RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mira Abderrahmane de Béjaïa
Faculté des Sciences Exactes
Département Informatique

Mémoire de fin de cycle

En vue de l'obtention du diplôme de master en Informatique
Option : Réseaux et systèmes distribués (RESYD)

Intégration de la stéganographie dans un protocole de routage des réseaux de capteurs sans fil

---

Présenté par :

STAMBOULI Abdelouadoud & SLIMI Abdelkrim

Devant le Jury composé de :

Président du jury :	M^{r} KHANOUCHE M^{ed} Essaid	Université de Béjaïa.
Examinatrice :	M^{me} BATTAT Nadia	Université de Béjaïa.
Examinateur :	M^{r} ABBACHE Bournane	Université de Béjaïa.
Promoteur :	M^{r} AISSANI Sofiane	Université de Béjaïa.

Juin 2014

Remerciements

Au nom d'Allah, le Tout Miséricordieux, le Très Miséricordieux.

En premier lieu, nous remercions Dieu, glorifié soit-Il, qui nous a donné la force nécessaire à l'accomplissement de cette tâche.

Nous remercions vivement M^{r} AISSANI Sofiane pour nous avoir honoré par son encadrement, sa disponibilité, sa patience, ses précieux conseils, ses remarques constructives et ses encouragements qui nous ont permis de mener à bien ce travail.

Nous remercions aussi M^{r} KHANOUCHE M^{ed} Essaid pour avoir accepté de présider le jury et pour son aide précieuse en terme de documentation, M^{me} BATTAT Nadia et M^{r} ABBACHE Bournane pour avoir accepté de faire parti du jury. Nous tenons à leurs exprimer toute notre gratitude pour avoir accepté de juger ce travail.

Nous remercions également tous nos enseignants qui nous ont suivis et formés pendant tout le cursus.

Un énorme merci à nos familles et amis et nos camarades de la promotion pour leurs encouragements, leurs éternel soutient et la confiance qu'ils ont en nos capacité.

LISTE DES ABRÉVIATIONS

AODV	**A**d hoc **O**n-demand **D**istance **V**ector.
CBRP	**C**luster **B**ased **R**outing **P**rotocol.
CE	**CE**ll-head.
CH	**C**uster-**H**ead.
CSMA	**C**arrier **S**ense **M**ultiple **A**ccess.
DSDV	**D**ynamic destination **S**equenced **D**istance **V**ector.
EROM	**E**rasable **R**ead **O**nly **M**emory.
FSR	**F**isheye **S**tate **R**Routing.
GIF	**G**raphics **I**nterchange **F**ormat.
GPRS	**G**eneral **P**acket **R**adio **S**ervice.
GPS	**G**lobal **P**ositioning **S**ystem.
GSR	**G**lobal **S**tate **R**outing.
IBPRF	**I**dentiy**B**ased **P**re-distribution using **R**andom **F**onction .
IEEE	**I**nstitute of **E**lectrical and **E**lectronics **E**ngineers.
LEACH	**L**ow-**E**nergy **A**daptive **C**lustering **H**ierarchy.
LEAP	**L**ightweight **E**xtensible **A**uthentication **P**rotocol.
LSB	**L**east **S**ignificant **B**it.
MAC	**M**edia **A**ccess **C**ontrol.
MANET	**M**obile **A**d hoc **NET**work.
MEMS	**M**icro-**E**lectro-**M**echanical **S**ystems.
RAM	**R**andom **A**ccess **M**emory.

RCSFs	**R**éseaux de **C**apteurs **S**ans **F**il.
RF	**R**adio **F**requency.
RSA	**R**ivets **S**hamir **A**dleman.
PEGASIS	**P**ower **E**fficient **G**athering in **S**ensor **I**nformation **S**ystem .
PIKE	**P**eer **I**ntermediaries **K**ey **E**tablisement.
PNG	**P**ortable **N**etwork **G**raphics.
SB	**S**tation de **B**ase.
SNKM	**S**ecurity **N**ode-based **K**ey **M**anagement.
SPIN	**S**ensor **P**rotocol for **I**nformation via **N**egotiation.
SPNS	**S**ecurity **P**rotocol for **N**etwork**S**ensor.
STKM	**S**panning **T**ree **K**ey **M**anagement.
TDMA	**T**ime **D**evision **M**ultiple **A**ccess.
TEEN	**T**hreshold-sensitive **E**nergy **E**fficient sensor **N**etwork protocol.
UMTS	**U**niversal **M**obile **T**elecommunication **S**ystem.
WSN	**W**ireless **S**ensor **N**etwork.
ZHLS	**Z**one-based **H**ierarchical **L**ink **S**tate protocol.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

TABLE DES FIGURES

INTRODUCTION GÉNÉRALE

L'évolution dans le domaine de la communication, principalement dans la communication sans fil et l'informatique mobile, gagne de plus en plus de popularité. Les nombreuses avancées techniques et technologiques dans les domaines de la micro-électronique, de la micro-mécanique et des technologies de communication sans fil ont permis ainsi de créer de petits objets communicants équipés de capteurs à un coût raisonnable. Ces nouveaux objets, appelés nœuds ou capteurs, sont équipés d'une unité de mesure, d'une unité de calcul, de mémoire et d'une radio pour communiquer. Enfin, pour l'alimentation, ces nœuds possèdent une pile ou un système de récupération d'énergie dans l'environnement. Le déploiement de ces noeuds capteurs forme un nouveau type de réseau appelé réseau de capteur sans fil (RCSF). Les réseaux de capteurs sont utilisés dans divers domaines d'applications : applications militaires, domotique, surveillance industrielle, de phénomènes naturels ou relevés de compteurs. Le faible coût de construction et la facilité de déploiement de tels réseaux ont contribué à leur popularité croissante.

L'un des défis à relever dans ce type de réseau, c'est le défi de la sécurité ou comment sécuriser ce type de réseau, en tenant compte des contraintes liées à ce dernier comme : faible puissance de calcul, énergie limitée, etc.

Si comme dans les autres types de réseaux, les solutions cryptographiques permettent de préserver leur intégrité, nous avons pu voir que leur coût en termes de temps d'exécution et d'énergie dépensée est en inadéquation avec les réseaux de type réseaux de capteurs sans fil. De plus, l'utilisation de la cryptographie dans des réseaux de capteurs sans fil pose un autre souci, en effet, le besoin de sécuriser des données malgré la difficulté de mettre en place une solution cryptographique efficace amène à proposer des solutions avec des algorithmes de cryptographie plus faible ou des failles de protocoles. Ces faiblesses de chiffrage peuvent amener à ce que des attaquants, qui ont interceptés les messages envoyés sur les ondes radios et qui possèdent des

dispositifs assez puissants, soient capables de les déchiffrer rapidement .

Alors le défi principal est de sécuriser les réseaux et atteindre les objectifs souhaités de la sécurité avec une faible consommation d'énergie. Pour répondre à cela, nous appliquons des mécanismes basés sur la stéganographie dans les protocoles de routage dans les réseaux de capteur sans fil, c'est à dire dissimuler des messages dans les messages de ces protocoles.

Ce document est organisé en quatre chapitres :

Dans le premier chapitre, nous présentons les réseaux de capteurs sans fil, leurs architectures de communication, les principaux facteurs et contraintes qui influencent la conception des réseaux de capteurs sans fil. Et on termine avec le concept de routage,en citant quelques exemples de ces protocoles de routage.

Dans le deuxième chapitre, nous abordons le concept de la gestion de clés dans les RCSFs où nous exposons les classifications existantes pour les protocoles de gestion de clés, en citant quelques exemples de ces protocoles et en apportant une étude critique à chaque classe de protocoles.

Dans le troisième chapitre, nous présentons la technique de dissimulation, c'est à dire la stéganographie, en expliquant les différentes possibilités de dissimulation dans les trames des couches PHY et MAC du protocole IEEE.802.15.4.

Le dernier chapitre est consacré à la présentation de notre protocole pour les RCSFs, que nous avons dénommé KMPS (Key Management Protocol with Steganography). Il se base sur l'organisation du réseau en cellules et clusters, il utilise la notion de la stéganographie pour le partage des clés et pour la mise en place des clés communes dans le réseaux pour assurer une meilleure consommation d'énergie, puis nous présenterons les résultats d'évaluation de notre protocole effectué à travers un simulateur développé en JAVA .

Enfin, nous concluons notre travail par une conclusion générale qui résume nos contributions et nos perspectives de recherches pour nos travaux futures.

CHAPITRE 1

GÉNÉRALITÉS SUR LES RÉSEAUX DE CAPTEURS SANS FIL (RCSF)

1.1 Introduction

Au cours des dernières décennies, nous avons assisté à une miniaturisation du matériel informatique, qui a apporté une nouvelle génération de réseaux informatiques et télécoms présentant des défis importants. Les réseaux de capteurs sans fil sont l'une des technologies visant à résoudre les problèmes de cette nouvelle ère de l'informatique embarquée et omniprésente.

La mise en œuvre de simples possibilités de traitement, de stockage, de détection et de communication à faible coût ouvre la porte à une multitude de nouvelles applications. Les réseaux de capteurs constituent une catégorie de réseaux sans fil comportant un très grand nombre de nœuds. Ils sont caractérisés par un déploiement très dense et à grande échelle dans des environnements souvent limités en terme de ressources. Les spécificités les plus frappantes de ces nœuds sont leurs capacités d'auto-organisation, de coopération, leur rapidité de déploiement, leur tolérance aux erreurs et leur faible coût.

En termes de domaines d'applications, les réseaux de capteurs ont connu un très grand succès, car ils détiennent un potentiel qui révolutionne de nombreux secteurs de notre économie et notre vie quotidienne, de la surveillance et la préservation de l'environnement, à la fabrication industrielle, en passant par l'automatisation dans les secteurs de transport et de la santé, la modernisation de la médecine, de l'agriculture, de la télématique et de la logistique [1].

Dans ce chapitre, nous présentons un ensemble de généralités sur les réseaux de capteurs,

leurs architectures, leurs caractéristiques ainsi que leurs domaines d'applications. Nous présentons aussi les principaux facteurs qui influencent la conception des réseaux de capteurs ainsi que les problématiques liées à ce dernier, nous abordons après la notion du routage dans les RCSFs en présentant les différentes méthodes utilisées et en définissant quelques protocoles qui seront utilisées plus tard dans ce mémoire.

1.2 Les réseaux sans fil

1.2.1 Définition

Un réseau sans fils est un réseau informatique qui connecte différents postes ou systèmes entre eux par ondes radio. Ces dernières sont plus exposéess aux perturbations et aux interférences que ne le sont les communications filaires. La norme IEEE 802.11, connu sous le nom de Wi-Fi est la norme la plus utilisée actuellement pour les réseaux sans fils. Le rayonnement géographique des ondes est relativement limité, étant donné la faible puissance d'émission des solutions matérielles actuelles. Pour cette raison, les réseaux sans fils se sont avant tout développés comme réseaux internes propre à un bâtiment, soit comme réseau d'entreprise, soit comme réseau domestique.

1.2.2 Les réseaux Ad Hoc

Un réseau sans fils ad-hoc (ou MANET) est formé par un ensemble d'hôtes qui s'organisent seuls et de manière totalement décentralisée, formant ainsi un réseau autonome et dynamique ne reposant sur aucune infrastructure filaire. Aucune infrastructure n'étant disponible, ces objets ont donc à découvrir dynamiquement leur environnement. Cependant, l'IETF qui représente l'organisme responsable de l'élaboration de standards pour Internet, définit les réseaux ad-hoc de la manière suivante : "Un réseau ad-hoc est un système autonome de plates-formes mobiles (par exemple un routeur interconnectant différents hôtes et équipements sans fils) appelées nœuds qui sont libres de se déplacer aléatoirement et sans contrainte. Ceci provoque des changements rapides et imprédictibles de la topologie du réseau. Ce système peut fonctionner d'une manière isolée ou s'interfacer à des réseaux fixes à travers des passerelles. Dans ce dernier cas, un réseau ad-hoc est un réseau d'extrémité.

Les réseaux ad-hoc sont idéaux pour les applications caractérisées par une absence d'une infrastructure préexistante, tels que les applications militaires, ou les autres applications de tactique comme les opérations de secours (incendies, tremblements de terre,...).

1.2.3 Les réseaux de capteurs

Les réseaux de capteurs sans fils sont considérés comme un type spécifique des réseaux ad-hoc où l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle d'hôtes et de routeurs.

1.2.4 Comparaison entre les réseaux de capteurs et les réseaux ad-hoc

Bien que ces deux types de réseaux ont plusieurs points en commun, cependant on peut mentionnés quelques différences illustrées ainsi dans le tableau suivant :

Réseaux de capteurs	**Réseaux Ad-Hoc**
Objectif cible.	Générique / communication.
Les nœuds collaborent pour remplir un objectif.	Chaque nœud a son propre objectif.
Flot de données tous vers un (Many-to-one) la communication repose sur la diffusion.	Flot de données tous vers tous (Any-to-any). La communication est de type point à point.
Très grand nombre de nœuds ayant tous un identificateur ID	Notion d'ID.
Energie est un facteur déterminant,nœud capteur sujet aux pannes	Débit est majeur.
Nombre de nœuds importants (forte scalabilité).	Nombre moyen de nœud.
Les entités interagissent essentiellement avec la nature ou l'environnement entre elles.	Les entités MANET sont utilisées directement par les êtres humains,comme les portables, les PDA,

TABLE 1.1 – Comparaison entre les réseaux de capteurs et les réseaux ad-hoc [2]

1.3 Architecture d'un capteur

Un capteur est composé de plusieurs éléments ou modules correspondant chacun à une tâche particulière d'acquisition, de traitement, ou de transmission de données. Il comprend également une source d'énergie [3].

1.3.1 Unité d'acquisition des données

Le principe de fonctionnement des détecteurs est souvent le même : il s'agit de répondre à une variation des conditions d'environnement par une variation de certaines caractéristiques électriques (par exemple pour une thermistance, une variation de température entraîne une variation de la résistance). Les variations de tension sont ensuite converties par un convertisseur

analogique-numérique afin de pouvoir être traitées par l'unité de traitement.

On trouve aussi des structures plus complexes pour détecter d'autres phénomènes : les MEMs (Microelectromechanical systems). Ils sont utilisés pour une grande variété de phénomènes physiques (accélération, concentration chimique,...).

1.3.2 Unité de traitement des données

Les microcontrôleurs utilisés dans le cadre de réseaux de capteurs sont à faible consommation d'énergie. Leurs fréquences sont assez faibles, moins de 10MHz pour une consommation de l'ordre de 1mW. Une autre caractéristique est la taille de leur mémoire qui est de l'ordre de 10Ko de RAM pour les données et de 10Ko de ROM pour les programmes [4] Cette mémoire consomme la majeure partie de l'énergie allouée au microcontrôleur, c'est pourquoi on lui adjoint souvent de la mémoire flash moins coûteuse en énergie.

Outre le traitement des données, le microcontrôleur commande également toutes les autres unités notamment le système de transmission.

1.3.3 Unité de transmission de données

Les composants utilisés pour réaliser la transmission sont des composants classiques. Ainsi on retrouve les mêmes problèmes que dans tous les réseaux sans-fil : la quantité d'énergie nécessaire à la transmission augmente avec la distance. Pour les réseaux sans-fil classiques (LAN, GSM) la consommation d'énergie est de l'ordre de plusieurs centaines de milliwatts, et on se repose sur une infrastructure alors que pour les réseaux de capteurs,le système de transmission consomme environ 20mW et possède une portée de quelques dizaines de mètres. Pour augmenter ces distances tout en préservant l'énergie, le réseau utilise un routage multi-sauts.

1.3.4 Source d'énergie

Pour des réseaux de capteurs sans fils autonomes, l'alimentation est une composante cruciale. Il y a essentiellement deux aspects : premièrement, stocker l'énergie et la fournir sous la forme requise ; deuxièmement, tenter de reconstituer l'énergie consommée par un réapprovisionnement grâce à une source externe au nœud-capteur telles les cellules solaires. Le stockage de l'énergie se fait traditionnellement en utilisant ses piles. À titre indicatif, ce sera souvent une pile normale d'environ 2.2 - 2.5 Ah fonctionnant à 1.5V [5].

1.4 Caractéristiques des réseaux de capteurs

1.4.1 Architecture d'un réseau de capteurs

Les réseaux de capteurs sans fil se composent généralement d'un grand nombre de capteurs, communiquants via des liens radio pour le partage d'informations et le traitement coopératif. Les données collectées par ces capteurs sont acheminées directement ou via un routage à un point de collecte, appelé station de base. La communication entre les stations puits et l'utilisateur se fait via Internet ou satellite [4].

La figure 1.1 illustre l'architecture d'une réseau de capteurs :

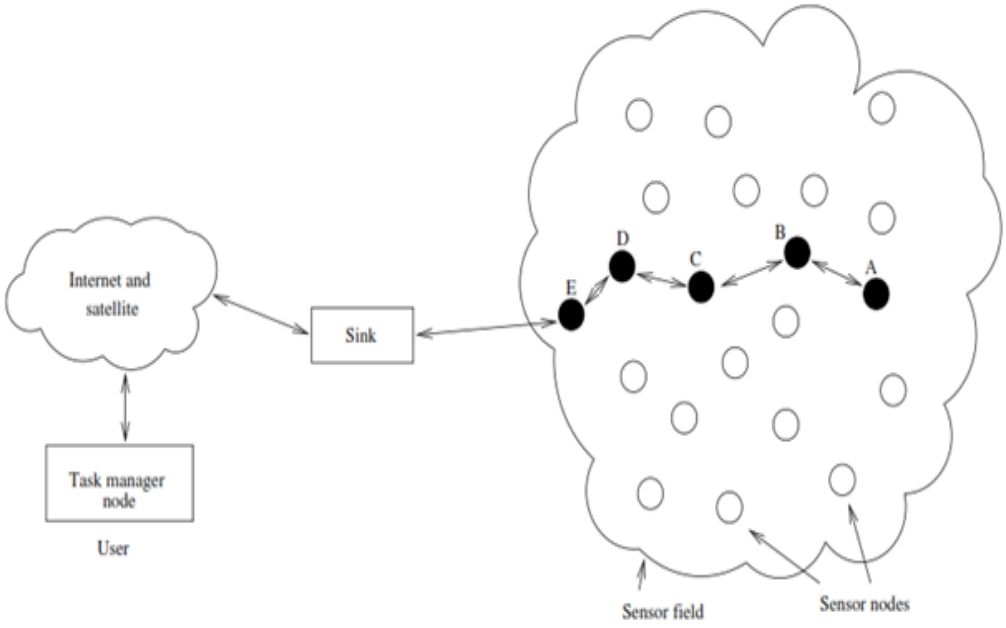

FIGURE 1.1 – Architecture d'un réseau de capteurs

Dans ces réseaux, chaque nœud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs point de collecte. Les RCSFs sont une instance particulière de la classe des réseaux ad-hoc, ils héritent leurs caractéristiques et partagent beaucoup de concepts existants avec ceux-ci.

La position des nœuds n'est pas obligatoirement prédéterminée. Ils sont dispersés à travers une zone géographique appelée champ de captage, les données captées sont acheminées grâce à un routage multi-sauts.

1.4.2 Caractéristiques des réseaux de capteurs sans fils

Les techniques des réseaux ad-hoc sont utilisées pour la réalisation d'un RCSF. Cependant, les protocoles et les algorithmes proposés dans les réseaux ad-hoc ne conviennent pas aux RCSF. Un réseau de capteurs a beaucoup de caractéristiques importantes, parmi celle-ci nous citons :

- **La durée de vie limitée :** L'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où le nombre de nœuds restants est trop bas pour permettre au réseau de bien fonctionner. Les capteurs utilisent leurs énergies à des fins de calcul et de transmission de données. Dans un RCSF chaque nœud joue le rôle d'émetteur et de routeur, ainsi, une défaillance énergétique d'un nœud capteur peut affecter des changements significatifs à la topologie du réseau et imposer une réorganisation couteuse de ce dernier.

- **Ressources limitées :** Habituellement, les nœuds capteurs ont une taille très petite, ce facteur de forme limite la quantité de ressources qui peuvent être mises dans ces nœuds, par conséquence la capacité de traitement et de mémoire est très limitée.

- **Bande passante limitée (Média de transmission) :** Différents médias sans fils (radio, infrarouge, optique), sont utilisés par les nœuds capteurs pour se communiquer. Le médium utilisé doit être compatible avec l'environnement de l'application ; cependant, la majorité des capteurs communiquent par l'utilisation d'un circuit RF. En raison de la puissance limitée, les nœuds capteurs ne peuvent pas supporter des débits très élevés.

- **Réseau auto-organisé :** Une configuration manuelle d'un réseau est en pratique impossible à réaliser ci cause du grand nombre de nœuds et leurs déploiements dans des environnements hostiles. Par ailleurs, des nœuds peuvent quitter le réseau en tombant en panne (manque d'énergie, panne physique, etc.), et d'autres peuvent l'intégrer. Par conséquent, il est essentiel que le réseau s'auto-organise.

- **Topologie dynamique :** La topologie des réseaux de capteurs change d'une manière fréquente et rapide du fait que les nœuds peuvent être déployés dans des milieux difficiles (par exemple un champ de bataille), ainsi que la défaillance très probable des nœuds capteurs. Cependant, les nœuds capteurs et les nœuds finaux (les nœuds de destination) où ils doivent envoyer l'information capturée peuvent être mobiles, et donc la transmission de messages en provenance ou vers un nœud mobile est un autre défi. Ainsi, la capture peut être aussi bien statique que dynamique dépendant de l'application.

- **L'agrégation des données :** Les techniques d'agrégation des données concernent le traitement des données par le réseau, permettent de réduire le nombre de messages et par conséquent réduire la consommation en énergie. Dans les RCSFs, les données produites par les nœuds capteurs sont très corrélées, ce qui implique l'existence de redondances de données. Les utilisateurs sont intéressés par le phénomène qui est saisi par les données générées par chaque nœud et par conséquent, ces réseaux fournissent la possibilité d'agréger les données afin de réduire la largeur de la bande passante.

1.4.3 Architecture protocolaire

La pile de protocoles utilisée par le puits (sink) ainsi que par tous les nœuds-capteurs est donnée dans la figure 1.2. Cette pile de protocoles combine routage et gestion d'énergie et intègre les données avec les protocoles réseau. Elle communique de manière efficace (en termes d'énergie) à travers le support sans fil et favorise les efforts de coopération entre les nœuds-capteurs. La pile de protocoles comprend une couche application, une couche transport, une couche réseau, une couche liaison de données, une couche physique, un plan de gestion d'énergie, un plan de gestion de mobilité et un plan de gestion des tâches. Selon les tâches de détection, différents types de logiciels d'application peuvent être construits et utilisés dans la couche application. La couche transport contribue au maintien du flux de données si l'application du réseau de capteurs l'exige. La couche réseau s'occupe de l'acheminement des données fournies par la couche transport.

