الجمهورية الجزائرية الديمقراطية الشعبية

وزارة التعليم العالي والبحث العلمي

جامعة سعيدة د. مولاي الطاهر

كلية التكنولوجيا

قسم: الإعلام الآلي

Mémoire de Master

Spécialité : Réseaux Informatiques et Systèmes Réparties

Thème

Système d'authentification dans l'internet des objets : Étude et sécurisation

Présenté par :

BELAIDI Kheira

TABINA Sabrina

Dirigé par :

Mme TALEB Fadia.

Promotion 2021 - 2022

Dédicaces

A mes chers parents,

Que nulle dédicace ne puisse exprimer ce que je leurs dois, pour leur bienveillance, leur affection et leur soutien...Trésors de bonté, de générosité et de tendresse, en témoignage de mon profond amour «Que Dieu vous garde».

A ma chère sœur Fatma Zahra et mes frères Ahmed et Belaid,

En témoignage de mes sincères reconnaissances pour les efforts qu'ils ont consenti pour l'accomplissement de mes études. Je leur dédie ce modeste travail en témoignage de mon grand amour et ma gratitude infinie.

A tous mes amis,

Pour leur aide et leur soutien moral durant l'élaboration du travail de fin d'études.

A toute ma Famille,

Pour leur soutien continu, leur grande confiance en mes capacités, leur amour et leur grande Appréciation pour moi.

A tous ceux dont l'oubli du nom n'est guère celui du cœur...

« Il n'y a pas de « si » ni de « mais », il faut réussir ».

KHEIRA

Dédicaces

Je dédie ce modeste travail

A mes chers parents

TABINA ben moussa, MELOUKI Habiba et à ma grand-mère et grand père pour leurs sacrifices et leurs encouragements que dieu les protègent

A celle avec qui j'ai partagé de bons moments

A ma sœur Imen et mes frère Amer, Younes,

A ceux qui m'ont encouragé et aidé et qui seront fière de moi

Ma grande famille que j'aime beaucoup,

A ceux que j'aime et que je respecte : mes vraies amies

A mon binôme et sa famille

SABRINA

Remerciements

C'est avec un grand plaisir que je réserve ces quelques lignes en Signe de gratitude et de profonde reconnaissance à tous ceux qui, de près ou de loin ont contribué à la réalisation et l'aboutissement de ce travail.

Je tiens tout d'abord à remercier Mme ***TALEB Fadia*** *pour son soutien, sa supervision, ses efforts, sa gentillesse et surtout pour son aide précieuse tout au long de l'élaboration de ce travail.*

Je remercie vivement les honorables membres du jury Monsieur ***BENYAHIA Kada*** *et Monsieur* ***CHAIBI Hassen*** *qui ont accepté d'évaluer mon travail.*

Je m'acquitte, enfin, volontiers d'un devoir de gratitude et de remerciements à tous mes enseignants pour la qualité de l'enseignement qu'ils ont bien volume prodiguer durant mes études afin de me fournir une formation efficiente.

Résumé

L'Internet des objets (IoT) est un paradigme prometteur qui étale la connexion Internet de nos jours pour interconnecter différents types d'objets intelligents, autre que les ordinateurs et les téléphones mobiles, pour un mode de vie beaucoup plus sophistiqué et une qualité de service améliorée dans différents domaines d'application tel que les villes intelligentes et les appareils intelligents. Et exploite d'autres technologies telles que RFID et WSN. En effet, la maturité de l'Internet des objets dépend sans aucun doute de la sécurité des communications et la protection de la vie privée des utilisateurs. Toutefois, les hétérogénéités technologiques et matérielles, ainsi que la nature asymétrique des communications entre les objets, font de la sécurité de l'IoT, un problème crucial. Dans ce contexte, de nombreuses solutions ont été proposées pour la standardisation de la sécurité d'IoT. Parmi ces solutions est l'utilisation d'un protocole efficace et sécurisé.

Ce mémoire se penche sur l'une de ces solutions, qui est les protocoles d'authentification. Nous analysons la sécurité de plusieurs protocoles d'authentification RFID récentes. Notre travail consiste aussi à établir une comparaison entre les différents protocoles existants basés sur les codes des correcteurs d'erreurs en termes de sécurité et de performance on utilisant l'outil AVISPA.

Mots-Clés : Internet des objets ; Protocole d'authentification; RFID ; AVISPA.

ملخص

إنترنت الأشياء(IoT) هو نموذج واعد ينشر اتصال الإنترنت في الوقت الحاضر لربط أنواع مختلفة من الكائنات الذكية، بخلاف أجهزة الكمبيوتر والهواتف المحمولة، من أجل نمط حياة أكثر تطوراً وجودة خدمة محسنة في مجالات التطبيق المختلفة مثل المدن الذكية والأجهزة الذكية. وتستفيد من التقنيات الأخرى مثل تحديد الترددات الراديوية(RFID) وشبكات الاستشعار اللاسلكية(WSN). في الواقع، إن نضج إنترنت الأشياء يعتمد بلا شك على أمن الاتصالات وحماية خصوصية المستخدم. ومع ذلك، فإن عدم التجانس التكنولوجي والمادي، فضلاً عن الطبيعة غير المتكافئة للاتصالات بين الأشياء، تجعل أمن إنترنت الأشياء مشكلة حاسمة. في هذا السياق، تم اقتراح العديد من الحلول لتوحيد أمان إنترنت الأشياء من بين هذه الحلول استخدام بروتوكول فعال وآمن .

تركز هذه الرسالة على أحد هذه الحلول وهو بروتوكولات المصادقة. نقوم بتحليل أمان العديد من بروتوكولات مصادقة في انظمة تحديد الترددات الراديوية الحديثة. يتمثل عملنا أيضًا في إنشاء مقارنة بين مختلف البروتوكولات الحالية بناءً على أكواد تصحيح الأخطاء من حيث الأمان والأداء باستخدام أداة أفيسبا (AVISPA).

كلمات البحث: انترنت الأشياء ؛ بروتوكول المصادقة ؛ *تحديد الترددات الراديوية؛ أفيسبا*.

Abstract

The Internet of Things (IoT) is a promising paradigm that spreads the Internet connection nowadays to interconnect different types of smart objects, other than computers and mobile phones, for a much more sophisticated lifestyle and quality of improved service in different application areas such as smart cities and smart devices. And leverages other technologies such as RFID and WSN. Indeed, the maturity of the Internet of Things undoubtedly depends on the security of communications and the protection of user privacy. However, technological and material heterogeneities, as well as the asymmetrical nature of communications between objects, make IoT security a critical issue. In this context, many solutions have been proposed for the standardization of IoT security. Among these solutions is the use of an efficient and secure protocol.

This thesis focuses on one of these solutions, which are authentication protocols. We analyze the security of several recent RFID authentication protocols. Our work also consists in establishing a comparison between the various existing protocols based on error correcting codes in terms of security and performance using the AVISPA tool.

Keywords: Internet of things; Authentication protocol; RFID; AVISPA.

Table des matières

Chapitre 02 : La sécurité dans l'Internet des Objets

Chapitre 03 : Les protocoles d'authentification dans l'IdO

Liste des figures

Liste des tableaux

Liste des abréviations

IoT	**Internet of Things**
IdO	**Internet des Objets**
RFID	**Radio Frequency Identification**
M2M	**Machine to Machine**
M2C	**Machine to Cloud**
SDN	**Software Defined Networking**
GPS	**Global Positioning Satellite**
Wi-Fi	**Wireless Fidelity**
MQTT	**Message Queuing Telemetry Transport**
CoAP	**Constrained Application Protocole**
Dos	**Denial of Service**
WSN	**Wireless Sensors Network**
UDP	**User Datagram Protocol**
TCP	**Transmission Control Protocol**
AES	**Advanced Encryption Standard**
IETF	**Internet Engineering Task Force**
AVISPA	**Automated Verification Internet Protocol and its Applications**

Introduction générale

À l'heure actuelle, l'Internet des objets (IdO) constitue l'un des développements majeurs dans le domaine des communications sans fil. Ce nouveau modèle permet l'interaction entre des organismes hétérogènes tels que : des capteurs, des marqueurs d'identification par radiofréquence (RFID), des téléphones mobiles, etc et cela afin d'atteindre des objectifs communs.

Dans l'Internet des objets, chaque objet est traduisible, guidé et lisible en ligne. Ces objets peuvent envoyer et recevoir des données de manière indépendante, ouvrant ainsi de nouveaux horizons pour de nombreux domaines d'applications intelligentes : villes intelligentes, industrie, santé, etc. La grande puissance de l'internet des objets tient du fait que ses objets communiquent, analysent, traitent et gèrent des données de manière indépendante. Cependant, il existe de nombreux problèmes liés à la sécurité de cette haute technologie, tels que : l'usurpation d'identité, le vol d'informations et la modification de données, etc. Ces derniers représentent un réel danger pour ce type de système et peuvent entrainer régulièrement des incidents. En effet, leur principe consiste à exploiter les différentes failles existantes sur le système afin d'atteindre les objectifs désirés. Le besoin en sécurité se fait de plus en plus présent et beaucoup de solutions ont émergé. La sécurité dans les IdO est un domaine de recherche extrêmement vaste, il existe beaucoup d'applications différentes en raison de la complexité des IdO et des différentes strates que nous devons sécuriser.

La prospérité de l'Internet des objets ne peut être atteinte que lorsqu'une bonne sécurité est garantie. Il est nécessaire de créer un fichier politique de sécurité qui empêche tout objet nuisible ou non autorisé d'accéder au système IoT dans le but de lire ou de modifier les données. Exécuter un service ou rejoindre un réseau doit être impérativement précédé par deux actions importantes : prouver son identité et disposer des droits d'accès nécessaires.

Notre projet de fin d'étude a pour objectif d'aborder aux systèmes d'authentification dans l'Internet des Objets et d'étudier les protocoles d'authentification pour faire une comparaison. En fait, dans notre recherche, nous avons abordé deux aspects : la sécurité en générale et l'authentification dans IdO en particulier.

Ce mémoire est organisé en trois chapitres : le premier chapitre est une survole des IdO. A savoir, leurs architectures, leurs caractéristiques, leurs défis et leurs technologies de communication, ainsi que les différents modes de communication existants. Tandis que le deuxième chapitre présente les propriétés de sécurité, leurs mécanismes et les différents types d'authentification. Dans le troisième chapitre, nous décrivons quelques travaux qui proposent de nouveaux protocoles d'authentification applicables aux IoT. Nous présentons l'outil AVISPA ainsi que quelques résultats de validation formelle des protocoles décrits auparavant. Une comparaison nous a ensuite permis d'évaluer et de définir le meilleur protocole d'authentification. Nous avons aussi proposé une solution pour améliorer l'un de ces derniers protocoles.

Chapitre 1 :
Internet des objets

1.1 Introduction

Nous vivons à une époque où le comportement numérique évolue vers des médias de plus en plus mobiles. Les objets connectés sont devenus incontournables dans notre quotidien et chaque jour nous découvrons de nouvelles choses. Ces nouvelles expériences nous offrent une excellente occasion de transformer et d'améliorer notre expérience utilisateur. Ainsi, les objets connectés sont de plus en plus présents dans nos usages. Ils offrent de nouvelles opportunités dans l'utilisation des technologies numériques et de communication.

Dans ce premier chapitre, nous présentons les principaux concepts de l'Internet des Objets (IdO) « en anglais Internet of Things (IOT) », tout d'abord nous donnons quelques définitions de base sur les IoT et leur historique ; ensuite nous parlons de l'architecture en trois couches de l'internet des Objets !!!(Perception, réseau, applicative), puis nous énonçons les caractéristiques générale des IoT comme la connectivité, la sécurité, l'intelligence et l'identité. Nous expliquons par la suite les défis majeur des IoT (l'hétérogénéité, les ressources limités, la sécurité, etc) et les technologies de communication entre la couche physique, la couche liaison de données, la couche réseau, la couche transport et enfin la couche application. L'internet des Objets se distingue par deux technologies clés : les RFID et les réseaux de capteurs. Enfin, nous parlons sur les domaines d'application (les villes intelligente, le smart Grid et les appareils intelligents).

1.2 Définition de l'internet des objets et leur historique :

1.2.1 Définition de l'IoT

Selon l'Union internationale des télécommunications, l'Internet des objets (IdO) est "une infrastructure mondiale pour la société de l'information, qui permet la fourniture de services avancés en interconnectant des objets (physiques ou virtuels) à l'aide de technologies d'information et de communication interopérables existantes ou en évolution". En fait, la définition de ce qu'est l'Internet des objets n'est pas statique. Elle croise des dimensions conceptuelles et techniques.

L'Internet des objets tire pleinement parti des objets pour fournir des services pour toutes sortes d'applications, tout en garantissant le respect des exigences de sécurité et de confidentialité. Elle souligne enfin que, dans une perspective plus large, l'Internet

des objets peut être considéré comme un concept ayant des implications pour les technologies et la société.

Par conséquent, l'Internet des objets est « un réseau de réseaux qui permet, via des systèmes d'identification électronique unifiés et des appareils mobiles sans fil, d'identifier directement et sans ambiguïté des entités numériques et des objets physiques et ainsi de pouvoir récupérer, stocker, transmettre et manipuler, sans interruption entre le monde physique et virtuel, les données liées »**[1].**

1.2.2 Historique de l'IoT

L'émergence de l'Internet des objets ce n'est qu'un résultat de convergence entre multiples technologies, à savoir l'Internet, la communication sans fil, les systèmes embarqués, systèmes microélectroniques et la nanotechnologie [2]. Dans cette section, nous citons les évènements les plus marquants sur le chemin de la concrétisation de l'IoT.

En 1999, la désignation Internet des objets a été prononcée pour la toute première fois par Kevin Ashton. Après, en 2000 la société LG annonce son premier réfrigérateur intelligent connecté à Internet. De plus, la technologie RFID (Radio Frequency IDentification) qui est l'une des technologies constitutionnelles de l'IoT, a commencé à être massivement déployée vers les années 2003 et 2004. D'autre part, une initiative très intéressante a été prise en 2008 ; un groupe de recherche appelé IPSo Alliance s'est consacré à promouvoir l'utilisation du protocole IP (Internet Protocol) pour les réseaux d'objets miniatures intelligents [2].

De nombreux travaux de recherches ont été succédés et se sont tous concentrés autour de la réalisation, dans les meilleures conditions, de la vision de l'Internet des objets et la mener à sa maturité en dépit de tous les défis soulevés. Cela avec la considération des progrès technologiques continus dans le marché des dispositifs intelligents et dans le domaine de technologies de télécommunication (comme : le cloud computing, le concept du SDN (Software-Defined Networking) [3], etc.).

1.3 Architecture de l'Internet des objets :

L'architecture des IoT définit les éléments physiques, la disposition technique, la configuration du réseau, les procédures d'exploitation et les formats de données utilisés. Jusque-là aucun standard d'architecture des IoT n'a été imposé, en effet, différentes architectures ont été proposées, comme par exemple :

- Architecture à trois couches [4].
- Architecture à quatre couches [5], [8].
- Architecture orientée service [6,7].

- Architecture à cinq couches [5, 9].

Dans le cadre de ce travail, nous avons choisi de mettre en relief l'architecture à trois couches. Elle est composée d'une couche de perception, une couche réseau et une couche application [4,5].

1.4 Architecture IoT trois couches

1.4.1 Couche de perception :

Cette couche de perception est la couche physique de l'architecture IoT. Elle comprend tout ce qui est capteur, actionneur, système embarqué. Elle permet donc de collecter une grande quantité de données suivant l'application et le besoin. Elle assure aussi la communication avec l'environnement et permet de détecter d'autres objets intelligents présents dans ce dernier.

1.4.2 Couche réseau :

Les données obtenues par les dispositifs intelligents doivent être transmises, distribuées, stockées et analysées. C'est la couche réseau qui en est responsable. Elle doit aussi assurer la connexion des objets intelligents entre eux et assurer leur liaison aux périphériques réseau et aux serveurs **[10]**.

1.4.3 Couche applicative :

L'utilisateur communique avec cette couche applicative. Il est chargé de fournir au client les ressources logicielles nécessaires. Par exemple, dans une application domotique qui permet d'automatiser et de contrôler à distance les appareils fonctionnant dans une maison par un simple appuie sur un bouton de l'application, où les utilisateurs appuient sur un bouton de l'application pour allumer une machine à café, par exemple. La couche application est chargée de fournir au client des ressources spécifiques à l'application. Il spécifie différentes utilisations de l'IoT, telles que les maisons intelligentes, les villes intelligentes et la santé intelligente.

1.5 Caractéristiques générale de l'IOT

Les caractéristiques générales de l'IoT sont les suivantes:

1.5.1 Connectivité

Est une exigence importante de l'infrastructure IoT. Les objets de l'IoT doivent être connectés à l'infrastructure IoT. N'importe qui, n'importe où, n'importe quand, la connectivité doit être garantie à tout moment, sans connexion, rien n'a de sens **[11].**

1.5.2 La sécurité

Les applications IOT couvrent également le domaine de la sécurité. Ils offrent la possibilité de fournir une protection supplémentaire dans l'environnement de travail, tant pour les opérateurs que pour l'équipement.
D'autres détecteurs supervisent les niveaux de chaleur, de pressage ou de vibration, et en cas de divergence des valeurs de référence, envoient des alertes aux stations concernées, notamment sur les Smartphones des employés responsables afin qu'ils puissent réagir en conséquence. Dans les machines plus avancées, les machines réagissent d'elles-mêmes en s'arrêtant ou en diminuant le régime.

1.5.3 Intelligence et identité

C'est un domaine d'activité qui consiste à attribuer des identifiants uniques (UID) ainsi que des métadonnées à des appareils et des objets, leur permettant de communiquer et de se connecter.

L'extraction de connaissances à partir des données générées est très importante. Par exemple, un capteur saisi des données, mais ces données ne seront utiles que si elles sont interprétées correctement. Chaque appareil IoT a une identité unique. Cette identification est utile pour suivre l'équipement et parfois pour interroger son état **[11].**

1.6 Les défis de l'IoT

1.6.1 L'hétérogénéité des dispositifs :

L'un des principaux défis auxquels est confronté l'Internet des Objets, est la capacité à gérer l'hétérogénéité des dispositifs en termes de ressources, de normes et de standards de communication.

En effet, les objets connectés sont destinés à différentes applications, une fois déployés, ils vont récolter des données grâce à des capteurs et c'est à l'application embarquée de les traiter. Le traitement des données varie en complexité suivant l'application, il peut être simple « vérifier si une température est bien comprise dans un intervalle » ou complexe « détection d'une intrusion grâce à une vidéo ». Cette complexité permet de définir des contraintes matérielles et logicielles spécifiques. Les contraintes matérielles se résument à : puissance de calcul, portée de transmission, capacité de stockage et autonomie en terme d'énergie.

L’hétérogénéité des objets connectés peut aussi concerner l’aspect sécurité puisque l’exigence en sécurité peut différer d’un objet connecté à l’autre.

La gestion de l’hétérogénéité est alors primordiale afin d’assurer l’interopérabilité des différents objets connectés et garantir un bon fonctionnement de l’IoT **[10].**

1.6.2 Les ressources limitées :

Un autre problème courant avec les appareils IOT est qu'ils sont souvent limités en termes de ressources (puissance de traitement, capacité de stockage, autonomie énergétique, portée), ceci afin de conserver des objets compacts ayant un coût minimal. Cette limitation en ressource peut être nuisible et empêcher la mise en œuvre d’une sécurité renforcée par exemple. Pour cette raison, de nombreux appareils n'offrent pas ou ne peuvent pas offrir de fonctions de sécurité avancées. Par exemple, les capteurs qui surveillent l'humidité ou la température ne peuvent pas gérer le cryptage avancé ou d'autres mesures de sécurité ; Comme les autres appareils IoT " set and forget " - placés sur le terrain ou sur un appareil et laissés en fin de vie - ils reçoivent rarement les mises à jour de sécurité ou les correctifs qu'ils devraient.

Les progrès dans le monde de la technologie et de la miniaturisation font qu’aujourd’hui, de plus en plus d’objets plus puissants, aussi compactes et peu chers sont disponibles. Toutefois, malgré cette évolution, les ressources limitées dans les objets connectés restent une contrainte majeure.

Prenons à titre d’exemple la contrainte de l’autonomie en énergie dans les dispositifs intelligents. L’approvisionnement en énergie se fait soit par batterie, pile ou par des panneaux solaires, etc. La mise en place d’une sécurité doit se faire en prenant en compte ces ressources limitées, il faut donc être attentif à la consommation énergétique liée aux calculs cryptographiques mais aussi aux communications. Le contraire mènerait à un échec inévitable **[12].**

1.6.3 La mobilité :

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

La gestion de tels réseaux dynamiques devient très difficile dans de grands environnements distribués car déterminer si un dispositif peut être accepté dans le

réseau revient à déterminer le niveau de confiance qu'on lui accorde. C'est donc, une problématique de sécurité.

Dans les réseaux dynamiques, la gestion de la mobilité peut se faire de plusieurs façons. Le dispositif intelligent (dont on rappelle qu'il est intégré à un dispositif physique, c'est-à-dire un objet) doit détecter son mouvement afin d'identifier qu'il va probablement quitter l'emplacement qu'il occupe alors dans la topologie du réseau pour se connecter à un autre endroit ou même à un autre réseau. Cette détection peut se faire par un scan passif des messages des participants aux réseaux ou via le suivi des émissions de balises (« beacons »). Une autre façon de gérer la mobilité est l'intégration dans les protocoles de messages de signalisation et de contrôle de la localisation des nœuds dans le réseau **[13].**

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

1.6.4 La sécurité :

Un certain nombre de défis empêchent de sécuriser les appareils IOT et de fournir une sécurité de bout en bout dans un environnement IOT. Étant donné que l'idée de mettre en réseau des appareils et d'autres objets est relativement nouvelle, la sécurité n'a pas toujours été considérée comme une priorité absolue lors de la phase de conception du produit. De plus, comme l'Internet des objets est un marché émergent, de nombreux concepteurs et fabricants de produits sont plus intéressés à commercialiser rapidement leurs produits qu'à prendre les mesures nécessaires pour renforcer la sécurité dès le départ**[14].**

1.7 Les deux modes de communication dans l'IdO :

La communication entre objets connectés se fait suivant deux modes : le mode Machine à Machine M2M (« Machine-to-Machine ») et le mode Machine à Cloud M2C(« Machine-to-Cloud »).

