République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITÉ MOULOUD MAMMERI DE TIZI OUZOU

Faculté de génie électrique et d'informatique

Département d'informatique

Mémoire de fin de cycle

en vue de l'obtention du diplôme de master 2 en Informatique

Spécialité : Conduite de Projet Informatique

Thème

Proposer un schéma de gestion de clés dynamiques dans un réseau de capteurs sans fil mobile

---

Présenté par :

M^{lle} BRIK Ouardia
M^{lle} CHOUGGAR Melissa

Proposé et dirigé par :

Mr RAMDANI Mohamed

Devant le jury composé de :

M^{me} AOUDJIT Rachida	Présidente
Mr SADI Samy	Examinateur

Promotion 2017/2018

Remerciement

Nous remercions d'abord Dieu le tout puissant qui nous a donné la force, le courage et la volonté pour accomplir ce travail.

Nous tenons à *exprimer notre grande gratitude* à *notre promoteur Monsieur RAMDANI Mohamed pour avoir accepter de nous encadrer tout au long de ce travail, pour sa disponibilité, son amabilité, ses conseils et suggestions et pour toute l'aide morale qui ne cesse de nous prodiguer. Il nous a prodiguer beaucoup de connaissances et conseils dans le domaine des réseaux de capteurs.*

Nos remerciements s'adressent à Madame la présidente AOUDJIT Rachida, par l'honneur qu'elle nous fais de présider ce jury de soutenance, nous lui exprimons notre gratitude profonde.

Nous tenons également à *remercier Monsieur SADI Samy qui a aimablement accepter d'examiner et de juger notre travail et pour l'intérêt qu'il y porte.*

Un grand merci pour nos familles, pour leurs soutien permanent, leurs présence et leurs encouragement.

Nous remercions profondément nos proches et très chers amis Adel, Lounis et Fateh qui nous en toujours soutenus et encouragé au cours de la réalisation de ce mémoire et tout au long de notre parcours universitaire, nous avons vraiment passé d'excellents moments ensemble.

Enfin, nous adressons nos vifs remerciements à *toute personne qui nous a aidé et apporté de plus pour arriver ici.*

Dédicace

A mes chers parents,

A mes frères et ma sœur,

A mes neveux,

A ma chère binôme et sa famille,

A toute ma famille et mes amis,

A tous ceux qui m'aiment et que j'aime .

Ouardia

Dédicace

A la mémoire de mon frère,

A la mémoire de mes grand-parents paternaux,

A la mémoire de mon grand-père maternel,

A mes chers parents, à ma sœur,

A ma meilleure amie, sa famille et tous mes amis,

A ma chère binôme et sa famille,

A tous ceux qui m'aiment et que j'aime.

Melissa

Résumé

Un réseau de capteur sans fils est un ensemble de nœud capteurs communicants entre eux via des liens sans fils, ayant des caractéristiques particulières et des ressources limitées en termes d'énergie, capacité de calcul et mémoire de stockage.

Les RCSF sont devenus de plus en plus un choix intéressant grâce à leurs performances montrées dans les différents domaines de leurs utilisations (facilité de déploiement, la mobilité,...). Ainsi, ils peuvent être implémentés dans des systèmes critiques différents comme les champs de bataille, le domaine de la santé, les bâtiments intelligents, les industries, la surveillance de l'environnement, etc.

Toutefois, en raison des ressources limitées de capteurs ainsi les environnements de déploiements hostiles et ouverts, ce type de réseau doit faire face à de nombreux types d'attaques qui peuvent nuire au travail des RCSF et empêcher leur bon fonctionnement. La sécurité de ces réseaux donc devient de plus en plus primordiale, et un grand challenge dans un domaine assez sensible que les RCSF. Par conséquent, des mécanisme de sécurité doivent s'adapter à la nature des réseaux de capteurs en raison des contraintes liées à ces derniers.

La gestion de clés cryptographiques dans un réseau de capteurs est donc un mécanisme important dans la configuration d'un système cryptographique, ainsi le choix d'une solution cryptographique revient à un grand challenge en raison des contraintes liés à ce domaine. En effet, à un niveau de sécurité équivalent, les schémas de chiffrement à clés publiques sont nettement plus lents que les schémas symétriques. Cependant, des avancées majeurs dans le cadre des cryptosystèmes asymétriques ainsi l'utilisation des courbes elliptiques en cryptographie ont marqué de bons résultats et de meilleures performances tout en s'adaptant à la nature des RCSFs et leurs caractéristiques.

Dans ce travail, nous nous concentrons d'abord sur le besoin d'assurer une meilleure sécurité et une bonne gestion de clés en étudiant les schémas proposés dans le domaine des réseaux de capteurs. Par la suite, nous proposons un nouveau schéma de gestion de clés dynamiques dans le cadre d'un réseau de capteurs à mobilité, en utilisant le système de cryptage intégré aux courbes elliptiques (ECIES) qui assure l'authentification et le chiffrement des données échangées, en prenant compte des contraintes et limites des réseaux de capteurs sans fil.

Mots clés :

RCSF, cryptographie des courbes elliptiques (ECC), gestion de clés, mobilité.

Table des matières

Table des figures

Liste des tableaux

Liste des abréviations

CH : Cluster head
DH : Diffie Hillman
ECC : Elliptic Curve Cryptograph
ECDH : Elliptic Curve Diffie-Hellman
ECDSA : Elliptic Curve Digital Signature Algorithm
ECIES : Elliptic Curve Integrated Encryption Scheme
FIFO : First In First Out
ID : identifiant d'un nœud
GPS : Global Positioning System
LEACH : Low Energy Adaptive Clustering Hierarchy
MAC : Message Authentication Code
micro-PKI : Micro Public Key Infrastructure
NI : Nœud intermediaire
NIS : Nœud isolé
NM : Nœud membre
NN : Nœud normal
NesC : langage dérivé du langage C
PKC : Public key cryptography
PKI : public key infrastructure
RCSF : réseaux de capteurs sans fil
SB : Station de Base
TDMA : Time division Multiple Access
TinyOs : Tiny operating system
TOSSIM : TinyOS SIMulator
QoS : Quality of Service

Introduction générale

Introduction générale

Au cours de ces dernières années, plusieurs progrès dans les domaines de microélectroniques, micro-mécaniques ainsi, les technologies de communications sans fils ont permit l'apparition et le développement des dispositifs embarqués miniaturisés appelés mini-capteurs, à faible cout et de plus en plus performants. Ces derniers, sont déployés facilement et en grand nombre dans la nature, coopérant entre eux sans aucune infrastructure externe et d'une façon autonome, communicant via des liens sans fils à courte portée. Constituant une infrastructure distribuée appelée réseau de capteurs sans fil (RCSF).

Un réseau de capteurs est constitué d'un nombre important de composants électroniques caractérisés par leurs tailles réduites, cette miniaturisation permet et facilite le déploiement dense de ces composants dans des endroits hostiles et difficiles d'accès.

De nos jours, les RCSFs sont omniprésents dans notre vie quotidienne grâce à la facilité, l'adaptation, ainsi que le caractère de déploiement et les liens de communications sans fil. On les trouve dans différentes applications de divers domaines, et souvent sont utilisés dans des applications critiques nécessitants un niveau de sécurité très élevé. Cela rend la sécurité un objectif crucial dans un tel réseau.

Pour atteindre ces objectifs dans un réseau assez sensible qu'un RCSF, la gestion de clés est une solution nécessaire dont les nœuds du réseau ont besoin pour garantir une meilleure sécurité des clés cryptographiques utilisées pour l'échange de données au sein du réseau. Souvent, les schémas de gestion de clés proposés dans la littérature sont basés sur l'utilisation de la cryptographie symétrique grâce à sa simplicité et sa facilité en la comparant à la cryptographie asymétrique qui est plus complexe et gourmande en ressources. Néanmoins, des études récentes ont montré l'efficacité des algorithmes asymétriques et leurs applicabilités en s'adaptant à la nature des réseaux sans fils.

Dans ce cadre, nous nous sommes particulièrement intéressées aux cryptosystèmes asymétriques basés sur la théorie des courbes elliptiques (ECC), en raison de leurs performances et les résultats obtenus en utilisant ces algorithmes. Ces derniers offrent des niveaux de sécurité supérieurs ou égales aux niveaux de sécurité donnés par les autres algorithmes asymétriques avec des tailles de clés nettement inférieures. D'où notre objectif de proposer un schéma de gestion de clés dynamiques pour les réseaux de capteurs sans fil à mobilité, en prenant compte les contraintes et ressources limitées de ce type de réseaux.

Organisation du mémoire

Ce mémoire est constitué de quatre chapitres organisés comme suit :

Pour mieux cerner les enjeux de notre étude, nous présentons dans le premier chapitre les concepts généraux des réseaux de capteurs sans fils, les domaines d'applications, les contraintes et caractéristiques des RCSFs. Nous présentons également les problèmes de sécurité dans de tels types de réseaux et la théorie de cryptographie basée sur les courbes elliptiques (ECC). Nous terminons ce chapitre par la classification des schémas de gestion de clés existant dans le monde des RCSFs.

Le second chapitre, concerne les schémas de gestion de clés dans les réseaux de capteurs. Nous détaillons quelque schémas de gestion de clés existants dans la littérature selon leurs méthodes cryptographiques utilisées.

Dans le troisième chapitre, nous décrivons les outils de développement ainsi les protocoles implémentés pour la mise en place de notre solution.

Le dernier chapitre est réservé à la simulation et la comparaison des résultats de notre solution avec les autres solutions proposées pour la gestion de clés dans le monde des réseaux de capteurs.

Enfin, notre mémoire s'achève par une conclusion générale qui résume les grands points qui ont été abordés, ainsi nos perspectives pour les travaux futurs dans le domaine des RCSFs.

Chapitre 1

Généralités sur les RCSFs

1.1 Introduction

Les progrès connus dans les domaines de micro-électronique, micro-mécanique, ainsi les technologies de communication sans fil, font l'apparition et le développement des dispositifs embarqués miniaturisés appelés mini-capteurs. Ces derniers ont connu un déploiement intense dans la nature. La coopération entre eux et leurs isolation de toute intervention externe, constituent un réseau de capteur sans fil devenu au fil du temps un domaine de recherche très actif.

1.2 Définition d'un capteur

Un capteur est un dispositif électronique miniaturisé, autonome, avec des ressources limitées. Capable de collecter des informations de son environnement, les traiter et les transmettre via des liens sans fil à d'autres entités de son entourage, sur des distances limitées.

FIGURE 1.1 – Exemple d'un nœud capteur

1.3 Architecture d'un capteur

Généralement, un nœud capteur est composé de quatre unités fondamentales [1] :

Unité d'alimentation : les capteurs sont alimentés par des batteries, et souvent ces batteries sont non rechargeables et irremplaçables, ce qui rend l'énergie, la ressource la plus précieuse dans les réseaux de capteurs, car elle influe directement sur la durée de vie des capteurs, ainsi celle du réseau entier.

Unité de détection : composée d'une unité de captage qui consiste à mesurer des grandeurs physiques et les transformer en signaux analogiques, et d'un convertisseur analogique numérique (CAN) qui transforme les signaux analogiques en un signal numérique, compréhensible par l'unité de traitement.

Unité de traitement : composée d'un processeur et d'une unité de mémoire réduite. Elle permet d'effectuer les traitements et les calculs, ainsi que le stockage de données.

Unité de communication : les capteurs sont capables de communiquer entre eux via des liens sans fils, grâce à une connexion radio, formant ainsi un réseau de capteurs sont fils. Cette unité est responsable de toutes les émissions/réceptions radio sur le canal de transmission.

Un nœud capteur peut également contenir, selon son domaine d'application d'autres modules supplémentaires tels qu'un système de localisation GPS, ou bien un système générateur d'énergie (cellule solaire).

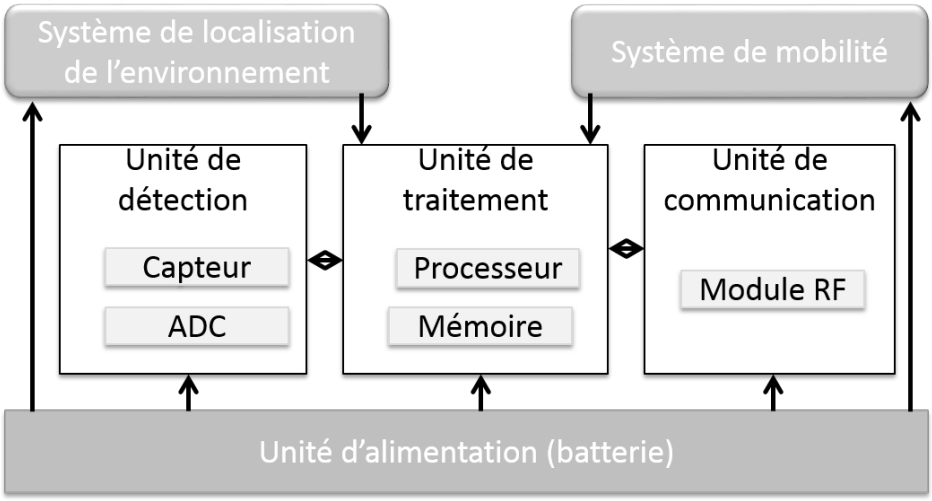

FIGURE 1.2 – Unités d'un capteur

1.4 Les réseaux de capteurs sans-fil

Un réseau de capteurs sans fil est un type particulier des réseaux ad'hoc, constitué d'un ensemble de nœuds capteurs (ou mote) dispersés aléatoirement ou d'une manière précise dans une zone géographique appelée « zone de captage », coopérant entre eux pour répondre à un objectif précis, de surveiller, détecter et traiter des phénomènes physiques captés, ainsi les envoyer à d'autre points de collecte appelés « puits » (Station de Base) qui est un nœud particulier du réseau doté d'une puissance de calcul supérieure et une alimentation quasi limitée. Par la suite, le puits (ou sink en anglais) transmet ces données à un ordinateur central «Gestionnaire de tâches », grâce à des ondes radio sur une distance limitée, pour l'analyse et prise de décision[2].

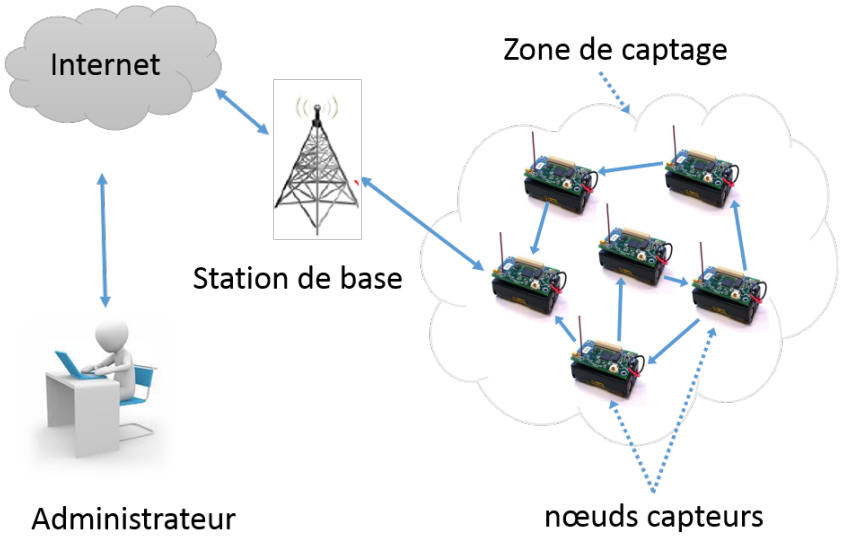

FIGURE 1.3 – Architecture d'un réseau de capteurs sans fil

1.4.1 Domaines d'application

De plus en plus, les RCSFs sont présents dans notre vie quotidienne grâce à leur facilité de déploiement, le cout réduit, la miniaturisation et le caractère de communication sans fil des réseaux de capteurs. Parmi les domaines où ces réseaux peuvent offrir des meilleures contributions , Nous citons :

Le domaine militaire : Le domaine militaire a été le moteur initial pour le développement des réseaux de capteurs. Les RSCFs sont utilisés pour surveiller les champs de bataille, la détection des intrusions, la communication et la reconnaissance. Les capteurs sont déployés dans des zones hostiles et isolés pour l'exploration, la surveillance des mouvements ou encore la détection de tout phénomène ou risque afin d'envisager certaines mesures lors d'une sortie d'exploration ou un ratissage dans ces terrains à risque.

Le domaine environnemental : Les capteurs peuvent être exploités pour détecter les catastrophes naturelles. En occurrence, on cite les feux de forets, les inondations, les éruptions volcaniques, les séismes, la sècheresse ou encore les avalanches, ...etc. La surveillance des déplacements de la faune dans l'environnement est aussi considéré, spécialement les animaux en voie d'extinction qui nécessitent une attention particulière et un soin intense pour préserver leur espèce rare.

Le domaine médical : Parmi ses applications, on peut citer la surveillance des malades à distance comme le cas du capteur de glucose implanté sous la peau ou un type de capteur accroché au poignet des malades.

On cite aussi, la surveillance des personnes âgées atteints de maladies chroniques dans le cadre de la géochronologie.

Le domaine commercial : Les capteurs peuvent être utilisés pour contrôler la qualité des produits, la gestion des inventaires, la surveillance de l'état du matériel, la gestion de la température de préservation des aliments ou le taux d'humidité.

Le contrôle d'édifice : L'utilisation des capteurs dans la surveillance des locaux en détectant et signalant tout évènement susceptible de nuire à la sécurité des locaux. Cela se fait par la détection des incendies ou tout changement de température pouvant nuire aux installations électriques ou équipements informatiques. C'est le cas considéré dans les DataCenters, où le système de refroidissement est un élément basique et crucial pour garantir la sécurité du matériel et une prolongation de sa durée de vie.

1.4.2 Les caractéristiques des RCSFs

Les principales caractéristiques des réseaux de capteurs sans fil sont [3] :

— Un grand nombre de nœuds : les RCSFs hébergent des milliers de nœuds voire des millions ;
— Accès sans fils ;
— Absence d'infrastructure fixe : Les RCSFs sont des réseaux ad'hoc ;
— Ressources limitées (calcul, énergie et mémoire) ;
— Topologie dynamique : La topologie d'un RCSFs peut être dynamique, ceci est due à la mobilité des nœuds qui peuvent s'attacher à des unités mobiles (GPS par exemple), ou due à l'ajout ou la suppression d'un ou plusieurs nœuds après le déploiement pour élargir le réseau ou pour supprimer les nœuds défaillants ;
— bande passante assez faible ;
— Sécurité physique limitée : Les réseaux de capteurs sont souvent déployés dans des environnements ouverts et sans surveillance. Par conséquent, ils peuvent facilement être interceptés et corrompus.

1.4.3 Les contraintes des RCSFs

Les réseaux de capteurs ne sont pas constitués de la même manière que les autres réseaux (les réseaux classiques), cette différence a causé des contraintes particulières à ce te type du réseau. Parmi ces contraintes, on distingue :

Les ressources limitées : Un réseau de capteurs sans fils est constitué d'un grand nombre de nœuds capteurs,ces derniers sont limités en termes d'énergie, de puissance de calcul et de capacité de communication.