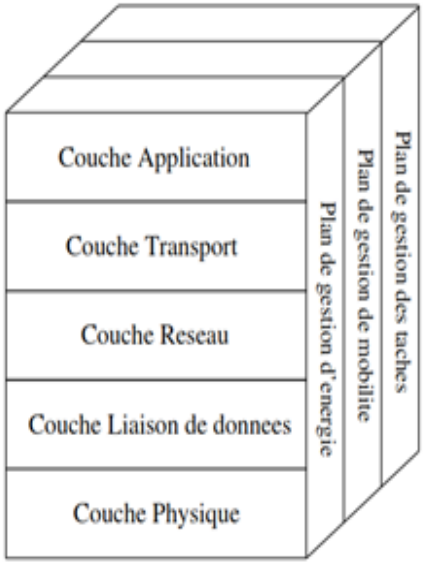

FIGURE 1.2 – la pile protocolaire des réseaux de capteurs

Comme l'environnement est sujet au bruit et que les nœuds-capteurs peuvent être mobiles, le protocole MAC doit tenir compte de la consommation d'énergie et doit être en mesure de réduire les collisions entre les nœuds voisins lors d'une diffusion par exemple. La couche physique répond aux besoins d'une modulation simple mais robuste, et de techniques de transmission et de réception. En outre, les plans de gestion d'énergie, de mobilité et des tâches surveillent et gèrent la consommation d'énergie, les mouvements, et la répartition des tâches entre les nœuds-capteurs. Ces plans aident les nœuds-capteurs à coordonner les tâches de détection et à réduire l'ensemble de la consommation d'énergie.

1.5 Les Problématiques et contraintes liées au RCSFs

Un réseau de capteurs possède plusieurs contraintes qui influencent sur la conception et la mise en place d'un réseau de capteurs sans fils parmi lesquels :

1.5.1 La topologie du réseau

Le déploiement d'un grand nombre de nœuds nécessite une maintenance de la topologie. Cette maintenance se fait en trois phases :

- **Déploiement :** les nœuds peuvent être soit éparpillés en masse ou bien placés un par un dans le champ de perception.
- **Post-déploiement :** les capteurs peuvent bouger, ne plus fonctionner, etc...
- **Redéploiement :** des nœuds additionnels sont déployés pour remplacer les nœuds mal fonctionnant.

1.5.2 L'environnement

Les capteurs sont souvent déployés en masse dans des endroits hostiles tels que des champs de bataille au-delà des lignes ennemies, à l'intérieur de grandes machines, etc. Par conséquent, Ils doivent pouvoir fonctionner sans surveillance dans des régions géographiquement éloignées ou inaccessibles.

1.5.3 La tolérance aux pannes

Certains nœuds capteurs peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, d'une défaillance matérielle ou d'une interférence, la tolérance aux pannes

c'est la capacité de maintenir les fonctionnalités du réseau sans interruption en cas de défaillance d'un nœud capteur.

1.5.4 Le passage à l'échelle (scalabilité)

Le nombre de nœuds déployés peut atteindre le million. Un nombre aussi important de nœuds engendre un trafic énorme dans le réseau, ce qui entraine des congestions et des erreurs de communication. Un tel déploiement, nécessite que le protocole utilisé pour la communication soit capable, de détecter les erreurs et de contrôler le flux, et nécessite aussi que la station de base soit équipe d'une capacité de stockage suffisante Pour accueillir les informations reçues à partir des nœuds capteurs.

Au futur Les capteurs sans fils sont destinés à avoir des dimensions microscopiques permettant de les disperser par milliers voire des dizaines de milliers dans la zone à étudier, ce qui rend ce problème imposant et parmi les plus important dans les RCSFs.

1.5.5 Connectivité

La densité élevée des nœuds dans les réseaux de capteurs les empêche d'être complétement isolés les uns des autres. Ceci, cependant, n'empêche pas la topologie du réseau d'être variable. En outre, la connectivité dépend de la distribution aléatoire des nœuds.

1.5.6 La tolérance aux intrusions

L'absence d'une protection physique des nœuds capteurs ainsi que la nature des liens sans fils utilisés pour la communication, rend le réseau vulnérable aux attaques, la tolérance aux intrusions implique la tolérance aux vulnérabilités.

1.5.7 Le support de transmission

Afin d'être installés sans difficulté dans des zones ciblées et sans induire d'importants couts de câblage, les capteurs utilisent des liens radiofréquences pour coopérer entre eux au sein du réseau, trois grandes normes sont utilisées : IEEE 802.11x(Wifi), IEEE 802.15.1(Bluetooth), IEEE 802.15.4/(Zigbee).

Le tableau 1.2 reprend les différentes caractéristiques de ces 3 protocoles de communication sans fil.

Protocole	**Bluetooth**	**Wifi**	**Zigbee**
Norme IEEE	802.15.1	802.11x	802.15.4
Durée de vie moyenne sur pile	plusieurs jours	plusieurs heures	plusieurs années
Nombre de nœud Maximum	8	2007	65 536
Débit Théorique Maximum	1 Mb/s	320 Mb/s	250 Kb/s
Bande de Fréquence	2.4 GHz	2.4 GHz ; 5 GHz	868/915 MHz ; 2.4 GHz
Porté théorique Maximum	100 mètres	300 mètres	100 mètres

TABLE 1.2 – Comparaison des protocoles de communication pour les RCSFs [6]

1.5.8 Les contraintes matérielles

Parmi les contraintes matérielles liées aux RCSFs, on peut citer :

- **La dimension :** La taille réduite des nœuds capteurs peut présenter plusieurs avantages, et elle permet un déploiement flexible et simple du réseau. Cependant, la puissance des batteries utilisées pour alimenter les nœuds capteurs est limitée, par la petite taille de ces derniers.
- **La puissance de calcul :** Les réseaux de capteurs sont différents par rapport aux réseaux traditionnels. Parmi les principaux points de différence nous pouvons citer la puissance de calcul. Les nœuds capteurs utilisent souvent des microcontrôleurs de faibles fréquences.
- **La consommation énergétique :** Comme les nœuds capteurs sont des composantes microélectroniques, ils sont équipés d'une ou plusieurs batteries normales et irremplaçables, par conséquent cette ressource et la plus précieuse dans un réseau de capteurs car la durée de vie d'un nœud capteur dépend fortement de la durée de vie de sa batterie. De ce fait, la faible consommation d'énergie est une exigence principale pour les applications ou une longue durée de vie du réseau est nécessaire.

1.5.9 La gestion des ressources

Le point crucial qui caractérise les RCSFs est la limitation des ressources : l'énergie disponible, la puissance de calcul. La complexité se situe au niveau des restrictions de ressources à considérer dans les algorithmes de cryptage, le contrôle des erreurs, etc.

1.5.10 La gestion des données collectées

La gestion des données collectées rassemble l'ensemble des traitements subis par les données durant leurs cycles de vie au sein du RCSFs. Les phases d'acquisition et de présentation des données à l'utilisateur viennent s'ajouter aux actions.

1.5.11 L'adressage

Le problème de l'adressage est lié à celui de la mise à l'échelle. Le nombre important de capteurs sans fils déployé oblige à se poser des questions sur l'adressage à utiliser. Deux méthodes sont utilisées :

- **Identification unique des capteurs :** consiste à offrir un identifiant unique à chaque capteur.
- **Identification par localisation :** consiste à se centrer sur les données en leurs associant un repère spatial et temporel.

1.5.12 La sécurité

Les RCSFs nécessitent dans nombreuses applications des solutions qui assurent la sécurité des informations circulant sur le réseau. La sécurité des informations circulant dans le réseau doivent répondre à plusieurs prérequis : [6]

- **Confidentialité :** le réseau doit s'assurer que les données transmises soient confidentielles et ne puissent être lues par des dispositifs ou personnes autres que ceux ayant droit de le faire.
- **Authentification :** L'authentification des capteurs est nécessaire pour s'assurer que l'identité déclarée par un capteur est bien celle du capteur déclarant.
- **Intégrité des données :** Les données circulant sur le réseau ne doivent pas pouvoir être altérées au cours de la communication. Il faut donc s'assurer que personne ne puisse capturer et modifier les données du réseau.
- **Fraîcheur des données :** Par fraîcheur des données, nous entendons savoir si la donnée est récente ou non. Cela signifie qu'il faut s'assurer que la donnée transmise corresponde à un état présent.
- **Disponibilité du réseau :** Le réseau doit pouvoir être disponible à tout instant, c'est-à-dire que l'envoi d'information ne doit pas être interrompu, de même que la circulation de l'information ne doit pas être stoppée.
- **Auto organisation :** Les capteurs du réseau doivent être capables, après avoir été déployés, de s'auto-organiser et surtout de se sécuriser eux-mêmes, sans autres interventions extérieures (mettre en place des clés cryptographiques).

Avant de mettre en place une application de réseau de capteurs il faut essentiellement résoudre la problématique du routage puis vient ensuite la sécurité [7].

1.6 Le Routage dans les RCSFs

Le routage permet l'acheminement des informations vers une destination donnée à travers un réseau de connexion. Le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau tout en tenant compte des critères de performance comme la consommation énergétique. Le problème consiste à trouver l'investissement de moindre coût qui assure le routage du trafic nominal et garantit la qualité de service.

Le problème qui se pose dans le contexte des réseaux de capteurs est l'adaptation de la méthode d'acheminement utilisée avec le grand nombre de nœud existant dans un environnement caractérisé par de changements de topologies, de modestes capacités de calcul, de sauvegarde, et d'énergie. Selon [8], le routage dans les réseaux de capteurs peut être classé selon la topologie du réseau, on aura donc le routage à plat et le routage hiérarchique.

1.6.1 Le Routage non hiérarchique (plat)

Dans le routage plat chaque nœud joue typiquement le même rôle et les nœuds capteurs collaborent pour accomplir la tâche globale du réseau. En raison du nombre important des nœuds capteurs, il n'est pas faisable d'assigner un identifiant global pour chaque nœud. Cette considération a mené au routage centré-données, où la station de base envoie des requêtes à certaines régions du réseau et attend des retours de données à partir des nœuds capteurs situés dans ces régions. Les premiers travaux sur ce type de routage et le protocole SPIN[9] qui permet de disséminer des informations sur le réseau de manière ciblée. Le fonctionnement du protocole SPIN permet de réduire la charge du réseau par rapport aux méthodes de diffusion traditionnelles telles que l'inondation.

On trouve aussi le protocole DSDV [10] ; un protocole proactif de routage basé sur l'algorithme distribué de Bellman-Ford [11], AODV [12] ; protocole similaire à DSDV mais il est réactif, il demande une route que lorsqu' il en a besoin, GSR [13] ; un protocole similaire à DSDV mais n'utilisant pas l'inondation à chaque modification topologique, FSR [14] ; un protocole amélioré de GSR basé sur l'utilisation de la technique "œil de poisson" [15].

1.6.2 Le routage hiérarchique

Dans un routage hiérarchique, des nœuds à grande énergie peuvent être employés pour traiter et envoyer l'information, alors que des nœuds à énergie réduite peuvent assurer la capture à proximité de la cible. La création des clusters et l'assignation des tâches spéciales aux têtes de clusters peuvent mieux supporter le passage à l'échelle, l'augmentation de la durée de vie et l'efficacité énergétique du système global. Le routage hiérarchique est une manière efficace de réduire la consommation énergétique dans un cluster en exécutant l'agrégation et la fusion de données afin de diminuer le nombre de messages transmis à la station de base.

Dans ce type de routage on trouve plusieurs travaux, parmi eux : ZHLS [16] ; basé sur la décomposition du réseau en un ensemble de zones disjointes sans élire de représentants contrairement à d'autres protocoles hiérarchiques, CBRP [17] ; un protocole réactif dont l'ensemble des nœuds du réseau est décomposé en clusters, TEEN [18] ; un protocole utilisant la technique de clustering pour les applications critiques où le changement de certains paramètres peut être brusque.

L'un des protocole les plus utilisés dans les réseaux de capteurs sans fils est le protocole LEACH[19] qui est un protocole basé sur le clustering et possède des propriétés qui permettent d'économiser l'énergie et d'augmenter la durée de vie du réseau.

Dans la partie qui suit on va détailler le fonctionnement de ce protocole et on présentera ces avantages et inconvénients.

1.6.3 Le protocole LEACH

Le protocole LEACH est un protocole qui utilise la technique du clustering ; les nœuds du réseau s'organisent eux-mêmes dans des clusters avec un nœud qui joue le rôle du cluster-head. Dans les protocoles qui utilisent aussi le clustering il est facile de voir que les nœuds malchanceux qui sont choisi comme cluster-head meurent rapidement puisque ils épuisent leur énergie ce qui met fin à la durée de vie utile pour tous les nœuds du cluster, ainsi LEACH comprend une rotation aléatoire c'est-à-dire les cluster-heads ne sont pas fixes et le nœud avec la plus grande quantité d'énergie restante poura lui-même devenir cluster-head et éviter ainsi le problème cité auparavant. On addition à cela : LEACH effectue une fusion de données locale qui consiste à compresser la quantité de données étant envoyé à partir des clusters à la station de base, réduisant encore la dissipation d'énergie et l'amélioration de la durée de vie du système. LEACH est exécuté en deux phase : la phase "setup" dans laquelle les clusters-heads

sont sélectionnés et les clusters sont formés suivis de la phase "steady-state" dans laquelle le transfert de données vers la station de base aura lieu.

1.6.3.1 La phase d'initialisation (setup phase) :

Elle commence d'abord par choisir les différents cluster-heads ; chaque nœud décide ou non de devenir cluster-head pour le tour courant, cette décision est basée sur un pourcentage suggérer des clusters-heads pour le réseau (déterminer à priori) et sur le nombre de fois que ce nœud a était cluster-head jusqu'ici.

Pour bien illustrer ça on prend un exemple : Soit un nœud n qui doit prendre une décision de devenir cluster-head ou non. Pour cela, il choisit un nombre aléatoire entre 0 et 1, si ce nombre est inférieur à un seuil *T(n)* il est alors cluster-head pour le tour courant, ce seuil *T(n)* est définit comme suit :

$$T(n) = \begin{cases} \frac{P}{1-P*(r \bmod 1/P)} & \text{si} \quad n \epsilon G \\ 0 & \text{sinon} \end{cases}$$

Où P représente le pourcentage suggérer des clusters-heads, r représente le tour courant, G c'est l'ensemble des nœuds qui n'ont pas était cluster-head dans les $1/P$ tours précédents. En utilisant ce seuil, chaque nœud sera cluster-head a un certain moment, dans le tour 0 (r=0) chaque nœud possède une probabilité P de devenir cluster-head, les nœuds choisit pour être clusters-heads dans ce tour ne peuvent plus l'être durant les $1/P$ tours suivants.

Après la formation des cluster-heads pour le tour actuel, chacun de ces derniers diffuse un message d'annonce aux autres nœuds. Pour cette étape, les cluster-heads utilisent un protocole CSMA pour la diffusion du message et la même énergie d'émission, les autres nœuds doivent maintenir leurs récepteurs actifs pour recevoir le message des cluster-heads. Après la réception des messages annonces, chaque nœud qui n'est pas cluster-head pour ce tour décide du cluster dans lequel il va appartenir, ce choix est basé sur la force du signal du message annonce reçu d'un cluster-head.

La figure 1.3 présente l'algorithme d'élection des cluster-heads :

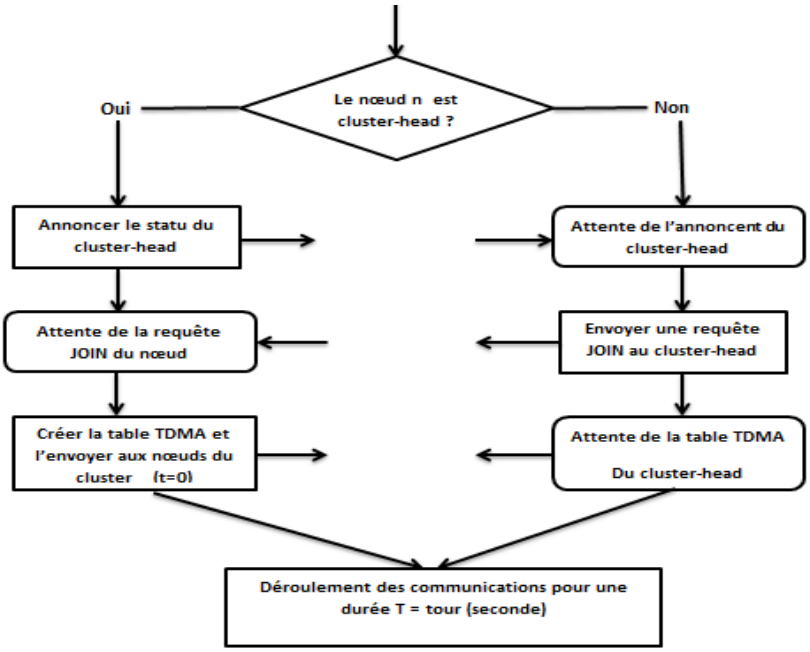

FIGURE 1.3 – Schéma illustrant l'algorithme de création des clusters [20]

Après que chaque nœud ait choisi le cluster auquel il va appartenir, ce dernier doit avertir le cluster-head qu'il sera membre de son cluster, en utilisant toujours un protocole CSMA, il envoie un message au cluster-head, pour cette étape les récepteurs des cluster-heads doivent être activés. Après que le cluster-head ait reçu les messages des nœuds qui voudraient être inclus dans le cluster et basé sur le nombre de nœuds dans ce cluster, le cluster-head crée un calendrier TDMA pour dire à chaque nœud quand il peut transmettre, ce calendrier est alors diffusé vers tous les nœuds du cluster.

1.6.3.2 La phase de transmission (steady-state phase) :

Une fois les clusters créés, la transmission des données peut commencer. En supposant que les nœuds ont toujours des données à envoyer, il envoie pendant leurs temps alloués par le cluster-head, cette transmissions utilise une quantité minimale d'énergie (choisie en fonction de la force du message d'annonce reçu), la radio de chaque nœud non cluster-head est désactivé pendant l'attente jusqu'à ce que son temps de transmission, minimisant ainsi la consommation d'énergie. Le cluster-head doit garder toujours son récepteur actif pour recevoir toutes les données des nœuds du cluster. Lorsque toutes les données ont été reçues, le cluster-head exécute des fonctions de traitement de signaux pour compresser les données en un signal unique. Ce

signal est envoyé à la station de base. Etant donné que la station de base est loin, il s'agit d'une transmission à haute énergie. C'est le fonctionnement général du protocole LEACH pour un cycle, après un certain temps qui est déterminer à priori un autre cycle commence est on recommence toutes les étapes depuis le choix des cluster-heads.

1.6.3.3 Avantages du protocole LEACH

- L'auto-configuration des clusters se fait indépendamment de la station de base (algorithme distribué).
- Les données sont fusionnées pour réduire la quantité d'informations transmise vers la station de base.
- La consommation d'énergie est partagée sur l'ensemble des nœuds prolongeant ainsi la durée de vie du réseau.
- L'utilisation des techniques TDMA/CDMA permet d'avoir une hiérarchie et de réaliser des clusterings sur plusieurs niveaux. Ces derniers permettent d'économiser d'avantage d'énergie.

1.6.3.4 Inconvénients du protocole LEACH

- LEACH choisit aléatoirement la liste des clusters heads et il ne pose aucune contrainte sur leur distribution ainsi que sur leur niveau d'énergie. Ainsi, les clusters heads peuvent se concentrer dans un même endroit et par conséquent, il pourrait exister des nœuds isolés (sans cluster head) pouvant se déclarer.
- Sans justifier leur choix, les auteurs fixent le pourcentage optimal de CHs pour le réseau à 5 pour-cent du nombre total des nœuds. Néanmoins, la topologie, la densité et le nombre de nœuds peuvent être différents dans d'autres réseaux.
- Aucune suggestion n'est faite à propos du temps de réélection des CHs (temps des itérations).
- Les CHs les plus éloignés de la station de base meurent rapidement par rapport à ceux qui sont proches de la station.
- L'énegie résiduelle des nœuds n'est pas prise en compte.

Pour essayer de contrer ces failles plusieurs améliorations du protocole LEACH on était proposé comme LEACH-C[21], PEGASIS[22], etc. Nous allons détaillé une de ces améliorations du protocole LEACH c'est le CELL-LEACH[23]que nous allons utiliser dans notre proposition dans les chapitres à venir.

1.6.4 Le protocole CELL-LEACH

Le protocole CELL-LEACH est une amélioration du protocole LEACH, un algorithme de routage basé sur le clustering. Ce protocole de routage prend en considération l'énergie résiduelle de nœuds pour prolonger la durée de vie des réseaux de capteurs.

Dans cette méthode proposée, le réseau de capteurs est divisé en sections qui sont appelées cellules. Chaque cellule comprend plusieurs capteurs. Un capteur qui est à l'intérieur de la cellule est sélectionné en tant que chef de la cellule (cell-head). Chaque groupe de sept cellules proches forment un cluster, chacun muni d'un capteur qui est connu en tant que tête de cluster (cluster-head). Les clusters et les cellules créés resteront les mêmes tant que réseau fonctionne, seul les cell-heads et les cluster-heads changent dynamiquement.