1.7.1 Machine à Machine M2M :

Dans un mode de communication Machine à Machine M2M, les objets intelligents peuvent échanger et partager des données de manière décentralisée, sans passer par un système centralisé. Si nous prenons le cas des maisons intelligentes, un réseau domestique (HAN) se compose de divers objets intelligents tels que des lumières intelligentes, des horloges intelligentes et des capteurs de température. Ces objets peuvent communiquer directement sans intervention humaine de diverses manières. Par exemple, un capteur de lumière peut détecter l'absence de lumière solaire et

envoyer un message pour allumer une lumière intelligente sans aucune intervention humaine **[13].**

1.7.2 Machine à Cloud M2C :

À l'inverse de mode M2M, le mode de communication Machine à Cloud M2C fournit une communication entre les objets intelligents et le cloud. Par exemple, un thermostat intelligent peut se connecter au service de prévision météorologique disponible sur le cloud pour déterminer s'il doit modifier la température (en envoyant des requêtes M2M aux appareils concernés tels que les radiateurs ou les climatiseurs) à la demande de l'utilisateur **[13].**

1.8 Les technologies de communication de l'Internet des Objets :

Avec la montée en nombre des objets connectés et le désir des solutions sans-fils, les protocoles et technologies *Internet of Things* (IoT) deviennent de plus en plus populaires. Il existe de nombreux protocoles pour l'IoT classifiés selon les différents cas d'usage et/ou selon leur portée sans-fil. De nombreux protocoles existent actuellement, tel que X-10, ZigBee, Bluetooth, LoraWan, Thread, Sigfox ouWi-Fi. Chaque technologie a ses avantages, ses inconvénients ainsi qu'un domaine de prédilection. Certaines se focalisent sur le développement de maisons connectées et d'autres se focalisent sur des infrastructures de plus grande ampleurs (usines ou même des villes).

Malgré les différences que ces technologies présentent, elles ont un but commun : la création de réseau de taille importante avec une faible consommation énergétique.

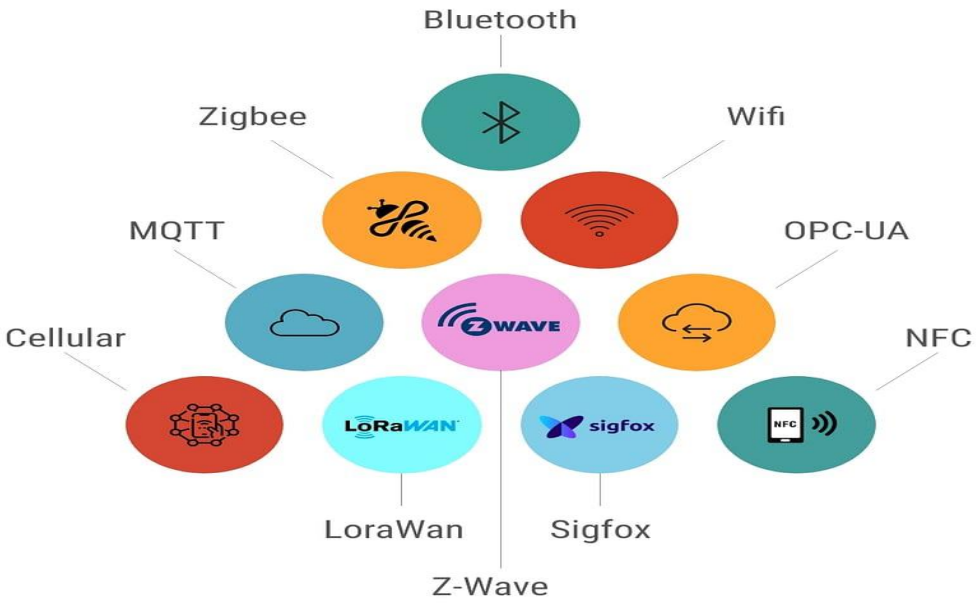

Figure 1:top 10 des technologies de l'Internet des Objets.

1.8.1 Les technologies des couches physiques et liaison de données

- **X-10:**

Le protocole filaire X-10 utilise la ligne d'alimentation électrique pour connecter et contrôler différents dispositifs intelligents d'une maison. En effet, il fonctionne sur le principe des CPL (Courants Porteurs en Ligne) pour communiquer entre l'émetteur et le récepteur en envoyant et en recevant des signaux sur les fils d'alimentation électrique **[15]**. X-10 offre de nombreux avantages: il est simple et facile à implémenter et à mettre en œuvre, son installation ne nécessite pas de nouveaux câblages et il permet de relier jusqu'à 256 dispositifs entre eux. Il ne peut envoyer qu'une seule commande à la fois afin d'éviter les collisions. En terme de sécurité du protocole, elle est absente puisque toute personne ayant accès physiquement à la ligne électrique peut envoyer des commandes. Toutefois, il est possible de se protéger des attaques extérieures en installant un module de filtrage qui empêchera les signaux émis en amont de l'installation de rentrer dans la maison mais aussi d'éviter que des informations sortent de la maison.

- **Bluetooth:**

Le Bluetooth est un protocole de communication sans fil, visant à connecter des appareils mobiles entre eux. Il permet une connexion entre plusieurs périphériques et l'échange bidirectionnel de données et de fichiers sur une très courte distance. Le principal avantage du Bluetooth réside dans le fait de pouvoir réaliser une connexion entre deux appareils sans aucune liaison filaire en utilisant des ondes radio sur la bande de fréquences de 2,4 GHz. Très vite, le Bluetooth a investi le monde de la téléphonie mobile et de l'informatique.

En 2010, la version 4.0 du Bluetooth introduit un mode de fonctionnement Low Energy (BLE) pour les objets connectés. Contrairement au Bluetooth, le BLE a pour but de fournir les mêmes fonctionnalités que le Bluetooth classique avec un coût et une consommation d'énergie réduits. Dans la version 4.1, BLE a introduit l'utilisation de l'algorithme de chiffrement AES (« Advanced Encryption Standard ») avec compteur (CTR, « CounTeR mode ») en mode CBCMAC (AES-CCM) pour la première fois dans une spécification Bluetooth, afin de fournir à la fois l'authentification et la confidentialité des données transmises. La dernière version de ce standard est Bluetooth 5.0 [16] qui permet de doubler la portée de communication, de multiplier le débit des transmissions basse consommation de 1 à 2 Mbps, d'améliorer l'interopérabilité et de réduire les interférences avec les autres technologies sans fil comme le Wi-Fi.

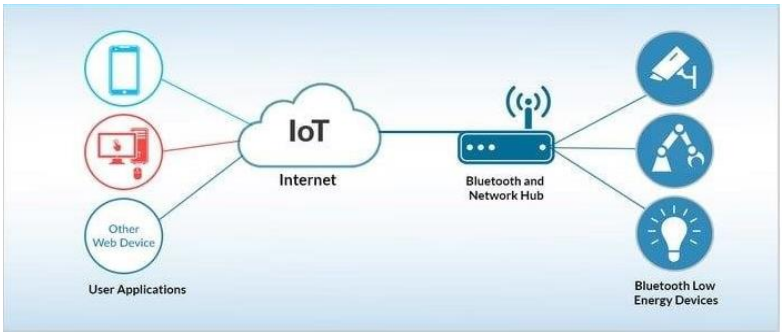

Figure 2 : Le Bluetooth dans les IoT.

- **ZigBee:**

ZigBee est un protocole de haut niveau permettant la communication d'équipements personnels ou domestiques équipés de petits émetteurs radios à faible consommation; il est basé sur la norme IEEE802.15.4 pour les réseaux à dimension personnelle. Cette technologie a pour but de permettre une communication à courtes distances **[17]**.

Parce qu'il utilise un réseau maillé, le protocole ZigBee offre une plus grande portée et une couverture plus stable que les autres réseaux .Dans un réseau maillé, chaque périphérique est connecté à un autre sans hiérarchie. C'est cette absence de hiérarchie qui permet d'étendre la portée de vos appareils connectés. Vous n'avez donc pas besoin d'ajouter de répéteurs. Aussi il consomme beaucoup moins d'énergie et il offre plus grande sécurité des données **[18]**.

Figure 3 : ZigBee dans l'IoT.

- **SigFox :**

Sigfox est une technologie de communication cellulaire, il est spécialisé dans l'IoT grâce à un réseau bas débit dit "0G". Il contribue à l'Internet des objets en permettant l'interconnexion via une passerelle. Sa technologie radio UNB (« Ultra narrow band ») lui permet de bâtir un réseau cellulaire bas-débit, économe en énergie. Ce réseau UNB, combiné à une diversité fréquentielle, temporelle et spatiale, permet une longue portée et une haute qualité de service. **[19]**

- **LoRaWan:**

LoRaWAN est un protocole de télécommunication radio permettant la communication à bas débit d'objets connectés. Il émet en France sur la bande de fréquence 868 MHz. Le signal radio est émis sur une grande largeur spectrale, pour limiter au maximum le risque d'interférence avec des signaux parasites. Il se base sur l'utilisation de la technique de modulation LoRa, cette dernière est utilisée pour les communications spatiales et militaires. Ce protocole de communication permet d'envoyer des données en intérieur (indoor), en sous-sol (deep indoor) et en extérieur (outdoor) **[20]**.C'est la technologie de communication la plus aboutie en France, en effet, Bouygues et Orange utilisent LoRa pour connecter les objets.

- **Wi-Fi:**

Le Wi-Fi, ou Wifi, est un réseau local lancé en 1999 qui utilise des ondes radioélectriques pour relier entre eux, plusieurs appareils informatiques dans le but de faciliter la transmission de données. Le terme est une abréviation de Wireless Fidelity qui peut être traduite en français par "fidélité sans fil". Régi par les normes IEEE 802.11, le Wifi est principalement utilisé pour relier des appareils (ordinateurs portables, PDA, etc.) à des liaisons haut débit. On le retrouve en particulier dans le domaine d'Internet avec des appareils nomades connectés au réseau Wifi plutôt que par un câble Ethernet. Le Wifi permet un très important débit data (environ **400 Mb/s** pour les modems grand public récents), de manière fiable et sécurisée **[21]**.

Aujourd'hui nous sommes à la version 6 le **Wifi 6**, aussi appelé Wifi AX, cette nouvelle génération en cours de commercialisation, va permettre d'améliorer considérablement la connectivité des objets connectés **[22]**.

- **Comparaison entre différentes technologies de communication sans fil utilisées dans l'IoT :**

Dans ce tableau, nous avons comparé cinq protocoles ou technologies de communication sans fils utilisées dans l'IdO et cela suivant différents critères : débits, sécurité, fréquences, portée et topologie.

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

Tableau 1 : Comparaison entre différentes technologies de communication sans fil utilisées dans l'IoT[13].

1.8.2 Les technologies de la couche réseau

- **6LoWPAN :**

6LoWPAN est le nom d'une norme Internet Engineering Task Force (IETF) qui définit une approche pour le routage du protocole Internet version 6 (IPv6) sur des réseaux sans fil à faible consommation. 6LoWPAN vise à apporter les avantages de la mise en réseau IP standard aux réseaux maillés et de capteurs à faible consommation, qui, dans le passé, utilisaient souvent des technologies propriétaires. 6LoWPAN utilise une représentation compacte d'IPv6 et peut être implémenté dans une pile de protocoles à faible encombrement appropriée pour les appareils dont la puissance de traitement et la mémoire sont limitées **[23].**les réseaux LoWPAN sont constitués d'une multitude de nœuds. Ils sont organisés en topologie de type « mesh » ou en étoile. Un protocole de routage permettant de supporter de tels réseaux doit être mis en place. Celui-ci doit en plus répondre aux contraintes des nœuds eux-mêmes (faibles CPU et mémoire) ainsi qu'à celles du 802.15.4 (faible débit et petits paquets). De par leur taille, les équipements 802.15.4 sont facilement transportables. La mobilité doit donc être prise en compte. Afin de contrôler l'intégrité des données transmises sur un réseau 6LoWPAN, la sécurisation du transfert des données devrait être implémentée au niveau IP, en plus de celle offerte par IEEE 802.15.4 (via AES).

1.8.3 Les technologies de la couche transport

Pour la couche transport, nous nous intéresserons seulement à deux protocoles de communication qui sont : MQTT et CoAP.

➢ MQTT :

MQTT Message Queuing Telemetry Transport est un protocole de messagerie de publication/abonnement conçu pour les communications M2M (**section 1.7**) légères. Il a été initialement développé par IBM et est maintenant un standard ouvert. Il consomme peu d'énergie et ne nécessite que peu de ressources processeur et mémoire. Ces caractéristiques le rendent idéal pour une utilisation dans des environnements contraints. MQTT a un modèle client/serveur, où chaque capteur est un client et se connecte à un serveur, appelé courtier, via TCP.MQTT est orienté message. Chaque message est un bloc de données discret, opaque pour le courtier. Chaque message est publié à une adresse, appelée sujet. Les clients peuvent s'abonner à plusieurs sujets. Chaque client abonné à un sujet reçoit chaque message publié dans le sujet **[24].**Dans le terme de sécurité Les courtiers MQTT peuvent exiger une authentification par nom d'utilisateur et mot de passe des clients pour se connecter. Pour garantir la confidentialité, la connexion TCP peut être cryptée avec SSL/TLS.

➢ CoAP :

Constrained Application Protocol est le protocole d'application contrainte du groupe IETF CoRE (Constrained Resource Environments). Comme HTTP, CoAP est un protocole de transfert de documents. Contrairement à HTTP, CoAP est conçu pour les besoins des appareils contraints. CoAP fonctionne sur UDP, pas sur TCP. Les clients et les serveurs communiquent via des datagrammes sans connexion. Il permet d'utiliser la diffusion UDP et la multidiffusion pour l'adressage. Il suit un modèle client/serveur. Les clients font des requêtes aux serveurs, les serveurs renvoient des réponses. Les clients peuvent utiliser les ressources GET, PUT, POST et DELETE. CoAP est conçu pour interagir avec HTTP et le Web RESTful au sens large via de simples proxys. Le protocole CoAP utilise DTLS (« Datagram Transport Layer Security ») pour offrir les mêmes garanties que TLS pour le protocole TCP. Il est à noter que DTLS adresse aussi les problèmes de perte et de réordonnancement des paquets **[24].**

➢ **Comparaison entre MQTT et CoAP :**

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

Tableau 2 : Comparaison entre MQTT et CoAP [24].

1.8.4 Les technologies de la couche application

Pour la couche application nous avons choisi de présenter un protocole de réseau mesh sans fils appelé Thread. Ce dernier a pour objectif de connecter en toute sécurité et de manière fiable les dispositifs intelligents d'une maison. Thread peut gérer plus de 250 nœuds appartenant à un même réseau. Il se caractérise par une faible latence des communications (moins de 100 millisecondes). Thread peut supporter plusieurs couches applicatives qui s'exécutent sur IPv6 comme CoAP et MQTT **[25].**

1.9 Deux technologies clés de l'IdO

Les systèmes WSN et RFID sont deux technologies majeures de l'Internet des Objets. Leur intégration aux objets intelligents offre de nouvelles capacités de communication. Ces dernières années, ces deux technologies ont été de plus en plus utilisées. Nous les présentons en détail dans cette section.

1.9.1 La technologie RFID

La RFID Radio Frequency Identification n'a besoin d'aucune source énergétique pour fonctionner. C'est là son grand atout. La RFID sert à identifier des biens, des machines, des personnes, des animaux… On utilise le protocole pour de la gestion de stocks et la traçabilité des produits. On compte aujourd'hui plus de 20 milliards de tags (badges) RFID dans le monde **[26]**.

Le système RFID est une technologie qui permet de mémoriser et de récupérer des informations à distance grâce à une étiquette qui émet des ondes radio **[27].**Il fonctionne de la manière suivante :

- L'étiquette RFID (ou transpondeur ou tag) est elle-même équipée d'une puce reliée à une antenne, l'antenne permet à la puce de transmettre les informations (numéro de série, poids...) qui peuvent être lues grâce à un lecteur émetteur-récepteur **[20]**.
- Une fois les informations transmises au lecteur RFID équipée d'une antenne intégrée ou externe, celui-ci n'a plus qu'à convertir les ondes-radios en données et celles-ci pourront être lues par un logiciel RFID **[27].**

1.9.2 Les réseaux de capteurs :

Un réseau de capteurs sans fil (WSN) est un réseau ad hoc avec un grand nombre de *nœuds* qui sont des micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils peuvent être aléatoirement dispersés dans une zone géographique, appelée *« champ de captage »* correspondant au terrain d'intérêt pour le phénomène capté **[28].**

1.10 Domaines d'applications :

Plusieurs domaines d'application sont touchés par l'IdO, Parmi ces principaux domaines nous citons:

1.10.1 Les villes intelligentes

Dans les villes intelligentes, les maisons, les routes, les bâtiments, les véhicules, les magasins, les parkings seront connectés à internet et annonceront leur présence à d'autres objets connectés. Cela pourrait aider à maîtriser le trafic routier, faire gagner du temps aux citoyens (notamment aux automobilistes), fournir des informations pertinentes, en temps réel, sur l'emplacement d'un utilisateur, d'une maison intelligente, d'un parking, d'un hôtel, d'un restaurant ou mieux encore de l'hôpital le plus proche. Il est possible aussi dans une ville intelligente d'avoir des informations générales sur la ville comme : la température, l'humidité, les niveaux de rayonnement…etc. Ce concept pourrait aider les autorités à gérer certaines tâches de manière plus facile comme : la dépollution, l'éclairage urbain, etc. Notons que la coexistence massive de multiples technologies est nécessaire pour mise en œuvre du projet de ville intelligente, notamment des réseaux de capteurs **[29].**

Pour la création d'une ville intelligente, il est indispensable de faire appel aux IoT et plus particulièrement aux réseaux de capteurs.

L'application nécessite une planification minutieuse à chaque étape, soutenue par l'accord du gouvernement et des citoyens pour mettre en œuvre la technologie IoT à

tous égards. Grâce à l'Internet des objets, les villes peuvent être améliorées à plusieurs niveaux : infrastructures, transports, etc **[30].**

Figure 4 : les villes intelligentes.

1.10.2 Grille Intelligente « Smart Grid »

Le « Smart Grid » (ou « réseau électrique intelligent ») désigne la nouvelle génération des systèmes énergétiques électriques desservant chaque habitant, chaque entreprise et chaque service d'infrastructure dans une ville. Ce réseau a été mis au point avec des technologies de communication et de connectivité pour favoriser une utilisation plus efficace des ressources.

Les technologies qui confèrent l'aspect « intelligent » à ce réseau énergétique basé sur l'IoT comprennent des dispositifs sans fil tels que des capteurs, des modules radio, des passerelles et des routeurs cellulaires.

Ces appareils fournissent la connectivité de pointe afin de résoudre les problèmes de décision en matière d'utilisation de l'énergie, d'économie d'électricité, et permettent aux autorités électriques de rétablir plus rapidement le courant après une panne et de moderniser la technologie actuelle vieillissante dont l'entretien est coûteux **[31].**

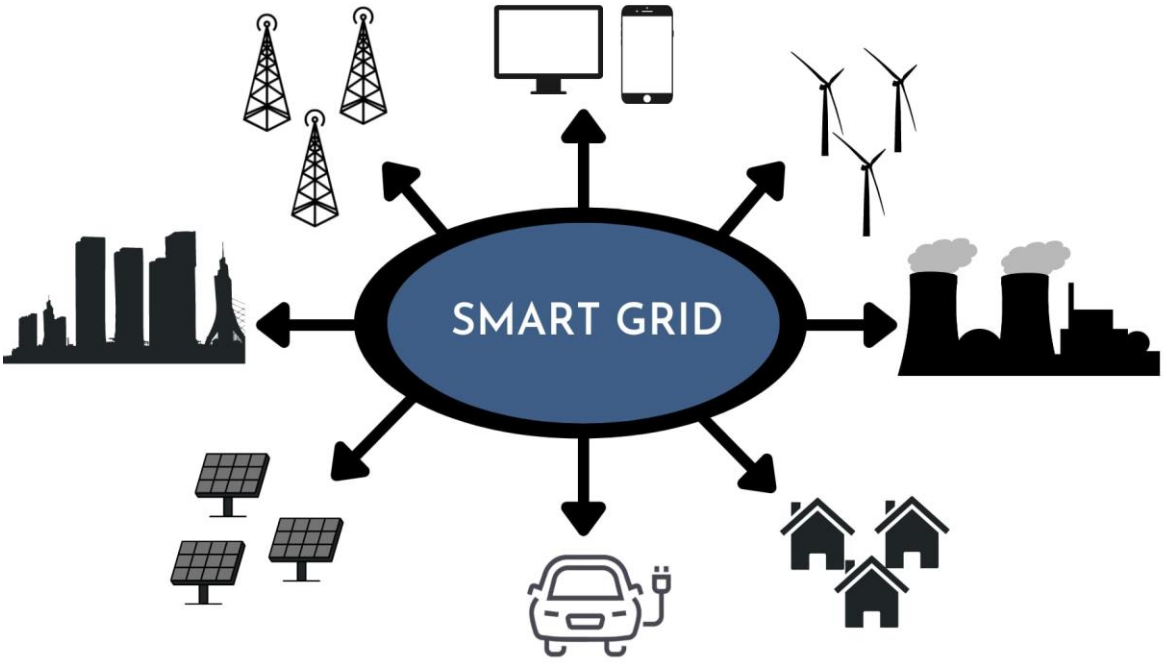

Figure 5 : Grille Intelligente .

1.10.3 Les appareils intelligents « Smart Devices »

Les appareils intelligents jouent un rôle fondamental dans l'Industrie d'aujourd'hui. Ils sont au centre de l'Internet des objets (IoT) et des villes intelligentes.

"Un appareil intelligent est un appareil électronique sensible au contexte,capable d'effectuer un calcul autonome et de se connecter à d'autres appareils avec ou sans fil pour l'échange de données ». Cette définition proposait qu'un appareil intelligent ait trois caractéristiques principales, à savoir la sensibilité au contexte, l'informatique autonome et la connectivité. Cette définition est conforme à l'idée principale de l'IoT qui est que toute « chose » peut faire partie de l'IoT. Une chaise peut devenir une chaise intelligente si nous ajoutons un capteur, un tout petit peu de capacités informatiques et une connectivité réseau **[32].**

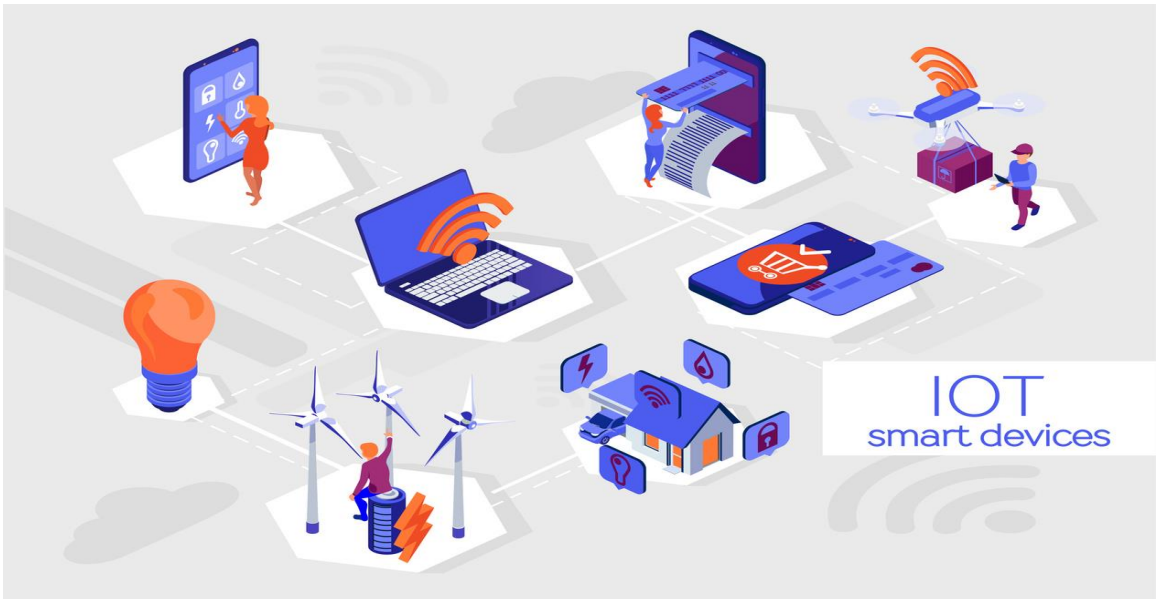

Figure 6 : Smart devices**.**

- Sensibilisation au contexte:

Capacité d'un système ou d'un composant de système à collecter des informations sur son environnement à un moment donné et à adapter ses comportements en conséquence.

