الجمهورية الجزئرية الدَيموقراطية الشَعبية
République Algérienne Démocratique et Populaire
وزارة التَعليم العالي والبحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة البليدة 1
Université de BLIDA 1
كليَة العلوم
Faculté des Sciences
قسم الإعلام الآلي
Département d'Informatique

Mémoire de Fin d'Études

En vue d'obtention du diplôme de Master

Option : Sécurité des Systèmes d'Information
&
Système Informatique et Réseau

Conception et développement d'un schéma d'authentification basé sur la technologie blockchain dans l'Internet des Objets

Réalisé par :

Mounir HABABOU
Hichem BENHAMMOU

Organisme d'accueil :

Présidente :	Mme.N. TOUBALINE	(USDB)
Examinateur :	Mr. Y. DOUGA	(USDB)
Promotrice :	Mme. S. AROUSSI	(USDB)
Encadreur :	Mr. H. KHEMISSA	(CERIST)

Remerciement

Tout d'abord, nous remercions ALLAH de nous avoir donné la patience et l'énergie pour réaliser ce travail.

Nos plus sincère remerciements pour notre Promotrice Madame Aroussi Sanaa pour la confiance que vous nous avez accordée, et les conseils et remarques que vous nous avez donner, Et de nous avoir orienté, corrigé notre travail et encouragé, nous avoir orienté. Nous avons acquis beaucoup de connaissance et savoir-faire bénéfique au cours de nos discussions

Nous tenant à remercie tout particulièrement Monsieur Khemissa hamza attaché de recherche à CERIST et promoteur de thèse, de nous avoir orienté.

Nous aimerions exprimer notre gratitude à tous les chercheurs et spécialistes, [les membres de jurée] qui ont pris le temps de discuter des différent problème et solution de notre sujet. Chacun de ces échanges nous a aidé à faire avancer notre projet.

Et nous remercions nos amis qui nous ont aidés avec des idées et des encouragements, nous citons notamment cher ami : Mehdi Nacer KERKAR.

Résumé

L'évolution de l'Internet des objets (Internet of Things, IoT) transforme les perceptions traditionnelles de l'Internet existant en un concept d'appareils intelligents qui interagissent les uns avec les autres. Les réseaux de capteurs sans fil jouent un rôle clé et prennent en charge divers domaines d'application IoT comme la e-santé. Les problèmes de sécurité sont, cependant,undes obstacles à leur utilisation, l'authentification des différentes entités interconnectées et la gestion des entités lors de l'authentification est une partie majeure de ce problème. Dans notre travail, nous proposons un schéma d'authentification compatible avec la faible puissance de calcul des appareils IoT connectés et ses ressources limitées. Notre schéma est basé sur la technologie blockchain qui offre un haut niveau de sécurité et assure la sécurité des identités du nœud participant dans le réseau. Les analyses de sécurité effectuées montrent que le schéma a une résistance contre plusieurs attaques telles que l'usurpation d'identité et l'attaque Dos.
Mots-clés : Internet des objets (IoT), E-santé, authentification, Blockchain.

Abstract

The evolution of the Internet of Things (IoT) transforms traditional perceptions of the existing Internet into a concept of smart devices that interact with each other. Wireless sensor networks play a key role and support various IoT application domains such as the e-health. Security concerns are, however, an obstacle to their use, the authentication of different interconnected entities among these problems, and the management of identities during authentication is a major part of this problem, in our work we propose a authentication scheme which is compatible with the low computing power of connected IoT devices and its limited resources, the scheme is based on Blockchain technology which offers a high level of security and ensures the security of the identities of the participating nodes in the network, Security analyzes we have done on our proposed scheme shows that the scheme has resistance against several attacks such as impersonation and Dos attack.

Keywords: Internet of things (IoT), E-health, authentication, Blockchain.

ملخّص

أدى تطور إنترنت الأشياء إلى تحويل التصورات التقليدية للإنترنت الحالي إلى مفهوم الأجهزة الذكية التي تتفاعل مع بعضها البعض. تلعب شبكات الاستشعار اللاسلكية دورًا رئيسيًا وتدعم مجالات تطبيقات إنترنت الأشياء المختلفة مثل العلاج الإلكتروني. ومع ذلك ، فإن المخاوف الأمنية هي عقبة أمام استخدامها ، ومصادقة الأجهزة المترابطة المختلفة من بين هذه المشاكل ، وإدارة الهويات أثناء المصادقة هي جزء رئيسي من هذه المشكلة ، في عملنا نقترح نظام مصادقة متوافق مع قوة الحوسبة المنخفضة للأجهزة المتصلة ومحدودية مواردها ، يعتمد المخطط على تقنية البلوكشين التي توفر مستوى عالٍ من الأمان وتضمن أمان هويات الأجهزة المشاركة في الشبكة ، والتحليلات الأمنية التي أجريناها على المخطط يُظهر أنَ المخطط المقترح لديه مقاومة ضد عدة هجمات مثل انتحال الهوية وهجوم الحرمان من الخدمة.

الكلمات المفتاحية: أنترنتالأشياء ،العلاجالإلكتروني،البلوكشين.

Table des matières

Liste des figures :

Liste des tableaux :

Liste des acronymes

IoT	Internet des objets
Dos	Denial of Service attack
TIC	Technologies de l'information et de la communication
RFID	Radio Frequency IDentification
SOA	Service-Oriented Architecture
SIS	Système d'Information Santé
SIH	Système d'Information Hospitalier
WIFI	Wireless Fidelity
XOR	eXclusive OR
WSN	Wireless Sensor Networks
ECC	Elliptic Curve Cryptography
SGBD	Système de gestion de base de données
RGPD	Règlement général sur la protection des données
E/S	Entrée / sortie
@ IP	Adresse Internet Protocol
@ MAC	Adresse Media Access Control
RSA	Rivest Shamir et Adleman (technologie de cryptage à clé publique)
AES	Advanced Encryption Standard
3DES	Triple Data Encryption Standard
DES	Data Encryption Standard
DSS	Digital Signature Standard
RC4	Rivest Cipher 4(algorithme de chiffrement)
BLE	Bluetooth Low Energy
WAN	Wide Area Network
PAP	Password Authentication Protocol
CHAP	Challenge-Handshake Authentication Protocol
MD5	Message Digest 5 (fonction de hachage cryptographique)
EAP	Extensible Authentication Protocol
MITM	man-in-the-Middle
POS	Proof-of-Stake
POW	Proof of Work
KYB	Know-Your-Business
KYC	Know-Your-Customer
POB	preuve de combustion
POC	preuve de capacité
POA	preuve d'activité
POE	preuve d'existence
POI	preuve d'intelligence
POL	preuve de chance
P2P	Peer-to-peer

ICP	Infrastructure de clés publiques
WSN	Wireless Sensor Network
DAPP	applications décentralisées
EVM	Ethereum Virtual Machine
API	Application Programming Interface
HTTP	Hypertext Transfer Protocol
ENS	Ethereum Name Service
POM	Project Object Model
ECC	Elliptic-Curve Cryptography
TRNG	True Random Number Generator
IoV	Internet of Vehicles
IoE	Internet of Energy
IoS	Internet of Sensors
IoMT	Internet of Medical Things
PUF	Physical Unclonable Function
TRNG	True Random Number Generator
TPM	Trusted Platform Module
TEE	Trusted Execution Environment
SFA	Single factor authentication
2FA	two-factor
MFA	multi-factor authentication
Md5	Message Digest 5
SHA	Secure Hash Algorithm
PBKDF2	Password-Based Key Derivation Function 2eme version

Introduction Générale

Récemment, un nouveau paradigme domine le monde des communications qui est l'internet des objets (Internet of Things, IoT) permettant aux objets physiques connectés ayant leur propre identité numérique de communiquer les uns avec les autres [1]. Ce réseau crée en quelque sorte une passerelle entre le monde physique et le monde virtuel. La plupart de ces objets IoT devraient être basés sur une technologie de communication sans fil à faible coût avec des capacités limitées en termes de calcul et de stockage[2][3]. Les systèmes IoT étant de plus en plus chargés de détecter et de gérer des écosystèmes très complexes, cela pose de nombreux défis de sécurité pour le développement de l'IoT dans tous les domaines tels que l'e-santé qui fait référence à l'utilisation des Technologies de l'Information et de la Communication (TIC) pour le transfert et l'échange à distance de données de santé. Dans notre travail, nous nous concentrons sur la sécurité de l'authentification dans le contexte IoT[4]. Le processus d'authentification doit garantir un canal sécurisé pour échanger des données entre les utilisateurs et les objets IoT et doit également garantir la sécurité des identités des nœuds participants dans le réseau de communication [5][6].

Pour atteindre notre objectif, nous avons utilisé un schéma d'authentification léger pour réseaux de capteurs sans fil hétérogènes dans le contexte de l'Internet des objets proposé par H. Khemissa et D. Tandjaoui dans [1]. Ce schéma est basé sur des nombres aléatoires, des fonctions de hachage pour assurer l'intégrité des messages échangés et de fonction de concaténation. Il vise à établir un canal sécurisé entre un nœud de capteur et une station de base dans une application e-santé classique [3]. Par ailleurs, les identités des appareils sont stockées à l'intérieur des utilisateurs et des passerelles, ce qui rend la sécurité des identités dépend de la sécurité d'un appareil, en augmentant ainsi le risque de les perdre et le risque de nombreuses cyberattaques telles que les attaques par déni de service, les attaques par usurpation d'identité, ainsi que les attaques liées à la base de données, la technologie Blockchain peut être une meilleure solution pour gérer le stockage des identités. En effet, le stockage Blockchain est un moyen de sauvegarder des données dans un réseau décentralisé, qui utilise l'espace disque inutilisé des utilisateurs sur le réseau pour stocker des données et cela peut offrir beaucoup d'avantages tels que : la décentralisation, la confiance, l'immuabilité, la transparence, la traçabilité, etc, Ces avantages peuvent aider à prévenir la plupart des cyberattaques mentionnées précédemment [7].

Notre contribution consiste à établir un schéma d'authentification basé sur la technologie blockchain dans l'Internet des Objets, et à implémenter un simulateur de réseau pour donner une meilleure vision à notre schéma proposé et pour pouvoir faire quelques analyses de sécurité nécessaires.

Ce présent mémoire est divisé en quatre chapitres :

- Dans le chapitre 1, nous allons présenter un état de l'art sur l'IoT, ses utilisations dans le domaine de l'e-santé et des approches de sécurité des appareils IoT.
- Dans le chapitre 2, nous présenterons également un état de l'art sur la technologie blockchain et les protocoles d'authentification, également un bref aperçu de l'utilisation de dans les processus d'authentification.

- Dans le chapitre 3, nous présenterons le schéma d'authentification proposé dans [1] et ensuite nous présenterons notre contribution.
- Dans le chapitre 4, nous présenterons l'implémentation de notre simulateur.

Enfin, nous terminerons par une conclusion de notre travail et quelques perspectives de notre contribution.

Chapitre 1 : La sécurité dans l'internet des objets et les applications e-santé.

L'Internet des objets (IoT) a beaucoup de potentiel pour améliorer de nombreux aspects liés à l'industrie des soins de santé. La connectivité des aux capteurs intelligents à l'internet dans les dispositifs médicaux peut aider les médecins à surveiller à distance leurs patients. Cela peut présenter l'avantage supplémentaire d'améliorer l'engagement et la satisfaction des patients grâce à plus de temps d'interaction avec leurs médecins. De plus, les hôpitaux commencent maintenant à utiliser l'IoT pour surveiller la sécurité et la santé de leurs patients et suivre leur historique médical. Le potentiel de l'IoT dans l'industrie médicale ne fait que commencer à se concrétiser [2].Dans ce chapitre, nous présentons les principaux concepts de l'internet des objets (IoT), des applications é-santé et la sécurité des applications e-santé dans l'IoT.

1. L'internet des objets

L'Internet des objets (IoT) est un nouveau paradigme qui progresse rapidement dans le scénario de communication sans fil moderne. Ce nouveau paradigme permet l'interaction de différents objets tels que des capteurs, des étiquettes d'identification par radiofréquence (RFID), les téléphones portables, etc, pour atteindre des objectifs communs [3][4].

1.1. Définition

L'Internet des objets signifie sémantiquement «un réseau mondial d'objets interconnectés adressable de manière unique, basé sur des protocoles de communication standard» [5]. Cela implique un grand nombre d'objets (hétérogènes) impliqués dans le processus. L'adressage unique des objets et la représentation et le stockage des informations échangées deviennent les problèmes les plus difficiles, amenant directement à une troisième perspective, «orientée sémantique», de l'IoT [3].

Dans la figure 1, les concepts, technologies et critères clés sont mis en évidence et catégorisés en référence visions IoT qui contribuent à leur meilleure caractérisation. De cette illustration, il apparaît clairement que le paradigme IoT devrait être le résultat de la convergence des trois perspectives principales.

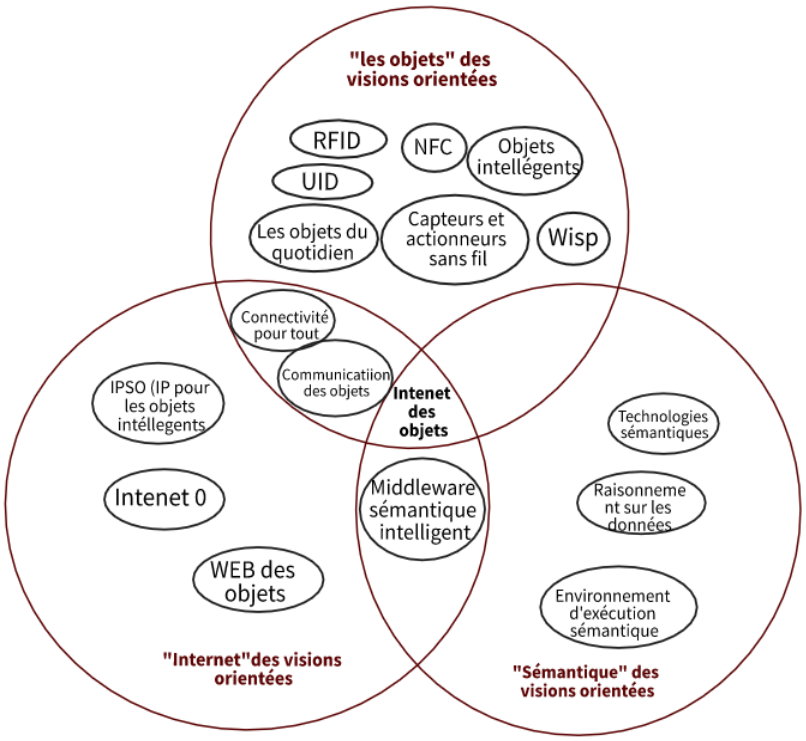

Figure 1 : Paradigme de «l'Internet des objets» suite à la convergence de différentes visions [3]

1.2. Technologies utilisées

Afin de mettre en œuvre le paradigme de l'IoT et de le rendre utilisable, les technologies sans fil jouent un rôle clé. Aujourd'hui, le rapport entre les radios et les humains approche la valeur de 1 pour 1 [6]. Cependant, ce rapport pourrait augmenter de plus en plus avec une réduction en termes de taille, de poids, de consommation d'énergie et de coût de la radio. Et cela augmentera l'utilisation du radios et le rendra plus utile dans de nombreuses applications comme les applications de e-santé, transports intelligents, maisons intelligentes, etc, qui sont constituées d'un ou plusieurs lecteurs et de plusieurs étiquettes RFID. Les balises sont caractérisées par un identifiant unique et sont appliquées aux objets [3].

Les systèmes RFID peuvent être utilisés pour surveiller des objets dans le temps réel, sans avoir besoin d'être en ligne, cela permet de cartographier le monde réel dans le monde virtuel. Par conséquent, ils peuvent être utilisés dans une gamme incroyablement large de scénarios d'application, allant de la logistique à l'e-santé et à la sécurité.

Les réseaux de capteurs joueront également un rôle crucial dans l'IoT. En fait, ils peuvent coopérer avec les systèmes RFID pour mieux suivre l'état des choses, c'est-à dire leur emplacement, leur température, leurs mouvements, etc [3].

Ainsi, ils peuvent augmenter la prise de conscience d'un certain environnement et, ainsi, agir comme un pont supplémentaire entre le monde physique et numérique. L'utilisation de réseaux de capteurs a été proposée dans plusieurs scénarios d'application, tels que la surveillance environnementale, la e-santé, le transport intelligent surveillance des systèmes, des installations militaires et industrielles [3].

Un réseau de capteurs est composé d'un certain nombre de nœuds communiquant de manière collaborative afin de transmettre les résultats à un nœud généralement particulier qui joue le rôle d'une station de base (voir Figure 2). Un réseau de capteurs dans un contexte IoT doit permettre une extensibilité, scalabilité et surtout optimiser la consommation d'énergie des différents nœuds qui le composent.

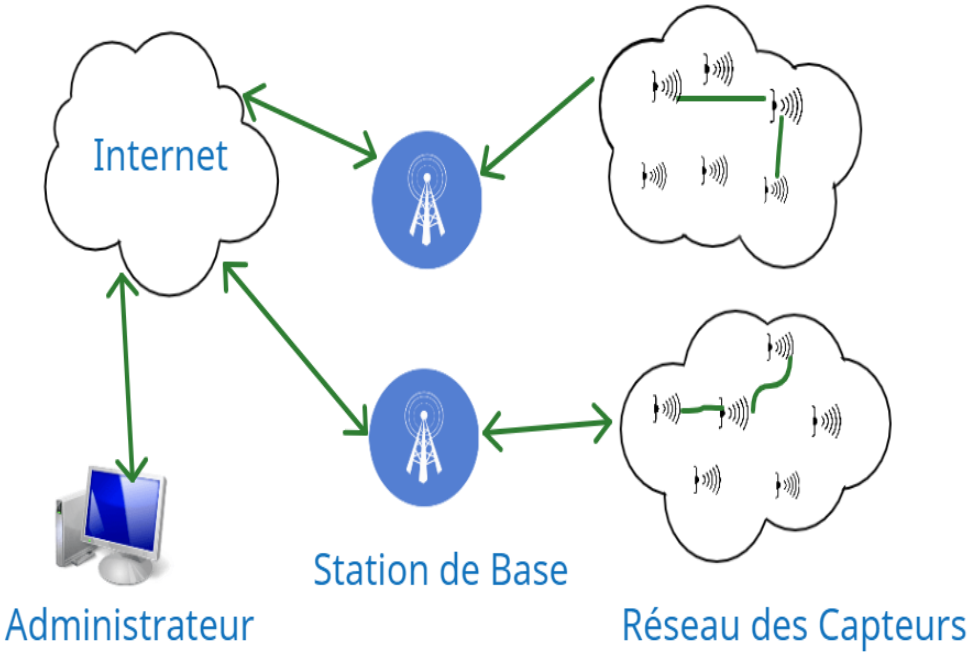

Figure 2:Architecture d'un réseau de capteurs sans fil [9]

1.3. Domaines d'utilisation

L'IoT est utilisé dans divers domaines comme [3]:

- **Domaine du transport et de logistique :** Certains moyens de transport tel que les voitures, les trains, et même les vélos sont équipés de capteurs et des lecteurs RFID et d'autre objets intelligents et même des routes sont équipés des étiquette RFID et des capteurs pour envoyer des information utiles et importantes au contrôle de circulation.

- **Domaine de la santé :** l'Internet des Objets (capteurs, et étiquettes intégrées..), permet un suivi en temps réel d'un patient avec la collecte et la détection automatiques des urgences, qui visent principalement à réduire le temps de traitement en fonction des résultats collectés et à automatiser les processus médicaux. Le suivi des patients devient plus facile, et l'identification et l'authentification des patients sont également très importantes car elles visent à réduire les incidents préjudiciables aux patients.

- **Domaine des environnements intelligents :** En utilisant des ensembles d'objets intelligents pour faciliter la vie quotidienne, tels que les maisons intelligentes, les villes intelligentes, les installations industrielles intelligentes, etc.

- **Domaine personnel et social :**Dans ce domaine, les applications permettent de créer et de développer des relations sociales, en effet, les choses peuvent déclencher automatiquement la transmission de messages à des amis pour leur faire savoir ce que nous faisons ou ce que nous avons fait dans le passé, comme les réseaux sociaux actuels, les jeux de console connectés, etc[3].

- **Domaine des applications futuristes :** Les applications mentionnées précédemment sont déjà déployées et disposent de toutes les ressources nécessaires, mais il y a des applications que restent à déployer dans le futur étant donné que leur mise en œuvre est complexe, Parmi ces applications : le taxi robot, Modèle d'information sur la ville, etc.

2. Les applications e-santé

La e-santé, ou santé électronique, décrit l'ensemble des moyens et services liés à la santé qui utilisent les nouvelles technologies de l'information et de la communication. L'e-santé fait appel à Internet, aux applications pour Smartphones et aux objets connectés,

L'avenir de e-santé s'appuie sur des capteurs implantés dans ou autour du corps humain, connectés à la station de base qui collecte des données privées sensibles liées à la santé des nœuds capteurs via des canaux sécurisés, puis transmet les données collectées aux soignants [1].

2.1. Le Système de Santé Electronique

L'e-santé inclut (figure 3) [10]:

- Les SIS (Système d'Information Santé) et les SIH (Système d'Information Hospitalier) qui constituent la base d'informations digitale de l'E-santé en milieu hospitalier et permet l'organisation des flux d'informations, dossiers médicaux ou encore la gestion des cartes vitales en interne des établissements médicaux.

- La télé-santé qui comprend les actes de prévention et de soins réalisés à distance : information via des portails grand public, sites de promotion de la santé, systèmes d'alerte téléphonique, prescriptions électroniques à distance, et la télé-santé regroupe :
  - La télémédecine (actes médicaux réalisés à distance par un médecin) : consultation par vidéoconférence, téléassistance d'un médecin lors d'une intervention, télésurveillance du patient, télé-expertise (échange des avis des médecins).
  - La m-santé (mobile santé) qui comprend les applications numériques pour Smartphones ou objets connectés (bracelets...) en lien avec la santé. [10].

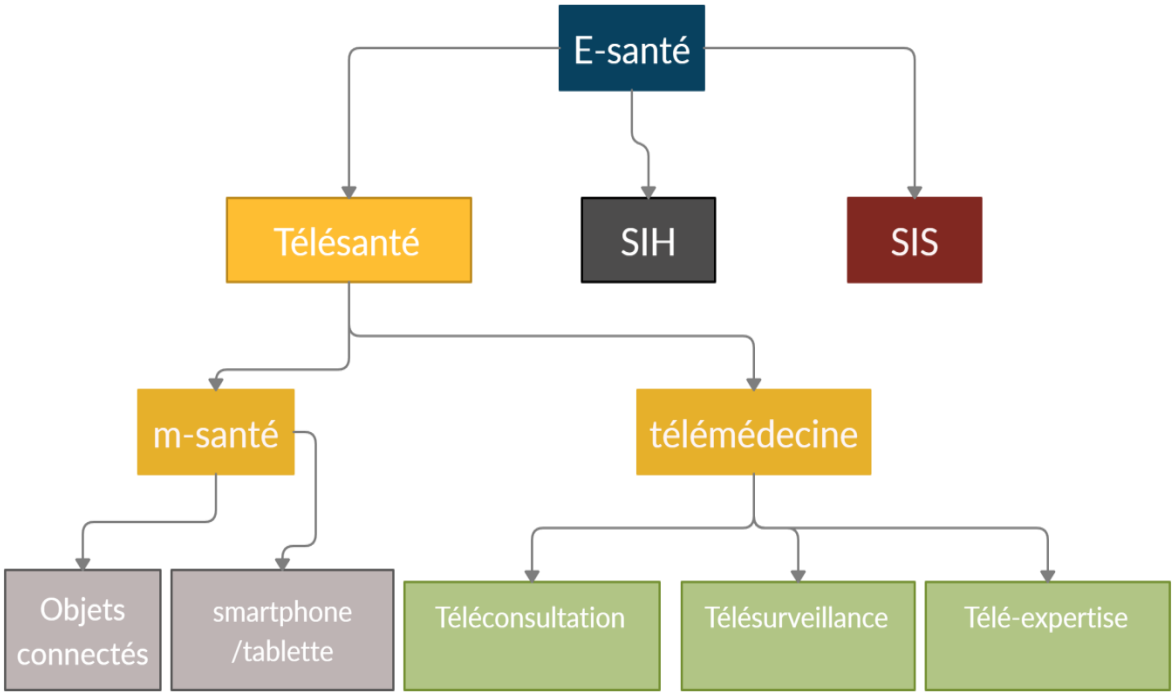

Figure 3:Organisation de l'e-santé [10]

L'internet des objets a rapidement transformé la prestation de soins. Les équipements et les capteurs sont de plus en plus « intelligents » et génèrent toujours plus de données nécessaires aux équipements médicaux, aux professionnels et profitant ainsi aux patients, en réduisant les coûts et en améliorant leur satisfaction. Les données ainsi collectées facilitent, adaptent, améliorent, anticipent ou réorganisent les soins des patients.

