REPUBLIQUE DU SENEGAL

UN PEUPLE UN BUT UNE FOI



MINISTERE DE L'ENSEIGNEMENT SUPERIEUR, DE LA RECHERCHE ET DE l'INNOVATION

UNIVERSITE GASTON BERGER DE SAINT-LOUIS



U.F.R SCIENCES APPLIQUÉES ET DE TECHNOLOGIE

Département : Physique Appliquée

Option : Électronique et Télécommunications

Mémoire de Fin d'étude pour l'obtention du

Diplôme de Master en Ingénierie
Électronique et Télécommunications

Sujet :

PROTOCOLE DE CONFIDENTIALITÉ ET D'INTÉGRITÉ DANS L'IOT BASÉ SUR LA TECHNOLOGIE BLOCKCHAIN

Présenté par : ***Encadrant :***

MOUSSA NDIAYE **Pr. Chérif DIALLO**

Soutenu le 11 Décembre 2024, devant le jury :

Pr. ABDOU KARIM	DIALLO	Professeur Titulaire	Président
Pr. MOUSTAPHA	SENE	Professeur Assimilé	Examinateur
Dr. GABRIEL JEAN	P. TEVY	Docteur	Examinateur
Pr. CHERIF	DIALLO	Professeur Titulaire	Encadrant

Année Académique : 2022/2023

DEDICACES

Louanges à Dieu, le miséricordieux,sans Lui, rien de tout cela n'aurait été possible. Je dédie ce précieux travail aux êtres les plus chers à mon très cher père **Abdoulaye Ndiaye** pour son soutien indéfectible, et ma très chère maman **Awa Ndiaye** qui n'a jamais cessé de m'encourager dans les moments les plus difficiles, ainsi que pour leurs sacrifices incomparables.

Je remercie également mon frère **Ousmane Ndiaye**, qui m'a aidé financièrement tout au long de mon parcours universitaire.

À toutes mes sœurs **(Ndeye Abi, Khadidiatou et Marietou)**, frères **(Mame Ndiaye, Pape Ndiaye, Ibrahima et Ousmane)**, famille et amis. Les mots ne suffisent pas à exprimer l'attachement, l'amour et l'affection que je ressens pour vous.

Je n'oublie pas non plus tous mes enseignants qui m'ont guidé depuis mon enfance, et à tous ceux que j'aime et qui m'aiment en retour.

REMERCIEMENT

Je remercie Allah, le Tout-Puissant et le miséricordieux, pour ses innombrables bénédictions et pour la guidance constante qu'Il m'a offerte durant tout mon parcours académique. Sa bienveillance et Sa grâce ont été les piliers qui m'ont soutenu à chaque étape, m'inspirant force et persévérance pour franchir chaque défi rencontré. Un remerciement particulier et sincère à mes parents pour les efforts qu'ils ont fournis. Ils ont toujours été présents et ce travail est un témoignage de ma gratitude et de mon profond respect envers eux.

Je tiens à exprimer toute ma reconnaissance à mon encadrant, **Pr. Chérif Diallo**, Professeur titulaire de l'Université Gaston Berger pour son expertise, sa patience et son accompagnement précieux durant cette recherche. Ses encouragements et son soutien ont été déterminants pour la réussite de ce travail. Je tiens à exprimer toute ma reconnaissance aux professeurs et enseignants chercheurs de l'**UFR SAT**. Leur disponibilité et leur engagement constant ont joué un rôle fondamental dans mon parcours académique, contribuant profondément à mon épanouissement intellectuel et à mes réussites universitaires.

Je remercie chaleureusement chaque membre de ma famille **(mes parents, mes tantes, mes frères et mes sœurs)** pour l'amour inconditionnel, le soutien indéfectible et les souvenirs inestimables que vous m'avez donnés tout au long de mon parcours. Votre présence et vos encouragements ont été des piliers essentiels pour mon développement et mon épanouissement.

Je tiens à remercier mon team **Zéro Franc Squad** depuis la licence 1 on est ensemble ; c'est grâce a eux j'ai su que la famille n'est pas forcement biologique. Je remercie aussi mon groupe de projet **(Souleymane Bore, Fatou Diop et Ibrahima Diaby)** avec eux on faisait ensemble tous les projets.

Je souhaite exprimer toute ma reconnaissance à mes camarades de promotion **P27 (MPI et MASS)** ainsi qu'à toute la **classe MIETEL** pour les moments d'apprentissage et les expériences enrichissantes que nous avons partagés ensemble au cours de cette formation.

Je tiens à exprimer un remerciement tout particulier à ma meilleure , ma confidente et ma partenaire, **Ndeye Fatou Fall**. Sa présence a été d'une grande aide durant ma dernière année d'études, et je suis vraiment ravi de l'avoir à mes côtés.

RESUME

L'Internet des Objets, ou Internet of Things (IoT) en anglais, désigne un réseau de dispositifs physiques connectés à Internet capables de collecter, d'échanger et de transmettre des données en temps réel. Ces objets connectés incluent une large gamme d'appareils, tels que des capteurs, des systèmes domestiques intelligents, des équipements industriels, et bien d'autres. Cependant, cette interconnectivité accrue expose les dispositifs IoT à des risques de sécurité, notamment en matière de confidentialité et d'intégrité des données. Parmi les vulnérabilités majeures, on trouve les risques de piratage, la manipulation des données et les failles de confidentialité.

La technologie Blockchain, de plus en plus envisagée pour sécuriser l'IoT, offre un cadre de protection en garantissant l'intégrité et la confidentialité des données échangées entre les dispositifs. Ce mémoire explore précisément les contributions de la Blockchain dans le renforcement de la sécurité de l'IoT, en partant des principes fondamentaux de cette technologie et en examinant les divers domaines d'application de l'IoT et ses principales vulnérabilités.

Dans ce contexte, un protocole de sécurité inspiré de la Blockchain est proposé, utilisant des mécanismes de vérification et de signature numérique pour sécuriser les échanges au sein des réseaux IoT. Ce protocole assure l'authenticité des échanges et limite l'accès aux données sensibles aux seuls dispositifs autorisés. En terme, cette étude montre que l'intégration de la Blockchain dans les systèmes IoT constitue une solution efficace pour renforcer la sécurité de ces environnements interconnectés, offrant une meilleure résistance face aux défis de confidentialité et d'intégrité propres aux réseaux IoT.

Mots clef : Sécurité, Confidentialité, Intégrité, Authentification, Internet des objets, Protocole, Blockchain

ABSTRACT

The Internet of Things (IoT) refers to a network of physical devices connected to the Internet that are capable of collecting, exchanging, and transmitting data in real-time. These connected objects encompass a wide range of devices, such as sensors, smart home systems, industrial equipment, and many more. However, this increased interconnectivity exposes IoT devices to security risks, particularly concerning data privacy and integrity. Major vulnerabilities include the risks of hacking, data manipulation, and privacy breaches.

Blockchain technology, increasingly considered for securing the IoT, provides a protective framework by ensuring the integrity and confidentiality of data exchanged between devices. This thesis specifically explores the contributions of Blockchain in enhancing the security of IoT, starting from the fundamental principles of this technology and examining the various application domains of IoT along with its main vulnerabilities.

In this context, a security protocol inspired by Blockchain is proposed, utilizing verification and digital signature mechanisms to secure exchanges within IoT networks. This protocol ensures the authenticity of exchanges and restricts access to sensitive data to authorized devices only.

In term, this study demonstrates that the integration of Blockchain into IoT systems represents an effective solution for strengthening the security of these interconnected environments, providing better resilience against the privacy and integrity challenges inherent to IoT networks.

Keywords : Security, privacy, Integrity, Authentication, Internet of Things, Protocol, Blockchain

Liste des abreviations

AES	Advanced Encryption Standard
API	Application Programming Interface
BCT	BlockChain Technology
CP-ABE	Ciphertext Policy Attribute Based Encryption
DES	Data Encryption Standard
DLT	Distributed Layer Technology
DDoS	Distributed Denial of Service
DoS	Denial of Service
AES	Advanced Encryption Standard
DPoS	Delegue Proof of Stake
ECC	Elliptic Curve Cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
IBM	International Bussiness Machines
IDEA	International Data Encryption Algorithme
IdO	Internet des Objets
IIOT	Industriel Internet of Things
IoT	Internet of things
IT	Informations Technology
LoRa	Long Rang
LPWAN	Low Power Wide Area Network
LTEM	LTEM Long Term Evolution for Machine
M2M	Machine to Machine
MD5	MD5 Message-Digest algorithm 5
NFC	Near Field Communication

NIST	National Institute of Standards and Technology
PBFT	PBFT Pratical Byzantine Fault Tolerance
P2P	Peer to Peer
PGP	Pretty Good Privacy
PK	Public Key
POA	Proof of Autority
POET	Proof of Elased Time
POS	Proof of State
POV	PoW Proof of Work
RC	Ron's Cipher
RFID	Radio Frequency IDentification
RSA	Rivest, Shamir and Adleman
SHA	Secure Hash Algorithm
STI	Système de Transport Intelligente
TLS	Transport Layer Security
V2X	Vehicle to Everything
WEP	Wired Equivalent Privacy
WPA	Wi-Fi Protected Access
WSN	Wireless Sensor Networks

TABLE DES MATIÈRES

DEDICACES i

REMERCIEMENT ii

RESUME iv

ABSTRACT v

Liste des abreviations vi

LISTE DES FIGURES x

INTRODUCTION GÉNÉRALE 1

1 Généralité sur l'internet des Objets 3

1.1 Introduction . . . 3
1.2 Définition de l'Internet des Objets . . . 4
1.3 Origine de l'Internet des Objets . . . 5
1.4 Architecture de l'Internet des Objets . . . 6
1.5 Les domaines d'applications de l'Internet des Objets . . . 10
1.6 Les enjeux de l'Internet des Objets . . . 13
1.7 Les attaques de l'IdO . . . 17
1.8 Conclusion . . . 18

2 La Technologie Blockchain 20

2.1 Introduction . . . 20
2.2 Historique . . . 21
2.3 Définition . . . 21
2.4 Propriétés de la blockchain . . . 23
2.5 Structure de la blockchain . . . 24
2.6 Architecture de la blockchain . . . 26
2.7 Les différents types de blockchain . . . 27
2.8 Blockchain Publique . . . 28

2.9 Blockchain Privée 29
2.10 Blockchain hybride 29
2.11 Blockchain consortium 30
2.12 Mécanisme de la blockchain 31
2.13 Plateformes de la blockchain 32
2.14 Conclusion 41

3 Etablissement de Sessions IoT Sécurisées par la Blockchain 42
3.1 Introduction 42
3.2 Problèmes de confidentialité 43
3.3 Problèmes d'intégrité des données 43
3.4 Solution pour améliorer la confidentialité et l'intégrité dans l'IoT 44
3.5 Objectif proposé 45
3.6 Conclusion 60

CONCLUSION GÉNÉRALE 61

LISTE DES FIGURES

1.1 : Internet des Objets 5
1.2 Architecture de l'IoT 7
1.3 Technologies de communication dans l'IoT 8
1.4 Domaines d'application de l'Internet des objets 11

2.1 Exemple simplifié d'une blockchain 22
2.2 Structure de la blockchain 25
2.3 Réseau Décentralisé vs Centralisé 26
2.4 Différents types de blockchain 27
2.5 Mécanisme de la blockchain 32
2.6 Cryptographie symétrique et Cryptographie asymétrique 38
2.7 Tableau comparatif des tailles de clé (en bits) par niveau de sécurité 39
2.8 Signature numérique et vérification 40
2.9 Principe de hachage 41

3.1 Architecture du système décentralisée 46
3.2 Création des blocs d'objets 52
3.3 Générée une paire de clé ECC en python 54
3.4 Chiffrement et Déchiffrement du message 55
3.5 Authentification des nœuds voisins 57
3.6 Signature numérique 58
3.7 Vérification de hachage 59
3.8 Message modifié 60

INTRODUCTION GÉNÉRALE

L'Internet des Objets (**IdO**), ou « Internet of Things » (IoT) en anglais, est un concept introduit par **Kevin Ashton** en 1999 pour décrire la connexion entre le réseau Internet et le monde physique grâce à des capteurs intelligents. Depuis, l'IdO a évolué pour englober divers dispositifs embarqués dans des environnements informatiques qui détectent, communiquent et fournissent des services aux utilisateurs, qu'ils soient humains ou automatisés. Ces dispositifs, intégrés dans des objets quotidiens ou industriels, permettent ainsi d'interagir avec leur environnement et de transmettre des données en temps réel, favorisant des applications dans des domaines variés tels que la domotique, la santé, et les infrastructures urbaines intelligentes. L'Internet des Objets (IoT) a révolutionné de nombreux secteurs en offrant des solutions interconnectées et intelligentes, intégrant des dispositifs variés allant des capteurs industriels aux appareils domestiques. Cette interconnexion massive améliore l'efficacité, optimise les processus et facilite la collecte de données en temps réel. Cependant, en raison de la nature distribuée et souvent peu sécurisée de ces dispositifs, l'IoT est exposé à des risques accrus en matière de sécurité, notamment au niveau de la confidentialité et de l'intégrité des données transmises.

La Blockchain, avec ses propriétés de décentralisation, d'immutabilité et de transparence, apparaît comme une solution potentielle pour sécuriser l'IoT. En intégrant des mécanismes de vérification et de signature numérique, elle pourrait renforcer la fiabilité des échanges entre dispositifs IoT, en s'assurant que les données transmises ne sont ni modifiées ni falsifiées, et que seuls les dispositifs autorisés peuvent y participer.

Dans ce contexte, ce mémoire propose un protocole inspiré de la Blockchain visant à améliorer la confidentialité et l'intégrité des communications dans les systèmes IoT. Notre travail est

structuré sur trois chapitres.

Dans le premier chapitre, nous introduirons l'Internet des Objets (IoT) en abordant son architecture, ses domaines d'application, les défis, les vulnérabilités et les services de sécurité.

Le deuxième chapitre nous explorerons la technologie Blockchain en présentant ses principes généraux, son architecture, son fonctionnement, ses domaines d'application, ainsi que ses mécanismes de sécurité.

Enfin, le dernier chapitre, nous allons parler sur établissement de sessions IoT sécurisées par la blockchain.

Chapitre

1 Généralité sur l'internet des Objets

Sommaire

1.1 Introduction 3
1.2 Définition de l'Internet des Objets 4
1.3 Origine de l'Internet des Objets 5
1.4 Architecture de l'Internet des Objets 6
1.4.1 Couche de détection 7
1.4.2 Couche réseau 7
1.4.3 Couche service 9
1.4.4 Couche Application 10
1.5 Les domaines d'applications de l'Internet des Objets 10
1.5.1 Santé 11
1.5.2 La domotique 11
1.5.3 Ville Intelligente 12
1.5.4 Transport intelligent 12
1.5.5 L'agriculture 13
1.6 Les enjeux de l'Internet des Objets 13
1.6.1 Sécurité 13
1.6.2 Hétérogénéité 14
1.6.3 Sensibilité au contexte 14
1.6.4 Sureté dans l'Internet des Objets 14
1.6.5 Avantages et Défis de l'IoT 15
1.7 Les attaques de l'IdO 17
1.8 Conclusion 18

1.1 Introduction

La prochaine génération d'Internet est l'Internet des objets. L'Internet des objets combine des objets dotés d'une connectivité internet de puissances capacités de collecte et d'analyse de données. L'Internet des objets est un monde intelligent potentiellement bénéfique, révolutionnaire

et entièrement interconnectée, porteurs de progrès, d'efficacité et d'opportunités.

L'Internet des objets est un écosystème qui émerge rapidement et qui présente un potentiel énorme pour offrir des avantages significatifs tels que la rentabilité, l'amélioration de l'utilisation des actifs, l'amélioration de l'efficacité des processus et l'augmentation de la productivité. Les produits liés à l'internet des objets utilisent une technologie intégrée pour communiquer, directement ou indirectement, entre eux ou avec l'internet [1]. L'Internet des objets est une convergence d'appareils traditionnels connectes et communiquant de manière intelligente, tels que les ordinateurs de bureau et les ordinateurs portables, et d'objets quotidiens de nature diverse avec une technologie intégrée, pour devenir un grand système d'information. La croissance explosive des appareils et applications mobiles et la facilite d'accès à la connectivité sans fil sont les moteurs de l'émergence de l'Internet des objets.

