République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option : *Réseaux et Systèmes Distribués (R.S.D)*

Thème

GESTION DE CLES DANS LES RESEAUX DE CAPTEURS CORPORELS SANS FIL (WBAN)

Réalisé par :

- ***SEBBAH*** *ABDERREZAK*
- ***CHERRAK*** *SOUFIANE*

Présenté le 02 Juin 2016 devant le jury composé de MM.

- *Mr* ***BENAMAR*** *Abdelkrim* *(Président)*
- *Mr* ***MANA*** *Mohammed* *(Encadreur)*
- *Mr* ***BENMAMMAR*** *Badr* *(Examinateur)*
- *Mr* ***LEHSAINI*** *Mohammed* *(Examinateur)*

Année Universitaire : 2015-2016

Remerciement

Ce travail, intitulé «Gestion de clés dans les réseaux capteurs corporels sans fil (WBANs)» s'inscrit dans le cadre d'un projet de fin d'étude mené au niveau du Département d'informatique de la Faculté des sciences de l'Université Abou-Bekr Belkaïd de Tlemcen, pour l'obtention du diplôme de Master en Informatique, option Réseaux et Systèmes Distribués.

Au terme de ce projet, nous tenons à remercier Monsieur **M.MANA** Maitre de conférence à l'Université de Tlemcen, pour l'honneur qu'il nous à fait de bien vouloir nous encadrer, pour l'aide et le temps qu'il nous a bien voulu nous consacré et pour les conseils qu'il nous a donné lors de la réalisation de ce manuscrit.

Nous tenons à remercier également Monsieur **A.BENAMAR** d'avoir accepté de présider le Jury, Monsieur **LEHSSAINI** et Monsieur **BENMAMMAR** d'avoir accepté d'examiner ce travail.

Nous adressons nos plus sincère remerciement à tous nos proches et nos collègues qui nous ont toujours soutenus et encouragé.

Enfin, nous tenons à exprimer notre profonde gratitude à toutes celles et ceux qui nous ont apporté leur soutien, leur amitié ou leur expérience tout au long de ce travail de mémoire.

Dédicace

Je tiens à exprimer ma profonde reconnaissance A DIEU : pour m'avoir donné la force dans les moments difficiles d'éditer ce mémoire.

A mes parents:

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit. Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi.

A mes sœurs et mes frères a mes amis et amies de par le monde qui n'ont cessé de m'encourager.

A mon professeur encadreur Mr MANA pour son aide et sa précieuse attention.

SEBBAH ABDEREZAK

Dédicace

Je tiens à exprimer ma profonde reconnaissance A DIEU : pour m'avoir donné la force dans les moments difficiles d'éditer ce mémoire.

A mes parents:

Ma mère, qui a œuvré pour ma réussite, de par son amour, son soutien, tous les sacrifices consentis et ses précieux conseils, pour toute son assistance et sa présence dans ma vie.

Mon père, qui peut être fier et trouver ici le résultat de longues années de sacrifices et de privations pour m'aider à avancer dans la vie. Puisse Dieu faire en sorte que ce travail porte son fruit. Merci pour les valeurs nobles, l'éducation et le soutient permanent venu de toi.

A mes sœurs et mes frères a mes amis et amies de par le monde qui n'ont cessé de m'encourager.

A mon professeur encadreur Mr MANA pour son aide et sa précieuse attention.

CHERRAK SOUFIAN

Sommaire

Sommaire

Introduction générale

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 permis de produire à coût raisonnable des capteurs de quelques millimètres cubes de volume, susceptibles de fonctionner en réseau appelé communément réseau de capteur sans fil.

Dans un scénario d'application classique, les capteurs sont déployés dans un champ d'intérêt afin de mesurer certains phénomènes physiques et de faire remonter les informations collectées à une station de base, nommée le nœud puits. Ce dernier a plus de ressources que les autres nœuds et peut traiter les informations reçues localement. Le nœud puits est administré par un utilisateur via un réseau externe (internet, satellite, etc.).

Les domaines d’application des réseaux de capteurs sans fil sont nombreux et variés du fait notamment de la variété des capteurs. Parmi les domaines qui ont été révolutionnés par les réseaux de capteurs sans fil, on trouve le domaine médical.

Les réseaux de capteurs sans fil utilisés dans le domaine médical sont appelés réseaux de capteurs corporels ou simplement WBANs (*Wireless Body Area Networks)*. Ces dispositifs ont révolutionné les systèmes de télémédecine en améliorant la qualité du soin et en réduisant les coûts énormes associés à des patients ambulants dans les hôpitaux. Grâce aux réseaux de capteurs corporels, la surveillance des patients peut avoir lieu en temps réel même en dehors de l'hôpital (dans l'environnement familier, voire professionnel, ce qui est devenu le souhait de tout patient) et sur une plus longue période.

La nature vulnérable des communications radios est une caractéristique qui augmente les risques de menaces et d'attaques contre les réseaux de capteurs corporels. Ces menaces et ces attaques peuvent poser de sérieux problèmes à la vie sociale de l'individu portant ces dispositifs de captage. Les gens ayant une intention malveillante peuvent utiliser les données privées du patient pour mettre sa vie en danger. Donc il est nécessaire et très important de protéger les réseaux de capteurs corporels afin de garantir une meilleure acceptabilité de ses systèmes par la communauté humaine.

La mise en place d'un système de sécurité pour les réseaux de capteurs corporels est un défi, car les capteurs sont limités en termes de complexité de calcul, de mémoire de stockage et d'énergie. En outre, le réseau a une topologie Ad hoc d'où l'absence d'une d'une infrastructure fixe. Toutes ces contraintes rendent impossible l'application des méthodes traditionnelles de sécurité aux réseaux de capteurs corporels. Le développement des solutions de sécurité légères mais efficaces est une nécessité.

Notre travail consiste à développer un protocole efficace et léger de gestion de clés dans les réseaux de capteurs corporel tout en tenant compte des limites des capteurs en termes de complexité de calcul, de stockage et particulièrement de l'énergie afin de garantir une longue durée de vie du réseau.

Ce manuscrit est organisée en trois chapitres en plus d'une introduction générale et d'une conclusion générale.

Le premier chapitre présente les réseaux de capteurs sans fil et en particulier les réseaux de capteurs corporels.

Dans le deuxième chapitre nous abordons en premier lieu les attaques et menaces qui peuvent compromettre le réseau corporel sans fil et les solutions introductives de gestion de clés. En deuxième lieu, nous présentons en détail notre mécanisme de gestion de clés dans les réseaux de capteurs corporels.

Le chapitre trois présente les résultats de simulation de notre protocole de gestion de clés dans les réseaux de capteurs corporels.

Chapitre I
Réseaux de capteurs corporels sans fil

I.1 – Introduction

L'évolution dans le domaine des communications sans fil et l'informatique mobile gagne de plus en plus de popularité et les composants mobiles deviennent de plus en plus fréquents (PDA, LAPTOPS, HANDSETS….). Ceci a permis l'apparition d'un nouveau type de réseaux sans fils appelé réseaux de capteur sans fils (RCSF en Français ou WSN en anglais) qui sont devenu de plus en plus populaires du fait de leur facilité de déploiement. Les WSN ont révolutionné de nombreux secteurs et notamment le secteur médical. En effet, ils permettent de contrôler les patients à distance. Les WSN appliqués au domaine médical sont appelés réseaux de capteurs corporels sans fil ou WBAN (Wireless Body Area Network).

Dans ce chapitre, nous allons présenter brièvement les réseaux de capteurs sans fil WSN et notamment les réseaux de capteurs corporels sans fil WBANs.

I.2 – Les réseaux de capteurs sans fil

Un réseau de capteurs sans fil (RCSF) est constitué d'un grand nombre d'entités autonomes communément appelées «capteurs sans-fil» **[1].** Dans un scénario d'application classique, les capteurs sont déployés dans un champ d'intérêt afin de mesurer certains phénomènes physiques et de faire remonter les informations collectées à une station de base, nommée le nœud «puits» .Ce dernier est administré par l'utilisateur via un réseau externe (satellite, internet,…).

Dans le cas d'un réseau de petite échelle (corps humain par exemple), les capteurs seront dans le voisinage direct du puits (un réseau de type étoile à un saut). Cependant, dans le cas d'un réseau à grande échelle (forêt, barrage, champs de batail…), les capteurs ne sont pas tous dans le voisinage du puits et les messages seront acheminés du nœud source vers le puits en transitant par plusieurs nœuds, selon un mode de communication multi-sauts comme l'illustre la figure I-1.

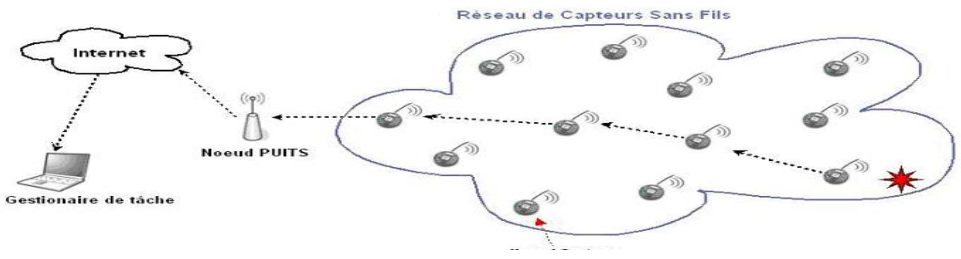

Figure I.1- *Les réseau de capteurs sans fil* ***[1].***

I.3 – Architecture d'un capteur sans fil

Comme l'illustre la figure I-2, un nœud capteur est composé fondamentalement de quatre unités élémentaires **[2] [3] [4]:**

- Unité de Captage: Ce composant est l'unité qui contient le ou les capteurs embarqués sur le nœud. Habituellement, un convertisseur analogique-numérique (CAN) convertit les signaux provenant des capteurs (signaux analogiques) en signaux interprétables par l'unité de traitement (signaux numériques).
- Unité de Traitement: Elle est généralement constituée d'un microcontrôleur dédié et d'une mémoire. Les microcontrôleurs utilisés dans ce cadre sont à faible consommation d'énergie. Leurs fréquences sont assez faibles, moins de10 MHz pour une consommation de l'ordre de 1 mW. La taille de la mémoire est de l'ordre de 10 Ko de RAM pour les données et de 10 Ko de ROM pour les programmes **[3].** Cette unité fournit aux capteurs la capacité d'exécuter des calculs sur les données et les conserver selon un scénario programmé.
- Unité de Communication: Elle est le plus souvent constituée d'un transmetteur radio qui fournit au capteur la capacité de communiquer avec les autres au sein du réseau. Elle met en œuvre des protocoles de communication sans fil dépendant de la technologie utilisée (par exemple 802.11, 802.15.1, 802.15.4, etc.).
- Unité de Puissance: c'est la source d'énergie qui alimente le reste des unités. Cette unité se trouve généralement sous la forme de batteries standards de basse tension.

Les capteurs peuvent également avoir d'autres modules en fonction des applications pour lesquelles ils sont conçus. Par exemple, les nœuds peuvent avoir une unité de localisation afin d'identifier leur position géographique en utilisant un récepteur GPS. Aussi, certains capteurs peuvent être équipés d'un mobilisateur pour qu'ils puissent se déplacer.

