Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

THESE

Présentée

A L'UNIVERSITE DE TLEMCEN
FACULTE DES SCIENCES

Pour l'obtention du diplôme de

DOCTORAT

Spécialité : " Informatique"

Par

Mme LABRAOUI Nabila

LA SÉCURITÉ DANS LES RÉSEAUX SANS FIL AD HOC

Soutenue en 2012 devant le Jury:

M.A. CHIKH	Maîtres de Conférences à l'Université de Tlemcen	Président
A. BALLA	Professeur à l'Ecole Supérieure en Informatique	Examinateur
M. FEHAM	Professeur à l'Université de Tlemcen	Examinateur
L. SEKHRI	Maître de Conférences à l'Université d'Oran	Examinateur
M. ALIOUAT	Maître de Conférences à l'Université de Sétif	Directeur de Thèse
M. GUEROUI	Maître de Conférences à l'Université de Versailles, France	Co-encadrant

Résumé

Depuis leur création, les réseaux de communication sans fil ont connu un engouement fulgurant qui ne cesse de croître au sein des communautés scientifiques et industrielles, permettant aux utilisateurs un accès à l'information et aux services électroniques, indépendamment de leur position géographique. Ainsi, le paradigme sans fil a vu naître, au cours de son évolution, diverses architectures dérivées, telles que les réseaux cellulaires, les réseaux locaux sans fil, les réseaux WiMax, etc. Durant la dernière décennie, un nouveau type de réseau sans fil a suscité un grand intérêt auprès de la communauté scientifique, il s'agit des réseaux ad hoc et des réseaux de capteurs sans fil (WSN). Les réseaux de capteurs sans fil sont un type particulier de réseaux ad-hoc dédiés à une application spécifique. Ils constituent un nouveau domaine de recherche qui s'est créé pour offrir des solutions économiquement intéressantes et facilement déployables à la surveillance à distance et au traitement des données dans les environnements complexes et distribués. Les WSN sont constitués de nœuds déployés en grand nombre en vue de collecter et transmettre des données environnementales vers un ou plusieurs points de collecte, d'une manière autonome. Ces réseaux ont un intérêt particulier pour les applications militaires, environnementales, domotiques, médicales, et bien sûr les applications liées à la surveillance des infrastructures critiques.

Dans cette thèse, nous nous somme intéressés au problème de sécurité dans les réseaux de capteurs sans fil. Nous avons abordé cette problématique en investiguant deux axes de recherche, à savoir la sécurité des données agrégées et la sécurité de la localisation. Notre première contribution consiste à proposer un algorithme robuste nommé RAHIM pour sécuriser l'agrégation des données dans les réseaux de capteurs à architecture hiérarchique. Alors que notre seconde contribution consiste à proposer un autre protocole de défense nommé WFDV pour sécuriser l'algorithme de localisation DV-Hop contre l'attaque critique wormhole. Plusieurs simulations ont été effectuées pour démontrer la faisabilité et l'efficacité de nos solutions dans un environnement hostile et sans surveillance.

Mots clés : Réseaux de capteurs sans fil (WSN), sécurité des WSN, sécurité des données agrégées, sécurité de la localisation.

Abstract

Since their creation, wireless communication networks have witnessed a huge success that continues to grow within scientific and industrial communities, allowing the users an access to the information and to the electronic services, independently of their geographical position. So, during its evolution, the wireless paradigm has given birth to various derivative architectures, such as cellular, WiMax and wireless local area networks. During the last decade, a new type of wireless networks has drawn growing attention from the scientific community; it consists in ad hoc networks and wireless sensor networks (WSN). The wireless sensor networks are a particular type of ad hoc networks dedicated to a specific application. They constitute an emerging research area which can offer solutions economically interesting and easily deployed to the remote monitoring and the data processing in the complex and distributed environments. A WSN consists of a large number of embedded processing units, called sensors, whose main function is the collection and the transmission of parameters related to the surrounding environment, to one or many monitoring nodes. These networks have a particular interest for the military applications, environmental, habitat, medical and applications related to the monitoring of the critical infrastructures.

In this thesis, we focus on security issues in WSN. We address these problems by exploring two research areas: secure data aggregation and secure localization. Our first contribution consists in proposing a robust algorithm named RAHIM to secure data aggregation in cluster-based sensor networks. Whereas our second contribution consists in proposing another defense mechanism named WFDV to secure the DV-Hop localization algorithm against wormhole attacks. Intensive simulations were carried out to assess the practicality and the effectiveness of our proposed solutions in a hostile and unattended environment.

Keywords: Wireless sensor networks (WSN), WSN security, secure data aggregation, secure localization.

Abstract (arabe)

A mon cher mari,

A mes petits poussins et ma raison de vivre :

Lilia, Redouane et Rafik.

Je vous aime !

Remerciements

Avant tout, le grand et le vrai merci à Allah qui m'a donné la force et la vie pour accomplir cette tâche, qui au début paraissait une mission difficile.