Les caméras, les microphones et les récepteurs GPS (Global Positioning Satellite), les radars et les capteurs LIDAR sont tous des sources potentielles de données pour l'informatique contextuelle. Un système sensible au contexte peut collecter des données via ces sources et d'autres et répondre selon des règles préétablies ou via l'intelligence informatique.

- Informatique autonome :

L'aspect clé de l'informatique autonome est un appareil ou plusieurs appareils effectuant des tâches de manière autonome sans la commande directe de l'utilisateur. Par exemple, lorsque nos smartphones font une suggestion en fonction de notre géolocalisation ou de la météo. Pour effectuer cette tâche "simple", un smartphone doit être autonome et utiliser des données contextuelles pour prendre des décisions.

- Connectivité :

La connectivité fait référence à la capacité d'un appareil intelligent à se connecter à un réseau de données. Sans connectivité, il est inutile qu'un appareil intelligent soit autonome et sensible au contexte. La connectivité réseau est une fonctionnalité cruciale qui permet à un appareil de faire partie de l'Internet des objets. La connectivité réseau peut être filaire ou sans fil.

1.11 Conclusion

Dans ce premier chapitre nous avons présenté en détail divers principes et notions de l'Internet des Objets. Nous avons commencé par présenter un bref historique, suivi par une description de l'architecture des IoT. Nous nous sommes ensuite intéressés aux différentes caractéristiques des IoT et les défis qu'ils doivent relever. Notre intérêt s'est très clairement porté sur l'aspect sécurité, puisque cela fait l'objet de notre thématique. Nous avons aussi parlé de certaines technologies de communication et cité certains domaines d'applications des IoT. A la fin de ce chapitre nous avons expliqué les deux principaux modes de communication, à savoir, le mode M2M et le mode M2C.

Dans la suit, nous nous intéresserons plus amplement à l'aspect sécurité des IoT. Nous parlerons en particulier de la gestion des identités d'objets et des divers mécanismes d'authentification dans ce contexte.

Chapitre 02 : La sécurité dans l'Internet des Objets.

2.1 Introduction :

La sécurité de l'IoT est un enjeu majeur pour la pérennité et la compétitivité des entreprises et administrations. La Federal Trade Commission (FTC) des États-Unis a souligné dans un rapport que le déploiement prévu de la technologie IoT ouvrira divers problèmes de sécurité et de confidentialité pour les utilisateurs de l'IoT, qui doivent être bien résolus. Pour bon nombre de ces applications IoT critiques, l'utilisation des données corrompues de manière malveillante peut avoir de graves conséquences. Les objectifs de la sécurité conventionnelle tels que l'authentification, la confidentialité et l'intégrité des données sont essentiels pour les objets, les réseaux et les applications IoT.

Ce chapitre fait le tour de la question de la sécurité dans les systèmes IoT. Nous commençons par une présentation des propriétés et des mécanismes de sécurité. Ensuite, nous représentons la notion d'identité et d'authentification dans les IoT. Ensuite, nous accordons une attention particulière à l'authentification telle que ses types et ses protocoles, à la fin nous citons les défis et les problèmes rencontrés dans l'authentification.

2.2 Les propriétés de sécurité et les mécanismes :

La **sécurité des systèmes d'information** (**SSI**) ou plus simplement **sécurité informatique** peut être définie comme étant le fait de protéger ou d'assurer du bon fonctionnement d'un système et garantir les résultats attendus de sa conception. En d'autres termes, la sécurité représente l'ensemble des politiques et des pratiques adoptées qui visent à empêcher une utilisation non autorisée, le mauvais usage, la modification ou le détournement du système **[33]**. Depuis cette définition, nous pouvons extraire les propriétés et les mécanismes de sécurité présentés ci-dessous.

2.2.1 Les propriétés de sécurité

- **Confidentialité :**

La confidentialité est le mécanisme qui permet de cacher une donnée, et de cacher même l'information de son existence. Elle permet aussi d'empêcher toute entité non autorisée d'avoir accès à ces données. En général, ce service repose sur des

algorithmes mathématiques qui permettent de déformer un texte brut et de lui redonner sa forme initiale grâce à une ou plusieurs clés de chiffrement et de cryptages.

Dans l'Internet des objets, en particulier dans le contexte de la communication, la confidentialité signifie assurer la protection des données échangées entre les appareils intelligents contre l'interception de personnes non autorisées et cela en utilisant des mécanismes dédiés.

➢ Intégrité :

L'intégrité garantit que les données ne sont ni falsifiées ni modifiées ou altérées ni supprimées par une entité non autorisée. Cela est assuré grâce aux points suivants :

- Empêcher la modification des informations par des utilisateurs non autorisés
- Empêcher la modification non autorisée ou involontaire des informations par des utilisateurs autorisés
- Préserver une cohérence interne et externe :

Cohérence interne – Elle garantit que les données sont cohérentes sur le plan interne.

Cohérence externe – Elle garantit que les données stockées dans la base de données sont cohérentes avec le monde réel.

Diverses méthodes de chiffrement peuvent contribuer à assurer l'intégrité en confirmant qu'un message n'a pas été modifié pendant sa transmission. Une modification peut rendre un message inintelligible ou, pire encore, inexact. Imaginez les graves conséquences que pourraient avoir des modifications non découvertes apportées à des dossiers médicaux ou à des prescriptions de médicaments. Si un message est altéré, le système doit comporter un mécanisme indiquant que le message a été corrompu ou modifié **[34]**.

➢ La disponibilité :

C'est la garantie d'accès à un service ou à des ressources afin de maintenir le bon fonctionnement du système Internet des Objets. Ainsi, assurer la disponibilité signifie construire le système pour réduire l'impact d'un déni de service (Denial of service (Dos)), pour un service auquel un attaquant pourrait essayer de bloquer l'accès.

➢ Authentification :

L'authentification permet de prouver l'identité d'une entité. Dans un système IoT, l'authentification représente la première barrière de sécurité pour empêcher une personne tierce et non autorisée d'accéder aux données des dispositifs intelligents. En

fait, il existe de nombreuses méthodes d'authentification qui peuvent être classées en 4 catégories **[34]** :

- **Authentification avec ce que nous savons « Ce que je connais »**, c'est-à-dire que l'entité prouve son identité avec des informations classifiées, qui ne sont connues que d'un nombre limité d'êtres légitimes. En général, le nombre d'éléments concernés ne dépasse pas 2 (par exemple, client et serveur). Les mécanismes les plus couramment utilisés dans cette catégorie sont Mots de passe et numéros d'identification personnels (Personal Identity Number (PIN)).
- **Authentification avec ce que nous avons « Ce que je possède »**, l'entité dans cette catégorie s'authentifie à l'aide des données stockées. . Ces données peuvent être aussi confidentielles qu'une clé pré-partagée (Pre-Shared Key (PSK)),ou publiques comme les certificats numériques.
- **Authentification avec qui nous sommes « Ce que je montre »,** cela concerne généralement les utilisateurs humains, qui ont des propriétés biométriques qui leur sont propres telles que Voix, empreinte digitale, iris et veines.
- **Authentification de notre façon d'agir « Ce que je fais »**, cette catégorie est basée sur les profils comportementaux de chaque utilisateur. Chaque entité a un chemin certains travaux, par exemple, sa façon de taper sur le clavier, et les montres conditions de travail habituelles, environnement de travail habituel, etc.

➢ Non-répudiation :

La non-répudiation est un mécanisme garantissant qu'une opération ne peut être niée par celui qui l'avait établie. La non-répudiation assure que l'émetteur du message ne pourra pas nier dans le futur avoir émis le message. La non-répudiation peut être assurée en utilisant un mécanisme cryptographique appelé : signature numérique **Figure7**. Ce mécanisme repose sur un système cryptographique asymétrique, il est calculée en utilisant la clé privé de l'émetteur et vérifiée en utilisant la clé publique de l'émetteur.

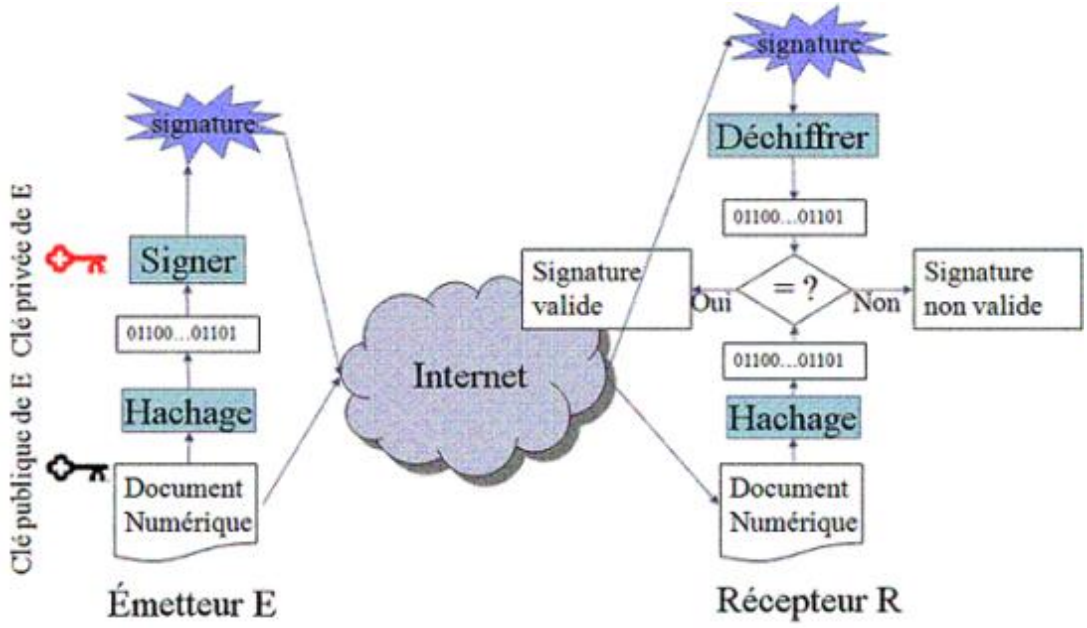

Figure 7 : Signature numérique et non répudiation.

2.2.2 Les mécanismes de sécurités

Dans cette section, nous proposerons les principaux types de mécanismes de sécurité. En général, les mécanismes de sécurité visent à protéger l'accès aux actifs (c'est-à-dire les données et les ressources) du système contre les diverses menaces. Ainsi, l'utilisation de mécanismes de sécurité permet de mettre en œuvre des services de sécurité pour empêcher la détection et/ou la modification non autorisée des données et/ou l'accès non autorisé aux ressources. Par exemple, pour assurer l'accès aux ressources, seuls les utilisateurs autorisés peuvent configurer un service d'authentification. Cela peut être obtenu grâce à l'utilisation de divers mécanismes de sécurité : affichage du nom d'utilisateur et du mot de passe, technologie de défi-réponse utilisant le cryptage symétrique ou signature numérique, etc. **[35].**

- **Les mécanismes de chiffrement**

Pour réaliser le cryptage, le processus de conversion du texte clair en un message crypté, il est possible d'utiliser un chiffrement symétrique ou un chiffrement asymétrique. L'objectif du chiffrement est de protéger la confidentialité des informations échangées entre les entités communicantes **[35].**

- **Le chiffrement symétrique**

Afin d'échanger des données en toute sécurité, à l'aide d'un algorithme de chiffrement symétrique, l'expéditeur et le destinataire (ou les destinataires) doivent utiliser un secret partagé (c'est-à-dire clé). La notion de symétrie vient du fait que c'est la même

clé, qualifiée de secrète, que les entités connectées utilisent à la fois pour le cryptage (le processus de conversion d'un message clair en un message crypté) et le décryptage (le processus de conversion d'un message crypté en un message clair). La clé doit rester secrète et ne pas être dévoilée aux entités du système qui n'y ont pas droits (par exemple, la clé peut être partagée au sein d'un groupe si nécessaire et devenir une clé de groupe, cependant, il sera difficile d'exclure un membre de groupe car vous devez à nouveau partager la clé et cela peut être contraignant) **[36][37]**. Le problème principal du chiffrement symétrique est qu'il nécessite un partage de clé avant que des données secrètes et chiffrées ne soient échangées. Le partage de la clé se fait via un canal sécurisé (c'est-à-dire un canal sur lequel l'attaquant n'a aucun contrôle), ce qui est difficile à réaliser concrètement. Un échange à main propre peut être envisagé pour limiter les prises de risque.

Les méthodes les plus connues pour le chiffrement symétrique sont le DES (Data Encryption Standard), le 3 DES (Triple DES) et l'AES (*Advanced Encryption Standard)* .

- ### Le chiffrement asymétrique

Pour effectuer un chiffrement asymétrique, chaque entité doit posséder deux clés cryptographiques : une clé privée (connue uniquement de l'entité qui la possède) et une clé publique (tout le monde peut y accéder) **[38]**. La taille des clés est plus importante et elle peut aller jusqu'à 4 096 bits, rendant ce type d'algorithme ou ce type de chiffrement extrêmement fiable, voire même totalement sécurisé à 99% jusqu'à preuve du contraire.

Le RSA est le plus connues pour le chiffrement asymétrique, aussi appelé chiffrement à clé publique.

➢ Les fonctions de hachage

Les fonctions de hachage sont des fonctions qui prennent en entrée des données de longueur arbitraire et produit en sortie une version condensée de ces mêmes données mais cette fois-ci de longueur fixe. Ce condensé est plus communément appelé "haché de données ou empreinte". Selon la fonction de hachage utilisée, le volume de sortie produit varie.

Les fonctions de hachage sont largement utilisées dans différents protocoles cryptographiques, dans l'authentification, les signatures, etc. **[35]**

Les principales propriétés que doit avoir une fonction de hachage cryptographique sont **[39]:**

- L'empreinte d'un message doit dépendre de tous les bits du message, si un seul bit change, le haché ou l'empreinte ne doit avoir aucune relation avec le haché du texte d'origine.
- Une fonction de hachage cryptographique doit être résistante aux préimage, à la seconde pré-image et aux collisions. Les trois problèmes suivants doivent être très difficiles:

Pré-image: étant donné un haché h choisi aléatoirement. Trouver un message m tel que H(m)=h.

Seconde pré image: étant donné un haché h choisi aléatoirement. Trouver un message m′ tel que H(m)=H(m′).

Collision: trouver deux messages m et m′, tels que m≠m′ et H(m)=H(m′).

On définit la résistance d'une fonction aux collisions, aux pré images et aux deuxièmes pré images par rapport à la difficulté de résoudre ces problèmes en pratique. Cette difficulté est évaluée par rapport aux nombres d'opérations nécessaires pour que la meilleure attaque générique contre une fonction de hachage idéale réussisse.

Il existe plusieurs fonctions de hachage ayant différents algorithmes, les plus usuelles sont : MD5, SHA-1, SHA-2, RIPEMD, Tiger…etc.

➢ La signature de numérique

La cryptographie n'est pas seulement du chiffrement, d'autres tâches peuvent être réalisées pour assurer la sécurité des données comme la signature électronique. C'est un procédé semblable à la signature manuscrite à quelques différences près. Elle engage la responsabilité du signataire sur le contenu du message signé. Une signature numérique ne doit pas pouvoir être reniée et doit pouvoir être vérifiée par tout le monde. Contrairement à une signature manuscrite qui est physiquement liée à un document, pour arriver à coller une signature électronique au message, il faudrait que cette dernière dépende du signataire mais aussi du message signé.

Une signature numérique a la même utilité qu'une signature manuscrite. Cependant, une signature manuscrite peut être facilement imitée, alors qu'une signature numérique est pratiquement infalsifiable. Les procédures de signature précédentes ont un coût élevé pour signer de longs messages car la signature est aussi longue que le message. On double donc la longueur du texte à crypter. Pour réduire la longueur de la signature on peut utiliser une fonction de hachage cryptographique. **[39]**

Prenons l'exemple de la signature de Schnorr. C'est une solution de signature numérique qui utilise l'algorithme d'authentification décrit par Claus Peter Schnorr **[40,41].** La Sécurité de cet algorithme est basée sur la difficulté de calculer le logarithme discret.

L'algorithme d'authentification a lieu entre P (vérificateur) qui veut prouver sa légitimité et V (vérificateur) qui veut la vérifier. Le fonctionnement général de l'algorithme consiste à définissez des paramètres généraux incluant l'ensemble des G générés par g pour q (q est un nombre premier).

La preuve P contient une clé secrète x ∈ Z
∗Q et une clé publique y telle que y = g
x qui sait de V

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

1. P génère un entier aléatoire r tel que r ∈ Z ∗ q.
2. P calcule R = g r et envoie R à V.
3. V génère un entier aléatoire c tel que c ∈ Z ∗ q et l'envoie à P.
4. P calcule a tel que a = r −c · x et l'envoie à V.
5. V vérifie si R ? = g a · y c (= g a ·(g x) c = g (r−c·x) · g x·c = g r−c·x+x·c = g r = R). Si oui, alors P est une entité légitime qui connaît bien x **[35].**

Pour signer un message M en utilisant un schéma non interactif, selon les commandes Fiat Shamir **[35,36]** avec le protocole Schnorr, fonction de hachage h : G × {0, 1}∗ → Z

∗q est montré en public.

Le signataire doit effectuer les étapes suivantes :

1. Générer un entier aléatoire r tel que r ∈ Z ∗ q et calculer R = g r.
2. Calculer c = h(R,M).
3. Calculer a = r −c · x.

La signature est σ = (c,a).

Le destinataire connaissant la clé publique du signataire pourra vérifier que le message M est valide à partir de σ si c ≡ h(R,M) est valide sachant que R se calcule avec R = g a · y c (= g r−c·x · (g x) c = g r−c·x · g x·c = g r−c·x+x·c = g r)**[35]**.

Les signatures Schnorr sont reconnues pour leur simplicité, leur rapidité et leur efficacité. Si ils étaient peu utilisés car ils sont protégés par des brevets, et ceux-ci ont décliné et ils gagnent en popularité face à d'autres méthodes comme les signatures ECDSA ("Elliptic Curve Digital Signature algorithme ») **[42].**

2.3 Technologies de communication de l'IoT et leurs mécanismes de sécurité :

❖ La sécurité dans LoRaWAN :

La politique de sécurité de LoRaWAN assure les objectifs de base de la cryptographie qui sont l'authentification des objets, la confidentialité et l'intégrité des données. Cette politique définie également des techniques de partage de clés **[43]**

- **Authentification des objets et partage des clés** :

La première méthode d'authentification et de distribution de clés s'appelle l'Over The Air Activation (OTAA). Chaque objet est muni d'une clé d'application (AppKey), qui est une clé symétrique unique de 128 bits pré-partagée avec le serveur réseau de LoRaWAN. Afin qu'un objet puisse être pair à un réseau LoRaWAN, il doit d'abord lancer une requête d'association (join request) au réseau. Cette requête qui contient l'identifiant unique d'application (AppEUI), l'identifiant unique de l'objet (DevEUI), et un nonce fortuit (pour éviter les attaques par cryptanalyse) de l'objet (DevNonce) d'une taille de 2 octets, doit être fini par un code d'intégrité de message (Message Integrity Code (MIC)). Le MIC (4 octets) représente la souscription du message en utilisant AppKey. **[44]**

❖ La sécurité dans 6LoWPAN

Comme avec la plupart des technologies IEEE 802.15.4, 6LoWPAN assure la confidentialité. En revanche, il ne connaît pas de méthode d'authentification spécifique, Ni pour gérer les clés. Un travail intéressant a été proposé par pour finaliser la Méthode d'authentification qui utilise le protocole d'authentification extensible Generic Pre-SharedKey (EAP-GPSK), qui est basé sur le cryptage symétrique**[49].**

- **Authentification et mécanisme d'échange des clés :**

La **figure 8**, L'architecture réseau de 6LoWPAN définie est présentée dans L'architecture du réseau consiste en un appareil qui se connecte à un Coordinateur du réseau personnel (CPAN). Un CPAN peut être un routeur, une passerelle ou tout autre appareil capable d'acheminer Paquets de données entre différents réseaux. Cette structure contient également un Serveur d'authentification Serveur d'authentification (AS) utilisé pour publier un ou plusieurs Fonction d'authentification, d'autorisation et de comptabilité/audit (AAA**)[50]. La figure 9** décrit le mécanisme d'authentification mutuelle proposé pour 6LoWPAN. Tout d'abord l'appareil envoie un message démarrage du protocole d'authentification extensible (message de démarrage EAP) au CPAN.

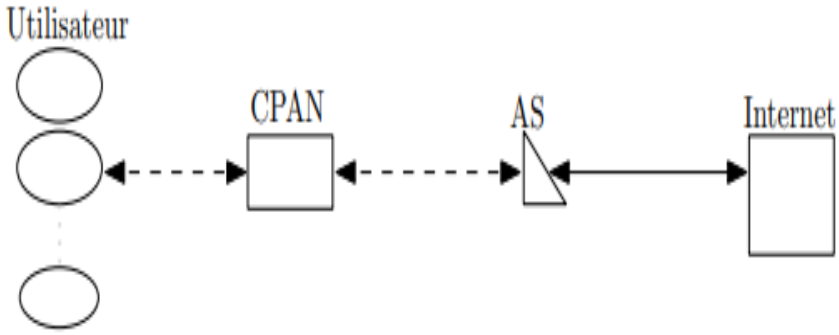

Figure 8 : Une architecture d'un réseau 6LoWPAN.