Enfin, s'il est nécessaire qu'un nœud soit maintenu en activité pendant une très longue période de temps, un générateur de puissance, tel que des cellules solaires, serait utile afin de tenir le nœud alimenté électriquement sans avoir à changer ses batteries **[4].**

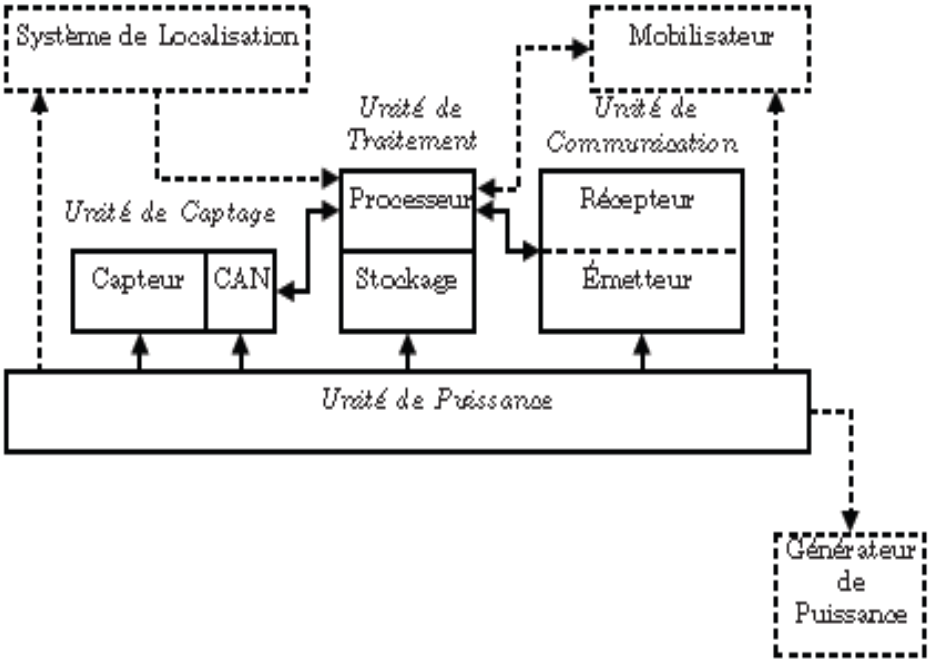

Figure. I-2 Architecture générale d'un nœud de capteur **[2].**

I.4 –Systèmes embarqués pour les capteurs

Les systèmes embarqués sont des systèmes d'exploitation prévus pour fonctionner sur des machines de petite taille, telles que des nœuds de capteurs. Les systèmes d'exploitation pour les nœuds de RCSF sont généralement moins complexes que les autres systèmes d'exploitation. Ceci à cause des exigences particulières des applications de réseau de capteurs et des contraintes de ressources des nœuds de capteurs. Plusieurs systèmes d'exploitation sont conçus pour les nœuds de RCSF. Parmi ces systèmes nous citons TinyOS **[5]**, Contiki**[6]** et MANTIS OS **[7]** .

I.4.1 – Contiki

Contiki est un système d'exploitation open source. C'est un système configurable modulaire pour les réseaux de capteurs. L'architecture hybride du noyau Contiki autorise deux modes de fonctionnement, soit multitâche ou soit basé sur les événements. Contiki est un système d'exploitation conçu pour prendre le moins de place possible, avec une faible empreinte mémoire. Pour cela, le code est écrit en langage C. Un système utilisant Contiki contient des processus, qui peuvent être des applications ou des services, c.à.d. un processus proposant des fonctionnalités à une ou plusieurs applications. La communication entre processus se fait par l'envoi d'événements.

Le noyau Contiki reste, nativement, un système d'exploitation basé sur les événements. Pour obtenir le mode multitâche, une bibliothèque doit être installée. Les fonctions associées à cette bibliothèque n'accèdent pas directement à l'ensemble des ressources du capteur sans fil. Elles doivent, dans certains cas, faire appel à la partie du noyau dédié à la gestion des événements. Cette structure à deux niveaux a pour conséquence une dégradation des performances du système quand le mode multitâche est activé.

I.4.2 – TinyOS

TinyOS est un système d'exploitation open source pour les réseaux de capteurs sans fil qui trouve sa genèse au sein du laboratoire d'informatique de l'université de Berkeley et qui a été l'un des premiers systèmes d'exploitation conçus pour les réseaux de capteurs miniatures. En effet, TinyOS est le plus répandu des OS pour les réseaux de capteurs sans-fil. Il est capable d'intégrer très rapidement les innovations en relation avec l'avancement des applications et des réseaux eux-mêmes tout en minimisant la taille du code source en raison des problèmes inhérents de mémoire des capteurs.

La conception de TinyOS a été entièrement réalisée en NesC, langage orienté composant syntaxiquement proche du C. La bibliothèque des composants de TinyOS est particulièrement complète puisqu'on y retrouve des protocoles réseaux, des pilotes de capteurs et des outils d'acquisition de données. Un programme s'exécutant sur TinyOS est constitué d'une sélection de composants systèmes et de composants développés

spécifiquement pour l'application à laquelle il sera destiné (mesure de température, du taux d'humidité, etc.).

TinyOS s'appuie sur un fonctionnement événementiel, c'est-a-dire qu'il ne devient actif qu'a l'apparition de certains événements ; par exemple, l'arrivée d'un message radio. Le reste du temps, le capteur se trouve en état de veille, garantissant une durée de vie maximale connaissant les faibles ressources énergétiques des capteurs. Cependant, l'allocation statique de la mémoire et la perte des composants lors de la génération de l'exécutable, constituent les limites de ce système et rendent la reconfiguration dynamique de l'image présente sur le capteur impossible.

I.4.3 – MANTIS OS

MANTIS (MultimodAl NeTworks of In-situ micro Sensor) OS apparu en 2005, a été conçu par l'université du Colorado . C'est un système d'exploitation léger et multitâche pour les capteurs, adapté aux applications où plusieurs traitements, chacun associé à un ou plusieurs processus, sont en concurrence pour accéder aux ressources du capteur sans fil.

Il dispose d'un environnement de développement Linux et Windows. La programmation d'application sur MANTIS OS se fait en langage C. Son empreinte mémoire est faible : 500 octets en mémoire RAM et 14 kilo-octets en mémoire flash. C'est un système modulaire dont le noyau supporte également des entrées/sorties synchrones et un ensemble de primitives de concurrence.

L'économie d'énergie est réalisée par MANTIS à l'aide d'une fonction de veille appelée sleep function qui désactive le capteur lorsque toutes les taches actives sont terminées . MANTIS est un système dynamique ; les modifications applicatives peuvent être réalisées pendant le fonctionnement. MANTIS apporte une compatibilité avec le modèle événementiel TinyOS à travers TinyMOS (MOS est la contraction de MantisOS), dont son noyau est équipé.

I.5 – Protocoles de communications sans fil

Parmi les grandes normes radios qui ont été utilisées pour des applications à bases de réseaux de capteurs nous citons Bluetooth/Wibree /ZigBee/Wifi **[8] [9] [10]:**

I.5.1 – La norme IEEE 802.15.1 / Bluetooth

Initialement, la norme Bluetooth a été proposée pour transmettre la voix et les données elle avait pour objectif préalable de permettre des communications sur de courtes distances avec un débit de communication limitée.

Ses caractéristiques ont ainsi retenu l'attention des développeurs de capteurs. Par exemple les capteurs BtNode sont conçus pour une communication de type Bluetooth. Pour autant, le protocole Bluetooth n'est pas le protocole le plus utilisé dans les réseaux de capteurs, bien qu'il puisse répondre en partie aux problèmes de préservation de l'énergie, car il est gravement handicapé par la taille limitée du réseau qu'il peut former .

I.5.2 – La norme Wibree (Ultra Low Power Bluetooth)

Elle est considérée comme une version allégée de la norme Bluetooth fonctionnant dans la bande de fréquence des 2,4 GHz. Wibree n'utilise pas de sauts de fréquences, Cette norme prend en charge une topologie en étoile avec un maître et sept esclaves**.**

Afin de réduire la consommation d'énergie de Bluetooth, Wibree utilise une puissance de transmission et un débit symbole faibles. La consommation d'énergie de Wibree est l'équivalent de 10% de celle d'une connexion par Bluetooth. Sa limite principale est la faible portée de communication: (5 -10 m).

I.5.3 – La norme IEEE 802.15.4 / Zigbee

Elle est conçue pour être utilisée dans les communications à très faible puissance et sur des distances réduites. Cette technologie est utilisée dans les réseaux de capteurs sans fil **[10].** Par rapport à Bluetooth, cette technologie fournit une faible latence ; une couche physique « DSSS : Direct Sequence Spread Spectrum » permet aux nœuds de basculer en mode sommeil sans perdre la synchronisation. Le protocole Zigbee est basé sur le standard

IEEE 802.15.4 qui définit sa couche PHY et MAC et qui permet de prolonger théoriquement la durée de vie d'un nœud sur plusieurs années. L'autre point fort de ce protocole est qu'il propose le déploiement de réseau dense à plus de 65000 nœuds avec une portée de l'ordre de 100 mètres pour un débit de 250 Kbits/s. Ces caractéristiques en font aujourd'hui le principal protocole utilisé dans les réseaux de capteurs.

I.5.4 – La norme IEEE 802.11x/WiFi

Le protocole de communication WiFi est le protocole le plus utilisé pour toutes les applications sans fil. Il offre une large bande passante (de 11 à 320 Mbits/s) ce qui a permis de démocratiser l'utilisation de la technologie sans-fil dans les réseaux classiques WLANs.

Les premiers capteurs sans-fil ont eu recours à ce protocole pour permettre la communication entre nœuds. Cependant, le standard de communication WiFi n'apparait plus actuellement comme une solution viable pour les réseaux de capteurs sans fil, du fait d'un besoin énergétique trop important pour son utilisation. La durée de vie des capteurs sans fil alimentés par des piles ne dépasse que rarement quelques heures.

C'est pourquoi, les applications de capteurs à base de communication sans fil WiFi sont très peu répandues.

La figure suivante illustre le positionnement des différents standards de communication.

