الجمـــهوريـــة الجـــزائريـــة الديـمقراطيـــــــة الشـــعبيـــــة
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعـلـيـم الـعـالـــي والـبـحـث الـعـلـمي
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
جـامـعـة عمّـــار ثـلـيـــجـــي بالأغـــواط
UNIVERSITE AMAR TELIDJI LAGHOUAT
كـلـيـــــة الـعـلـــــوم
FACULTE DES SCIENCES
DEPARTEMENT D'INFORMATIQUE

Mémoire de MASTER

Domaine : Mathématiques et Informatique
Filière : Informatiques
Option : Réseaux, Systèmes et Applications Réparties

Par:
Behitila mohammed bachir
Laouti mostapha farouk

THEME

La sécurité dans les réseaux WBAN

Soutenu publiquement le 26-06-2019 devant le jury composé de:

Mr.Y.guellouma	*M.C.(B)*	*Président*
Mr.F.Bousbaa	*M.C.(A)*	*Examinateur*
MrTahari Abdou el karim	*M.C.(A)*	*Encadreur*
MrBahache Mohammed	*M.C.(A)*	*Co-Encadreur*

N …… Année Universitaire 2018/2019

REMERCIEMENTS

C'est avec un immense plaisir que je réserve ces quelques lignes en signe de gratitude et de reconnaissance à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce travail.

Je souhaite adresser, en premier lieu, mes remerciements les plus sincères à mon encadrant Dr TAHARI abdou el karim pour ses conseils lucides et pertinents, sa patience et son précieux suivi tout au long de la réalisation de ce travail. Sa disponibilité, ses qualités pédagogiques et humaines, et ses compétences m'ont apporté un encadrement déterminant dans toutes les phases de ce travail. Qu'il trouve ici le témoignage de mon profond respect.

Mes remerciements vont aussi à mon Co-Encadrant MR MED.BAHACHE, pour sa disponibilité et l'aide précieuse qu'il m'a apportée. Sa compréhension, son soutien et son suivi m'ont permis de mener à bien ce travail.

Je tien également remercier les membres du jury d'avoir consacré une partie de leur temps à la lecture de ce mémoire et pour l'intérêt qu'ils ont porté à ce travail.

Mes remerciements s'étendent à tous mes enseignants du département d'Informatique de l'Université AMMAR THLIDJI.

Je remercie enfin toutes les personnes qui ont contribué de près ou de loin à l'accomplissement de ce travail.

Laouti mostapha farouk

Remerciements

Je souhaiterais remercier l'équipe de recherche de la formation génie logiciel du temps qu'elle m'a réservé, le temps que chacun de ses membres m'a accordé, et plus globalement, pour toutes les informations, références bibliographiques, réflexions, corrections, que chacun m'a apporté et qui ont nourrit ce travail.

Je remercie également le département math et informatique de m'avoir appris à aimer le monde numérique et digital.Je remercie également mes enseignants pour la qualité de l'enseignement qu'ils m'ont prodigué au cours de ces 5 années passées à l'université Ammar Thlidji Laghouat.

Je remercie tout particulièrement Mr Tahari et Mr Bahache (respectivement encadreur et Co-Encadreur de ce mémoire) qui m'ont laissé une large part d'autonomie dans ce travail tout en m'aiguille sur des pistes de réflexions riches et porteuses.Je souhaiterais aussi remercier tout le corps administratif, ainsi que toutes les personnes qui souhaiteraient voir un jour notre université au meilleur rang.

Je remercie enfin l'ensemble des mes proches Surtout mon père et ma mère qui m'ont aidé et motivé durant ce cursus rempli d'embuches, je les remercie pour l'aide qu'ils m'ont apporté dans la réalisation de ce travail.

Je veux remercier les personnes que j'ai vu lors Pendant ma carrière universitaire, les personnes optimistes que j'ai pu croiser, les personnes qui font beaucoup avec peu de moyens.

Merci à tous !

Behitila med bachir

Table des matières

Table des figures

Liste des tableaux

Table des abréviations

Abréviation	Signification
RCSF	Réseaux de Capteurs Sans Fil
MANETs	Mobile Ad hoc Networks
ADC	Analog-to-Digital Converter
DSN	Distributed Sensor Networks
WSN	Wireless sensor network
WBAN	Wireless body area network
DARPA2	Defense Advanced Research Projects Agency
RSA	(Rivest–Shamir–Adleman) algorithm
DES	(Data Encryption Standard) algorithm
MD 4/5	Message Digest
AES	Advanced Encryption Standard
SHA	Secure Hash Algorithm
ECC	elliptic curve cryptography
SPAN	Security Protocol ANimator
AVISPA	Automated Security Protocol Analysis

Introduction général

L'essor des nouvelles technologies ainsi que les progrès effectués dans les domaines des micro-électroniques, des télécommunications, des réseaux et du traitement de l'information ont entraîné l'apparition de nouveaux outils et objets communicants qui améliorent notre qualité de vie. Parmi ces objets communicants nous intéressons aux capteurs médicaux.

Au cours des dernières décennies et grâce à l'avancée des systèmes embarqués et des technologies sans fil, les Réseaux de Capteurs Sans Fil (RCSF), ou « WSN », sont de plus en plus utilisés dans de nombreux domaines. Parmi ces domaines, nous nous intéressons aux RCSF pour les applications médicales. Imaginons un ensemble de petits appareils électroniques, autonomes, équipés de capteurs et capables de communiquer entre eux via un médium sans fil, chacun de ces petits appareils constitue un nœud capteur médical. Ces nœuds déployés sur le corps du patient ou dans son environnement forment ensemble un réseau des capteurs sans fil médicaux qui est capable de surveiller l'état de santé du patient en collectant des informations physiologiques et de communiquer ensuite ces informations à une équipe médicale à distance.

L'un des défis majeurs du monde de ces dernières décennies a été l'augmentation continue de la population des personnes âgées dans le Monde . D'où la nécessité de fournir des soins de qualité à une population en croissance rapide, tout en réduisant les coûts des soins de santé.

Dans ce contexte, de nombreux travaux de recherche portent sur l'utilisation des réseaux de capteurs sans fil médicaux dans les systèmes WBAN (Wireless Body Area Network), pour faciliter et améliorer la qualité du soin et de surveillance médicale à distance. Ces réseaux sont caractérisés par la mobilité de leurs nœuds capteurs, par leur facilité de déploiement et leur auto-organisation, ce qui est un point avantageux pour la surveillance des personnes âgées, des personnes à mobilité réduite, des personnes à risques et des personnes ayant des maladies chroniques ainsi que pour la surveillance de leur environnement de vie.

Les réseaux des capteurs sans fil médicaux sont utilisés aujourd'hui dans la médecine pour surveiller certains signes vitaux comme la température, la pression

artérielle ou le rythme cardiaque,… etc. Donc les systèmes WBAN permettent non seulement d'améliorer la qualité de vie des patients, mais aussi le suivi des patients en temps réel et d'intervenir le plus rapidement possible dans les cas d'urgences.

Ces réseaux de capteurs médicaux sans fils soulèvent de nouveaux défis technologiques en termes de sécurité et de protection contre les anomalies et les attaques. Le mode de communication sans fil utilisé entre ces capteurs et l'unité de traitement (puits ou sink en anglais) accentue ces vulnérabilités.

Les systèmes WBAN de surveillance médicale à distance sont vulnérables à différents types d'attaques et d'anomalies. Parmi ces attaques et anomalies, il y en a ceux qui visent la disponibilité et l'intégrité du système et donc qui peuvent avoir d'une façon indirecte une influence très dangereuse sur la qualité de soin et sur la vie des patients et d'autres qui visent la confidentialité du système et donc peuvent avoir une influence sur la confidentialité des données médicales.

En effet les vulnérabilités dans un système WBAN de surveillance médicale à distance se décomposent en deux parties principales.

La première partie se compose des anomalies possibles dans les nœuds capteurs et les attaques possibles sur le réseau des capteurs médicaux et sur le médium de communications sans fils entre ces capteurs et le nœud de collecte. La deuxième partie se compose des attaques possibles sur les communications à haut débit entre le système WBAN et le serveur médical.

L'objectif de notre travail est d'effectuer une étude comparative entre deux protocoles aux réseaux WBAN dans l'environnement de simulation "AVISPA" et en exploitant les fonctionnalités de la plateforme de simulation de réseaux de capteur sans fil "SPAN".

Ce rapport est organisé en 4 Chapitres :

-Chapitre 1 : Dans ce chapitre nous présentons les capteurs sans fil avec leurs architectures puis les concepts de base des réseaux des capteurs sans fil suivi d'une comparaison entre les réseaux sans fil et les réseaux adhoc.

- Chapitre 2 : Dans ce chapitre les réseaux sans fil corporelle "WBAN" seront exposés. Nous commençons par les capteurs médicaux ainsi que leurs architectures, puis les

réseaux WBAN (architecture, topologie ainsi que leurs contraintes) seront présentés, Enfin, on termine par une comparaison entre les réseaux WBAN et réseaux ADHOC.

- **Chapitre 3 : Dans** ce chapitre on présente les notions de base relié à la cryptographie symétrique et asymétrique telle que le chiffrement et ces déférents types.

- **Chapitre 4** : Dans ce chapitre on a choisi deux protocole précis sur la sécurité dans les réseaux WBAN, Nous expliquons les deux protocoles choisis et nous implémente en utilisant le langage HLPSL. Enfin nous vérifions leur sécurité en utilisant outil "AVISPA"

Chapitre I Généralités sur les réseaux de capteurs sans fils

Introduction

Les progrès dans le domaine de l'électronique miniaturisée et les communications sans fil ont donné naissance à des composants capables de prélever des grandeurs environnementales, physiologiques etc. Ces composants sont appelés des nœuds capteurs et ils ont la capacité de s'auto-organiser pour former un réseau de capteurs sans fil (RCSF).

Les RCSF permettent de faciliter le suivi et le contrôle à distance de l'environnement physique avec une meilleure précision. Ils peuvent aussi être déployés pour exploiter diverses applications (environnementales, militaires, médicales, etc). En outre, un réseau de capteurs est constitué généralement d'un grand nombre de nœuds capteurs car ces derniers sont sujets à pannes accidentelles ou intentionnelles. Chaque nœud est composé principalement d'un ou plusieurs capteurs, d'une unité de traitement et d'un module de communication. Ces nœuds communiquent entre eux selon une certaine topologie du réseau afin d'acheminer les informations à un centre de contrôle distant de la zone de leur déploiement. La mise en place d'un RCSF pose de nombreux problèmes, car ces nœuds capteurs sont caractérisés par des ressources très limités, que soit énergie, traitement, stockage ou communication, de ce fait ces facteurs doivent être prises en considération lors de conception et la mise en place d'un RCSF.

Ce chapitre sera consacré pour présenter les RCSFs.

Nous commençons par une définition d'un capteur, son architecture, ses types et voir comment ces derniers sont déployés pour former un réseau de capteurs sans fil. Ensuite, la topologie, et les spécificités des RCSFs seront étudiés, ainsi que les domaines d'application des réseaux de capteurs sans fil .Ensuite une petite conclusion pour terminer le chapitre

I.1 Le capteur

I.1.1 Définition d'un capteur

Un capteur est un dispositif ayant pour tâche de transformer une mesure physique observée en une mesure généralement électrique qui sera à son tour traduite en une donnée binaire exploitable et compréhensible par un système d'information. [1]

Exemple : Parmi les différents types de mesures enregistrées par les capteurs, on peut citer entre autres : la température, l'humidité, la luminosité, l'accélération, la distance, les mouvements, la position, la pression, la présence d'un gaz, la vision (capture d'image), le son, … etc

I.1.2 Architecture d'un capteur

Dans cette section, nous distinguons les deux parties qui composent un capteur :

a) Architecture matérielle

La figure 1 est l'illustration la plus générale de l'architecture d'un capteur dit intelligent.

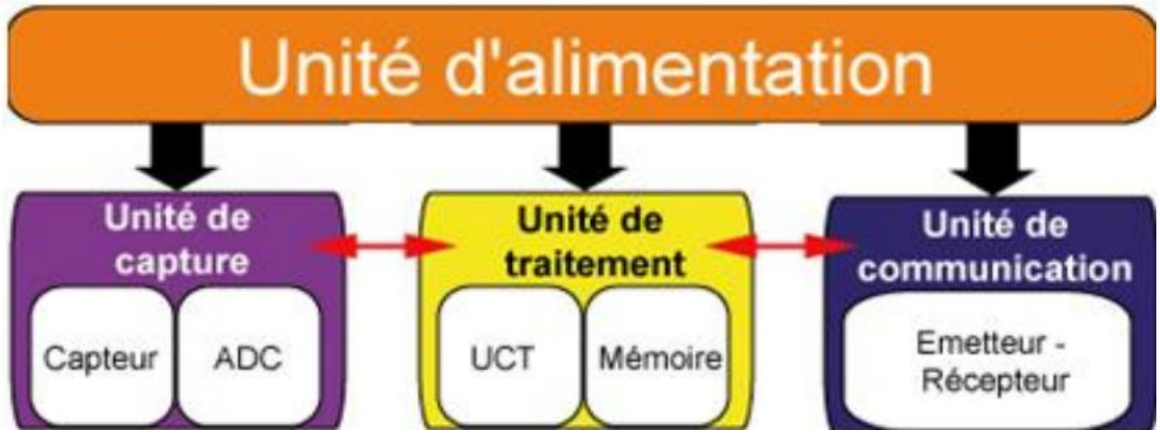


Figure I-1 Architecture d'un capteur sans fil [1]

Cette architecture s'articule autour de quatre unités :

- **L'unité de traitement :** c'est l'unité principale du capteur. Elle est généralement représentée par un processeur couplé à une mémoire vive. Son rôle est de contrôler le bon fonctionnement des autres unités. Sur certains capteurs elle peut embarquer un système d'exploitation pour faire fonctionner le capteur. Elle peut aussi être couplée à une unité de stockage, qui servira par exemple à y enregistrer les informations transmises par l'unité d'acquisition de données. - l'unité d'acquisition : elle permet la mesure des grandeurs physiques ou analogiques et leur conversion en données numériques. Elle est composée du capteur lui-même et de l'ADC qui permet

la conversion des données. Le capteur est chargé de récupérer les signaux analogiques qu'il transmet à l'ADC qui a pour rôle de transformer et de communiquer les données analogiques en données numériques compréhensibles pour l'unité de traitement. [1]

- **L'unité de communication :** elle a pour fonction de transmettre et recevoir l'information. Elle est équipée d'un couple émetteur/récepteur pour communiquer au sein du réseau. Il existe cependant d'autres possibilités de transmission (optique, infrarouge, etc. ..).

- **L'unité d'alimentation** : c'est un élément primordial de l'architecture du capteur, c'est elle qui fournit en énergie toutes les autres unités. Elle correspond le plus souvent à une batterie ou une pile alimentant le capteur, dont les ressources limitées en font une problématique propre à ce type de réseau puisque ces derniers sont généralement déployés dans des zones non accessibles. La réalisation récente d'unité d'alimentation à base de panneaux solaires tente d'apporter une solution pour prolonger sa durée de vie [2].

Par ailleurs, un capteur peut être doté d'autres unités. Citons, entre autres, la possibilité pour assurer la mobilité du capteur, ou une unité spécifique de capture comme une caméra pour de l'acquisition vidéo.

b) Architecture Logicielle

La contrainte énergétique des capteurs exige l'utilisation de systèmes d'exploitation légers tels que TinyOS [3] ou Contiki [4]. Cependant, TinyOS reste toujours le plus utilisé et le plus populaire dans le domaine des RCSF. Il est libre et est utilisé par une large communauté de scientifiques dans des Simulations pour le développement et le test des algorithmes et protocoles réseau.

I.1.3 Types de capteurs

Il existe un grand nombre de capteurs, avec des fonctionnalités diverses et variées.

La plupart des capteurs dépendent de l'application pour lesquels ils ont été conçus (capteur aquatique, sous-terrain, etc. . .). Il est plus intéressant de décrire les capteurs les plus utilisés et leur évolution au cours du temps.

En l'occurrence, la figure 2 illustre l'évolution des capteurs au cours de dernières années. Cette représentation met en avant l'importance des travaux de recherche de l'université de Berkeley dans l'essor des réseaux de capteurs, surtout sachant que l'entreprise Xbow (aussi appelé Crossbow) qui fait jusqu'à aujourd'hui office de référence dans la fabrication de capteurs est née au sein de la célèbre université californienne. [5]

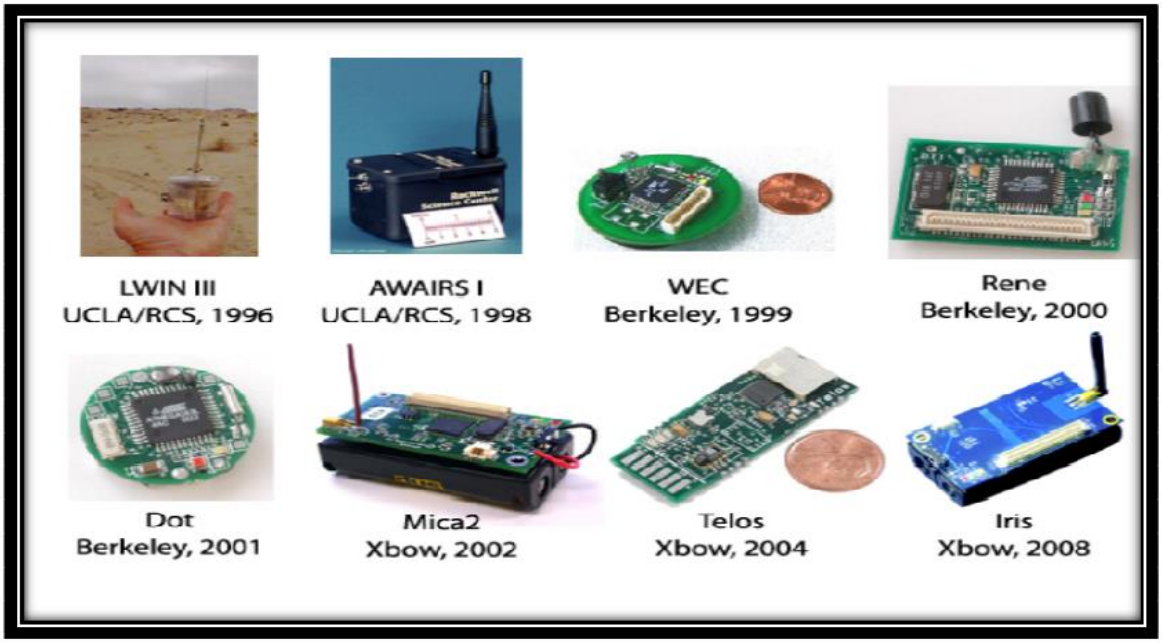


Figure I-2 Quelques exemples de capteurs [6]

Les capteurs fabriqués par Xbow (famille de capteurs Mica et Telos) sont sans aucun doute les plus utilisés dans les expériences et travaux de recherche. Ces capteurs sont capables de mesurer plusieurs métriques (température, humidité, luminosité, etc. . .) et s'articulent pour la plupart d'entre eux autour du Chipcon CC2420 qui est devenu le standard au niveau des modules de transmission utilisant le protocole de communication IEEE 802.15.4.