Dans ce chapitre, nous allons présenter l'Internet des objets (IdO) en abordant son historique et son architecture à travers ses différentes couches. Nous examinerons également ses principaux domaines d'application et ses utilisateurs les plus connus. Enfin, nous discuterons de ses avantages, de ses limites, des défis et sécurités auxquels il fait face.

1.2 Définition de l'Internet des Objets

L'Internet des Objets (IdO), ou Internet of Things (IoT) en anglais, est un concept visant à relier le monde virtuel de l'Internet au monde réel à travers des objets intelligents. Ces objets possèdent des capacités de communication et d'échange de données via Internet [2].
Selon l'Union Internationale des Télécommunications (UIT), l'IdO est défini comme « une infrastructure mondiale pour la société de l'information, permettant d'offrir des services évolués en interconnectant des objets physiques et virtuels grâce aux technologies de l'information et de la communication, qu'elles soient existantes ou en évolution ». En d'autres termes, l'IdO utilise les capacités de détection, de traitement de l'information, d'action et de communication des entités autonomes pour créer de nouveaux services et applications. Par exemple, il permet

l'interconnexion de multiples capteurs de température, d'humidité, de pression, etc. [3].
Conceptuellement, l'Internet des Objets décrit des objets physiques connectés dotés d'une identité numérique propre et capables de communiquer entre eux, formant ainsi une passerelle entre le monde physique et le monde virtuel. Techniquement, l'IoT repose sur l'identification numérique directe et standardisée (comme une adresse IP, les protocoles SMTP, HTTP, etc.) d'un objet physique via un système de communication sans fil, tel qu'une puce RFID (Radio Frequency Identification), le Bluetooth ou le Wi-Fi [7].
Ces objets connectés constituent collectivement la base de l'Internet des Objets, permettant aux personnes et aux objets d'être connectés en tout temps et en tout lieu.
La figure 1.1 ci-dessous illustre ce concept.



FIGURE 1.1 – : Internet des Objets

1.3 Origine de l'Internet des Objets

En 1999, Kevin Ashton, co-fondateur de l'Auto-ID Center au MIT (Massachusetts Institute of Technologie), a évoqué pour la première fois le concept de l'Internet des Objets lors d'une présentation chez Procter & Gamble. Son objectif était de sensibiliser les directeurs de P&G aux puces RFID (identification par radiofréquence). Cependant, pour capter l'intérêt de son auditoire en intégrant la tendance technologique de l'année, il a intitulé sa présentation « Internet

des objets ». L'idée d'appareils connectés, cependant, remonte aux années 1970.

Grâce au Web, les développeurs pouvaient surveiller l'état des machines et vérifier si une boisson froide était disponible. L'IoT a ensuite progressé avec l'essor des technologies sans fil (comme le Wifi), des MEMS (systèmes micro-électromécaniques), des micro services et de l'Internet. Cette évolution a rapproché la technologie opérationnelle (OT) et la technologie de l'information (IT), permettant ainsi l'analyse de données non structurées générées par les machines pour identifier des axes d'amélioration.

Le fondement de l'Internet des Objets repose sur la connectivité M2M (Machine to Machine), où les machines communiquent entre elles via un réseau sans intervention humaine. L'IoT représente ainsi un réseau de milliards de nœuds (capteurs), reliant des personnes, des systèmes et diverses applications pour collecter et partager des données. Toutefois, le concept d'écosystème de l'Internet des objets ne s'est concrétisé qu'au milieu des années 2010. Cette avancée est en partie due au gouvernement chinois, qui a fait de l'IoT une priorité stratégique, notamment dans le cadre de sa stratégie de reconnaissance faciale et de surveillance de la population [6].

1.4 Architecture de l'Internet des Objets

Une architecture IoT se compose de quatre couches : la couche de détection, la couche réseau, la couche service et la couche application. Ce découpage est particulièrement pertinent car il couvre tous les aspects liés aux problèmes de sécurité dans l'IoT. D'un point de vue technique, une architecture IoT intègre une grande diversité de dispositifs et de technologies de communication réseau. La figure 1.2 illustre cette organisation en couches ainsi que les différents éléments présents dans chacune d'elles.

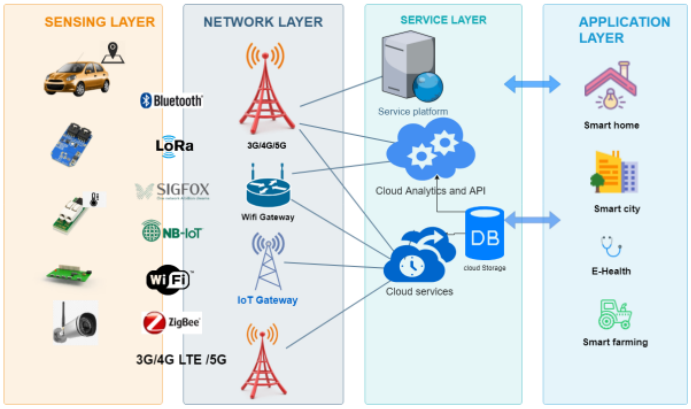


FIGURE 1.2 – Architecture de l'IoT

1.4.1 Couche de détection

Les objets peuvent être classés en deux catégories : les objets contraints et les objets non contraints. Les objets contraints possèdent de faibles capacités de calcul, de stockage et d'énergie, et sont également limités par leur taille. Parmi les exemples d'objets contraints, on trouve les sondes de température, les Arduino Nano, et les bracelets connectés (smart bands). Il est important de noter que, pour des raisons de gestion énergétique, ces objets ne sont actifs que pendant le temps de communication [8].

Les objets non contraints, en revanche, disposent de capacités supérieures à celles des objets contraints et ne sont pas limités par leur taille. Ces objets peuvent être directement connectés à Internet car ils possèdent les capacités nécessaires pour supporter plusieurs technologies de communication. Parmi les exemples d'objets non contraints, on peut citer les montres intelligentes (smart watches) et les smartphones.

1.4.2 Couche réseau

La couche réseau, ou "network layer" en anglais, située au-dessus de la couche de détection, gère toutes les activités réseau d'une infrastructure IoT. Cela inclut la connectivité, le routage, l'interconnexion, l'acheminement des paquets, la qualité de service, la gestion de l'adressage et l'adaptation des protocoles, etc. Cette couche héberge des passerelles (Gateway) qui servent de point de passage vers Internet pour les objets contraints, en assurant l'adaptation des protocoles

et le routage des données.

Elle est également responsable du transport des données provenant de la couche de détection vers les services et applications qui consomment ces données. Une variété de technologies de communication réseau est utilisée dans cette couche. Ces technologies peuvent être divisées en trois catégories : les réseaux étendus à faible puissance (LPWAN : Low Power Wide Area Networks), les réseaux étendus (WAN : Wide Area Networks) et les réseaux locaux/personnels (LAN : Local Area Networks et WPAN : Wireless Personal Area Networks) [8].

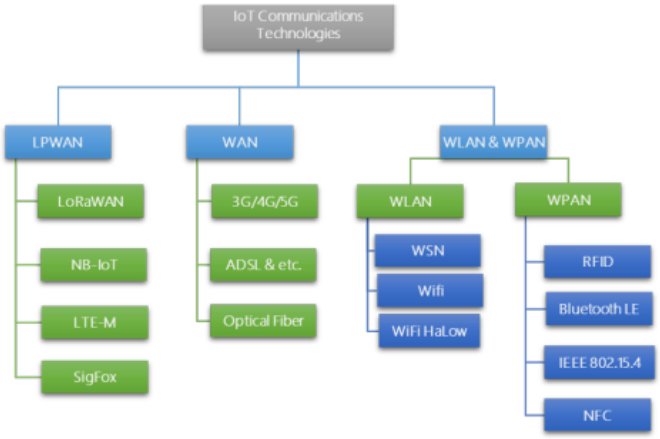


FIGURE 1.3 – Technologies de communication dans l'IoT

La figure 1.3 illustre la répartition de ces différentes technologies de communication dans les différentes catégories.

Dans la catégorie des réseaux étendus à faible puissance, on trouve :

LoRa WAN (Long Range Wide Area Networks) de l'alliance LoRa ;

Sigfox de l'entreprise SigFox ;

NB-IoT (Narrowband Internet of Things) du 3GPP ;

LTE-M (Long Term Evolution – M) du 3GPP.

Dans la catégorie des réseaux étendus, on trouve :

Les réseaux mobiles 3G/4G/5G ;

Les réseaux haut débit filaires (ADSL, Fibre optique).

Dans la troisième et dernière catégorie, on trouve :

Les technologies de réseaux de capteurs (WSN : Wireless Sensor Network) ;

Les technologies de communication paire à paire à courte portée, telles que :

BLE : Bluetooth Low Energy.

IEEE 802.15.4 ;

Les technologies de communication à très courte portée :

RFID ;

NFC (Near Field Communication) ;

Les technologies de communication à moyenne portée en réseau local :

WiFi HaLow IEEE 802.11ah (Low Power WiFi) ;

Les variantes des normes sans fil WiFi.

Les protocoles réseau mis en œuvre dans cette couche incluent notamment le protocole 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) standardisé par l'IETF.

1.4.3 Couche service

La couche service, ou "service layer", située au-dessus de la couche réseau, fournit et gère les services nécessaires pour les utilisateurs et les applications de l'architecture. Elle héberge les services de stockage, de traitement et d'analyse des données collectées par la couche de détection. Elle est également responsable de la mise en œuvre des logiques métiers des applications, ce qui implique l'extraction des informations pertinentes à partir de l'énorme quantité de données recueillies par les objets. À ce niveau, on trouve généralement des services d'analyse de données Big Data, des services d'apprentissage automatique et des services d'interfaces de programmation d'application (API : Application Programming Interface).

Cette couche inclut également des services de support opérationnel, de configuration et de mise à jour des dispositifs, de reporting et de facturation. Elle héberge également les services de gestion de la qualité de service ainsi que la plupart des opérations de sécurité (même si la sécurité est transversale à toutes les couches), telles que le contrôle d'accès, la gestion des identifications et la gestion des processus métiers [8].

1.4.4 Couche Application

La couche supérieure de l'architecture de l'IoT, perceptible par l'utilisateur final, a pour objectif de gérer et de fournir des applications en se fondant sur les données capturées par la couche de perception et traitées par l'unité de traitement de l'information. Elle permet aux utilisateurs finaux d'accéder à des services personnalisés sur le réseau, adaptés à leurs besoins, grâce à l'utilisation de divers appareils portables et terminaux. Cette couche se subdivise en deux sous-couches : la couche de support aux applications et les applications IoT [10].
La couche de support aux applications, située juste au-dessus de la couche de transmission, assure le soutien à une gamme de services et prend en charge l'exécution de calculs intelligents ainsi que le traitement ou la fusion des données. Elle opère la reconnaissance et le filtrage des données afin de les classer en données valides, invalides, malveillantes, ou indésirables, selon les besoins des différents services. Cette couche peut adopter différentes configurations en fonction des services offerts, et tire parti des fonctionnalités du middleware, chargé des calculs intelligents.
Les applications IoT peuvent être classées en trois catégories selon leurs fonctionnalités : la collecte d'informations, l'analyse et les prises de décision. Les applications collectrices d'informations sont responsables de recueillir les données provenant des nœuds de perception et de les stocker localement. L'analyse implique le prétraitement hors ligne des données collectées afin de créer un modèle générique pour évaluer les données futures. Les applications décisionnelles sont ensuite utilisées pour prendre les mesures appropriées en fonction des analyses réalisées.

1.5 Les domaines d'applications de l'Internet des Objets

Les avantages offerts par l'Internet des objets couvrent un large éventail d'applications, permettant d'améliorer la vie des personnes dans divers contextes : à la maison, au travail, en déplacement, à l'hôpital, etc. La figure 1.4 illustre quelques-uns des principaux domaines d'application de l'IoT.



FIGURE 1.4 – Domaines d'application de l'Internet des objets

1.5.1 Santé

Les applications de l'IoT sont particulièrement bénéfiques pour le suivi de l'état de santé d'une personne. Elles intègrent un certain nombre d'objets comprenant divers types de capteurs [5]. Dans le domaine de la santé, l'IoT permettra le déploiement de réseaux personnels pour le contrôle et le suivi des signes cliniques, notamment pour les personnes âgées. Les objets connectés permettront de surveiller la tension artérielle, le rythme cardiaque, la qualité de la respiration, ou encore la masse graisseuse. Ceci facilitera la télésurveillance des patients à domicile et apportera des solutions pour l'autonomie des personnes à mobilité réduite [23].

1.5.2 La domotique

La domotique, ou "maison connectée", est considérée comme l'apogée de l'Internet des objets. Ce concept, connu sous le nom de "science de la maison", se définit comme l'ensemble des appareils connectés utilisés pour créer un bâtiment intelligent. Grâce à des plateformes dédiées appelées box domotiques, ces dispositifs peuvent interagir entre eux et être contrôlés de manière centralisée via une application unique. Ainsi, l'ensemble de la maison devient connecté et contrôlable depuis un smartphone, qu'il s'agisse de la gestion des lumières ou de l'énergie [1].

De manière générale, les technologies domotiques offrent trois utilités principales :

Le confort : en améliorant la qualité de vie et en automatisant certaines tâches. Les économies

d'énergie : en installant des modules qui surveillent les dépenses des utilisateurs et fournissent des conseils personnalisés. La sécurité : grâce à des systèmes d'alarme centralisés et des scénarios de sécurité.
La domotique couvre de nombreux secteurs et permet diverses fonctionnalités, parmi lesquelles

— La création d'automatismes ;

— Le pilotage à distance ;

— Le contrôle de l'éclairage ;

— La gestion du chauffage ;

— La centralisation de la sécurité.

1.5.3 Ville Intelligente

Les technologies de l'information et de la communication peuvent aussi contribuer à améliorer la qualité de vie des citoyens en offrant des services tels que la télésanté, l'enseignement en ligne et la sécurité publique intelligente. De plus, elles jouent un rôle crucial dans la promotion d'une économie plus durable en optimisant l'utilisation des ressources naturelles et en soutenant l'émergence de nouvelles industries écologiques.

1.5.4 Transport intelligent

Le transport intelligent repose sur trois conceptions principales : l'analyse des transports, le contrôle des véhicules connectés et la gestion du trafic. L'analyse des transports comprend la prédiction de la demande et la détection d'anomalies. Le routage des véhicules, le contrôle de la vitesse et la gestion du trafic sont regroupés sous le terme de contrôle du transport, qui est étroitement lié aux véhicules connectés via la communication V2X (Véhicule-to-Everything). Globalement, ces systèmes sont régis par une diffusion multi-technologie

.

1.5.5 L'agriculture

Dans ce domaine, des réseaux de capteurs connectés à l'IoT peuvent être utilisés pour surveiller l'environnement des cultures. Cela permettra une meilleure prise de décision en agriculture, notamment en optimisant l'irrigation, l'utilisation des intrants et la planification des travaux agricoles. De plus, ces réseaux peuvent aider à combattre la pollution de l'air, du sol et de l'eau, contribuant ainsi à améliorer la qualité de l'environnement en général

.

1.6 Les enjeux de l'Internet des Objets

De nombreuses recherches cruciales, dans diverses directions, sont nécessaires pour une adoption à grande échelle de l'IoT. Dans cette section, nous abordons les principaux enjeux liés au concept de l'IoT.

1.6.1 Sécurité

Un problème fondamental persistant dans l'Internet actuel, et qui doit être résolu, est celui de la sécurité. Les attaques constituent un défi majeur pour l'IoT en raison de la capacité limitée des "objets" (dispositifs) utilisés et de l'accessibilité physique aux capteurs [6]. Ce problème est aggravé par le fait que les défaillances aléatoires des objets créent des vulnérabilités exploitables par les attaquants. Pour répondre aux exigences réalistes du système, impliquant un fonctionnement de longue durée et sans surveillance, les applications IoT doivent pouvoir continuer à fonctionner de manière satisfaisante en présence d'attaques et se rétablir efficacement après celles-ci. Les solutions peuvent nécessiter le téléchargement d'un nouveau code, ce qui est également vulnérable aux attaques. De plus, le système doit être capable de s'adapter à de nouvelles attaques imprévues lors de son déploiement initial.

1.6.2 Hétérogénéité

L'IoT se caractérise par des objets et des systèmes reposant sur des configurations matérielles, logicielles et réseaux très variés. Ces objets peuvent interagir entre eux ou avec différentes plateformes de services à travers des réseaux hétérogènes. Les applications et systèmes à mettre en œuvre devront donc prendre en compte cette hétérogénéité de l'IoT

.