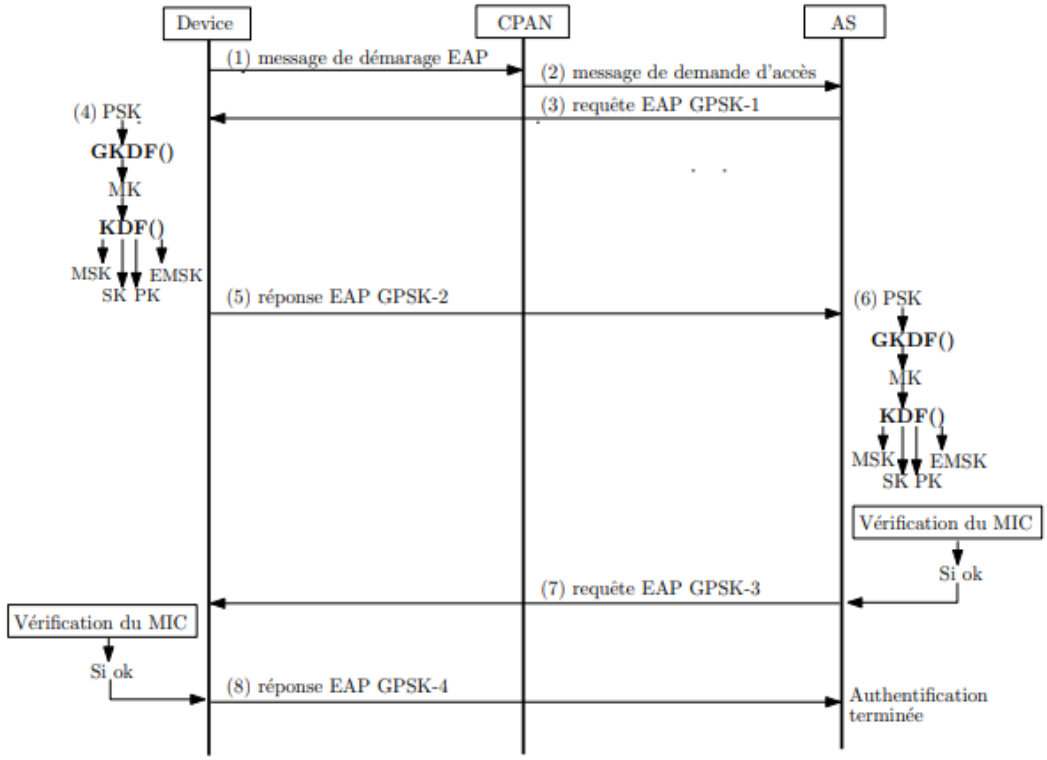

Figure 9 : Mécanisme d'authentification mutuelle EAP-GPSK (proposé pour 6LoWPAN).

- **Confidentialité et intégrité de données :**

Pour protéger les données échangées, elle recommande l'utilisation de la norme AESCCM*, un algorithme qui assure la sécurité et la confidentialité des services.

- **La sécurité dans ZigBee**

Pour des raisons de sécurité, il est requis au niveau de la couche application et de la couche réseau. Chaque couche est responsable de la sécurisation de son volume de données.

- **Confidentialité et intégrité de données :**

Les services de confidentialité et d'attention sont fournis par un cryptage authentifié couche application et couche réseau. En fait, les messages sont doublement protégés au niveau des deux couches séparément, en utilisant la norme AES-CCM.

2.4 Identification dans l'IdO :

Dans le monde informatique, **L'identification** est une phase qui consiste à établir l'identité de l'utilisateur. Elle permet de répondre à la question : "Qui êtes-vous ?". **L'authentification** est une phase qui permet à l'utilisateur d'apporter la preuve de **son identité**. Elle intervient après la phase d'identification. Elle permet de répondre à la question : "Êtes-vous réellement cette personne ?".

Différents systèmes IdO requièrent divers justificatifs pour confirmer une identité. Ce justificatif se compose souvent d'un mot de passe. Mais il peut également s'agir d'autres formes d'authentification, en fonction du service et des caractéristiques de sécurité dont l'objet doit disposer. L'ensemble de la procédure comporte deux phases principales : l'identification et l'authentification réelle. Cette dernière permet de s'assurer que tout est sécurisé et que l'utilisateur a accès aux bons outils.

2.4.1 L'identité dans l'IoT :

La connexion sécurisée des appareils intelligents à un réseau représente un défi pour de nombreux responsables de la sécurité, car ils doivent avoir accès à une grande variété de données et d'applications qui dépendent de l'utilisateur et de leur finalité. Selon les prévisions de Gartner, d'ici à 2020, plus de 20 milliards d'"objets" connectés, aux rôles et tâches les plus divers, seront exploités par les particuliers et les entreprises du monde entier. Cependant, il ne suffit pas d'intégrer ces appareils dans la stratégie de sécurité existante sans faire attention à l'utilisation de l'identité des objets **[51]**.

2.4.1.1 Le principe d'identité :

L'identité des objets, comme celle des êtres humains, présente certaines caractéristiques qui permettent de la définir. Un objet a besoin d'une identité pour pouvoir disposer d'un accès relativement rapide et administrativement moins lourd à certains systèmes, certaines données, certains dossiers, etc. Cependant, le principe qui consiste à accorder "autant d'accès que nécessaire et aussi peu d'accès que possible" s'applique. Ce contrôle d'accès est déterminé par l'identité d'un objet. Les identités des objets sont également uniques, puisqu'elles permettent de chiffrer, d'exécuter des codages ou de contrôler globalement le fonctionnement d'autres objets. Par exemple,

l’information contenue dans l’identité d’un objet, et reconnue n’est autre que : que fait cet appareil, où se trouve-t-il, s’il est considéré comme fiable, et si oui l’autorisé ou non à y accéder. L’identité des objets dépend de certificats numériques et de clés cryptographiques qui enregistrent et identifient mutuellement leurs informations, afin d’instaurer la confiance nécessaire à une communication et à un accès à l’information sécurisés **[51]**.

- **Cycle de vie** : certaines entités IoT peuvent avoir une durée de vie assez longue. Le dossier médical électronique (DME) de la personne. Quant aux autres entités, leur cycle de vie est très court.
- **Relations** : il est important de savoir comment une entité IoT est liée à d'autres entités, y compris non seulement d'autres éléments, mais également des entités externes telles que des propriétaires, des administrateurs et d'autres parties responsables.
- **Sensibilisation au contexte** : la gestion des identités et des accès (IAM) pour les entités IoT doit dépendre du contexte. Il peut être approprié, par exemple, qu'une entité accède à une autre entité ou à un autre système dans certaines circonstances, et lui accorder l'accès d'une autre manière est inapproprié, voire dangereux.
- **Authentification** : l'authentification multi facteur est efficace pour identifier les humains, mais est moins efficace pour les questions liées à l'IdO car de nombreuses méthodes - la vérification biométrique, par exemple - ne sont pas pertinentes. Il est nécessaire de trouver d'autres moyens d'authentifier en toute sécurité les identités IoT.

2.4.1.2 Système de gestion d’identité (IdM) :

IdM est un système ou mécanisme de gestion des identités, leur authentification les autorisations liée à ces identités .Les modèles des IdM ont été classés selon trois types : isolé, centralisé et fédéré. **[51]**.

Dans le modèle isolé qui est le plus simple, le fournisseur de service SP (Service provider qui offre un ou plusieurs services aux utilisateurs) joue à la fois le rôle d’un SP et d’un fournisseur d’identité IdP (Identity provider qui sert à identifier chaque utilisateur), ce qui signifie que toutes les informations stockées pour une identité et les différentes opérations d’un utilisateur sont gérées par un seul serveur.

Le modèle centralisé Implémentation du modèle central dans le modèle client-serveur. Ce modèle caractérise les fonctions SP et IdP. Le fournisseur d'identité IDP, le stockage de l'identité de l'utilisateur, l'authentification de l'utilisateur et l'authentification de l'utilisateur gèrent tous les services IDP. La forme principale est bien ajustée, ce qui provoque une occurrence particulière dans le problème de la vie privée.

Dans le modèle fédéré, Sous la forme unifiée, protocoles, règles et normes. Il est établi entre les fournisseurs de services, de sorte que les identités de différents domaines d'identité peuvent être identifiées au cours d'une session même s'il existe une correspondance entre différents identifiants appartenant au même utilisateur dans différents domaines. Par conséquent, les utilisateurs d'un domaine peuvent accéder aux services d'un autre sans avoir à s'authentifier à nouveau. Par conséquent, les utilisateurs peuvent accéder à tous les services du domaine fédéré avec une seule connexion.

2.4.2.3 Outils et technologies de gestion des identités

Les outils et technologies répondent également aux problèmes de sécurité IdM, parmi les différentes méthodes d'authentification utilisées dans IdM figurent les mots de passe, les jetons, les cartes à puce et les données biométriques. Chacun d'eux a un niveau de sécurité, un contexte et un domaine d'application différents.

Depuis plusieurs années, les mots de passe et les codes sont largement utilisés dans divers systèmes et applications. De nos jours, les cartes à puce et la biométrie avec système de communication RFID sont des tendances très populaires. Dans ce qui suit, nous définirons et examinerons brièvement les deux dernières méthodes d'authentification **[51]**.

- **Les cartes à puce**

Les cartes à puce sont utilisées comme technologies de confidentialité, ainsi que pour contrôler l'accès aux informations personnelles dans un système particulier. Il est doté d'un certain espace de stockage, d'une capacité de calcul et d'un système de cryptage des données. Les cartes à puce sont équipées d'un processeur cryptographique pour la génération et le stockage des clés, afin d'assurer une authentification forte et des signatures électroniques**.**

- **La biométrie**

La biométrie est utilisée actuellement comme technologie d'authentification, qui mesure et analyse les caractéristiques distinctives et mesurables du corps humain afin de s'authentifier à un système particulier. Il existe plusieurs identificateurs biométriques, tels que l'ADN, les empreintes digitales…. Ce type d'authentification est très utilisé dans les entreprises t les publiques, elle permet une correspondance entre les individus et les identificateurs enregistrer dans les systèmes d'authentification. D'où, il est très difficile de falsifier ou d'utilisé l'identificateur biométrique d'un individu.

2.5 L’authentification dans l’IoT :

Avec le développement du domaine de l'Internet des objets et le besoin illimité du monde pour celui-ci. Nous devons prendre soin de la sécurité dans tous ses aspects, en utilisant plusieurs phases comme l'authentification, dont nous parlerons en profondeur dans ce qui suit.

Ci-dessous, nous présenterons les différentes formes d'authentification. La classification des solutions d'authentification va nous aider à mieux comprendre pourquoi il est essentiel de disposer d’un système d’authentification approprié dans un système IoT.

2.5.1 Définition

En termes simples, l’authentification constitue la procédure de reconnaissance de l’identité d’un utilisateur. Elle s’exécute au début d’une application et valide les utilisateurs pour s’assurer qu’ils remplissent toutes les conditions de sécurité **[52]**. L'authentification de l’objet est le processus d'authentification générale d'un objet ou d’un appareil sur des réseaux câblés ou sans fil lorsque l’objet est un «demandeur» cherchant à accéder à des informations ou à les partager ou à réaliser un autre type d'interaction numérique. L'authentification de l’objet se produit de différentes manières dans diverses configurations informatiques, mais implique généralement un «certificat numérique» comme dans le protocole SSL utilisé sur Internet **[53].**

2.5.2 Les différents types d’authentification :

Dans IoT, le but est de chercher à mettre en place des systèmes connectés plus sophistiqués pour sécuriser leurs utilisateurs et s’assurer que la sécurité soit assurée. Cela explique pourquoi il existe tant de types d’authentification, qui tentent de couvrir une grande variété d’exigences.

- **Authentification primaire/ Facteur unique :**

Cette méthode permet de sécuriser le plus simplement possible l’accès à un système. Dans ce cas, il suffit de faire correspondre un seul justificatif d’identité pour se vérifier en ligne. Il s’agit le plus souvent d’un mot de passe, la forme d’authentification la plus populaire.

- **Authentification à deux facteurs (2FA)**

Une couche de sécurité supplémentaire, garantissant qu’un système est plus sûr pour ses utilisateurs. De fait, après avoir saisi son identifiant et son mot de passe, l’utilisateur doit franchir une étape supplémentaire. Il fournit un autre élément d’information, qui peut être un code PIN, une réponse à une « question secrète », un numéro envoyer par SMS/email. Il offre même ses caractéristiques biométriques (en utilisant Face ID ou Touch ID par exemple).

- **Authentification unique (SSO)**

Le SSO permet de s'authentifier en toute sécurité auprès de plusieurs comptes en ligne en utilisant un seul ensemble d'informations d'identification. En fait, ce système est utilisé chaque fois que la connexion est possible à l'aide de Google, Apple, Facebook ou un autre fournisseur. Le principe repose sur un certificat que le fournisseur de services et le fournisseur d'identité ont échangé. Celui-ci envoie les informations d'identité à un fournisseur de services par le biais de ce certificat afin de savoir qu'elles proviennent d'une source fiable.

- **Authentification multi-facteurs (MFA) :**

Avec l'AMF, il faut fournir deux facteurs de vérification ou plus pour accéder au système. Le système se compose d'une application, d'un compte en ligne ou d'un VPN. Cette méthode d'authentification représente la principale composante d'une politique de gestion des identités et des accès. L'utilisation de l'AMF pour les systèmes diminue les chances de réussite d'une cyberattaque.

2.5.3 Les différents protocoles d'authentification:

Dans le domaine des technologies de l'information, un protocole désigne l'ensemble spécial de règles que les points d'extrémité d'une connexion de télécommunication utilisent lorsqu'ils communiquent. Les protocoles spécifient les interactions entre les entités communicantes.

- **Norme d'authentification FIDO2**

FIDO2 désigne la combinaison de la spécification de l'Alliance FIDO pour les protocoles client-authentificateur (CTAP) et de la spécification d'authentification Web (WebAuthn) du World Wide Web Consortium (W3C). Ces derniers permettent aux utilisateurs de s'authentifier auprès de services en ligne à partir d'environnements mobiles et de bureau à l'aide d'un dispositif d'authentification interne ou externe.

- **Pretty Good Privacy**

PGP (Pretty Good Privacy) peut être utilisé pour signer, crypter et décrypter presque tout ! Pretty Good Privacy (PGP) est un protocole de chiffrement des données qui utilise une combinaison de chiffrement symétrique et asymétrique pour permettre à deux parties d'échanger des données en toute confidentialité.

- **Secure Shell (SSH)**

Secure Shell (SSH) est un protocole réseau qui permet des communications sécurisées entre un client SSH et un serveur SSH sur un réseau non sécurisé (par exemple, Internet). Classiquement, SSH offre deux mécanismes d'authentification de haut niveau : les mots de passe et la cryptographie à clé publique. L'authentification par clé publique est généralement considérée comme plus sûre. En effet, elle évite de devoir stocker des mots de passe et élimine la possibilité qu'une personne compromise…

- **FIDO – Identité rapide en ligne**

L’authentification Fast Identity Online (FIDO) est un ensemble de spécifications techniques ouvertes. Elles définissent des mécanismes d’authentification des utilisateurs qui réduisent la dépendance aux mots de passe. À ce jour, l’Alliance FIDO a publié trois ensembles de spécifications. Ces derniers fournissent un moyen standard d’interfacer un authentificateur matériel à second facteur.

- **Protocole client – Authentificateur (CTAP/CTAP2)**

Le protocole CTAP ou Client To Authenticator Protocol est une spécification décrivant comment une application établit des communications avec un dispositif d’authentification. Cette spécification fait partie du projet FIDO2 et de la spécification WebAuthN du W3C.

- **Protocole d’authentification extensible (EAP)**

Extensible Authentification Protocol (EAP) est un cadre d’authentification, et non un mécanisme d’authentification spécifique. Ce protocole est fréquemment utilisé dans les réseaux sans fil et les connexions point à point. En outre, il fournit certaines fonctions communes et la négociation de méthodes d’authentification appelées méthodes EAP. Le protocole EAP peut prendre en charge plusieurs mécanismes d’authentification sans avoir à en pré-négocier un en particulier. Il existe actuellement environ 40 méthodes différentes définies.

- **Connexion sécurisée, rapide et fiable (SQRL)**

Secure, Quick, Reliable Login, ou SQRL (prononcé « squirrel »), désigne un projet de norme ouverte pour l’identification et l’authentification anonymes. Ces derniers sont sécurisés pour des utilisateurs de sites et d’applications Web. Il a été proposé par son inventeur, Steve Gibson, comme un substitut facile à utiliser aux noms d’utilisateur, aux mots de passe et à la MFA. SQRL a été conçu pour éliminer l’authentification par nom d’utilisateur et mot de passe sur les sites Web distants.

- **Echange de clés sur internet (IKE)**

Internet Key Exchange (IKE) est le protocole utilisé pour établir un canal de communication sécurisé et authentifié entre deux parties. Il utilise généralement des certificats PKI X.509 pour l’authentification et le protocole d’échange de clés Diffie-Hellman pour établir un secret de session partagé. En fait, IKE fait partie du protocole de sécurité Internet (IPSec), qui est responsable de la négociation des associations de sécurité (SA).

2.5.4 Classification des solutions d'authentification IoT :

L’authentification est une sécurité dans les IoT. En effet, elle joue un rôle central dans la sécurité globale des systèmes IoT. Le schéma d'authentification des appareils IoT garantit que ces appareils peuvent être fiables ce qu'ils prétendent être. Un tel schéma fournit à chaque appareil IoT une identité unique (peut être dynamique dans le temps)

qui peut être vérifié lorsque cet appareil tente de se connecter au réseau IoT. Cela donne la possibilité, entre autres, de suivre chaque dispositif tout au long de son cycle de vie (le cas échéant), pour échanger en toute sécurité avec lui, pour l'empêcher d'exécuter du code malveillant, et même s'il arrivait qu'un l'appareil IoT a présenté un comportement inattendu suspect, ses privilèges peuvent simplement être révoqué **[53].**

Dans cette section, nous présentons une taxonomie générale des schémas d'authentification IoT figure 10.

- **Couche IoT :** cela dépend de la couche où la méthode d'authentification est implémentée. Dans cette classification, nous considérons l'architecture IoT la plus courante qui consiste en trois couches couche de perception, couche réseau ou Couche d'application. Cependant, cette classification peut être adaptée à n'importe quelle couche architecture IoT.
- **Domaine d'application :** différents schémas d'authentification sont utilisés pour différents domaines d'application ou environnement IoT différent: machine à machine (Machine to Machine (**M2M**)), Internet des véhicules (Internet of Vehicles (**IoV**)), Internet de l'énergie (Internet of Energy (**IoE**)), Internet des capteurs (Internet of Sensors (**IoS**)) et Internet des objets médicaux (Internet of Medical Things (**IoMT**)).
- **Basé sur 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 (les solutions matérielles implicites qui utilisent le matériel "existant" lors de l'authentification (par exemple, Physical Unclonable Fonction (PUF) ou générateur de nombres aléatoires réels (True Random Number Generator (TRNG)) . Des solutions matérielles explicites qui nécessitent l'utilisation de composants supplémentaires dédiés aux opérations (cryptographique ou autre) effectuées lors de l'authentification (par exemple, Trusted Platforme Module (TPM), Environement d'execution de confiance (TEE)).
- **Facteur d'authentification :** En fonction du nombre de facteurs pris en compte pour l'authentification d'un appareil, une solution peut être classée comme une authentification à un seul facteur (SFA), une authentification à deux facteurs (2FA) ou une authentification à plusieurs facteurs (MFA). Par exemple, si seul le mot de passe de l'utilisateur est utilisé, le schéma d'authentification est appelé schéma à facteur unique. Un système d'authentification à deux facteurs peut utiliser un mot de passe utilisateur et une carte à puce pour authentifier les utilisateurs. Un authentification Multi-Facteur peut utiliser des facteurs supplémentaires tels que les informations de localisation, la biométrie, etc.
- **Accès utilisateur** : différentes méthodes sont utilisées pour l'accès utilisateur ; Certains utilisent la cryptographie à clé public et peut être soit basée sur l'utilisation de certificats numériques (Certificate-based) ou reposant uniquement sur la paire de clés publique/privée (Certificate-Less). D'autres

solutions reposent sur l'utilisation de clés pré-partagées. Solutions hybrides le mélange des deux usages existe aussi.

- **Algorithmes cryptographiques** : 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 surcharge par rapport aux algorithmes asymétriques. Dans cette catégorie de solutions, on peut distinguer les schémas d'authentification utilisant des algorithmes symétriques et ceux utilisant des algorithmes symétriques légers qui ont été introduits pour les appareils contraints tels que les objets IoT. Une autre catégorie des solutions s'appuient uniquement sur la cryptographie asymétrique lors de la phase d'authentification, et peuvent être divisés en ceux utilisant des algorithmes traditionnels (par exemple, RSA) et ceux 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 légèreté. Enfin, des solutions hybrides mélangeant deux ou toutes les méthodes précitées existent également.
- **Procédure d'authentification** : La procédure d'authentification peut être unidirectionnelle, authentification bidirectionnelle ou tripartite à trois voies. Dans l'authentification unidirectionnelle, une seule des deux entités souhaitant communiquer s'authentifiera à l'autre entité. Dans une authentification bidirectionnelle, les deux entités souhaitant communiquer se vérifieront. Ce type d'authentification est également appelé authentification mutuelle. Dans l'authentification à trois voies, les deux entités communicants sont authentifiées à l'aide d'une troisième entité appelée entité centrale d’authentification. 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'un tiers de confiance (entité centralisée), 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. La structure hiérarchique utilise différents niveaux et la structure plate n'a pas de niveau structure.

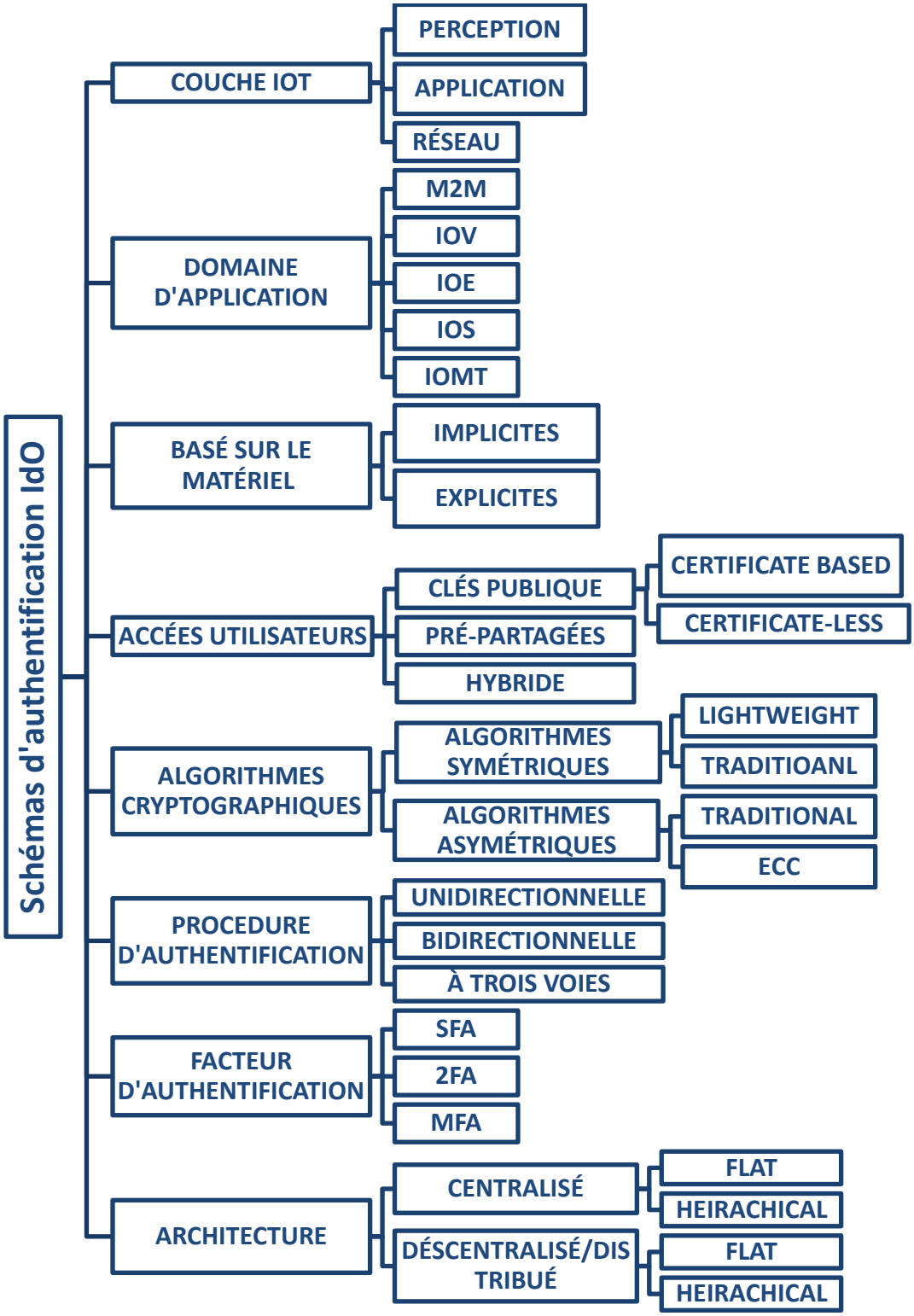

Figure 10 : Taxonomie des schémas d'authentification IdO.

