UNIVERSITÉ DE SAAD DAHLAB DE BLIDA

Faculté des Sciences

Département d'Informatique

MÉMOIRE DE MAGISTER

Spécialité : Informatique Répartie et Mobile

PROBLÈMES DE SÉCURITÉ DANS LES RÉSEAUX DE CAPTEURS AVEC PRISE EN CHARGE DE L'ÉNERGIE

Par :

RAMDANI MOHAMED

Devant le jury composé de :

D. Bennouar, Maitre de Conférence, MCA, U. de Blida	Président
N. Boustia, Maitre de Conférence, MCA, U. de Blida	Examinateur
N. Chikhi, Maitre de Conférence, MCB, U. de Blida	Examinateur
M. Benmohamed, Professeur, U. de Constantine	Rapporteur

Blida, Novembre 2013

Remerciements

Tout d'abord et avant tout, je remercie vivement le bon Dieu, le tout puissant qui nous guide le chemin droit, de m'avoir permis d'emprunter le chemin de la recherche et de m'avoir donné suffisamment de courage et de patience pour accomplir ce travail.

Ma reconnaissance et mes remerciements sincères se tournent en premier lieu vers mon encadreur, le professeur Mohamed Benmohammed, qui m'a permis de mener à bien mes travaux. Ce mémoire doit beaucoup à ses conseils précieux, à sa rigueur, à son calme et à sa disponibilité.

J'adresse également mes vifs remerciements aux membres du jury qui ont accepté d'évaluer mon travail, le président du jury Mr Djamel Bennouar et les deux examinateurs Mr Nacim Chikhi et Mlle Narimene Boustia.

Je tiens à remercier chaleureusement mes chers parents d'avoir encadré mon potentiel dès mon jeune âge, mes chères sœurs, mon cher frère et sa femme ainsi toute ma famille, mes amis et proches pour leur soutien inconditionnel et sans faille.

Enfin, je remercie tous ceux qui ont contribué de près ou de loin pour la réalisation de ce travail, tous ceux qui m'ont accompagné et soutenu.

Merci à tous et toutes.

Mes dédicaces spéciales vont particulièrement pour ma nièce Narimene et mes deux neveux Ziane et Fouzi. Que le bon dieu les garde pour nous.

RÉSUMÉ

Un réseau de capteurs sans fils (RcSF) est un réseau ad hoc particulier. Il est utilisé, en général, pour contrôler un environnement particulier. Il est constitué d'un ensemble de capteurs communicants par des liaisons sans fil. Les RcSF interviennent dans des applications particulières : militaires, médicales, environnementales, pour la surveillance des infrastructures critiques dans des zones sinistrées et hostiles.

Une des contraintes principales dans les réseaux de capteurs sans fil est la protection des communications. Le prolongement de la durée de vie du réseau par le déploiement des protocoles de routage et de sécurité adéquats permet une gestion efficace de l'énergie. Le rechargement des batteries, dont la capacité est limitée, dans des zones hostiles est souvent impossible. Pour cela, les RcSF nécessitent des mécanismes de sécurité efficaces et peu couteux en énergie.

La plupart des protocoles de gestion de clés proposés dans la littérature se basent sur des mécanismes de chiffrement symétrique. Une grande partie de ces protocoles utilise la méthode de pré-distribution de clés. Dans ce mémoire, nous avons étudié, dans un premier temps, les techniques cryptographiques proposées dans la littérature ainsi que les différents protocoles de gestion de clés existants et nous avons proposé par la suite une solution qui pourrait satisfaire conjointement aux deux contraintes majeures liées à l'établissement de communications sécurisées et à la gestion efficace de l'énergie.

ABSTRACT

A Wireless Sensors Network (WSN) is a special ad hoc network. It is used in general to monitor a particular environment. It consists of a plurality of sensors communicating with wireless links. The WSNs are involved in specific applications: military, medical, environmental, for the monitoring of critical infrastructure in the affected and hostile areas.

One of the major constraints in wireless sensor networks is the protection of communications. Extending the life of the network through the deployment of routing protocols and adequate security enables efficient energy management. Recharge the batteries, which the capacity is limited in hostile areas, is often impossible. For this, the WSN security mechanisms require efficient and inexpensive energy.

Most key management protocols proposed in literature are based on symmetric encryption mechanisms. Most of these protocols use the pre-distribution of keys. In this project, we studied, in a first time, cryptographic techniques proposed in the literature and various protocols of management key existing and we subsequently proposed a solution that satisfies both the major constraints linked to establishment of secure communications and efficient energy management.

LISTE DES ILLUSTRATIONS, GRAPHIQUES ET TABLEAUX

TABLE DES MATIERES

INTRODUCTION

Les récentes avancées réalisées dans les domaines de la micro-électronique et des technologies de communication sans fil ont permis de créer de petits appareils de cout raisonnable, de taille minuscule et autonomes appelés micro-capteurs ou capteurs. Ces dispositifs multifonctionnels, considérés comme de véritables systèmes embarqués, sont équipés d'une unité de capture, une unité de calcul, une unité de stockage et d'une unité radio pour communiquer avec le monde extérieur.

Les nœuds capteurs collaborent et s'auto-organisent pour former un réseau de capteur sans fil (RcSF) capable de superviser son environnement de déploiement souvent hostile, inaccessible et sans aucune intervention humaine, ce qui peut s'avérer très utile pour de nombreuses applications militaires, civiles, environnementales, agricoles, médicales, industrielles, domestiques…etc. La collaboration des nœuds capteurs a pour objectif d'acheminer les données captées à partir du champ de captage vers une destination finale sur le réseau (souvent une ou plusieurs stations de base). La forte densité de ce type de réseaux favorise un mode de communication en multi-sauts économe en énergie et en nombre de paquets échangés, et favorise également le partitionnement du réseau en plusieurs niveaux hiérarchiques. Plusieurs solutions sont proposées dans la littérature pour assurer un routage efficace d'informations en contournant les nombreuses contraintes influençant le bon fonctionnement du réseau. Deux de ces contraintes sont essentielles : i) assurer une consommation raisonnable d'énergie afin de prolonger la durée de vie des nœuds capteurs et par conséquent la durée de vie du réseau et ii) garantir des communications sécurisées où les utilisateurs légitimes sont authentifiés et l'information véhiculée est fraiche, disponible et confidentielle.

Les ressources limitées des capteurs, l'absence de la sécurité physique et la nature vulnérable des communications sans fil sont des caractéristiques qui peuvent augmenter le niveau de risque d'attaques ayant pour objectif la récupération des informations sensibles et confidentielles circulant sur le réseau afin de nuire au fonctionnement global du système. Ces caractéristiques, entre autres, rendent les solutions de sécurité développées pour les réseaux filaires ou sans fil inapplicables aux RcSF. Par conséquent, plusieurs domaines de recherche sont apparus ces dernières années proposant des solutions de sécurité capables de

remédier aux insuffisances des nœuds capteurs et aux vulnérabilités du médium de communication sans fil utilisé.

Parmi ces solutions adéquates l'emploi des mécanismes cryptographiques notamment la cryptographie à clé secrètes (symétrique) moins gourmande en ressources par rapport à la cryptographie à clés publiques (asymétrique). Le principe de la cryptographie symétrique repose sur le partage d'une clé commune entre chaque paire de nœuds désirant communiquer. Après le déploiement des nœuds, l'établissement des clés partagées est très problématique dans le contexte des RcSF et la plupart des mécanismes de gestion de clés sont vulnérables aux types d'attaques par compromission. La plupart des solutions de gestion de clés existantes se basent sur la méthode de pré-distribution de clés avant le déploiement. Elles peuvent être classées selon la façon de partager les clés communes (déterministes ou probabilistes), selon l'emplacement des nœuds (géographiques) ou selon le nombre minimum (t) de nœuds que l'attaquant doit compromettre pour casser le système de cryptage du réseau (dits t-secure).

La solution que nous avons proposée est un protocole de gestion de clés déterministe et distribué, utilisant un mécanisme particulier basé sur la cryptographie sur les courbes elliptiques (ECC) afin de résoudre le problème de la clé partagée en respectant les contraintes et caractéristiques des réseaux de capteurs sans fil.

Contribution

Dans ce travail, nous avons développé une approche de sécurité et un mécanisme de gestion de clés qui permet d'installer des clés secrètes entre chaque paire de nœuds du réseau, en présence ou non de nœuds malicieux ou compromis et en respectant les limites physiques et énergétiques des nœuds capteurs. Les clés installées et partagées par les nœuds sont utilisées par le protocole de routage proposé afin de construire une topologie de communication sécurisée garantissant un échange réduit de paquets de données et une consommation d'énergie optimisée. Les attaques sont détectées et éliminées par les nœuds. De plus, notre solution propose un mécanisme de mise à jour de clés permettant d'augmenter la tolérance aux pannes et aux intrus.

Notre contribution est récapitulée dans les points suivants :

- Notre schéma de gestion de clés permet de conserver un maximum d'énergie dissipée dans le processus de découverte de voisinage et d'installation de clés secrètes en réduisant le nombre de paquets de données et en optimisant les calculs des paramètres et des clés.

- Notre protocole de sécurité du routage sécurise tous les types de liens (nœud-nœud, nœud-nœud chef, nœud chef-nœud puits) entre les entités du réseau pour n'importe quel type de topologie (plate ou hiérarchique), ce qui permet de garantir l'authentification des nœuds légitimes du réseau, la fraicheur, la confidentialité, l'intégrité et la disponibilité des données.
- Une solution implémentée par un code simple et portable qui ne nécessite aucun matériel supplémentaire.

Organisation du mémoire

- Le premier chapitre présente un aperçu général sur les réseaux de capteurs sans fil.
- Le deuxième chapitre présente un état de l'art sur le routage de données dans les RcSF. Plusieurs architectures et contraintes sont discutées, des solutions et des métriques de performance sont présentées.
- Le troisième chapitre est entièrement consacré à la sécurité du routage dans les RcSF. Une taxonomie d'attaques et quelques mécanismes de sécurité sont présentés.
- Dans le quatrième chapitre, nous étudions quelques outils cryptographiques et quelques protocoles de gestion de clés dédiés aux réseaux de capteurs sans fil.
- Le cinquième chapitre présente notre contribution pour la problématique d'établissement de clés symétriques pour chaque paire de nœuds communicants, et notre approche de sécurisation des communications. On présente les principes et les objectifs de notre solution ainsi que son implémentation.
- Le dernier chapitre présente des résultats et des performances en comparant notre protocole proposé au protocole TinyKeyMan. Nos résultats montrent l'efficacité de notre protocole en termes de gestion de clés, de sécurité, du nombre de paquets échangés et de la consommation d'énergie par rapport à son vis-à-vis.

CHAPITRE 01
GENERALITES SUR LES RCSF

Ce chapitre est consacré à des généralités sur les réseaux de capteurs sans fil (RcSF). Nous allons aborder des définitions, des notions générales, des caractéristiques, des domaines d'application, l'historique, l'architecture et les contraintes de conception de ce type de réseaux Ad Hoc particulier.

1. Introduction

Les récents progrès et les nombreuses avancées technologiques dans les domaines de la micro-électronique et les progrès atteints dans les domaines d'intégration et de la miniaturisation ont permis la fabrication d'entités miniaturisées, communément appelées capteurs, faibles en cout et de plus en plus performantes, capables de se disposer dans l'environnement de manière aisée sans altération du paysage ambiant. Egalement, le progrès des technologies de communication sans fil a permis aux capteurs d'être plus autonomes ce qui favorise un déploiement facile et rapide dans des endroits difficile d'accès ou complètement inaccessibles. Sur un champ de captage, les capteurs coopèrent entre eux sans aucune intervention externe (humaine, infrastructure de base…etc.) pour former une infrastructure de communication dite réseau de capteurs sans fil [8].

Un réseau de capteurs sans fil est composé de plusieurs milliers de capteurs communicants via des liaisons sans fil placés dans des endroits précis ou dispersés aléatoirement sur une zone à surveiller, capables de collecter, traiter et s'auto-organiser afin de transmettre des informations sur leur environnement [65].

2. Définition et architecture d'un RcSF

Un Réseau de Capteur Sans Fil (RcSF ou WSN pour Wireless Sensor Network en anglais) est un type particulier de réseaux Ad Hoc [7] dédié à une application spécifique telle que la surveillance des troupes ennemies sur un champ de bataille, surveillance médicale, détection des feux de forets, suivi du mouvement des animaux, analyse de la qualité de l'air, météo…etc. Il est composé d'un ensemble de dispositifs très petits nommés nœuds capteurs et d'une ou de plusieurs stations de base appelées « Sink nodes » ou nœuds puits qui sont considérés comme l'interface entre le réseau de capteurs et l'utilisateur final. (Figure 1.1).

Les nœuds capteurs, dont le nombre peut atteindre des dizaines de millions d'éléments pour certaines applications, sont des entités caractérisées par leur cout très réduit, leur taille minuscule généralement en quelques millimètres de volume et leurs ressources limitées en calcul, en mémoire et notamment en énergie. Ils sont déployés sur une zone de capture, soit aléatoirement (largage par avion ou par hélicoptère par exemple) ou d'une manière déterministe en choisissant leurs emplacements, dans le but de collecter des données de leur environnement telles que les grandeurs physiques comme l'intensité de la luminosité, la température, l'humidité, les vibrations…etc., et de les router vers la station de base. Ils participent en conséquence à un partage organisé d'informations par des traitements coopératifs.

La station de base, jouant à la fois le rôle de collecteur final et de passerelle vers d'autres réseaux, sert à collecter l'ensemble des informations provenant des nœuds capteurs et de les transmettre par d'autres moyens (réseau filaire, internet, satellite…etc) à un utilisateur final. De plus, l'utilisateur final peut utiliser la station de base comme une passerelle pour diffuser ses requêtes sur le réseau (voir figure 1.1).

2.1. <u>Anatomie d'un nœud capteur :</u> Un nœud capteur est doté, principalement, de quatre unités [7] [85] (Figure 1.2) :

2.1.1. <u>Unité de captage ou d'acquisition</u>

Cette unité est responsable de la collecte des données. Elle est constituée de deux dispositifs : un dispositif de transformation des données interceptées en signaux analogiques et un dispositif de conversion de ces signaux analogiques en signaux numériques compréhensibles par l'unité de traitement.

2.1.2. <u>Unité de traitement</u>

Composée d'un microprocesseur et d'une mémoire de stockage, son rôle est d'effectuer des traitements sur les données captées en exécutant les commandes et les instructions contenues dans les protocoles de communication pré-chargés sur le capteur. Elle est équipées de deux interfaces : i) interface par unité d'acquisition et ii) interface par unité de communication.

2.1.3. <u>Unité de communication</u>

Elle est responsable des émissions et réceptions des données sur le medium de communication.

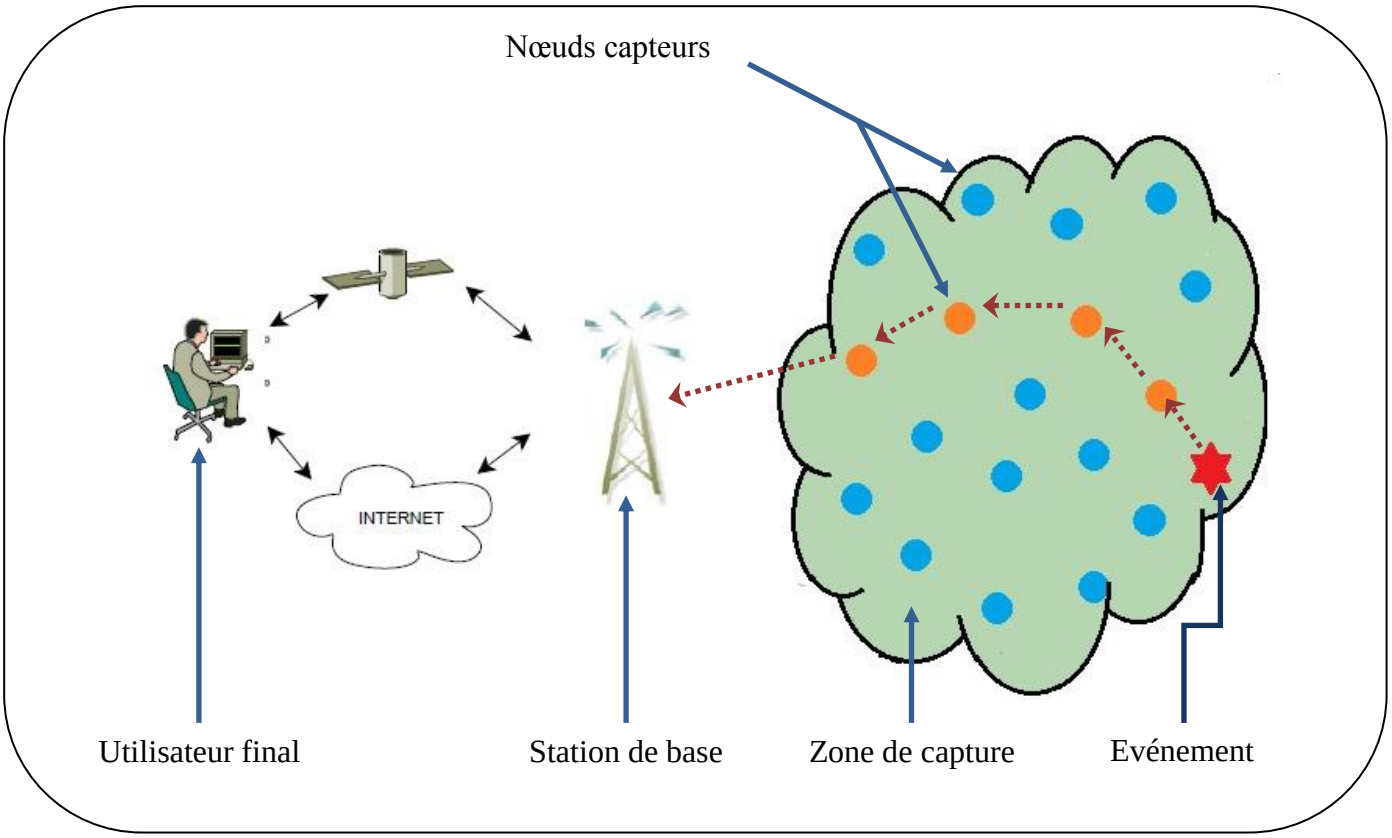

Figure 1.1 : Exemple d'un réseau de capteurs sans fil

2.1.4. Unité d'alimentation

Composée généralement d'une ou de plusieurs batteries souvent irremplaçables et non rechargeables ayant des ressources énergétiques limitées. Cette unité est responsable de gérer l'alimentation en énergie de tous les composants du nœud capteur.

Mise à part ces quatre unités, un nœud capteur peut être doté par d'autres composants tels que le GPS (système de positionnement) et/ou d'un mobilisateur (pour d'éventuels déplacements).

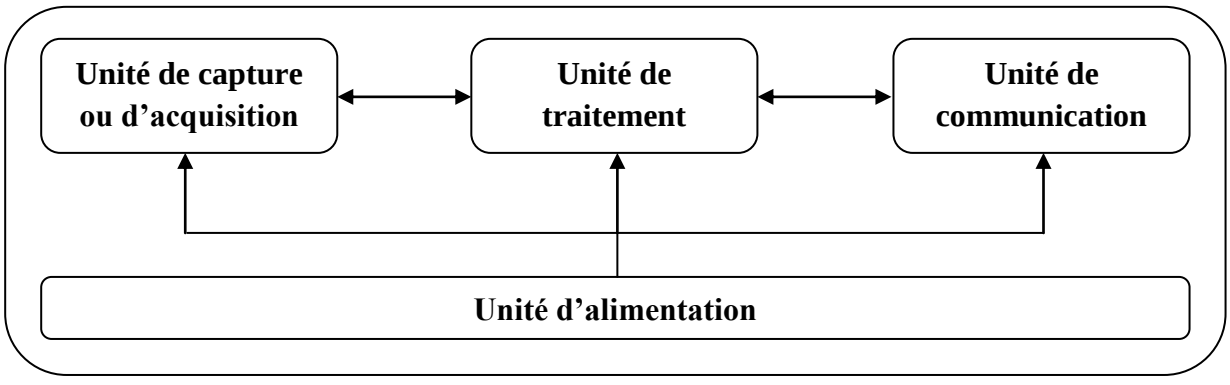

Figure 1.2 : Anatomie d'un nœud capteur sans fil [7]

Plusieurs modèles de capteurs sont disponibles sur le marché, les plus utilisés sont : Micaz, développé par l'université de Berkeley, Imote2, TelosB, commercialisés par la société Crossbow Technology…etc.

2.2. Architecture d'un réseau de capteurs sans fil : Selon [20], il existe deux types d'architecture pour les réseaux de capteurs sans fil : Les RcSF plats et les RcSF hiérarchiques.

2.2.1. Les réseaux de capteurs sans fil plats

Un RcSF plat est un réseau homogène où tous les nœuds ont les mêmes ressources en terme d'énergie, de calcul et de mémoire. Cette architecture est utilisée pour des réseaux ayant un déploiement massif des nœuds (plusieurs nœuds capteurs / m^2) et un mode de communication sans fil en multi-sauts (c'est-à-dire que l'information envoyée par un nœud récolteur doit transiter par plusieurs nœuds intermédiaires avant d'atteindre sa destination finale sur le réseau et sans aucun traitement supplémentaire sur la donnée transportée). Ces deux conditions rendent le passage à l'échelle très critique et la consommation d'énergie pour le routage des informations énorme.

2.2.2. Les réseaux de capteurs sans fil hiérarchiques

Contrairement au type d'architecture présenté ci-dessus, un RcSF hiérarchique est un réseau où tous les nœuds ne possèdent pas les mêmes rôles et par conséquent les mêmes ressources. En effet, l'introduction d'un certain nombre de nœuds puissants permet de décharger une grande partie des nœuds ordinaires de plusieurs taches du réseau. Ainsi, les taches complexes et gourmandes en ressources sont exécutées par les nœuds intégrés et les taches de base (comme la capture) sont assumées par les nœuds ordinaires.

3. Histoire des Réseaux de capteurs Sans Fil

L'apparition des réseaux de capteurs ne date pas d'aujourd'hui. Leur première apparition fut à la fin des années 1970 et au début des années 1980. Jusqu'aux années 1990, il fallait un câblage encombrant et couteux pour acheminer les données d'un capteur vers un point de collecte central. Les progrès du domaine des technologies sans fil (Wireless) a permis aux réseaux de capteur d'avoir une autre dimension et un champ d'application vaste et varié.

Les premiers pas étaient l'œuvre de recherches menées par l'armée américaine dans le but de concevoir des entités autonomes, servant de support technique pour les soldats, capables de surveiller les zones de combat. Effectivement, la croissance spectaculaire des RcSF a commencé avec l'implication de l'agence américaine pour les projets de recherches

avancée de défense (DARPA ; Defense Advanced Research Projects Agency) dans le financement des projets de recherches et de développement sur les RcSF notamment le programme SensIT (Sensor Information Technology) [29] de 1999 à 2002.

A partir des années 2000, les universités et instituts, entre autre Stanford, Berkeley et Massachusetts Institute of Technology, ont démocratisé leur développement dans tous les domaines pour constituer aujourd'hui l'une des technologies les plus influentes de notre vie quotidienne. On parle [22] de trois générations de RcSF :

Générations / Caractéristiques	1ere Génération 1980 - 1990	2eme Génération 2000 – 2003	3eme Génération 2010
Energie (autonomie)	Grosses batteries (1 Heure à 1 Jour)	Batteries AA (1 jour à 1 semaine)	Solaire (1 mois à plusieurs années)
Taille	Grande boite à chaussures	Boite de cartes	Particule de poussière
Poids	Kilogrammes	Grammes	Négligeable
Technologie	En étoile (un seul saut)	Client / server, peer to peer	En Multi-sauts (peer to peer)
Déploiement	Un seul capteur	Mise en place à la main	Embarqué

Tableau 1.1 : Générations des RcSF [22]

4. Applications des Réseaux de capteurs Sans Fil

Les réseaux de capteurs sans fil (RcSF) ont un champ d'application vaste et diversifié. Ceci est rendu possible par leur cout faible, leur taille réduite, le support de communication sans fil utilisé et la large gamme des types de capteurs disponibles. Un autre avantage est la possibilité de s'auto-organiser et d'établir des communications entre eux sans aucune intervention humaine, notamment dans des zones inaccessibles ou hostiles, ce qui accroit davantage le nombre de domaines ciblés par leur application (environnement, catastrophes naturelles, bâtiments intelligents, la santé, l'agriculture, l'industrie…etc.). Nous présentons dans ce qui suit les domaines les plus ciblés par les RcSF [7] :

- **Domaine militaire :** Les RcSF sont le résultat de la recherche militaire. Ils sont utilisés dans la surveillance des champs de bataille pour connaitre exactement la

position, le nombre, l'armement (chimique, biologique, nucléaire…etc.), l'identité et le mouvement des soldats et ainsi empêcher leur déploiement sur des zones à risques.

Exemple d'application : Le déploiement d'un réseau de capteurs sans fil sur les cotes Est et Ouest des Etats-Unis destiné à repérer les sous marins silencieux et les navires, notamment soviétiques, s'approchant du territoire américain [22] [105].

- **Domaine civil :** Apparus dans plusieurs contextes notamment dans la surveillance des habitations (concept de bâtiments intelligents), des infrastructures, des installations et des zones à risques. Leur utilisation permet de réduire considérablement le budget consacré à la sécurité des humains tout en garantissant des résultats sûrs et fiables.

- **Domaine agricole et environnemental :** Les réseaux de capteurs sans fil sont très utiles dans la protection de l'environnement. Ils peuvent être utilisés pour la détection des feux de forets, des inondations, surveillance des volcans, contrôle de la qualité de l'air par le suivi de l'évolution de la densité moyenne de Co_2 [8], le déplacement des animaux…etc. Dans le domaine agricole, on cite le déploiement des capteurs sur un champ agricole afin d'identifier les zones sèches et permettre leur irrigation à temps.

  Exemple d'application : Le projet CORIE [106]. On déploie des capteurs temps-réel pour récolter des informations visant à fournir des indications objectives sur la variabilité spatiale et temporelle de la partie inférieure de la rivière Columbia.

- **Domaine industriel :** Le suivi des chaines de production dans une usine, détection des dysfonctionnements de machines, suivi du mouvement des marchandises dans les entrepôts de données, suivi du courrier, des colis expédiés…etc. sont, entre autres, des exemples concrets d'application des RcSF dans le domaine industriel.

- **Domaine de la santé :** Un moyen très efficace pour le domaine médical et le suivi temps-réel de l'état des patients, notamment ceux atteints de maladies chroniques, ils permettent de collecter des informations physiologiques de meilleure qualité [3] [12] [67] pouvant être stockées pour une longue durée ou alors détecter des comportements anormaux chez des personnes âgées ou handicapées comme les chutes, les chocs, les cris…etc.

  Exemple d'application : Utilisation de gélules multi-capteurs pour la transmission d'images de l'intérieur du corps humain sans avoir recours à la chirurgie [25].

- **Applications domestiques :** Les RcSF sont l'un des moyens les plus importants dans la lutte contre le réchauffement climatique. En effet, l'intégration des capteurs dans

des murs ou sur les plafonds des maisons permet d'économiser un maximum d'énergie en gérant au mieux l'éclairage et le chauffage en fonction de la localisation des personnes. Egalement, les capteurs peuvent être embarqués dans des appareils électroménagers (aspirateurs, micro-ondes, climatiseurs, réfrigérateurs…etc.) et d'interagir entre eux et avec un réseau externe pour assurer un meilleur contrôle à distance de ces appareils par leur propriétaire [77].

La liste des applications des RcSF est non exhaustive. Ils peuvent être utiles dans d'autres domaines comme la sécurité alimentaire, les télécommunications, la robotique ou dans des applications traditionnelles (automobile, aéronautiques, applications commerciales…etc.).

5. Notions fondamentales

5.1. Modèle en couche de la pile réseau

Les réseaux de capteurs sans fil (RcSF) sont des réseaux qui se basent sur un support de communication sans fil. Cette solution de communication doit être couplée avec les contraintes matérielles et énergétiques des capteurs.

Les standards de communication utilisées pour les réseaux filaires et pour les autres types de réseaux sans fil (WiFi, WiMAX, Bluetooth…etc.) ne sont plus adaptés aux RcSF car les mécanismes de communication et de sécurité développés pour ces standards ne prennent pas en considération les ressources limitées des capteurs au niveau des couches protocolaires. Ainsi, il était primordial de développer une nouvelle norme de communication garantissant, à la fois, une fiabilité maximale et un respect aux limites des capteurs essentiellement celle liée à la consommation d'énergie. Cette norme est communément appelée IEEE 802.15.4 [45].

La norme IEEE 802.15.4 concerne uniquement les deux premières couches du modèle OSI : la couche physique et la couche liaison de données. Parmi ses caractéristiques : i) Trois types de débit (20 Ko/S, 40 Ko/S et 250 Ko/S) ; ii) Distance maximale de 100 mètres entre deux entités communicantes ; iii) Utilise le protocole CSMA/CA pour l'accès au canal de transmission.

La norme IEEE 802.15.4 est exploitée principalement par les réseaux personnels sans fil, appelés également réseaux individuels sans fil ou réseaux domestiques sans fil (WPAN pour Wireless Personal Area Network). Elle a les avantages suivants :

- Garantit un maximum de fiabilité dans les transferts de données entre les entités communicantes.

- Offre un accès déterministe au support au niveau de la couche liaison de données ce qui avantage les applications nécessitant des garanties de délais.
- Inclut des mécanismes de détection d'intrusions au niveau de la couche physique.
- Empêche les collisions de données au niveau de la couche liaison de données.
- Garantit un faible coût et une consommation optimisée de l'énergie.

Bien que cette norme offre des avantages énormes pour les RcSF, elle demeure incapable d'établir des communications en multi-sauts, ce qui est insuffisant pour ce genre de réseaux.

Afin de palier à cette insuffisance, plusieurs protocoles de routage ont été développés au niveau de la couche réseau en se basant sur les couches physiques et liaison de données de la norme IEEE 802.15.4. ZigBee [107] est le meilleur exemple de cette suite de protocoles de haut niveau. Les principales tâches des couches de la pile protocolaire [8] sont données par le tableau suivant :

Couche	**Taches**
Physique	Codage, filtrage, contrôle de puissance, sélection de canaux, émission et réception des données sur le support physique, modulation.
Liaison de données	Méthode d'accès au canal, retransmission des données, validation des trames, mécanismes de cryptage, contrôle de puissance.
Réseau	Routage, découverte des voisins, sécurité du transfert de données, gestion des tables de routage, allocation de ressources.
Transport	Contrôle de flux, retransmission des données.
Application	Codage, compression, fonction d'agrégation, collecte.

Tableau 1.2 : Taches de la pile protocolaire

5.2. Notion de routage

L'acheminement des données entre le nœud collecteur et la station de base via un réseau de connexion est une tâche difficile qui nécessite un travail de collaboration de l'ensemble des nœuds capteurs participant à ce transfert. Pour un acheminement optimal, les fonctions de routage (procédures, protocoles…etc.) doivent tenir compte des ressources limitées des nœuds notamment de leur niveau d'énergie.

Parmi les facteurs qui affectent le processus de routage, le nombre important des nœuds du réseau, leurs contraintes matérielles et énergétiques, l'environnement de déploiement et la topologie du réseau. Pour y remédier, plusieurs protocoles de routage ont été proposés dans la littérature comme des solutions aux différentes contraintes engendrées par ces facteurs.

On distingue deux modes de transmission d'informations dans les RcSF : i) Envoi direct (en un seul saut) : c'est-à-dire que le nœud émetteur peut atteindre directement la station de base (SB). Cette transmission est possible si la SB est directement accessible par le nœud émetteur. ii) Envoi par nœuds intermédiaires (en multi-sauts) : dans le cas où la station de base n'est pas accessible par le nœud émetteur, on utilise la méthode d'envoi par nœuds intermédiaires. Ceci permet de créer des routes entre le nœud émetteur et sa destination.

Selon l'objectif de l'application, le routage dans les RcSF se classe en : i) Routage centré sur les données : La station de base émet une requête à certaines régions du réseau afin de recevoir uniquement les données attendues par l'initiateur de la requête, ii) Routage hiérarchique : consiste à diviser le réseau en plusieurs zones. Le but est de diminuer la consommation d'énergie par des fonctions d'agrégat exécutées localement par le responsable de chaque zone, iii) Routage basé sur la localisation : les messages échangés sur le réseau utilisent uniquement la position des capteurs et leur emplacement. Ces positions sont exploitées pour acheminer les données sur le réseau. Nous allons étudier la notion de routage dans les RcSF plus profondément dans le chapitre suivant de ce mémoire.

5.3. Notion de protocole

L'échange d'informations (données, messages, requêtes…etc.) sur le réseau nécessite un support physique (dans le cas des réseaux filaires) ou des équipements de transmission sans fil (antennes radio), et un ensemble de règles, de méthodes et de procédures assurant un standard de communication entre les entités, éventuellement de plusieurs types, du réseau. L'ensemble de ces moyens sont regroupés dans un même et unique langage appelé « protocole ». Or, un protocole assure la liaison des équipements par la définition de moyens physiques et procéduraux garantissant l'interconnexion entre eux, en procédant à la définition et le contrôle des flux d'informations échangées.

Un protocole est géré par des logiciels spécifiques installés sur des supports d'interconnexion comme les routeurs ou les commutateurs.

Dans les RcSF, plusieurs protocoles sont déployés aux différents niveaux de la couche protocolaire pour assurer la communication entre les équipements du réseau (les nœuds capteurs entre eux ou avec la station de base) et la sécurité des informations échangées. On cite les protocoles TRAMA, S-MAC pour la couche liaison de données, TEEN, APTEEN, LEACH pour la couche réseau ou Pike, LEAP comme protocoles cryptographiques assurant la sécurité des données. Dans la suite de notre travail, nous nous focaliserons sur les protocoles de routage et les protocoles cryptographiques.

Les protocoles de routage permettent d'acheminer correctement les données tandis que les protocoles cryptographiques assurent la sécurisation de ces informations circulant sur le réseau. Ces protocoles doivent être conçus d'une manière à réduire les coûts en communication et en consommation énergétique.

Le fonctionnement d'un protocole de routage ou de gestion de clés se sert de la connaissance de son voisinage. Les informations du voisinage sont obtenues par les mécanismes de découverte de nœuds voisins tels que les algorithmes (dans le cas de déploiement aléatoire), les informations préinstallées sur les nœuds capteurs (quand on connait au préalable l'emplacement des nœuds capteurs sur la zone de captage) ou les appareils de positionnement comme le GPS (dans le cas de routage géographique).

5.4. Notion de sécurité

La sécurité informatique est l'ensemble de techniques visant à protéger les données et s'assurer que celles-ci soient exploitées uniquement par les personnes ou périphériques autorisés. L'analyse de risques permet d'évaluer le degré des menaces et des failles de sécurité (vulnérabilités). Les résultats de cette analyse sont explorés par une politique de sécurité qui définit les mécanismes et les techniques à mettre en œuvre afin de garantir la disponibilité, l'intégrité et la confidentialité des données.

La sécurité d'un réseau de capteurs ne doit pas concerner uniquement la sécurisation de chacune des entités qui le composent mais c'est de définir une politique de sécurité globale (un protocole) combinant à la fois les solutions proposées sur chaque nœud et une solution englobant l'ensemble des protections appliquées à tous les nœuds car, et en l'absence d'une politique de sécurité globale, la compromission d'un nœud par un attaquant lui permettra d'accéder à toutes les informations communiquées sur le réseau.

Les approches traditionnelles de sécurisation des réseaux filaires ou d'autres types de réseaux sans fil (WiFi, WiMax, réseaux ad hoc…etc.) sont inappropriées pour les réseaux de capteurs sans fil pour des raisons liées aux contraintes sur les ressources des nœuds

capteurs. Les mécanismes employés dans ce type de solutions sont inadaptables aux RcSF pour les raisons suivantes : i) la cryptographie à clé publique : gourmande en ressources, notamment énergétiques et stockage mémoire ; ii) les clés préenregistrées sur les capteurs : elles nécessitent une mémoire importante pour stocker au minimum les *(n-1)* clés des *n* nœuds du réseau ; iii) le serveur de distribution de clés : en l'absence d'infrastructures fixes, la capacité des attaquants accroit et se concentre sur la station de base d'où elle devient le seul point faible à exploiter donc facilement accessible.

La sécurité des réseaux de capteurs sans fil est au cœur de plusieurs travaux de recherche proposant des solutions adéquates et efficaces d'autant que les RcSF ont souvent des objectifs stratégiques. Ces solutions ont pour souci la garantie de la confidentialité, l'intégrité et la disponibilité des données échangées par l'emploi de mécanismes d'authentification des utilisateurs. Toute politique de sécurité doit prendre en considération les contraintes suivantes : i) les ressources limitées des capteurs ; ii) le type de communication multi-sauts utilisé d'où l'impossibilité d'atteindre directement la station de base ; iii) absence ou peu d'infrastructures ce qui oblige les nœuds capteurs à assurer leur sécurité en dépit de leurs ressources limitées. Nous allons étudier plus profondément la notion de sécurité dans le chapitre 03 de ce mémoire.

5.5. Notion d'énergie

Plusieurs manières sont possibles pour un dysfonctionnement total d'un réseau de capteurs sans fil : attaque externe sur les nœuds, perte de liaisons avec la station de base, dégâts matériels, interférences…etc. Toutes ces causes sont externes. Un RcSF peut toutefois s'auto détruire par l'épuisement de ses capacités énergétiques. En effet, la durée de vie d'un réseau dépend essentiellement de la durée de vie de ses nœuds capteurs. Parallèlement, la durée de vie d'un nœud capteur dépend en grande partie de la durée de vie de sa source d'énergie. Beaucoup de travaux de recherche s'orientent actuellement sur la façon de trouver d'autres moyens d'alimentation en ressources énergétiques pour les capteurs tels que le moyen de convertir l'énergie lumineuse en énergie électrique, intégrer d'autres sources comme les dispositifs d'énergies renouvelables (modules solaires)…etc.

Un capteur est déployé pour fonctionner durant des années. Une meilleure gestion de son énergie garantit ce fonctionnement. D'après [37], un capteur utilise son énergie pour les opérations suivantes :

5.5.1. Energie de capture : c'est l'énergie consommée pour accomplir les taches suivantes : traitement et conversion analogique / numérique des signaux, activation de la sonde de capture…etc.

5.5.2. E*nergie de traitement :* c'est l'énergie dissipée dans l'exécution des opérations relatives aux traitements effectués sur les données reçues des autres unités.

5.5.3. Energie de communication : représente environ 80% de l'ensemble de l'énergie consommée par un capteur [62]. Elle est destinée aux opérations d'émission et de réception des informations. Elle est déterminée par deux facteurs essentiels : la quantité de données échangées et la puissance du signal qui dépend de la distance séparant les entités communicantes.

L'optimisation de la consommation d'énergie est l'objet de plusieurs efforts et contributions dans le domaine de la recherche scientifique. Les solutions retenues dans la littérature et qui s'imposent comme les plus économes en énergie sont [1]: agrégation de données spatio-temporelle, routage efficace en énergie et ordonnancement de l'interface radio.

5.6. Notion de clustering

La clusterisation est une technique qui consiste à diviser le réseau en groupe de nœuds également appelés clusters ou grappes (Figure 1.3). Chaque cluster est dirigé par un chef (Cluster Head) désigné soit par élection par les nœuds de son cluster, soit choisi et imposé par la station de base [37] [38]. Ce choix ou cette élection peuvent se faire sous plusieurs contraintes : la capacité énergétique du nœud capteur, son emplacement géographique, la topologie du réseau…etc.

Le rôle d'un Cluster Head est de récupérer les informations des nœuds formant le cluster, de les traiter et de les envoyer vers la prochaine destination. Des traitements spécifiques peuvent être appliqués sur les données reçues (selon leur nature), agrégation de données par exemple, avant leur transmission vers la station de base. Un CH (Cluster Head) peut être remplacé par un autre nœud du groupe pour l'une des raisons suivantes : i) CH défectueux ; ii) CH compromis ; iii) CH isolé ; iv) dans le souci de répartir équitablement la consommation d'énergie entre tous les nœuds capteurs du groupe.

La topologie en clusters est définie selon plusieurs paramètres : i) le déploiement des nœuds sur une zone géographique ; ii) le modèle de fonctionnement du réseau ; iii) le processus d'élection des CHs…etc. L'objectif de cette technique est de présenter le meilleur moyen de répartition et de conservation de la consommation d'énergie par un traitement et une transmission contrôlée des données récoltées. Le nœud d'un cluster peut atteindre en un seul saut ou en K sauts (k étant le nombre de nœud intermédiaires participant dans le processus de communication entre le nœud initial et sa destination) le

Cluster Head selon la distance qui le sépare de ce dernier et de sa puissance de transmission.

Afin de transmettre une information sur le réseau, deux types de communication sont mis en œuvre : intra-cluster et inter-cluster. Ceci permet de réduire le nombre de nœuds intervenant dans le processus de communication et ainsi réduire son coût.