— Limitation de mémoire et capacité de calcul : Un capteur est un composant électronique simple avec une petite quantité de mémoire et de puissance de calcul, prenant par exemple le capteur le plus couramment utilisé : MicaZ, ne dispose que de 4KO de RAM, 128 KO

de ROM et 512 KO de mémoire Flash, avec un micro-contrôleur de 8MHz-Atmel 128Mega.

- Limitation en énergie : Un nœud capteur est muni d'une source d'énergie limitée (batterie souvent non rechargeable).
- La limitation en puissance de transmission : la communication dans les RCSFs est une communication sans fil, dépend souvent de la coopération locale des nœuds en se basant sur l'acheminement de données par un routage multi-sauts.

Une communication non fiable : La communication dans un réseau de capteurs est une communication sans fils. De ce fait, chaque nœud capteur se trouvant dans une zone de captage peut écouter tous les messages transmis sur le canal de transmission, et donc l'application d'un bruit (le brouillage) sur ce canal rend le capteur incapable de transmettre ces messages car le canal peut apparaitre occupé en permanence. la grande menace de sécurité des nœuds capteurs est la nature du médium de communication sans fils, qui est distingué par sa nature non sécurisée. Dans un tel réseau, le médium sans fil est ouvert et accessible par n'importe qui au contraire des réseaux filaires. Cela facilite aux intrus d'intercepter des paquets valides et/ou d'injecter d'autres paquets malveillants en raison de la nature à accès ouvert du support de communication sans fil. En outre, endommager des paquets peut être du à une congestion élevée dans le nœud capteur. La communication elle même peut être non fiable dans des canaux fiables et cela est peut être du à une collision de paquets.

Des risques **inattendus** : Les capteurs sont déployés dans des zones géographiques distantes, ouvertes aux ennemis et sans aucune surveillance. Et donc les nœuds capteurs sont exposés aux attaques physiques, ce qui peut provoquer la destruction ou la capture du nœud, ou bien sa compromission par des entités malicieuses[4].

L'environnement hostile : les nœuds capteurs sont déployés souvent dans des environnements hostiles, ces capteurs sont exposés à la destruction ou la capture par des intrus. Ce type de déploiement facilite pour les attaquants l'accès physique aux capteurs qui sont déployés aléatoirement dans la nature. L'attaquant peut capturer le nœud, extraire des données légitimes (des clés cryptographiques par exemple) ou le détruire entièrement. De ce fait, les environnements hostiles sont un véritable challenge pour les chercheurs de sécurité[5].

Le tableau suivant résume les principales caractéristiques des capteurs les plus courants.

Capteur	MicaZ	WSN430	TelosB	Imote2
Processeur	Atmel AT- Mega 128L	TI MSP430	TI MSP430	Intel PXA271 XScale
Vitesse de processeur	16MHz	8 MHz	8MHz	13-416MHz
RAM	4KO	10KO	10KO	
Espace programmable	128KO	48 KO	48 KO	32KO
Flash	512 KO	1 MO	256 KO	32KO
Batterie	2xAA	PoLiFlex	2/3 A	3x AAA
Fréquence (MHz)	2400-2483	315/433/ 868/915	2400-2483	2400-2483
Débit de données (kb/s)	250	250	250	500
Système supportés	TinyOs,SOS MantisOS, Nano-RK	TinyOS, Contiki, FreeRTOS	Contiki,SOS TinyOS, MantisOS	Microsoft .NET, MicroLinux, TinyOS

TABLE 1.1 – Caractéristiques de certains capteurs disponibles sur le marché

1.4.4 Les facteurs de conception des RCSFs

Pour concevoir un réseau de capteurs, plusieurs facteurs doivent être pris en compte [1]. On considère :

1. La tolérance aux pannes : Le fonctionnement de capteurs aux cours de cycle de vie du réseau peut être interrompu. Ces pannes ne doivent pas affecter le fonctionnement global de réseau, ainsi que ce dernier doit être capable de maintenir ses fonctionnalités malgré la défaillance de certains nœuds.
2. La gestion et consommation d'énergie : Généralement, les nœuds capteurs sont alimentés par des batteries et souvent leurs rechargement et remplacement est impossible. Ce qui rend la durée de vie du nœud fortement dépendante de la durée de vie de sa batterie. De ce fait, l'énergie est une ressource critique dans les réseaux de capteurs sans fil.
3. La limitation des ressources : Les capteurs sont des composants électroniques limités en mémoire, en énergie, aussi en puissance de calcul et en capacité de transmission.
4. Passage à l'échelle (scalabilité) : Dans un réseau dense le nombre de nœuds capteurs peut atteindre des milliers voire des millions. Ce nombre

important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que la station de base soit équipée d'une mémoire suffisante pour le stockage des informations reçues.

5. L'agrégation de données : La transmission de données est une tâche gourmande en ressource (énergie). Cependant, une approche répandue en se basant sur la diminution de la quantité des données transmises vers la station de base, qui consiste à agréger les données en éliminant les données redondantes au niveau des nœuds intermédiaires.
6. Le déploiement : Souvent les nœuds capteurs sont dispersés aléatoirement dans des zones hostiles, distantes et sans aucune surveillance, les capteurs doivent être conçus pour résister aux différentes conditions climatiques (la chaleur, le froid, la pression, etc.).
7. La mobilité : L'introduction des éléments mobiles dans les capteurs ont rendu ces derniers capables de se déplacer dans leur environnement et d'atteindre de nouvelles positions dans le réseau, ce qui engendre le changement de la topologie. Aussi, la mobilité représente une autre source de dissipation d'énergie du nœud et influence sa durée de vie, ainsi que la durée de vie du réseau. En conséquent, les RCSFs nécessitent des mécanismes de sécurité plus adéquats afin de répondre aux contraintes liées à la mobilité des nœuds.
8. Le routage dans les RCSF : Le routage est la procédure d'acheminement des informations d'un nœud source vers un nœud destinataire à travers un réseau de connexion. Chaque nœud est susceptible d'être à la disposition des autres nœuds pour participer à la transmission et retransmission des informations émises sur le réseau par un ou plusieurs nœuds n'ayant pas la possibilité d'atteindre directement la destination. Comme les RCSFs ont des caractéristiques particulières en ce qui concerne la capacité de calcul, de stockage et l'énergie des nœuds capteurs, donc le routage est une tâche bien compliquée qui nécessite la collaboration de tous les nœuds appartenant à un même réseau.

Les types de protocoles de routage dans les RCSFs La classification des protocoles de routage peut prendre différents critères [6]. On considère alors les classifications suivantes :

(a) Protocoles basés sur la structure du réseau :

i. Routage à Plat ou centré : Tous les nœuds ont les mêmes rôles, chaque nœud communique avec son voisin direct. Cependant, les nœuds les plus proches de la base épuiseront leurs batteries rapidement, vu que chaque donnée acheminée vers la station de base passera obligatoirement par ces nœuds là.

ii. Hiérarchique : Application du principe de clustering où chaque cluster contient un cluster-Head. Ce dernier reçoit toutes les données des nœuds appartenant à son cluster, les traitent (suppression des redondances par le principe d'agrégation, etc) pour qu'il

puisse les acheminer à la station de base. Dès lors que le réseau augmente, l'élection du cluster-Head devient plus gourmande en ressource.

(b) Protocoles basés sur la localisation : Identifié l'emplacement géographique des nœuds afin de calculer le plus court chemin depuis le nœud émetteur vers le nœud de destination, plus économique en termes d'énergie, d'un coté, puisque évite les méthodes déterministes et probabiliste pour la recherche des routes. La localisation évite aussi le broadcast des requêtes, elle permet de les envoyer qu'aux nœuds spécifiques. De l'autre coté, le GPS consomme beaucoup d'énergie.

(c) Protocoles basés sur le mode de transmission :

i. Routage proactif : Une table de routage détermine le chemin à emprunter par chaque donnée.

ii. Routage réactif ou à la demande : Des routes crées à la demande ce qui permet d'économiser de l'énergie.

(d) Protocole de routage basé sur la QoS : Essayer de satisfaire les métriques : délai de transmission, l'énergie consommée, ...

(e) Le routage hybride (fusion de proactif et de réactif) : Des tables de routages sont établies avec les nœuds voisins dont le saut est au maximum deux, au-delà c'est un routage réactif.

1.5 La sécurité dans les RCSFs

Souvent, les RCSFs sont utilisés dans des applications critiques, ce qui nécessite un niveau de sécurité élevé. De ce fait, des mécanismes de sécurité doivent être mis-en place afin de garantir l'intégrité des données échangées, leur authentification, la disponibilité, la confidentialité ainsi la non-répudiation et le contrôle d'accès aux données.

1.5.1 Les défis de sécurité

La sécurité est un grand challenge dans les réseaux de capteurs sans fil. En effet, trois défis majeurs auxquels une solution de sécurité doit s'adapter, ces derniers sont :

— Pouvoir fournir une solution qui minimise la consommation d'énergie et maximise les performances de sécurité, vue que la ressource d'énergie est une ressource critique, toute solution de sécurité à un impact direct sur l'énergie de nœud.

— Répondre à la problématique relative à la mobilité des nœuds.

— Trouver une alternative aux schémas de sécurité des réseaux filaires, à cause de leurs inapplicabilité sur les RCSFs due au caractère sans-fil de ses réseaux.

1.5.2 Les attaques de sécurité dans les RCSFs

Les réseaux de capteurs sans fil sont plus vulnérables aux attaques de sécurité que les réseaux filaires [7] [8] en raison de la nature du médium de transmission sans fils. En effet, les RCSFs sont déployés dans des fréquences radio ouvertes, ce qui rend l'écoute du médium de communication assez facile. De plus, les RCSFs ont une vulnérabilité supplémentaire car les nœuds sont souvent déployés dans un environnement hostile ou difficile où ils sont pas physiquement protégés. Nous avons opté à la classification des attaques les plus fréquentes en deux grands types d'attaques : les attaques actives et les attaques passives [5][9] .

1.5.2.1 Attaques passives du réseau :

Ce type d'attaque se produit lorsque un attaquant surveille et étudie le trafic réseau qui le traverse mais sans altérer son fonctionnement (sans modifier les données). Le but de l'attaquant est d'intercepter les données échangées sur le réseau sans bousculer son fonctionnement [9]. Plusieurs attaques dérivent de ce type, nous citons :

1. Monitor& eavesdropping (L'écoute du réseau) : C'est l'attaque la plus courante et la plus facile contre la vie privée, l'attaquant écoute le trafic via lequel les données sont transférées. Et vu la nature du médium, il peut facilement récupérer le contenu des communications échangées, l'écoute (eavesdropping) peut alors agir facilement contre la protection de la vie privée.
2. Trafic analysis (L'analyse du trafic) : Même lorsque les données transférées sont cryptées, il reste encore possible d'analyser le pattern de communication. Cette attaque peut provoquer une attaque de déni de service (DoS) et également une attaque sur les nœuds qui jouent un rôle important dans le réseau, en présence d'un attaquant qui analyse leurs activités. C'est une attaque passive sur les réseaux de capteurs sans fils en terme de confidentialité.
3. Camouflage adversairies : Un attaquant peut insérer un nouveau nœud (malveillant) ou compromettre un nœud du réseau, dans le but de faire participé ce nœud au routage de paquets au sein du réseau, comme étant un nœud normal, donc il construit de mauvaise route et achemine les paquets de manière erronées en effectuant l'analyse de confidentialité. Camauflage adversairies est aussi une attaque passive sur le réseau de capteurs sans fil en terme de confidentialité.

1.5.2.2 Attaques actives :

Dans ce type d'attaque, l'attaquant supprime ou modifie les informations interceptées pour perturber, ralentir et modifier le fonctionnement du réseau, ou le bloquer carrément et produire un déni de service (DoS). Parmi ces attaques, nous citons :

1. Attaque physique d'un nœud : Les nœuds capteurs sont déployés dans des environnements accessibles aux ennemis , ce qui les rend vulnérables aux attaques physiques que se soit pour récupérer le matériel cryptographique tel que les clés, reprogrammer le capteur attaqué ou retirer le capteur du réseau, extraire les informations sensibles ou détruire le capteur carrément[9].
2. Attaque de déni de service (DoS) : Ce type d'attaque consiste à réduire les performances du réseau en créant une surcharge sur le réseau en produisant des opérations malicieuses pour perturber le fonctionnement des nœuds.
3. Les attaques de routage : Ces attaques sont appliquées sur les protocoles de routage qui permettent l'acheminement des paquets d'un nœud source vers la destination, parmi ces attaques on définit :
   (a) Spoofed routing information : L'attaquant se met entre le nœud émetteur de l'information à transmettre et la station de base, dans le but d'altérer les informations routées et déséquilibrer le trafic réseau.
   (b) selective packet forwarding : Dans un RCSF, on suppose que tous les nœuds transmettent les paquets de données reçues, mais dans cette attaque, l'attaquant crée un nœud malveillant qui ne transmet pas la totalité des messages qu'il a reçu.
   (c) sinkhole/Blackhole attack : Dans ce type d'attaque, l'attaquant introduit des nœuds compromis avec des capacités supérieures, nœuds beaucoup plus performants, attirant plus les nœuds voisins. Ces nœuds transmettent leurs données via ces nœuds compromis, ces derniers peuvent modifier, supprimer et retransmettre de faux paquets vers le destinataire.
      — **Sinkhole attack :** C'est quand l'attaquant empêche la station de base d'obtenir une collecte complète des informations dans le réseau, ce qui provoque une sérieuse attaque dans la couche supérieure "application". L'attaquant peut facilement attirer tous le trafic dans une zone spécifique et fait de sorte que le nœud malveillant apparait comme étant un nœud du réseau plus attirant que les autres [10] et par la suite il fait une suppression sélective des données.
      — **Blackhole attack :** Un nœud falsifie les informations de routage afin que les données soient dirigées vers lui, sans les retransmettre par la suite. Il est possible aussi que le nœud se place dans un endroit stratégique du réseau et supprime par la suite tous les messages qu'il doit transmettre.
   (d) sybil attack : Dans un réseau de capteurs, tous les nœuds coopèrent entre eux et fonctionnent en collaboration afin d'accomplir les tâches souhaitées. Un attaquant introduit un nœud malveillant qui apparait comme étant un ensemble de nœuds utilisant les identités des nœuds légitimes du réseau. Cela influe le routage de données, l'agrégation et l'envoie de fausses données.

(e) Wormhole attack : L'attaquant reçoit les paquets du réseau d'une location, et les retransmettre vers une autre location.

(f) Hello flood attack : Des protocoles utilisent les paquets hello en diffusion pour la découverte du voisinage et la création de la topologie du réseau. Donc on trouve un type d'attaque par la diffusion de plusieurs messages Hello jusqu'à inondation du réseau et empêcher d'autre communications. Un nœud capteur sera convaincu que le nœud émetteur de message est un nœud du réseau et lui est très proche et il le choisit comme son cluster head, afin de lui envoyer ses données pour les acheminer vers la station de base. Cette attaque est la plus simple à produire dans les réseau de capteurs .

4. Node replication attacks (attaques de réplication de nœuds) : Dans cette attaque, l'attaquant ajoute un nœud capteur au réseau existant en copiant l'ID du nœud de capteur déjà existant . Cette attaque réduit les performances du réseau en utilisant une corruption de paquets ou leurs mauvais acheminements.
5. False Node : Cette attaque se produit quand un attaquant ajoute un nœud malicieux dans le réseau, ce dernier est généralement doté de meilleures performances pour attirer plus les messages, vu que certains algorithmes de routage prennent en considération la puissance en termes d'énergie. Il introduit des données erronées dans le réseau qui seront transmises et échangées au sein de réseau. La grande majorité des attaques dans les RCSFs sont due à cause de ce genre d'attaque (les données erronées) .
6. Node outage : Cette situation est introduite lorsque un nœud cesse de fonctionner. Dans ce cas, le CH cesse de fonctionner, les protocoles du réseau doivent être assez robustes afin de minimiser l'effet de l'attaque d'outage en proposant un autre itinéraire pour l'acheminement de données.
7. Node Malfunction : Un nœud défectueux génère des données inexactes et incorrectes altérant ainsi l'intégrité des données, en particulier s'il s'agit d'un nœud d'agrégation tel que le Cluster-Head.
8. Passive information gathering : Un intrus doté d'un récepteur suffisamment puissant et d'une antenne bien conçue peut facilement capter les données. Il intercepte les messages contenant l'emplacement physique des nœuds, ce qui lui permettra de localiser les nœuds du réseau et de les détruire.

1.5.3 La cryptographie

La cryptographie est un mécanisme de sécurité qui est relativement beaucoup plus sûr et fiable. Basée sur l'utilisation des méthodes mathématiques pour la transformation des messages originaux en une suite de données incompréhensible et qui ne peuvent pas être interpréter directement par des tierces parties.

La cryptographie est mise en place afin de garantir la confidentialité, l'intégrité et l'authentification de données échangées. Dans cette optique, nous citons les principaux objectifs de sécurité et de cryptographie [5] :

- La confidentialité : Garantir que les données d'un nœud ne sont accessibles qu'à son nœud destinataire. La confidentialité est un issue important dans la sécurité du réseau, un nœud capteur ne doit pas divulguer ses données que aux nœuds destinataires .
- L'authentification : Les attaques dans un RCSF ne consiste pas seulement en l'altération de paquet mais aussi en l'injection de nouveaux paquets erronés. L'authentification de données permet d'assurer et de vérifier l'identité de nœud source/destinataire. Elle est assurer à travers des mécanismes symétriques ou asymétriques lors de l'envoie et/ ou réception de paquets par le partage de clés secrètes entre les nœuds.
- L'intégrité : L'intégrité est nécessaire pour assurer la fiabilité de données dans le réseau et d'assurer que les données ne sont pas altérées pendant leurs transmissions. Même si la confidentialité du réseau est garantie, il reste toujours possible de compromettre l'intégrité. Cette dernière est mise en menace quand :
  - Un nœud malicieux est présent sur le réseau et injecte des données erronées.
  - Des conditions non stables en raison de perte de données ou des données endommagées (due au canal sans fil)
- La disponibilité de données : Définit la capacité du réseau à assurer ses services pour maintenir son bon fonctionnement, en garantissant aux parties communicantes la présence ainsi l'utilisation de l'information au moment souhaité.
- Non répudiation : Prévenir que le destinataire nie la communication (un nœud ne peut pas nier l'envoi ou la réception d'un message).

Il existe deux systèmes de cryptographie, la cryptographie symétrique qui utilise une seule clés dite symétrique pour le chiffrement et le déchiffrement. Dans ce cas, les communicants se mettent d'accord sur une clé secrète que seul eux connaissent, il leur faut alors un moyen sûr pour s'échanger la clé. La cryptographie asymétrique, qui est à l'opposé de la première, utilise une paire de clés : clé privée/clé publique. Tel que, la clé pour coder le message est connue de tout le monde mais ne permet pas d'en déduire la clé qui permet de décrypter ce message. Cette clé n'est connue que par le destinataire.