Dans le contexte d'application é-santé, l'Internet des objets est fondamental. En effet, la conception d'un système intelligent de prise de décision clinique, matérialisé par le stockage des données collectées sur les patients et leur accessibilité universelle, procurerait au médecin un excellent appui durant la phase de traitement (voir figure 4)[11].

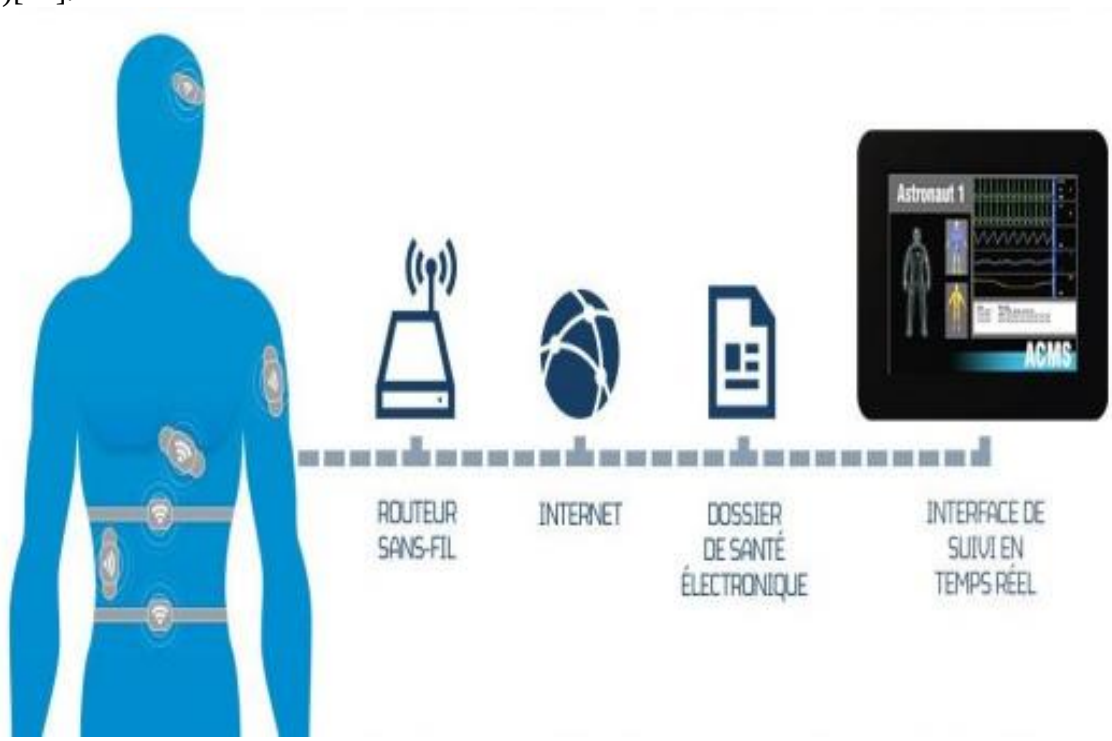

Figure 4: un système de santé électronique [11]

2.2. Attaques dans les applications é-santé

L'Internet des objets prend en charge de nombreuses applications d'e-santé, ce qui signifie que toute attaque qui menace l'IoT menace également des applications e-santé.

L'IoT est vulnérable à un nombre considérable d'attaques. Il existe diverses attaques sur des schémas d'authentification d'utilisateurs distants tels que le dictionnaire, men-in-the-middle, le texte en clair, la carte à puce perdue, la modification, le déni de service (DOS), la divulgation de clé de session, l'emprunt

d'identité, l'initié, etc. Ces attaques peuvent être gênantes pour un utilisateur légitime lors de l'accès à un système dans un but spécifique.

Formellement, une menace est une cause potentielle d'incident, qui peut résulter en un dommage à l'ensemble des systèmes d'information du domaine de la santé (et pas seulement de la médecine, le champ d'application étant ainsi très large) incluant les méthodes et technologies d'exploitation et d'analyse des données collectées à partir de ces systèmes d'information variés et diversifiés qui peut toucher :

Confidentialité : La confidentialité consiste à rendre l'information intelligible à d'autres personnes que les seuls acteurs de la transaction.

L'intégrité : Vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle).

La disponibilité : L'objectif de la disponibilité est de garantir l'accès à un service ou à des ressources.

L'authentification : L'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. Un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.

Les attaques qui menacent les applications e-santé (système d'information) peuvent être classées en 2 catégories:

Attaques passives : consistent à écouter sans modifier les données ou le fonctionnement du réseau, elles sont généralement indétectables mais une prévention est possible.

Attaques actives : consistent à modifier des données ou des messages, à s'introduire dans des équipements réseau ou à perturber le bon fonctionnement de ce réseau. Noter qu'une attaque active peut être exécutée sans la capacité d'écoute. De plus, il n'y a généralement pas de prévention possible pour ces attaques, bien qu'elles soient détectables (permettant ainsi une réponse adéquate).

Dans le tableau 1, nous présentons les principales attaques IoT dans les applications e-santé, notamment [11]:

- l'attaque de dictionnaire qui tente de deviner des mots de passe communs basés sur le dictionnaire.
- l'attaque men-in-the-middle qui est implémentée pour reconnaitre l'information.
- l'attaque en clair qui est utilisée lorsque le texte chiffré est volé.

- l'attaque perdue de carte à puce qui est introduite lorsqu'une carte à puce est perdue, puis un attaquant peut appliquer des procédures pour acquérir l'information.
- l'attaque de modification qui est implémenté pour modifier les informations en d'autres termes, l'attaquant modifie les informations puis retransmet les données à nouveau.

Nom de l'attaque	But et résultat de l'attaque	Menace	Active ou Passive
Dos	- Saturer un serveur ou bloquer le trafic. - Rendre un service non disponible.	- Intégrité. - Disponibilité. - Confidentialité.	Active
Man-in-the-Middle	- Intercepter les communications entre deux parties contrôler la conversation. - écouter, modifier ou supprimer des données.	- Intégrité. -Confidentialité.	Active
L'usurpation d'identité	- Vol d'identité. -Réaliser des actions frauduleuses. -Prendre délibérément l'identité d'une autre personne vivante.	-Confidentialité. -Authentification.	Active
Flooding	- épuiser la mémoire et l'énergie des nœuds. - Saturer le réseau.	- Intégrité. - Disponibilité.	Active
Wardriving	- Utilisée pour pouvoir accéder à internet au nom d'une autre personne. - Parcourir tous les lieux ou le Wifi est déployée afin de découvrir toutes les bornes Wifi existantes noter l'adresse géographique.	-Confidentialité.	Passive
Sniffing	- Capturer les trames circulent local et afficher leur contenus (entêtes des sur un réseau protocoles, id des	-Confidentialité.	Passive

	user, MDP non crypté, etc.).		

Tableau1 : principales attaques IoT dans les applications l'e-santé [1]

3. La sécurité des applications e-santé dans l'IoT

Les systèmes IoT étant de plus en plus chargé de détecter et de gérer des applications e-santé, des questions sur la sécurité et la fiabilité des données transmises vers et depuis les appareils IoT devient rapidement un enjeu majeur préoccuper(figure 5)[8][12].

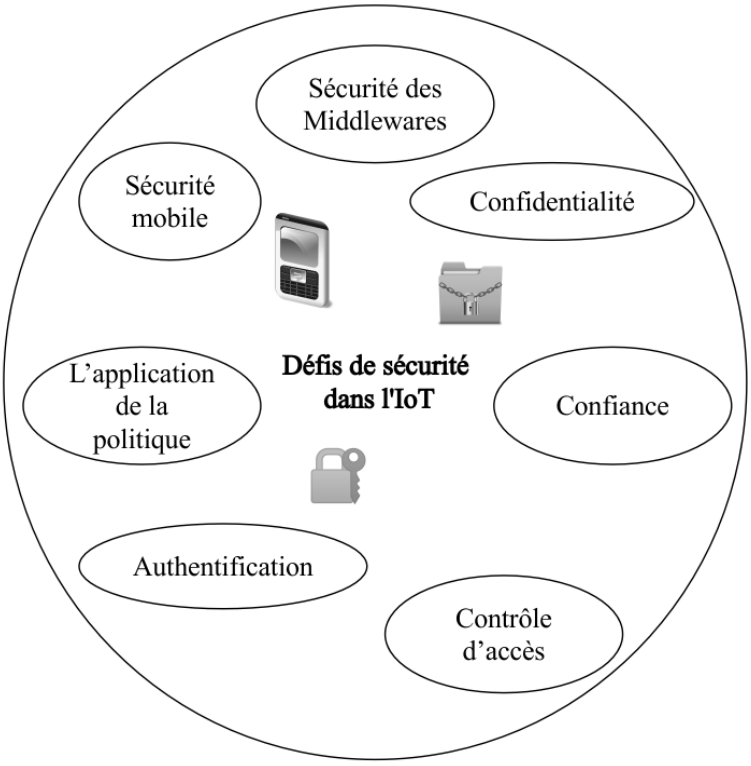

Figure 5: Principaux problèmes de sécurité dans l'IoT [13]

De plus, les systèmes IoT ont une puissance de calcul limité, ce qui signifie que les contre-mesures de sécurité traditionnelles et l'application de la protection de confidentialité ne peuvent pas être utilisées pour sécuriser les systèmes et technologies IoT[14]. Donc, pour offrir une meilleure sécurité et pour gagner l'acceptation des utilisateurs, il est nécessaire de définir de meilleurs modèles de

sécurité, de confidentialité et de confiance qui aident à sécuriser les systèmes et applications IoT[15].

En effet, il est nécessaire de définir un meilleur moyen de sécuriser l'intégrité des données de l'utilisateur (des objets) ainsi que son identité et sa confidentialité afin de prévenir l'accès et les utilisations illégales des données[16]. Il est nécessaire de mettre en place un mécanisme d'authentification et d'autorisation fort et adapté (pour la puissance de calcul des systèmes IoT), concernant la confidentialité, les appareils peuvent gérer les informations sensibles (par exemple, les habitudes des utilisateurs) donc la confidentialité doit être assurée avec un bon mécanisme, enfin, la confiance est un problème fondamental de données car les données des utilisateurs sont manipulées et utilisées par différents appareils pour garantir leurs droits et leurs besoins[17].

Dans ce qui suit, nous présentons les exigences de sécurité IoT qui doivent être garantir dans les applications e-santé comme : authentification, contrôle d'accès, confidentialité et confiance[18].

3.1. Authentification

L'authentification est le processus de vérification de l'identité d'un utilisateur, d'un processus ou d'un appareil. La vérification est basée sur un facteur d'authentification, qui peut dépendre de plusieurs facteurs [19] :

a) Connaissance : Quelque chose dont l'utilisateur (ou toute entité qui doit être authentifiée) est informé (par exemple, les mots de passe et les clés pré-partagées).

b) Propriété : Quelque chose qui appartient à l'utilisateur, comme les éléments qu'il possède (par exemple, les cartes à puce).

c) Biométrique : Quelque chose sur l'utilisateur, comme les caractéristiques physiologiques et biométriques (par exemple, le visage, les empreintes digitales et la voix).

d) Comportement : Une activité prise par un utilisateur ; un type d'authentification qui vérifie les identités par l'observation de comportements (par exemple, des gestes ou des contacts).

e) Emplacement : Quelque part où se trouve l'utilisateur ; c'est-à-dire les informations de localisation de l'utilisateur (par exemple, GPS, adresse IP).

Dans un contexte IoT, tous les dispositifs IoT doivent pouvoir confirmer leur identification afin de se voir accorder des capacités d'exploitation particulières (par exemple, envoyer/recevoir des informations) ou accéder aux ressources du système d'information. Un ou plusieurs des facteurs d'authentification décrits ci-dessus peuvent être utilisés comme preuve d'identité (informations d'identification), qui est souvent basée sur des informations d'identification similaires conservées dans un

emplacement sécurisé. L'authentification (et donc la preuve d'identité) peut être effectuée sur l'ensemble de l'appareil, ou sur des éléments individuels de l'appareil, pour vérifier qu'aucun composant tiers potentiellement dangereux n'est lié au système[20]. Nous détaillerons ce processus d'authentification dans le chapitre suivant.

3.2 Contrôle d'accès

Le contrôle d'accès est un élément fondamental de la sécurité des données qui détermine qui est autorisé à accéder aux informations et aux ressources de l'entreprise et à les utiliser. Grâce à l'authentification et à l'autorisation, les politiques de contrôle d'accès garantissent que les utilisateurs sont bien qu'ils prétendent être et qu'ils ont un accès approprié aux données de l'entreprise.

Il y a les détenteurs de données et les collecteurs de données que les détenteurs de données (utilisateurs et objets) devraient être identifiés et authentifiés par les collecteurs de données pour leur donner le droit d'alimenter les collecteurs de données avec les données dont ils ont besoin [13].

Dans IoT, il est également nécessaire de traiter des flux de données non seulement des données discrètes (sur les bases de données traditionnelles), le contrôle d'accès pour un flux de données est plus intensif en calcul que dans le SGBD traditionnel (Système de gestion de base de données), dans les flux de données les requêtes doivent être exécuté directement sur les flux entrants, et cela nécessite plus d'énergie de calcul, et beaucoup de travaux sont réalisés dans ce domaine.

3.3. Confidentialité

Le nouveau règlement général européen sur la protection des données (RGPD) étant devenu exécutoire le 25 mai 2018 [21], la protection des données des utilisateurs et la sécurisation de la confidentialité des utilisateurs sont urgentes et prédominantes à résoudre pour toute application IoT. Les données des utilisateurs ne peuvent pas être capturées ni utilisées à leur insu. La confidentialité a la plus haute priorité pour tous les développements d'applications existantes et futurs, y compris les systèmes IoT. Les identités des utilisateurs doivent ne pas être identifiable ni traçable. En vertu de la nouvelle législation, le traitement des données doit impliquer:

1) Traitement légal, équitable et transparent - mettant l'accent sur la transparence pour les personnes concernées.

2) Limitation de la finalité - avoir une fin légale et légitime pour le traitement des informations dans le premier endroit.

3) Minimisation des données - s'assurer que les données sont adéquates, pertinentes et limitées, et que les organisations sont capturant suffisamment la quantité minimale de données nécessaires pour atteindre l'objectif spécifié.

4) Traitement précis et à jour - obligeant les responsables du traitement à s'assurer que les informations restent exactes, valides et adaptées à l'usage prévu.

5) Limitation du stockage sous une forme permettant l'identification - décourageant les données inutiles redondance et réplication

6) Confidentiel et sécurisé - protéger l'intégrité et la confidentialité des données en s'assurant qu'elles sont sécurisées (qui s'étend aux systèmes informatiques, aux enregistrements papier et à la sécurité physique)

7) Responsabilité - la démonstration de la conformité. En tant que déclaration bien connue en matière de sécurité, il existe des problèmes de sécurité à toutes les couches de perception, de réseau et d'application.

Un autre problème de sécurité peut être résolu de manière efficace et efficiente à un certain niveau de couche, tel que dans la mise en œuvre du composant de confidentialité sur la couche d'application. Dans un système de santé, les patients doivent être totalement conscients de qui collecte et utilise leurs données. Ils devraient également avoir les contrôles sur les données et avec qui ils veulent partager, comment et où leurs données sont utilisées. Les applications doivent fournir services et interface pour permettre aux utilisateurs de gérer leurs données. Les utilisateurs doivent disposer des outils qui leur permettant de conserver leur anonymat dans ce monde super-connecté. Le même scénario peut être appliqué à des systèmes tels que maison intelligente, transport intelligent, etc. Les applications IoT peuvent collecter les informations et données personnelles des utilisateurs de leurs activités quotidiennes. Beaucoup de gens considéraient que les données ou informations prédites à partir des données comme privées. L'exposition de ces informations pourrait avoir un impact indésirable ou négatif sur leur vie. L'utilisation du système IoT ne devrait pas causer de problèmes de fuite de confidentialité. Toutes les applications IoT qui ne répond pas avec ces exigences de confidentialité pourrait être interdites par la loi. Le système IoT doit sérieusement considérer la mise en œuvre de la confidentialité selon les 7 principes de protection des données, fournissant un soutien centré sur l'utilisateur pour la sécurité et la vie privée à partir de ses propres fondations [22].

3.4. Confiance

La confiance est une condition préalable dans l'environnement IoT car elle est largement distribuée et fiable sur des données qualitatives [23]. En fait, le concept de confiance est une question importante et il est utilisé dans différents contextes et il a des significations différentes, c'est une notion difficile sans consensus clair dans la littérature scientifique.[24].

Dans les technologies de l'information et de la communication (TIC), la confiance est décrite dans une variété des significations potentielles et est considérée comme une dimension cruciale des interactions numériques en intégrant la confiance dans les machines et les humains [25]. l' IoT n'est pas une exception, la confiance est la mesure dans laquelle un utilisateur peut faire confiance à l'environnement auquel il appartient et aux utilitaires avec lesquels il travaille. Par conséquent, la confiance dans l' IoT est la capacité de l'objet à servir un utilisateur sans causer de préjudice à l'utilisateur ou

ses données. L'utilisateur peut comprendre les services distribués impliqués, ainsi que le concept de sécurité et de résilience face aux menaces. [26].

La confiance est également connue comme le niveau de confiance qu'une entité peut garantir à d'autres entités pour des services spécifiques dans un contexte particulier Bien que la confiance soit fréquemment utilisée en référence à des personnes, il peut également être lié à un appareil ou à tout système, qui soulignent l'importance de mesurer le niveau de confiance dans une communauté numérique. [26].

En conséquence, la confiance dans l'IoT est intégrée dans trois couches: utilisateur à appareil, à partir d'un appareil à un autre, et d'un appareil à un utilisateur [25]. Ainsi, la confiance peut être fragmentée en confiance d'entité, confiance de machine et confiance de données. La confiance de la machine indique la nécessité d'interagir avec des dispositifs fiables tels que des actionneurs et des capteurs. C'est un défi dans l'environnement IoT car il n'est pas toujours possible d'établir la confiance dans les appareils. [27].

L'Internet des objets a introduit de nouveaux défis par rapport à ceux déjà existants, la réputation est liée à la confiance, et elle peut être définie comme une mesure de confiance dans laquelle une entité conserve des informations dignes de confiance sur d'autres entités, résultant en un « réseau » de confiance [25].

L'hétérogénéité est l'un des facteurs les plus importants à prendre en compte lorsqu'il s'agit de la confiance IoT. L'hétérogénéité vient de l'idée de l'Internet des objets (IoT), dans laquelle les appareils IoT interagissent avec le monde physique à travers une variété de choses qui ne communiquent que via une interface [28].

À mesure que le nombre d'appareils connectés à l'IoT augmente, la quantité de communications, de transactions et de données augmente également. Par conséquent, les systèmes de confiance doivent évoluer avec le nombre croissant d'appareils et le développement de mécanismes d'application prenant en charge l'évolutivité est essentiel, et de nouvelles approches doivent être examinées en fonction de leur capacité à gérer un nombre croissant d'éléments dans le réseau. [29].

La gestion des identités est également un aspect important de l'Internet des objets, tout comme les systèmes de confiance et de réputation. Le fait que l'identité des objets ne soit pas la même que le mécanisme de base, que les objets puissent avoir une identité de base et de nombreuses identités différentes, et que ces objets puissent aussi cacher leur véritable identité sont tous des éléments importants de cette difficulté. [30].

De plus, garantir que les exigences de confiance sont satisfaites est lié au contrôle d'accès et à la gestion des identités, Il s'agit d'un problème critique car l'environnement IoT est défini par une variété d'appareils qui doivent traiter et manipuler les données conformément aux souhaits et aux droits des utilisateurs. Le contrôle de l'état du monde virtuel doit être pris en charge. Les utilisateurs doivent pouvoir contrôler leurs services, ainsi que d'avoir accès à des outils qui caractérisent précisément toutes leurs interactions, afin de développer une cartographie mentale précise de leur environnement virtuel. [31].

4. Conclusion

Ce chapitre se composait de deux parties : La première partie a été consacrée à la présentation de l'IoT et ses domaines d'application, et dans la deuxième partie, nous avons défini l'usage des applications IoT sur le domaine E-santé et présenté quelques exigences de sécurité dans les applications IoT.Dans ce qui suit, nous nous concentrons sur le processus d'authentification dans les applications IoT, et l'utilisation de la technologie blockchain dans ce contexte.

Chapitre 2: Authentification et Blockchains dans l'Internet des objets

L'authentification est difficile dans l'IoT car elle nécessite des infrastructures qui ne seront pas disponibles dans les scénarios IoT. Dans ce chapitre, nous allons commencer par définir le concept et les attaques courantes sur l'authentification dans l'IoT. Ensuite, nous présentons les principaux concepts de blockchain.

1. L'authentification dans l'internet des objets

L'authentification est un processus, par lequel un système informatique certifie l'identité d'une personne (objet) ou d'un ordinateur (login,@ email, @ IP, @ MAC, etc.). Le but de ce processus étant d'autoriser la personne (objet) à accéder à certaines ressources sécurisées, il va comparer les informations des utilisateurs autorisés stockées dans une base de données (en local ou sur un serveur d'authentification) à celles fournies, l'accès sera autorisé seulement si les informations sont identiques [32].

1.1. Le rôle de l'authentification dans la sécurité IoT

L'authentification IoT est un modèle permettant de renforcer la confiance dans l'identité des machines et des appareils IoT, de protéger les données et de contrôler l'accès lorsque les informations transitent par un réseau non sécurisé tel qu'Internet. L'authentification nécessaire pour les appareils et les machines IoT connectées peut être fiable pour se protéger contre les commandes de contrôle d'utilisateurs ou d'appareils non autorisés et aide à empêcher les attaquants de se faire passer pour des appareils IoT dans l'espoir d'accéder aux données sur les serveurs telles que les conversations enregistrées, les images et autres informations potentiellement sensibles[3],[31].

En effet, l'authentification permet d'éviter différents types de cyber-attaques, comme [31]:

- Hameçonnage (Phishing) :L'attaquant emploie une liste de numéros de téléphone ou d'adresses e-mail pour envoyer un message enjoignant le destinataire à agir urgemment. (Par exemple, l'utilisateur peut être invité à se connecter pour vérifier des transactions.) Il redirige généralement l'utilisateur vers un site Web factice afin d'obtenir son nom d'utilisateur et son mot de passe.
- Harponnage (Spearphishing) :L'attaquant cible un petit nombre de personnes à l'aide de messages bien rédigés et crédibles qui les concernent directement, souvent au moyen de contenu personnalisé (nom de l'utilisateur, action ou

évènement récemment survenu, etc.) À l'instar du hameçonnage, il enjoint les utilisateurs à agir afin qu'ils dévoilent leurs identifiants.

- Remplissage d'identifiants(Credentialstuffing): L'attaquant tire parti du fait que les utilisateurs emploient souvent les mêmes identifiants sur plusieurs comptes en tentant d'utiliser des combinaisons de nom d'utilisateur/mot de passe volées pour accéder à plusieurs sites et applications.
- Attaques de force brute et force brute inverse : L'attaquant emploie un programme permettant de générer des noms d'utilisateur/mots de passe potentiels pour tenter d'accéder à une ressource. (Les attaques par dictionnaire constituent un type d'attaque en force brute). L'attaquant peut également tenter d'utiliser les mots de passe les plus couramment employés (de type « Password123 ») sur différents comptes.
- Attaques de l'intercepteur (MITM) : Le programme de l'attaquant s'insère dans les communications entre l'utilisateur et une application (par exemple en usurpant un réseau Wi-Fi public). Il recueille ensuite les identifiants de connexion saisis par l'utilisateur, voire détourne le jeton de session.

1.2. Classification de l'authentification IoT

Il existe plusieurs méthodes par lesquelles nous pouvons obtenir une authentification forte pour sécuriser les communications des appareils IoT. Les méthodes, appelés aussi schéma d'authentification peuvent être classifiées selon plusieurs critères comme indiqué dans la (figure 7) [32]. Ils peuvent être classifié selon:

- **La couche IoT :**

Où la méthode d'authentification est implémentée. Dans cette classification, nous considérons l'architecture IoT la plus courante qui se compose de trois couches (1) couche de perception, (2) couche réseau ou (3) couche application. Cependant, cette classification peut être adaptée à n'importe quelle architecture en couches pour l'IoT.

- **Le domaine d'application**

Différents schémas d'authentification sont utilisés pour les différents domaines d'application ou environnement IoT : Machine to Machine (M2M), Internet des véhicules (IoV), Internet de l'énergie (IoE), Internet des capteurs (IoS) et Internet des objets médicaux (IoMT).

- **Le matériel**

L'authentification basée sur le matériel utilise les caractéristiques physiques du matériel pour traiter l'authentification. Sur la base de ce critère, on peut distinguer entre les solutions matérielles implicites qui utilisent le matériel « existant » lors de

l'authentification (par exemple, la fonction physique non clonable (PUF) [43] ou le générateur de vrais nombres aléatoires (TRNG)[44]) et, des solutions matérielles explicites qui nécessitent l'utilisation d'un composant supplémentaire dédié aux opérations (cryptographiques ou autres) effectuées lors de l'authentification (par exemple, Module de plate-forme de confiance (TPM)[45] , Environnement d'exécution de confiance (TEE)[46]).