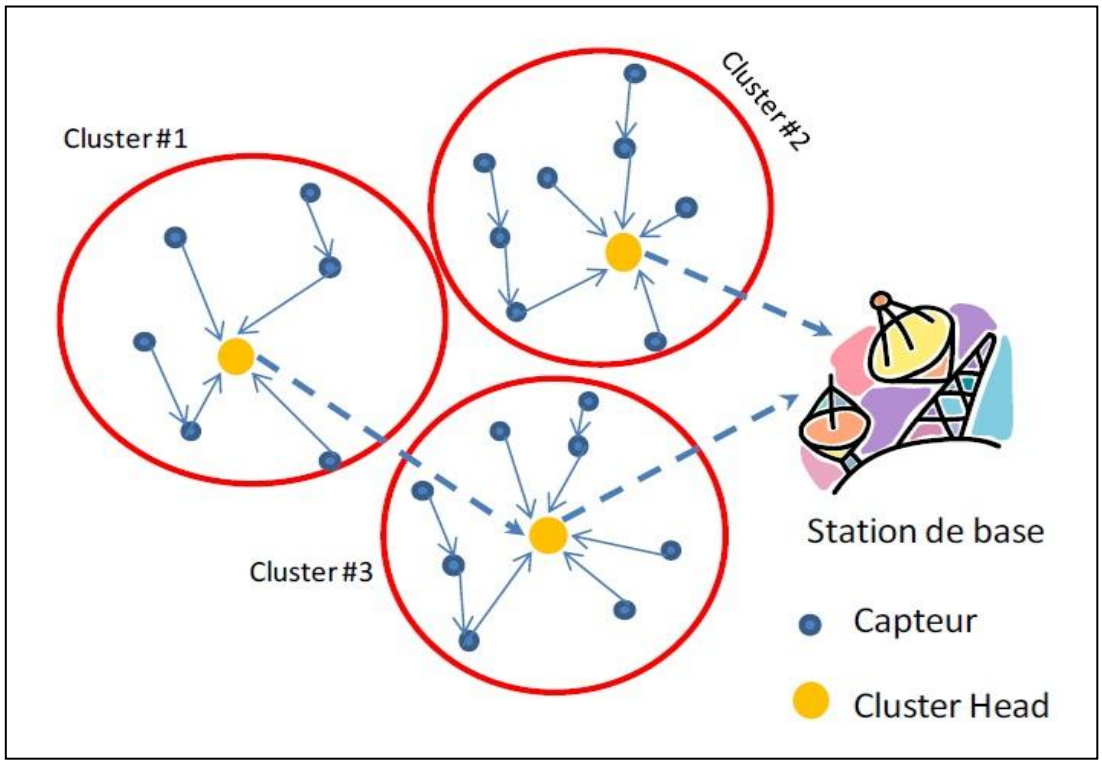

Figure 1.3 : Architecture en clusters [18]

Plusieurs protocoles sont conçus pour la création et la gestion des topologies en clusters. On cite: LEACH (Low-Energy Adaptive Clustering Hierarchy) [37], LEACH-C (Low-Energy Adaptive Clustering Hierarchy-C) [38], TEEN (Threshold sensitive Energy Efficient sensor Network protocol) [63], APTEEN (Adaptive Threshold sensitive Energy Efficient sensor Network protocol) [64]…etc.

Parmi les avantages du clustering, le renforcement du passage à l'échelle et l'augmentation de la durée de vie du réseau. En effet, on retient dans [36] que le clustering donne des meilleurs résultats que les approches traditionnelles basées uniquement sur le routage en multi-sauts. En revanche, cette approche est inefficace pour les applications temps-réel à cause des délais de traitement des données par les CHs qui sont parfois très lents ce qui provoque des retards de transmissions aux protocoles de routage temps-réel.

5.7. Notion d'agrégation

La densité du réseau et l'emplacement des capteurs, souvent proches l'un de l'autre, fait que les nœuds peuvent capter les mêmes données. Afin d'éviter la duplication des données captées et transmises, on utilise la technique d'agrégation. Cette technique exécutée au niveau des Cluster Head (CHs) consiste à appliquer un ensemble de fonctions d'agrégat (Min, Max, Moyenne, Somme…etc.) sur les données captées par l'ensemble des nœuds de son cluster. Ces fonctions permettent ainsi de remplacer les lectures individuelles par une lecture globale et collective, évitant des redondances sur les calculs et les transmissions des données effectuées individuellement sur chaque nœud capteur.

Sur l'exemple donné par la figure 1.4 et pour les quatorze messages envoyés par les sept nœuds du cluster vers le nœud Cluster Head (nœud agrégateur), un seul message résumant l'ensemble des informations reçues est transmis à la prochaine destination. Cette opération d'agrégat réduit la quantité d'informations circulant sur le réseau et par conséquent diminue le trafic et économise l'énergie dissipée dans la transmission. Or plusieurs contributions scientifiques ont démontré que la transmission d'un bit sur un réseau de capteurs sans fil est équivalente à l'exécution de mille instructions par le processeur qui demande une quantité d'énergie conséquente.

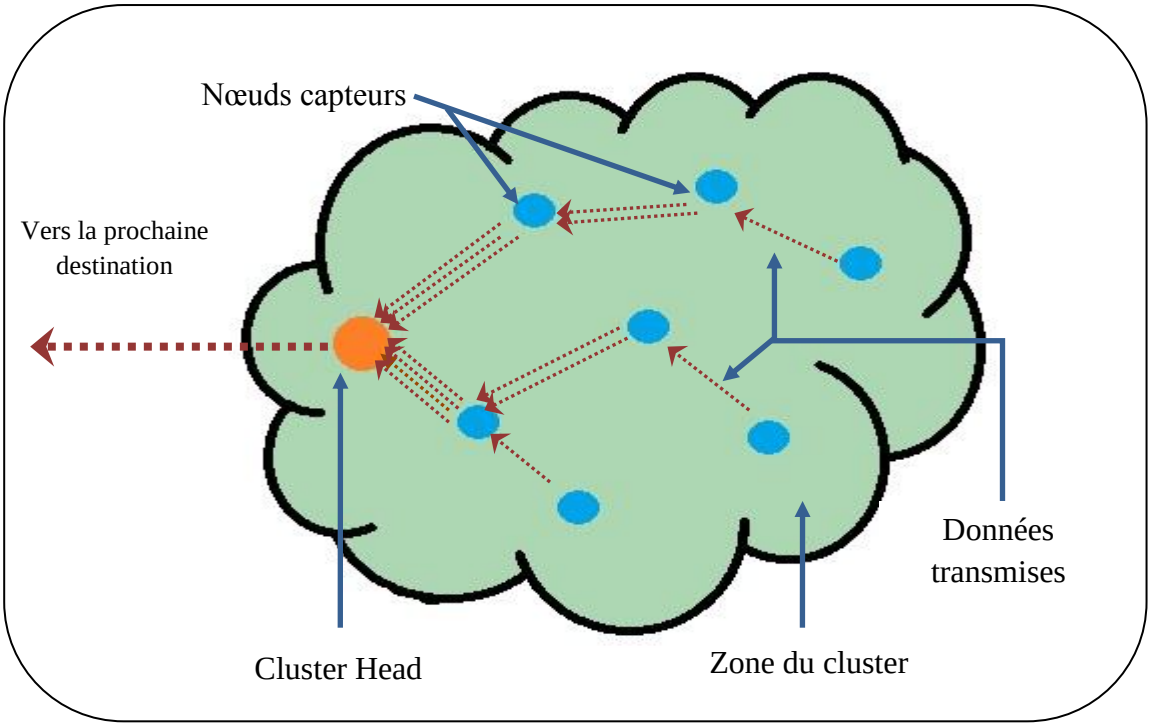

Figure 1.4 : Exemple d'agrégation de données

On distingue deux types de techniques d'agrégation : i) agrégation centralisée s'appuyant sur un protocole de clusterisation (le rôle du protocole est de créer, organiser,

gérer et maintenir l'architecture en clusters du réseau) et ii) une architecture distribuée où le réseau est vu de façon globale pour former une architecture arborescente.

Cependant, l'agrégation de données ne s'applique pas à tous les types des RcSF et elle concerne uniquement les données ayant des similitudes et des parties communes entre les messages reçues.

Le problème de la sécurisation des informations échangées, soit en inter-cluster ou en intra-cluster, se pose également pour la technique d'agrégation, et les formes d'attaque sont pratiquement les mêmes que dans tous les types de réseaux informatiques. Deux méthodes de sécurisation sont utilisés par les nœuds capteurs et particulièrement par le Cluster Head : i) la confidentialité de bout en bout : le CH applique directement les fonctions d'agrégat sur les données chiffrées reçues ; ii) la confidentialité saut par saut : le CH possède la clé de chiffrement, il déchiffre les données reçues, applique les fonctions d'agrégat puis chiffre de nouveau les nouvelles données agrégées en vue de leur expédition vers la prochaine destination.

L'inconvénient principal des techniques d'agrégation de données est la maitrise des données récoltées. En effet, avant toute application de fonctions d'agrégat, on doit connaitre au préalable la nature des données à agréger. Ceci est rendu complexe par le déploiement aléatoire des capteurs sur des zones vastes d'où la difficulté de créer des zones de calcul communs.

5.8. Environnements de simulation

Dans cette section, nous allons étudier quelques environnements de simulation utilisés dans le cadre des réseaux de capteurs sans fil. Avant tout déploiement effectif des capteurs sur un champ de captage, une série de tests et de simulations sont effectuées sur le code source pouvant être injecté sur les capteurs du réseau. Cette étape a pour objectif de tester l'efficacité réelle des protocoles de routage et de sécurité développés et de connaitre au préalable leurs failles et limitations. Une condition nécessaire au bon fonctionnement des tests serait d'exécuter les instructions du code sur une machine disposant d'un environnement d'exécution similaire à celui du capteur. Cette condition est garantie par un système d'exploitation spécialement dédié aux RcSF, c'est-à-dire un ensemble de fonctions, de procédures et de mécanismes produisant un code minimisé respectant les ressources limitées en mémoire et en capacités de calculs des capteurs. Cependant, les systèmes d'exploitation classiques sont inadaptés pour les RcSF pour de multiples raisons : i) la non prise en compte des contraintes énergétiques et ressources physiques des capteurs ; ii) un code énorme pour la mémoire d'un capteur ; iii) image mémoire

importante…etc. Plusieurs systèmes ont été proposés pour les réseaux de capteurs sans fil : TinyOs, MantisOS, Senses, Contiki…etc. TinyOs est le plus populaire et le plus utilisé, pourquoi nous l'avons choisi pour nos ateliers pratiques dans la simulation.

TinyOs a été développé par l'université de Berckley. Principalement dédié aux systèmes embarqués, il s'agit d'un système open source complètement écrit en langage C et ses composants ont été ré-implémentés en langage Nesc. Il se distingue par une programmation simple et puissante et un code portable sur les différentes plateformes existantes. Nesc est un langage de programmation orienté composants dont la syntaxe est proche du langage C. Il existe deux types de composants en Nesc : les modules et les configurations. Les modules définissent le code de l'application et implémentent des interfaces qui représentent l'unique point d'accès au composant. Les configurations permettent d'assembler les composants en reliant les interfaces du composant aux interfaces des autres composants. L'avantage de programmer en Nesc est la possibilité de construire des composants pouvant être des systèmes complets.

Pour les besoins des tests, plusieurs simulateurs ont été développés : Tossim, PowerTossim, NS2, NS3, Omnet++, JSim, Avrora…etc. Le simulateur Tossim se distingue par sa capacité à modéliser le comportement de l'implémentation par une compilation directe du code et la simulation de sa pile réseau. PowerTossim est une extension de Tossim avec la particularité d'estimer les capacités énergétiques des capteurs. NS2 demeure, actuellement, l'outil le plus utilisé. Le tableau 1.2 présente les simulateurs les plus utilisés, leurs avantages et leurs limitations [87].

A noter que le système TinyOs et les applications compilés sont installées directement et en un seul exécutable sur le nœud capteur.

Simulateur	**Langage / Plateforme**	**Avantages**	**Limitations**
NS-2	C++	Facilité d'ajout de nouveaux protocoles, un grand nombre de protocoles sont disponibles publiquement, disponibilité d'un outil de visualisation.	Supporte un nombre limité (environ une centaine) de capteurs, prend en charge uniquement les protocoles MAC 802.11 et TDMA.

Tossim / PowerTossim	Nesc	Exactitude d'exécution du code compilé, disponibilité d'un outil de visualisation.	Problèmes de synchronisation lors de la compilation
JSim	Java	Modélisation de la consommation d'énergie, architecture orientée composants.	Capacités de simulation trop faibles, supporte un seul protocole MAC (802.11).

Tableau 1.3 : Avantages et limites de certains simulateurs populaires

6. Caractéristiques des RcSF

Un réseau de capteurs sans fil (RcSF) possède plusieurs caractéristiques [7] dont :

- Ressources limitées des capteurs en calcul, en mémoire et en énergie.
- Durée de vie limitée.
- Mode de communication direct ou en multi-sauts.
- Densité importante des capteurs qui peuvent atteindre des dizaines de millions pour certaines applications.
- Possibilité de découper le réseau en clusters et d'utiliser les capteurs comme calculateurs ou des agrégateurs.
- La coopération entre les nœuds capteurs pour les taches complexes.
- Absence d'un identifiant global pour les capteurs.
- Deux modes de fonctionnement : « Un à plusieurs » où la station de base diffuse des informations aux différents capteurs ; et « Plusieurs à un » où les nœuds capteurs diffusent des informations à la station de base.

7. Contraintes de conception d'un RcSF

7.1. Passage à l'échelle

Le nombre de capteurs déployés sur une zone de captage peut atteindre plusieurs centaines de milliers voire plusieurs millions pour certaines applications. Le bon fonctionnement du réseau est conditionné par la définition d'un schéma de déploiement efficace respectant la propriété de haute densité. Le passage à l'échelle est défini comme la

possibilité de déployer un grand nombre de nœuds sur une petite surface. Il est donné par la valeur calculant les distances entre les nœuds.

Le passage à l'échelle est utilisé pour connaitre exactement la densité, le rayon d'émission et le nombre moyen de voisins d'un nœud donné. Ces informations sont d'une importance capitale pour bien modéliser le trafic avant toute implémentation réelle. A noter que le passage à l'échelle est plus critique dans les réseaux de capteurs que dans les autres réseaux sans fil.

7.2. Tolérance aux pannes

Le fonctionnement d'un ou de plusieurs capteurs peut être interrompu au cours du cycle de vie du réseau. Les causes de ces défaillances sont multiples : i) manque en ressources énergétiques, ii) dégâts matériels, iii) interférences environnementales, iv) compromission des nœuds …etc. Ces pannes ne doivent pas affecter le fonctionnement global du réseau. La tolérance aux pannes se définie alors comme la capacité du réseau à continuer à fonctionner normalement sans interruption même après le dysfonctionnement d'un ou de plusieurs de ses nœuds capteurs.

7.3. Environnement de déploiement

Dans la majorité des applications, les nœuds capteurs sont déployés dans des zones distantes, hostiles et sans aucune surveillance ni intervention humaine. Les capteurs doivent être conçus pour résister aux différentes conditions climatiques telles que la chaleur, l'humidité, le froid, la pression …etc.

7.4. Topologie du réseau

L'ajout de nouveaux capteurs sur la zone de captage ou la défection d'un ou de plusieurs nœuds capteurs du réseau peut causer une instabilité de la topologie du réseau.

7.5. Contraintes matérielles

On peut citer :

- Consommation stricte et mesurée de l'énergie.
- Un coût faible.
- Fonctionnement autonome des capteurs.
- S'adapter aux conditions de déploiement (environnementales, climatiques …etc.)
- Une portée radio limitées.

- Faible débit.
- Etc.

7.6. Economie d'énergie

Actuellement et hormis quelques expérimentations non encore abouties pour de nouvelles sources d'énergie (solaire par exemple), la batterie, dont la capacité est limitée et son remplacement ou rechargement est souvent impossible, est considérée comme l'unique alimentation en ressources énergétiques des capteurs. L'énergie d'un capteur est consommée par toutes ses unités (traitement, communication, capture, maintenance et réorganisation du réseau …etc.) ce qui donne à négocier parfaitement son utilisation par les différents protocoles utilisés dans la gestion du réseau.

8. Comparaison entre un RcSF et un réseau Ad Hoc

Un réseau de capteurs sans fil (RcSF) est un réseau Ad Hoc particulier. Les deux types de réseau ont plusieurs points communs et plusieurs divergences. Les points communs essentiels se résument en :

- Fonctionnement sans infrastructure fixe.
- Utilisation de support sans fil pour les communications.
- Le médium utilisé pour l'échange des informations est l'air.
- Les communications sont vulnérables aux problèmes d'interférences.
- L'emploi de protocoles de routage multi-sauts.
- Les protocoles d'accès au médium sont typiquement en mode half-duplex.

Les points de divergences sont également multiples [80] :

- Le nombre de nœuds formant un RcSF est beaucoup plus supérieur au nombre de nœuds formant un réseau Ad Hoc.
- La densité de déploiement est importante dans les réseaux de capteurs que dans les réseaux Ad Hoc.
- La mobilité est faible dans les RcSF contrairement aux réseaux Ad Hoc.
- Le risque de tomber en panne dans les RcSF est plus important que dans les réseaux Ad Hoc.
- Changement fréquent de la topologie du réseau de capteurs par rapport aux réseaux Ad Hoc.

- Le type de communication est généralement en broadcast (un-à-plusieurs ou plusieurs-à-un) dans les RcSF et en point-à-point dans les réseaux Ad Hoc.
- Les capacités en calcul, mémoire et énergie sont trop faibles dans les RcSF contrairement aux réseaux Ad Hoc.
- Les entités d'un réseau de capteurs sans fil sont souvent inaccessibles par les humains. Les entités d'un réseau Ad Hoc sont directement accessibles.

Le tableau 1.3 illustre les convergences et les divergences entre un RcSF et un réseau Ad Hoc :

Réseau / Propriétés	RcSF	Ad Hoc
Infrastructure	**Sans infrastructure fixe**	
Support de communication	**Sans fil**	
Médium	**Air**	
Problèmes d'interférences	**Vulnérables aux problèmes d'interférences**	
Routage	**Envoi direct, Multi-sauts**	
Accès au médium	**Half-duplex**	
Nombre de nœuds	**Plusieurs centaines de milliers**	**Une dizaine**
Densité de déploiement	**Importante**	**Moyenne**
Mobilité	**Faible**	**Forte**
Pannes	**Fréquent**	**Rare**
Topologie du réseau	**Changements fréquents**	**Fixe**
Type de communication	**Broadcast**	**Point-à-point**
Capacités en ressources	**Trop faible**	**Aucune limitation**
Accessibilité	**Inaccessible**	**Facilement accessible**

Tableau 1.4 : Comparaison entre un RcSF et un réseau Ad Hoc

9. Conclusion

Le cout de plus en plus faible, la taille réduite, la large gamme, la facilité de déploiement, l'auto-organisation des capteurs, le support de communication sans fil, la tolérance aux pannes sont les caractéristiques des nœuds capteurs qui offrent des possibilités énormes de développement de réseaux pour un champ d'application vaste et varié. Ce vaste

champ d'application confirme l'idée de penser que les réseaux de capteurs sans fil feront dans les années à venir une partie intégrante de notre vie quotidienne et changeront certainement notre manière de vivre.

Cependant, la conception d'un réseau de capteurs robuste et fiable doit satisfaire certaines contraintes liées à la durée de vie et à la sécurité. En effet, une consommation d'énergie acceptable, le respect des limites physiques en calcul et en mémoire et l'adaptation à l'environnement sont des paramètres à prendre en considération avant toute idée de conception d'un RcSF. De plus, afin d'assurer une fiabilité maximale du système, différents mécanismes de sécurité sont développés dans le but d'éliminer toute menace capable d'atteindre à la sécurité des informations échangées, souvent confidentielles et stratégiques.

Dans ce chapitre, nous avons défini ce qu'est un RcSF. Nous avons évoqué brièvement les notions d'énergie, de routage et de sécurité, des notions que nous allons développer profondément dans les chapitres à venir de ce mémoire.

Dans le chapitre suivant, nous allons commencer à étudier la notion de routage dans les réseaux de capteurs sans fil.

CHAPITRE 02
ROUTAGE DANS LES RCSF : ETAT DE L'ART

Ce chapitre est consacré à la notion de routage dans les réseaux de capteurs sans fil. Nous allons commencer par définir les différentes architectures de communication entre les entités du réseau, les contraintes de conception, la contrainte d'énergie et les critères de performance des protocoles chargés de gérer ces communications. Selon l'objectif et la nature d'une application d'un RcSF, nous allons classifier et présenter les principaux protocoles de routage adéquats pour un routage peu couteux en énergie.

1. Introduction

Le routage dans les réseaux de capteurs sans fil est la procédure d'acheminement d'informations d'un nœud source collecteur vers une destination à travers un réseau de connexion. Chaque nœud est susceptible d'être à la disposition des autres nœuds capteurs pour participer à la transmission et retransmission des informations émises sur le réseau par un ou un ensemble de nœuds n'ayant pas la possibilité d'atteindre directement la destination. Cette dernière peut être un nœud agrégateur, un chef de zone dans le cas de réseaux hiérarchiques ou la station de base pour les réseaux à plat.

En raison des caractéristiques particulières des RcSF qui les distinguent des autres réseaux traditionnels sans fil, notamment en capacités de calculs, de stockage et d'énergie des nœuds capteurs, le routage est une tache ardue et compliquée nécessitant un travail de collaboration de tous les nœuds appartenant au réseau. On peut classifier les approches du routage selon l'objectif du réseau (récolte périodique, événementiel, requête spécifique sur une région…etc.), selon l'architecture du réseau (plat, hiérarchique, arborescente…etc.), selon la façon d'atteindre une destination (proactif, réactif et hybride), selon les taches assignées aux protocoles (Qualité de service, contrôle de flux, négociation…etc.), etc.

2. Types de routage

Le routage dans les réseaux de capteurs sans fil se classe généralement en [66]:

2.1. Routage à plat : appelé également routage centré données (data centric) [50] où tous les nœuds ont les mêmes tâches à accomplir. C'est la première approche utilisée dans l'acheminement des données dans les RcSF. Elle se base sur la collaboration de tous les nœuds du réseau. Les propriétés des données sont spécifiées par un système de

dénomination par attribut (attribut, valeur) [4] [89] en raison de la difficulté d'affecter un identificateur global à chaque nœud vu leur nombre important. Parmi leurs avantages, la simplicité d'où la possibilité d'établir des communications sans surcoût où chaque nœud n'aura besoin que des informations de ses voisins directs. L'inconvénient est l'épuisement des ressources en énergie des nœuds proches de la station de base car tout le trafic vers cette dernière passe obligatoirement par eux.

2.2. Routage hiérarchique : cette approche est basée sur la formation de clusters (zones communes). Le principe est de router les données récoltées par chaque nœud du cluster à son chef de zone (Cluster Head), qui et après des traitements sur leurs parties communes, les transmettra à la prochaine destination (Si le CH ne pourra pas atteindre directement la station de base, les informations seront routées vers le prochain chef de zone). L'avantage est la réduction des coûts en communication et en énergie en minimisant le nombre de messages circulant sur le réseau, étant donnée que les CHs appliquent des fonctions d'agrégat sur les données du cluster ce qui permet de les combiner. L'inconvénient concerne la taille du réseau. En outre, quand la taille du réseau augmente, le processus d'élection du Cluster Head devient critique et gourmand en ressources.

2.3. Routage basé sur la localisation : l'identification des emplacements géographiques des nœuds capteurs sur la zone de captage est d'une importance capitale pour les mécanismes de routage de données dans les RcSF. Ces informations de localisation permettent le calcul des positions des capteurs et les distances qui les séparent afin de construire les chemins les plus courts entre un nœud source et sa destination. Cette approche de routage est plus économe en énergie car elle dispense les nœuds capteurs d'employer les méthodes aléatoires ou probabilistes pour rechercher les routes. De plus, la localisation des nœuds (et par conséquent de leurs régions) permet de diffuser des requêtes uniquement à ces régions et éviter leur diffusion en mode broadcast (diffusion globale à tous les nœuds) et ainsi réduire le nombre de transissions d'une manière significative [94] [97]. L'inconvénient est la nécessité d'équiper les nœuds capteurs avec un système de localisation par satellite comme le GPS qui consomment énormément d'énergie.

2.4. Routage proactif : le calcul de routes se fait à priori ce qui facilite l'acheminement des données. Les informations des chemins à suivre par chaque donnée source vers une destination sur le réseau sont stockées dans une table de routage. Les tables de routage doivent être mises à jour régulièrement afin de corriger certains chemins coupés en

raison du changement de topologie dus aux défaillances ou à la mobilité de certains nœuds capteurs. Cette mise à jour est assurée par la diffusion périodique des paquets de contrôle sur le réseau, ce qui n'est pas évident pour des réseaux de grande taille comme les réseaux de capteurs sans fil. L'établissement de routes se fait indépendamment des besoins réels de l'application et un bon nombre de ces routes est sauvegardé pour ne jamais être utilisées. Une autre limite concerne la taille des tables de routage, notamment pour des réseaux de grande taille, qui pourrait dépasser les capacités de stockage des nœuds capteurs.

2.5. *Routage réactif :* également appelé routage à la demande, le routage réactif permet de créer les routes selon les besoins de l'application. Lorsqu'une requête est diffusée sur le réseau, la procédure de découverte de routes est lancée [4][93] par les nœuds concernés par cette requête, et les réponses sont acheminées sur les routes créées. Cette procédure est lancée également pour des applications event-driven (applications orientées événements) pour chaque événement intéressant détecté. L'avantage d'établir des routes à la demande est la conservation d'énergie par rapport au routage proactif. La recherche de routes peut causer des lenteurs pour l'acheminement des données ce qui n'est pas approprié aux applications interactives et temps-réel.

2.6. *Routage hybride (à la fois proactif et réactif) :* c'est une combinaison des deux concepts de routage proactif et réactif. Des tables de routage sont stockées sur les nœuds capteurs de façon à établir des routes sur leur voisinage proche (généralement en deux sauts maximums). Au-delà de leur voisinage, le routage devient réactif et des procédures de recherche de routes sont lancées. Cette approche combine les avantages des deux autres approches proactive et réactive et réduit considérablement la taille des tables de routage ainsi que les délais d'établissement de routes.

Le problème pour les approches de routage dans les réseaux de capteurs sans fil se manifeste par la façon d'adapter ces solutions aux fortes contraintes matérielles (taille des tables de routage, puissance des algorithmes de calcul de chemins) et énergétiques des nœuds capteurs, ainsi qu'aux caractéristiques particulières aux RcSF (taille du réseau, passage à l'échelle, tolérance aux pannes, topologie…etc.).

Plusieurs protocoles sont proposés dans la littérature pour palier aux problèmes du routage et minimiser les coûts de communication, de transmission de données et de la consommation d'énergie. Un protocole de routage calcule le coût de communication entre une source et sa destination en utilisant des algorithmes de découverte de routes conçus d'une

manière à respecter les contraintes et les caractéristiques propres aux RcSF. Parmi ces protocoles, nous citons : LEACH, TEEN, APTEEN, SPIN, Directed Diffusion, GEAR…etc.

3. Les architectures de communication dans les RcSF

Une communication est définie par l'échange d'informations entre les entités du réseau. Elle peut être établie suite à un événement déclenché par un nœud capteur ou à une requête diffusée sur le réseau par la station de base, l'utilisateur final de l'application, un nœud chef de zone…etc.

Les stratégies de communication dépendent de plusieurs facteurs : domaine d'application du réseau (stratégique, confidentiel, publique …etc.), l'environnement de déploiement (hostile, inaccessible), les objectifs du réseau (temps réel, régulier, événementiel…etc.).

Les réseaux de capteurs sans fil sont souvent utilisés dans des environnements hostiles sans aucune infrastructure de base, ne possédant aucune information sur l'emplacement des nœuds capteurs et par conséquent sur la topologie du réseau (locale ou globale). Dans des conditions similaires, les nœuds capteurs doivent s'auto-organiser et construire une infrastructure de communication autonome qui leur permet d'interagir entre eux et d'établir un cheminement de n'importe quel nœud capteur vers n'importe quelle destination sur le réseau.

Pendant le processus d'acheminement de données et d'informations, un nœud capteur joue plusieurs rôles à la fois : collecteur, intermédiaire ou collecteur et intermédiaire en même temps. Les nœuds intermédiaires participent à la mise en place d'architectures de communications multi-sauts. Dans certains cas et ce qui est rare pour les RcSF, les nœuds capteurs peuvent atteindre en un seul saut (directement) la station de base si ces deux entités se trouvent dans la même portée radio de communication. Ce type de communication en un seul saut ne nécessite aucune collaboration ni architecture de communication pour le routage de données et d'informations.

La figure 2.1 montre un exemple d'un réseau constitué de plusieurs nœuds capteurs. Il s'agit d'une communication multi-sauts entre le nœud collecteur C_4 et la station de base et d'une communication en un seul saut entre C_1 et la station de base. A partir de cette organisation, on distingue quatre types de communications dans les réseaux de capteurs sans fil :

a. Communication directe entre un nœud collecteur et un autre nœud collecteur (cas des nœuds C_4 et C_5) : ce type de communication est généralement utilisé dans des opérations

locales telles que la procédure de création de routes ou pendant le processus de clusterisation [68].

b. Communication entre un nœud collecteur et un nœud intermédiaire (cas des nœuds C_4 et C_3) : Puisque le nœud collecteur n'est pas en mesure d'atteindre directement la station de base, ses données collectées sont transmises à un nœud intermédiaire se trouvant dans sa portée radio.

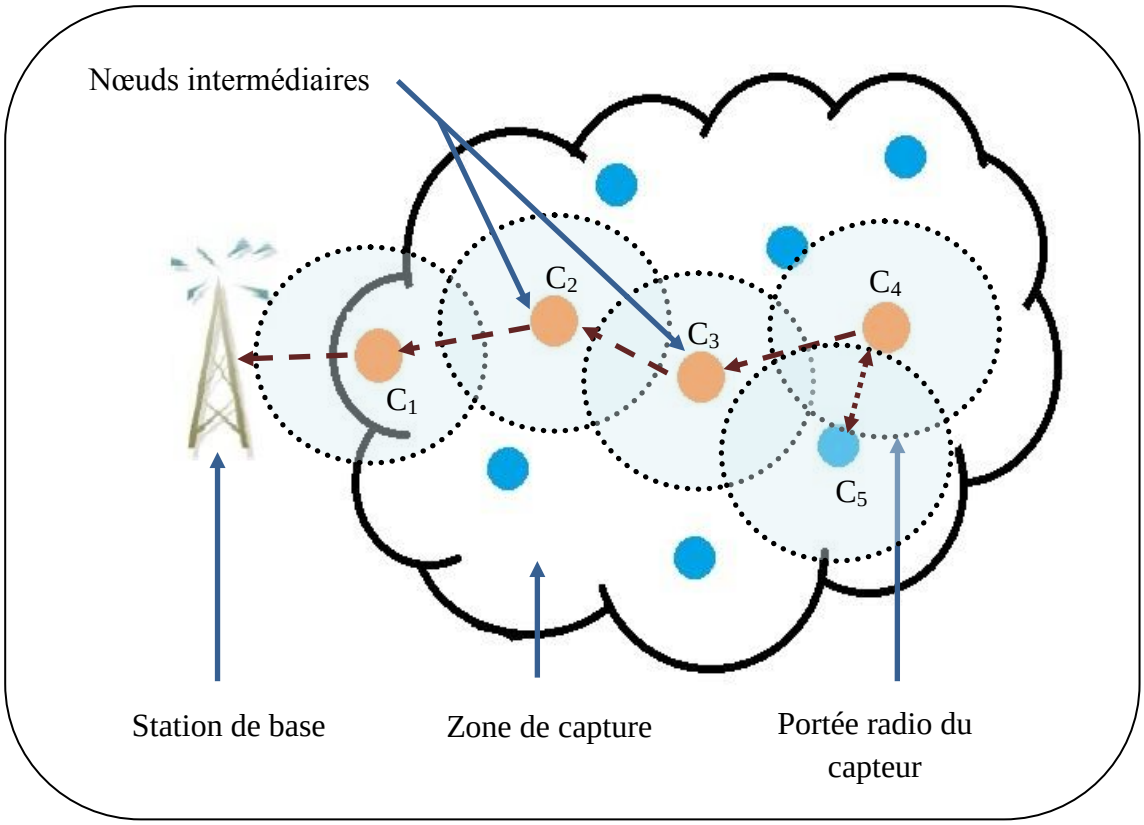

Figure 2.1 : Communication multi-sauts dans un réseau de capteurs sans fil

c. Communication entre un nœud intermédiaire et un autre nœud intermédiaire (cas de C_3 et C_2) : souvent en unicast [71], ce type de communication a pour rôle d'installer un réseau de connexion entre le nœud collecteur et la station de base.

d. Communication entre un nœud intermédiaire et la station de base (cas du nœud C_1) : c'est la dernière étape avant la réception finale des données par la station de base. Cette communication peut concerner également une transmission directe entre un nœud collecteur et la station de base.

4. Contraintes de conception d'un protocole de routage

Nous allons présenter dans ce qui suit quelques contraintes liées aux caractéristiques des protocoles de communication et de routage de données dans les réseaux de capteurs sans fil

pour la conception de protocole robustes et fiables. En effet, trouver un chemin entre un nœud source et sa destination se heurte à un certain nombre de difficultés dues à l'obligation d'optimiser la consommation d'énergie, respecter les capacités faibles des nœuds capteurs et assurer une qualité de services offerts. Ces difficultés sont causées par le support de communication sans fil utilisé, la taille du réseau, le déploiement, souvent, aléatoire des nœuds, la dynamicité du réseau, du passage à l'échelle…etc.

4.1. Capacités réduites des capteurs

Les capacités réduites des capteurs en calcul, en mémoire et en énergie empêchent le développement de mécanismes complexes, volumineux et complets qui s'exécutent sur des processeurs puissants et qui nécessitent des tailles mémoire importantes. De même, l'une des recommandations majeures à prendre en compte est de répartir équitablement l'énergie consommée lors de la découverte et la maintenance des routes. De plus, l'agrégation de données, la clusterisation, contrôle de la puissance du signal sont, entre autres, des techniques de conservation d'énergie appliquées par un protocole de routage lors de la transmission de données entre les différentes entités du réseau, utilisant une architecture de communication adéquate.

4.2. La taille du réseau

Le nombre de nœuds d'un réseau de capteurs sans fil est souvent important et sa topologie change fréquemment avec la défaillance ou l'ajout de certains nœuds capteurs. Ce changement de topologie provoque une maintenance et une mise à jour régulière des schémas de gestion de routes ; deux opérations gourmandes en énergie. Ainsi, la réorganisation du réseau et la mise à jour des tables de routage, ou parfois le changement de l'approche de routage utilisée selon les nouvelles donnes, implique la participation d'un grand nombre de nœuds capteurs, ce qui est difficile à gérer par les protocoles de routage.

4.3. Déploiement des nœuds

Les nœuds d'un RcSF peuvent être déployés d'une manière sélective ou aléatoire. Dans un environnement de déploiement sélectif où les nœuds capteurs sont placés à la main, les protocoles de routage peuvent utiliser directement les informations pré-chargées sur les capteurs avant leur déploiement sur la zone de captage. On parle alors de routage prédéterminé. Ce genre de déploiement ne pose pas trop de problèmes aux protocoles de routage car les informations sur l'emplacement des capteurs sont statiques et les mises à jour des tables de routage s'effectuent d'une manière déterministe. Cependant, les informations sur les positions des capteurs dispersés aléatoirement sur une zone à surveiller

sont difficiles à connaitre ou à prévoir et l'emplacement des capteurs est quasi inconnu. Dans ce cas, on procède alors à la définition de nouveaux mécanismes chargés de la découverte des routes (algorithmes de Djikstra, algorithme de Nilson, algorithmes génétiques, technique du spanning tree…etc.).

4.4. La dynamicité du réseau

Dans la plupart des applications des réseaux de capteurs sans fil, l'emplacement des nœuds sur une zone de captage est souvent statique et les positions des capteurs sont fixes et invariables, ce qui maintient régulièrement à jour l'état des liens et les informations de routage stockées sur les nœuds capteurs. Néamoins, des facteurs externes comme les vents, les orages, et autres conditions climatiques ainsi que le facteur humain (d'une manière accidentelle ou ciblée) et animal peuvent changer le positionnement géographique des nœuds capteurs et engendrer l'apparition ou la disparition de liens entre les nœuds ce qui va altérer au processus de routage. Le problème de la gestion de la mobilité au sein d'un RcSF est l'objet de plusieurs projets de recherche proposant des solutions nouvelles [83] aux problèmes identifiés [101] mais soulevant de nouveaux défis notamment la gestion des communications entre nœuds mobiles [51] et l'adaptation des schémas anciens aux nouvelles informations volatiles et non permanentes.

4.5. Tolérance aux pannes

Un nœud participant à l'acheminement de données sur un réseau de capteurs est susceptible de tomber en panne suite à l'épuisement de ses réserves d'énergie, à une défection matérielle, à une compromission…etc. ce qui ne doit pas altérer le fonctionnement du protocole de routage. Un mécanisme de gestion et de prise en charge de pannes doit être prévu et mis en place.

4.6. Qualité de service

Un protocole de routage doit veiller au respect des délais d'acheminement et à préserver l'intégrité des données émises car un message, une donnée ou une information arrivant après un certain délai ne serait peut être plus utilisable surtout dans des applications temps réel.

5. La contrainte d'énergie pour le routage

La consommation d'énergie est la métrique la plus importante dans l'évaluation de la durée de vie des capteurs et des performances d'un réseau de capteurs sans fil. La source d'énergie des capteurs, conçus pour fonctionner durant des mois et même des années et alimentés par des batteries de capacité limitée, doit être utilisée de façon optimale pour toute

tache effectuée, notamment la communication et l'échange d'informations, entre les entités du réseau. Plusieurs métriques et techniques sont utilisées par les protocoles de routage afin d'optimiser les sources de consommation d'énergie au niveau de la couche réseau.

5.1. Sources de consommation d'énergie

Les sources de consommation d'énergie au niveau de la couche réseau sont diverses [10] :

5.1.1. La longueur des chemins : Les paquets de données suivent des chemins en un certain nombre de sauts. Le coût d'un saut en termes d'énergie est mesuré par la distance qui sépare les deux nœuds concernés par la communication, et le coût global du routage est la somme des énergies consommées à tous les sauts. Les chemins les plus longs sont les plus gourmands en énergie.

5.1.2. La qualité des liens : la retransmission des données suite à l'interruption du cheminement entre la source et la destination est une opération qui engendre un coût énergétique supplémentaire.

5.1.3. Le mode de communication : le routage point-à-point est inefficace pour les RcSF du fait qu'il consomme beaucoup d'énergie au niveau des nœuds capteurs. Par ailleurs, les modes de communication adéquats sont en effet le mode « one-to-many » et le mode « many-to-one ».

5.1.4. Le routage de paquets inutiles : certains paquets de données deviennent inutiles si on ne respecte pas les échéances temporelles des transmissions, fixées soit par l'émetteur ou par le récepteur. Router des données dont l'échéance est expirée engendre une consommation d'énergie inutile.

5.1.5. Le choix d'un chemin : La non-prise en compte de l'énergie des nœuds lors du processus d'établissement de chemins peut provoquer l'épuisement des capacités énergétiques de certains nœuds.

5.2. Techniques économes en énergie

Parmi les techniques et les métriques les plus économes en énergie utilisées par les protocoles de routage, on cite :

5.2.1. Ajustement des puissances de transmission : L'énergie de transmission consommée par le nœud émetteur lors de l'envoi d'un message dépend directement de la puissance de transmission. Certains protocoles de routage prennent en compte les

distances entre les nœuds afin d'optimiser le choix sur la puissance de transmission minimale à utiliser et de réduire la consommation d'énergie.

5.2.2. Distribution des charges : La solution de distribution des charges entre les nœuds participants au routage de données est très efficace aux problèmes de congestion de charges où les paquets émis empruntent toujours les mêmes nœuds, ce qui provoque l'épuisement rapide de leur énergie. En effet, les protocoles de routage calculent le coût énergétique de tous les chemins entre la source et sa destination et choisissent le chemin le moins couteux en termes d'énergie [23]. Le calcul du coût énergétique se fait par le nœud source sur la base des puissances de transmission.

5.2.3. La formation des grappes : La technique de partitionnement du réseau, notamment ceux ayant une densité importante de nœuds, pour la formation de grappes permet une meilleure gestion du routage sur plusieurs niveaux. Chaque sous ensemble (grappe) est constitué d'un certain nombre de nœuds possédant des propriétés ou des tâches communes, regroupés autour d'un nœud chef considéré comme le nœud collecteur de la grappe. Par ailleurs, le nœud collecteur coordonne et ordonnance les activités des nœuds membres du groupe en autorisant certains nœuds à se mettre en veille, ou bien en ajustant les puissances de transmissions ou alors en organisant les instances de transmission afin d'éviter les collisions et par conséquent les retransmissions. Le risque d'épuisement de l'énergie des nœuds chef est pris en charge par la station de base qui procède à leur remplacement par d'autres nœuds à chaque intervalle de temps ou à chaque diminution critique de leur capacité énergétique.

5.2.4. La réduction de données : La réduction de données consiste à empêcher deux ou plusieurs nœuds qui couvrent une même zone de captage de transmettre des données décrivant le même événement, en éliminant les données redondantes par des nœuds dits agrégateurs, ce qui réduit le nombre de transmissions et par conséquent la quantité d'énergie dissipée.

5.2.5. Négociation des échanges de données : Les nœuds du réseau entament une négociation sur les données à transmettre en diffusant un message contenant une métadonnée décrivant les données à transmettre. Les nœuds intéressés par ces données manifestent leur intérêt et les reçoivent en entier. La sélection des nœuds destinataires par l'intérêt réduit la bande passante et la consommation d'énergie du réseau.

6. Critères de performance des protocoles de routage

Nous allons présenter, dans cette section, certaines métriques permettant de mesurer l'efficacité et la robustesse des protocoles de routage face aux menaces et contraintes des RcSF. Les performances d'un protocole de routage se mesurent par sa capacité à assurer les tâches de construction et de maintenance de routes et de la transmission d'informations sur le réseau à moindre coûts et à des temps de traitement et de transfert raisonnables. Il est primordial de considérer la consommation d'énergie et l'intégrité des données transmises comme des métriques essentielles pour toute évaluation d'un protocole de routage destinée aux RcSF.

6.1. Consommation d'énergie

Le nœud capteur doit utiliser son énergie d'une façon optimale pour ses activités de détection, de traitement et de communication. Un protocole de routage doit gérer les périodes d'activité et d'inactivité des capteurs en incluant des modes « en marche » et « en veille » et avoir la notion de temps pour se mettre en veille et se réveiller. Le mode « en veille » permet au nœud capteur d'éteindre son interface radio et d'empêcher une perte d'énergie due aux écoutes actives et en permanence inutiles de son environnement.