La figure 1.4 montre l'organisation de réseau en cluster et cellule par le CELL-LEACH

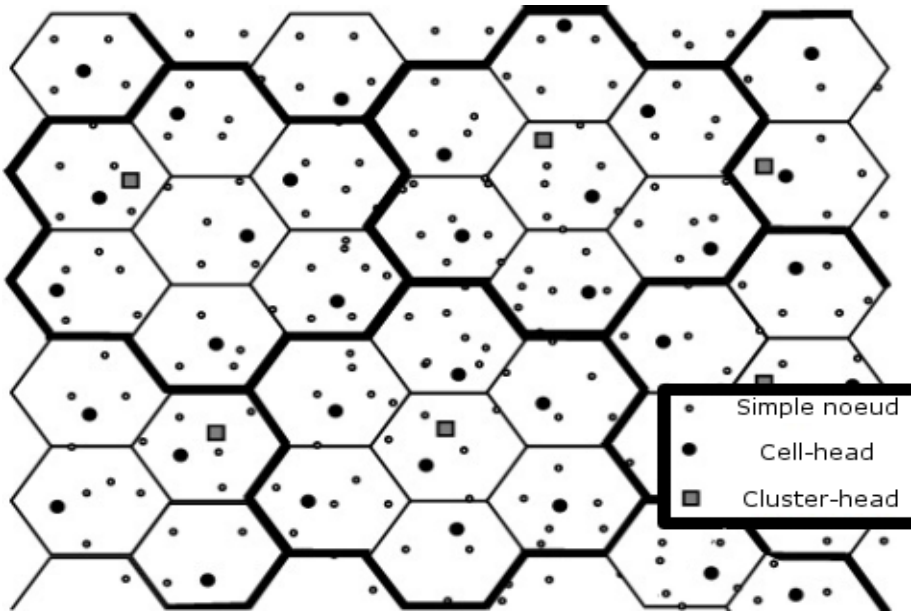

FIGURE 1.4 – L'organisation du réseau par le CELL-LEACH

Après la configuration du réseau, un message est transmis de la station de base à tous les cluster-heads. Ensuite chaque cluster-head transmet le message à tous les cell-heads. Enfin chaque cell-head transmet le message à tous les capteurs à l'intérieur de la cellule. Lorsqu'un cell-head reçoit une réponse de l'un des capteurs lui appartenant, il alloue une limite de temps TDM (Time Division Multiplexing). Chaque capteur doit transmettre sa réponse au cell-head dans le temps désigné. Cette méthode est également utilisée pour transférer des données du

cell-head vers le cluster-head.

Pour pouvoir envoyer le message d'un cluster-head vers la station de base, chaque cluster-head va conserver les informations de localisation des autres clusters dans sa table. Ce tableau est mis à jour à chaque fois qu'un cluster-head est choisi. En utilisant cette table, le chemin le plus court est sélectionné pour envoyer des données de cluster-head vers la station de base. la figure 1.5 montre l'acheminement des messages depuis le nœud de la cellule vers la station de base.

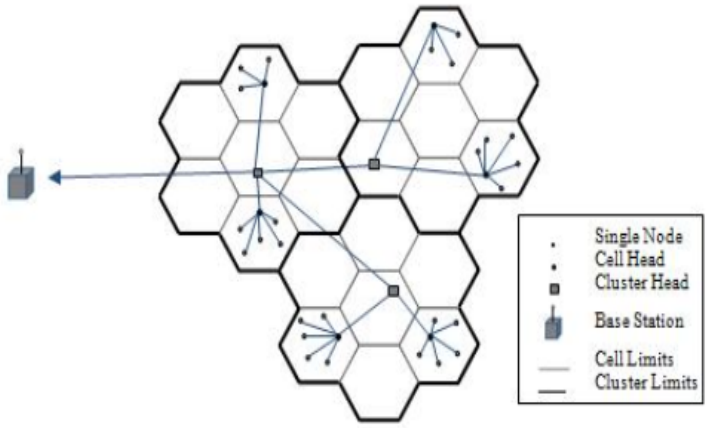

FIGURE 1.5 – La communication dans le protocole Cell-LEACH

La sélection du cell-head et du cluster-head se fait de la même façon. Au début, après le déploiement du réseau, le cell-head à l'intérieur de chaque cellule et le cluster-head à l'intérieur de chaque cluster est déterminé de manière aléatoire, étant donné que tous les capteurs possèdent la même énergie. Après une certaine période, chaque ancien cell-head et ancien cluster-head est remplacé par un nouveau dynamiquement.

Une méthode d'évaluation de l'énergie moyenne consommée est utilisée par ce protocole. A la fin d'une période de temps, tous les capteurs d'une cellule transmettent un message contenant leur quantité d'énergie restante au cell-head. Leurs $E_{avg-cell}$ sont calculées et transmises au cluster-head. Celui qui possède l'énergie la plus élevée sera choisi comme étant le nouveau cell-head. Après le cluster-head va calculer la moyenne des $E_{avg-cell}$ pour déterminer l'énergie du cluster $E_{avg-cluster}$. Cette méthode conduit à une consommation d'énergie équilibrée des nœuds dans le processus de sélection des cluster-heads et cell-heads. Ce protocole peut contrôler l'énergie des nœuds capteurs et prolonge la durée de vie du réseau sans dégradation des performances.

1.6.4.1 Méthode de calcul de l'énergie moyenne restante dans un cluster

Après chaque tour, les nœuds d'une cellule envoient leurs énergies restantes vers le cell-head. L'énergie moyenne restante de la cellule ($E_{avg-cell}$) sera alors calculé et envoyé au cluster-head. Ensuite, le nœud de la cellule qui a la plus grande quantité d'énergie restante deviendra le nouveau cell-head. Après l'envoi des $E_{avg-cell}$ au cluster-head ce dernier va calculer $E_{avg-cluster}$ et procéder a la désignation du nouveau cluster-head. Le calcul de $E_{avg-cell}$ ne se fait pas à chaque tour, en effet à chaque tour le nœud avec la maximum énergie comparé au $E_{avg-cell}$ sera nommé nouveau cell-head à chaque tour. L'étape du calcul de $E_{avg-cell}$ sera répétée lorsqu'il n y a plus aucun nœud qui possède une énergie résiduelle supérieure à $E_{avg-cell}$ et c'est le même principe pour $E_{avg-cluster}$.

1.7 Conclusion

Les réseaux de capteurs restent une nouvelle technologie peu accessible au grand publique. La flexibilité, la tolérance aux fautes, le prix réduit et les moyens rapides de déploiement des réseaux de capteurs annoncent un futur prometteur à cette technologie. Cependant, la mise en place d'une application de RCSF doit prendre en considération les contraintes qui caractérisent les nœuds-capteurs. Dans ce chapitre, nous avons présenté les contraintes relatives aux RCSFs avec les problématiques liées à ces derniers, mais aussi nous avons mis l'accent sur le routage dans ce type de réseau en présentant les diffèrentes classifications et en détaillant les protocoles de routage qui seront utilisés dans la suite de ce document.

CHAPITRE 2

GESTION DES CLÉS DANS LES RÉSEAUX DE CAPTEURS SANS FIL

2.1 Introduction

La gestion de clés est un service primordial pour la sécurité de n'importe quel système basé sur la communication. Sous les contraintes des RCSFs, la conception d'un système de gestion de clés est un grand défi. Sélectionner une solution cryptographique appropriée pour les RCSFs est un autre défi.

Pour qu'un système fonctionne, il est important que chacun des utilisateurs disposent d'un ensemble de clés secrètes utilisées lors des opérations. Cela implique de les générer et les distribuer d'une manière sécurisée entre eux ou leurs offrir un moyen de générer ces clés. Ce dernier doit être en mesure d'enregistrer et de gérer ses clés d'une manière sûre [24].

La cryptographie à clé publique (asymétrique) fournit des mécanismes plus sûrs et fiables mais exige un espace mémoire assez grand et de haute capacité de calcul, ce qui la rend inappropriée pour les RCSFs.

Cependant, des recherches [25] [26] ont montré qu'il est possible d'appliquer la solution à clé publique aux réseaux de capteurs en choisissant les bons algorithmes et les paramètres appropriés. Les études [26] [27] [28] montrent que la cryptographie de courbe elliptique ECC (Elliptic Curve Cryptography) [29] a un avantage significatif par rapport au RSA, car elle réduit le temps de calcul ainsi que la quantité de données transmises et stockées. Quelques autres travaux [30] [31] [32] se concentraient sur la mise en œuvre de l'ECC dans les RCSFs.

La cryptographie à clé secrète possède ses propres qualités qui en fait la préférée pour les RCSFs. Par conséquent, les schémas de gestion de clés proposés pour les RCSFs sont basés sur la cryptographie symétrique.

2.2 Phases de gestion de clés

La gestion de clé comporte plusieurs phases [33] :

2.2.1 Phase de pré-distribution des clés

Chaque nœud doit avoir une clé ou un ensemble de clés avant le déploiement. On utilise la pré-distribution pour installer des clés dans les nœuds avant le déploiement afin de sécuriser la transmission des paquets.

2.2.2 Découverte du voisinage

Après le déploiement, chaque nœud doit découvrir ses voisins qui sont à sa portée et avec lesquels il partage des clés. Un lien existe entre deux nœuds capteurs seulement s'ils partagent une clé. Ce schéma ne laisse aucune occasion à un attaquant de découvrir la clé partagée entre deux nœuds voisins. [33]

2.2.3 Etablissement de clés de chemin

Une clé de chemin " path Key " est importante pour les nœuds non liés directement mais qui sont reliés par un chemin multi sauts afin de sécuriser la communication de bout en bout, cette clé doit être différente de la clé partagée entre les nœuds voisins.

2.2.4 Isolation des nœuds malveillants

L'identification et l'isolation des nœuds anormaux est importante pour continuer les opérations dans les RCSFs car ils agissent comme des nœuds intermédiaire et la compromission d'un de ses nœuds intermédiaire implique la compromission de toute la communication. Il existe plusieurs raisons pour qu'un nœud ne fonctionne pas normalement parmi elles : l'épuisement de son énergie, l'endommagement par un attaquant, etc[33].

2.2.5 Mise à jour et renouvellement de la clé

Une fois que la clé a expiré, la mise en service de nouvelles clés est primordiale et doit être faite d'une manière sécurisée, efficace et conforme à une consommation d'énergie réduite.

2.2.6 Latence d'établissement des clés

Réduire la latence résultante des communications et conserver l'énergie constitue un objectif primaire dans le processus de gestion des clés. Tout schéma de gestion des clés devrait prendre la réduction de latence comme un facteur crucial.

2.3 Solutions Introductives

2.3.1 Clé partagée par le réseau

La solution la plus simple 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 a la vulnérabilité suivante :

- Elle 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.

2.3.2 Clé partagée par-paire de nœuds

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). L'avantage de cette solution est que la résilience est parfaite car la compromission d'un nœud n'affecte pas la sécurité des autres nœuds.

Par contre cette solution n'est pas appropriée aux RCSF car elle exige une capacité mémoire importante pour stocker les N-1 clés (N peut être grand), et l'ajout de nouveaux nœuds est difficile parce que les nœuds existants ne possèdent pas les clés de ces nouveaux nœuds.

2.3.3 Solution basée sur la station de base (SPINS)

Les auteurs de SPINS [34] ont proposé une méthode pour achever l'établissement de clé entre deux nœuds à l'aide de la station de base. Cette dernière est considérée comme tierce partie de confiance avec laquelle chaque nœud partage une clé secrète. 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 permet une connectivité totale, où chaque nœud peut partager une clé avec n'importe quel autre nœud du réseau et une résilience parfaite contre la compromission d'un nœud.

Par contre, cette solution n'est pas appropriée aux RCSFs car elle ne permet pas le passage à l'échelle à cause du nombre de messages requis, entre la station de base et les nœuds capteurs, afin d'installer des clés symétriques entre deux nœuds communicants.

2.4 Classification des protocoles de gestion de clés

La gestion des clés est le processus par lequel des clés cryptographiques sont produites, enregistrées, protégées, transférées, chargées, employées, et détruites [35]. Cette gestion est habituellement décrite par le procédé de pré-distribution de clés qui exige un chargement d'information secrète dans les nœuds capteurs avant leur déploiement dans le réseau. Cette information secrète, déployée dans le réseau, peut être une clé secrète, ou de l'information auxiliaire qui aide les nœuds à dériver la clé secrète réelle. Il existe deux types d'approches pour la gestion des clés dans les réseaux sans fils : approche de cryptographie symétrique et approche de cryptographie asymétrique.

2.4.1 Approche de cryptographie asymétrique

Avant le déploiement, chaque nœud du réseau possède les clés maîtresses publiques et privées, puis chaque nœud A génère sa paire de clés. Après le déploiement, les nœuds échangent les clés (échange des clés publiques et une signature par la clé maîtresse pour la vérification des clés publiques reçues). Ensuite, une clé symétrique peut être générée et échangée entre les nœuds encryptées par leurs clés publiques. L'utilisation de la cryptographie asymétrique dans les RCSF permet la scalabilité et la résistance contre la capture des nœuds. Cependant cette approche est exigeante (augmentation des coûts, beaucoup de calcul, consommation d'énergie,etc) et rend les RCSF vulnérables aux attaques de déni de service par épuisements de batteries. Parmi les

approches existantes on trouve :

2.4.1.1 TinyPK

TinyPK (Tiny Public Key) [36] est un projet qui emploie le crypto-système de RSA pour manipuler la distribution des clés symétriques. Mettre en application un système de clé publique exige une quantité modeste d'infrastructure comprenant une autorité de certificat (CA). La clé publique de CA est pré-chargée sur chaque nœud, les normes actuelles ralentissent l'exécution de TinyPK. Le temps d'exécution pour le chiffrement est proportionnel à la taille de la clé utilisée. Les auteurs de TinyPK admettent qu'actuellement sa mise en place est trop lente pour des RCSF. Ils suggèrent de l'utiliser comme méthode d'authentification et de déplacer, si possible, les exécutions de calcul couteux (calcul qui demande une énergie considérable) vers des dispositifs puissants du réseau. Bien que la cryptographie à clé public possède beaucoup d'avantages, elle est actuellement infaisable pour une transmission de nœud-à-nœud dans des réseaux de capteurs.

2.4.1.2 TinyECC

C'est la cryptographie sur les courbes elliptiques, ces dernières peuvent être utilisées pour des opérations asymétriques comme des échanges de clés sur un canal non sécurisé, on parle de ECC (Elliptic Curve Cryptosystem) [37]. L'usage des courbes elliptiques en cryptographie a été suggéré par Neal Koblitz et Victor Miller en 1985.

L'avantage d'ECC est l'emploi de clés plus courtes que d'autres méthodes de cryptographique asymétrique telle que RSA, tout en fournissant un niveau équivalent ou plus élevé de sécurité. ECC emploie des points sur une courbe elliptique pour dériver une clé publique de 160 bits qui est équivalente, relativement au niveau de sécurité, à une clé de 1024 bits de l'algorithme RSA [38].

Par conséquent, une plus petite taille de clé permet d'exécuter plus rapidement l'opération de chiffrement/déchiffrement et exige un besoin moindre en mémoire. Un inconvénient, cependant, est que l'exécution des opérations de chiffrement/déchiffrement par ECC prend plus de temps que dans la cryptographie symétrique.

2.4.2 Approche de cryptographie symétrique

2.4.2.1 Absence de pré-distribution de clés

Aucune pré-distribution de clés, ce mécanisme considère la réalité des RCSF. Si on ne sait pas où et quand des nœuds sont déployés, il serait difficile pour lancer une attaque active.

INF [39], un schéma de gestion de clés prévu pour les RCSFs, suppose un déploiement de masse et que les nœuds sont statiques. INF installe des clés symétriques entre les nœuds et leurs voisins d'un seul saut. La sécurité est basée sur la surprise : elle est fondée sur le modèle réaliste de l'attaquant, dans la phase de déploiement de réseau, n'importe quel attaquant peut seulement surveiller un petit pourcentage des voies de transmission.

En premier temps, chaque nœud génère simplement une clé symétrique et l'envoie dans l'espace libre à ses voisins. Une approche de chuchotement de clé est employée, c.-à-d., la clé est au commencement transmise à un niveau de puissance bas. La puissance de transmission est alors augmentée jusqu'à ce que la clé soit entendue par au moins un voisin d'un seul saut et qu'une réponse soit reçue.

Il y a très peu de probabilité d'intercepter une telle communication locale et de faible portée, et il n'est pas possible de surveiller tous les nœuds déployés. En outre, l'échange de clés a une durée de quelques secondes à comparer avec la durée de vie du réseau qui se compte en mois voire en années.

2.4.2.2 Les protocoles basés sur la pré-distribution de clés

Les protocoles basés sur la pré-distribution de clés dans les RCSFs consistent à un chargement des clés dans les nœuds avant leur déploiement. La figure ci-dessous illustre une taxonomie des solutions de gestion des clés basées sur la pré-distribution :

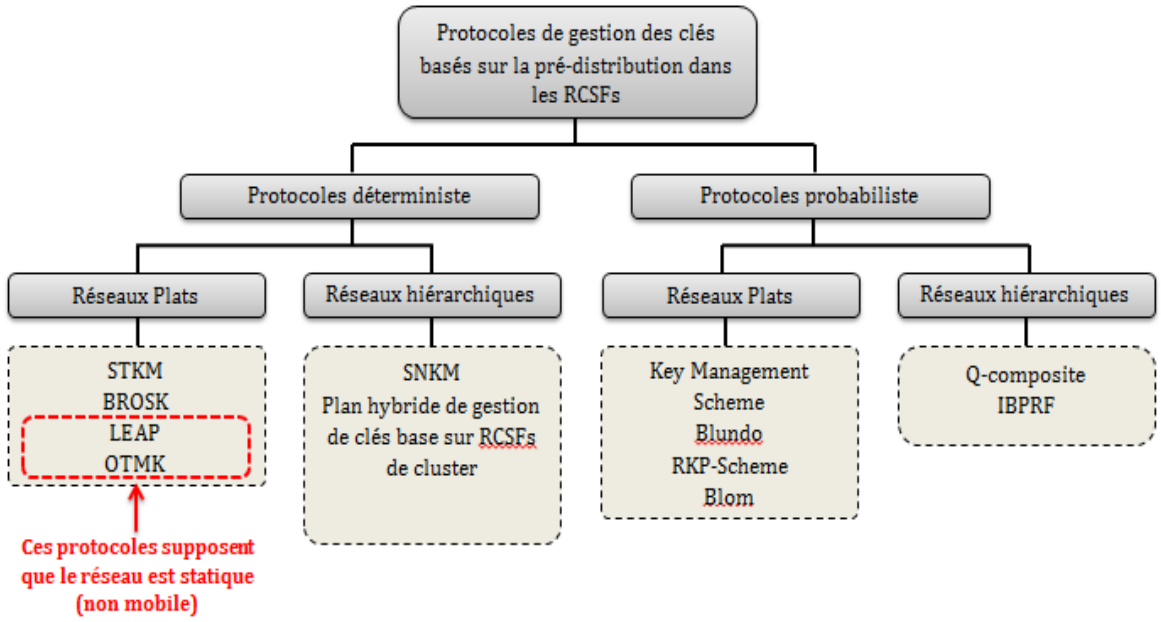

FIGURE 2.1 – Taxonomie de pré-distribution de clé pour les RCSF [40]

Protocoles de gestion de clés déterministes

Le principe de ces protocoles consiste à utiliser une clé maîtresse pour dériver des clés qui seront utilisé entre paire de voisins dans le réseau. D'une autre manière chaque nœud sera capable d'établir une clé par-paire avec ses voisins. Ces protocoles sont les premiers à avoir été proposé dans la littérature.

► **Protocole SPINS**

Le protocole SPINS [34] (Security Protocols for Sensor Networks) dispose de deux blocs de sécurité, à savoir SNEP, qui utilise deux mécanismes de sécurité, le premier consiste à chiffrer les données pour assurer leur confidentialité et le second de calculer un code MAC pour assurer l'authentification et l'intégrité de données, et uTESLA qui utilise la diffusion (broadcast) authentifié. Pour permettre cette authentification, la station de base rajoute au message un code MAC calculé à partir d'une clé secrète. Le protocole SPINS propose une solution garantissant la confidentialité, l'authenticité des donnée mais présente quelques inconvénients. En effet le protocole utilise un chiffrement DES qui n'est pas si sûr que ça et uTESLA nécessite un envoi de donnée de façon permanente ce qui consomme beaucoup d'énergie.

► **Protocole PIKE**

Le protocole PIKE [41] consiste à utiliser un ou plusieurs nœuds de capteurs comme un intermédiaire de confiance pour faciliter l'établissement de clés. L'idée de base du PIKE est d'utiliser des nœuds de capteurs comme des intermédiaires de confiance et établir des clés partagées entre les nœuds. Chaque nœud partage une clé " pairwise " avec $O(\sqrt{n})$ (n est le nombre de nœud dans le réseau), ces clés sont déployé tel que entre deux nœuds A et B, il est possible de trouver un nœud C dans le réseau qui partage une clé paire unique avec A et B. A peut alors en toute sécurité passer le message d'établissement de clé à C, puis ce dernier va le transmettre à B. Soit un réseau avec n nœud et on associe à chaque nœud un ID de la forme (x,y) tel que : x et y appartiennent à 0,1,2,3,...,.$\sqrt{n-1}$. Chaque nœud se voit chargé une clé secrète partagés par-paires, chaque clé est unique et partagé uniquement entre deux nœuds.

Par exemple un nœud (x,y) partage une clé K(x,y)(1,y) avec le nœud (1,y) et K(x,y),(2,y) avec le nœud (2,y) et ainsi de suite, donc chaque nœud aura $2(\sqrt{n-1})$ clés et le nombre de clé unique sera de $n(\sqrt{n-1})$. Cette méthode de distribution de clé permet à deux nœuds A et B de trouver un nœud de confiance intermédiaire C qui partage une clé avec A et B au même temps, ce nœud sera utilisé pour la communication.

► **Protocole LEAP**

Le protocole LEAP [42] propose la construction de quatre types de clés :

- *Une clé individuelle* : qui assure la communication entre un nœud et la station de base et qui est installé dans chaque nœud avant le déploiement de ce dernier.

- *Une clé par-paire* : qui assure la communication entre deux nœuds voisins, construite à partir d'une clé maitresse commune à tous les nœuds.
- *Une clé cluster* : qui assure la communication d'un nœud avec l'ensemble de ses voisins.
- *Une clé de groupe* : qui assure la communication entre tous les nœuds du réseau

► **Protocole BROSK**

Le protocole BROSK [43] est un protocole qui utilise un système entièrement ad-hoc pour négocier la clé de session et peut effectuer ce processus de négociation de clé efficacement. En outre l'évolutivité de BROSK est significatif en particulier lorsqu'ils sont appliqués à des réseaux de capteurs à grande échelle.

Soit un nœud de capteur va essayer de négocier une clé de session partagée par la diffusion du message de négociation clé. Chaque nœud tente de diffuser le message suivant : IDA|NA||MACK(IDA|NA). Ici IDA est l'identifiant du nœud A et chaque nœud devra avoir un ID différent. Une fois un nœud reçoit le message de présentation diffusé par son voisin, il peut construire la clé de session partagée par générer le MAC de deux nonces.