Une comparaison entre ces deux mécanismes est présentée dans la figure suivante :

Cryptographie Symétrique	Cryptographie Asymétrique
Simplicité des calculs	Calculs plus complexes (gourmande en ressources)
Grand espace de stockage : n(n-1)/2 clés	Faible espace de stockage : 20 clés uniquement
Basé sur la pré-distribution de clés	Ne nécessite pas une pré-distribution de clés

TABLE 1.2 – Cryptographie Symétrique VS Cryptographie Asymétrique

On considérait la cryptographie symétrique comme étant le seul système applicable sur les RCSFs, cependant de ressentes études se tournaient vers la cryptographie asymétrique et leurs applicabilités sur ce type de réseau. Ce contournement est due à l'efficacité de cette dernière et le haut niveau de sécurité qu'elle offre en prenant aussi en considération les points cités dans le Tableau 1.2. Le résultat de ces études est une démonstration de l'efficacité d'utilisation des solutions à clé publique dans les réseaux de capteurs sans fil et leur adaptabilité pour un tel type de réseau.

1.5.4 Les attaques cryptographiques

La cryptographie aussi menacée par différentes attaques, dans le but de récupérer le plus grand nombre d'informations sur le système cryptographique utilisé et les secrets échangés entre les différentes entités[11]. Parmi ces attaques nous citons :

- L'attaque à texte chiffré seulement : Le cryptanalyste possède les textes chiffrés avec le même algorithme de plusieurs messages, l'attaquant doit être capable de fournir un algorithme qui permettra de déchiffrer et de décrypter les prochains messages chiffrés.
- L'attaque à texte en clair connu : Le cryptanalyste possède non seulement les textes chiffrés de plusieurs messages mais aussi les textes en clair correspondants, dans le but d'arriver à extraire la clé de chiffrement utilisée pour chiffrer ces messages, ce qui lui permettra de déchiffrer les nouveaux messages chiffrés avec la clé retrouvée.
- L'attaque à texte en clair choisi : Cette attaque ressemble à l'attaque à texte en clair connu mais elle tire avantage du fait que le cryptanalyste est capable de choisir les textes en clair ce qui lui donneront le plus d'informations sur la clé de chiffrement.
- L'attaque à texte en clair choisi adaptive : Considérée comme un cas particulier de l'attaque à texte en clair choisi. Le cryptanalyste peut non seulement choisir les textes en clair mais il peut également adapter ses choix en fonction des textes chiffrés précédents.
- L'attaque à texte chiffré choisi : Le but de l'attaquant est de retrouver la clé a l'aide d'un dispositif qui fait le déchiffrement.
- L'attaque à clé choisie.

1.5.5 La cryptographie des courbes elliptiques

Les algorithmes de cryptographie asymétriques sont généralement plus couteux par rapport à ceux de cryptographie symétriques, ainsi les clés générées sont plus longues et nécessitent des calculs importants. De ce fait, la cryptographie asymétrique est moins souhaitée dans le cas des réseaux de capteurs sans fil à cause des ressources limitées de ces derniers. En effet, un nœud capteur est muni d'une petite taille mémoire, une faible puissance de calcul et d'une ressource énergétique critique, l'utilisation de la cryptographie

asymétrique est donc rarement utiliser dans ce domaine. Cependant, des optimisations de ces algorithmes rendu possible l'implémentation de tels algorithmes dans les réseaux de capteurs. La cryptographie des courbes elliptiques est un nouveau cryptosystème asymétrique fondu sur le problème de logarithme discret sur les courbes elliptiques, il permet d'améliorer les primitives cryptographiques existantes en réduisant la taille des clés [12]. Soient F_q un corps fini à q éléments et $\bar{F}_q$ la clôture algébrique de F_q (tout polynôme de degré supérieur ou égal à 1, à coefficients dans $\bar{F}_q$, admet au moins une racine dans $\bar{F}_q$). Une courbe elliptique E est ensemble de couple (x,y) appartenant à $F_q \times \bar{F}_q$, vérifiant ainsi l'équation de Weierstast E suivante :

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x$$

$$+a_6$$

Pour leur usage en cryptographie, les facteurs a_1, $a_2$et a_3 doivent avoir une valeur nulle. Avec les cryptographes a_4 est devenu a et$a_6$est devenu b, d'où l'équation devienne $E : y^2 = x^3 + ax + b$, où $a, b \epsilon Fq \times \bar{F}_q$ sont des constantes telles que $4a^3 + 27b^2 \neq 0$. Et donc une courbe elliptique E sur un corps fini F_q se compose de l'ensemble de toutes les paires de coordonnées affines (x, y) pour$x, y \epsilon F_q$ qui satisfont une équation de la forme ci-dessus avec un point à l'infini [13] : $\mathcal{O}(x, y) \epsilon Fq \times \bar{F}qU\mathcal{O}$.

Protocoles cryptographiques basée sur les courbes elliptiques

Pour utiliser les courbes elliptiques, il faut trouver un problème difficile à résoudre comme la factorisation (le cas de RSA : factorisation d'un nombre en ses facteurs premiers). Considérant l'équation suivante : $Q = KP$ où Q, P appartiennent à la courbe E $(Q, P \epsilon E$) et $K < P$, il est facile de calculer Q connaissant Ket P, mais il est extrêmement difficile de trouver K si on connait Q et P : il s'agit du Problème de Logarithme Discret dans les courbes elliptiques ($LogP(Q)$) [14].
Vue la complexité de résoudre ce problème, plusieurs protocoles de cryptographie ont été proposé. Nous citons les principaux algorithmes qui se basent sur la théorie des courbes elliptiques.

1. L'échange de clé de Diffie Hellman
   L'échange de clé de Diffie Hellman basé sur les courbes elliptiques ou comme souvent abrégé ECDH (***E****lliptic* Curve Diffie Hellman) est un protocole d'échange de clés anonyme qui permet à deux pairs, chacun ayant un couple de clé privée/publique basé sur les courbes elliptiques, d'établir un secret partagé à travers un canal de communication non sécurisé [14]. Ce secret partagé peut être employé directement comme une clé de chiffrement ou être utilisé pour dériver une autre clé qui, à son tour, peut être utilisée pour chiffrer les communications.

Soient deux participants à une conversation Alice et Bob, ils rendent publique une courbe elliptique E définie sur un corps fini $\mathbb{Q}$ et un point $P(x_1, y_1)$ de cette courbe ($P \epsilon E(\mathbb{Q})$).
Alice choisi un élément $x \epsilon \mathbb{Q}\text{-}\{\mathcal{O}\}$ qui sera sa clé privée. Alice génère alors sa clé publique : $X = xP$ .....(1)
Bob choisit y, $y \epsilon \mathbb{Q}\text{-}\{\mathcal{O}\}$, qui sera sa clé privée. Bob génère alors sa clé publique$Y = yP$ .....(2)
Ensuite, ils échangent les résultats de (1) et (2), puis ils génèrent une clé secrète comme suit :
— Alice calcule : $K = xY$.....(3)
— Bob calcule :$K' = yX$.....(4)
On obtient$(3) \Longleftrightarrow (4) : K = xY = x(yP) = yX = K'$ où K ou K' est le secret commun, et si une entité malveillante intercepte les clés publiques X et Y et elle connait P, elle ne peut pas découvrir la clé secrète car il lui faut connaitre x ou y hors que pour cela des calculs de logarithme discrets doivent être fait, ce qui est très difficile.

2. ELLIPTIC CURVE INTEGRATED ENCRYPTION SCHEME (ECIES)

   La cryptographie par les courbes elliptiques n'est pas limitée pour les échanges de clés symétrique, mais aussi utilisée pour chiffrer directement les données.**ECIES** (**E**lliptic **C**urve **I**ntegrated **E**ncryption Scheme) est un algorithme de chiffrement à clé publique basé sur les courbes elliptiques qui est une variante de celui d'El Gamal standardisée. Ce système dérive une clé de chiffrement en bloc et une clé MAC à partir d'un secret commun, les données sont chiffrées d'abord avec un chiffrement symétrique, puis le texte chiffré est MAC (Message Authentication Code) chiffré sous un schéma d'authentification et enfin, le secret commun est chiffré avec la partie publique d'une paire de clés publique/privée . ECIES utilise différents types de fonctions : Fonction de dérivation de clé (KDF), schéma de chiffrement symétrique, fonction de hachage. Ce protocole est détaillé dans les chapitres suivants.

3. Elliptic Curve Digital Signature Algorithm

   ECDSA est un algorithme de signature à clé publique, proposé en 1992 par Scott Vanstone, en réponse à un appel d'offre pour les signatures numériques du NIST(National Institute of Standards and Technology). La méthode de signature de cet algorithme est une variante de l'algorithme de signature DSA (Digital Signature Algorithm), basée sur les courbes elliptiques avec une variation légère de la méthode de signature EL Gamal.

   Exemple : Soient E une courbe elliptique d'ordre p, Q un point appartenant à cette courbe et H la fonction de hachage dans $[1, p-1]$. supposons que Alice doit signer et envoyer un message $m \epsilon [1, p-1]$. Pour signer ce message Alice doit créer une clé privée K_A et une clé publique $P_A = K_A.Q$. Ainsi, Bob doit disposer de la clé publique de Alice pour qu'il puisse vérifier authenticité de la signature. Ensuite, les deux participants Alice et Bob suivent les étapes résumées dans le tableau suivant :

Signature cote Alice	Vérification cote Bob
Calculer $e = H(m)$	Calculer $e = H(m)$
choisir $K\epsilon[1, p]$ au hasard	Vérifier que $r, s\epsilon[1, p-1]$
Calculer $K.Q$	Calculer $w = s^{-1} \text{mod } p$
Convertir $x(KQ)$ en un entier $\bar{x}$	Calculer $t_1 = ew$,$t_2 = rw \text{ mod } p$
Calculer $r = \bar{x} \text{ mod} p$	Calculer $X = t_1Q + t_2 + P_A$
Calculer $s = K^{-1}(e + K_A r) \text{mod } p$	Convertir $x(X)$ en un entier $\bar{x}$
Envoyer(m, r, s)	Accepter si $\bar{x} = r \text{ mod } p$

TABLE 1.3 – Les étapes de l'algorithme ECDSA

1.5.6 Schéma de gestion de clés

La gestion de clés est un mécanisme important dans le processus de configuration d'un système cryptographique. La conception d'un système de gestion de clés dans un RCSFs est un grand défi vue les contraintes liées à ce domaine, ainsi le choix d'une solution cryptographique revient à un autre défi. Il existe plusieurs classifications et distribution de clés dans la littérature, nous avons choisi de faire une classification qui regroupe l'ensemble des schémas de gestion de clés en deux familles, la première famille utilise la cryptographie symétrique et la deuxième utilise la cryptographie asymétrique comme le présente la figure ci-dessous :

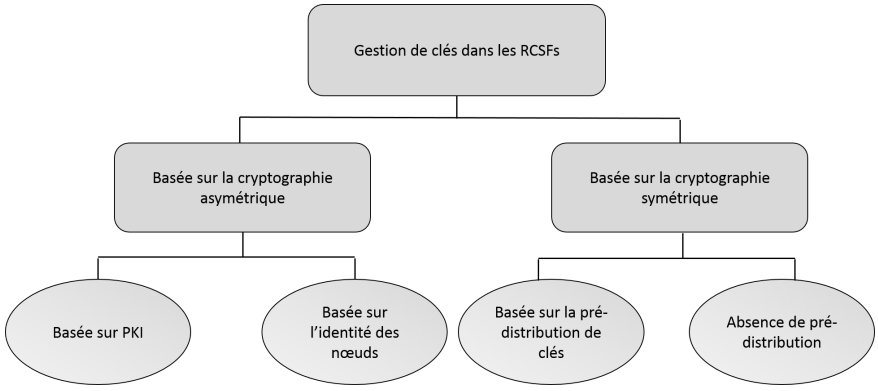

FIGURE 1.4 – Classification des schémas de gestion de clés pour les réseaux de capteurs

1.6 Conclusion

Les réseaux de capteurs sans fil sont un domaine de recherche actif, en plein développement. Ils sont omniprésents dans notre vie quotidienne et souvent, ils sont utilisés dans des applications critiques nécessitant une sécurité importante.

Dans ce chapitre nous avons présenté les concepts généraux des RCSFs, nous avons décris les grands problèmes qui limitent les RCSFs : le besoin de sécurité dans de tels systèmes, ainsi les mécanismes de sécurité adaptés aux RCSFs. Nous consacrons le chapitre suivant pour présenter le mécanisme de gestion de clés, en détaillant les schémas existants dans la littérature.

Chapitre 2

État de l'art sur la gestion de clés dans les RCSFs

2.1 Introduction

La cryptographie à clé secrète ou aussi appelée cryptographie symétrique est la méthode de sécurité la plus préférable et dominante dans les RCSFs grâce à ces calculs rapides et non couteux de clés, ainsi la taille de ses dernières est relativement petite. Tandis que, la cryptographie asymétrique est une autre solution qui fournit des mécanismes plus sûr et fiables pour l'authentification et la distribution de clés, mais qui exige un espace mémoire important et une haute puissance de calcul, ce qui la rend inapproprié pour les réseaux de capteurs sans fil qui se caractérisent par leurs ressources limitées[15]. Cependant, durant ces dernières années, les chercheurs ont montré la possibilité d'appliquer de tels systèmes de cryptographie aux réseaux de capteurs. En effet, des améliorations des algorithmes de cryptographie, en tenant compte des contraintes des RCSFs, ont donné de bons résultats. Des études montrent que la cryptographie basée sur les courbes elliptiques (ECC) donne des résultats significatifs par rapport aux autres systèmes de sa catégories (RSA, DH, ...etc.) et avec une taille de clés beaucoup plus réduite avec un cout de calcul minimisé. La cryptographie asymétrique possède des avantages importants face à la cryptographie à clés secrète, ce qui la rend un domaine de recherche très actif et attirant pour les spécialistes de sécurité de nos jours. La gestion et l'établissement de clés cryptographiques est un service primordial afin de garantir un haut niveau de sécurité au sein d'un RCSF, cependant cette tâche est très difficile vue les contrainte et caractéristiques d ce type du réseaux. Dans ce chapitre, nous présentons des schémas de gestion de clés existants dans la littérature.

2.2 Problématique

La gestion de clés est un processus très important pour le système cryptographique, elle permet de produire des clés, de les distribuer, les enregistrer, les transférer, les renouveler ou de les supprimer. Pour cela, la gestion de clés doit répondre à quelques problématiques :

- Comment ajouter un nœud au réseau et pouvoir établir une communication avec lui ?
- Comment détecter une défaillance d'un nœud ou l'intrusion d'un nœud et pouvoir révoquer sa clé du réseau ?
- Comment assurer le renouvellement des clés à chaque écoulement d'un délai fixe ?
- Comment assurer une gestion meilleure de clés dans les réseaux de capteurs mobiles ?

2.3 Pourquoi la gestion de clés ?

La gestion de clés est considérée comme étant une tâche importante et difficile à mettre en place au sein d'un réseau de capteur sans fils, cela est dû aux contraintes et caractéristiques de ce type de réseaux, ainsi la nature de déploiement des nœuds et l'environnement naturel dont ils sont déployés. Un rôle capital est alors joué par le mécanisme de gestion de clés dans la sécurité de tout système basé sur la communication, en vue des points suivants :

- La gestion des clés est très importante dans le processus de cryptographie.
- Que se soit pour la cryptographie, la signature numérique ou MAC, l'utilisation de bonnes clés cryptographiques est l'élément de base.
- La sécurité des clés implique la sécurité du réseau entier.
- La confiance accordée aux informations reçues (Assurer une fiabilité en vers les informations reçues) .
- Une intrusion durant l'échange de clés signifie l'échec de la sécurité des échanges dans un système.
- Le renouvellement des clés est très important et essentiel dans les RCSFs.
- La suppression de clés suite à l'isolation d'un nœud susceptible d'être malveillant.

2.4 Protocoles de gestion de clés

La sécurité du réseau est basée sur la gestion de clés qui est difficile à implanter à cause des contraintes liées à ce type du réseau. Il existe plusieurs classifications et distribution de clés dans la littérature, nous avons choisi de faire une classification qui regroupe l'ensemble des schémas de gestion de clés en deux familles, la première utilise la cryptographie symétrique et la deuxième utilise la cryptographie asymétrique. Dans ce qui suit nous détaillerons les principaux schémas de la figure 1.5 présentée dans le chapitre précédant.

2.4.1 Schémas symétriques

La plupart des protocoles de gestion de clés dans la littérature se base sur des mécanismes de chiffrement symétrique car ils sont moins couteux comparant aux autres et plus facile à implémenter [16]. La solution symétrique est réalisée en trois phases principales :

- Pré-distribution de clés (Key predistribution) : Les clés sont stockées dans la mémoire avant le déploiement constituent le porte-clés (key ring) du nœud. S'il existe une clé commune entre deux nœuds, ils peuvent alors créer un lien sécurisé entre eux.
- Découverte de la clé commune (Shared-key discovery) : Le protocole de communication après le déploiement est chargé de découvrir la clé commune entre deux nœuds voisins.
- Etablissement ou constitution d'un chemin sécurisé par des clés (path-key establishment) : S'il n'existe pas de clé commune entre deux nœuds voulant communiquer, il faut alors trouver un chemin sécurisé entre eux. Ce chemin passe par un ensemble de nœuds qui contient déjà des liens sécurisés. Une fois ce chemin établi, les deux nœuds peuvent l'utiliser pour communiquer en toute sécurité.

La figure suivante résume la classification des schémas de gestion de clés selon l'utilisation des systèmes cryptographique symétriques :

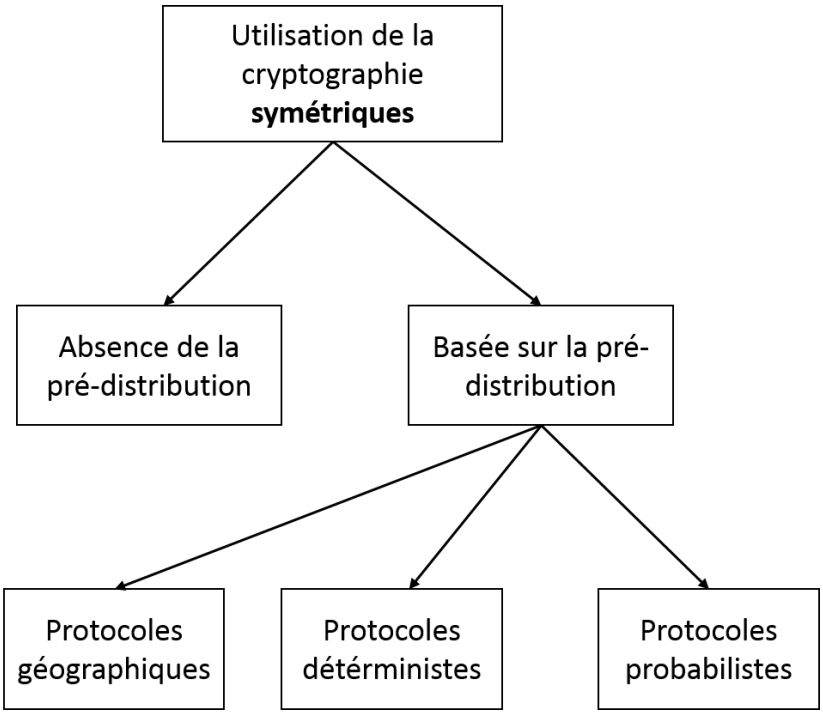

FIGURE 2.1 – schémas de gestion de clés symétriques

2.4.1.1 Absence de pré-distribution

Les schémas de cette catégorie ne prend aucune considération de pré-distribution de clés. En effet si un attaquant ne connait pas où et quand les nœuds sont déployés, il lui sera difficile de lancer une attaque active. Mais, il reste souvent présent après le déploiement et il peut surveiller les communications dans le réseau [17].