- **Le facteur d'authentification**

Selon le nombre des facteurs pris en compte pour authentifier un appareil, une solution peut être classée comme une authentification à un facteur (SFA), une authentification à deux facteurs (2FA) ou une authentification à plusieurs facteurs (MFA). Par exemple, si seul le mot de passe utilisateur est nécessaire, le schéma d'authentification est appelé schéma à facteur unique. Un schéma d'authentification à deux facteurs peut utiliser un mot de passe utilisateur et une carte à puce pour authentifier les utilisateurs. Une authentification multi facteur peut utiliser des facteurs supplémentaires tels que des informations de localisation, biométriques, etc [47].

- **L'accès utilisateur**

Différentes méthodes sont utilisées pour l'accès utilisateur, certaines utilisent la cryptographie à clé publique et peuvent être basées sur l'utilisation de certificats numériques (avec certificat) ou uniquement sur la paire de clés publique/privée (Sans certificat), d'autres les solutions reposent sur l'utilisation de clés pré-partagées. Des solutions hybrides mixant les deux usages existent également.

- **L'algorithme cryptographique**

Les algorithmes cryptographiques utilisés lors de la phase d'authentification peuvent également être utilisés comme critère de classification. Certains mécanismes d'authentification reposent uniquement sur des algorithmes symétriques compte tenu de leur faible surcoût par rapport aux algorithmes asymétriques. Dans cette catégorie de solutions, on peut distinguer les schémas d'authentification utilisant des algorithmes symétriques traditionnels et ceux utilisant des algorithmes symétriques légers qui ont été introduits pour les appareils contraints tels que les objets IoT. Une autre catégorie de solutions repose uniquement sur la cryptographie asymétrique pendant la phase d'authentification et peut être divisée en celles utilisant des algorithmes traditionnels (par exemple, RSA) et celles reposant sur la cryptographie à courbe elliptique (ECC). Une troisième catégorie de solutions repose sur l'utilisation de fonctions de hachage compte tenu de leur nature légère. Enfin, des solutions hybrides mélangeant deux ou toutes les méthodes mentionnées ci-dessus existent également.

- **Procédure d'authentification**

La procédure d'authentification peut être une authentification unidirectionnelle, bidirectionnelle ou tripartite à trois voies. Dans l'authentification unidirectionnelle, une seule des deux entités souhaitant communiquer s'authentifiera auprès de l'autre entité. Dans une authentification bidirectionnelle, les deux entités souhaitant communiquer se vérifieront mutuellement. Ce type d'authentification est également appelé authentification mutuelle. Dans l'authentification à trois voies, les deux entités communicantes sont authentifiées à l'aide d'une troisième entité appelée authentification centrale. L'entité centrale authentifie les deux entités à l'aide d'une authentification mutuelle.

- **Architecture d'authentification**

Deux architectures d'authentification sont utilisées pour traiter la procédure d'authentification, architecture centralisée et distribuée (décentralisée). Dans une architecture distribuée, les deux entités communicantes sont authentifiées à l'aide d'une authentification directe. Dans une architecture centralisée, les entités communicantes sont authentifiées à l'aide d'une troisième entité centralisée de confiance, qui partage et gère les identifiants entre les entités pour la procédure d'authentification. Dans les deux architectures, la structure peut être hiérarchique ou plate. Hiérarchique utilisé différents niveaux et plat n'a pas de structure de niveau [48].

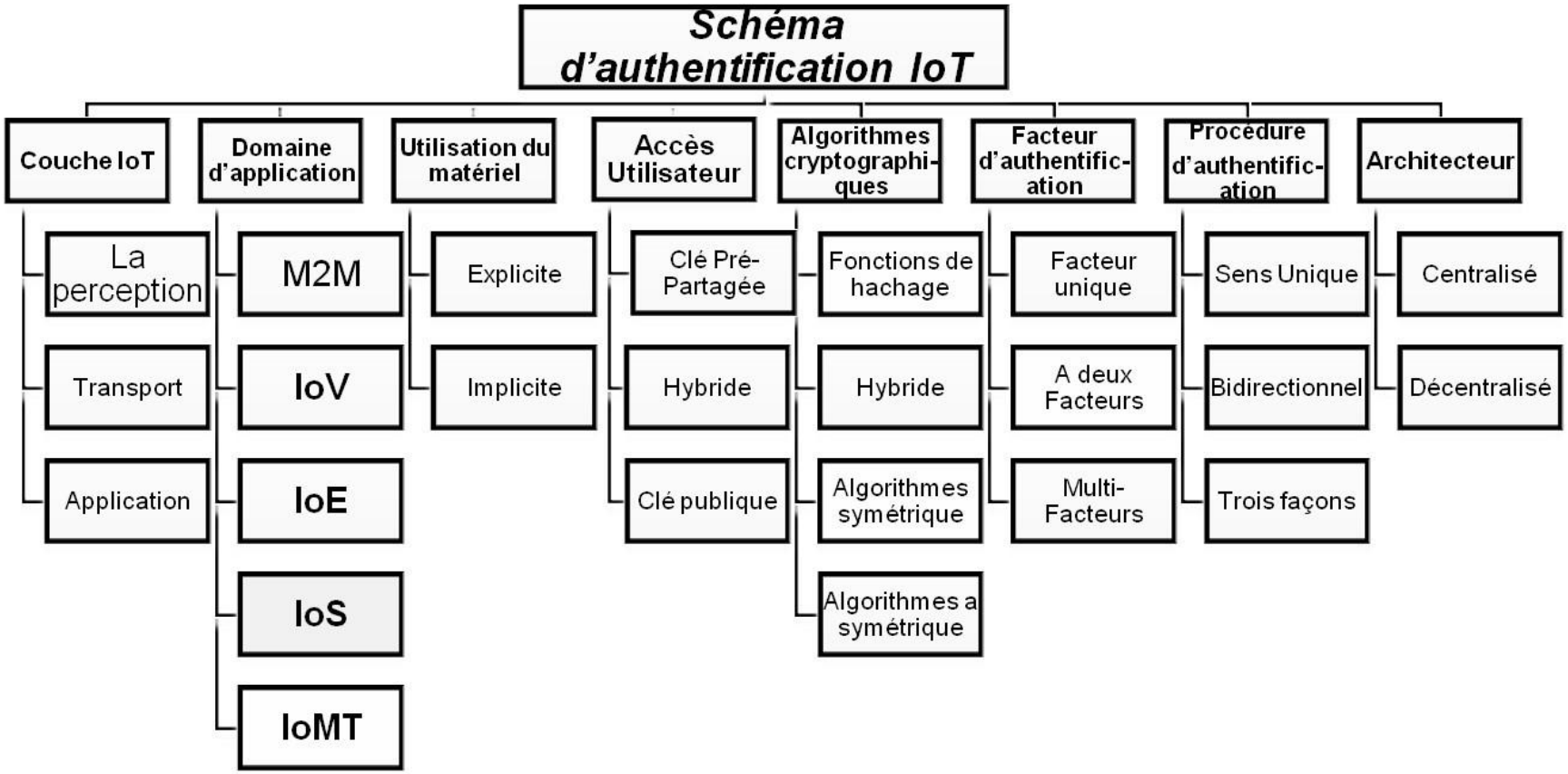

Figure 6:Taxonomie des schémas d'authentification IoT [32]

Dans notre travail, nous nous intéressons au schéma d'authentification proposé dans [1] qui est basé sur les nombres aléatoires, les fonctions HMAC pour assurer l'intégrité des messages échangés plus la fonction OU exclusif (Xo) et utilise aussi la fonction de concaténation. Ce schéma d'authentification vise à fournir une authentification mutuelle et à établir une communication sécurisée entre un nœud de capteur et un utilisateur distant pour des réseaux de capteurs sans fil hétérogènes dans le contexte de l'IoT. A l'aide de la figure 7, nous allons classer ce schéma dans le tableau suivant :

Couche IoT	Domaine D'application	Utilisation du matériel	Accès Utilisateur	Algorithme Cryptographique	Procédure d'authentification	Facteur d'authentification	Architecture
Application	Internet des Capteurs	Implicite	Clé publique	Symétrique	Bidirectionnel	A deux facteurs	Centralisé

Tableau2 : Taxonomie de Schéma dans [1]

Dans la prochcaine section nous présenterons le schéma d'authentification qui nous a inspiré pour réaliser notre travail.

2. Le schéma d'authentification léger pour réseaux de capteurs sans fil hétérogènes dans le contexte de l'IoT [1]

Dans cette section, nous présentons le schéma d'authentification proposé dans [1]. Nous commençons par présenter l'architecture du réseau. Ensuite, nous présentons le fonctionnement du schéma et ses avantages en termes de performances et de sécurité.

2.1. Architecture de réseau

L'architecture réseau est principalement composée des nœuds de capteurs, du nœud de passerelle et de l'utilisateur distant comme illustré dans la figure 11.
Le système de communication proposé permet de transmettre les données collectées à partir d'un nœud de capteur directement à l'utilisateur distant mobile après une authentification mutuelle réussie entre un nœud de capteur et l'utilisateur distant.

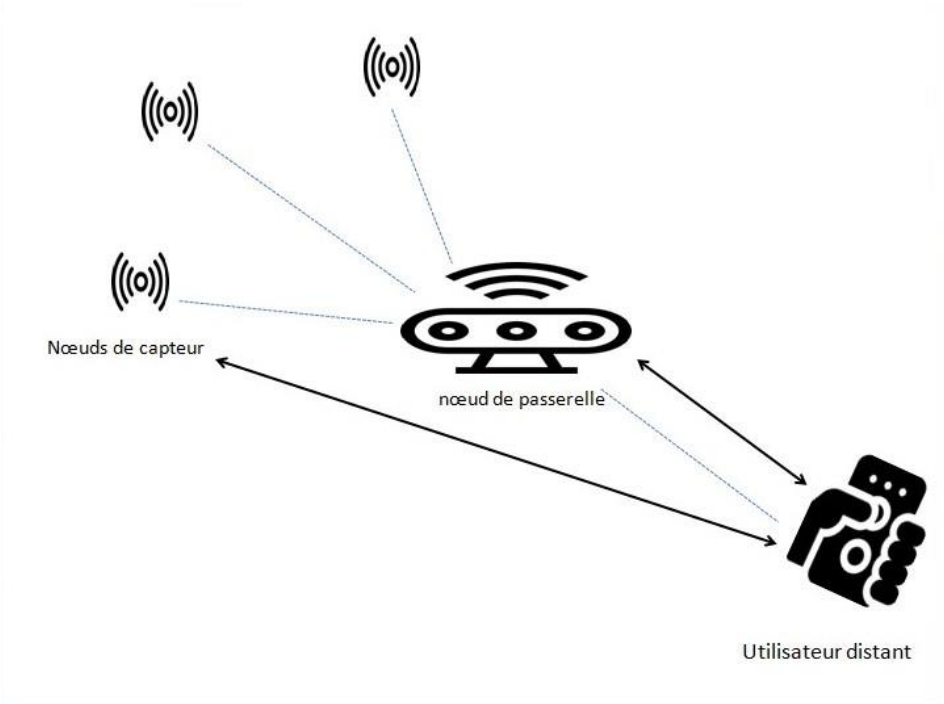

Figure 7 : Architecture de réseau [1]

2.2. Fonctionnement de schéma

Ce schéma fournit une authentification mutuelle entre un nœud de capteur et l'utilisateur distant d'un réseau de capteur sans fils (WSN). Il est divisé en trois phases : enregistrement, authentification et l'établissement de clés partagées.

2.2.1. La phase d'enregistrement

La phase d'enregistrement entre les nœuds de capteur, le nœud de passerelle et l'utilisateur distant est importante dans le fonctionnement de ce schéma. Cette phase est divisée en deux parties. Le nœud capteur et le nœud passerelle sont impliqués dans la première partie. Le nœud de passerelle et l'utilisateur distant terminent la deuxième partie d'enregistrement. .

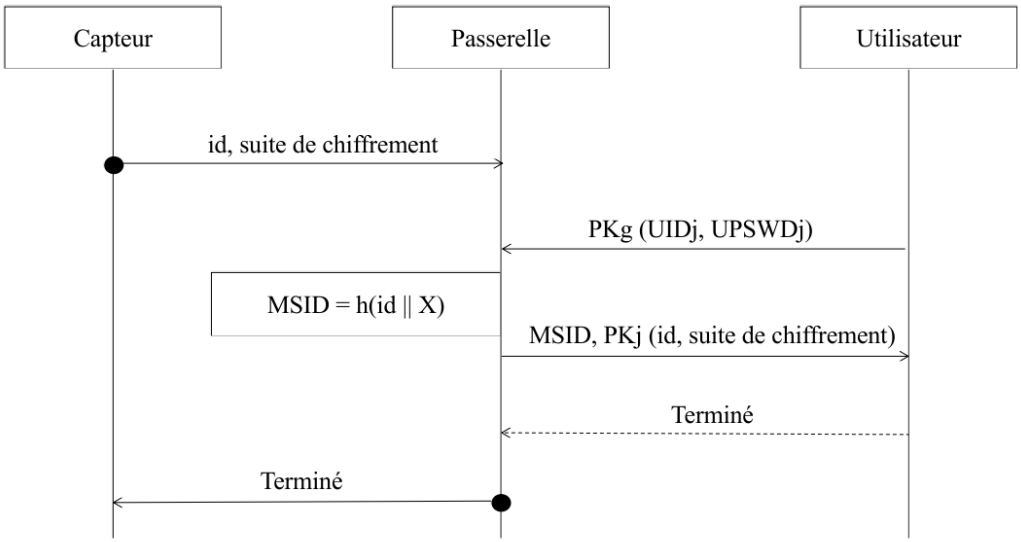

Figure 8:La phase d'enregistrement [1]

Comme présenté dans la figure 12 :

- Via un canal sécurisé, le nœud capteur envoie son identifiant id et une liste de suites de chiffrement prises en charge au nœud passerelle de réseau (la communication entre le nœud capteur et le nœud passerelle est censée être préalablement sécurisée).

- Le nœud de passerelle sélectionne la suite de chiffrement à utiliser et utilise l'identité de capteur Id et la clé secrète X pour calculer l'identité masquée du nœud de capteur MSId. Ensuite, il transmet un message contenant l'identité masquée MSID du nœud capteur, ainsi que le cryptage à la fois de l'identité du nœud capteur Id et de la suite de chiffrement sélectionnée à l'aide de la clé publique PK_j de l'utilisateur distant. (Lors du pré-déploiement du réseau, le nœud passerelle connaît la clé secrète du nœud capteur, ainsi que la clé publique de l'utilisateur distant).

- En réponse, l'utilisateur distant envoie un message crypté contenant la suite de chiffrement sélectionnée, complétée par la clé secrète du capteur.

- Enfin, le nœud passerelle envoie le message Terminé au nœud capteur.

- A la réception du message par le nœud capteur, l'enregistrement la phase se termine avec succès.

Les informations de sécurité sont stockées à la fin de l'étape d'enregistrement dans une table de liaison comme illustré dans le tableau 2, (ce tableau est enregistré à la fois dans le nœud passerelle et l'utilisateur distant).

Nœud	suite de chiffrement	Identité masquée : $MSId_i = h(Id_i \| X_i)$
Identifiant 1	Chiffre1 & X_1	$MSID_1$
Identifiant 2	Chiffre2 & X_2	$MSID_2$
Identifiant 3	Chiffre3 & X_3	……

Tableau3 : Informations relatives à la sécurité [1]

2.2.2. La phase d'authentification

Pendant la phase d'authentification, le nœud capteur et l'utilisateur distant s'authentifient mutuellement. Le processus d'authentification doit être complété par chaque nœud de capteur qui souhaite se connecter avec l'utilisateur distant. Le schéma d'authentification est le suivant (Figure13) :

a) Le nœud de capteur génère un nonce aléatoire de 8 octets N et envoie un message à l'utilisateur distant qui comprend le nonce créé, son identité masquée MSId et un HMAC (MSId N, Id).

b) A la réception du message par l'utilisateur distant, le message est vérifié en calculant le HMAC associé. Si la vérification est réussie, l'utilisateur distant génère également un nonce aléatoire M sur 8 octets, et transmet au nœud passerelle un message composé de l'identité masquée du nœud capteur MSId, du nonce N reçu, du nonce M, et HMAC (MSId, N, M).

c) Lorsque le nœud passerelle reçoit ce message, il calcule le HMAC correspondant et le vérifie. Si la vérification est réussie, le nœud passerelle crée un nonce aléatoire S de 8 octets et le XOR avec la valeur reçue N : ($T = N \oplus S$). À son tour, le nœud de passerelle délivre un message à l'utilisateur distant qui comprend les nonces N et M reçus, la valeur calculée T et un HMAC (M, Id, S).

d) A la réception du message par l'utilisateur distant, la valeur S est calculée comme suit : ($S = N \oplus T$) et le message est vérifié en calculant le HMAC associé. Si la vérification est réussie, l'utilisateur distant génère également un nonce aléatoire W sur 8 octets, applique un XOR avec la valeur S comme : ($Z = W \oplus S$), et envoie au nœud capteur un message composé par : le nonce N reçu , la valeur Z , la valeur S et un HMAC (N, Id, W).

e) Lorsque le nœud capteur reçoit le message, il calcule la valeur W comme ($W = Z \oplus S$), et le message est vérifié en calculant le HMAC associé. Si la vérification est réussie, l'authentification mutuelle entre les objets est terminée avec succès.

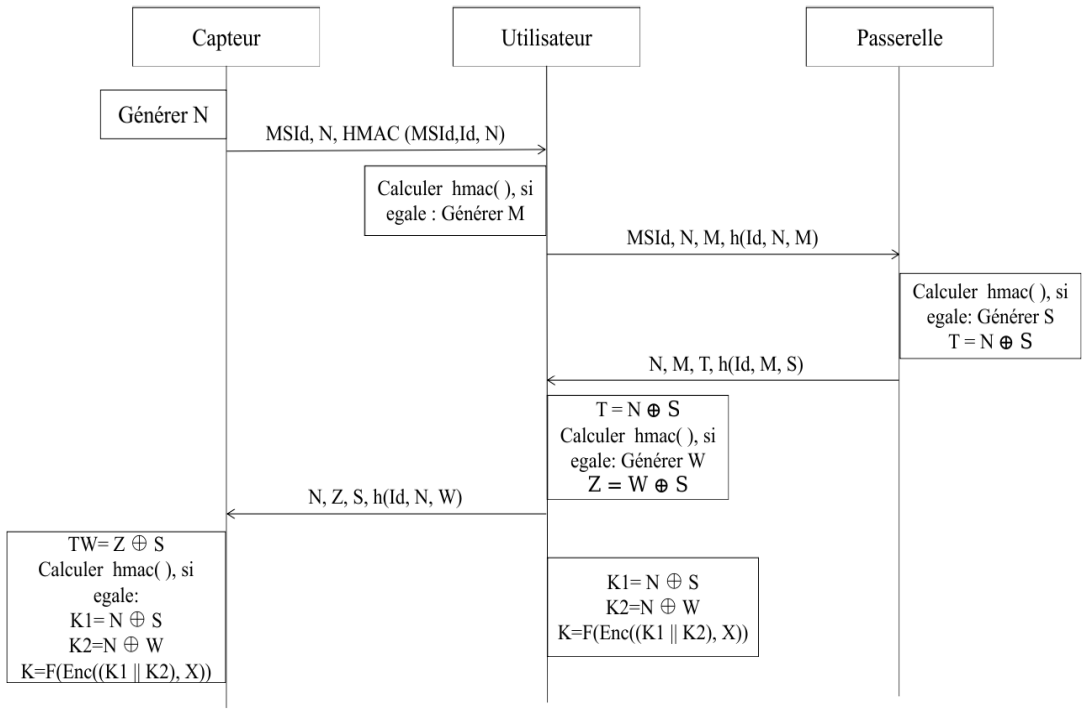

Figure 9: La phase d'authentification [1]

Pour faciliter la lecture, les notations utilisées dans la figure 13 sont définies dans le tableau 3.

Notation	**La description**
‖	Concaténation
⊕	Opération ou exclusif (Xor)
N	valeur du nombre aléatoire du nœud du capteur
M	Première valeur du nombre aléatoire de l'utilisateur distant
S	Valeur du nombre aléatoire du nœud de passerelle
W	Deuxième valeur du nombre aléatoire de l'utilisateur distant
H()	Une fonction de hachage à sens unique
Enc(N,X)	Cryptage AES-128 de la valeur N à l'aide de la clé secrète X

Déc(N,X)	Décryptage AES-128 de la valeur N à l'aide de la clé secrète X
HMAC( )	Code d'authentification de message de hachage à clé
F()	Si (N != 16 octets) : La Fonction F applique une fonction de hachage h() qui renvoie 16 octets en sortie

Tableau4 : Notations utilisées

2.2.3. L'établissement de clés partagées

Lorsque la phase d'authentification est terminée, une clé symétrique partagée K est établie pour sécuriser le canal de communication.

Une fonction personnalisée calcule cette clé comme K=F(Enc(K1 || K2, X)). Tout d'abord, les valeurs K1 et K2 sont calculées en effectuant un Xor de la valeur N avec les nonces S et W. Ensuite, la concaténation des deux valeurs K1 et K2 est prise et chiffrée avec la clé secrète associée du nœud capteur X. La clé K obtenue doit avoir une longueur de 128 bits.

2.3. Analyses

D'une part, les contributeurs de [1] ont fait une analyse théorique et une analyse automatique de la sécurité qui montre les avantages de ce schéma en termes de sécurité. Parmi ces avantages.

- La résistance contre les cyber attaques comme l'attaque par rejeu, l'attaque de l'homme de milieu, l'attaque d'usurpation d'identité et l'attaque par déni de service grâce à l'utilisation de la fonction de hachage HMAC et le masquage de l'identifiant de nœud de capteur durant les échange des messages d'authentification .

- Des fonctionnalités avancées qui améliorent le niveau de sécurité telles que l'authentification mutuelle, la protection de l'identité, l'intégrité des données, l'indépendance de la synchronisation, l'évolutivité.

D'autre part, pour évaluer des performances de ce schéma, les contributeurs de [1] se sont concentrés sur le nœud de capteur car il a une faible puissance de calcul et une faible énergie. L'énergie nécessaire pour exécuter les primitives cryptographiques a été calculée avec l'énergie nécessaire pour envoyer et recevoir des données. Les résultats obtenus montrent que le schéma proposé permet également d'économiser de l'énergie dans les différents cas d'un échec d'authentification.

3. La technologie de Blockchain

Le blockchain est une technologie récente qui aide beaucoup à augmenter le niveau de sécurité des réseaux IoT dans l'authentification et le contrôle d'accès. Dans cette section, nous définissons la technologie blockchain, ses caractéristiques et les avantages qu'elle donne à la sécurité des données.

3.1. Définition

La technologie blockchain est sous forme d'une base des données distribuée dans un réseau peer-to-peer qui fournit une méthode de décentralisation des systèmes. Bien que ne garantisse pas la décentralisation, il garantit que le stockage des données et les transactions sont distribués. Le paradigme décentralisé a le potentiel d'augmenter l'égalité de stockage et la disponibilité des informations et des ressources. Un réseau peer-to-peer[1] peut utiliser la blockchain pour faciliter les transactions entre les nœuds (figure 6). De la monnaie, des votes, des données sur la santé, des idées, des prédictions, … sont exemples qui pourraient tous être échangés. Jusqu'à présent, l'accent a été mis sur les crypto-monnaies comme Bitcoin.[50].

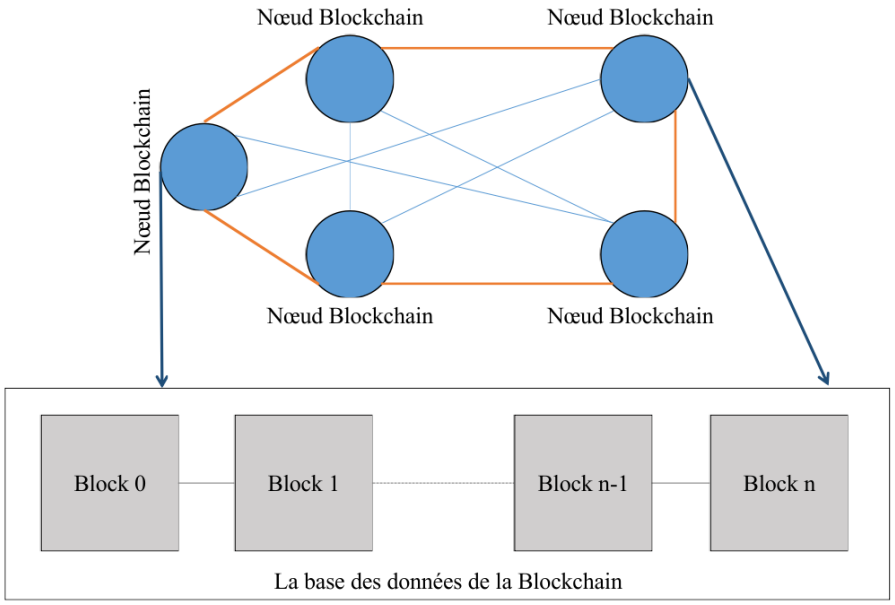

Figure 10: Réseau blockchain

Formellement, Blockchain est une chaîne de blocs qui sont connectés ensemble et sont en croissance continue en stockant les transactions sur les blocs. La structure des blocs est présentée dans la figure 7 :

[1]Est un réseau qui contient un groupe d'ordinateurs indépendants appelés nœuds qui sont interconnectés pour partager des données entre eux.