1.6.3 Sensibilité au contexte

L'IoT repose principalement sur la collecte de données des capteurs des objets connectés. Cette collecte reflète les changements dynamiques qui surviennent dans l'environnement des objets. Ces derniers doivent aussi s'adapter aux changements de contexte (comme un déplacement géographique), au nombre d'objets présents (ajout ou retrait d'objets en temps réel), et à la configuration réseau (par exemple, les changements fréquents de réseau d'accès, comme le Wifi ou la 4G, et d'adresse IP). L'état des objets évolue également de manière dynamique, avec des périodes d'activité et de connectivité intermittentes, passant de l'état de veille à l'état actif, ou encore de l'état connecté à l'état déconnecté.

1.6.4 Sureté dans l'Internet des Objets

Les périphériques IdO sont déployés par différentes personnes / autorités / entités. Ces dispositifs ont des conditions de fonctionnement différentes, des fonctionnalités, des résolutions, etc. Ainsi, permettre une intégration transparente de ces appareils est un énorme défi. Le degré de complexité augmente de nombreux plis lorsque certains de ces simples dispositifs sont fusionnés pour former un réseau complexe

.

1.6.5 Avantages et Défis de l'IoT

Tout d'abord, l'IoT peut améliorer l'efficacité opérationnelle et réduire les coûts grâce à la surveillance et au contrôle à distance des équipements. Cependant, cette capacité peut également augmenter les risques pour la sécurité et la confidentialité des données sensibles, ce qui exige une attention particulière pour protéger les informations personnelles et professionnelles. De plus, la complexité croissante des systèmes IoT pose des défis supplémentaires en matière de gestion et de maintenance [8].

1.6.5.1 Avantages

Automatisation : Dans les industries manufacturières, l'automatisation peut s'avérer particulièrement bénéfique en permettant aux machines d'accomplir des tâches répétitives et monotones. Cela libère les travailleurs pour qu'ils puissent se concentrer sur des tâches plus complexes et créatives. De plus, l'automatisation améliore la précision et la qualité des produits finis.

Optimisation : Les entreprises peuvent tirer parti de la technologie pour améliorer leur communication interne et externe. Elle permet également de mieux comprendre les besoins des clients et de leur offrir des solutions plus personnalisées. En somme, l'utilisation de la technologie peut avoir un impact positif sur tous les aspects d'une entreprise, allant de la gestion des opérations à la satisfaction des clients.

Prévention : L'IoT offre la possibilité de surveiller les équipements en temps réel et de prédire les besoins en maintenance, réduisant ainsi les coûts et augmentant la productivité. En intégrant l'IoT dans le secteur industriel, de nombreux avantages peuvent être obtenus en termes de sécurité, d'efficacité et de rentabilité.

Personnalisation : La personnalisation peut améliorer la satisfaction des clients et augmenter leur fidélité, ce qui impacte positivement les résultats financiers de l'entreprise. Toutefois, il est

crucial de respecter la vie privée des clients et de veiller à ce que les données soient collectées et utilisées de manière éthique et transparente.

Innovation : La technologie IoT permet de créer des produits et services innovants qui répondent aux besoins des consommateurs. En outre, la concurrence pousse les entreprises à améliorer leur efficacité opérationnelle et à réduire leurs coûts, ce qui peut se traduire par des économies pour les clients.

1.6.5.2 Défis

Confidentialité : Les entreprises doivent être transparentes sur la manière dont elles collectent et utilisent les données, ainsi pour protéger ces informations contre les violations de sécurité et les abus. Il est également crucial que les utilisateurs soient informés de leurs droits en matière de protection des données et qu'ils aient la possibilité de contrôler l'utilisation de leurs informations personnelles.

Coût : Cependant, les avantages potentiels de l'IoT en termes de productivité, d'efficacité et de rentabilité pour une entreprise peuvent être considérables. Il est donc essentiel d'évaluer soigneusement le coût initial par rapport aux avantages à long terme avant de prendre une décision d'investissement dans un tel système.

Complexité : L'intégration de l'IoT peut entraîner une complexité accrue dans les environnements informatiques modernes, avec des problèmes potentiels de compatibilité et de sécurité. Cela nécessite une gestion des ressources efficace et une expertise technique spécialisée pour maintenir un réseau stable et sécurisé. De plus, les mises à jour fréquentes des logiciels et des équipements peuvent également compliquer la gestion du réseau.

Dépendance : En outre, une dépendance excessive à la technologie peut avoir des conséquences néfastes sur la santé mentale et physique, telles que la fatigue oculaire, l'isolement social et le

stress. Il est donc essentiel de trouver un équilibre entre l'utilisation de la technologie et les activités hors ligne pour préserver un mode de vie sain et équilibré.

1.7 Les attaques de l'IdO

(i) Attaques contre les objets : Ces attaques nécessitent un accès physique aux objets et sont généralement difficiles à réaliser.

— Attaque contre la disponibilité (capture physique) : Certains objets sont installés dans des endroits non protégés. Un attaquant peut rendre un objet inaccessible en le capturant ou en le détruisant physiquement. Ces attaques, qui compromettent la disponibilité des objets, sont également appelées attaques contre l'intégrité physique [2].

— Accès non autorisé et extraction de données sensibles : Un attaquant peut accéder physiquement à un objet non protégé et en extraire des informations sensibles stockées. Les objets peuvent être exposés à des attaques par canal latéral (side channel attacks). Parmi ces attaques, on trouve :

- Analyse de la consommation électrique : Un attaquant cherche à retrouver des informations sensibles (comme une clé privée) à partir de la consommation de courant lors des opérations cryptographiques.

- Injection de faute : Cette attaque consiste à perturber l'exécution d'un algorithme pour extraire des informations, en utilisant des moyens comme la surchauffe du processeur ou l'émission de laser.

— Micrologiciel vulnérable : Les objets connectés fonctionnent grâce à des micrologiciels (firmwares). Un micrologiciel est un programme intégré dans la mémoire de l'objet pour assurer son bon fonctionnement. Certains micrologiciels peuvent contenir des bugs ou des failles de sécurité. Ces vulnérabilités exposent les objets à des menaces telles que l'attaque par débordement de mémoire tampon. L'application de correctifs via des mises à jour OTA (On The Air) par des canaux non sécurisés augmente les risques de clonage. Un attaquant peut capturer le firmware, cloner un objet, ajouter des portes dérobées et remplacer un objet légitime par un objet cloné.

— Clonage et usurpation d'objet : Certains fabricants peu scrupuleux peuvent cloner les caractéristiques

d'objets authentiques et insérer des portes dérobées dans les objets clonés. Ces objets clonés fonctionnent normalement mais peuvent envoyer secrètement des informations à des tiers à des fins d'espionnage. Lors de l'installation, un attaquant peut se faire passer pour un installateur légitime, remplacer un objet authentique par un objet compromis ayant les mêmes caractéristiques. Cet objet compromis peut falsifier les informations de l'utilisateur et retransmettre des informations sensibles à l'attaquant.

(ii) Attaques visant les communications : Ces attaques peuvent être menées dès qu'un attaquant a une connectivité avec un ou plusieurs objets [2].

— Exécution de code malveillant : Les objets de la couche de détection sont vulnérables à l'exécution de codes malveillants, tels que des malwares, virus ou chevaux de Troie. Dès qu'un attaquant obtient un accès non autorisé à ces objets, il peut y exécuter ces codes malveillants. L'objectif principal de ces attaques est souvent de prendre le contrôle à distance des objets. Par exemple, en 2016, le botnet Mirai a été utilisé pour contrôler des milliers d'objets IoT et lancer une attaque de déni de service distribué (DDoS).

— Déni de service : La faible quantité de mémoire et d'énergie des objets les rend particulièrement vulnérables aux attaques de déni de service. Ces attaques compromettent la disponibilité des objets en rendant leurs ressources inutilisables pour les utilisateurs légitimes. Un attaquant peut continuellement envoyer un grand nombre de messages à un objet, épuisant ainsi ses ressources (drainage de la batterie, débordement de la mémoire, etc.).

— Interception des informations brutes : La transmission des données au niveau de la couche de détection est sujette aux attaques d'interception. Les données collectées et transmises sont souvent sensibles et concernent la vie privée des utilisateurs ou des infrastructures critiques. Un attaquant peut intercepter ces transmissions et en extraire des informations sensibles.

1.8 Conclusion

Dans ce chapitre, nous avons présenté une étude détaillée sur l'Internet des Objets (IoT), en commençant par sa définition et ses domaines d'application. Nous avons ensuite examiné les principaux composants de l'IoT, les vulnérabilités, les attaques ainsi que les services de

l'Internet des Objets et ses défis. Parmi ces défis majeurs, la tolérance aux fautes se distingue comme une technique essentielle pour garantir la sûreté de fonctionnement. Dans le chapitre suivant nous aborderons la technologie de la blockchain et son fonctionnement en matière de sécurité.

Chapitre

2 La Technologie Blockchain

Sommaire

2.1 Introduction 20
2.2 Historique 21
2.3 Définition 21
2.4 Propriétés de la blockchain 23
2.5 Structure de la blockchain 24
2.6 Architecture de la blockchain 26
2.7 Les différents types de blockchain 27
2.8 Blockchain Publique 28
2.9 Blockchain Privée 29
2.10 Blockchain hybride 29
2.11 Blockchain consortium 30
2.12 Mécanisme de la blockchain 31
2.13 Plateformes de la blockchain 32
2.13.1 Bitcoin 32
2.13.2 Ethereum 33
2.13.3 Hyperledger Fabric 33
2.13.4 IOTA 33
2.13.5 Procedure de consensus 34
2.13.6 Cryptographie Symétrique 36
2.13.7 Cryptographie Asymétrique 37
2.13.8 Signature Numérique 39
2.13.9 Fonction de hachage 40
2.14 Conclusion 41

2.1 Introduction

L'IoT permet la connexion de milliards de dispositifs physiques à Internet, facilitant la collecte, l'échange et l'analyse de données en temps réel. Cependant, cette interconnexion

massive pose des défis importants en matière de sécurité, de confidentialité et de gestion des données. La blockchain, avec ses caractéristiques de décentralisation, de transparence et d'immuabilité, offre des solutions potentielles pour relever ces défis. En les combinant, il est possible d'améliorer la sécurité et la fiabilité des systèmes IoT, d'assurer une meilleure gestion des identités et des autorisations, et de garantir l'intégrité des données collectées et échangées. Dans ce chapitre, nous allons expliquer la technologie de la blockchain ainsi que ses mécanismes de sécurité.

2.2 Historique

1991 : La technologie blockchain est initialement conceptualisée comme un système d'horodatage et de certification de documents par un collectif de chercheurs, Stuart Haber et W. Scott Stornetta.
1995 : Le New York Times met en place la première blockchain dans son journal. Cette technologie, qui est toujours active, est la plus longue blockchain de l'histoire.
1994-2005 : Nick Szabo développe les concepts de base des cryptomonnaies, notamment le concept de minage, et lance Bitgold, le précurseur de Bitcoin [11].
2009 : Satoshi Nakamoto, dont l'identité reste inconnue à ce jour (il pourrait s'agir de Nick Szabo, d'un collectif, ou d'un individu inconnu), publie un article qui pose les bases de Bitcoin.
2013-2015 : Vitalik Buterin développe Ethereum, qui est à la fois une monnaie cryptographique (la deuxième après Bitcoin) et une plateforme applicative distribuée.
Époque actuelle : La blockchain pourrait s'insérer comme élément d'un méta-système combinant les technologies de big data (cloud), la blockchain (intégrité référentielle) et l'intelligence artificielle (deep learning, machine learning), avec des projets comme Deep Brain Chain et Cortex Virtual Machine.

2.3 Définition

Une blockchain, ou chaîne de blocs, est une base de données distribuée (ledger) qui conserve un enregistrement permanent et infalsifiable des données transactionnelles. Elle est totalement décentralisée et fonctionne sur un réseau pair à pair (peer-to-peer). Chaque nœud du réseau

conserve une copie du ledger, éliminant ainsi le risque d'un point unique de défaillance. Toutes les copies sont mises à jour et validées simultanément.

Bien que la blockchain ait été initialement créée pour résoudre le problème de la double dépense en crypto-monnaie (monnaie virtuelle), elle peut être utilisée dans de nombreux autres cas. En effet, la blockchain est un moyen sécurisé de gestion et de protection de toutes sortes de données, qu'elles soient monétaires ou non [16].

Le ledger est composé d'un ensemble de blocs. Chaque bloc contient deux parties :

Le corps du bloc : Il contient les transactions, également appelées faits (facts), que la base de données doit enregistrer. Ces faits peuvent être des transactions monétaires, des données médicales, des informations industrielles, des logs systèmes, etc.

L'entête du bloc (header) : Il contient des informations concernant le bloc, telles que l'horodatage (timestamp), le hash des transactions, etc., ainsi que le hash du bloc précédent.

Cette structure fait que l'ensemble des blocs forme une chaîne de blocs liés et ordonnés. Plus la chaîne est longue, plus il est difficile de la falsifier. En effet, si un utilisateur malicieux veut modifier ou échanger une transaction sur un bloc, il doit :

— Modifier tous les blocs suivants, car ils sont liés par leurs hashes.

— Changer la version de la chaîne de blocs stockée par chaque nœud participant.

La figure suivante illustre un exemple simplifié d'une blockchain.

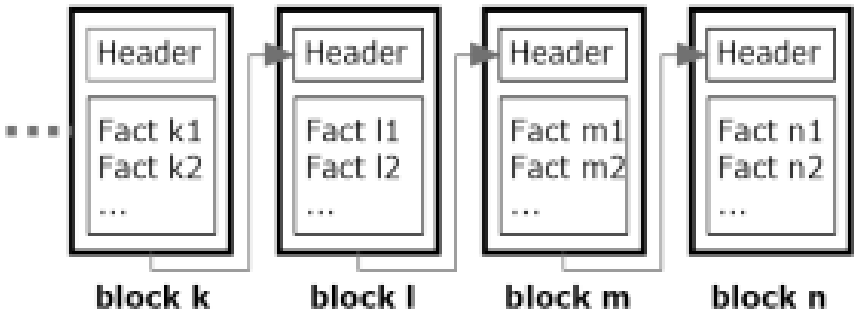


FIGURE 2.1 – Exemple simplifié d'une blockchain

Il existe deux types d'objets participants dans une blockchain :

— Objets en mode passif : Ces objets peuvent uniquement lire les faits (transactions) enregistrés.

—Objets en mode actif (mineurs) : Ces objets peuvent à la fois lire et écrire des faits.

Pour ajouter un nouveau bloc à la blockchain, il faut suivre les étapes suivantes :

— Les transactions sont regroupées dans un bloc.

— Les mineurs vérifient que les transactions du bloc respectent les règles définies.

— Les mineurs exécutent un mécanisme de consensus pour valider le bloc ajouté.

—Une récompense est donnée aux mineurs qui valident le bloc.

— Les transactions vérifiées sont stockées dans la blockchain.

Pour prouver la validation honnête d'un bloc, il existe plusieurs mécanismes de validation, les plus utilisés étant : Proof of Work (PoW) et Proof of Stake (PoS).

2.4 Propriétés de la blockchain

La blockchain est une technologie de registre distribué qui a plusieurs caractéristiques clés qui la distinguent d'autres systèmes de gestion de données. Voici les principales caractéristiques de la blockchain :

• **Décentralisation** : Dans les systèmes de transaction centralisés traditionnels, chaque transaction doit être validée par une autorité centrale de confiance, comme une banque centrale, ce qui peut entraîner des coûts élevés et créer des goulots d'étranglement au niveau des serveurs centraux. À l'inverse, sur un réseau blockchain, les transactions peuvent se faire directement entre deux pairs (P2P) sans nécessiter l'authentification d'une autorité centrale. Cela permet de réduire considérablement les coûts associés aux serveurs (y compris ceux de développement et d'exploitation) et de diminuer les problèmes de performance liés aux serveurs centraux.

• **Persistance** : Comme chaque transaction sur le réseau blockchain doit être confirmée et enregistrée dans des blocs distribués à travers tout le réseau, il est presque impossible de falsifier ces transactions. De plus, chaque bloc diffusé est validé par d'autres nœuds, et les transactions sont vérifiées en conséquence. Ainsi, toute tentative de falsification serait rapidement détectée.