2.4.1.2 Basé sur la pré-distribution

Les schémas basés sur la pré-distribution reposent sur le principe de distribution de clés statiques, où les nœuds sont pré-chargés par des clés dans leurs mémoires avant le déploiement dans la nature, d'une manière que chaque nœud a une clé commune qu'il partagera avec ses voisins afin d'établir des communications secrètes. La plupart des schémas proposés pour les RCSFs sont basés sur la pré-distribution. Nous pouvons classer les schémas de cette catégorie en trois (3) sous catégories :

1. Schémas déterministes

   Avant le déploiement, une clé commune est chargée sur tous les nœuds du réseau, après le déploiement chaque nœud prendra connaissance de ses

voisins, il permet d'assurer que chaque nœud peut établir une clé avec ses voisins, c'est-à-dire, chaque nœud du réseau établira une clé unique avec n'importe quel nœud du réseau, ce qui permet d'assurer une connectivité totale du réseau. Une seule clé avant le déploiement appelée « **pairwise key** » qui va servir à établir des pairs de clés**,** cependant, il nécessite un espace mémoire assez important pour le stockage de clés surtout dans le cas des réseaux denses. Le chiffrement d'un même message plusieurs fois avec une méthode déterministe, obtient toujours le même texte chiffré.

Schema de G.Jolly, M. Kusku, P. Kokate et M. Younis

Les auteurs G.Jolly, M. Kusku, P. Kokate et M. Younis ont proposé un protocole déterministe basé sur la pré-distribution de clés, dédié aux applications du domaine militaire. Ce schéma est fondé sur l'architecture hiérarchique du réseau (c-à-d, elle est basée sur le principe de clustering)[17]. Les nœuds du réseau sont regroupés en cluster selon des centres d'intérêts, ils sont liés entre eux par des nœuds ayant des performances élevés en matière d'énergie, de capacité de calcul et de stockage, ces nœuds sont dit passerelles. Aussi, il utilise des nœuds de commandes représentant la tierce partie de confiance de tous les nœuds du réseau. Ce protocole se déroule comme suit :

— Distribution de clés : Chaque nœud du réseau est pré-chargé avec deux clés secrètes avant son déploiement. Une clé partagée avec le nœud passerelle et une autre partagée avec le nœud de commande. Les passerelles partagent des clés entre elles, avec le nœud de demande et avec les nœuds avec qui elle partage le même centre d'intérêt (cluster). Ce protocole tire d'avantage un gain d'énergie sur les opérations d'échange de clés en raison qu'aucune opération supplémentaire sera effectué sur ces clés. Par contre, la compromission d'une clé implique la compromission de sécurité de l'information communiquée.

— Construction de groupe :Chaque nœud du réseau diffuse un message "HELLO" pour la découverte de son voisinage, ce message contient l'ID du nœud et l'ID de la passerelle qui contient la clé partagée. En réception, chaque nœud reçoit une réponse de la passerelle correspondante.

— Révocation de clés : La révocation consiste à éliminer les nœuds compromis du réseau. En effet, si un nœud de commande est compromis, le nœud passerelle l'élimine de son groupe et supprime toutes les routes passant par ce nœud. Par contre, si c'est le nœud passerelle qui est compromit alors le nœud commande le supprime et choisit une autre passerelle.

— Renouvellement de clés : Le nœud commande ré-établira de nouvelles clés, les transmettent aux passerelles. Ces derniers, transmettent à leurs tour une clé pour chaque nœud.

— Ajout d'un nouveau nœud : Un nœud nouvellement arrivé sur le réseau, sera pré-chargé avec deux clés secrètes. Le nœud de commande

transmet un message contenant l'ID et la clé du nouveau nœud à une passerelle choisis au hasard. Cette dernière, procède à l'intégration du nouveau nœud après avoir exécuté l'algorithme de reformation de groupe.

2. Schémas probabilistes

Le principe du protocole probabiliste est qu'avant le déploiement, un ensemble de clés est choisi aléatoirement à partir d'un nombre important de clés, avec condition pour que chaque deux nœuds voisins puissent avoir au moins une clé commune entre eux avec une certaine probabilité après le déploiement. La pré-distribution de clé est réalisée en plusieurs étapes : Commençant par la génération d'un « Pool » de clés avec leurs identifiants. Puis, un choix aléatoirement d'un ensemble de clé K dans le Pool P servant de « porte-clés » (Key ring) pour chaque nœud. Ces porte-clés sont stockés dans la mémoire des nœuds, par contre, l'association entre la liste des identifiants de porte-clés et l'identifiant de nœuds sont stockés dans la mémoire du nœud puits. Les méthodes probabilistes sont faciles et simples à mettre en œuvre dans la phase de distribution de clés après le déploiement, mais possèdent un inconvénient qui réside dans le fait que la taille de l'ensemble des clés pré-distribuées va augmenter considérablement avec l'augmentation de la taille du réseau, ce qui influence la capacité mémoire des nœuds capteurs. Ainsi, ces méthodes ne sont pas sécurisées contre les attaques de capture physique des nœuds [11]. Différents schémas de pré-distribution de clés sont proposés dans la littérature, nous définissons les suivants :

Schémas de L.Eschenauer et D.Gligor : Schéma aléatoire de pré-distribution de clés

Ce schéma est un schéma de gestion de clés basé sur les probabilités partagées entre les nœuds d'un graphe aléatoire sans possibilité substantielle de calcul et de communication. Proposé par L.Eschenauer et D.Gligol en 2002. Il repose sur la distribution des clés d'une façon aléatoire, ce qui implique la création d'un grand ensemble de clés par un administrateur du réseau et avant le déploiement, un groupe de cet ensemble choisi aléatoirement est distribué à chaque nœud[17] [18].

Processus de déroulement :

— Pré-distribution de clés : Avant le déploiement, chaque nœud génère un ensemble de clés P (souvent entre 2^{17} et 2^{20} clés), de cet ensemble P un groupe m de clés choisi aléatoirement. Ces derniers (m clés) sont stockées dans la mémoire du nœud. Le nombre de clés de l'ensemble P : |P| est choisi d'une manière que deux sous-ensembles aléatoires de P de taille m auront une certaine probabilité d'avoir au moins une clé en commun, par exemple :pour une probabilité de p=0.5 on a besoin

d'un sous ensemble de taille m=75 clés et de l'ensemble P de taille S=10000 clés.

— Découverte de clés partagées (après déploiement), ou les nœuds découvrent leurs voisins dans la portée de communication sans fils avec les quels ils sont en mesure de communiquer d'une façon sécurisée car ils possèdent une clé identique dans leurs trousseau, Le moyen le plus simple pour deux nœuds de découvrir s'ils partagent une clé est que chaque nœud diffuse, en clair, la liste des identifiants des clés sur leur trousseau de clés. Après la phase de découverte du voisinage, vient la phase de d'établissement de chemins. En effet, le réseau est représenté sous forme d'un graphe connecté formé de liens sécurisés , les nœuds peuvent alors utiliser des liens existants pour mettre en place des clés partagées avec leurs voisins qui ne partageaient pas de clés en commun avec eux.

— la révocation : La phase de révocation est une étape importante dans le processus de gestion de clés. En effet , la révocation d'un nœud compromis se fait par la suppression de son trousseau de clés. Cela est fait par l'annonce d'un message de révocation de la part de la station de base contenant la liste m d'identificateurs de clés, afin que ces clés soient retirer des trousseaux de clés des autres nœuds. Aussi, il est à prendre en considération les ressources limitées des nœuds capteurs,ainsi la durée de vie de clé peut expirer et donc la nécessité de renouvellement de cette clé qui consiste à établir de nouveaux clés .

L'inconvénient de ce schéma réside dans le nombre de clés qui peut être très grand surtout quand il s'agit d'un réseau dense. Aussi, si un attaquant arrive à compromettre un ou plusieurs nœuds, une grande partie des clés de ce système sera connues par l'attaquant, ce qui lui permettra de calculer les clés de communication établies entres les nœuds.

Schéma de Lanying Li, et Xin Wang : « A high security dynamic secret key management scheme for Wireless Sensor Networks »

Proposé par Lanying Li, et Xin Wang , dans le but de minimiser la consommation d'énergie. En effet ce système divise le réseau en trois niveaux comme le montre la figure 2.2 suivante . Dans le premier niveau, le réseau est organisé avec les nœuds normaux (NN), ainsi le second avec les nœuds chef de groupe (CH) et le dernier niveau avec la station de base (SB).

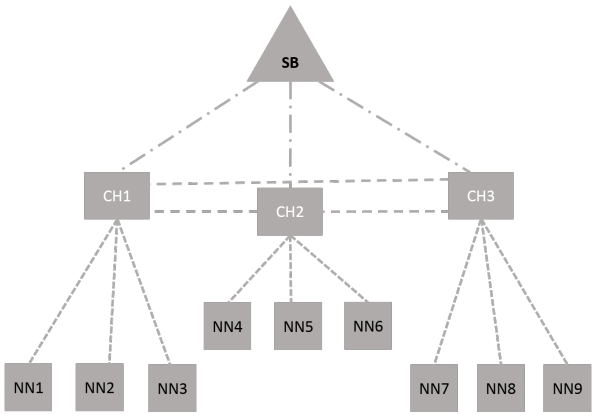

FIGURE 2.2 – Les différents niveaux du réseau

Les nœuds de ce réseau communiquent entre eux via la règle de fils-père pour les nœuds normaux et le chef de groupe, ainsi pour le chef de groupe avec la station de base, aussi un cluster head peut établir une connexion avec un autre cluster head et échange leur données avec lui (communication entre les frères). Ce schéma se déroule en quatre phases : La pré-distribution de clés, établissement de clés par paire, renouvellement de clés, et ajout et suppression de nœuds du réseau[7].

Processus de déroulement :

— Phase de pré-distribution :
Pendant la phase de pré-distribution, la station de base pré-charge tous les nœuds du réseau avec un identifiant unique pour chacun et d'une règle uniforme pour tous les nœuds sans exception, y compris la station de base elle même.
— Phase d'établissement de clés par paire :
Après le déploiement, vient l'étape d'établissement de clés par paire pour les trois niveaux du réseau. Pour ce faire, la station de base génère une matrice Ckp aléatoirement avec N nœuds qui est égale au nombre de colonnes de la matrice générée. Avec p le nombre de clés générées, k représente la longueur de clés nécessaires et P-N représente le maximum des nœuds capteurs qui pourrais être ajouter de nouveau au réseau. Cette phase permettra d'établir des clés entre les différentes paires de nœuds comme suit :
— Établissement de clés entre la station de base et le Cluster Head :
Après avoir élus tous les CHs, la Station de Base diffuse un message à tous ces clusters heads, pour qu'ils puissent commencer le processus d'établissement de clés, ainsi les CHs envoient leurs IDs à la SB. Cette dernière reçoit les IDs de tous les clusters heads, les utilise dans la référence de la matrice Ckp déjà générée et obtient la valeur de la colonne

(car les IDs représentent les numéros de colonnes dans la matrice). Puis, elle utilisera la valeur de KM et la règle uniforme pour le calcul des clés par paire, ensuite, la SB envoie la valeur de KM et la liste des identifiants (IDs) générée aux clusters heads correspondants. A la réception, chaque CH vérifie son ID dans la matrice Ckp. Les clusters heads calculent leurs clés en se basant sur la valeur de la colonne qu'ils ont et le KM reçu de la SB, et enregistrent les clés.

— Établissement de clés entre deux CHs :

Les CHs ont déjà une chaine binaire qui représente un état initial utilisé pour générer les clés qui sont nécessaires pour être calculées avant. Chaque CH a une valeur de colonne pré-chargée sur lui ; Les CHs voisins s'envoient leurs valeur, de sorte que chaque CH va avoir la valeur de sa colonne $C_i$et celle de son voisin C_j . Les deux CHs qui ont fait l'échange avant calculent $C_i$et C_j en conséquence, et le résultat est utilisé comme valeur de l'état initial de la règle uniforme. Les CHs calculent respectivement, selon la règle uniforme et la valeur KM comme nombre d'itérations, puis enregistrent les nouvelles clés.

— Établissement de clés entre CH et les nœuds normaux (NNs) :

Dans cette étape, les nœuds normaux (NN) envoient leurs IDs au cluster heads correspondants (le chef de groupe au quel ils appartiennent), le CH à son tour, envoie la liste des IDs reçu à la SB. Cette dernière interroge la matrice Ckp, et envoie par la suite la valeur de la colonne en fonction de la liste des IDs, et fixe la valeur de KM avec elle (SB) pour le cluster head. Le CH reçoit les informations envoyées par la station de base, puis fait ces calculs. Ensuite, il diffuse les KMs et la liste des identifiants pour les noeuds de son cluster, ces derniers(les NNs) vérifient à leurs tours, les IDs reçus s'ils sont inclus dans la liste reçue. Les nœuds normaux calculent en utilisant KM et la règle uniforme leurs clés et les enregistrent ces dernières.

— Phase de renouvellement de clés :

Le renouvellement de clés correspond à la mis-à-jour et le ré-établissement de clés de réseau. Dans cette approche (ce schéma), le renouvellement se fait lors de capture de cluster head par une entité non autorisée, sachant que cette capture est détectée par la station de base. La station de base doit informer les autres CHs du réseau de cette capture pour rafraichir leurs clés, ainsi elle génèrera un nouveau nombre aléatoire en tant que valeur pour KM. Puis l'ensemble de tous les étapes d'établissement de clés seront répétées mais en utilisant la nouvelle valeur de KM et la même chaine binaire qui a été affecté à chaque nœud.

— Ajout et suppression de nœuds :

Ce schéma permet d'ajouter de nouveaux nœuds, ainsi de supprimer des nœuds anciens du réseau. Pour qu'un nouveau nœud puisse rejoindre le réseau, il doit être pré-chargé de mêmes informations nécessaires(ID qui représente le numéro de colonne dans la matrice Ckp, la valeur de la colonne et la une règle uniforme). Lorsque le nœud est déployé dans le réseau, il diffuse son ID pour le CH le plus proche de lui l'intercepte et le

passe à son tour à la SB pour vérifier son identité. La station de base renvoie la réponse au CH pour qu'il commence le processus d'établissement de clés (entre le CH et le nouveau NN) si l'ID de ce nouveau nœud est validé. Et pour la suppression, dans cette approche se fait uniquement pour les nœuds normaux car c'est eux qui ont une quantité d'énergie moindre par rapport aux CHs. Le nœud normal informe son CH dès que son niveau d'énergie atteint ses limitées (puissance épuisée), le CH va effacer l'ID de ce nœud et sa clé associée.

3. Schémas géographique

   Les schémas de gestion de clés géographiques utilisent la location géographique du nœud pour augmenter la connectivité du réseau et gérer facilement les clés entre les nœuds voisins. Les nœuds capteurs utilisent des modules spéciaux intégrés comme les GPS (Global Positioning System) ou des algorithmes de localisation, afin de pouvoir suivre l'emplacement géographique des nœuds à tout instant après leurs déploiements.

2.4.2 Schémas asymétriques

Ces schémas utilisent des mécanismes asymétriques pour l'établissement d'une clé à deux nœuds ou un groupe de nœuds dans le réseau, de tel schémas sont connus par leurs résistances (résilience) contre les attaques dans un tel réseau [11]. Parmi les schémas de gestion de clés utilisant les mécanismes asymétrique, on peut citer :

2.4.2.1 Les schémas basés sur la PKI

— Le protocole micro-PKI (micro Public Key Infrastructure)

Proposé par Munivel et al, consiste en une version simplifiée des PKI conventionnelles. La station de base possède une paire de clés (une clé publique et une privée) ; la clé publique est utilisée par les nœuds pour authentifier la station de base et la clé privée pour le déchiffrement de données envoyées par les nœuds au niveau de la station de base. Avant le déploiement la clé publique de la station de base est stockée dans tous les nœuds. Les auteurs incluent deux types d'authentifications, le premier se produit entre le nœud et la station de base, telle que le nœud génère une clé symétrique de session et pour le chiffrement utilise la clé publique de la station de base. La clé chiffrée est transmise à la station de base sans être déchiffrée en chemin car les nœuds ne possèdent pas la clé privée de la station de base. Ainsi, à la réception, la station de base déchiffre la clé de session et la stocke dans une table [11]. Le second type se déroule entre n'importe quel couple de nœuds du réseau en passant par la station de base, celle-ci est vue comme étant un authentificateur entre un couple de nœuds. L'un des deux nœuds envoie une requête contenant l'identifiant de l'autre nœud à la station de base, cette dernière génère à la réception une clé aléatoire et la chiffre avec la clé de session du nœud émetteur de la requête. Pour assurer l'intégrité des messages échangés, les auteurs proposent d'intégrer à chaque nœud un code

MAC avec utilisation de la même clé de cryptage des messages. Pour les nœuds arrivés au réseau, avant le déploiement, ils ont uniquement besoin de stocker la clé publique de la station de base.

— Le protocole TinyPk

Proposé par Watro et al, fondé sur le principe de Diffie-Hellman à clé publique afin d'établir une clé secrète entre deux nœuds dans le RCSF. Le TinyPk utilise une autorité de confiance pour la signature des clés publiques des nœuds. Cette clé (CA) est distribuée aux nœuds de réseau avant le déploiement pour qu'il puisse vérifier les clés de ces voisins après la phase de déploiement.

2.4.2.2 Les protocoles basés sur l'identité des nœuds

Les protocoles de cette catégorie utilisent l'identité des nœuds pour l'établissement de la clé publique, ils sont capables d'offrir les mêmes niveaux de cryptographie qu'une PKI et sans avoir besoin d'échanger des clés privée/publiques. Parmi ces protocoles on peut citer :

— **les protocoles PKKE & CBKE**

Ces protocoles utilisent l'identité de nœuds pour créer une seule clé partagée entre chaque paire de nœuds dans le réseau. La création de cette clé nécessite une interaction entre les deux nœuds, en utilisant des méthodes d'envoie et de réception de messages sur les deux cotés et cela avant la mise en place de la clé. Pour réduire l'énergie des nœuds souhaitent partager un message secret et les nœuds intermédiaires , plusieurs méthodes ont été proposées pour éliminer ces interactions. Ces méthodes sont connues dans le domaine de cryptographie comme ID-NIKDS (Identify-Based Non-Interactive Key Distribution Scheme).

— **le protocole C4W**

Les nœuds sont en mesure de calculer les clés publiques des autres nœuds en utilisant leurs identités, ce qui permet de remplacer le rôle de certificats. Cette méthode (C4W) utilise le principe de Diffie Hellman pour l'échange des clés et sans utilisation de certificat.