6.2. Temps de traitement

C'est le temps pris par un nœud capteur pour effectuer des opérations de calcul sur les données récoltées ou reçues. Ce temps doit être raisonnable pour ne pas causer des retards de transmission d'informations pour des applications critiques et temps réel.

6.3. La mobilité des nœuds capteurs

La position des capteurs sur la zone de captage n'est pas toujours fixe. Un nœud capteur peut devenir mobile et changer sa position selon les besoins de l'utilisateur. Des traitements spécifiques pour la maintenance des liens et la mise à jour des informations de routage sont à prévoir lors de la conception d'un protocole de routage.

6.4. Modes de transmission

Choisir un mode de communication adéquat à la structure et à la topologie du réseau de déploiement.

6.5. Sécurité des échanges

Consiste l'envoi périodique de paquets de contrôle afin d'éviter des collisions et des pertes de données. Un paquet de contrôle peut contenir le nombre de bits émis, l'adresse ou

l'identificateur de destination et des informations sur le routage. Ces paquets sont nécessaires afin d'assurer la disponibilité et l'intégrité des données transmises.

7. Classification des protocoles de routage

Les protocoles de routage sont conçus différemment pour répondre aux objectifs d'un réseau de capteurs sans fil. Plusieurs applications des capteurs exigent un routage efficace, sécurisé et économe en énergie assurant une bonne qualité de service et des temps de traitement et de transmission convenables. Le choix de conception d'un protocole de routage pour les RcSF est assez vaste et nous pouvons les classifier par différentes façons. En effet, certaines applications de réseaux de capteurs sont mises en place pour récolter des données périodiques sur leur environnement tandis que d'autres sont chargées de répondre à des évènements importants produits à l'intérieur ou à proximité de la zone à surveiller, et d'autres encore sont conçues pour capter des informations précises sur des zones ciblées (répondre aux requêtes diffusées par les utilisateurs de l'application).

Les techniques de transmission de données sur un réseau de capteurs sans fil différent selon plusieurs critères : i) la structure du réseau : protocoles de routage plats, hiérarchiques ou basés sur la localisation ; ii) les services rendus : protocoles de routage basés sur la qualité de service, sur la négociation, sur le multi-chemin, sur les interrogations ; iii) mode de fonctionnement du réseau : découverte de routes, maintenance et mise à jour des informations de routage, diffusion de requêtes de données ; iv) type d'application : périodique, temps réel, interactive, événementielle. Le tableau 2.1 présente une classification détaillée des protocoles de routage dans les RcSF :

Critère	Classification	Définition
Structure du réseau	Hiérarchique	Le réseau est organisé en clusters. Le routage s'effectue sur plusieurs niveaux (intra-cluster et inter-cluster).
	Plat	Tous les nœuds ont le même rôle et collaborent entre eux pour accomplir le routage.
	Géographique	Les informations de localisation des nœuds sont utilisées pour le routage de données.
Les fonctions du protocole	Qualité de service	Le réseau doit satisfaire la qualité des données avec une consommation raisonnable d'énergie.
	Négociation	Eliminer les transmissions redondantes et établir des communications selon les ressources du réseau.

	Multi-chemins	Utiliser des chemins multiples afin d'augmenter les performances du réseau en maintenant des chemins alternatifs.
Mode de transmission	Proactif	Les chemins sont établis à priori.
	Réactif	Les chemins sont établis à la demande selon les besoins.
	Hybride	Combine les deux techniques proactive et réactive.
Type d'application	Event-driven	Réagir à des changements soudains.
	Time-driven	Prélèvement périodique des données.
Origine de la requête	Initiée par la source	Les données sont envoyées à un certain intervalle ou quand le nœud capture certains événements.
	Initiée par la destination	Les nœuds répondent aux requêtes envoyées par la destination.
Mode de fonctionnement	Découverte	Permet aux nœuds de connaitre des services et de s'auto-organiser pour accomplir leurs taches.
	Interrogation	Propager une requête sur le réseau pour un intérêt particulier.
	Acheminement	C'est le service de communication qui permet d'acheminer les données vers la destination.
Paradigmes de communication	Node centric	Les communications se basent sur l'identification des nœuds participants.
	Data centric	Les communications se basent sur les données à transmettre.
	Position centric	Les communications se basent sur la position des nœuds.

Tableau 2.1 : Classification des protocoles de routage [13]

8. Les principaux protocoles de routage

Le tableau 2.2 [6] résume les principales contributions de la communauté scientifique pour le routage de données dans les RcSF.

8.1. LEACH: LEACH (*Low*-Energy Adaptive Clustering. Hierarchy) est l'un des protocoles les plus populaires pour les réseaux de capteurs sans fil [6][9]. Son principe est de former des zones communes de calcul et de traitements en se basant sur la puissance du signal et le niveau d'énergie des nœuds capteurs. Chaque zone est dirigée par un chef de zone, jouant le rôle d'agrégateur et de routeur, en effectuant des traitements sur les données reçues de son cluster et leur expédition vers la prochaine destination. Ce rôle de chef de zone est échangé entre les nœuds d'un cluster afin de répartir équitablement la consommation d'énergie entre eux.

8.1.1. ***Fonctionnement :*** Le protocole LEACH se déroule en rounds. Chaque round se compose de deux phases : Phase de construction et phase de communication. La phase de construction consiste à définir les clusters et élire les Clusters Head (CHs), la phase de communication est responsable de la transmission des données captées.

Principaux protocoles de routage dans les RcSF			
Structure du réseau	**Protocole de routage plat**	**Protocole de routage hiérarchique**	**Protocole de routage basé sur la localisation**
	SPIN, Cougar, Directed Diffusion, Rumour Routing …etc.	LEACH, TEEN, APTEEN, PEGASIS …etc.	Gear, GAF, Speed, SAR …etc.
Type de protocole	**Protocole de routage basé sur la négociation**	**Protocole de routage basé sur la qualité de service**	**Protocole de routage basé sur l'interrogation**
	SPIN	SPEED	Directed Diffusion

Tableau 2.2 : Les principaux protocoles de routage dans les RcSF

Les deux phases doivent être exécutées en même temps par les nœuds du réseau afin de garantir un meilleur fonctionnement du protocole. Après une période de temps passée, une requête est diffusée sur le réseau par la station de base afin d'évaluer les

performances des nœuds chefs de zones en terme d'énergie et procéder, le cas échéant, à son remplacement.

Le protocole LEACH est conçu pour des applications time-driven. Son utilisation dans des applications event-driven est inapproprié car il emploie le schedule TDMA et par conséquent un mode de transmission proactif.

8.1.2. ***Caractéristiques :*** Economie d'énergie, minimisation du nombre de messages circulant sur le réseau.

8.2. TEEN et APTEEN : TEEN [63] (Threshold sensitive Energy Efficient sensor Network *protocol) et son extension APTEEN [64] (AdaPtive* Threshold sensitive Energy Efficient sensor Network protocol) sont des protocoles de routage hiérarchiques orientés données qui conviennent aux applications sensibles à des changements soudains des attributs de la zone à surveiller. En effet, ces deux protocoles ont une réactivité importante à des changements improvisés des valeurs captées par les nœuds capteurs. De plus, APTEEN possède des caractéristiques supplémentaires par rapport au protocole TEEN lui permettant de répondre plus rapidement aux besoins des utilisateurs des applications temps réel en modifiant les paramètres, notamment la périodicité, du protocole TEEN.

8.2.1. ***Fonctionnement :*** Le protocole TEEN favorise le mode réactif pour le transfert de données et son complément APTEEN introduit le mode proactif en modifiant au besoin les messages émis par le chef de zone.

La majorité des comportements de TEEN et APTEEN sont semblables aux comportements du protocole LEACH, sauf que dans les deux premiers protocoles cités, on ne transmet pas de schedule TDMA mais un message contenant des informations sur la tache demandée au capteur, la valeur critique après laquelle les membres doivent envoyer des rapports de données et la valeur du seuil représentant un changement minimal obligeant les nœuds à envoyer le nouveau rapport de données. De plus, les nœuds qui exécutent les commandes de APTEEN doivent respecter un délai maximum toléré entre deux émissions de rapport. Ce délai garantit un comportement proactif et permet aux applications time-driven d'envoyer des informations périodiques sur leurs environnements. Pour cela, le protocole APTEEN peut être employé pour des applications event-driven et time-driven.

8.2.2. ***Caractéristiques :*** répondre aux applications event-driven et time-driven en introduisant un mode de transmission réactif nécessaire aux applications critiques et un mode de transmission proactif favorable aux applications périodiques.

8.3. PEGASIS et Hierarchical PEGASIS : PEGASIS [60] (Power-Efficient GAthering in Sensor Information Systems) et Hierarchical PEGASIS [61] sont des versions améliorées du protocole LEACH. Pour ces deux protocoles, le réseau est vu comme un arbre, les nœuds forment des chaines plutôt que des clusters. Un nœud transmet et reçoit uniquement les données de son voisin. Chaque nœud est considéré comme un nœud agrégateur, ses données reçues sont traitées et envoyées au prochain nœud de la chaine. Ainsi, toutes les données capturées sont fusionnées et transmises par un seul nœud désigné pour communiquer avec la station de base [60]. Les nœuds qui transmettent à la station de base sont choisis, pour un intervalle de temps bien défini, selon la politique de remplacement Round Robin dans le but de répartir équitablement l'énergie consommée durant un round de transmission. Hierarchical PEGASIS se présente comme une variante de PEGASIS qui tente de réduire les délais de transmission par l'envoi simultané de messages et la prise en considération de la métrique (énergie – retard).

8.3.1. ***Fonctionnement :*** Le principe est d'organiser le réseau sous forme d'arbre hiérarchique où les nœuds collecteurs sont considérés comme des feuilles et la station de base comme la racine. Les données captées transitent d'une feuille à la racine par des nœuds intermédiaires formant une chaine. A la réception d'un paquet de données, le nœud intermédiaire procède à son traitement avant son expédition vers son voisin direct de la chaine. Le dernier nœud de la chaine (appelé leader) transmet les données fusionnées à la station de base.

8.3.2. ***Caractéristiques :*** Economie d'énergie, répartition équitable des tâches et des ressources.

8.4. SPIN : SPIN [39] (Sensor Protocols for Information via Negociation) est un protocole de routage qui se base sur le principe de méta-données lors du processus de diffusion de requêtes sur le réseau. Le récepteur de la requête a la possibilité de choisir d'accepter la donnée ou non. C'est le principe de la négociation. SPIN est considéré comme le premier protocole data centric destiné aux RcSF. Tous les nœuds du réseau sont traités comme des nœuds de destination [2] dont la tâche principale est de recueillir un point de vue globale sur leur environnement. Egalement, SPIN s'attaque à la redondance de données transmises sur un réseau de capteurs sans fil.

8.4.1. ***Fonctionnement :*** SPIN se fonde sur un modèle de négociation inter nœuds pour le traitement des messages reçus contenant la nature de la tâche à réaliser. Le principe est d'émettre un message ADV, contenant la description de la donnée à transmettre, à tous les nœuds du réseau. Chaque nœud récepteur du message ADV décidera de la réception ou non de la donnée à transmettre, selon sa description. Ainsi, si le nœud est intéressé par l'information, il envoi une réponse sous forme de requête au nœud émetteur du message ADV. Par la suite et à la réception de toutes les réponses, le nœud émetteur commence à envoyer les données aux nœuds intéressés.

8.4.2. ***Caractéristiques :*** Economie d'énergie, éviter les problèmes de l'inondation et de l'implosion, palier aux problèmes d'adaptation aux ressources.

8.5. Directed Diffusion (DD) : Directed Diffusion ou diffusion dirigée [46] est un protocole de propagation de données. C'est l'un des protocoles les plus intéressants dans le routage data centric des RcSF. Son utilité réside dans sa capacité à réduire considérablement le nombre d'opérations de la couche réseau par la création de plusieurs chemins pour le routage d'informations et ainsi permettre une économie d'énergie conséquente aux capteurs. Cette réduction d'opération passe par une mise en place de quatre éléments : i) la nomination des données (décrire les intérêts) ; ii) propagation des intérêts ; iii) propagation des données ; iv) renforcement des chemins de transport de données. Le protocole DD à travers ces quatre éléments définit au préalable la nature de la donnée à capter, décrit cette donnée sous forme d'intérêt et diffuse cette requête sur le réseau. Les nœuds concernés commencent à capter et propager les données sur des chemins différents. A la réception des premières données par la station de base, elle procède au choix puis au renforcement des meilleurs chemins.

8.5.1. ***Fonctionnement :*** La station de base définit une requête d'intérêt, décrit par un schéma attribut-valeur, sur une information particulière à vouloir récupérer sur l'environnement de déploiement des capteurs. Ensuite, la requête sera diffusée par une inondation globale du réseau et les nœuds concernés commencent à récolter les informations décrites par cette requête. Les données récoltées seront transmises suivant des gradients relatifs à l'intérêt pour déterminer le chemin à suivre vers la station de base. La station de base, et après avoir reçue les premières bonnes informations, renforce les chemins vers les nœuds émetteurs.

8.5.2. ***Caractéristiques :*** Réduction du nombre d'opérations de la couche réseau, économie d'énergie.

8.6. Rumour Routing : Le protocole Rumour Routing [11] utilise un procédé probabiliste, similaire au gossiping, afin de trouver un compromis entre l'inondation des intérêts et la propagation de données. Ce procédé probabiliste repose sur la méthode Monté-Carlo qui suppose que la probabilité que deux lignes se croisant au sein d'une région rectangulaire est de 0.69, et si on utilise cinq lignes passant par un point, la probabilité qu'une autre ligne se croise avec l'une des cinq lignes est de 0.997. Par conséquent, si on considère la source et le puits deux points distincts, on établit un certain nombre de mi-chemin depuis la source et d'autres mi-chemin depuis le puits et on aura une forte probabilité que deux mi-chemin se joignent créant ainsi un chemin complet entre la source et la destination. Ce procédé permet d'éviter l'inondation de données.

8.7. GEAR : Le protocole GEAR [99] (Geographic and Energy Aware Routing) est un protocole de routage basé sur les informations géographiques des nœuds capteurs. Les informations géographiques sont utiles pour restreindre le champ de diffusion des requêtes d'intérêts et ne prendre en considération que les régions ciblées.

8.8. MECN : Le protocole MECN [84] (Minimum Energy Communication Network) est un protocole de routage géographique conçu pour des nœuds capteurs équipés d'un GPS à basse puissance. L'idée est de trouver un sous réseau composé d'un nombre minimum de nœuds, proches entre eux, afin d'avoir une puissance de transmission faible et moins couteuse en énergie entre deux nœuds particuliers du réseau.

Le tableau 2.3 présente la comparaison de quelques protocoles de routage dans les réseaux de capteurs sans fil.

Protocole	Architecture	Type du routage	Prends en compte l'énergie	Communication	Agrégation de données	Qualité de service (QoS)
LEACH	Hiérarchique	A base de clusters	Oui	Multi-sauts	Oui	Non
TEEN / APTEEN	Hiérarchique	A base de clusters	Oui	Multi-sauts	Oui	Non
PEGASIS / Hierarchical PEGASIS	Hiérarchique	A base de chaines	Oui	Multi-sauts	Non	Non

SPIN	Data centric	Négociation	Oui	Multi-sauts	Oui	Non
Directed diffusion (DD)	Data centric	Multi-chemins	Oui	Multi-sauts	Oui	Non
Rumour Routing	Data centric	Inondation	Oui	Multi-sauts	Non	Non
GEAR	Géographique	Basé sur les informations géographiques des capteurs	Oui	Multi-sauts	Non	Non
MECN	Géographique	Basé sur les informations géographiques du GPS	Oui	Multi-sauts	Non	Non

Tableau 2.3 : Comparaison des protocoles de routage dans les RcSF

9. Conclusion

Les progrès dans les réseaux de capteurs sans fil ont mené à de nouvelles solutions pour le routage de données dans ce type de réseaux où la contrainte d'énergie est une considération essentielle. Plusieurs facteurs ont influencé la conception de ces solutions et des techniques robustes, mesurées par les métriques citées précédemment, ont été développées pour garantir des communications efficaces en termes d'énergie.

Nous avons décrit profondément dans les sections précédentes de ce chapitre les notions de routage et d'énergie dans les RcSF. Nous avons présenté quelques protocoles de communication, les plus populaires et les plus appropriés, dépassant les contraintes propres aux RcSF et satisfaisant, à un certain degré, les métriques de performance, notamment celle liée à l'économie d'énergie des nœuds capteurs.

Dans le chapitre suivant, nous allons étudier plus profondément la notion de sécurité dans les RcSF, aussi importante que primordial pour la fiabilité du réseau, en présentant des solutions efficaces et peu couteuses en énergie.

CHAPITRE 03
SÉCURITÉ DU ROUTAGE : MENACES ET SOLUTIONS

Ce chapitre propose un état de l'art sur la sécurité des réseaux de capteurs sans fil. D'autant que la confidentialité des informations échangées sur un RcSF est primordiale, la sécurité est donc une dimension importante pour ces réseaux. Les solutions proposées pour les réseaux ad hoc classiques ne peuvent pas s'appliquer aux RcSF car les capteurs sont limités par leur puissance d'énergie et de calcul ce qui rend la conception d'une politique de confiance un défi de taille. Nous allons présenter une taxonomie des attaques sur les réseaux de capteurs sans fil, en précisant leur nature et leurs objectifs, les vulnérabilités et les failles de sécurité exploitées. Nous allons présenter par la suite quelques défis, politiques et solutions de sécurité pour ce type de réseaux.

1. Introduction

Le succès des RcSF est lié, en grande partie, à leur efficacité dans le monitoring de leur environnement et la capacité de ses nœuds à intervenir, d'une façon autonome, dans des fonctions sensibles. Cette intervention consiste à protéger les transmissions des données captées, à moindre cout, dans leur intégralité sans aucun accès ou modification par une autre entité étrangère. L'environnement de déploiement des RcSF regroupe des domaines d'application très critiques où la sécurité du système représente un enjeu majeur et l'existence de failles est intolérable.

En effet, les RcSF connaissent actuellement une grande extension et une large utilisation dans différents types d'applications, souvent sensibles, exigeant une sécurité renforcée. Toutefois, les nœuds capteurs disposent de capacités mémoires, énergétiques et de traitement très limitées. Ces applications font que l'introduction des mesures de sécurité classiques est restreinte. En d'autres termes, la protection des communications dans les réseaux de capteurs sans fil nécessite un niveau de sécurité élevé entre les différents liens de communication. Cependant, la sécurisation des liens n'est pas suffisante pour garantir la fiabilité du réseau. En effet, la sécurisation des entités du réseau permet d'empêcher toute attaque visant à i) compromettre les nœuds, ii) récupérer leur identité, iii) récupérer les informations captées et échangées avec les autres nœuds capteurs, iv) altérer le bon

fonctionnement du réseau par l'introduction de fausses informations par le nœud compromis…etc.

La sécurisation des RcSF est la source de plusieurs travaux de recherche proposant des solutions économes en énergie et qui respectent d'autres contraintes propres aux capteurs telles que la puissance de calcul et les capacités de stockage [32]. Par ailleurs, l'absence d'infrastructure fixe, le déploiement dans des endroits ouverts et hostiles proches des territoires ennemis rendent les réseaux de capteurs sujet à différents types d'attaques qui visent à exploiter leurs vulnérabilités et se permettre des privilèges d'accès pour modifier, d'insérer ou de supprimer une partie ou la totalité des informations qui circulent sur le réseau.

Deux types de vulnérabilités sont à distinguer pour les RcSF : logiques et physiques. Les vulnérabilités physiques se résument essentiellement aux ressources limitées des nœuds capteurs notamment en termes d'énergie, de stockage et de calcul. Les vulnérabilités logiques dérivent des vulnérabilités physiques, principal handicap au développement de mécanismes complexes, complets et fiables. Par conséquent, les contraintes matérielles et énergétiques s'imposent dans le modèle de conception d'un mécanisme de sécurité fiable.

Les vulnérabilités liées à la sécurité du routage dans les réseaux de capteurs sans fil se résument en :

- La technologie sans fil sous jacente. Quiconque possédant le récepteur adéquat peut potentiellement écouter ou perturber les messages échangés ;
- Les nœuds eux-mêmes sont des points de vulnérabilité du réseau car une attaque peut compromettre un composant laissé sans surveillance ;
- L'absence d'infrastructure fixe pénalise l'ensemble du réseau dans la mesure où il faut faire abstraction de toute entité centrale de gestion pour l'accès aux ressources ;
- Les mécanismes de routage sont d'autant plus critiques dans les RcSF que chaque nœud participe à l'acheminement des paquets à travers le réseau. De plus, les messages de routage transitent sur les ondes radio.

Plusieurs solutions, basées sur des approches algorithmiques et sur des approches cryptographiques, sont proposées pour les RcSF. Nous nous intéresserons dans la suite de notre travail uniquement aux solutions cryptographiques basées sur la gestion de clés. La taille réduite des clés cryptographiques et le besoin d'une consommation raisonnable en énergie des crypto-systèmes ont amené les chercheurs à proposer plusieurs contributions dans ce contexte. Habituellement, la gestion de clés exige le chargement d'informations secrètes

(clés) sur les nœuds capteurs avant leur déploiement. Cette information secrète peut être une information auxiliaire, sous forme d'une clé, nécessaire à la génération de toutes les clés secrètes réelles à utiliser dans la sécurisation des communications. Le chargement et la génération de ces clés est conditionné par la capacité de stockage réduite des nœuds capteurs, d'où la difficulté de concevoir des protocoles cryptographiques complexes [43].

Afin de mieux comprendre l'aspect sécurité dans les RcSF (contexte, difficultés, objectifs et solutions), nous allons, tout d'abord, répondre aux deux questions posées ci-dessous puis nous présenterons cinq sections sur les principes d'attaques et d'attaquants, leurs objectifs, les défis, les politiques, les menaces et les solutions de sécurité.

- Pourquoi les RcSF sont vulnérables ?
    - Des vulnérabilités liées aux réseaux ad hoc ;
    - Contraintes matérielles et énergétiques ;
    - Débit faible et une portée radio limitée ;
    - Déploiement sur des zones à risque proches ou au cœur des zones ennemies ;
    - Absence d'infrastructures fixes ;
    - Absence de sécurité physique ;
    - De nouvelles technologies émergentes ;
    - Les attaquants sont souvent mieux équipés ;
    - La sécurité ne peut pas être sure à 100 %.
- Pourquoi les solutions de sécurité ne peuvent jamais être sures à 100 % ?
    - Difficulté de remplacer la politique de sécurité en cas d'attaques massives sur le système ;
    - Différents types d'attaques simultanées sur un même réseau ;
    - Plus les mécanismes de sécurité sont confrontés à des contraintes, moins ils sont efficaces.

2. Principes d'attaques et d'attaquants

Une attaque [108] peut être définie comme une tentative d'accès illégale à une ressource du système. Les attaques visent essentiellement les liens de communication et les entités du réseau afin de s'autoriser à récupérer et manipuler les données échangées. Un attaquant est une personne qui s'intéresse au fonctionnement du réseau dans le but de s'adjuger les moyens

et le pouvoir de déjouer la sécurité du système. Cela est possible par la maitrise des techniques utilisées pour le sécuriser, et ainsi causer son dysfonctionnement partiel ou total par l'usurpation d'identités et la compromission de ses nœuds. L'existence d'un nœud compromis est très problématique car cela nécessite de revoir complètement la politique de sécurité appliquée. Les attaquants de réseaux de capteurs sans fil se classent en plusieurs catégories en fonction de leurs objectifs et de leurs capacités de nuisance. On distingue :

- *Attaquant passif* : son objectif est de récupérer les informations qui circulent sur le réseau sans toutefois toucher à son fonctionnement. Ces mêmes informations peuvent être utilisées pour générer des attaques actives sur le réseau.
- *Attaquant actif* : il vise à détruire le réseau par la compromission ou la destruction physique de ses nœuds.
- *Attaquant externe* : il est considéré comme entité étrangère au réseau. il essaye d'infiltrer le réseau de l'extérieur en exploitant certaines failles de sécurité.
- *Attaquant interne* : il revendique son appartenance au réseau afin d'accéder aux ressources du ou fournies par le système.

3. Objectifs des attaques

Un attaquant peut opérer à deux niveaux : s'attaquer aux informations échangées entre les nœuds et s'attaquer aux nœuds eux-mêmes. Les objectifs et les motivations d'un attaquant sont multiples ; on cite les principaux [92] :

- Obtenir un accès au système.
- Espionnage : récupérer les données qui circulent sur le réseau.
- Perturbation : injection de données erronées, génération de fausses alertes, dénis de services…etc.
- Détournement : la compromission des nœuds et leur détournement de leurs fonctions initiales.

4. Défis et challenges de sécurité dans les RcSF

Les contraintes propres aux capteurs et les types d'attaques diversifiées auxquels ils doivent faire face font de la sécurité du routage de données dans les RcSF un défi majeur. Plusieurs attaques ont été recensées contre les protocoles de routage [28] [100] et plusieurs solutions ont été développées et proposées. Les solutions de sécurité dédiées aux réseaux ad hoc ne peuvent pas s'appliquer aux RcSF pour les mêmes motifs cités précédemment dans ce mémoire. Pour cela, les solutions conçues pour les RcSF ont un seul défi principal à relever :

concevoir des mécanismes de sécurité privilégiant la consommation d'énergie tout en maximisant les performances de sécurité. Les principaux challenges que doit relever un protocole de sécurité dans les réseaux de capteurs sans fil sont :

- *Authentification des nœuds du réseau* : on doit vérifier qu'un nœud correspond bien au nœud qu'on a déployé et l'information reçue est bien celle émise par ce nœud.
- *L'intégrité des données transmises* : les données ne doivent subir aucune modification au cours de leur transmission du nœud source à la destination.
- *La confidentialité des données échangées* : les données ne doivent jamais être accessibles aux nœuds non autorisés.
- *La disponibilité des données* : les données doivent être disponibles au moment opportun à leur utilisation.
- *La fraicheur des données :* elle permet de vérifier si la donnée reçue est récente ou non. Cette vérification garantit que ces données ne reflètent pas un événement passé qui n'a plus cours.
- *Collaboration et auto-organisation :* après le déploiement, les nœuds capteurs doivent être capables de fonctionner en collaboration et s'auto-organiser pour se sécuriser eux-mêmes en absence d'une sécurité physique et d'une infrastructure fixe de sécurisation synonyme d'absence d'une tierce partie de confiance. Le développement des relations de confiance entre les nœuds capteurs, l'échange et l'établissement des clés de cryptage et la gestion des clés de session entre nœuds communicants sont les fonctionnalités que doivent assurer les nœuds capteurs en s'auto-organisant et en se collaborant.

Afin d'atteindre les objectifs de sécurité cités précédemment, les solutions de sécurité doivent relever les défis suivants :

- Minimiser la consommation d'énergie : l'objectif est de concevoir des mécanismes de sécurité performants et économes en énergie. L'énergie dissipée pour assurer les fonctions de sécurité est utilisée pour le calcul, la transmission des certificats de sécurité, les chiffrages, déchiffrages, signatures, vérifications des signatures et le stockage des paramètres de sécurité.
- S'adapter à l'environnement de déploiement : dans la plupart des applications des réseaux de capteurs sans fil, les nœuds sont déployés dans des zones hostiles ce qui rend leur compromission assez facile. Les mécanismes de sécurité doivent détecter et réagir

rapidement à la capture et à la compromission des nœuds afin de changer les paramètres de sécurité appliqués.

- Contourner l'absence de la sécurité physique des capteurs : la topologie des RcSF les rends exposés à différents types d'attaques pouvant survenir de toute part et qui cible n'importe quel entité du réseau. l'absence d'une sécurité physique, contrairement aux réseaux filaires où la sécurité est renforcée par des firewalls, doit être contournée par des techniques de détection d'intrusions.
- Développer des schémas propres aux communications sans-fil : les solutions classiques de sécurité pour les réseaux filaires et autres types de réseaux sans fil sont inapplicables aux RcSF. Les mécanismes de sécurité doivent coupler le médium de communication sans-fil utilisé aux caractéristiques spécifiques aux réseaux de capteurs sans fil.

5. Politiques de sécurité dans les RcSF

Une politique de sécurité dans les réseaux de capteurs sans fil est un ensemble de mécanismes, d'algorithmes, de procédures et de schémas cohérents conçus pour maintenir un certain niveau de sécurité. Une politique de sécurité doit assurer au minimum : i) un contrôle d'accès : détecter et interdire l'ajout de nouveaux éléments corrompus au réseau ; ii) l'intégrité et iii) la confidentialité des données transmises. Les protocoles de sécurité reposent sur les politiques suivantes :

- Sécurité des communications : choisir des outils adéquats (comme la cryptographie symétrique) pour l'établissement de liens sécurisés entre les nœuds communicants. Ces outils doivent respecter les limites des nœuds capteurs.
- Sécurité du routage et des données agrégées : l'accès aux données pour les agréger, les filtrer, les compresser ou décompresser doit se faire uniquement par les nœuds autorisés du réseau.
- Sécurité physique des nœuds : protéger les nœuds contre toute tentative de compromission, détecter et mettre en quarantaine les nœuds compromis.

6. Stratégies de sécurité pour le routage

La sécurité du routage dans les RcSF concerne les différents niveaux de communication selon la topologie du réseau et le type du routage appliqué pour acheminer les informations. Par ailleurs, les politiques de sécurité mettent en avant la sécurité des liens et des communications directes dites bout-à-bout pour ensuite obtenir la sécurité globale du réseau (voir figure 3.1), soit :

- Sécurité des liens : il s'agit de sécuriser les communications directes entre les entités communicantes.
- Sécurité intra-cluster : protéger les communications établies entre les nœuds membres d'un cluster (dans le cas où le routage de données à l'intérieur du cluster est en multi-sauts) ou bien entre les nœuds membres et le Cluster-Head.
- Sécurité inter-cluster : cette protection concerne la sécurité des informations échangées entre les Cluster-Head.
- Sécurité nœud du réseau – station de base : le but est de protéger les requêtes diffusées par la station de base aux nœuds du réseau et les informations transmises par les nœuds à la station de base.

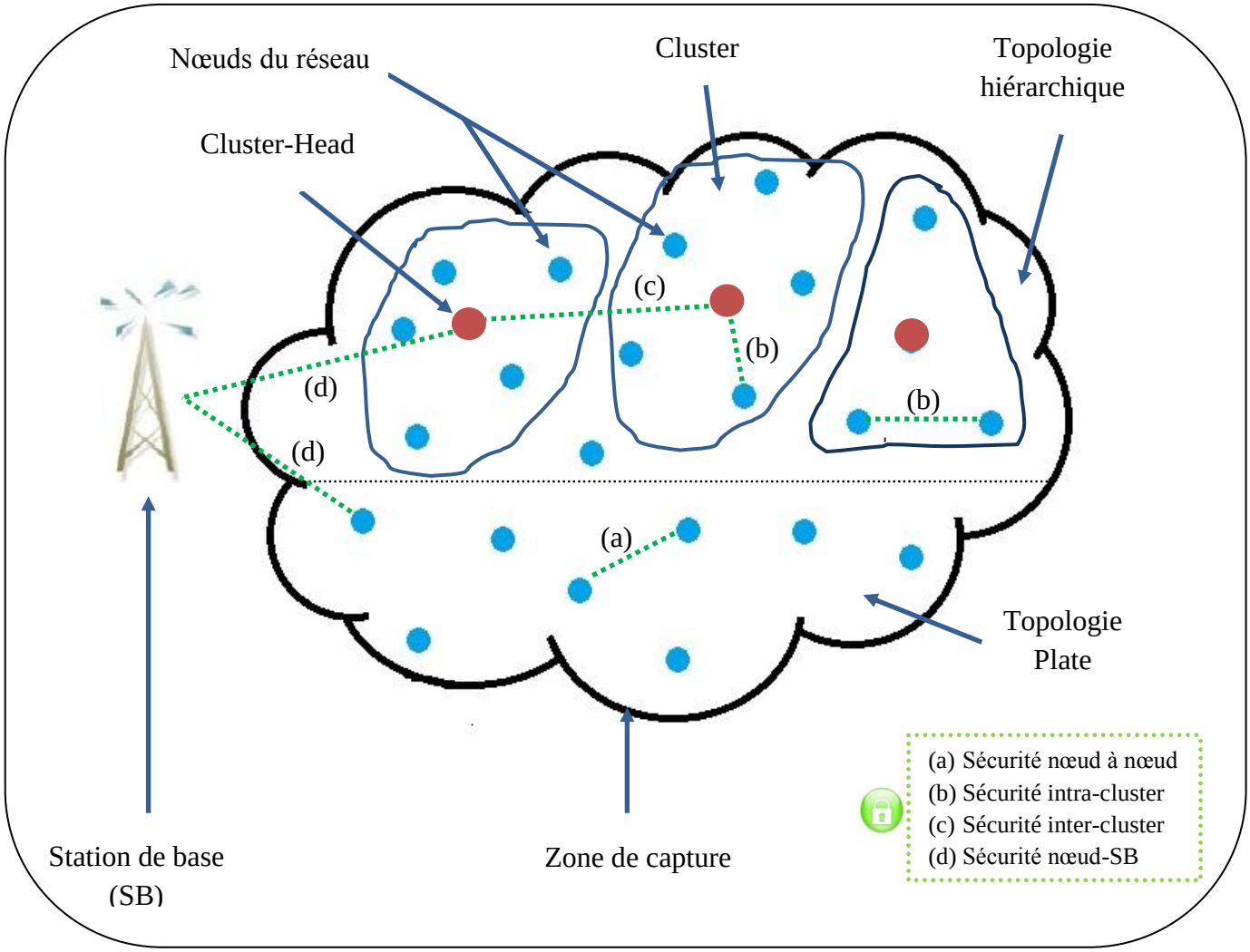

Figure 3.1 : Stratégies de sécurité dans un RcSF

7. Menaces et solutions

7.1. Taxonomie des attaques

Les menaces sur les réseaux de capteurs sans fil sont nombreuses et leurs objectifs sont multiples. On distingue des attaques sur la construction et la maintenance des routes et d'autres sur les informations contenues et échangées par les nœuds du réseau. Les attaques peuvent être passives ou actives, physiques ou logiques. Dans cette section, nous allons présenter une série des menaces (liste non exhaustive) les plus redoutables et les plus connues sur le routage de données dans les RcSF [103].

7.1.1. Attaques passives

Dans ce type d'attaques, généralement, l'attaquant passe inaperçu. En effet, son objectif est d'écouter le réseau sans chambouler ou altérer son fonctionnement. Pendant ce temps, aucun paquet de données n'est émis sur le réseau par l'attaquant ce qui rend sa détection très difficile. L'objectif de ces attaques est d'analyser les paquets de données circulant sur le réseau et d'extraire des informations précieuses, d'une part, et d'analyser les chemins empruntés par ces paquets de données afin de se procurer des informations stratégiques sur le fonctionnement global du réseau comme la position des stations de base, l'identité des nœuds chargés d'agrégation…etc. d'autres parts. Les informations obtenues par un attaquant passif peuvent être utilisées pour créer des attaques actives [65]. Exemple : l'attaque Eaversdropping.

7.1.2. Attaques actives

Contrairement aux attaques passives, les attaques actives visent à modifier l'état du réseau. En effet, un attaquant émet périodiquement des paquets de données sur le réseau pour différents objectifs qu'on va énumérer en présentant quelques techniques d'attaques sur le routage, les plus répandues pour les RcSF. Les menaces actives appartiennent principalement à quatre catégories (illustrées par la figure 3.2) :

- Interruption : Vise la disponibilité des données.
- Interception : Vise la confidentialité des données.
- Modification : Vise l'intégrité des données.
- Fabrication : Vise l'authenticité des données.

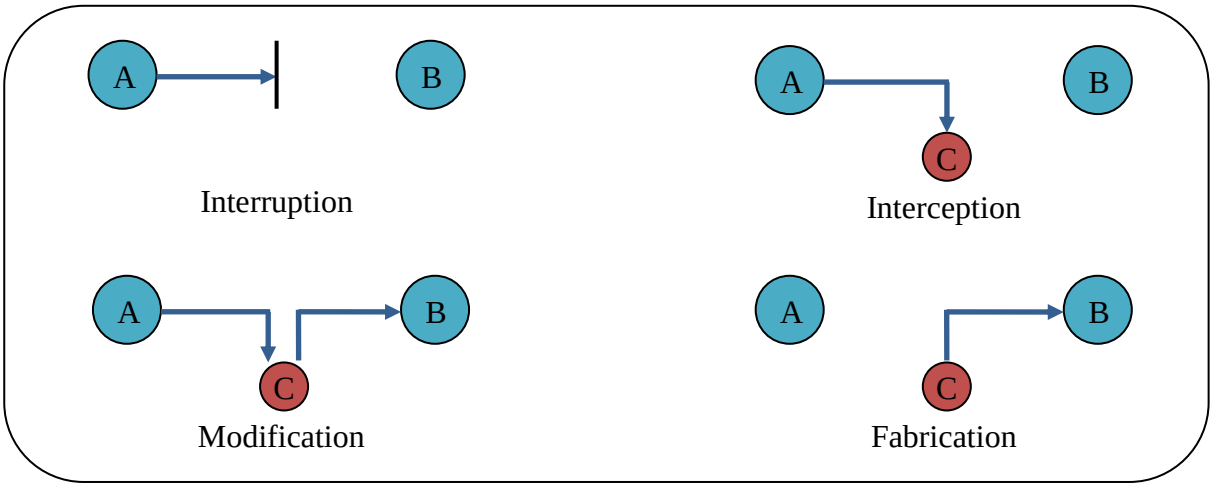

Figure 3.2 : Types d'attaques actives

7.1.2.1. Jamming attack : cette attaque sévit au niveau de la couche physique, son objectif est de causer un déni de service en émettant des signaux à une certaine fréquence, proche de celle utilisée dans le réseau, sur le médium de communication afin de le saturer pour que les nœuds ne puissent pas communiquer entre eux.

7.1.2.2. Usurpation d'identité : c'est l'une des attaques les plus dommageables dans les réseaux informatiques en général et les réseaux de capteurs en particulier. Elle permet à un attaquant de confisquer l'identité de sa victime pour s'acquérir les privilèges qui lui sont associés. Typiquement, l'attaquant se limite uniquement à emprunter l'identité de l'un des nœuds communicants, la source ou la destination, pour pouvoir lire et transmettre des messages en utilisant les coordonnées de sa victime. Cette attaque est initiée par un type d'attaquant appelé man-in-the-middle (la technique de l'homme au milieu).

7.1.2.3. Sinkhole attack (l'attaque du trou de la base) : l'attaquant devient attractif en proposant des chemins optimaux pour atteindre la station de base avec des connexions puissantes ce qui pousse les nœuds émetteurs à modifier leurs tables de routage pour acheminer les données par ce nœud malicieux. Ainsi, toutes les informations qui y transitent pourront être récupérées par l'attaquant [52].

7.1.2.4. Wormhole attack (l'attaque du trou de ver) : ce type d'attaques nécessite au moins deux nœuds malicieux. Les deux nœuds sont liés par une liaison radio puissante ou par une liaison filaire. Ce chemin attaquant-attaquant est à un saut donc plus rapide et plus optimal ce qui facilite la création de deux nœuds

attaquants de type Sinkhole attack et permettre à chacun des deux de récupérer des informations dans un point du réseau, les modifier, puis les transmettre sur l'autre point du réseau en ignorant les nœuds intermédiaires. Les informations compromises seront envoyées vers la station de base. Les protocoles victimes de ce type d'attaques sont ceux basés sur i) la latence de routes, ii) la première route découverte et iii) le nombre de sauts pour atteindre la destination [52].

7.1.2.5. Blackhole attack (l'attaque du trou noir) : le principe est d'insérer un nouveau nœud ou bien compromettre un nœud du réseau pour obliger un maximum de voisins à modifier leurs tables de routage et faire transiter leurs données émises par ce nœud malicieux. Les informations reçues par ce dernier seront détruites et ne seront jamais réinsérées sur le réseau. Blackhole attack vise souvent les architectures hiérarchiques et plus particulièrement les nœuds agrégateurs.