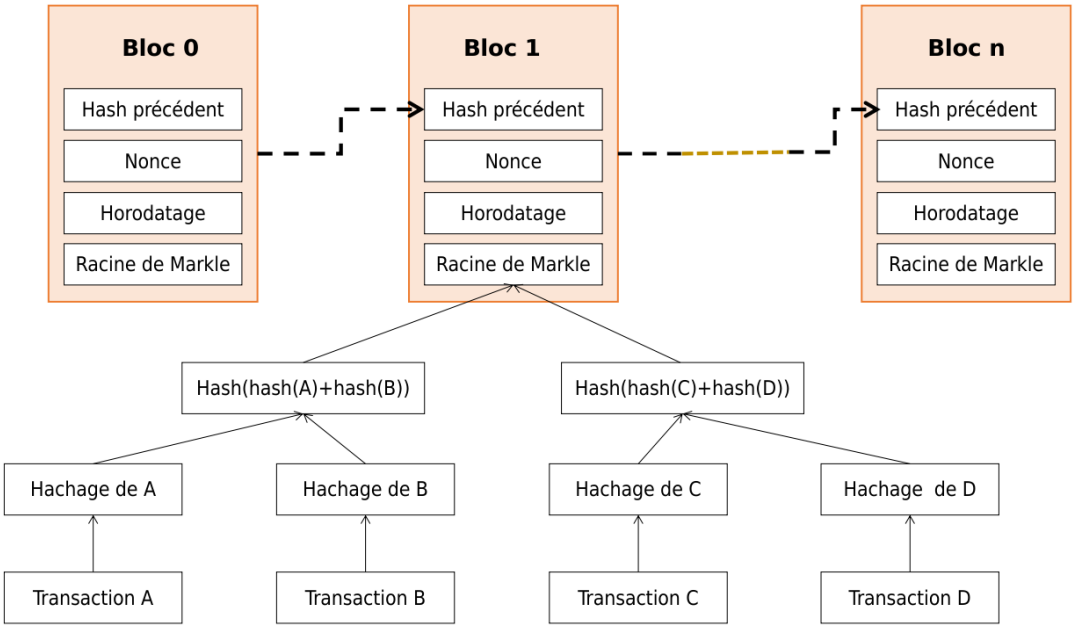

Figure 11: Structure de [52]

Dans ce qui suit, nous définissons les concepts de base de blockchain que nous utilisons tout au long du document :

- **Les blocs :** Un bloc contient un en-tête de bloc et des données de bloc. L'en-tête de bloc contient des métadonnées pour ce bloc. Les données de bloc contiennent une liste de transactions validées et authentiques qui ont été soumises au réseau blockchain. Il convient de noter que chaque implémentation de blockchain peut définir ses propres champs de données. Cependant, de nombreuses implémentations de blockchain utilisent des champs de données comme les suivants :
  - En-tête de bloc :
    - Le nombre de bloc.
    - La valeur de hachage de l'en-tête de bloc précédent.
    - Une représentation de hachage des données de bloc.
    - Un horodatage.
    - La taille du bloc.
    - La valeur de nonce.
  - Bloc de données :
    - Une liste des transactions et des événements du Ledger inclus dans le bloc.
    - D'autres données peuvent être présentes.
- **Bloc Genesis :** Est le premier bloc de transaction dans la blockchain.
- **Transactions :** Une transaction représente une interaction entre les parties. Avec les crypto-monnaies, par exemple, une transaction représente un transfert de la crypto-monnaie entre les utilisateurs du réseau blockchain. Pour les scénarios interentreprises, une transaction peut être un moyen d'enregistrer les activités se produisant sur des actifs numériques ou physiques. Bien qu'elles soient principalement utilisées pour transférer des actifs numériques, les transactions peuvent être plus généralement utilisées pour transférer des données. Dans le cas

des systèmes de contrats intelligents, les transactions peuvent être utilisées pour envoyer des données, traiter ces données et stocker des résultats sur .

- **Ledgers :** Un Ledger est un ensemble de transactions.
- **Un Contrat intelligent « Smart contract »** présente l'exécution d'un code stocké dans le blockchaine.

- **Nœud :** Est un ordinateur (ou tout autre système ou appareil physique) relié à un réseau blockchain (peer-to-peer). Il reçoit également une copie complète de et peut exécuter certaines fonctions telles que la création, la réception et l'envoi d'informations, ainsi que la validation des transactions.
- **Mining :** Est le processus de validation d'un bloc de transactions à ajouter à qui est réalisé par le mineur.
- **Les mineurs :** Sont des utilisateurs de blockchain qui consiste à vérifier les transactions qui stockés dans les blocs
- **Base de données blockchain :** Une base de données blockchain est une base de données distribuée, tolérante aux pannes et à ajout uniquement qui stocke les données dans des blocs, bien que tous les utilisateurs de aient accès aux blocs, ils ne peuvent ni les supprimer ni les modifier. Parce que chaque bloc a une valeur de hachage de son prédécesseur, les blocs sont liés dans une chaîne, chaque bloc contient un certain nombre de transactions qui ont été validées. Le réseau blockchain se compose de nœuds qui maintiennent dans un système distribué peer-to-peer, chaque bloc comprend également un horodatage indiquant l'heure de création du bloc et un nombre aléatoire (nonce) pour les activités cryptographiques, les blocs sont accessibles à tous les nœuds, bien qu'ils ne soient pas totalement sous leur contrôle.

- **Un Nonce :** une abréviation de "numéro utilisé une seule fois," qui est un nombre ajouté à un bloc haché ou crypté dans un blockchain, lorsqu'il est haché de nouveau afin de répondre aux restrictions de niveau de difficulté.
- **Le hash d'un bloc précédent (Previous Hash)** permet d'assurer que le bloc n'a pas été altéré par un tiers. Ceci relie les blocs ensemble (dans une chaîne) parce que les hachures sont cryptographiquement dérivées des données du bloc. Cela empêche la fraude, car un changement dans n'importe quel bloc de l'histoire invaliderait tous les blocs suivants que toutes les haches suivantes changeraient et tout le monde exécutant le blockchain remarquerait.

- **Arbre de Markle :** Est un type de structure de données qui comprend plusieurs hachages ou horodatages stockés dans . C'est une méthode de stockage de données dans en utilisant des hachages.

- **Hachage:** Le hachage est une méthode de calcul d'une sortie unique pour une entrée de pratiquement n'importe quelle taille à l'aide d'une fonction de hachage cryptographique. Les fonctions de hachage cryptographique sont utilisées à diverses fins dans un réseau blockchain, notamment :
  - Création d'identifiants uniques.
  - Sécurisation des données de bloc.
  - Sécurisation de l'en-tête de bloc.

3.2. Couches de Blockchain

Afin de bien comprendre , cinq couches principales sont présentes dans un nœud Blockchain et sont considérées comme un point commun entre les technologies Blockchain. Cette illustration ci-dessous représente ces cinq couches [53]:

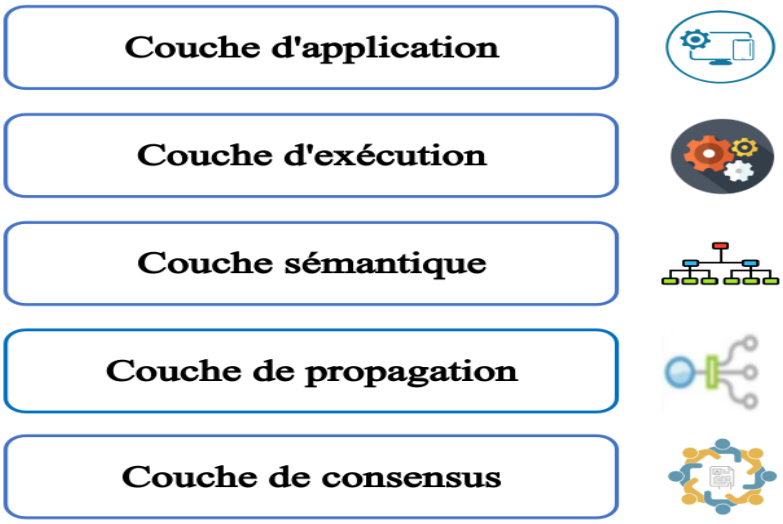

Figure 12 : Couches de blockchain [53]

- **Couche d'application :**Contient tous les codes de fonctionnalités nécessaires à la création d'une application pour les utilisateurs finaux. Les constructions de programmation côté client, les scripts, les API, les cadres de développement, etc. sont inclus.

- **Couche d'exécution :** Pour s'assurer que les transactions sont exécutées correctement, tous les nœuds d'un réseau Blockchain doivent exécuter un script ou un programme. Pour cela, la couche d'exécution est l'endroit où les fonctions requises de la couche d'application sont exécutées.

- **Couche sémantique :** La couche sémantique est considérée comme une couche logique car elle définit la logique et les règles du système. Toutes les instructions de fonction exécutées dans la couche d'exécution sont validées dans la couche sémantique.

- **Couche de propagation :** La couche de propagation, également appelée couche de communication peer-to-peer, permet aux nœuds de se découvrir, de communiquer et de se synchroniser selon la logique du réseau. La couche de propagation définit la transaction et la propagation du réseau pour assurer tout cela, ainsi que la stabilité du réseau Blockchain. Par conséquent, lorsqu'une transaction est effectuée ou qu'un nœud souhaite soumettre un bloc valide, l'ensemble du réseau est averti.

- **Couche de consensus :** La couche de consensus est la couche de base deblockchain. Son objectif principal est que tous les nœuds se mettent d'accord sur un état de registre unique et cohérent. Selon le cas d'utilisation, plusieurs méthodes pour obtenir un consensus entre les nœuds peuvent exister.

3.3.Les notions de preuve et de consensus

La fonctionnalité d'un blockchain peut être résumée comme suit : un nœud fournissant des données fraîches enregistre les données dans un bloc puis diffuse le nouveau bloc accessible au réseau de . Les nœuds qui obtiennent le nouveau bloc utilisent le hachage pour vérifier le bloc. Le payload est ajoutée à un bloc si elle est correcte. Tous les nœuds membres exécutent des algorithmes de preuve de travail (PoW) ou de preuve de participation (PoS) sur le bloc. Une fois le consensus obtenu, le nouveau bloc est mis dans et tous les nœuds le valident [45].

PoW, qui est utilisé dans le contexte de Bitcoin, est la preuve la plus connue. PoW a été créé pour se protéger contre les attaques de spam et de déni de service. Pour lutter contre un hypothétique 51% du taux de hachage du réseau, la puissance de hachage totale du réseau blockchain était nécessaire, Tous les clients peuvent participer au hachage. PoW s'est avéré être un cauchemar énergétique lorsque Bitcoin est arrivé près de 15 ans plus tard, et la course aux richesses minières a commencé. En 2012, les performances totales du réseau Bitcoin ont dépassé celles du supercalculateur le plus productif au monde [56]. [57] Les protocoles PoW sont lents. Les cycles d'horloge et la puissance du processeur sont les ressources rares requises pour PoW [58].

Pour modifier un bloc à l'aide de PoW, un attaquant ou un agresseur potentiel doit extraire tous les blocs avant celui qu'il souhaite modifier. Le coût de l'extraction d'un seul bloc est assez coûteux car il implique des procédures de calcul intensif comme la résolution de défis cryptographiques difficiles, le coût de calcul élevé associé au paradigme PoW, d'autre part, est le principal inconvénient de l'approche blockchain utilisée dans le contexte de Bitcoin. La complexité requise de l'extraction d'un bloc nécessite des quantités massives d'électricité et de ressources informatiques, ce qui rend la méthode non viable dans les domaines où l'évolutivité est un problème. PoW n'est pas un paradigme approprié pour des secteurs tels que la publicité en ligne, où l'évolutivité est un problème majeur, en particulier pour les mises en œuvre à un stade précoce.[50].

Le système distribué doit s'accorder sur un seul état partagé, qui est la difficulté du consensus blockchain, que les paradigmes du PoW visent à résoudre. Les mécanismes de consensus de sont actuellement lents, longs et peu énergivores. Le PoS est l'alternative préférée au PoW pour parvenir à un consensus. Il existe de nombreuses options différentes pour PoW, cependant, on ne sait pas comment leurs fonctionnalités de sécurité et leurs incitations se comparent les unes aux autres. Les alternatives consistent à rendre l'utilisation de l'énergie moins inefficace ou à remplacer les crypto-puzzles dénués de sens utilisés dans le PoW par des problèmes significatifs [24]. PoW, PoS, dPoW, preuve de combustion (PoB), preuve de capacité

(PoC), preuve d'activité (PoA), preuve d'existence (PoE), preuve d'intelligence (PoI) , la preuve de chance (PoL), le registre d'ondulation, le réseau de foudre et les chaînes de blocs croisées sont les principaux mécanismes de consensus. [59] contient des descriptions détaillées des méthodes de consensus mentionnées ci-dessus [50].

3.4. Les types de

Les blockchains **privées**, **publiques** et **hybrides** [55] sont les trois types de blockchain. Les autorisations d'écriture sont surveillées par une autorité décisionnelle centralisée dans un blockchain totalement privée, tandis que les autorisations de lecture sont soit publiques, soit restreintes. Un blockchain privé est un registre autorisé qui permet la mise sur liste blanche (ou liste noire) d'une identité d'utilisateur via un processus organisationnel de Know-Your-Business (KYB) et Know-Your-Customer KYC). Chaque internaute a accès à des blockchains décentralisées publiques. Tous les participants peuvent voir quels blocs sont ajoutés à la chaîne et jusqu'où elle a progressé. Pour le processus de validation, les blockchains publiques entièrement décentralisées nécessitent un mécanisme de consensus.
L'étendue de la décentralisation, ou comment ils assurent l'anonymat, est la principale différence entre les blockchains publiques et privées. Entre les deux extrêmes, un continuum existe, aboutissant à des blockchains partiellement décentralisées. Les blockchains de consortium sont un hybride entre les blockchains publiques et privées avec un seul nœud hautement fiable. Le continuum s'applique également à la consommation d'énergie; par rapport à une blockchain privée de confiance, les blockchains publiques, en particulier celles mettant en œuvre la méthode de consensus PoW sans confiance, nécessitent des quantités massives d'énergie [50].

3.5. Les avantages de

Les principaux avantages de la technologie sont :

- **Désintermédiation :** Blockchain est un système décentralisé, donc, il n'est pas nécessaire de travailler avec une organisation tierce ou un administrateur central. Cela signifie que le système fonctionne sans l'utilisation d'un intermédiaire, tous les participants à prenant des décisions. Chaque système contient une base de données, qui doit être protégée car le système interagira avec des organisations tierces. La base de données peut être piratée ou les données peuvent tomber entre de mauvaises mains. Le processus de sécurité de la base de données peut prendre beaucoup de temps et coûter très cher. Si la technologie Blockchain est utilisée, cela peut être évité car les transactions Blockchain ont leur propre preuve de validité et leur autorité pour faire respecter les limitations. Cela signifie également que les transactions peuvent être vérifiées et traitées de manière indépendante [67], [68].

- **La confiance :** La confiance de repose sur la croyance de deux ou plusieurs participants, qui ne se connaissent pas. Il y a l'idée principale est les transactions réelles et non sans valeur entre ces personnes inconnues. La confiance peut être

encore augmentée, car il peut y avoir plus de processus et d'enregistrements partagés [68], [69].

- **L'immuabilité :** L'immuabilité est obtenue en acceptant et en partageant des transactions tout au long de . Il sera impossible de modifier ou de supprimer une transaction une fois celle-ci connectée à . Cela dépend aussi du type de système : si le système est centralisé, il peut être modifié ou supprimé car une seule personne ne prend la décision. Cependant, si le système est décentralisé, comme , chaque transaction liée à est dupliquée sur chaque ordinateur du réseau. En raison de cet avantage, la technologie Blockchain est immuable et indestructible. Les utilisateurs de Blockchain ont un contrôle total sur toutes les transactions et informations. Lorsqu'un intrus a la capacité de traitement pour remplacer ou effacer des informations sur toutes les machines, y compris celles de , avant que le prochain bloc ne soit enregistré ici, il est possible de modifier ou de supprimer des informations dans . Si contient un petit nombre d'ordinateurs, la technologie est plus vulnérable aux agressions ; néanmoins, si contient un grand nombre d'ordinateurs, le système devient plus sûr et plus transparent [52].

- **La transparence :** La transparence de est obtenue grâce au mécanisme de copie des transactions. Chaque transaction est dupliquée sur l'un des ordinateurs du réseau Blockchain, comme indiqué ci-dessus. Chaque membre a accès à toutes les transactions, ce qui signifie que toute activité est visible pour tous les participants à . Personne ne peut rien faire sans réfléchir. [67] [70].

- **Traçabilité :** est conçue de telle manière qu'elle puisse montrer d'éventuels défauts et, si nécessaire, les corriger. Cet avantage rend la traçabilité de la technologie Blockchain [71].

- **La cohérence :** La forte sécurité de la technologie Blockchain est réalisée au point d'entrée individuel du réseau. Parce que tous ceux qui rejoignent le Blockchain se voient attribuer une identité unique liée à leur compte. Un autre facteur de sécurité de est la cohérence de la chaîne de hachage cryptographique. Lorsqu'un nouveau bloc est formé, la valeur de hachage pour le nouveau bloc doit être calculée. La valeur du hachage précédent est presque certainement incluse dans le nouveau hachage. De manière générale, le hachage contient le type, le numéro d'identification du bloc, la valeur du hachage précédent, l'heure à laquelle le bloc a été produit, le numéro d'identification de l'utilisateur, le niveau du mineur et la racine merkle, qui contient des informations sur les transactions passées et leurs hachages . La clé de nœud génère ce hachage automatiquement. Il est impossible de modifier les informations de la valeur de hachage [71] dans ce scénario [52].

- **Traitement plus rapide** : Traditionnellement, le traitement et le paraphe d'une transaction dans une organisation bancaire prennent beaucoup de temps. L'utilisation de la technologie Blockchain réduit le temps de traitement et d'initialisation de plusieurs ordres de grandeur – d'environ 3 jours à des minutes voire des secondes [79], [81].

Comme nous l'avons vu, la technologie blockchain présente d'énormes avantages en termes de sécurité et de stockage de données, et elle est utilisée dans différents domaines, y compris le domaine de la santé et l'IoT. Dans ce qui suit, nous voyons comment le blockchain peut être utilisé pour renforcer la sécurité du processus d'authentification dans les systèmes IoT.

4. L'utilisation de la blockchain dans l'authentification

Dans cette section, nous présentons l'utilisation de la technologie blockchain dans le processus d'authentification des systèmes IoT dans le domaine de l'e-santé. Nous examinons différents protocoles d'authentification basés sur la technologie blockchain et nous voyons l'avantage d'utiliser cette nouvelle technologie pour améliorer le niveau de sécurité dans le processus d'authentification.

Dans [72], les auteurs proposent un modèle d'identification et d'enregistrement d'un dispositif IoT inséré dans une application smart city. Ils développent une passerelle API pour vérifier l'identité et authentifier les messages de signature reçus par les appareils IoT, à l'aide de Blockchain et de Smart Contracts. Cette stratégie de sécurité utilise les paradigmes de réseau Edge et Fog Computing. Ces paradigmes ont un potentiel dans les implémentations IoT, car ils sont efficaces et économiques, même avec de faibles taux de transfert de données. Ils sont adaptés aux demandes où l'intelligence de l'application est proche des dispositifs producteurs d'informations. L'idée principale est d'envoyer des documents consolidés, signés, vérifiés et identifiés.

Dans [37], l'auteur propose un schéma sécurisé de stockage et de protection des données IoT basé sur le blockchain. L'Edge computing est intégrée pour aider à gérer le stockage des données et les petits appareils IoT effectuant des calculs. La cryptographie sans certificat est adoptée pour mettre en place un système d'authentification pratique pour les applications IoT basées sur le blockchain qui surmonte l'inconvénient de la cryptographie sans certificat en offrant une plate-forme pour diffuser la clé publique d'un utilisateur. Un algorithme détaillé sur la façon de traiter les transactions dans un tel système et sur la façon d'obtenir l'authentification et la responsabilité est fourni. Il s'agit du premier article abordant le problème de la création d'un système de stockage sécurisé et responsable pour les données IoT à grande échelle, et le premier à combiner l'informatique de pointe, la cryptographie sans certificat et le blockchain dans son ensemble pour servir les applications IoT.

Le travail effectué dans [73] propose une approche originale appelée bulles de confiance, dans laquelle des zones virtuelles sécurisées sont créées, où les appareils peuvent communiquer de manière totalement sécurisée. L'approche des bulles de confiance peut être appliquée à de nombreux contextes, services et scénarios IoT. Il s'appuie sur un blockchain publique, ainsi, il bénéficie de toutes ses propriétés de sécurité. En outre, il a défini les exigences de sécurité qu'un schéma d'authentification

IoT doit garantir et a construit un modèle de menace. L'évaluation de cette approche montre sa capacité à répondre aux exigences de sécurité demandées ainsi que sa résilience face aux attaques. De plus, il a fourni une étude approfondie de ses consommations de temps et d'énergie, où différents appareils ont été évolués.

Les auteurs dans [74] ont proposé une conception de système et la mise en œuvre d'une solution basée sur Blockchain utilisant des contrats intelligents Ethereum pour l'authentification des appareils IoT à grande échelle, de manière décentralisée sans tiers intermédiaire. Ils ont mis en œuvre le contrat intelligent Ethereum proposé en utilisant le langage Solidity. L'authentification à grande échelle des appareils IoT est présentée en impliquant des nœuds de brouillard qui sont utilisés pour soulager les appareils IoT de la charge de traitement des tâches d'authentification et de la surcharge de connectivité impliquée dans l'interfaçage avec le réseau Ethereum Blockchain. Ils ont présenté et discuté les détails de l'architecture du système, les interactions et les échanges de messages entre les participants, y compris les diverses activités en chaîne et hors chaîne. De plus, ils ont montré et discuté comment les fonctionnalités globales et les opérations du mécanisme d'authentification régi par le contact intelligent ont été mises en œuvre et testées. Différents scénarios de test ont été présentés pour vérifier les différentes fonctionnalités du système à l'aide de Remix IDE. Les auteurs ont également fourni une analyse de sécurité et montré que la solution d'authentification proposée atteint les objectifs de sécurité et résiste aux attaques connues telles que l'écoute clandestine, la relecture et le DoS. Comme perspective, , ils sont en train de construire un prototype de système entièrement fonctionnel impliquant de vrais appareils IoT basés sur Arduino et Raspberry PI connectés à des nœuds de brouillard équipés du client Ethereum pour être connectés au vrai réseau public Ethereum qui héberge le code de contrat intelligent.

Comme nous le voyons, la technologie blockchain a donné de nombreuses solutions pour renforcer le processus d'authentification dans les systèmes IoT et e-santé. La question qui se pose est comment pouvons-nous utiliser cette technologie dans le schéma [1] pour renforcer sa sécurité ? La réponse à cette question fait l'objet du chapitre suivant.

5. Conclusion

Dans ce chapitre, nous avons présenté le processus d'authentification et son importance. On a aussi présenté le schéma d'authentification qui nous a inspiré a réaliser nore travail Puis, nous avons présenté la technologie blockchain qui est utilisée récemment dans les systèmes IoT pour renforcer la sécurité et le stockage des données. Nous avons également présenté son utilisation dans les protocoles d'authentification. Dans le chapitre suivant, nous présenterons notre contribution qui est un schéma d'authentification basé sur la technologie blockchain.

.

Chapitre 3 : Notre schéma d'authentification basé sur la technologie blockchain dans l'IoT

Dans cette section, nous allons commencer par présenter le schéma d'authentification que nous avons proposé. En se basant sur les mêmes opérations et fonctions présentées dans le schéma précédent, notre schéma intègre la technologie blockchain pour stocker de manière sécurisée les identités des composants disposés à communiquer entre eux qui sont l'utilisateur distant et les nœuds de capteurs. Ensuite, nous allons effectuer une analyse de sécurité et une analyse des performances de notre proposition en indiquant ses avantages par rapport à la version d'origine du schéma.

1. Architecture du réseau

Notre architecture de réseau (figure 14) est composée principalement d'un ou de plusieurs nœuds de capteurs hétérogènes, passerelles, utilisateurs distants, et d'un réseau blockchain. Ces composants interagissent pour transmettre les données collectées par les capteurs aux utilisateurs distants autorisés. Pour sécuriser ces échanges, des processus d'enregistrement et d'authentification mutuelle doivent être effectués. Ces processus seront détaillés par la suite.