I.1.4 Exemple de capteur sans fil

Le modèle que nous allons présenter est le MicaZ

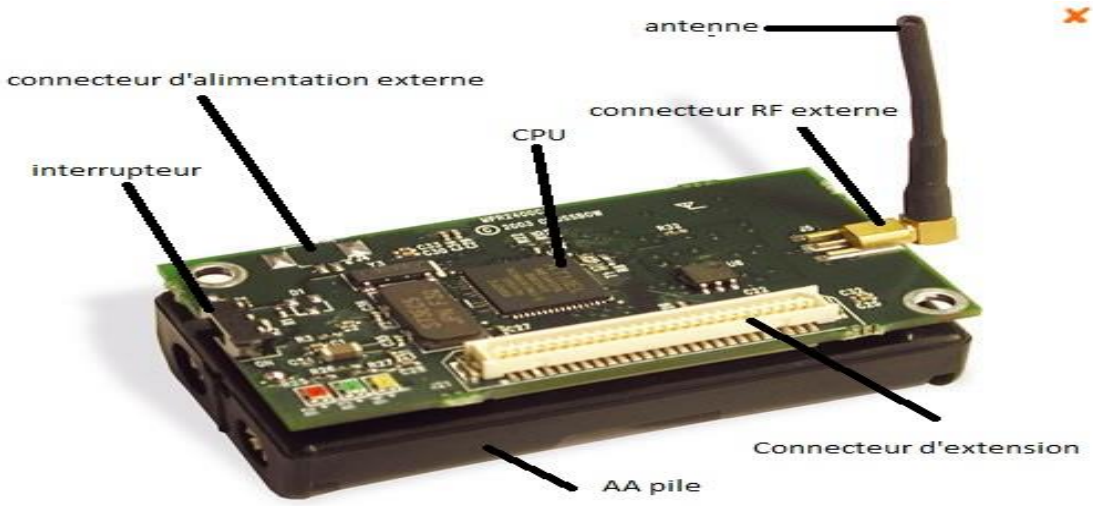


Figure I-3 capteur Micaz [1]

On remarque que les capteurs sont constitués de quatre principaux groupes de composants ayant chacun leur propre rôle [5] :

- **Processeur et mémoire :** composé du processeur qui effectue les traitements, de la mémoire RAM qui stocke les données temporaires et de la mémoire flash qui contient le système d'exploitation. Le MicaZ est constitué d'une mémoire flash de 512 Kb permettant de stocker plus de 100 000 mesures ainsi que d'une mémoire flash de 128 Kb
- **Communication :** composé d'une antenne et d'un système de communication radio an de pouvoir émettre et recevoir des signaux sans fil.
- **Alimentation :** composé de la batterie fournissant l'énergie nécessaire au fonctionnement et assurant ainsi l'autonomie du capteur.
- **Interactions :** composé des interfaces, de système de capture, de LEDs et qui permet au capteur d'interagir avec son environnement.
  Le capteur illustré dans la Figure 2, est alimenté par deux piles AA et fonctionne sur une plage théorique allant de 2,7 à 3,3 Volts. La portée de ce capteur peut atteindre une trentaine de mètres en intérieur et s'étendre jusqu' à 100 mètres en milieu ouvert.

I.2 Réseaux de capteurs sans fil (RCSF)

I.2.1 Description des réseaux de capteurs

Un RCSF est composé d'un ensemble de nœuds capteurs. Ces nœuds capteurs sont organisés en champs « sensed area » (voir figure I-4). Chacun de ces nœuds sa la capacité de collecter des données et de les transférer au nœud passerelle (dit "sink" en anglais ou puits) par l'intermédiaire d'une architecture multi-sauts. Le puits transmet ensuite ces données par Internet ou par satellite à l'ordinateur central « Gestionnaire de taches» pour analyser ces donner et prendre des décisions. [6]

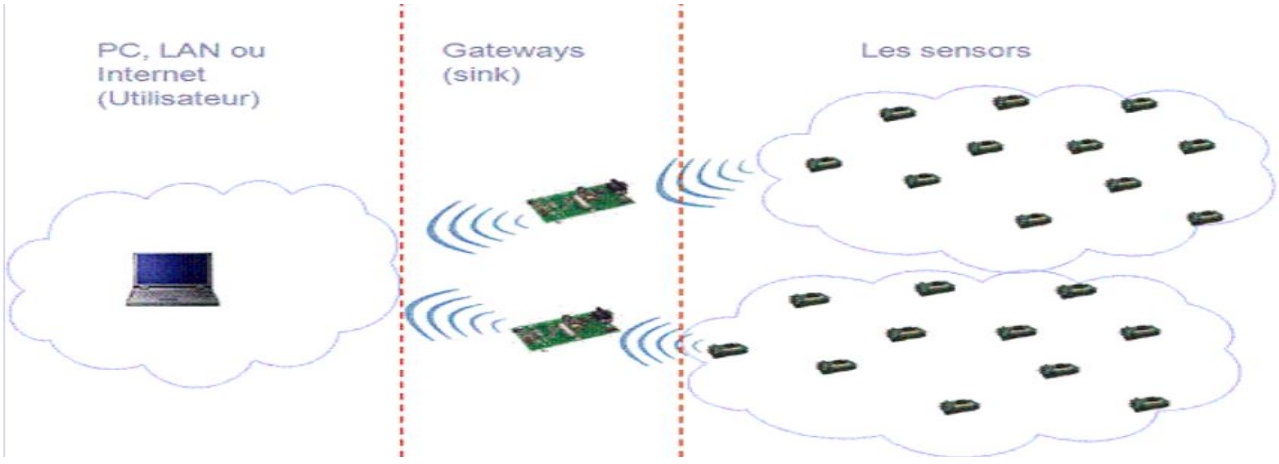


Figure I-3 Architecture d'un RCSF [6]

I.2.2 Similarité et différence entre (RCSF) et réseaux Adhoc (MANETs)

Les réseaux de capteurs sont considérés comme un type particulier des réseaux ad-hoc et sont significativement différents des réseaux MANET (Mobile Ad-hoc NETwork) traditionnels. Le tableau I-1 résume les similarités et les différences entre les réseaux de capteurs sans fil et les MANETs:

Réseau de capteurs sans fils	**Réseau MANET (Ad-Hoc)**
1. Utilisation d'un médium sans fil.	1. Utilisation d'un médium sans fil.
2. Réseau auto-configurable	2. Réseau auto-configurable
3. Topologie dynamique	3. Topologie dynamique
4.Mode de transmission: many to one	4.Mode de transmission: one to one (any to any)
5. Utilisation du broadcasté	5. Communication point à point
6. La mobilité des nœuds est restreinte.	6. Mobilité des nœuds.
7. Grand nombre de nœuds (de l'ordre de mille) nœuds n'ayant pas tous un ID	7. Nombre de nœuds moyen (de l'ordre de cents) Notion d'ID
8. Des petits nœuds plus susceptibles aux pannes, avec moins de capacité de traitement et de stockage.	8. Des nœuds ayant plus de capacité de traitement et de stockage.
9. Nœuds collaborent pour remplir un objectif Commun	9. Chaque nœud a son propre objectif
10. Portée radio est de l'ordre de 40 m	10. Portée radio est dans l'environs de 250 m
11. Energie (facteur déterminant)	11. Débit majeur

Tableau I-1: Similarités et les différences entre (RCSF)

et réseaux Ad-hoc. [7]

I.2.3 Pile protocolaire dans un RCSF

Un RCSF est une série de connexions entre les capteurs leur permettant de communiquer. Le contenu, la portée, la taille, la vitesse et la fiabilité du réseau dépend d'un ensemble de protocoles et de leur implémentation. Les protocoles sont un moyen de communication prédéterminé. Conceptuellement, il est utile de représenter

l'ensemble de ces protocoles sous forme d'une pile, c'est ce qu'on appelle la pile protocolaire. La pile protocolaire [6] utilisée par la station de base, ainsi que tous les autres capteurs du réseau, est illustrée dans la figure 5 Elle comprend la couche application, la couche transport, la couche réseau, la couche liaison de données, la couche physique, le plan de gestion de l'énergie, le plan de gestion de la mobilité et le plan de gestion des tâches.

- **La couche application** : suivant la fonctionnalité des capteurs, différentes applications peuvent être utilisées et bâties sur cette couche.
- **La couche transport** : elle sert à maintenir le flux de données en cas de nécessité dans les applications utilisées, particulièrement lors d'une connexion avec Internet.
- **La couche réseau :** elle s'occupe du routage des données fournies par la couche transport.
- **La couche liaison de données** : comme l'environnement des réseaux de capteurs est bruyant et les nœuds peuvent être mobiles, la couche MAC doit garantir une faible consommation d'énergie et un taux de collision minime entre les données diffusées par les nœuds voisins.
- **La couche physique :** elle s'occupe des techniques d'émission, de réception, démodulation et d'encryptage de données.

Les niveaux de gestion d'énergie, de mobilité et de tâches sont responsables du contrôle de l'énergie consommée, des mouvements des nœuds et de la distribution des tâches à travers toute la pile protocolaire, ces niveaux permettent aux capteurs de coordonner leurs tâches et minimiser la consommation d'énergie.

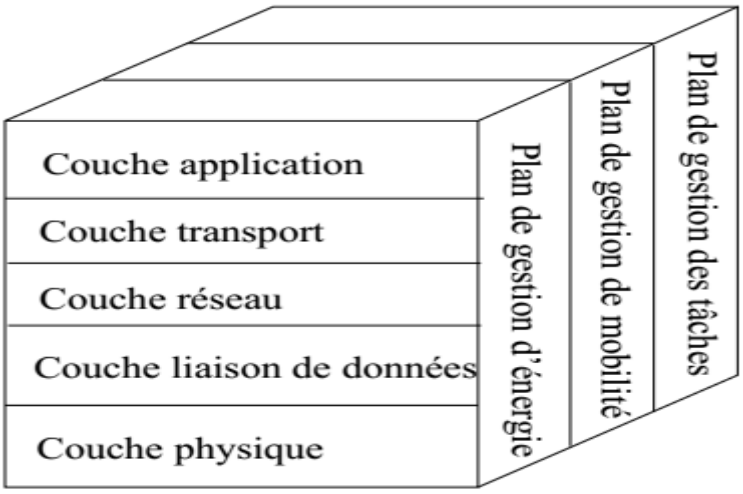


Figure I-4 Pile protocolaire des réseaux de capteurs sans fils [8]

I.3 Domaine d'application

Les RCSF peuvent avoir beaucoup d'applications (voir figure I-6). Parmi elles, nous citons :

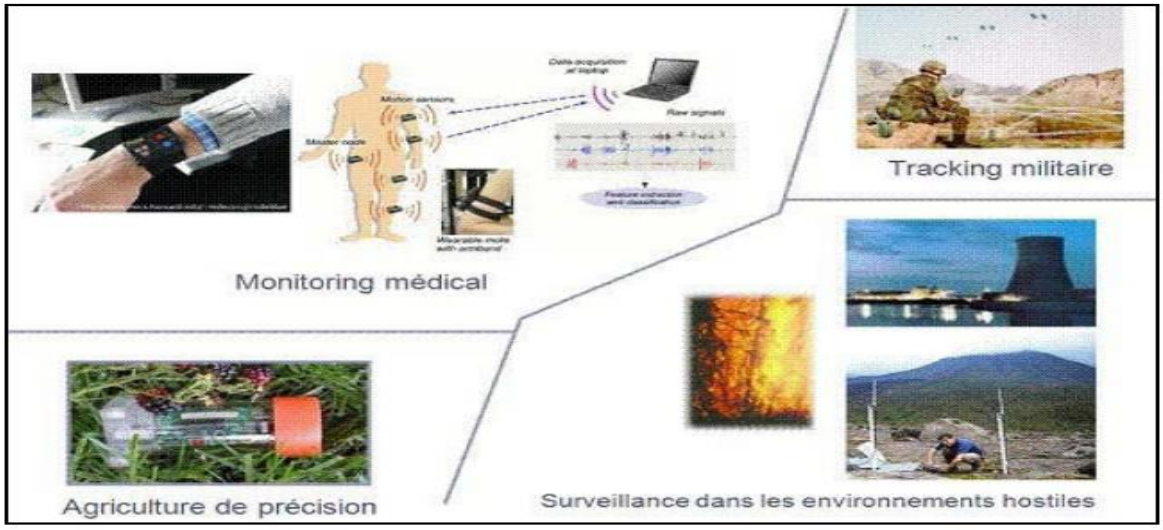


Figure I-5 Quelques domaines d'applications des RCSF [9]

- **Applications médicales**
  Dans le domaine de la médecine, les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers à une étape précoce,). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du cœur, ...etc à l'aide des capteurs ayant chacun une tâche bien particulière. Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient. D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, ...etc) chez les personnes dépendantes (handicapées ou âgées). En outre, des chercheurs américains ont implanté des capteurs à l'échelle nano dans le sang pour détecter quelques maladies à une étape précoce. Comme il existe d'autres applications telles que celles qui permettent la surveillance des lieux contre les intrusions ou les risques. Par exemple, les capteurs placés dans les conduites permettant de transporter le gaz. [2]

- **Applications environnementales**
  Le contrôle des paramètres environnementaux par les réseaux de capteurs peut donner naissance à plusieurs applications. Par exemple, le déploiement des thermo capteurs dans une forêt peut aider à détecter un éventuel début d'incendie et par suite faciliter la lutte contre les feux de forêt avant leur propagation. Le déploiement des capteurs chimiques dans les milieux urbains

peut aider à détecter la pollution et analyser la qualité d'air. De même leur déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.). En outre, dans le domaine de l'agriculture, les capteurs peuvent être utilisés pour réagir convenablement aux changements climatiques par exemple le processus d'irrigation lors de la détection de zones sèches dans un champ agricole. Cette expérimentation a été réalisée par Intel-Research-Laboratory and Agriculture and AgriFood Canada sur une vigne à British Columbia. [10]

- **Applications à la surveillance**
  L'application des réseaux de capteurs dans le domaine de la sécurité peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure. Le déploiement d'un réseau de capteurs de mouvement peut constituer un système d'alarme qui servira à détecter les intrusions dans une zone de surveillance. . [9] [11]

- **Applications militaires**
  Le déploiement rapide, l'auto-organisation et la tolérance aux pannes sont des caractéristiques qui ont rendu les réseaux de capteurs efficaces pour les applications militaires. Plusieurs projets ont été lancés pour aider les unités militaires dans un champ de bataille et protéger les villes contre des attaques, telles que les menaces terroristes. Le projet DSN au DARPA2 était l'un des premiers projets dans les années 80 ayant utilisés les réseaux de capteurs pour rassembler des données distribuées. De même, un réseau de capteurs peut être déployé dans un endroit stratégique ou hostile, afin de surveiller les mouvements des forces ennemies, ou analyser le terrain avant d'y envoyer des troupes (détection des armes chimiques, biologiques ou radiations). L'armée américaine a réalisé des tests dans le désert de Californie. [11]

Chapitre II Les Réseaux de capteurs corporels sans fil

I.2.1 Introduction

L'un des défis majeurs du monde de ces dernières décennies a été l'augmentation continue de la population des personnes âgées et les cas nécessitant des soins de santé quotidiens et une intervention rapide dans le monde. D'où la nécessité de fournir des soins de qualité à une population en croissance rapide, tout en réduisant les coûts des soins de santé. Dans ce contexte, de nombreux travaux de recherche portent sur l'utilisation des réseaux de capteurs sans fil dans les systèmes WBAN, pour faciliter et améliorer la qualité du soin et de surveillance médicale à distance.

I.2.2 Définition de capteur médical

Un capteur est un dispositif ayant pour tâche de transformer une mesure physique observée en une mesure généralement électrique qui sera à son tour traduite en une donnée binaire exploitable et compréhensible pour un système d'information. Un capteur médical se constitue d'un capteur équipe d'un circuit ´électronique spécifique capable de mesurer un ou plusieurs paramètres physiologiques. [12]**.**Donc : capteur + circuit électronique spécifique = capteur médical

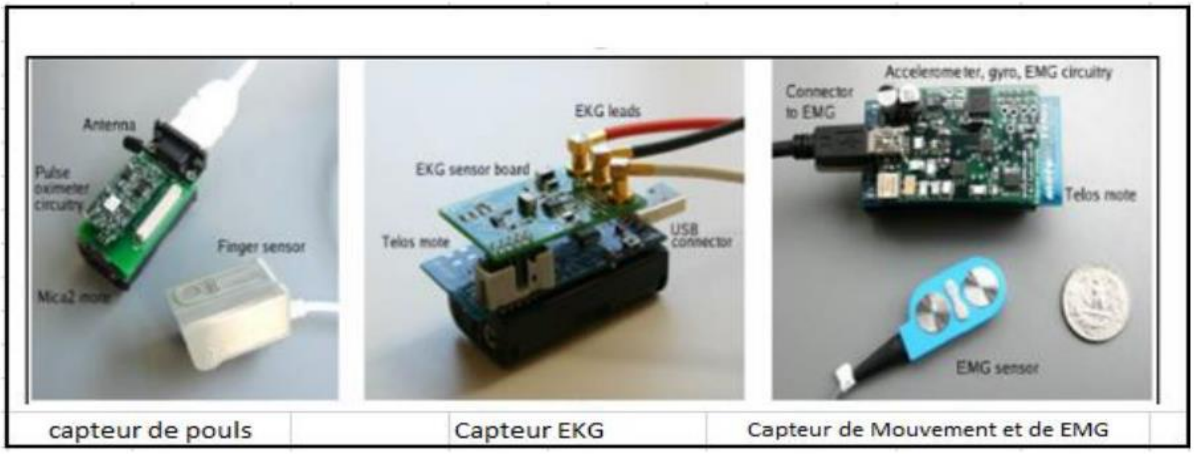


Figure II-1 : Exemple de capteur médical [13]

I.2.3 Architecture des capteurs médical

La Figure 2 représente un nœud-capteur qui est composé de plusieurs éléments ou modules. Chaque module correspond à une tâche particulière de captage et d'acquisition, de traitement ou de transmission de données. Il comprend également une source d'énergie.

Chaque capteur est composé de quatre unités : l'unité d'acquisition des données, l'unité de traitement, l'unité de communication et l'unité de contrôle d'énergie. [14]

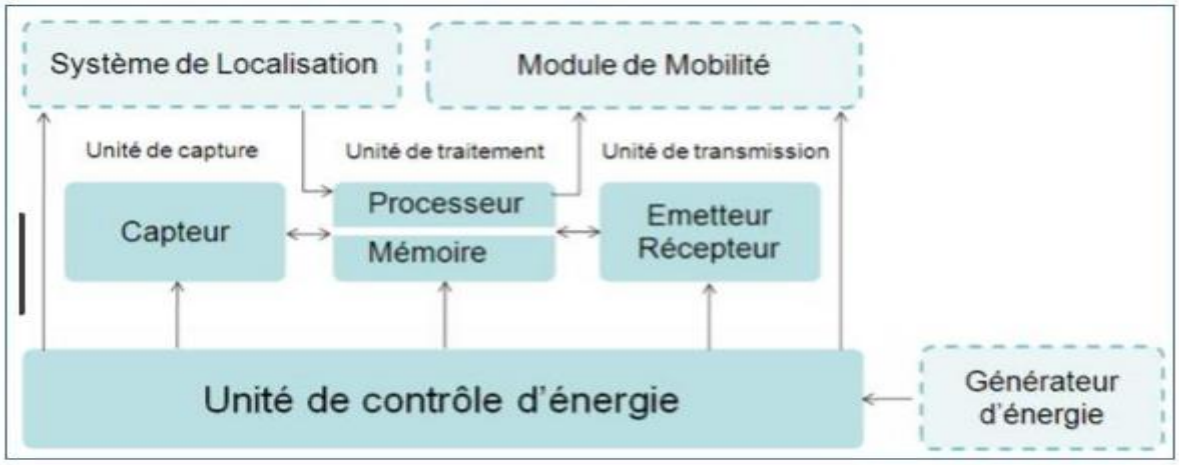


Figure II-2 : architecture d'un capteur médical [15]

I.2.4 Definition de WBAN

Un réseau de capteurs sans fil est considéré comme un type particulier de réseaux ad hoc où des nœuds capteurs couvrent une zone d'intérêt afin de mesurer une grandeur physique ou surveiller un évènement et réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte à l'aide d'une connexion sans fil**.** [16]

I.2.5 Architecture générale d'un réseau WBAN

La Figure 2.1 illustre l'architecture générale d'un système WBAN de surveillance médicale, où plusieurs types de capteurs corporels envoient leurs données mesurées à un serveur par le biais d'une connexion sans fil. Ces données sont ensuite transmises (via Internet, par exemple) à une équipe médicale pour obtenir un diagnostic en temps réel, à une base de données médicale pour être enregistrées, ou bien à un équipement correspondant qui émet une alerte d'urgence.