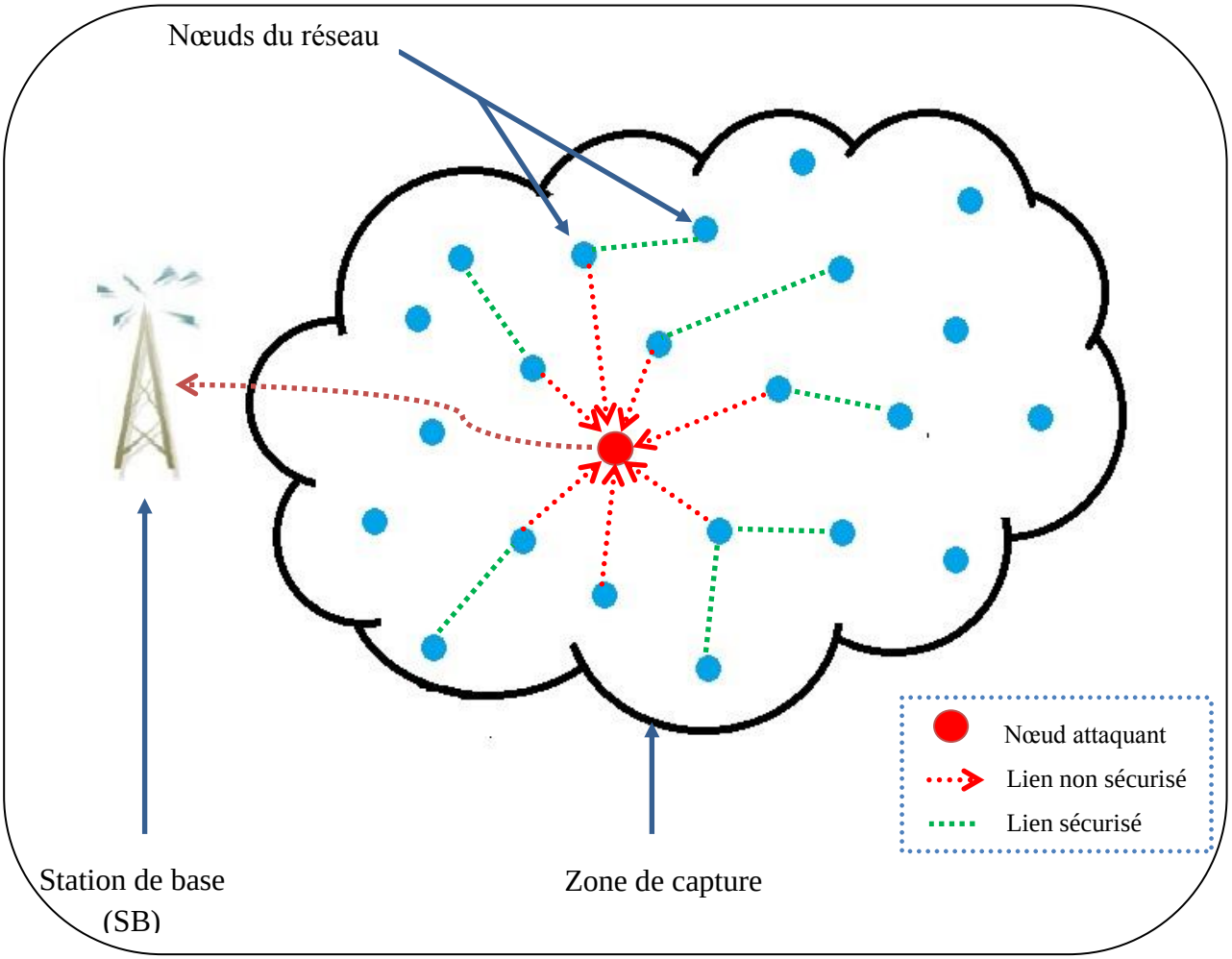

Figure 3.3 : Attaque Sinkhole

7.1.2.6. Greyhole attack (l'attaque du trou gris) : l'attaque du trou gris est une variante améliorée de l'attaque du trou noir. Tout comme le principe de Blackhole, Greyhole attack procède à la modification des tables de routage des nœuds du réseau par l'insertion d'un nouveau nœud ou la compromission d'un nœud du réseau, mais à la différence que les informations récupérées par l'attaquant du trou gris ne seront pas toutes détruites et quelques informations non critiques seraient achemineés correctement. Ce comportement semble normal aux autres nœuds du réseau d'où sa détection est rendue plus difficile.

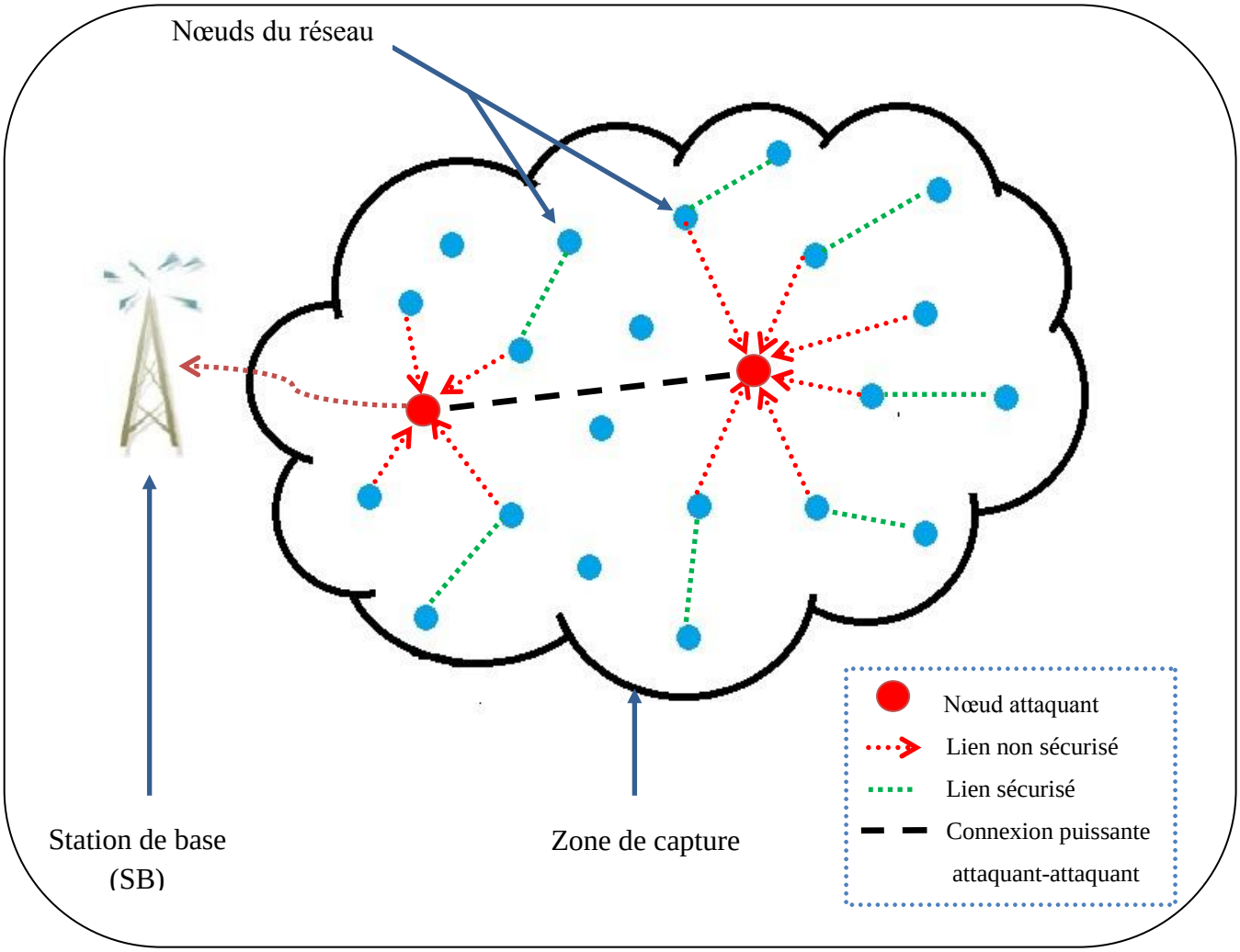

Figure 3.4 : Attaque Wormhole

7.1.2.7. Rushing attack : cette attaque vise particulièrement les protocoles de routage basés sur la première route découverte [42]. Lors du processus d'établissement de routes par la diffusion de requêtes, et à la réception d'une requête de construction, l'attaquant émet automatiquement une réponse via un ou

plusieurs nœuds malicieux insérés dans le réseau même si aucune information sur la destination n'est disponible sur sa table de routage (c'est-à-dire que l'attaquant n'est pas concerné par la requête). Cette réponse lui permet, à fortes chances, d'être choisi comme nœud intermédiaire dans le routage de données. Les informations qui lui seront destinées seront alors compromises.

7.1.2.8. Routing table poisoning : l'attaquant encombre le réseau avec de fausses informations sur des routes fictives ce qui contraint les nœuds à mettre à jour excessivement leurs tables de routage. Les mises à jour engendrent un débordement et les tables de routage seront remplies et saturées par de fausses informations sur les routes.

7.1.2.9. Sybil attack : l'objectif de cette attaque est de faire passer le nœud malicieux pour plusieurs nœuds en lui endossant plusieurs identités, afin de créer une multitude de chemins passant par ce nœud qui ne sont en réalité qu'un seul chemin. Sybil attack s'intéresse aux protocoles basés sur l'instauration d'une redondance de chemin pour assurer la fiabilité du réseau.

7.1.2.10. Flooding attack : l'objectif de cette attaque est de provoquer un déni de service. En effet, un ou plusieurs nœuds malicieux du réseau effectuent des envois réguliers de massages à une puissance d'émission forte dans le but de saturer le réseau.

7.1.2.11. Hello flooding attack : l'objectif de cette attaque est de consommer l'énergie des nœuds capteurs, notamment les plus éloignés, par un envoi continu, à un signal puissant des messages de découverte du voisinage de type HELLO. Les nœuds destinataires du message essayent de répondre au nœud malicieux même s'ils sont situés à des distances lointaines ne permettant pas de l'atteindre. A force de tenter de lui répondre, tous les nœuds concernés par ce message HELLO consomment l'intégralité de leur énergie.

7.1.2.12. Insertion de boucles infinies : cette attaque vise à consommer l'énergie des nœuds capteurs en modifiant leurs tables de routage de manière à générer des boucles infinies entre les nœuds.

7.1.2.13. Injection ou altération de messages : le principe est d'injecter sur le réseau des messages véhiculant de fausses informations sur le routage ou bien récupérer puis altérer les messages circulant sur le réseau. L'objectif d'une telle

attaque est de perturber la fonction du routage et donc le fonctionnement global du réseau.

7.1.2.14. Réplication de données : cette attaque vise la fraicheur des données échangées sur le réseau. En effet, l'attaquant récupère et enregistre des informations au temps (T) puis les reproduire sur le réseau au temps (T+N). Ceci va tromper le système de surveillance et des décisions erronées seront prises. Pour illustrer ce cas, on cite l'exemple d'un RcSF ayant pour mission la détection d'un départ d'incendie. Si un incendie se déclare au temps (T), l'attaquant récupère cette information et va la reproduire ultérieurement faisant croire au centre de contrôle qu'un nouvel incendie s'est produit.

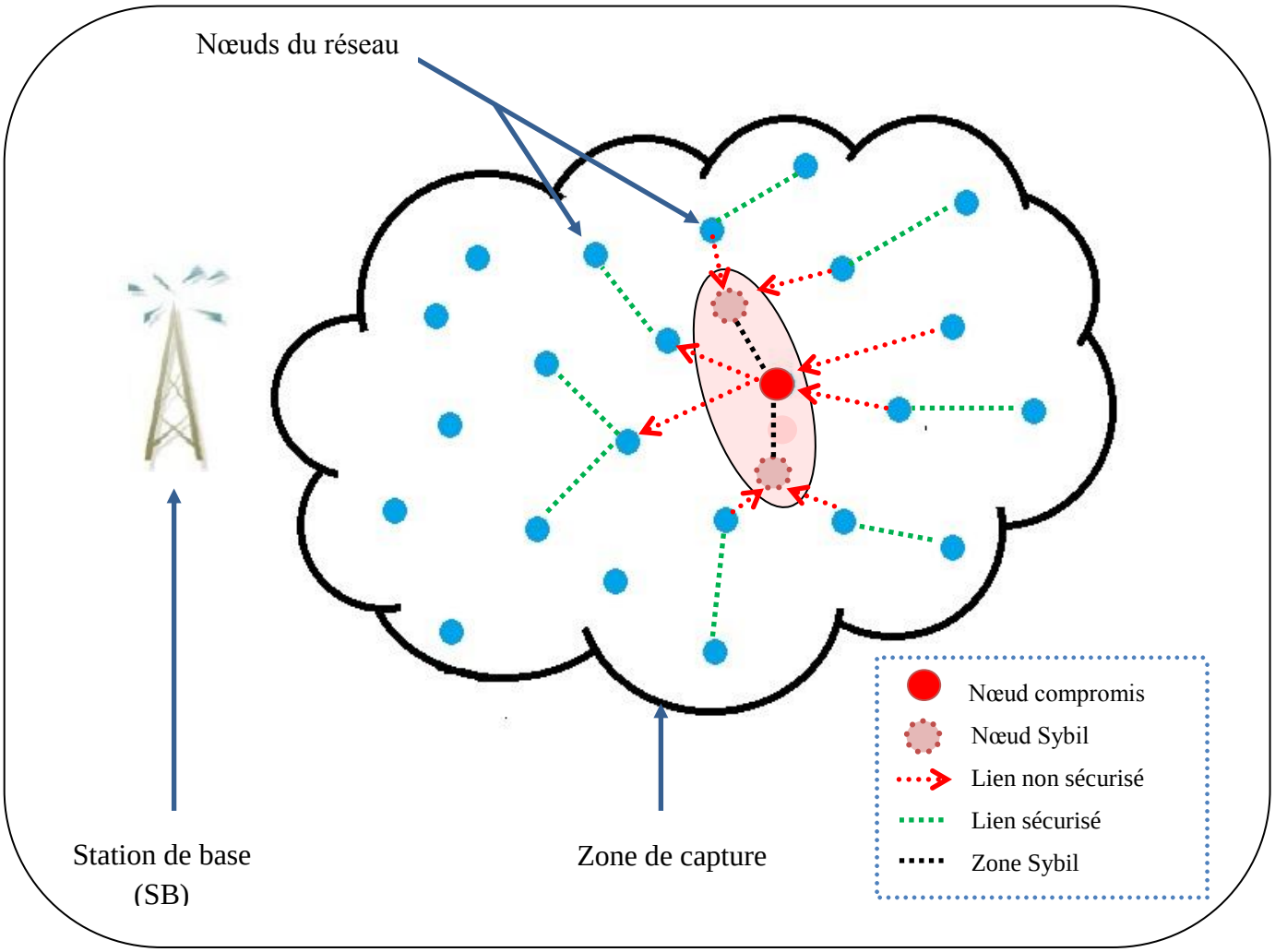

Figure 3.5 : Attaque Sybille

7.1.3. Attaques physiques

Dans la plupart des applications de réseaux de capteurs sans fil, les capteurs sont déployés dans des zones inaccessibles au cœur du territoire ennemi. Le risque d'exposition aux attaques ennemies est très présent, parmi elles on cite :

7.1.3.1. Destruction physique : l'ennemi ou toute personne étrangère à l'application du RcSF, se trouvant sur le périmètre de déploiement, peut subtiliser des nœuds capteurs pour les détruire et ainsi faire perdre au réseau sa connectivité et le diviser en sous réseaux incapables de communiquer entre eux et avec la station de base, ou bien les récupérer pour divulguer leurs informations (données, clés, informations de routage…etc.) et de les modifier selon ses besoins. De plus, l'attaquant peut les reprogrammer puis les réinsérer sur le réseau et ainsi devenir des nœuds malicieux fonctionnant à son service [78] [95].

7.1.3.2. Attaques spécifiques aux types de capteurs : en déduisant le type et la tâche du capteur déployé, l'attaquant peut le faire réagir en produisant un événement attendu par le capteur, comme allumer une lampe devant un capteur de luminosité ou allumer une flamme devant un capteur thermique. Le capteur réagit et transmet de fausses informations en continu à la station de base jusqu'à l'épuisement de son énergie.

7.1.3.3. Empêcher le capteur de se mettre en veille : l'attaquant essaye d'empêcher un nœud capteur de se mettre en veille par différents moyens. Le fonctionnement sans cesse du capteur lui causera la perte de son énergie.

7.1.4. Vers et virus informatiques

Les virus et les vers informatiques s'attaquent à des machines reliées en réseau afin de les compromettre et de générer un dépassement de capacité mémoire. Comme les capteurs sont des mini-ordinateurs possédant un processeur, une mémoire et un système d'entrée / sortie, tout porte à penser que des attaques par virus et vers y sont possibles. Cependant, et en raison des caractéristiques particulières des capteurs, les attaques par compromission à distance semble une tâche ardue pour les attaquants [21].

7.2. Mécanismes de sécurité

Plusieurs solutions ont été développées pour contrer les attaques sur les RcSF. On distingue des solutions spéciales au routage où la fonction de sécurité concerne la protection de la donnée transmise, d'autres solutions sont conçues pour protéger les communications entre les nœuds capteurs en assurant la fiabilité des liaisons par des

fonctions cryptographiques, et enfin des solutions qui se chargent de protéger les nœuds du réseau, de détecter et d'exclure les nœuds malicieux par des mécanismes de détection d'intrusions. Dans [27], on propose les mécanismes simples, les plus efficaces et les plus appropriés aux RcSF, regroupés selon l'objectif de la solution proposée :

7.2.1. Partitionnement de données

C'est une solution proposée dans le but d'empêcher les attaquants de récupérer l'intégralité des informations qui circulent sur le réseau. Le principe est de découper, au niveau du nœud source, l'information à transmettre en plusieurs paquets de tailles fixes et chaque paquet de données sera envoyé sur un chemin différent. L'entité de destination finale, et après la réception de tous les paquets de données, procède à la reconstitution du message initial émis par le nœud source. Afin de saisir intégralement l'information échangée entre la source et sa destination, un attaquant doit scruter tous les chemins utilisés dans la transmission du message, ce qui est difficile voire impossible vu la taille des réseaux et le nombre important de chemins possibles entre chaque paire de nœuds. Cependant, cette solution est gourmande en énergie car elle implique un grand nombre de nœuds pour acheminer chaque paquet vers l'entité destinatrice. Un exemple d'une telle solution est donné par la figure (3.6).

7.2.2. La cryptographie

Le mot cryptographie est composé de deux mots grecs : « crypto » qui signifie *caché* et « graphie » qui signifie *écrire*, d'où la signification complète de la cryptographie est « *l'écriture secrète* ». La cryptographie est définie comme la science qui convertit les informations *en clair* en informations *cryptées* c'est-à-dire codées. Le principe est donné par la figure (3.7). La plupart des mécanismes de sécurité des communications sont basés sur des outils cryptographiques utilisant des informations secrètes représentées par des nombres premiers, dites clés, combinées, en entrée d'une opération cryptographique, au message à coder pour produire le message crypté. On distingue quatre types de clés utilisées dans les opérations cryptographiques : clé individuelle, clé par paire, clé de groupe et clé globale. Plusieurs outils cryptographiques sont utilisés pour assurer la sécurité du routage, des entités du réseau et des liens.

Pour assurer un système cryptographique fiable, deux communicants doivent choisir soigneusement leur clé de chiffrement/déchiffrement et doivent appliquer les principes suivants, car un attaquant peut essayer un certain nombre de possibilités et par chance tomber rapidement sur la solution :

- La sécurité doit se reposer sur le secret de la clé et non pas sur la sécurité de l'algorithme.
- Le déchiffrement sans connaissance préalable de la clé doit être impossible en un temps raisonnable.
- Calculer la clé à partir du texte en clair et du texte chiffré doit être impossible en un temps raisonnable.

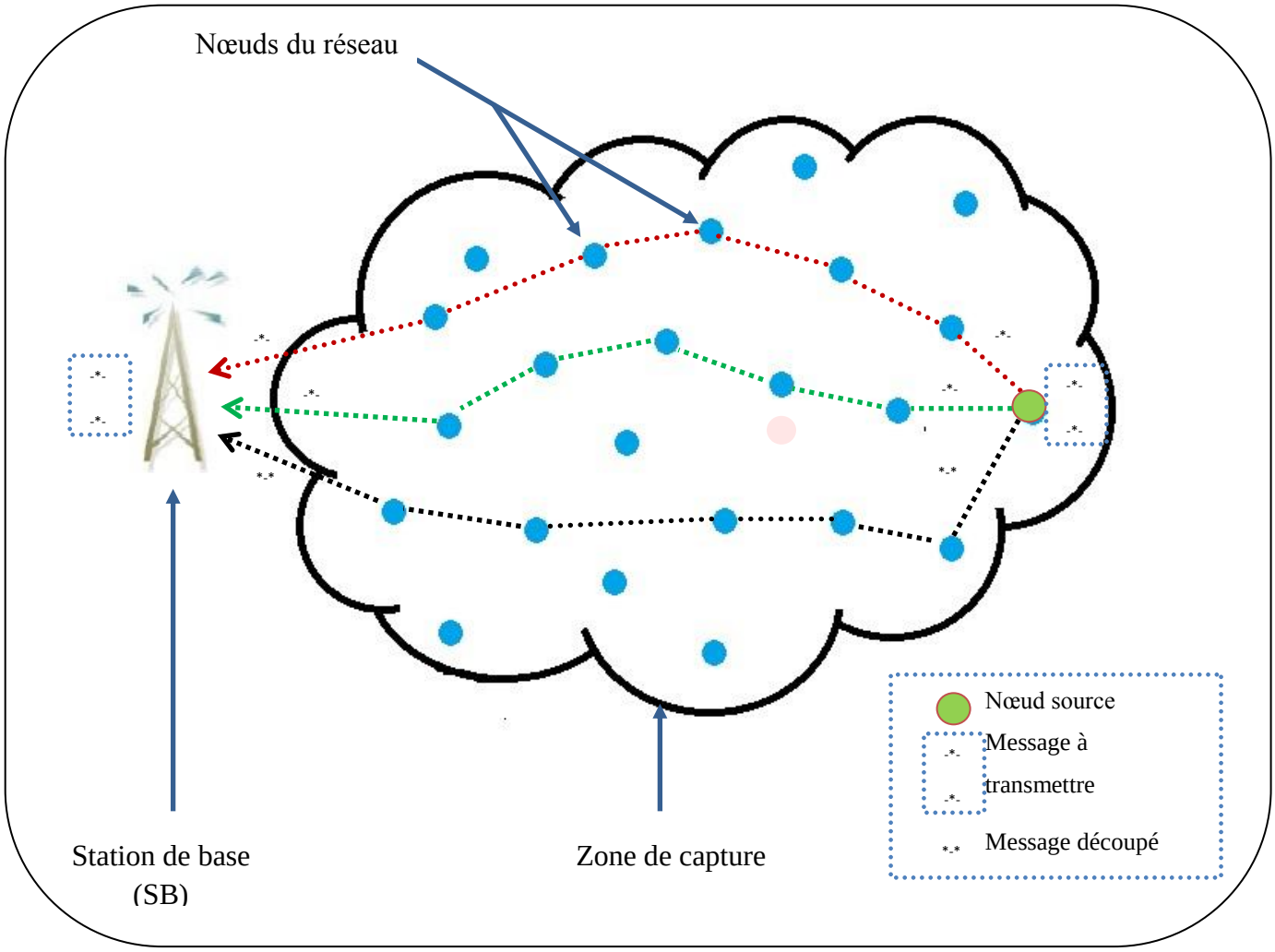

Figure 3.6 : Technique de partitionnement de données

Nous allons étudier plus profondément les techniques de la cryptographie et de gestion de clés dans le chapitre à suivre.

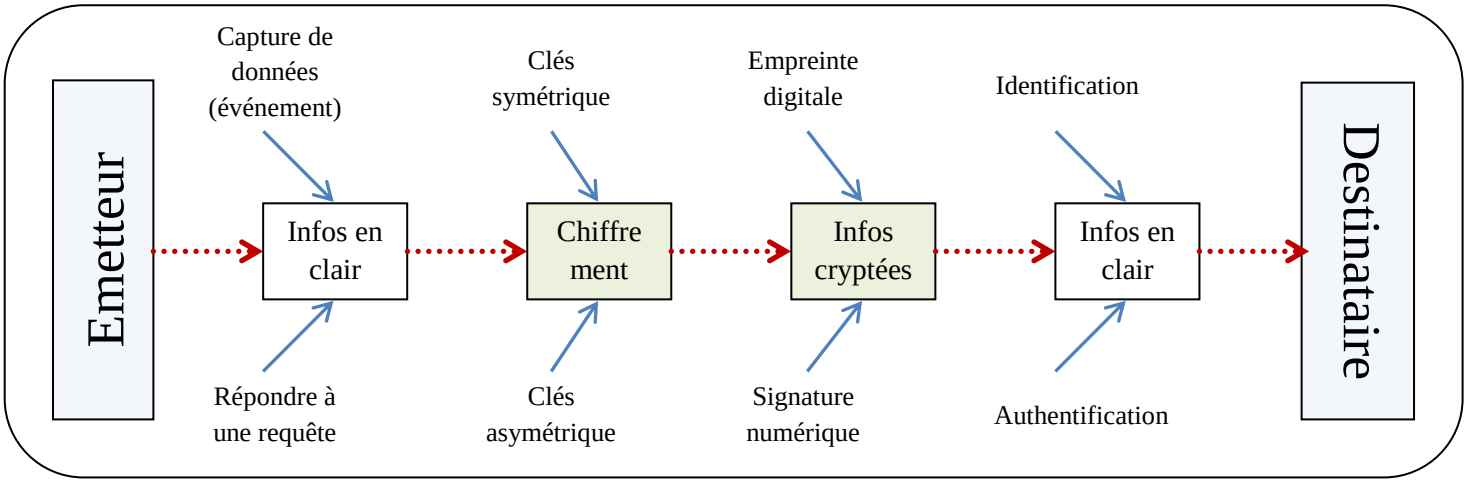

Figure 3.7 : Principe de la cryptographie

7.2.3. Détection d'intrusions

7.2.3.1. Détection d'intrusions par localisation : pour les besoins de cette solution proposée par [59], les nœuds capteurs doivent être équipés de moyens qui permettent de connaitre leurs positions géographiques (exemple le GPS). Ce type de nœuds capteurs sont dits capteurs balises. Si un capteur souhaite faire partie du réseau, il adresse une demande d'insertion aux capteurs balises afin d'estimer sa localisation par rapport au domaine d'écoute. Les capteurs balises vont quadriller par la suite leurs zones d'écoutes respectives et demandent aux nœuds qui ont reçu la demande d'insertion de voter pour la zone de quadrillage qu'ils peuvent entendre. La zone qui obtient le plus de voix sera considérée comme celle où est censé se trouver le capteur.

7.2.3.2. Détection d'intrusions par indice de confiance : contrairement à la solution de détection d'intrusions présentée précédemment où des moyens supplémentaires et couteux sont exigés, la détection d'intrusions par indice de confiance consiste à générer une alerte pour tout scénario ou comportement suspect détecté.

- **Approche par comportements :** observer le comportement du système et le comparer au comportement normal. Si le comportement observé est différent du comportement normal, on conclu que le système présente des anomalies et fait objet d'intrusions. L'avantage est la rapidité et la simplicité de détection alors que l'inconvénient est le nombre important de faux positifs que le système détecte à cause d'éventuels changements inattendus qui ne correspondent pas à des attaques.

- **Approche par scénarios :** son fonctionnement ressemble beaucoup au fonctionnement des anti-virus et des anti-trojan des systèmes informatiques classiques. En effet, on utilise une base de signatures où sont répertoriés les différents scénarios d'attaques. Les données reçues sont analysées afin de détecter d'éventuels scénarios d'attaques prédéfinis dans la base de signatures. Cette approche présente l'avantage de précision de diagnostic par rapport à l'approche par comportement mais elle est incapable de détecter les nouvelles attaques non répertoriées.

Le tableau (3.1) résume les attaques selon leurs objectifs ciblés et les mécanismes appropriés pour les contrer.

8. Menaces sur les protocoles de routage

Pour qu'un protocole de routage pour les réseaux de capteurs sans fil soit efficace, il doit être conçu pour contrecarrer les menaces sur les informations communiquées par les nœuds du réseau. Toutefois, les différentes contraintes des RcSF obligent une conception assez simple des protocoles de routage et par conséquent assez vulnérable aux attaques.

Les attaques sur les protocoles de routage dans les RcSF sont multiples : accès non autorisé, les écoutes passives, usurpation d'identités, dénis de services (DoS), modification, destruction et falsification des messages du réseau…etc. Les objectifs de ces attaques peuvent avoir pour but de modifier le protocole lui-même, de l'empêcher de construire et de maintenir le réseau, le soumettre à insérer des fausses informations sur les tables de routage des capteurs afin de perturber la topologie du réseau.

L'objectif commun des attaques est de perturber le trafic et de le faire passer par un nœud contrôlé par l'attaquant. Les solutions visant à prévenir ces attaques font souvent appel à la cryptographie.

Type d'attaque	Objectif ciblé	Mécanisme de sécurité
Attaques passives	Analyse du trafic	- Authentification des messages - Cryptographie - Partitionnement
Jamming **Flooding**	Déni de service	- Détection précoce d'une quantité excessive de paquets émis sur le réseau.
Sinkhole	Disponibilité	- Cryptographie - Authentification de la source

Wormhole **Blackhole** **Greyhole**	L'intégrité Confidentialité Fraicheur	- Authentification des messages
Sybil	Fiabilité du réseau	- Authentification de la source - Cryptographie - Contrôle des certificats de sécurité - Détection d'identificateurs multiples sur le réseau
Hello flooding	Consommation d'énergie des capteurs	- Authentification de la source - Cryptographie
Rushing **Routing table poisoning**	Informations de routage	- Authentification des messages - Contrôle d'accès aux tables de routage - Cryptographie
Capture physique	Récupérer les informations chargées sur le capteur	- Détection d'intrusions
Empêcher le capteur de se mettre en veille	Consommation d'énergie des capteurs	- Cryptographie - Authentification des messages - Authentification de la source - Vérification des certificats de sécurité - Contrôle d'accès aux tables de routage

Tableau 3.1 : Objectifs et solutions des attaques

9. Conclusion

Le développement de mécanismes de sécurité spécifiques aux réseaux de capteurs sans fil a attiré une grande part d'intention parmi les chercheurs de la communauté scientifique du domaine. En outre, les solutions de sécurité proposées pour les autres réseaux sans fil (ad hoc, Wifi, Bluetooth…etc.) ne sont pas applicables directement dans les RcSF. En effet, l'absence de sécurité physique et les limites en ressources énergétiques des capteurs, entre autres, s'imposent comme contraintes importantes dans la conception des solutions de sécurité efficaces et économes en énergie pour les réseaux de capteurs sans fil.

Dans ce chapitre, nous avons présenté un état de l'art sur la sécurité du routage dans les RcSF. Nous avons défini l'objectif des attaquants qui profitent des spécificités réduites de ce type de réseau pour exploiter les failles de sécurité des solutions existantes. D'autre part, les

moyens et les mécanismes de sécurité existants dans la littérature pour garantir un niveau de sécurité élevé pour les échanges de données sur le réseau.

Dans ce qui suit, nous allons étudier de façon plus complète les techniques de la cryptographie et de gestion de clés dans les RcSF.

CHAPITRE 04
CRYPTOGRAPHIE ET GESTION DE CLÉS DANS LES RCSF

Les points abordés dans ce chapitre traitent les techniques de la cryptographie et de gestion de clés dans les réseaux de capteurs sans fil. Nous avons abordé dans le chapitre précédent des solutions de sécurité pour le routage de données sans toutefois prendre réellement en considération les capacités limitées de calcul, de stockage et les ressources faibles en énergie des capteurs. Parmi ces solutions, nous allons nous intéresser dans ce qui suit à la cryptographie, une solution qui répond le mieux aux objectifs de sécurité et aux capacités limitées des RcSF. Ensuite, nous allons étudier la gestion de clés dans les systèmes cryptographiques, et définir son objectif, ses contraintes de mise en place, ses phases de déroulement et nous allons terminer par présenter quelques schémas de gestion de clés pour les RcSF.

1. Introduction

La cryptographie est une technique de sécurité simple utilisant le principe de clé, occupant une taille mémoire minimal et facile à calculer, pour le cryptage et le décryptage des informations à émettre ou à recevoir. Elle peut définir quatre types de clés :

- *Clé individuelle :* chaque nœud possède une clé personnelle partagée uniquement avec la station de base. Après chaque envoi, le nœud émetteur chiffre son message avec sa clé personnelle et restera anonyme sur le réseau jusqu'à atteindre la station de base. Cette solution présente des avantages de sécurité pour sa simplicité mais demeure inappropriée pour les architectures où on effectue des traitements sur les données (exemple : agrégation de données).
- *Clé globale :* une seule et même clé est partagée par tous les nœuds du réseau. Un message est chiffré et déchiffré par la même clé (principe de clé symétrique). Cette solution est très économe en énergie mais moins sure et moins sécurisante. Toutefois, si un attaquant récupère la clé, il pourra déchiffrer tous les messages circulant sur le réseau.
- *Clé partagée par paire de nœuds :* c'est une solution plus sûre mais plus couteuse en énergie. Le principe est que chaque nœud du réseau partage une clé avec son voisin, ainsi si un nœud possède *n* nœuds voisins, il doit stocker *n* clés. Après chaque envoi, le

nœud source chiffre son message avec la clé de son voisin et après chaque réception, le nœud destinataire déchiffre le message avec sa clé et le chiffre à nouveau avec la clé de son voisin, et ce, jusqu'à atteindre la station de base. Une quantité énorme d'énergie sera dissipée dans les opérations de chiffrement et de déchiffrement pour chaque envoi.

- *Clé partagée par groupe de nœuds :* les clés sont établies sur deux niveaux : i) une clé partagée par les nœuds du même cluster et ii) une clé partagée par les nœuds chefs de zones. Le chiffrement se fait avec une clé individuelle entre chaque paire de nœuds du même cluster et avec une clé globale entre les nœuds chefs de zone.

Selon les applications et le niveau de sécurité désiré, la clé est peut être un nombre premier qui varie de 32 (2^5) bits à plusieurs milliers (2^{20}) ou plus. On utilise les nombres premiers car il n'existe, du moins pour le moment, aucune méthode connue pour les retrouver rapidement. Egalement, la sécurité ne se repose plus uniquement sur le secret partagé mais sur des problèmes, notamment mathématiques, connus de tous (exemple : la factorisation) et d'une information connue de tous (la clé publique).

2. Les outils de la cryptographie

Les outils cryptographiques utilisant le principe de clé pour sécuriser les liens de communication sont nombreux, on cite :

2.1. Le chiffrement

Le chiffrement est un système basé sur des clés cryptographiques pour assurer la confidentialité des données. Les clés peuvent être privées (symétriques) ou publiques (asymétriques).

2.1.1. *Le chiffrement symétrique :* une même clé est utilisée par une paire de nœuds pour chiffrer et déchiffrer les messages échangés entre eux en utilisant un algorithme de chiffrement symétrique. Le chiffrement symétrique se fait soit en fractionnant les données en bit à bit (chiffrement par flots) ou bien en blocs de taille fixe (chiffrement par blocs).

- Les algorithmes de chiffrement par blocs : Ce sont les algorithmes les plus utilisés dans les systèmes cryptographiques. Les algorithmes de chiffrement par blocs (Block Cipher en anglais) décomposent les données à transmettre en blocs de tailles généralement égales (comprises entre 64 et 512 bits) et sont ensuite chiffrés les uns après les autres selon différents modes (ECB, CBC, OFB, CTR…etc.). Outre la confidentialité, les algorithmes de chiffrement par blocs peuvent être

utilisés dans d'autres outils de cryptographie comme les fonctions de hachage et les codes MAC (Message Authentication Code).

- Les algorithmes de chiffrement par flots : ils convertissent les données en clair en texte chiffré bit par bit en combinant les flux de bits en clair avec un flux de bits aléatoires. L'opération de combinaison est souvent bijective (un ou exclusif (XOR)). Le déchiffrement est réalisé par l'inversement de l'opération bijective en combinant les données chiffrées avec le même flux aléatoire de bits pour retrouver les données en clair. Le flux de bits aléatoire est généré avec un générateur pseudo-aléatoire initialisé avec la clé de chiffrement et un vecteur public d'initialisation (IV) différent pour chacun des messages à chiffrer.

L'inconvénient des algorithmes de chiffrement symétrique est la difficulté de distribuer les clés car chaque nœud a besoin d'une clé partagée avec un autre nœud du réseau, et si on considère que le nombre de nœuds du réseau est *n*, le système aura à gérer *n*(n-1)/2* clés. Malgré cela, les algorithmes de chiffrement à clé privée ne requièrent aucune opération mathématique complexe et gourmande en énergie durant les phases de chiffrement et de déchiffrement.

Exemples d'algorithmes symétriques : AES (Advanced Encryption Standard) [74], DES (Data Encryption Standard) [73], RC4 (Rivest Cipher 4) [34].

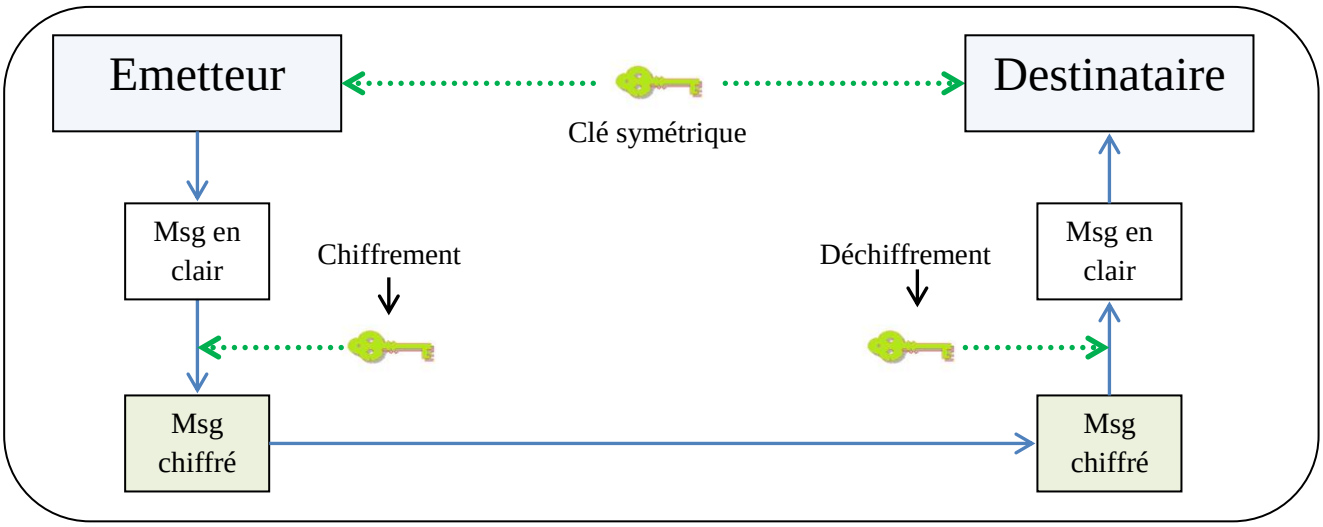

Figure 4.1 : Chiffrement symétrique

2.1.2. *Le chiffrement asymétrique :* deux clés sont générées par le récepteur du message : une clé privée maintenue chez le récepteur et une clé publique diffusée à tous les nœuds émetteurs. La clé publique sert au chiffrement d'un message et la clé

privée sert au déchiffrement du même message. La puissance de ce mode de chiffrement repose sur l'impossibilité de déduire la clé privée à partir de la clé publique. La fonction de distribution de clés est fortement simplifiée où chaque nœud a besoin uniquement d'une paire de clés, et si on considère que le nombre de nœuds du réseau est *n*, le système aura à gérer *2*n* clés. Cependant, la complexité des opérations mathématiques et le stockage de toutes les clés publiques de tous les autres nœuds exigent des capacités de traitement et de stockage importantes synonyme d'une consommation conséquente d'énergie.

Exemples d'algorithmes asymétriques : RSA (Rivest Shamir Adleman) [109], ECC (Elliptic Curve Cryptography) [53] [69].

2.1.3. *Le chiffrement mixte :* bien que le chiffrement asymétrique soit pratique et le chiffrement symétrique est performant, la combinaison des deux techniques s'avère intéressante et elle a tendance à devenir un système cryptographique moderne. Pour cela, on génère et on échange avec une clé publique une clé de session symétrique et on utilise cette dernière pour le chiffrement des messages échangés.

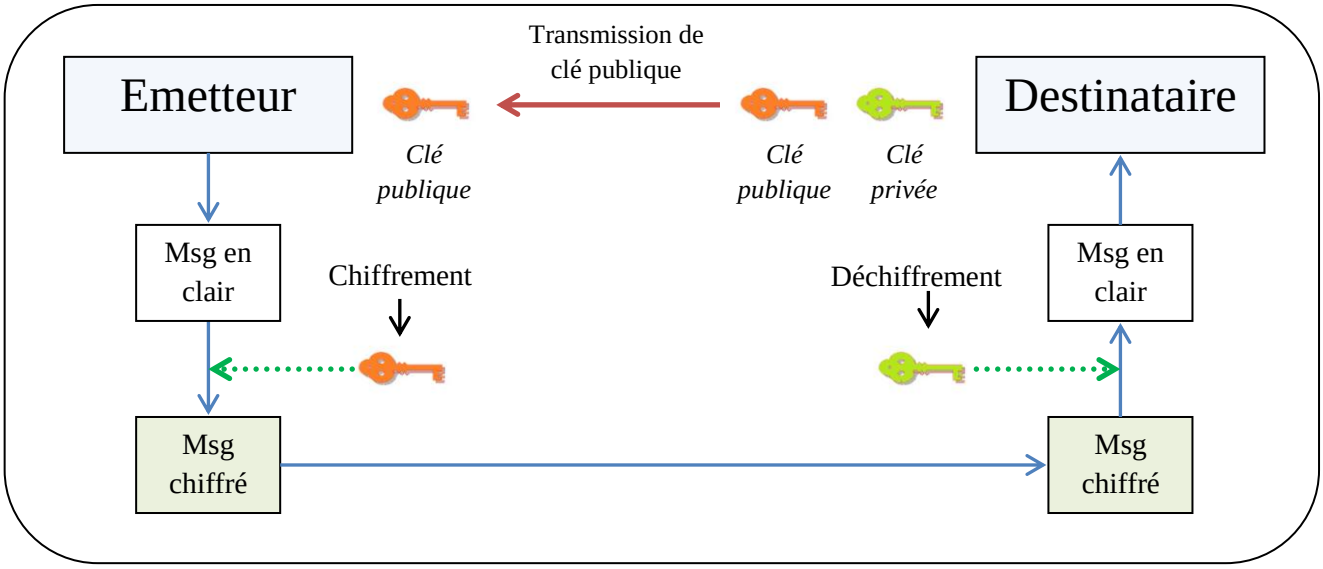

Figure 4.2 : Chiffrement asymétrique

2.2. La signature digitale

C'est un système cryptographique qui repose sur des clés asymétriques pour assurer la non-répudiation de la source. L'émetteur produit une signature digitale et signe son message avec sa clé privée. Le récepteur tente de déchiffrer le message reçu en utilisant

sa clé publique. Si le message est déchiffré, l'émetteur ne pourra plus nier l'émission de ces données par la suite.

2.3. Les fonctions de hachage

C'est un système qui assure l'intégrité des informations qui circulent sur le réseau. La fonction de hachage sert à calculer une courte empreinte de taille fixe à partir d'une information de taille arbitraire pour remplir les trois conditions suivantes :

- Calculer l'empreinte facilement à partir du contenu du message.
- La difficulté de trouver le contenu du message à partir de l'empreinte (la fonction est à sens unique, c'est-à-dire à partir de H(M), il est impossible de trouver M).
- La difficulté de trouver une même empreinte pour deux messages aléatoires différents (c'est-à-dire à partir de H(M) et M, il est impossible de trouver $M' \neq M$ tel que $H(M') = H(M)$) et cela mène à la résistance aux collisions.