• **Anonymat** : Les utilisateurs peuvent interagir avec le réseau blockchain en utilisant une adresse générée aléatoirement. En outre, ils peuvent créer plusieurs adresses pour minimiser l'exposition de leur identité. Ce système évite à une partie centrale de détenir des informations privées sur les utilisateurs, préservant ainsi une certaine confidentialité des transactions. Cependant,

il est important de noter que la blockchain ne peut pas garantir une confidentialité totale en raison de ses contraintes intrinsèques [4].

• **Auditabilité** : Puisque chaque transaction sur la blockchain est validée et enregistrée avec un horodatage, les utilisateurs peuvent facilement vérifier et retracer les transactions antérieures en accédant à n'importe quel nœud du réseau distribué. Par exemple, dans la blockchain Bitcoin, chaque transaction peut être retracée jusqu'aux transactions antérieures de manière itérative, ce qui améliore la traçabilité et la transparence des données stockées.

• **Immuabilité** : Les transactions sont stockées sur divers nœuds du réseau distribué, rendant la falsification de la blockchain publique quasiment impossible. Cependant, dans le cas d'une blockchain de consortium ou privée, si la majorité des membres du consortium ou de l'organisation dominante décidait de manipuler la blockchain, il serait théoriquement possible de renverser ou de falsifier ces données.

• **Transparence** : Toutes les transactions sur une blockchain sont visibles pour tous les participants du réseau. Cela permet une transparence complète, ce qui est particulièrement bénéfique pour des applications qui nécessitent une piste d'audit claire et transparente.

• **Sécurité** : La blockchain utilise des techniques cryptographiques avancées pour sécuriser les données. Chaque bloc de données est lié au précédent par un hachage cryptographique, ce qui rend pratiquement impossible la modification des données sans que cela soit détecté.

2.5 Structure de la blockchain

La structure de la blockchain repose sur plusieurs éléments essentiels qui interagissent pour former un registre distribué, sécurisé et immuable. Voici un aperçu des principaux composants qui la constituent :

La structure d'un bloc dans une blockchain illustre les composants essentiels qui forment un bloc typique dans un réseau blockchain. Un bloc est constitué de deux parties principales : l'en-tête de bloc et le corps de bloc. L'en-tête contient plusieurs éléments clés, tels que la version du bloc, le hachage de la racine de l'arbre de Merkle, l'horodatage, le champ nBits, le nonce, ainsi que le hachage du bloc parent [12].

Version du bloc : Indique l'ensemble des règles de validation qui régissent la signature et la création des blocs dans la blockchain.

Hachage de la racine de l'arbre de Merkle : Représente la valeur de hachage de l'ensemble des transactions contenues dans le bloc, assurant l'intégrité et l'authenticité des données.

Horodatage : Enregistre l'heure actuelle, exprimée en secondes (temps universel).

nBits : Définit le seuil cible pour qu'un hachage de bloc soit considéré comme valide.

Nonce : Il s'agit d'une valeur variable, généralement aléatoire, qui ne doit pratiquement jamais être répétée. Dans le contexte de la blockchain, et en particulier pour les crypto-monnaies comme le Bitcoin, le nonce est un nombre de 4 octets que les mineurs doivent résoudre pour générer un hachage valide, conformément aux exigences de difficulté. Le premier mineur à trouver le nonce valide reçoit une récompense en bitcoins [9].

Hachage du bloc parent : Une valeur de hachage hexadécimale de 256 bits qui fait référence au bloc précédent dans la chaîne, reliant ainsi chaque bloc à celui qui le précède pour former la chaîne de blocs.

Le corps du bloc est constitué de deux éléments :

Un compteur de transactions, qui enregistre le nombre de transactions contenues dans le bloc.

Une liste de transactions, qui contient les données de chaque transaction incluse dans le bloc.

Enfin, des mécanismes cryptographiques asymétriques sont utilisés pour valider l'authenticité des transactions sur la blockchain, garantissant ainsi la sécurité et la fiabilité du réseau.

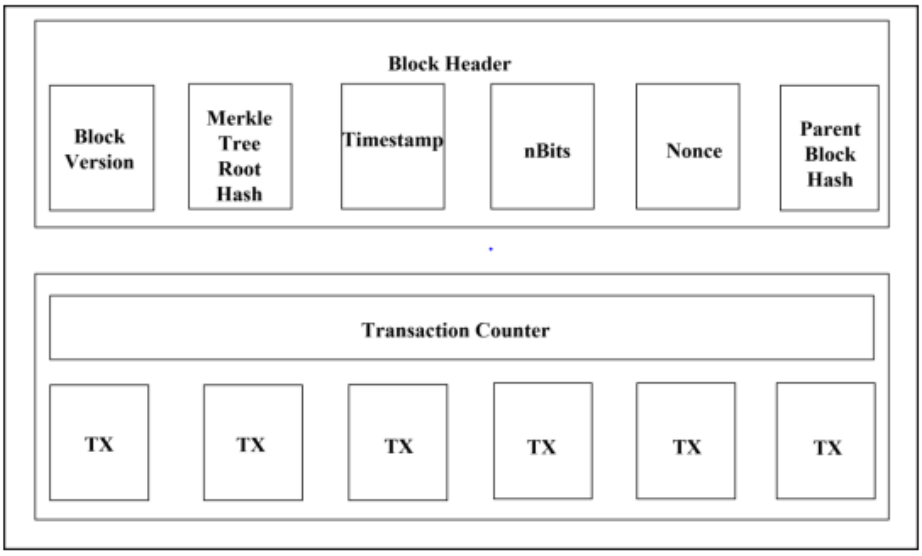


FIGURE 2.2 – Structure de la blockchain

2.6 Architecture de la blockchain

La couche réseau de la technologie blockchain constitue l'infrastructure sous-jacente qui permet la communication entre les participants du réseau. Elle intègre des mécanismes tels que la structure du réseau (mise en réseau pair-à-pair), la transmission des données, et la sécurité des informations. La mise en réseau pair-à-pair assure que chaque participant finit par recevoir les informations, à l'image d'un système de diffusion décentralisé. La transmission des données se concentre sur le transfert efficace des transactions et des blocs à travers le réseau. Quant à la sécurité des données, elle garantit l'intégrité et l'authenticité des informations échangées. L'optimisation de ces mécanismes est cruciale pour améliorer la rapidité, la sécurité et l'évolutivité des systèmes blockchain. [3].

La figure 3 illustre un schéma comparant les structures décentralisées et centralisées. Dans une architecture décentralisée, chaque nœud ou ordinateur agit à la fois comme serveur et utilisateur, ce qui leur permet de partager des données directement entre eux, renforçant ainsi la sécurité. En revanche, dans un système centralisé, un serveur unique stocke l'ensemble des données du réseau, ce qui crée un point de vulnérabilité potentiel.

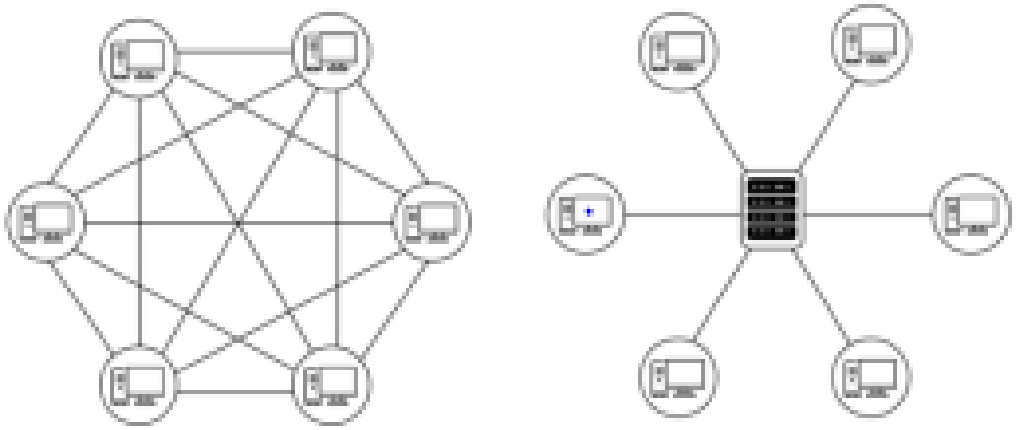


FIGURE 2.3 – Réseau Décentralisé vs Centralisé

2.7 Les différents types de blockchain

Les blockchains peuvent être classifiées selon leur architecture, leurs usages et le niveau d'implication dans le processus de consensus. On distingue notamment les blockchains avec autorisation et sans autorisation. Cette catégorisation repose sur la manière dont les nœuds participants rejoignent le réseau pour valider ou ajouter des transactions à un bloc.

Dans une blockchain sans autorisation, tous les nœuds actifs peuvent valider les transactions après avoir investi des ressources pour authentifier le processus d'ajout de données. L'accès au réseau en tant que nœud validateur est libre et ouvert à tous. À l'inverse, les blockchains avec autorisation restreignent l'accès aux nœuds externes, empêchant ces derniers de participer à la validation des transactions et à la mise à jour des blocs.

Il existe quatre grandes catégories de blockchains : les blockchains publiques, les blockchains privées, les blockchains de consortium, et les blockchains hybrides [18].

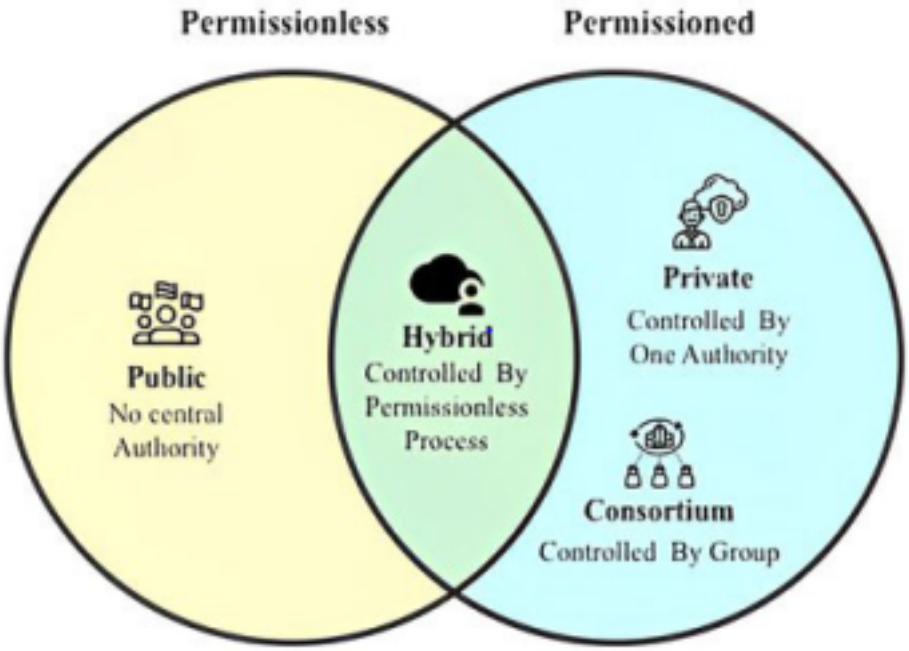


FIGURE 2.4 – Différents types de blockchain

2.8 Blockchain Publique

Une blockchain publique est un registre numérique décentralisé qui permet l'enregistrement et la vérification transparentes des transactions sur un réseau distribué d'ordinateurs. Fonctionnant selon un mécanisme de consensus, plusieurs participants (ou nœuds) s'accordent sur la validité des transactions tout en maintenant une chaîne chronologique de blocs sécurisés. Chaque bloc contient un ensemble de transactions et est cryptographiquement lié au bloc précédent, formant ainsi une chaîne immuable.
Une des principales caractéristiques des blockchains publiques est leur accessibilité ouverte : n'importe qui peut rejoindre le réseau en tant que nœud, participer à la validation des transactions, et consulter l'historique complet des transactions. Cette ouverture favorise un haut niveau de transparence et de confiance entre les participants.

Caractéristiques principales des blockchains publiques [21] :

Décentralisation : La décentralisation garantit que le réseau fonctionne de manière indépendante, sans qu'une autorité centrale ne puisse en dicter le fonctionnement. Chaque participant a une voix dans la gouvernance du réseau.

Transparence : Tous les participants peuvent consulter l'intégralité de l'historique des transactions. Cette transparence garantit une responsabilité accrue et empêche la falsification des enregistrements.

Sécurité : Pour protéger l'intégrité du réseau, les blockchains publiques utilisent des mécanismes de consensus tels que la preuve de travail (Proof of Work) ou la preuve d'enjeu (Proof of Stake). Ces systèmes garantissent la sécurité contre les accès non autorisés et les comportements malveillants.

Confiance minimisée : L'une des caractéristiques fondamentales des blockchains publiques est la possibilité d'interagir et d'effectuer des transactions directement entre participants, sans avoir besoin de recourir à des intermédiaires, éliminant ainsi la nécessité de faire confiance à un tiers.

2.9 Blockchain Privée

Une blockchain privée opère dans un réseau fermé, réservé exclusivement aux entités autorisées. Contrairement aux blockchains publiques, elle propose un accès strictement contrôlé. Seuls des participants spécifiques, appelés nœuds, sont autorisés à valider les transactions et à gérer la blockchain dans un environnement privé. Cette structure garantit une confidentialité accrue, car seuls les membres approuvés peuvent interagir avec le registre. Les blockchains privées sont largement adoptées par les organisations qui requièrent une gestion sécurisée et autorisée des transactions.

Caractéristiques clés des blockchains privées :

Accès contrôlé : Les participants doivent obtenir une autorisation d'une autorité centrale ou d'un groupe d'entités pour pouvoir rejoindre le réseau et valider les transactions.

Confidentialité accrue : Ces blockchains sont souvent dotées de fonctionnalités renforcées qui protègent les informations sensibles, accessibles uniquement aux membres autorisés.

Haute évolutivité : Avec un nombre de participants limité, les blockchains privées sont capables de traiter un volume de transactions plus important que les blockchains publiques, ce qui se traduit par une meilleure performance en termes de rapidité et d'efficacité.

Centralisation : Contrairement aux blockchains publiques, les blockchains privées sont plus centralisées, car une autorité centrale ou un consortium détient le contrôle du réseau et des processus de validation.

2.10 Blockchain hybride

La blockchain hybride combine les avantages des blockchains publiques et privées, tout en surmontant leurs limitations respectives. Elle permet aux différentes parties d'un réseau de fonctionner sur des blockchains distinctes, avec une interopérabilité fluide entre les éléments publics et privés.

Caractéristiques principales des blockchains hybrides :

Flexibilité : Les blockchains hybrides offrent une flexibilité optimale, permettant aux utilisateurs

de choisir la blockchain la plus adaptée à chaque cas d'utilisation, tout en restant au sein d'un même réseau.

Interopérabilité : Elles permettent un échange fluide de données et d'actifs entre les composantes publiques et privées, assurant ainsi une intégration harmonieuse des différentes parties du réseau.

Confidentialité : La blockchain hybride garantit une meilleure confidentialité en utilisant des blockchains privées pour gérer les données sensibles, tout en profitant de la transparence et de la sécurité des blockchains publiques pour d'autres activités.

Personnalisation : Les organisations peuvent configurer des blockchains hybrides selon leurs besoins spécifiques, équilibrant ainsi les exigences en matière de confidentialité, de sécurité et de performance.

2.11 Blockchain consortium

La blockchain de consortium est un registre numérique qui combine des aspects autorisés et décentralisés, permettant une collaboration sécurisée entre un groupe restreint de participants. Contrairement aux blockchains publiques, ouvertes à tous, une blockchain de consortium est restreinte à des nœuds de confiance, sélectionnés et approuvés par les membres du consortium. Ce contrôle d'accès assure une meilleure confidentialité et permet un traitement plus rapide des transactions comparé aux blockchains publiques.

Le consensus au sein d'une blockchain de consortium est atteint via des mécanismes sur mesure, adaptés aux besoins des organisations participantes. Ces mécanismes peuvent inclure des versions modifiées de la preuve d'enjeu ou de la tolérance aux pannes byzantines, avec une priorité accordée à l'efficacité et à la rapidité plutôt qu'à une participation entièrement ouverte. Seuls les participants approuvés sont habilités à valider les transactions et à maintenir l'intégrité de la blockchain, renforçant ainsi la sécurité et la confidentialité. Ce modèle convient parfaitement aux secteurs nécessitant un partage de données sécurisé entre plusieurs parties prenantes, comme la gestion de la chaîne d'approvisionnement, la finance ou les soins de santé.

Caractéristiques clés des blockchains de consortium :

Semi-décentralisation : Ces blockchains se situent à mi-chemin entre la décentralisation des

blockchains publiques et la centralisation des blockchains privées. Le consensus et la gouvernance sont partagés entre des participants préalablement approuvés, établissant un équilibre entre décentralisation et contrôle.