► **Protocole STKM**

Le protocole STKM [33] est basé sur l'idée de construire un arbre couvrant de manière sécurisée et conservant l'énergie après un déploiement aléatoire des nœuds ; par la suite, cet arbre sera utilisé pour le renouvellement de clés. La station de base est l'initiatrice de l'algorithme, et chaque nœud tire profit des messages reçus, même si le message n'est pas destiné à ces nœuds, cela permet la réduction de nombre de messages transmis, et par conséquent, minimiser la consommation d'énergie.

Avant le déploiement, la station de base assigne un identificateur unique pour chaque nœud capteur du réseau. Chaque nœud possède une clé partagée avec la station de base pour chiffrer les messages du nœud vers la station de base et une autre clé pour les messages du sens inverse. Ces deux clés sont utilisées pour sécuriser les communications entre les nœuds et la station de base et vice versa.

Après le déploiement, chaque nœud copie sa clé K_r dans sa mémoire volatile (RAM) et la supprime de la mémoire non volatile (EROM). Si un attaquant capture (accès physique) un nœud après le déploiement, il n'aura pas accès à la clé. La station de base diffuse un message dans le but de découvrir ses nœuds voisins. Il utilise un compteur initialisé à zéro et reflète le niveau dans l'arbre (la station de base étant la racine). Le MAC du compteur et de l'identificateur de la source du message est calculé, le tout est chiffré par la clé K_r.

A la fin de la phase précédente, chaque nœud partage une clé symétrique avec la station de base et avec son nœud père, et la clé K_r partagée par tout le réseau. Si un nœud père détecte qu'un de ses fils est malveillant, il ignore ses messages et le supprime de la liste des fils. Dans le cas inverse, où un fils détecte que son nœud père est malveillant et si sa liste de voisins n'est pas vide, il choisit un de ses voisins comme père en lui envoyant un message pour l'informer ; le nœud père confirme au nœud fils par l'envoi d'un autre message et les deux calculent leur clé symétrique partagée.

► **Protocole OTMK**

Le protocole OTMK [44] est une solution de gestion de clé basé sur le concept de la clé initiale transitoire de LEAP. Il permet l'établissement de clés par-paires entre les nœuds voisins.

Ce protocole consiste à préconfigurer chaque nœud du réseau avant la phase de déploiement avec une même clé initiale. Chaque nœud diffuse un message JOIN pour établir des clés par-paires avec ses voisins. Lorsqu'un nœud reçoit le message, il génère un nombre aléatoire et envoie le message REPLY au nœud émetteur qui va le décrypter et vérifier le nonce. Si la vérification est validée, le nœud émetteur enregistre le nœud récepteur comme étant son voisin vérifié puis génère la clé par-paire. Afin de réduire les chances de compromission, chaque nœud détruit la clé initiale après l'établissement de la clé par-paire.

Dans le cas de la compromission de la clé initiale, l'adversaire peut injecter des nœuds malveillants dans le réseau, ce qui lui permet d'intercepter toutes les clés par-paires qui sont en train d'être échangées en observant les messages REPLY. Pour contrer ce problème, plusieurs solutions existent. Par exemple, la station de base utilise l'authentification pour chaque nouveau nœud rejoignant le réseau. Cette approche entraîne beaucoup de trafic de données et de consommation d'énergie pour un réseau de capteurs à grande échelle, mais il est toujours pratique pour un petit réseau.

► **Protocole SNKM**

Le protocole SKNM [45] est un schéma de gestion de clé basé sur la sécurité des nœuds pour les clusters dans les RCSFs. Ce schéma utilise plusieurs genres de clés. Ainsi, les nœuds peuvent choisir différentes clés pour le chiffrage et l'authentification selon les différents types de paquets de données. Ce protocole a été proposé afin d'améliorer le degré de sécurité des clusters Head et réduire l'énergie consommé pour l'établissement des clusters. Selon les différents niveaux de sécurité des nœuds, les auteurs de SNKM adoptent différents schémas de sécurité et différents types de clés. Le cluster Head joue un rôle important dans les RCSFs, ainsi sa sécurité doit être assurée. Si un comportement anormal du cluster Head est détecté, il doit être remplacé immédiatement.

– Les nœuds de sécurité : Selon une fonction aléatoire, le nœud calcule un nombre aléatoire. Si ce nombre est plus grand que T, ce nœud peut être un nœud de sécurité. Dès qu'un nœud détecte un comportement anormal de ses voisins, il envoie un rapport ua nœud de sécurité.

- La clé par-paire : Avant le déploiement, les nœuds ne connaissent pas leurs voisins. Ainsi, lorsqu'un nouveau nœud rejoint le réseau, il essaye de découvrir ses voisins en diffusant un message avec son ID et se met en attente de réponse de ses voisins. Quand un voisin reçoit le message, il lui répond par un ACK et chiffre l'information avec la clé publique. Ensuite dès que le nouveau nœud reçoit l'ACK, il calcule alors la clé par-paire entre eux.
- La clé du cluster : Cette clé est négociée par les nœuds de sécurité. Au début, chaque nœud de sécurité produit une clé aléatoire et l'envoie vers tous les autres nœuds de sécurité avec une estampille de temps. Les nœuds de sécurité comparent les estampilles de temps et prennent la clé aléatoire avec une estampille de temps minimale comme la nouvelle clé du cluster. Le cluster Head récupère cette clé puis la chiffre avec une la clé par-paire pour informer tous les autres nœuds.
- La clé publique : Cette clé est utilisée pour chiffrer les informations de diffusion (Broadcast) et doit être mis à jour régulièrement. Le protocole utilise une fonction aléatoire unidirectionnelle pour générer la chaine de clé d'authentification. En outre, dans ce protocole de gestion des clés, les informations de diffusion ne sont traitées que par les nœuds de sécurité et non par tous les nœuds. Dans certains cas, les nœuds peuvent recevoir les informations de diffusion et ne communiquent qu'avec le cluster Head et les nœuds de sécurité.

► Un plan hybride de gestion de clé basé sur les clusters des réseaux sans fils

Ce protocole [46] est basé sur la construction d'un arbre de clé de dimension d entre le cluster head et la SB, ce schéma considère que le cluster head a une capacité plus élevé en traitement et en stockage d'informations que les nœuds normaux.

Ce schéma se déroule en trois phases :

- **Phase de pré-distribution de clé :** avant le déploiement, chaque nœud charge une clé principale K_m à l'avance et produit aléatoirement son identificateur. Et chaque cluster head sera stockés également une clé privée, une clé de session PK(r)qui est initialement égale au K_m.
- **Phase d'établissement de clé :** cette phase est divisée en deux types :
  - Construire un arbre principal entre cluster head et la station de base : chaque membres de l'arbre stocke une clé privée K_i, SB et le cluster head partagent une fonction PK(r)= E(K_m,R). SB chiffre un nombre aléatoire R produit par elle avec chaque clé privée K_i et l'envoie alors à chaque cluster head. Selon la clé Km, le cluster head produit alors une clé de session en utilisant la fonction précédente

- Etablir la clé de communication entre les membres du même cluster : selon la clé k_m , le nœud génère une clé unique et une paire de clé adjacente basées sur l'emplacement. Comme le nœud u, il aura une clé unique basé sur son emplacement cette clé unique sera : k_u = FKM(ID_u,S_u) avec S_u qui présente les informations de localisations de u. afin de communiquer avec les nœuds voisins, ce message doit être diffusé : S_u, MACk_m(ID_u,S_u) qui donne un accès directe à la clé de session.
- **Phase de maintenance de clé :**dans cette partie l'auteur traite quelques situations comme : l'ajout de nouveau nœud au cluster, la suppression d'un nœud, le remplacement des clusters heads.

Protocoles de gestion de clés probabilistes :
Dans ces schémas les clés utilisées durant le fonctionnement du réseau sont choisie aléatoirement parmi un grand ensemble de clés, ces clés seront chargées dans les capteurs avant le déploiement, ainsi deux nœud voisins en une probabilité d'avoir partagé une clé qui appartient au deux sous-ensembles des voisins.

► **Key-Management Scheme**

Ce protocole probabiliste [47] consiste à configurer chaque nœud avec un ensemble de clés choisie aléatoirement à partir un ensemble de clés plus grand générées par la station de base, on trouve alors que chaque paire de nœuds partage au moins une clé commune avec une certaine probabilité. Cette probabilité est donnée par la formule suivante :

$$P(n) = 1\text{-}\frac{|E|!}{(|E|-1)!} * \frac{1}{|E|^n}$$

Cette formule calcule la probabilité de tirer deux fois le même élément parmi n élément d'un ensemble E. Cette méthode a quelques inconvénients comme la dégradation de la sécurité une fois le nombre de nœuds compromis augmente, aussi il peut y avoir deux nœuds qui ne partagent aucunes clés.

► **Protocole q-Composite**

Le protocole q-Composite [48] est identique à [47] sauf qu'à la place d'une clé partagée exigée pour la communication, chaque paire de nœuds doit partager au moins ($q \geqslant 2$) clés pour pouvoir établir un lien de communication sécurisé. LA nouvelle clé utilisée pour la communication entre ces deux nœuds est le hash de toutes les clés partagées entre eux. Plus le nombre de clés partagées augmente plus la résilience contre la capture du nœud augmente. Autrement,

lorsque le nombre exigé de clés partagées augmente, il devient plus difficile à un attaquant avec un ensemble donné de clés de casser un lien. Cependant, pour préserver une probabilité donnée que deux nœuds partagent des clés suffisantes pour établir un lien sécurisé, il est nécessaire de réduire la taille de l'ensemble des clés. Ceci permet à un attaquant de gagner un plus grand échantillon de l'ensemble de clés en cassant peu de nœuds.

► **Protocole Blom**

Dans le protocole Blom [49], un (k - 1) * n matrice G et (k -1) * (k - 1) matrice carrés D sont construits d'abord, où n est la taille du groupe et k est le seuil attendu. Dans la phase d'initialisation du groupe, chaque membre choisit aléatoirement un vecteur ligne de la matrice A, où A = (G puis T *D) et un vecteur colonne correspond dans la matrice G. Supposons que membre de un correspondant sélectionne la ligne (i) de A et la colonne(i) de G, et b sélectionne la ligne (j) de A et la colonne (j) de G. Une fois a et b veulent communiquer entre eux, ils échangent leurs vecteurs colonnes stockées, puis multiplier leur vecteur ligne stockée avec colonne vecteur du partenaire. Après les calculs, un obtient l'entrée (i, j) de la matrice K (K = GTDG) b obtient le (j, i) entrée de la même matrice. Comme K est une matrice symétrique, les deux entrées ont la même valeur qui peut être travaillé par paires en tant que clé unique entre a et b.

► **Protocole RKP Scheme**

Le protocole RKP scheme [50] sélectionne au hasard un jeu de clés à partir d'un grand nombre de clés et installe les clés dans la mémoire de chaque capteur. Après le déploiement, les capteurs peuvent mettre en place des clés en utilisant les clés préinstallés. Depuis les "schemes", RKP exigent un nombre limité de clés préinstallées dans les capteurs, un capteur peut ne pas partager une clé avec tous ses voisins.

Ce système permet à n'importe quelle paire de nœuds dans un réseau de trouver une paire de clés de manière sécurisée tant que pas plus de λ nœuds sont compromises. Le système est construit sur deux matrices : une matrice connue publiquement G de taille $(\lambda + 1) * N$,une matrice secrète D de taille $(\lambda + 1) * (\lambda + 1)$ créé par le centre de distribution de clés. La matrice A de taille $N * (\lambda + 1)$ est alors créé comme $A = (D * G)^T$. Chaque ligne de A représenté les clés distribuées à un membre du groupe et le numéro de ligne peut servir d'ID d'un capteur. Etant donné que $K = A * G$ est une matrice symétrique, les nœuds i et j peuvent générer une clé partagée (K_{ij} ou K_{ji}) à partir de leurs pré-distribution de clés privées, où k_{ij} est l'élément

dans K située dans la $i^{ième}$ ligne et la $j^{ième}$ colonne.

Ce système possède les propriétés suivantes :

- Une fois que les deux nœuds i et j possèdent les clés préinstallés du même espace de clés $A^{(t)}$, ils peuvent en tirer une clé partagée :Kij(t) = $K_{ji}{}^{(t)}$
- Si les lignes x du même espace de clés $A^{(t)}$ sont pré-distribuées à x capteurs et $x < \lambda$, tout sous-ensemble de capteurs x ne peut pas s'entendre pour établir les secrets dans d'autres capteurs.
- L'identifiant d'un capteur est représenté par le numéro de ligne de la matrice de clé A. Aucun autre capteur peut passer pour ce capteur, depuis la ligne de A est distribué uniquement à ce capteur.

Le protocole RKP possède quelques limitations qui le rendent vulnérable aux attaques. Les capteurs sont des dispositifs à faible coût et fonctionnent dans un environnement sans surveillance pour de nombreuses applications, ils ne peuvent pas être considérés comme inviolable. Comme pour RKP Scheme, d'autres travaux [51] [52] [53] envisagent la connaissance du déploiement de chaque nœud pour améliorer à la fois l'utilisation de la mémoire.

► **Protocole IBPRF**

L'objectif du protocole IBPRF [54] est de concevoir un protocole qui réduit essentiellement la surcharge de communication pour établir des clés par-paires directes entre capteurs lors de la phase d'établissement des clés. Le schéma d'IBPRF se déroule en quatre différentes phases :

- Pré-distribution de clé : pour chaque nœud capteur, le serveur de clés génère aléatoirement une clé maitresse puis sélectionne un ensemble S de nœuds avec des ID générés aléatoirement d'un autre ensemble considérés comme des voisins physiques probable. Pour chaque ID de l'ensemble S, le serveur de clés génère une clé symétrique.
- Etablissement de la clé directe : Après le déploiement, chaque nœud localise tous ses voisins physiques. Si le nœud possède une clé par-paire avec l'un des voisins, il lui envoie un message court contenant son ID. Après la réception du message par un nœud voisin, il calcule une clé symétrique en utilisant l'ID du nœud émetteur puis établit une clé par-paire directe entre eux puis l'utiliser pour leurs futures communications.
- Etablissement d'un chemin : Pour que deux nœuds A et B établissent un chemin entre eux, il faut d'abord que le nœud A trouve un chemin pour atteindre B en passant par des nœuds intermédiaires. Pour cela, le nœud A va produire une clé partagée aléatoire entre lui et le nœud B et le chiffre avec la clé partagée entre A et A1, son nœud voisin, puis la lui envoie. Après réception, le nœud A1 va le déchiffrer avec la clé partagée entre les deux

nœuds A et A1 puis le chiffre avec la clé partagée entre les nœuds A1 et A2, puis la lui envoie, et ainsi de suite jusqu'à ce que le nœud B reçoit la clé.

– Mobilité des nœuds : Si un nœud A ne partage aucune clé avec un nœud voisin B alors il génère une clé par-paire et la chiffre en y ajoutant son ID puis la lui envoie. A la réception du message, le nœud B va la déchiffrer et établir un lien avec le nœud A. La clé par-paire sera utilisée pour leurs futures communications.

2.5 Comparaison entre les protocoles de gestion de clés

2.5.1 Métriques d'évaluation

Pour l'évaluation des solutions de gestion de clés proposées pour les RCSFs, les paramètres suivants sont souvent utilisés dans la littérature :

- **Efficacité :** elle dépend de la quantité de mémoire nécessaire pour enregistrer les clés et le nombre de messages échangés pour la gestion des clés.
- **Scalabilité :** c'est la capacité d'augmenter le nombre de nœuds sans porter atteinte au bon fonctionnement du réseau.
- **Révocation :** cette propriété sert à savoir si le RCSF a la possibilité du retrait d'un nœud en panne ou quand son opération n'est pas correcte.
- **Résilience :** c'est la résistance d'un protocole à la compromission d'un nœud par un adversaire.
- **Connectivité :** c'est la probabilité que deux nœuds en communication (nœuds voisins) partagent au moins une clé.
- **Mobilité :** si le protocole la supporte ou non.
- **Topologie de réseau :** c'est le type de topologie supportée par le protocole.

2.5.2 Comparaison entre les protocoles de gestion de clés

Dans cette partie, nous allons faire une comparaison entre les différents protocoles de gestion de clés étudiés dans ce chapitre, on se basant sur les métriques d'évaluations citées auparavant.

	Efficacité		Scalabilité	Révocation	Résilience	Connectivité	Mobilité	Topologie du réseau
	Complexité en mémoire	Complexité en communication						
TinyPK / TinyECC	Dépend du nombre de voisins à un saut (d)	Pour chaque nœud : 2 x d	Bonne	Faible	Faible	100%	OUI	/
LEAP	(3 x d) + 2 + la chaine de clés pour uTESLA	(2 x d) + 1	Bonne	Bien	Très bien avant T_{min}	100%	NON	Plate
OMTK	d + 1	d + 1	Bonne	Bien	Parfaite avant et après T_{min}	100%	NON	Plate
Q-Composite	2*m	d + 1	Moyenne	Très bien	Dépend de m	p' < p	OUI	Plate
Blom	2(λ+1)	d + 1	Moyenne	Bien	λ-secure	100%	OUI	Plate
Blundo	t + 1	d + 1	Excellente	Bien	t-secure	100%	OUI	Plate
SNKM	d - 1	d + 1	Excellente	Bien	t-secure	100%	OUI	Plate
Plan hybride de gestion de clés base le cluster des RCSFs	Un peu complexe	Relativement petite	Bien	Très bien	Dépend de m	100%	Oui	Hiérarchique
STKM	3+nombres de fils	d+1	Bien	Bien	Très bien	100%	Oui	Plate
IBPRF	m+1	<d+1	Bien	Bien	Parfaite	Dépend de m	Oui	Hiérarchique

m : La taille du réseau. d : nombre de voisins à un saut.

FIGURE 2.2 – Comparaison entre les protocoles de gestion de clés

Les protocoles de gestion de clés déterministes qui sont basés sur une clé initiale et qui achèvent une connectivité totale , sont moins coûteux en espace mémoire utilisée. Les protocoles de gestion de clés probabilistes qui pré-charge les nœuds avec un trousseau de clé, sont plus coûteux en terme d'espace mémoire. La connectivité dans ces derniers est variante et dépend de la taille du trousseau de clés pré-chargées dans les nœuds capteurs.

Les protocoles basés sur une topologie hiérarchique, présentent une bonne résilience car ils centralisent la mission d'établissement de clés au niveau des chefs de groupes, qui sont supposés être sécurisés. Les protocoles basés sur une topologie plate, présentent une résilience parfaite si la compromission des nœuds aura lieu après la phase d'établissement de clés. Les protocoles probabilistes essayent d'atteindre une meilleure résilience par la combinaison d'autres mécanismes au protocole de base. Par contre, les protocoles de gestion de clés sans pré-distribution sont faibles en termes de résilience contre la capture des nœuds.

2.6 Conclusion

Dans ce chapitre, nous avons étudié quelques protocoles et solutions de gestion de clés proposées pour les RCSFs. Nous avons pu constaté que les protocoles basés sur la méthode de pré-distribution sont les plus appropriés aux RCSFs pour leurs faibles coûts.

Il est à noter que la gestion des clés est l'un des secteurs les plus importants dans la sécurité des RCSF, ce qui a amené à effectuer beaucoup de travaux afin d'avoir un schéma performant qui assure un niveau élevé de sécurité, optimise les métriques de performances et conserve l'énergie.

CHAPITRE 3

STÉGANOGRAPHIE DANS LES RÉSEAUX DE CAPTEURS SANS FIL

3.1 Introduction

La sécurité est un enjeu majeur des technologies numériques modernes. Infrastructure de télécommunication (GSM, GPRS, UMTS), réseau sans fils (bluetooth, WiFi, WiMax), Internet, systèmes d'informations, routeurs, systèmes d'exploitation, applications informatiques, toutes ces entités présentent des vulnérabilités : faille de sécurité, défaut de conception ou de configuration. Ces systèmes tombent en panne, subissent des erreurs d'utilisation et sont attaqués de l'extérieur ou de l'intérieur par des pirates, des cybercriminels. [55]

Dans les réseaux de capteurs il existe beaucoup de solutions de sécurité [56] qui sont les solutions basées sur les approches algorithmiques comme la découverte de voisinage sécurisée, et les solutions basées sur les approches cryptographiques qui consistent à utiliser la cryptographie symétrique ou asymétrique dans les réseaux de capteurs.

Néanmoins, l'utilisation de ces approches dans certains cas n'est pas toujours efficace car dans les réseaux de capteurs, la comparaison de puissance d'action est souvent en faveur de l'attaquant. Par-là, nous entendons qu'un attaquant a des moyens de calculs (ordinateurs, grappes de calcul, etc.) nettement supérieur aux réseaux de capteurs, qui peuvent lui permettre de déchiffrer le message codé. Donc si un attaquant intercepte un message chiffré il va savoir que c'est un message important et va essayer par tous les moyens qui sont à sa disposition de le déchiffrer, donc dans ce cas il est préférable de cacher l'existence d'un tel message en utilisant des nouvelles solutions comme la stéganographie.

3.2 La stéganographie

La stéganographie est l'art de dissimuler des données dans d'autres données. Son objectif est de faire passer inaperçu un message dans un autre message, de cacher l'existence d'un tel message contrairement à la cryptographie où le message chiffré peut circuler librement et vu par d'autres personnes sans jamais pouvoir le déchiffrer et le lire car son intégrité est liée au secret de la clé de chiffrement uniquement connu par les personnes qui sont censées accéder au contenu du message.