Exemple de fonctions de hachage : MD5 (Message Digest5) [86], SHA-1 (Secure Hash Algorithm) [31].

2.4. Le code d'authentification de messages MAC

C'est un système cryptographique qui permet de vérifier l'authenticité de l'origine et l'intégrité des informations reçues. Le code d'authentification de messages MAC (Message Authentication Code) est produit par des fonctions de hachage à clé symétrique puis envoyé avec les données. Le récepteur recalcule le code MAC avec la même clé (puisque on utilise une clé symétrique) et le compare au code reçu. Si les deux codes sont identiques, la source est authentifiée et les données sont enregistrées, sinon la source est suspectée et les données sont considérées comme altérées.

Exemples de code MAC : HMAC (keyed-Hash Message Authentication Code) [49], CBC-MAC [75].

3. Avantages et inconvénients de la cryptographie

3.1. Cryptographie asymétrique

- **Avantages**
  - Permet la scalabilité.
  - Efficace contre la capture des nœuds.
  - Facilité de la distribution des clés.

- Signature facile des messages.
- Nombre réduit de clés à distribuer.

- **Inconvénients**
  - Taille des clés.
  - Trop lente.
  - Gourmande en ressources physiques et énergétiques.
  - Vulnérable aux attaques de type déni de service où un attaquant inonde le réseau avec des certificats numériques illégales obligeant les nœuds à calculer des clés publiques jusqu'à épuisement de leur énergie.

3.2. Cryptographie symétrique

- **Avantages**
  - Calcul non couteux et rapide des clés.
  - Les clés sont relativement courtes (en moyenne 128 bits).
  - Facilité d'implantation sur hardware.

- **Inconvénients**
  - La distribution de clés entre les entités communicantes est très problématique.
  - Le nombre de clés à gérer accroit sensiblement avec le nombre de nœuds déployés.
  - Certaines propriétés sont difficiles à réaliser (exemple : la signature).

4. Cryptographie symétrique Vs Cryptographie asymétrique ?

Dans les systèmes basés sur la cryptographie symétrique, les faiblesses des nœuds capteurs sont contournées par la réduction des opérations de calculs et de stockage de clés. Contrairement aux systèmes symétriques, la cryptographie asymétrique nécessite des temps de calcul importants et une consommation d'énergie énorme due au calcul des algorithmes à clés publiques et à la transmission des certificats numériques utilisés par le destinataire pour authentifier la clé publique reçue. Cependant, la gestion de clé dans les systèmes à clé privée (symétriques) est beaucoup plus compliqué à cause des difficultés rencontrées lors de la distribution, l'échange et la découverte des clés entre nœuds voisins. Un problème que résout le système à clés publiques par l'introduction de certificats numériques pour une seule paire

de clés (privée / publique) par nœud utilisée dans le chiffrement et le déchiffrement des messages.

Bien que le chiffrement à clé publique possède de gros avantages, il demeure inapproprié pour les RcSF pour sa lenteur d'exécution et son cout considérable en termes de ressources physiques et énergétiques. Cependant et malgré que le chiffrement à clé privée possède des inconvénients liés au problème de l'établissement de clés entre les nœuds, il demeure le système le plus approprié pour les RcSF.

Les caractéristiques spéciales aux réseaux de capteurs sans fil ne permettent pas d'utiliser des méthodes cryptographiques complexes. En effet, le temps de calcul et la consommation d'énergie dans les traitements doivent être raisonnables. Pour ces deux raisons, l'emploi de techniques de cryptographie simples est indispensable. Dans ce qui suit, nous allons étudier en détail la fonction de gestion de clés pour la conception des systèmes cryptographiques simples et efficaces pour les RcSF.

5. La gestion de clés dans les RcSF

5.1. Définition

La gestion de clés constitue la fonction basique la plus critique dans la conception d'un système cryptographique. C'est le processus par lequel les clés sont générées, affectées, stockées, protégées, vérifiées, révoquées, renouvelées et détruites [98]. La gestion de clés assure le couplage des caractéristiques propres aux RcSF aux exigences de sécurité habituelles telles que la disponibilité, l'intégrité, la confidentialité et l'authentification des entités communicantes dans le but de sécuriser le routage et de renforcer la coopération entre les nœuds du réseau.

En effet, la sécurité des clés chargées sur les nœuds ou partagées entre eux est égale à la sécurité du réseau entier. Pour cela, la gestion de clés doit tenir compte des ressources limitées des nœuds capteurs et de la possibilité de leur compromission d'où la plupart des solutions proposées s'adaptent à ces contraintes et utilisent la cryptographie symétrique convenable à ce type de réseaux. Cela malgré les efforts déployés pour prouver qu'on peut utiliser le principe de cryptographie asymétrique ou combiner les deux types de chiffrement pour une sécurité meilleure et économe en ressources. Les résultats de ces efforts sont, pour le moment, peu convaincants.

Pour qu'un système soit entièrement sécurisé, chacune des entités du réseau doit disposer d'un ensemble de clés privées (dans un système à clés privées) ou de paire de clés publiques / privées (dans un système à clés publiques) afin de chiffrer (déchiffrer) les

informations émises (reçues). Les clés sont générées et distribuées sur les capteurs de différentes manières : i) pré-chargées sur les nœuds capteurs avant leur déploiement, ii) générées sur les nœuds capteurs après le déploiement, iii) générées par la station de base et distribuées sur les nœuds capteur après le déploiement…etc.

5.2. Objectifs de la gestion de clés

La gestion de clés dans les réseaux de capteurs sans fil permet de :

- Sécuriser le routage d'informations.
- Sécuriser l'agrégation de données.
- Renforcer la coopération entre les nœuds en utilisant des mécanismes d'authentification.
- Garantir un système cryptographique fiable et sécurisé en sécurisant les liens et en protégeant les nœuds contre la compromission.

5.3. Contraintes de mise en place d'un système efficace

Les contraintes qui découlent des propriétés des RcSF sont résumées sur la figure ci-dessous :

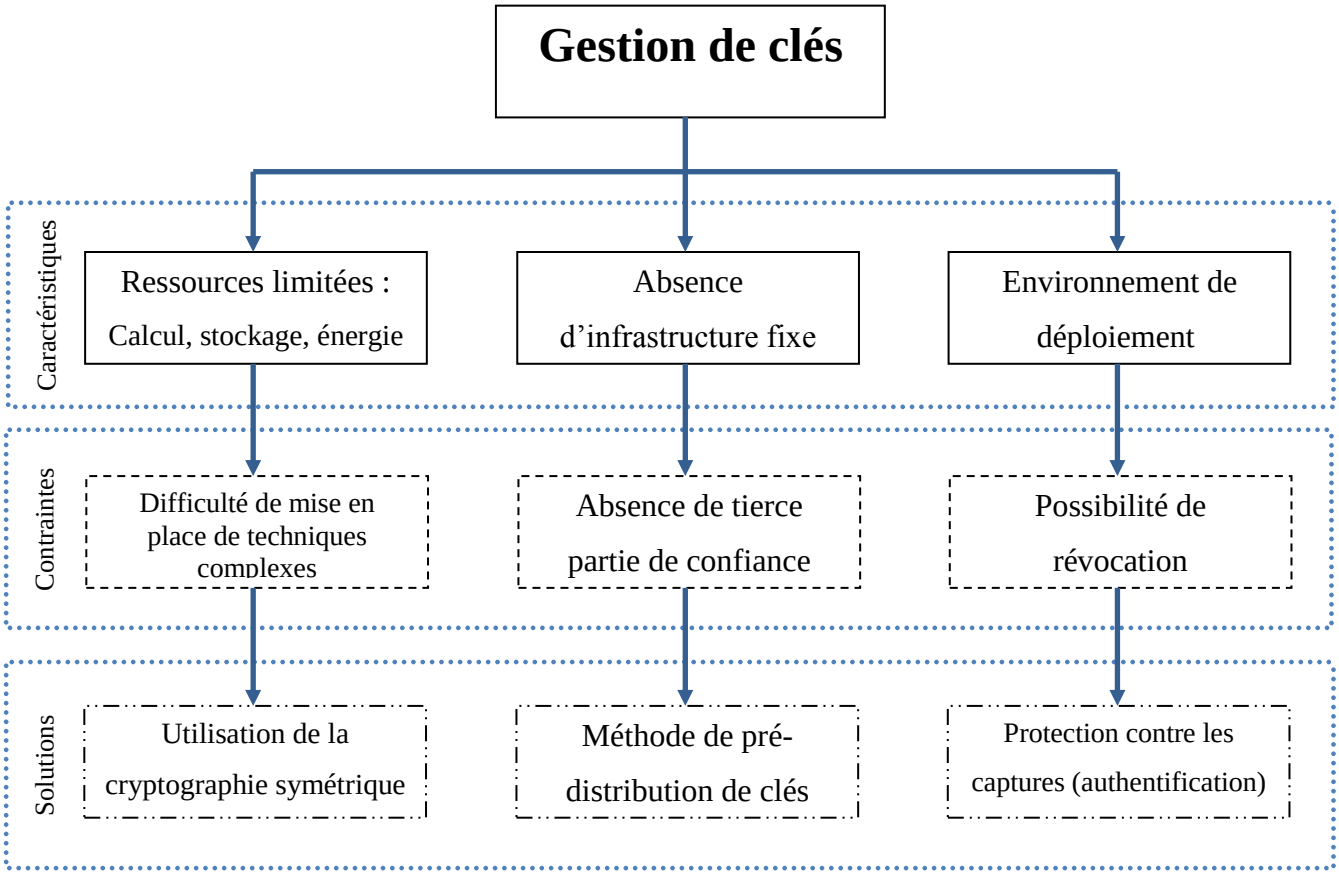

Figure 4.3 : Contraintes de mise en place d'un système de gestion de clés

Le problème de gestion de clés dans les RcSF est décomposé par [44] en :

5.3.1. *Pré-chargement des clés :* dans des applications où le déploiement est aléatoire, l'établissement de clés se fait avant le déploiement des capteurs sur la zone de captage.

5.3.2. *Découverte du voisinage :* après le déploiement, chaque nœud capteur procède à l'identification de ses voisins afin d'établir des liens sécurisés entre eux et de former un graphe connexe. Un lien existe entre deux nœuds du réseau si et seulement s'ils partagent une clé.

5.3.3. *Etablissement de chemins sécurisés :* établir des liens entre des nœuds non liés directement est un problème complexe car le routage dans les RcSF est à multi-sauts et la clé de chemin « path key » partagée entre une paire de nœud d'un chemin doit être différente de la clé partagée par les nœuds voisins à cette même paire de nœuds.

5.3.4. *Mise en quarantaine des nœuds suspectés :* un nœud suspect est un nœud qui ne fonctionne pas comme indiqué pour l'une des raisons suivantes : i) compromis par un attaquant, ii) incapable d'assurer ses taches car il a épuisé totalement son énergie. Les nœuds suspects doivent être isolés pour ne pas altérer le bon fonctionnement du système.

5.3.5. *La mise à jour des clés :* le « re-keying » constitue un défi majeur pour le système de gestion de clés car les clés découvertes par les nœuds attaquants doivent être révoquées et détruites, d'autres doivent être générées et distribuées sur les nouveaux nœuds capteurs ou bien en remplacement des clés supprimées.

5.4. Etapes de gestion de clés

Les phases de gestion de clés dans les réseaux de capteurs sans fil sont données par la figure ci-dessous :

(1) A partir d'un paramètre de sécurité choisi, un algorithme de génération de clés produit un ensemble de clés aléatoires de *n* bits.

(2) Les clés sont pré-chargées sur les capteurs avant le déploiement.

(3) Après le déploiement, les clés sont chargées sur les capteurs via une tierce partie de confiance et via des liens sécurisés.

(4) Les clés sont stockées soit sur la mémoire vive (Ram) des capteurs ou bien sur les mémoires mortes (Eprom), selon la stratégie de sécurité choisie.

(5) Des requêtes de découverte de voisinage sont lancées par les nœuds capteurs.

(6) Des liens sécurisés entre chaque paire de nœuds voisins sont établis.

(7) Une vérification systématique est lancée à chaque intervalle de temps ou à chaque tentative de compromission détectée sur le réseau afin de neutraliser la menace.

(8) Le renouvellement des clés se fait soit i) volontairement : après une période de temps, ii) préventivement : lors d'une tentative d'accès illégale par un attaquant ou bien iii) obligatoirement : après la compromission d'un ou de plusieurs nœuds capteurs du réseau.

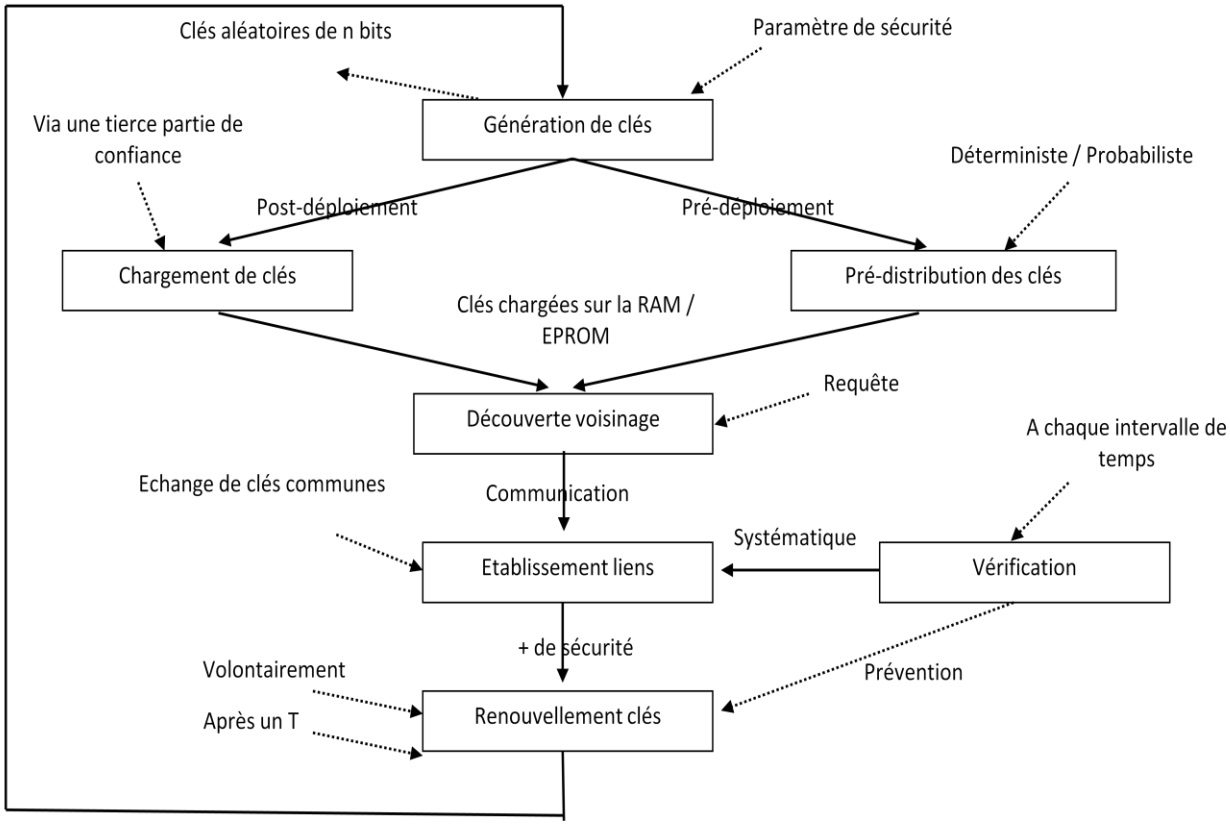

Figure 4.4 : Schéma de gestion de clés

5.5. Classification des méthodes de gestion de clés

Les méthodes de gestion de clés peuvent être classées en quatre catégories de protocoles : déterministes, probabilistes, géographiques et dits t-secure.

5.5.1. Les protocoles déterministes

C'est la première solution proposée dans la littérature. La génération de clés entre deux voisins se fait d'une manière déterministe. Une clé commune est pré-chargée sur tous les capteurs avant le déploiement. Après le déploiement, chaque nœud

prendra connaissance de ses voisins et générera une clé par-paires avec eux et la clé commune sera effacée après cette génération.

Les inconvénients sont, entre autres, le nombre élevé de messages échangés entre les nœuds pour l'établissement des liens sécurisés, et la possibilité de récupération par un attaquant de la clé maitre qui lui permet de générer toutes les autres clés du réseau.

Exemple de protocoles déterministes : LEAP [104], Pike, Brosk, schéma de Jolly, Kusku et Younis [48].

5.5.2. Les protocoles probabilistes

L'ensemble M des clés utilisées est choisi au hasard à partir d'un ensemble de clés générées aléatoirement où |M| >> |N|. N étant le nombre de nœuds du réseau. Chaque capteur est pré-chargé avec un sous ensembles R de clés de l'ensemble M où |R| < |M|. La probabilité que deux nœuds A et B partagent une clé K parmi les M générées est donnée par :

$$P = \frac{|M|!}{(|M| - m)!} \frac{1}{|M|^m}$$

m est le nombre d'éléments de M.

Pour que la probabilité P soit importante (P > 0.7), on doit augmenter l'espace mémoire réservé aux clés du sous ensemble R.

Les inconvénients sont, entre autres, l'absence d'authentification entre chaque paire de nœuds et le besoin d'augmenter l'espace mémoire réservé aux clés stockées sur le capteur pour les réseaux de grande taille.

Beaucoup de travaux [26] [41] [57] supposent connaitre la position des nœuds voisins (avec une certaine probabilité) pour réduire le nombre de clés stockées en mémoire. D'après l'étude faite par [96], les protocoles probabilistes ont peu d'intérêt par rapport aux protocoles déterministes.

Exemples de protocoles probabilistes : schéma de Eschenauer et gligor [31], schéma de Chan, perrig et Song [23].

5.5.3. Les protocoles géographiques

Ils utilisent la position des nœuds pour augmenter la connectivité du réseau et gérer convenablement et aisément les clés entre nœuds voisins. Ils nécessitent des

composants spéciaux comme le GPS ou des algorithmes de localisation pour localiser les nœuds après le déploiement.

Plusieurs variantes sont développées pour associer la position des nœuds et la création des clés entre eux. Par exemple, on associe la position d'un nœud à un polynôme bi-varié permettant de créer des clés entre deux nœuds voisins [58].

Les inconvénients sont, entre autres, les couts des composants supplémentaires : le GPS et les algorithmes de découverte du voisinage (temps de calcul, espace mémoire, énergie).

Exemples de protocoles géographiques : LBK (Location-Based Keys) [102], LKE (Location-aware Key Establishment).

5.5.4. Les protocoles t-secure

L'objectif est de résister aux attaques visant à compromettre tous les nœuds du réseau. Le principe est qu'un nombre inférieur ou égal à t de nœuds compromis de l'ensemble n des nœuds du réseau ($t < n$) ne permettra pas de compromettre tous le réseau. Ils s'appuient en majorité sur deux schémas cryptographiques particuliers : le schéma de Blom [17] et le schéma de Blundo et al [16].

Pour construire les clés partagées, le schéma de Blom se base sur des matrices spéciales tandis que le schéma de Blundo et al se base sur l'évaluation des polynômes symétriques.

Plusieurs travaux ont combiné le modèle probabiliste avec les schémas de Blom et de Blundo et al pour générer des clés secrètes aux moindres couts de chargement sur les nœuds capteurs.

Les inconvénients sont, entre autres, l'impossibilité des nouveaux nœuds de vérifier l'identité des nœuds précédemment déployés, et dans les cas ou ces derniers sont compromis, ils répondent aux nouveaux nœuds sans possibilité de vérification.

Exemples de protocoles : [13], [19].

5.6. Exemples de schémas de gestion de clés

5.6.1. Schéma de gestion de clés de G. Jolly, M. Kusku, P. Kokate et M. Younis

Jolly et al [48] ont proposé un protocole déterministe de gestion de clés basé sur la méthode de pré-distribution de clés ayant une architecture hiérarchique et reprenant les notions du clustering appliquées au routage de données dans les RcSF. En effet, les nœuds capteurs sont regroupés selon des centres d'intérêt et liés entre eux par des

passerelles appelés « superbe-nœuds » et caractérisées par des ressources énergétiques et des capacités de calcul et de stockage relativement élevées par rapport aux autres nœuds du centre d'intérêts. Dans cette architecture, on utilise également des nœuds de commande responsables des missions de sécurité du réseau et représentent la tierce partie de confiance de tous les nœuds.

On suppose dans ce schéma que les nœuds d'un groupe utilisent un mode de communication direct avec la passerelle lors des opérations de gestion de clés, et que les passerelles utilisent également le même mode et sont atteignables en 1-saut. De plus, les auteurs supposent que les nœuds et les passerelles n'ont aucune connaissance à postériori de la topologie du réseau et sont aléatoirement déployés.

Déroulement du protocole : les fonctions du protocole définissent comment les clés sont distribuées, gérées, révoquées et renouvelées. On tient compte également de l'intégration des nouveaux nœuds capteurs déployés.

- **Distribution des clés :** chaque nœud stocke dans sa mémoire deux clés secrètes : une clé partagée avec la passerelle et une autre clé avec le nœud de commande. Les passerelles partagent des clés entre elles et avec le nœud de commande. On suppose que les passerelles peuvent stocker toutes les clés des nœuds de leurs groupes mais cela est moins sécurisé pour effectuer le stockage de toutes ces clés et les clés partagées avec les autres passerelles. Toutes les clés sont distribuées et chargées sur les capteurs avant le déploiement et aucune action supplémentaire de distribution, immédiate ou après le déploiement, n'est envisagée et un gain d'énergie sur les opérations d'émission / réception de clés est tiré profit.

- **Construction des groupes :** après le déploiement, chaque nœud diffuse un message « Hello » de découverte de voisins contenant son identificateur et l'identificateur de la passerelle qui contient la clé partagée. Chaque passerelle exécute l'algorithme de formation de groupe en se basant sur les clés partagées avec les nœuds capteurs. A la fin de cette étape, chaque nœud reçoit une réponse de la passerelle.

- **Révocation des clés :** si un nœud est compromis, le nœud de commande et la passerelle l'expulsent du groupe en ignorant les routes qui passent par lui. Si une passerelle est compromise, le nœud de commande l'expulse et choisit une autre passerelle pour la remplacer. La nouvelle passerelle reçoit les identificateurs des nœuds de son groupe ainsi que les clés partagées avec eux. Les nœuds de son

nouveau groupe ainsi que les autres passerelles seront informés par ce remplacement en acceptant ses messages et en ignorant les messages de la passerelle compromise.

- **Renouvellement de clés :** le nœud de commande produit les nouvelles clés et les transmet aux passerelles. Chaque passerelle transmet à son tour une clé pour chaque nœud de son groupe.

- **Ajout de nouveaux capteurs :** le nouveau capteur sera pré-chargé avec deux clés secrètes. Le nœud de commande transmet un message contenant l'identificateur et la clé du nouveau capteur à une passerelle sélectionnée au hasard. La passerelle procède à l'intégration du nœud capteur après l'exécution de l'algorithme de reformation de groupes.

5.6.2. LEAP (Localized Encryption and Authentication Protocol)

LEAP [104] est un protocole déterministe de gestion de clés pour les réseaux de capteurs sans fil à la fois localisé et d'authentification qui limite l'impact d'un nœud compromis sur son voisinage. LEAP utilise quatre types de clés sur chaque nœud capteur : une clé individuelle partagée avec la station de base, une clé par-paire partagée avec un autre nœud capteur, une clé du groupe partagée avec des nœuds voisins du groupe et une clé globale partagée avec tous les nœuds du réseau. Le choix porté sur ces quatre clés permet de minimiser l'implication de la station de base dans le processus de gestion de clés, ce qui réduit en conséquence le trafic et la consommation d'énergie.

Déroulement du protocole : les auteurs de LEAP supposent que pour compromettre un nœud capteur, un attaquant nécessite un temps minimal T_{min} qu'il faut pour copier le contenu de la mémoire du nœud compromis. LEAP essaye d'exploiter ce temps pour permettre à deux nœuds voisins d'établir une clé symétrique, à partir de la clé K_{in} pré-chargée sur chaque capteur avant le déploiement, et de supprimer cette dernière de la mémoire du nœud compromis en un temps $T < T_{min}$.

- **Pré-distribution des clés :** chaque nœud C dérive une clé principale Kc à partir de la clé K_{in} en utilisant une fonction pseudo-aléatoire *f* comme suit :

$Kc = f_{kin}(c)$.

- **Découverte du voisinage :** après le déploiement, chaque nœud capteur diffuse un message « Hello » avec son identificateur Id_c afin de découvrir ses voisins, et initialise un Timer qui se déclenche après un temps T_{min}. Chacun des voisins du

nœud C répond par un message ACK contenant son identificateur Id_v et sa clé Kv authentifiée par le nœud C comme suit : $Kv = f_{kin}(v)$.

- **Calcul de la clé par paire :** le nœud C calcule sa clé par paire avec son voisin V de cette façon : $Kcv = f_{kv}(c)$.De même, le nœud V peut calculer sa clé par paire avec le nœud C de la même manière, c'est-à-dire : $Kcv = f_{kc}(v)$.

- **Révocation des clés :** lorsque le timer expire après le temps T_{min}, le nœud C efface la clé k_{in} et toutes les clés principales de ses voisins de sa mémoire, sans toutefois effacer sa clé principale Kc.

- **Ajout de nouveaux nœuds :** les nœuds nouvellement déployés dérivent leurs clés principales et leurs clés par paire partagées avec leurs voisins avec la même clé initiale K_{in}. Les autres étapes restent inchangées.

Sécurité du protocole : A la fin de toutes les phases de déroulement du protocole, tous les nœuds du réseau auront établis des clés principales et des clés par paire utilisées avec leurs voisins. De plus, aucun nœud du réseau ne possède la clé initiale K_{in}. Ainsi, un attaquant peut toutefois écouter le réseau sans pouvoir injecter des informations erronées ou déchiffrer les messages échangés entre les nœuds. Même si un attaquant réussisse à compromettre un nœud après un temps Tmin, il n'obtient que sa clé principale, et quand il sera détecté, ses voisins suppriment les clés partagées avec lui.

Cependant, le problème de ce schéma est la possibilité de lancement d'attaques de consommation de ressources par des adversaires qui peuvent exploiter les messages « Hello » non authentifiés et d'injecter un grand nombre de messages de ce type, synonyme de communications inutiles et consommatrices de ressources. Parmi les solutions proposées afin de corriger la faille, on retient: pré-charger le nouveau nœud avec la clé globale courante et l'utiliser pour authentifier les messages « Hello ». Cette solution n'est pas trop efficace car un nœud attaquant peut récupérer la clé globale en compromettant n'importe quel nœud du réseau.

5.6.3. Schéma de Eschenauer et Gligor

Eschenauer et Gligor [31] ont proposé un schéma de gestion de clés basé sur le principe de la clé probabiliste partagées entre les nœuds d'un groupe aléatoire. L'idée de base de ce schéma est de distribuer aléatoirement un certain nombre de clés à chaque capteur. Deux capteurs A et B peuvent établir une communication sécurisée et échangent des messages s'ils partagent une même clé, c'est-à-dire $\exists$ Ka $\in$ K(A) et $\exists$ Kb $\in$ K(B) tel que Ka = Kb.

Déroulement du protocole

- **Pré-distribution de clés :** un sous ensemble de *m* clés est choisi au hasard pour chaque nœud parmi un ensemble *P* de clés générées tel que | m | < | P | et de telle manière que deux sous ensembles aléatoires *m1* et *m2* de *P* auront une certaine probabilité d'avoir en commun une clé.

- **Découverte de clés partagées :** après le déploiement, les nœuds découvrent leurs voisins en diffusant un message « Hello » et les nœuds qui répondent par un message ACK sont destinataires de la liste des identités K_{id} des clés possédées. Chacun des nœuds voisins à un nœud compare la liste des identités des clés reçues à sa liste de clés et si une clé au minimum est commune entre les deux listes, elle devient une clé de session entre les deux nœuds voisins. A la fin de cette phase, un graphe connecté formé de liens sécurisés est construit. Une phase d'établissement de chemin de clés serait alors lancée.

- **Révocation de clés :** un nœud contrôleur annonce la liste de clés à supprimer de la mémoire de tous les nœuds du réseau possédant ces clés. Cette liste de clés à révoquer appartient à un nœud compromis détecté par le nœud contrôleur. Afin d'authentifier cette liste et d'empêcher des adversaires de générer des listes de clés à supprimer de la mémoire des nœuds du réseau, le nœud contrôleur génère une clé K_e pour chiffrer le message envoyé, avec la liste de clés, aux nœuds concernés qui le déchiffre avec la même clé K_e partagées avec lui.

- **Renouvellement de clés :** si des clés expirent après un certain temps, elles seront supprimées par le nœud lui-même. Les liens cassés sont réparés au lancement de la nouvelle phase de découverte de clés et d'établissement de chemins sécurisés.

Les limites de ce schéma, entres autres la nature des liens et l'absence d'authentification symétrique, sont corrigées par le schéma de Chan, Perrig et Song que nous allons étudier dans la section suivante. Les corrections portent essentiellement sur les mécanismes de renforcement de liens et d'authentification nœud à nœud.

5.6.4. Schéma de Chan, Perrig et Song

Le schéma proposé par Chan, Perrig et Song [23] est un ensemble d'amélioration au schéma de Eschenauer et gligor et porte sur les corrections suivantes : i) un nœud doit partager *q* clés ($q > 1$) avec un autre nœud pour établir un chemin de clés sécurisé, ii) on doit changer les clés partagées K par des valeurs aléatoires K' afin de renforcer la sécurité des liens, iii) introduire un protocole d'authentification nœud à nœud.

5.6.4.1. Schéma des clés q-composite

Une paire de nœuds (A, B) doit partager *q* clés, au lieu d'une seule, pour établir un lien sécurisé telle que (K_{A1}, K_{A2},…, $K_{Aq} \in m_a$) soit identique à (K_{B1}, K_{B2},…, $K_{Bq} \in m_b$). La nouvelle clé extraite de ces deux ensembles est le hash de clés communes tel que $K = \text{hash}(K_1, K_2, \ldots, K_q)$.

5.6.4.2. Renforcement de la sécurité des liens

L'objectif de renforcer la clé utilisé pour la sécurité d'un lien par chemins multiples, une technique explorée pour la première fois par Anderson et Perrig [5] au début des années 2000, est d'empêcher un nœud attaquant d'exploiter une liste de clés récupérée sur un nœud compromis pour compromettre la sécurité d'un lien entre deux nœuds utilisant une clé appartenant à cette liste confisquée. En effet, si un attaquant compromet un nœud C et récupère sa liste de clés $m_c = \{K_1, K_2,\ldots, K_m\}$, il se peut que deux nœuds A et B établissent un lien en utilisant une clé $K_{AB} \in m_c$. Dans ce cas, l'attaquant utilise la clé K_{AB} pour déchiffrer les messages échangés entre les nœuds A et B. Pour éviter un tel scénario, les auteurs suggèrent que le nœud A connait les différentes routes qui mènent vers le nœud B avec un certain nombre de sauts *h* et les utilise pour envoyer *j* valeurs aléatoires v_1, v_2,…, v_j sur les *j* routes qui mènent vers B. La nouvelle clé K' sera calculée par le nœud B à partir des *j* valeurs reçues comme suit : $K' = K \Phi v_1 \Phi v_2 \Phi \ldots \Phi v_j$. Par conséquent, plus le nombre de chemin entre deux nœuds augmente, plus le degré de sécurité entre eux augmente.

Cependant, le risque qu'un attaquant puisse écouter et déchiffrer les messages circulant sur un chemin augmente avec sa longueur. Il suffit qu'un seul lien de ce chemin soit non sécurisé pour que le chemin entier soit non sécurisé. Pour y remédier, les auteurs de ce schéma se sont intéressés aux chemins multiples à 2-sauts pour le renforcement des clés de chemin.

5.6.4.3. Authentification nœud à nœud

Logiquement, deux nœuds du réseau définissent un lien sécurisé et renforcé s'ils partagent *q* clés secrètes et sont tous les deux des nœuds légitimes. Cette dernière condition est difficile à vérifier avec les deux mécanismes cités ci-dessus car rien ne laisse présupposer que le nœud A ne communique pas en réalité avec un nœud C et partage avec lui le même sous ensemble de clés K partagé avec le nœud B, ce qui pousse les auteurs à définir la propriété suivante : « un protocole a

la propriété d'authentification nœud à nœud si n'importe quel nœud peut assurer l'identité des nœuds avec lesquels il communique ». Cette propriété soutient les solutions de sécurité suivantes : i) éviter de communiquer avec des nœuds compromis en s'assurant de leur identité, ii) permettre aux différents nœuds capteurs de résister aux attaques de réplique en maintenant les identités des nœuds déjà insérés et en rejetant les liaisons additionnelles établies par leurs identités, iii) exécuter des actions, d'une façon autonome et loin de la station de base, par les nœuds en rejetant les liaisons avec les nœuds compromis et en révoquant les clés subtilisées par les attaquants, ceci améliore le temps de réaction aux intrusions détectées par le réseau.

5.6.5. LBK (Location-based Keys)

Liu et Ning [54] ont proposé un protocole de gestion de clé qui se base sur les emplacements géographiques des nœuds capteurs appelé Location-based Keys. Les auteurs de ce schéma considèrent trois hypothèses comme pré-requis : i) tous les nœuds du réseau sont statiques, ii) de nouveaux nœuds peuvent être ajoutés au réseau à tout moment, iii) il est souvent possible de déterminer approximativement les positions des nœuds capteurs.

Les nœuds peuvent alors utilisés leurs informations sur leur emplacement pour partager des clés par paire avec leurs voisins. Le principe de ce schéma est le suivant : avant le déploiement, chaque nœud est pré-chargé avec une coordonnée (x,y) pour définir son endroit prévu sur un secteur bidimensionnel appelé le champ cible. Après le déploiement, les nœuds sont placés dans des endroits réels pouvant être différents des endroits prévus, cette différence est dite l'erreur de déploiement.

Déroulement du protocole

- **Pré-distribution des clés :** les nœuds sont pré-chargés par la station de base avec des clés par paire suivant les coordonnées choisies pour les endroits prévus (on suppose que les deux nœuds A et B possèdent une forte probabilité d'apparaitre dans un même endroit réel), cette probabilité dépend de l'erreur de déploiement. Plus précisément, pour chaque nœud capteur *u*, la station de base découvre l'ensemble S de ses voisins les plus proches de son endroit prévu. Pour chaque nœud $v \in S$, la station de base génère une clé aléatoire unique K*uv* si aucune autre clé n'est assignée à ce couple (*u*, *v*), et distribue ensuite la clé K*uv* aux nœuds capteurs *u* et *v*.

- **Découverte de clés :** après le déploiement, si deux nœuds capteurs *u* et *v* souhaitent établir une communication, il suffit de vérifier s'ils partagent une clé pré-distribuée. Cette information est obtenue via la station de base.

- **Etablissement de liens :** après le déploiement, si les deux nœuds *u* et *v* ne partagent aucune clé pré-distribuée (l'erreur de déploiement est jugée importante), ils peuvent s'associer à un voisin intermédiaire qui partage des clés par paire avec chacun d'eux pour établir une clé de session. Pour cela, l'un des deux nœuds (supposons le nœud *u*) diffuse un message Broadcast avec son identificateur et l'identificateur du nœud *v*, le nœud *y* intercepte le message, et vérifie s'il partage une clé K*uy* avec *u* et une clé K*yv* avec le nœud *v*. Si oui, le nœud *y* transmet au nœud *u* un message contenant les informations E(K*uy*(K)) et E(K*yv*(K)). La clé de session est calculée à partir des clés K*uy* et K*yv*. Le nœud *u* transmet à son tour un message contenant l'information E(K*yv*(K)) au nœud *v*. A noter que le nœud *u* peut recevoir plusieurs réponses à sa requête, il ne choisi qu'une seule d'entre elles.

6. Conclusion

Ce chapitre nous a permis d'étudier la cryptographie et la gestion de clés dans les RcSF. Nous avons déduit que la cryptographie à clé symétrique est la plus appropriée aux réseaux de capteurs sans fil pour sa simplicité et l'aisance de sa mise en œuvre. Bien que la cryptographie symétrique respecte les contraintes des capteurs, contrairement à la cryptographie asymétrique, la fonction de gestion de clés entre les entités du réseau demeure un problème critique et difficile à résoudre sans recourir à la méthode de pré-distribution de clés, la plus appropriée pour les RcSF pour son cout faible. Nous avons présenté néanmoins quelques schémas de gestion de clés utilisant les méthodes de pré-distribution de clés. Malgré que plusieurs solutions de ces schémas paraissent prometteuses, il existe encore certains défis à relever qui nécessitent une prise en considération par les solutions de sécurité.

Dans le chapitre suivant, nous allons présenter nos solutions proposées pour la sécurité du routage et la gestion de clés dans les RcSF.

CHAPITRE 05
CONTRIBUTION ET IMPLÉMENTATION

Nous avons étudié dans les chapitres précédents les notions fondamentales des réseaux de capteurs sans fil (RcSF) et principalement les notions de routage, d'énergie et de la sécurité des échanges de données. Parmi toutes les solutions proposées pour l'acheminement sécurisé d'informations à un cout minimum en consommation énergétique et en capacités de calcul, nous avons choisi de travailler sur une topologie hiérarchique à deux niveaux et un système de sécurité basé sur la gestion de clés. Les deux choix sont des choix efficaces et économes en énergie. Les sections suivantes décrivent les fondements et les principes du protocole proposé.

1. Introduction

Les réseaux de capteurs sans fil (RcSF) sont souvent des réseaux denses déployés dans des zones à risques pour des applications sensibles où l'information récoltée est d'une importance capitale pour l'utilisateur final du réseau. Les réseaux denses nécessitent l'échange d'un grand nombre de messages lors des phases de construction et de maintenance, synonyme d'une consommation conséquente d'énergie. Ajoutons à cela, les conditions de déploiement et la nature de l'information captée exige un niveau de sécurité important pour contrer les différents types d'attaques sur la topologie du réseau et le routage de données tout en respectant les contraintes des RcSF et le maintien de leurs performances. Notre contribution s'inscrit dans ce contexte de protection des communications et du routage de données dans les RcSF en développant deux solutions pour le routage et la sécurité des échanges, la première étant basée sur une architecture hiérarchique à deux niveaux et la seconde sur un schéma de gestion de clés hybride, simple et efficace, introduisant un système de cryptage asymétrique pour l'échange de clés privées entre les nœuds capteurs.

En effet, la première solution proposée est une solution de routage qui permet de minimiser le nombre de messages diffusés en Broadcast par les nœuds dans les phases de construction et de maintenance de la topologie du réseau. Ce gain en messages échangés implique un gain sur la bande passante (minimiser la perte de données due aux collisions) et sur l'énergie consommée (maximiser la durée de vie du réseau). L'autre solution est un schéma sécurisé de gestion de clés avec des algorithmes cryptographiques qui assure à la fois l'authentification, l'intégrité, la confidentialité et la non-répudiation des données transmises. Elle inclut un système hybride d'échange de clés fondé sur la technique de la cryptographie

sur les courbes elliptiques (ECC), une technique que nous allons étudier en détail dans les sections suivantes de ce chapitre.

L'utilisation de la cryptographie sur les courbes elliptiques (ECC) est motivée par la taille réduite des clés pour une sécurité égale ou supérieure aux autres méthodes de chiffrement asymétrique et d'échange de clés existantes. Le problème sur lequel se base les ECC est le problème du logarithme discret, intéressant pour la cryptographie dès qu'il devient exponentiel pour un attaquant.

Le choix d'une architecture hiérarchique sur deux niveaux permet de gagner en bande passante et en énergie consommée tandis qu'un système de gestion de clés hybride garantit un échange de clés sécurisé entre chaque paire de nœuds communicants et assure également la protection du réseau contre la capture des nœuds en introduisant des mécanismes de génération, de distribution, de révocation et de renouvellement de clés.

Parmi les motivations qui nous ont poussés à travailler sur une architecture en clusters où les communications et le routage de données sont sécurisés par un schéma de gestion de clés, on cite :

- Le protocole utilise une architecture à deux niveaux : les nœuds membres d'un cluster transmettent leurs données vers le Cluster Head (CH), le CH les agrège et transmis le résultat au nœud puits.
- Durant la phase de construction de la topologie en clusters, le nombre de messages échangés est réduit.
- La protection des clés est égale à la protection du réseau entier.
- La gestion des clés constitue la base des systèmes cryptographiques.
- Le problème de gestion de clés est l'un des problèmes les plus compliqués dans la sécurisation d'un réseau.
- Utilisation d'un système asymétrique pour la génération et l'échange de clés : sûr et produit des clés de petite taille pour les capacités physiques des RcSF.
- Utilisation de la cryptographie symétrique pour sécuriser les communications et le routage : simple, rapide et performant.
- Un bon schéma de gestion de clés permet souvent d'éviter la compromission, la capture, l'analyse du trafic, l'atteinte à l'intégrité et à la fraicheur des données.

2. Notions mathématiques

Avant de commencer à étudier des travaux antérieurs sur l'utilisation des outils de la cryptographie, notamment le chiffrement et l'échange de clés par les ECC, et avant de présenter notre schéma de gestion de clé qui se base justement sur ces outils, nous allons commencer d'abord par donner les principales notions mathématiques sur lesquelles se basent l'ensemble de ces outils.

2.1. Complexité : la théorie de la complexité fournit une méthodologie d'analyse de degré de complexité de calcul des algorithmes et des techniques de cryptographie. En outre, elle détermine le temps de traitement minimal et l'espace de stockage nécessaire pour résoudre un problème sur une machine théorique dite machine de Turing. Selon leur complexité, les problèmes peuvent être résolus en un temps polynomial (raisonnable) avec des algorithmes polynomiaux. Ces problèmes sont dits *solubles* et appartiennent aux problèmes de la classe P. Par contre, les problèmes qui ne peuvent pas être résolus en un temps polynomial sont dits *non solubles* car calculer leurs solutions devient rapidement impossible. Ce sont souvent des problèmes résolus avec des algorithmes exponentiels et appartiennent aux problèmes de la classe NP. Le tableau (5.1) donne, en moyenne, les temps nécessaires pour casser un algorithme selon sa classe de complexité.