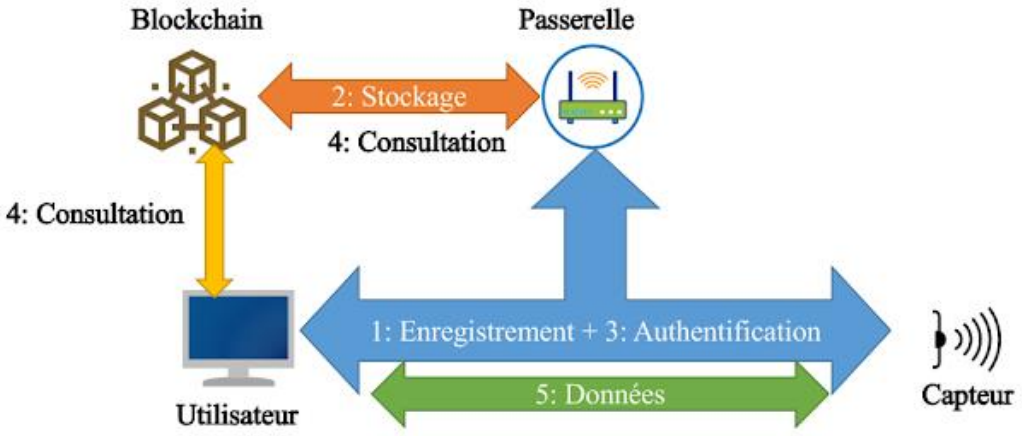

Figure 13: Notre architecture du réseau

Formellement, les composants de notre architecture sont :

- **capteur:** qui est un dispositif IoT, chargé de collecter des données et de les transmettre à l'utilisateur (le collecteur de données). Identifié par un ID et MSID (identifiant masqué), le capteur a moins de puissance de calcul, de capacité de stockage et d'énergie que l'utilisateur et la passerelle. Il a la capacité d'effectuer des chiffrements symétriques et au moins un chiffrement asymétrique. Au contraire, le nœud passerelle et l'utilisateur distant peuvent effectuer n'importe quelle opération de calcul.

- **Utilisateur:** qui est destiné à collecter les données envoyées par les capteurs et à les utiliser à des fins médicales. Il se connecte au capteur à distance via un téléphone ou un ordinateur. La connexion entre l'appareil utilisateur et le capteur est établie à l'aide d'un troisième appareil appelé la passerelle.

- **Passerelle:** qui est un appareil placé entre le capteur et l'utilisateur distant, permet d'établir la connexion entre ces appareils. Considérée comme un nœud de blockchain, elle a une adresse blockchain pour stocker les identités de l'utilisateur et du capteur (les deux appareils qui devaient se connecter l'un avec l'autre) dans le réseau blockchain.

- **Blockchain:** qui comprend une base de données qui contient les données stockés dans le Blockchain et un réseau des nœuds (passerelle) participants comme le montre la figure suivante :

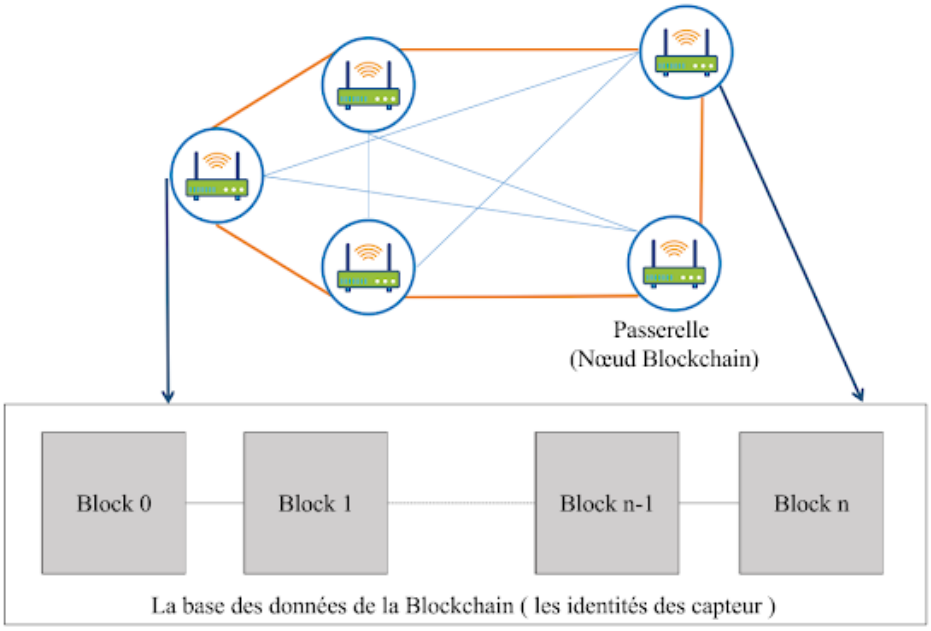

Figure 14: Réseau blockchain

2. Fonctionnement

Le schéma d'authentification proposé assure une authentification mutuelle entre un nœud de capteur et un utilisateur distant dans un réseau de capteurs hétérogène sans fil. Ce schéma passe par deux phases qui sont :

- **La phase d'enregistrement :** où les nœuds de capteurs doivent enregistrent auprès de l'utilisateur distant à travers la passerelle.

- **La phase d'authentification:** où le nœud de capteur et l'utilisateur distant doivent s'authentifient mutuellement avant d'échanger des données. Dans cette phase, une clé de session est aussi établie entre eux.

Dans ce qui suit, nous décrirons en détail les phases mentionnées précédemment.

2.1. La phase d'enregistrement

Dans cette phase, le nœud capteur et l'utilisateur distant envoient au nœud passerelle leurs identifiants qui seront enregistrés dans le réseau blockchain, De même que [1], nous supposons que la communication entre les trois nœuds concernés par l'enregistrement est réalisée dans un canal préalablement sécurisé pour que les informations soient transmises en clair (non cryptées ou chiffrés).
La figure 16 montre les étapes pour qu'un capteur soit enregistré dans le réseau :

1. Le nœud de capteur envoie un message de demande d'enregistrement au nœud de passerelle. Ce message est composé de son identifiant «id», d'une suite de chiffrement qui contient la clé secrète de nœud de capteur et un algorithme de hachage et de l'adresse de l'utilisateur distant. La passerelle contacte le destinataire pour l'envoyer ses informations d'identification.

2. L'utilisateur destinataire envoie un message contenant ses informations (identifiant «uid» et mot de passe «upswd») au nœud passerelle.

3. Le nœud de passerelle calcule le MSID en utilisant l'identifiant (id) de capteur et son algorithme de hachage. Les informations reçues seront enregistrées dans le blockchain en utilisant l'adresse blockchain du nœud passerelle. L'index des informations enregistrées est le MSID.

4. Le nœud de passerelle envoi l'adresse d'enregistrement des informations avec l'indice MSID à l'utilisateur distant.

5. L'utilisateur distant répond avec un message «terminé».

6. Enfin, le nœud de passerelle envoie de sa part un message «terminé» au nœud de capteur.

Lorsque le nœud de capteur reçoit le message «terminé» la phase d'enregistrement se termine avec succès et l'utilisateur distant sauvegarde l'adresse d'enregistrement et l'indice MSID dans le tableau 4.

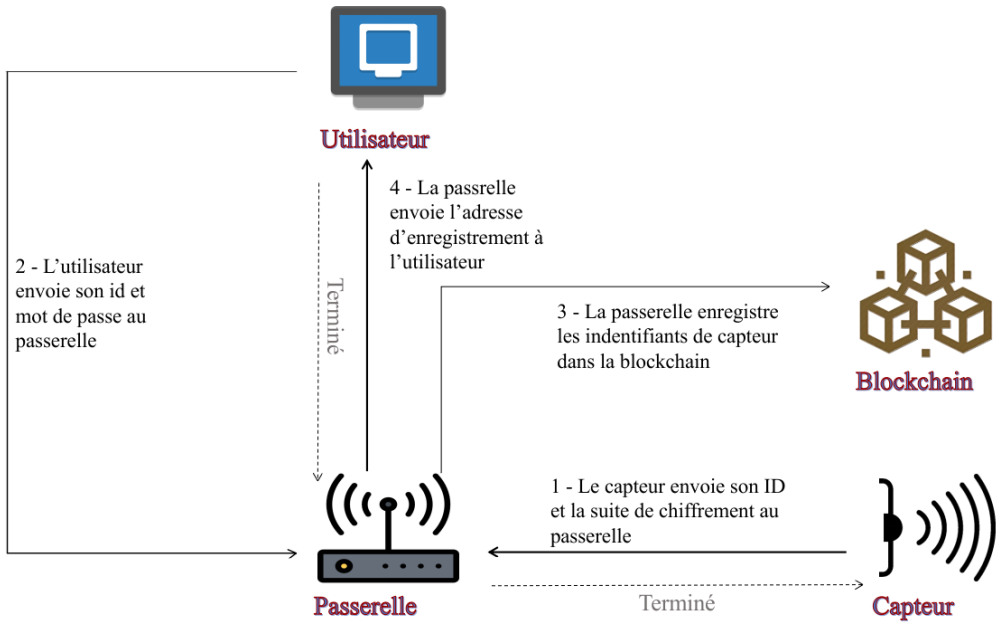

Figure 15: La phase d'enregistrement

L'indice «MSID»	Adresse d'enregistrement
$Msid_1$	$Addr_1$
$Msid_2$	$Addr_2$
↓	↓
$Msid_n$	$Addr_m$

Tableau5 : Enregistrement de l'utilisateur

2.2. La phase d'authentification

A chaque connexion, les capteurs enregistrés doivent s'authentifier mutuellement pour assurer la sécurité de la communication avec l'utilisateur distant. Comme présenté dans figure 17, le processus d'authentification s'exécute comme suit :

1- Le nœud capteur génère une valeur aléatoire N de 8 octets.

2- Le nœud capteur envoie un message contenant la valeur générée N avec son MSID et une valeur de hachage h(MSID, id, N) de son id, MSID et la valeur générée N, à l'utilisateur distant.

3- L'utilisateur distant reçoit le message du nœud capteur et utilise le MSID pour trouver l'adresse blockchain enregistré dans le tableau des informations. Puis, il utilise le MSID et l'adresse blockchain pour trouver l'identifiant du nœud de capteur et son algorithme de hachage.

4- Le blockchain vérifie l'adresse et envoie un message vide si l'adresse est invalide, le id si l'adresse est valide.

5- L'utilisateur distant calcule une valeur de hachage h(MSID, id, N) et la compare avec la valeur de hachage reçue si elles sont différentes l'authentification sera échouée sinon une valeur aléatoire M sera générée.

6- L'utilisateur distant envoie un message composé de MSID, N, M et une valeur de hachage h(id, N, M) calculé, au nœud de passerelle.

7- Lorsque le nœud de passerelle reçoit le message de l'utilisateur distant, le nœud de passerelle utilise son adresse blockchain et le MSID pour trouver le id de nœud de capteur et son algorithme de hachage.

8- vérifie l'adresse et envoie un message vide si l'adresse est invalide, le id si l'adresse est valide.

9- Le nœud de passerelle calcule une valeur de hachage h(i, N, M) et la compare avec la valeur de hachage reçue si elles sont différentes l'authentification sera échouée sinon une valeur aléatoire S sera générée et la valeur T sera calculé en utilisant l'opérateur ou exclusif (xor) $T = N \oplus S$.

10- Le nœud de passerelle envoie un message contient le MSID, N, M, T et une valeur de hachage h(id, M, S) à l'utilisateur distant.

11- L'utilisateur distant calcule la valeur S en utilisant le ou exclusif (xor) $S = N \oplus T$. Après avoir calculé S, une valeur de hachage h(id, M, S) sera calculée. L'utilisateur distant vérifie si le la valeur de hachage calculé est égale à la valeur de hachage reçu, s'ils sont différents l'authentification sera échouée sinon une valeur aléatoire W sera générée, et la valeur $Z = W \oplus S$ sera calculé.

12- L'utilisateur distant calcule une valeur de hachage h(id, N, W) et l'envoie au nœud de capteur avec les valeurs N, Z, S.

13- Le nœud de capteur reçoit le message de l'utilisateur distant et vérifie la valeur de h() si ils sont égaux la partie authentification sera terminée avec succès sinon une l'authentification sera échouée.

14- L'utilisateur distant calcule la valeur k1 = k1 = N $\oplus$ S et la valeur k2 = N $\oplus$ W, et puis il calcule K=F(Enc((K1 || K2), X)), la clé k sera utilisé pour l'échange des données avec le nœud de capteur (chiffrement symétrique).

15- Le nœud de capteur calcule la valeur k1 = k1 = N $\oplus$ S et la valeurk2 = N $\oplus$ W, et puis il calcule K=F(Enc((K1 || K2), X)), la clé k sera utilisé pourl'échange des données avec l'utilisateur distant (chiffrement symétrique).

Pour faciliter la lecture, les notations utilisées dans notre schéma d'authentification proposé sont résumées dans le tableau 5 suivant:

Notation	Description
N	Valeur du nombre aléatoire du nœud de capteur
M	Première valeur du nombre aléatoire de l'utilisateur distant
S	Valeur du nombre aléatoire du nœud de passerelle
W	Deuxième valeur du nombre aléatoire de l'utilisateur distant
$\oplus$	OU exclusif (Xor)
h()	Une fonction de hachage à sens unique
k1	$k1 = N \oplus S$
k2	$k2 = N \oplus W$
k	K=F(Enc((K1 || K2), X))
Enc(Y, X)	Chiffrement AES de la valeur Y à l'aide de la clé secrète X
F(Y)	Si (Y! = 16 octets): La fonction F applique une fonction de hachage h () qui renvoie 16 octets en sortie.

Tableau6 : Notations utilisées

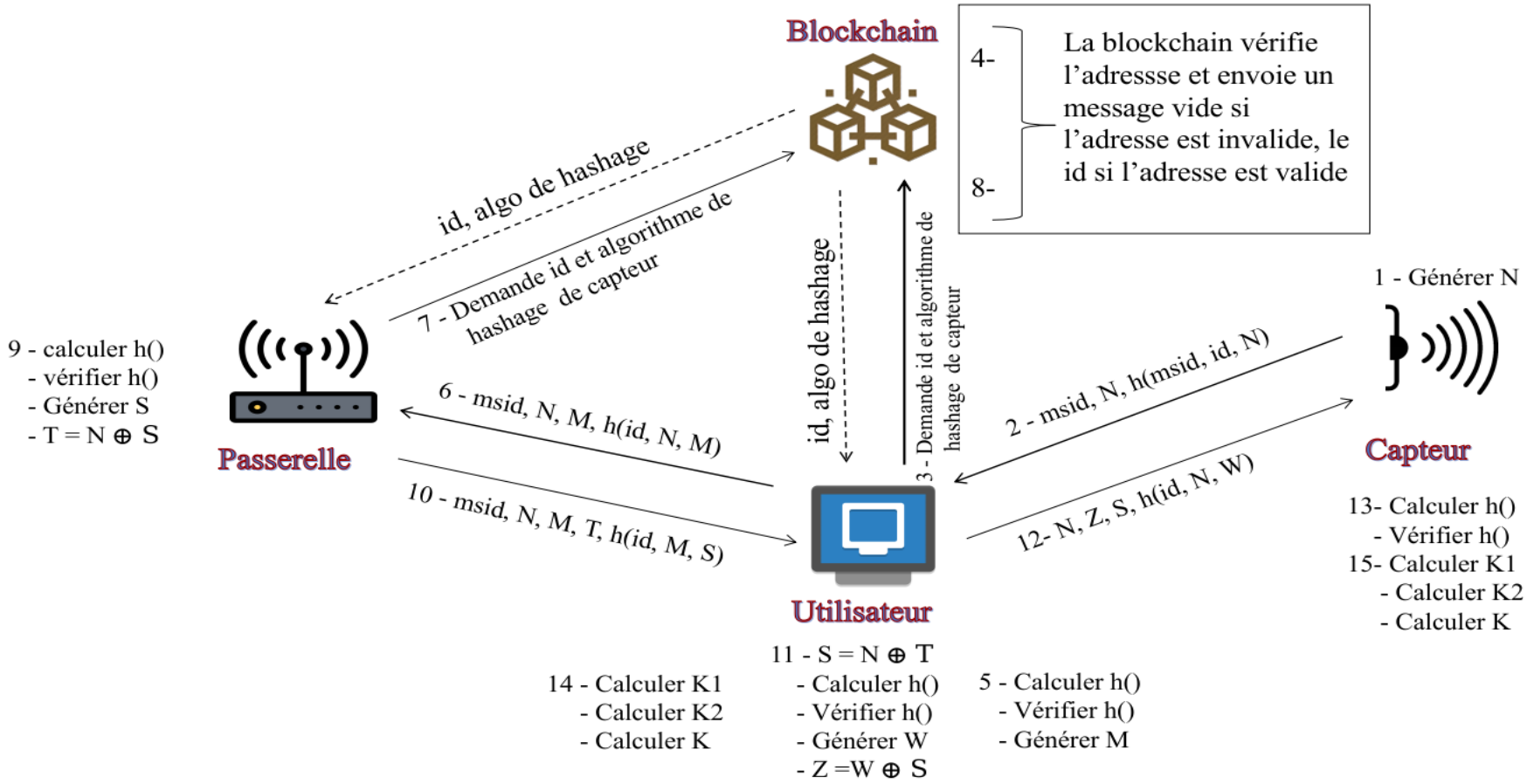

Figure 16: La phase d'authentification

3. Analyses de notre schéma

Dans cette section, nous analysons notre schéma proposé en termes de sécurité et de performances. Nous commençons par parler des performances en termes du coût de l'énergie et de la puissance de calcul nécessaires pour effectuer le processus d'authentification. Ensuite, nous procédons à une analyse approfondie en terme de sécurité en définissant comment notre processus d'authentification peut résister à plusieurs cyber attaque, et comment un administrateur peut surveiller le système et le sécuriser. Nous mentionnons également l'avantage d'utiliser la technologie blockchain dans le schéma proposé.

3.1. Analyses de performance

La consommation d'énergie et la puissance de calcul et même la capacité de stockage dans le nœud utilisateur et le nœud passerelle ne sont pas un problème car ces nœuds ont plus que suffisamment de ressources pour effectuer les opérations et les calculs présentés dans notre schéma. Mais le nœud du capteur a moins d'énergie et de puissance de calcul, ce qui nécessite d'effectuer moins de calculs. Dans notre schéma proposé, nous avons conservé les mêmes calculs et opérations effectués dans le schéma [1] pour le nœud de capteur. Ainsi, l'analyse des deux schémas en termes de performances au niveau du capteur serait la même. Voici quelques avantages des deux schémas en termes de performances :

- Après une comparaison entre le [1] et quelques autres schémas qui sont [75] [76] [77], en terme de coût de communication dans le nœud capteur, le schéma [1] (et par conséquent, notre schéma proposé) a moins de coûts de communication, et cela nécessite moins d'énergie pour effectuer la communication, donc la consommation d'énergie est réduite, ce qui rend le schéma plus performant.

- En ce qui concerne la capacité de stockage, le nœud de capteur après la phase d'enregistrement stocke son identifiant (id), l'identifiant masqué (MSID), l'algorithme de hachage et la clé secrète (X), ce qui fait environ 50 octets à stocker. Après la phase d'authentification, le capteur doit stocker le (Id, MSID, X, algorithme de hachage, N, Z, S, W, K1, K2, K) qui nécessite environ 120 octets de capacité de stockage. Lorsque l'établissement de la clé est terminé, le capteur peut supprimer le (N, Z, S, W) et ne conserve que (id, MSID, X, algorithme de hachage), ce qui nécessite environ 64 octets, ce qui est considéré comme un faible coût de stockage et améliore les performances du schéma.

- Comme évaluation énergétique, les contributeurs de [1] ont fait une comparaison entre leur schéma et quelques autres en termes de consommation d'énergie, et en conséquence le nœud de capteur dans [1] (par conséquent notre schéma proposé) a une faible consommation d'énergie, ce qui rend le système beaucoup plus performant.

Le nœud capteur est le nœud le plus faible du réseau en termes d'énergie et de puissance de calcul, et notre schéma offre moins de consommation d'énergie et moins de quantité des données pour le stockage, ce qui le fait un meilleur schéma pour l'utilisation en termes de performances.

3.2. Analyses de sécurité :

Notre solution hérite, entre autre les avantages du schéma [1], les avantages de blockchain en termes de sécurité :

- **Analyse théorique :** De même que [1], notre schéma offre une résistance aux différentes attaques possibles comme :
  - **Attaque de l'homme au milieu :** si un intermédiaire a reçu un message pendant la phase d'authentification, tout ce qu'il obtient de ce message est une identité masquée d'un capteur et quelques nombres aléatoires. Avec ce niveau, le système n'aura aucun problème, et toutes les informations critiques sont toujours en sécurité.

  - **Attaque par rejeu :** Si un intermédiaire reçoit un message échangé entre deux des trois nœuds concernés par l'authentification (le nœud capteur, le nœud passerelle et l'utilisateur distant) lors de la phase d'authentification, un nouveau nombre aléatoire sera généré. Dans notre schéma proposé, de nouveaux nombres aléatoires sont générés à chaque authentification, donc le nombre reçu lors des échanges de messages précédents sera refusé, et le message sera considéré comme un rejeu d'un ancien message.

  - **Attaque par déni de service :** Si le réseau est inondé par des messages dans le processus d'authentification, tous les messages reçus sont vérifiés par l'utilisation de nombres aléatoires, et cela rend le message acceptable ou non. un message inacceptable provoque l'échec de l'authentification.

  - **Usurpation d'identité :** L'identifiant du capteur échangé dans le réseau est l'identifiant masqué (MSID) donc un attaquant ne peut pas usurper l'identité du capteur. Aussi, l'attaquant ne peut pas obtenir l'identité d'un utilisateur ou d'une passerelle sans calculer exactement le même hachage de l'identifiant de capteur, par conséquent.

  Et voici quelques fonctionnalités avancées de sécurité qui contribuent à augmenter la valeur de la sécurité dans notre schéma proposé :

  - **Authentification mutuelle :** Dans tout schéma de communication, l'authentification mutuelle est si importante puisqu'elle donne confiance aux deux parties ; l'expéditeur et le destinataire du message savent exactement avec qui ils communiquent.

- **Intégrité des données :** Tous les messages transmis pendant la phase d'authentification sont vérifiés avec une valeur de hachage calculée, ce qui garantit l'intégrité du message reçu.

- **Établissement de la clé :** La communication et les données transmises entre le nœud capteur et l'utilisateur distant sont cryptées et sécurisées à l'aide de la clé calculée après la phase d'établissement de la clé.

- **Protection de l'identité du nœud de capteur :** Après la phase d'enregistrement, un identifiant masqué (MSID) de l'identifiant du capteur est calculé. Pendant la phase d'authentification, seul le MSID est transmis, ce qui rend le nœud du capteur toujours anonyme.

- **Anonymat de l'utilisateur distant:** L'utilisateur distant ne partage jamais son identité pendant la phase d'authentification ce qui en fait toujours un élément anonyme dans le réseau.

- **Extensibilité et évolutivité:** Pendant la phase d'enregistrement, de nombreux nœuds de capteurs peuvent participer à l'enregistrement et ils seront enregistrés avec leur identifiant et leur MSID calculé, ce qui rend le schéma proposé toujours extensible.

- **La technologie blockchain :** Le blockchain est une nouvelle technologie qui domine le monde de la sécurité et de la confidentialité des données et des communications dans les réseaux de communication comme Internet, etc., en raison du mécanisme de sécurité offert par cette technologie. Dans notre schéma proposé, nous avons utilisé le blockchain comme base de données. pour stocker l'identité des nœuds participants dans le processus d'authentification, ce qui fait que notre schéma proposé résiste à ces types de menaces :

  - **Les attaques sur les bases des données :** l'identité des nœuds est stockée de manière distribuée, tous les nœuds participants à l'intérieur du réseau blockchain ont une copie des données stockées, ce qui rend presque impossible pour un attaquant de cibler les données stockées.

  - **Usurpation d'identité :** dans notre schéma proposé, l'authentification doit être effectuée à chaque fois que le capteur ou l'utilisateur se connecte au réseau, et chaque authentification est enregistrée dans , ce qui facilite aux administrateurs de suivre les authentifications et détecter le faux capteur / utilisateur en surveillant les comportements des capteurs et des utilisateurs à l'intérieur du réseau.

  - **Le risque de perdre l'appareil d'un utilisateur distant ou la passerelle :** Si l'appareil d'un utilisateur distant ou une passerelle tombe en panne, l'identité des capteurs ne sera jamais perdue car elle est stockée de manière distribuée, toutes les identités et l'historique de l'authentification sont stockés dans tous les nœuds participant au réseau blockchain.