2.5 Critère de comparaison

Des métriques sont employées pour comparer les différents protocoles de gestion des clés[19], parmi ces métriques nous citons [17][11] :

— **L'efficacité** : la gestion de clés doit prendre en considération les ressources limitées du réseau, en terme de mémoire, en communication et en traitement.

  — Complexité en mémoire : Consiste en capacité mémoire nécessaire pour le stockage et la sauvegarde de clé.

  — Complexité en communication : C'est le nombre de messages échangés pour la gestion de clés.

  — Complexité en traitement : Quantité de cycles de processeur nécessaires pour établir une clé.

- **La connectivité** : La probabilité et le degré pour lequel un couple de nœud peut partager une clé.
- **La scalabilité** : Le nombre de clés généré doit être flexible avec la taille du réseau même après le déploiement (lors de l'ajout de nouveaux nœuds).
- **La topologie** : La topologie de réseau est supportée par le protocole ou pas.
- **La mobilité** : Le protocole de gestion de clés prend t-il en considération la mobilité des nœuds du réseau ou pas[19].
- **La dynamicité** : La gestion de nouveaux nœuds arrivants au réseau ou le départ de certains d'autres doit être gérer d'une façon dynamique.
- **La résilience** ou résistance contre la capture des nœuds : Consiste en la résistance du protocole utilisé à la compromission d'un nœud par un attaquant, la méthode de gestion de clés doit être capable de détecter les nœuds compromis et authentifier les nœuds du réseau avant la phase de distribution de clés.
- **Le renouvellement et la révocation de clés** : Pour une meilleure sécurité du réseau, les clés cryptographiques doivent être renouveler et ré-établir de nouveau. Ainsi, révoquer les clés expirées ou celles découvertes par un attaquant et supprimer tous les liens sécurisés avec elles.

2.6 Conclusion

La gestion de clés cryptographiques dans un réseau de capteurs sans fils est un processus très important et nécessaire pour garantir un haut niveau de sécurité dans un réseau assez vulnérable. Dans ce chapitre, nous avons présenté la classification des méthodes de gestion de clés selon le système cryptographique utilisé, ainsi nous avons détaillé quelques schémas de gestion de clés existant dans la littérature, en constatant que le choix entre les schémas symétriques et asymétriques diffère en fonction du niveau de sécurité souhaité dans le réseau. Nous notons que les solutions symétriques peuvent être choisies pour leurs rapidités et les solutions asymétriques pour la résistance contre les attaques.

Chapitre 3

Simulation et mise en pratique

3.1 Introduction

Dans ce chapitre, nous présentons les outils utilisés dans notre solution de gestion de clés dans un réseau de capteurs sans fil. Nous commençons par la description de l'environnement du travail. Pour atteindre nos objectifs, nous avons utilisé la plateforme TinyOs dédié aux composants embarqués et le langage de programmation NesC. Nous présentons aussi le simulateur TOSSIM utilisé et les protocoles implémentés dans notre solution proposée.

3.2 TinyOs

3.2.1 Présentation

TinyOs est un système d'exploitation conçu pour les composants embarqués à faible puissance, développé par l'université américaine Berkeley de Californie, destiné aux réseaux de capteurs miniaturés. En effet, TinyOs est le plus répondu des OS pour les réseaux de capteurs sans fils, conçu pour prendre en charge les opérations intensives simultanées requises par les capteurs du réseau avec un minimum de matériel. Son fonctionnement se base sur une architecture évènementielle, il propose à l'utilisateur une gestion très précise de la consommation du capteur et permet de mieux s'adapter à la nature de la communication sans fil. TinyOs est un ensemble de composants logiciels qui peuvent être assemblés en un seul exécutable, qui sera chargé sur le capteur, permettant de réduire la taille du code nécessaire pour sa mise en place et donc il répond à la contrainte de mémoire liée aux réseaux de capteurs. La bibliothèque TinyOs comprend des protocoles réseaux, des services de distribution, les drivers pour capteurs

et les outils d'acquisition de données ce qui la rend particulièrement complète [20].Sa conception a été entièrement réalisée en NesC, langage orienté composants, syntaxiquement proche du C. TinyOs donc a été crée pour répondre aux caractéristiques et nécessités des réseaux de capteurs sans fils, telle que :

- Une taille mémoire réduite ;
- Une basse consommation d'énergie ;
- Efficacité en calcul et consommation d'énergie ;
- Communication fondamentale.

3.2.2 Caractéristiques de TinyOs

TinyOs le système d'exploitation dédié aux réseaux de capteurs sans fil, se distingue des autres par ses caractéristiques qu'on peut citer [20] :

Event-driven : TinyOs s'appuie sur le fonctionnement évènementiel, c'est-à-dire il devient actif qu'à l'apparition de certains évènements. Le reste de temps, le capteur se trouve en état de veille afin de garantir une durée de vie maximale aux faibles ressources énergétiques du capteur.

Non-Préemptif : TinyOs ne permet pas la préemption des tâches, tel qu'une tâche ne peut interrompre une autre tâche, par contre, il donne la priorité à l'interruption matérielle.

Consommation : TinyOs est conçu afin de minimiser la consommation énergétique des capteurs, ainsi en absence d'évènement il se met automatiquement en veille.

Non temps réel : TinyOs n'est pas un système temps réel car il n'est pas prévu pour manipuler des niveaux de priorité, pour mieux respecter les échéances, dans les tâches. TinyOs est donc basé sur une structure à deux niveaux de planification :

- Les évènements : Ils sont utilisés pour réaliser de petits processus (par exemple quand le compteur du "timer" arrive à son terme). De plus ils peuvent interrompre les tâches qui sont exécutées.
- Les tâches : Les tâches sont pensées pour réaliser une plus grande quantité de traitements et elles ne sont pas critiques dans le temps. Les tâches sont exécutées complètement, mais l'initialisation et la terminaison d'une tâche sont des fonctions séparées.

Modularité : Application constituée en composants, l'OS et l'application sont compilés en un seul exécutable.

Communication : Modèle event/command, ordonnancement FIFO non préemptif.

3.3 NesC

3.3.1 Présentation

NesC (prononcé Nisse C) est le langage de programmation utilisé par TinyOs, syntaxiquement proche du langage C, compilé vers le langage C avant sa compilation en binaire. Constitué d'interfaces et de composants. Le composant est l'élément de base d'une application NesC [20][21].

Un composant implémente des interfaces utilisées par d'autres composants pour communiquer avec eux, ainsi il exécute des commandes, lance des évènements, et il dispose d'un frame pour stocker l'état local. Une application NesC est représentée comme étant un ensemble de composants reliés entre eux pour former un seul exécutable. Chaque composant fournit (provides) et implémente (uses) plusieurs interfaces qui permettent de l'utiliser. Il existe deux types de composants :

- Module : Elément basic de programmation en NesC, ce type de fichier contient le code de l'application et peut utiliser ou fournir une ou plusieurs interfaces.
- Configuration : Les configurations sont utilisées pour lier les composants entre eux, et connecter les interfaces utilisées par ces composants, permettent de décrire l'architecture de l'application. Il est à noter que les éléments connectés doivent être compatibles : 'interface' à 'interface', 'command' à 'command' et 'event' à 'event'.

3.3.2 Command, events et tasks

Une commande en NesC est toute fonction peut être appelée par une interface ; elle ne doit pas être bloquante et elle peut appeler des commandes sur d'autre composants. Les commandes sont des appels de haut vers le bas, c'est-à-dire de composants applicatifs vers des composants proches du matériel. Tandis que, les évènements remontent les signaux du bas vers le haut, ainsi, ils peuvent appeler des commandes, signaler d'autre évènements, poster des tâches mais ne peuvent pas être signalé par des commandes. Aussi, un évènement peut interrompre une tâche, tandis que le contraire n'est pas accepté. Les tâches sont exécutées selon un ordonnancement FIFO et pas de mécanisme de préemption entre elles. Elles sont utilisées pour réaliser un travail qui nécessite beaucoup de calcul et nécessite une longue durée. Une tâche peut être poster par une commande ou un évènement.

Une tâche est un élément de contrôle indépendant défini par une fonction sans argument retournant un void (vide).

3.4 Simulation et mise en pratique

La simulation permet aux utilisateurs de déboguer, faire des tests et analyser des algorithmes dans un environnement contrôlé. Les simulateurs fournissent des structures virtuelles ainsi ils permettent de créer une configuration du réseau souhaité en utilisant des composants simulés ayant des mêmes spécifications qu'en réseau mis en environnement réel.

3.4.1 Outils de simulation

L'outil que nous avons utilisé pour la simulation de notre travail est le simulateur TOSSIM et son extension Power TOSSIM.

3.4.1.1 TOSSIM

TOSSIM est un outil très efficace pour le débogage et le test des programmes TinyOs. Il est très utile de savoir que notre code fonctionne bien, avant de le télécharger sur les capteurs. TOSSIM simule des applications TinyOs entièrement en remplaçant les composants par des implémentations de simulation [16][22] , il permet de simuler le comportement d'un capteur, par exemple, lors de l'envoie et/ou réception de messages via des ondes radios au sein d'un réseau de capteurs. TOSSIM est souvent utilisé avec une interface graphique : TinyViz, pour une meilleure compréhension et visualisation de l'état du réseau.(voir Annexe B)

3.4.1.2 Power TOSSIM

L'énergie est une ressource précieuse dans les réseaux de capteurs sans fils. En raison du coût et de la difficulté de déployer en réseau de capteurs. Il est impératif de pouvoir obtenir le profil de puissance d'une application avant son déploiement réel. Nous présentons Power TOSSIM, une extension de modélisation de puissance à TOSSIM, simulateur de TinyOs qui n'est pas capable d'évaluer l'énergie [23]. Power TOSSIM modélise précisément l'énergie consommée par les applications TinyOs [16], ainsi, un nœud qui ne possède plus d'énergie s'arrête de fonctionner, ce qui permet d'exécuter la simulation jusqu'à la mort du réseau. Power TOSSIM et TOSSIM exigent que les applications à simuler doivent être écrite en NesC[23].

3.5 Protocoles implémentés

3.5.1 Protocole de routage implémenté

Nous avons implementé le protocole LEACH qui est un protocole de routage hierrachique basé sur le clustering dynamique dans les réseaux de capteurs ho-

mogènes. Il est caractérisé par la consommation réduite d'énergie et son auto-organisation des clusters [24].

Il est fondé sur deux hypothèses [25][26] :

— La station de base est située en dehors de la zone de captage.
— Tous les noeuds sont homogènes, ils disposent tous de la même capacité d'énergie excepté la SB.

L'auto-organisation des clusters, offre une autonomie en terme de décisions et une décentralisation des contrôles. Cette pratique réduit la consommation d'énergie mais ne garantit pas une répartition équitable, vis-à-vis de la position des CHs ou le nombre de nœuds dans chaque cluster.

Le rôle de CH est affecté à chaque noeud du réseau pour garantir une consommatin équitable en terme d'énergie, vu que les traitements fait par les CHs sont gourmands en cette ressource.

3.5.1.1 Les phases de fonctionnement de LEACH

L'algorithme se déroule en round, chaque round débute par une phase d'initialisation, élection des CHs et formation des clusters, suivie d'une phase de transmission. (Voir Annexe C).

Phase d'initialisation: Phase d'élection de CHs et de formation de clusters

La sélection des CHs est basée sur le pourcentage de CHs voulu. Cette phase peut être divisée en trois parties :

1. Phase d'annonce: L'élection d'un nombre déterminé de CH se fait ainsi :
   — Chaque nœud choisit un nombre aléatoire compris entre 0 et 1. Se nombre est comparé à un seuil P(t) calculer par la SB par la formule :
   $$P(t) = \left\{ \frac{K}{N - K * \left(rmod\left(\frac{N}{K}\right)\right)} \right\}$$
   Où K : est le nombre de CH désirés.
   Et $N - K * \left(rmod\left(\frac{N}{K}\right)\right)$ est le nombre de nœuds éligibles d'être CHs.
   — Une fois le CH élu, Il diffuse son statut vers tous les nœuds et attend des demandes d'appartenances.
   — Les nœuds reçoivent les annonces de CHs.
2. Phase d'organisation de groupes ou Clustering:
   — Chaque nœuds pouvant recevoir plusieurs annonces de CH, traite toutes les annonces et choisit d'appartenir au CH le plus proche en se basant sur la puissance du signal (on considérera la profondeur du nœud dans le réseau). Si les profondeurs sont égales, le choix se fait de manière aléatoire [27], sinon le CH de profondeur minimum sera choisit.

— Une demande d'appartenance est alors envoyé au CH choisit après l'écoulement de la durée nécessaire de traitements de toutes les annonces de CH.

3. Phase d'ordonnancement
   — Le CH reçoit les demandes, il peut alors calculer le nombre des membres de son cluster et leurs crée la table TDMA. Ensuite, le CH affecte à chacun un slot qu'il utilisera exclusivement pour émettre et recevoir des données pour et à partir des autres membres. L'ensemble des slots de ses membres s'appelle un frame.
   — Chaque nœud reçoit son slot depuis le CH choisit et pourra commencer la transmission de ses données une fois la durée de son slot atteinte. Donc, en dehors de son slot un nœud capteur éteindra sa radio et restera endormi pour économiser son énergie.
   — En plus du slot, chaque CH dispose d'un code de propagation CDMA aléatoire qu'il affecte à ses membres afin d'éviter des interférences, entre les clusters adjacents, durant leurs transmissions.

Phase de transmission:

Phase d'une durée plus longue et durant laquelle un transfert de données est fait vers la SB [28],

— Chaque nœud capteur envoie ça donnée captée à son CH à travers son slot.
— Le CH attend que la durée du frame s'écoule (durée nécessaire pour que tous ses nœuds membres aient envoyés leurs données) ; pour qu'il puisse agréger les données reçues et transmettre en un seul saut le résultat à la SB.
— La SB reçoit toutes les données agrégées à partir des CH et termine le round courant et annonce le début d'un nouveau round permettant ainsi de synchroniser tous les nœuds pour recommencer les deux phases, puisque la SB ne débute un round qu'après avoir reçu toutes les données.
— Ce processus est répété jusqu'à ce que tous les nœuds soient élus CH une fois , c'est à ce moment là alors, que le round est réinitialisé à 0.

3.5.1.2 Inconvénients de LEACH[24]

— Il est supposé que chaque nœud est capable d'atteindre la SB pour lui communiquer les données.
— Le choix aléatoire des CHs peut entrainer un isolement de certains nœuds (des nœuds n'appartenant pas à un cluster), si les CHs sont concentrés dans une zone.
— La transmission des données est périodique, cependant dans certains cas elle peut ne pas être nécessaire, donc il y aura une consommation d'énergie inutile.

— La transmission des données agrégées se fait en un seul saut, donc une grande consommation d'énergie en résultera du fait de l'éloignement de certain CHs.
— L'élection périodique des CH est une charge supplémentaire dû aux messages d'avertissement dans la phase d'élection de CH, ce qui sera débiter du gain d'énergie.
— Les noeuds les plus distants du CH consomment beaucoup plus d'énergie dans la transmission, donc meurent plus vite.
— Le protocole LEACH n'est pas sécurisé donc vulnérable à plusieurs types d'attaques.

3.5.2 Protocole de sécurité

Nous utilisons pour notre contribution le protocole de sécurité basé sur les courbes elliptiques ECIES (**E**lliptic **C**urve **I**ntegrated **E**ncryption Scheme) ou le Système de Cryptage Intégré à courbe elliptique en français. ECIES est un système de cryptage hybride proposé par Victor Shoup en 2001, conçu pour être sémantiquement sécurisé en présence d'un adversaire capable de lancer des attaques comme les attaques à texte en clair choisi et les attaques à texte chiffré choisi. Il assure le chiffrement et l'authentification des messages, il permet aussi d'échanger des clés cryptographiques en toute sécurité via un canal non sécurisé. Pour chiffrer un message, il faut d'abord encoder le texte en claire m comme un point de coordonnées (x,y), c'est ce point là qui sera chiffré par la suite. Pour cela, deux entités souhaitant échanger des messages secrets rendent publique une courbe elliptique E d'ordre premier$p$ et un point G appartenant à cette courbe, Ils doivent également choisir une clé privée et générer la clé publique correspondante [6][12]. Supposant que Alice et Bob veulent échanger des messages secrets, alors ils doivent d'abord disposer de ces informations :

— KDF (Key Derivation Function) : Une fonction de dérivation de clé qui permet de générer plusieurs clés à partir d'une valeur secrète de référence.
— MAC (Message Authentication Code) : Code transmis avec les données dans le but d'assurer l'intégrité de ces dernières.
— S YM : Algorithme de chiffrement symétrique.
— $E(Fp)$: La courbe elliptique utilisée avec le point de générateur G dont $ord_p(G) = n$.
— KB : La clé publique de Bob $KB = kB.G$, où $kB\epsilon[1, n-1]$ est sa clé privée.

Le tableau suivant résume les étapes de déroulement de cet algorithme :

Le chiffrement de coté Alice	Le déchiffrement coté de Bob
Choisir un entier $k\epsilon[1, n-1]$ et calculer$R = k.G$.	Rejeter le message si $R < E(Fp)$.
Calculer $Z = k.KB$.	calcule $Z = kB.R = kB.k.G = k.KB$.
Générer les clés$(k1, k2) = KDF(abscisse(Z), R)$.	Générer les clés$(k1, k2) = KDF(abscisse(Z), R)$.
coder le message $C = SYM(k1, M)$.	Générer le code $MACt\prime = MAC(k2, C)$.
Générer le code$MACt = MAC(k2, C)$.	Rejeter et ne pas accepter le message si$t\prime \neq t$.
Envoyer (R, C, t)à Bob.	Déchiffrer le message$M = SYM - 1(k1, C)$.

TABLE 3.1 – Les étapes de l'algorithme ECIES (chiffrement et déchiffrement)

La valeur critique de ce protocole est l'entité k, elle permet à Bob de calculer $Z = k.KB$ et de générer le couple (k_1, k_2) qui sera utilisé pour le chiffrement et l'authentification de messages. Et vue que le problème de logarithme discret est difficile à résoudre , Alice peut envoyer $R = k.G$ sans problème.

Ainsi dans notre nous utilisons ECIES pour sécuriser les chemins de protocole de routage LEACH, en sécurisant le chemin entre la Station de base et le Cluster Head, le chemin entre le nœud membre et son Cluster Head. Et comme LEACH possède l'inconvénient qu'un nœud peut être isolé et n'appartient à aucun cluster, dans ce cas, le nœud isolé choisit un nœud dis intermédiaire (le plus proche à lui) à qui il envoie ses données, ce dernier après les avoir reçues et agrégées, les envoie au cluster head du cluster au quel il appartient. Donc le chemin entre le nœud isolé et le nœud intermédiaire doit être aussi sécurisé.

3.6 Conclusion

Dans ce chapitre, nous avons présenté l'environnement de développement (système d'exploitation TinyOs et le langage de programmation NesC). Ainsi, le protocole de routage utilisé pour l'acheminement de données avec le protocole de sécurité implémenté pour la gestion de clés cryptographiques dans un RCSF. Le chapitre suivant sera consacré pour détailler le schéma proposé avec une comparaison des résultats obtenus.