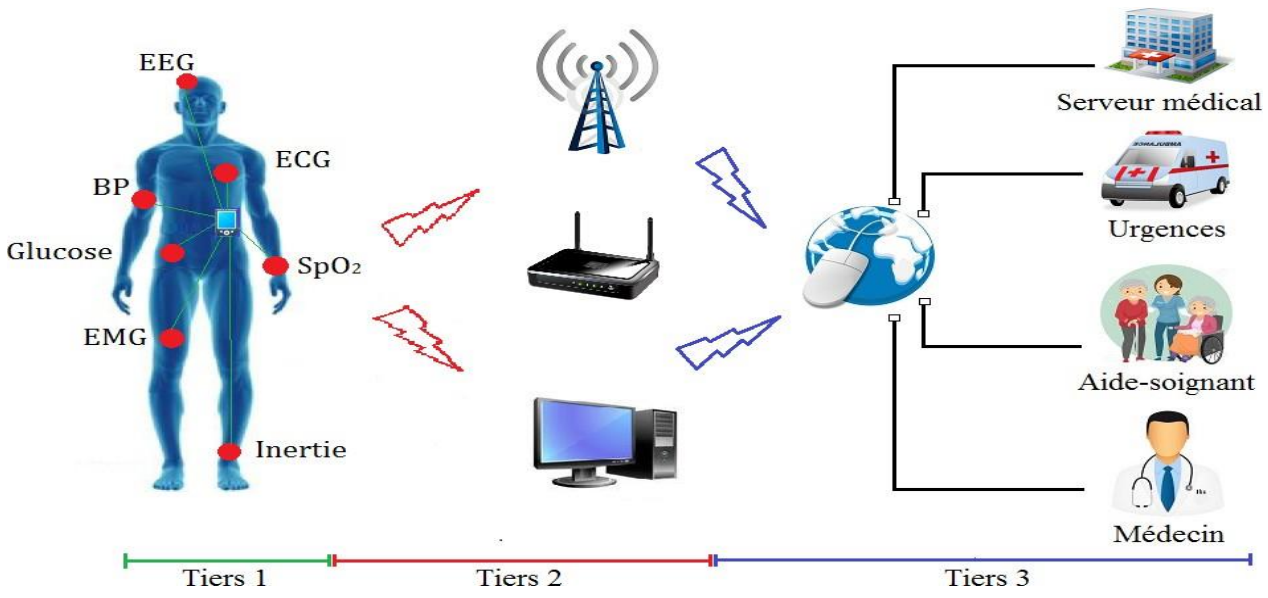


Figure II-3 : Architecture générale d'un système WBAN de surveillance médicale. [17]

L'architecture des communications dans un WBAN peut être décomposée en trois tiers:

- **Tiers 1 : communications intra-WBAN**

Ce tiers concerne les communications se déroulant autour du corps humain, à savoir les communications entre les différents capteurs corporels ainsi que les communications entre les capteurs corporels et le point de collecte (serveur personnel ou nœud coordinateur). Ce dernier peut être un dispositif caractérisé par une puissance de calcul et une réserve d'énergie plus importantes par rapport aux capteurs corporels [18].

- **Tiers 2 : communications inter-WBAN**

Ce tiers se compose des communications entre le point de collecte et un ou plusieurs points d'accès. Les points d'accès peuvent être déployés dans le cadre de l'infrastructure, ou être placés stratégiquement dans un environnement dynamique pour gérer les situations d'urgence. [18]

- **Tiers 3 : communications extra-WBAN**

Ce tiers rassemble les communications entre le point d'accès et l'équipe médicale localisée, par exemple, dans un hôpital et cela via le réseau Internet ou un réseau cellulaire. Les communications extra-WBAN peuvent améliorer la surveillance médicale en permettant aux personnels de la santé (médecins et infirmières) d'accéder à distance aux informations médicales des patients et d'intervenir dans les cas d'urgence. [18]

I.1 Topologies des réseaux WBAN

Dans cette section, nous décrivons les topologies les plus utilisées pour le déploiement des réseaux WBAN. Nous distinguons les topologies suivantes : point-à-point, étoile, et maille. La Figure 4 représente ces trois topologies.

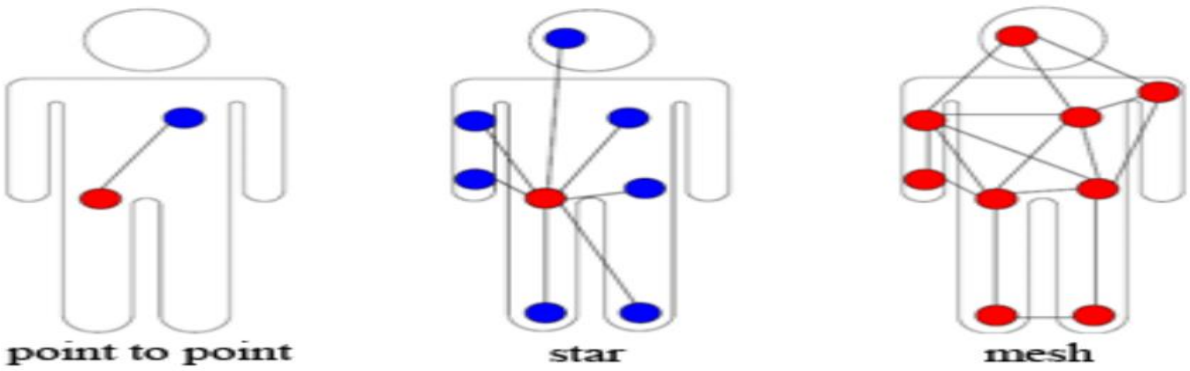


Figure II-4 : Les topologies dans les réseaux WBAN [12]

- **Topologie Point-à-point :**

C'est la topologie la plus simple dans les réseaux. Cette topologie est destinée à une seule liaison, par exemple entre un collecteur de données et un nœud capteur. Le principal avantage de cette topologie est la simplicité qui permet

souvent l'utilisation d'un protocole simple, la faible latence et le débit élevé. Les inconvénients comprennent ses fonctionnalités limitées ainsi que sa faible couverture. [12]

- **Topologie en Etoile :**

  Une topologie dans laquelle tous les nœuds sont connectés par l'intermédiaire d'un nœud central est une topologie en étoile (Star en anglais). Ces nœuds peuvent seulement envoyer ou recevoir un message à ou de l'unique nœud central. Il ne leur est pas permis de s'échanger des messages directement entre eux. Le nœud central joue le rôle d'un relais entre les différents nœuds À ce jour, cette topologie est la plus proposée et utilisé pour les réseaux WBAN. Cette topologie présente des avantages qui peuvent être résumés par la simplicité, la faible consommation d'énergie des nœuds et la moindre latence de communication entre les nœuds et le nœud central. Par contre, son inconvénient majeures la vulnérabilité du nœud central car tout le réseau est géré par un seul nœud. [2]

- **Topologie en Maille** :

  Une topologie avec une connectivité complète entre les nœuds est une topologie maillée (Mesh en anglais). Dans ce cas (dit « communication multi-sauts »), tout nœud peut échanger avec n'importe quel autre nœud du réseau s'il est à portée de transmission. Un nœud voulant transmettre un message à un autre nœud hors de sa portée de transmission, peut utiliser un nœud intermédiaire pour envoyer son message au nœud destinataire. L'avantage d'utiliser la topologie en maille est la possibilité de passer à l'échelle du réseau, avec redondance et tolérance aux fautes et une bonne couverture. Par contre, les inconvénients d'une telle topologie sont l'importante consommation d'énergie induite par la communication multi-sauts ainsi que la latence créée par le passage des messages à travers plusieurs nœuds avant d'arriver au nœud destinataire. L'utilisation d'une topologie maillée est une considération primordiale dans toutes les situations dans lesquelles la fiabilité et la communication flexible sont prioritaires par rapport à l'efficacité énergétique et la durée de vie du réseau. [12]

Topologie	Avantages	Inconvénients
Point-à-point	-Simplicité -Faible latence -Débit élevé	-Fonctionnalités limitées -Faible couverture
Etoile	-Simplicité -Faible consommation d'énergie -Faible latence -Bande passante élevée	-Vulnérabilité du nœud central
Maille	-Redondance -Tolérance aux fautes -Bonne couverture	-Consommation d'énergie importante -Latence élevée

Tableau II-1 : Les avantages et les inconvénients des topologies dans les réseaux WBAN. [12]

I.2 Les contraintes des réseaux WBAN

Un réseau WBAN est un réseau spécial qui a un certain nombre de contraintes par rapport à un réseau informatique classique. Ces contraintes sont le résultat des limitations concernant la mémoire du capteur, sa réserve énergétique, sa capacité de traitement ainsi que l'utilisation d'une communication sans fil. Les contraintes dans un réseau de capteurs sans fil médicaux sont classées en deux catégories : contraintes matérielles et contraintes réseau. Le Tableau 2 résume ces contraintes.

Contraintes matérielles	Contraintes réseau
-Mémoire et espace de stockage limités -Energie Limitée -Capacité de calcul limitée -Faible débit	-Communication incertaine

Tableau *II-2 :* Tableau des contraintes de sécurité dans WBAN.

a) Contraintes matérielles

Toutes les approches de sécurité nécessitent une certaine quantité de ressources pour leur mise en œuvre, dont une mémoire, de l'espace pour le code et de l'énergie pour alimenter le capteur. Toutefois, ces ressources sont très limitées dans un minuscule capteur sans fil notamment dans ceux implantés dans le corps humain. [17]

- **Mémoire et espace de stockage limités**

Un capteur est un petit dispositif avec une mémoire très réduite et un espace de stockage limité. De ce fait, pour construire un mécanisme de sécurité efficace, il est nécessaire de limiter la taille du code de l'algorithme. [17]

- **Énergie limitée**

L'énergie est un autre facteur important à considérer lors de la conception de mécanismes de sécurité. Par exemple, dans le cas de capteurs implantés dans le corps humain, il est très important de limiter la consommation en énergie et de prolonger la durée de vie de la batterie. De ce fait, ajouter des mesures de sécurité a nécessairement un impact significatif sur la consommation en énergie , par exemple, exécuter les fonctions de chiffrement et de déchiffrement, échanger des clés, etc.. [19]

- **Capacité de calcul limitée**

Malgré les progrès réalisés dans la fabrication de capteurs de plus en plus puissants, les capteurs actuels possèdent une capacité de calcul très réduite. Cette faible capacité de calcul ne permet pas d'utiliser des algorithmes complexes, et particulièrement des algorithmes cryptographiques coûteux en ressources CPU. [20]

- **Portée radio limitée et faible débit**

La majorité des capteurs possèdent une portée radio de quelques dizaines de mètres variable selon l'environnement et la fréquence radio utilisée. De plus, le débit actuel dans les réseaux corporels sans fil ne dépasse pas les quelques dizaines de mégabits par seconde. [19]

b) Contraintes réseau

Les communications sans fil sont en général incertaines. En effet, des paquets peuvent être perdus ou endommagés à cause de la transmission radio. Ce manque de fiabilité dans la communication constitue un problème additionnel pour les nœuds capteurs. [19]

I.3 Comparaison entre les réseaux WBAN et les réseaux WSN

Nous présentons ici les différences entre WBAN et WSN qui sont classifiées selon plusieurs facteurs. Le Tableau 3 résume ces différences.

Réseau / Facteur	WBAN	WSN
Déploiement	-Sur le corps humain	Dans des endroits qui ne sont pas facilement accessibles
Densité	-Pas dense	-Dense
Débit	-Actions périodiques	-Actions à des intervalles irréguliers
Latence	-Facilement accessibles -temps de latence réduit	-Difficilement accessibles -temps de latence élevé
Mobilité des nœuds	-Nœuds mobiles	-Nœuds stationnaires

Tableau II-3 : Différences entre WBAN et WSN [12]

I.4 Conclusion

Les WBANs représentent une technologie prometteuse les applications de soins de santé et de divertissements de la prochaine génération. Cependant, ces réseaux apportent un nouvel ensemble de défis notamment en termes de sécurité.

Dans ce chapitre, Après une introduction générale sur les capteurs sans fil, nous avons présenté les WBANs et l'architecture générale de ces réseaux. Enfin, nous avons cité les contraintes dans les WBANs et plus particulièrement dans les applications de surveillance médicale.

Chapitre III

La cryptographie

III.1 Introduction

La cryptographie est une science très ancienne. Des recherches indiquent qu'un scribe égyptien a employé des hiéroglyphes non conformes à la langue pour écrire un message. De ce temps-là et au long de l'histoire, la cryptographie a été utilisée exclusivement à des fins militaires. Aujourd'hui, les réseaux informatiques exigent une phase de cryptographie comme mécanisme fondamental afin d'assurer la confidentialité présentons de l'information numérique. Dans ce chapitre nous présentons les notions de base de la cryptographie.

III.2 Terminologie

- **Texte en clair** : c'est le message à protéger.
- **Texte chiffré** : c'est le résultat du chiffrement du texte en clair.
- **Chiffrement** : c'est la méthode ou l'algorithme utilisé pour transformer un texte en clair en texte chiffré.
- **Déchiffrement** c'est la méthode ou l'algorithme utilisé pour transformer un texte chiffré en texte en clair.
- **Clé** : c'est le secret partagé utilisé pour chiffrer le texte en clair en texte chiffré et pour Déchiffrer le texte chiffré en texte en clair. On peut parfaitement concevoir un algorithme qui n'utilise pas de clé, dans ce cas c'est l'algorithme lui-même qui constitue la clé, et son principe ne doit donc en aucun cas être dévoilé.
- **Cryptographie** : cette branche regroupe l'ensemble des méthodes qui permettent de chiffrer et de déchiffrer un texte en clair afin de le rendre incompréhensible pour quiconque n'est pas en possession de la clé à utiliser pour le déchiffrer.
- **Cryptanalyse** : c'est l'art de révéler les textes en clair qui ont fait l'objet d'un chiffrement sans connaître la clé utilisée pour chiffrer le texte en clair.
- **Cryptologie** : il s'agit de la science qui étudie les communications secrètes. Elle est composée de deux domaines d'étude complémentaires : la cryptographie et la cryptanalyse.
- **Décrypter** : c'est l'action de retrouver le texte en clair correspondant à un texte chiffré sans posséder la clé qui a servi au chiffrement. Ce mot ne devrait donc être employé que dans le contexte de la cryptanalyse.

- **Crypter** : en relisant la définition du mot décrypter, on peut se rendre compte que le mot crypter n'a pas de sens et que son usage devrait être oublié. Le mot cryptage n'a pas plus de sens non plus.
- **Coder, décoder** : c'est une méthode ou un algorithme permettant de modifier la mise en forme d'un message sans introduire d'élément secret.

III.3 Définition Cryptographie

Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre inintelligibles sans une action spécifique. Le verbe crypter est parfois utilisé mais on lui préfèrera le verbe chiffré.

Le fait de coder un message de telle façon à le rendre secret s'appelle chiffrement. La méthode inverse, consistant à retrouver le message original, est appelée déchiffrement.

Le chiffrement se fait généralement à l'aide d'une clef de chiffrement, le déchiffrement nécessite quant à lui une clef de déchiffrement. On distingue généralement deux types de clefs :

- Les clés symétriques : il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.

- Les clés asymétriques : il s'agit de clés utilisées dans le cas du chiffrement asymétrique (aussi appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement [21] [22]

III.4 Algorithmes de la cryptographie

Un algorithme de cryptographie ou un chiffrement est une fonction mathématique utilisée lors du processus de cryptage et de décryptage. Cet algorithme est associé à une clef (un mot, un nombre, ou une phrase). Afin de crypter une donnée avec des clés différentes le résultat du cryptage variera également. La sécurité des données cryptées repose entièrement sur deux éléments : l'invulnérabilité de l'algorithme de cryptographie et la confidentialité de la clef. [22]

Un système de cryptographie est constitué d'un algorithme de cryptographie, ainsi que de toutes les clefs et tous les protocoles nécessaires à son fonctionnement.

III.4.1 Algorithmes symétriques (clef secrète)

Un algorithme symétrique est un algorithme qui permet de transformer un texte en clair en texte chiffré en utilisant une clé et de retransformer le texte chiffré en texte en clair en utilisant la même clé.

Le secret de la communication est uniquement assuré par la clé qui est utilisée lors de la phase de chiffrement et de déchiffrement. L'algorithme utilisé ne fait pas partie du secret.

On parle d'algorithmes symétriques car c'est la même clé qui sert à la fois au chiffrement et au déchiffrement du message. [23]

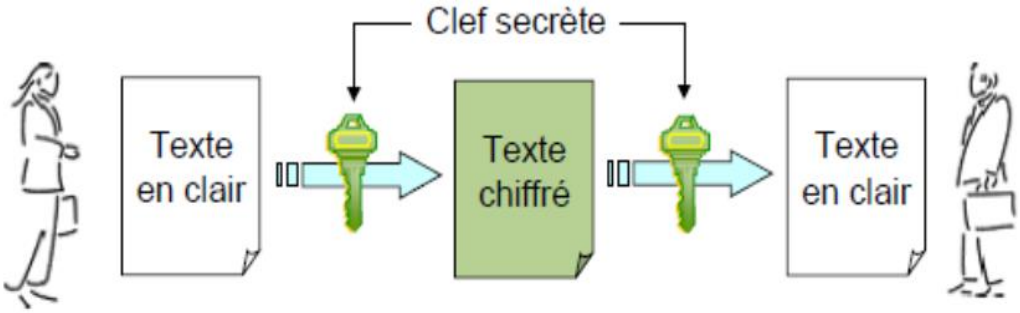


Figure III-1 : Principe de l'algorithme symétrique [21]

Les algorithmes symétriques sont de deux types :

- Les algorithmes de chiffrement en continu, qui agissent sur le message en clair un bit à la fois.
- Les algorithmes de chiffrement par bloc, qui opèrent sur le message en clair par groupes de bits appelés bloc.