2.5.5 Défis d'authentification IoT et problèmes ouverts

L'examen de la littérature a permis d'identifier un certain nombre de questions ouvertes qui doivent être traitées pour combler les lacunes des schémas

d'authentification existants ; certaines questions ont peut-être été abordées dans la littérature, mais de manière incomplète **[53]** :

- De nombreux schémas d'authentification incluent une phase d'initialisation obligatoire pour les appareils nouvellement ajoutés où une intervention physique sur la passerelle est nécessaire pour installer et configurer les informations d'identification de ces appareils (par exemple, l'installation de PSK clés publiques) qui n'est pas un choix évolutif.

- L'utilisation d'algorithmes asymétriques traditionnels ou de certificats X.509 qui nécessitent des coûts de calcul élevés pour les objets IoT à ressources limitées.

- La nature dépendante de l'architecture des schémas d'authentification qui les inadaptables à d'autres architectures IoT.

- La nature centralisée de certains systèmes d'authentification qui crée des points de défaillance et rend le schéma non évolutif.

2.6 Conclusion :

La sécurité et l'authentification sont importantes dans notre monde, en particulier dans un monde où l'Internet des objets s'est répandu, et c'est ce que nous avons expliqué ci-dessus. Nous avons commencé par un aperçu des propriétés et des mécanismes de sécurité. Ensuite, nous avons représenté le concept d'identité et d'authentification dans l'Internet des objets. Après, nous avons accordé une attention particulière à l'authentification telle que ses types et ses protocoles, et enfin cité les défis et les problèmes ouverts de l'authentification.

Le chapitre suivant décrit trois protocoles d'authentification et se concentre sur la vérification et la comparaison de ces protocoles à l'aide de l'outil AVISPA en ajouton notre proposition.

Chapitre 3 : protocoles d'authentification pour l'IdO.

3.1 Introduction

Actuellement, la question d'authentification est très importante dans de nombreuses applications. Parmi les systèmes prenant en charge les protocoles d'authentification qui se sont développés rapidement ces dernières années, on note l'utilisation de la Radio Frequency Identification (RFID) dans divers domaines d'IdO. La principale caractéristique de ces systèmes est leur utilisation limitée des ressources informatiques, telles que : mémoire, processeur, etc.

La vérification de la sécurité du protocole d'authentification dépend de sa force face à des attaques. Nous avons choisi l'outil AVISPA, qui est le plus populaire pour faire la vérification automatique des protocoles de sécurité.

Ce chapitre se concentre sur la vérification des protocoles Authentification des systèmes RFID à l'aide de l'outil AVISPA après modélisation de ces protocoles en HLPSL. Cela nous permet de comparer les protocoles et de déterminer lequel est le plus sûr et nous permet aussi d'ajouter notre proposition.

3.2 Etat de l'art

Au fil du temps, les chercheurs ont mis au point de plus en plus de protocoles d'authentificationpour résoudre les problèmes de sécurité et de confidentialité qui menacent les systèmes tels que les systèmes RFID**.**

En 2006, Lopez et al. [54–56] ont proposé une famille de protocoles RFID ultralégers nommés UMAP (Ultralightweight Mutual Authentication Protocol) : LMAP [51] (Lightweight Mutual Authentication Protocol), M2AP [56] (Minimalist Mutual Authentication Protocol) et EMAP [54] (Efficient Mutual Authentication Protocol). Ces trois protocoles n'implémentent que quelques opérations binaires très simples sur les balises, y compris ou exclusif (XOR), ou (OR) et addition modulo n (+), où n est la longueur de la chaîne binaire utilisée dans les protocoles. Ces régimes sont assez efficaces pour les étiquettes RFID à faible coût en raison de leur faible coût de calcul et de l'économie d'espace de stockage côté étiquette. Cependant, les littératures [57,58,59,60] ont révélé de nombreuses attaques contre ces trois protocoles.Li et Wang [59] ont présenté deux désynchronisations attaques et une attaque de divulgation complète sur LMAP (ces attaques peuvent également être appliquées à M2AP). Leurs attaques actives sont basées sur l'utilisation de la faiblesse de LMAP pour couper la potentielle relation entre certains messages (comme la relation entre les

messages C et D). De plus, les auteurs [59] ont proposé deux mécanismes (envoi de D et statut de stockage) pour protéger le protocole de leurs attaques. Cependant, l'envoi de D n'est pas utile car un attaquant peut toujours savoir si son attaque est réussie ou non en vérifiant le prochain IDS du tag (nouveau ou ancien).De plus, en raison du problème inhérent à la communication entre deux entrées, il doit y avoir être une entrée ne peut pas assurer l'état de l'autre, ce qui signifie qu'un côté (lecteur/étiquette) ne peut pas connaître l'état de l'autre côté (étiquette/lecteur) dans le système RFID.C'est donc encore un problème pour mécanisme de stockage d'état pour garantir l'intégrité du bit d'état des deux côtés. Plus tard, [57,58,60] ont présenté plusieurs approches pour des étiquettes RFID entièrement compromises dans LMAP, M2AP et EMAP.

En 2007, Chien [61] a développé le protocole SASI ayant le mérite de fournir une forte authentification et intégrité forte comme revendiqué. Malheureusement, [62] ont souligné que SASI est vulnérable à divers types d'attaques, à savoir l'attaque DoS2 (déni de service), l'attaque de désynchronisation, l'attaque de traçage de l'anonymat, l'attaque de relecture et l'attaque de divulgation complète.Après ce protocole, les chercheurs ont apporté plusieurs modifications et ajouts aux protocoles pour assurer une sécurité complète jusqu'à Jeon et Yoon [63] en 2013 ont développé un nouveau protocole RFID ultraléger nommé RAPLT (RFID Authentication Protocol for Low-cost Tags) basé sur deux nouvelles opérations, Fusion (Mer) et séparation (Sep). Cependant, Zhuang et al.[64]ont constaté que RAPLT n'est pas sécurisé en ce sens qu'il résiste aux attaques de désynchronisation et aux attaques par rejoue et qu'il protège l'intégrité des données et la confidentialité des utilisateurs.

Dans la suite de ce chapitre, nous expliquerons en détail les principaux protocoles que nous allons étudier.

3.3 Présentation de l’outil AVISPA et le modèle de l’attaquant

AVISPA (Automated Validation of Internet Security Protocols and Applications) a été développé en 2004 par Basin et Al dans le cadre d’un projet européen. C’est un outil d’analyse automatique destiné à aider à la validation de protocoles. Il possède deux enjeux majeurs : être performant, tout en garantissant l’accessibilité aux non-spécialistes du domaine [65].

AVISPA fournit un langage formel hautement expressif et standardisé pour modéliser les protocoles de sécurité et les spécifications de propriété appelé HLPSL (High Level Protocol Specification Language).

3.3.1 Architecture de l’outil AVISPA

Cet outil prend en entrée une spécification écrite sous format HLPSL, cette dernière est traduite en un format de plus-bas niveau appelé format intermédiaire (**Intermediate Format IF**) à l'aide d'un traducteur appelé **hlpsl2if Translator**. Le fichier résultant est ainsi directement interprété et la vérification de la spécification du protocole par les back-ends peut commencer [66].

AVISPA dispose de quatre différents back-ends comme illustré sur la **figure 11** et qui implémentent des techniques d'analyse allant de la falsification de protocole (découverte d'attaque sur le protocole d'entrée) à des méthodes de vérification à base d'abstractions pour des nombres fini ou infini de sessions. Si une propriété de sécurité est violée dans la spécification, l'analyseur produit la trace de séquence d'événement qui mène à une faille et affiche quelle propriété est violée [66].

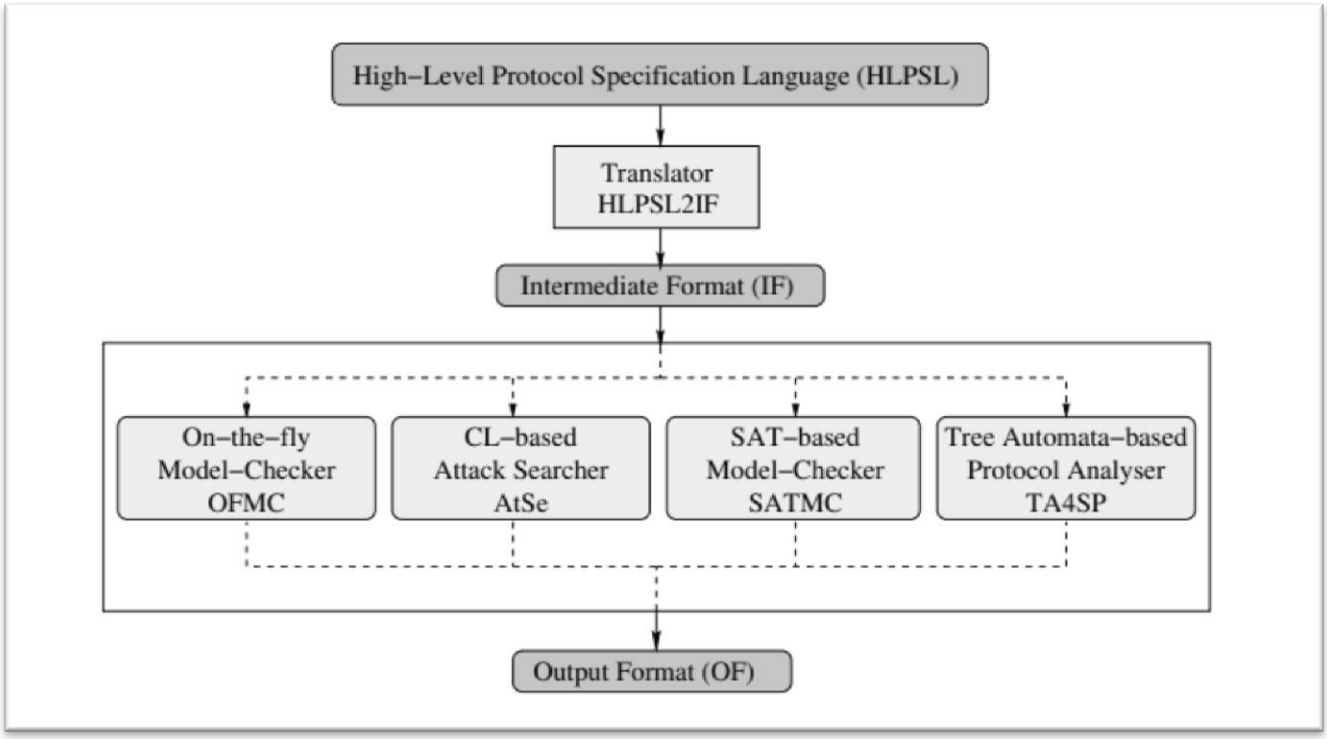

Figure 11 : Architecture de l'outil AVISPA.

➢ On-the-fly Model-Checker (OFMC)

Historiquement, l'OFMC a commencé au sein du projet AVISS, et puis OFMC a mûri au sein du projet AVISPA. Il effectue une vérification bornée en explorant le système de transition décrit par une spécification IF. OFMC implémente des techniques symboliques correctes et également complètes. Il supporte la spécification des opérateurs à propriétés algébriques tels que le OU exclusif ou encore l'Exponentielle [66].

➢ Constraint logic- based Attack Searcher (CL-AtSe)

C’est un outil basé sur des techniques de résolution de contraintes et implémente une procédure de décision. Il permet de faire une traduction d’une spécification d’un protocole de sécurité sous forme de relations de transition au format IF, vers un ensemble de contraintes qui peuvent être utilisées pour trouver des attaques sur le protocole en question. Il procède à une simplification du protocole d’origine en éliminant les états redondants par application de propriétés heuristiques. Sa modularité permet d’intégrer facilement d’autres spécifications de fonctions cryptographiques[65].Les deux méthodes (la traduction et la vérification sont totalement automatiques et prises en charge par l’outil CL-AtSe sans intervention d’outils externes [66].

- **SAT-based Model-Checker (SATMC)**

Cet outil a été développé au laboratoire DIST à Gênes (Italie) . Il construit une formule propositionnelle codant un déploiement borné du système de transition de IF ,l’état initial et l’ensemble des états représentant la violation des propriétés de sûreté spécifiées en IF (a fortiori en HLPSL). La formule propositionnelle est ensuite donnée à résoudre à un solveur SAT, sélectionné parmi ces quatre : zCHAFF, mCHAFF, SIM et SATO. Ensuite, tout modèle satisfaisant cette formule est retourné sous forme d’attaque[66].

- **Tree-Automata-based Protocol Analyzer (TA4SP)**

La particularité de cet outil de vérification est qu’à partir d’un état initial il fait soit une sur-approximation ou une sous-approximation des connaissances de l’intrus en utilisant des automates d’arbres. Cette méthode permet de savoir si un certain état est accessible ou non et que l’intrus peut savoir certaines connaissance ou non et ainsi de conclure l’absence d’attaque sur le secret pour des scénarios exécutés un nombre indéterminé de fois[65].

3.3.2 L’utilisation de l’outil AVISPA selon [65]

a. On commence par la spécification du protocole à tester grâce au langage HLPSL, ainsi que les propriétés à vérifier.

b. On lance AVISPA à l’aide d’une invite de commandes toute en précisant l’analyseur (back-end) qu’on va utiliser et par défaut AVISPA utilise OFMC back-end.

c. Ensuite AVISPA, après analyse, il déclare que soit le protocole est sûr (mais peut être sous certaines conditions), ou bien le protocole présente des failles et dans ce cas, on pourra changer la spécification du protocole et poursuivre depuis l’étape « a ».

3.3.3 Présentation de langage HLPSL

Les protocoles à tester par l'outil AVISPA sont écrits dans le langage HLPSL qui est inspiré de TLA (Time Procedural Logic). Il permet également la représentationLe protocole de sécurité par les états/transitions du système que la vérification mettra en œuvre des propriétés de sécurité exprimées dans une logique temporelle linéaire.

Toutes les entités intervenant dans le protocole en question sont représentées dans la spécification HLPSL. Cette description est contenue dans un fichier avec l'extension « .hlpsl ». Il sera traduit afin de créer autre fichier dont l'extension « .if » à l'aide du traducteur hlpsl2if. Ensuite, différents back parties utilisent ce format pour vérifier le protocole donné.

Le langage HLPSL est basé sur la définition des rôles et une liste des propriétés de sûreté à vérifier ;

Les rôles

Un rôle peut être considéré comme un processus indépendant peut être paramétré en prenant un ou plusieurs arguments et peut également déclarer des variables locales. On distingue deux types de rôles différents:

- **Rôles basiques** : il représente les connaissances et le comportement initial de chaque participant (agent) est impliqué dans le protocole spécifié.
- **Rôles de composition :** qui représentent les scénarios des rôles basiques et permet de combiner les autres rôles soit en parallèle ou en séquentiel, et définir ainsi le protocole en tant que session.

Déclaration des propriétés de sûreté

Les propriétés de sûreté du protocole à évaluer par AVSPA telles que le secret, l'authentification sont déclarées dans une section à part et afin de les vérifier, il est nécessaire d'introduire dans la spécification des éléments événementiels appelés signaux. En HLPSL, il existe différents signaux permettant d'exprimer les propriétés de sûreté [65];

- **secret(E, id, S)** : déclare que l'information E est un secret partagé par un ensemble S d'agent.
- **witness(A, B, id, E)** : pour la propriété d'authentification (faible) de l'agent A auprès de B grâce à la donnée E. Cet objectif sera identifié par la constante id dans la section réservée à la déclaration des propriétés de sécurité.
- **request (B, A, id, E)** : pour l'authentification forte entre A et B. Elle déclare que B demande une vérification de la valeur E. La fonction de id est la même que pour witness.
- **wrequest(B, A, id, E)** : elle est similaire à request(), mais cette fois pour l'authentification faible.

La déclaration des objectifs ou des propriétés à vérifier se fait dans une section à part. Chaque propriété est identifiée par une constante qui réfère le prédicat défini (secret, witness, request et wrequest) pour une transition donnée [66].

3.4 Le modèle Dolev & Yao:

Dans le cadre de la modélisation de protocoles de sécurité, il est nécessaire de modéliser également l'intrus, c'est-à-dire de définir son comportement et de le limiter. Pour cela, les hypothèses utilisées sont rassemblées sous le nom de « modèle de Dolev-Yao ». Ce modèle est basé sur deux hypothèses importantes qui sont: le chiffrement parfait et l'intrus est le réseau. Le chiffrement parfait assure en particulier qu'un intrus ne peut déchiffrer un message m chiffré avec une clé k que s'il possède l'inverse de cette clé. La seconde hypothèse "l'intrus est le réseau" signifie que l'intrus peut intercepter et remplacer les messages envoyés par les acteurs honnêtes du protocole, et leur envoyer des messages sous une fausse identité [67].

3.5 Le protocole d'authentification FDW

Feldhofer, Dominikus et Wolkerstorfer [FDW04] ont montré qu'il est possible d'implémenter le chiffrement symétrique AES sur une étiquette RFID. Dans leur article, ils discutent deux protocoles simples d'authentification unilatérale et mutuelle [68]. Dans ce qui suit, on vérifie ce dernier protocole.

Symbole	Signification
R, T	Nom d'agent honnête (un participant honnête du protocole), R : Lecteur ; T : Tag
m	Message
Nt, Nr	Nonce (nombre aléatoire « frais »)
h	fonction de hachage
,	Concaténation
K	Clé symétrique partagée entre R et T
ID	Identificateur partagé entre le tag et le lecteur
{m}_K	Le message m crypté avec K
R Æ T : m	R envoie un message m à T

Tableau 3 : Notation utilisé dans le protocole FDW.

3.5.1 Description

Dans ce protocole, chaque couple de lecteur R et tag T possède une clé unique et partagée K. Il y a trois étapes de FDW :

Etape 01 : Le lecteur lance le protocole par l'envoi d'un nonce Nr frais au tag.

Etape 02 : Le tag génère un nombre nonce Nt et crypte la paire (Nt, Nr) avec la clé partagée K, et l'envoie au lecteur.

Etape 03 : Le lecteur déchiffre le message en utilisant la même clé partagée et inverse l'ordre des deux nonces, crypte le message avec la même clé partagée et l'envoie au tag.

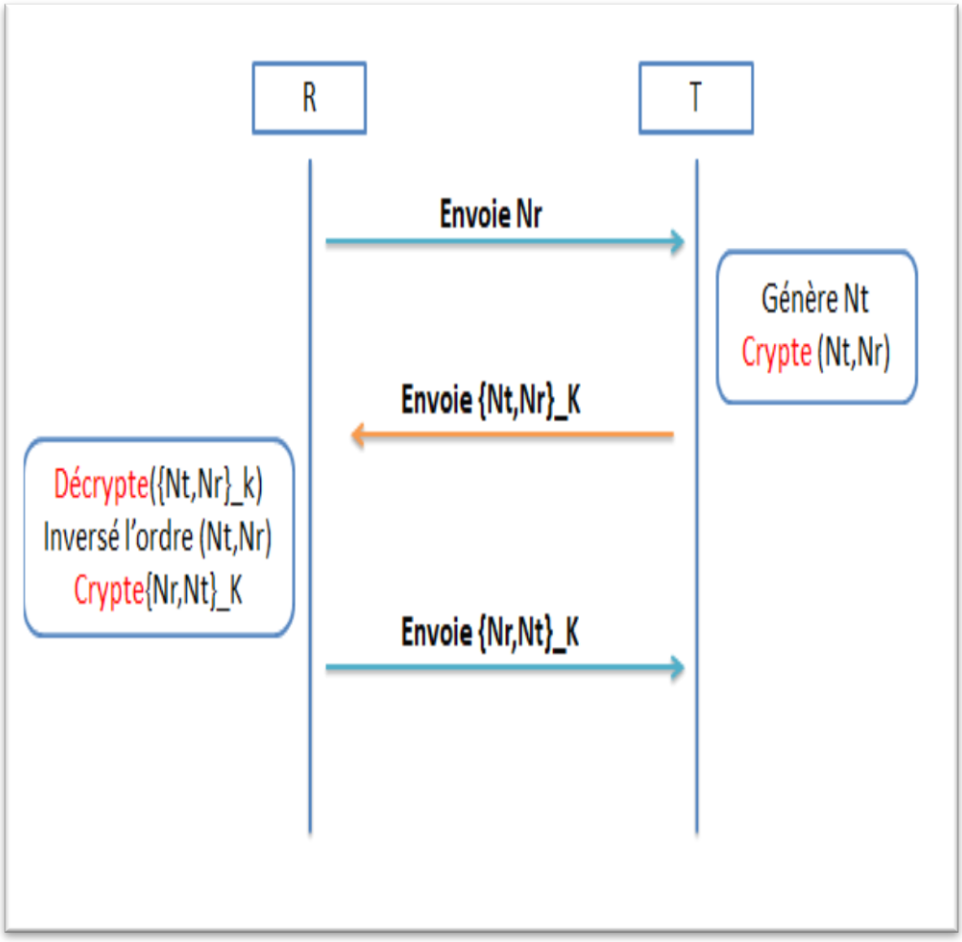

Figure 12 : La phase d'authentification dans le protocole FDW.

Nous pouvons aussi le présenter par la notation Alice-Bob suivante:

- R → T : Nr
- T → R : {Nt, Nr} _K
- R → T : {Nr, Nt} _K

3.5.2 Code HLPSL et explication

Le code HLPSL de ce protocole est présenté dans la **figure 13**. Dans cette partie, nous expliquons les différents rôles de ce script.