Accès autorisé : Seuls les participants ayant obtenu une autorisation préalable peuvent intégrer le réseau et participer au processus de validation, garantissant un environnement sécurisé et contrôlé.

Évolutivité accrue : Grâce à un nombre limité de participants et des mécanismes de consensus plus ciblés, les blockchains de consortium offrent une meilleure évolutivité que les blockchains publiques, avec une capacité à traiter un volume plus important de transactions.

Confidentialité renforcée : Les blockchains de consortium mettent un fort accent sur la confidentialité, permettant aux participants de gérer strictement l'accès aux données sensibles à l'intérieur du réseau.

2.12 Mécanisme de la blockchain

Une blockchain est un registre décentralisé qui enregistre toutes les transactions au sein d'un réseau pair-à-pair (P2P). Cette technologie permet aux participants de valider des transactions sans l'intervention d'un intermédiaire pour leur finalisation. Elle trouve des applications dans divers domaines tels que les transferts de fonds, le règlement de transactions, les systèmes de vote, l'Internet des objets (IdO), et bien d'autres secteurs [11].

Étant donné que la création de nouvelles données sur la blockchain est un processus concurrentiel, plusieurs blocs peuvent être proposés simultanément au réseau pour validation. Le premier bloc à fournir une preuve de travail valide est ajouté à la chaîne. Cette nouvelle chaîne est ensuite validée par le reste du réseau, qui recalculera la preuve de travail, avant d'être diffusée à tous les nœuds.

Il existe deux types de nœuds sur le réseau blockchain : les utilisateurs, qui génèrent des informations, et les mineurs, qui produisent les preuves de travail. Chaque mineur et utilisateur détient une copie locale de tout ou partie de la chaîne.

Plusieurs mineurs travaillent en parallèle pour trouver la preuve de travail et créer de nouveaux

blocs, dans l'espoir d'obtenir une récompense sous forme de cryptomonnaie. Cependant, pendant que l'un d'entre eux résout un bloc, un autre mineur peut valider un bloc différent et le diffuser à une autre partie du réseau, ce qui est possible en raison du délai de propagation des données sur le réseau.

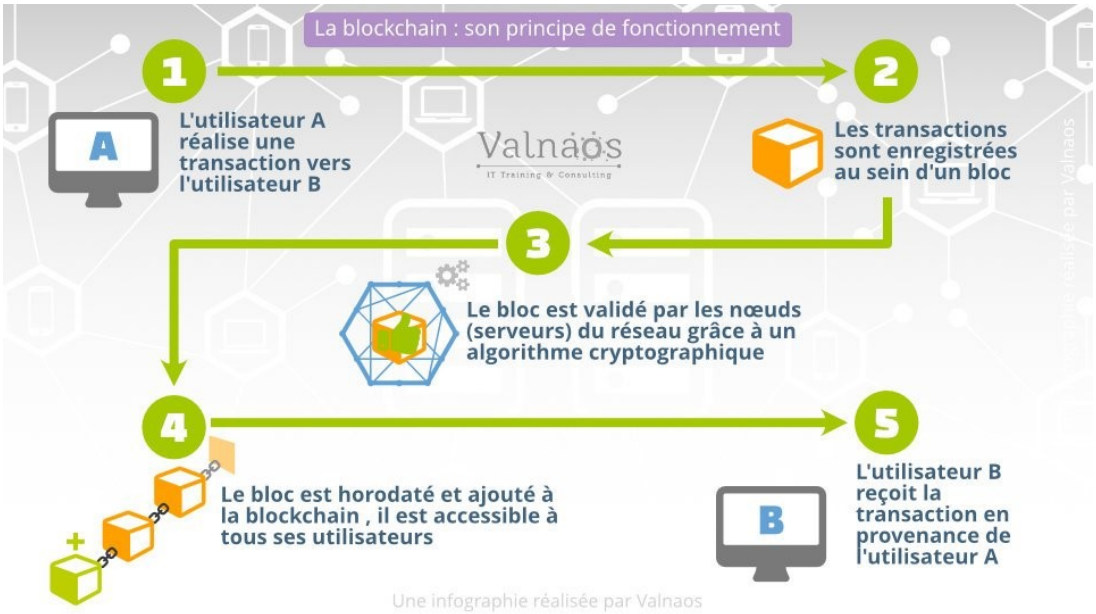


FIGURE 2.5 – Mécanisme de la blockchain

2.13 Plateformes de la blockchain

La blockchain est en constante évolution, et de nombreuses plateformes existent aujourd'hui. Les exemples les plus notables incluent Bitcoin, Ethereum et Hyperledger Fabric liés à l'internet des objets [17].

2.13.1 Bitcoin

Bitcoin est une crypto-monnaie qui repose sur un réseau peer-to-peer pour vérifier les transactions d'actifs numériques. Il utilise un protocole de consensus de preuve de travail (Proof of Work) pour valider les transactions via un mécanisme décentralisé, éliminant ainsi le risque de point de défaillance unique souvent associé aux systèmes centralisés de validation financière. La blockchain Bitcoin assure la sécurité des transactions en utilisant un horodatage et des fonctions

de hachage cryptographique pour créer un registre public immuable. Contrairement aux systèmes traditionnels de transfert électronique de fonds, qui dépendent d'une autorité de confiance, Bitcoin utilise des preuves cryptographiques pour valider les transactions. Les utilisateurs peuvent créer des portefeuilles de paiement en utilisant des adresses spécifiques, sans besoin d'une tierce partie centralisée.

2.13.2 Ethereum

Ethereum, en plus de valider les transactions financières, permet également l'utilisation de contrats intelligents via la machine virtuelle Ethereum (EVM). Sa cryptomonnaie, appelée Ether (ETH), est utilisée pour financer ces opérations. La plateforme Ethereum offre une plus grande flexibilité que Bitcoin, car elle permet non seulement de traiter des transactions financières, mais également de gérer d'autres types de données à travers des contrats intelligents. Cela permet de créer des applications décentralisées (DApps), qui fonctionnent de manière autonome sur le réseau.

2.13.3 Hyperledger Fabric

Hyperledger Fabric (HLF), développé sous l'égide de la Linux Fondation, est une plateforme de registre distribué open source, avec des autorisations. Elle soutient le développement de contrats intelligents en utilisant des langages de programmation de haut niveau tels que Golang et Java. Les mécanismes d'autorisation avancés de HLF permettent de gérer l'isolement des registres et de faciliter les transactions privées entre différentes organisations. Grâce à son architecture modulaire et flexible, HLF s'adapte à une variété d'applications industrielles basées sur la blockchain [16].

2.13.4 IOTA

IOTA est une blockchain publique conçue pour faciliter les contrats intelligents, notamment pour les transactions entre appareils IoT (Internet des objets). Grâce à son mécanisme de consensus

innovant appeler Tangle, elle se distingue des plateformes classiques comme Ethereum ou Bitcoin. Plutôt que de recourir à des blocs ou à des mineurs, Tangle exige que chaque appareil IoT valide deux transactions avant de pouvoir initier la sienne. Cette structure sans blocs, sans frais de transaction et sans mineurs permet de résoudre les problèmes d'évolutivité et de coûts, tout en offrant une solution idéale pour les appareils IoT aux ressources limitées, leur permettant d'enregistrer des données de manière efficace.

2.13.5 Procedure de consensus

Un algorithme de consensus est un processus employé dans les réseaux distribués, tels que les blockchains, afin de garantir que tous les participants (ou nœuds) parviennent à un accord sur une version unique de l'état du réseau, même en cas de défaillances ou de comportements malveillants. Ces algorithmes jouent un rôle crucial pour assurer la sécurité, l'intégrité et la cohérence des données partagées au sein du réseau.
Il existe différents types de mécanismes de consensus en fonction du protocole adopté [15] et [22] :

2.13.5.1 La preuve de travail (PoW)

Dans ce mécanisme, les nœuds, appelés mineurs, doivent résoudre des problèmes mathématiques complexes. Le premier nœud à trouver la solution la partage avec les autres nœuds, qui la vérifient avant d'ajouter la transaction au registre distribué (DL). L'un des principaux inconvénients de ce consensus est sa consommation énergétique extrêmement élevée. Ce mécanisme est utilisé par Bitcoin.

2.13.5.2 La preuve d'enjeu (POS)

Dans ce modèle, les nœuds, appelés validateurs, sont choisis pour valider des blocs en fonction du nombre et de l'ancienneté de leurs parts (jetons). Les validateurs avec les parts les plus importantes et les plus anciennes ont plus de chances d'être sélectionnés.

2.13.5.3 La preuve d'enjeu déléguée (DPoS)

Ce mécanisme repose sur un sous-ensemble de validateurs PoS, pour lesquels la tâche de validation est déléguée par les autres participants du réseau. Les détenteurs de parts votent pour les validateurs qui effectueront les validations de blocs.

2.13.5.4 La preuve d'enjeu louée (LPoS)

Ici, les nœuds peuvent louer leurs actifs à d'autres nœuds pour augmenter les chances d'être sélectionnés comme validateurs. Les récompenses issues de la validation sont ensuite partagées entre les membres du groupe de nœuds participants.

2.13.5.5 La preuve d'autorite (PoA)

Les validateurs sont présélectionnés et autorisés à valider les blocs en fonction de leur identité et de leur réputation. Ce mécanisme repose sur la confiance accordée aux validateurs plutôt que sur des ressources matérielles ou financières. Ethereum utilise ce type de consensus dans certains réseaux privés ou autorisés.

2.13.5.6 La preuve d'activité (PoAC)

Ce consensus hybride commence par un mécanisme de PoW, où les mineurs résolvent des problèmes pour récolter des récompenses. Une fois ces récompenses obtenues, le système bascule vers un modèle PoS, où les récompenses sont utilisées comme mise (stake) pour continuer la validation.

2.13.5.7 Protocole de sécurité

Cela peut-être définit comme étant une méthode ou un ensemble de processus mis en place pour protéger un système, des informations ou un réseau contre les menaces, les attaques et les accès non autorisés. Ces mécanismes sont indispensables pour garantir la confidentialité, l'intégrité et la disponibilité des données.

2.13.5.8 Cryptographie

La cryptographie est la science permettant de sécuriser les communications et les données en les rendant inaccessibles ou incompréhensibles aux personnes non autorisées. Elle repose sur des algorithmes mathématiques complexes qui garantissent la confidentialité, l'intégrité et l'authenticité des informations. Essentielle dans le monde numérique, la cryptographie protège les transactions bancaires, les emails et les échanges sur Internet. Cependant, il existe deux types de cryptographie : symétrique (clé secrète ou privée) et asymétrique (clé publique) [19].

2.13.6 Cryptographie Symétrique

La cryptographie symétrique est une méthode de cryptographie dans laquelle une seule et même clé est utilisée à la fois pour chiffrer et déchiffrer les informations. Cela signifie que les deux parties, l'expéditeur et le destinataire, doivent avoir accès à cette clé secrète partagée pour pouvoir échanger des messages en toute sécurité. Si la clé privée est perdue ou volée, la sécurité du système devient vulnérable [18]. En effet, on peut citer quelques algorithmes symétriques :

—**Data Encryption Standard (DES)** : En 1977, le Data Encryption Standard (DES) est établi comme norme de chiffrement par le National Bureau of Standards des États-Unis. Rapidement critiqué par la communauté cryptographique, le DES est jugé trop faible en raison de sa clé de 64 bits, dont seulement 56 sont réellement efficaces. À la fin des années 1990, plusieurs attaques ont mis en évidence ses faiblesses. En 1999, par exemple, distributed.net a réussi à déchiffrer la clé en moins de 24 heures en utilisant environ 100 000 ordinateurs connectés à Internet.
Pour remédier à cette situation, l'agence américaine National Institute of Standards and Technology (NIST) a lancé en 1997 un appel à candidatures pour trouver un remplaçant au DES. Actuellement, l'AES est considéré comme la norme de référence pour le chiffrement.

—**Triple-DES (3DES)** : C'est une méthode de chiffrement qui applique l'algorithme DES trois fois sur chaque bloc de données. Cela augmente la sécurité par rapport au DES standard, mais il est moins efficace que l'AES et est désormais considéré comme obsolète en raison de sa vulnérabilité face à des attaques modernes. 3DES est souvent remplacé par l'AES dans les applications de sécurité actuelle.

—**Advanced Encryption Standard (AES)** : En 2001, un variant du candidat Rijndael a été sélectionné et est devenu l'Advanced Encryption Standard (AES). Cet algorithme se décline en trois configurations, utilisant des clés de 128, 192 et 256 bits. Après deux décennies de cryptanalyse, l'AES est toujours jugé mathématiquement sûr et demeure l'algorithme de chiffrement symétrique le plus largement adopté dans le monde.

—**RC (Ron's Cipher ou Ron's Code)** : Ron's Cipher, également appelé Ron's Code (RC), fait référence à une série d'algorithmes de chiffrement symétrique développés par Ron Rivest. Cette famille comprend les algorithmes RC2 à RC6. Parmi eux, RC4 est le plus connu et a été largement utilisé dans des applications telles que les protocoles SSL/TLS et le chiffrement des réseaux sans fil via WEP.

Cependant, RC4 a été critiqué pour ses nombreuses vulnérabilités, notamment des faiblesses dans sa génération de clés, ce qui a entraîné son retrait progressif de nombreux systèmes de sécurité modernes. Des solutions plus sûres, comme l'AES, sont désormais préférées pour le chiffrement des données.

—**International Data Encryption Algorithm (IDEA)** : C'est un algorithme de chiffrement par blocs conçu pour offrir un haut niveau de sécurité dans les communications et le stockage de données. Il a été développé en 1991 par Xuejia Lai et James Massey. IDEA utilise une clé de 128 bits et opère sur des blocs de 64 bits, ce qui le rend efficace et sécurisé pour de nombreuses applications.

L'algorithme est connu pour sa résistance aux attaques cryptographiques telles que les attaques par force brute. Il a été largement utilisé dans les systèmes de chiffrement comme PGP (Pretty Good Privacy). Bien que des alternatives plus récentes comme l'AES (Advanced Encryption Standard) aient gagné en popularité, IDEA reste un exemple marquant de la cryptographie moderne et a influencé de nombreux algorithmes ultérieurs.

2.13.7 Cryptographie Asymétrique

La cryptographie asymétrique, également appelée cryptographie à clé publique, est une technique qui repose sur l'utilisation d'une paire de clés : une clé publique et une clé privée. Contrairement à la cryptographie symétrique, qui utilise une seule et même clé pour chiffrer

et déchiffrer les données, ces deux clés sont distinctes mais mathématiquement liées. La clé publique : Cette clé peut être partagée librement avec n'importe qui. Elle est utilisée pour chiffrer des messages ou pour vérifier des signatures numériques et la clé privée : Cette clé est gardée secrète par le propriétaire. Elle sert à déchiffrer les messages qui ont été chiffrés avec la clé publique correspondante ou à signer des messages afin d'attester de leur authenticité. En effet, on peut citer quelques algorithmes asymétriques .

—**RSA (Rivest, Shamir et Adleman)** : RSA est un algorithme de cryptographie asymétrique largement utilisé pour le chiffrement et la signature numérique. Il a été proposé en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman. RSA repose sur la difficulté de factoriser de grands nombres premiers.

—**Diffie-Hellman** est un algorithme de cryptographie asymétrique conçu pour établir un secret partagé entre deux parties, facilitant ainsi l'échange sécurisé de clés. Proposé en 1976 par Whitfield Diffie et Martin Hellman, il est souvent considéré comme l'un des pionniers des protocoles de cryptographie moderne.

—**ECC (Elliptic Curve Cryptography en français Cryptographie sur les Courbes Elliptiques)** : La Cryptographie sur les Courbes Elliptiques (ECC) est une méthode de cryptographie asymétrique qui repose sur les propriétés mathématiques des courbes elliptiques. Elle offre un niveau de sécurité équivalent à d'autres systèmes de cryptographie, comme RSA, mais avec des clés beaucoup plus courtes, ce qui la rend plus efficace.