Classe	**Complexité**	**Nombre d'opérations pour n = 10^6**	**Temps nécessaire pour 10^6 opérations / seconde**
Linéaire	O (n)	10^6	1 seconde
Quadratique	O (n^2)	10^{12}	11,6 jours
Cubique	O (n^3)	10^{18}	32 000 années
Exponentiel	O (2^n)	10^{301030}	10^{301006} fois l'âge de l'univers

Tableau 5.1 : Temps nécessaire pour casser un algorithme selon sa complexité

2.2. Logarithme discret : le problème du logarithme discret est une application réciproque de l'exponentiation qui consiste à retrouver un entier λ tel que $h = g^{\lambda} \bmod p$ (h est un élément appartenant à un groupe multiplicatif G d'un corps fini F d'ordre n, g est un générateur du groupe G, p est un grand nombre premier et λ est le plus petit entier satisfaisant l'équation $h = g^{\lambda} \bmod p$ avec ($0 \leq \lambda < n$). λ est le logarithme discret de

l'élément h du groupe G. on notera l'exponentiation de g en λ par [λ] g = g x g x … x g (λ fois). Le principe est exprimé par la facilité de calculer h connaissant g, p et λ et par l'impossibilité (en un temps raisonnable) de déterminer λ même si on connait h, p et g car il revient à calculer $\lambda = Log_g (h) \ mod \ p$, c'est-à-dire calculer le logarithme discret en base g (λ). La complexité de calcul de λ est de O (√|G| log |G|) en temps et de O (√|G|) en mémoire. Le temps de calcul de la solution n'est pas polynomial alors le problème du logarithme discret appartient aux problèmes de la classe NP. Les parties communicantes utilisent alors des *fonctions à sens unique*, faciles à calculer et difficile à résoudre, qui se basent sur un secret dit la trappe pour réduire les temps de calculs et l'espace de stockage dans la résolution du problème. Dans les systèmes cryptographiques, la trappe représente la clé secrète.

2.3. Les courbes elliptiques : nous donnons ici une brève définition des courbes elliptiques en s'appuyant principalement sur l'ouvrage 'Guide to Elliptic Curve Cryptographie' [40] et sur le cours de l'arithmétique des courbes elliptiques [70]. Les courbes elliptiques sont des objets mathématiques très complexes et très riche en même temps. Elles interviennent en cryptographie dans le problème de la factorisation des entiers et la génération des secrets communs entre deux entités communicantes. Elles sont représentées par des courbes algébriques non-singulières de degré 3 ou 4, autrement dit, l'ensemble des couples (x, y) ∈ K^2 vérifiant l'équation $y^2 = f(x)$ où le degré de la fonction f est 3 ou 4. Exemple : $y^2 = x^3 - 18x + 25$.

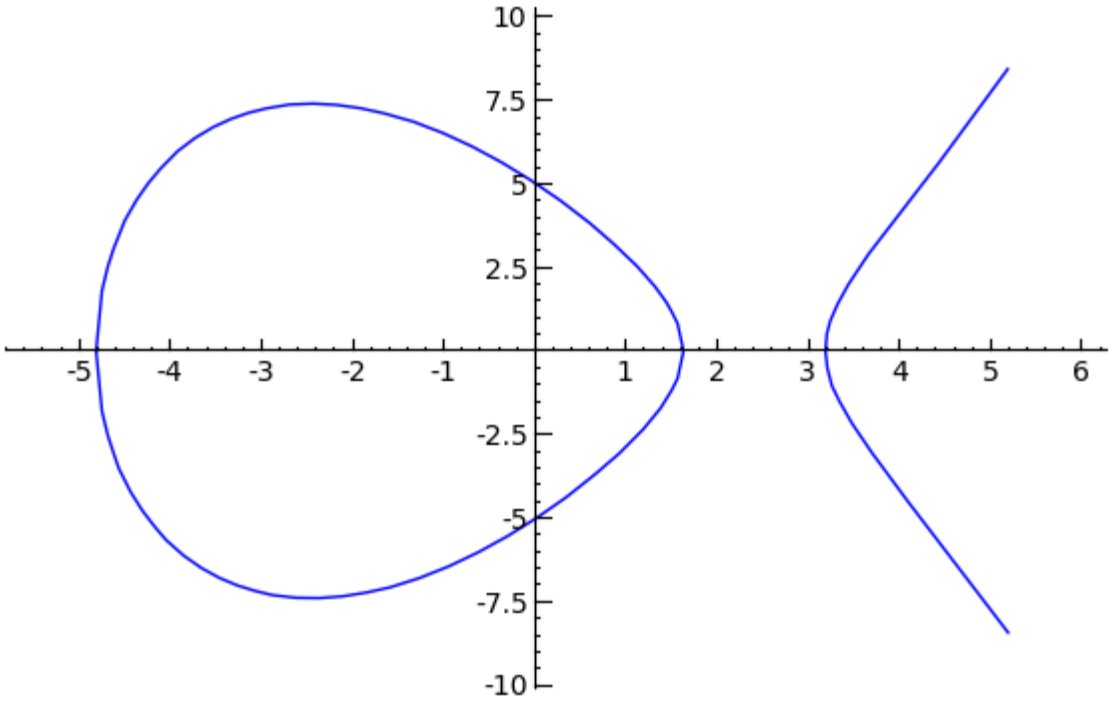

Figure 5.1 : Courbe elliptique $y^2 = x^3 - 18x + 25$ sur R [47]

2.4. Le logarithme discret sur les courbes elliptiques : de manière analogue au problème du logarithme discret sur un groupe multiplicatif, on peut définir le problème du logarithme discret sur les courbes elliptiques. Etant donné une courbe elliptique E définie sur un corps fini F_q, un point $P \in E (F_q)$ d'ordre n et un point $Q \in \{P\}$, le problème est de retrouver l'entier $k \in [0, n-1]$ tel que $Q = [k] P$. l'ordre n du point P est donné par : $\{[i] P ; 0 \leq i < n\}$. plusieurs méthodes sont proposés pour effectuer la multiplication scalaire d'un point ([k] P), on cite : l'algorithme Double_and_add, la méthode NAF (Non-Adjacent Form)…etc.

A la lumière de cette petite introduction au principe du logarithme discret sur les courbes elliptiques qui nous a permis d'avoir une vision sur la complexité théorique des algorithmes utilisant ce principe, nous avons décider de l'intégrer dans notre solution afin de permettre un échange de clés symétriques sécurisé et à un cout raisonnable entre chaque paire de nœuds et d'utiliser les clés générées pour sécuriser les communications et le routage de données sur le réseau.

3. Travaux antérieurs

Cette section présente quelques travaux antérieurs sur la cryptographie qui permettent d'assurer la confidentialité, l'intégrité, l'authentification, la signature et la fraicheur des données.

3.1. AES (Advanced Encryption Standard) : c'est une méthode de chiffrement par blocs de 128, 192 et de 256 bits [74]. Chaque bloc est transformé en une matrice de 16 (4x4) octets à quoi on applique les opérations suivantes pour atteindre un texte chiffré à partir du texte en clair (figure 5.2) :

- Sub_Byte : substitution de chaque élément du texte en clair.
- Shift_Rows : décalage des lignes en fonction de leurs numéros.
- Mix_Columns : multiplication de chaque colonne par un polynôme pour obtenir une transformation linéaire.
- Add_Round_Key : application de l'opérateur mathématique XOR entre la matrice transformé et une sous partie de la clé.

L'avantage de l'algorithme AES est sa rapidité et son économie en termes d'énergie et de calcul. L'inconvénient concerne la façon d'échanger la clé de chiffrement.

3.2. RC4 (Rivest Cipher 4) : crée en 1987 par Ron Rivest, RC4 [34] est considéré comme l'algorithme le plus utilisé dans sa catégorie d'algorithmes de chiffrement par flot. Son fonctionnement est le suivant :

- La clé K utilisée est d'une longueur variable. Elle peut avoir une taille de 8 à 2048 bits (comprise entre 1 et 256 octets).
- La clé est utilisée pour initialiser un vecteur S de 256 octets. Initialement, les cellules du vecteur S reçoivent des valeurs égales à leurs positions c'est-à-dire : S[0]= 0, S[1] = 1, …., S[255] = 255.
- On crée un vecteur temporaire T de longueur égale à S destiné à recevoir la clé K et utilisé pour produire la permutation initiale de S.
- Pour la génération des flux, la clé K n'est pas utilisée. Pour chaque S[i], on procède à un échange de contenu avec S[j]. Ensuite, on calcule un entier t = (S[i] + S[j]) mod 256 nécessaire pour déduire la clé K = S[t].
- La valeur de la clé K est utilisée pour les opérations de chiffrement et de déchiffrement. Le chiffrement est obtenu en réalisant une opération XOR entre un octet de la clé K et du message en clair. Le déchiffrement est obtenu en réalisant un XOR entre un octet de la même clé K et du texte chiffré.

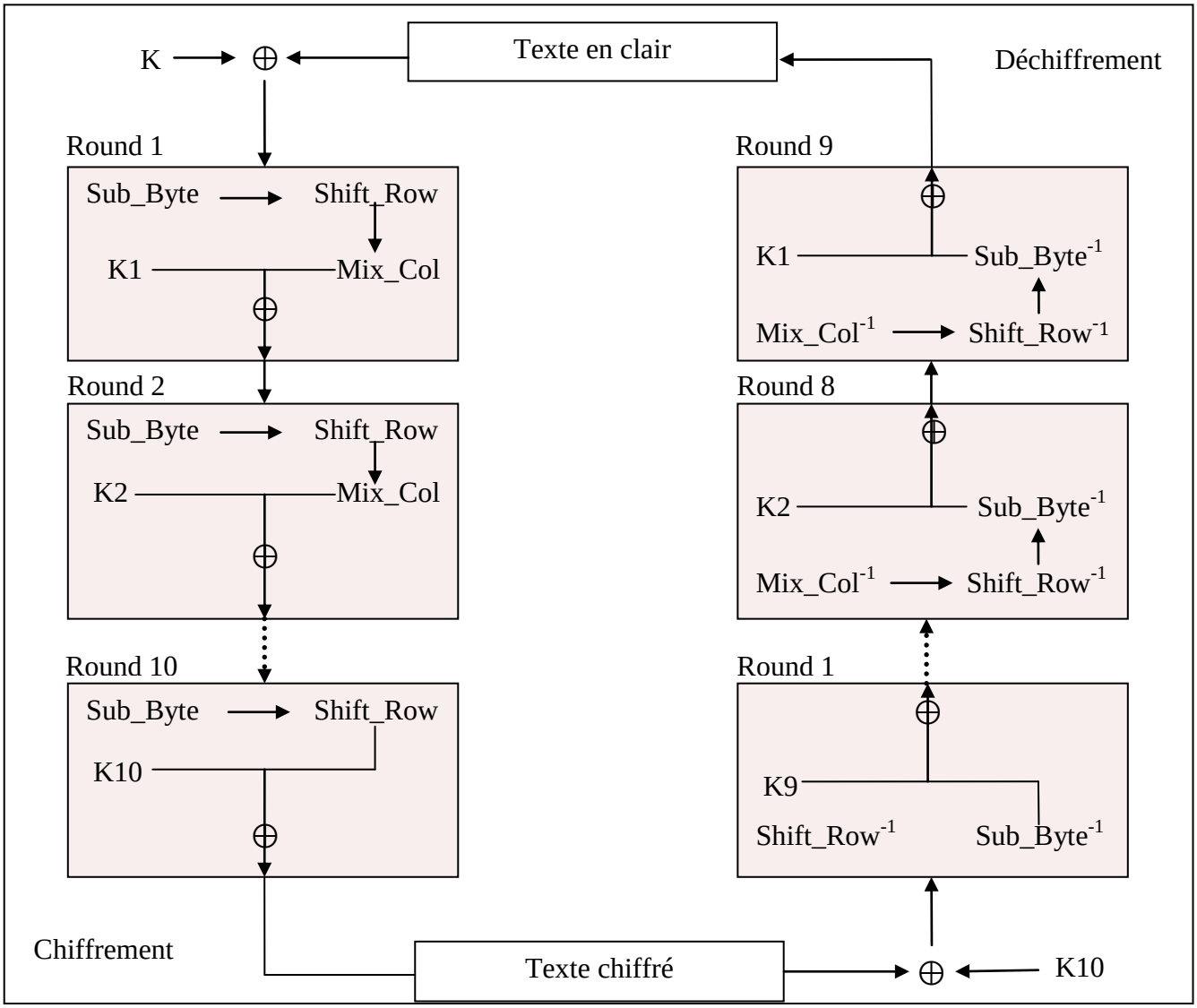

Figure 5.2 : Schéma général de l'algorithme AES

Une comparaison des deux algorithmes de chiffrement AES et RC4 est donné par le tableau (5.2) [90] ci-dessous :

Algorithme de chiffrement	**Taille des clés (en bits)**	**Complexité**	**Mode de chiffrement**	**Niveau de sécurité**	**Rapidité d'exécution**
AES	128, 192 et 256	Nouveau et complexe	Chiffrement par bloc	Très sécurisé	Moins rapide
RC4	Entre 8 et 2048	Vieux et simple	Chiffrement par flot	Moins sécurisé	Plus rapide

Tableau 5.2 : Comparaison des algorithmes de chiffrement AES et RC4 [89]

3.3. RSA (Rivest Shamir Adelman) : inventé en 1977 par Ron Rivest, Adi Shamir et Len Adelman, le système de cryptage RSA [109] est un système à clé publique qui devint rapidement une référence. Son principe de fonctionnement est le suivant :

- A génère deux grands nombres premiers p et q par des algorithmes probabilistes.
- A calcule (n = p * q) et φ (n) = φ (p, q) = (p-1) (q-1) et détermine e tel que e ^ φ (n) = 1, c'est-à-dire PGCD (e, φ(n)) = 1.
- A détermine d tel que ed ≡ 1 mod (φ(n)).
- A envoi le couple (e, n) à B. (e, n) est la clé publique de A.
- Le couple (d, n) est la clé privée de A.
- B chiffre son message M en utilisant la clé publique (e, n) de A et obtient le message chiffré C calculé comme suit : $C = M^e \bmod n$, et le transmet à A.
- A déchiffre le message C et obtient le message original en exécutant l'opération suivante : $M = C^d \bmod n$.

L'atout majeur de RSA est le niveau de sécurité élevé qu'il garantit car il est facile pour deux communicants A et B de créer un grand nombre premier à partir du produit de deux autres nombres premiers et il est difficile, pour des nombres premiers assez grands, à un attaquant de retrouver la factorisation en deux nombres premiers pour le grand nombre premier généré. Ses inconvénients majeurs sont son temps de calcul important et la taille de ses clés (au minimum 1024 bits).

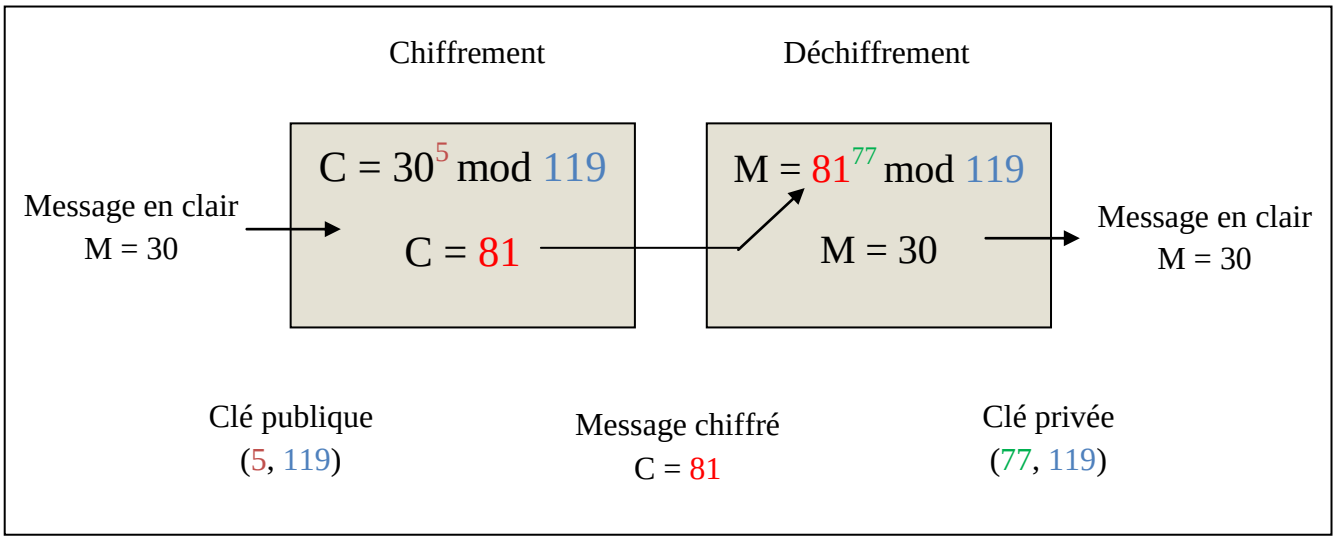

Figure 5.3 : Exemple de déroulement de l'algorithme RSA

3.4. Diffie-Hellman : du nom de ses auteurs Whitfield Diffie et Martin Hellman, Diffie-Hellman [91] est un algorithme asymétrique très simple pour l'échange de clés entre deux nœuds communicants A et B sans qu'un troisième puissent calculer ce secret même s'il possède toutes les informations utilisées par A et B. Son principe est :

- A et B choisissent publiquement deux grands nombres premiers p et g tel que $1 < g < p$. p doit être suffisamment grand (512 ou 1024 bits) pour garantir un niveau de sécurité élevé.
- A choisit secrètement $a < p$ et calcule $X = g^a \bmod p$.
- B choisit aléatoirement $b < p$ et calcule $Y = g^b \bmod p$.
- A transmet X à B et b transmet Y à A. Les transmissions se font sur un canal non sécurisé.
- A calcule sa clé privée $K_{(a,b)} = g^{ab} \bmod p = Y^a \bmod p$.
- B calcule sa clé privée $K_{(a,b)} = g^{ab} \bmod p = X^b \bmod p$.
- A et B partagent une clé secrète symétrique $K_{(a, b).}$

3.5. ECC (Elliptic Curve Cryptography) : c'est une méthode d'échange de clés proposée indépendamment par Miller [69] en 1985 et Koblitz [53] en 1987. Ce système cryptographique se base sur le modèle asymétrique et permet aussi bien de chiffrer et de signer. Les clés utilisées sont plus courtes pour une sécurité égale ou supérieure aux autres systèmes de chiffrement asymétrique (RSA, DH, …etc.). Une comparaison de ces trois systèmes asymétriques (ECC, RSA, DH) est donnée par le tableau (5.3) [30]. Le principe de la cryptographie sur les courbes elliptiques est le suivant :

- Repose sur le problème du logarithme discret pour les courbes elliptiques.
- A et B choisissent publiquement n (n est le plus petit entier positif sur une courbe elliptique tel que $nQ = O$ (Q est un point de la courbe et O est le point identité de la courbe: $Q + O = Q$)).
- A choisit $a < n$ qui sera considéré comme sa clé secrète.
- B choisit sa clé secrète $b < n$.
- A génère $P_A = [a]\ G$ et B génère $P_B = [b]\ G$. (G est un autre point de la courbe elliptique).
- A et B échangent publiquement P_A et P_B.
- A calcule sa clé privée $K_{(a, b)} = a\ P_B$.

- B calcule sa clé privée $K_{(a, b)} = b\ P_A$.
- A et B partagent la même clé privée $K_{(a, b)}$.

Système de cryptage	Symétrique	ECC	DH / RSA
Taille des clés (en bits)	80	163	1024
	128	283	3072
	192	409	7680
	256	571	15360

Tableau 5.3 : Comparaisons des tailles des clés pour un niveau de sécurité équivalent [30]

3.6. MD5 (Message Digest 5) : Conçu par Ronald Rivest [86] en 1991, c'est une fonction de hachage cryptographique et c'est le dernier d'une série d'algorithmes Message Digest (MD2, MD4, …etc.). L'empreinte est calculée comme suit :

- Ajouter des padding, si nécessaire, pour atteindre un message T d'une longueur de 448 bits.
- Ajouter la longueur réelle du message à émettre (64 bits) après les 448 bits pour atteindre un message M de longueur de 512 bits. Si la longueur du message à émettre dépasse 64 bits, on ne prendra en compte que les bits de poids faible.
- Initialiser quatre buffers de 32 bits chacun (A, B, C, D) qui constituent l'IV (vecteur initial).
- Concaténer les résultats des additions des registres A, B, C, D de l'IV avec la valeur de la variable chainée obtenue par la manipulation du i^{eme} bloc pour avoir le résultat final.
- Le condensé obtenu est de 128 bits.

3.7. SHA-1 (Secure Hash Algorithm) : SHA-1 [31] est l'algorithme le plus utilisé dans la famille des algorithmes SHA (SHA1, SHA256, SHA384, SHA512, …etc.). SHA-1 est une fonction de hachage qui prend en entrée un texte en clair de taille inférieure à 2^{64} bits et produit en sortie un condensé de 160 bits comme suit :

- Le message à condenser doit être de longueur égale à 448 bits.
- La longueur à ajouter doit être de taille égale à 64 bits afin d'atteindre une taille totale du message égale à 512 bits.
- On utilise 05 buffers de 32 bits (A, B, C, D, E).
- On exécute quatre rondes sur vingt itérations chacun. Les rondes sont similaires mais utilisent des fonctions primitives différentes.
- Le condensé final est le condensé attendu.

Le tableau (5.4) donne une comparaison entre les deux fonctions de hachage MD5 et SHA-1 :

Fonction de hachage	**Taille de l'empreinte (en bits)**	**Taille des unités de traitement (en bits)**	**Nombre d'opérations**	**Longueur maximale du message M en bits)**
MD5	128	512	64 (4 rondes de 16 itérations)	∞
SHA-1	160	512	80 (4 rondes de 20 itérations)	$2^{64} - 1$

Tableau 5.4 : Comparaisons des fonctions de hachage MD et SHA-1

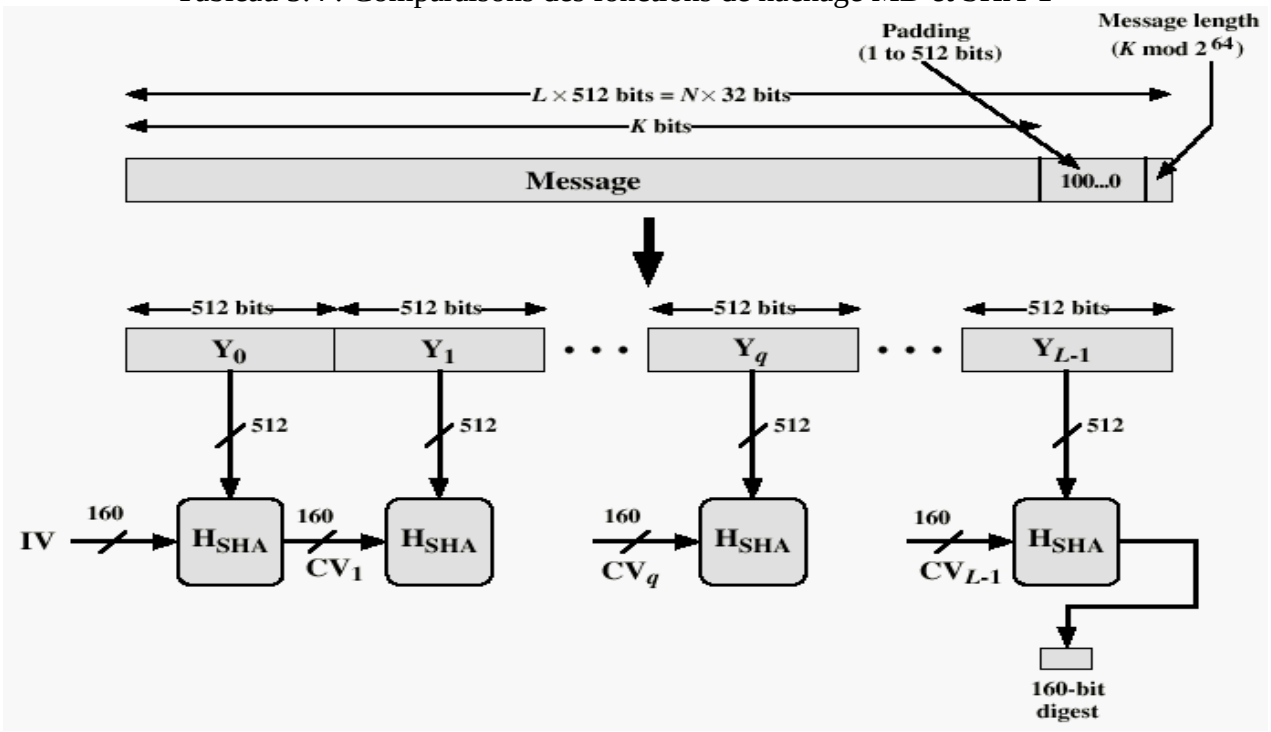

Figure 5.4 : Principe de l'algorithme SHA-1

3.8. HMAC (keyed-Hash Message Authentication Code) : La construction et l'analyse des HMACs [49] ont été publiées pour la première fois en 1996 par Mihir Bellare, Ran Canetti, et Hugo Krawczyk. Le but est de créer des messages d'authentification MAC en utilisant des fonctions de hachage en combinaison avec une clé cryptographique. Les HMACs sont utilisés pour vérifier simultanément l'intégrité et l'authenticité d'un message. N'importe quelle fonction de hachage peut être utilisée dans le calcul du HMAC et le nom de l'algorithme résultant est une combinaison des deux, comme : HMAC-MD5 ou HMAC-SHA256…etc. le HMAC est calculé comme suit :

- Ajouter des zéros à gauche de la clé K pour créer un flux de b bits (K+).
- Produire S_i = K+ (XOR) ipad en b bits.
- Concaténer le message M à S_i (M || S_i).
- Appliquer une fonction de hachage H au flux obtenu : H (M || S_i).
- Produire S_0 = K+ (XOR) opad en b bits.
- Concaténer S_0 avec H (M || S_i) : (S_0 || H (M || S_i)).
- Appliquer une fonction de hachage H au flux obtenu : H (S_0 || H (M || S_i)).
- Le code ainsi obtenu est : $HMAC_K$ (M) = H [(K⊕opad) || H [(K⊕ipad) || M]].
- Avec : b : le nombre de bits d'un bloc du message M.

  K : la clé qui permet le calcul du MAC.

  Ipad : 0x36 (répété b/8 fois).

  Opad : 0x5C (répété b/8 fois).

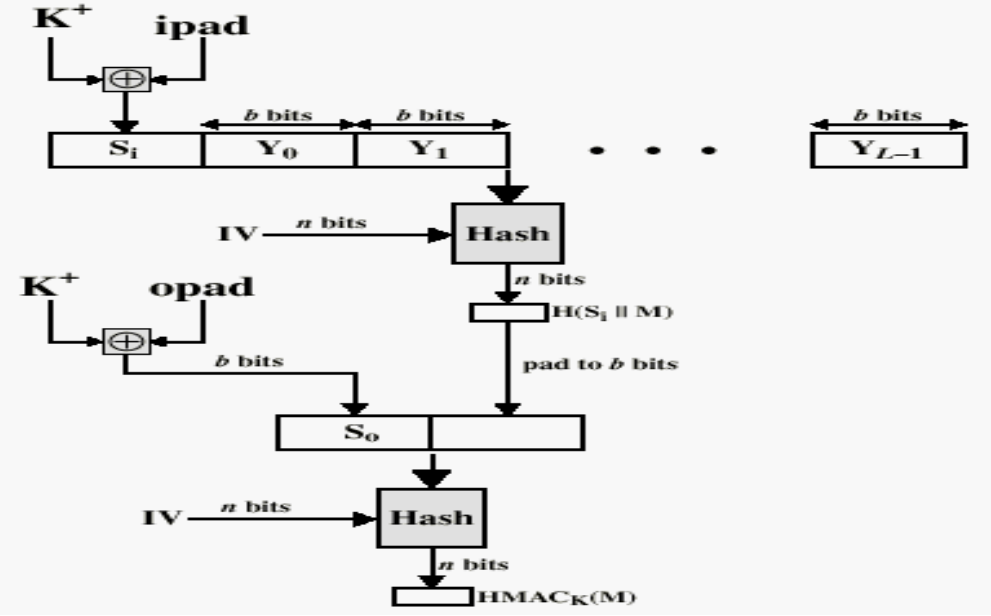

Figure 5.5 : Principe de l'algorithme HMAC

3.9. El Gamal [33]**:** c'est un algorithme de cryptographie asymétrique qui se base sur le problème du logarithme discret, crée par Tahar El Gamal et utilisé pour la signature des messages. Son déroulement est le suivant :

- A est l'entité qui veut signer un message M.
- Soient p un nombre premier et $S_k = x$ ($x < p$) une clé privée de A.
- Soit $P_k = (y, g, p)$ la clé publique de A avec $g < p$ et $y = g^x \bmod p$.
- Soit un nombre entier k tel que PGCD (K, p-1) = 1. k doit être gardé secret par A et B.
- A calcule $a = g^k \bmod p$.
- A calcule b tel que $M = (xa + kb) \bmod (p-1)$.
- Le couple (a, b) représente la signature du message M.
- A la réception du message M par B, B vérifie la validité de la signature (a, b) comme suit : $y^a a^b \bmod p = g^M \bmod p$.
- Une condition nécessaire pour maintenir le secret de la méthode est d'utiliser une nouvelle valeur de k à chaque signature.

4. ECC et cryptographie

4.1. Définition

Afin d'assurer un niveau de sécurité appréciable en utilisant un système ECC, il faut trouver un problème difficile à résoudre comme le problème de la factorisation d'un produit en ses facteurs premiers, utilisé par le système RSA. Pour cela, nous considérons l'équation suivante : $Q = [k] P$ où $Q, P \in E_p$ et $k < p$. Comme souligné déjà plus haut, il est aisé de calculer Q connaissant k et P mais il est très ardu et voire impossible pour un k assez grand, de déterminer k sachant P et Q. Ce problème est le problème du logarithme discret, très difficile à résoudre dans le contexte des courbes elliptiques. La cryptographie sur les courbes elliptiques (ECC) se base sur ce problème pour l'échange de clés, le chiffrement et même la signature des messages entre deux entités communicantes. Elle a été présentée indépendamment par [53] et [69] au milieu des années 80, et récemment, ECC attire l'attention des chercheurs en raison de la taille trop courte de ses clés pour un niveau de sécurité égal ou supérieur aux autres systèmes de sa catégorie (RSA, Diffie-Hellman…etc.).

On sait que le principe de la cryptographie publique repose sur un couple de clés (publique, privée) et son secret est la difficulté de retrouver la clé privée à partir de la clé publique. Plusieurs problèmes mathématiques ont été utilisés dans la cryptographie publique pour atteindre ce secret et qui doivent respecter certaines exigences, notamment celles liées à la taille des clés, le temps de traitement et la quantité de mémoire utilisée. Dans le système ECC, le problème employé est celui du logarithme discret, reposant sur des clés de tailles réduites, des temps de traitement raisonnables et des capacités de stockage moins importantes. Les ECC conviennent bien aux RcSF car elles respectent leurs contraintes en calculs, en temps de traitements, en capacités de stockage et en consommation d'énergie.

4.2. Echange de clés

Soit un grand nombre premier p et deux paramètres a, b satisfaisant l'équation (y^2 mod p) = ((x^3 + ax + b) mod p). a et b permettent de définir un E_p (a, b) d'ordre n très élevé. On choisit au hasard un point de départ G (x_1, y_1) dans E_p (a, b) qu'on rendra publique avec E_p (a, b). L'échange de clés entre les deux entités communicantes A et B se réalise comme suit :

- A choisit x < n et B choisit y < n respectivement comme clés secrètes.
- A calcule sa clé publique P_A = [x] G et B calcule également sa clé publique P_B = [y] G.
- A et B échangent publiquement leurs clés publiques P_A et P_B.
- A génère sa clé privée $K_{(A, B)}$ = [x] P_B et B génère sa clé privée $K_{(A, B)}$ = [y] P_A.
- La clé $K_{(A, B)}$ est la clé symétrique partagée de manière sécurisée par A et B.

La figure 5.6 illustre les étapes d'établissement de clés entre les nœuds A et B.

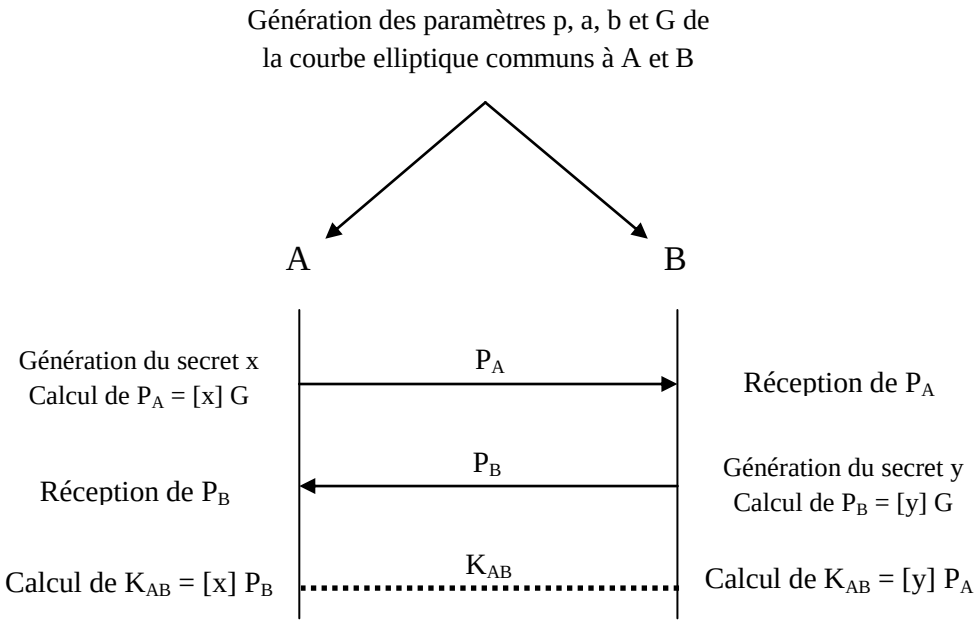

Figure 5.6 : Etapes d'établissement de clés symétriques

4.3. Taille des clés

Dans un système cryptographique, la taille des clés représente le paramètre essentiel pour mesurer son niveau de sécurité. Or, plus la taille des clés augmentent, l'attaquant prendra plus de temps à retrouver le secret de la clé et casser le système cryptographique surtout quand il s'agit d'attaques génériques basées sur la recherche exhaustive ou sur la technique du paradoxe des anniversaires (on cite particulièrement les algorithmes de Pohlig-Hellman [76] et de Pollard-Rho [81]). Pour un système cryptographique basé sur les ECC, une sécurité de 2^{80} opérations exigera des corps d'ordre minimum de 2^{160}, c'est-à-dire choisir un nombre premier $p \geq 160$ bits. Si on généralise pour une sécurité 2^n opérations, il faut créer des nombres premiers de taille supérieure ou égale à 2*n. Une petite comparaison avec le système RSA [35], ce dernier étant considéré comme le système de cryptage le plus utilisé dans sa catégorie, et pour une sécurité de 2^{80} opérations, ECC nécessite des clés de 160 bits tandis que RSA nécessite des clés de 1024 bits. Autrement dit, pour une sécurité égale, les clés utilisées par RSA sont sept fois plus longues que les clés utilisées par ECC. Encore, pour une sécurité de 2^{112} et 2^{128} opérations, ECC nécessite des clés de 224 et 256 bits respectivement, tandis que RSA nécessite des clés de 2048 et 3072 bits respectivement. Le tableau ci-dessous résume cette comparaison.

Niveau de sécurité (nombre d'opérations)	**Taille des clés**		**Durée de vie du crypto-système**
	RSA	ECC	
2^{80}	1024	160	Environ 04 années
2^{112}	2048	224	Environ 20 années
2^{128}	3072	256	Environ 30 années

Tableau 5.5 : Taille des clés des systèmes RSA et ECC pour une sécurité égale [35]

4.4. Avantages et inconvénients

Le tableau (5.6) présente les avantages et les inconvénients de la cryptographie sur les courbes elliptiques.

Avantages	Inconvénients
+ Taille des clés inférieure pour une sécurité égale. + Calculs moins lourds. + Utilisation moindre de mémoire. + La taille des clés croit moins vite que les autres systèmes pour une meilleure sécurité. + Permet à la fois l'échange de clés, le chiffrement et la signature. + Convient aux systèmes embarqués. + Basé sur le problème du logarithme discret difficile et appartenant à la classe des problèmes NP. + Cryptanalyse par algorithmes exponentiels.	- Complexes. - Nécessite des travaux d'optimisation essentiellement destinés aux systèmes mobiles.

Tableau 5.6 : Avantages et inconvénients des systèmes basés sur ECC

Au travers de toutes les comparaisons faites sur le système ECC et les autres systèmes de cryptage asymétrique, on conclut que l'utilisation de la cryptographie sur les courbes elliptiques convient aux réseaux de capteurs sans fil pour la taille réduite de ses clés et pour tous les autres avantages cités dans le tableau ci-dessus. Son utilisation est encore plus importante dans la résolution du problème d'échange de clés jusque là considéré comme le point faible des systèmes cryptographiques et des schémas de gestion de clés. La sécurité de cet échange de clé est plus élevée et fournit des clés de taille réduite tant que les problèmes de la classe NP le resteront.

La section suivante sera consacrée à la présentation de notre solution pour le routage de données et la sécurité des communications, en utilisant un système hybride basé sur le système cryptographique ECC pour l'établissement des clés symétriques entre les entités communicantes et leur utilisation pour sécuriser l'échange de données.

5. Notre solution

5.1. Aperçu du protocole

Le protocole proposé est un protocole de routage hiérarchique sécurisé. Les principes des approches de routage et de sécurité sont présentés ci-dessous. Le schéma de gestion de clés [82] est déterministe et permet de sécuriser toutes les étapes d'initialisation, de construction et de maintenance de l'architecture du réseau en garantissant des communications intra-cluster, inter-cluster, station de base (SB) – Cluster Head (CH), SB – Nœuds relais et CH – nœuds relais sécurisées. Lors de la phase de pré-distribution, la SB pré-charge tous les nœuds avec une clé initiale K_{init} utilisée pour authentifier les messages de construction de la topologie du réseau. Après l'expiration de la phase de construction, la clé K_{init} sera supprimée de la mémoire de tous les nœuds. La confidentialité des informations durant toute la durée de vie du réseau est garantie par le chiffrement symétrique des paquets de données, en utilisant des clés générées à partir d'un système asymétrique (ECC). L'authentification et l'intégrité de données sont assurées par l'algorithme HMAC, la signature des messages peut être obtenue par l'algorithme de El Gamal, exécuté sur les mêmes données du système ECC.

L'approche du routage proposée permet de construire une topologie hiérarchique maintenue durant toute la durée de vie du réseau. On procède au remplacement des CHs et des nœuds relais dès que leurs ressources en énergie atteignent un niveau bas afin d'éviter le partitionnement du réseau. La gestion de clés se déroule en plusieurs phases : pré-distribution, génération et échange, révocation et renouvellement des clés et l'assignation de clés aux nouveaux nœuds déployés sur le réseau.

5.1.1. Principe de la solution de routage

- Utilise une architecture hiérarchique sur deux niveaux.
- La construction se fait progressivement à partir de la SB.
- Le premier niveau regroupe la SB et ses nœuds voisins directs, considérés comme des nœuds relais.
- Chaque nœud relais diffuse une requête de découverte de voisinage et construit son cluster composé de ses voisins directs comme membres dont le CH est lui-même. Il choisira parmi ses voisins les plus éloignés un CH à qu'il annoncera son élection. Le CH désigné diffuse à son tour une requête de découverte de voisinage et annonce à ses membres qu'il est le CH du cluster.
- Chaque CH du niveau supérieur désigne un CH pour le niveau directement inférieur à son niveau. Le CH désigné construit son cluster et informe ses membres sur cette désignation.
- Le dernier CH (au dernier niveau) remontera une requête contenant son ID et un tableau contenant l'ensemble des identificateurs de ses membres.
- Chaque CH récepteur du message ajoutera son ID et un tableau des identificateurs de ses membres.
- La SB regroupe l'ensemble des informations reçues dans une grille symétrique qui lui sert de support sur l'organisation du réseau.

Réorganisation du réseau

- Chaque CH de niveau supérieur procède au remplacement du CH fils (de niveau inférieur) sans toutefois changer la topologie du réseau.
- Si tous les nœuds d'un cluster sont passés par le rôle de CH, une nouvelle procédure de création de clusters sera lancée par le nœud du niveau supérieur et ne concernera que les nœuds des niveaux inférieurs.
- Si un nœud relais tombe en panne ou épuise son énergie, il sera remplacé par un autre nœud relais.

Caractéristiques:

- Une solution qui convient bien aux réseaux denses comme les RcSF.

- Le nombre de messages pour la construction et la réorganisation de la topologie du réseau est très réduit : seuls les CH désignés sont autorisés à diffuser des messages de type Broadcast.
- Structure en clusters : économie d'énergie par l'emploi de techniques d'agrégation.
- Connaissance parfaite de l'organisation et de la topologie du réseau par la SB, ce qui lui permet de cibler directement des zones spécifiques.
- La coopération entre les nœuds pour remonter les données captées à la SB.
- Tolérance aux pannes : on peut rapidement contourner un CH en panne en désignant un autre CH parmi les nœuds du cluster (tous les nœuds sont voisins).
- Réorganisation locale synonyme d'un nombre très réduit de messages de reconstruction : économie d'énergie et de bande passante.