Cependant dans la stéganographie, on cherche à cacher l'existence d'un message, uniquement les personnes auxquelles le message est destiné connaissent son existence. Pour pouvoir faire cela, on prend un message secret (secret message), on le cache au sein d'un objet de couverture (cover object), on aura alors un objet stego (stego object) qui devra être d'apparence similaire a l'objet de couverture mais qui contient le message secret.
La figure 3.1[57] présente une description du mécanisme de la stéganographie :

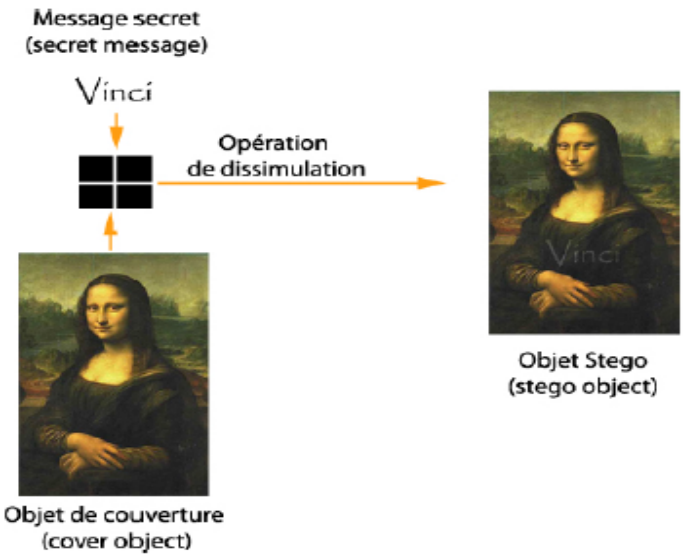

FIGURE 3.1 – Description du mécanisme de la stéganographie

La stéganographie existe depuis longtemps, bien avant l'invention de l'ordinateur. En effet dans la Grèce antique, l'historien grec Hérodote cite l'utilisation de la stéganographie par les soldats grecs pour contrer l'invasion perse[58]. En chine, on écrivait un message sur de la soie recouverte ensuite par de la cire pour former un boule qui sera avalé par le messager. Vers le 1^{er} et 2^{eme} siècle, l'encre invisible fut réalisée en écrivant avec du lait ou du jus de citron sur

une feuille, l'exposition de cette dernière sous une source chaude (fer à repasser chaud, flamme de bougie...) révèle le message. Durant la seconde guerre mondiale, la stéganographie fut utilisé fréquemment par les agents allemands durant leurs opérations.

La stéganographie a su évoluer avec le temps. Aujourd'hui son principal champ d'application se retrouve dans le domaine du numérique. Ainsi la stéganographie utilise la plupart de nos médias numériques actuels tels que les images numériques, le son numérique, les protocoles de communication ou bien encore la programmation informatique pour dissimuler des informations.

Dans la programmation informatique on trouve trois types de stéganographie : [57]

- **Stéganographie pure :** c'est le concept général de la stéganographie, ou le secret ne correspond qu'a la méthode utilisée pour la dissimulation du message.
- **Stéganographie à clé secrète :** elle correspond à combiner la méthode de la stéganographie pour la dissimulation et au cryptage de ce message avant cette dissimulation.
- **Stéganographie à clé publique :** La stéganographie à clé publique reprend la définition de la stéganographie à clé symétrique hormis le fait que le chiffrement ne se fait plus à l'aide d'une clé partagée mais grâce à l'utilisation de clé publique et privée.

L'utilisation de l'image comme objet de couverture est le plus répandu au cours de ces dernières années, cela est dû aux possibilités de dissimulation qu'offrent ces images, il existe beaucoup de techniques de dissimulation dans les images comme la méthode LSB[59] qui consiste à dissimuler le message secret dans les bits du poids faible de l'image, c'est-à-dire, modifier les bits les moins importants dans l'image donc l'image ne parait pas modifié à l'œil nue.

La figure 3.2 présente le codage d'un pixel et la couleur correspondante à ce pixel :

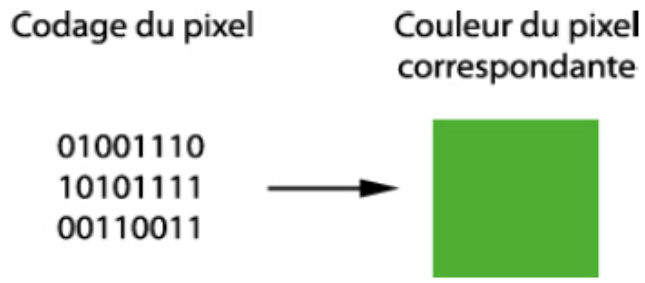

FIGURE 3.2 – Codage d'un pixel et couleur correspondante [57]

Dans la figure 3.3 qui suit, on va utiliser la technique de dissimulation LSB sur le codage précédent :

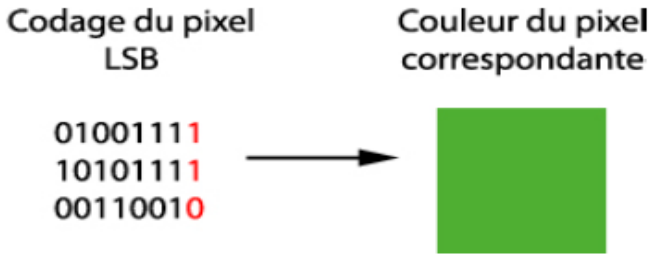

FIGURE 3.3 – Dissimulation LSB dans un pixel [57]

Une autre technique de stéganographie répandue dans les images numériques consiste à utiliser les propriétés des images PNG et GIF qui compressent la taille d'une image avec l'utilisation d'un panel de couleurs. Le panel de couleurs dans ces images numériques est en quelque sorte le catalogue de toutes les couleurs qui se retrouvent dans l'image. La dissimulation alors dans ce cas consiste à modifier l'ordre des images selon leur code correspondant.

3.3 La stéganographie dans les RCSFs

Il existe peu de travaux qui sont attachés à la dissimulation des messages lors des communications dans les réseaux de capteurs sans fils. En effet, dans [60] [61], les auteurs ont pour objectif de créer un canal secret stéganographique au sein de la couche PHY du protocole de communication IEEE.802.15.4 en utilisant la correspondance (symbole de 4 bits - séquence de 32 bits) traduite par la puce de transmission des données illustré dans la figure 3.4.

4-bit symbol	32-chip chip sequence	4-bit symbol	32-chip chip sequence
0x0	0x744AC39B	0x8	0xDEE06931
0x1	0x44AC39B7	0x9	0xEE06931D
0x2	0x4AC39B74	0xA	0xE06931DE
0x3	0xAC39B744	0xB	0x06931DEE
0x4	0xC39B744A	0xC	0x6931DEE0
0x5	0x39B744AC	0xD	0x931DEE06
0x6	0x9B744AC3	0xE	0x31DEE069
0x7	0xB744AC39	0xF	0x1DEE0693

FIGURE 3.4 – Correspondance symbole de 4 bits - séquence de 32 bits puce [61]

L'un des travaux les plus intéressants et qu'on va utiliser dans ce document est le travail de D. Martins dans [6], qui a prouvé qu'il existe dans les couches PHY et MAC du protocole IEEE 802.15.4 plusieurs possibilités de dissimulation d'informations, cela constitue un bon environnement exploitable pour une communication stéganographique à l'aide de capteurs.

Dans ce qui suit on présentera les travaux proposé par [6].

3.3.1 la norme IEEE 802.15.4

Le protocole IEEE 802.15.4 est le plus utilisé pour la communication dans les réseaux de capteurs car il est capable de transférer de petits fichiers sur de grands réseaux. La communication de ce protocole est peu coûteuse en énergie et l'accès au canal de communication est rapide.
Ce protocole utilise les couches physiques (PHY) et d'accès au médium (MAC) qui est utilisé par le protocole ZigBee[62] comme représentée par la figure 3.5 :

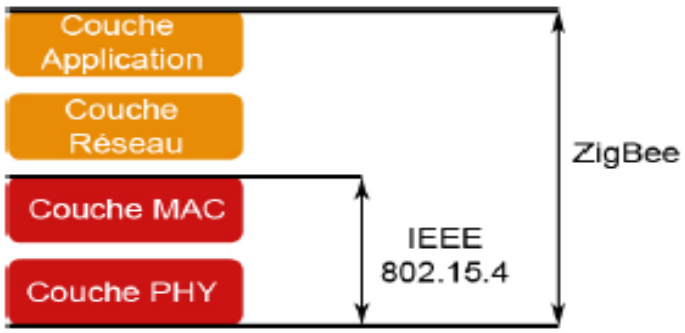

FIGURE 3.5 – Représentation de la pile du protocole IEEE 802.15.4/ZigBee

Ce protocole propose trois topologies :

- **Réseau en étoile (star network) :** ce type de topologie supporte un nombre limité de nœuds. Ce réseau est composé de plusieurs capteurs et/ou routeurs (esclave) reliés à un coordinateur désigné comme nœud central (maitre). Toutes les communications entre les esclaves passent obligatoirement par le maitre

- **Réseau maillé (mesh network) :** tous les nœuds peuvent communiquer entre elles si elles se trouvent dans la même zone de couverture. Un nœud est un capteur s'il possède

au plus un voisin. Dans le cas contraire, ce nœud peut être un capteur ou un routeur. Il est nécessaire qu'il ait un coordinateur dans le réseau malgré l'aspect pair à pair de cette topologie.

- **Réseau en arbre clustérisé (clustered tree network) :** Le nœud coordinateur est le nœud central qui constitue la racine de l'arbre. Il est relié aux nœuds routeurs de chaque cluster de nœuds où sont contenues les feuilles de l'arbre à savoir des nœuds capteurs et/ou routeurs. On parle aussi de topologie hiérarchique puisque le coordinateur est le supérieur hiérarchique des nœuds routeurs auxquels il est relié, qui sont à leur tour les supérieurs hiérarchiques des autres nœuds situés dans les sous-réseaux de l'arbre.

Le protocole IEEE 802.15.4 offre deux modes d'accès au médium (MAC) :

- **Mode non beacon :** l'accès au médium se fait via CSMA-CA. Le coordinateur du réseau reste par défaut en attente de données. Un nœud doit vérifier si le canal est libre avant d'envoyer des données. Dans le cas contraire, il doit attendre une période aléatoire avant de vérifier à nouveau, et ainsi de suite.

- **Mode beacon :** l'accès au médium se fait aussi via CSMA-CA. Dans ce mode, le coordinateur observe des périodes de veille pour préserver son énergie. Pour la communication, il envoie des trames beacon pour signaler le début d'une période pendant laquelle les capteurs peuvent accéder au canal pour communiquer avec le coordinateur.

3.3.2 Dissimulation dans les couches PHY et MAC

3.3.2.1 Dissimulation dans La couche PHY

La couche PHY du protocole IEEE 802.15.4 contrôle l'activation et la désactivation de la transmission radio, la commutation des canaux de transmission et l'évaluation de la qualité du signal. La couche PHY envoie une trame PHY dont le champ PHY Service Data Unit contient la trame MAC.

La structure de la trame PHY est représentée dans la figure 3.6 :

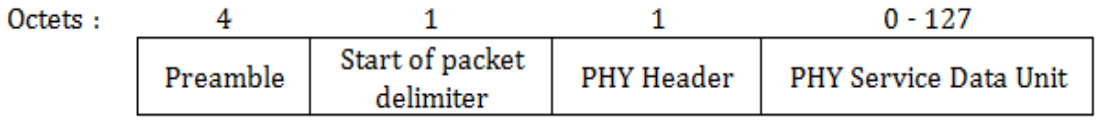

FIGURE 3.6 – Structure d'une trame de la couche PHY du protocole IEEE 802.15.4

Le champ PHY Header qui est codé sur 1 octet (8 bits) détermine la longueur du champ PHY Service Data Unit qui contient les données à transmettre. Ce champ PHY Header n'utilise que 7 bits, le $8^{ème}$, étant réservé, peut être utilisée pour y dissimuler de l'information.

La capacité de dissimulation à l'aide de cette technique correspond à 1 bit pour une trame PHY comprise entre 9 et 133 bits. L'intérêt de ce bit parait limité mais permet tout de même de transmettre de très petites informations.

3.3.2.2 Dissimulation dans La couche MAC

Les possibilités de dissimulation de données dans la couche MAC sont beaucoup plus nombreuses que dans la couche PHY. La couche MAC utilise 4 types de trames :

► La trame de données

La trame de données permet la communication des données. La structure générale d'une trame de données de la couche MAC du protocole IEEE 802.15.4 est représentée dans la figure 3.7 :

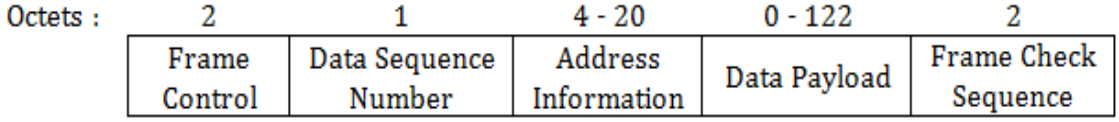

FIGURE 3.7 – Structure d'une trame de données de la couche MAC du protocole IEEE 802.15.4

Dans cette trame, des informations peuvent être cachées dans les champs Frame Control, Data Sequence Number et Address Information.

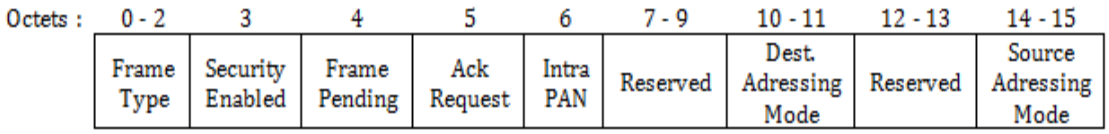

FIGURE 3.8 – Structure du champ Frame Control

- Le champ Frame Control de taille de 2 octets (16 bits) possède 5 bits réservés qui ne sont pas utilisés. Il est donc possible d'utiliser ces 5 bits de ce champ pour y cacher de l'information.
- Le champ Data Sequence Number contient un nombre codé sur un octet, utilisé pour les paquets d'acquittement afin de spécifier quel numéro de paquet a été acquitté. Cette valeur est initialisée aléatoirement pour chaque communication entre capteurs et incrémentée après chaque paquet envoyé. Il est possible de définir volontairement un numéro pour l'utiliser comme objet de couverture stéganographique, donc y cacher de l'information.
- Le champ Address Information, qui varie entre 4 et 20 octets, permet de déterminer le coordinateur source et de destination, et les adresses de l'expéditeur et du destinataire du paquet.

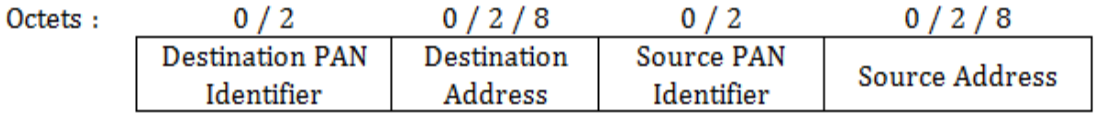

FIGURE 3.9 – Structure du champ Address Information

Il est possible de dissimuler des informations dans le champ Source Address, le destinataire du message ne connaitra pas l'expéditeur mais peut tout de même traiter l'information. Dans certains cas, le destinataire n'a pas besoin de connaitre l'expéditeur pour traiter les informations reçus.

De plus, le protocole IEEE 802.15.4 permet deux types d'adressage, un adressage court sur 2 octets et un adressage étendu sur 8 octets qui sont utilisés en fonction de la taille du réseau. Il est donc possible d'utiliser les bits non utilisés pour dissimuler de l'information sans masquer l'identité de l'expéditeur.

En définitive une trame de données de la couche MAC du protocole IEEE 802.15.4 peut être utilisée comme objet de couverture pour dissimuler jusqu'à 77 bits en utilisant les 3 méthodes décrites précédemment.

► **La trame beacon**

Les trames beacon sont utilisées uniquement pour les réseaux dont l'accès au canal se fait en mode beacon. La structure générale d'une trame beacon de la couche MAC du protocole IEEE 802.15.4 est représentée par la figure 3.10 :

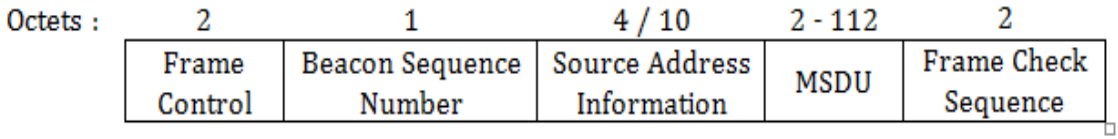

FIGURE 3.10 – Structure d'une trame beacon de la couche MAC du protocole IEEE 802.15.4

Le champ Frame Control, ainsi que le champ Source Address contenu dans le champ Source Address Information offrent les mêmes possibilités de cacher des données comme expliqué pour la trame de données.

Le champ Data Sequence Number d'une trame de données est remplacé par un champ Beacon Sequence Number de taille identique qui a pour mission de déterminer le numéro Beacon de la trame pour une communication du protocole IEEE 802.15.4. Ce champ possède le même fonctionnement que le champ Data Sequence Number.

Au final, la capacité de dissimuler de l'information au sein d'une trame beacon est équivalente à celle d'une trame de donnée pour la couche MAC du protocole 802.15.4, soit au total 29 et 77 bits selon que l'adressage soit court ou étendu.

► **La trame d'acquittement**

Les trames acquittement de la couche MAC servent à acquitter les trames de données et les trames de commande reçues. Elles sont envoyées immédiatement après réception de ces trames. La structure générale d'une trame d'acquittement de la couche MAC du protocole IEEE 802.15.4 est représentée par la figure 3.11 :

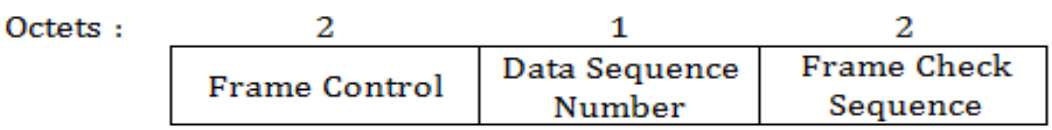

FIGURE 3.11 – Structure d'une trame d'acq de la couche MAC du protocole IEEE 802.15.4

Dans une trame d'acquittement, les champs Frame Control et Data Sequence Number sont identiques aux champs de même nom de la trame de données. Le champ Frame Control offre la même possibilité de dissimulation que celui d'une trame de donnée, mais le champ Data Sequence Number doit contenir le numéro de la trame qu'il acquitte et non une variable aléatoire. L'utilisation de ce champ est donc difficilement exploitable.
Au final, une trame d'acquittement de la couche MAC du protocole IEEE 802.15.4 peut donc accueillir jusqu'à 5 bits dissimulés.

▶ La trame de commande

Les trames de commandes de la couche MAC du protocole 802.15.4 transmettent des requêtes via des numéros de commande. Ces commandes servent entre autres à demander à intégrer le réseau, à s'en dissocier, à demander des données au coordinateur ou à réserver une période GTS (Garanteed Time Slot) pour l'envoi des données. La structure générale d'une trame de commande de la couche MAC du protocole IEEE 802.15.4 est représentée par la figure 3.12 :

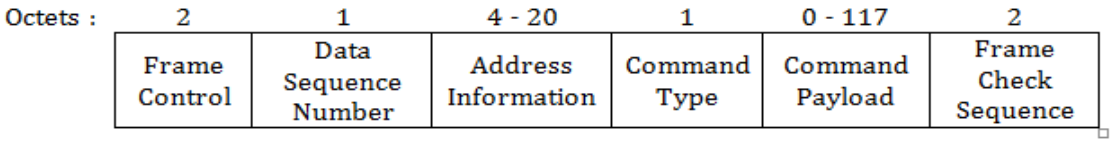

FIGURE 3.12 – Structure d'une trame de contrôle de la couche MAC du protocole IEEE 802.15.4

Les champs Frame Control, Data Sequence Number et Address Information d'une trame de contrôle sont identiques en tout point aux champs de même nom d'une trame de donnée de la couche MAC et possèdent aussi la même utilité. Les possibilités d'y dissimuler des informations sont donc identiques à celles trouvées dans la trame de données et correspondent au total à 77 bits.

▶ Autres possibilités de la couche MAC

Il existe d'autres moyens à part les trames de la couche MAC de dissimuler des informations, parmi ces possibilités [57] il y a :

– **Taille de la trame :** comme la taille d'une trame est variable, on peut jouer sur cette dernière pour dissimuler des informations.

- **Ordre des trames :** l'ordre d'envoi des trames peut servir à envoyer des messages stéganographique. Par exemple chacune des 4 trames correspond à un séquençage de 2 bits (trame de donnée : 00, trame beacon : 01, trame d'acquittement : 10, trame de contrôle : 11) et on veut envoyer ce message : 00111001 donc on aura le séquencement suivant : trame de donnée -> trame de contrôle -> trame d'acquittement -> trame beacon.
- **Le délai d'envoi :** le délai d'envoi entre chaque trame peut permettre de dissimuler un message stéganographique. Deux capteurs s'envoient des messages avec un temps d'attente Δ qui va correspondre au code binaire 00. Ainsi, 2Δ va correspondre au code binaire 01, 3Δ à 10 et 4Δ à 11.
- **Création de collision :** lors d'une communication entre deux nœuds A et B, un nœud C provoque volontairement une collision pour délivrer un message d'une manière stéganographique à A.

3.4 Conclusion

La conception de réseaux de capteurs sans fil est un domaine de recherche très actif. Ces capteurs sont utilisées dans des applications qui ont souvent besoin d'un niveau de sécurité élevé car ils fournissent des services essentiels, voire vitaux.

En l'absence de mécanismes de sécurité appropriés, le déploiement de ces réseaux de capteurs demeure vulnérable à de nombreuses attaques.

Dans ce chapitre, nous avons abordé l'utilisation de la stéganographie dans les RCSFs qui constitue un moyen efficace pour sécuriser les messages échangés et les préserver des attaques de personnes malveillantes. L'exploitation des trames de données permet une meilleure sécurisation et permet une minimisation de la consommation d'énergie.

Dans le chapitre suivant nous allons utiliser cette technique lors de la proposition de notre protocole.

CHAPITRE 4

PROPOSITION ET SIMULATION

4.1 Introduction

Le problème majeur dans la communication dans les réseaux de capteurs sans fils, c'est de trouver une manière d'établir une communication sécurisée et peu coûteuse. Cela consiste à élaborer des clés cryptographiques entre les nœuds du réseau en introduisant des algorithmes de gestion de clés d'une part et la construction des routes d'acheminement en utilisant des protocoles de routage d'autre part.

L'utilisation de ses méthodes de manière séquentielle induit à un nombre important de messages ce qui entraine une consommation coûteuse en terme d 'énergie, ce qui justifie l'utilisation de la notion de la stéganographie. En dissimulant les messages utilisés dans les protocoles de gestion de clés dans ceux du protocole de routage en utilisant les possibilités proposées par David Martins.

Dans ce chapitre, nous proposons un protocole basé sur la stéganographie en combinant un protocole de gestion de clés que nous avons proposé et qui utilise la pré-distribution et un protocole de routage hiérarchique.

4.2 Protocole proposé

Notre protocole KMPS (Key Management Protocol with Steganography) est un protocole de gestion de clés basé principalement sur l'exploitation du protocole de routage CELL-LEACH pour effectuer la gestion de clé du réseau, c'est-à-dire, la sécurisation du réseau par des clés

cryptographiques pour permettre une bonne sécurité, en effet on va exploiter les messages générés par le protocole CELL-LEACH qui ont pour but d'organiser le réseau pour mettre en place les clés du réseaux c'est-à-dire dissimuler les messages générés par le protocole de gestion de clé à l'intérieur des messages du protocole CELL-LEACH. L'utilisation de la stéganographie pour cette dissimulation a pour but de diminuer le nombre de messages échangés entre les capteurs du réseau.