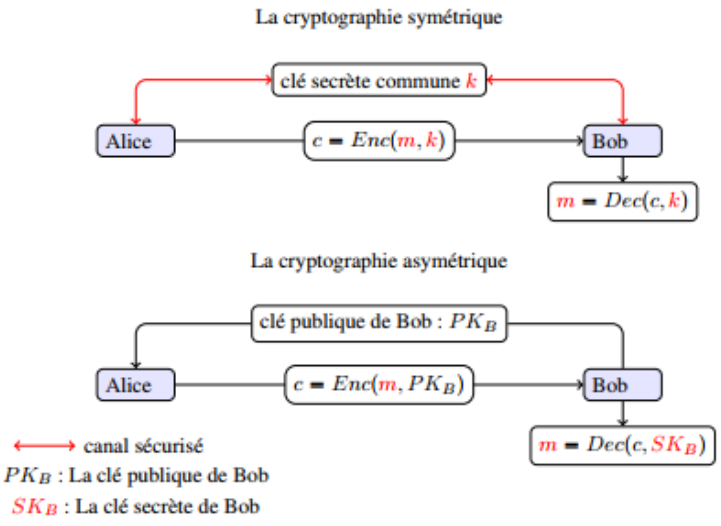


FIGURE 2.6 – Cryptographie symétrique et Cryptographie asymétrique

Comme l'illustre le tableau comparatif des tailles de clé (en bits) par niveau de sécurité ci-dessous, l'ECC peut offrir la même force de chiffrement qu'un système basé sur l'algorithme RSA, tout en utilisant des clés beaucoup plus courtes. En effet, une clé ECC de 256 bits équivaut à une clé RSA de 3072 bits, ce qui est donc considérablement plus long que les clés de 2048 bits utilisées de nos jours. En revanche, la taille de la clé symétrique est plus courte que celle des clés asymétriques, mais il est important de noter qu'une seule clé est partagée.

Bits de sécurité	Taille de module RSA	Taille de clef ECC
80	1024	160
112	2048	224
128	3072	256
192	7680	384
256	15360	512

FIGURE 2.7 – Tableau comparatif des tailles de clé (en bits) par niveau de sécurité

2.13.8 Signature Numérique

Une signature numérique est un procédé cryptographique permettant de garantir l'intégrité et l'authenticité d'un message ou d'un document électronique. Elle joue un rôle essentiel dans les communications sécurisées et les transactions en ligne, en s'assurant que le contenu n'a pas été modifié et qu'il provient bien de l'expéditeur déclaré [20].

Le processus de création et de vérification d'une signature numérique suit généralement les étapes suivantes :

1. Création de la signature

   — Hachage du message : Le message est d'abord soumis à une fonction de hachage cryptographique (comme SHA-256) pour générer un condensé de longueur fixe, appelé "empreinte" ou "hash".

   — Chiffrement de l'empreinte : L'empreinte est ensuite chiffrée avec la clé privée de l'expéditeur, créant ainsi la signature numérique. Seul le détenteur de la clé privée peut produire cette signature.

2. Vérification de la signature

   — Déchiffrement de la signature : Le destinataire utilise la clé publique de l'expéditeur pour déchiffrer la signature, récupérant ainsi l'empreinte originale.

   — Hachage du message reçu : Le destinataire applique également la fonction de hachage au message reçu pour obtenir une nouvelle empreinte.

   — Comparaison des empreintes : Si l'empreinte déchiffrée correspond à celle nouvellement sélectionnée, cela confirme que le message n'a pas été altéré et que la signature est valide.

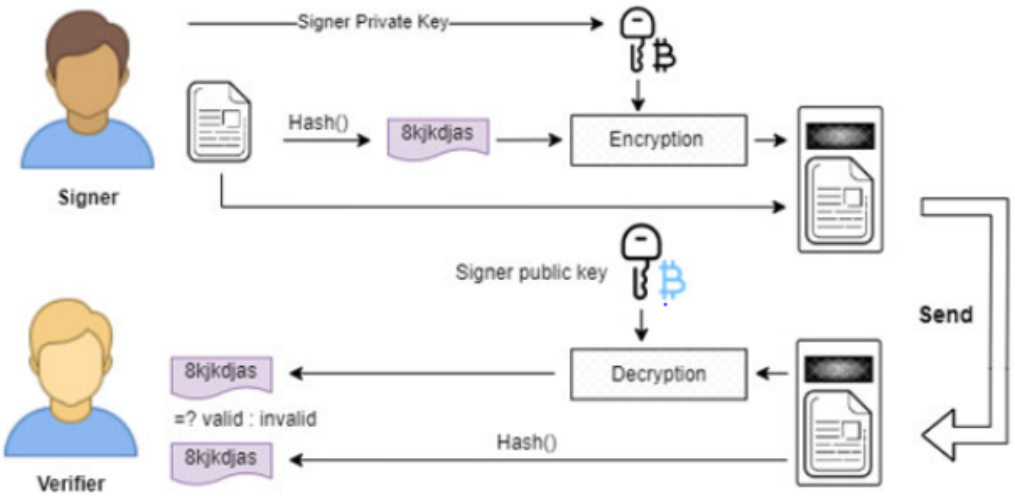


FIGURE 2.8 – Signature numérique et vérification

2.13.9 Fonction de hachage

Une fonction de hachage est un algorithme qui convertit des données de taille variable, telles qu'un fichier, un message ou un mot de passe, en une sortie de taille fixe appelée empreinte ou hash, généralement représentée sous forme de chaîne hexadécimale. Une fonction de hachage cryptographique possède plusieurs propriétés : elle est déterministe, produisant toujours la même empreinte pour une même entrée, rapide à calculer, résistance aux collisions (trouver deux entrées différentes générant la même empreinte est extrêmement difficile) et à la préimage (il est presque impossible de retrouver l'entrée à partir de l'empreinte), et elle présente une forte diffusion, où tout petit changement dans l'entrée provoque une modification significative de l'empreinte. Parmi les algorithmes de hachage les plus connus figurent : MD5 (128 bits),

autrefois populaire mais désormais considéré comme non sécurisé en raison de sa vulnérabilité aux collisions ; SHA-1 (160 bits), plus robuste que MD5 mais également obsolète en raison de failles de sécurité ; SHA-256 (256 bits), largement utilisé pour l'intégrité des données et les signatures numériques, appartenant à la famille SHA-2 ; et SHA-3, basé sur l'algorithme Keccak, offrant une sécurité renforcée avec des tailles d'empreintes variables (224, 256, 384 ou 512 bits), surpassant SHA-2 en robustesse [20].

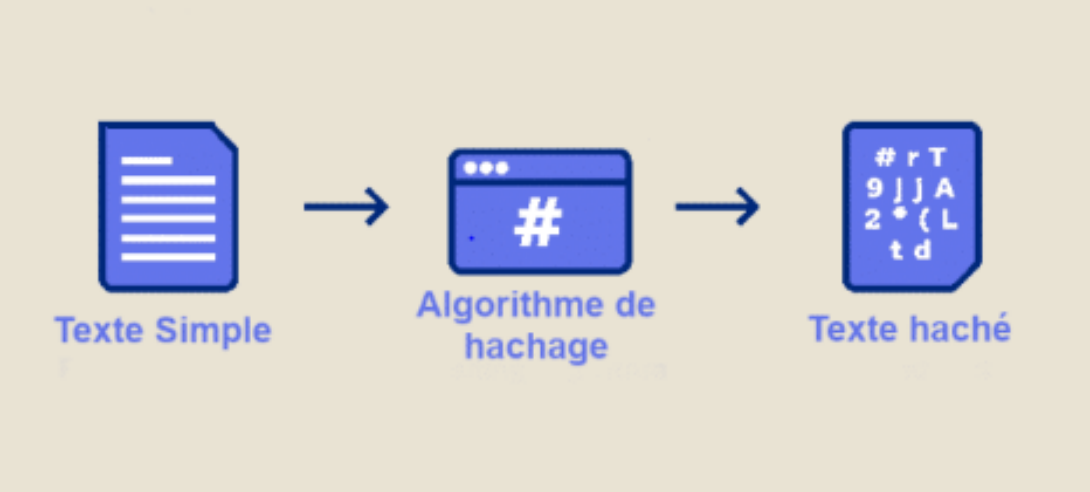


FIGURE 2.9 – Principe de hachage

2.14 Conclusion

La blockchain, en tant qu'innovation majeure, apporte un modèle décentralisé, sécurisé et transparent qui renforce la confiance dans divers secteurs en éliminant les intermédiaires et en offrant un registre immuable. Ses applications s'étendent à des domaines tels que la finance, la logistique, les services publics et la gouvernance. De plus, sa convergence avec l'Internet des objets (IoT) ouvre de nouvelles perspectives pour la sécurisation et l'automatisation des échanges de données entre objets connectés. Dans le chapitre suivant, nous examinons les méthodes de confidentialité pour l'Internet des objets basés sur la blockchain, en les catégorisant selon leurs caractéristiques.

Chapitre

3 Etablissement de Sessions IoT Sécurisées par la Blockchain

Sommaire

3.1 Introduction 42
3.2 Problèmes de confidentialité 43
3.3 Problèmes d'intégrité des données 43
3.4 Solution pour améliorer la confidentialité et l'intégrité dans l'IoT 44
3.5 Objectif proposé 45
3.5.1 Architecture proposée 45
3.5.2 Protocole de confidentialité et d'intégrité 47
3.5.3 Etablissement d'une session de communication entre un Serveur S, un Utilisateur U et un Nœud N 48
3.5.4 Phase de déroulement 51
3.6 Conclusion 60

3.1 Introduction

L'Internet des Objets (IoT) constitue un réseau de dispositifs intelligents et interconnectés qui bouleverse notre quotidien en intégrant des capteurs, des objets connectés et des systèmes autonomes dans divers secteurs tels que la santé, les transports, les villes intelligentes, et les environnements domestiques. Avec la prolifération de ces dispositifs, capables de collecter et d'échanger en temps réel des volumes considérables de données, nos interactions avec la technologie sont profondément transformées. Cependant, cette évolution rapide engendre des défis significatifs, en particulier concernant la confidentialité et l'intégrité des données. Les objets connectés capturent et transmettent des informations souvent sensibles, comme des données de géolocalisation, des habitudes de vie ou encore des paramètres de santé.

Dans ce cadre, l'établissement de sessions sécurisées dans l'IoT par la blockchain est une

approche innovante pour résoudre les défis liés à la sécurité, à la confidentialité et à l'intégrité des communications entre les appareils connectés.
Cependant, on distingue deux types de problèmes : Confidentialité et Intégrité.

3.2 Problèmes de confidentialité

Dans l'IoT, la confidentialité fait référence à la protection des données personnelles collectées et transmises par les dispositifs connectés. Ces données incluent des informations sensibles telles que les données de santé, la localisation, et les habitudes de vie. Les risques associés concernent principalement le vol de ces informations ou leur exposition à des tiers non autorisés [14].

—**Vérification de l'authenticité des dispositifs** : Chaque appareil IoT doit faire l'objet d'une vérification pour garantir qu'il est authentique et qu'il est autorisé à communiquer sur le réseau. Cela permet d'empêcher les dispositifs non autorisés ou malveillants d'accéder aux informations sensibles ou de recueillir des données personnelles sans le consentement des utilisateurs.

—**Chiffrement et signature numérique des données** : Les informations échangées par les appareils IoT peuvent être protégées grâce au chiffrement et à la signature numérique. Le chiffrement garantit la confidentialité des données en empêchant leur interception par des tiers, tandis que la signature numérique assure que l'expéditeur des informations est légitime. Cela évite l'introduction de dispositifs frauduleux ou de fausses données dans le réseau.

3.3 Problèmes d'intégrité des données

L'intégrité des données assure que celles-ci restent inchangées tout au long de leur transmission ou de leur stockage. Dans un contexte IoT, toute altération des données peut compromettre la fiabilité des systèmes critiques, tels que les infrastructures industrielles ou les dispositifs médicaux [13].

—**Vérification de l'intégrité des données** : Les mécanismes de hachage cryptographique permettent de s'assurer que les données transmises entre les dispositifs IoT n'ont pas été modifiées. Lors de

l'envoi, les données sont converties en un hachage (une valeur unique), qui est ensuite comparé à celui obtenu à la réception. Si les valeurs diffèrent, cela signale une altération potentielle.

—**Signature numérique pour garantir l'intégrité** : En complément du hachage, la signature numérique certifie que les données proviennent d'une source authentique et n'ont pas été modifiées durant leur transfert. Chaque message envoyé par un dispositif IoT est signé avec une clé privée, et la clé publique correspondante permet au destinataire de vérifier son authenticité et son intégrité.

3.4 Solution pour améliorer la confidentialité et l'intégrité dans l'IoT

—**Chiffrement des communications** : Le chiffrement des données est essentiel pour protéger la confidentialité et l'intégrité des informations échangées par les dispositifs IoT. Cela empêche les attaquants de lire ou de modifier les données interceptées.

—**Authentification renforcée** : L'authentification des dispositifs et des utilisateurs à l'aide de mécanismes robustes, comme l'authentification multifactorielle, est cruciale pour s'assurer que seuls les dispositifs autorisés ont accès aux réseaux IoT.

—**Mises à jour de sécurité régulières** : Les fabricants doivent proposer des mises à jour régulières pour corriger les vulnérabilités connues et renforcer la sécurité des dispositifs IoT.

—**Blockchain pour la traçabilité et l'intégrité des données** : La blockchain peut être utilisée pour garantir l'intégrité des données dans un système décentralisé. Elle permet d'enregistrer de manière immuable les transactions et les échanges de données, ce qui rend plus difficile la falsification ou la modification des informations.

3.5 Objectif proposé

L'objectif principal de notre approche est de concevoir un mécanisme de confidentialité et d'intégrité dans les systèmes IoT en s'inspirant des principes fondamentaux de la blockchain. Cette approche vise à sécuriser les échanges de données entre dispositifs IoT en garantissant que les informations sensibles sont protégées contre les accès non autorisés et les altérations. Notre solution repose sur des concepts tels que la décentralisation, la transparence, et l'immutabilité de la blockchain, permettant ainsi de créer un réseau où la confidentialité des données est assurée et où leur intégrité peut être vérifiée. En intégrant des techniques de chiffrement, de hachage et de signature numérique, notre approche garantit que seuls les dispositifs autorisés peuvent échanger des informations de manière sécurisée et que les données transmises n'ont pas été modifiées en cours de route. Cette architecture vise à fournir une solution robuste et fiable pour les écosystèmes IoT, les rendant plus résistants aux attaques, tout en préservant la confidentialité et l'intégrité des données.

3.5.1 Architecture proposée

Pour une architecture IoT intégrée à la blockchain avec un objectif de confidentialité et d'intégrité, voici les couches principales, chacune jouant un rôle essentiel dans la sécurisation et le fonctionnement du système.

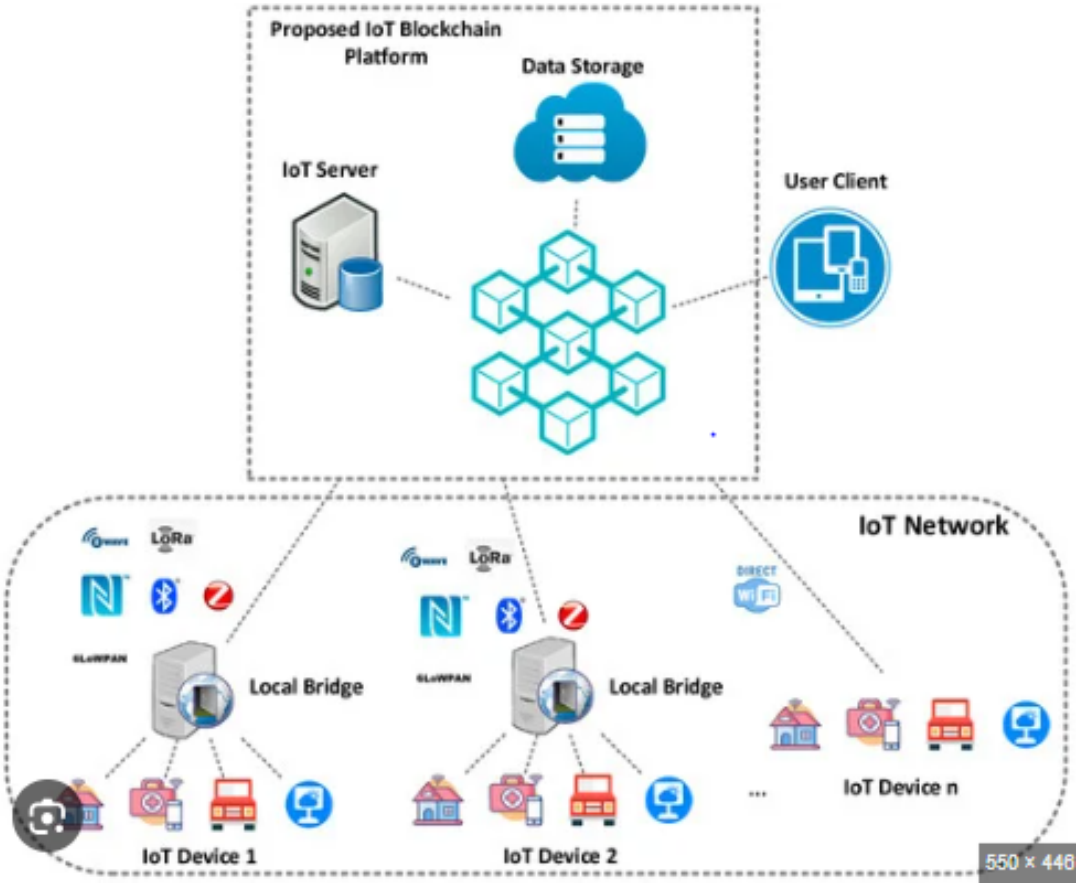


FIGURE 3.1 – Architecture du système décentralisée

Nœuds IoT (dispositifs connectés) : Inclut les capteurs, appareils intelligents et équipements industriels. Chaque nœud collecte des données en temps réel, générant ainsi de précieuses informations.