I. Algorithmes de chiffrement en continu

Qui opèrent sur le message en clair un bit à la fois. Le principe consiste à générer un flux pseudo aléatoire et de le combiner avec l'information bit à bit par l'opération XOR. A la réception, on applique le même mécanisme, et on restitue l'information. [23]

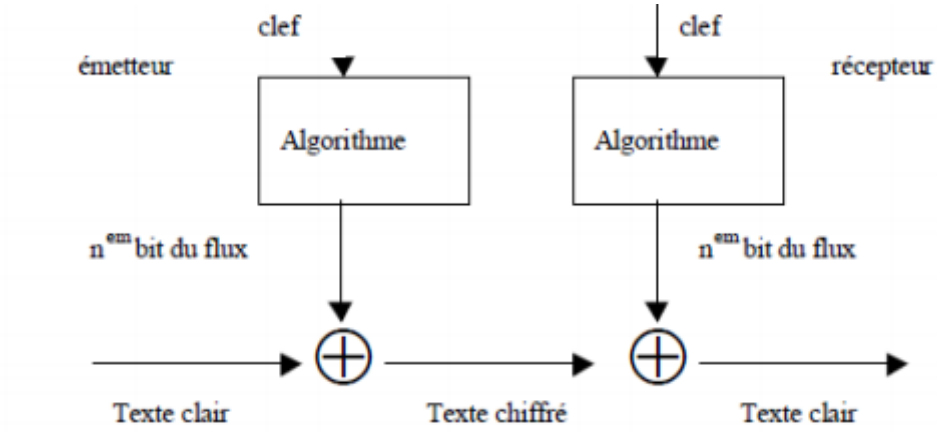


Figure III-2 : Chiffrement en continu [23]

II. Algorithmes de chiffrement par bloc

Qui opèrent sur le message en clair par groupe de bit. La taille typique des blocs est 64 bits, ce qui est assez grand pour interdire l'analyse et assez petit pour être pratique. [2]

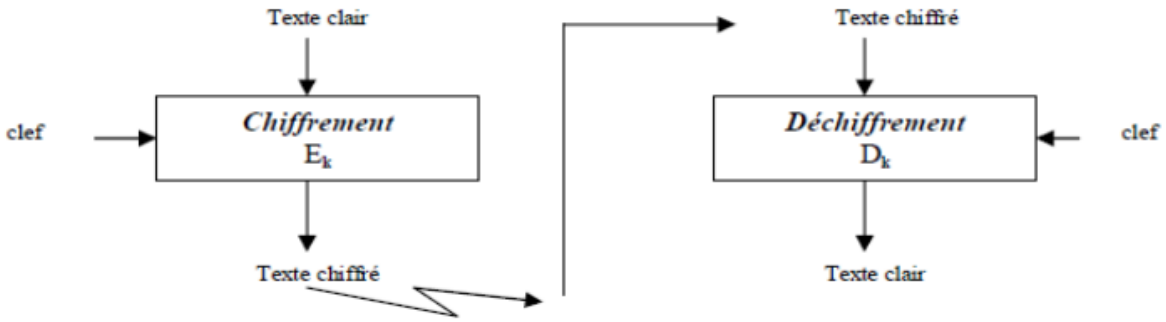


Figure III-3 : Chiffrement par bloc [24]

III. Exemple algorithmes symétrique

- Chiffrement par bloc

		DES	3DES	IDEA	RC4	RC5/6	Blowfish	AES
Nom réel		**Data Encryption Standard**	**Triple Data Encryption Standard**	**International Data Encryption Algorithm**	**Rivest Cipher 4**	**Rivest Cipher 5/6**	**Blowfish**	**Advanced Encryption Standard**
Date		**1973**	**1978**	**1992**	**1987**	**1994**	**1993**	**1998**
LONGEUR	**Clé**	**64 bits (56 effectifs)**	**192 bits (168 effectifs)**	**128 bits**	**jusqu'a 256 bits**	**entre 0 et 2040 bits**	**entre 40 et 448 bits**	**128, 192, 256 bits**
	Bloc	**64 bits**	**64 bits**	**64 bits**	**Flux**	**32, 64, 128 bits**	**64 bits**	**128 bits**

Tableau 4-1 : chiffrement par bloc

III.4.2 Algorithmes asymétriques (clef publique)

Les algorithmes symétriques vus sont tous fiables mais ils posent un problème, c'est celui de l'échange de la clé : comment transmettre de manière fiable à mon interlocuteur la clé de chiffrement utilisée pour chiffrer le message que je lui envoie ? Il y a bien sûr le téléphone, mais il y a aussi les écoutes téléphoniques.

Les algorithmes asymétriques ont été inventés pour pallier précisément le problème de transmission sécurisée de la clé.

On parle d'algorithmes asymétriques car ce n'est pas la même clé qui sert au chiffrement et au déchiffrement. Dans le cas de ces algorithmes, on parlera alors de clé privée et de clé publique. Ces deux clés, clé privée et clé publique, sont intimement liées par une fonction mathématique complexe. [23]

Les algorithmes asymétriques possèdent 2 modes de fonctionnement :

- Le mode chiffrement dans lequel l'émetteur chiffre un fichier avec la clé publique du destinataire pour chiffrer. Le destinataire utilise sa clé privée pour déchiffrer le fichier.
  Dans ce mode, l'émetteur est sûr que seul le destinataire peut déchiffrer le fichier.

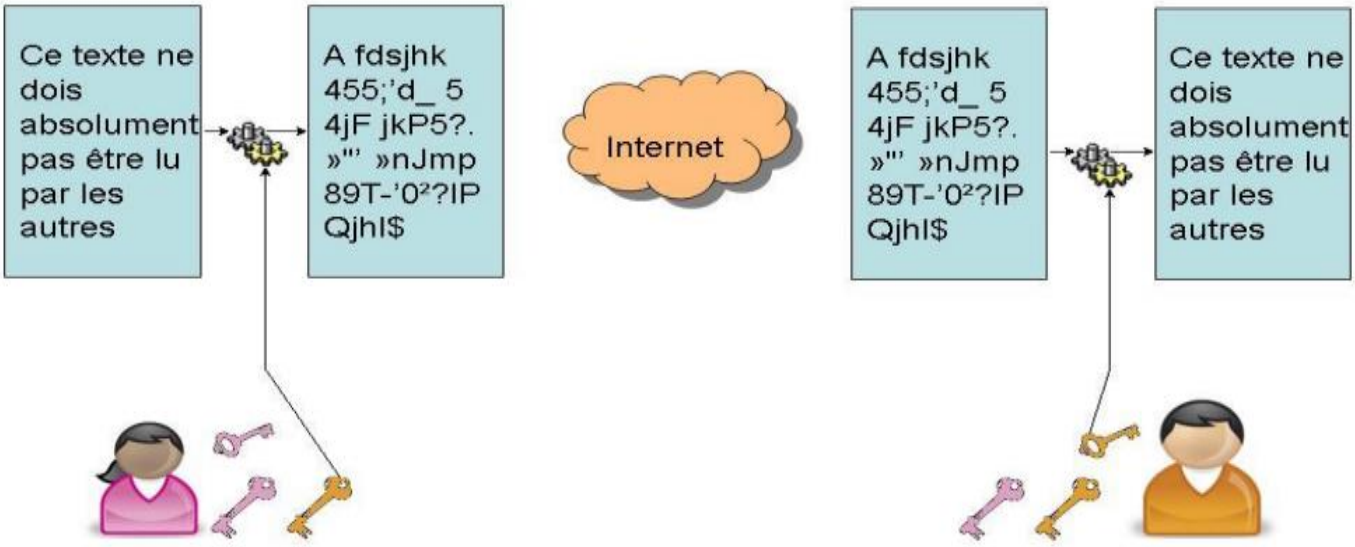


Figure III-4 : Chiffrement avec l'algorithme asymétrique. [25]

- Le mode signature dans lequel l'émetteur signe un fichier avec sa propre clé privée. Le destinataire utilise la clé publique de l'émetteur pour vérifier la signature du fichier. Dans ce mode, le destinataire est sûr que c'est bien l'émetteur qui a envoyé le fichier.

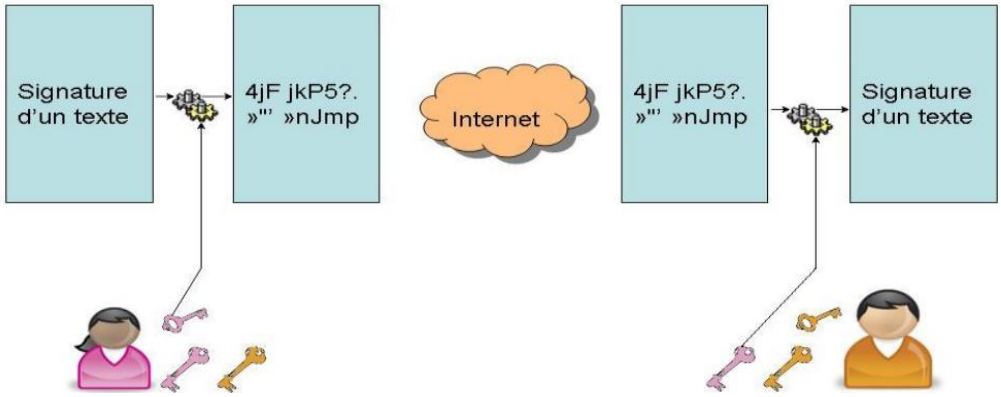


Figure III-5 : Signature avec l'algorithme asymétrique. [25]

Exemple algorithmes asymétrique

Quelques algorithmes de cryptographie asymétrique très utilisés :

- **RSA** (chiffrement et signature).
- **DSA** (signature).
- **Protocole d'échange de clés Diffie-Hellman** (échange de clé) .
- **ECC :**La cryptographie basée sur les courbes elliptiques (ECC) a été proposée dans les années 80. Récemment, ECC a attiré beaucoup l'attention des chercheurs en raison de son exigence de clés de plus courte taille comparativement à d'autres techniques de cryptographie à clé publique à l'instar de RSA en particulier dans le domaine des systèmes embarqués où les dispositifs ont une puissance de calcul très limitée. Dans, ont démontré qu'ECC peut atteindre le même niveau de sécurité en utilisant une clé plus courte que celle de RSA (une clé de 160 bits dans ECC est équivalente à une clé de 1024 bits dans RSA).

III.5 Fonction de hachage

Une fonction de hachage est une fonction permettant d'obtenir un condensé d'un texte c'est-à-dire une suite de caractères assez courte représentant le texte qu'il condense.

La fonction de hachage doit être telle qu'elle associe un et un seul haché à un texte en clair). D'autre part, il doit s'agir d'une fonction à sens unique (one-way function) afin qu'il soit impossible de retrouver le message original à partir du condensé. La figure suivante nous résume le tout. [25]

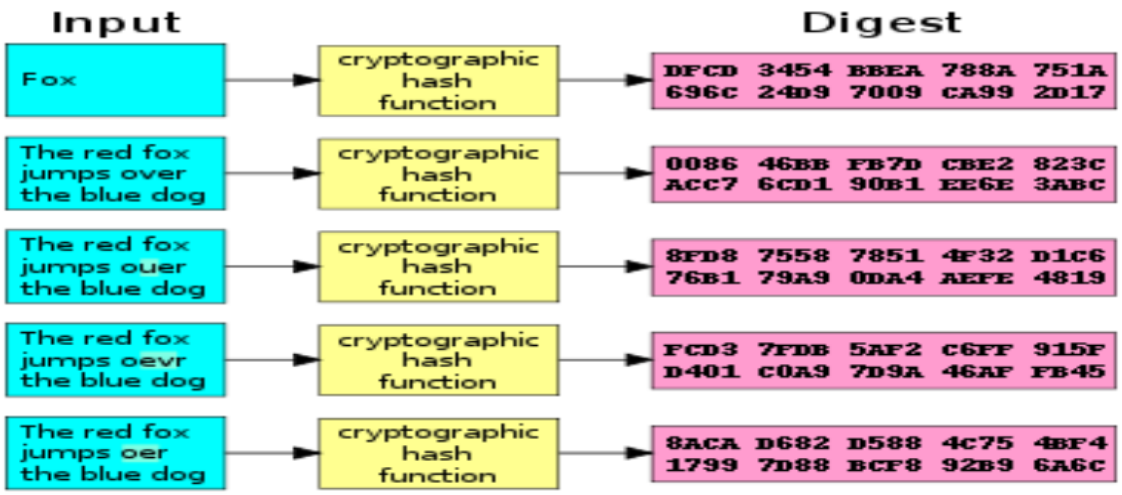


Figure III-6 : Fonction de hachage [25]

Parmi les fonctions de hachage usuelles, on trouve :

- **MD4 et MD5** (Message Digest) furent développées par Ron Rivest. MD5 produit des hachés de 128 bits en travaillant les données originales par blocs de 512 bits. Mais au cours de temps des failles on été découvertes, les calculs prend beaucoup de temps, le md5 n'est donc considéré comme sur ainsi il provoque des débordements de mémoire dans les RCSF.
- **SHA-1 , SHA-2** : (Secure Hash Algorithm 1), comme MD5, il est basé sur MD4. Il fonctionne également à partir de blocs de 512 bits de données et produit par contre des condensés de 160 bits en sortie. Le SHA-2et est destiné à remplacer SHA-1. Les différences principales résident dans les tailles de hachés possibles : 256, 384 ou 512 bits. Il sera bientôt la nouvelle référence en termes de fonction de hachage. Mais l'inconvénient Il nécessite plus de ressources que MD5, alors plus de consommation d'énergie et gaspillage du temps, comme il nous offre qu'une image sur le haché.
- **Base64** : est un résumé de l'information utilisant 64 caractères, sélectionnés pour être disponibles sur la majorité des tables de caractères hachage utilise les valeurs binaires. Base 64 consiste à découper le message en groupe de 6 bits (on complète avec des 0 si besoin). Chaque groupe de 6 bits a une valeur en base 10, on y associe le caractère de même rang dans l'alphabet. Base64 permet d'éviter la surcharge et d'éviter le temps de traitement très long.

III.6 Cryptage symétrique vs cryptage asymétrique

Cryptage symétrique :	**Cryptage asymétrique :**
-Chiffrement à clé privé (utilisation une clé pour crypter qui fonctionne aussi pour décrypter -Très facile -Très rapide -les clés de chiffrement symétrique doivent être conservées en toute sécurité - vous devez vous assurer que chaque personne qui a besoin de la clé, il obtient sans aucun risque de le sortir.	-Chiffrement à clé publique (utilisation deux clés un pour crypter **clé publique** et autre pour décrypter **clé privée** -Difficile par rapport au cryptage symétrique -Plus lent -les clés publiques qu'ils utilisent sont sans danger pour être publié n'importe où parce que pour obtenir la clé privée à partir d'une clé publique peut prendre des centaines d'années de travail.

Tableau III-2 : Cryptage symétrique vs cryptage asymétrique

III.7 Conclusion

Dans ce chapitre nous avons présenté une introduction générale sur la cryptographie, nous avons distingué deux classes importantes des méthodes de chiffrement, c'est le cryptage symétrique à clé secrète et le cryptage asymétrique a clé publique. Nous avons aussi montré la puissance et la faiblesse de chaque type d'algorithme de chiffrement. Dans le chapitre suivant, nous allons présenter les outils AVISPA, qui sont utilisés pour valider la sécurité des protocoles d'authentification étudiés.

Chapitre IV
La mise en œuvre

Introduction

Les réseaux WBANs utilisés dans le domaine de santé, pour assurer une collecte de données, ces données généralement sont partagées entre plusieurs utilisateurs, de ce fait un contrôle d'accès est indispensable, car ces données vont être exploitées afin de prendre des interventions nécessaires.

Donc avant de permettre l'accès aux données sensibles et privées des patients en temps réel, l'utilisateur externe (médecin) doit être authentifié. Dans ce chapitre nous avons expliquons deux protocoles de sécurité et les analysons, puis faire une vérification via l'outil AVISPA.

Protocol N° 1 [26]

1. PRINCIPE

C'est un système ayant pour objectif d'assurer une communication sécurisée entre tout le composant du WBAN. Ce système utilise une authentification d'utilisateur par mot de passe et groupe basée sur les droits d'accès fournis aux utilisateurs authentiques dans les réseaux WBAN. Et Il offre une meilleure sécurité par rapport aux autres systèmes de contrôle d'accès associés, car il prend en charge l'authentification mutuelle entre l'utilisateur, la station de base et le nœud du capteur, résiste aux attaques de type "dos privilégié", "privilégié-initié", "violation de la carte à puce" et "man in the middle". .

Il prend en charge les ajouts de nœuds dynamiques après le déploiement initial de nœuds sur le réseau. Il prend également en charge le déploiement de nouveaux nœuds pour les nouveaux patients et ne nécessite pas d'informations actualisées provenant de la carte à puce de l'utilisateur. Il prend en charge une modification locale du mot de passe de l'utilisateur sans l'aide de la station de base (serveur médical). Il établit une clé de session secrète entre l'utilisateur et un nœud de capteur afin que la même clé puisse être utilisée pour la communication secrète future de données en temps réel à l'intérieur du réseau WBAN. En vérifiant la sécurité de manière formelle à l'aide de l'outil AVISPA, nous montrons que le protocole est également sécurisé contre les attaques passives et actives telles que la relecture et les manœuvres "man in the middle".

2. LES PHASES de protocole

Le protocole comprend les phases suivantes :

- **Pre-deployment,**
- **post-deployment,**
- **registration,**
- **login,**
- **authentication,**
- **password change**
- **dynamic node addition**

Ces phases sont décrites dans les sous-sections suivantes, Mais avant les détailler il faut donner les notations suivantes (voir tableau1)

Symbole	Description
Sn_i	Identifiant du nœud de capteur i
U_j	$j^{ième}$ utilisateur
BS	Station de base
PW_j	Mot de passe de l'utilisateur U_j
G_{idj}	Identifiant de groupe de l'utilisateur U_j
APM_j	Masque de privilège d'accès de l'utilisateur U_j
X	Clé privée de la station de base
K_{BS}	Clé publique de la station de base
MK_{Si}	Clé principale du nœud de capteur SN_i
RM_{uj}	Nombre aléatoire pour l'utilisateur U_j
K_i	Clé secrète du nœud SN_i partagée avec la station de base.
H(.)	Fonction de hachage sécurisée unidirectionnelle
T_i	Temps d'amorçage pour le nœud SN_i
A||B	Les données A sont concaténées avec les données B
$E_K(M)$	Cryptage symétrique à l'aide de la clé K
$D_K(M)$	Décryptage symétrique à l'aide de la clé K
X →Y:M	L'entité X envoie le message M à l'entité Y

Tableau 1 : Notations utilisées dans le protocole

1. **PRE-DEPLOYMENT** :

   Cette phase sert à pré charger les supports de chiffrement sur tous les nœuds de capteurs avant leur déploiement. Cette opération est effectuée hors ligne par le serveur d'installation des clés. Le serveur de configuration est la station de base (le serveur médical) Cette phase est mise en œuvre hors ligne par la station de

base avant le déploiement des nœuds de capteur sur le corps du patient (champ cible). [26]

2. **POST-DEPLOYMENT,**

Cette phase aide les nœuds de capteur et la station de base à établir des connexions sécurisées entre eux. Dès qu'ils sont déployés, leur première tâche consiste à localiser les voisins physiques dans leurs limites de communication. Pour une communication sécurisée entre les nœuds de capteur, ils doivent établir des clés secrètes par paire entre eux. Puisque la base est de résoudre le problème du contrôle d'accès des utilisateurs, ils supposent que les nœuds d'un réseau WBAN peuvent établir des clés secrètes à l'aide des schémas d'établissement de clés existants. Par exemple, nous pouvons utiliser un schéma d'établissement de clé sécurisé sans condition [27]pour l'établissement de clé par paire entre les nœuds de chaque cluster.