Chapitre 4

Implémentation et résultats

4.1 Introduction

En se basant sur les protocoles de routage et de sécurité décris précédemment, nous présentons la solution que nous suggérons pour atteindre un objectif fixé, celui de proposer un schéma de gestion de clés dynamique pour les réseaux de capteurs sans fil à mobilité, en utilisant un système de chiffrement asymétrique basé sur les courbes elliptiques : ECIES. De ce fait, ce chapitre est dédié à la présentation de notre solution et la description de son fonctionnement. Puis, nous terminons par une comparaison de nos résultats obtenus avec les résultats des autres solutions proposées dans le domaine des réseaux de capteurs.

4.2 Les facteurs de conception

Des facteurs influençant sur la durée de vie du réseau et son efficacité doivent être pris en considération avant la mise en environnement réel du réseau, dans cette optique nous citons :

- La consommation énergétique : L'énergie est une ressource critique dans le cadre des réseaux de capteurs, les nœuds capteurs disposent d'une quantité d'énergie limitée.
- La scalabilité : L'une des caractéristiques des réseaux de capteurs sans fil est qu'ils peuvent contenir des centaines voir des milliers de nœuds capteurs, ainsi le réseau doit être capable de fonctionner avec ce nombre de nœuds.
- La connectivité : La connectivité est un problème majeur dans les réseaux de capteurs, l'architecture de notre réseau est une architecture hiérarchique, on dit que le réseau est connecté si et seulement s'il existe au moins un chemin entre chaque paire de nœud. c'est-à-dire il existe au moins une route entre chaque paire de nœud du réseau, des chemins entre

cluster head et station de base et des chemins entre un nœud membre du cluster avec son chef.
— L'agrégation de données : L'agrégation est une technique très importante voir nécessaire pour réduire la quantité des informations redondantes transmises par les nœuds capteurs et qui seront reçues par la station de base. Cette réduction permet de conserver l'énergie du capteur et améliorer sa durée de vie.
— Le modèle de communication : Repose sur une architecture orientée évènements.
— La mobilité : Les nœuds capteurs sont parfois amenés à changer leur positions à causes de phénomènes naturels ou trainés par des animaux .

4.3 Description de la solution implémentée

La solution implémentée a pour objectif de sécuriser le protocole de routage LEACH en garantissant une gestion optimale et sécurisée des clés cryptographes générées à partir d'un protocole de sécurité basé sur la cryptographie asymétrique des courbes elliptiques. Notre solution vise à garantir un meilleur routage de données pour notre schéma de gestion tout en respectant la contrainte d'énergie. Pour cela on a essayé de remédier à quelques problèmes posés par LEACH allant jusqu'à être considérés comme étant ses inconvénients. Entre autre le problème de connectivité du réseau, où chaque nœud est supposé capable d'atteindre la SB, qui a été résolu par l'adoption d'une architecture à deux sauts au maximum avec le principe du nœuds isolé. Ce principe qui vise principalement à régler le problème d'isolement de certains nœuds dans le cas où les CHs sont concentrés dans une zone précise et la prolongation de la durée de vie des nœuds isolés en réduisant la distance de transmission radio des données en considérant un et un seul nœud intermédiaire, d'où le nom donné à ce principe. Une amélioration aussi est apportée à l'issu de ce principe au niveau de l'agrégation qui ne se faisait qu'à un seul saut contrairement à notre solution qui supporte deux sauts. La solution tente aussi à minimiser le nombre de paquets échangés dans le réseau, précisément pour le schéma de gestion en adoptant le principe d'échange de clés à la demande qui garantie aussi une optimisation dans l'espace de stockage des clés échangées. Cette gestion de clés garantit un ensemble de chemins sécurisés pour l'acheminement des données depuis les nœuds isolés situés au bas de l'architecture jusqu'à la SB qui est en tête. De ce fait, on dispose de trois types de liens sécurisés : SB-CH , CH-membre où CH-noeud isolé,noeud_isolé-noeud_intermédiaire (NIS-NI).

4.4 Structure de réseau

Comme le protocole de routage LEACH sur lequel notre architecture est basée, possède la possibilité qu'un nœud ne soit membre d'aucun cluster, c'est-à-dire isolé du réseau. Pour ce cas, notre solution propose une communication à 2

sauts au maximum pour le protocole LEACH. Par le billet de cette structure (la communication est à un seul saut dans le cas normal, et à 2 sauts dans le cas de nœud isolé). Il existe ainsi au moins un chemin entre chaque paire de nœud (chemin direct à 1 seul saut ou par l'intermédiaire à 2 sauts).

Donc, dans notre cas, l'architecture du réseau est une architecture hiérarchique à un seul saut, ou parfois deux sauts dans le cas de présence du nœuds isolés, fondée sur le protocole LEACH. La figure suivante explique les différents niveaux de cette architecture réseau :

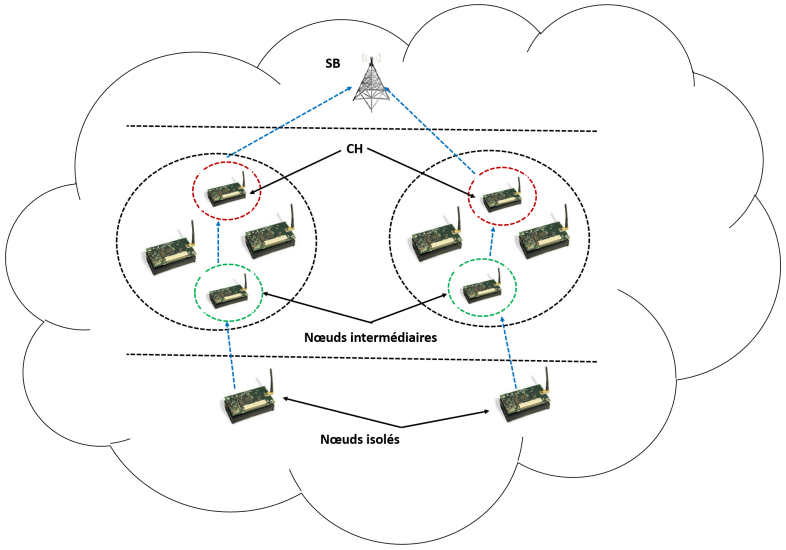

FIGURE 4.1 – Les différents niveaux du réseau

Le niveau supérieur représente la station de base, cette dernière considérée comme n'ayant pas de contrainte sur les capacités de calcul, stockage, énergie et ne peut pas être compromise. Dans le second niveau, on trouve les clusters ou chaque cluster est constitué d'un nœud chef : le cluster head (CH) et des nœuds membres (NM) dont certains peuvent se transformés en nœuds intermédiaires pour les nœuds du niveau inférieur. Le niveau inférieur et le dernier niveau qui regroupe les nœuds n'ayant pas de cluster, c'est-à-dite ils n'appartiennent à aucun cluster, ces nœuds sont appelés nœuds isolés . Les communications se font suivant hiérarchie du réseau, de la SB passant par les CHs allant vers les nœuds membres ou passant par les nœuds intermédiaires (NI) allant vers les nœuds membres et vise vers ça.

Le nœud isolé pour déterminé son nœud intermédiaire envoie un message en broadcast pour annoncé son état, les nœuds membres du cluster reçoivent ce message et lui envoie leurs réponses. Le premier nœud, appartenant à un cluster et disposant d'un slot ,ayant répondu sera le puit vers lequel le nœud isolé transitera ses données. Dans notre modèle ce nœud est dit : nœud intermédiaire.

Le nœud isolé envoie ces données à son intermédiaire durant un slot déjà définit, le nœud intermédiaire à son tour reçoit les données du nœud isolé, les traite, les agrège puis il les envoie à son cluster head.

La figure ci-dessous résume l'architecture et le fonctionnement général du réseau :

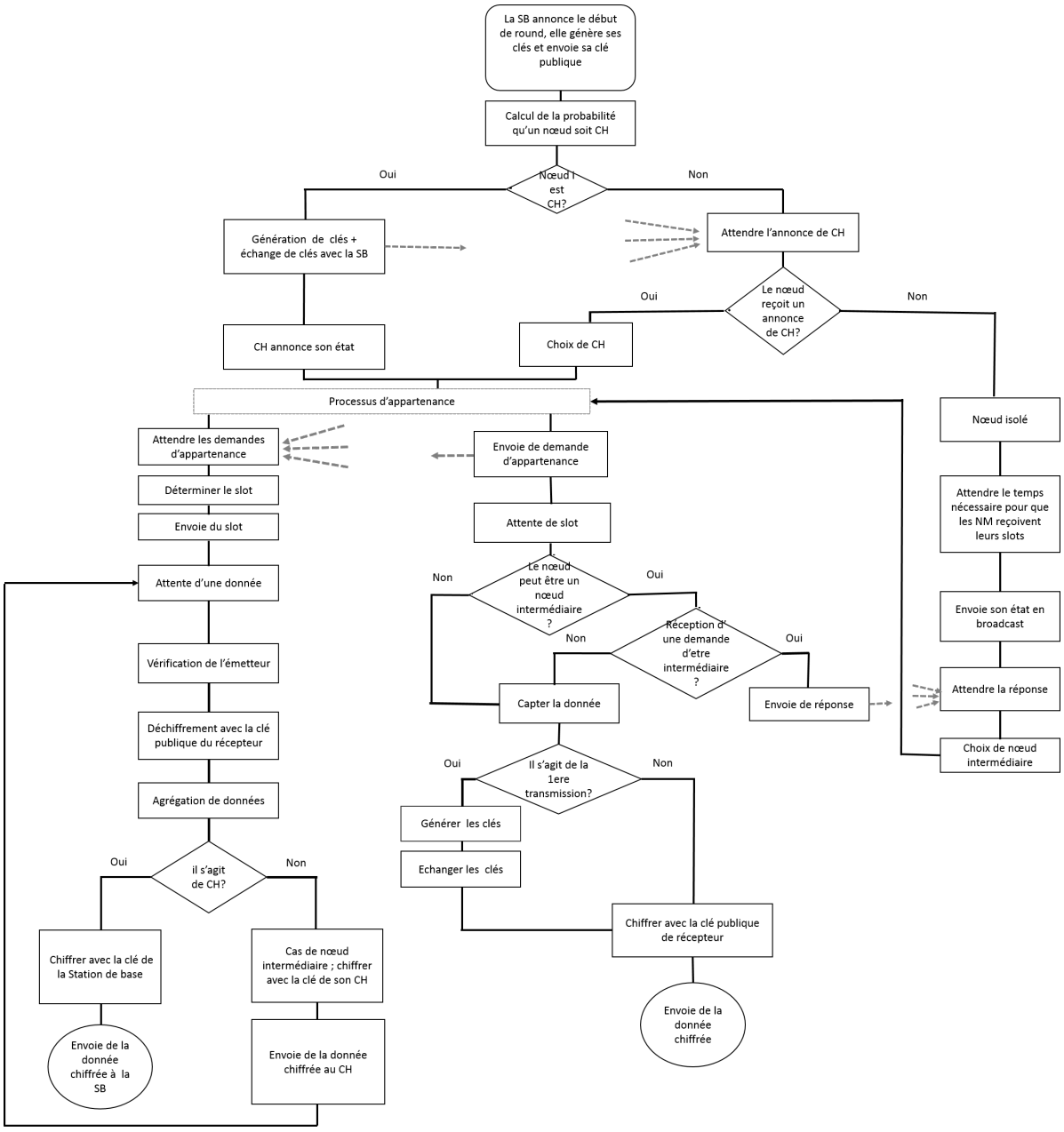

FIGURE 4.2 – Structure du réseau

4.5 Phases de gestion de clés

Pour améliorer la sécurité des capteurs, la gestion de clés de cryptage est un processus important à mettre en place, mais revient aussi difficile à implanter. Ceci est dû à la topologie dynamique et aux ressources limitées des réseaux de capteurs sans fil. Avant de présenter les phases de déroulement du schéma de notre solution proposée, nous décrivons les points sur lesquels notre schéma est basé :

- Le réseau de capteurs peut être un réseau mobile, c'est-à-dire tout nœuds est capable de changer sa position dans le réseau.
- Les nœuds du réseau sont homogènes (ils ont tous les mêmes capacités de mémoire, de stockage, d'énergie et de calcul).
- La station de base n'a pas de contrainte sur les capacité de calcul, d'énergie et de stockage. Aussi elle ne peut pas être compromise.
- Génération de clés asymétriques en utilisant les courbes elliptiques (ECC).
- Chaque capteur est chargé avec un identificateur (ID) unique délivrée par la station de base.
- La sécurité des clés cryptographiques implique la sécurité du réseau.
- Mise à jour des clés et ajout de nouveaux nœuds : La possibilité d'ajouter de nouveaux nœuds au réseau et de les attribuer des clés, mettre à jour, révoquer et détruire les clés des nœuds interceptés.

Notre schéma proposé se déroule comme suit :

- Génération des clés : Chaque nœud du réseau génère une paire de clé publique/privée. La clé publique est connue par tout les nœuds, à l'opposé de la clés privée qui reste confidentielle pour chaque nœud. La station de base, le cluster head ainsi les autres nœuds membres génèrent aléatoirement tous leur propre paire de clés basée sur les courbes elliptiques (ECC).
- Échange de clés à la demande :

1. Echange de clés SB-CH : L'échange de clés entre la SB et chaque CH débute par l'annonce de la SB du déclenchement du nouveau round, où celle-ci véhicule ce message d'annonce de son adresse publique. Les nœuds du réseau recevant et traitant ce message, évaluent leurs probabilité d'être CH, si cette dernière et inférieure à celle déterminée par la SB, ce nœud est alors élu CH et sauvegarde par la même occasion la clé publique de la SB.
2. Echange de clés MB-CH ou NIS-NI :Dans notre approche, l'échange de clés se fait uniquement à la demande. C'est-à-dire, c'est au moment ou un nœud émetteur décide de transmettre ses données, pour la première fois, à un nœud récepteur du réseau, que ces deux échangent leurs clés. Cela permet un gain intense en terme de paquets émis, d'énergie et de mémoire de stockage, puisqu'il est fréquent d'avoir durant un round des nœuds ne désirant pas entré en communication avec leurs Chs et donc une

dissipation des ressources est résultée dans le cas de l'approche standard d'échange de clés.

La figure suivante résume cette étape :

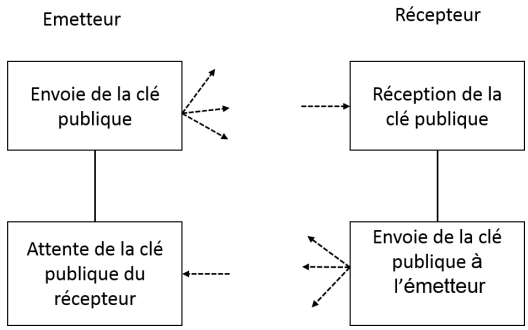

FIGURE 4.3 – Échange de clés

— Vérification d'identité du nœud : A la réception du CH de données chiffrées avec sa clé publique, le Cluster Head vérifie d'abord que l'émetteur n'est pas révoqué puis vérifie son identité. Si cet ID appartient au IDs de sa table des membres, alors il déchiffre la donnée avec sa clé privée. Sinon (ID de nœud émetteur n'existe pas dans la table des IDs des membres de CH) le cluster head signale ce nœud à la station de base en lui envoyant un message pour vérifier l'identité de ce nœud dans tous le réseau. La station de base à son tour, vérifie si ce nœud appartient au réseau, si c'est le cas alors c'est un nœud qui a changé sa position dans le réseau, due par effet de mobilité, ou bien c'est un nouveau nœud arrivé sur le réseau, en réponse, elle envoie un message au CH pour l'inclure dans ce cluster. Par contre, si l'ID n'existe pas dans la liste des nœuds du réseau, la station de base envoie un broadcast pour révoquer ce nœud du réseau et supprimer sa clé publique et toute les routes sécurisées avec elle.
— Révocation de nœud et suppression de clés : La révocation de clés est une phase intervenante dans le cas de compromission d'un nœud capteur, ce qui nécessite la suppression de la clé de sécurité de ce nœud, ainsi tous les liens sécurisés par cette clé, supprimer tous les liens avec ce nœud compromis. Le processus de vérification de la révocation d'un nœud est résumé dans la figure suivante :

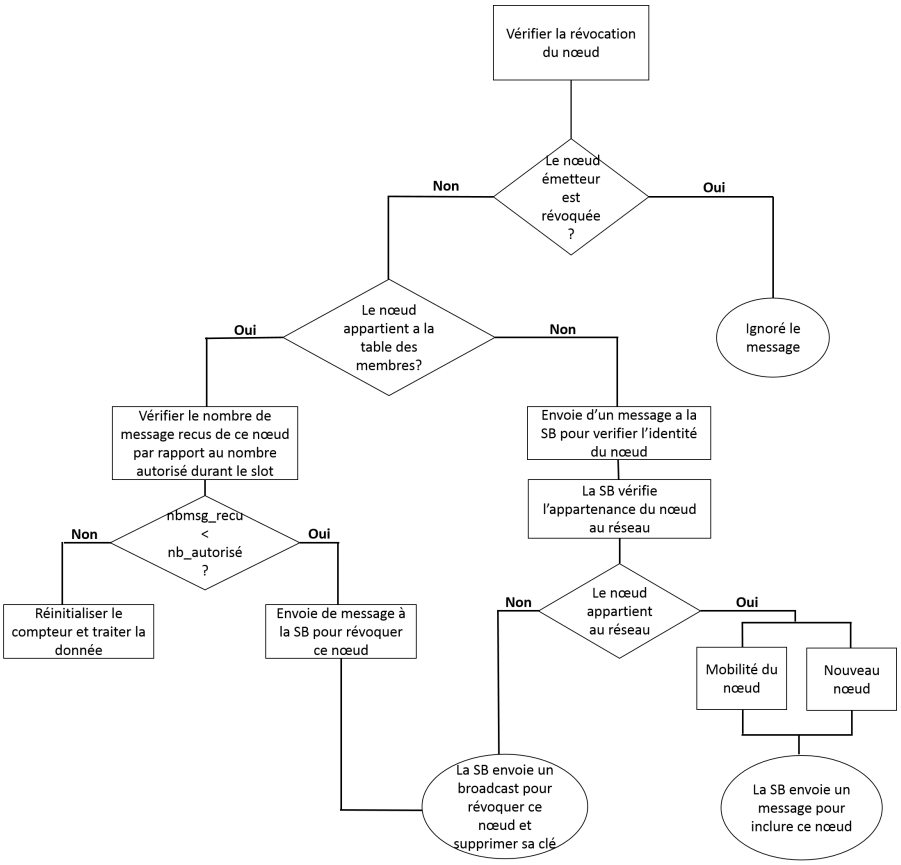

FIGURE 4.4 – Vérification de la révocation d'un nœud

— Reconstruction et Renouvellement de clés : Toute fois, la durée de vie d'une clé cryptographique partagée entre deux nœuds du réseau peut atteindre sa limite, et donc cette clé sera expirée et ne sert plus à protéger le lien sécuriser par cette clé. Dans ce cas, une phase de renouvellement (re-keying) est nécessaire et doit être mise en service. le re-keying ou le renouvellement en français consiste à re-générer et ré-établir les clés de nouveau, c'est le retour vers la phase de génération de clés. Le re-keying est équivalent à la révocation de clés de chaque nœud du réseau sans attendre la réception d'un message de broadcast de la part de la station de base. Dans notre schéma, le renouvellement se fait à chaque round. Cela est dû au fait que la durée de vie des clés est proportionnelle à la durée d'un round. Donc à chaque nouveau round de nouveaux clusters sont élus ainsi de nouveaux liens sont établies, il est alors intéressant et suffisant de procéder au re-keying lors de l'établissement de la nouvelle architecture pour éviter que des clés n'expirent en phase de communication et causé

ainsi une perte de données.