La figure suivante présente le principe du protocole de routage proposé:

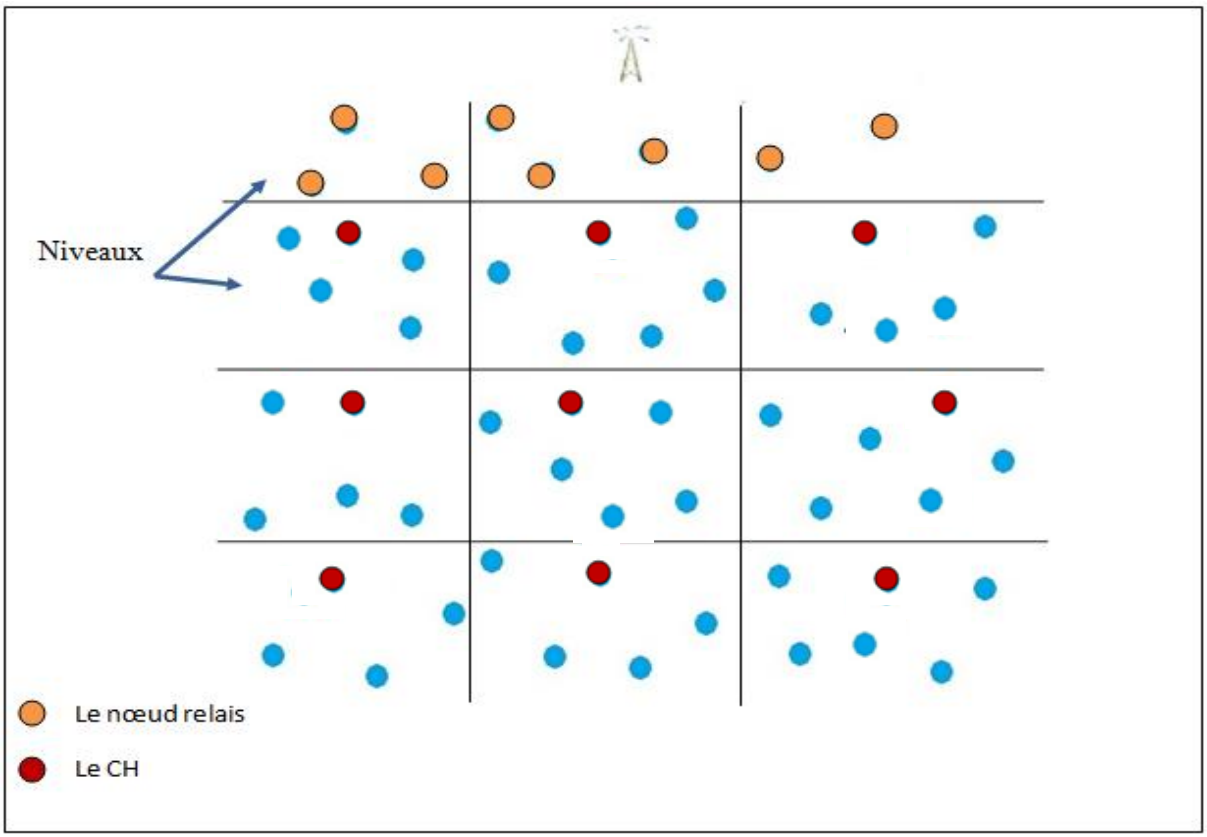

Figure 5.7 : Principe général de la solution de routage proposée

5.1.2. Principe du schéma de gestion de clés

La sécurité du routage et des communications s'effectue sur 03 phases [82] : avant le déploiement, avant l'expiration de la période Tmin dite période de confiance et après l'expiration du temps Tmin de confiance.

La technique utilisée pour la sécurité est une technique hybride à la fois symétrique et asymétrique. On calcule des clés publiques ECC en utilisant le principe du logarithme discret. Le problème est exponentiel d'où la solution pour l'attaquant devient impossible à calculer (le problème est NP-complet) et l'algorithme devient intéressant pour la cryptographie. Les clés publiques sont échangées par les entités communicantes du réseau afin d'établir des clés symétriques.

Avant le déploiement, toutes les clés publiques et paramètres de génération de clés symétriques de chaque nœud sont calculés par la station de base et pré-chargés sur les capteurs.

Au déploiement et avant l'expiration de la période de confiance, tous les échanges sont sécurisés par la clé K_{init} en utilisant des fonctions de hachage à sens unique (non réversibles) et des codes d'authentification MAC afin d'assurer l'intégrité, la confidentialité et l'authentification.

Après Tmin, les nœuds communicants utilisent les clés symétriques générées lors de la construction de la topologie du réseau.

Les clés symétriques sont révoquées et renouvelées après chaque détection de nœuds compromis.

5.2. Hypothèses

Le protocole de sécurité proposé se base sur les hypothèses suivantes :

- Le réseau est statique et le déploiement est aléatoire..
- Les nœuds capteurs sont homogènes, ils ont les mêmes capacités physiques, énergétiques et de communication.
- L'acheminement des données se fait via une communication multi-sauts.
- Les nœuds CHs peuvent exécuter des taches supplémentaires sur les données reçues de leurs membres de cluster.
- Les nœuds intermédiaires relaient les données reçues sans traitement supplémentaire.

- Les nœuds sont déployés avec des systèmes de détection d'intrusion (SDI). Les nœuds CHs assument la tache de détection d'intrusion en acceptant les messages des membres de leurs clusters indiquant des attaques malveillantes et en remontant cette information à la station de base.
- Les requêtes de construction et de reconstruction du réseau sont lancées simultanément à chaque niveau de construction ou de reconstruction.
- Les puissances de transmission des nœuds capteurs sont identiques et le temps mis par un message émis par la source pour atteindre une destination peut être remplacé par la distance qui les sépare.
- Les nœuds capteurs utilisent des horloges communes.
- Un attaquant peut opérer à n'importe quel niveau de la topologie du réseau.
- Un attaquant ne peut jamais compromettre une station de base.
- La compromission d'un nœud implique un accès complet de l'attaquant à sa mémoire.
- Aucune contrainte n'est imposée sur les capacités en calcul, en stockage et en énergie de la station de base.

5.3. Notations et terminologie

Notation	Description
p, g, x, P_x	Paramètres de génération et d'échange de clés de la méthode ECC.
K_{init}	Clé initiale pour authentifier les messages de construction du réseau.
A → B : M	Message M diffusé par A pour B.
A → * : M	Message M diffusé par A en broadcast à tous ses nœuds voisins.
$K_{(A, B)}$	Clé symétrique partagée par A et B.
Id_A	Identité du nœud A.
(E (K, Id_A, Motif, Valeur), Autres valeurs)	Chiffrement du message à transmettre avec les paramètres suivants : - K : clé de chiffrement / déchiffrement. - Id_A : identité de l'émetteur. - Motif : l'objectif du message : Join (joindre le cluster), Replay (Accepter d'appartenir au cluster), Share (partager une clé), Renew (renouvellement des clés), design (désigner le récepteur comme CH), cannot (refuser la tache de CH)

	- Valeur : une valeur transmise soit pour établir une clé commune ou bien elle comporte une donnée de type tableau. - Autres valeurs : des valeurs optionnelles partagées publiquement et qui sert souvent à changer les paramètres de la méthode ECC.
MonCH	Identité du nœud CH du cluster.

Tableau 5.7 : Notation utilisée dans notre solution

5.4. Phases de déroulement du protocole

5.4.1. Phase de pré-distribution

La station de base effectue les opérations suivantes :

- Calculer les paramètres qui seront utilisés pour exécuter l'algorithme de la cryptographie sur les courbes elliptiques (ECC) : un grand nombre premier p et un point G.
- Initialiser un tableau T contenant un identificateur pour chaque nœud du réseau, une clé secrète x utilisée pour établir une clé privée entre un nœud et ses voisins, une valeur calculée P_x (Voir le tableau (5.7)). P_x est calculée comme suit : $P_x = g^x \bmod p$.
- Générer une clé initiale commune K_{init}.
- Charger chaque capteur avec les informations suivantes : ID, K_{init}, p, x, P_x.

5.4.2. Phase de construction

a) La station de base (SB) diffuse un message de découverte de voisinage :

SB → * : E (K_{init}, Id_{SB}, Join, 0) ……(1).

b) A la réception du message (1), les nœuds voisins de la SB répondent par le message suivant : V → SB : E (K_{init}, Id_x, Replay, P_x).

c) La SB reçoit tous les messages des voisins V et établit une clé de session avec eux en calculant $K_{(SB, V)} = (P_x)^y$. Puis elle envoi à chaque nœud voisin le message suivant : SB → V : E (K_{init}, Id_{SB}, P_y). Les nœuds voisins calculent de la même manière la clé de session avec la SB : $K_{(SB, V)} = (P_y)^x$. A la fin de cette étape, (N/10) de nœuds sont désignés comme des CHs. (N est le nombre de nœuds du réseau).

d) Les nœuds CHs diffusent un message de découverte de voisinage : CH → * : E (K_{init}, Id_{CH}, Join, 0) ……(2).

e) Les nœuds récepteur du message (2) répondent par un message Replay et ignorent tous les autres messages reçus : V → CH : E (K_{init}, Id_V, Replay, P_x).

f) Le nœud CH reçoit tous les messages de ses nœuds voisins qu'il considère comme des nœuds membres de son cluster. Pour chaque nœud membre, calcule $K_{(CH, m)} = (P_x)^y$ et lui transmet le message suivant : CH → M : E (K_{init}, Id_{CH}, Share, P_y). Le nœud membre initialise sa variable MonCH = Id_{CH} et calcule sa clé de session avec son Cluster Head : $K_{(CH, m)} = (P_y)^x$.

g) Après l'établissement de clés symétriques avec chacun des nœuds de son cluster, le CH diffuse un message contenant un secret x que tous les nœuds du cluster doivent utiliser pour établir une clé commune pour authentifier les messages en Broadcast du cluster. Le message est : CH → M : E (K_{init}, Id_{CH}, Share, x).

h) Le CH et chaque membre de son cluster calcule la clé privée commune en utilisant le paramètre x.

i) A la fin de cette étape, tous les nœuds CH auront établis trois types de clés : une clé partagée avec la station de base, une clé partagée avec chacun des nœuds du cluster et une clé commune partagée avec tous les nœuds du cluster. Les nœuds membres établissent deux types de clés : une clé partagée avec le CH et une clé commune partagée avec tous les nœuds du cluster.

5.4.3. Phase de reconstruction

La reconstruction de la topologie du réseau se fait à chaque fois que le niveau en ressources énergétiques d'un nœud CH atteint un niveau beaucoup plus inférieur aux autres niveaux des autres nœuds du réseau. Cette constatation sera faite par la station de base qui diffuse un message de reconstruction du cluster ayant pour chef le nœud à remplacer et l'identité du nouveau nœud CH du cluster. Ce dernier envoi un message en Broadcast à tous les nœuds membres du cluster pour les informer de ce changement en modifiant leurs variables MonCH avec l'identité du nouveau chef et en supprimant les clés partagées avec l'ancien CH et lancent des messages d'établissement et d'échange de clés avec le nouveau CH.

5.4.4. Phase de renouvellement

Cette mesure vise à renforcer la sécurité du réseau. la station de base (SB) diffuse un message de renouvellement de clés SB → CH : (E ($K_{(SB, CH)}$, Id_{SB}, Renew, 0), p, g, P_y)…(1) avec p et g les deux nouvelles données générées publiquement et utilisées dans le calcul de la clé de partage. Le message (1) sera relayé par tous les nœuds CH.

Chaque nœud CH lance la procédure de renouvellement de clés avec ses nœuds membres du cluster. Durant la procédure de renouvellement, chaque nœud CH calcule $P_y = g^y \bmod p$ et le transmet à tous ses nœuds membres et chaque nœud membre calcule son $P_x = g^x \bmod p$ et le transmet au CH. Les deux parties génèrent une clé de session en calculant $(P_y)^x$ du coté des nœuds membres et $(P_x)^y$ du coté du CH.

5.4.5. Phase de révocation

La phase de révocation se déclenche à chaque compromission d'un nœud du réseau. Si le nœud compromis est un nœud membre d'un cluster, son CH supprime de sa mémoire la clé partagé avec lui, le place en quarantaine, ignore ses messages et le signale à la station de base pour ne jamais le désigner CH lors des phases de reconstruction du réseau. Si le nœud compromis est un CH, la station de base supprime la clé partagée avec lui, désigne un autre CH parmi les membres du cluster du CH compromis qui se charge d'informer et d'avertir tous les nœuds membres du cluster. Aussitôt informés, ils procèdent à la suppression des clés partagées avec le CH compromis, ignorent ses messages, le mettent en quarantaine et remplacent leurs variables MonCH par l'identité du nouveau CH.

5.4.6. Insertion des nouveaux nœuds

La station de base pré-charge un nouveau nœud avec les valeurs (Id, x, P_x) et diffuse un message à tous les nœuds CH comportant l'identité du nouveau nœud à insérer. Après le déploiement, le nouveau nœud diffuse un message d'appartenance à un cluster et le nœud CH le plus proche du nouveau nœud vérifie son identité avec les identités des nouveaux nœuds à ajouter sur le réseau. Si l'identité du nœud est vérifiée, le nœud CH établit une clé de session avec lui en lui transmettant un message Join.

6. Conclusion

Dans ce chapitre, nous avons proposé un protocole de routage sécurisé et économe en énergie pour les réseaux de capteurs sans fil. La solution de routage proposée permet la construction d'une topologie en clusters moins gourmande en ressources physiques et en consommation d'énergie. La solution de sécurité est une fonction de gestion de clés basée sur la cryptographie sur les courbes elliptiques (ECC) pour générer des clés symétriques et résoudre le problème d'échange de clés entre les nœuds capteurs du réseau après leur déploiement. Cette solution garantit un niveau de sécurité élevé en utilisant des clés de petites tailles.

Notre protocole parait prometteur et présente des mécanismes nouveaux pour les fonctions de construction des topologies hiérarchiques et de gestion de clés en combinant les avantages des chiffrements asymétrique et symétrique afin d'établir et d'échanger, en toute sécurité, des clés de taille réduites convenables aux capacités limitées en calcul et en mémoire des RcSF.

Les sections du chapitre suivant tentent de confirmer l'efficacité de notre protocole en présentant des résultats de simulation et des comparaisons avec des travaux antérieurs sur la sécurité des échanges dans les RcSF.

CHAPITRE 06
SIMULATIONS ET RÉSULTATS

Dans ce chapitre, nous allons présenter les résultats des simulations effectuées sur notre solution proposée. Au vu des résultats obtenus, nous avons classé les données dans des tableaux et graphes relativement à nos métriques et en fonction de la taille du réseau. Nous avons comparé notre solution de sécurité et de gestion de clés à une solution combinant les approches probabilistes et dites t-secure que nous avons implémentée dans un même environnement TinyOs. Ces comparaisons sont faites à travers des résultats de simulation obtenus à partir du système de sortie de débogage du simulateur Tossim et de sa variante PowerTossim.

1. Introduction

Les problèmes du nombre et de la taille des paquets de données échangés durant les phases de déroulement du protocole (découverte de voisinage, installation des clés, renouvellement des clés et insertion des nouveaux nœuds) ainsi que le problème de la consommation d'énergie par les nœuds capteurs sont deux concepts majeurs déterminant la durée de vie d'un réseau de capteurs sans fil (RcSF). C'est pour cette raison que la plupart des solutions proposées essayent davantage de réduire le nombre de messages émis et reçus et la quantité d'énergie consommée dans les opérations cryptographiques durant tout le cycle de vie du réseau.

Plus haut dans ce mémoire (chapitre 04), nous avons donné une classification des méthodes de gestion de clé en déterministes, probabilistes, géographiques et dites t-secure. Nous avons évoqué l'absence d'authentification entre chaque paire de nœuds et le problème d'espace mémoire réservé aux clés stockées comme principaux inconvénients des méthodes et protocoles probabilistes et nous avons cité quelques travaux d'amélioration pour ces méthodes. Justement, beaucoup de travaux ont été faits dans ce sens en combinant les modèles probabilistes aux modèles dits t-secure (Blom et Blundo entre autres) avec des corrections faites sur l'insertion des nouveaux nœuds. Dans la littérature, l'idée de combiner le modèle probabiliste avec les schémas dits t-secure est illustrée par le protocole TinyKeyMan [56]. En adoptant les deux modèles, TinyKeyMan souffre de quelques inconvénients : A chaque construction ou reconstruction de routes pour une installation ou une mise à jour des clés, le nombre de paquets échangés entre les nœuds accroit rapidement avec la taille du réseau et la probabilité que deux nœuds partagent un même secret ou trouvent un voisins commun pour établir une clé symétrique devient de plus en plus faible.

Par conséquent, on peut dire que ces deux approches combinées n'offrent pas de compromis entre la taille du réseau et le nombre de messages échangés. Dans ce contexte, nous avons présenté une nouvelle solution déterministe qui assure une probabilité complète (100%) d'établir une clé symétrique entre chaque paire de nœuds sans le partage d'aucune information. Ce gain de messages transmis et reçus est synonyme d'une consommation réduite de l'énergie. Les opérations de calcul de secrets et de clés sont également optimisées pour une consommation raisonnable de ressources physiques et énergétiques des nœuds capteurs. Dans ce qui suit, nous allons présenter les métriques et paramètres utilisés, la librairie TinyKeyMan, et les résultats obtenus par rapport à ces métriques et en fonction du nombre de nœuds du réseau.

2. Environnement de simulation

Afin de réaliser notre simulation et d'évaluer les performances de notre protocole, nous avons choisi d'utiliser l'environnement de travail Tinyos [55] et d'effectuer nos tests avec le simulateur Tossim [54] et sa variante PowerTossim [88]. Les choix de l'environnement de travail et du simulateur sont déjà expliqués dans la section (5-8) du premier chapitre de ce mémoire.

2.1. Les paramètres de simulation

Pour le modèle expérimental, nous avons utilisé un réseau formé de 100 à 1000 nœuds dispersés aléatoirement sur une zone de 1000 x 1000 m (voir le tableau 6.1). Environ 10% de ces nœuds assument la tache de Cluster Head (CH). La portée radio d'un capteur est de 15.6 mètres, le modèle radio est simple c'est-à-dire que le taux d'erreur de transmission est nul, la taille d'un paquet de données est de 40 octets. A noter que l'utilisation d'un modèle simple de transmission inclut une transmission parfaite de bits où la probabilité que deux nœuds émettent en même temps est très faible en raison de l'emploi du protocole CSMA de Tinyos [54]. Cela évite des collisions et des chevauchements et par conséquent les pertes de données sont négligeables.

Nombre de nœuds du réseau (N)	100 à 1000 nœuds
Nombre de clusters	10% (N / 10)
Nombre de nœuds d'un cluster	En moyenne 10
Nombre de nœuds intrus	De 0 à 10% de N
Modèle de topologie	Aléatoire statique
Modèle radio	CC1000

Portée radio d'un capteur	15.6 mètre
Taille d'un paquet de données	40 octets
Type du paquet	AM
Durée de la simulation (virtuelle)	60 secondes
Puissance de réception Rx	7.0 mA
Puissance d'émission Tx à 0 dBm	8.5 mA

Tableau 6.1 : Paramètres de la simulation utilisés

2.2. Les métriques

Pour l'évaluation des performances de notre solution, nous nous basons sur les métriques suivantes :

- Le nombre de clés stockées dans la mémoire des nœuds capteurs.
- La taille des clés stockées dans la mémoire des nœuds capteurs.
- Le nombre de paquets de données échangés durant les différentes phases de déroulement du protocole : initialisation (découverte du voisinage), installation des clés symétriques, révocation des clés découvertes par les nœuds intrus, renouvellement des clés et insertion des nouveaux nœuds.
- La consommation d'énergie moyenne par tous les nœuds du réseau en fonction de la taille du réseau.
- La consommation d'énergie par composant : énergie consommée durant les opérations de calcul des clés et l'énergie consommée pour l'émission / réception des paquets de données.

Le nombre de clés stockées dans la mémoire des nœuds est obtenu par l'exécution de notre protocole de sécurité et de gestion de clés sur le protocole LEACH [37], considéré comme la référence des protocoles hiérarchiques et qui convient mieux aux besoins en simulation de notre solution.

La taille des clés ECC présentes dans la mémoire des nœuds capteurs et utilisées par notre protocole de sécurité et de gestion de clés est calculée sur la base des données présentées sur le tableau (6.2):

Taille du nombre premier P	160 bits
Taille du secret x	< 160 bits
Taille d'un point Px	320 bits
Taille d'une clé publique	320 bits
Taille d'une clé privée	160 bits

Tableau 6.2 : Taille des clés ECC et de leurs paramètres de calcul

Les messages échangés ont une taille de 40 octets et sont de type AM. Un message d'authentification et à la fois d'intégrité de données et de cryptage (puisque nous utilisons la technique HMAC-SHA1 pour authentifier les messages) a une taille minimale de 63 octets (41 octets pour le texte chiffré C, 20 octets (160 bits) pour le bloc hmac et 2 octets pour les informations d'identification) ce qui nécessite l'émission et la réception de deux paquets pour chaque opération cryptographique d'authentification. Un message d'installation de clés par paire (clés symétriques) a une taille minimale de 42 octets (40 octets pour le point public Px et 2 octets pour les informations d'identification) ce qui nécessite également l'envoi et la réception de deux paquets pour chaque opération d'installation de clés entre chaque paire de nœuds.

L'énergie consommée pour émettre ou recevoir un paquet est obtenue par l'exécution d'un script PowerTossim sur un fichier d'extension (.trace) généré avec les paramètres de simulation cités ci-dessus et par le plugin « Power profile » de l'outil graphique TinyViz du simulateur Tossim. Cette énergie peut être calculée également de la manière suivante si on considère à 3V la tension nécessaire dans les opérations radio :

- L'énergie consommée pour recevoir un paquet : ERx = 7mA x 3V = 21 mW.
- L'énergie consommée pour émettre un paquet : ETx = 8.5mA x 3V = 25.5 mW.

3. Simulation et évaluation des performances

Comme nous l'avons indiqué ci-dessus, nous avons implémenté notre protocole en utilisant l'environnement TinyOs incluant le simulateur Tossim qui permet de tester les performances d'une application dans un réseau sans fil virtuel. Néanmoins avec le simulateur Tossim, on peut récupérer le nombre de paquets échangés lors des différentes opérations cryptographiques du protocole mais ne permet pas d'évaluer exactement la consommation énergétique d'un nœud. Pour cela, nous avons obtenu cette consommation avec sa variante

PowerTossim qui fournit des résultats détaillés sur la dépense d'énergie par les différents composants d'un nœud capteur (CPU, radio, …etc.).

Afin de comparer les performances de notre solution avec des solutions existantes, nous avons implémenté notre protocole et les autres protocoles de la littérature dans la même configuration réseau, c'est-à-dire avec les mêmes paramètres et métriques de simulation. Les deux autres protocoles utilisés sont TinyKeyMan [56] et TinyECC [72] (et plus particulièrement son module ECDH pour l'échange de clés).

Toutes les opérations cryptographiques que nous avons utilisées dans notre solution ont été programmées à l'aide des librairies de TinyECC : HMAC-SHA1 pour l'authentification, l'intégrité et le cryptage, ECDH pour l'échange des clés. La librairie ECDH offre une solution optimisée pour le calcul des clés symétriques et d'utilisation de la mémoire des nœuds qui convient aux réseaux de capteurs sans fil (RcSF).

3.1. Description des solutions utilisées pour l'évaluation

TinyKeyMan fournit une librairie efficace d'établissement et de gestion des clés de session entre chaque paire de nœuds capteurs. Les techniques d'établissement de clés sont basées sur les polynômes bi-variables de degré *t*, proposés par [15], générés et distribués avant le déploiement aux capteurs selon le schéma de [32]. Dans [56], les auteurs proposent deux méthodes de pré-distribution de polynômes :

a) Assignement aléatoire d'un sous ensemble de polynômes à chaque nœud capteur. Le secret est calculé en utilisant l'identification du capteur avec les polynômes pré-chargés.

b) Selon un schéma en grille sous forme d'hypercube de polynômes où chaque nœud capteur se voit attribuer un système de coordonnées unique dans l'espace. A partir de ces cordonnées, on génère l'ensemble des secrets du nœud.

Le calcul des secrets se fait avant le déploiement et il est optimisé par une technique d'évaluation polynomiale afin de réduire les couts en calcul et en mémoire.

Au déploiement, deux nœuds peuvent établir une clé symétrique s'ils 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 et donnée par la figure (6.5). Pour cela, les deux nœuds échangent les listes des identifiants de leurs polynômes. Si aucun polynôme n'est commun aux deux nœuds, on essaye de trouver un nœud voisin qui partage une clé symétrique avec chacun d'eux. Le nœud voisin calcule ensuite une clé aléatoire,

appelée clé de chemin (Key Path), qu'il va transmettre aux deux nœuds pour l'utiliser comme clé symétrique.

Les auteurs de ce schéma considèrent que la technique des polynômes à deux variables de degré *t* permet d'établir des clés de session entre chaque paire de nœuds même en présence d'intrus et de nœuds compromis. Un nœud attaquant ne peut déterminer la clé partagée par une paire de nœuds avec un polynôme de degré *t* s'il ne compromet pas à la fois *t* nœuds.

L'avantage de ce schéma est qu'un nœud peut déterminer rapidement et à moindre cout s'il peut établir une clé symétrique avec n'importe quel nœud voisin. Les techniques d'optimisation utilisées sur le calcul des secrets rendent l'emploi des schémas probabilistes possibles et efficaces pour les réseaux à ressources limitées comme les réseaux de capteurs sans fil. Pour tous ces avantages, nous avons choisi d'utiliser cette approche, à la fois probabiliste et t-secure, pour évaluer les performances de notre solution déterministe et qui utilise d'autres techniques de calcul et d'échange de clés.

3.2. Installation des clés

3.2.1. Utilisation de la mémoire

3.2.1.1. Nombre et la taille des clés stockées

Dans le tableau (6.3), nous avons calculé le nombre et la taille des clés stockées par un nœud Cluster Head (CH) et par un nœud normal membre d'un cluster dans une topologie hiérarchique. Dans cette topologie, chaque nœud CH partage trois types de clés : i) une clé partagée avec son nœud parent dans la hiérarchie, ii) une clé partagée avec chacun des nœuds de son cluster et iii) une clé commune partagée avec tous les nœuds de son cluster. Un nœud membre d'un cluster partage deux types de clés : i) une clé personnelle avec son CH et ii) une clé commune avec son CH et tous les autres nœuds du cluster. On note que tous les nœuds du réseau partagent entre eux à la phase d'initialisation une clé commune qui sert à authentifier et sécuriser toutes les communications durant la phase d'installation des clés par paires. Cette clé n'est pas prise en compte car elle sera supprimée à la fin de la phase d'initialisation et d'installation des clés.

La taille des clés est calculée selon les informations données par le tableau (6.2) ci-dessus. Le tableau (6.3) donne le nombre et la taille des clés stockées par chaque nœud du réseau où Nc est le nombre moyen de nœuds d'un cluster.

Type du nœud	Nombre de clés stockées	Taille des clés stockées
Nœud CH	(Nc + 2)	20 (Nc + 2) octets
Nœud membre	2	40 octets

Tableau 6.3 : Le nombre et la taille des clés stockées dans notre méthode

3.2.1.2. Nombre de paquets échangés

Les figures (6.1) et (6.2) présentent le nombre de paquets de données échangés lors de l'installation des clés sur des réseaux de taille variant de 100 à 1000 nœuds capteurs en présence ou non de nœuds attaquants. Les messages échangés concernent ceux des opérations d'authentification, de découverte de voisinage et de calcul des clés secrètes.

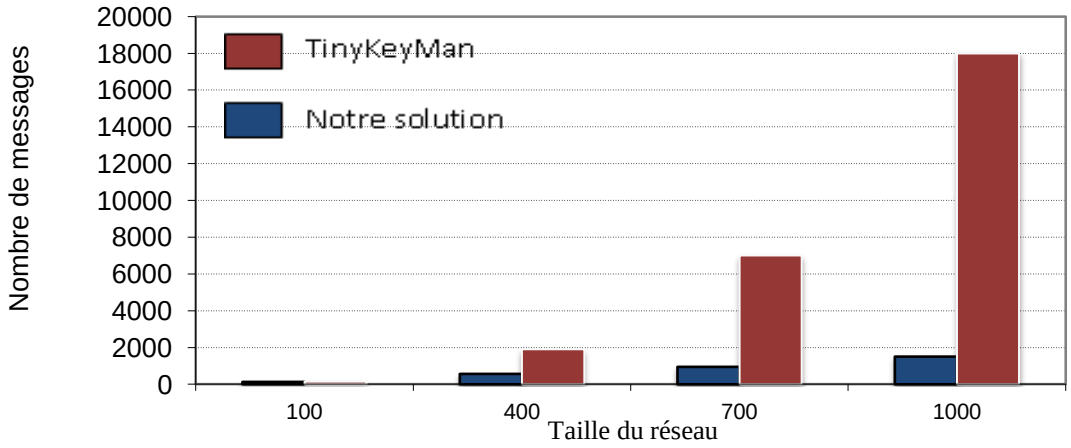

Figure 6.1 : Nombre de paquets échangés en absence d'intrus

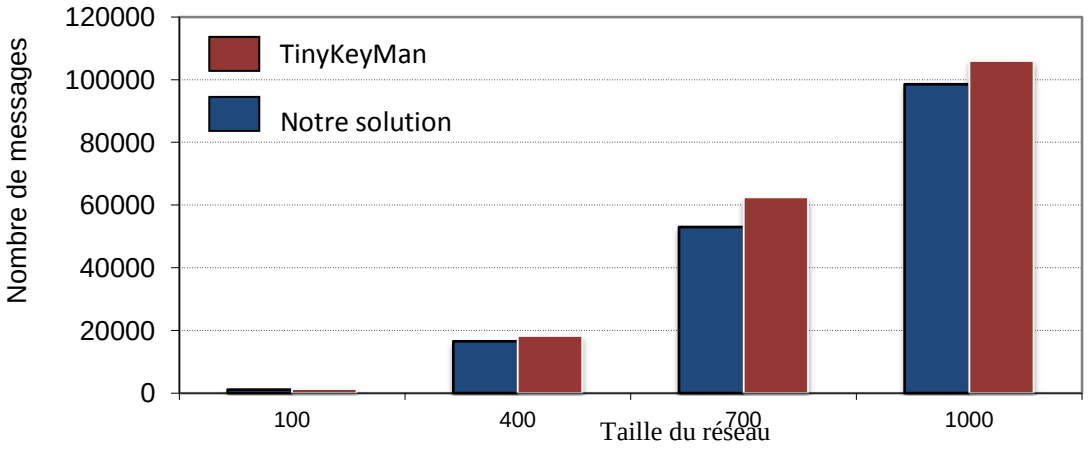

Figure 6.2 : Nombre de paquets échangés en présence d'intrus

Les deux figures ci-dessus résument les résultats obtenus après avoir réalisé différentes expériences. Sur la figure (6.1) où on suppose l'absence de nœuds attaquants, la complexité en communication de notre protocole est meilleure

notamment pour des réseaux denses comme les RcSF et sont négligeables pour les réseaux de petite taille. Notre solution échange moins de paquets, environ 10 fois moins que la solution probabiliste du protocole TinyKeyMan. Cette différence est justifiée par la figure (6.5) où la probabilité que deux nœuds partagent le même polynôme devient rapidement faible avec l'augmentation du nombre de nœuds du réseau d'où la nécessité de recourir à une tierce partie de confiance engendrant en conséquence des communications supplémentaires.

Dans la figure (6.2) où on suppose la présence de 10% de nœuds attaquants de l'ensemble des nœuds du réseau, le nombre d'informations échangées par radio est beaucoup plus important (environ une différence de 500% de paquets) et cela est dû à la diffusion d'un grand nombre de fausses informations de contrôle. Chaque nœud du réseau essaye alors d'authentifier l'émetteur puis d'ignorer ses futurs messages dans le cas d'une intrusion. Dans le protocole TinyKeyMan, les informations produites par les nœuds malicieux comportent une forte probabilité de partage d'un même polynôme ce qui touche à un plus grand nombre de nœuds.

Notre solution échange moins de paquets pour installer les clés ce qui garantie un temps d'installation plus court et un niveau de sécurité plus élevé.

3.2.2. Consommation d'énergie

3.2.2.1. Consommation d'énergie moyenne

En utilisant l'outil PowerTossim, nous avons évalué la consommation moyenne d'énergie d'un nœud capteur durant les phases de découverte de voisinage et d'installation des clés de session. Cette énergie est calculée sur la base des instructions exécutées pour les opérations cryptographiques (calcul du MAC par l'émetteur, vérification du MAC par le récepteur, calcul du secret x, calcul du point public Px et le calcul de la clé symétrique) et pour les opérations radio (émission et réception des messages de découverte de voisinage, d'échange du point Px et du MAC). La figure (6.3) montre la variation de l'énergie consommée en fonction du nombre de nœuds du réseau en présence ou non de nœuds attaquants.

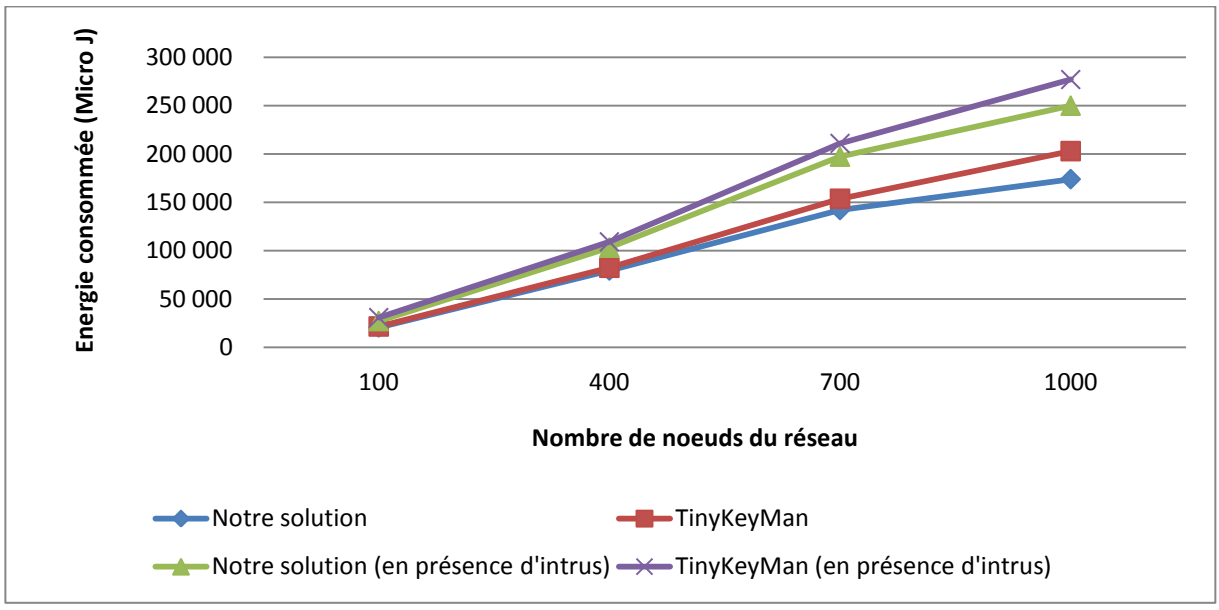

Figure 6.3 : Variation de la consommation d'énergie en fonction de la taille du réseau

Dans les deux approches, on remarque que la quantité d'énergie dépensée accroit, à environ 1000% entre N = 100 et N = 1000 nœuds, avec la taille du réseau. Cette augmentation de consommation est causée principalement par le nombre croissant de paquets échangés et par les opérations de calcul et de vérification effectuées par les nœuds capteurs. Par exemple, pour N = 100 nœuds et un nombre d'intrus variant de 0 à 10%, un nœud CH reçoit en moyenne entre 10 et 20 requêtes d'établissement de clés, effectue 10 à 20 vérifications sur les MACs reçus et calcule 11 clés de sessions (une clé avec chacun des nœuds de son cluster et une clé commune à tous les nœuds du cluster). Un nœud normal reçoit entre 01 et 20 requêtes de découverte de voisinage, effectue 01 à 20 vérifications sur les MACs reçus et calcule 02 clés de sessions (une clé partagée uniquement avec son CH et une autre clé partagée avec tous les nœuds du cluster). Par contre et pour N = 1000, un nœud CH reçoit en moyenne entre 100 et 200 requêtes d'établissement de clés, effectue 100 à 200 vérifications sur les MACs reçus et calcule 101 clés de sessions. Un nœud normal reçoit entre 01 et 200 requêtes de découverte de voisinage, effectue 01 à 200 vérifications sur les MACs reçus et calcule 02 clés de sessions.

3.2.2.2. Consommation d'énergie par composant

Comme on l'a déjà indiqué plus haut dans ce mémoire, les clés ECC étaient pour longtemps inutilisables pour les RcSF et cela est dû à la complexité des opérations de calcul des clés publiques (du point Px) traduite par une consommation énorme d'énergie

et de mémoire. Mais avec les optimisations réalisées sur cette méthode, il semble que cette consommation se réduit significativement et la technique convient de plus en plus aux réseaux à ressources limitées. La figure (6.4) démontre cette hypothèse.

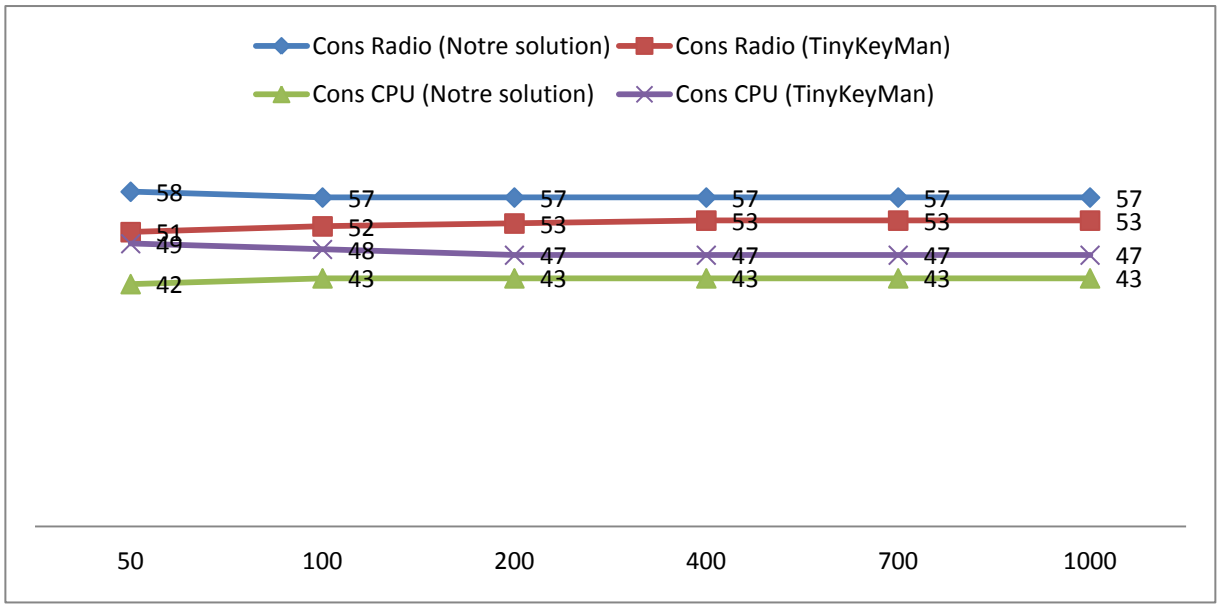

Figure 6.4 : Consommation d'énergie par composant

Ces simulations permettent de conclure que notre approche consomme moins d'énergie dans les opérations cryptographiques (authentification, intégrité, chiffrement et établissement de clés secrètes) et garantit aux nœuds capteurs une durée de vie plus longue comparativement au protocole TinyKeyMan.

3.2.3. Connectivité

La connectivité se définie comme la probabilité qu'un nœud puisse établir un lien sécurisé avec l'ensemble de ses voisins.

Dans les schémas déterministes, tous les nœuds sont pré-chargés avec une clé de génération. Cette clé est utilisée, après le déploiement et lors des phases de construction de la topologie du réseau et d'installation des clés, pour dériver des clés par-paires avec tous les nœuds voisins. Dans notre cas, le secret commun est l'ensemble des paramètres de définition de la courbe elliptique pour le calcul des clés publiques et des clés secrètes. Deux nœuds voisins A et B ont une probabilité égale à 1 d'établir un lien sécurisé. Dans les schémas probabilistes, les nœuds sont pré-chargés avec des sous ensembles de clés. Après le déploiement, deux nœuds peuvent établir un lien s'ils partagent au moins une clé commune dans leurs sous ensembles de clés. Dans le cas de TinyKeyMan, les nœuds sont pré-chargés avec des sous ensembles de polynômes bi-variables de degré t pour le calcul des clés privées entre chaque paire de

nœuds. La probabilité que deux nœuds voisins A et B partagent un secret varie selon la taille des sous ensembles de secrets partagés et selon la taille du réseau. La probabilité d'établir un lien sécurisé dépend de la probabilité de partager un secret. La figure 6.5 illustre la variation de la connectivité en fonction de la taille du réseau.