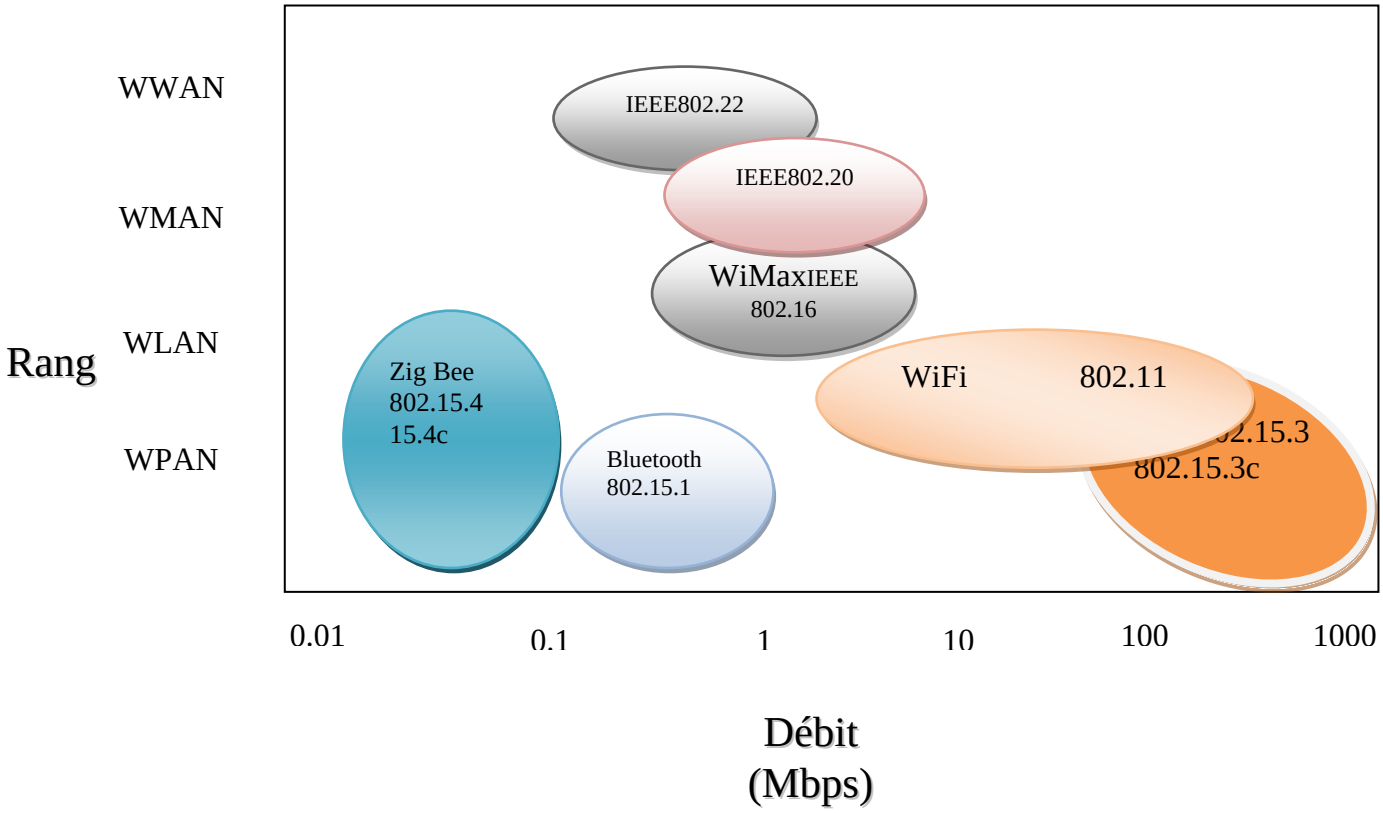

Figure I.3- *Différents standards de communication* ***[8].***

I.6– Application des réseaux de capteurs sans fil

Il existe plusieurs domaines d'applications des réseaux de capteurs parmi lesquels on peut citer les applications militaires, environnementales, industrielles et de surveillance en général **[11] [12] [13].**

- **Applications militaires :** Les réseaux de capteurs sans fil peuvent constituer des modules intégrés dans les systèmes militaires de commandes, contrôle, communication, calcul, intelligence, surveillance, reconnaissance et ciblage. Ces systèmes sont appelés communément par les systèmes C4ISRT. De plus, il existe d'autres applications militaires auxquelles les réseaux de capteurs peuvent être appliqués.

- **Ciblage** les réseaux de capteurs peuvent être également incorporés dans les systèmes de guidage des munitions intelligentes.

- **Détection des incendies de forêts** les capteurs peuvent être déployés d'une façon aléatoire, et dense dans n'importe quel type de forêt ce qui permet facilement de détecter tout incendie avant qu'il se propage et devienne incontrôlable.

- **Détection des inondations** Les réseaux de capteurs peuvent être également utilisés pour détecter les inondations. Parmi les exemples d'utilisation des réseaux de capteurs pour la détection des inondations on trouve le système ALERT (Automated Local Evaluation in Real Time) déployé aux Etats Unies d'Amérique. Ce dernier englobe plusieurs types de capteurs hydrologiques qui détectent la pluie, le niveau d'eau, ainsi que d'autres capteurs météorologiques qui servent à la détection de la température, la pression, etc.
- **Agriculture** Les réseaux de capteurs sont capables d'apporter des bénéfices considérables au domaine de l'agriculture, grâce à leur habilité de surveiller les taux de pesticides dans l'eau potable, le degré d'érosion du sol, et le niveau de pollution de l'air en temps réel.

Le suivi et la surveillance des médecins et des patients au sein de l'hôpital
Chaque patient aura des petits nœuds capteurs légers qui lui sont attachés où chacun de ces nœuds aura sa tache spécifique, par exemple, l'un des capteurs peut être employé pour détecter les battements du cœur et un autre pour la pression du sang, etc.,…

Autres applications d'autres domaines d'application des réseaux de capteurs peuvent être envisagées. Dans la construction, on y trouve des applications de surveillance des structures de bâtiments, ainsi que des applications liées à l'automatisation des maisons, le pilotage de robots, etc.

Figure I.4- Exemples de surveillance militaire, environnementale, médical **[11].**

I.7–Les réseaux WBAN

I.7.1–Définition

Un réseau de capteurs corporels sans fil ou WBAN est un réseau constitué de mini-capteurs portés ou implantés dans le corps humain. Chaque nœud capteur est

généralement capable de détecter une ou plusieurs caractéristiques physiologiques à partir du corps humain ou de son environnement et de faire remonter les informations collectées au nœud puits. Comme ce dernier a plus de ressources que les autres nœuds, il peut traiter localement les informations reçues et générer des alarmes si nécessaire (figure I-14). La communication entre les nœuds du réseau WBAN et le puits se fait sans fil [**14]**.

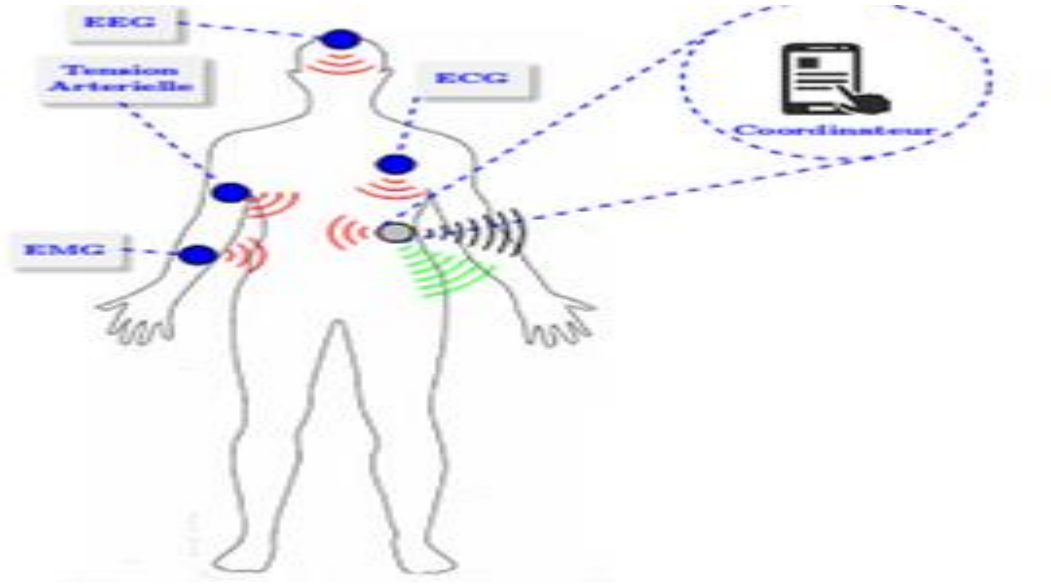

Figure I.5- Un WBAN **[14].**

I.7.2–Architecture d'un WBAN

Le schéma de la figure I-6 présente l'architecture générale d'un système de soin de santé (télémédecine) **[14].** Il est constitué du réseau WBAN, du réseau externe et le serveur médical.

Le réseau WBAN permet de collecter des informations médicales sur le sujet et de transmettre ces informations au serveur médical via le réseau externe qui peut être filaire ou sans fil (Internet, GSM,…etc.). Le nœud puits joue le rôle d'une passerelle entre le réseau WBAN et le réseau externe. Le serveur médical stocke, traite et gère efficacement l'énorme quantité de données biomédicales des patients. Ces données médicales sont ensuite observées et analysées par un médecin.

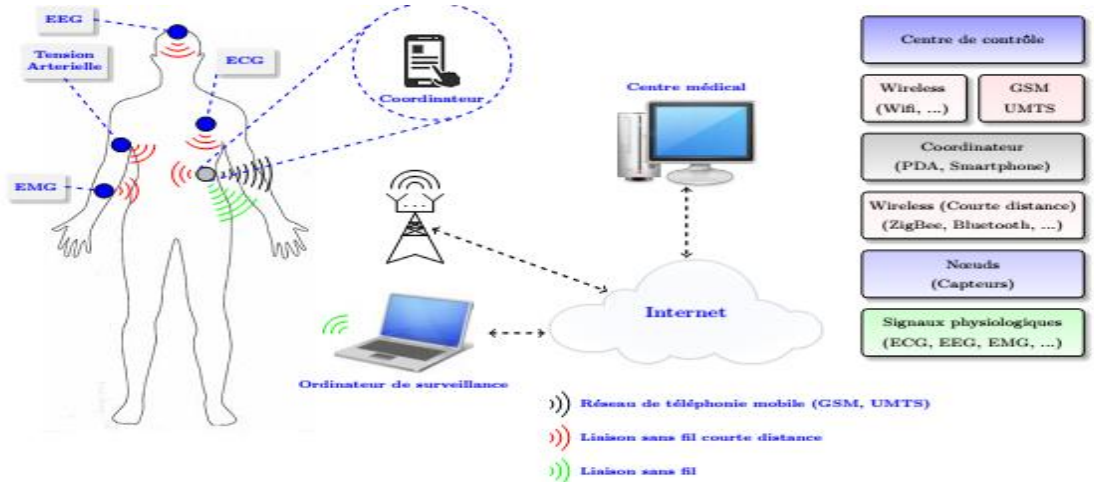

Figure I.6- Architecture d'un système de surveillance médical **[14].**

I.7.3–Différence entre WBAN et WSN

Dans ce tableaux nous présentons les différences entre WBAN et WSN qui sont classifiées selon plusieurs facteurs.

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 I.1- Différences entre WBAN et WSN

I.7.4– 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, maille et arbre.

I.7.4.1 – 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.

I.7.4.2 – Topologie en étoile

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 majeur est la vulnérabilité du nœud central.

I.7.4.3 – 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 la portée de transmission. Un nœud voulant transmettre un message à un autre nœud hors de sa portée, peut utiliser un nœud intermédiaire pour véhiculer son message au 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.

I.7.4.4 – Topologie en arbre

Une topologie en arbre (Tree en anglais) contient un sommet avec une structure de branches au-dessous. Les connexions entre les nœuds sont structurées hiérarchiquement, ce qui signifie que chaque nœud peut être un fils à un nœud de niveau supérieur et un père à un nœud de niveau inférieur.

Cette topologie divise le réseau en sous-parties de sorte qu'il devient plus facile à gérer. Elle présente une bonne tolérance aux fautes, une bonne couverture, une bande passante élevée et une faible latence. Mais toutefois, les nœuds pères peuvent consommer beaucoup d'énergie.

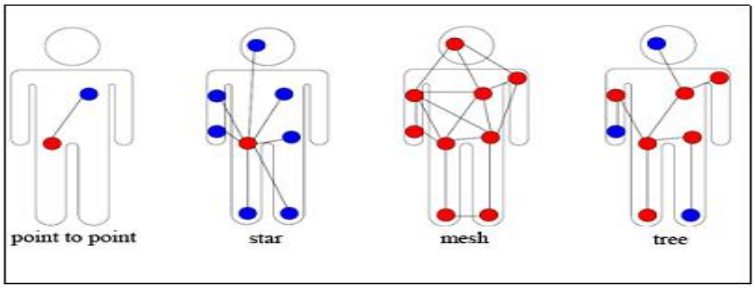

Figure I.7- Les topologies dans les réseaux WBAN.

I.8–Conclusion

Les réseaux WBANs ont amélioré considérablement le secteur médical. En effet, ils permettent une surveillance à distance et en temps réel des patients par l'équipe médicale ce qui permet une intervention immédiate en cas de nécessité.