— Ajout d'un nouveau nœud : Si un nouveau nœud veut joindre le réseau, l'administrateur du réseau doit charger la clé publique du CH de la zone de déploiement du nœud, ainsi qu'un slot. Comme ça après le déploiement, le nouveau nœud peut procédé directement à l'échange de données et sa clé public avec son CH. Ce dernier en recevant la donnée de ce nouveau nœud va lancer la procédure de vérification d'identité et par la suite ajouter ce nouveau nœud à sa table de membre.
La figure ci-dessous résume le processus de gestion de clés du notre schéma :

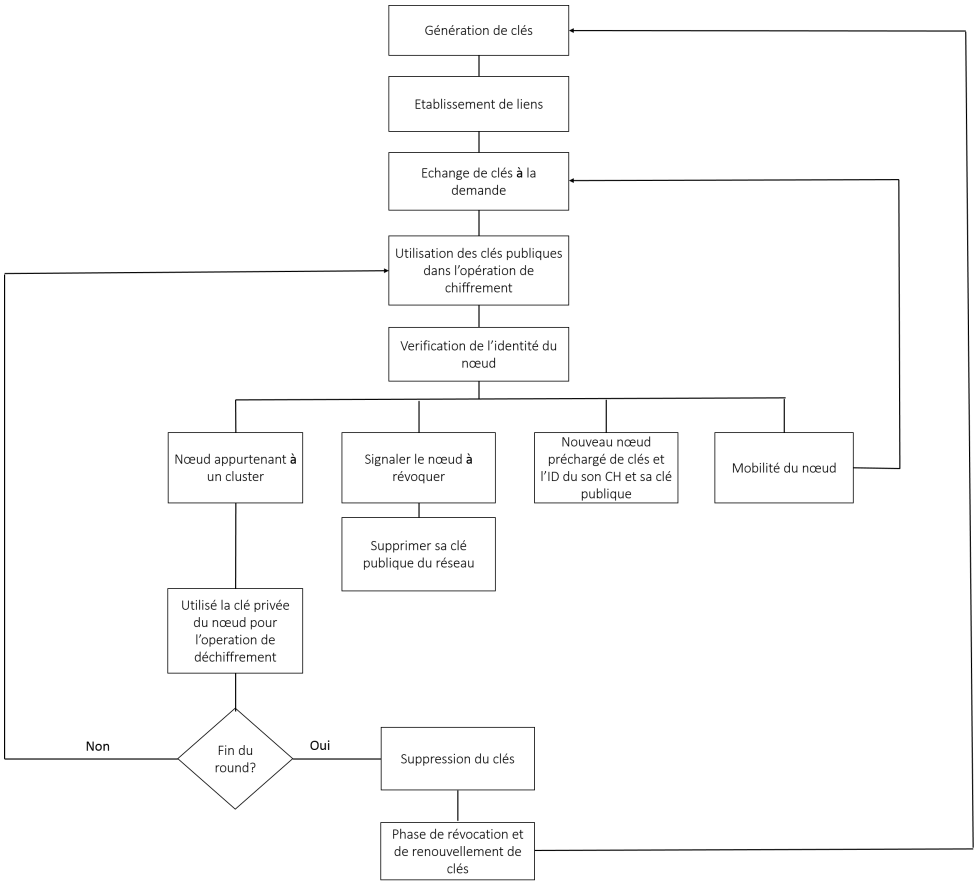

FIGURE 4.5 – Schéma de gestion de clés

— Analyse de sécurité : Des mesures de prévention des attaques sont prises en considération :
Un attaquant qui intercepte les clés publiques (clés de chiffrement) n'est pas en mesure d'obtenir la clé privée (clé de déchiffrement).De ce fait

un attaquant qui intercepte un message chiffré n'est pas en mesure de le déchiffrer.
Une mesure de prévention contre l'attaque des nœuds malveillants n'appartenant pas au RCSF est prise en considération, en vérifiant l'identité du nœud essayant d'entrer en communication avec un CH, ainsi que son appartenance au réseau.
A la réception de tout message par le CH, il vérifie, avant de le traiter, que ce nœud n'est pas révoqué et puis qu'il appartient réellement à sa table de membres, dans le cas contraire, il signalera ce nœud à la SB pour vérifier son appartenance au réseau.
Une autre mesure est prise en considération en ne tolérant la réception qu'un nombre précis de messages "mrec", par le CH d'un nœud donné durant la période de slot qu'il lui est dédiée.

4.6 Simulation des résultats

4.6.1 Métriques d'évaluation utilisées

— La consommation énergétique : Comme les nœuds capteurs sont des composants microélectronique, ils sont équipés par des ressources énergétiques limitées. La solution proposée prend en considération cette contrainte et tente à optimiser cette ressource.
— le nombre de paquets envoyés .
— le nombre de sauts : le nombre de saut est au maximum 2 sauts, l'architecture du réseau est à un seul saut entre le cluster head et son membre, le cluster head avec la station de base. Cependant, elle est à deux sauts dans le cas d'un nœud isolé en passant par le nœud intermédiaire qui situe entre le nœud isolé et le chef du cluster.

4.6.2 Paramètres de simulation

Nombre de nœuds du réseau	De 100 à 700
Nombre de clusters	10 % de N
Nombre de nœuds intrus	10 % de N
Modèle de topologie	Aléatoire
Taille des clés de notre solution	192 bits
Taille des clés des solutions de comparaison	160 bits
Taille du paquet	40 octets
Type de paquets	AM
Durée de simulation	60s

TABLE 4.1 – Paramètres de simulation

4.6.3 Solutions de comparaisons

1. TinyKeyMAN :

   TinyKeyMan fournit une implémentation sur TinyOs pour l'établissement de clés par paires dans les réseaux de capteurs sans fil, cela en utilisant le schéma de pré-distribution de clés basé sur un pool polynomiale. Il inclue l'implémentation du schéma d'assignation de sous ensemble aléatoire et du schéma basé sur la grille, ces derniers ont un certains nombre de propriétés intéressantes, notamment la probabilité élevée d'établir des clés par paires, la tolérance à la capture des nœuds et la réduction des couts généraux de communication et de calcul.

   Le calcul du secret se fait avant le déploiement et il est sécurisé par une technique d'évaluation polynomiale pour minimiser les couts en calcul et mémoire. Ainsi qu'au déploiement, deux nœuds peuvent établir une clé symétrique s'il partagent le même secret, la probabilité que deux nœuds partagent un polynôme et établissent une clé symétrique est obtenue en fonction de la taille du réseau.

   Pour ce faire, les deux nœuds échangent les listes des identifiants et leurs polynômes et si aucun polynôme n'est en commun, ils essaient d'obtenir un nœud voisin qui partage une clé symétrique avec chacun des deux. Ce nœud voisin calcul une clé aléatoire dite clé de chemin "Key Path" qu'il va transmettre par la suite aux deux nœuds afin qu'ils puissent l'utiliser comme clé symétrique.

   L'avantage de ce schéma réside dans la rapidité et le cout réduit pour l'établissement de clés symétriques entre n'importe quels nœuds voisins.

2. Ramdani and al :

La solution basée sur le protocole de gestion de clés déterministe et distribué, basée sur la cryptographie des courbes elliptiques (ECC) : Cette solution a comme but de résoudre le problème de clés partagées en respectant les contraintes et caractéristiques des RCSFs [15].

Un protocole de routage (LEACH) utilise les clés installées et partagées par les nœuds pour construire une topologie de communication énergétique optimale.

4.7 Evaluation des résultats

1. Nombre de paquets émis lors de l'échange de clés : Dans la figure qui suit, une comparaison est faite entre une solution standard où l'échange de clés se fait à la réception du slot depuis le CH, et notre solution qui est basée sur la gestion des clés à la demande.

   Habituellement, on est amené à échanger et stocker des clés dont le nœud ne s'en sert pas durant la durée de vie de la clé, et donc elle sera renouvelée avant même d'être utilisée impliquant une perte d'énergie, consommée lors de l'échange de clés, et d'espace de stockage. il est alors intéressant d'étudier et de présenter une comparaison entre le nombre de paquets émis dans la solution standard et celui du principe d'échange de clé à la demande dans la figure qui suit.

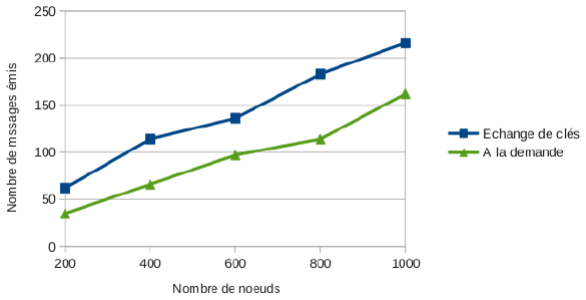

FIGURE 4.6 – Comparaison entre nombre de messages émis lors de l'échange de clés

2. Nombre de paquets émis en présence ou absence d'intrus

   Une étude sur le nombre de paquets émis est présentée si-dessous en tenant compte du cas d'absence d'intrus, dans la figure (4.7), et la présence de 10 % d'intrus du nombre de nœuds du réseau, figure (4.8). Il s'agit des messages émis dans la phase d'échange de clés.

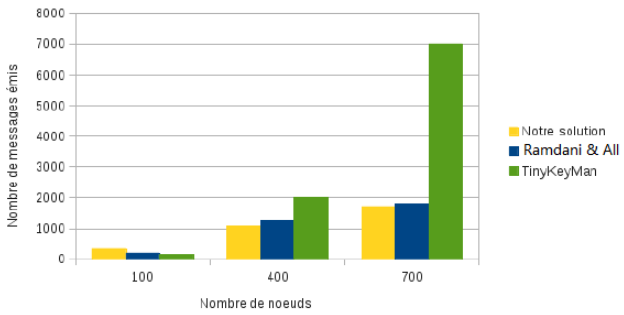

FIGURE 4.7 – Comparaison entre nombre de messages émis en absence d'intrus

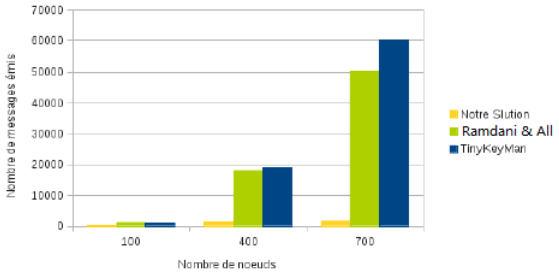

FIGURE 4.8 – Comparaison entre nombre de messages émis en présence d'intrus

La première figure représente une légère différence dans le nombre de paquets émis avec la solution de comparaison Ramdani and Al, cependant on constate une très grande différence par rapport à la solution TinyKeyMan résultant de la méthode d'échange de clés à la demande. L'efficacité de notre solution se constate par le nombre du paquets résultants à la présence de 10 % d'intrus du nombre de nœuds du réseau, qui est dû à la détection immédiate d'un nœud malveillant dès qu'il aura dépassé un nombre précis du messages durant sa période de slot ou par les traitements de vérification d'identité du nœud effectué par la station de base.

4.8 Conclusion

Dans ce chapitre, une présentation de la solution implémentée, de son architecture, son schéma de gestion, a été faite suivie par une simulation des résultats obtenus en les comparant avec deux autres solutions existantes et ayant des paramètres similaires. Les résultats ont démontré l'efficacité de notre solution vis à vis du nombre de paquets émis durant les phases de gestion de clés en assurant une meilleure connectivité et une détection rapide et efficace de nœuds intrus.

Conclusion générale

Conclusion générale

Un réseau de capteur sans fils est un ensemble de nœuds capteurs autonomes, utilisés pour collecter des informations, les traitées de les transmettent vers des unités de calcul. De nos jours, les RCSF sont devenus un domaine de recherche très actif grâce à leurs performances et leurs résultats montrés dans les domaines de leurs utilisation, ils peuvent être implémentés dans des systèmes critiques différents comme les champs de bataille, le domaine de la sante, les bâtiments intelligents, les industries et dans la surveillance de l'environnement. Vu la sensibilité de ces domaines d'applications, la sécurité devient de plus en plus primordiale, des mécanismes de sécurité doivent s'adapter à la nature des réseaux de capteurs en raison des contraintes liées à ce domaine. Durant le cadre de ce projet de fin d'étude, nous avons proposé un schéma de gestion de clés dynamiques pour un réseau de capteur sans fil à mobilité , en se basant sur un système cryptographique basé sur les courbes elliptiques (ECC) en prenant compte des contraintes et ressources limitées des réseaux de capteurs. Ainsi, les résultats de simulation ont démontré l'efficacité de notre solution vis à vis du nombre de paquets émis durant les phases de gestion de clés en assurant une meilleure connectivité et une détection rapide et efficace de nœuds intrus.

Perspectives

Pour la suite de ce projet, nous avons envisagé les perspectives suivantes :

1. Le choix du CH basé sur la plus courte distance entre le nœud membre et le CH et non la profondeur, afin d'économiser l'énergie de transmission à grande distance.
2. Régler le problème engendré par le choix du CH, basé sur la probabilité inférieure à celle déterminé par la SB, qui est la possibilité d'avoir un round sans CH, y remédier en se basant sur le principe du nœud isolé.
3. Le choix du nœud intermédiaire par le nœud isolé, non seulement en se basant sur le nœud le plus proche appartenant au réseau mais aussi sur son énergie résiduelle.
4. Implémentation de notre solution dans un environnement réel.

Bibliographie

[1] Boubiche Jallal Eddine,Une approche inter couche(cross-layer) pour la sécurité dans les RCSFs, en vue d'obtention de diplôme de doctorant en Science en Informatique,Batna : Université Lhadj Lkhedar.

[2] D. Navarro, F. Mieyeville, Simulation de réseaux de capteurs sans fil, Institut des Nanotechnologies de Lyon (INL), 2012.

[1] Said HARCHI, Un protocole de session dans les réseaux de capteurs sans fils,en vue d'obtention du grade de Docteur de l'universite de LORRAINE, 2013.

[4] Rupinder Singh†, Jatinder Singh‡, Ravinder Singh. ATTACKS IN WIRELESS SENSOR NETWORKS : A SURVEY, IJCSMC, Vol. 5, Issue. 5, May 2016

[5] G. Padmavathi, D. Shanmugapriya, A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks, International Journal of Computer Science and Information Security (IJCSIS), Vol. 4, No. 1 & 2, 2009.

[6] Benayed Abdelhak, Implémentation et sécurisation du protocole de routage AODV optimisé pour les RCSFs OAODV, en vue d'obtention de diplôme Master en communication, spécialité Réseaux mobiles et services de télécommunication, Université Aboubakr Belkaïd– Tlemcen, 2016-2017.

[7] Boucheneb Sonia, Tahir Ouissam, Gestion de clés basée sur des clusters dans les réseaux de capteurs sans fils, Mémoire de fin de cycle en vue d'obtention du diplôme de master recherche en Informatique, Option : Réseaux et systèmes distribués, Bejaia :Université A/ Mira de Bejaia,2015-2016.

[8] Wassim Drira, Chakib Bekara, Maryline Laurent. Sécurité dans les réseaux de capteurs sans fil : conception et implémentation. [Rapport de recherche] Dépt. Logiciels-Réseaux (Institut Mines-Télécom- Télécom SudParis) ; Services répartis, Architectures, MOdélisation, Validation, Administration des Réseaux (Institut Mines-Télécom-Télécom SudParis-CNRS). 2008.

[9] Abdul Wahid, Pavan Kumar, A Survey On Attacks, Challenges and Security Mechanisms In Wireless Sensor Network. International Journal for Innovative Research in Science & Technology (IJIRST),Volume 1, Issue 8, January 2015.

[10] Ismail Mansour, Gerard Chalhoub and Michel Misson, Security architecture for multi-hop wireless sensor networks, LIMOS-CNRS, Clermont Universite, 63177 Aubiere cedex, France.

[11] Ismail Mansour, Contribution à la sécurité des communications des réseaux de capteurs sans fil, en vue d'obtention du grade de Docteur, Discipline : Informatique, Université Blaise Pascal - Clermont-Ferrand II, 2013. Français

[12] YANBO SHOU, Cryptographie sur les courbes elliptiques et tolérance aux pannes dans les réseaux de capteurs, en vue d'obtention du grade de Docteur, Spécialité : Informatique, université de Franche-Comté,2014.

[13] NEAL KOBLITZ, ALFRED MENEZES, SCOTT VANSTONE, The State of Elliptic Curve Cryptography. Designs, Codes and Cryptography, 19, 173–193 (2000) °c 2000 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.

[14] Balthazar BAUER Pierre DONAT-BOUILLUD, Victor DURAND, Courbes elliptiques et cryptographie, 2011.

[15] Ramdani Mohamed, Problèmes de sécurité dans les réseaux de capteurs avec prise en charge de l'énergie, Mémoire de Magister , Spécialité : Informatique Répartie et Mobile, Université de Saad Dahlab de Blida, Novembre 2013.

[16] Cedric RAMASSAMY, Analyse des protocoles des réseaux de capteurs sans-fils, en vue de l'obtention du diplôme DOCTORAT DE L'UNIVERSITÉ DES ANTILLES ET DE LA GUYANE, Informatique, Amérique : U.F.R. des SCIENCES,23/11/2012

[17] DJAMA Lynda, MEBARKI Soraya, Protocole de gestion de clés dans les réseaux de capteurs sans fil, en vue d'obtention de diplôme master recherche en informatique, option : Réseaux et Systèmes Distribués, Université A/Mira de Béjaia, 02/07/2016.

[18] Laurent Eschenauer, Virgil D. Gligor, A Key Management Scheme for Distributed Sensor Networks.

[19] Noureddine LASLA, La gestion de clés dans les réseaux de capteurs sans-fil, en vue d'obtention dudiplome Magister en Informatique, option : Informatique Industrielle, Institut National de formation en Informatique (I.N.I) Oued-Smar, Alger, 07/06/2008.

[20] Lalam Mustapha, TinyOs : TinyOs un système opératoire pour de petits capteurs embarqués en réseau. Tizi-Ouzou : UMMTO, 2015-2016.

[21] Simulateur de TinyOs, Philip Levis and Nelson Lee, TOSSIM : A Simulator for tinyos Network ,pal@cs.berkeley.edu,September 17, 2003.

[22] Philip Levis, Nelson Lee, Dennis Chi and David Culler, TOSSIM : Visualizing the Real World, UC Berkeley : NEST Retreat, January 2003.

[23] Chris Merlin ECE 245 | Spring'09, A Tutorial for Programming in TinyOS, WCNG : University of Rochester. January 26, 2009.