Le point principal de notre schéma d'authentification est le nœud passerelle, car il reçoit l'adresse blockchain qui est l'adresse où les identifiants du nœud capteur et de l'utilisateur distant sont stockés. Cette adresse est partagée avec l'utilisateur distant, et une copie des données stockées est partagée avec tous les nœuds participants dans le blockchain, ce qui nous conduit à :

- ✓ Pour obtenir les données stockées, le blockchain dispose d'un mécanisme de contrôle d'accès qui permet uniquement aux utilisateurs qui ont l'adresse utilisée pour stocker les données de les obtenir. Ce qui évite que les données ne soient mal utilisées par des utilisateurs indésirables, les copies des données stockées dans les périphériques différents sont hachées, de sorte que les autres nœuds ne peuvent pas voir les données stockées, à moins qu'ils n'aient l'adresse d'enregistrement.

- ✓ Avoir le nœud de passerelle comme point principal de la gestion de la sécurité et des identités oblige à avoir un administrateur qui permet de garder une trace des nœuds avec lesquels l'adresse d'enregistrement est partagée et de surveiller également l'utilisation de l'adresse. Le blockchain dispose d'un mécanisme qui aide l'administrateur à obtenir les informations à chaque fois qu'une authentification entre un utilisateur distant et un nœud de capteur a été effectuée.

- ✓ Après une phase d'enregistrement le nœud passerelle stocke dans l'identifiant du capteur (id) et son identifiant masqué (MSID), la valeur secrète X et l'algorithme de hachage et l'identifiant de l'utilisateur distant avec son mot de passe qui nécessite environ 70 octets. Nous supposons que la passerelle est connectée à 100 des capteurs et 10 utilisateurs, s'ils se sont tous enregistrés les uns avec les autres, nous obtenons 1000 enregistrements et 70000 octets à stocker. Nous supposons que le nœud de passerelle est dans un réseau blockchain où il y a 100 nœuds, et tous ont la même quantité de données ce qui signifie que tous auront une quantité de 7000000 octets ou 7 Mo. Le nœud passerelle peut avoir des Gigas d'octets comme capacité de stockage ce qui signifie que le coût de stockage dans le nœud de passerelle n'est pas un problème dans notre schéma proposé.

4. Conclusion :

Dans ce chapitre nous avons présenté notre schéma d'authentification basé sur la technologie blockchain dans le contexte IoT. Nous avons utilisé comme une base des données distribuée et nous avons analysé théoriquement le schéma en termes de performances et de sécurité. Cependant, une implémentation des fonctionnalités du schéma donnera une meilleure vue sur le schéma et ses avantages. Cela fait l'objet du chapitre suivant.

Chapitre 4: Implémentation & Tests

Dans ce chapitre, nous commençons par décrire les outils et les technologies que nous avons utilisés pour mettre en œuvre la solution que nous avons proposée. Ensuite, nous présentons comment nous avons utilisé le blockchain pour stocker les identités et sécuriser la connexion entre le capteur et l'utilisateur, tout cela dans un simulateur qui contient l'utilisateur, la passerelle et les capteurs. Enfin, nous allons inclure quelques interfaces de notre simulateur.

1. Environnement de développement :

Nous avons créé un simulateur sous forme d'une application DAPP basé sur la technologie blockchain Ethereum, comme le montre la figure 18 suivante :

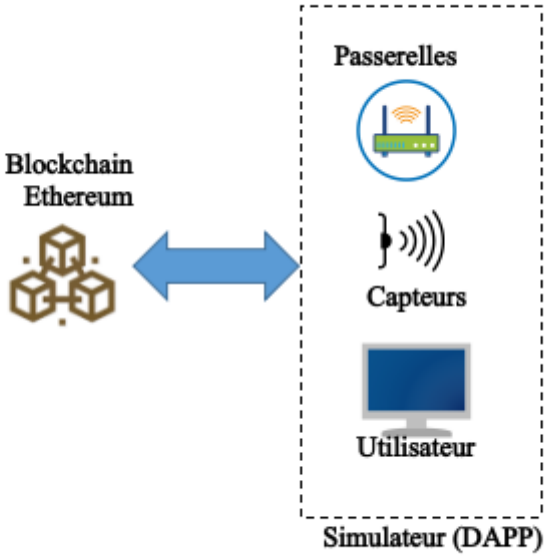

Figure 17: Environnement de développement.

- **Coté blockchain :** nous avons opté pour Ethereum (Figure 19) que nous allons décrire en détails dans la section suivante.

- **Coté application décentralisée :** nous avons utilisé Web3j qui permet aux utilisateurs d'interagir avec un nœud Ethereum en utilisant le langage de programmation Java.

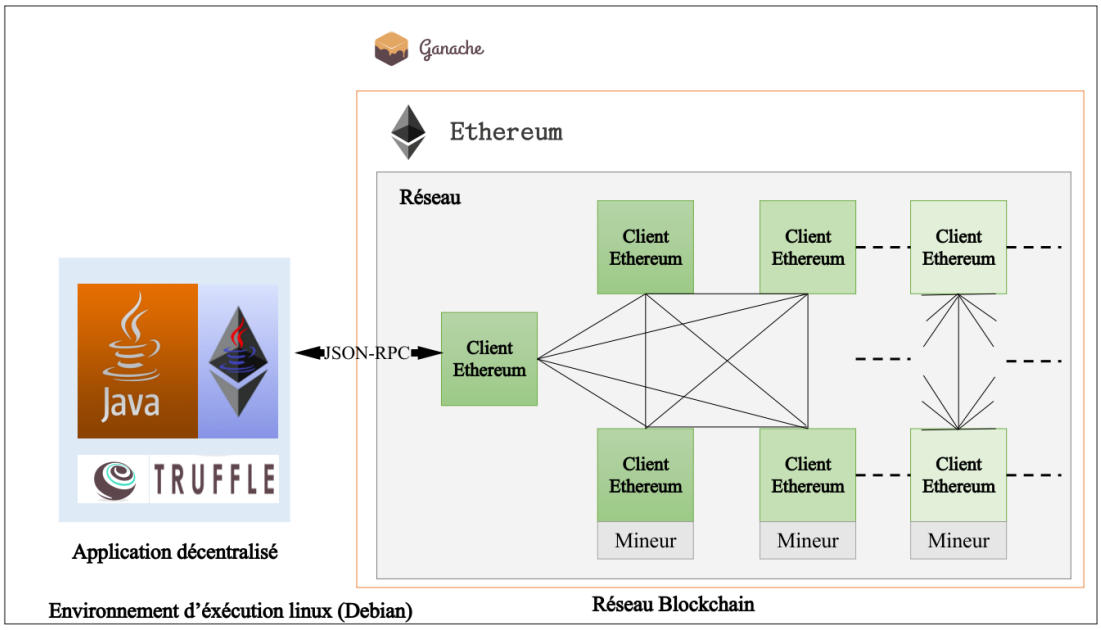

Figure 18 Technologies utilisées

1.1. Blockchain Ethereum :

Il existe un seul ordinateur canonique dans l'univers Ethereum (connu sous le nom de machine virtuelle Ethereum ou EVM) dont tout le monde sur le réseau Ethereum s'accorde sur l'état. Chaque nœud Ethereum (participant au réseau Ethereum) enregistre une copie de l'état actuel de cet ordinateur. Quel que soit le participant, il peut également diffuser une demande pour que cet ordinateur effectue le calcul de son choix. Lorsqu'une telle requête est diffusée, d'autres membres du réseau vérifient, valident et effectuent (ou « exécutent ») le calcul. L'état de l'EVM change en conséquence, et cela est validé et propagé dans tout le réseau.

On appelle "requêtes de transaction" les requêtes de calcul ; l'enregistrement de toutes les transactions, ainsi que l'état actuel de l'EVM, sont conservés dans , qui est ensuite stocké et accepté par tous les nœuds.

Les mécanismes cryptographiques garantissent que les transactions ne peuvent pas être falsifiées après avoir été vérifiées comme valides et ajoutées à ; les mêmes mécanismes garantissent également que toutes les transactions sont signées et exécutées avec les « autorisations » appropriées (seule Alice devrait être en mesure d'envoyer des actifs numériques à partir du compte d'Alice)[78].

Les principaux composants de Ethereum[78]:

- **Les nœuds :** L'état EVM est stocké sur des machines réelles. Les nœuds échangent des informations sur l'état de l'EVM et les nouvelles modifications d'état entre eux. En diffusant une requête d'exécution de code depuis un nœud,

tout utilisateur peut demander l'exécution de code. Le réseau Ethereum est composé de tous les nœuds Ethereum ainsi que de leurs communications.

- **EVM :** La machine virtuelle Ethereum est l'ordinateur virtuel global dont chaque participant sur le réseau Ethereum stocke et accepte l'état. Tout participant peut demander l'exécution de code arbitraire sur l'EVM ; l'exécution du code change l'état de l'EVM..

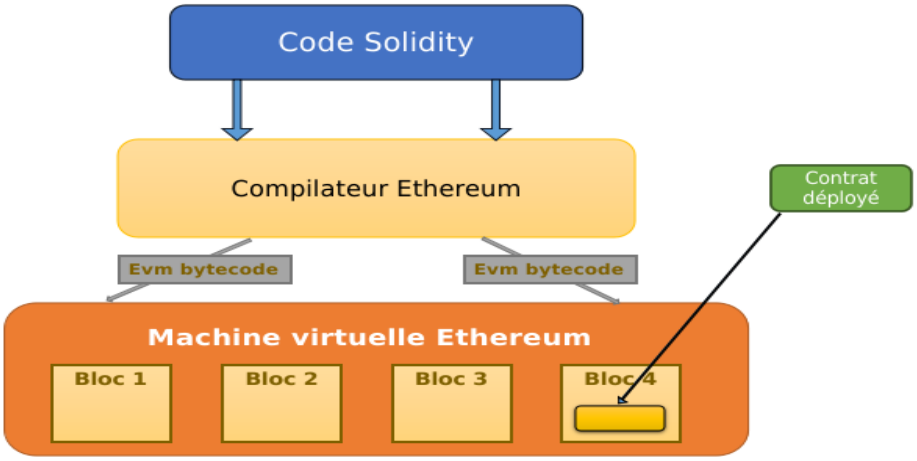

Figure 19: Machine virtuelle Ethereum

- **Les transactions :** Une « demande de transaction » est le mot formel pour une demande d'exécution de code sur l'EVM, tandis qu'une « transaction » est une demande de transaction terminée et le changement associé dans l'état de l'EVM. A partir d'un nœud, tout utilisateur peut diffuser une demande de transaction vers le réseau. Toute exécution de code provoque un changement d'état dans l'EVM, qui est diffusé à tous les nœuds du réseau lors de l'engagement.

- **Les blocs :** Les blocs sont des collections de transactions liées par un hachage du bloc précédent dans la chaîne. Étant donné que les hachages sont formés de manière cryptographique à partir de données de bloc, cela connecte les blocs (dans une chaîne) et empêche la fraude car une modification dans n'importe quel bloc précédent invalide tous les blocs suivants car tous les hachages suivants changent, et tous ceux qui exécutent le blockchain le remarquent.

- **Les contrats intelligents :** Tout simplement, un "contrat intelligent" est un logiciel qui s'exécute sur d'Ethereum. C'est une collection de code (ses fonctions) et de données (son état) qui vit sur Ethereum à une seule adresse. Ils ne sont pas contrôlés par l'utilisateur ; au lieu de cela, ils sont déployés sur le réseau et exécutés comme programmé. Les comptes d'utilisateurs peuvent ensuite interagir avec un contrat intelligent en envoyant des transactions qui entraînent l'exécution d'une fonction par le contrat intelligent. Les contrats intelligents, comme les contrats réguliers, peuvent définir des règles et les faire appliquer

automatiquement via la programmation. Par défaut, les contrats intelligents ne peuvent pas être supprimés et les interactions avec eux sont irréversibles, Les contrats intelligents peuvent être écrits à l'aide de langages relativement conviviaux pour les développeurs, ce qui est une fonctionnalité intéressante d'Ethereum, **Solidity** est **un** langage le plus actif et le mieux entretenu.

- **Solidity :** Solidity est un langage à accolades de haut niveau orienté objet pour la mise en œuvre de contrats intelligents. Il est influencé par C++, Python et JavaScript, et est conçu pour cibler la machine virtuelle Ethereum (EVM). Il est de type statique, prend en charge l'héritage, les biblIoThèques et les types complexes définis par l'utilisateur, entre autres fonctionnalités[79].

Le tableau 6 représente le code que nous avons utilisé pour créer le contrat intelligent qui est dans notre travail :

Code Solidity	Explication du code
pragma solidity ^0.8.4;	La version de Solidity que la compilation utilisera pour compiler le programme
constructor () { owner=msg.sender; }	Le constructeur du contrat, ce constructeur initialise l'adresse du propriétaire de ce contrat (l'adresse de celui qui l'a déployé).
modifier onlyOwner (){ require(msg.sender == owner); _; }	Le modificateur, ce modificateur précise qui peut utiliser la fonction à laquelle il est connecté, la seule adresse autorisée ici est l'adresse du propriétaire
struct identifiers { string id; string X; string hashAlgo; string msid; string uid;string upswd; }	La structure que nous avons utilisée pour stocker les identifiants du capteur et de l'utilisateur dans
mapping(string => identifiers) identifiers_list;	La création de tableau qui contient les informations des capteurs et des utilisateurs avec lesquels les capteurs sont enregistrés (l'indice d'enregistrement est le MSID de capteur).
event create (string id, string X, string hashAlgo,string indexed msid, string uid,string upswd);	L'événement de création d'une connexion (la phase d'enregistrement)
function setIdentifiers (string memory _id, string memory _X,string memory _hashAlgo,string memory _msid,string memory _uid,string memory _upswd) public onlyOwner{	La fonction que nous avons utilisée pour stocker les données dans .

identifiers_list[_msid].id=_id;
identifiers_list[_msid].X=_X;
identifiers_list[_msid].hashAlgo
=_hashAlgo;
identifiers_list[_msid].msid=_m
sid;
identifiers_list[_msid].uid=_uid;
identifiers_list[_msid].upswd=_
upswd;
 	
emit create
(_id,_X,_hashAlgo,_msid,_uid,_
upswd);
}	

function getID(string memory
msid) public view onlyOwner
returns (string memory) {
emit getting_id
(identifiers_list[msid].id);
return identifiers_list[msid].id;
}
	La fonction pour récupérer l'id du capteur, il y a des getters pour toutes les données stockées donc on peut récupérer toutes les informations, tous les fonctions sont signés par le modificateur que nous avons présenté plus haut, donc seulement les utilisateurs qui ont l'adresse du déploiement du contrat intelligent peuvent obtenir ces informations ou les stocker.

emit create (_id,_X,_hashAlgo,_msid,_uid,_ upswd);	Cette ligne représente l'envoie de l'événement «create» afin qu'un administrateur ou un contrôleur puisse écouter à l'événement et obtenir l'information lorsqu'un capteur est enregistré auprès d'un utilisateur.

Tableau7 : La création du contrat intelligent

1.2. Web3 Java Ethereum Dapp API (Web3j)

Nous avons travaillé avec le langage de programmation java, et pour implémenter les fonctionnalités d'Ethereum avec java, nous avons besoin d'utiliser le framework web3j. Web3j permet de travailler avec des contrats intelligents et de se connecter aux

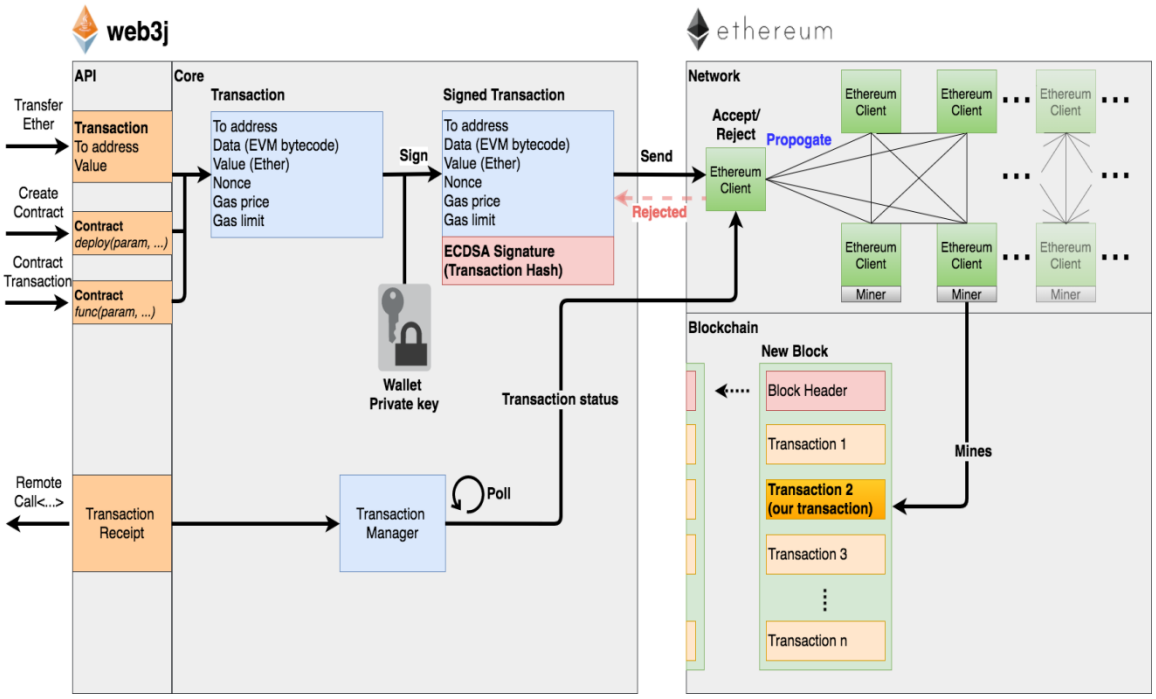

Figure 20:Les transactions Ethereum - web3j [80]

clients du réseau Ethereum (comme illustré dans la figure 21), qui est légère, hautement modulaire, réactive et sécurisée [77].
Les caractéristiques de Web3j sont [80]:

- Implémentation complète de l'API client Ethereum JSON-RPC sur HTTP et IPC et prise en charge du portefeuille Ethereum.
- Génération automatique de wrappers de contrats intelligents Java pour créer, déployer, traiter et appeler des contrats intelligents à partir de code Java natif (formats de définition Solidity et Truffle pris en charge).
- API fonctionnelle réactive pour travailler avec des filtres.
- Prise en charge du service de noms Ethereum (ENS).
- Prise en charge des nœuds Ethereum hébergés.
- Prise en charge des normes de jetons ERC20 et ERC721.
- Outils en ligne de commande.
- Compatible Android.

Dans notre mise en œuvre, nous avons utilisé les commandes suivantes :

✓ **Installation** :$ curl -L get.web3j.io | sh && source ~/.web3j/source.sh

✓ **Création d'un nouveau projet :**$ web3j new

✓ **Génération de wrappers d'un contrat intelligent (figure 22):** $ web3j import

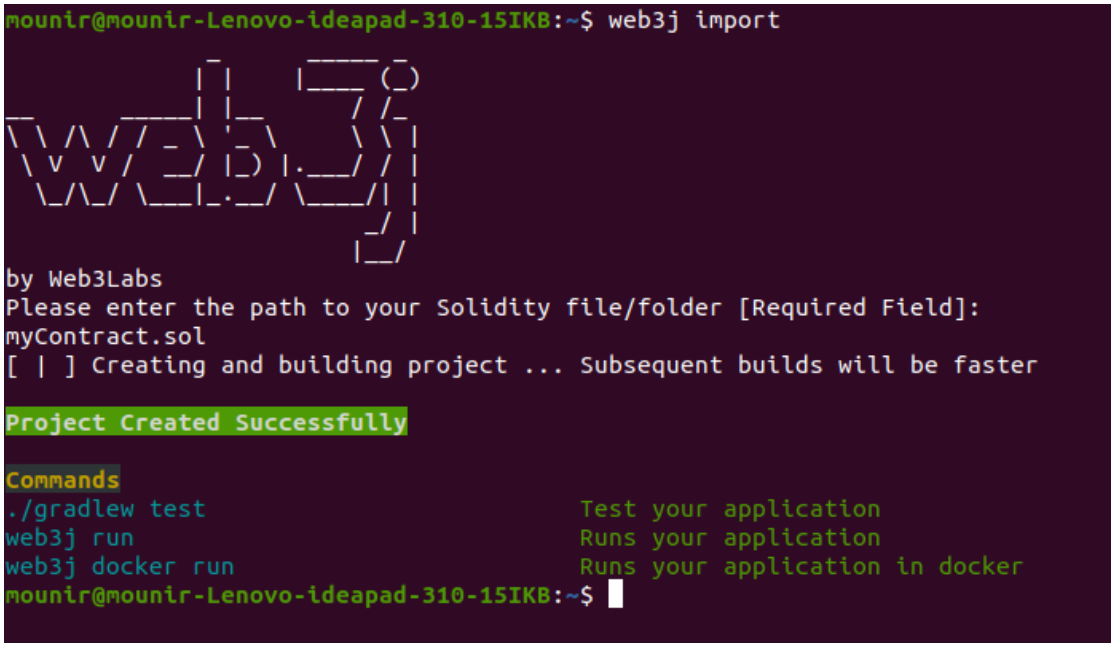

Figure 21: Génération de wrappers d'un contrat intelligent

Voici un aperçu de fichier «connexion.java» résultant de cette commande (le fichier complet est dans : https://github.com/mounirProgrammer/IoT_authentication_simulator_blockchain dans le package «pfe») :

```
public RemoteCall<String> getX(String msid) {
    final org.web3j.abi.datatypes.Function function = new org.web3j.abi.datatypes.Function(FUNC_GETX,
            Arrays.<Type>asList(new org.web3j.abi.datatypes.Utf8String(msid)),
            Arrays.<TypeReference<?>>asList(new TypeReference<Utf8String>() {}));
    return executeRemoteCallSingleValueReturn(function, String.class);
}

public RemoteCall<TransactionReceipt> setIdentifiers
    (String _id, String _X, String _hashAlgo, String _msid, String _uid, String _upswd) {
    final org.web3j.abi.datatypes.Function function = new org.web3j.abi.datatypes.Function(
            FUNC_SETIDENTIFIERS,
            Arrays.<Type>asList(new org.web3j.abi.datatypes.Utf8String(_id),
            new org.web3j.abi.datatypes.Utf8String(_X),
            new org.web3j.abi.datatypes.Utf8String(_hashAlgo),
            new org.web3j.abi.datatypes.Utf8String(_msid),
            new org.web3j.abi.datatypes.Utf8String(_uid),
            new org.web3j.abi.datatypes.Utf8String(_upswd)),
            Collections.<TypeReference<?>>emptyList());
    return executeRemoteCallTransaction(function);
}
```

Figure 22:Wrapper d'un smart contrat

1.3. Truffle

Truffle est un environnement de développement de classe mondiale, un cadre de test et un pipeline d'actifs pour les chaînes de blocs utilisant la machine virtuelle Ethereum (EVM), visant à faciliter la vie de développeur. Avec Truffle, on obtient [81]:

- Compilation, liaison, déploiement et gestion binaire de contrats intelligents intégrés.
- Tests de contrats automatisés pour un développement rapide.
- Framework de déploiement et de migrations extensible et scriptable.
- Gestion de réseau pour le déploiement sur n'importe quel nombre de réseaux publics et privés.
- Gestion des packages avec EthPM & NPM, utilisant la norme ERC190.
- Console interactive pour la communication directe des contrats.
- Pipeline de build configurable avec prise en charge d'une intégration étroite.
- Lanceur de script externe qui exécute des scripts dans un environnement Truffle.

Dans notre projet, nous avons utilisé les commandes suivantes :

- ✓ **Installation :**$ npm install -g truffle

- ✓ **Initialisation :**$ npm truffle init

- ✓ **Compiler un smart contrat :** Pour compiler un contrat il faut le mettre dans le fichier /contracts créé par la commande init et exécuter la commande: $ npm truffle compile

1.4. Ganache

Ganache est un blockchain personnel pour le développement rapide d'applications distribuées Ethereum et Corda. Il peut être utilisé tout au long du cycle de développement, en permettant de développer, déployer et tester les applications décentralisées dans un environnement sécurisé et prévisible. Ganache est disponible en deux versions : une interface utilisateur et une interface de ligne de commande [82].