Les WBANs sont en toute évidence exposés à différentes menaces et attaques vu la nature du support de transmission sans fil. En outre, l'information communiquée dans les réseaux WBANs est cruciale car elle comprend des mesures médicales sur la personne portant ce type de réseau. Donc, cette information doit être gardée secrète et confidentielle afin de ne pas mettre la vie de la personne en danger. La protection des réseaux WBANs contre les attaques et menaces est déterminante pour l'acceptabilité de ces systèmes par les individus. Nous abordons dans le chapitre suivant d'une part les attaques et menaces qui peuvent compromettre le réseau WBAN et d'autre part, nous présentons en détail notre mécanisme de gestion de clés pour les réseaux WABNs.

Chapitre II
Gestion de clés dans les réseaux corporels sans fil

II.1 – Introduction

La sécurité des réseaux WBANs est un enjeu majeur car d'une part, les informations collectées par les nœuds capteurs sont vitales et doivent être sécurisée et d'autre part le support de transmission est sans fil ce qui permet à n'importe qui de capter les informations échangées entre les nœuds capteurs et la station de base.

Un autre défi majeur consiste à développer des solutions de sécurité adaptées car les mécanismes de sécurité traditionnelles ne sont pas applicables dans le contexte des WBANs parce qu'ils nécessitent des ressources gourmandes en termes de stockage, traitement et énergie.

Dans ce chapitre, et en premier lieu, nous allons brièvement présenter l'aspect sécurité dans les réseaux WBANs, les défis à relever et les problèmes de sécurité. En deuxième lieu, nous allons présenter notre protocole de gestion de clés pour les réseaux WBANs.

II.2 – Menaces et attaques

Quelques menaces sont inhérentes aux WBANs, d'autres liées a la technologie retenue. Nous distinguons deux catégories: passives et actives.

- **Attaques passives*:*** Dans ce type d'attaques, généralement, l'attaquant passe inaperçu. En effet, son objectif est d'écouter le réseau sans chambouler ou altérer son fonctionnement. Pendant ce temps, aucun paquet de données n'est émis sur le réseau par l'attaquant ce qui rend sa détection très difficile. L'objectif de ces attaques est d'analyser les paquets de données circulant sur le réseau et d'extraire des informations précieuses.

- **Attaques actives:** Contrairement aux attaques passives, les attaques actives visent à modifier l'état du réseau.

II.3 – Objectifs de la sécurité

Lorsque nous abordons le problème de sécurité, nous visons à atteindre certains objectifs, dont les principaux sont les suivants :

- **Confidentialité des données** : consiste à rendre les informations inintelligibles.
- **Intégrité des données :** L'assurance que l'information n'est ni modifiée, ni altérée, ni détruite de façon erronée ou sans autorisation. **L'authentification :** Savoir et pouvoir vérifier l'identité d'un nœud qui veut communiquer avec un autre nœud.
- **La Localisation** (traçabilité): Le but est de garder privé l'emplacement physique des capteurs.
- **La non répudiation :** Un ensemble de processus, règles et mécanismes permettant d'associer de façon irréfutable un paquet à sa source. Le nœud ne peut pas nier avoir envoyé un paquet.
- **L'anonymat :** Le but est de prévenir la révélation de l'identité d'un nœud qui a performé une action à l'intérieur d'un groupe de nœuds qui ont des attributs identiques.

II.4 – défi de la sécurité

L'essentiel défi consiste à minimiser la consommation de l'énergie tout en maximisant les performances de sécurité. En effet, ces performances et les mécanismes de sécurité utilisés sont fortement influencés par les capacités et les contraintes du nœud capteur. L'énergie constitue l'essentiel des capacités du nœud [**15**]. La plus grande partie d'énergie consommée par un nœud pour assurer la sécurité est liée aux :

- Calcul requis pour les fonctions de sécurité, tels que le chiffrage, le déchiffrage,
- La signature des données et la vérification de la signature.
- Energie requise pour le stockage des paramètres de sécurité, tel que le stockage de la clé de chiffrement.
- Nombre de messages échangé

II.5 – Techniques cryptographiques

La cryptographie protège le réseau uniquement contre les attaques externes .Elle comprend un ensemble de techniques qui sont fréquemment utilisées dans le monde informatique pour assurer la confidentialité, l'intégrité et l'authenticité des données. L'application de la cryptographie implique souvent des calculs intensifs et la gestion des données volumineuses, qui ne posent aucun problème pour des plateformes possédant une puissance de calcul suffisante et un accès de mémoire rapide **[16] [17].**

On distingue deux types de système cryptographique, les systèmes cryptographiques symétriques et les systèmes cryptographiques asymétriques.

II.5.1 – Système cryptographique symétrique

Dans les systèmes cryptographiques symétriques la clé utilisée dans le processus de chiffrement et la même que la clé utilisée dans le processus de déchiffrement.

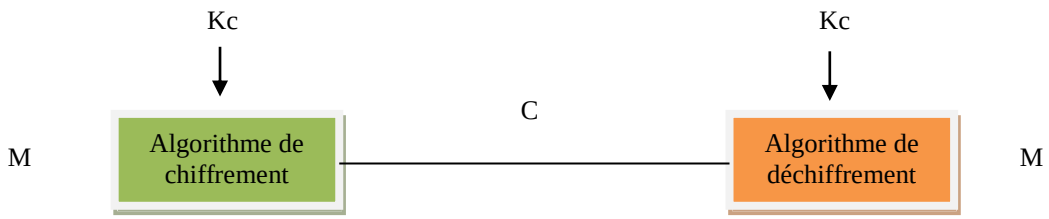

Figure II.1- Schéma de système cryptographique symétrique **[17]**

II.5.2 – Système cryptographique asymétrique

Un système cryptographique asymétrique repose sur l'utilisation d'une clé publique qui est diffusée) et d'une clé privée gardée secrète, l'une permettant de coder le message et l'autre de le décoder.

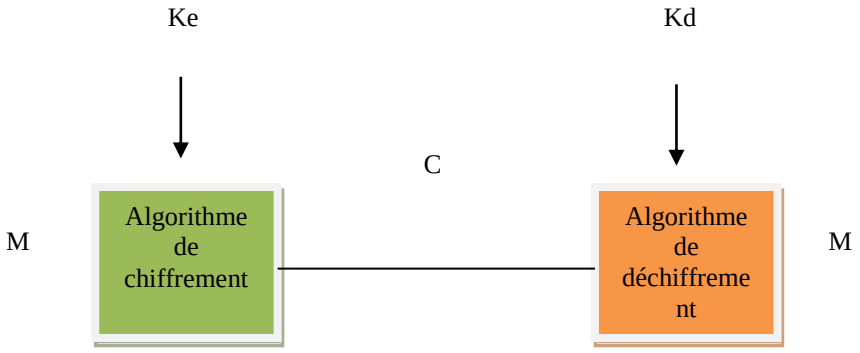

Figure II.2- Schéma de système cryptographique asymétrique **[17].**

II.6 – 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 **[18].**

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.

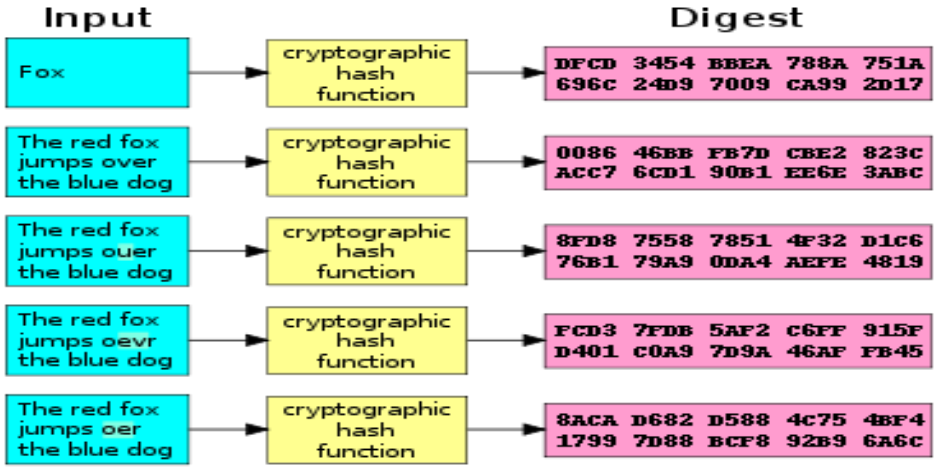

Figure II.3- Fonction de hachage**[18]**.

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.

  Base64 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.

II.7 – Mécanisme de gestion de clés dans les WBANs

La gestion de clés vise à offrir les quatre foncions principales suivantes : la génération, la distribution, le stockage et la vérification de clés

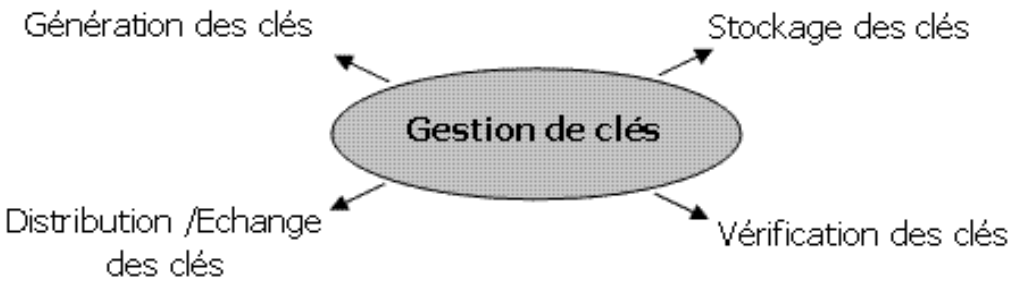

Figure II.4- Fonctions de la gestion de clés.

Sous les contraintes des réseaux WBANs, la conception d'un mécanisme de gestion de clés est un grand défi.

La cryptographie à clé publique est une solution très efficace qui fournit des mécanismes plus sûrs et fiables pour l'authentification et la distribution des clés. Traditionnellement, la cryptographie asymétrique exige un espace mémoire assez grand et de haute capacité de calcul, ce qui la rend inappropriée pour les réseaux de capteurs. Cependant, des recherches récentes ont montré la faisabilité d'appliquer la solution à clé publique aux réseaux de capteurs en choisissant les bons algorithmes et les paramètres appropriés.

Pour cette raison, la plupart des schémas de gestion de clés proposes pour les WBANs sont basés sur la cryptographie symétrique**[19][20].**

Le problème majeur avec la cryptographie symétrique est de pouvoir trouver une méthode qui achève l'établissement de clé entre les nœuds. La solution commune est d'utiliser une méthode de pré-distribution dans laquelle les clés sont chargées dans les nœuds capteurs avant le déploiement.

II.7.1 -Objectifs de la gestion des clés

L'établissement de clés cryptographiques entre les nœuds du réseau permet de :

- Sécuriser le routage
- Sécuriser l'agrégation
- Sécuriser les données échangées
- Assurer l'authentification
- Contraintes de conception

La figure II.5 résume les contraintes découlant des propriétés des réseaux de capteurs à prendre en compte dans la conception d'une solution de gestion de clés pour ce type de réseaux.