NPasserelle IoT sécurisée : Fonctionne comme une interface entre les nœuds IoT et le réseau blockchain, assurant la communication. Elle regroupe les données, les pré-traite, et applique des mécanismes de sécurité.

NRéseau Blockchain (réseau de validation distribué) :

—**Nœuds validateurs** : Ce sont des serveurs ou équipements puissants responsables de la vérification des transactions et du maintien du consensus au sein du réseau.

—**Contrats intelligents (Smart Contracts)** : Scripts programmés pour automatiser les processus de validation et gérer les règles de confidentialité et d'intégrité des données.

Système de gestion des clés : Assure la gestion des clés de chiffrement utilisées pour les signatures numériques, en gérant la distribution et le renouvellement des clés pour chaque nœud.

Stockage décentralisé sécurisé : Espace de stockage crypté et distribué pour les données critiques (situées hors blockchain) afin d'éviter les surcharges.

3.5.2 Protocole de confidentialité et d'intégrité

La confidentialité et l'intégrité dans un environnement IoT utilisant la blockchain :

Confidentialité : Assurer que seules les parties autorisées peuvent accéder aux données transmises dans le réseau IoT.

Intégrité : Garantir que les données échangées n'ont pas été altérées en cours de transmission.

Authenticité : Valider l'identité des dispositifs IoT et des nœuds de communication pour prévenir toute intrusion malveillante.

Étape 1 : Initialisation et Authentification des Dispositifs

— Chaque dispositif IoT, nœud de serveur et utilisateur reçoit une paire de clés publique/privée générée à l'aide d'un algorithme de cryptographie asymétrique, comme l'ECC (Elliptic Curve Cryptography).

— Les clés publiques sont enregistrées dans une blockchain dédiée qui permet à chaque participant de vérifier l'authenticité d'un autre participant. — Les dispositifs IoT s'authentifient mutuellement en échangeant des certificats numériques signés, garantissant ainsi leur identité et empêchant les appareils non autorisés de s'intégrer dans le réseau.

Étape 2 : Établissement de la Session Sécurisée

— Une fois authentifiés, les dispositifs IoT et les nœuds de serveur établissent une clé de session partagée en utilisant un protocole de chiffrement symétrique. Cette clé est utilisée pour chiffrer les données échangées afin d'assurer leur confidentialité.

— La clé de session est elle-même protégée par un chiffrement asymétrique pour éviter toute interception.

Étape 3 : Vérification de l'Intégrité des Données

— Avant de transmettre des données, chaque dispositif IoT utilise une fonction de hachage cryptographique (par exemple SHA-256) pour générer un hachage unique des données.

— Ce hachage est ensuite signé numériquement avec la clé privée du dispositif, garantissant ainsi l'intégrité et l'authenticité des données à la réception.

Étape 4 : Transmission et Validation sur la Blockchain

— Les données chiffrées et signées sont transmises à la passerelle IoT et ensuite vers les nœuds validateurs de la blockchain. — Les nœuds validateurs vérifient la signature numérique et comparent le hachage reçu avec celui généré pour confirmer que les données n'ont pas été modifiées. — Si les données passent cette vérification, elles sont ajoutées à la blockchain, créant ainsi une trace immuable qui assure la transparence et la non-répudiation.

Étape 5 : Surveillance et Révocation de la Session

— Le serveur central surveille en permanence les activités des dispositifs IoT pour identifier les comportements suspects.

— Si une anomalie est détectée, le serveur peut révoquer les clés de session ou interdire temporairement un dispositif de participer au réseau.

— Toutes les clés de session sont réinitialisées périodiquement pour renforcer la sécurité du réseau et empêcher toute exploitation de failles potentielles.

3.5.3 Etablissement d'une session de communication entre un Serveur S, un Utilisateur U et un Nœud N

Une session de communication entre un utilisateur U, un serveur S, et un nœud N dans un contexte sécurisé inspiré de la blockchain et de l'Internet des objets (IoT) :

1. **Étape d'authentification initiale** :

   — Utilisateur U : Pour initier la communication, l'utilisateur doit s'authentifier auprès du serveur S. Il envoie ses informations d'identification (par exemple, un identifiant et une signature numérique générée par sa clé privée).

   — Serveur S : Le serveur vérifie l'identité de U en validant la signature numérique avec la clé publique de l'utilisateur. Si l'identité de U est validée, le serveur génère un jeton de session temporaire unique et le renvoie à U.

2. **Étape d'authentification du nœud** :

   — Serveur S : Une fois l'utilisateur U authentifié, le serveur établit un lien sécurisé avec

le nœud N qui fera partie de la session de communication. Le serveur envoie une demande d'authentification à N pour s'assurer qu'il est un nœud de confiance dans le réseau.

— Nœud N : Le nœud répond avec sa propre signature numérique, validée par le serveur S pour confirmer son authenticité.

3. **Établissement de la session sécurisée** :

   — Serveur S : Après validation, le serveur crée une session commune sécurisée, dans laquelle U et N sont liés. Le serveur peut générer une clé de session partagée pour chiffrer les échanges entre U et N.

   — Utilisateur et Nœud : L'utilisateur U et le nœud N reçoivent chacun la clé de session ou un certificat d'autorisation, signé par S, qui leur permet de vérifier la légitimité des communications dans la session.

4. **Communication sécurisée et validation des transactions** :

   — Utilisateur U : Il peut maintenant transmettre des informations au nœud N via le serveur S. Chaque message envoyé est signé numériquement avec sa clé privée, garantissant l'authenticité des données.

   —Nœud N : Il utilise la clé de session partagée ou la signature numérique pour vérifier l'intégrité et l'authenticité des messages reçus de U. Toutes les transactions ou commandes envoyées sont validées par S et éventuellement enregistrées dans la blockchain pour assurer traçabilité et transparence.

5. **Fermeture de la session** :

   — Fin de la session : Lorsque la session est terminée, le serveur S révoque le jeton de session et invalide la clé de session partagée.

A) OUTILS DE SIMULATION

A.1) Python 3.11.3

Python est un langage de programmation de haut niveau, interprété et orienté objet, réputé pour sa polyvalence et sa facilité d'utilisation. Créé par Guido van Rossum et publié pour la première fois en 1991, Python se démarque par une syntaxe claire et lisible, ce qui le rend accessible aux

débutants et apprécié des développeurs expérimentés.

Python est un langage polyvalent, adapté à de nombreux domaines comme le développement web, le développement de logiciels, l'automatisation, l'analyse de données, et l'intelligence artificielle. Étant un langage interprété, Python exécute le code ligne par ligne grâce à un interpréteur, ce qui facilite le débogage et les tests interactifs. Il est également orienté objet, permettant ainsi la structuration du code en classes et objets pour une meilleure modularité. Grâce à sa bibliothèque standard riche et à un écosystème étendu, Python offre une multitude de modules et de frameworks facilitant le développement d'applications complexes. Enfin, sa compatibilité multiplateforme permet son utilisation sur divers systèmes d'exploitation, rendant Python extrêmement adaptable pour un large éventail de projets.

A.11) Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code source gratuit et largement apprécié, développé par Microsoft. Léger, rapide et hautement extensible, VS Code s'adapte à de nombreux langages de programmation et technologies, et fonctionne sur les principales plateformes, notamment Windows, macOS et Linux. L'interface sobre et intuitive de VS Code crée un espace de travail ergonomique, offrant des fonctionnalités avancées et une prise en charge étendue pour divers langages, tels que C/C++, Java, PHP, Python, JavaScript, HTML/CSS, JSON, Markdown, Node.js, et bien plus encore. L'arborescence des fichiers est visible dans la colonne de gauche, et les fichiers peuvent être ouverts simultanément sous forme d'onglets. Plusieurs modes d'affichage permettent aux utilisateurs d'organiser leur espace de travail selon leurs préférences. VS Code intègre également des outils de recherche et de gestion de versions, dont Git, facilitant les commits et le suivi des modifications directement depuis l'éditeur. Parmi les points forts de VS Code se trouve le débogage intégré pour plusieurs langages, permettant aux développeurs de tester et corriger leurs applications sans quitter l'éditeur. L'écosystème d'extensions est un autre atout majeur : il permet aux utilisateurs d'ajouter des fonctionnalités supplémentaires, d'intégrer des outils tiers, et de personnaliser leur environnement pour mieux répondre à leurs besoins.

3.5.4 Phase de déroulement

a) Création des blocs objet

Chaque nœud possède une chaîne de blocs, et chaque bloc stocke des informations comme un index, des données, un hash et une référence au hash du bloc précédent.

Bloc : Chaque bloc contient un index, un timestamp, des données (par exemple, des mesures IoT), un hash calculé à partir du contenu du bloc, et le hash du bloc précédent.

Nœud : Le nœud est responsable de créer et gérer sa propre chaîne de blocs. Il peut ajouter de nouveaux blocs avec des données spécifiques.

Affichage de la chaîne : La méthode afficher chaine permet de visualiser tous les blocs d'un nœud avec leurs informations clés.

Le code va afficher une liste de blocs pour le nœud IoT, montrant les informations sur chaque bloc, y compris son index, timestamp, données, hash et le hash précédent.

```python
import hashlib
import time

# Classe pour représenter un Bloc
class Bloc:
    def __init__(self, index, data, previous_hash):
        self.index = index  # Position du bloc dans la chaîne
        self.timestamp = time.time()  # Temps de création du bloc
        self.data = data  # Données contenues dans le bloc (ex : données de capteur)
        self.previous_hash = previous_hash  # Hash du bloc précédent
        self.hash = self.calculate_hash()  # Calcul du hash du bloc actuel

    def calculate_hash(self):
        # Génère un hash unique basé sur le contenu du bloc
        block_string = str(self.index) + str(self.timestamp) + str(self.data) + str(self.previous_hash)
        return hashlib.sha256(block_string.encode()).hexdigest()

# Classe pour représenter un Noeud
class Noeud:
    def __init__(self, nom):
        self.nom = nom  # Nom ou identifiant du nœud
        self.chain = []  # Liste de blocs formant la chaîne du nœud

    def creer_bloc_initial(self):
        # Crée le premier bloc de la chaîne, sans lien vers un bloc précédent
        bloc_initial = Bloc(index=0, data="Bloc Initial", previous_hash="0")
        self.chain.append(bloc_initial)

    def ajouter_bloc(self, data):
        # Ajoute un nouveau bloc avec les données fournies
        dernier_bloc = self.chain[-1]
        nouveau_bloc = Bloc(index=dernier_bloc.index + 1, data=data, previous_hash=dernier_bloc.hash)
        self.chain.append(nouveau_bloc)
```

```
    def afficher_chaine(self):
        # Affiche la chaîne de blocs
        for bloc in self.chain:
            print(f"Index: {bloc.index}")
            print(f"Timestamp: {time.ctime(bloc.timestamp)}")
            print(f"Données: {bloc.data}")
            print(f"Hash: {bloc.hash}")
            print(f"Hash Précédent: {bloc.previous_hash}\n")

# Création d'un nœud et ajout de blocs à la chaîne
noeud1 = Noeud("Nœud IoT 1")

# Initialiser la chaîne avec le bloc initial
noeud1.creer_bloc_initial()

# Ajouter des blocs avec des données
noeud1.ajouter_bloc("Mesure Température: 22°C")
noeud1.ajouter_bloc("Mesure Humidité: 60%")
noeud1.ajouter_bloc("Mesure Pression: 1012 hPa")

# Afficher la chaîne de blocs
noeud1.afficher_chaine()
```

FIGURE 3.2 – Création des blocs d'objets

```
Index: 0
Timestamp: Mon Oct 28 11:15:11 2024
Données: Bloc Initial
Hash: 4e2437d12a3788c8dd5bf7f1b1768bab5f62f0204122ea0a17b45e7e350d6db9
Hash Précédent: 0

Index: 1
Timestamp: Mon Oct 28 11:15:11 2024
Données: Mesure Température: 22°C
Hash: 4730ab3b50059d31d8bbddee3e35723de05dd43b5685b3f9a9e201c74ca8c2b7
Hash Précédent: 4e2437d12a3788c8dd5bf7f1b1768bab5f62f0204122ea0a17b45e7e350d6db9

Index: 2
Timestamp: Mon Oct 28 11:15:11 2024
Données: Mesure Humidité: 60%
Hash: 008a39e8f0c776cfad6d2487a33bf17e4c94acedb4440d5deb6d582fc1f5bf76
Hash Précédent: 4730ab3b50059d31d8bbddee3e35723de05dd43b5685b3f9a9e201c74ca8c2b7

Index: 3
Timestamp: Mon Oct 28 11:15:11 2024
Données: Mesure Pression: 1012 hPa
Hash: ba758d7392963faa0968fc2a4816cfc5b4581214bc0b1942da5454bf5f4b5555
Hash Précédent: 008a39e8f0c776cfad6d2487a33bf17e4c94acedb4440d5deb6d582fc1f5bf76
```

b) Générée une paire de clef

Le nœud va faire une demande aux nœuds voisins leurs clé publique tout en assurant une communication de sécurité pour l'IoT basé sur la blockchain.

En effet, chaque nœud va envoyer leur clef de façon d'un point a un autre pour authentifier.

```python
from cryptography.hazmat.primitives.asymmetric import ec
from cryptography.hazmat.primitives import serialization

# Générer une clé privée ECC
private_key = ec.generate_private_key(ec.SECP256R1())  # Courbe elliptique SECP256R1 (par exemple)

# Générer la clé publique correspondante
public_key = private_key.public_key()

# Afficher la clé privée (format PEM)
private_pem = private_key.private_bytes(
    encoding=serialization.Encoding.PEM,
    format=serialization.PrivateFormat.PKCS8,
    encryption_algorithm=serialization.NoEncryption()
)

# Afficher la clé publique (format PEM)
public_pem = public_key.public_bytes(
    encoding=serialization.Encoding.PEM,
    format=serialization.PublicFormat.SubjectPublicKeyInfo
)

print("Clé Privée ECC:")
print(private_pem.decode('utf-8'))

print("Clé Publique ECC:")
print(public_pem.decode('utf-8'))
```

FIGURE 3.3 – Générée une paire de clé ECC en python

```
Clé Privée ECC:
-----BEGIN PRIVATE KEY-----
MIGHAgEAMBMGByqGSM49AgEGCCqGSM49AwEHBG0wawIBAQQgzgxv3PCt1GtLhIsC
54nyIjXLPz65rfgfCnVXKYpKRTWhRANCAARj9TVKhvvYdn3jNbvqw1NdkT+hq0rc
8iyKWBJ8D/t6f+IukGEu7SlVqqlEoFPmQVGgXCLJa3P1IhHZufPI9rWq
-----END PRIVATE KEY-----

Clé Publique ECC:
-----BEGIN PUBLIC KEY-----
MFkwEwYHKoZIzj0CAQYIKoZIzj0DAQcDQgAEY/U1Sob72HZ94zW76sNTXZE/oatK
3PIsilgSfA/7en/iLpBhLu0pVaqpRKBT5kFRoFwiyWtz9SIR2bnzyPa1qg==
-----END PUBLIC KEY-----
```

c) Chiffrement du message (Ciphertext)

Pour chiffrer un message dans un scénario IoT, nous pouvons utiliser un chiffrement asymétrique (pour échanger des clés de session) ou un chiffrement symétrique (pour des messages plus grands ou des sessions continues). En effet, on va montrer comment chiffrer un message avec une clé publique en utilisant le chiffrement asymétrique RSA, ainsi que la manière de le déchiffrer

avec la clé privée correspondante.