Pour avoir la figure 24, il faut suivre les étapes suivantes :

- Commande de téléchargement pour le ganache-cli:

$ npm install ganache-cli-[version] -g

- Commande de lancement :

$ ganache-cli

- Pour Ganache UI, il faut télécharger le fichier ganache.AppImage et le mettre exécutable avec la commande

$ chmod a+x ganache.AppImage

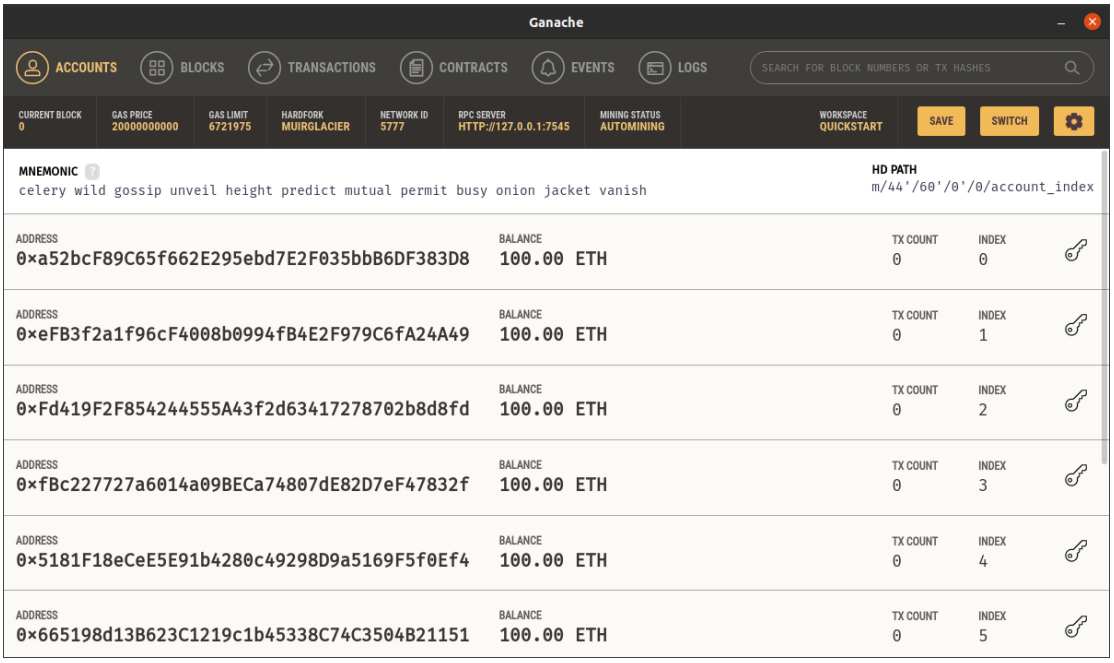

Figure 23: Interface Ganache

1.5. Java

Java est un langage de programmation orienté objet, développé par Sun Microsystems. Il permet de créer des logiciels compatibles avec de nombreux systèmes d'exploitations (Windows, Linux, Macintosh, Solaris). Pour créer notre application java nous avons utilisé l'environnement eclipse et les API Java suivants :

- **Java swing :** que nous avons utilisé pour créer les interfaces graphique de notre simulateur.

- **Apache maven :** est un outil de gestion de projet et de compréhension de projets logiciels. Maven peut gérer la construction, le reporting et la documentation d'un projet à partir d'une information centrale grâce au concept de modèle objet de projet (POM) [83].

Pour créer une application DAPP en utilisant java maven il est nécessaire d'ajouter quelques dépendances dans le fichier pom, voici le contenu de fichier «pom.xml» dans notre projet :

```xml
<project xmlns="http://maven.apache.org/POM/4.0.0" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 https://maven.apache.org/xsd/maven-4.0.0.xsd">
  <modelVersion>4.0.0</modelVersion>
  <groupId>io.kauri.tutorials.java-ethereum</groupId>
  <artifactId>java-ethereum</artifactId>
  <version>0.0.1-SNAPSHOT</version>
  <properties>
    <maven.compiler.target>1.8</maven.compiler.target>
    <maven.compiler.source>1.8</maven.compiler.source>
  </properties>
  <dependencies>
    <dependency>
      <groupId>org.web3j</groupId>
      <artifactId>core</artifactId>
      <version>4.3.0</version>
    </dependency>
  </dependencies>
</project>
```

Figure 24: Fichier pom.xml

2. Implémentation de notre schéma d'authentification

Pour implémenter notre simulateur, nous avons créé trois classes pour chaque nœud dans le réseau. Voici les constructeurs de ces classes :

```
public Utilisateur(String uid, String upswd, Connection_gen generator) {
    this.uid = uid;
    this.upswd = upswd;
    this.generator=generator;

}
```

Figure 25: Constructeur de la classe «Utilisateur»

```
public Passerelle(String id_pas,Connection_gen generator){
    this.id_pas=id_pas;
    this.generator=generator;

}
```

Figure 26: Constructeur de la classe «Passerelle»

```
public Capteur(String ID,String algoHash,String X){
    this.ID=ID;
    this.algoHash=algoHash;
    this.X=X;
    MSID=H(ID+X,algoHash);

}
```

Figure 27 : Constructeur de la classe «Capteur»

Pour l'enregistrement et l'authentification des capteurs, nous avons créé une classe «Enreg-Auth» mettant en place ces deux phases. Voici le constructeur de cette classe :

```
public Enreg_Auth (Passerelle pas, Capteur cap, Utilisateur user){
    this.pas=pas;
    this.cap = cap;
    this.user = user;
}
```

Figure 28 : Constructeur de la class «Enreg_Auth»

Cette classe contient deux fonctions («Enreg» et «Auth»). Les valeurs retenus par ces classes seront affichés dans les interfaces enregistrement et d'authentification respectivement. Voici la fonction qui fait l'enregistrement :

```
public String[] Enreg (){
    String[] affichage = new String[3];
    System.out.println("hello there");
    String capt = null,pasr = null, util = null;
    capt = cap.enregMsg1(pas);
    util = user.enregMsg2(pas);
    pasr = pas.enregMsg1(cap, user);
    pasr = pasr + "\n" +pas.enregMsg2(cap, user);
    pasr = pasr + "\n" +pas.enregMsg3(cap, user);
    pasr = pasr + "\n" +pas.enregMsg4(cap, user);
    util = util + "\n" +user.enregMsg4(pas);
    util = util + "\n" +user.enregMsg5(pas);
    pasr = pasr + "\n" +pas.enregMsg5(cap, user);
    pasr = pasr + "\n" +pas.enregMsg6(cap, user);
    capt = capt + "\n" +cap.enregMsg6(pas);
    System.out.println("capt = "+capt);
    System.out.println("pasr = "+pasr);
    System.out.println("util = "+util);
    affichage[0] = capt;
    affichage[1] = pasr;
    affichage[2] = util;
    return affichage;
}
```

Figure 29: La fonction «Enreg»

Et voici la structure de la fonction «Auth» sans traiter les cas de connexion échoué et les attaques (la version complète dans : https://github.com/mounirProgrammer/IoT_authentication_simulator_blockchain)

```
public String[]  Auth (){
    String[] affichage = new String[3];
    String capt = null,pasr = null, util = null,rcv = null,attaque = null;
    String msg1 = null, msg2 = null, msg3 = null, msg4 = null,titre = "";
    capt = cap.authMsg1(user);
    capt = capt+"\n"+cap.authMsg2(user);
    util = user.authMsg2(cap,pas);
    rcv = user.authMsg3(cap,pas);
    util = util+"\n"+rcv;
    util = util+"\n"+user.authMsg4(cap,pas);
    util = util+"\n"+user.authMsg5(cap,pas);
    pasr = pas.authMsg5(user);
    pasr = pasr+"\n"+pas.authMsg6(user);
    rcv = pas.authMsg7(user);
    pasr = pasr+"\n"+rcv;
    pasr = pasr+"\n"+pas.authMsg8(user);
    util = util+"\n"+user.authMsg8(cap,pas);
    rcv = user.authMsg9(cap,pas);
    util = util+"\n"+rcv;
    util = util+"\n"+user.authMsg10(cap,pas);
    capt = capt+"\n"+cap.authMsg10(user);
    util = util+"\n"+user.authMsg11(cap,pas);
    capt = capt+"\n"+rcv;
    affichage[0] = capt;
    affichage[1] = util;
    affichage[2] = pasr;
    return affichage;
}
```

Figure 30: la fonction «Auth»

De plus, pour le chiffrement, nous avons utilisé le chiffrement AES en utilisant la fonction suivante:

```
public void setKey(String myKey)
{
    MessageDigest sha = null;
    try {
        key = myKey.getBytes("UTF-8");
        sha = MessageDigest.getInstance("SHA-1");
        key = sha.digest(key);
        key = Arrays.copyOf(key, 16);
        secretKey = new SecretKeySpec(key, "AES");
    }
    catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    catch (UnsupportedEncodingException e) {
        e.printStackTrace();
    }
}
```

Figure 31: Chiffrement AES

Notons ici que pour la fonction d'hachage, nous avons implémenté trois fonctions différentes : md5[2] (figure 28), SHA (SHA-1, SHA-256 SHA-384, SHA-512)[3] (figure 33) et PBKDF2[4].

```
static public  String md5Hash(String clear) {
    String generatedHash = null;
    try {
        // Create MessageDigest instance for
        MessageDigest md = MessageDigest.getInstance("md5");
        //Get the hash's bytes
        byte[] bytes = md.digest(clear.getBytes());
        //This bytes[] has bytes in decimal format;
        //Convert it to hexadecimal format
        StringBuilder sb = new StringBuilder();
        for (int i = 0; i < bytes.length; i++) {
            sb.append(Integer.toString((bytes[i] & 0xff) + 0x100, 16).substring(1));
        }
        //Get the complete hash in hex format
        generatedHash = sb.toString();
    } catch (NoSuchAlgorithmException e) {
        e.printStackTrace();
    }
    return generatedHash;
}
```

Figure 32: Fonction de hachage MD5

```
static public String shaHash (String clear) {
    String generatedHash = null;

    try {
        MessageDigest md = MessageDigest.getInstance("SHA-1");
        byte[] bytes = md.digest(clear.getBytes());
        StringBuilder sb = new StringBuilder();
        for(int i=0; i< bytes.length ;i++)
        {
            sb.append(Integer.toString((bytes[i] & 0xff) + 0x100, 16).substring(1));
        }
        generatedHash = sb.toString();
    }
    catch (NoSuchAlgorithmException e)
    {
        e.printStackTrace();
    }
    return generatedHash;
}
```

Figure 33 : Fonction de hachage SHA

[2]Algorithme de hachage pouvant être utilisé pour créer une valeur de chaîne de 128 bits à partir d'une chaîne de longueur arbitraire [84].

[3]Une fonction de hachage cryptographique conçue par la *National Security Agency* des États-Unis (NSA)[84].

[4]PBKDF2 défini dans la RFC 2898, est une fonction de dérivation de clé (KDF) spécifique [85].

```
public static String PBKDF2Hash (String clear) {
    int iterations = 1000;
    char[] chars = clear.toCharArray();
    byte[] hash = null;
    String toReturn = null;
    try {
    byte[] salt = getSalt();
    PBEKeySpec spec = new PBEKeySpec(chars, salt, iterations, 64 * 8);
    SecretKeyFactory skf;
                    skf = SecretKeyFactory.getInstance("PBKDF2WithHmacSHA1");
                    hash = skf.generateSecret(spec).getEncoded();
                    toReturn = iterations + ":" + toHex(salt) + ":" + toHex(hash);
            } catch (NoSuchAlgorithmException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            } catch (InvalidKeySpecException e) {
                    // TODO Auto-generated catch block
                    e.printStackTrace();
            }
    return toReturn;

}
```

Figure 34: Fonction de hachage PBKDF2

3. Présentation du simulateur

Dans cette page, l'utilisateur doit mettre ses informations pour accéder (ces informations seront utilisées dans le processus d'authentification) :

Figure 35: Interface de log In

Dans l'interface principale de simulateur, l'utilisateur peut créer des capteurs (Figure 39) et passerelles (Figure 38) et les ajouter dans la liste des capteurs/passerelles (non enregistrés ni authentifiés) situés dans le réseau. Il peut enregistrer des capteurs et authentifier les capteurs de cette liste avec lui.

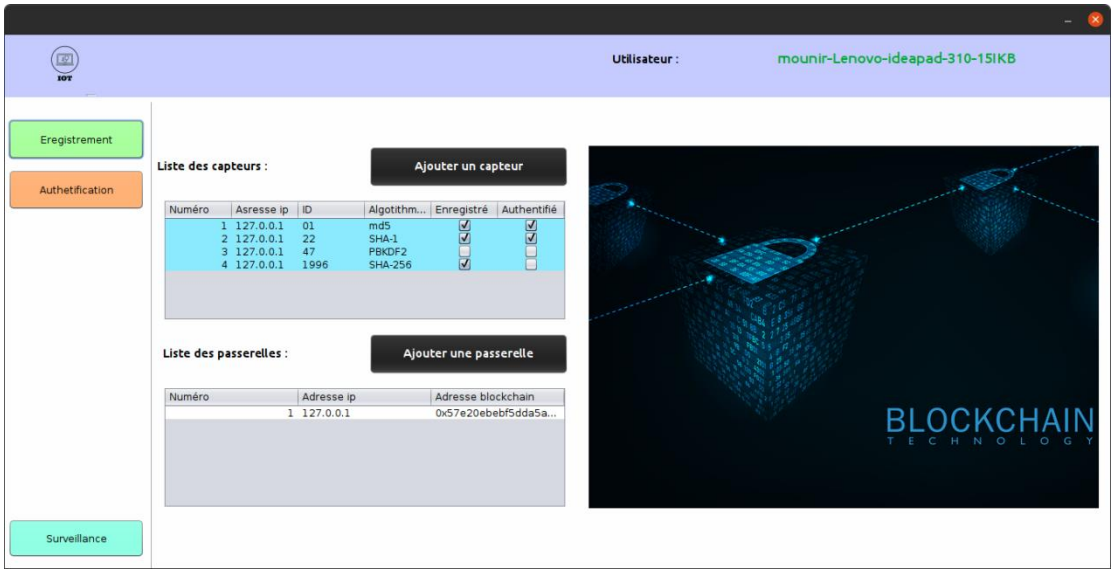

Figure 36: Interface d'un utilisateur

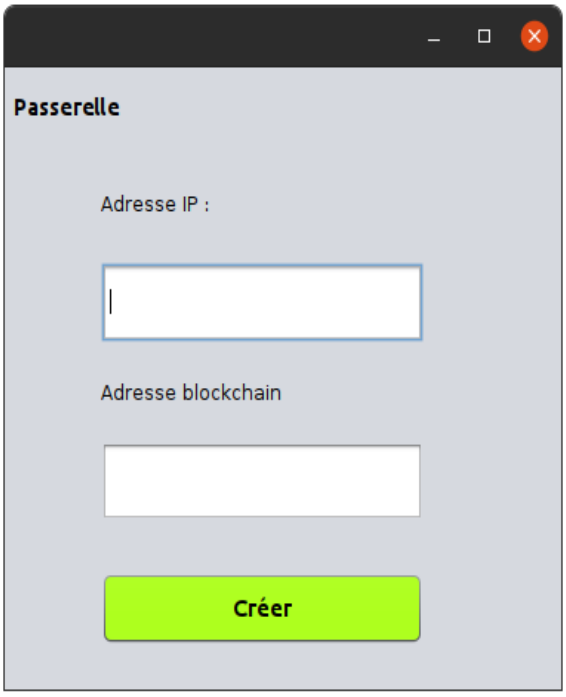

Figure 37: Ajout d'un passerelle

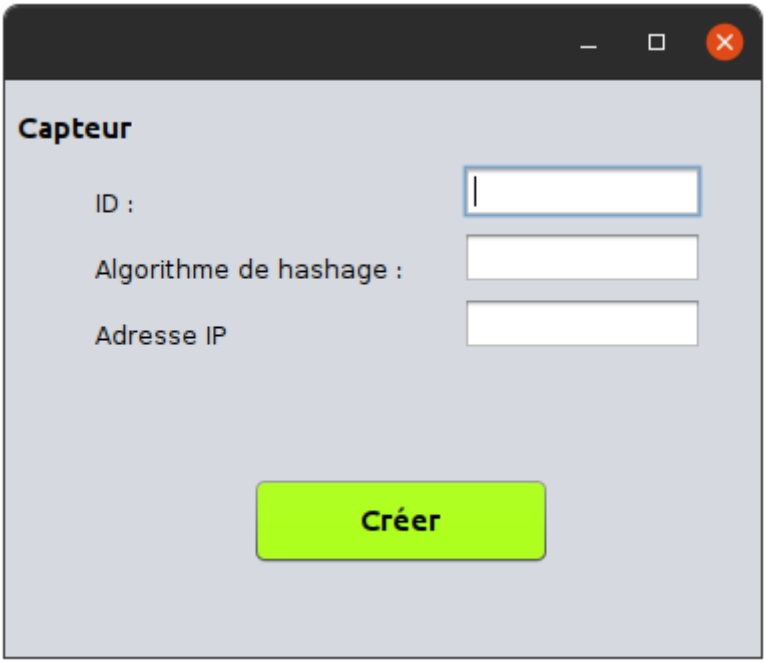

Figure 38: Ajout d'un capteur

En cliquant sur le bouton enregistrer et choisissant le numéro de capteur à enregistrer et la passerelle à utiliser, l'interface suivante s'affiche :

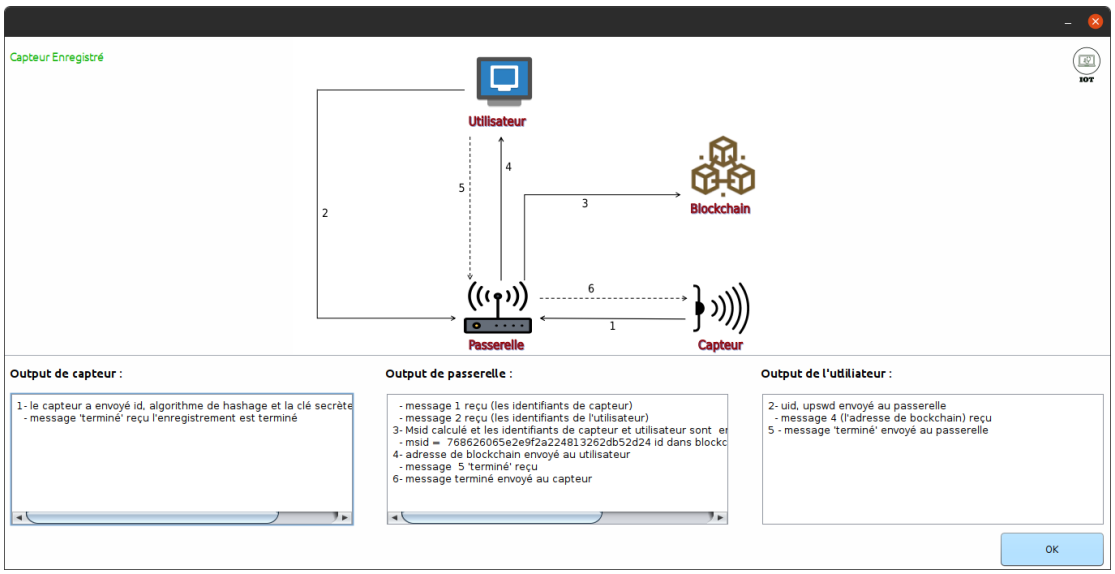

Figure 39: Processus d'enregistrement

En cliquant sur le bouton « authentification », l'utilisateur peut choisir s'il veut faire une authentification simple sans aucune attaque ou une authentification avec l'attaque de l'homme au milieu ou l'attaque par rejeu. Dans les deux derniers cas, il

choisit la partie du message où il veut que l'attaque soit effectuée. La figure 41 suivante permet à l'utilisateur à soumettre son choix:

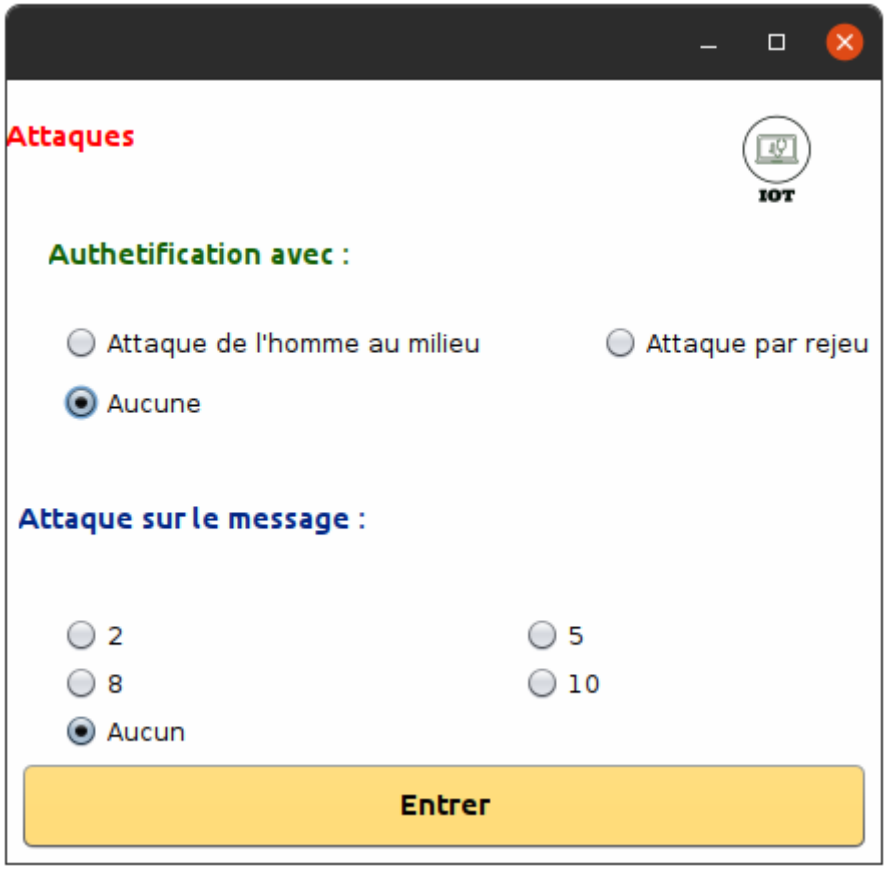

Figure 40: choix d'attaque

Après choix, trois interfaces différentes peuvent s'afficher :

- Figure 42 : si l'utilisateur a voulu une authentification sans aucune attaque.
- Figure 43 : en cas où d'une attaque d'un homme du milieu dans le message 5
- Figure 44 : en cas où d'une attaque par rejeu dans le message 10.

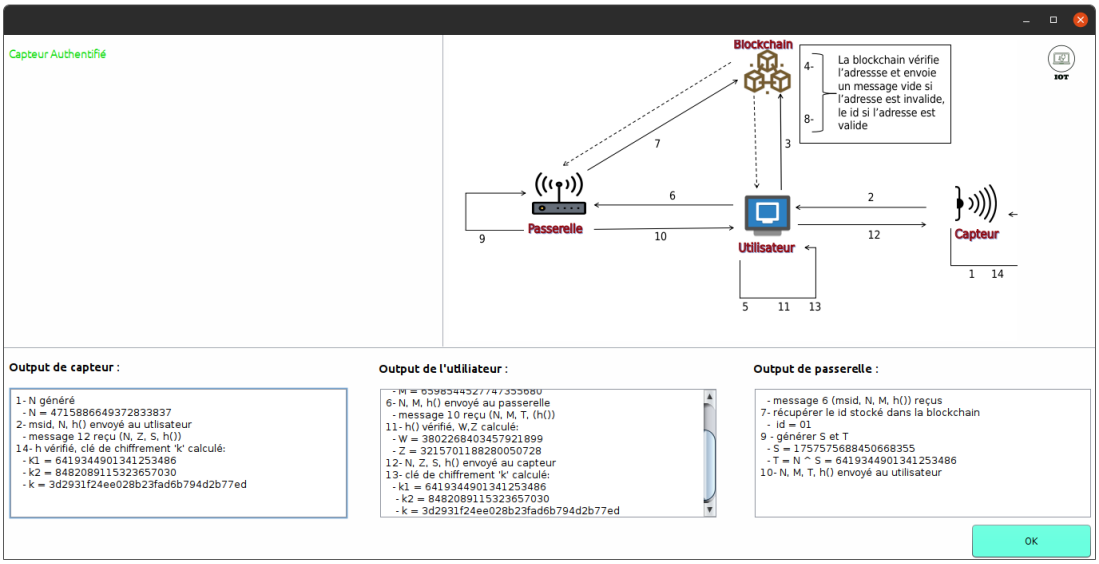

Figure 41: Processus d'authentification

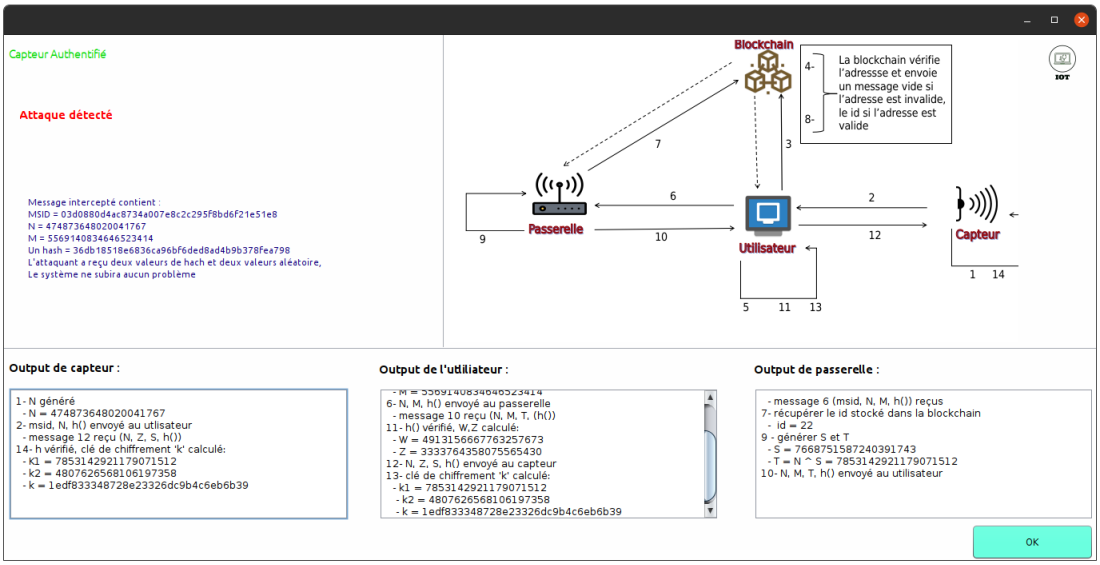

Figure 42:Authentification avec l'attaque de l'homme au milieu dans le message 5

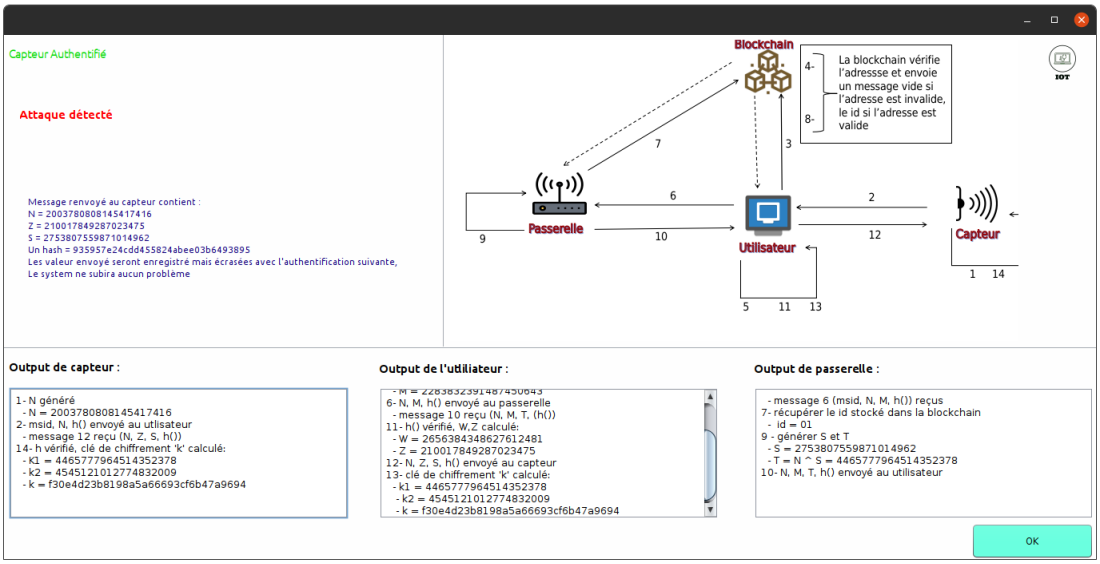

Figure 43: Authentification avec une attaque par rejeu dans le message 10

Comme nous l'avons présenté précédemment, la technologie blockchain a la particularité d'envoyer un message avec tout événement émis. En cliquant sur le bouton "surveillance", une interface (Figure 45) contenant les messages envoyés avec toute inscription apparaîtra.