KMPS est un protocole hybride qui utilise à la fois un chiffrement symétrique utilisés par les nœuds et asymétrique propre à la station de base.

KMPS utilise cinq types de clés :

- une clé commune partagée entre un cell-head et ses nœuds et un cluster-head et ses cell-head.
- une clé initiale partagée avec tous les nœuds du réseau, utilisée pour l'établissement de la clé commune.
- une clé secrète unique pour chaque nœud, utilisée par la station de base pour communiquer avec lui, utilisée aussi pour l'établissement de la clé commune.
- la clé publique de la station de base partagée avec tous les nœuds, utilisée pour la mise à jour de la nouvelle clé secrète.
- la clé privée de la station de base réservée, utilisée pour déchiffrer les messages envoyés par les nœuds pour les informer du changement de leurs clés secrètes.

4.3 Hypothèse

Nous définissons les hypothèses suivantes :

- Les clusters et les cellules ne changent pas durant toute la durée de vie du réseau, seuls les clusters-heads et les cells-heads changent de manière dynamique.
- Les nœuds capteurs sont homogènes (c'est-à-dire les nœuds sont similaires dans leurs capacité de traitement, d'énergie et de stockage).
- Le déploiement est aléatoire.
- La compromission d'un nœud signifie que les informations stockées dans sa mémoire sont connues par l'attaquant.
- La mort de chaque capteur n'est causée que par l'épuisement de son énergie.
- La station de base n'a pas de contrainte sur la capacité de stockage et de calcul et ne peut pas être compromise.
- Un attaquant ne pourra pas compromettre un nœud avant que la phase de pré-distribution

des clés soit terminée.

- Il n'existe pas de nœud isolé (c'est-à-dire que tous les nœuds se trouvent dans des clusters et cellules).

4.4 Notation

Les notations suivantes vont être utilisées lors de la présentation du protocole KMPS :

Notation	**Déscription**
n	Nombre de nœuds
A	Nœud du réseau
CH	Cluster-head
CE	Cell-head
ID_A	Identifiant d'un nœud
ID_{CH}	Identifiant d'un cluster-head
ID_{CE}	Identifiant d'un cell-head
K_A	Clé secrète d'un nœud
K_{CH}	Clé secrète d'un cluster-head
K_{CE}	Clé secrète d'un cell-head
K_{IN}	Clé initiale
$K_{CH,CE}$	Clé commune entre un cluster-head et un cell-head
$K_{CE,A}$	Clé commune entre un cell-head et un nœud
K_{SB}	Clé privée de la station de base
K_P	Clé publique de la station de base
H	Fonction de hachage
nonce	Nombre aléatoire
T_{min}	Temps nécessaire pour l'établissement des clés communes

TABLE 4.1 – Description des notations utilisé dans la proposition

4.5 Fonctionnement du réseau

Le protocole KMPS se compose de six phases importantes pour le bon fonctionnement du réseau :

4.5.1 Phase de pré-distribution des clés

La station de base attribue à chaque nœud du réseau un identifiant unique, une clé secrète, une clé initiale identique pour tous les nœuds ainsi que la clé publique de la station de base.

La station de base possède une clé privée K_{SB}, utilisée pour déchiffrer les messages envoyés par les nœuds pour l'informer du changement de leurs clés secrètes.

4.5.2 Phase de hiérarchisation et mise en place des clés communes

Après le déploiement des nœuds, la station de base divise le réseau en plusieurs zones. Pour chaque zone, elle choisit aléatoirement un nœud qui deviendra un cluster-head et lui envoie un message SB-JOIN avec l'identifiant du nœud choisi.
Lorsqu'un nœud capte le message SB-JOIN, si son ID est mentionné, alors il devient un cluster-head, sinon il l'ignore.

Ensuite, chaque cluster-head va découvrir les nœuds qui formeront le cluster en envoyant un message CL-TEAM. Tous les nœuds qui vont recevoir ce message et qui ne font pas encore parti d'un cluster vont lui répondre par un message CL-TEAM-ACK.

Après, chaque cluster-head va sélectionner sept nœuds aléatoirement qui vont devenir des cell-heads. Pour cela, chque cluster-head va envoyer un message CL-JOIN contenant l'identifiant du cluster-head ID_{CH}, les identifiants des cell-heads sélectionnées ID_{CE1}, ID_{CE2},..., ID_{CE7}, à l'intérieur de ce message sera caché d'une manière stéganographique un autre message qui contiendra la clé secrète du cluster-head K_{CH}, un nonce généré aléatoirement par ce dernier, le tout chiffré avec la clé initiale K_{IN}, puis se met en attente de réponses.

$$CH \text{ -> } CE : ID_{CH} + (ID_{CE1}, ID_{CE2},\ldots, ID_{CE7}) + (K_{CH}, \text{nonce})K_{IN}$$

Lorsqu'un nœud intercepte le message CL-JOIN, il vérifie si son identifiant est mentionné dans ce message. S'il ne le trouve pas, il ignore le message. Sinon, il lui répond par un message CL-JOIN-ACK contenant son identifiant ID_{CE}, à l'intérieur de ce message sera caché d'une manière stéganographique un autre message qui contiendra la clé secrète du cell-head K_{CE}, le nonce envoyé par le cluster-head, le tout chiffré avec la clé secrète du cluster-head K_{CH}.

$$CE \text{ -> } CH : ID_{CE} + (K_{CE}, \text{nonce})K_{CH}$$

Enfin, chaque cluster-head et ses cell-heads vont calculer leurs clés communes chacun de son coté en utilisant le nonce partagé comme suit :

$$K_{CH,CE} = (ID_{CE}, \text{nonce})K_{CE}.$$

Les cell-heads ayant été définis, chacun va découvrir les nœuds qui formeront les cellules en diffusant un message CE-JOIN contenant l'identifiant du cell-head ID_{CE}, à l'intérieur de ce message sera caché d'une manière stéganographique un autre message qui va contenir sa clé secrète K_{CE}, un nonce généré aléatoirement par ce dernier, le tout chiffré avec la clé initiale K_{IN} et se met en attente de réponses.

$$CE \text{ -> } A : ID_{CE} + (K_{CE}, \text{nonce})K_{IN}.$$

Lorsqu'un nœud A intercepte le message CE-JOIN, s'il appartient déjà à une autre cellule, il ignore le message. Sinon, il lui répond donc avec un message CE-JOIN-ACK pour l'informer qu'il accepte de faire partie de sa cellule. Ce message va contenir son identifiant ID_A, à l'intérieur de ce message sera caché d'une manière stéganographique qui contiendra sa clé secrète K_A et le nonce envoyé par le cell-head, le tout chiffré par K_{CE}. Il envoie aussi le haché des deux identifiants ID_{CE} et ID_A chiffrés avec la clé initiale K_{IN} pour assurer l'authentification.

$$A \text{ -> } CE : ID_A + (K_A, \text{nonce})K_{CE} + H((ID_A, ID_{CE})K_{IN}).$$

Lorsque le cell-head reçoit le message CE-JOIN-ACK du nœud A, il récupère l'identifiant ID_A et exécute la fonction de hachage H sur les deux identifiants ID_{CE} et ID_A pour authentifier le nœud A. De cette manière, le nonce est partagé entre les deux entités d'une manière plus sûre.

Enfin, chaque entité va calculer la clé commune de son coté, cette clé est calculée en utilisant le nonce partagé comme suit :

$$K_{CE,A} = (ID_A, \text{nonce})K_A$$

L'algorithme de génération de clé est détaillé comme suit :

Algorithm 1 Génration de clé commune entre CH et CE

POUR CHAQUE CLUSTER-HEAD CH

Début

Diffuser message CL-JOIN : ID_{CH}+ID_{CE1}+ID_{CE2}+...,ID_{CE7} avec (K_{CH},nonce)K_{IN} caché ;

Attente réponse ;

if nœud A reçoit CL-JOIN **then**

if $ID_A \epsilon$ CL-JOIN **then**

obtenir ID_{CH},K_{CH} et nonce ;

envoyer message CL-JOIN-ACK : ID_{CE} avec (K_{CE},nonce)K_{CH} caché ;

générer clé commune $K_{CH,CE}$:(ID_{CE},nonce)K_{CE}

end if

end if

if CH reçoit CL-JOIN-ACK **then**

obtenir ID_{CE},(K_{CE} et nonce ;

générer clé commune $K_{CH,CE}$:(ID_{CE},nonce)K_{CE}

end if

Après la mise en place des clés communes entre les nœuds du réseau, chaque nœud va supprimer les clés secrètes qu'ils possèdent pour éviter qu'elles soient interceptées par un attaquant. Ensuite, chaque nœud renouvèle sa propre clé secrète en envoyant un message à la station de base contenant son identifiant ID_A et sa nouvelle clé secrète K_{A+} chiffrée avec la clé publique K_P. Ce message sera déchiffré par la station de base avec sa clé privée K_{SB}.

$$A \text{ -> } SB : ID_A + (K_{A+})K_P.$$

4.5.3 Phase de communication

Cette phase permet l'établissement d'une communication soit entre les nœuds d'une cellule avec le cell-head, soit entre les cell-heads d'un cluster avec le cluster-head ou encore entre les clusters-heads avec la station de base.

Il existe deux types de communication :

- Les nœuds d'une cellule communiquent avec le cell-head en utilisant la clé commune partagée entre eux et c'est la même méthode qu'utilisent les cell-heads d'un cluster pour communiquer avec le cluster-head.
- La station de base communique avec les clusters-heads en utilisant leurs clés secrètes.

Algorithm 2 Génration de clé commune entre CE et les nœuds

POUR CHAQUE CELL-HEAD CE

Début

Diffuser message CE-JOIN : ID_{CE} avec (K_{CE},nonce)K_{IN} caché ;

Attente réponse ;

if A est libre **then**

obtenir ID_{CE},K_{CE} et nonce ;

envoyer message CL-JOIN-ACK : ID_A avec H((ID_{CE},ID_{CH})K_{IN}) (K_{CE},nonce)K_{CH} caché ;

générer clé commune $K_{CE,A}$:(ID_A,nonce)K_A

end if

if CE reçoit CE-JOIN-ACK **then**

obtenir ID_A,(K_A,H() et nonce ;

générer clé commune $K_{CE,A}$:(ID_A,nonce)K_A

end if

4.5.4 Phase de mise à jour

La phase de mise à jour permet de se fait à chaque nouveau tour. Tous les nœuds vont supprimer les clés communes déjà établies. La phase de mise à jour se fait en trois étapes :

4.5.4.1 Mise à jour des clusters-heads et cell-heads

Pour la sélection des nouveaux clusters-heads et cell-heads, on utilise la méthode proposée par le protocole CELL-LEACH.

4.5.4.2 Mise à jour de la clé initiale

La station de base met à jour la clé initiale K_{IN} en diffusant un message SB-UPDATE avec un autre message caché de manière stéganographique qui va contenir la nouvelle clé initiale K_{IN+} chiffrée avec la clé secrète de chaque nœud.

$$SB \text{ -> } A : (K_{IN+})K_A.$$

4.5.4.3 Mise à jour des clés communes

Chaque nouveau cluster-head va envoyer un message CL-UPDATE contenant son identifiant ID_{CH} avec un autre message caché d'une manière stéganographique contenant sa clé secrète K_{CH} et un nonce généré aléatoirement, chiffrés avec la clé initiale K_{IN} puis se met en attente de réponses.

$$CH \text{ -> } A : ID_{CH} + (K_{CH}, nonce)K_{IN}.$$

Lorsqu'un nœud intercepte le message CL-UPDATE, si le nœud n'est pas un cell-head alors il ignore le message. Sinon, il répond par un message CL-UPDATE-ACK contenant son identifiant ID_{CE} avec un autre message caché d'une manière stéganographique contenant sa clé secrète K_{CE} et le nonce envoyé par le cluster-head, chiffrés avec la clé secrète K_{CH}. Il envoie aussi le haché des deux identifiants ID_{CH} et ID_{CE} chiffré avec la clé initiale K_{IN} pour assurer l'authentification.

$$CE \text{ -> } CH : ID_{CE} + (K_{CE}, nonce)K_{CH} + H((ID_{CH}, ID_{CE})K_{IN}).$$

Enfin, chaque entité va calculer la clé commune de son coté en utilisant le nonce partagé comme suit :

$$K_{CH,CE} = (ID_{CE}, nonce)K_{CE}.$$

Ensuite, chaque nouveau cell-head va envoyer un message CE-UPDATE contenant son identifiant ID_{CE} avec un autre message caché d'une manière stéganographique contenant sa clé secrète K_{CE} et un nonce généré aléatoirement, chiffrés avec la clé initiale K_{IN} puis se met en attente de réponses.

$$CE \text{ -> } A : ID_{CE} + (K_{CE}, nonce)K_{IN}.$$

Lorsqu'un nœud intercepte le message CE-UPDATE, si le nœud n'appartient pas à la cellule alors il ignore le message. Sinon, il répond par un message CE-UPDATE-ACK contenant son identifiant ID_A avec un autre message caché d'une manière stéganographique contenant sa clé secrète K_A et le nonce envoyé par le cell-head, chiffrés avec la clé secrète K_{CE}. Il envoie aussi le haché des deux identifiants ID_{CE} et ID_A chiffrés avec la clé initiale K_{IN} pour assurer l'authentification.

$$A \text{ -> } CE : ID_A + (K_A, nonce)K_{CE} + H(ID_{CE}, ID_A)K_{IN}$$

Enfin, chaque entité va calculer la clé commune de son coté en utilisant le nonce partagé comme suit :

$$K_{CE,A}=(ID_A,nonce)K_{IN}.$$

Le renouvellement des clés secrètes des nœuds du réseau se fera de la même façon décrite dans la phase 2.

4.5.5 Phase de supression d'un nœud

Il existe plusieurs raison pour supprimer un nœud et le faire retirer du réseau : l'épuisement de sa source d'énergie, l'endommagement par un attaquant, etc.

Dans cette solution il existe deux cas qui nécessitent la suppression d'un nœud :

- Le cell-head détecte un nœud de sa cellule comme étant malveillant. Dans ce cas, il ignore ses messages, supprime la clé commune établie avec lui et l'exclue de sa liste des nœuds de sa cellule puis envoie un message qui contient l'identifiant du nœud compromis vers la station de base.
- La station de base détecte un comportement anormal d'un nœud. Dans ce cas, elle envoie un message au cell-head de la cellule où se trouve ce nœud pour l'informer de sa compromission. De ce fait, le cell-head va ignorer ses messages, supprimer la clé commune établie avec lui et l'exclure de sa liste des nœuds de sa cellule.

4.5.6 Phase d'ajout d'un nouveau nœud

On ajoute des nœuds au réseau soit pour maintenir une bonne connectivité, soit pour remplacer les nœuds supprimés. Quand un nouveau nœud N est déployé, la station de base va lui affecter un nouvel identifiant, une clé secrète unique pour lui, la clé initiale utilisé pour le tour en cours ainsi que la clé publique de la station de base.

Après le déploiement de ce nœud dans une cellule précise, le cell-head le découvre puis lui envoie un message CE-JOIN, le nœud lui répond par un message CE-JOIN-ACK, puis chaque entité va calculer la clé commune de son coté. Le contenu des messages sont décrites dans la phase 2.

4.6 Discussion

Le schéma proposé est un protocole déterministe avec une topologie hiérarchique car les nœuds sont organisés en groupes appelés cellules, ces dernières sont groupés pour former des clusters. L'utilisation de la stéganographie a pour objectif de diminuer de manière considérable le nombre de messages circulant dans le réseau, mais aussi de dissimuler certains messages important, permettant une sécurité meilleure. La compromission de la clé initial K_{IN} par un attaquant lui permet de récupérer toutes les informations stockés dans la mémoire du nœud compromis et de les utiliser pour perturber les communications, mais cela ne pourra pas durer longtemps car la clé initiale K_{IN} est mise à jour d'une manière périodique, mais aussi si un nœud

compromis est détecté soit par le cell-head ou par la SB il sera alors isolé et supprimé du réseau.

Ce qui concerne les objectifs de la sécurité, KMPS assure la propriété d'authentification entre le nœud cell-head et les nœuds qui forment la cellule avec l'utilisation de la fonction de hachage H.

Pour assurer la confidentialité des données transmises on a opté pour une combinaison entre la stéganographie et la cryptographie, même dans cette dernière on a utilisé une combinaison entre le chiffrement symétrique et asymétrique.

KMPS assure aussi la scalabilité puisque il permet d'ajouter des nouveaux nœuds au réseau pour remplacer les nœuds compromis ou épuisés, l'ajout des ces nouveaux nœuds se fera sans aucune contrainte sur le bon fonctionnement du réseau aide à augmenter la résilience, mais aussi d'augmenter la connectivité du réseau.

Les inconvénients de ce protocole c'est la non mobilité des nœuds car une fois les cellules et les clusters formés on pourra plus les modifiés la complexité en mémoire surtout pour les cells-head qui doivent stockés les trois clés K_{IN}, K_P, K_{CE}, plus les clés commune établie avec tous les nœuds de la cellule.

4.7 Exemple illustratif

Pour bien expliquer le fonctionnement de notre protocole on présente l'exemple suivant : soient les nœuds A, B, C, D, E, F et une station de base SB.

Dans la phase de pré-distribution la station de base SB attribue pour tous les nœuds du réseau un ID unique, une clé sécrète, une clé initiale et la clé publique de la station de base.

Après le déploiement du réseau, la station de base diffuse le message SB-JOIN pour choisir le cluster-head, dans ce cas le nœud A sera choisi alors :

SB -> $\star$: ID_A.

Le nœud A vérifie que son ID est mentionné dans le message alors il devient cluster- head, alors il diffuse un message CL-TEAM pour former le cluster et trouver les nœuds qui vont appartenir à ce cluster

A -> $\star$: ID_A.

On suppose que les nœuds B, C, D, E, F intercepte le message de A et lui répondent positivement alors le cluster sera composé comme suit A, B, C, D, E, F avec A comme cluster-head.

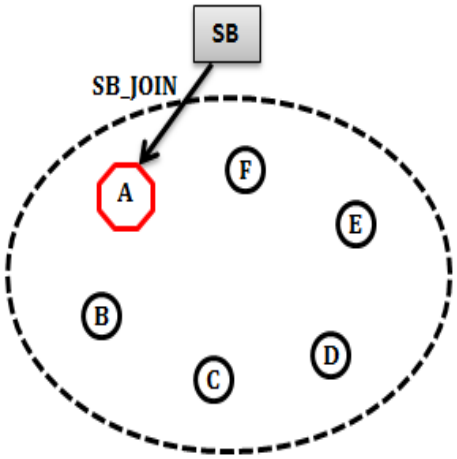

FIGURE 4.1 – Sélection du cluster-head

Une fois le cluster formé, les nœuds A va choisir aléatoirement les nœuds cell-head, supposons que les nœuds C, F sont choisis et envoie le message CL-JOIN :
A -> $\star$: ID_A, ID_C, ID_F + Message caché (K_{CH}, nonce) K_{IN} .

Les nœuds C et F vont vérifier que leurs ID sont mentionnés dans le message, il répond avec le message suivant :
C -> A : ID_C + Message caché (K_C, nonce) K_A .
F -> A : ID_F + Message caché (K_F, nonce) K_A .

Après l'échange des messages, les nœuds C et F vont devenir des cell-heads et la clé commune entre les nœuds C et A et entre F et A va être établie comme suit :
A -> C : (ID_C, nonce)K_C
A -> F : (ID_F, nonce)K_F

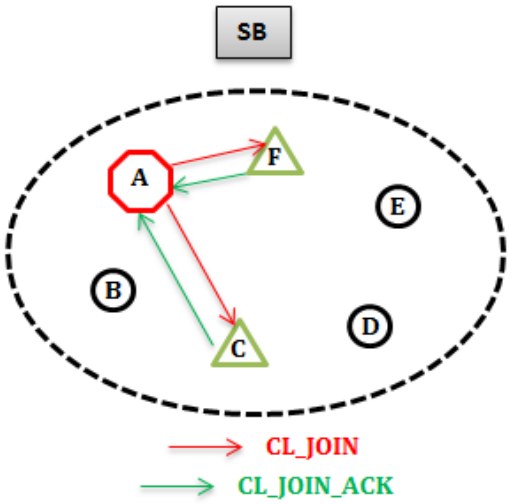

FIGURE 4.2 – Sélection des cell-heads

Après la définition des cell-head, il est temps de former les cellules pour cela les nœuds cell-heads C, F diffuse les messages suivant :
C -> $\star$: ID_C +Message caché (K_C, nonce)K_{IN}
F -> $\star$: ID_F + Message caché (K_F, nonce)K_{IN}

On suppose que les nœuds A et B interceptent le message du nœud C. Les nœuds E et D interceptent le message du nœud F. Alors la réponse des nœuds sera comme suit :
A -> C : ID_A . Puisque le nœud A est le cluster-head et la clé commune ayant déjà était définis.
B -> C : ID_B + Message caché (K_B, nonce)K_C + H((ID_B, ID_C)K_{IN}).
E -> F : ID_E + Message caché (K_E, nonce)K_F + H((ID_F, ID_E)K_{IN}).
D -> F : ID_D + Message caché (K_D, nonce)K_F + H((ID_F, ID_D)K_{IN}).

Après l'échange des messages les clés communes vont être établies comme suit :
$K_{B,C}$= (ID_B, nonce)K_B
$K_{E,F}$= (ID_E, nonce)K_E
$K_{D,F}$= (ID_D, nonce)K_D

Alors l'état du réseau va devenir comme suit :
Cellule 1 : A, B, C avec le nœud C comme cell-head.
Cellule 2 : D, E, F avec le nœud F comme cell-head.