Pour atteindre les résultats souhaités de permettre aux utilisateurs autorisés appartenant à différents groupes (médecins, infirmières, équipe d'assurance maladie, patients, etc.) d'avoir accès aux données en temps réel permettant de surveiller l'état d'un patient à partir des capteurs situés à l'intérieur du réseau WBAN, ils ont communication entre les nœuds de capteurs et les utilisateurs autorisés. [26]

3. **REGISTRATION,**

Dans la phase d'enregistrement, un utilisateur doit s'inscrire auprès de la station de base pour accéder aux données en temps réel d'un nœud de capteur spécifique dans un réseau WBAN. L'utilisateur identifie son identifiant, son mot de passe et son identifiant de groupe, ainsi qu'un numéro aléatoire.

Après réception de l'information, la station de base calcule la valeur de hash partagée et secrète pour l'utilisateur.La station de base génère enfin une carte à puce inviolable et l'envoie à utilisateur via un canal sécurisé et stocke le information de l'utilisateur [26] . Cette phase d'enregistrement est résumée Par l'algorithme de la Figure1

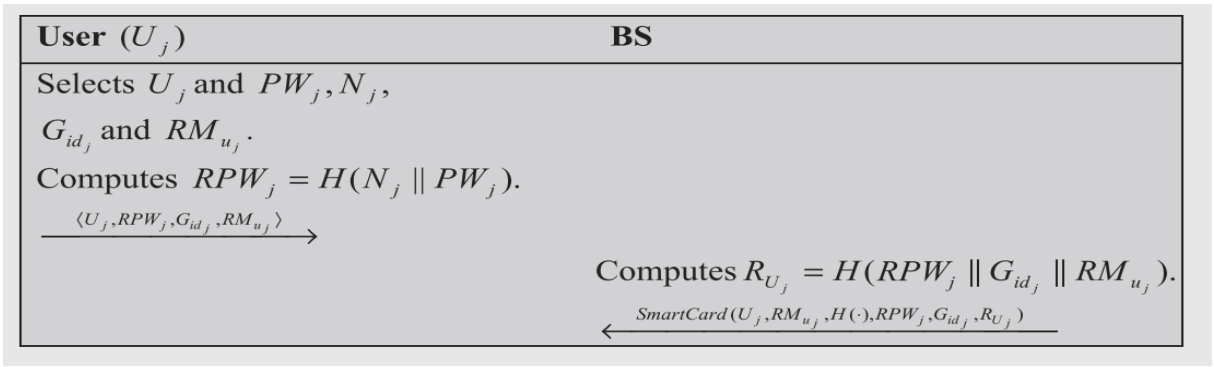


Figure 1: La phase d'inscription [26]

4. AUTHENTIFICATION

L'authentification des capteurs est nécessaire pour s'assurer que l'identité déclarée par un utilisateur est bien celle du déclarant. En l'absence d'un mécanisme permettant d'authentifier clairement un nœud du réseau, de nombreuses attaques peuvent se mettre en place

Pour cela, ce protocole effectue plusieurs opérations, Le plus important :

- l'utilisateur calcule la valeur chiffrée et la valeur de hachage, en envoyant le message de demande d'authentification suivant au nœud de capteur SNi.

-Après avoir reçu le message de demande d'authentification de l'utilisateur, le nœud de capteur SNi effectue plusieurs opérations pour vérifier si l'utilisateur est légitime

-Si la vérification a succès, le nœud de capteur vérifie la signature.

-Si la vérification de la signature échoue, le nœud de capteur considère l'utilisateur comme illégal et la phase se termine immédiatement. Le nœud de capteur envoie un accusé de réception à l'utilisateur et à la station de base, et répond à la requête de l'utilisateur,

-Sinon, la phase d'authentification se termine immédiatement. [26]

La Figure 2 représente l'algorithme de la phase authentification

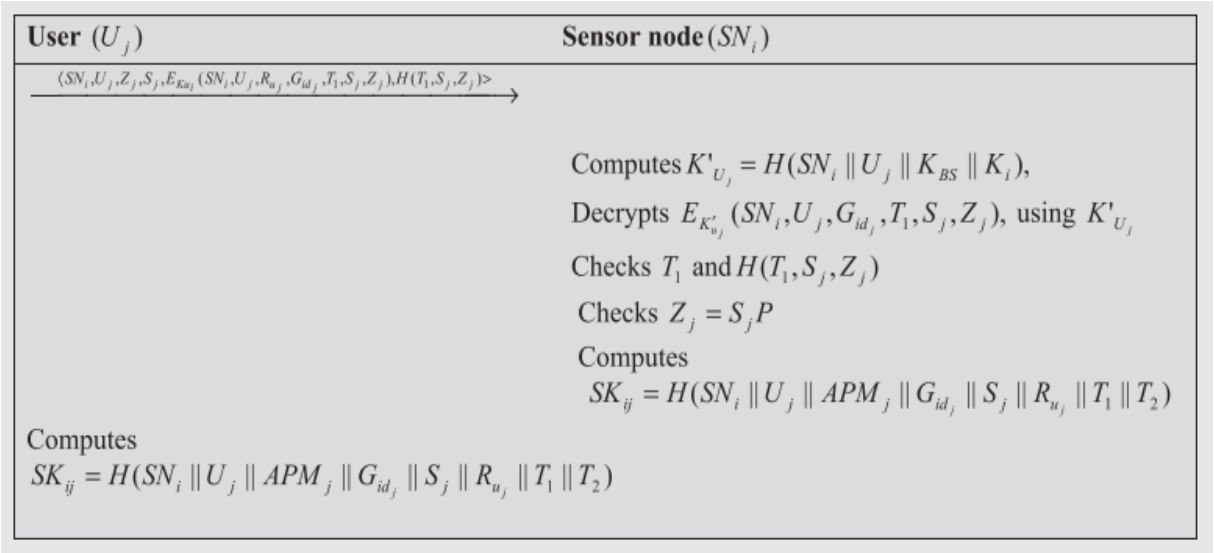


Figure 2 : La phase authentification [26]

5. **PASSWORD CHANGE**

   Dans cette phase, un utilisateur peut changer de mot de passe librement et complètement localement pour des raisons de sécurité sans contacter le Station de base. [26]

6. **DYNAMIC NODE ADDITION**

   Le déploiement de nouveaux nœuds dans les réseaux de capteurs est inévitable en raison de la perte de nœuds de capteurs résultant de l'épuisement de la puissance après des semaines ou des mois de fonctionnement. Certains nœuds peuvent être compromis et nécessiter un remplacement. [26]

3. Analyse du protocole

Dans cette section, nous effectuons des analyses de fonctionnalité et de sécurité de protocole

Vérification formelle de sécurité du protocole avec AVISPA back-ends

Nous avons implémenté le protocole N°1 dans le langage HLPSL. Dans cette implémentation, nous avons trois rôles de base, le nœud de capteur SNi, la station de base et l'utilisateur Uj. [28]

Nous avons également défini la session et l'environnement en protocole.

1. Rôle Utilisateur

Le tableau2 illustre la spécification du rôle de l'utilisateur Uj dans HLPSL.

Nous écrivons le rôle d'utilisateur Dans la partie de déclaration, nous définissons le nœud "utilisateur", "serveur", "serveur "et leur Propriété. Pendant la phase d'enregistrement, Uj envoie le message « Uj, RPWj, Gidj, RMj » sécurisée à la station de base par l'opération Snd () [26] [29]. Le canal de déclaration de type (dy) indique le canal pour le modèle de menace Dolev–Yao. Uj attend alors que la carte à puce contenant les informations sécurisées dans le message « Uj, RMuj, H(.), RPWj, Gidj, Ruj » de la BS à partir de l'opération Rcv ().

L'intrus aura la capacité d'intercepter, d'analyser et / ou de modifier les messages transmis sur le canal non sécurisé. Pendant la phase de connexion, Uj envoie le message de demande de connexion à la station de base. En réponse, la station de base envoie la réponse par message à Uj. Au cours de la phase d'authentification, Uj envoie enfin le message de demande d'authentification au nœud de capteur SNi. Voir la table

```
role user (U,BS,SN :agent,
        MKsi : symmetric_key,
        MKuj : symmetric_key,
        H    : hash_func,
        Snd,Rcv:channel(dy))
played_by U
def=local State : nat,
Uj,RPWj,APMj,RMuj,Nj,Pwj,UKj: text,
Ruj,Kuj,SNi,Sj,Zj,Ki,Kbs,GIdj,RNui:text,
T1,T2:text
const snode_server,server_user, user_server, user_snode,
subs1,subs2,subs3,subs4,subs5,subs6: protocol_id
init state :=0
trasition
1.State =0/\Rcv(start)=|>
 State':=1/\RPWj':=H(PWj.Nj)
          /\RMuj':=new()
          /\Snd(U.BD.{Uj.RPWj'.GIdj.RMuj'}_MKuj)
2.State=1/\Rcv(BS.U.{Uj.GIdj.H(H(PWj.Nj).GIdj.RMuj')
.H.H(PWj.Nj)}_MKuj)=|>
State':=2/\secret ({Ki},subs1,{SN,BS})
        /\secret ({MKsi},subs2,{SN,BS})
        /\secret ({RMuj'},subs3,{U,BS})
        /\secret ({Kbs},subs4,{SN,BS})
        /\secret ({APMj,GIdj},subs5,{U,BS})
        /\secret ({PWj,Nj},subs6,U)
        /\T1':=naw()
/\Snd(U.BS.Uj.H(H(H(PWj.Nj).GIdj.RMuj).T1').T1')
        /\witness(U,BS, user_server,T1')
3.State=2/\Rcv(BS.U.{Sj.Zj.SNi.H(SNi.Uj.Kbs.Ki)}_H(H(H(PWj.Nj).GIdj.RMuj').Uj.T1'.T2').T2'.T1')=|>
State':=3/\UKj':=H(Ruj.Uj.T1'.T2')
  /\Snd(U.SN.SNi.Uj.Zj.Sj.{SNi.Uj.Ruj.GIdj.T1'.Sj.Zj}_Kuj'.H(T1'.Sj.Zj))
        /\witness(U,SN, user_snode,T1')
end role
```

Tableau 2 : la spécification du rôle de l'utilisateur Uj .

2. Rôle station de base

Nous écrivons le rôle server Dans la partie de déclaration la même chose avec le role précèdent partie,et dans la partie transition le serveur "station de base" au cours de la phase post-déploiement, il reçoit un message du nœud de capteur SNi ce message qui contient les informations nécessaires pour établi une connexion entre les nœuds et station de base, puis passe à la phase d'enregistrement.

Pendant cette phase après la réception sécurisée du message de l'utilisateur Uj, la station de base envoie de manière sécurisée la carte à puce contenant les Informations confidentielles "les clés, la valeur de hash " contenues dans le message à l'utilisateur Uj.

Dans la phase de connexion, lorsque la BS reçoit un message de l'utilisateur Uj, elle envoie les messages à Uj et le nœud de capteur SNi. Le tableau3 Montre la spécification du rôle de la station de base dans le langage HLPSL.

```
role server (BS,SN,U :agent,
            MKsi : symmetric_key,
            MKuj : symmetric_key,
            H    : hash_func,
            Snd,Rcv:channel(dy))
played_by BS
def=
local State : nat,
RPWj,RMuj,Ruj,Kbs,Kuj,Sj,Zj,T2,APMj,GIdj,Nj,Pwj,UKj:text,
SNj,Uj,Ki,Ti,T1,M3:text
const snode_server,snode_user,user_server,user_snode,
subs1,subs2,subs3,subs4,subs5,subs6: protocol_id
init state :=0
transition
1.State =0/\Rcv(SN.BS.SNi.Ti.{Ki.SNi.Ti}_MKsi)=|>
 State':=1/\Kuj':=H(SNi.Uj.Kbs.ki)

2.State=1/\Rcv(U.BS{Uj.H(PWj.Nj).GIdj.RMuj'}_MKuj)=|>
State':=2/\Snd(BS.U.{Uj.GIdj.H(H(PWj.Nj).GIdj.RMuj').H.H(PWj.Nj)}_MKuj)
            /\secret ({Ki},subs1,{SN,BS})
            /\secret ({MKsi},subs2,{SN,BS})
            /\secret ({RMuj'},subs3,{U,BS})
            /\secret ({Kbs},subs4,{SN,BS})
            /\secret ({APMj,GIdj},subs5,{U,BS})
            /\secret ({PWj,Nj},subs6,U)
            /\request(SN,BS,snode_server,Ti)

3.State=2/\Rcv(U.BS.Uj.H(H(H(PWj.NJ).GIdj.RMuj').T1').T1')=|>
State':=3/\M3':=xor(APMj,GIdj)
            /\T2':=new ()
             /\Snd(BS.U.{Sj.Zj.Kuj.SNi}_UKj.T2'.T1')
            /\Snd(BS.SN.SNi.Uj.{SNi.Uj.GIdj.T1'}_Ki)
            /\witness(BS,SN,snode_server,T2')
            /\request(U,BS,user_server,T1')
end role
```

Tableau 3 : la spécification du rôle de la BS

3. Rôle de nœud de capteur

Nous avons implémenté la spécification de rôle pour le nœud de capteur SNi dans le langage HLPSL. Nous écrivons le rôle de nœud de capteur SNi "sensor node" Dans la partie de déclaration la même chose avec les rôles précède, et dans la partie transition, commence par la phase post-déploiement, le nœud de capteur SNi envoie un message à la BS pour la définition et établi une connexion. Pour cela pendant la phase de connexion, le nœud de capteur SNi reçoit un message de la station de base, pour Passer à la phase suivante, la phase d'authentification, le nœud capteur reçoit le message de demande d'authentification de l'utilisateur Uj.

Witness (A, B, ID, E) déclare pour une propriété d'authentification (faible) de A par B sur E, déclare que l'agent A est témoin de l'information E;

Cet objectif sera identifié par l'identifiant constant dans la section des objectifs. Requeté (B, A, ID, E) demande une propriété d'authentification forte de A par B sur E, déclare que

l'agent B demande une vérification de la valeur E; cet objectif sera identifié par l'identifiant constant dans la section des objectifs. L'intrus est toujours désigné par i.

Le tableau4 Montre la spécification du rôle le nœud de capteur SNi dans le langage HLPSL.

```
role snode (SN,BS,U :agent,
        MKsi : symmetric_key,
        H    : hash_func,
        Snd,Rcv:channel(dy))
played_by SN
def=
local State : nat,
SNi,Ti,Ki,Kbs:text
Uj,APMj,GIdj,RPWj,RMuj,T1,T2,Sj,Zj,Ruj,Kuj,Nj,Pwj,UKj: text
Const snode_server,user_server,snode_user,user_snode,
subs1,subs2,subs3,subs4,subs5,subs6: protocol_id
init state :=0
trasition
1.State =0/\Rcv(start)=|>
 State':=1/\Ti':=new()
          /\secret ({Ki},subs1,{SN,BS})
          /\secret ({MKsi},subs2,{SN,BS})
          /\secret ({RMuj},subs3,{U,BS})
          /\secret ({Kbs},subs4,{SN,BS})
          /\secret ({APMj,GIdj},subs5,{U,BS})
          /\secret ({PWj,Nj},subs6,U)
          /\Snd(SN.BS.SNi.Ti.{Ki.SNi.Ti}_MKsi)
          /\witness(SN,BS,snode_server,Ti')
2.State=1/\Rcv(BS.SN.SNi.Uj.{SNi.Uj.xor(APMj,GIdj).H(H(PWj.Nj).GIdj.RMuj').T1'.T2'}_MKsi{SNi.
Uj.GIdj.T1'}_Ki)=|>
State':=2/\request(BS,SN,snode_server,T2')
3.State=2/\Rcv(U.SN.SNi.Uj.Zj.Sj.{SNi.Uj.H(H(PWj.Nj).GIdj.RMuj').GIdj.T1'.Sj.Zj}_H(SNi.Uj.Kbs.Ki).
h(T1'.Sj.zj))=|>
State':=3/\request(U,SN,user_snode,T1')
end role
```

Tableau 4 : la spécification de rôle pour le nœud de capteur Sni

4. Rôle Session et environnement

Enfin, les spécifications dans le langage HLPSL pour le rôle de session, objectif et environnement sont spécifiées sur les TABLEUAX. 5 et 6. Dans le segment de session, tous les rôles de base (Snode, serveur et user) sont instanciés avec des arguments concrets.

Le rôle de niveau supérieur (environnement) est toujours défini dans la spécification du langage HLPSL. Ce rôle contient les constantes globales et une composition d'une ou plusieurs sessions, où l'intrus peut jouer certains rôles en tant qu'utilisateurs légitimes. L'intrus participe également à l'exécution du protocole sous forme de session concrète. La version actuelle de HLPSL prend en charge les objectifs d'authentification et

de confidentialité standard. Dans le protocole, six objectifs de confidentialité et quatre authentifications sont vérifiés.

```
role session(SN,BS,U:agent,
MKsi:symmetric_key,
        MKuj:symmetric_key,
        H:hash_func)
def=
local US,UR,SS,SR,VS,VR:channel(dy)
composition
snode(SN,BS,U,MKsi,H,US,UR)
/\server(BS,U,SN,MKsi,MKuj,H,SS,SR)
/\user(U,BS,SN,MKsi,MKuj,H,VS,VR)
end role
```

Tableau 5 :la specification de role session

```
role environment()
def=
const sn,bs,u :agent,
        mksi:symmetric_key,
        mkuj:symmetric_key,
        h:hash_func,
rpwj,ruj,sj,zj,kuj,ki,rmuj,ti,t1,t2,apmh,gidj,kbs,sni,uj:text,
snode_server,snode_user,user_server,user_snode,subs1,subs2,subs3,subs4,subs5,subs6:
protocol_id
intruder_knowledge={u,bs,sn,h,uj,sni,uj}
composition
session(sn,u,bs,mksi,mkuj,h)/\
session(u,sn,bs,mksi,mkuj,h)/\
session(u,sn,bs,mksi,mkuj,h)
end role
goal
secrecy_of subs1
secrecy_of subs2
secrecy_of subs3
secrecy_of subs4
secrecy_of subs5
secrecy_of subs6
authentication_on snode_server
authentication_on user_server
authentication_on user_snode
authentication_on snode_user
end goal
environment()
```

Tableau 6 : la spécification de rôle goal et environnement

Nous avons simulé protocole 1 pour les back-ends OFMC et CL-AtSe à l'aide de l'outil AVISPA.

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

C'est un outil basé sur des techniques de résolution de contraintes. 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. Les deux méthodes de la traduction et la vérification sont totalement automatiques et prises en charge par l'outil CL AtSe sans intervention d'outils externes. Les possibilités de CL–AtSe ont été étendues lors du projet AVISPA pour supporter de vérification des protocoles intégrant les opérateurs algébriques (Xor ou Exp).

- **On the fly Model Checker (OFMC)**

Pour but d'analyser la sécurité des protocoles cryptographiques. 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.

Les résultats de la simulation sont illustrés aux Fig. 20 et 21. Le résumé des résultats est le suivant :

OFMC signale que le protocole est sécurisé.

CL-AtSe indique que le protocole est sécurisé.

Il est donc clair que ce protocole est sécurisé contre les attaques passives et actives, y compris les attaques de type "rejouer" et "intercepteur".