Rôles Basiques

```
role reader ( R,T: agent, K: symmetric_key, SND,REC: channel(dy))
  played_by R  def=  local  State : nat, Nr, Nt : text
    const sec_N1 : protocol_id
    init State := 0
    transition
       1. State = 0  /\ REC(start)   =|>  State' := 1  /\ Nr' := new()
       /\ SND(Nr')  /\ witness(R,T,aut_reader,Nr')
       2. State = 1  /\ REC({Nt'.Nr'}_K)   =|>   State' := 2
       /\  SND({Nr.Nt'}_K)  /\ secret(Nt',sec_N1,{R,T})
       /\ request(R,T,aut_tag,Nt')
    end role
role tag ( T,R: agent,K: symmetric_key, SND,REC: channel(dy))
   played_by T     def=
      local  State  : nat, Nt,Nr  : text
       const sec_N2 : protocol_id
      init State := 0
      transition
       1. State = 0  /\ REC(Nr')   =|>   State' := 1  /\ Nt' := new()
       /\ SND({Nt'.Nr'}_K)
       /\ secret(Nt',sec_N2,{T,R})     /\ witness(T,R,aut_tag,Nt')
       2. State = 1 /\ REC({Nr.Nt}_K)  =|>  State' := 2
       /\ request(T,R,aut_reader,Nr)
    end role
```

Rôles Composants

```
role session(T,R : agent,K : symmetric_key) def=
  local St,Rt,Sr,Rr : channel(dy)
  composition
  tag(T,R,K,St,Rt) /\ reader(R,T,K,Sr,Rr)
end role
role environment() def=
   const t1,t2,r : agent, k1,k2 : symmetric_key,
   aut_tag, aut_reader :protocol_id
   intruder_knowledge = {t1,t2,r}
   composition
   session(t1,r,k1) /\ session(t2,r,k2)
end role
```

Propriétés

```
goal
secrecy_of sec_N2, sec_N1
authentication_on aut_reader % authentification du lecteur par Nr
authentication_on aut_tag  % authentification du tag par Nt
end goal
  environment()
```

Figure 13 : la spécification du protocole FDW en HLPSL.

Le rôle **environment** s'intéresse au scénario de vérification suivant : deux sessions du protocole en parallèle noter par le symbole /\ concernant un tag légitime et un même lecteur légitime. Les connaissances initiales de l'intrus sont les noms d'agents t qui représente le tag, et l'agent r qui représente le lecteur. Cette spécification permet de détecter les attaques par relais s'ils existent. Les données aut_reader et aut_tag sont des constantes qui permettent d'identifier les propriétés d'authentification du lecteur et d'authentification du tag respectivement.

La section goal nous permet de spécifier les objectifs de sécurité permettant aux outils AVISPA de faire une recherche sur les attaques.

3.5.3 Test avec avispa

Après la vérification de ce protocole par les outils AVISPA, Le résultat est montré sur la **figure 16.**

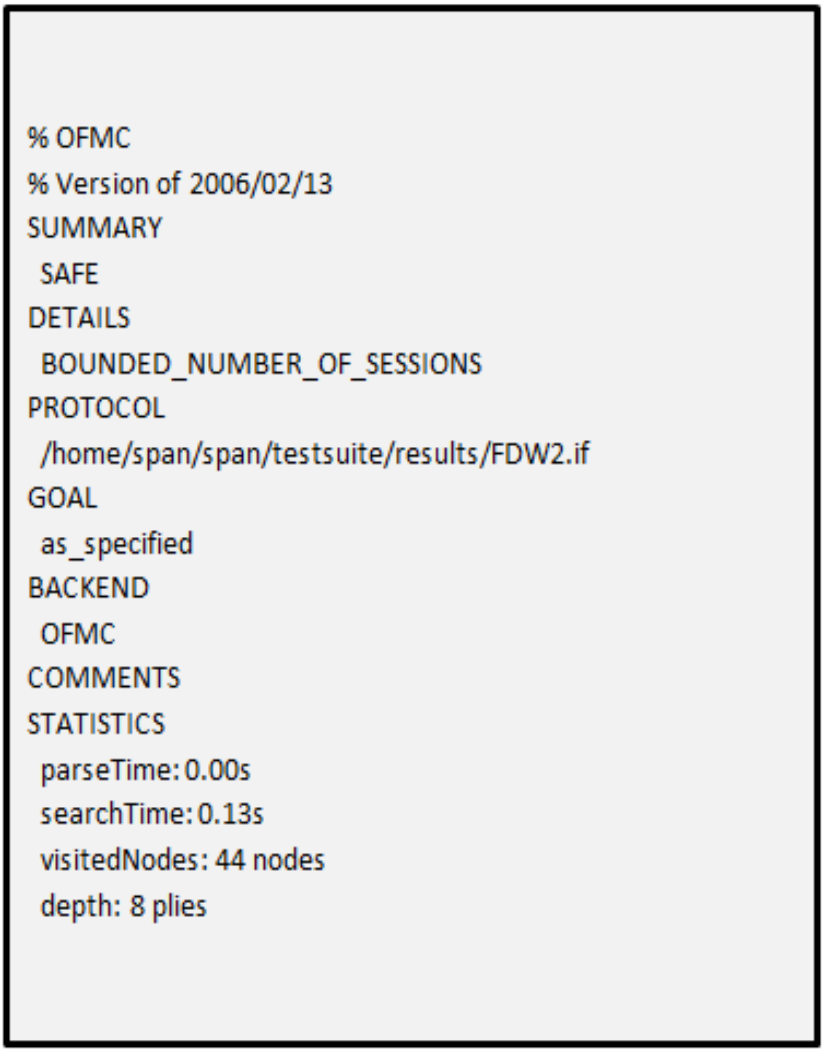

Figure 14 : Le résultat de la vérification de protocole FDW par AVISPA.

Ce résultat signifie clairement qu'il n'y a pas d'attaque pour la confidentialité du nombre Nt (vérifiée par sec_N1 et sec_N2) détectée, ni pour l'authentification d'étiquette ni pour l'authentification de lecteur. Ainsi on peut conclure quele diagnostic de la plateforme AVISPA pour ce protocole est sécurisé.

3.6 Le protocole d'authentification Wei et al

Ce Protocole d'authentification a été proposé par Wei et al . Il utilisait les primitives : fonction de hachage, nonce (number used once) et opérateur xor.

Symbole	Signification
R, T	Nom d'agent honnête (un participant honnête du protocole), R : Lecteur ; T : Tag
m	Message
Nt, Nr	Nonce (nombre aléatoire « frais »)
h	fonction de hachage
,	Concaténation
ID	Identificateur partagé entre le tag et le lecteur

Tableau 4 : Notation utilisé dans le protocole Wei et al.

3.6.1 Description

Dans ce protocole, le lecteur R et le tag T partagent les valeurs secrètes s et ID d'identificateur selon les étapes suivantes ;

Étape 01 : Le lecteur génère un nombre aléatoire NR et interroge les tags avec NR.

Étape 02 : Après avoir reçu NR, le tag génère un nombre aléatoire NT et calcule h (s⊕ NR⊕NT), puis renvoie NT et h(s⊕NR⊕NT) au lecteur.

Étape 03 : Après avoir reçu NT et h(s⊕ NR⊕NT) du tag, le lecteur calcule h(RID⊕ NR),et envoie NR, h(s⊕NR⊕NT), NT, h(RID⊕ NR) au serveur.

Étape 04 : Après avoir reçu un message d'authentification du lecteur, le serveur vérifie si NR correspond à NR(old), s'ils correspondent, l'authentification est réussie. Sinon, l'authentification a échoué.

Étape 05 : Le serveur vérifie s'il existe certains RID* dans la table RID de la base de données, ce qui pourrait donner h(RID*⊕ NR) = h(RID⊕ NR). S'il existe un tel enregistrement, l'application d'authentification serait considérée comme provenant d'un lecteur légitime, sinon l'authentification échoue.

Étape 06 : Par la suite, le serveur vérifie s'il existe un certain s* dans l'ID de table dela base de données, ce qui pourrait faire h(s*⊕NR⊕NT) = h(s⊕ NR⊕NT). S'il existe un telrecord, le tag serait considéré comme un tag légitime, alors le serveur génère unnombre aléatoire Ndb et calcule h (id⊕Ndb), puis envoie Ndb , h(id⊕Ndb) aulecteur, le serveur doit ensuite mettre à jour NR(old), NR(new), s(old)et s(new).

Étape 07 : Après avoir reçu Ndb, h(id⊕NR) du serveur, le lecteur enverrait Ndb,h(id⊕Ndb)à le tag.

Étape 08 : Après avoir reçu Ndb, h(id⊕Ndb) du lecteur, le tag calculerait h(id⊕Ndb), Si le résultat est égal au h(id⊕Ndb) reçu, alors l'objet de l'authentification mutuelle est atteint, le tag doit mettre à jour s = h(id⊕Ndb⊕ NT), sinon, l'authentification est échoué .

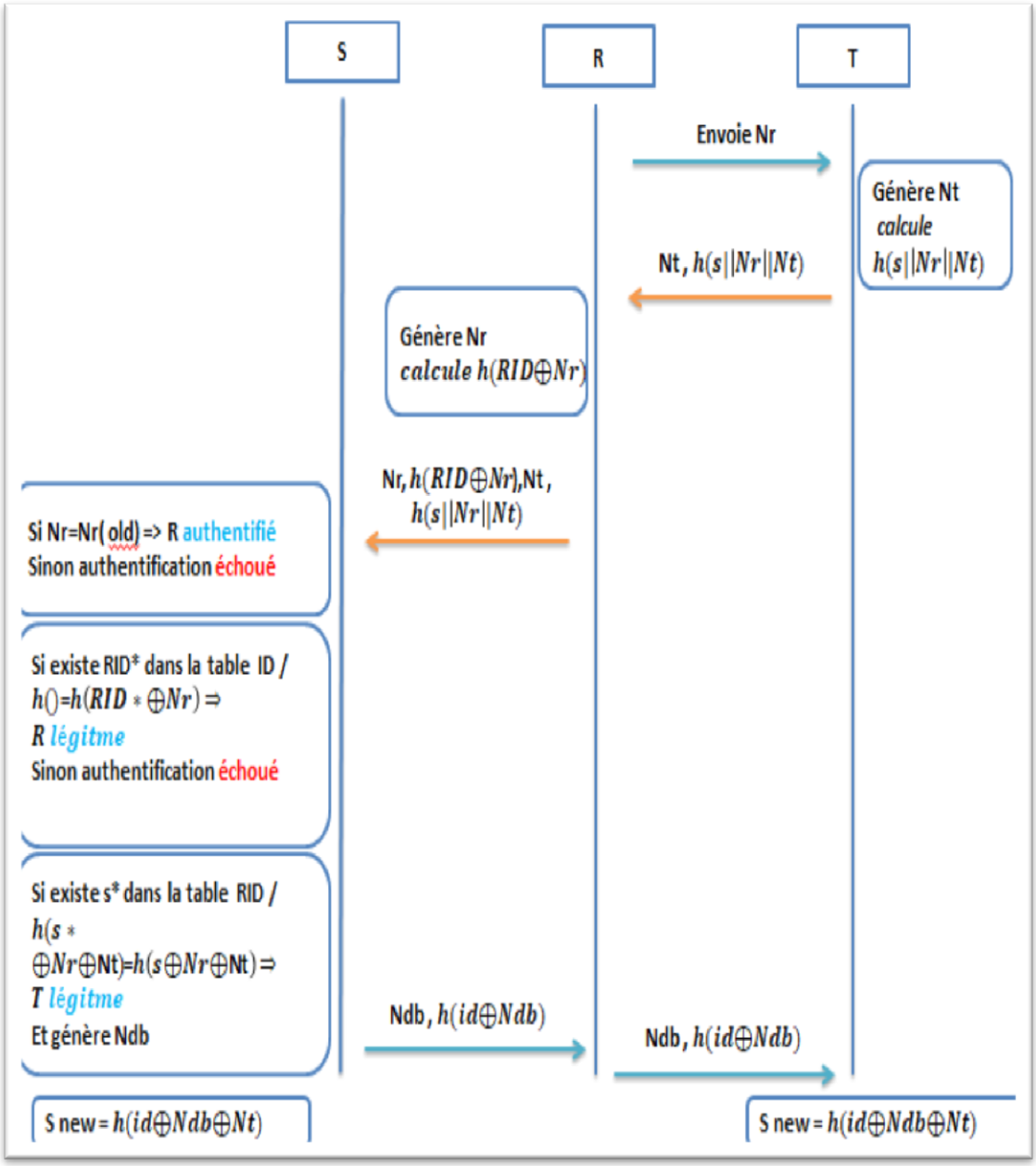

Figure 15 : La phase d'authentification dans le protocole Wei et al.

3.6.2 Code HLPSL et explication

Comme nous l'avons dit dans la section 3.6 ce protocole utilise les primitives : fonction de hachage, nonce et opérateur xor. Ces primitives sont prises en charge dans HLPSL.

```
role reader ( R,T: agent, ID,RID, S: text, H : hash_func, Snd,Rec:
channel(dy))
 played_by R
 def=
 local State : nat,
 Nr, Nt, Ndb : text
 init State := 0
 transition
 1. State = 0 /\ Rec(start) =|> State' := 1 /\ Nr' := new() /\
Snd(Nr')
 2. State = 1 /\ Rec(H(xor(xor(S,Nr),Nt')).Nt')
 =|> State' := 2 /\ Ndb' := new() /\ Snd(H(xor(ID,Ndb')).Ndb') /\
secret(ID,sec_id,{R,T})
 /\ request(R,T,aut_tag,Nt') /\ witness(R,T,aut_reader,Ndb')
end role
role tag ( T,R: agent, ID,RID,S: text, H : hash_func,Snd,Rec:
channel(dy))
 played_by T
 def=
 local State : nat,
 Nt, Nr,Ndb : text
 %const sec_k2 : protocol_id
 init State := 0
 transition
 1. State = 0 /\ Rec(Nr') =|> State' := 1 /\ Nt' := new()
 /\ Snd(H(xor(xor(S,Nr'),Nt')).Nt') /\ witness(T,R,aut_tag,Nt')

 2. State = 1 /\ Rec(H(xor(ID,Ndb')).Ndb')
 =|> State' := 2 /\ request(T,R,aut_reader,Ndb')
end role
```

Figure 16 : la spécification de protocole Wei et al.en HLPSL.

Pour les deux roles **reader** et **tag** est dit lecteur, avec les paramètres R et T de type agent, id et RID de type texte et H de type fonction de hachage. Les paramètres Rec et Snd sont de type canal, indiquant qu'il s'agit de canaux sur lesquels l'agent joue le rôle de lecteur qui communiquera. L'attribut du type de canal, dans ce cas (dy), indique le modèle d'intrus à prendre en compte pour ce canal.

Le paramètre R apparaît dans la section played_by, ce qui signifie, intuitivement, que R désigne le nom de l'agent qui joue le rôle de lecteur. Notez également la section locale qui déclare les variables locales du lecteur : State qui est un nat (un nombre naturel) et nombres aléatoires de type texte, Nr, Nt et Ndb . La variable locale State est initialisée à 0 dans la section d'initialisation.

Concernant la partie transition, la première transition du rôle lecteur signifie : si la valeur de State est 0 et le message dans le canal REC est start alors : Nr prend un nouveau random valeur envoyée sur le canal SND. Le but (goal) witness(R,T, aut_server,Nr') doit être lu "agent R affirme que l'on veut être le pair de l'agent T, en s'accordant sur la valeur Nr' dans un effort d'authentification identifié par l'identifiant de protocole aut_server."

Pour la deuxième transition, si la valeur de State est 1 et le message H(xor (xor (Nr,Nt'),S)).Nt' sur la canal REC alors la variable State est mise à 2, et le lecteur envoie le message H(xor(id,Ndb').Ndb' sur le canal SND. Pour le prédicat secret cela signifie "la nouvelle valeur stockée dans S est un secret à partager uniquement entre les agents R et T". Le requête de prédicat (R,T,aut_tag,Nt') doit être lue, « l'agent R accepte la valeur Nt' et maintenant s'appuie sur la garantie que l'agent T existe et s'accorde avec lui sur cette valeur ».

```
role session(R,T : agent,ID,RID,S : text, H: hash_func)
def=
 local Sa,Ra,Sb,Rb : channel(dy)
 composition
 reader(R,T,ID,RID,S,H,Sa,Ra) /\ tag(T,R,ID,RID,S,H,Sb,Rb)
end role
role environment() def=
const r,t : agent,
 id,rid,s,id1,s1: text,
h: hash_func,
 aut_reader, aut_tag, sec_id : protocol_id
 intruder_knowledge = {r,t,h}
 composition
 session(r,t,id,rid,s,h)
 /\ session(r,t,id1,s1,h)
end role
goal
 secrecy_of sec_id
authentication_on aut_tag
authentication_on aut_reader
end goal
environment()
```

Figure 17 : la spécification de protocole Wei et al.en HLPSL.

Dans le role session, on déclare généralement tous les canaux utilisés par les rôles de base. Le type de canal prend un attribut supplémentaire (qui spécifie l'intrus modèle un suppose pour ce canal). Ici, le canal de déclaration de type (dy) représente le modèle d'intrus de « Dolev et Yao ». Ainsi, le lecteur et l'étiquette peuvent envoyer et recevoir sur quel que soit le canal qu'ils veulent; lorsque l'intrus est le réseau, la connexion prévue entre certaines variables de canal n'est pas pertinente. Dans notre spécification, le lecteur envoie sur Sa des messages à tag qui les reçoit sur Rb.

3.6.3 Test avec AVISPA

Après la vérification de ce protocole par les outils AVISPA détectent une trace d'attaque sur l'authentification des tags .les résultat est montré sur la figure 18.

Dans ce résultat de trace, i représente le l'intrus, (r, 3) le lecteur et (t,4) le tag.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/span/span/testsuite/results/WA.if
GOAL
  replay_protection_on_aut_reader
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.20s
  visitedNodes: 60 nodes
  depth: 5 plies
ATTACK TRACE
i -> (r,3): start
(r,3) -> i: Nr(1)
i -> (t,3): Nr(1)
(t,3) -> i: h(s XOR Nr(1) XOR Nt(2)).Nt(2)
i -> (r,3): h(s XOR Nr(1) XOR Nt(2)).Nt(2)
(r,3) -> i: h(id XOR Ndb(3)).Ndb(3)
i -> (t,3): h(id XOR Ndb(3)).Ndb(3)
i -> (t,6): x277
(t,6) -> i: h(s XOR x277 XOR Nt(5)).Nt(5)
i -> (t,6): h(id XOR Ndb(3)).Ndb(3)
```

Figure 18 : Le résultat de la vérification de protocole Wei et al par AVISPA.

Msg1 : Le lecteur génère un nonce NR et l'intrus capture et stocke le nonce au cours de la communication.

Msg2 : L'intrus génère un autre nonce NR' et l'envoie au tag.

Msg3 : Le tag génère une instance du nonce NT et l'envoie avec le hachage fonction h (NR'⊕ NT ⊕s) à l'intrus.

Msg4 : L'intrus renvoie la fonction reçue au lecteur avec NR'⊕ NR ⊕ NT.

Msg5 : Le lecteur envoie le message h(id⊕Ndb), Ndb au tag. Ce message fait ne dépend pas de l'attaque découverte.

Donc l'attaque sur l'authentification des tags est réalisée dans Msg4.

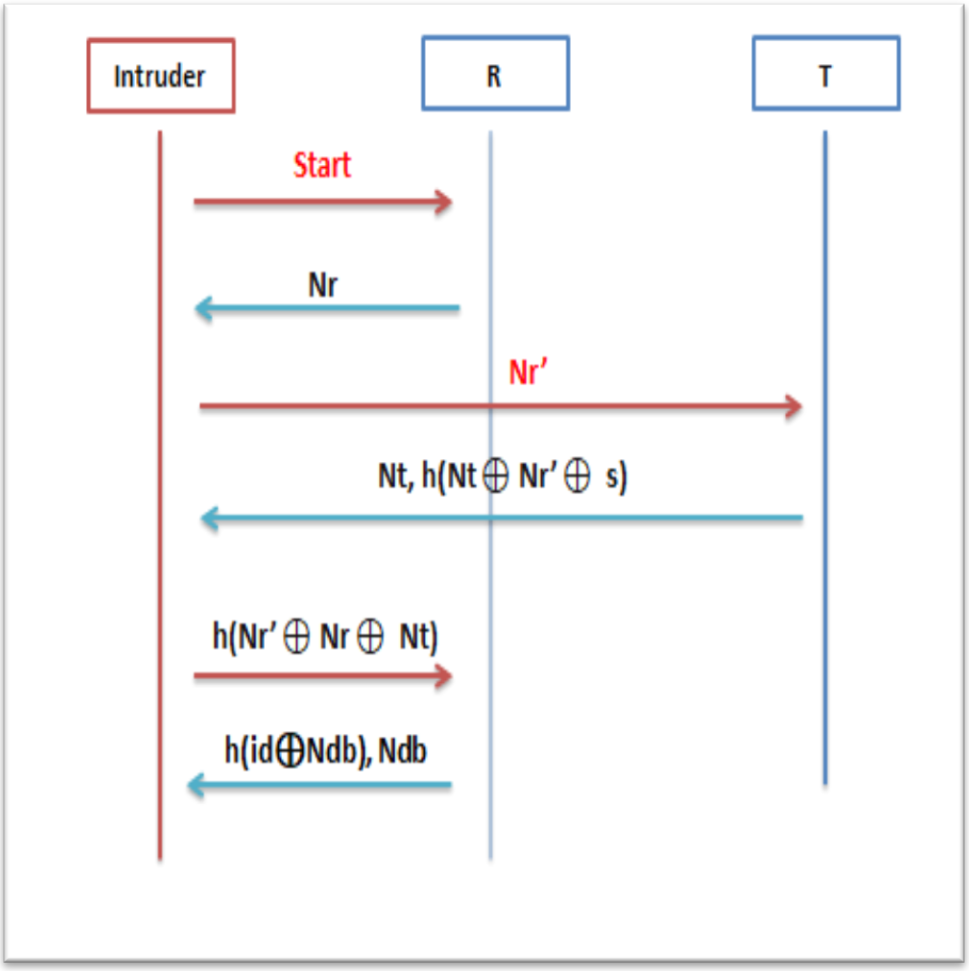

Figure 19 : Attaque de trace sur le protocole Wei et al (WHC).

3.6.4 Solution amélioré

Concernant le protocole de Wei et al. , la solution proposée par les auteurs [65] est de changer la primitive XOR (⊕) entre le nonce NR et NT par la concaténation (‖).Par conséquent, la nouvelle fonction de hachage est h(NR‖(NT⊕s)). Après correction la spécification en HLPSL présenté dans la figure 20.

```
role reader ( R,T: agent, ID,RID, S: text, H : hash_func, Snd,Rec:
channel(dy))
 played_by R
 def=
 local State : nat,
 Nr, Nt, Ndb : text
 init State := 0
 transition
 1. State = 0 /\ Rec(start) =|> State' := 1 /\ Nr' := new() /\
Snd(Nr')
 2. State = 1 /\ Rec(H(xor(S,Nr),Nt').Nt')
 =|> State' := 2 /\ Ndb' := new() /\ Snd(H(xor(ID,Ndb')).Ndb') /\
secret(ID,sec_id,{R,T})
/\ request(R,T,aut_tag,Nt') /\ witness(R,T,aut_reader,Ndb')
end role
role tag ( T,R: agent, ID,RID,S: text, H : hash_func,Snd,Rec:
channel(dy))
 played_by T
 def=
 local State : nat,
 Nt, Nr,Ndb : text
 %const sec_k2 : protocol_id
 init State := 0
transition
 1. State = 0 /\ Rec(Nr') =|> State' := 1 /\ Nt' := new()
 /\ Snd(H(xor(S,Nr'),Nt'),Nt') /\ witness(T,R,aut_tag,Nt')
 2. State = 1 /\ Rec(H(xor(ID,Ndb')).Ndb')
 =|> State' := 2 /\ request(T,R,aut_reader,Ndb')
end role
role session(R,T : agent,ID,RID,S : text, H: hash_func)
def=
local Sa,Ra,Sb,Rb : channel(dy)
 composition
 reader(R,T,ID,RID,S,H,Sa,Ra) /\ tag(T,R,ID,RID,S,H,Sb,Rb)
end role
```

Figure 20 : la spécification de la solution amélioré de protocole Wei et al. en HLPSL.

La vérification automatisée du protocole de Wei et al. donne le résultat suivant :

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/waprop.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.06s
  visitedNodes: 20 nodes
  depth: 4 plies
```

Figure 21 : Le résultat de la vérification de protocole «Wei al. Amélioré » par AVISPA.