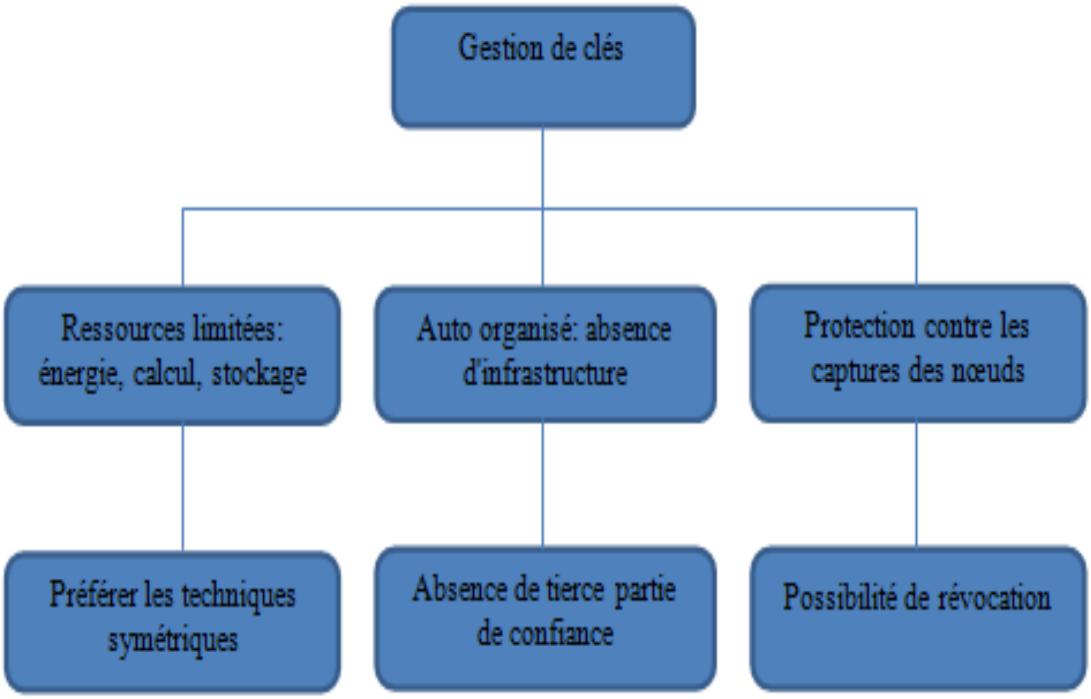

Figure II.5- Contraintes de conception de solutions de gestion de clés.

Comme a été déjà mentionné précédemment, les nœuds capteurs possèdent des ressources limitées en termes de calcul, stockage et énergie. Les réseaux de capteurs ont une structure Ad-hoc d'où l'absence d'une infrastructure. Aussi les nœuds du réseau

sont sujets à compromission. Toutes ces contraintes doivent être prises en considération lors de l'établissement d'un protocole de gestion de clés.

II.7.2 –Solutions introductives

II.7.2.1 Solution 1 : une clé partagée par le réseau

Cette solution consiste à utiliser une clé unique partagée par tous les nœuds du réseau. Les avantages de cette solution sont :

- Gestion simple des clés, car il suffit de pré-charger les nœuds, avant le déploiement, par une seule clé.
- Toutes les communications peuvent être chiffrées simplement en utilisant un minimum de mémoire (stockage d'une seule clé).

Par contre cette méthode ne présente aucune résilience contre la compromission d'un nœud, parce que si un attaquant compromet un nœud du réseau, et étant donné que tous les nœuds du réseau communiquent entre eux en utilisant la même clé, dans ce cas, la sécurité de tout le réseau est menacée.

II.7.2.2 Solution 2 : deux clés partagée par paire de noeuds

Dans cette solution, chaque nœud est pré-chargé avec (N-1) clés secrètes, chacune de ces clés est connue seulement par ce nœud et un des (N-1) autres nœuds (N étant le nombre de nœuds dans le réseau). Cette solution permet une parfaite résilience car la compromission d'un nœud n'affecte pas la sécurité des autres nœuds. Par contre cette solution n'est pas appropriée aux réseaux de capteurs car: Elle exige une capacité mémoire importante pour stocker les (N-1) clés. L'ajout de nouveaux nœuds est difficile parce que les nœuds existants ne possèdent pas les clés de ces nouveaux nœuds.

II.7.2.3 Solution 3 : basée sur la station de base

Cette dernière est considérée comme tierce partie de confiance avec laquelle chaque nœud partage une clé secrète **[21].** Pour que deux nœuds puissent communiquer entre eux d'une manière sécurisée, la station de base transmet une clé symétrique à chacun de ces nœuds en utilisant la clé secrète partagée avec eux. Cette solution présente les avantages suivants :

- Une connectivité totale, où chaque nœud peut partager une clé avec n'importe quel autre nœud du réseau.
- Une résilience parfaite contre la compromission d'un nœud.

II.8 – Notre protocole de gestion de clefs

Dans cette partie, nous allons présenter notre protocole de gestion de clés dans les réseaux WBANs, qui vise à établir des clés de session entre les nœuds du réseau WBAN et la station de base avec un minimum de consommation énergétique.

II.8.1 – Hypothèses

Nous supposons que :

- Chaque nœud capteur est créé avec un identificateur de périphérique unique (UID) qui n'est connu que par ce nœud capteur.
- Les identificateurs de tous les nœuds doivent être programmés manuellement dans la station de base
- Chaque UID agit comme un secret initial partagé entre le nœud capteur et la station de base
- Chaque UID n’est jamais échangé en clair
- Un mécanisme d'autoprotection de dispositif pourrait être employé afin d'assurer que la mémoire est vidée si toute tentative est faite pour manipuler physiquement le dispositif afin de récupérer les données confidentielles.

II.8.2 – Phase d'établissement de clés entre un nœud capteur et la station de base

Cette phase a pour objectif d'établir efficacement et sûrement des clés cryptographiques symétriques entre les nœuds capteurs et la station de base.

La figure suivante illustre notre approche de génération et de distribution de clés entre un nœud capteur et la station de base.

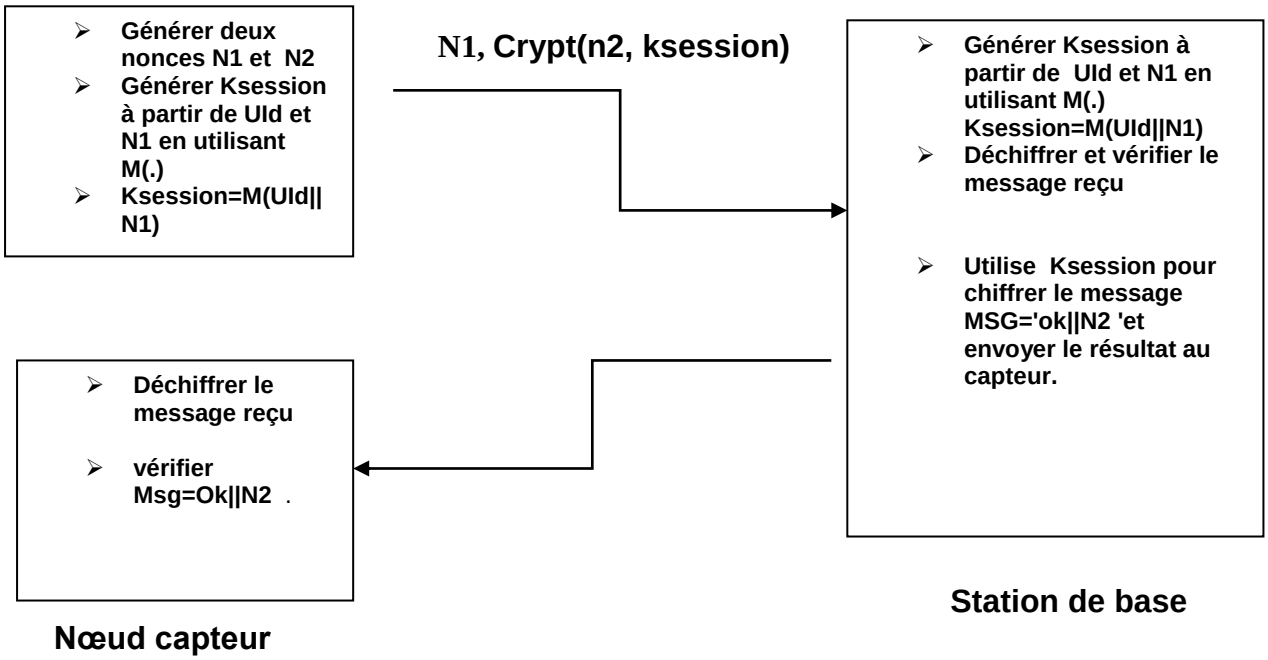

Figure II.6- Phase d'établissement de communication entre les nœuds et la station de base

Chaque nœud qui vise à établir une clé de session symétrique avec la station de base effectue les étapes suivantes :

- Etape 1: générer deux nonces N1 et N2

- Étape 2: générer la clé de session à partir de son UID et le nonce N1 en utilisant la fonction de hachage M(.), Ksession = M (UID | | N1)

- Étape 3: crypter un message et le nonce N2 avec Ksession
- Étape 4: envoyer le message vers la station de base.

Sur réception du message, la station de base génère la clé de session Ksession à partir de l'UID et le nonce N1 en utilisant la fonction de hachage M (.). Ensuite, elle vérifie le message Ok . Si la vérification est réussie, la station de base utilise la clé calculée (Ksession) pour envoyer les informations chiffrées suivantes au nœud initiateur du protocole.

II.8.3 – Analyse du protocole en termes de services de sécurité

Confidentialité: Cet aspect est assuré par l'utilisation du chiffrement symétrique pour chiffrer le trafic échangé entre la station de base et les nœuds de capteurs.

Intégrité et authenticité: L'intégrité et l'authenticité peuvent être assurées en utilisant le « base 64 » calculé et joint à chaque paquet transmis ou échangé entre la station de base et tout nœud du réseau.

Non rejeu : assuré par le nonce N2.

II.9 – Conclusion

La sécurité des réseaux de capteurs corporels contre les éventuelles attaques et menaces est indispensable. Le développement des mécanismes de protection pour les réseaux WBANs est un défi majeur car ces systèmes sont d'une part limités en termes de ressources mémoire, CPU et énergie, d'autre part ces réseaux ont une topologie ad hoc i.e. sont sans infrastructure ce qui rend les solutions de sécurité traditionnelles inapplicables dans ce type de réseau. Dans ce chapitre, nous avons développé un protocole léger de génération et de distribution de clés cryptographiques dans les réseaux de capteurs corporels afin de protéger les informations véhiculées dans ces réseaux. Le chapitre suivant est consacré à l'évaluation et la validation de notre solution en utilisant le système d'exploitation Contiki et le simulateur Cooja

Chapitre III

Réalisation et simulation

III.1 – Introduction

Dans cette partie, nous allons tout d'abord aborder l'environnement de travail, ensuite nous allons détailler l'implémentation de notre protocole de gestion de clés, et enfin nous allons présenter les résultats de simulation.

III.2 – Environnement de travail et outils de développement

L'implémentation de notre protocole nécessite l'utilisation de différents outils et logiciels bien spécifiques aux réseaux de capteurs corporels sans fil, tels que le langage C, Contiki qui est un système d'exploitation dédié aux équipements à ressources limitées, le simulateur Cooja qui est fourni avec Contiki.

III.3 – Les étapes de développement du protocole

III.3.1 – Installation de Contiki 2.7

Dans cette partie, nous allons décrire les étapes d'installation de Contiki 2.7 sous K-Ubuntu.

- En premier lieu, nous avons tout d'abord téléchargé Contiki 2.7 à partir du lien suivant : http://downloads.sourceforge.net/project/contiki/Contiki/Contiki%2027/contiki - 2.7.zip.
- Puis nous avons décompressez le fichier dans **/ home / user / : unzip contiki - 2.7.zip.**
- Ensuite nous avons renommé le dossier de contiki - 2.7 à contiki : **mv contiki - 2.7 contiki**
- Enfin nous avons installé tous les paquets nécessaires :**sudo apt- get install build- essential binutils- msp430 gcc- msp430 msp430- libc msp430mcu mspdebug binutils- avr gcc- a vr gdb- avr avr- libc avrdude openjdk- 7- jdk openjdk- 7- jre ant libncurses5- dev**

III.3.2 – Comment utiliser Cooja avec Contiki

Cooja est un simulateur proposé par Contiki afin d'émuler les différents capteurs sur lesquels seront chargés un système d'exploitation et des applications. Cooja permet ensuite de simuler les connexions réseaux et d'interagir avec les capteurs.