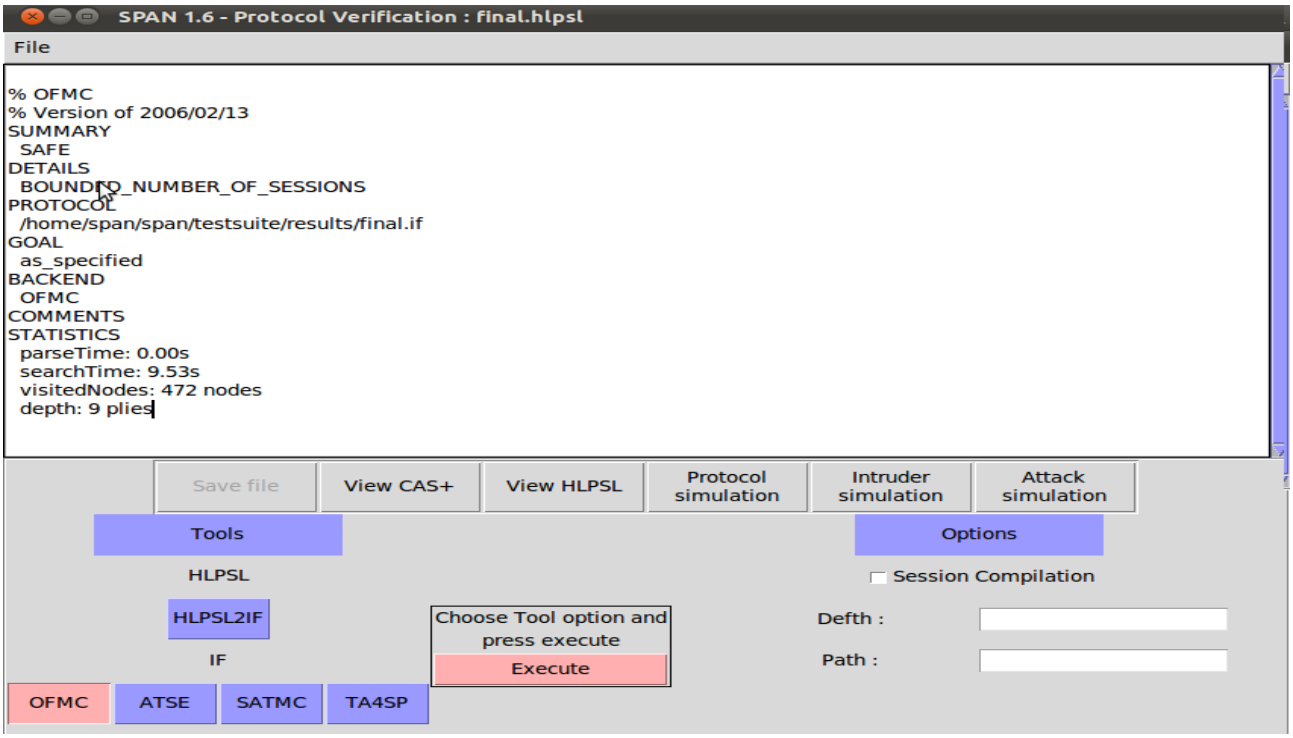


Figure 3: Le résultat de l'analyse utilisant OFMC

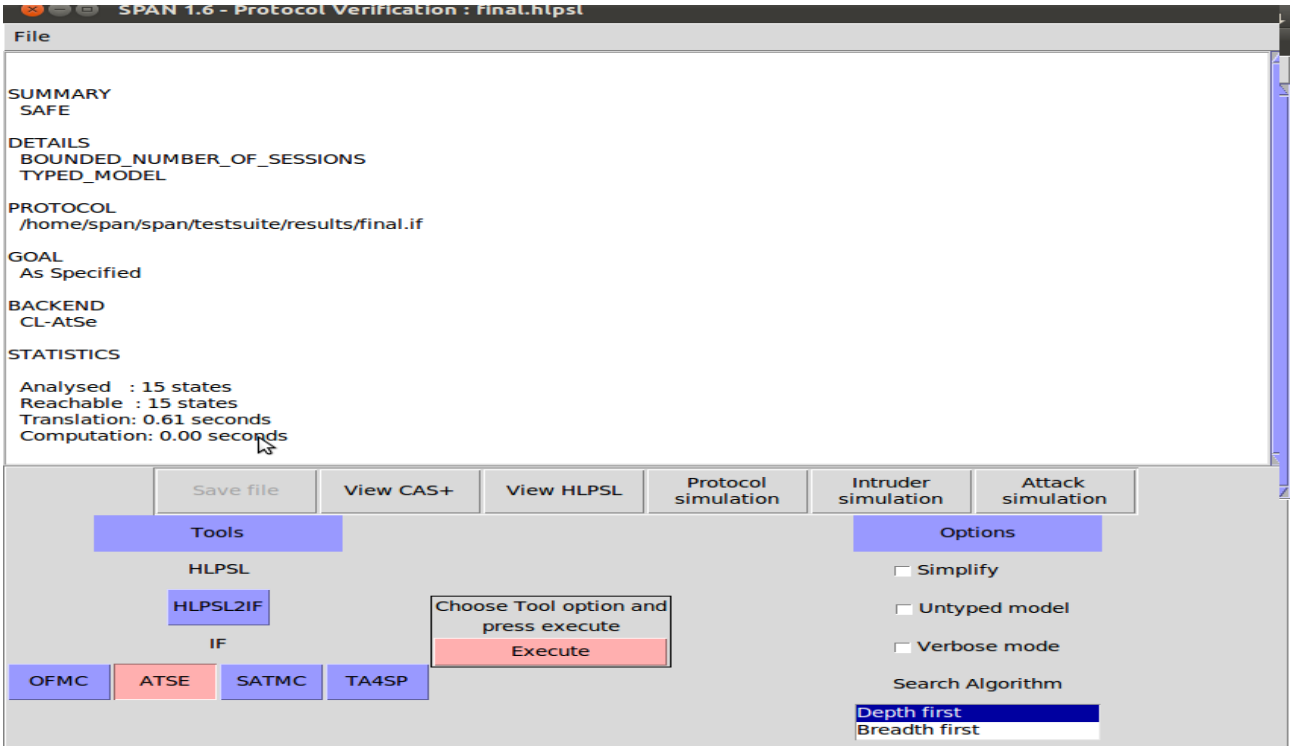


Figure 4: Le résultat de l'analyse utilisant CL-AtSe

Après avoir testé ce protocole sous l'outil AVISPA, nous avons confirmé que ce protocole est fiable et sécurisé

Protocole N°2 [29]

1. Principe

Ce protocole de sécurité est proposé pour l'authentification des utilisateurs de WBAN en prenant en compte l'authentification mutuelle, l'établissement de la clé de session, l'actualisation des données et la confidentialité.

Ils considèrent que les WBAN sont constitués de quelques utilisateurs (avec la carte à puce pouvant être capturée ou volée par l'adversaire A), de centaines de nœuds de capteur (ces nœuds peuvent être capturés par A) et d'un nœud de passerelle sécurisé. Compte tenu de ces entités, ils conçoivent le protocole qui comprend quatre phases :

2. LES PHASES

I. La phase déploiement

Au cours de cette phase, ils sélectionnent un nœud informatique de hautes performances et de confiance comme passerelle GWN qui attribue une identité unique IDSNj à chaque nœud de capteur SNj et charge une clé secrète unique Kgsnj = h (IDsnj ||Kgwn) dans la mémoire de SNj et envoie des commentaires historiques enregistré communauté

II. La phase Enregistrement

Dans cette phase, un utilisateur légitime Ui envoie les informations d'identification secrètes hachées [30]à GWN à l'aide d'un canal de communication sécurisé et le GWN fournit une carte à puce (constituée d'un paramètre secret connu uniquement du GWN) SCi à Ui. [29]

III. Phase d'authentification

Dans cette phase, ils utilisent la procédure de reproduction Rep (.) De l'extracteur flou [31]pour l'authentification de l'utilisateur (Ui) avec son identifiant biométrique bruyant B'i, et utilisent la procédure de courbe elliptique Diffie-Hellman [32]pour partager la clé de session commune (SK) entre l'utilisateur (Ui) et le nœud de capteur SNi. [29]

IV. Phase de mise à jour des informations d'identification de l'utilisateur

Si un utilisateur légitime est authentifié à l'aide de son identité (IDui), de son mot de passe (PWui), des informations biométriques (Bi) et de la carte à puce (SCi), il peut mettre à jour son mot de passe et ses informations biométriques [29]

Enfin, ils valident automatiquement la sécurité de ce protocole à l'aide de l'outil AVISPA [28](version v1.1) basé sur le modèle d'intrus Dolev-Yao avec des sortie OFMC et CL-AtSe.

3. Analyse du protocole

Dans cette section, nous effectuons des analyses de fonctionnalité et de sécurité de protocole

Vérification formelle de sécurité du protocole avec AVISPA back-ends

La spécification HLPSL du protocole comprend certaines sections importantes :

1. Les Rôles de base :

    - Chaque rôle peut avoir un paramètre tel que Ui (utilisateur), GWN(getaway), SNj de type agent et Kui1, Kgsnj de type symmetric_key.
    - Les paramètres RCV et SND désignent les canaux de communication de l'agent pour la réception et l'envoi des informations.
    - Le paramètre (dy) représente le modèle d'intrus Dolev-Yao pour le canal.
    - Les fonctions H, Gen, Rép, EccMul, Enc, Dec et XOR correspondant à la fonction de hachage, au générateur d'extracteur flou, à la reproduction à l'extracteur flou, à la multiplication scalaire à courbe elliptique, au cryptage, au décryptage et aux opérations logiques XOR.
    - Le terme hash_func représente toutes les fonctions qui ne sont pas facilement inversibles parceque les opérateurs arithmétiques aléatoires non inversibles et ne sont pas supportés dans HLPSL.
    - Le terme "play_by Ui "indique que le rôle User est joué par Ui.

Les spécifications HLPSL des rôles d'Utilisateur, GWN et SNj sont présentées dans les tableaux 12 à 14 respectivement.

2. Transitions: Les transitions sont déclarées par étapes. Il s'agit d'un déclencheur qui se déclenche lorsqu'un événement se produit. Pour tous les États d'une transition, si un message est reçu sur le canal RCV, la transition est déclenchée et attribue une nouvelle valeur à l'État.
3. Les Rôles composés: Il crée un ou plusieurs rôles de base à exécuter ensemble et représente les sessions impliquées dans le protocole. L'opérateur ^ représente l'exécution parallèle des rôles. Et La spécification HLPSL de la session de protocole proposée est présentée dans le tableau 15
4. Environnement: Il consiste en une composition globale de constante et de session dans laquelle l'adversaire peut jouer un rôle en tant qu'utilisateur autorisé. Et La spécification HLPSL de l'environnement du protocole proposé est présentée dans le tableau 16.
5. Objectif de sécurité: Ce module spécifie l'objectif de sécurité du protocole. Les prédicats importants utilisés dans ce module sont les suivants:
    - secret ({PWi, Bi, SIGi '}, sub1, Ui): indique que les informations {PWi, Bi, SIGi'} sont secrètement partagées avec Ui et peuvent être reconnues avec une identité constante sub1 dans la section objectif.

- Witness (Ui, GWN, gateway_user_gu, Tui, Alpha '): il représente la faible authenticité de Ui par GWN et Ui est le témoin des données {Tui, Alpha'}. L'identité de cet objectif est représentée par gateway_user_gu dans la section des objectifs.
- request (Ui, SNj, user_sensor_us, Skey '): représente la forte authenticité de Ui by SNj sur Skey avec une identité user_sensor_us.
- Symboles: La concaténation (.) Est utilisée pour la composition du message) et les virgules (,) est utilisé dans le cas d'arguments multiples d'événements ou de fonctions.

1. Role utilisateur

```
Role user(Ui, GWN, SNj: agent,
Xui1, Kgsnj: symmetric_key,
H,Gen, Rep, EccMul, Enc, Dec, XOR: hash_func,
SND, RCV: channel(dy))
played_by Ui def=
local
State: nat,
IDui, IDsnj, PWui, Bi, Bi1, SIGi, SIGi1, TAUi, PBi, PBi1, P, Kui1, Rui, Aui, Bui,Wui, Alpha, Beta,
Gamma, Ysnj, Ysnj1, Tui, Tgwn, Xui, X, Beta1, Kui, Rsnj, Gamma1, Skey, Skey1: text
const sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 0
transition
0. State = 0 ^ RCV (start) = .
State': = 2 ^ SIGi': = Gen(Bi)
            ^ PBi': = H(PWui.SIGi')
^ secret(PWui,Bi,SIGi', sub1, Ui)
          ^ SND (IDui.PBi')
2. State = 2 ^ RCV (P.Aui'.Bui'.Wui') = .
State': = 5 ^ Rui': = new()
            ^ Tui': = new()
            ^ secret(Rui', sub2, Ui)
            ^ SIGi1': = Rep(Bi1.TAUi)
            ^ PBi1': = H(PWui.SIGi1')
            ^ Kui1': = XOR(Wui, H(IDui.PBi1'))
            ^ Xui': = EccMul(Rui'.P)
            ^ Xui1': = EccMul(Rui'.Kui1')
^ secret(Xui1', sub3, Ui, GWN)
            ^ Alpha': = Enc(IDsnj.Tui)
            ^ SND(IDui.Xui'.Alpha')
^ witness(Ui, GWN, gateway_user_gu, Tui,Alpha')
6. State = 5 ^ RCV(Beta1') = .
State': = 6 ^ Ysnj1': = Dec(Beta1')
            ^ Skey': = EccMul(Rui'.Ysnj1')
            ^ request(Ui,SNj, user_sensor_us, Skey')
end role
```

Tableau 7 : Spécification du rôle de l'Ui dans HLPSL

2. Rôle Gateway

```
role gateway(Ui, GWN, SNj: agent,
Xui1, Kgsnj: symmetric_key,
H,Gen, Rep, EccMul, Enc, Dec, XOR: hash_func,
SND, RCV: channel(dy))
played_by GWN def=
local
State: nat,
IDui, IDsnj, PWui, Bi, Bi1, SIGi, SIGi1, TAUi, PBi, PBi1, P, Kui1, Rui, Aui, Bui,Wui, Alpha,
Beta,Gamma, Ysnj, Tui, Tgwn, Xui, X, Beta1, Kui, Rsnj, Gamma1, Skey, Skey1: text
const sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 1
transition
1. State = 1 ^RCV (IDui.PBi')= .
State': = 3 ^ X': = new()
            ^ Kui': = EccMul(H(IDui.X').P)
            ^ Aui': = XOR(PBi'.H(XOR(IDui.X')))
            ^ Bui': = H(IDui.PBi'.XOR(IDui.X'))
            ^ secret(X',sub4, GWN)
            ^ Wui': = XOR(H(IDui.PBi).Kui')
            ^ secret(Kui', sub5, GWN,Ui)
            ^ SND(P.Aui'.Bui'.Wui')
3. State = 3 ^ RCV(IDui.Xui'.Alpha')= .
State': = 4 ^ Tgwn': =new()
^request(GWN, Ui, gateway_user_gu, Alpha')
^ IDsnj': = Dec(Alpha')
             ^ Rsnj': = new()
             ^ Ysnj': = EccMul(Rsnj'.P)
           ^ Beta': = Enc(IDsnj'.Ysnj'.Tgwn)
           ^secret(Kgsnj, sub6, GWN,SNj)
           ^Gamma': = Enc(IDui.Skey'.Beta'.Tgwn')
           ^ SND(Gamma')
           ^ witness(GWN, Ui, gateway_user_gu, Tgwn')
end role
```

Tableau 8 : Spécification du rôle de GWN dans HLPSL

3. Role noued de capteur «sensor node »

```
role sensor(Ui, GWN, SNj: agent,
Xui1, Kgsnj: symmetric_key,
H,Gen, Rep, EccMul, Enc, Dec, XOR: hash_func,
SND, RCV: channel(dy))
played_by SNj def=
local
State: nat,
IDui, IDsnj, PWui, Bi, Bi1, SIGi, SIGi1, TAUi, PBi, PBi1, P, Kui1, Rui, Aui, Bui,Wui, Alpha,
Beta,
Gamma, Ysnj, Tui, Tgwn, Xui, X, Beta1, Kui, Rsnj, Gamma1, Skey, Skey1: text
const sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
init
State: = 4
transition
4. State = 4 ^RCV (Gamma') = .
State': = 5^ Skey1': = Dec(Gamma'.Kgsnj)
^ secret(Skey1', sub7, SNj)
^ Beta1': = Dec(Gamma')
^ secret(Skey1', sub8, SNj)
^ SND(Beta1')
end role
```

Tableau 9 : Spécification du rôle de SNj dans HLPS

4. Role session

```
role session(Ui,GWN,SNj:agent,
Xui1, Kgsnj:symmetric_key,
H,Gen, Rep, EccMul, Enc, Dec, XOR: hash_func)
def=
local GWNUi,RUi,GWNSNj,RSNj,GWNGWN,RGWN:channel(dy)
composition
user(Ui, GWN, SNj, Xui1,Kgsnj,H,Gen, Rep, EccMul, Enc, Dec,XOR,GWNUi, RUi)
        ^ sensor(Ui, GWN, SNj,Xui1, Kgsnj, H,Gen, Rep, EccMul, Enc, Dec, XOR,GWNSNj,
RSNj)
        ^ gateway(Ui, GWN, SNj, Xui1, Kgsnj,H,Gen, Rep, EccMul, Enc, Dec, XOR, GWNGWN,
RGWN)
end role
```

Tableau 10 : Spécification de la session de protocole proposée en HLPSL

5. Role environnement

```
role environment()
def=
const ui, gwn, snj: agent,
xui1,kgsnj,kig: symmetric_key,
h,gen, rep, eccMul, enc, dec, xOR: hash_func,
sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8,
gateway_sensor_gs, gateway_user_gu, user_sensor_us: protocol_id
intruder_knowledge = ui,gwn,snj,kig
composition
session(ui,snj,gwn,xui1,kig,h,gen, rep, eccMul, enc, dec, xOR)
      ^ session(ui,snj,gwn,kgsnj,kig,h,gen, rep, eccMul, enc, dec, xOR)
       ^ session(ui,snj,gwn,kig,kgsnj,h,gen, rep, eccMul, enc, dec, xOR)
end role

Goal
secrecy_of sub1, sub2, sub3, sub4, sub5, sub6, sub7, sub8
authentication_on gateway_sensor_gs, gateway_user_gu, user_sensor_us
end goal
environment()
```

Tableau 11 : Spécification de l'environnement et goal de protocole proposé en HLPSL

Utilisé OFMC

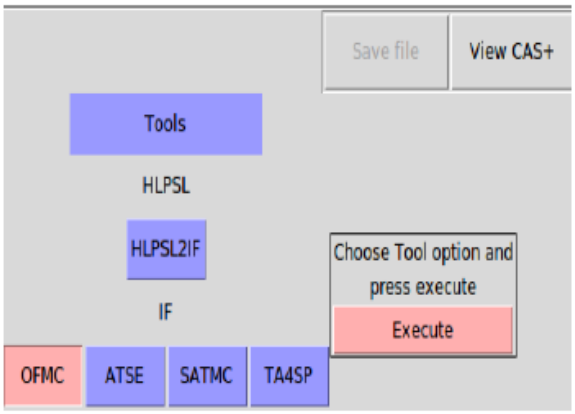
SUMMARY
SAFE

DETAILS
BOUNDED_NUMBER_OF_SESSIONS

PROTOCOL
/home/cmb-lab-22/Desktop/Proto.if

GOAL
as_specified
BACKEND
OFMC

STATISTICS
Time: 984 ms
parseTime: 0 ms
visitedNodes: 456 nodes
depth: 9 piles



Utilisé ATSE

SUMMARY
SAFE

DETAILS
BOUNDED_NUMBER_OF_SESSIONS
TYPED_MODEL

PROTOCOL
/home/cmb-lab-22/Desktop/Proto.if

GOAL
as_specified
BACKEND
CL-AtSe

STATISTICS
Analysed: 1956 states
Reachable: 1956 states
Translation: 0.06 s
Computation: 0.01 s

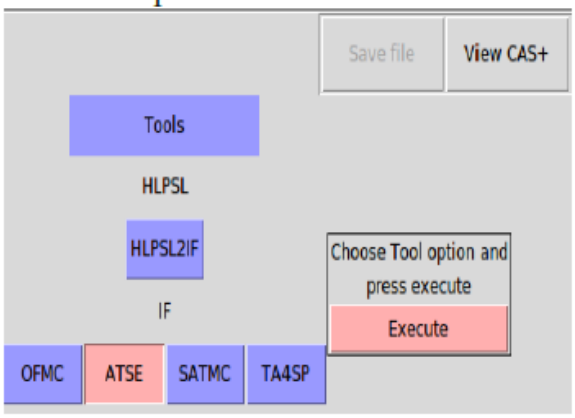


Figure 4 : Résultat de la vérification de sécurité obtenu à l'aide de l'outil AVISPA

Après avoir testé voir le figure 4ce protocole sous l'outil AVISPA, nous avons confirmé que ce protocole est fiable et sécurisé

Conclusion

Dans ce chapitre, nous avons évalué les performances des protocoles que nous avons choisis. Pour ce faire nous avons simulé les deux protocoles par les back-ends OFMC et CL-AtSe à l'aide de l'outil AVISPA.
Les objectifs d'authentification et de confidentialité sont atteints, alors On peut dire que ces protocoles sont valides.