— Génération de la clé RSA : Nous générons une clé privée et en dérivons la clé publique pour le chiffrement.

— Chiffrement avec OAEP Padding : Le schéma de padding OAEP (Optimal Asymmetric Encryption Padding) est utilisé pour sécuriser le chiffrement RSA.

— Déchiffrement : Seul le détenteur de la clé privée peut déchiffrer le message chiffré.

Ce processus garantit que le message reste confidentiel et ne peut être lu que par le destinataire prévu, assurant la confidentialité des communications dans un environnement IoT.

```python
from cryptography.hazmat.primitives.asymmetric import rsa
from cryptography.hazmat.primitives.asymmetric import padding
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives import serialization

# Générer une paire de clés RSA (pour l'exemple)
private_key = rsa.generate_private_key(public_exponent=65537, key_size=2048)
public_key = private_key.public_key()

# Message à chiffrer
message = b"Message important pour l'utilisateur IoT"

# Chiffrer le message avec la clé publique
ciphertext = public_key.encrypt(
    message,
    padding.OAEP(
        mgf=padding.MGF1(algorithm=hashes.SHA256()),
        algorithm=hashes.SHA256(),
        label=None
    )
)

print("Message chiffré:", ciphertext)

# Déchiffrer le message avec la clé privée
plaintext = private_key.decrypt(
    ciphertext,
    padding.OAEP(
        mgf=padding.MGF1(algorithm=hashes.SHA256()),
        algorithm=hashes.SHA256(),
        label=None
    )
)

print("Message déchiffré:", plaintext.decode())
```

FIGURE 3.4 – Chiffrement et Déchiffrement du message

```
Message chiffré: b'\x192}4\x13d[?\xc751\xbb\x02\xa0b\x1d\x9fs\xb0j\t\x8c#\xb51\r)c\xa1z}:\xe0kv\xeev\xc65B\xcd\xd5x8\xd3\xc0\xce;\xb9F\xac\x89\xad\xe6-\x
8e\x8e\x9dY\t\x01\xd0\xfe\x0b\xc8\x16r\xcf\x8b\xd3Yf\xe9-J\x80,\x8f\x0f\xa1Y\xc5\xe4]\xa7\xc9\xa2gR\xb1\xe2u\x80o\xfd8\xe9\x80\x9e&>\x9f\xd2Xy\xfd\x12N\x
e46\x95Nn\xbaJQ\x04\xd6\x95\xe6\x94\x87\xe2\x15\xc8\xa6)@#\xb2U\x05\xc4\xce.\xc9\xd7}+\x8b\xce\xaf\xea\xdfp1yx\x16.\xa9\x9bW\x0b\xe1\x11.\x1a\xa9*R\x82\x
df\x96\xb2\xdb"\x01\xe0I-\x951\x1b\xce\xc0v\x90\x88<\x16}L\xba\xdc\x0e\xe3R&\x90\xcf\xb6[\x9cs_*\xac>\xa0\xc44\x17u\x0e"\xb6\xb3\x81\xf5\x89\n\xc3m\x06\x
b2d\xcfz\xbc\xdd\xa2\x0b\x03\x12O\x9c\x02\x0c\xfd\x06\x00\xc5\x03\xd1\x93\xac\xd5\x07q\x81\xfe\xd9\x82\x9e\xa6\xb0i9\x9c\x18\xe1\xdf\xfbxn'
Message déchiffré: Message important pour l'utilisateur IoT
```

d) Authentification des nœuds voisins

Dans un environnement IoT sécurisé basé sur la Blockchain, une demande d'authentification aux nœuds voisins consiste généralement en une vérification mutuelle d'identité entre un nœud émetteur et ses voisins avant d'établir une session de communication. Cela garantit que seuls les nœuds autorisés peuvent échanger des informations. Ce type de processus renforce la confidentialité et l'intégrité des échanges en minimisant le risque d'accès par des entités non autorisées.

— Clés ECC : Chaque nœud génère sa propre paire de clés ECC pour signer et vérifier les messages.

— Signature de la demande : $N1$ signe sa demande avec sa clé privée, ce qui garantit que seule la clé publique correspondante (connaissable de $N2$ et $N3$) peut valider la signature.

— Vérification de la demande : Le nœud $N2$ utilise la clé publique de $N1$ pour vérifier la signature.

Si la vérification réussit, l'authenticité est confirmée et une session de communication peut être initiée.

Cette approche d'authentification entre nœuds voisins permet de bâtir un réseau IoT sécurisé en assurant que seules les entités de confiance participent aux échanges.

```python
from cryptography.hazmat.primitives.asymmetric import ec
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives.asymmetric.utils import encode_dss_signature, decode_dss_signature

# Générer les clés ECC pour les nœuds
private_key_n1 = ec.generate_private_key(ec.SECP256R1())
public_key_n1 = private_key_n1.public_key()

private_key_n2 = ec.generate_private_key(ec.SECP256R1())
public_key_n2 = private_key_n2.public_key()

# Initialisation de la demande d'authentification
message = b"Demande d'authentification de N1 avec nonce et timestamp"
signature = private_key_n1.sign(
    message,
    ec.ECDSA(hashes.SHA256())
)

# Vérification de l'authenticité par N2
try:
    public_key_n1.verify(signature, message, ec.ECDSA(hashes.SHA256()))
    print("Authentification réussie : N1 est un nœud de confiance.")
except:
    print("Échec de l'authentification : N1 n'est pas reconnu.")
```

FIGURE 3.5 – Authentification des nœuds voisins

```
Authentification réussie : N1 est un nœud de confiance.
```

e) Signature numérique

Un message est signe à l'aide de sa clé privée avec un algorithme. — Génération de la clé : Une paire de clés ECC est générée pour signer et vérifier.

— Création du hachage : Le message est transformé en bytes (UTF-8), puis le hachage SHA-256 est calculé.

— Signature : La signature utilise le hachage et la clé privée de l'utilisateur.

Cela garantit que seules les parties possédant la clé privée peuvent générer une signature valide, permettant ainsi de garantir l'authenticité du message.

```python
from cryptography.hazmat.primitives import hashes
from cryptography.hazmat.primitives.asymmetric import ec
from cryptography.hazmat.primitives.asymmetric.utils import Prehashed
from cryptography.hazmat.primitives import serialization
from cryptography.hazmat.backends import default_backend

# 1. Génération d'une paire de clés ECC
private_key = ec.generate_private_key(ec.SECP256R1(), default_backend())
public_key = private_key.public_key()

# 2. Message à signer
message = "Ce message doit être signé".encode('utf-8')

# 3. Calcul du hachage du message avec SHA-256
digest = hashes.Hash(hashes.SHA256(), backend=default_backend())
digest.update(message)
message_hash = digest.finalize()

# 4. Signature du hachage avec la clé privée
signature = private_key.sign(
    message_hash,
    ec.ECDSA(Prehashed(hashes.SHA256()))
)

# Affichage de la signature en hexadécimal
print("Signature:", signature.hex())
```

FIGURE 3.6 – Signature numérique

```
Signature: 304502205f250f6b96b492b76059e46dde7bb6bf3a37734780c3123d43019838b9a8212c02
La signature est valide.
```

e) Vérification de hachage

Pour vérifier un hachage, on compare le hachage original avec celui calculé à partir des mêmes données. Si on trouve le même résultat donc le message n'a pas été modifie sinon le message a été modifié.

— Encodage du message : La fonction encode le message en bytes pour le rendre compatible avec hashlib.

— Calcul du hachage : Le hachage est calculé avec SHA-256 et converti en format hexadécimal

pour une lecture simplifiée.

— Retour du hachage : La fonction retourne le hachage sous forme de chaîne hexadécimale.

Il est utilisé pour assurer l'intégrité des données dans un système, garantissant que toute modification du message entraînera un hachage différent.

```python
import hashlib

def calculer_hachage(message):
    # Encoder le message en bytes
    message_bytes = message.encode('utf-8')
    # Calculer le hachage SHA-256
    hachage = hashlib.sha256(message_bytes).hexdigest()
    return hachage

# Exemple d'utilisation
message = "Ceci, abcd, efgh09"
print("Hachage SHA-256 du message :", calculer_hachage(message))
```

FIGURE 3.7 – Vérification de hachage

```
Hachage SHA-256 du message : e72923f56c987339caa966f2761818f9ad0785f2c899969cf18801732f733c58
```

Modification du message

Une modification du message en cours de transmission pourrait être tentée par un attaquant. Dans ce cas, le contenu du message serait altéré, générant une empreinte (hachage) différente de l'empreinte d'origine. Même un léger changement dans le message entraîne une transformation complète de son hachage, rendant toute altération immédiatement détectable grâce à cette signature unique.

```python
import hashlib

def calculer_hachage(message):
    # Encoder le message en bytes
    message_bytes = message.encode('utf-8')
    # Calculer le hachage SHA-256
    hachage = hashlib.sha256(message_bytes).hexdigest()
    return hachage

# Exemple d'utilisation
message = "Ceci, abcd, efgh24"
print("Hachage SHA-256 du message :", calculer_hachage(message))
```

FIGURE 3.8 – Message modifié

```
Hachage SHA-256 du message : 56b00a9e69d5aa78a703710c4683fe340a1bc2b9496becfdcf2cf0623f8fd5da
```

3.6 Conclusion

Dans ce chapitre, nous avons exposé l'architecture proposée ainsi que le protocole de confidentialité et d'intégrité destiné aux systèmes IoT. Notre approche de sécurité s'inspire de la technologie Blockchain, qui offre une résistance notable face aux attaques. Après une analyse approfondie, nous pouvons conclure que notre solution est adaptée aux défis posés, en particulier en ce qui concerne les menaces ciblant l'IoT.

CONCLUSION GÉNÉRALE

L'Internet des Objets (IoT) est en train de redéfinir notre façon de vivre et d'interagir avec notre environnement. En reliant des millions de dispositifs intelligents, l'IoT permet une collecte et une analyse de données sans précédent, ouvrant la voie à des innovations dans des domaines variés tels que la santé, l'industrie, les transports et les villes intelligentes.

Cependant, cette transformation numérique s'accompagne de défis majeurs, notamment en matière de sécurité, de confidentialité et d'intégrité des données. La prolifération de dispositifs connectés expose les utilisateurs à des risques de violations de données et de cyberattaques, ce qui soulève des questions critiques sur la confiance et la protection des informations personnelles. Pour relever ces défis, il est impératif de mettre en place des infrastructures de sécurité robustes et des normes régissant l'utilisation des technologies IoT. Les solutions innovantes, telles que l'intégration de la blockchain, peuvent jouer un rôle clé dans l'amélioration de la sécurité et de la transparence des systèmes IoT. La technologie Blockchain représente une avancée significative dans le domaine de la gestion des données et des transactions numériques. En offrant un cadre décentralisé, transparent et immuable, la Blockchain transforme la manière dont les informations sont stockées, vérifiées et échangées. Grâce à ses caractéristiques uniques, elle assure une meilleure sécurité, une plus grande confiance et une traçabilité accrue des transactions dans divers secteurs, allant de la finance à la logistique, en passant par la santé et l'Internet des Objets (IoT).

Cependant, malgré son potentiel immense, la Blockchain fait face à des défis notables, notamment en matière de scalabilité, d'interopérabilité et de consommation d'énergie. Pour que cette technologie réalise pleinement son potentiel, il est crucial de développer des solutions innovantes qui répondent à ces problématiques tout en respectant les principes fondamentaux de décentralisation et de sécurité.

La montée en puissance de l'Internet des Objets (IoT) révolutionne la manière dont nous interagissons

avec notre environnement, mais elle soulève également des défis significatifs en matière de sécurité, de confidentialité et d'intégrité des données. Dans ce contexte, l'intégration de la technologie Blockchain se révèle être une solution prometteuse, offrant des mécanismes robustes pour renforcer la protection des systèmes IoT.

Dans ce mémoire, nous avons aborder 3 chapitres a savoir dans le premier chapitre sur la généralité de l'Internet des objets, dans le deuxième chapitre sur la technologie de la blockchain et enfin le dernier chapitre sur établissement de sessions IoT sécurisées par la blockchain.

Ainsi, nous pouvons concevoir dans le cas réel l'utilisation de l'Intellgence Artificielle dans l'IoT basé sur la technologie de la Blockchain.

Bibliographie

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things : A survey. *Computer Networks*, 54(15) :2787–2805, 2010.

[2] Alessandro Bassi, Martin Bauer, Martin Fiedler, Thorsten Kramp, Rob van Kranenburg, Sebastian Lange, and Stefan Meissner, editors. *Enabling Things to Talk*. Springer Berlin Heidelberg, Berlin, Heidelberg, 2013.

[3] M. Belotti, N. Božic, G. Pujolle, and S. Secci. A vademecum on blockchain technologies : When, which, and how. *IEEE Communications Surveys Tutorials*, 21(4) :3796–3838, 4 2019. 4th Quarter.

[4] J. Bonneau, A. Narayanan, A. Miller, J. Clark, J. A. Kroll, and E. W. Felten. Mixcoin : Anonymity for bitcoin with accountable mixes. In *Proceedings of the International Conference on Financial Cryptography and Data Security*, pages 486–504, Berlin, Heidelberg, 2014. Springer.

[5] G. Eysenbach. What is e-health ? *Journal of Medical Internet Research*, 3(2), juin 2001.

[6] Futura. Internet des objets : définition et explications, juin 2021. Consulté le 19 octobre 2024.

[7] Actualité Informatique. Qu'est-ce que l'iot (internet of things – internet des objets), juin 2021. Consulté le 19 octobre 2024.

[8] ITU. ITU-T Recommendation Y.4000/Y.2060 : Overview of the Internet of things, juin 2012.

[9] N. Kopal. Solving classical ciphers with cryptool 2, 2018. Consulté le 19 octobre 2024.

[10] Maciej Kranz. *Building the Internet of Things : Implement New Business Models, Disrupt Competitors, Transform Your Industry*. John Wiley & Sons, 2016.

[11] Le Monde. La première blockchain de l'histoire date de 1995 et elle est imprimée sur papier, septembre 2018. Consulté le 19 octobre 2024.

[12] A. Lohachab, A. Lohachab, and A. Jangra. A comprehensive survey of prominent cryptographic aspects for securing communication in post-quantum iot networks. *Internet of Things*, 9 :100174, 2020.

[13] M. Löf. Decentralized transactions in a centralized environment : A blockchain study within the transport industry, 2017. Consulté le 19 octobre 2024.

[14] Cristian Martín, Jaime Chen, Enrique Soler, Manuel Díaz, and Ana Reyna. On blockchain and its integration with iot : Challenges and opportunities. *Future Generation Computer Systems*, 88 :173–190, nov 2018.

[15] M. B. Mollah, J. Zhao, D. Niyato, K.-Y. Lam, X. Zhang, A. M. Y. M. Ghias, L. H. Koh, and L. Yang. Blockchain for future smart grid : A comprehensive survey. *IEEE Internet of Things Journal*, 8(1) :18–43, jan 2021.

[16] Satoshi Nakamoto. Bitcoin : A peer-to-peer electronic cash system, 2008. Consulté le 19 octobre 2024.

[17] Satoshi Nakamoto. Bitcoin v0.1 released. *Mail Archive*, 9, 2009. Consulté le 19 octobre 2024.

[18] A. Schneider. Decentralization : Conceptualization and measurement. *Studies in Comparative International Development*, 38(3) :32–56, sep 2003.

[19] Claude E. Shannon and Warren Weaver. *The Mathematical Theory of Communication*. University of Illinois Press, Urbana and Chicago, 1949.

[20] Nick Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2(9), 1997. Consulté le 19 octobre 2024.

[21] D. Tapscott and A. Tapscott. *Blockchain Revolution : How the Technology Behind Bitcoin is Changing Money, Business, and the World.* Penguin, Baltimore, MD, USA, 2016.

[22] P. Zheng, Z. Zheng, J. Wu, and H.-N. Dai. Xblock-eth : Extracting and exploring blockchain data from ethereum. *IEEE Open Journal of Computer Society*, 1 :95–106, 2020.

[23] T. Zonta, C. A. da Costa, R. da Rosa Righi, M. J. de Lima, E. S. da Trindade, and G. P. Li. Predictive maintenance in the industry 4.0 : A systematic literature review. *Computers & Industrial Engineering*, 150 :106889, 2020.