Ce résultat a clairement montré qu'aucune attaque n'a été détectée lors de l'authentification. Nous pouvons donc conclure que ce protocole est sécurisé.

3.7 Protocole d'authentification RFID basé sur le hachage

Ce Protocole basé sur les fonctions Hachages unidirectionnels sans collisions. Il repose sur deux phases : une phase de configuration préalable qui se déroule «hors ligne» et une phase d'identification intervenant «en ligne».

Symbole	Signification
S, T	Nom d'agent honnête (un participant honnête du protocole), S : Serveur ; T : Tag
Msg_i	Message
r_s ,r_t	Nonce (nombre aléatoire « frais »)
H1,H2,H3	fonctions de hachage
,	Concaténation
K	Clé symétrique partagée entre S et T
$ID_{i,jx}$	Identifiant secret de l'étiquette Tagi numéro jx,ID_i si Tagi possède juste un seul identifiant
K_{Gi}	Clé secrète du groupe i
E_Y	Fonction de chiffrement en utilisant la clé y

Tableau 5 : Notation utilisé dans la spécification du protocole d'authentification basé sur le hachage.

3.7.1 Description

Dans ce protocole**,** le serveur et les étiquettes dans le protocole stockent trois fonctions de hachage H1, H2 et H3. Comme nous le verrons par la suite, ces fonctions de hachages permettre à la fois de renforcer la sécurité de protocole mais également de réduire la complexité du temps de recherche coté serveur.

Les étapes suivantes expliquent l'interaction entre le serveur (S) et une étiquette (Tag_i.) au cours de la phase d'identification ;

Étape 01 : Le serveur S génère un nombre aléatoire r_s et le transmet àTag_i..

Étape 02 : En réponse, Tag_i génère d'abord un nombre aléatoire rt et choisit un identifiant ID_{i,j_r}dans $\Omega_{i,j}$.

Puis, il calcule $Msg_i = E_{k\,G_i}$ ($r_s \| r_t \|$H1 (K_{Tagi}) $\|$ ID_{i,j_x}).

Tag_i envoie le message Msg_i = à S.

Étape 03 : Après la réception de la réponse, le serveur déchiffre Msg_i en essayant toutes les clés de groupe dans le système pour retrouverID_{i,j_x} .

Ensuite, S utilise H1 (K_{Tagi}) pour retrouver la correspondance K'_{Tagi}dans la base de données et vérifie si H1 (K'_{Tagi}) ? = H1 (K_{Tagi}). Si une correspondance est trouvée, S accepteTag_i. Sinon S rejette Tag_i.

Le serveur calcule K^{+1}_{Tagi}= H2 (K_{Tagi}) et envoie H3 ($K^{+1}_{Tagi} \| r_t$) àTag_i.

Étape 04 :Tag_i reçoit H3 ($K^{+1}_{Tagi} \| r_t$) et vérifie si H3 ($K^{+1}_{Tagi} \| r_t$) ? = H3 (H2 (K_{Tagi})$\| r_t$) .Si une correspondance est trouvée, le serveur est authentifié et Tag_i met à jour sa clé suivant K^{+1}_{Tagi} = H2 (K_{Tagi}).

Pour éviter les attaques par désynchronisation le serveur enregistre la clé K^{-1}_{Tagi} deTag_i.

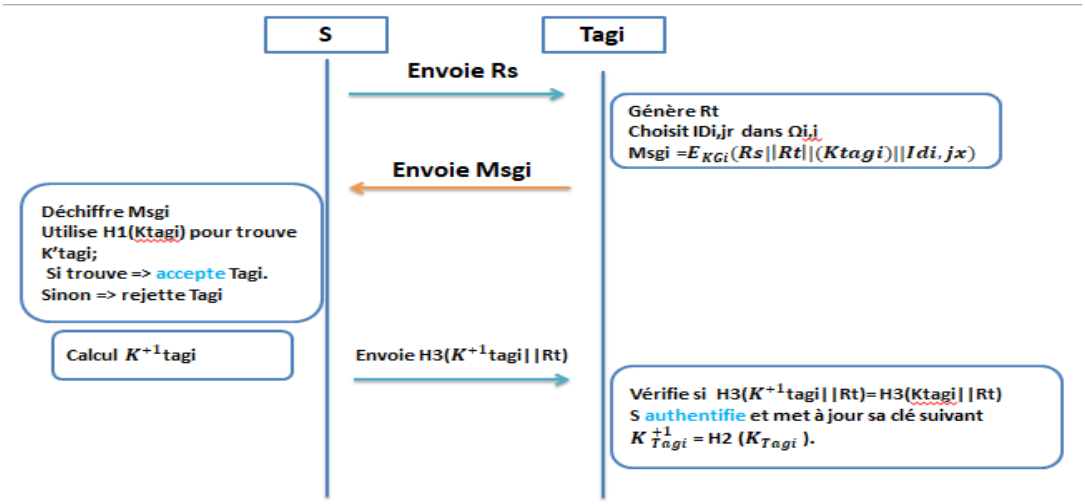

Figure 22 : La phase d'authentification dans le protocole d'authentification basé sur le hachage.

3.7.2 Code HLPSL et explication

Pour la spécification en HLPSL présenté dans les **Figures 23** ,**24** et **25** dans cette partie, on explique le principe des rôles basiques. Il existe deux rôles de base « role servers » et « role TagT » qui expliquent l'activité du serveur.

Il existe deux agents T, S qui utilisent trois fonctions de hachage H1, H2, et H3 et une opération de concaténation CONCAT.

```
role serverS (S , T : agent ,
H1 , H2 , H3 , CONCAT :hash_func ,
SND , RCV : channel ( dy ) )
played_by S
def=
local State: nat ,
Idti : text ,
Kti , KGI : symmetric_key ,
Rs , Rt , MSG, MSGs: message
const sec_kti , sec_kgi , sec_idt , auth_kti : protocol_id
init State  :=0
transition
1. State = 0 /\ RCV (start)
=|> State':=2 /\ Rs':= new ()
/\ SND ( Rs')
2. State  = 2
/\ RCV ( MSG')
/\MSG'= { CONCAT ( Rs , Rt' , H1 ( Kti ) , Idti )} _KGI
=|> State':=4
/\ request (S , T , auth_kti , Kti )
/\ Kti':= H2 ( Kti )
/\ MSGs':= H3 ( Kti' , Rt )
/\ SND ( MSGs')
/\ secret ( Idti , sec_idt , {T , S })
/\ secret ( Kti , sec_kti , {T , S })
/\ secret ( KGI , sec_kgi , {T , S })
/\ witness (S , T , auth_kti , Kti')
end role
```

Figure 23 : la spécification de protocole RFID basé Sur le hachage en HLPSL.

```
role tagT ( T , S : agent ,
H1 , H2 , H3 , CONCAT : hash_func,
SND , RCV : channel ( dy ) )
played_by T
def=
local State: nat,
Idti : text ,
KGI,Kti: symmetric_key ,
Rs , Rt , MSG , MSGs: message
const sec_kti , sec_kgi , sec_idt , auth_kti : protocol_id
init State :=1
transition
1.State = 1 /\ RCV ( Rs')
=|> State':=3
/\ Rt':= new()
/\Idti':=new()
/\ MSG':= { CONCAT ( Rs', Rt', H1 ( Kti ) , Idti')} _KGI
/\ SND ( MSG')
/\ secret ( Idti', sec_idt , {T , S })
/\ secret( Kti , sec_kti , {T , S })
/\ secret ( KGI , sec_kgi , {T , S })
/\ witness (T , S , auth_kti , Kti )
2. State = 3 /\ RCV ( MSGs')/\ MSGs'= H3 ( H2 ( Kti') , Rt )
=|> State':=5
/\ Kti':= H2 ( Kti')
/\ request (T , S , auth_kti , Kti')
end role
role session ( S , T:agent ,
H1 , H2 , H3 , CONCAT : hash_func)
def=
local ST , RT , SS , RSS : channel ( dy )
composition
serverS ( S , T , H1 , H2 , H3 , CONCAT , SS , RSS )
/\tagT ( T , S , H1 , H2 , H3 , CONCAT , ST , RT )
end role
```

Figure 24 : la spécification de protocole RFID basé Sur le hachage en HLPSL.

Le secret de la clé secrète de l'étiquette T, la clé secrète du groupe et son identité Kti, Kgi, Idti sont modélisées à l'aide des prédicats secret(Kti, sec_kti, {T, S}), secret (KGI, sec_kgi, {T, S}), et secret (Idti, sec_idt, {T, S}),) qui sont gérés par le protocol_id : sec_kTi, sec_kgi et sec_idt respectivement. Les paramètres Idti et (Rs, Rt) sont gardés secrets pour T et S.

L'authentification mutuelle est réalisée via les objectifs witness et request, c'est-à-dire witness(T, S, auth_kti, Kti), request(S, T, auth_kti, Kti), witness(S, T, auth_kti, Kti'), request (T, S, auth_kti, Kti').

witness(T, S, auth_kti, Kti')déclare que l'agent T prétend être l'homologue de l'agent S, et qu'ils sont en accord sur la valeur Kti'. auth_Kti représente l'authentification de Kti' indiquée dans la section goal alors que request(S, T, auth_kti, Kti') déclare que l'agent S accepte la valeur Kti' et s'appuie désormais sur la garantie que l'agent T existe et l'accepte pour cette valeur. S et de l'étiquette Tag_i

```
role environment ()
def=
const
a , b , i : agent ,
h1 , h2 , h3 , concat : hash_func,
sec_kti , sec_kgi , sec_idt , auth_kti : protocol_id
intruder_knowledge = { a , b , i , h1 , h2 , h3 , concat }
composition
session (a , b , h1 , h2 , h3 , concat )
/\ session (a , i , h1 , h2 , h3 , concat )
/\ session (i , b , h1 , h2 , h3 , concat )
end role
goal
secrecy_of sec_kti , sec_kgi , sec_idt
authentication_on auth_kti
end goal
environment ()
```

Figure 25 : la spécification de protocole RFID basé Sur le hachage en HLPSL.

3.7.3 Test avec AVISPA

Après la vérification de ce protocole par les outils AVISPA, Le résultat est montré sur la **figure 26**.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/RFID.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.04s
  visitedNodes: 24 nodes
  depth: 4 plies
```

Figure 26 : Le résultat de la vérification de protocole RFID basé Sur le hachage par AVISPA.

Ce résultat signifie en clair qu'il n y a pas d'attaque par rejeu, ce qui est rejoint ce qui a été mentionné dans [71].

3.8 Les solutions proposées

Selon **[69]**, plusieurs attaques sont possibles si on prend en compte les propriétés algébriques de l'opérateur XOR dans les primitives du chiffrement. Dans ce sens, Lawrence C. Paulson a démontré que le protocole Bull **[70]** est correct pour la propriété de confidentialité sous l'hypothèse de l'encryptions parfaite mais sa correction s'écroule si on prend en compte les propriétés algébriques de l'opérateur XOR.

L'idée de notre proposition concernant le protocole de Wei et al. et leur version amélioré est de changer tout primitive XOR ($\oplus$) par une concaténation ($\|$).Par

conséquent, les nouvelles fonctions de hachage sont h(NR||(NT||s)) et h(ID|| Ndb'). Après correction la spécification en HLPSL présenté dans la **figure 27.**

```
role reader ( R,T: agent, ID,RID, S: text, H : hash_func, Snd,Rec:
channel(dy))
 played_by R
 def=
 local State : nat,
 Nr, Nt, Ndb : text
 init State := 0
 transition
 1. State = 0 /\ Rec(start) =|> State' := 1 /\ Nr' := new() /\
Snd(Nr')
 2. State = 1 /\ Rec(H(S,Nr,Nt').Nt')
 =|> State' := 2 /\ Ndb' := new() /\ Snd(H(ID,Ndb').Ndb') /\
secret(ID,sec_id,{R,T})
/\ request(R,T,aut_tag,Nt') /\ witness(R,T,aut_reader,Ndb')
end role
role tag ( T,R: agent, ID,RID,S: text, H : hash_func,Snd,Rec:
channel(dy))
 played_by T
 def=
 local State : nat,
 Nt, Nr,Ndb : text
 %const sec_k2 : protocol_id
 init State := 0
transition
 1. State = 0 /\ Rec(Nr') =|> State' := 1 /\ Nt' := new()
 /\ Snd(H(S,Nr',Nt'),Nt') /\ witness(T,R,aut_tag,Nt')

 2. State = 1 /\ Rec(H(ID,Ndb').Ndb')
 =|> State' := 2 /\ request(T,R,aut_reader,Ndb')
end role
role session(R,T : agent,ID,RID,S : text, H: hash_func)
def=
local Sa,Ra,Sb,Rb : channel(dy)
 composition
 reader(R,T,ID,RID,S,H,Sa,Ra) /\ tag(T,R,ID,RID,S,H,Sb,Rb)
end role
```

Figure 27 : la spécification de la solution proposé de protocole Wei et al.en HLPSL.

La vérification automatisée du protocole de Wei et al. donne le résultat suivant :

Figure 28 : Le résultat de la vérification de protocole «Wei al. Amélioré » par AVISPA.

Ce résultat a clairement montré qu'aucune attaque n'a été détectée lors de l'authentification .On peut également conclure que le temps d'exécution en utilisant la concaténation est inférieur au temps d'exécution en utilisant la primitive xor. Nous pouvons donc conclure que ce protocole est sûr.

Concernant le protocole FDW contrairement à la proposition de protocole Wei et al (WHC) l'idée de notre proposition est de changer la concaténation (||) par la primitive XOR (⊕) .Par conséquent, les nouveaux changements sont {xor (Nr,Nt') }_K . Après le changement la spécification en HLPSL présenté dans la **figure 29.**

```
role reader ( R,T: agent, K: symmetric_key, SND,REC: channel(dy))
 played_by R def= local State : nat, Nr, Nt : text
 const sec_N1 : protocol_id
 init State := 0
 transition
 1. State = 0 /\ REC(start) =|> State' := 1 /\ Nr' := new()
 /\ SND(Nr') /\ witness(R,T,aut_reader,Nr')
 2. State = 1 /\ REC({xor(Nt',Nr)}_K ) =|> State' := 2
 /\ SND({xor(Nr,Nt')}_K) /\ secret(Nt',sec_N1,{R,T})
 /\ request(R,T,aut_tag,Nt')
 end role
role tag ( T,R: agent,K: symmetric_key, SND,REC: channel(dy))
 played_by T def=
 local State : nat, Nt,Nr : text
 const sec_N2 : protocol_id
 init State := 0
 transition
 1. State = 0 /\ REC(Nr') =|> State' := 1 /\ Nt' := new()
 /\ SND({xor(Nt',Nr')}_K)
 /\ secret(Nt',sec_N2,{T,R}) /\ witness(T,R,aut_tag,Nt')
 2. State = 1 /\ REC({xor(Nr,Nt')}_K) =|> State' := 2
 /\ request(T,R,aut_reader,Nr)
 end role
```

Figure 29 : la spécification de la proposition de protocole FDW en HLPSL.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/span/span/testsuite/results/fdwks.if
GOAL
  authentication_on_aut_reader
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.02s
  visitedNodes: 1 nodes
  depth: 1 plies
ATTACK TRACE
i -> (t1,3): x229
(t1,3) -> i: {Nt(1) XOR x229}_k1
i -> (t1,3): {x229 XOR Nt(1)}_k1
```

Figure 30 : Le résultat de la vérification de protocole «FDW Amélioré » par AVISPA.

Le résultat de la vérification des outils AVISPA détectent une trace d'attaque sur l'authentification **figure 30** cela montre que plusieurs attaques sont possibles Si l'on considère les propriétés algébriques de l'opérateur XOR dans les bases de la cryptographie donc le protocole sera insécurisé.

Concernant le protocole RFID basé sur le hachage comme la proposition de protocole FDW l'idée est de changer la concaténation (||) par la primitive XOR (⊕) .Par conséquent, les nouveaux changements présentés dans les **figures 31 et 32.**

```
role servers (S , T : agent ,
H1 , H2 , H3 , CONCAT : hash_func,
SND, RCV : channel ( dy ) )
played_by S
def=
local State : nat ,
Idti : text ,
Kti , KGI : symmetric_key ,
Rs , Rt , MSG , MSGs : message
const sec_kti , sec_kgi , sec_idt , auth_kti : protocol_id
init State :=0
transition
1. State = 0 /\ RCV ( start )
=|> State':=2
/\ Rs':= new ()
/\SND (Rs')
2.State = 2 /\RCV ( MSG')
/\ MSG'= { CONCAT ( xor(xor(xor(Rs , Rt'), H1 ( Kti )), Idti ))} _KGI
=|> State':=4
/\ request (S , T , auth_kti , Kti )
/\ Kti':= H2 ( Kti )
/\ MSGs':= H3 (xor( Kti', Rt) )
/\ SND ( MSGs')
/\ secret ( Idti , sec_idt , {T , S })
/\ secret( Kti , sec_kti , {T , S })
/\ secret( KGI , sec_kgi , {T , S })
/\ witness (S , T , auth_kti , Kti')
end role
```

Figure 31 : la spécification de la solution proposée de protocole RFID basé sur le hachage en HLPSL.

```
role tagt ( T , S : agent ,
H1 , H2 , H3 , CONCAT : hash_func ,
SND, RCV : channel ( dy ) )
played_by T
def=
local State : nat ,
Idti : text ,
Kti , KGI : symmetric_key ,
Rs , Rt , MSG , MSGs : message
const sec_kti , sec_kgi , sec_idt , auth_kti : protocol_id
init State:=1
transition
1. State = 1 /\ RCV (Rs')
=|> State':=3
/\ Rt':= new () /\ Idti':= new()
/\MSG':= {CONCAT( xor(xor(xor(Rs' ,Rt'), H1 ( Kti )), Idti'))} _KGI
/\ SND( MSG')
/\ secret( Idti', sec_idt , {T ,S })
/\ secret( Kti , sec_kti , {T ,S })
/\ secret( KGI , sec_kgi , {T , S })
/\witness(T , S , auth_kti , Kti )
2. State= 3 /\ RCV ( MSGs')/\ MSGs'= H3 ( xor(H2 ( Kti') , Rt) ) =|> State':=5
/\ Kti':= H2 ( Kti')
/\ request(T , S , auth_kti , Kti')
end role
```

Figure 32 : la spécification de la solution proposée de protocole RFID basé sur le hachage en HLPSL.

Le résultat de la vérification dans la **figure 33** montre que la concaténation prendre moindre de temps de recherche que la primitive ‘ou exclusif ’.

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/rfidks.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 3.57s
  visitedNodes: 24 nodes
  depth: 4 plies
```