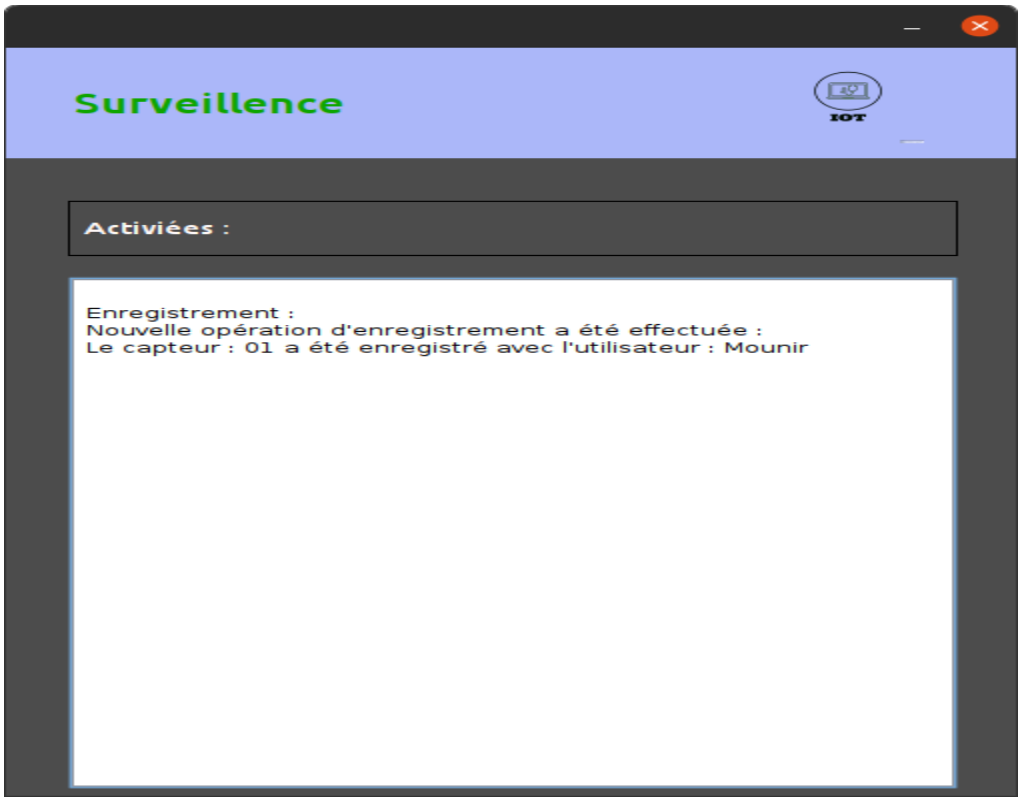

Figure 44: Interface de surveillance

4. Conclusion :

Nous avons présenté dans ce chapitre notre simulateur qui comprend un compte utilisateur, les passerelles et les capteurs, implémenté à l'aide du langage de programmation java et en utilisant la technologie de blockchain Ethereum et du framework web3j. Le simulateur montre la communication entre les appareils du réseau et le processus d'enregistrement et d'authentification suivant notre schème proposé.

Conclusion Générale et Perspectives

Récemment, l'utilisation de l'Internet des objets dans notre vie est en croissance, elle couvre de nombreux domaines différents comme les maisons intelligentes, les villes intelligentes, le domaine de la santé, etc. Ce paradigme est basé sur l'utilisation de différents appareils qui visent à collecter et partager les données avec les utilisateurs distants. L'authentification dans le contexte de l'IoT est très importante. C'est un processus essentiel pour la sécurité et le fonctionnement des appareils IoT, ce qui nécessite un processus d'authentification bien géré pour éviter de nombreuses attaques dangereuses comme les attaques par déni de service, l'attaque d'usurpation d'identité, etc., Pour assurer la sécurité contre ces attaques, un mécanisme de gestion et de sécurité des données est nécessaire pour sécuriser les identités des utilisateurs et des appareils inclus dans le réseau IoT. Le Blockchain est une technologie moderne de stockage et de transmission d'informations. Elle fonctionne sans organe central de contrôle, mais apporte transparence et sécurité grâce à la validation des transactions par les nœuds du réseau.

Dans notre travail, nous avons fait une étude sur l'IoT et son utilisation, notamment dans le domaine de la e-santé. Puis, nous avons présenté une étude sur la technologie blockchain et son utilisation dans l'authentification et ce dans le cadre de l'IoT et de domaine de l'e-santé. Ensuite, nous avons présenté notre schéma d'authentification inspiré du schéma d'authentification dans le contexte IoT du [58] et basé sur la technologie blockchain en tant qu'une base de données distribuée qui offre un haut niveau de sécurité et un moyen sûr de stocker les identités des appareils participants dans le processus d'authentification. L'analyse de la sécurité et des performances de notre schéma prouve que notre schéma a une faible consommation d'énergie et moins d'opérations de calcul au niveau du capteur, et aussi une résistance contre plusieurs cyberattaques liées aux bases de données et au stockage de données, ect. Enfin, nous avons créé un simulateur simple de réseau qui contient un utilisateur, des passerelles et des capteurs avec programmation en utilisant le langage java et le blockchain Etherum. Le but du simulateur est de clarifier le fonctionnement de notre schéma proposé et de faire quelques tests de sécurité qui sont nécessaires.

Comme perspective de notre travail, nous avons quelques propositions pour améliorer le niveau de sécurité du processus d'authentification.

- La mise en œuvre de notre schéma dans un simulateur IoT/Blockchaine ou dans un environnement réel pour avoir une meilleure vision sur le niveau de sécurité et les performances de notre schéma.

- Les protocoles d'authentification sont différents, donc de nombreuses recherches ont été faites dans ce domaine et le niveau de sécurité change d'un protocole à l'autre,

nous proposons d'établir une analyse de sécurité standard, pour faciliter le test du niveau de sécurité et de la protection des données dans n'importe quel protocole.

- Récemment, l'utilisation de l'intelligence artificielle (IA) dans notre monde prouve la puissance de la machine dans de nombreuses tâches. L'IA peut faire beaucoup de tâches qui nécessitent un haut niveau de calculs et aussi un haut niveau d'attention que l'homme ne peut pas faire parfaitement. Donc pour conserver les ressources humaines pour les meilleures tâches et pour obtenir une meilleure performance, nous proposons de créer un système d'IA pour surveiller le trafic dans le réseau pendant le processus d'authentification ainsi que les comportements des nœuds participants dans le réseau (temps d'authentification et d'enregistrement et les changements d'identités, ect) et cela aide à prévenir certains types d'attaques comme les attaque Dos et d'usurpation d'identité et tant d'autres menaces.

Références :

[1]Khemissa, H., & Tandjaoui, D. (2016, April). A novel lightweight authentication scheme for heterogeneous wireless sensor networks in the context of Internet of Things. In 2016 IEEE Wireless Telecommunications Symposium (WTS), pp. 1-6.

[2] National Intelligence Council. Disruptive Civil Technologies – Six Technologies with potential impacts on US Interests out to 2025 - Conference Report CR 2008-07. April 2008. Available at http://www.dni.gov/nic/NIC_home.html

[3] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," Computer networks, vol. 54, no. 15, pp. 2787-2805, 2010.

[4] Giusto, D. (2010). A. lera, g. morabito, l. atzori (eds.) the internet of things.

[5] INFSO D.4 Networked Enterprise & RFID INFSO G.2 Micro & Nanosystems , in co-operation with the Working Group RFID of the ETP EPOSS, "Internet of Things in 2020, Roadmap for the Future", Version 1.1 - 27 May, 2008

[6] Srivastava. Pervasive, Ambient, Ubiquitous: the Magic of Radio. European Commission Conference "From RFID to the Internet of Things", Bruxelles, Belgium, March 2006.

[7] Michael, J., Cohn, A. L. A. N., & Butcher, J. R. (2018). Blockchain technology. The Journal, 1(7).

[8] D. Miorandi, S. Sicari, F. De Pellegrini, I. Chlamtac, Survey internet of things: Vision, applications and research challenges, Ad Hoc Netw. 10 (7) (2012) 1497–1516.

[9] H. Khemissa and D. Tandjaoui, "A lightweight authentication scheme for ehealth applications in the context of internet of things," in 2015 9th International Conference on Next Generation Mobile Applications, Services and Technologies. IEEE, 2015, pp.90-95.

[10] Home | MHQ [Internet]. mHealth Quality. [cité 17 mars 2017]. Disponible sur: http://www.mHealth-Quality.eu

[11] T. Limbasiya and N.Doshi. An analytical study of biometric based remote user authentication schemes using smart cards.Computers & Electrical Engineering 2017

[12] M. Palattella, N. Accettura, X. Vilajosana, T. Watteyne, L. Grieco, G. Boggia, M.Dohler, Standardized protocol stack for the internet of (im- portant) things, Communications Surveys Tutorials, IEEE 15 (3) (2013) 1389–1406.

[13] S. Sicari, A. Rizzardi, L. A. Grieco, and A. Coen-Porisini, "Security, privacy and trust in internet of things: The road ahead," Computer networks, vol. 76, pp. 146-164, 2015.

[14] N. Ye, Y. Zhu, R.-C. b. Wang, R. Malekian, Q.-M. Lin, An efficient authentication and access control scheme for perception layer of internet ofvthings, Applied Mathematics and Information Sciences 8 (4) (2014) 1617–1624.

[15] R. H. Weber, Internet of t hings - new security and privacy challenges, Computer Law & Security Review 26 (1) (2010) 23–30.

[16] H. Feng, W. Fu, Study of recent development about privacy and security of the internet of things, in: 2010 International Conference on Web Information Systems and Mining (WISM), Sanya, 2010, pp. 91–95.

[17] R. Roman, J. Zhou, J. Lopez, On the features and challenges of security and privacy in distributed internet of things, Computer Networks 57 (10) (2013) 2266–2279.

[18] J. will Anderson, thrive L.Rainie, 2025, The internet of things Internet Project, http://www.pewinternet.org/2014/05/14/internet-of-things/, May 2014.

[19] NIST-CSRC. Computer security resource center - online glossary. Online, 2020. (Cited on pages 38, 39, 40 and 41.).

[20] Musa G. Samaila, Miguel Neto, Diogo A. B. Fernandes, Mário M. Freire, and Pedro R. M. Inácio. Security Challenges of the Internet of Things, pages 53–82. Springer International Publishing, Cham, 2017. (Cited on pages 39 and 40.)

[21] COMMISSION EUROPÉENNE,«Une meilleure protection et de nouvelles perspectives - Orientations de la Commission relatives à l'application directe du règlement général sur la protection des données à partir du 25 mai 2018[online].Mai 2018,https://eur-lex.europa.eu/legal-

content/FR/TXT/PDF/?uri=CELEX:52018DC0043&from=FR

[22] R. Roman, J. Zhou, and J. Lopez, "On the features and challenges of security and
privacy in distributed internet of things," Computer Networks, vol. 57, no. 10, pp. 2266 – 2279, 2013. Towards a Science of Cyber Security Security and Identity Architecture for the Future Internet.

[23] Vasilomanolakis, E., et al.: On the security and privacy of internet of things architectures and systems. In: 2015 International Workshop on Secure Internet of Things (SIoT). IEEE (2015)

[24] Rose, K., Eldridge, S., Chapin, L.: The Internet of Things: An Overview, pp. 1–50. The Internet Society (ISOC) (2015)

[25] Levitt, T.: Internet of Things: IoT Governance, Privacy and Security Issues (2015)

[26] Leister, W., Schulz, T.: Ideas for a trust indicator in the internet of things. In: SMART (2012)

[27] Fritsch, L., Groven, A.-K., Schulz, T.: On the internet of things, trust is relative. In: International Joint Conference on Ambient Intelligence. Springer (2011)

[28] Ion, M., et al.: A peer-to-peer multidimensional trust model for digitalecosystems. In: 2008 2nd IEEE International Conference on Digital Ecosystems and Technologies. IEEE (2008)

[29] Daubert, J., Wiesmaier, A., Kikiras, P.: A view on privacy & trust in IoT. In: 2015 IEEE International Conference on Communication Workshop (ICCW). IEEE (2015)

[30] Eder, T., Nachtmann, D., Schreckling, D.: Trust and Reputation in the Internet of
Things (2013)

[31] Sicari, S., et al.: Security, privacy and trust in internet of things: the road ahead. Comput. Netw. 76, 146–164 (2015)

[32] Mohamed Amine Ferrag, Leandros A. Maglaras, Helge Janicke, Jianmin Jiang, and Lei Shu. Authentication Protocols for Internet of Things : A Comprehensive Survey.
2017 :1 41, 2017.

[33] Mardiana binti Mohamad Noor and Wan Haslina Hassan. Current research on Internet of Things (IoT) security : A survey. 148 :283–294, 2019.

[34] Loıc Ferreira. IoT & sécurité : Retour vers le futur ?, 2017.

[35] B Fouladi and S Ghanoun. Sécurity Evaluation of the Z-Wave Wireless Protocol. Page 6, 2013.

[36] Sooyeon Shin and Taekyoung Kwon. Two-Factor Authenticated Key Agreement Supporting Unlinkability in 5G-Integrated Wireless Sensor Networks. 6 :11229–11241, 2018.

[37] LI, Ruinian, SONG, Tianyi, MEI, Bo, *et al.* Blockchain for large-scale internet of things data storage and protection. *IEEE Transactions on Services Computing*, 2018, vol. 12, no 5, p. 762-771.

[38] Radek Fujdiak, Petr Blazek, Konstantin Mikhaylov, Lukas Malina, Petr Mlynek, Jiri Misurec, and Vojtech Blazek. On Track of Sigfox Confidentiality with End-to-End Encryption. In Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, pages 19 :1–19 :6. ACM, 2018.

[39] Usman Raza, Parag Kulkarni, and Mahesh Sooriyabandara. Low Power Wide Area Networks : An Overview, 2016

[40]51/ FIPS 197, Advanced Encryption Standard (AES). page 51.

[41] S. Goldfeder, R. Gennaro, H. Kalodner, J. Bonneau, J. A. Kroll, E. W. Felten, and A. Narayanan, “Securing bitcoin wallets via a new dsa/ecdsa threshold signature scheme,” 2015,
http://stevengoldfeder.com/papers/threshold_sigs.pdf

[42] C. Rackoff and D. R. Simon, “Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack,” in Proc. Annu. Int. Cryptology Conf., 1991, pp. 433–444.

[43]C. Marchand, L. Bossuet, U. Mureddu, N. Bochard, A. Cherkaoui, and V. Fischer.
Implementation and characterization of a physical unclonable function for IoT: A case study with the tero-puf. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 37(1):97–109, 2018. (Cited on page 44.)

[44] J. Yang, Y. Lin, Y. Fu, X. Xue, and B. A. Chen. A small area and low power true random number generator using write speed variation of oxidebased rram for IoT security application. In 2017 IEEE International Symposium onCircuits and Systems (ISCAS), pages 1–4, 2017. (Cited on page 44.)

[45] Hala Hamadeh, Soma Chaudhuri, and Akhilesh Tyagi. Area, energy, and time assessment for a distributed tpm for distributed trust in IoT clusters. Integration, 58:267 – 273, 2017. (Cited on page 44.)

[46] C. Lesjak, D. Hein, and J. Winter. Hardware-security technologies for industrial IoT: Trustzone and security controller. In IECON 2015 - 41st AnnualConference of the IEEE Industrial Electronics Society, pages 002589–002595, 2015. (Cited on page 44.)

[47] Ashok Kumar Das, Sherali Zeadally, and Debiao He. Taxonomy and analysis of security protocols for internet of things. Future Generation Computer Systems, 89:110 – 125, 2018. (Cited on page 44.)

[48] S. Agrawal and P. Ahlawat. A survey on the authentication techniques in internet of things. In 2020 IEEE International Students’ Conference on Electrical,Electronics and Computer Science (SCEECS), pages 1–5, 2020. (Cited on pages 44 and 45.)

[49] M. Swan, Blockchain: Blueprint for a New Economy, 1st ed. Beijing, China: O’Reilly, 2015
[50] Pärssinen, Matti et al. “Is Blockchain Ready to Revolutionize Online Advertising?” *IEEE Access* 6 (2018): 54884-54899.

[51] Yaga, D., Mell, P., Roby, N., & Scarfone, K. (2019). Blockchain technology overview. *arXiv preprint arXiv:1906.11078*.

[52]Golosova, Jūlija and A. Romanovs. “The Advantages and Disadvantages of the Blockchain Technology.” *2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE) (2018): 1-6.*

[53] Singhal, B., G. Dhameja, and P.S. Panda, How Blockchain Works, in Beginning Blockchain: A Beginner's Guide to Building Blockchain Solutions, B. Singhal, G. Dhameja, and P.S. Panda, Editors. 2018, Apress: Berkeley, CA. p. 31-148.

[54]BlockchainHub. Blockchains & Distributed Ledger Technologies. Accessed: Jan. 8, 2018. [Online]. Available: https://blockchainhub.net/blockchains-and-distributedledger-technologies-in-general/

[55] I.-C. Lin and T.-C. Liao, ‘‘A survey of blockchain security issues and challenges,’’ IJ Netw. Secur., vol. 19, no. 5, pp. 653–659, 2017.

[56]M. Pilkington, Blockchain Technology: Principles and Applications. Cheltenham, U.K.: HAL, 2016.

[57] Bytecoin. Proof of Stake, Proof of Work Comparison. Accessed: Aug. 22, 2017. [Online]. Available: https://bytecoin.org/blog/proof-of-stake-proof-of-workcomparison

[58] J. Mattila, ‘‘The blockchain phenomenon,’’ Book Blockchain Phenomenon Berkeley Roundtable Int. Econ., Berkeley, CA, USA, ETLA Work. Papers 38, 2016.

[59] P. Tasca, T. Thanabalasingham, and C. J. Tessone, ‘‘Ontology of blockchain technologies. Principles of identification and classification,’’ Cornell Univ., New York, NY, USA, Tech. Rep., 2017.

[60] C. Franko, “Borderless: A Governance Platform and Charity for a Global Society”

[61] DHL Trend Research, “Blockchain in Logistics”, 2018

[62] Followmyvote.com, “Why Online Voting” [online]. Available from: https://followmyvote.com

[63] Republic of Estonia E-Residency, “The new digital nation” [online]. Available from: https://e-resident.gov.ee/

[64] A scribe, “Lock in attribution, securely share and trace where your digital work spreads.” [online]. Available from: https://www.ascribe.io/

[65] Everledger, “Pioneers of digital provenance” [online]. Available from: https://www.everledger.io/about-us/about

[66] MedRec, “What is Medrec?” [online]. Available from: https://medrec.media.mit.edu/

[67] Blockchaintechnology, “Advantages & Disadvantages of Blockchain Technology” [online]. 2016. Available from: https://blockchaintechnologycom.wordpress.com/2016/11/21/advantagesdisadvantages/

[68] W. Fauvel, “Blockchain Advantages and Disadvantages” [online]. August 2017. Available from: https://medium.com/nudjed/blockchain-advantage-anddisadvantages-e76dfde3bbc0

[69] A. Songara, L. Chouhan, “Blockchain: A Decentralized Technique for Securing Internet of Things”. Conference paper, October 2017

[70] K. Biswas and V. Muthukkumarasamy, "Securing Smart Cities Using Blockchain Technology," 2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS), Sydney, NSW, 2016, pp. 1392-1393

[71] Dataflair team, “Advantages and disadvantages of Blockchain Technology” [online]. 2018. Available from: https://data-flair.training/blogs/advantages-anddisadvantages-of-blockchain/

[72] Panarello, Alfonso et al. “Blockchain and IoT Integration: A Systematic Survey.” *Sensors (Basel, Switzerland)* 18 (2018): n. pag.

[73] Hammi, M. T. et al. “Bubbles of Trust: A decentralized blockchain-based authentication system for IoT.” *Comput. Secur.* 78 (2018): 126-142.

[74] Almadhoun, Randa et al. “A User Authentication Scheme of IoT Devices using Blockchain-Enabled Fog Nodes.” *2018 IEEE/ACS 15th International Conference on Computer Systems and Applications (AICCSA)* (2018): 1-8.

[75] Q. Jiang, J. Ma, X. Lu, and Y. Tian, “An efficient two-factor user authentication scheme with unlinkability for wireless sensor networks,” Peer-to-peer Networking and Applications, vol. 8, no. 6, pp. 1070-1081, 2015.

[76] A. K. Das, “A secure and robust temporal credential-based three-factor user authentication scheme for wireless sensor networks,” Peer-to-peer Networking and Applications, vol. 9, no. 1, pp. 223-244, 2016.

[77] M. S. Farash, M. Turkanovic, S. Kumari, and M. Holbl, “An efficient user authentication and key agreement scheme for heterogeneous wire-less sensor network tailored for the internet of things environment,” Ad Hoc Networks, vol. 36, pp. 152-176, 2016.

[78] (03/08/2021) Ethereum docs. Disponible:

https://ethereum.org/en/developers/docs/

[79] (03/08/2021) Solidity v0.8.7. Disponible : https://docs.soliditylang.org/en/v0.8.7/

[80] (04/08/2021) Web3j 4.8.7. Disponible: https://docs.web3j.io/4.8.7/

[81] (04/08/2021) Truffle overview . Disponible: https://www.trufflesuite.com/docs/truffle/overview

[82] (10/08/2021) Ganache overview. Disponible : https://www.trufflesuite.com/docs/ganache/overview

[83] (15/08/2021) Maven what-is-maven. Disponible: https://maven.apache.org/what-is-maven.html

[84] Gupta, P., & Kumar, S. (2014). A comparative analysis of SHA and MD5 algorithm. *architecture*, *1*, 5.

[85] Visconti, A., Mosnáček, O., Brož, M., & Matyáš, V. (2019). Examining PBKDF2 security margin—Case study of LUKS. *Journal of Information Security and Applications*, *46*, 296-306.