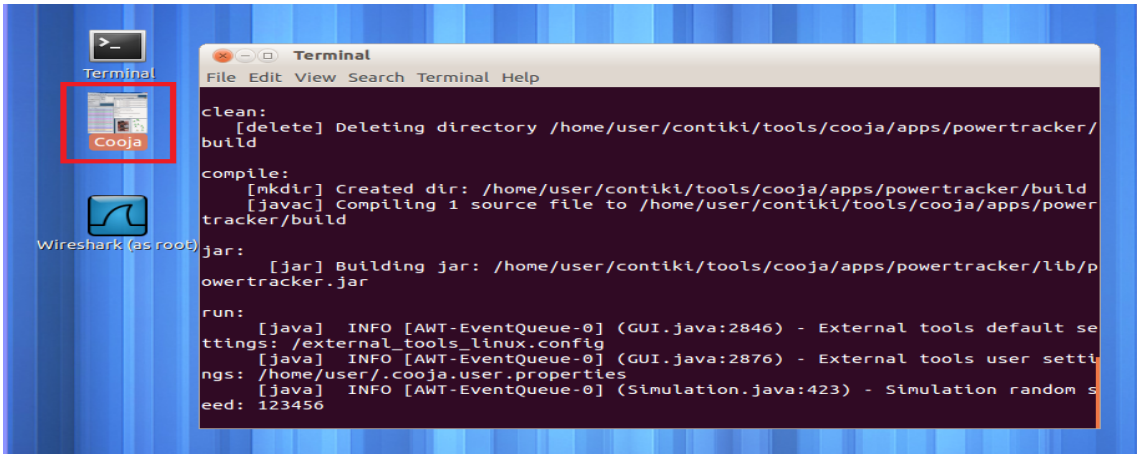

Figure III.1- Démarrage de Cooja

Une fois que Cooja démarre, il faut créer une nouvelle simulation (Figure suivante)

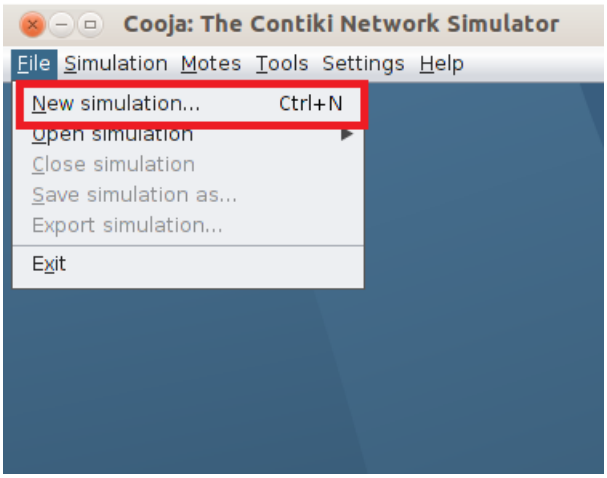

Figure III.2- Création d'une simulation

Après qu'une nouvelle simulation est créée, on peut créer les nœuds capteurs et lancer une simulation. La figure suivante montre un exemple d'interface de la simulation :

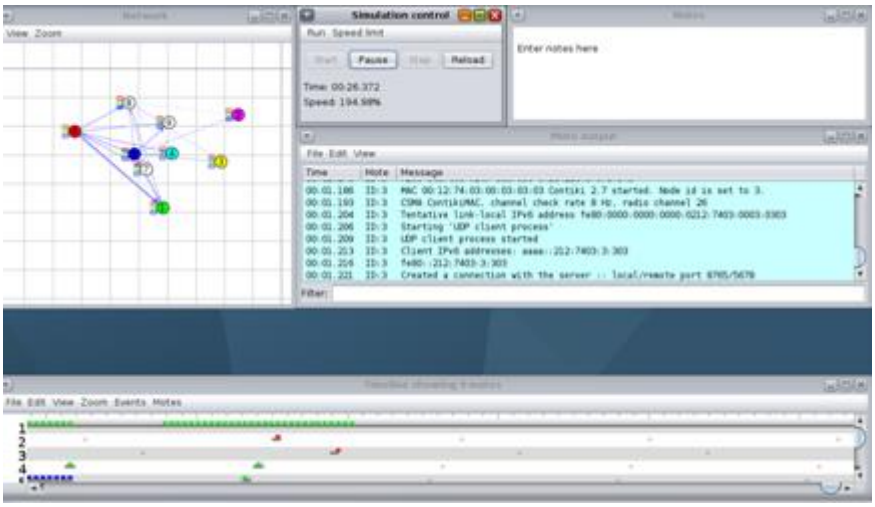

Figure III.3- Exemple de simulation

II.4- I-Implémentation et Evaluation de notre protocole de gestion de clés

Cette partie décrit l'implémentation et l'évaluation de notre protocole de gestion de clés dans les réseaux de capteurs corporels sans fil.

III.4.1 –La partie du code

Le code de notre protocole comprend deux parties, une partie concerne les capteurs clients et l'autre partie concerne la station de base.

III.4.1.1- Partie client

C'est la partie du protocole qui est embarquée sur les nœuds capteurs clients qui veulent établir des communications avec la station de base.

```
      unsigned char *msg= strcat(n1_1,id1_1);
      unsigned char tab[100];
      base64(msg,tab);


static void
send_packet(void *ptr)
{
  static int n2;
  static int seq_id;
  char buf[MAX_PAYLOAD_LEN];
  uint16_t light1;
      uint16_t light2;
      uint16_t temperature;
      uint16_t humidity;
      uint16_t battery;
      int n = 0;
      temperature = sht11_sensor.value(SHT11_SENSOR_TEMP);
      humidity = sht11_sensor.value(SHT11_SENSOR_HUMIDITY);
      battery = battery_sensor.value(0);
..



char *a= tab ;
//printf ("%s:::::\n",a);
char *c;

      //crypter(buf, c,a);

       if (seq_id==0){
        sprintf (buf, "N2:%d", n2);

       printf("N2:%d\n", n2);
 n2++;
c= encrypt(buf, a);
      // printf ("%s:::::\n",c);
     uip_udp_packet_sendto(client_conn, c, strlen(c),
                           &server_ipaddr, UIP_HTONS(UDP_SERVER_PORT));
       }
       else {
         printf("l'humidity :%u N2:%d\n",  humidity,n2);
         sprintf (buf, "l'humidity :%u N2:%d",  humidity,n2);
          c= encrypt(buf, a);
          uip_udp_packet_sendto(client_conn, c, strlen(c),
                           &server_ipaddr, UIP_HTONS(UDP_SERVER_PORT));
          n2++;
       }
       seq_id++;

  /*
```

Figure III.4- Partie client

III.4.1.2- Partie station de base

C'est la partie du protocole qui est embarquée sur la station de base.

```
static void
tcpip_handler(void)
{
  char *appdata;
  char *valeur ;
  if(uip_newdata()) {
    appdata = (char *)uip_appdata;
    appdata[uip_datalen()] = 0;
  }

   PRINTF("station recv  crypter '%s' from ", appdata);
    PRINTF("%d",
           UIP_IP_BUF->srcipaddr.u8[sizeof(UIP_IP_BUF->srcipaddr.u8) - 1]);
    PRINTF("\n");

unsigned const char *msg= strcat(n1_1,id1_1);
unsigned char tab[100];
base64(msg,tab);
unsigned char *msg1= strcat(n1_2,id1_2);
unsigned char tab1[100];
base64(msg1,tab1);
unsigned char *msg2= strcat(n1_4,id1_4);
unsigned char tab2[100];
base64(msg2,tab2);
unsigned char *msg3= strcat(n1_3,id1_3);
unsigned char tab3[100];
base64(msg3,tab3);
unsigned char *msg4= strcat(n1_5,id1_5);
unsigned char tab4[100];
base64(msg4,tab4);
unsigned char *msg5= strcat(n1_6,id1_6);
unsigned char tab5[100];
base64(msg5,tab5);
```

```
liste liste1;
int uid2=2;
int uid3=3;
int uid4=4;
int uid5=5;
int uid6=6;
int uid7=7;
int uid8=8;
int uid9=9;
int uid10=10;
int uid11=11;
  liste1 = ajouterclient(liste1, uid2,tab);
  liste1 = ajouterclient(liste1, uid3,tab1);
  liste1 = ajouterclient(liste1, uid4,tab2);
  liste1 = ajouterclient(liste1, uid5,tab3);
  liste1 = ajouterclient(liste1, uid6,tab4);
  liste1 = ajouterclient(liste1, uid7,tab5);
  liste1 = ajouterclient(liste1, uid8,tab6);
  liste1 = ajouterclient(liste1, uid9,tab7);
  //liste1 = ajouterclient(liste1, uid10,tab8);
  //liste1 = ajouterclient(liste1, uid11,tab9);

    //unsigned char *msg= strcat(n1,id1);

    char *tabk;

    int id=UIP_IP_BUF->srcipaddr.u8[sizeof(UIP_IP_BUF->srcipaddr.u8) - 1];
    tabk= searchkey(liste1,id);
    //printf("le client n-%d  sa cle est %s  \n",id,tabk);
   //int msgsize=strlen(appdata);//On diminue la taille de cryptage de 1
    //const char *s = strcat(n1,id1);
»       //tab=(char) rc_crc32(0, msg, strlen(msg));

»
 char *a=tabk;
```

Figure III.5- Partie station de base.

```
  c= encrypt(appdata, a);
  //*appdata=crypt(appdata,tab1,0,strlen(appdata));//On décrypte du 6ème caractère (après indentifiant), clé fixe d'exemple
 //appdata[msgsize]='\0';//On redéfinit la fin de chaîne
  printf("[Get data...]\n");
 PRINTF("\n");
 //appdata[0]='\0';//On raccourci franchement la chaîne

  PRINTF("station recv decrypter '%s' from ", c);
  PRINTF("%d",
         UIP_IP_BUF->srcipaddr.u8[sizeof(UIP_IP_BUF->srcipaddr.u8) - 1]);
  PRINTF("\n");

  PRINTF("station sending ok pour capteur %d\n",UIP_IP_BUF->srcipaddr.u8[sizeof(UIP_IP_BUF->srcipaddr.u8) - 1]);
  uip_ipaddr_copy(&server_conn->ripaddr, &UIP_IP_BUF->srcipaddr);

static int n2;
char buf[MAX_PAYLOAD_LEN];
sprintf(buf, "  ok pour capteur %d",UIP_IP_BUF->srcipaddr.u8[sizeof(UIP_IP_BUF->srcipaddr.u8) - 1]);
printf("\n");
char k[150];
crypter (buf, k,a);
     uip_udp_packet_send(server_conn, k, sizeof(k));
      uip_udp_packet_sendto(server_conn, k, strlen(k),
                    &client_conn, UIP_HTONS(UDP_CLIENT_PORT));



  uip_create_unspecified(&server_conn->ripaddr);

}
```

III.5 –Evaluation de notre protocole de gestion de clés

Dans cette partie nous allons évaluer les performances de notre protocole en termes d'énergie consommée. Pour ce faire, nous avons implémenté notre protocole sur cinq nœuds (quatre capteurs et la station de base) et ensuite nous avons lancé la simulation sur une période de deux minutes.

Les deux figures suivantes montrent respectivement les différentes phases d'établissement de clés et les messages échangés après la phase d'établissement de clés entre les nœuds capteurs et la station de base.