Conclusion générale

Un système de santé promettant une collecte continue et fiable et une analyse objective des aspects physiologiques et comportementaux d'un patient, tout en fournissant ces informations aux médecins, a été l'objectif des réseaux corporels sans fil (WBANs).

Ces réseaux sont apparus comme une technologie ayant le potentiel pour révolutionner la prestation des soins de santé dans les ambulances, les salles d'urgence, les salles d'opération, les cliniques et même dans nos maisons.

Toutefois, les WBANs sont encore au stade précoce de leur développement, et plusieurs défis de recherche doivent être surmontés afin qu'ils puissent être largement acceptés. L'authentification est l'un des principaux défis à relever tant les données collectées sont sensibles et directement associées à un patient particulier. Un intrus malintentionné peut tenter d'usurper l'identité d'un nœud légitime afin d'injecter des données pouvant mettre en danger la santé du patient.
De plus, la conception d'un mécanisme de sécurité pour les WBANs doit faire face à certaines contraintes en raison de l'utilisation de nœuds capteurs limités en termes de puissance de calcul, de réserve d'énergie, d'espace de stockage, etc.

Ce document a débuté par une étude générale sur les réseaux de capteurs sans fil. Nous avons présenté en premier lieu la définition, l'architecture et les type de capteurs. Ensuite nous avons effectué une description des réseaux de capteur sans fil, Une comparaison entre les MANETS et les RCSF et une pile protocolaire dans RCSF, en suite nous avons présenté les domaines d'application de ses réseaux.
Dans le 2eme chapitre (Les Réseaux de capteurs corporels sans fil) nous avons présenté définie les capteurs médicaux et l'architecture de ces capteurs ,ensuite nous sommes passé a la définition des WBANS ,et leurs architecture.la topologie et les contraintes de ces réseaux ensuite nous avons fait une comparaison entre les WBANs et les WSNs

Dans le 3eme chapitre nous avons défini la cryptographie, ensuite nous avons présenté les algorithmes de la cryptographie (symétrique et asymétrique). Les fonctions de hachage et dans la fin de ce chapitre nous avons fait une comparaison entre les Algorithmes symétriques (clef secrète) et les Algorithmes asymétriques (clef public).

Dans le quatrième chapitre, nous avons choisie deux protocoles de sécurité pour les étudier. Pour chaque protocole nous avons expliqué le principe de protocole (les cryptographies utilisées, les grandes phases du protocole) et nous avons fait une analyse formelle de sécurité du protocole avec AVISPA. Notre solution c'est que les deux protocoles sont des protocoles valides et sécurisé.
Enfin Pour conclure ce document, Nous n'avons pas pu atteindre les objectifs fixé dans le début de ce travail a cause des contraintes de temps qu'il y au Manque de documentation sur l'outil AVISPA .Alors, l'objet de nos futures recherches ces atteindre les objectifs fixé dans le début de ce travail.

Références

[1] J. Y. J. H. S. Z. D. L. J. F. A. Kris Lin, «enabling long-lived sensor networks through solar energy harvesting» chez *Proceedings of the 3rd international conference on Embedded networked sensor systems*, New York,USA, 2005.

[2] Y. Challal, « Réseaux de Capteurs Sans Fils », France: Université de Technologie de Compiègne, 17 Novembre 2008.

[3] «Tinyos» 2019. [En ligne]. Available: http:/www.tinyos.net/.

[4] A. B. G. e. T. V. Dunkels, «Contiki: a Lightweight and Flexible Operating System for Tiny Networked Sensors,» 2004.

[5] S. Sentilles, «Architecture logicielle pour capteurs sans-fil» Université de Pau et des Pays de l'Adour, juin 2016.

[6] W. S. Y. S. E. C. I.F. Akyildiz, «wireless sensor network : a survey» *computer networks,* vol. 38, pp. 393-422, 2002.

[7] B. Kechar, «« Problématique de la consommation de l'énergie dans les réseaux» universite oran, Algerie , 2010.

[8] D. Martins, «"Sécurité dans les réseaux de capteurs sans fil Stéganographie et réseaux de confiance» l'université de Franche-Comté, 2010.

[9] S. W. S. &. C. E. Akyildiz I. F., «Wireless Sensor Network : a servey» *Computer Networks,* vol. 38, pp. 393-422, 2012.

[10] V. S. Somerset, «"Intelligent and Biosensors",» Intech, January 2010.

[11] Y. Younes, «"Minimisation d'énergie dans un réseau de capteurs",» Université Mouloud Mammeri , Tizi-Ouzou, 2012.

[12] A. Makke, «Détection d'attaques dans un système WBAN de,» Université René Descartes - Paris V, paris , france, 2014.

[13] J. Gabay, «Technologie de capteurs pour santé et fitness,» *la contribution de Electronic Products,* 2015-10-28.

[14] A. M. Héribert, «Conception théorique d'un réseau WBAN pour,» UNIVERSITE D'ABOMEY CALAVI (UAC), 2017.

[15] B. Mohamed, «Surveillance de tout point d'une zone d'intérêt à l'aide d'un réseau de capteur multimédia sans fil,» Ecole nationale supérieure d'informatique, Oued- Smar Alger Algérie, 2013.

[16] A. Ahmad, «A Review of Wireless Sensor Networks Applications,» aout 2011. [En ligne]. Available: https://www.researchgate.net/publication/227352668. [Accès le 10 juin 2019].

[17] L. Z. S. W. &. C. V. Walters J. P., «Wireless Sensor Network Security : a survey,» *Security in Distributed, Grid, and Pervasive Computing,* vol. 14, pp. 367-404, 2014.

[18] A. M. L. J. S. D. &. J. A. Movassaghi S., «Wireless Body Area,» *In IEEE Communications Surveys and Tutorials,* vol. 16, pp. 1658-1686, 2014.

[19] W. Znaid, «Quelques propositions de solutions pour la sécurité des réseaux de capteurs,» 2010.

[20] S. M. &. R. J. Neves P., «Application of Wireless Sensor Networks to Healthcare Promotion,» *Journal of Communications Software and Systems,* vol. 4, pp. 181-190, 2008.

[21] D. W. Steffen Peter, «« A Survey on the Encryption of Convergecast Traffic with In-Network Processing »,» *IEEE Transactions on dependable and secure,* vol. 05.

[22] V. MA, «Les concepts de base de la cryptographie,» Université Paris-Est Marne-la-Vallée, [En ligne]. Available: http://igm.univ-mlv.fr/~dr/XPOSE2007/vma_PKI/liens_utiles.html. [Accès le 10 6 2019].

[23] J. Buchmann, Introduction to Cryptography, Hardcover.

[24] H. FERRADI, Écrivain, *Introduction à la cryptographie (cours 3):Chiffrement par bloc.*. Université Paris 13 Villetaneuse, 01/02/2016.

[25] «cryptographie,» [En ligne]. Available: http://www.securiteinfo.com/cryptographie/hash.shtml. [Accès le 11 06 2019].

[26] A. K. D. ,. J. K. S. Santanu Chatterjee, «A novel and efficient user access control scheme for wireless body area sensor networks,» *Journal of King Saud University,* pp. 182-201, 26 October 2013.

[27] M. L. Das, «Two-factor user authentication in wireless sensor networks,» *IEEE Transactions on Wireless Communications,* vol. 08, n° 13, pp. 1086-1090, mars 2009.

[28] T. A. team, «HLPSL Tutorial,» the European Community under the Information Society Technologies Programme, 2006.

[29] V. N. S. Anup Kumar Maurya, «Fuzzy Extractor and Elliptic Curve Based Efficient User Authentication Protocol for Wireless Sensor Networks and Internet of Things,» vol. 8, n° %1136, 30 October 2017.

[30] D. Stinson, «some observations on the theory of cryptographic hash functions,» vol. 38, p. 259–277, 2006.

[31] Y. Dodis, L. Reyzin et Smith, «Fuzzy extractors: How to generate strong keys from biometrics and other,» chez *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, 2004.

[32] V. Miller, «Use of elliptic curves in cryptography,» *In Advances in Cryptology-CRYPTO 85; Lecture Notes,* p. 417–426, 1986.

ANEXXE

Le langage de spécification HLPSL

Et la plateforme AVISPA

Sommaire

II.1. Introduction

Pour la vérification des protocoles de sécurité, il existe plusieurs outils d'analyses mais les outils AVISPA (*Automated Validation of Internet Security Protocols and Applications*) est la plus connue. Dans ce chapitre nous allons présenter le langage de spécification des protocoles des sécurités HLPSL (*High Level Protocol Specification Language*) et nous allons donner un exemple de spécification de protocole fil-rouge. Ensuite nous allons donner un bref aperçu sur la plateforme AVISPA et nous expliquent les quatre outils de vérification qui ont été intégrés dans la plate-forme : OFMC, CL-ATSE, SATMC et TA4SP. Enfin, nous allons montrer l'animateur de protocoles de sécurité SPAN (*Security Protocol ANimator*) .

II.2. Le langage de spécification HLPSL :

Le HLPSL (High Level Protocol Specification Language) est un langage expressif de la communication et de la modélisation des protocoles de sécurité pour l'outil AVISPA. Il permet ainsi la représentation d'un protocole de sécurité par un système d'états/transitions sur lequel la vérification des propriétés de sûreté exprimées en logique temporelle linéaire(LTL) sera effectuée.

C'est un langage de spécification modulaire basé sur la notion de rôles (participants) et de rôles composés (sessions, instances). Un rôle simple sert à décrire les actions d'un agent lors de l'exécution du protocole. Un rôle composé permet d'instancier plusieurs rôles simples afin de modéliser l'exécution du protocole entier. En plus de la notion de rôle fournie par HLPSL, ce langage possède les caractéristiques suivantes :

- Primitives cryptographiques variés (clés symétriques, clés publiques et privées, fonctions de hachage,…);
- Information typée (ou non), avec des types simples ou composés ;
- Propriétés algébriques supportées (concaténation, OU exclusif, exponentiation…);
- Canaux pour les échanges de messages ;
- Flux de contrôle assurant les transitions valides ;
- Propriétés de sécurité à vérifier : confidentialité, authentification, intégrité.

II.3. Structure d'une spécification HLPSL

Une spécification HLPSL est composée de trois parties : Une liste de définition des rôles, une liste des objectifs ou des propriétés de sécurité à vérifier, et enfin, l'instanciation du rôle principal (généralement sans arguments).

Dans notre exemple de spécification de protocole *fil-rouge* par le langage HLPSL. La description de ce protocole par notation Alice-Bob est comme ci-dessous :

1.A -> B : Kab, {M}Kab

II.3.1. Définition des rôles

Les rôles peuvent être définis comme des processus indépendants qui ont des noms, reçoivent des informations en paramètre et contiennent des déclarations locales. On distingue deux catégories de rôles :

II.3.1.1. Les rôles basiques

Cette catégorie décrite la connaissance initiale et le comportement de chaque entité (agent) intervenant dans le protocole spécifié. Ci-dessous, un exemple simplifié d'une déclaration de rôle pour l'agent "Alice" dans le protocole *fil-rouge*. Cet exemple est donné afin de mieux comprendre cette notion de rôle.

```
role alice (A,B : agent,          Kab :
symmetric_key,          SND,RCV:
channel(dy)) played_by Adef=

local State : nat,          M : text
const id1 : protocol_id init State
:= 0 transition
    1. State = 0  /\ RCV(start) =|>
      State':= 1 /\ M' := new()
/\ SND(Kab.{M'}_Kab)
                    /\ secret(M',id1,{A,B}) end role
```

La connaissance initiale liée à chaque rôle (dans notre exemple, concerne alice) est exprimée par une liste de paramètres qui sont : la clé publique Kab, les différents agents connus par Alice, un ensemble de canaux SND, RCV qui seront utilisés pour la communication. Dans la déclaration d'un rôle, on peut trouver une clause optionnelle "played_by" pour spécifier quel agent joue le rôle considéré. Dans le rôle Alice l'acteur principal est l'agent A (played_by A).

Nous pouvons trouver aussi une section qui contient une liste de variables locales et leurs types, qui peuvent être initialisées dans la section 'init'. L'extrait de spécification exprime la déclaration de deux variables *locales State* et *M*, précisant que la variable *State* est initialement instanciée à la valeur 0.

local State : nat,

M: text

init

State := 0

Une liste de transitions est définie comme dans la toute dernière partie du rôle. Cette dernière va définir le comportement de l'agent à travers de des transitions qu'il peut faire suivant une certaine condition pour changer d'état.

Concernant notre exemple (protocole *fil-rouge*), si l'agent A se trouve à l'état (State=0) et que sur le canal Rcv il peut lire le message "*start*" alors la transition est déclenchée et change d'état vers State' = 1. (State' signifie que l'ancienne valeur stockée dans State sera écrasée par la nouvelle valeur qui est "1"). La variable *M*est instanciée par une valeur aléatoire (nonce) grâce à l'instruction *M' :=new()*.Cette nouvelle valeur est d'abord chiffrée par Kab, puis concaténée à Kab et les envoyées sur le canal SND.

1. State = 0 /\ RCV(start) =|>

State':= 1 /\ M' := new()

/\ SND(Kab.{M'}_Kab)

II.3.1.2. Les rôles de composition

Chaque spécification doit avoir un rôle spécial qui peut s'appeler rôle de session et qui va représenter la structure des sessions pendant l'exécution du protocole. Dans l'exemple ci_dessous est représentée une session du protocole *fil-rouge* entre deux agents Alice et Bob. L'opérateur / \ indique que ces rôles devraient exécuter en parallèle. Les variables SA et SB sont des canaux d'émission des messages des agents A et B respectivement. Au contraire RA et RB sont des canaux de réception des messages des agents A et B respectivement.

```
role session(A, B: agent, Kab : symmetric_key)def=
local SA, RA, SB, RB: channel (dy) composition
alice(A,B,Kab,SA,RA)    /\ bob
(A,B,Kab,SB,RB)
 end role
```

Il existe un autre rôle de composition appelé environnement ou rôle principal. Ce rôle ne possède aucun paramètre et sert à définir l'état initial du système en précisant d'un côté, la connaissance initiale de l'intrus par la clause *intruder knowledge* et d'un autre côté, un nombre fini d'instances du rôle session.

Dans la spécification d'une session de protocole *fil-rouge* Le rôle ci-dessous exprime une session qui se déroule entre deux agents a et b en utilisant la clé kab. Les données a, b et kab sont des constantes associées à un type dans la section *const*. L'intrus connaît initialement les agents a et b.

```
role environment() def=   const a, b :
agent,        kab  : symmetric_key
intruder_knowledge = {a, b}
   composition        session(a,b,kab)
end role
```

II.3.2. Les objectifs de sécurité

Il existe différents signaux ou évènement. L'évènement *secret* permet de déterminer les propriétés de confidentialité alors que les deux autres évènements (*Witness, Request*) permettent la définition de deux types de propriété d'authentification : authentification faible et authentification forte. Ces signaux permettant d'exprimer les propriétés de sécurit :

- **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.

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) pour une transition donnée.

L'exemple (protocole *fil-rouge*) permet de déclarer la nouvelle valeur de M après exécution de la transition comme étant un secret partagé entre A et B. L'instruction M' := new () permet de générer aléatoirement une nouvelle valeur pour M.

State=0 ∧ RCV(start) =|>
State':=1 ∧ M' := new()
∧ SND(Kab.{M'} Kab) ∧ secret(M', id_sec,{A,B})

En HLPSL, nous pouvons exprimer les objectifs à vérifier à l'aide des macros suivantes dans une section HLPSL réservée et nommée goal :

Secrecy_ofsec_id Authentication_onauth_id
weakauthentication_onwauth_id

Les données *sec_id*, *auth_id* et *wauth_id* sont les identifiants attribués aux signaux.
Remarquons que pour exprimer une propriété d'authentification faible ou forte, nous devons spécifier deux signaux *witness*et *request*ayant le même identifiant.

II.3.3. Instanciation d'un rôle

La création d'une instance d'un rôle est comme l'appel d'une procédure, en donnant une valeur pour chaque paramètre déclaré dans le rôle, et bien sûr le nombre d'arguments ainsi que leurs types doit être les mêmes que ceux des paramètres du rôle. Pour le rôle principal, il suffit d'invoquer son nom (usuellement sans paramètre comme suit : *environment()*).

II.4. La plate-forme AVISPA

II.4.1. Description et Architecteur

Le projet AVISPA [23] a développé une plate-forme de vérification automatique de protocoles sécurités, appelée également AVISPA. Elle est accessible sur le réseau à l'adresse www.avispa-project.org Elle met à disposition le langage HLPSL qui est utilisé pour spécifier à la fois les protocoles et les propriétés à vérifier. Dans ce langage, deux symboles sont définies pour le chiffrement symétrique et asymétrique. Il permet aussi la déclaration des fonctions de hachage. Il existe également un opérateur de concaténation. Par contre, il n'y pas d'opérateur pour la signature. Le langage HLPSL permet de décrire l'échange des messages entre les participants pendant l'exécution d'un protocole. La plate-forme permet de vérifier trois propriétés : le secret, l'authentification faible et l'authentification forte. Cette dernière propriété est une combinaison entre l'authentification faible et la protection contre le rejeu. Après avoir spécifié le protocole dans HLPSL, celui-ci est traduit dans le langage IF (Intermediate Format) par le compilateur ou traducteur HLPSL2IF.

Le code IF peut ensuite être utilisé par des outils d'analyse différents qui effectuent la vérification. Quatre outils de vérification ont été intégrés dans la plate-forme AVISPA : OFMC, ATSE, SATMC et TA4SP.L'architecture de la plateforme AVISPA est montrée dans la Fig.1.