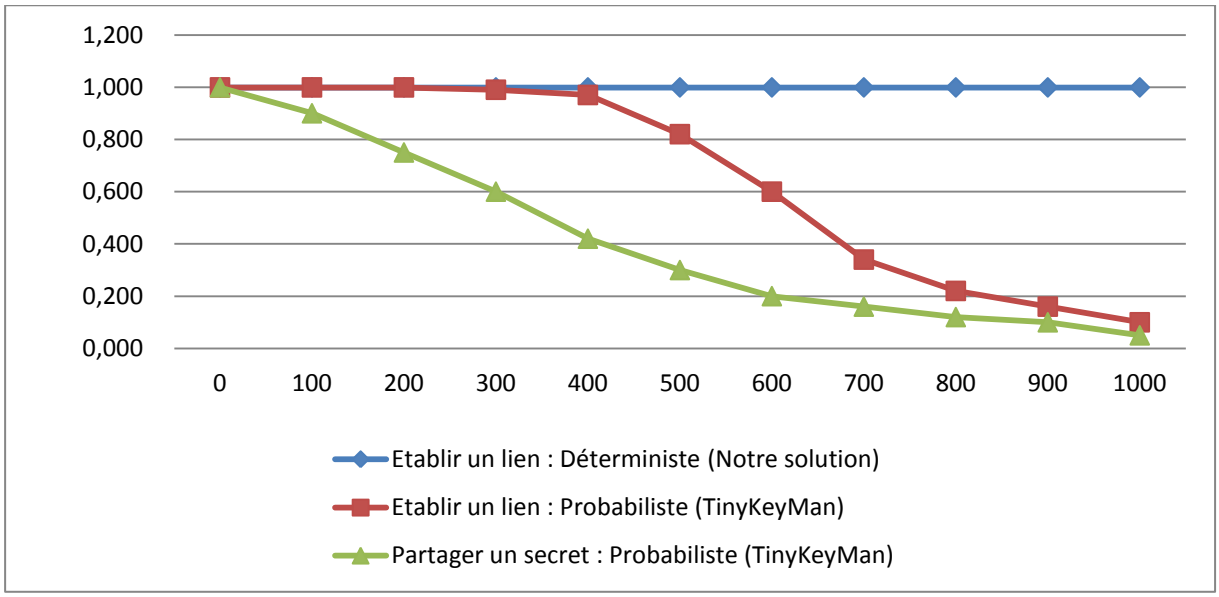

Figure 6.5 : Probabilité d'établir un lien

3.2.4. Sécurité des échanges

Dans les deux approches, les nœuds capteurs sont pré-chargés respectivement avec une clé globale d'authentification K_{init} et d'un secret commun utilisé pour la génération d'une nouvelle clé globale d'authentification. Les deux clés permettent de sécuriser tous les échanges de données et d'informations lors des phases d'initialisation et d'installation des clés secrètes. La phase d'installation de clés est vulnérable aux attaques passives qui consistent à récupérer les informations échangées par les nœuds pour tenter de retrouver le secret de génération et de lancer des attaques actives visant à compromettre le réseau.

Dans LEAP, un schéma de gestion de clés déterministe, les clés sont générées à partir d'une clé « master » en utilisant une fonction pseudo-aléatoire. Cette solution est peu résistante aux attaques massives car il suffit de compromettre un seul nœud du réseau pour divulguer le secret de la fonction pseudo-aléatoire et compromettre le réseau entier. Les autres approches déterministes basées sur une tierce partie de confiance pour l'établissement de clés résistent mieux mais consomment plus de ressources en conséquence.

Dans l'approche TinyKeyMan, un nœud attaquant peut utiliser les informations récupérées pour générer des sous ensembles de secrets de grande taille avec une certaine probabilité de partager un ou plusieurs secrets avec les nœuds du réseau. Le degré t des polynômes bi-variables utilisés dans cette approche lui garantit la propriété t-secure, mais si l'attaquant parvient à compromettre t+1 nœuds, le réseau entier sera compromis.

Les informations échangées par les nœuds exécutant notre protocole sont publiques et sont accessibles même pour les nœuds intrus. La difficulté de résoudre le problème du logarithme discret dans le contexte des courbes elliptiques permet à n'importe quelle paire de nœuds du réseau d'établir des liens sécurisés en présence de nœuds attaquants. Dans le tableau 5.5 ci-dessus, on estime la durée nécessaire pour un attaquant de casser un crypto-système basé sur des clés ECC de taille 160 bits à environ quatre années [35].

3.2.5. Résilience contre la capture

Comme dans LEAP, On suppose que les phases d'initialisation et d'installation de clés s'exécutent en un temps maximal Tmin et qu'un attaquant prendra au minimum un temps T > Tmin pour compromettre ou capturer un nœud et récupérer les informations contenues dans sa mémoire. Avant ce temps T, les phases d'initialisation et d'installation de clés soient achevées, la clé K_{init} est supprimée de la mémoire de tous les nœuds et de nouvelles clés symétriques sont installées entre chaque paire de nœuds. Après le temps T, le nœud compromis est facilement détectable et une opération de révocation du nœud est déclenchée. Si après Tmin un nœud est capturé ou compromis, la procédure de révocation et de renouvellement de clés sera déclenchée.

3.3. Cout de détection

Un autre critère d'évaluation des performances d'une solution est l'impact de détection d'une attaque par rapport au cout en énergie des ressources employées pour détecter et éliminer son comportement malicieux. Dans notre cas, on s'intéresse uniquement aux attaques sur la construction de la topologie du réseau et par conséquent sur le processus d'établissement de clés de session.

Comme notre protocole est distribué, ce cout est négligeable car les attaques sont automatiquement détectées et éliminées par les nœuds eux-mêmes en se basant sur

leurs informations locales pour le calcul des MACs. Ce cout de calcul pour la vérification de l'authenticité des messages reçus est très négligeable.

3.4. Renouvellement des clés

La procédure de renouvellement de clés, décrite précédemment dans la section (5.4.4), se déclenche après chaque période T, globalement pour tous les nœuds, dans le but de renforcer la sécurité du réseau. Si pendant cette période T un nœud est compromis, une nouvelle procédure de renouvellement de clés locale va se déclencher pour révoquer les clés partagées avec le nœud compromis et de les renouveler. Deux cas se présentent :

a. Si le nœud compromis est un nœud normal : dans ce cas le nœud CH du cluster supprime de sa mémoire la clé partagée avec lui et la clé commune partagée avec tous les nœuds du réseau et lance une nouvelle procédure de génération d'une nouvelle clé commune pour le cluster. Cette opération engendre un cout faible en calcul et en énergie pour les opérations de calcul des clés publiques et privées et des transmissions de ces nouvelles données cryptographiques.

b. Si le nœud compromis est un nœud CH : après cette détection de compromission, la station de base (SB) lance une procédure locale de renouvellement de clés en désignant un nouveau CH au cluster. Ce dernier régénère l'ensemble des clés publiques et privées nécessaires à l'établissement des clés secrètes et la clé commune. Cette opération est plus couteuse en énergie que dans le précédent cas car elle concerne un plus grand nombre de nœuds et par conséquent des opérations de calcul et de radio plus importantes.

3.5. Insertion des nouveaux nœuds

Le nouveau nœud à insérer sur le réseau est pré-chargé avec une clé publique Px pour établir une clé de session avec son nouveau CH. Cette insertion est acceptée sur la base des informations d'identification reçues par le CH de la station de base avant le déploiement du nœud et cela afin d'éviter un surcout en nombre de communications avec la station de base pour vérifier l'identité du nouveau nœud. Si le CH ne reçoit aucune information sur ce nouveau nœud, il refuse son insertion. Le cout de ces opérations de vérification et d'insertion (calcul de la clé publique du coté du CH et le calcul de la clé de session et de la clé commune) sont négligeables.

4. Conclusion

Dans ce chapitre, nous avons présenté les résultats obtenus avec les simulations effectuées pour l'évaluation des performances de notre solution. Notre solution est comparée avec une autre solution sécurisée, économe en énergie et qui assure une gestion efficace de clés en s'appuyant sur le modèle probabilistes et les schémas de Blom et Blundo et al. Les résultats obtenus ont montré l'intérêt de l'utilisation des méthodes déterministes pour les réseaux de grandes taille et à ressources limitées qui permettent de garantir des communications réduites, une connectivité totale et une consommation énergétique raisonnable. Notre solution s'inscrit dans cette logique de méthodes déterministes permet de résoudre le problème du secret partagé en utilisant la cryptographie sur les courbes elliptiques (ECC) dont les calculs et le choix des paramètres sont optimisés pour les ressources physiques et énergétiques des nœuds capteurs.

La solution que nous avons proposée présente une connectivité totale, une génération réduite du nombre et de la taille des paquets échangés ainsi qu'une consommation minimisée d'énergie comparativement aux autres méthodes.

CONCLUSION

Dans ce travail, nous avons étudié les caractéristiques essentielles et les notions fondamentales des réseaux de capteurs sans fil. Nous avons étudié plus particulièrement les notions de routage, de sécurité et d'énergie par la définition complète des contraintes, des besoins, des défis et des moyens de chacune d'elles mis à la disposition des nœuds capteurs du réseau pour un acheminement correcte, sécurisé et économe en énergie.

Plusieurs protocoles de routage, de sécurité et de gestion de clés sont présentés et plusieurs classifications ont été établies. Nous étions intéressés très particulièrement par les protocoles de routage hiérarchiques pour leur gestion du réseau d'une manière à minimiser l'énergie consommée et le nombre de paquets de données échangés. Une taxonomie d'attaques et de solutions de sécurité sont étudiées. Nous étions également intéressés par les protocoles de gestion de clés utilisant les méthodes de pré-distribution de clés et se basant sur la cryptographie à clés secrètes et nous avons proposé un mécanisme de gestion de clés efficace et peu couteux en énergie et en bande passante, basé sur des clés ECC 160 bits assurant une sécurité égale aux clés RSA de 1024 bits, et un protocole de routage sécurisé, conçus pour des réseaux de capteurs à grande échelle, permettant aux nœuds du réseau d'établir facilement des clés symétriques et de sécuriser leurs échanges d'informations avec moins de paquets échangés et un minimum d'énergie consommée.

Afin de démontrer l'efficacité de notre solution, nous l'avons comparée à une autre solution de sécurité et de gestion de clés TinyKeyMan utilisant le principe des polynômes à deux variables de degré *t* pour calculer et partager un secret commun.

Les résultats obtenus par notre protocole de sécurité et de gestion de clés ont montré l'intérêt de nos méthodes d'établissement de clés et de sécurisation du réseau sur le nombre de paquets échangés et sur l'énergie totale consommée en générant des clés de petite taille équivalentes en niveau de sécurité à des clés de plus grandes taille utilisées par des systèmes traditionnels comme RSA.

Perspectives

La solution que nous avons proposée exploite les optimisations apportées par les concepteurs de la librairie TinyECC afin de rendre l'utilisation de la cryptographie sur les courbes elliptiques possible pour les réseaux de capteurs sans fil. Une perspective de travail pourrait s'intéresser à explorer d'autres techniques d'addition et de doublement des points de la courbe afin de minimiser davantage la complexité des opérations de calcul. Il serait encore intéressant d'expérimenter d'autres solutions de génération de clés autre que celle proposée par Diffie-Hellman afin de comparer les couts en termes de consommation d'énergie et d'espace mémoire.

Pour la solution de routage que nous avons proposée, elle demeure intéressante dans la théorie mais nécessite tout de même des travaux d'expérimentation pour démontrer son application pour les réseaux de capteurs sans fil.

Afin de vérifier la pertinence et la robustesse des deux solutions, nous proposons de les appliquer dans un environnement réel composé de différents types de capteurs existants.

Les contributions de ce mémoire ont apporté des solutions à certains problèmes d'application des RcSF notamment celui lié à l'établissement de clés partagées entre les capteurs après le déploiement. Mais il reste certaines d'autres solutions à explorer et chacune d'elles constitue une perspective possible dans la continuité de ce mémoire.

REFERENCES

1. I. Amadou, G. Chelius, F. Valois, “PFMAC: Routage sans connaissance du voisinage efficace en énergie”. CFIP 2011 – Colloque Francophone sur l’ingénierie des protocoles. 2011.

2. C. B. Abbas, R. González, N. Cardenas, L. J. G. Villalba, “A proposal of a wireless sensor network routing protocol”, Springer Science and Business Media Telecommunication Systems. pp. 61–68. March 2008.

3. Andrews, P. Johnson, D.C, “Remote continuous monitoring in the home. Telemedicine and Telecare”. Vol. 2 (2), pp. 107-113. 2006.

4. M. Achir and L. Ouvry. "A routing protocol for wireless ad-hoc sensor networks : Multi-Path Source Routing Protocol (MPSR)", ICN’05 : 4th International Conference on Networking (IEEE), Ile de la Réunion. Avril 2005.

5. R. Anderson, A. Perrig, "Key infection : Smart trust for smart dust", Unpublished Manuscript. November 2001.

6. M. Ali, S. K. Ravula, “Real-time support and energy efficiency in wireless sensor networks”, Technical report, IDE0805. January 2008.

7. I.F. Akyildiz, W.S. Sankarasubramaniam, E. Cayirci, “Wireless Sensor Network: A Survey”. Computer networks, Vol. 38, pp. 393-422. 2002.

8. I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. l. Cayirci, "A survey on sensor networks", IEEE Communications Magazine, Vol. 40 (8). pp. 102-116. Août 2002.

9. K. Akkaya, M. Younis, “A Survey on Routing Protocols for Wireless Sensor Networks”, Journal of Ad Hoc Networks, Vol. 3 (3). pp. 325-349. May 2005.

10. M. Aissani, "Optimisation du routage dans les réseaux de capteurs pour les applications temps-réel", Thèse en cotutelle de Doctorat en Informatique des universités de Paris-Est et d’USTHB. Mars 2011.

11. D. Braginsky, D. Estrin, “Rumor Routing Algorithm for Sensor Networks”, 1st Workshop. Sensor Networks and Apps., Atlanta, GA. 2002.

12. H. Baldus, K. Klabunde, G. Muesch, “Reliable set-up of medical body-sensor networks”. in EWSN 2004. January 2004.

13. C. Bekara, M. Laurent-Maknavicius, “A new resilient key management protocol for wireless sensor networks”, in WISTP. pp.14–26. 2007.

14. R. V. Biradar, V. C. Patil, S. R. Sawant, R. R. Mudholkar, “Classification and comparison of routing protocols in wireless sensor networks”, Special Issue on Ubiquitous Computing Security Systems. UbiCC Journal. Vol. 4. pp. 704-711. 2009.

15. C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, M. Yung. “Perfectly secure key distribution for dynamic conferences”. In Advances in Cryptology – CRYPTO ’92, LNCS 740. pp. 471–486. 1993.

16. C. Blundo, A. De Santis, U. Vaccaro, A. Herzberg, S. Kutten, M, Yong, “Perfectly secure key distribution for dynamic conferences”, Inf. Comput., Vol. 146 (1). pp.1-23. 1998.

17. R. Blom, “Non-public key distribution”, In R. L. Rivest, A. Sherman, and D. Chaum, editors, Advanced in cryptology Proceedings of Crypto82, New York. pp. 231-236. 1983.

18. K. Beydoun, “Conception d’un protocole de routage hiérarchique pour les réseaux de capteurs”, Thèse de Doctorat. Décembre 2009.

19. D. A. Y. Cheng, “Efficient pairwise key establishment and management in static wireless sensor networks,” in the Second IEEE International Conference on Mobile ad hoc and Sensor Systems. 2005.

20. I. Chlamtac, I. Carreras, H Woesner, “From internets to bionets :biological kinetic service oriented networks”. The case study of Bionetic Sensor networks. CREATE-NET Research Consortium, Trento, Italy. 2005.

21. C. Castelluccia, A. Francillon, “Protéger les réseaux de capteurs sans fil”, In SSTIC08. 2008.

22. C. Chong, Y. Kumar, “Sensor networks: evolution, opportunities, and challenges”. Proceedings of the IEEE, Vol. 91 (8). 2003.

23. J. Chen, R. Lin, Y. Li, Y. Sun, “LQER: A Link Quality Estimation based Routing for Wireless Sensor Networks”, IEEE Sensors, Vol. 8(2). pp. 1025-1038. February 2008.

24. H. Chan, A. Perrig, D. Song, "Random key predistribution schemes for sensor networks", In IEEE Symposium on Security and Privacy, Berkeley, California. pp. 197-213. May 2003.

25. D. Davenport, B. Deb, F. Ross, “Wireless propagation and coexistence of medical body sensor networks for ambulatory patient monitoring”. In Sixth International Workshop on Wearable and Im-plantable Body Sensor Networks. BSN 2009, Berkeley, CA, USA, pp. 41-45. IEEE Computer Society. June 2009.

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

27. J. Deng, R. Han, S. Mishra, “Countermeasures against traffic analysis attacks in wireless sensor networks”, In SECURECOMM ’05 : Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks. pp. 113–126, Washington, DC, USA. 2005.

28. D. Djenouri, L. Khelladi, A. N. Badache, “A Survey of Security Issues in Mobile Ad Hoc and Sensor Networks, Communications Surveys & Tutorials”. IEEE 7, No. 4, pp. 2–28. 2005.

29. DARPA IPTO, “SensIT : Sensor Information Technology program”. 1999.

30. L. Ertaul, W. Lu. “ECC Based Threshold Cryptography for Secure Data Forwarding and Secure Key Exchange in MANET (I)”. In Networking 2005, LNCS 3462. pp. 102-113. 2005.

31. D. Eastlake, P. Jones, "US Secure Hash Algorithm 1 (SHA1)", RFC 3174. 2001.

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

33. T. El Gamal, "A public key cryptosystem and a signature scheme based on discrete logarithms". IEEE Trans. Inform. Theory, Vol. 31, pp. 469-472. 1985.

34. S. Fluhrer, I. Mantin, and A. Shamir. “Weaknesses in the key scheduling algorithm of RC4”. In Proc. 8th Workshop on Selected Areas in Cryptography, LNCS 2259. Springer-Verlag, 2001.

35. N. Gura, A. Patel, A. Wander, H. Eberle, S.C. Shantz. " Comparing elliptic curve cryptography and rsa on 8-bit cpus ". In Cryptographic hardware and embedded systems

CHES 2004 : 6th international workshop, Cambridge, MA, USA, August 11-13, 2004. Vol (6). pp 119. Springer-Verlag New York Inc. 2004.

36. S. Ghiasi et al, “Optimal Energy Aware Clustering in Sensor Networks”, SENSORS Journal, Vol. 2 (7), pp. 258-269. July 2002.

37. W. Heinzelman, A. Chandrakasan and H. Balakrishnan, ”Energy efficient communication protocols for wireless microsensor networks”, in Proceedings of the 33rd Hawaiian International Conference on Systems Science, pp. 3005-3014. January 2000.

38. W. Heinzelman, A. Chandrakasan and H. Balakrishnan, “An application-specific protocol architecture for wireless microsensor networks”, in IEEE Transactions,on Wireless Communications, pp. 660-670. Octobre 2002.

39. W. Heinzelman, J. Kulik, H. Balakrishnan, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks”, 5th ACM/IEEE Mobicom, Seattle, WA. pp. 174–85. 1999.

40. D. Hankerson, A. Menezes, S. Vanstone, “Guide to Elliptic Curve Cryptography”. Springer, 2004.

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

42. Y.C. Hu, A. Perrig, D.B. Johnson, “Rushing attacks and defense in wireless ad hoc network routing protocols”, Proceedings of the 2003 ACM workshop on Wireless security (New York, NY, USA), ACM Press. pp. 30–40. 2003.

43. F. Hu, N. K. Sharma, "Security considerations in adhoc sensor networks". Ad Hoc Networks 3, Elsevier Science, pp. 69–89, 2005.

44. F. Hu, J. Ziobro, J. Tillett, N. Sharma, "Wireless Sensor Networks: Problems and Solutions", Rochester Institute of Technology, Rochester, New York USA.

45. IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs). September 2006.

46. C. Intanagonwiwat, R. Govindan, D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks”, In the Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom'00). Boston, MA. August 2000.

47. T. Izard, "Opérateurs arithmétiques parallèles pour la cryptographie asymétrique". Thèse de doctorat. Décembre 2011

48. G. Jolly, M.C. Kuscu, P. Kokate, M. Younis, "A Low-Energy Key Management Protocol for Wireless Sensor Networks", IEEE Symposium on Computers and Communications (ISCC'03). 2003.

49. H. Krawczyk, M. Bellare, R. Canetti, "HMAC : Keyed-Hashing for Message Authentication", RFC 2104. 1997.

50. B. Krishnamachari, D. Estrin, S. Wicker. "Modelling data-centric routing in wireless sensor networks", IEEE INFOCOM. 2002.

51. R. Kuntz, T. Machiavel, “Accessing the medium in mobile and dense wsn”, In PIMRC’09. 2009.

52. C. Karlof, D. Wagner, “Secure routing in wireless sensor networks : Attacks and countermeasures”, Elsevier's AdHoc Networks Journal, Special Issue on Sensor Network Applications and Protocols. pp. 293-315. September 2003.

53. N. Koblitz, “Elliptic curve cryptosystems”. Mathematics of Computation, 48(177). pp. 203-209. 1987.

54. Levis P., Lee N., “TOSSIM: A Simulator for TinyOS Networks”. pal@cs.berkeley.edu, September 17, 2003.

55. P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, D. Culler : TinyOS: An operating system for Wireless Sensor Networks. In Weber, W., Rabaey, J., Aarts, E., eds.: Ambient Intelligence. Springer-Verlag, New York, NY (2004).

56. D. Liu, P. Ning. “Establishing pairwise keys in distributed sensor networks”. In Proceedings of the 10th ACM Conference on Computer and Communications Security (CCS03). 2003.

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

58. D. Liu, P Ning, “Location-based pairwise key establishments for static sensor networks”. In SASN '03 : Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, New York, NY, USA. pp. 72-82. 2003.

59. D. Liu, P. Ning, W. Du, "Detecting malicious beacon nodes for secure location discovery in wireless sensor networks. In ICDCS, IEEE Computer Society. pp. 609–619. 2005.

60. S. Lindsey, C. Raghavendra, “PEGASIS: Power-Efficient Gathering in Sensor Information Systems”, Proceedings of the IEEE Aerospace Conference. Vol. 3. pp. 1125-1130. Big Sky, MT, USA. March 2002.

61. S. Lindsey, C.S. Raghavendra, K. Sivalingam, “Data gathering in sensor networks using the energy delay metric”, In Proceedings of the IPDPS Workshop on Issues in Wireless Networks and Mobile Computing. April 2001.

62. J. Lu, F. Valois, M. Dohler, “Optimized Data Aggregation in WSNs using Adaptive ARMA”, Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM). 2010.

63. A. Manjeshwar, D. P. Agrawal, “TEEN : A Protocol for Enhanced Efficiency in Wireless Sensor Networks”. 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco, CA. April 2001.

64. A. Manjeshwar, D. P. Agrawal, “APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks”, In the Proceedings of the 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile computing. Ft. Lauderdale, FL. April 2002.

65. D. Martins, H. Guyennet, “Etat de l’art. Sécurité dans les réseaux de capteurs sans fil”. Manuscrit auteur, publié dans SAR-SSI 2008 : 3rd conference on security of network architectures and information systems, France. 2008.

66. N. Mejri, F. Kamoum, “Algorithme de Routage Hiérarchique MHEED à Plusieurs Sauts pour Les Grands Réseaux de Capteurs ”. In SETIT 2007, 4th International Conference:

Sciences of Electronic, Technologies of Information and Telecommunications. TUNISIA. Mars 2007.

67. R. Mehmet, Yuce, Ng. Peng Choong, Y. Jamil. Khan. “Monitoring of physiological parameters from multiple patients using wireless sensor network”. J. Medical Systems, Vol. 32(5), pp. 433-441. 2008.

68. N. Mitton, "Auto-organisation dans les réseaux sans fil multi-sauts à grande échelle", Thèse de Doctorat en Informatique et Réseaux de l’INSA de Lyon, INRIA Rhone Alpes, Lyon, France. Mars 2006.

69. V. Miller, “Use of elliptic curves in cryptography”. In Advances in cryptology – CRYPTO 85, Lecture notes in computer sciences. pp. 417-426. Springer. 1985.

70. N. Méloni, "Arithmétique pour la Cryptographie basée sur les Courbes Elliptiques ". Thèse de doctorat. Septembre 2007.

71. T. Nieberg, S. Dulman, P. Havinga, L. V. Hoesel, J.Wu, "Collaborative algorithms for communication in wireless sensor networks ", Ambient Intelligence : Impact on Embedded Systems, Kluwer Academic Publishers. Octobre 2003.

72. P. Ning and A. Liu. "TinyECC: Elliptic Curve Cryptography for Sensor Networks". 2008. Available on:http://discovery.csc.ncsu.edu/software/TinyECC/

73. National Institute of Standards and Technology, “Data Encryption Standard”. In FIPS Publication 46-2. 1993.

74. Nist Publication, “The Advanced Encryption Standard (AES)”. 2001.

75. Nist Publication. Computer Data Authentication, 1985.

76. S. Pohlig, M. Hellman, “An improved algorithm for computing logarithms over GF(p) and its cryptographic significance”. In IEEE Trans. Information Theory. Vol (24). pp 106-110. 1978.

77. E.M. Petriu, N.D. Georganas, D.C. Petriu, D. Makrakis, V.Z. Groza. “Sensor-based information appliances”. IEEE Instrumentation Measurement Magazine. Vol. 3 (4), pp. 31-35. December 2000.

78. B. Parno, A. Perrig, V. D. Gligor. “Distributed detection of node replication attacks in sensor networks”. In IEEE Symposium on Security and Privacy, IEEE Computer Society. pp. 49–63. 2005.

79. A. Perrig, R. Szewczyk, V. Wen, D. E. Culler, J. D. Tygar, “SPINS : security protocols for sensor networks”. In Mobile Computing and Networking. pp. 189-199. 2001.

80. C. Perkins, “Ad Hoc Networks”. Addison-Wesley, Reading, MA. 2000.

81. J. Pollard. “A monte-carlo method for factorization”. BIT Numerical Mathematics. Vol (15). pp. 331-334. 1975.

82. M. Ramdani, M. Benmohammed, “Un nouveau schéma de gestion de clés basé sur les ECC pour les réseaux de capteurs sans fil", International conference on networking and advanced systems (ICNAS 2013), Annaba, Algeria, June 19-20, 2013.

83. D. Roth, J. Montavont, T. Noel, “MOBINET : gestion de la mobilité à travers différents réseaux de capteurs sans fil", Inria. 2010.

84. V. Rodoplu, T. H. Ming, “Minimum energy mobile wireless networks”, IEEE Journal of Selected Areas in Communications. Vol. 17 (8). pp. 1333-1344. 1999.

85. V. Raghunathan, C. Schurgers, S. Park, M.B. Srivastava, “Energy-Aware Wireless Microsensor Networks”. IEEE Signal Processing Magazine, Vol. 19(2), pp. 40-50. 2002.

86. R. Rivest., "The MD5 Message-Digest Algorithm", RFC 1321.1992.

87. H. Sundani, H. Li, V. Devabhaktuni, M. Alam,P. Bhattacharya, “Wireless Sensor Network Simulators : A Survey and Comparisons”. International Journal Of Computer Networks (IJCN), Vol. 2 (5). 2010.

88. V. Shnayder, M. Hempstead, B. Chen, M. Welsh, “PowerTOSSIM: Efficient power simulation for TinyOS applications”. In ACM Conference on Embedded Networked Sensor Systems (SenSys). 2004.

89. N. Sadagopan, B. Krishnamachari, A. Helmy. "The ACQUIRE mechanism for efficient querying in sensor networks", Proceedings of the First International Workshop on Sensor Network Protocol and Applications, Anchorage, Alaska. pp. 149-155. Mai 2003.

90. N. Singhal, J.P.S. Raina. “Comparative Analysis of AES and RC4 Algorithms for Better Utilization”. In International Journal of Computer Trends and Technology- July to Aug Issue 2011. pp. 177-181. 2011.

91. M. Steiner, G. Tsudik, M. Waidner, “Diffie-Hellman key distribution extended to groups”. In Proceedings of the ACM Conference on Computer and Communications Security. 1996.

92. M. SAXENA, “Security in Wireless Sensor Networks: A Layer based Classification”, Purdue University. 2007.

93. D. Simplot-Ryl. "Some real-time issues in wireless sensor networks", Technical Report, IRCICA/LIFL, Université de Lille 1, CNRS UMR 8022, INRIA Futurs, Ecole d’été Temps Réel. 2005.

94. N. Thepvilojanapong. "A study on data collection and mobility control for wireless sensor networks", PhD Thesis, A Dissertation Submitted to the Department of Information and Communication Engineering,the University of Tokyo. December 2005.

95. X. Wang, W. Gu, K. Schosek, S. Chellappan, D. Xuan, “Sensor network configuration under physical attacks”, In Xicheng Lu andWei Zhao, editors, ICCNMC. Vol. 3619 of Lecture Notes in Computer Science. pp. 23–32. Springer, 2005.

96. D. Xu, J. Huang, J. Dwoskin, M. Chiang, R. Lee, “Re-examining probabilistic versus deterministic key management”, In Information Theory, ISIT 2007, IEEE International Symposium. pp. 2586-2590. 2007.

97. Y. Xu, J. Heidemann, D. Estrin. "Geography-informed energy conservation for ad-hoc routing", Proceedings of the 7th Annual International Conference on Mobile Computing and Networking. pp. 70-84, ACM Press New York, NY, USA. 2001.

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

99. Y. Yu, D. Estrin, R. Govindan, “Geographical and Energy-Aware Routing: A Recursive Data Dissemination Protocol for Wireless Sensor Networks”, UCLA Computer Science Department Technical Report, UCLA-CSD TR-01-0023. May 2001.

100. H. Yang, H. Luo, F. Ye, S. Zhang, L. Lu, “Security in mobile ad hoc networks : Challenges and solutions”, IEEE Wireless Communications. pp. 38-47. 2004.

101. G. Yang, B. Tong, D. Qiao, W. Zhang, “Sensor-aided overlay deployment and relocation for vast-scale sensor networks”, INFOCOM’08. 2008.

102. Y. Zhang, W. Liu, W. Lou, Y. Fang, “Location-Based Compromise-Tolerant Security Mechanisms for wireless Sensor Networks,” IEEE JSAC, special issue on Security in Wireless Ad Hoc Networks, vol. 24 (2), pp. 247. Fevrier 2006.

103. W. Znaidi, M. Minier, J.P. Babau, “An Ontology for Attacks in Wireless Sensor Networks”, Research Report RR-6704, INRIA. 2008.

104. S. Zhu, S. Setia, S. Jajodia, “LEAP: Efficient Security Mechanisms for Large-Scale distributed Sensor Networks”, In Proc of The 10th ACM Conference on Computer and Communications Security (CCS ‟03). pp. 62-72. 2003.

Références Webliographiques

105. fr.wikipedia.org/wiki/Sound_Surveillance_System#Notes_et_r.C3.A9f.C3.A9rences

106. http://www.stccmop.org/CORIE/ (CMOP 2012)

107. http:// www.zigbee.org

108. http://fr.wikipedia.org/wiki/Cat%C3%A9gorie:Attaque_r%C3%A9seau

109. http://fr.wikipedia.org/wiki/Rivest_Shamir_Adleman

APPENDICE A

Généralités sur les courbes elliptiques

1. Définition d'un groupe

Un groupe est un ensemble E non vide muni d'une loi de composition interne associative admettant un élément neutre et un élément symétrique pour chaque élément de l'ensemble E. Une loi de composition interne sur E est une application de (E x E) dans E notée (E x E) $\longrightarrow$ E.

2. Définition d'un corps

Un corps est un anneau commutatif où tous les éléments non nuls sont inversibles (c'est-à-dire ils admettent un élément symétrique) pour la multiplication. Un anneau est un ensemble muni de deux lois de composition appelées multiplication et addition tel que la multiplication est distributive par rapport à l'addition.

3. Définition d'une courbe elliptique

Une courbe elliptique définie sur un corps K est l'ensemble des points de coordonnées (x, y) satisfaisant l'équation cubique de Weierstrass, donnée par $Y^2 + a_1 XY + a_3 Y = X^3 + a_2 X^2 + a_4 X + a_6$ (a_i sont des coefficients dans K), auxquels on ajoute un point à l'infini noté O.

On appelle une courbe elliptique sur R toute courbe plane de l'équation cubique de Weierstrass simplifiée donnée par $Y^2 = X^3 + a X + b$, avec un discriminant $(4 a^3 + 27 b^2)$ non nul.

En cryptologie, les courbes elliptiques sont utilisées dans le corps F_p avec p un nombre premier strictement supérieur à 3 et assez grand pour assurer un niveau de sécurité élevé.

Un exemple d'une courbe elliptique est donné par la figure A.1 ci-dessous.

4. Loi de groupe

L'ensemble des points d'une courbe elliptique peut être muni d'une loi de groupe commutative. Cette loi permet de définir la multiplication d'un point par un scalaire (un

nombre entier), un moyen nécessaire pour introduire le problème du logarithme discret sur les courbes elliptiques.

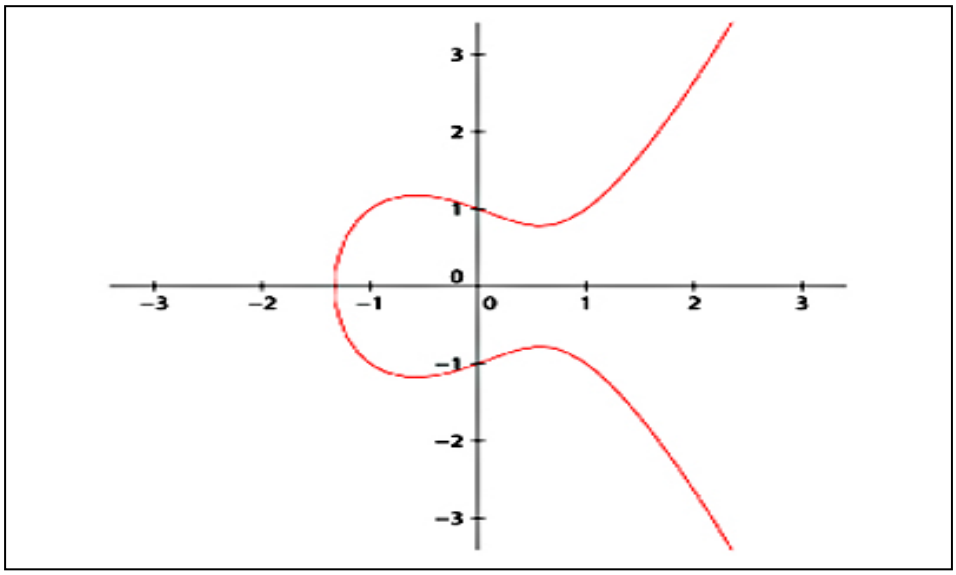

Figure A.1 : Courbe elliptique d'équation $Y^2 = X^3 + X - 1$

Il est possible d'additionner deux points d'une courbe elliptique mais il n'est pas possible de réaliser une multiplication sur eux. Par ailleurs, une succession d'additions permet de définir la multiplication d'un point par un entier. D'un point de vu complexité, temps de traitement et d'espace mémoire, la technique de multiplication par succession d'additions est beaucoup plus intéressante, dans le contexte des réseaux de capteurs, que l'exponentiation modulaire définie sur $(Z / pZ)^*$.

Soient $P = (X_1, Y_1)$ et $Q = (X_2, Y_2)$ deux points d'une courbe elliptique E :

- Le point $(X_1, -Y_1)$ est l'opposé du point P et noté –P.
- Si $P \neq Q$ et $-P \neq Q$ alors le point $R = P + Q$ de coordonnées (X_3, Y_3) est définie par :

$$X_3 = \frac{(Y_1 - Y_2)^2}{(X_1 - X_2)} - (X_1 - X_2) \quad \text{et} \quad Y_3 = Y_1 + \frac{(Y_1 - Y_2)}{(X_1 - X_2)} - (X_3 - X_1)$$

- Si P = Q alors le point $2P = (X_4, Y_4)$ est défini comme suit :

$$X_4 = \frac{(3X_1^2 + a)^2}{(2Y_1)} - 2X_1 \text{ et} \qquad Y_4 = X_1 + \frac{(3X_1^2 + a)}{2Y_1} - (X_3 - X_1)$$

- Si $X_1 = X_2$ et $Y_1 \neq Y_2$ alors R = O.

- Si P = Q et $Y_1 = 0$ alors R = O.

Les deux figures A.2 et A.3 illustre l'addition de deux points P et Q et l'addition d'un point P avec lui-même respectivement.

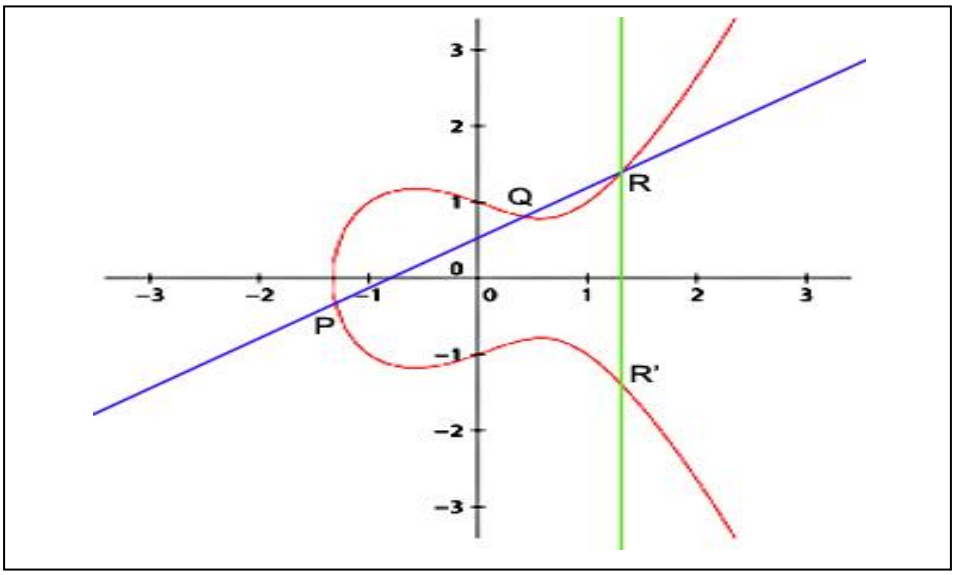

Figure A.2 : Addition de deux points P + Q = R

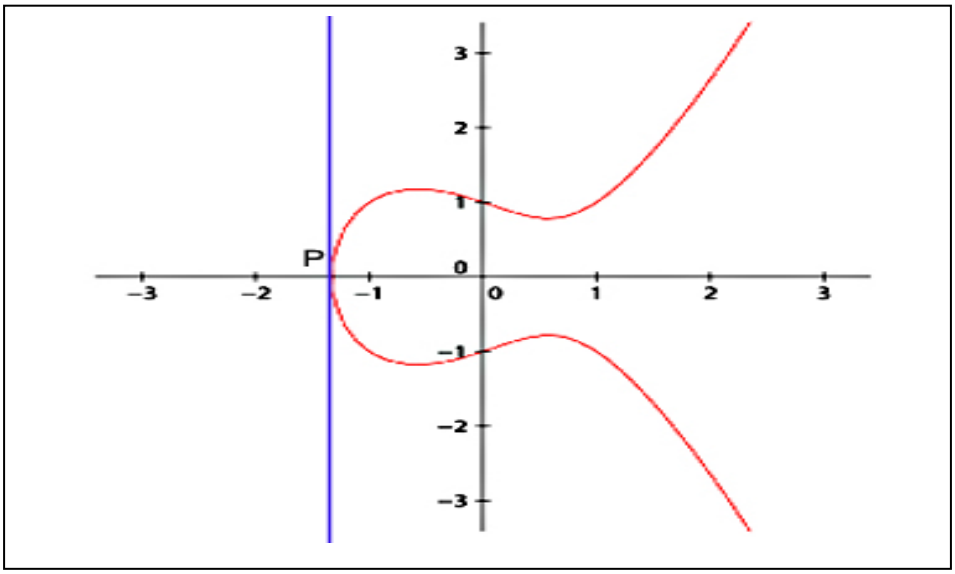

Figure A.3 : Addition d'un point avec lui-même 2P

On démontre l'existence de la loi de groupe sur les points de la courbe elliptique E par la vérification de l'identité P + Q + R = O où O est l'élément neutre noté le point distingué de E.

5. Problème du logarithme discret

Soit E une courbe elliptique définie sur un corps K et soient deux points P et G appartenant à E (K). Le problème du logarithme discret consiste à déterminer un entier n tel que P = nG.

Si K est un corps fini, le problème du logarithme discret est réputé être un problème difficile appartenant à la classe des problèmes NP-Complets non solubles en un temps déterministe. En conséquent, aucun algorithme sous exponentiel n'est connu, pour l'heure, pour le résoudre.

L'algorithme donné par la figure A.4 présente une solution parmi d'autres pour calculer la multiplication d'un point par un scalaire (P = nG).

Algorithme.

Entrée : E : courbe elliptique, K : corps fini, Q, G : deux points appartenant à E(K) et n : un entier.

Sortie : P : un point appartenant à E(K).

Etape1 :

- P = O.
- Si n = 0 alors retourner P. aller à Fin de l'algorithme ;
- Sinon N = n, Q = G
- Fin si.

Etape2 :

- Si n est impair alors P = Q + P
- Fin si.

Etape3 :

- N = (N / 2)
- Si N = 0 alors retourner P. aller à Fin de l'algorithme ;
- Sinon Q = 2Q. aller à l'étape 2.

Figure A.4 : Algorithme de multiplication d'un point par un entier

6. Courbe elliptique sur un corps fini

Soit K = F_p un corps fini à p éléments et E(F_p) une courbe elliptique sur F_p peut être définie par l'équation suivante : $Y^2 \pmod p = X^3 + a X + b \pmod p$ si le discriminant ($4 a^3 + 27 b^2$) est non nul avec a, b, X, Y sont des entiers variables positifs.

L'utilisation de la courbe E(F_p) pour la cryptographie exige que l'équation ci-dessus ($Y^2 \pmod p = X^3 + a X + b \pmod p$) doit être satisfaite avec des valeurs arbitraires de a, b et p.

APPENDICE B

Liste des acronymes

ACK	Acknowledgment
AES	Avanced Encryption standard
CBC	*Cipher* Block Chaining
CBR	Constant Bit Rate
CH	Cluster Head
CSMA	Carrier Sense Multiple Access
CTR	Clear To Send
DARPA	Defense Advanced Research Projects Agency
DES	Data Encryption standard
ECB	Electronic Codebook
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
ECIES	Elliptic Curve Integrated Encryption Scheme
GPS	Global Positioning System
HMAC	keyed-hash message authentication code
ID	Identifiant d'une entité du réseau
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Organization for Standardization
IV	Vecteur initial
LAN	Local Area Network
OFB	Output FeedBack
OSI	*Open Systems Interconnection*
MAC	Message authentication Code
MANET	Mobile Ad hoc NETwork

MD5	Message Digest 5
NS2	Network Simulator version 2
RC4	Rivest Cipher 4
RcSF	Réseaux de Capteurs Sans Fil
RSA	Rivest Shamir Adleman
SB	Station de Base
SDI	Système de Détection d'Intrusion
SHA	Secure Hash Algorithm