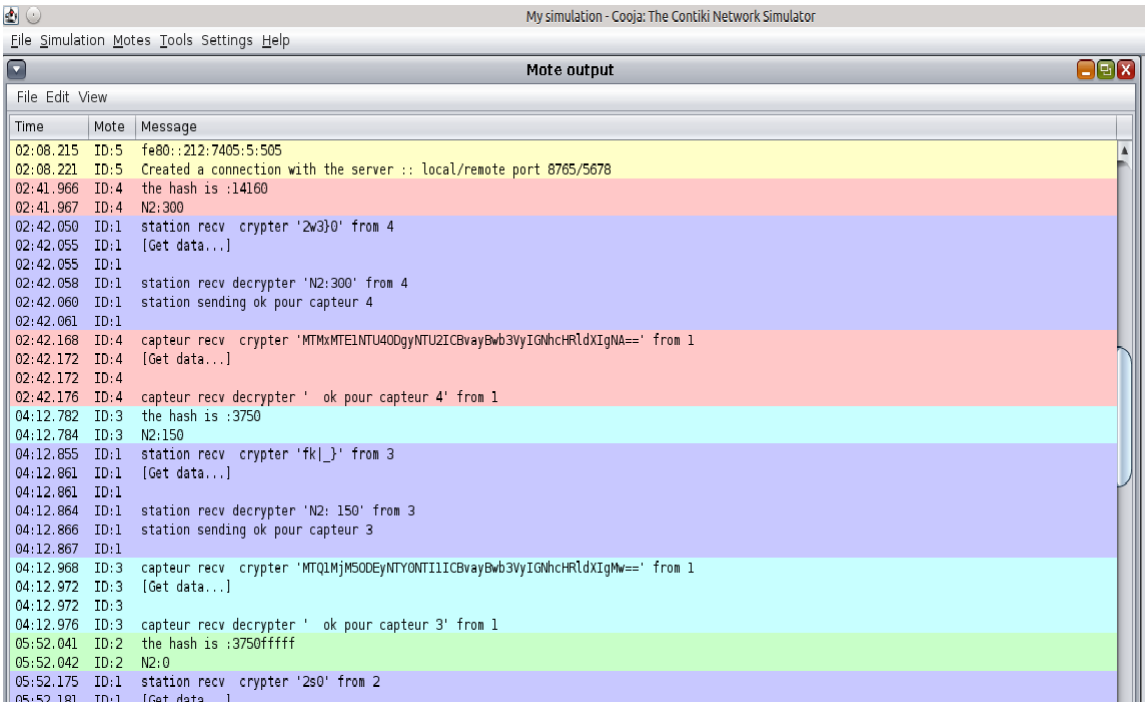

Figure III.6- Phases d'établissement de clés entre les nœuds capteurs et la

```
09:05.207  ID:1
10:02.690  ID:2  the hash is :3750fffff
10:02.692  ID:2  l'humidity :4160 N2:1
10:02.819  ID:1  station recv  crypter '!s!#-1[:GyeJn
10:02.820  ID:1  g|' from 2
10:02.828  ID:1  [Get data...]
10:02.828  ID:1
10:02.832  ID:1  station recv decrypter 'l'humidity :4160 N2:1' from 2
10:02.835  ID:1  station sending ok pour capteur 2
10:02.835  ID:1
10:02.932  ID:2  capteur recv  crypter 'MTIzNDU2Nzg5MTIzNDU2ICBvayBwb3VyIGNhcHRldXIgMg==' from 1
10:02.936  ID:2  [Get data...]
10:02.936  ID:2
10:02.940  ID:2  capteur recv decrypter '  ok pour capteur 2' from 1
10:43.011  ID:3  the hash is :3750
10:43.013  ID:3  temprature :37 N2:151
10:43.113  ID:1  station recv  crypter '91$
10:43.114  ID:1  'S:Pmk
10:43.115  ID:1  m{@qd' from 3
10:43.124  ID:1  [Get data...]
10:43.124  ID:1
10:43.128  ID:1  station recv decrypter 'temprature :37 N2:151' from 3
10:43.130  ID:1  station sending ok pour capteur 3
10:43.131  ID:1
10:43.220  ID:3  capteur recv  crypter 'MTIzNDU2Nzg5MQ==ICBvayBwb3VyIGNhcHRldXIgMw==' from 1
10:43.224  ID:3  [Get data...]
10:43.224  ID:3
10:43.228  ID:3  capteur recv decrypter '  ok pour capteur 3' from 1
12:08.934  ID:5  the hash is :3750ddd
12:08.935  ID:5  N2:605
12:09.061  ID:1  station recv  crypter '2o6}5' from 5
12:09.069  ID:1  [Get data...]
12:09.069  ID:1
12:09.072  ID:1  station recv decrypter 'N2:605' from 5
```

Figure III.7- Echanges sécurisés entre les capteurs et la station de base

III.5.1- Consommation énergétique

Nous avons utilisé PowerTrace pour calculer l'énergie consommée par les capteurs et la station de base et nous avons utilisé également la bibliothèque Python 'matplot' pour tracer et visualiser l'énergie consommée.

1/sudo apt-get install python-matplotlib *//installation de matplot*

" PROCESS_BEGIN ( ) "

2/powertrace_start(CLOCK_SECOND * 2);

#include "powertrace.h" *//inclusion de la fonctionnalité de Power trace*

3/APPS+=powertrace *//à ajouter dans le fichier Makefile*

La figure suivante illustre les données concernant l'énergie consommée par les capteurs toutes les deux secondes

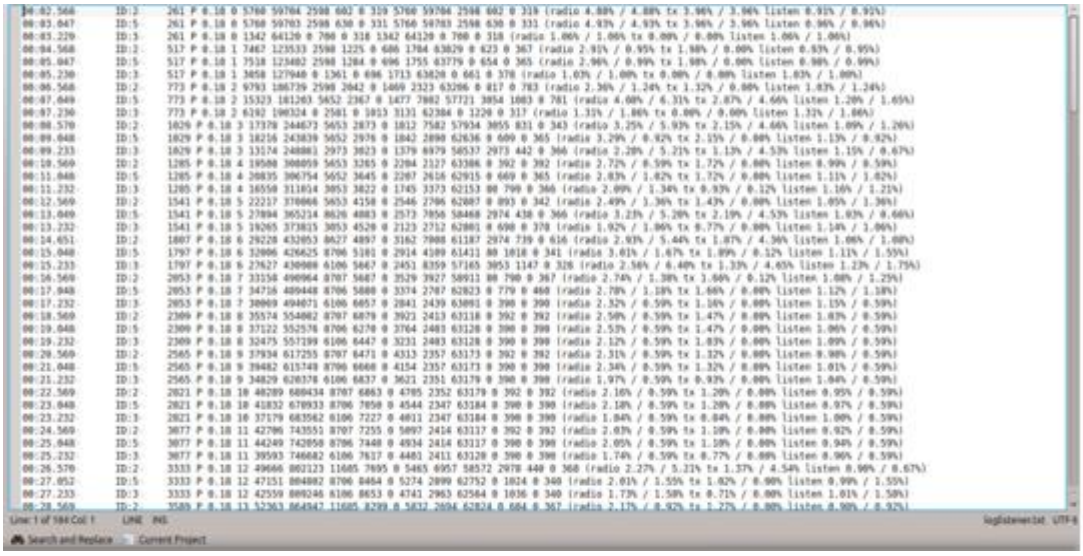

Figure III.8- Données concernant l'énergie consommée toutes les deux secondes

L'interprétation des données générées par Powertrace après simulation de notre protocole est donné par les graphes suivants :