Après l'établissement des clés communes, chaque nœud va supprimer les clés privées qu'ils possèdent et informent la station de base de ce changement :

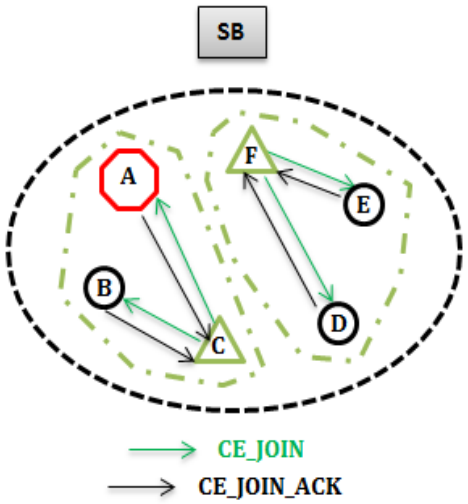

FIGURE 4.3 – Formation des cellules

A -> SB : $ID_A + (K_{A+})K_P$
B -> SB : $ID_B + (K_{B+})K_P$
C -> SB : $ID_C + (K_{C+})K_P$
D -> SB : $ID_D + (K_{D+})K_P$
E -> SB : $ID_E + (K_{E+})K_P$
F -> SB : $ID_F + (K_{F+})K_P$

Lors de la phase de communication, les nœds communiquent avec le cell-head qui à son tour communique avec le cluster-head qui envoi les messages à la station de base.

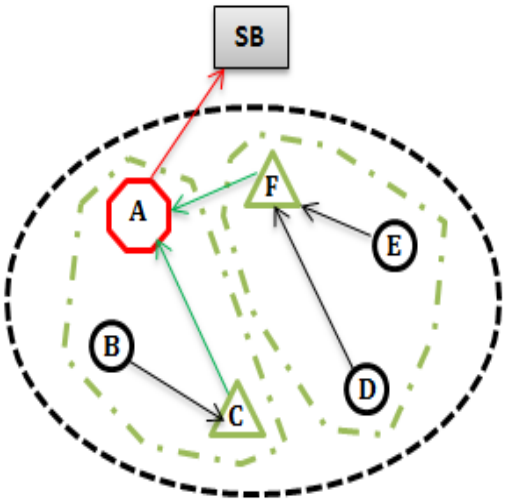

FIGURE 4.4 – Phase de communication

- **La phase de mise à jour**

Après chaque tour, la sélection d'un nouveau cell-head et d'un nouveau cluster-head va être nécessaire, pour cela chaque nœud d'une cellule envoi son énergie restante au cell-head qui calcule l'énergie moyenne restante de la cellule et l'envoi au cluster-head. Le nœud avec l'énergie maximum et supérieur à cette énergie moyenne va être sélectionné comme nouveau cell-head. On suppose que les nouveaux cell-heads sont les suivants : Le nœud B pour la cellule1 et le nœud D pour la cellule2.

Après la réception des énergies moyennes de la part des cellules, le cluster-head calcule l'énergie moyenne du cluster et le nœud avec l'énergie maximum supérieur à cette énergie va être élu nouveau cluster-head. On suppose que le nœud E est les nouveaux cluster-heads.

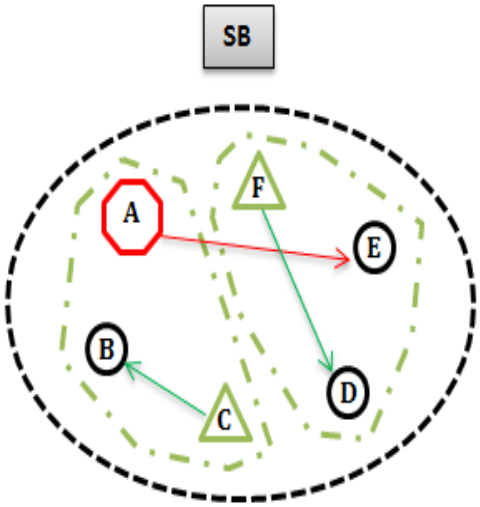

FIGURE 4.5 – Mise à jour du cluster-head et des cell-heads

Après cette sélection, le nœud E va diffuser le message CL-UPDATE pour établir une clé commune avec les nouveau cell-head :

E -> B : ID_E +Message caché (K_E, nonce)K_{IN}

E -> D : ID_E+ Message caché (K_E, nonce)K_{IN}

Les nœuds B, D vont intercepter ce message et vont répondre avec le message CL-UPDATE-ACK :

B -> E : ID_B+ Message caché (K_B, nonce)K_E + H((ID_E, ID_B)K_{IN})

D -> E : ID_D + Message caché (K_D, nonce)K_E + H((ID_E, ID_D)K_{IN})

La clé commune va être établie de la même manière que dans la première phase.

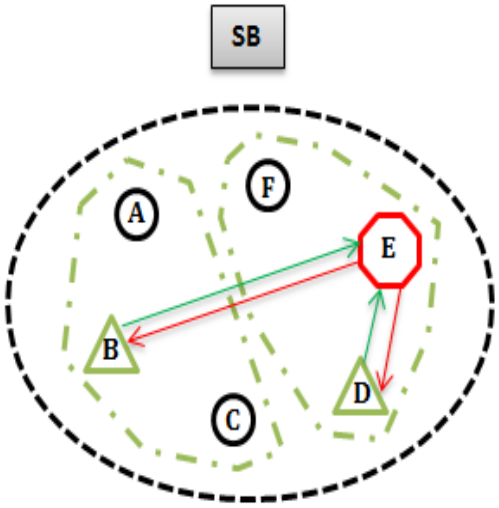

FIGURE 4.6 – Etablissement de la clé commune entre le cluster-head et ses cell-heads

Les nouveau cell-head B et D vont établir les clés communes avec les nœuds de leur cellule respective comme suit :

B -> A : ID_B +Message caché (K_B, nonce)K_{IN}

B -> C : ID_B +Message caché (K_B, nonce)K_{IN}

D -> E : ID_D + Message caché(K_D, nonce)K_{IN}

D -> F : ID_D + Message caché(K_D, nonce)K_{IN}

Les nœuds vont répondre :

A -> B : ID_A+Message caché (K_A, nonce)K_B + MAC (ID_B, ID_A)K_{IN}

C -> B : ID_C+ Message caché (K_C, nonce)K_B + MAC (ID_B, ID_C)K_{IN}

E -> D : ID_E+ Message caché (K_E, nonce)K_D + MAC (ID_D, ID_E)K_{IN}

F -> D : ID_F + Message caché (K_F, nonce)K_D + MAC (ID_F, ID_D)K_{IN}

Aprés l'échange des message le clé commune va être établie comme la phase précédente.

Après cette sélection, le nœud E va diffuser le message CL-UPDATE pour établir une clé commune avec les nouveau cell-head :

E -> B : ID_E +Message caché (K_E, nonce)K_{IN}

E -> D : ID_E+ Message caché (K_E, nonce)K_{IN}

Les nœuds B, D vont intercepter ce message et vont répondre avec le message CL-UPDATE-ACK : B -> E : ID_B+ Message caché (K_B, nonce)K_E + H((ID_E, ID_B)K_{IN})

D -> E : ID_D + Message caché (K_D, nonce)K_E + H((ID_E, ID_D)K_{IN})

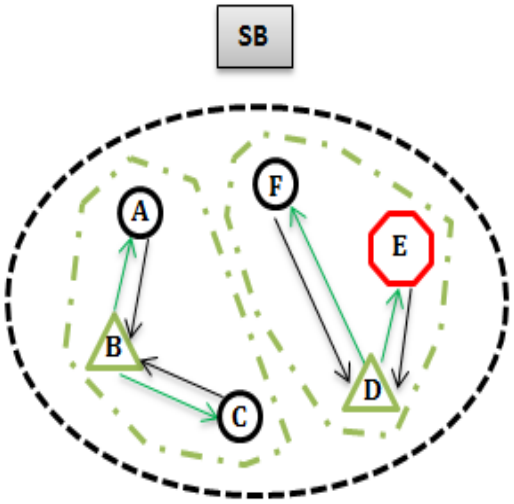

FIGURE 4.7 – Etablissement de la clé commune entre les cell-heads et ses nœuds

La clé commune va être établie de la même manière que dans la première phase.

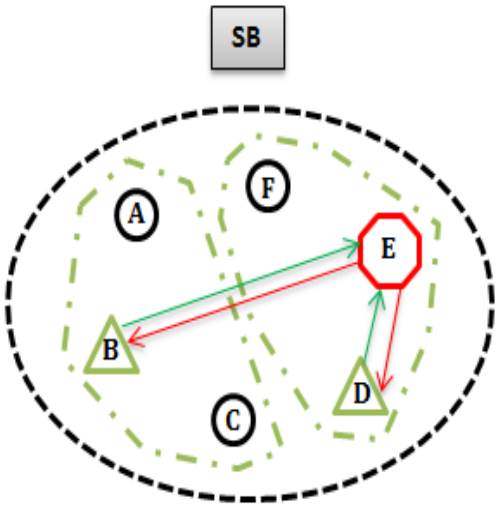

FIGURE 4.8 – Etablissement de la clé commune entre le cluster-head et ses cell-heads

Les nouveau cell-head B et D vont établir les clés communes avec les nœuds de leur cellule respective comme suit :

B -> A : ID_B +Message caché (K_B, nonce)K_{IN}

B -> C : ID_B +Message caché (K_B, nonce)K_{IN}

D -> E : ID_D + Message caché(K_D, nonce)K_{IN}

D -> F : ID_D + Message caché(K_D, nonce)K_{IN}

Les nœuds vont répondre :

A -> B : ID_A+Message caché (K_A, nonce)K_B + MAC (ID_B, ID_A)K_{IN}

C -> B : ID_C+ Message caché (K_C, nonce)K_B + MAC (ID_B, ID_C)K_{IN}
E -> D : ID_E+ Message caché (K_E, nonce)K_D + MAC (ID_D, ID_E)K_{IN}
F -> D : ID_F + Message caché (K_F, nonce)K_D + MAC (ID_F, ID_D)K_{IN}

Aprés l'échange des message le clé commune va être établie comme la phase précédente.

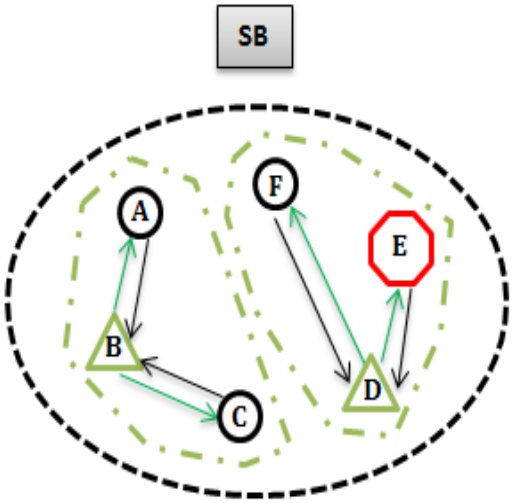

FIGURE 4.9 – Etablissement de la clé commune entre les cell-heads et ses nœuds

Maintemant on passe à la mise a jour de la clé initiale, la station de base va informer cahque nœud de la nouvelle clé :
SB -> A : (K_{IN}) K_A
SB -> B : (K_{IN}) K_B
SB -> C : (K_{IN}) K_C
SB -> D : (K_{IN}) K_D
SB -> E : (K_{IN}) K_E
SB -> F : (K_{IN}) K_F

- **Ajout d'un nouveau nœud**

On suppose qu'un nouveau nœud G est déployé dans le réseau alors la SB va lui affecter un ID, une clé secrète, la clé initiale, la clé publique et l'affecte à une cellule précise, le cell-head de cette cellule va alors établir une clé commune avec ce nouveau nœud comme décrit précédemment.

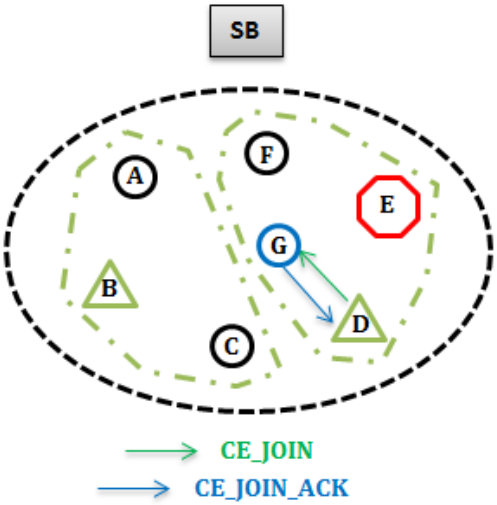

FIGURE 4.10 – Ajout d'un nouveau nœud

4.8 Simulation

La simulation informatique, ou simulation numérique, est une série de calculs effectués sur un ordinateur et reproduisant un phénomène physique. Elle aboutit à la description du résultat de ce phénomène, comme s'il s'était réellement déroulé. Cette représentation peut être une série de données, une image ou même un film vidéo.

Un simulateur peut réagir à des modifications de paramètres et modifier ses résultats en conséquence. Un simulateur de vol, par exemple, modifie la trajectoire calculée de l'avion en fonction des commandes transmises par l'utilisateur. [63]

Afin de simuler notre protocole, nous avons utilisé Java qui est à la fois un langage de programmation informatique orienté objet et un environnement d'exécution informatique portable. Le langage Java a la particularité principale d'être portable sur plusieurs systèmes d'exploitation tels que Windows, Mac OS ou Linux. C'est la plateforme qui garantit la portabilité des applications développées en Java.

Java aussi possède l'avantage d'être extensible à l'infini. Il utilise très peu de méthode native et la plupart des packages (librairies de classes) sont écrits eux-mêmes en java. Le package swing en est le parfait exemple. Java possède aussi l'avantage d'être très sécurisé, ce qui le différencie beaucoup de C++. On n'aura pas à s'occuper de libérer la mémoire à la fin de vie d'un objet. Un garbage collector (ramasse miettes) s'en charge pour nous[64].

Il permet de créer des applications autonomes et de doter les documents html de nouvelles fonctionnalités : animations interactives, applications intégrées, modèles 3D, etc. Ce langage est orienté objet et comprend des éléments spécialement conçus pour la création d'applications multimédia. [65]

4.8.1 Environnement de simulation

Notre modèle d'expérimentation est établi sur 150 nœuds dispersés aléatoirement sur une surface carrés de 100m x 100m. Nous supposons que tous les nœuds ont une position fixe durant toute la période de la simulation. Les paramètres de la simulation sont résumés dans le tableau suivant :

Paramètres	**Valeur**
La taille du réseau	100m x 100m
L'emplacement de la station de base	x=50, y=50
Le nombre de nœuds	150
Le nombre de Clusters	4
Le nombre de cellules	28
L'énergie initiale des nœuds	2 Joules
La taille des paquets	512 bits

TABLE 4.2 – Paramètres de simulation

4.8.2 Modèle énergétique

Dans notre solution, les capteurs dans le réseau se trouvent dans l'une des situations suivantes : Soit c'est un cluster-head (CH), soit c'est un membre à 1-saut (CM-1S) qui présente les cell-heads, soit c'est un membre à 2-sauts (CM-2S) qui présente les autres nœuds.

L'ensemble de ses nœuds assurent la couverture de la zone cible. En effet, les cluster-heads s'occupent du captage, de l'agrégation et de la transmission des données à la station de base. En revanche, les membres à un saut (CM-1S) effectuent le captage, l'agrégation de données en provenance des (CM-2S) et la transmission de ces données aux clusters-heads. Les membres à deux sauts, quant à eux, sont seulement responsables du captage des informations et de leur transmission aux CM-1S.

Cette organisation nous permet d'adopter le modèle énergétique proposé dans DEECIC [66], ce modèle est basé principalement sur celui proposé par Heinzelman et al [67].

Pour cela, l'énergie consommée par un membre à 2-saut (CM-2S) notée j dépend de la distance qui le sépare du membre à 1-saut (CM-S1) qui lui est associé noté i, (dans notre cas c'est le cell-head de la cellule à laquelle ce nœud j appartient). Cette énergie se calcule comme suit :

$$E(\text{CM-2S}_j) = L * E_{elec} + L * \varepsilon fs * \text{dist}^2(\text{CM-2S}_j, \text{CM - 1S}_i).$$

L'énergie consommée par un membre à 1-saut (CM-1S) notée i dépend du nombre de ces membres à 2-saut (CM-2S), cette énergie se calcule comme suit :

$$E\ (\text{CM - 1S})_i = L * E_{elec} * \text{Nbr}(\text{CM-2S}(\text{CM - 1S}_i)) + L * E_{DA} * (\text{Nbr}\ (\text{CM-2S}\ (\text{CM - 1S}_i)) +1) + L * (E_{elec} + \varepsilon fs * \text{dist}^2\ ((\text{CM-1S})_i, \text{CH}_k)).$$

Finalement, l'énergie consommée par le cluster-head dépend du nombre de ces membres à 1-saut (CM-1S) et la distance qui le sépare de la station de base (SB) , cette énergie se calcule comme suit :

$$E(\text{CH}) = L * E_{elec} * \text{Nbr}(\text{CM-1S}(\text{CH}_k)) + L * E_{DA} * (\text{Nbr}(\text{CM-1S}(\text{CH}_k)) + 1) + L * (E_{elec} + \varepsilon fs * \text{dist}^4(\text{CH}_k, \text{SB}).$$

Ou :

- E_{elec} représente l'énergie électronique du transmetteur
- εmp et εfs représentent les énergies d'amplification.
- E_{DA} représente l'énergie d'agrégation consommée par un Cluster-head ou un (CM-1S).
- f représente le nombre de trames transmises par un cluster-head dans une période.
- L représente la taille du paquet de données.
- dist(i, j) représente la distance en mètres séparant le capteur i du voisin j.

4.8.3 Résultats de simulation

Le temps de simulation est divisé en plusieurs rounds, chaque round dure 10 secondes. Durant chaque round, on a mesuré la consommation d'énergie des nœuds pendant l'échange des messages entre eux et le nombre de messages échangés. Au début de la simulation, chaque nœud possède 2 Jouls.

L'énergie globale du réseau diminue à chaque round, un nœud est considéré comme épuisé si son énergie descend au-dessous de 0.5 Jouls. Le critère d'arrêt de la simulation est quand le nombre de nœuds du réseau passe au-dessous de 50 nœuds.

A la fin de la simulation, on aura trois graphe : le premier illustre le nombre de nœuds restants (actifs) par rapport au temps, le second illustre l'énergie moyenne globale consommée et le dernier illustre le nombre de messages échangés entre les nœuds

La figure 4.11 illustre un réseau de 150 nœuds déployés aléatoirement sur une surface de 100m x 100m.

Les nœuds normaux, les clusters-heads et les cell-heads sont affichés avec des points de couleurs différentes.

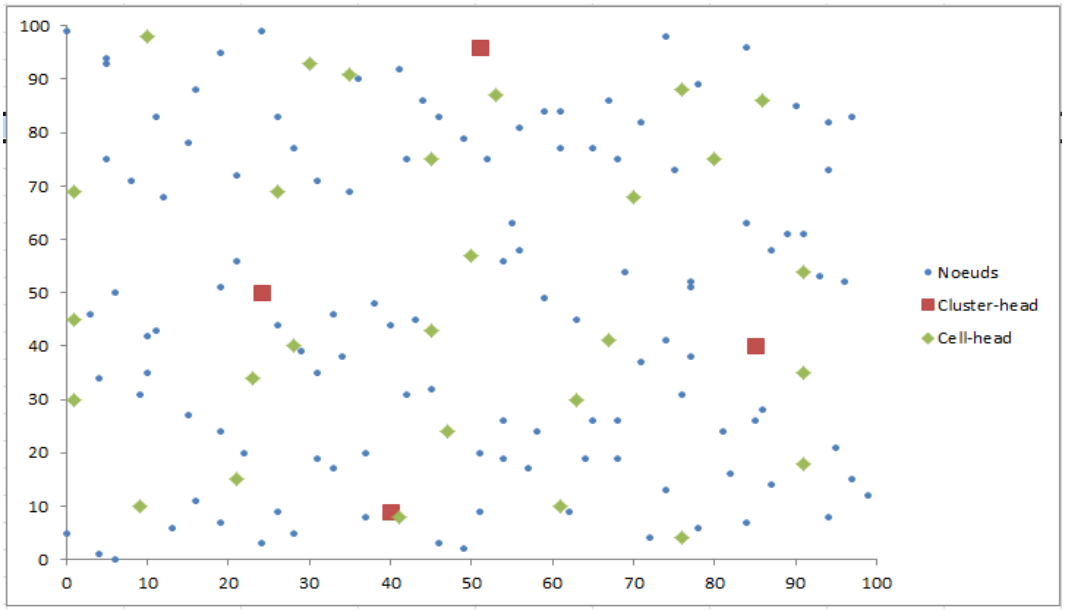

FIGURE 4.11 – le déploiement aléatoire de 150 nœuds

4.8.4 Analyse et évaluation des performances de notre protocole

Afin d'évaluer les performances de notre protocole, on a utilisé les métriques suivants :

- le nombre de nœuds restants
- la consommation d'énergie
- le nombre de messages échangés

Afin de comparer notre protocol, nous avons pris le protocole Cell-LEACH qui est un protocole de routage non sécurisé et le même protocole en lui ajoutant un protocole de gestion de clé qui est SNKM [45].

4.8.4.1 Le nombre de nœuds restants

Un nombre important de nœuds est nécessaire pour la couverture de l'espace surveillé et le maintien d'une bonne connectivité dans le réseau. La figure 4.12 illustre le nombre de nœuds restants dans le réseau de notre protocole et celle de Cell-LEACH seul et le duo Cell-LEACH/SNKM.

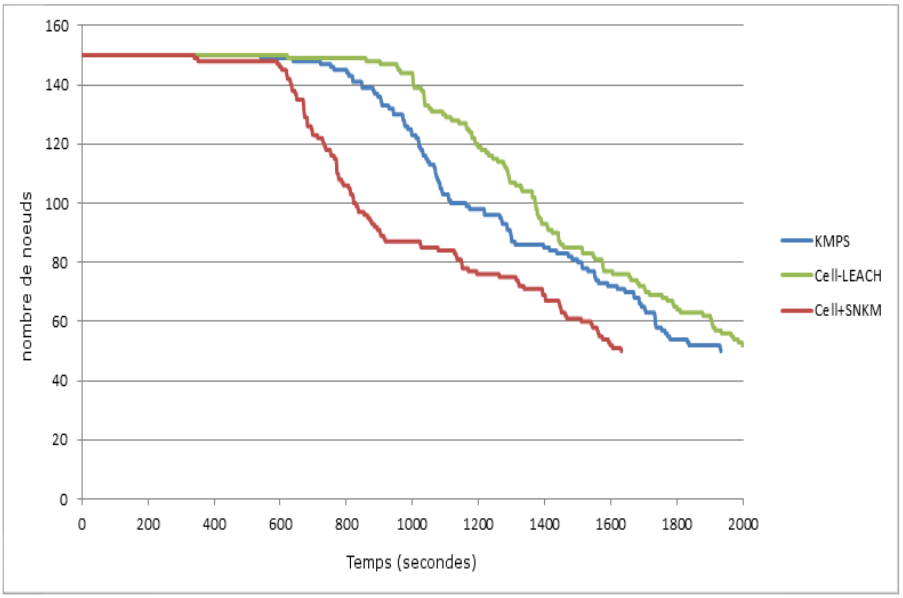

FIGURE 4.12 – Nombre de nœuds restants dans le réseau par rapport au temps

La figure nous montre que le nombre de nœuds du protocole KMPS diminue un peu plus rapidement par rapport au protocole Cell-LEACH (non sécurisé) à cause du message supplémentaire utilisé pendant la phase de la mise à jour. En comparant le protocole KMPS avec le duo Cell-LEACH/SNKM, on constate que KMPS permet au réseau de tenir plus longtemps, ceci est dû à l'utilisation de la stéganographie qui réduit au maximum le nombre de messages échangés.