[24] MoussaouiOmar, Routage hiérarchique basé sur le clustering : garantie de QoS pour les applications multicast et réseaux de capteurs, Thèse de doctorat en Informatique, Université de Cergy-Pontoise, 11/12/2006.

[25] Berrachedi Amel et Diarbakiri Amina, Sécurisation du protocole de routage hiérarchique LEACH dans les réseaux de capteurs sans fil, Mémoire de fin de cycle pour l'obtention du diplôme d'Ingénieur d'état en Informatique, Option : Systèmes d'informatiques (SIQ), Ecole nationale Supérieure d'Informatique (E.S.I), Oued-Smar, Alger, juin 2009.

[26] M. Zhu et M.D. Liu, An Improvement of LEACH Algorithm Based on Tinyos for Wireless Sensor Network, 2016 6th International Conference on Information Technology for Manufacturing Systems (ITMS 2016), College of Information Science and engineering, East China University of Science and Technology, Shangai 200237, China.

[27] Aleixandre Tudó Carlos, Clustering algorithms for Wireless Sensor Networks and Security threats, Master's Thesis under Eramus programme, Department of Computer Science and Engineering, Division of Distributed Computing and Systems group, Charlmers University of Technology, Göteborg, Sweden 2010.

[28] Diane Ibrahima, Optimisation de la consommation d'énergie par la prise en compte de la redondance de mesure dans les réseaux de capteurs, Thèse de doctorat : MITT : Domaine STIC : Réseaux, Télécoms, Systèmes et architecture, Université de Toulouse 3 Paul Sabatier 17/07/2014.

[29] Yassine Maleh, Pr. Abdellah Ezzati, Etude et développement d'un protocole de dymétrique pour sécuriser les communications des RCSF, laboratoire Veille et Technologies Emergente (LAVETE), faculté des sciences et techniques de Settat, Maroc.

Annexes

Annexe A : Environnement de développement

TinyOs

TinyOs est un système d'exploitation open source conçu pour les réseaux de capteurs intégrées sans fils. Repose sur une architecture basée sur les composants avec un modèle d'exécution piloté par les évènements correspond aux exigences liées aux contraintes de puissance et de mémoire.

Installation de TinyOs

L'installation de TinyOs s'est avérée ne pas être aussi simple que d'installer un simple logiciel ou une application normale.

Nous avons réussir a installer TinyOs 1.x sur une machine réelle Linux, doté de système d'exploitation Debian version 8, et cela n'a jamais fait facilement en raison des problèmes rencontrés lors d'installation des différents packages.

Vous trouvez les étapes d'installation de TinyOs 1.x sur ce site : http ://leme.tagus .ist.utl.pt/gems/PmWiki/index.php/resources/HowToInstallTinyOS

FIGURE 4.9 – Logo de TinyOs

Structure logicielle de TinyOs

Le système d'exploitation TinyOs s'appuie sur le langage de programmation NesC, ce dernier propose une architecture basée sur les composants, ce qui permet de réduire considérablement la taille mémoire de système et de ses applications, telle qu' une application typique est de l'ordre de 15KO dont l'OS de base est aux environs de 400 Octets, ainsi, une grande application ou une BDD sont de l'ordre de 64 KO.

Chaque composant correspond à un élément matériel (LEDs, Timer, ADC,...) et peut être réutilisé dans différentes applications. Les composants peuvent être des concepts abstraits ou bien des interfaces logicielles en E/S matérielles de la cible étudiée (carte ou dispositif électronique (mote)). L'implémentation des composants s'effectue en déclarant des taches, des commandes ou des événements. Les commandes et les événements sont des mécanismes de communication inter-composant, tandis que les tâches sont utilisées pour exprimer la concurrence intra-composant. Lors de l'appel d'une tâche, cette dernière est ajoutée a une file de type FIFO.les tâches s'exécutent de l'ordre de cette file car TinyOs ne dispose de mécanisme d'interruption entre les taches. Lorsque la file d'attente est vide, cela signifie qu'aucune tâche n'est exécutée et donc TinyOs met le capteur en veille, permettant d'économiser son énergie. L'ordonnanceur TinyOs dispose d'une file d'attente FIFO disposant de 7 tâches et deux niveaux de priorité (bas pour les tâches et haut pour les événements). Une commande est une requête au composant pour réaliser un service, comme l'initialisation à la lecture, tandis qu'un événement signale la terminaison d'un service.

TinyOs peut être installé sur différentes plateformes : Windows, Linux, MAC.

NesC

NesC est le langage de programmation utilisé par TinyOs, syntaxiquement proche du langage C

Les principaux caractéristiques de NesC

Les fichiers de NesC sont classés en trois types : interfaces, modules et configurations.

— Les interfaces définissent un ensemble de fonctions pouvant être utilisées de manière bidirectionnelle par n'importe quel composant.Les composants peuvent uniquement être liés les uns aux autres en utilisant et en implémentant des interfaces. Ces fonctions sont alors implémentées par le fournisseurs ou l'utilisateur de l'interface pour distinguer celles qui concernent les commandes de celles qui concernent les évènements . Toute fonction est précédée de Command ou Event, voici un exemple d'application :

```
interface sendMsg{
    // send a message
    command result_t send(uint16_t adresse,uint8_t lenght, TOS_MsgPtr msg );

    // an event indicating the previous message was sent
    event result_t sendDone(TOS_MsgPtr msg, result_t success);
    }
```

FIGURE 4.10 – Exemple d'une interface

Voici deux exemple d'interface ; une pour appeler une commande et une autre pour signaler un evenement :

Appeler une commande : call Send.send(1, sizeof(Message), &msg) ;

signaler un évènement : signal Send.sendDone(&msg, SUCCESS) ;

— Les modules sont les éléments de base de la programmation, ils permettent d'implémenter les composants et sont distingués par l'extension nom_composant.**M**

```
module AMStandard {
provides { interface SendMsg[uint8_t id];
        }
uses {event result_t sendDone(); }
}
implementation {
task void sendTask() {

signal sendDone(); signal SendMsg.SendDone();
}
command result_t SendMsg.send[uint8_t id](uint16_t addr,
uint8_t length, TOS_MsgPtr data) {

post sendTask();

return SUCCESS;
}
default event result_t sendDone() { return SUCCESS;}
```

FIGURE 4.11 – Exemple d'un module

— Les configurations sont utilisées pour assembler d'autres composants ensemble, elles se chargent d' unir les différents composants en fonctions des interfaces alors elles décrivent les liaisons entre les différents composants implémentés. Les fichiers de configuration ont une extension **.nc**

```
configuration GenericComm {
  provides {
        interface StdControl as Control;
        interface SendMsg[uint8_t id]; //parameterized by active message id
        interface ReceiveMsg[uint8_t id];
  command uint16_t activity();
          }
  uses { event result_t sendDone();}
       }
implementation {

components AMStandard, RadioCRCPacket as RadioPacket, TimerC,
NoLeds as Leds, UARTFramedPacket as UARTPacket,
HPLPowerManagementM;

Control = AMStandard.Control;
SendMsg = AMStandard.SendMsg;
activity = AMStandard.activity;
AMStandard.TimerControl -> TimerC.StdControl;
AMStandard.ActivityTimer -> TimerC.Timer[unique("Timer")];
}
```

FIGURE 4.12 – Exemple d'un fichier de configuration

Annexe B : Environnement de simulation

Pour mieux comprendre le comportement individuel des capteurs et le fonctionnement global du réseau, la simulation est une phase importante et nécessaire qui doit être exécutée avant le déploiement réel des nœuds capteurs.

il existe de nombreux simulateurs dans la littérature, chacun avec ses propriétés et caractéristiques. Le simulateur TOSSIM parmi les premiers simulateurs utilisés avec TinyOs, il permet :

- De simuler les applications TinyOs, il simule TinyOs et son exécution
- Tossim est un simulateur d'évènement, plutôt que de compiler une application pour un capteur, l'utilisateur compile dans cet environnement et tourne sur PC
- Il permet de tester et d'analyser des algorithmes dans un environnent contrôlé.
- Il peut être utilisée avec une interface graphique : TinyViz, qui permet de visualiser le comportement des nœuds capteurs.

Description de TinyViz

TinyViz est une application qui offre un aperçu graphique du comportement des nœuds aux seins de réseau, sans avoir a déployer les nœuds dans la nature.

elle permet l'analyse du réseau en basculant entre les différents modes disponibles. La figure 4.5 représente l'interface graphique de TinyViz

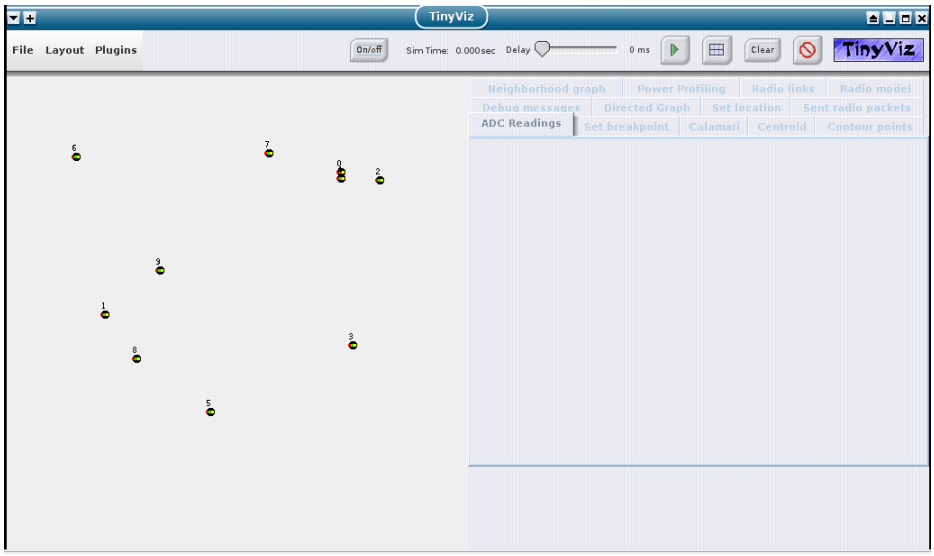

FIGURE 4.13 – Interface graphique de TinyViz

Dans la plus en haut (la partie superieure) on trouve toute les commandes qui peuvent intervenir dans la phase de simulation, elles sont décrites comme suit :

- On/Off : Met en marche ou éteint un capteur.
- Delay : Permet de préciser (sélectionner) la durée au bout de laquelle se déclenche le Timer.
- Play : Permet de lancer la simulation, ou la mettre en pause.
- Grilles : Permet d'avoir une grille pour situer les capteurs dans l'espace.
- Clear : Pour effacer tous les messages qui transitent entre les capteurs.
- Arrêt : Permet la fin de simulation.

La partie la plus à gauche, représente la zone d'aperçu où les résultats de simulation sont affichés. Elle décrit les capteurs selon la topologie utilisée. Et la partie plus à droite en dessous de la partie supérieure, représente la liste des plugins disponibles pour visualiser la simulation. Les plugins dont on s'est beaucoup servi sont « Debug messages » pour afficher tous les messages de type « dbg » afin de vérifier la nature des messages, et « Radio links » qui nous permet de voir, avec des flèches ou des cercles, si un capteur est en train d'émettre ou non.

Les échanges entre les capteurs peuvent être de deux manières :

- Unicast : L'échange de données se fait uniquement entre deux capteurs,
- Broadcast : Message émis par un capteur à l'ensemble des capteurs du réseau.

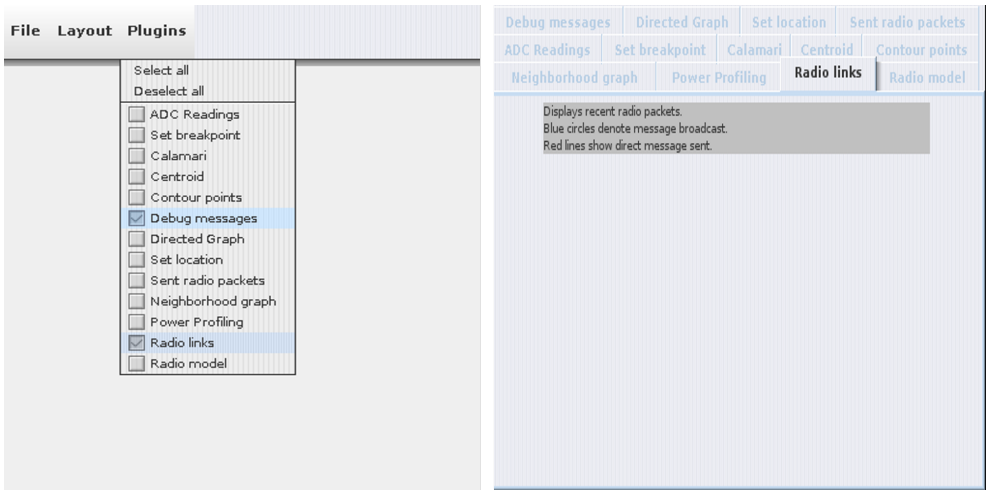

FIGURE 4.14 – L'interface de liste des plugins

Annexe C : Le protocole de routage LEACH

LEACH : (Low Energy Adaptive Clustering Hierarchy) est le protocole le plus populaire et le plus utilisé dans les réseaux de capteurs sans fil, considéré comme étant le premier protocole de routage hiérarchie basé sur les clusters. Il regroupe à la fois l'efficacité en consommation d'énergie et la qualité de l'accès au media. Il se base sur la décomposition du réseau en clusters et l'utilisation du concept d'agrégation de données pour une meilleure performance en termes de durée de vie. La construction de clusters est basée sur les zones là où il y a un fort signal reçu, l'utilisation de cluster permet de réduire et d'économiser la consommation énergétique. Le choix de cluster head se fait d'une manière aléatoire, chaque cluster head collecte les informations de ces membre et les transmettre à la station de base

Ce rôle de CH est affecté à chaque nœud du réseau pour garantir une consommation équitable en termes d'énergie, vu que les traitements fait par les CHs sont gourmands en cette ressource. Pour réduire le taux de transmission, le CH se charge de l'agrégation des données reçus depuis les autres nœuds pour transmettre, lui uniquement, le résultat par la suite à la SB.

Chaque round de ce processus est exécuté en deux phases :

1. Phase d'initialisation : Phase d'élection de CHs et de formation de clusters : L'élection d'un nombre déterminé de CH se fait ainsi :
   Chaque nœud choisit un nombre aléatoire compris entre 0 et 1, ce nombre est comparé à un seuil P(t) calculer par la SB par la formule :
   $P(t) = \frac{K}{N-K*(rmod(\frac{N}{K}))}$Où K : est le nombre de CH désirés. Et $N - K * (rmod(\frac{N}{K}))$est le nombre de nœuds éligibles d'être CHs.
   La particularité de LEACH est qu'il ne nécessite pas une communication entre les nœuds pour déterminer les CHs, ils sont choisis à l'aide d'une méthode probabiliste lors de chaque round.
   Un nœud peut recevoir plusieurs annonces de CHs, son choix se portera sur le CH le plus proche et donc celui qui dispose du plus puissant signal reçu.
2. Phase de transmission : Phase de transmission des données vers la SB.

Dans cette phase, chaque nœud envoie les données collectées au CH auquel il appartient. Ce dernier, se charge des traitements d'agrégations de toutes les données reçus de ses nœuds membres et de transmettre le résultat à la SB en un seul saut, avec un ordonnancement TDMA (Time Division Multiple Access) qui affecte à chaque nœud un intervalle de temps pour la transmission de ces données.

L'avantage de ce protocole est qu'un nœud est actif uniquement lors de la phase de transmission de ces données, sinon il se met en veille, ce qui permet de réduire la consommation d'énergie des nœuds, à part le cluster head qui est toujours actif pour recevoir les données venants des autres nœuds. Ces étapes seront les mêmes pour chaque round, avec une contrainte que un nœud déjà élu cluster head ne sera pas réélu une autre fois.

Attaques et contremesures pour LEACH:

Du fait que Leach n'est pas securisé, il est vulnerables a plusieurs types d'attaques dont nous pouvons citer:

4.8.0.1 Inhibiting node discovery

L'exercion du brouillage sur un réseau de capteurs, peut empecher certains noeuds de faire la decouverte du voisinage, ainsi, il semblera aux autres noeuds que la densité du réseau à diminuer. Un petit nombre de CHs est désigné, ce qui engendera une disipation d'énergie. La contremesure consiste à assurer qu'un noeud reconnaisse toujours les noeuds qu'il a découvert.

4.8.0.2 Cluster set-up channel blocking

Consiste à transmettre des donnés de façon continue sur le canal utilisé par le protocole MAC CDMA, servant aux noeuds non CH à envoyer leurs demandes d'appartenance aux CHs.L'accès bloquer au canal laissera des noeuds capteurs isolés et qui reessayeront toujours d'appartenir à un cluster ce qui affaiblira leurs ressources énergétiques.

4.8.0.3 Forged Base Station

Un attaquant peut usurper l'identité de la SB en utilisant un signal plus fort que celui du puits d'origine. Les CHs vont alors diriger leurs données vers ce noeud malvéillant avec une consommation d'énergie moindre. La contremesure consiste à programmer chaque noeud avec la position de la SB.

4.8.0.4 Spoofed CH

C'est une combinaison entre l'attaque Sink Hole (emettre un puissant signal, provenant d'un noeud malicieu, lui permettant de devenir le CH pour la majorité ou la totalité des noeuds du réseau) et l'attaque Hello Floods (le signal est sous forme de paquets Hello). Ce nouveau CH reçoie tous les messages du réseau, les modiffie ou les supprime ou bien meme cause des collisions entre les transmissions en manipulant l'ordonnanceur TDMA. La contremesure est de ne pas génerer une fréquence de CH plus elever que celle générer par ces antécédents, ou fixer un seuil de fréquence à ne pas dépasser.

4.8.0.5 Supported CH

Amplifier le signal des annonces de CH pour tromper les noeuds dans leur choix de CH le plus proche à eux, ainsi leur choix peut se porter sur un CH éloigné ce qui en gendrera une grande perte d'énergie de transmission ou l'arret de la transmission quand le CH est non attégnable. La contremesure consiste à se rappeler de la force du signal de chaque noeud élligible de devenir CH.

4.8.0.6 Ghost Nodes

Faire passer un noeud malveillant pour un groupe de neouds fontomes afin d'augmenter le nombre de noeuds et le nombre de CH donc causer une perte d'énergie.

4.8.0.7 Brute-force jamming attack

Brouillage d'une partie du réseau, en particulier les canaux de liaison entre les CHs et la SB en utilisant une petite quantité d'énergie. La contre mesure est d'utiliser la technique d'étalemlent du spectre obligeant le noeud malveillant à fournir beaucoup plus d'efforts.

4.8.0.8 Neighbors Interference

Un CH malveillant peut utiliser un meme code CDMA pour causer des interférences entre les voisins de différents clusters lors des communications. Ce type d'attaque fournit le meme résultat que le brouillage des signaux dans une zone.

La majorité des attaques visent les CHs puisqu'ils représentent la maille la plus sensible du réseau en vue de l'importance des roles qu'ils exercent d'agrégation et d'acheminement des données à la SB.