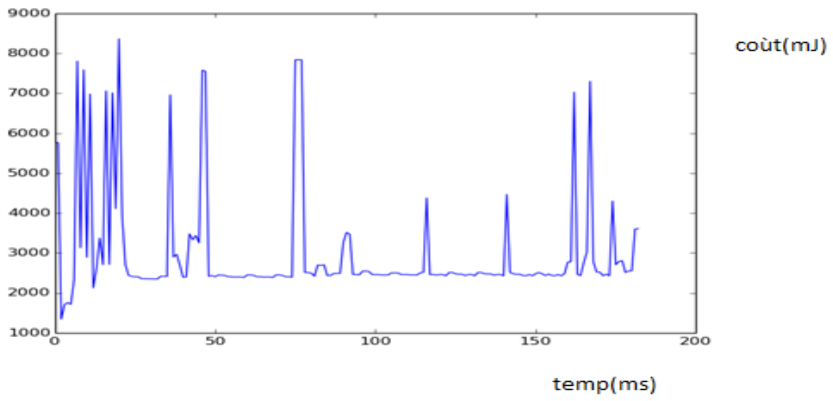

Figure III.9- Energie consommée par CPU

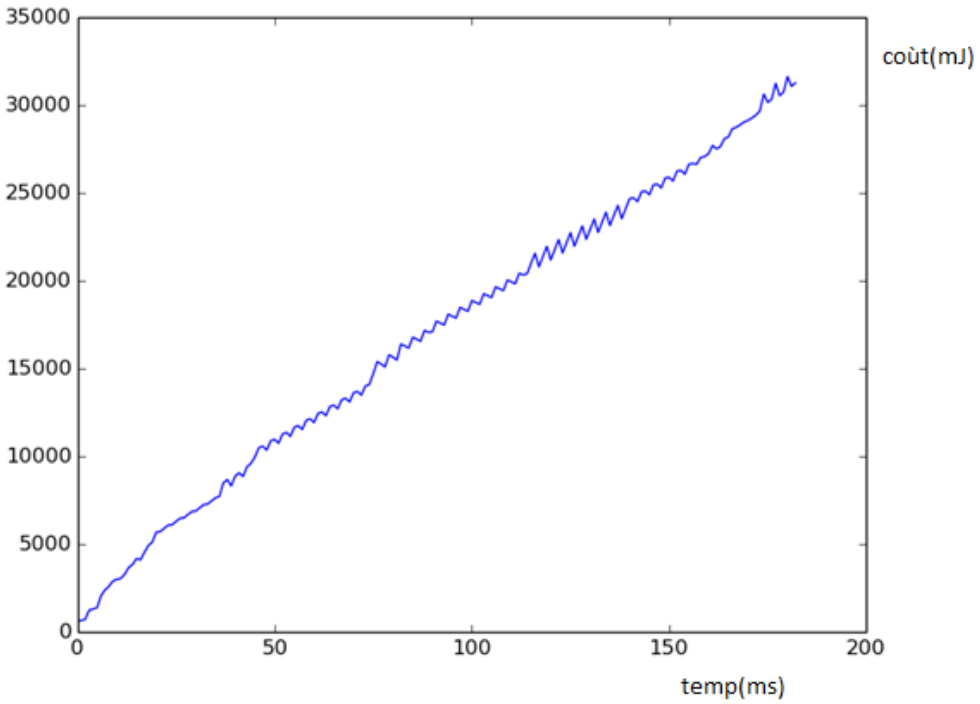

Figure III.10- Energie consommée pour la transmission

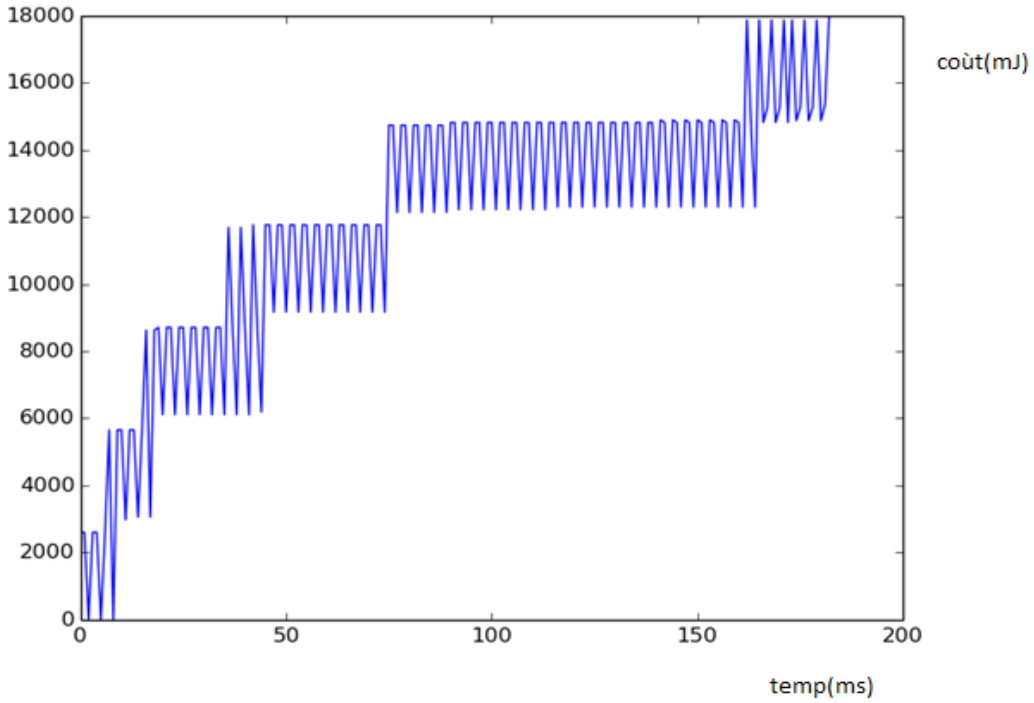

Figure III.11- Energie consommée pour la réception

III.6 – Conclusion

Dans ce chapitre nous avons décrit l'implémentation et la simulation de notre protocole de gestion de clés dans les réseaux de capteurs corporels sans fil sous Contiki-Cooja.

L'avantage de notre protocole consiste d'une part à l'utilisation de la cryptographie symétrique pour la gestion de clés et d'autre part à l'utilisation d'un HandShake à deux phases ce qui permet de minimiser la consommation d'énergie et par conséquent augmenter considérablement la durée de vie du réseau de capteurs corporels.

Conclusion générale

Dans ce mémoire, nous avons montré le besoin et la nécessité de sécuriser les réseaux de capteurs corporels sans fil. Nous avons étudié dans le premier chapitre les contraintes majeures qui doivent être prises en compte lors de la conception de toute architecture de sécurité pour les réseaux de capteurs corporels. Ces facteurs se résument en la faible puissance de calcul des nœuds capteurs, ainsi que la limite de leurs mémoires de stockage et particulièrement la durée de vie de leurs batteries embarquées.

Dans le deuxième chapitre, nous avons abordé la gestion de clés dans les réseaux de capteurs corporels. En effet, la gestion de clés constitue un service très important pour la sécurité de n'importe quel système de communication. Les mécanismes traditionnels de gestion de clés sont inappropriés pour les réseaux de capteurs corporels car ces derniers sont limités en termes de capacité de calcul, de stockage et d'énergie. Pour cela, nous avons développé un protocole de gestion de clés cryptographiques pour les réseaux WBANs en prenant en compte les contraintes des capteurs et notamment la contrainte d'énergie pour garantir une plus longue durée de vie du réseau.

Dans le troisième et dernier chapitre, nous avons mis l'accent sur l'implémentation et la simulation de notre protocole de gestion de clés dans les réseaux de capteurs corporels. L'implémentation et la simulation ont été réalisées sous Contiki Cooja.

Références bibliographiques

[1] David Martins, ”Sécurité dans les réseaux de capteurs sans fil Stéganographie et réseaux de confiance“, L'U.F.R. des Sciences et Techniques de l’université de Franche-Comté, 2010.

[2] Ian F. Akyildiz, Weilian Su, Yogesh Sankara subramaniam, and Eredal Cayirci."Wireless sensor networks: a survey". Computer Networks, The International Journal of Computer and Telecommunications Networking, v.38 n.4, pp.393-422,2002.

[3] David Culler, Deborah Estrin, and Mani Srivastava.Guest editors' introduction"Overview of sensor networks". Computer, 37(8) :41_49, August 2004

[4] Rahim KACIMI, these " Techniques de conservation d'énergie pour les réseaux de capteurs sans fil", Institut National Polytechnique de Toulouse, Septembre 2009

[5] Tinyos. http:/www.tinyos.net/, 2010

[6] Dunkels, A., B. Grönvall, et T. Voigt. Contiki: a Lightweight and Flexible Operating System for Tiny Networked Sensors. In Proceedings of the FirstIEEE Workshop on Embedded Networked Sensors, pages 455-462, Tampa, Florida, USA, 2004.

[7] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruenwald, A. Torgerson, and R. Han. Mantis OS : an embedded multithreaded operating system for wireless micro sensor platforms. Mob. Netw. Appl., 10(4) : 563-579, 2005.

[8] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, « A Survey on Sensor Networks », IEEE Communications Magazine, August 2002

[9] Jennifer Yick, Biswanath Mukherjee, DipakGhosal, « Wireless sensor network survey », Computer Networks 52 (2008) 2292–2330

[10] Manuel CAMUS, Thèse « Architecture de réception RF très faible coût et très faible puissance. Application aux réseaux de capteurs et au standard ZigBee » Université Paul Sabatier - Toulouse III, Février 2008

[11] Cristian Duran-Faundez, these " Transmission d'images sur les réseaux de capteurs sans fil sous la contrainte de l'énergie", Université Henri Poincaré, Nancy 1, juin 2009

[12] Puccinelli, D.; Haenggi, M., "Wireless sensor networks: applications and challenges of ubiquitous sensing", Circuits and Systems Magazine, IEEE

[13] Steve Warren1, Jeffrey Lebak1, Jianchu Yao3, Jonathan Creekmore2, Aleksandar Milenkovic2, and Emil Jovanov, « Interoperability and Security in Wireless Body Area Network Infrastructures », Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference Shanghai, China.

[14] R. Rivest., "The MD5 Message-Digest Algorithm", RFC 1321, 1992, April.

[15] Gérard Chalhoub,'MaCARI:Une methode d'acces déterministe et économe pour les réseaux capteurs sans fils',Thése de doctorat,Université Blaise Pascal

[16] Steffen Peter, Dirk Westhoff, « A Survey on the Encryption of Convergecast Traffic with In-Network Processing », IEEE Transactions on dependable and secure computing, VOL. 5, NO.

[17] Buchmann, Johannes "Introduction to Cryptography », Hardcover, ISBN 978-0-387-21156-5

[18] http://www.securiteinfo.com/cryptographie/hash.shtml

[19] G. Gaubatz, et al, " Public Key Cryptography in Sensor Networks-Revisited", ESAS'04 ,1st European Wksp, Security in Ad-Hoc and Sensor Networks

[20] Krzysztof Piotrowski et al, "How Public Key Cryptography In uences Wireless Sensor Node Lifetime", SASN'06, Alexandria, Virginia, USA, October 30, 2006.

[21] Adrian Perrig et al, "SPINS : Security Protocols for Sensor Networks", Mobile Computing and Networking, Rome Italy

Annexe: Explication du code

PROCESS_BEGIN ()

Cette macro définit le début d'un processus, il doit toujours apparaître dans une définition PROCESSUS fil ().

PROCESS_END ()

Cette macro définit la fin d'un processus. Il doit apparaître dans un PROCESS_THREAD (). Le processus se ferme lorsque la macro PROCESS_END est atteinte.

PROCESS_WAIT_EVENT ()

Attend un évènement à être affecté à ce processus.

PROCESS_THREAD (name, ev, data)

Cette macro bloque le processus en cours d'exécution jusqu'à ce que le processus reçoive un évènement.

PROCESS_YIELD_UNTIL (c)

Cédez le processus en cours jusqu'à ce qu'une condition se produise. Cette macro est différente de PROCESS_WAIT_UNTIL () (C : la condition d'attente).

Void* uip_appdata

Pointeur vers les données d'applications dans la mémoire tampon de paquets. Ce pointeur pointe vers les données d'applications quand l'application est appelée. Si l'application souhaite envoyer des données, elle peut utiliser cet espace pour écrire les données avant d'appeler uip_send ().

Struct uip_udp_conn

Pointeur vers la connexion TCP courante. Le pointeur uip_conn peut-être utilisé pour accéder à la connexion TCP courante.

Uip_datalen ()

La longueur de toutes les données d'entrée qui sont actuellement disponibles dans le tampon uip_appdata. La fonction test uip_data () doit d'abord être utilisée pour vérifier s'il y a des données disponibles.

#define udp_bind (conn, port)

Cette fonction est liée à une connexion et à un port UDP local spécifié. Lorsque la connexion est créée avec udp_new (), elle obtiendra un numéro de port local automatiquement. Si l'application a besoin de lier la connexion à un port local spécifié, cette fonction doit être utilisée.
Le numéro de port doit être prévu dans l'ordre des octets du réseau pour une conversion avec UIP_HTONS () est généralement nécessaire.
Conn : un pointeur vers la connexion UDP qui d'être liée.

#define uip_udp_send (len)

Envoyer un datagramme UDP de longueur len sur la connexion en cours. Cette fonction ne peut être appelée en réponse à un évènement UDP. Les données doivent être présentes dans le tampon uip_buf, à l'endroit indiqué par le pointeur uip_appdata.

Struc uip_ds6_addr

Structure d'adresse unicast.

#define MAC_MAX_PAYLOAD 120

Pointeur dans la mémoire tampon de Rime

Liste de figures

Liste des tableaux :

Liste des abréviations:

***WSN**:* Wireless Sensor Network

***RCSF** :* Réseau de Capteur Sans Fil

***UID** :* Unique Identification device

XOR : OU exclusif opérateur logique de l'algèbre de Boole

Résumé :

Les réseaux de capteurs corporels sans fil (WBANs) ont attiré un grand intérêt dans la dernière décennie, et ont apporté des solutions dans le domaine médical. Néanmoins, la sécurité de ces réseaux reste un défi majeur car les solutions traditionnelles sont irréalisables en raison des limites des capteurs en termes de ressources énergétique, mémoire et CPU.

Notre travail consiste à développer un protocole de gestion de clés en tenant en compte les ressources limitées des capteurs.

Mots-clefs : WBANs, la sécurité, ressources énergétique, mémoire , CPU.

Abstract:

Wireless Body Area Networks (WBANs) attracted great interest in the last decade and have provided solutions in the medical field . However, the security of these networks remains a major challenge because traditional solutions are not feasible because of the limitations of sensors in terms of energy resources, memory and CPU.

Our job is to develop a key management protocol, taking into account the limited resources of the sensors.

Keywords: WB ANs, security, energy resources, memory,CPU.

ملخص:

جذبت شبكات الاستشعار اللاسلكية (WBANs) اهتماما كبيرا في العقد الماضي وقدمت الحلول في المجال الطبي . ومع ذلك، لا يزال أمن هذه الشبكات تحديا كبيرا لأن الحلول التقليدية ليست مجدية بسبب القيود المفروضة على أجهزة استشعار من حيث موارد الطاقة والذاكرة و حدة المعالجة المركزية.

مهمتنا هي لوضع بروتوكول معالجة توليد وتوزيع مفاتيح، مع الأخذ بعين الاعتبار الموارد المحدودة لأجهزة الاستشعار.

الكلمات المفتاحية

أمن هذه الشبكات. موارد الطاقة .الذاكرة. وحدة المعالجة المركزية.