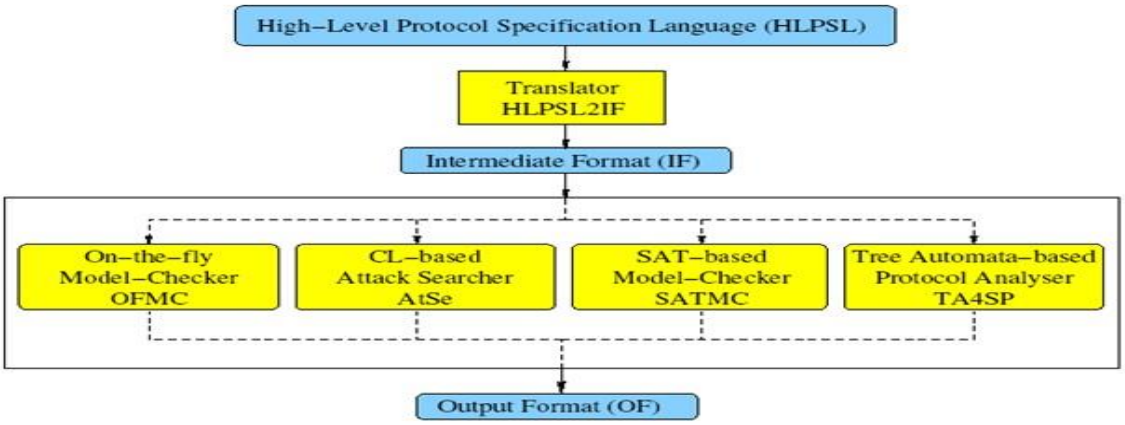


Figure 1 : l'architecture de la plateforme AVISPA

Une interface graphique (Figure3) est également disponible selon deux modes :

- Mode *basique* simplifiant les actions de l'utilisateur, et lançant les quatre outils en parallèle, le résultat de chacun étant ensuite visible ;
- M*ode expert* permettant de choisir l'outil d'analyse et de régler quelques paramètres d'analyse.

Lorsqu'une attaque est trouvée, un diagramme de séquence permet de visualiser les messages échanges qui ont mené à cette attaque.

II.4.2. Le traducteur HLPSL2IF

Le traducteur HLPSL2IF traduit automatiquement une spécification de HLPSL vers une spécification IF. Le traducteur fonctionne comme suit. Premièrement, il analyse la spécification de HLPSL, vérifiant qu'un nombre de conditions sont rempliés (par exemple que toutes les variables utilisées sont déclarées). Ensuite, il aplatit la structure hiérarchique des descriptions de rôles dans HLPSL et les traduit dans les règles d'étape du IF, décrivant les transitions agents honnêtes peuvent exécuter.

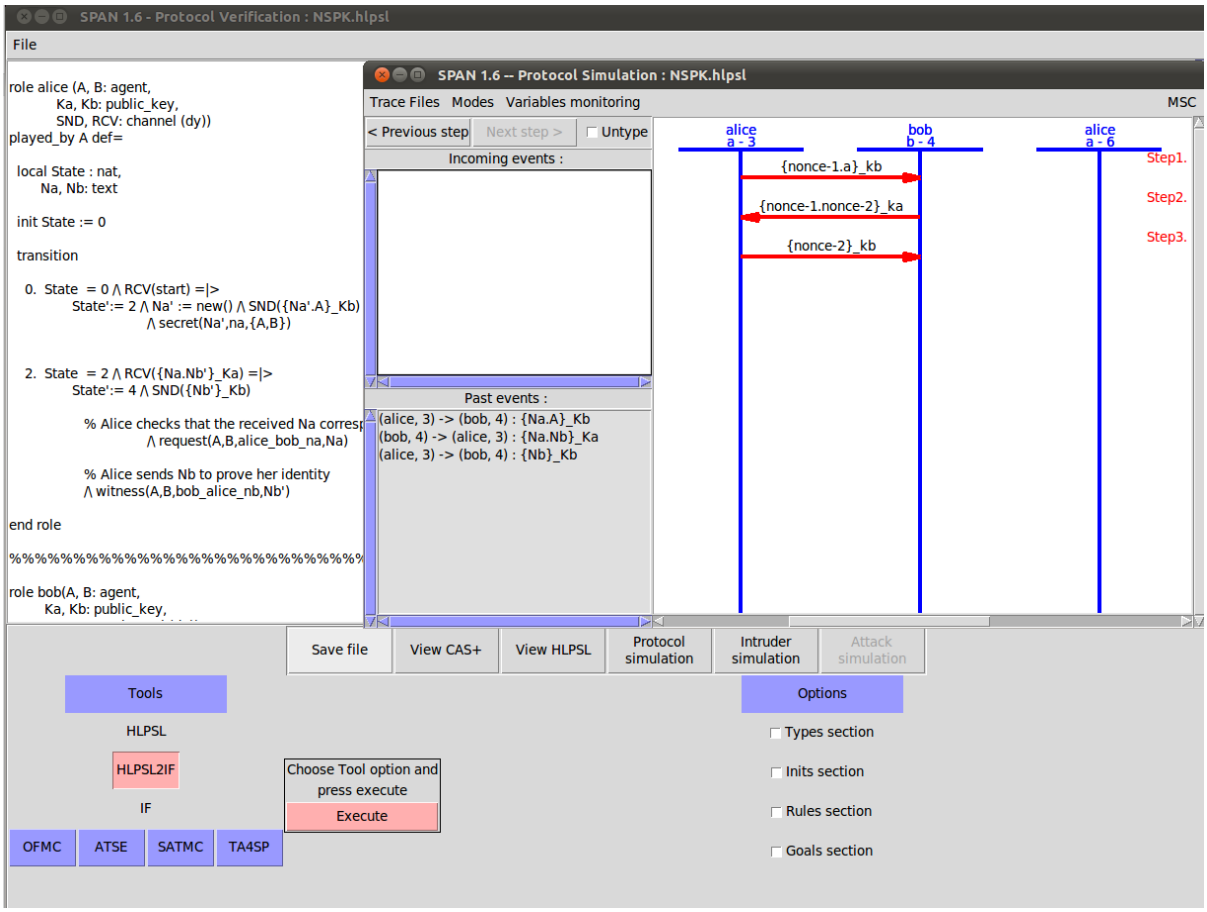


Figure .2: Interface graphique du logiciel AVISPA

L'état initial du IF est calculé de l'instanciation donnée dans le fichier HLPSL (déclarant quels agents sont de jouer quels rôles du protocole avec que). Enfin, les objectifs sont calculés comme un codage basé sur l'état des propriétés données dans le fichier HLPSL.

II.4.3. Langage format intermédiaire IF

Le IF (**I**ntermediate **F**ormat) est un langage de bas niveau, simple mais expressif pour spécifier des protocoles de sécurité et de leurs propriétés. Les spécifications IF peuvent être générées automatiquement par le traducteur HLPSL2IF de spécifications écrites dans le langage de haut niveau HLPSL. La première version de IF a été développé pendant le projet AVISS et cette nouvelle version est une refonte complète de la conception de la langage, basée sur l'expérience que les partenaires de projet ont fait avec une variété de problèmes d'analyse de protocole, afin d'être en mesure d'analyser les protocoles de sécurité Internet et les applications que nous allons considérerons dans le projet AVISPA.

II.4.4. Outils AVISPA

Les outils (ou back-ends) intégrés à AVISPA permettent de vérifier les propriétés de sécurité, selon les besoins. Ils prennent en entrée le format intermédiaire IF.

II.4.4.1. Constraint logic_based Attack Searcher (CL _AtSe)

C'est un outil basé sur des techniques de résolution de contraintes. 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 . Les deux méthodes de la traduction et la vérification sont totalement automatiques et prises en charge par l'outil CL_AtSe sans intervention d'outils externes. Les possibilités de CL–AtSe ont été étendues lors du projet AVISPA pour supporter de vérification des protocoles intégrant les opérateurs algébriques (**Xor** ou **Exp**).

II.4.4.2. On_the_ fly Model_ Checker (OFMC)

Pour but d'analyser la sécurité des protocoles cryptographiques. 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.

II.4.4.3. SAT_based Model_ Checker (SATMC)

Cet outil a été développé au laboratoire DIST à Gènes (Italie) . Il construit une formule propositionnelle codant pour un déroulement borné de la relation de transition spécifiée par IF, l'état initial et l'ensemble des états représentant une violation des propriétés de sécurité .La formule propositionnelle est ensuite à résoudre à un solveur SAT et tout modèle satisfaisant cette formule est retourné sous forme d'attaque.

II.4.4.4. Tree Automata based on Automatic Approximations for the Analysis of Security Protocols (TA4SP):

La particularité de cet outil de vérification est qu'à partir d'un état initial il fait soit une sur_ approximation (si le protocole est sûr pour un nombre illimité de sessions) ou une sous_ approximation (si un protocole est erroné) 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.

Les trois premiers outils cherchent des attaques, alors que le dernier essaie de démontrer la validation des propriétés, tâche beaucoup plus difficile et ne pouvant être appliquée que pour des propriétés relativement restreintes.

II.5. Utilisation de la plateforme AVISPA

L'interaction avec le type d'outil AVISPA est la suivante :

1. On commence par la spécification du protocole à tester grâce au langage HLPSL, ainsi que les propriétés à vérifier.
2. On lance AVISPA à l'aide d'une invite de commandes toute en précisant l'analyseur (back_end) qu'on va utiliser.
3. Ensuite AVISPA, après analyse, il déclare que soit le protocole est sûr, ou bien le protocole présente des failles.

II.6. Le résultat de vérification

Quand l'analyse d'un protocole est réussie, en trouvant ou pas des attaque possible, la sortie de AVISPA décrit précisément le résultat. Le premier résultat obtenu est de résumé les diagnostics de chaque outil de la plateforme AVISPA. Il existe trois types de messages :

- Le message *« UNSAFE»* signifie que le protocole n'est pas sûr et l'outil présente une trace d'attaque,
- Le message « *SAFE*» signifie que le protocole est sûr, et il n'y a pas détection des attaques,
- Le message « *INCONCLUSIVE »* signifie que l'outil n'a pas abouti a un résultat.

Dans l'exemple de protocole *fil-rouge*, le résultat de la vérification de ce protocole avec la plateforme AVISPA est :

```
AVISPA Tool Summary
OFMC    : UNSAFE
CL-AtSe : UNSAFE
SATMC   : UNSAFE
TA4SP   : INCONCLUSIVE
Refer to individual tools output for details
```

Apres avoir affiché un diagnostic général, une deuxième phase consiste à appuyer sur le bouton comportant le nom d'outil afin de visualisé la trace d'attaque s'il existe en détail, ainsi que les informations spécifiques de la vérification tels que :

- La première section *SUMMARY* indique si le protocole est sécurisé ou non, ou si l'analyse n'a pas été concluante,
- La deuxième section *DETAILS* décrit sous quelles conditions le protocole est déclaré sécuriser ou non, sous quelles conditions une attaque est trouvée, et finalement pourquoi l'analyse n'a pas été concluante,
- La section *PROTOCOL* rappelle le nom du protocole analysé,
- La section *GOAL* présente le but de l'analyse, comme par exemple la confidentialité de la clé de chiffrement des données,
- La section *BACKEND* désigne le traducteur des spécifications HLPSL, utilisé lors de l'analyse.
- La section *STATISTIC*S le temps de recherche, le temps d'analyse grammaticale, le nombre de nœuds visités et la profondeur.

Le résultat obtenu peut prendre trois formes : la forme texte, la forme MSC (Message Sequence Chart), ou enregistrer dans un fichier postscript. Et voilà un exemple de résultat de vérification du protocole KN avec l'outil OFMC sous forme texte :

```
% OFMC
% Version of 2006/02/13
SUMMARY
UNSAFE
DETAILS
  ATTACK_FOUND
PROTOCOL
  /home/avispa/web-interface-
computation/./tempdir/workfileThH3Ff.if GOAL
authentication_on_aut_server
BACKEND
  OFMC
COMMENTS STATISTICS
parseTime: 0.00s searchTime:
0.03s visitedNodes: 3 nodes
depth: 2 plies ATTACK
TRACE i -> (c,3): start
(c,3) -> i: idi.h(idi.xs) XOR Nc(1) i ->
(s1,3):idi.h(idi.xs) XOR Nc(1) (s1,3) -> i:h(h(idi.xs)
```

XOR Nc(1)).Nc(1).h(idi.xs) XOR Ns(2) i -> (c,3):
h(h(idi.xs) XOR Nc(1)).Nc(1).h(idi.xs) XOR x250
(c,3) -> i: h(h(idi.xs) XOR x250).x250

La figure 4 illustre le résultat de la vérification du protocole précédent sous forme MSC qui contient la trace d'attaque :

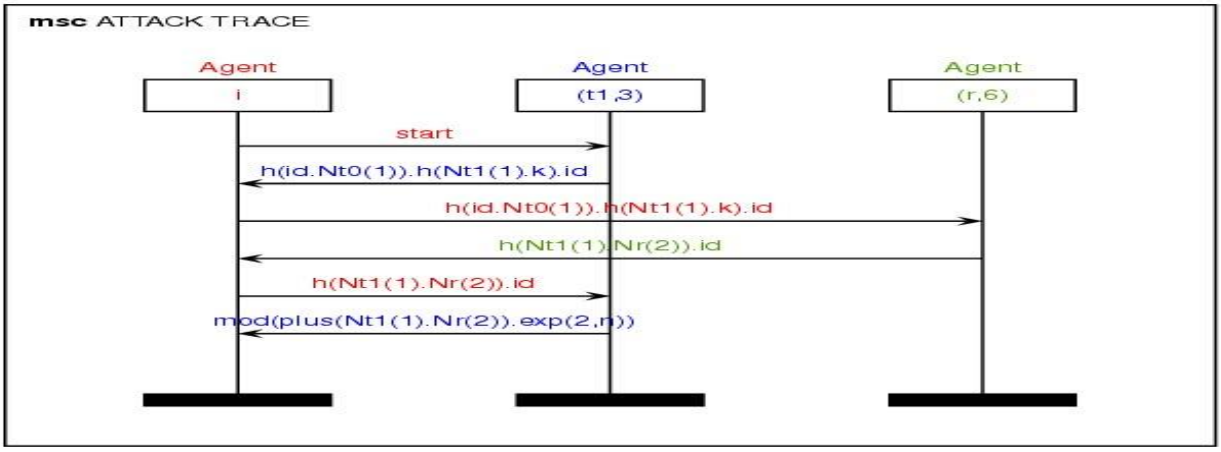


Figure .3 : Trace d'attaque sur le protocole KN (OFMC)

II.7. Outil graphique SPAN

SPAN (Security Protocol ANimator for AVISPA) est un animateur de protocole de sécurité. SPAN permet d'animer les spécifications HLPSL, c'est-à-dire produire interactivement des MSC (Message Sequence Charts) [30] qui peuvent être vus comme une trace "Alice & Bob" d'une spécification HLPSL. Cet outil permet de simuler le protocole, l'intrus et l'attaque.

Dans la Figure II.4, le cadre de droite (**1**) ainsi que dans la fenêtre à gauche **(2**), SPAN affiche les messages déjà envoyés. Dans le cadre de gauche (**3**), on voit les transitions possibles(les envois de messages) qu'on peut déclencher d'un double-clique. S'il n'y a plus de transitions, c'est qu'on est arrivé à la fin du protocole ou qu'il y a une erreur dans la spécification HLPSL.

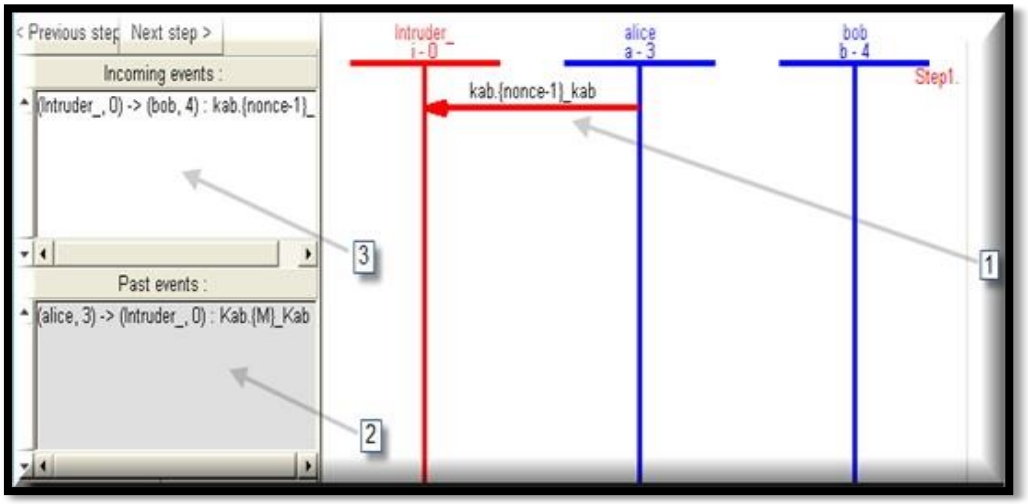


Figure 4 : L'animation de la spécification HLPSL de *fil-rouge*

SPAN a plusieurs utilités dont celle de vérifier que la spécification HLPSL correspond bien au protocole qu'on avait en tête avant même de chercher des attaques. La plus grande utilité de SPAN est la possibilité de reconstruire des attaques fournies par les outils de vérification. SPAN permet à l'utilisateur de choisir à chaque étape si les messages sont reçus par l'intrus ou non et affiche dans un cadre les connaissances de l'intrus.

II.8. Synthèse

Les motivations choisies pour la plateforme AVISPA pour vérifier les protocoles de sécurité sont multiples, elles sont en majorité :

- La nature du langage de spécification HLPSL, tel qu'un langage devient facile à manipuler, haut niveau, modulaire et expressif pour les protocoles et les propriétés de sécurité vérifiées.
- La plateforme AVISPA n'est pas un seul outil de vérification automatique, mais elle contient quatre back-end et chaque back-end possède des techniques particulières pour vérifier le protocole. Pendant la vérification, la considération des nombres des sessions est facteur important, les outils AVISPA sont partagés en deux catégories :
    1. OFMC, CL-Aste et SATMC (nombre fini de sessions),
    2. TA4SP (nombre non borné de sessions).
- La forme de la plateforme AVISPA est en plusieurs modes (Interface Web, mode Mac, Linux), ceci permet d'interaction avec les utilisateurs

الملخص:

عملنا يتعلق بالتحقق الآلي المبني على الصيغ لبروتوكولات التشفير. في هذه المذكرة، نقدم نمذجة بلغة HLPSL ونستخدم أرضية AVISPA لتحقق من سلامة الفئتين من البروتوكولات الأمنية: بروتوكولات أنظمة البطاقات الذكية و بروتوكولات المصادقة . الخصائص الأمنية التي تم تحليلها هي: السرية والمصادقة.

وفي الأخير أجرينا تحليللمختلف البروتوكولات المدروسة من ناحية درجة التعقيد في استعمال الأدوات التشفيرية والجبرية المستعملة في البروتوكولات

الكلمات المفتاحية: البروتوكول الأمني، الخصائص الأمنية، التحقق الآلي، ارضية AVISPA، لغة HLPSL

Abstract:

Our work focuses on the automatic formal verification of cryptographic protocols. In thispaper, we present a modeling language in HLPSL and using the AVISPA platform forverification of two categories of security protocols: the protocols of RFID systems and theprotocols of smart card systems. We verify the following properties: the confidentiality ofsecret data and authentication of the entities of the system. Finally, we present a analysisthe different protocols studied in term complexity of implementation ofcryptographic primitives and algebraic.

Key-words: security protocol, security properties, automatic verification, AVISPA platform,HLPSL language.

Résumé :

Notre travail s'article sur la vérification formelle automatique des protocolescryptographiques. Dans ce mémoire, nous présentons une modélisation en langage HLPSL etNous vérifions les propriétés de sécurités suivantes : la confidentialité des données secrètes etl'authentification des entités du système. Enfin nous présentons une étude les différents protocoles étudies en terme de complexité d'implémentation des primitivecryptographique et algébrique.

Mots-clés : Protocole de sécurité, propriétés de sécurité, vérification automatique, plateformeAVISPA, langage HLPSL