```
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  /home/span/span/testsuite/results/rfidproto.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 0.12s
  visitedNodes: 24 nodes
  depth: 4 plies
```

Figure 33: Résultat de vérification de la solution proposée de protocole RFID basé sur le hachage en HLPSL.

3.9 Comparaison entre les protocoles

Pour faire notre comparaison entre ces différents protocoles d'authentification présentés plus haut, nous allons réaliser dans ce qui suit ; un test sur deux ordinateurs déférents, une analyse de sécurité et une analyse de performance. Cela afin de déterminer le protocole qui présente les meilleures performances.

3.9.1 Cadre d'évaluation

Afin de mieux comparer les protocoles, nous avons choisis de les tester sur deux ordinateurs déférents **HP** (Processeur AMDA10-4655M APU with Radeon(TM) HD Graphics 2.00 GHz , RAM 7.19 Go) et **TOSHIBA** (Processeur Intel(R) Core (TM) i3-3120M CPU 2.00GHz RAM 3.89).

Protocole d'authentification FDW

	Résultat	**OFMC**			
Ordinateurs		**Temps d'analyse**	**Temps de Recherche**	**nœuds visités**	**Profondeur**
HP (RAM 7.19**Go)**	**Safe**	**0**	**0.07**	**44**	**8**
TOSHIBA (RAM 3.89**Go)**	**Safe**	**0**	**0.12**	**44**	**8**

Tableau 6 : Résultat de vérification dans le back-end OFMC de FDW.

	Résultat	**ATSE**			
Ordinateurs		**Analysés**	**Accessible**	**Traduction**	**Calcul**
HP (RAM 7.19Go)	**Safe**	**2394**	**868**	**0.0**	**0.0**
TOSHIBA (RAM 3.89Go)	**Safe**	**2394**	**868**	**0.01**	**0.02**

Tableau 7 : Résultat de vérification dans le back-end ATSE de FDW.

Protocole d'authentification Wei et al(WHC)

Résultat		OFMC			
Ordinateurs		Temps d'analyse	Temps de Recherche	nœuds visités	Profondeur
HP (RAM 7.19Go)	Unsafe	0	0.13	60	5
TOSHIBA (RAM 3.89Go)	Unsafe	0	0.25	60	5

Tableau 8 : Résultat de vérification dans le back-end OFMC de Wei et al.

Résultat		ATSE			
Ordinateurs		Analysés	Accessible	Traduction	Calcul
HP (RAM 7.19Go)	Unsafe	13	8	0.01	0.0
TOSHIBA (RAM 3.89Go)	Unsafe	13	8	0.03	0.0

Tableau 9 : Résultat de vérification dans le back-end ATSE de Wei at al.

Protocole Wei et al amélioré

Résultat		OFMC			
Ordinateurs		Temps d'analyse	Temps de Recherche	nœuds visités	Profondeur
HP (RAM 7.19Go)	Safe	0	0.03	20	4
TOSHIBA (RAM 3.89Go)	Safe	0	0.08	20	4

Tableau 10 : Résultat de vérification dans le back-end OFMC de Wei et al proposé.

Résultat		ATSE			
Ordinateurs		Analysés	Accessible	Traduction	Calcul
HP (RAM 7.19Go)	Safe	20	4	0.03	0.0
TOSHIBA (RAM 3.89Go)	Safe	20	4	0.08	0.02

Tableau 11 : Résultat de vérification dans le back-end ATSE de Wei et al proposé.

Protocole d'authentification RFID basé sur le hachage

Résultat		OFMC			
Ordinateurs		Temps d'analyse	Temps de Recherche	nœuds visités	Profondeur
HP (RAM 7.19Go)	Safe	0	0.06	24	4
TOSHIBA (RAM 3.89Go)	Safe	0	0.13	24	4

Tableau 12 : Résultat de vérification dans le back-end OFMC de protocole RFID basé sur le hachage.

Résultat		ATSE			
Ordinateurs		Analysés	Accessible	Traduction	Calcul
HP (RAM 7.19Go)	Safe	2	0	0.01	0.0
TOSHIBA (RAM 3.89Go)	Safe	2	0	0.04	0.0

Tableau 13 : Résultat de vérification dans le back-end ATSE de protocole RFID basé sur le hachage.

Donc après la comparaison entre les tableaux **8** et **10** et entre les tableaux **9** et **11** les résultats confirment ce que nous avons discuté dans la section **3.8**.

3.9.2 Analyse de sécurité

- **Attaque Sybil :** Une attaque Sybil est un type de menace de sécurité sur un système en ligne où une personne tente de prendre le contrôle du réseau en créant plusieurs comptes, nœuds ou ordinateurs. Ces derniers peuvent envoyer de fausses informations au serveur ou à l'application gestion des services, afin de prendre les décisions dont l'attaquant a besoin. Parmi les problèmes qui peuvent être causés par les attaques Sybil : les attaquants peuvent être en mesure de voter contre les nœuds honnêtes du réseau s'ils créent suffisamment d'identités fausses (ou d'identités Sybil)[72].

- **Attaque d'usurpation d'identité :** Contrairement à l'attaque Sybil où l'attaquant tente pour créer de nombreuses identités fausses ou virtuelles, l'attaquant tente d'usurper l'identité d'un utilisateur légitime pour obtenir ses privilèges [72].

- **Attaque par substitution de messages :** lors d'une attaque par substitution, l'attaquant intercepte les messages valides pendant les transmissions et les modifie de telle sorte que les destinataires acceptent les faux messages comme s'ils avaient été envoyés par expéditeur d'origine [72].

- **Déni de service :** Une attaque contre un ordinateur qui vise à rendre un service indisponible, empêchant les utilisateurs légitimes du service de l'utiliser. À l'heure actuelle, la grande majorité de ces attaques sont menées à partir de plusieurs sources, on parle alors d'une attaque par déni de service distribué **(DDOS**). Parmi les cyberattaques les plus dangereuses. Sa popularité est due à sa grande efficacité contre tout type de service, car elle ne nécessite aucune identification/exploitation des failles du protocole ou pour un service particulier, il suffit de le vider [73].

- **Attack par rejeu:** Une attaque par rejeu, parfois également appelée attaque par relecture, est une cyberattaque dans laquelle l'entité malveillante intercepte des réitère (on dit également rejoue, ou répète) une transmission de données valide en passant par un réseau. En raison de la validité des données d'origine (qui proviennent généralement d'un utilisateur autorisé), les protocoles de sécurité du réseau traitent l'attaque comme s'il s'agissait d'une transmission de données normale. Cette attaque peut être utilisée pour tromper les institutions financières en dupliquant des transactions, permettant ainsi aux attaquants de retirer de l'argent directement sur les comptes de leurs victimes. [72].

Critères d'évaluation	FDW	FDW proposé	Wei et al	Wei et al amélioré	RFID basé sur hachage
Identification	✓	✓	✓	✓	✓
Authentification mutuelle	✗	✗	✗	✓	✓
Intégrité des données	✓	✗	✗	✓	✓
Disponibilité	✓	✗	✗	✓	✓
Non répudiation	✓	✓	✓	✓	✓
Confidentialité	✓	✗	✗	✓	✓
Attaque Sybil	✗	✗	✗	✗	✗
Attaque d'usurpation d'identité	✗	✓	✓	✗	✗
Attaque par substitution de messages	✗	✓	✓	✗	✗
Déni de service	✗	✓	✓	✗	✗
Attaque par rejeu	✗	✓	✓	✗	✗

Tableau 14 : Analyse de sécurité.

Le tableau 14 montre en claire que notre proposition de protocole Wei et al. et le protocole RFID basé sur le hachage sont les meilleures en termes de tous les critères d'évaluation ci-dessus.

3.9.3 Analyse des performances

❖ Analyse de complexité

Dans le protocole RFID basé sur le hachage, la communication se déroule entre le serveur et l'étiquette Tagi contrairement au protocole de FDW et Wei et al. Dans **FDW**, la communication implique seulement le lecteur et le tag RFID. Le protocole de **Wei et al** implique les trois entités à savoir le serveur S, le lecteur R et l'étiquette T.

Dans protocole RFID basé sur le hachage les étiquettes RFID ont besoin de stocker un ensemble d'identifiants, la clé de groupe, la clé secrète K Tagi, d'implémenter trois fonctions de hachage H1, H2 et H3. Néanmoins, le protocole s'exécute plus rapidement du côté du serveur. Donc la complexité mémoire de protocole RFID basé sur le hachage a légèrement augmentée par rapport au protocole de FDW. mais reste moins élevée que celle de Wei et al. où le serveur a besoin de stocker les données d'identification du lecteur en plus de celle des tags.

Protocole	Cout de calculs		
	Serveur	Lecteur	Étiquette
FDW	0	1TC	1TC
Wei et al amélioré	1Th	1TC	1TC +2TH
RFID basé sur le hachage	τTC +2TH	0	1TC +3TH

Tableau 15 : Étude de complexité.

Les coûts de calculs des protocoles FDW, Wei et al amélioré et protocole basé sur le hachage respectivement résumés dans le **tableau 15**.

Le **tableau 15** montre que le protocole FDW est le plus performant en termes de coûts de calcul pour le serveur et l'étiquette RFID et le protocole basé sur le hachage est meilleur en termes de coûts de calcul pour le lecteur.

Notre proposition implique les trois entités à savoir le serveur S, le lecteur R et l'étiquette T comme le protocole Wei et al qui rend le cout de calcul le même.

❖ Les coûts de communication

Dans cette partie, nous avons calculé la taille totale en bits des messages transmis pendant la phase d'authentification et nous avons également compté le nombre d'échanges entre les différentes entités du système. Pour cela nous avons considéré que ;

- Le résultat de la fonction de hachage est de 256 bits.
- Les autres données sont fixées à 128 bits.
- Le résultat de la fonction de chiffrement correspond à la taille des données en clair fournies en paramètre.

Protocole	Serveur-Lecteur (bits)	Lecteur-Étiquette (bits)	Serveur-Étiquette (bits)	# d'échanges
FDW	**0**	**768**	**0**	**3**
Wei et al amélioré	**1792**	**1152**	**0**	**5**
RPBH	**0**	**0**	**1024**	**3**

Tableau 16 : Étude des coûts de communication.

Le Tableau 16 montre que le protocole RFID basé sur le hachage est meilleur en termes de coût de communication entre le serveur et les étiquettes que celui de Wei et al et FDW. Tandis que protocole Wei et al amélioré est meilleur entre le serveur et les étiquettes et entre le lecteur et les étiquettes.

3.10 Conclusion

Dans ce dernier chapitre, nous avons présenté un l'état de l'art des protocoles d'authentification utilisés dans les systèmes RFID. Parmi ces protocoles, nous avons choisis d'étudier quelques-uns, à savoir : FDW, Wei et al (WHC) ,Wei et al amélioré et un protocoles RFID basé sur l'utilisation d'une fonction de hachage.

Nous avons présenté ensuite l'outil de vérification AVISPA. Ce dernier nous a permis de vérifier formellement ces quatre protocoles choisis et a permis de montrer l'efficacité de ces protocoles à l'égard de certaines attaques.

Suite à nos vérifications, nous avons constaté que FDW et le protocole RFID basé sur le hachage permettent une bonne sécurité, par contre le protocole Wei et al est vulnérable vis-à-vis de certaines attaques. Une version améliorée de ce même protocole a permis de le renforcer et de le rendre plus sûr face à ces attaques. Les résultats de cette vérification nous ont aidés à suggérer des modifications au protocole Wei et al et FDW afin de prouver certaines conclusions.

Conclusion Générale

L'IoT constitue un domaine de recherche très riche. Ce type de réseaux peut être appliqué dans plusieurs domaines différents « militaire, médical, agricole, etc. ». Cependant, de nombreux défis et problèmes restent à résoudre. L'un des problèmes majeurs que nous pouvons rencontrer est celui de la sécurité. Il peut être causé par le fait que les objets connectés emploient des protocoles de sécurité vulnérables. Le réseau est par conséquent exposé à plusieurs types d'attaques internes ou externes. Dans le cadre de ce mémoire, nous nous sommes intéressés à l'aspect sécurité dans ce type de réseaux. Notre attention s'est portée plus particulièrement sur les systèmes d'authentification dans l'Internet des Objets. Pour cela, nous avons commencé par nous immerger dans le monde des IoT en passant en revue différentes notions de bases. Ensuite, nous nous sommes focalisés sur le côté sécurité, plus particulièrement, l'authentification. Nous avons dressé un état de l'art des différents protocoles d'authentifications proposés dans la littérature et nous avons choisis d'en étudier quelques un d'une manière un peu plus approfondie. Pour cela, nous avons procédé à une vérification formelle des protocoles choisis grâce à l'outil AVISPA.

Notre travail s'articule autour de trois chapitres :

Le premier chapitre a été dédié à une présentation générale d'IdO. Les différentes technologies de communication dans l'IdO et les domaines d'application.

Le deuxième chapitre parle de la sécurité dans l'IdO. Nous avons présenté les propriétés et les mécanismes de sécurité dans leur globalité, ensuite, nous nous sommes concentré plus particulière sur le mécanisme d'authentification (définition, différents types d'authentification, les protocoles et aussi les défis).

Finalement dans le troisième chapitre nous avons commencé par présenter l'outil AVISPA et le langage de spécification HLPSL. Ensuite, nous avons procédé à la vérification de trois protocoles d'authentification appliqués aux systèmes RFID, et nous avons fait une comparaison entre ces protocoles.

Ce projet nous a été d'un grand apport puisqu'il nous a permis d'approfondir nos connaissances dans le domaine d'IdO, principalement la sécurité. Il nous a permis de manipuler certains outils tels que SPAN et AVISPA.

Bibliographie

[1] Droua sohib, Terir Karim, Gestion de la confidentialité des données pour les dispositifs IOT (Internet of Things), Mémoire de fin d’´etude pour obtention du diplôme de Master, Université Mohamed Sadik Ben Yahia de Jijel, Algérie, 2020.

[2] Somia SAHRAOUI « Mécanismes de sécurité pour l’intégration des RCSFs à l’IoT (Internet of Things) », Université de Batna 2, 09/11/2016.45-46.

[3] A. Hakin, A. Gokhale, P. Berthou, D. C. Schmidt, T. Gayraud, Software-Defined Networking: Challenges and research opportunities for Future Internet, Computer Networks 75 (part A) (2014) 453–471.
[4] Ammar, M.; Russello, G.; Crispo, B. Internet of Things: A survey on the security of IoT frameworks. J. Inf.Secur. Appl. 2018, 38, 8–27.

[5] Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of things: A survey onenabling technologies, protocols, and applications. IEEE Commun. Surv. Tutor. 2015, 17, 2347–2376.

[6] Da Xu, L.; He, W.; Li, S. Internet of things in industries: A survey. IEEE Trans. Ind. Inform. 2014,10, 2233–2243.

[7] Hammoudeh, M.; Epiphaniou, G.; Belguith, S.; Unal, D.; Adebisi, B.; Baker, T.; Kayes, A.; Watters, P.A service-oriented approach for sensing in the Internet of Things: intelligent transportation systems andprivacy use cases. IEEE Sens. J. 2020. doi:10.1109/JSEN.2020.2981558.

[8] Suo, H.; Wan, J.; Zou, C.; Liu, J. Security in the internet of things: A review. In Proceedings of the2012 International Conference on Computer Science and Electronics Engineering, Hangzhou, China,23–25 March 2012; Voumn. 3, pp. 648–651.

[9] Mrabet, H.; Beghith, S; Alhomoud, A; Jemai, A.A Survey of IoT Security Based on a LayeredArchitecture of Sensing and Data Analysis. Sensors 2020, 20, 3625; doi:10.3390/s20133625

[10] Jean Paul Khorez EZIKOLA MAZOBA, L'étude de l'internet des objets et contrôle d'accès aux données, Mémoire Online, Université Panafricaine, 2015.

[11] Imad Saleh. Les enjeux et les défis de l’Internet des Objets (IdO). 1 Laboratoire Paragraphe Université Paris 8, 2018.

[12]Yassine HADDAB, Professeur à l’Université de Montpellier, Introduction à l’internet des objets (IdO – IoT)

[13] Emmanuel Baccelli, Internet des Objets défis sociétaux et domaines de recherche Scientifique pour l'Internet des Objets (IoT), nouvembre2021.

[14] Fatma Merabet. Solutions de sécurité pour l'internet des objets dans le cadre de l'assistance à l'autonomie à domicile. Cryptographie et sécurité [cs.CR]. Université de Limoges; Université Mouloud Mammeri (Tizi-Ouzou, Algérie), 2021.51-54.

[15] Martin Koppe, Les défis de l'Internet des objets, 27.09.2016

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

[17] Joe Decuir Standards Architect , gy CSR Technology Councilor, Bluetooth Architecture Review Board IEEE Region 6 Northwest Area chair 09.2015

[18] Habib NASSER, Nacer K. M'SIRDI, Aziz NAAMANE, Hassen MEKKI2 , Hatem TLEJANI3 . 1 LSIS, CNRS UMR 6168. Domaine Universitaire St Jérôme, Av. Escadrille Normandie – Niémen ; 13397. MARSEILLE Cedex 20. FRANCE.

[19] Chakkor, Saad , « E-diagnostic de processus physiques à base des méthodes de haute résolution »,2015

[20] Christian Cosquer, Julie Lanckriet « Les objets connectés et la Défense » Dans Revue Défense Nationale 2016/2 (N° 787), pages 97 à 103

[21] Jonathan Roux. Détection d'intrusion dans des environnements connectés sans-fil par l'analyse des activités radio. Informatique [cs]. Université Toulouse 3 Paul Sabatier (UT3 Paul Sabatier), 2020. Français. fftel-02880658v1f

[22] Karine Baillargeon-Audet1 Francis Fortin2 « Réseaux sans fil et éléments criminogènes » ,Canada 2013

[23] Ming-An Chung ,Wei-Hsuan Chang " Low-cost, low-profile and miniaturized single-plane antenna design for an Internet of Thing device applications operating in 5G, 4G, V2X, DSRC, WiFi 6 band, WLAN, and WiMAX communication systems" 30 December 2019

[24] ARADJ TAHA MOHAMED EL-AMINE « Usage du protocole MQTT dans une application de suivi ». Université Mohamed Khider – BISKRA .2017

[25] YAHIAOUI Ayoub, BENJENNA Hakim, ROOSE Philippe « Patrons temporels pour spécifier les systèmes auto-adaptatifs » Laboratoire LAMIS, Université Larbi

Tébéssi, Route de Constantine, 12002 Tébessa, Algérie . Laboratoire LIUPPA, Univ Pau & Pays Adour, EA 3000, 64600 Anglet, France.

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

[27] Souhayla, Ferhane « l'internet des objets révolutionne notre vie quotidienne: application pour une maison intelligent ».2021

[28] BACHOTI Youssef, BELHAJ SENDAGUE Bassim, RODRIGUES OLIVEIRA Joao Gabriel « PROJET RFID ». 25 janvier 2011.

[29] TAIEB BRAHIM, MOHAMED,Evaluation et amélioration du routage dans les réseaux sans fils Ad Hoc,Université de Sidi Bel Abbès - Djillali Liabes.

[30] DAHMANE Abir, BOUZIDI Meriem ,ELABED Sid ali, Implémentation d' n Réseau Prototype pour l'Internet des Objets , Université de Mohamed El-Bachir El-Ibrahimi - Bordj Bourreridj, 19/09/2021.

[31] Zendaoui, Lokmane Boulkamh, Chouaib,Internet des objets pour le controle de l'eclairage d'une maison.Université Oum El Bouaghi,2019.

[32] Meftah, ZOUAI ,*Une approche cloud computing basée IoT pour la maison intelligente.* Thèse de doctorat, Université de mohamed kheider biskra. (2021)

[33] Djebar Yacine, Management des systemes d'information internationaux (sii), université 8 mai 1945 de Guelma. 2016/2017

[34] Fatma Merabet. Solutions de sécurité pour l'internet des objets dans le cadre de l'assistance à l'autonomie à domicile. Cryptographie et sécurité [cs.CR]. Université de Limoges; Université Mouloud Mammeri (Tizi-Ouzou, Algérie), 2021. Français. ffNNT : 2021LIMO0037ff. fftel-03326960 (page37)

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

[36] Alexandra Boldyreva, Nathan Chenette, Younho Lee, and Adam O'neill. Order-preserving symmetric encryption. In Annual International Conference on the Theory and Applications of Cryptographic Techniques, pages 224–241. Springer, 2009.

[37] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Annual International Cryptology Conference, pages 537–554. Springer, 1999

[38] Monika Agrawal and Pradeep Mishra. A comparative survey on symmetric key encryption techniques. International Journal on Computer Science and Engineering, 4(5) :877, 2012.

[39] TALEB, F. Support de cours “Cryptologie”, département d’informatique, Université Dr Moulay TAHAR, Saida, Algérie, 2019.

[40] Claus-Peter Schnorr. Efficient identification and signatures for smart cards. In Conference on the Theory and Application of Cryptology, pages 239–252. Springer, 1989.

[41] Uriel Feige, Amos Fiat, and Adi Shamir. Zero-knowledge proofs of identity. Journal of cryptology, 1(2) :77–94, 1988.

[42] Amos Fiat and Adi Shamir. How to prove yourself : Practical solutions to identification and signature problems. In Conference on the Theory and Application of Cryptographic Techniques, pages 186–194. Springer, 1986

[43] Hartwig Mayer. Ecdsa security in bitcoin and ethereum : a research survey. CoinFaabrik, June, 28 :126, 2016.

[44] [N.Sornin (Semtech), M.Luis (Semtech), T.Eirich (IBM), T.Kramp (IBM), O.Hersent (Actility). LoRaWAN Specification.LoRa Alliance, 1.0, 2015

[45] A. N. Bikos and N. Sklavos.LTE/SAE Security Issues on 4G Wireless Networks. IEEE Security Privacy, 11(2) :55–62, March 2013.

[46]] Chao-Chen Yang Min-Shiang Hwang and Cheng-YehShiu. An authentication scheme for mobile satellite communication systems.acmsigops operating systems review. 37(4)(10.21494), 2003.

[47] Pascal Urien and Marc Loutrel. La carte `a puce eap, un passeport pour la s´ecurit´e des r´eseaux ´emergentswi-fi. 5`emes Journ´eesR´eseaux, JRES2003, Lille, France, 2003.

[48]Denis Dessales. Conception d’un r´eseau de capteurs sans fil, faible consommation, d´edi´e au diagnostic in-situ des performances des bˆatiments en exploitation. Poitiers, 2011.

[49] Gabriel Montenegro NandakishoreKushalnagar and Christian Schumacher. overlowpowerwirelesspersonal area networks (6lowpans) : overview, assumptions, problemstatement, and goals. ietf, ietf, rfc4919 ipv6. 5`emes Journ´eesR´eseaux, JRES2003, Lille, France, 2007.

[50]NandakishoreKushalnagar, Gabriel Montenegro, and Christian Schumacher. IPv6 over low-power wireless personal area networks (6LoWPANs) : overview, assumptions, problem statement, and goals. IETF, IETF, RFC4919, August 2007. 58,

[51]Khemissa Hamza. Protocole de gestion d'authentification et de les identités dans l'Internet des Objets .Université des sciences et de technologies Houari Boumediene (pages 25 et 39)

[52] kevunie , Authentification : guide complet sur ce mode de sécurisation en ligne,27 septembre 2021 .

[53] Achraf Fayad. Secure authentication protocol for Internet of Things. Networking and Internet Architecture [cs.NI]. Institut Polytechnique de Paris, 2020. English44-53.

[54] Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. (2006). EMAP: An efficient mutual authentication protocol for low-cost RFID tags. In O. T. M. Federated (Ed.) Conferences and workshop: IS workshop, Montpellier, France.

[55] Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. (2006). LMAP: A real lightweight mutual authentication protocol for low-cost RFID tags. In Proceedings of 2nd workshop RFID security. Graz, Austria: Ecrypt.

[56] Peris-Lopez, P., Hernandez-Castro, J. C., Tapiador, J. M. E., & Ribagorda, A. (2006). M2AP: a minimalist mutual-authentication protocol for low-cost RFID tags. In Proceedings of 2006 international conference on ubiquitous intelligence and computing, Wuhan and Three Gorges.

[57] Alomair, B., Lazos, L., & Poovendran, R. (2007). Passive attacks on a class of authentication protocols for RFID. In K.-H. Nam & G. Rhee (Eds.), International conference on information security and cryptologyICISC 2007. Seoul, Korea: Lecture notes in computer science.

[58] Barasz, M., Boros, B., Ligeti, P., Loja, K., & Nagy, D. A. (2007). Breaking LMAP. In: Conference on RFID security, Malaga, Spain.

[59] Li, T. & Wang, G. (2007). Security analysis of two ultra-lightweight RFID authentication protocols. In Proceedings of 22nd IFIP TC-11 Int'1 information security conference, Sandton, Ganteng, South Africa.

[60] Li, T., & Deng, R. (2007). Vulnerability analysis of EMAP-an efficient RFID mutual authentication protocol. In Proceedings of second international conference on availability, reliability, and security (AReS'07), Vienna, Austria.

[61] Chien, H. Y. (2007). SASI: A new ultralightweight RFID authentication protocol providing strong authentication and strong integrity. IEEE Transaction on Dependable and Secure Computing, 4, 337–340.

[62] Castro, H., Tapiador, M. E., Lopez, P., & Quisquater, J. (2008). Cryptanalysis of the SASI ultralightweight RFID authentication protocol with modular rotations, arXiv preprint arXiv:0811.4257.

[63] Jeon, I., & Yoon, E. (2013). A new ultra-lightweight RFID authentication protocol using merge and separation operations, 7, 2583–2593.

[64] Zhuang, X., Zhu, Y., & Chang, C. C. (2013). Security Analysis of Ultralightweight RFID Protocols. Technique Report

[65] Malika MOHAMEDI, Samia IKERBANE.(2016). Vérification automatique d'un protocole de sécurité dans les systèmes RFIDs à base d'outils AVISPA & SPAN.

[66] Rafik Kheddam .(2010).Génération de tests de protocoles cryptographiques par mutations de modèles.

[67] Noureddine Chikouche , Mohamed Benmohammed. Vérification automatique des protocoles d'authentification des systèmes RFID.

[68] Xu Zhuang · Yan Zhu & Chin-Chen Chang. << A New Ultralightweight RFID Protocol for Low-Cost Tags: R2AP>>. Wireless Pers Commun (2014) 79:1787–1802 DOI 10.1007/s11277-014-1958-x. Springer Science+Business Media New York 2014, 26 July 2014.

[69]Jaouhar Fattahi, « Analyse des protocoles cryptographiques par les fonctions témoins » , université Laval , Québec, Canada p38-39

[70]Lawrence C. Paulson. Mechanized proofs for a recursive authentication protocol. In In 10th IEEE Computer Security Foundations Workshop, pages 84–95. IEEE Computer Society Press, 1997.

[71] Chikouche Noureddine. « Problèmes de Sécurité dans les Systèmes Embarqués (Security Problems in Embedded Systems) » 17 mars 2016.

[72] Mohamed Tahar Hammi. Sécurisation de l'Internet des objets. Réseaux et télécommunications [cs.NI]. Université Paris-Saclay, 2018. Français. ffNNT : 2018SACLT006ff. fftel-01997261f.

[73] Hijab Ali. Implémentation d'un protocole d´etection d'un serveur d'authentification dans l'internet des objets. master's thesis,. 2017.