4.8.4.2 La consommation d'énergie

Les ressources énergétiques déterminent la durée de vie du réseau et doit être soigneusement prise en compte dans la conception de n'importe quelle application. Pour cette raison, on a

mesuré et comparé la consommation d'énergie de KMPS avec les deux autres protocoles. La figure 4.13 illustre la consommation d'énergie moyenne globale du réseau des 3 protocoles.

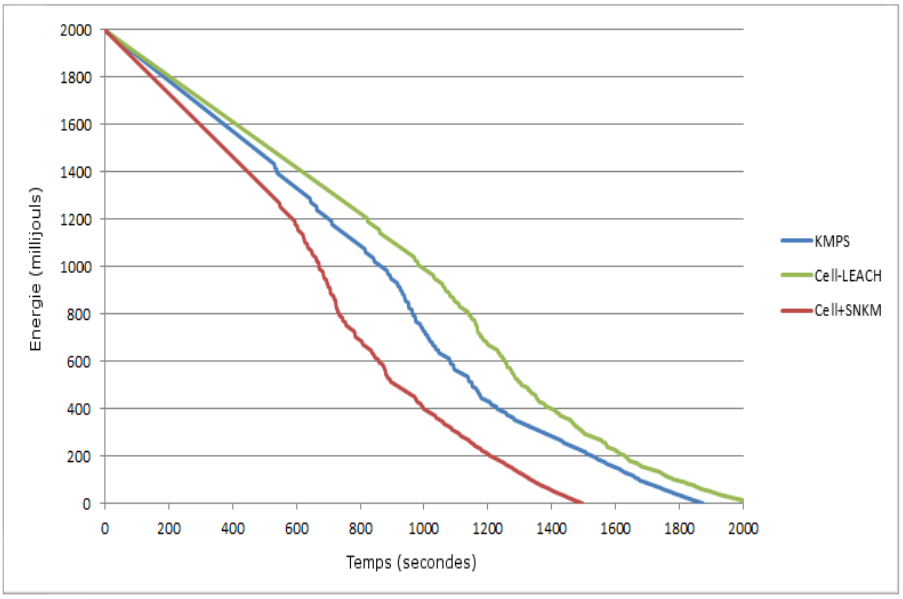

FIGURE 4.13 – consommation d'énergie moyenne globale du réseau par rapport au temps

La figure nous montre que la consommation d'énergie du réseau du protocole KMPS est légèrement supérieure au le protocole Cell-LEACH (non sécurisé), cela est dŭ à la présence de la sécurité dans KMPS (chiffrements/déchiffrements des messages, gestion des clés). On remarque aussi que la consommation d'énergie de KMPS est moins importante que celle du duo Cell-LEACH/SNKM.

4.8.4.3 Le nombre de messages echanges

La figure 4.14 illustre le nombre de messages échangés du réseau des 3 protocoles.

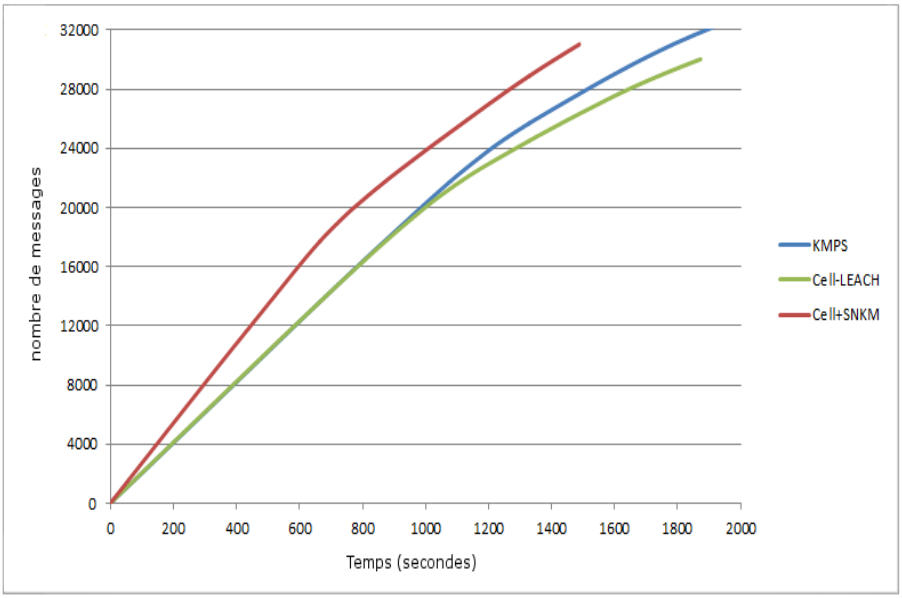

FIGURE 4.14 – Le nombre de messages échangés du réseau par rapport au temps

On remarque de le protocole KMPS échange moins de messages par rapport au duo CELL-LEACH/SNKM car il intègre protocole de gestion de clés dans un protocole de routage et l'utilisation de la stéganographie permet de réduire d'une manière considérable le nombre de messages dans le réseau. On remarque aussi que le nombre de messages échangés est très similaire entre KMPS et Cell-LEACH car le protocole KMPS envoie les mêmes messages que le protocole Cell-LEACH. La seule différece est que KMPS envoie un message supplémentaire pendant la phase de la mise à jour.

4.9 Conclusion

Dans ce chapitre, nous avons évalué les performances de notre approche KMPS en la comparant avec le protocole Cell-LEACH seul d'un coté et avec une combinaison du CELL-LEACH/SNKM d'un autre coté qui sont des protocoles très récents. Cette comapraison est faite à base de plusieurs paramètres de performances, tels que le le nombre de nœuds restants, la consommation d'énergie et le nombre de messages échangésdans le réseau.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

L'avènement récent de la technologie des réseaux de capteurs sans-fil, conjugué au progrès de miniaturisation des composants et à l'allongement de la durée de vie des batteries, annoncent un futur prometteur à cette technologie. De plus, le développement de nouveaux capteurs plus performants permettra d'étendre d'avantage les domaines d'applications déjà nombreux.

Les RCSFs constituent des sujets de recherche innovants pour diverses disciplines des sciences et techniques de l'information et de la communication mais avec toutefois des contraintes spécifiques s'érigeant en défis certains à relever. Parmi les problèmes posés à l'heure actuelle dans ce type de réseaux, la sécurité en est un véritable et auquel une solution adéquate doit être apportée.

Dans ce mémoire, nous avons mis en avant les caractéristiques essentielles des réseaux de capteurs sans-fils, ainsi que les besoins et les défis de la sécurité dans ces derniers. Nous avons étudié aussi quelques schémas de gestion de clés qui permettent d'offrir le service de sécurité de base pour n'importe quel système basé sur la communication.

L'ensemble des protocoles de gestion de clés proposés pour les RCSFs se basent principalement sur la cryptographie à clé symétrique et la méthode de pré-distribution de clés afin d'achever l'établissement de clés entre les entités communicantes dans le réseau. Nous avons étudié un ensemble de ces protocoles de gestion de clés qui sont classés dans plusieurs catégories selon la topologie du réseau (hiérarchique ou plate) et la façon dont les nœuds voisins partagent des clés communes (probabiliste ou déterministe).

Après l'étude de ces solutions, nous avons constaté que le défi dans la conception des schémas de gestion de clés est de trouver un compromis entre un système efficace et les contraintes

caractérisant les RCSFs.

Nommé KMPS (Key Mangement Protocol with Steganography), notre solution montre à travers les résultats de l'évaluation et de la simulation qu'elle peut fournir plus de sécurité avec moins d'exigence que d'autres solutions.

Afin de continuer le travail, nous proposons les perspectives suivantes :

- Implementer le protocole et le tester sur des capteurs réels.
- Intergrer notre protocole de gestion de clés dans d'autres prtotocoles de routages.
- Comparer notre protocole avec plus de protocoles.
- Adapter notre protocole au RCSFs mobiles et essayer d'exploiter les messages engendrés par la mobilité pour cacher les informations et diminuer le nombre de messages échangés.

BIBLIOGRAPHIE

[1] : K. Holger, A. Willig. Protocols and architectures for Wireless sensor networks. Wiley, 2005

[2] : B. Krishnamachari. Networking Wireless sensors. Cambridge University Press, 2006.

[3] : I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. Wireless sensor networks : a survey. Computer networks, 38(4) : 393x422, 2002.

[4] : I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. A survey on sensor networks. IEEE Communications Magazine, 40(8) : 102x114, august 2002.

[5] : K. Holger, A. Willig. Protocols and architectures for Wireless sensor networks. Wiley, 2005.

[6] : D. Martins, H. Guyennet. Steganography in MAC Layers of 802.15.4 Protocol for securing Wireless Sensor Networks. In IWNS 2010, 2nd IEEE Int. Workshop on Network Steganography, Nanjing, China, pages 824-828, November 2010.

[7] : G.D. Sousa. Etude en vue de la réalisation de logiciels bas niveau dédié aux réseaux de capteurs sans fils : microsystème de fichiers. Heudiasuc, France .17/11/2008.

[8] : K. Beydoun. Conception d'un protocole de routage hiérarchique pour les réseaux de capteurs. Thèse pour l'obtention du grade de docteur en informatique. Université de Franche-Comté. 2009.

[9] : W. Heinzelman, J. Kulik, H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. 5th annual ACM/IEEE international conference on Mobile computing and networking . August 1999, pp. 174 - 185.

[10] : C.E. Perkins, P. Bhagwat. Highly dynamic Destination-Sequenced Distance-Vector routing (DSDV) for mobile computers. ACM SIGCOMM Computer Communication Review. October 1994, Vol. 24, 4.

[11] : D. Walden The Bellman-Ford Algorithm and "Distributed Bellman-Ford" . 2009.

[12] : C.E. Perkins, E.M. Royer, S.R. Das. Ad hoc on demand distance vector (AODV) routing. In IETF, Internet Draft, draft-ietf-manet-aodv-05.txt. 2000.

[13] : C. Tsu-Wei, M. Gerla Global state routing : a new routing scheme for ad-hoc wireless networks. IEEE International Conference on Communications. 1998, Vol. 1, pp. 171-175.

[14] : M. Gerla, X. Hong, G. Pei. Fisheye State Routing Protocol (FSR) for Ad Hoc Networks. INTERNET-DRAFT. 17 June 2002. "http ://tools.ietf.org/html/draft-ietf-manet-fsr-03".

[15] : L. Stevens, K. Kleinrock Fisheye : A Lenslike Computer Display Transformation. s.l. : UCLA, Computer Science Department, 1971. Technical report.

[16] : M. Joa-Ng, I.-T. Lu. A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks. IEEE Journal on Selected Areas in Communications 17 (1999), no. 8, p. 1415-1425.

[17] : M. Jiang, J. Li, Y.C. Tay. Cluster Based Routing Protocol (CBRP) . Internet draft, IETF-MANET Working Group. 1999.

[18] : A. Manjeshwar, D.P. Agrawal. TEEN : a routing protocol for enhanced efficiency in wireless sensor networks. Proceedings 15th International Parallel and Distributed Processing Symposium. 2001, pp. 2009-2015.

[19] : W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan. Energy-efficient Communication Protocol for Wireless Microsensor Networks. Proceedings of the IEEE Hawaii International Conference on System Sciences. 2000, Vol. 2, p. 10.

[20] : Z. Lum. An Application-Specific Protocol Architecture for Wireless Microsensor Networks. "http ://fr.slideshare.net/zhendong/leachprotocol".

[21] : W.R. Heinzelman, A.P. Chandrakasan, H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. IEEE Transactions Wireless Communications. October 2002, Vol. 1, 4, pp. 660-670.

[22] : S. Lindsey, C.S. Raghavendra PEGASIS : Power-efficient gathering in sensor information systems. IEEE Aerospace Conference Proceedings. 2002, Vol. 3, pp. 3-1130.

[23] : A. Yektaparast, F.-H. Nabavi, A. Sarmast. An Improvement on LEACH Protocol (Cell-LEACH). International Conference on Advanced Communication Technology (ICACT). Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Khouzestan, Iran. Février 2012.

[24] : Y. Challal. Réseaux de capteurs sans fils : Système intelligents pour de transfert. Université de technologie de Compiègne, Heudiasuc, France. 2008.

[25] : G. Gaubatz et al. Public Key Cryptography in Sensor Networks-Revisited. ESAS '04 : 1st European Wksp, Security in Ad-Hoc and Sensor Networks, 2004.

[26] : K. Piotrowski et al. How Public Key Cryptography Influences Wireless Sensor Node Lifetime. SASN '06, Alexandria, Virginia, USA, October 30, 2006.

[27] : A. S. Wander et al. Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks. PerCom '05, March 2005.

[28] : N. Gura et al. Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs. Boston, Massachusetts : 6th International Workshop on Cryptographic Hardware and Embedded Systems, August 2004.

[29] : I.F. Blake, G. Seroussi, P.S. Nigel Advances in Elliptic Curve Cryptography. London Mathematical Society Lecture Note Series (No. 317), April 2005.

[30] : A. Liu, P. Ning. TinyECC : Elliptic Curve Cryptography for sensor networks (version 0.3). February, 2007.

[31] : D. J. Malan, M. Welsh, M. D. Smith. A Public-Key Infrastructure for Key Distribution in TinyOS based on Elliptic Curve Cryptography. Proc. 1st IEEE Int'l. Conf. Sensor and Ad Hoc Communications and Networks, Santa Clara, CA, Oct. 2004.

[32] : H. Wang, B. Sheng, Q. Li. Elliptic curve cryptography-based access control in sensor networks. Int. J. Security and Networks, Vol. 1, Nos. 3/4, 2006.

[33] : M. Mohamed Lamine et A. Makhlouf. Protocole Efficace de Gestion des Clés dans les Réseaux de Capteurs Sans-Fil. UAMB, Ecole Doctorale en informatique ReSyD Bejaia, Algérie. Département informatique, UFAS Setif, Algérie. Novembre 2009.

[34] : A. Perrig et al. SPINS : Security Protocols for Sensor Networks. Mobile Computing and Networking, Rome Italy, July 2001.

[35] : Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, M. Galloway. A survey of key management schemes in wireless sensor networks. Computer Communications 30, Elsevier, May 2007.

[36] : R.J. Watro, D. Kong, S. F. Cuti, C. Gardiner, C. Lynn, P. Kruus. TinyPK : securing sensor networks with public key technology. In ACM SASN'04, pp. 59-64, 2004.

[37] : D. Hankerson, A. Menezes, S. Vanstone. Guide to Elliptic Curve Cryptography. Springer, 2004.

[38] : A.S. Wander, N. Gura, H. Eberle, V. Gupta, S.C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. Pervasive Computing and Communications, IEEE International Conference, pp. 324-328, on 8-12 March 2005.

[39] : L. Eschenauer, V.D. Gligor. A key-management scheme for distributed sensor networks. In Proceedings of the 9th ACM conference on Computer and communications security, November 2002.

[40] : X. Du et al. An improved key distribution mechanism for large-scale hierarchical wireless sensor networks. Security Issues in Sensor and Ad Hoc Networks, January 2007, Pages 35-48.

[41] : H. Chan, A. Perrig. Pike : peer intermediaries for key establishment in sensor networks. In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer andCommunications Societies, 13-17 March 2005, Miami, FL, USA, pages 524_535. IEEE, 2005.

[42] : X. Du et al. An improved key distribution mechanism for large-scale hierarchical wireless sensor networks. Security Issues in Sensor and Ad Hoc Networks, January 2007, Pages 35-48.

[43] : B. Lai, S. Kim, I. Verbauwhede. Scalable session key construction protocol for wireless sensor networks. In IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES), page 7, 2002.

[44] : J. Deng, C. Hartung, R. Han, S. Mishra. A Practical Study of Transitory Master Key Establishment for Wireless Sensor Networks. Proc First IEEE Int'l Conf Security and Privacy for Emerging Areas in Comm. Networks (SecureComm '05), Sept. 2005.

[45] : B. Jiana, E. Xu. An Energy-efficient Security Node-based Key Management Protocol for WSN. Department of Information Science and Technology, Université de Bohai Jinzhou, China. 2013.

[46] : P. Zhao, Y. Xu, M. Nan. A Hybrid Key Management Scheme Based on Clustered Wireless Sensor Networks. Department of Mathematics and Computer Science, Anhui Normal University, China, 2012.

[47] : L. Eschenauer, V.D. Gligor. A key-management scheme for distributed sensor networks. In Vijayalakshmi Atluri, editor, Proceedings of the 9th ACM Conference on Computer and Communications Security, CCS 2002, Washingtion, DC, USA, November 18-22, 2002, pages 41_47, New York, NY, USA, 2002. ACM.

[48] : H. Chan, A. Perrig, D. Song. Random key predistribution schemes for sensor networks. In IEEE Security and Privacy, 2003. Proceedings. 2003 Symposium on, pages 197_213, 11-14 2003.

[49] : R. Blom. An optimal class of symmetric key generation systems. In Proceedings of the Eurocrypt 84 Workshop on Advances in Cryptology : Theory and Application of Cryptographic Techniques. Springer Verlag, 1985, pages 335-338.

[50] : Z. Yu, Y. Guan. A robust group-based key management scheme for wireless sensor networks. In : Proceedings of IEEE Wireless Communications and Networking Conference (WCNC 2 005). New Orleans, LA USA : IEEE Press, 2005, pp.13-17.

[51] : W. Du, J. Deng, Y. S. Han, S. Chen, P.K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge. In Proceedings of IEEE NFOCOM'04, Hong Kong : IEEE Press, 2004. 586-597.

[52] : D. Liu, P. Ning. Improving key pre-distribution with deployment knowledge in static sensor networks. ACM Transactions on Sensor Networks, 2005, 1(2) : pp.204-239.

[53] : D. Huang, M. Mehta, D. Medhi, L. Harn. Location-aware key management scheme for wireless sensor networks. In Proceedings of ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04), Washington DC, USA : ACM Press, 2004. pp.29-42.

[54] : A.K. Das, D. Giri. An Identity Based Key Management Scheme in Wireless Sensor Networks. Department of Computer Science and Engineering Indian Institute of Technology, Kharagpur 721 302, India. Mars 2011.

[55] : S. Athmani. Protocole de sécurité pour les réseaux de capteurs sans fil. Mémoire de Magister en informatique. Université Hadj Lakhder, Batna. 2010.

[56] : W. Znaidi. Quelques propositions de solutions pour la sécurité des réseaux de capteurs sans fil. Thèse pour l'obtention du grade de docteur en informatique et mathématique. L'Institut National des Sciences Appliquées de Lyon. 2010.

[57] : D. Martins. Sécurité dans les réseaux de capteurs sans fil : Stéganographie et réseaux de confiance. Thèse pour l'obtention du grade de docteur en informatique. Université de Franche-Comté, 2010.

[58] : F.-B. Huyghe. Histoire de la cryptologie : Une technologie du secret. In "http ://www.huyghe.fr/actu_109.htm".

[59] : A. Cheddad, J. Condell, K. Curran, P. McKevitt. Digital image steganography : Survey and analysis of current methods. Signal Processing, 90(3) :727-752, March 2010.

[60] : T. Kho. Steganography in the 802.15.4 physical layer. Technical report, 2007.

[61] : S. Lanzisera, A.M. Mehta, K. Pister. Steganography in 802.15.4 wireless communication. In Advanced Networks and Telecommunication Systems, 2008. ANTS '08. 2nd International Symposium on, pages 1_3, Mumbai, 2008.

[62] : Z. Alliance. "In http ://www.zigbee.org/".

[63] : "http ://www.futura-sciences.com/magazines/hightech/infos/dico/d/informatique-simulation-informatique-11319/". (dernière consultation : Juin 2014)

[64] : "http ://gaetan.dussaux.free.fr/cours/java/1.htm". (dernière consultation : Juin 2014)

[65] : "http ://www.6ma.fr/lexique/informatique/java/". (dernière consultation : Juin 2014)

[66] : Z. Liu, Q. Zheng, L. Xu, X. Guan. A distributed energy-efficient clustering algorithm with improved coverage in wireless sensor networks. Future Generation Computer Systems, vol 28(5) : 167-739, 2011.

[67] : W.B. Heinzelman, A.P. Chandrakasem, H. Balaknishnan An application-specific protocol architecture for wireless microsensor networks. IEEE Trans on wireless Communications, vol 4(1) pp. 660-670, 2002.

Résumé

La gestion des clés dans les réseaux de capteurs sans fil (*RCSFs*) est reconnue comme un domaine de recherche très actif vu les spécificités de ce type de réseau, où la consommation de l'énergie et la sécurité sont considérés comme les défis majeurs. Cependant, l'utilisation de la cryptographie pour atteindre les objectifs de sécurité est limitée par la faible puissance de calcul et la durée de vie limité de ce type de réseau.

Dans ce mémoire nous présentons notre proposition, un protocole de gestion de clé basé sur l'utilisation de la stéganographie, qui consiste à la dissimulation des messages du protocole de gestion des clés dans ceux du protocole de routage, afin de minimiser la consommation d'énergie et prolonger la durée de vie du réseau.

Mots-clés :
Réseaux de capteurs sans fil, Gestion de clé , Stéganographie, Sécurité, Cryptographie.

Abstract

Key management in wireless sensor networks (*WSNs*) is recognized as a very active area of research given the specificities of this type of network, where the consumption of energy and security are considered major challenges. However, the use of cryptography to reach a high level of security is limited by the low computing power and the limited lifetime of this type of network.

In this memory we present our proposal, a key management protocol based on the use of steganography, which involves hiding key management protocol messages in those of the routing protocol to minimize the consumption of energy and the extended life of the network.

Keywords :
Wireless **s**ensor **n**etworks, Key Mangement, Cryptography, Steganography, Security.