Je tiens à remercier en premier lieu Mr. Makhlouf Aliouat, mon directeur de thèse pour son encouragement, son expérience, ses conseils et sa sympathie qui m'ont permis de mener à bien cette thèse.

Je suis très reconnaissante envers Mr Mourad Gueroui, mon co-encadrant, pour sa disponibilité malgré la distance qui nous sépare, son ouverture d'esprit, sa générosité et les nombreuses discussions fructueuses qui ont animé ces quatre années de thèse. Je le remercie également de m'avoir accueillie au sein du laboratoire PRISM à l'université de Versailles.

Je suis très honorée par la présence de Mr Mohamed Amine Chikh, qui a accepté de présider le jury de ma thèse, je suis également très honorée par la présence de Mr Amar Balla, Mr Larbi Sekhri, et Mr Mohamed Feham qui ont accepté d'être les rapporteurs de cette thèse. Qu'ils trouvent ici mes plus vifs remerciements pour l'effort qu'ils ont fait pour lire mon manuscrit et l'intérêt qu'ils ont porté à mon travail.

Ce travail a été réalisé au sein du laboratoire Systèmes et Technologie de l'Information et de la Communication (STIC) de l'université de Tlemcen. Je tiens donc à remercier encore une fois le directeur du laboratoire Mr Feham de m'avoir donné l'opportunité d'effectuer mes travaux dans de bonnes conditions.

Je remercie ma mère de m'avoir donné l'amour de la lecture dès ma tendre enfance. Je remercie chaleureusement mon père pour tous ses encouragements et pour m'avoir soigneusement corrigé ce mémoire. Merci aussi à tous les membres de ma famille d'avoir cru en moi.

Je réserve une pensée particulière à mon mari Mohamed pour avoir toujours su trouver les mots qu'il faut dans les meilleurs comme dans les pires moments et pour m'avoir soutenue, encouragée et supportée durant ces quatre années de thèse. « *Ton aide, ta générosité, ton soutien ont été pour moi une source de courage et de confiance. Qu'il me soit permis aujourd'hui de t'assurer mon profond amour et ma grande reconnaissance* ».

Table des matières

Deuxième partie:

Les contributions à la recherche

Liste des illustrations

Liste des tableaux

INTRODUCTION GENERALE

1. Contexte générale

Incontestablement, ce début de vingt et unième siècle est placé sous le signe de *technologie de l'information et de la communication*. Après le phénomène Internet, la démocratisation des technologies sans fil révolutionne les moyens de communication avec notamment l'apparition de réseaux spontanés ou réseaux ad hoc. En effet, les premières retombées révolutionnaires de ces progrès ont été les ordinateurs personnels, suivis ensuite par les ordinateurs portables, les assistants personnels et finalement les téléphones portables. Actuellement, les micro-capteurs constituent le dernier maillon de cette chaîne d'évolution qui sera sans doute suivie par les produits de la nanotechnologie dans le futur.

Le paradigme de l'informatique pervasive est devenu désormais une réalité et a connu un essor remarquable ces dernières années en s'imposant progressivement et sûrement dans notre quotidien. La convergence des progrès réalisés d'une part dans le domaine de la microélectronique et d'autre part dans les technologies de communication sans fil a permis de produire à coût raisonnable des capteurs communicants peu consommateurs en énergie. Ces petites entités électroniques, dont l'objectif est de récolter des grandeurs physiques de leur environnement proche (luminosité, mouvement, température, pression barométrique, etc.), et éventuellement de les traiter, constituent les briques de base des réseaux de capteurs sans fil et ont été à l'origine de leur émergence.

Les réseaux de capteurs sans fil sont l'une des technologies visant à résoudre les problèmes de cette nouvelle ère de l'informatique embarquée et omniprésente. Cette nouvelle technologie promet de révolutionner notre façon de vivre, de travailler et d'interagir avec l'environnement physique qui nous entoure. Des capteurs communicants sans fil et dotés de capacités de calcul facilitent une série d'applications irréalisables ou trop chères il y a quelques années. Aujourd'hui, des capteurs minuscules et bon marché peuvent être littéralement éparpillés sur des routes, des structures, des murs ou des machines, créant ainsi une sorte de « seconde peau numérique » capable de détecter une variété de phénomènes physiques et biologiques.

La diminution des coûts matériels, ainsi que l'élargissement de la gamme des capteurs disponibles, a permis d'étendre le champ d'application des réseaux de capteurs sans fil. Le

domaine militaire a été un moteur initial dans le développement de ces technologies pour l'analyse de terrains dangereux ou la surveillance de mouvements. Les applications environnementales se sont ensuite multipliées, pour la détection de feux de forêts, la surveillance d'activité volcanique ou sismique, ou encore le suivi du déplacement d'animaux. On utilise aussi les réseaux de capteurs pour des applications médicales comme la veille épidémiologique, ou dans un but commercial, pour l'optimisation des processus de stockage, ou encore dans l'agriculture de précision, et la construction de maisons intelligentes.

Dans beaucoup d'applications des réseaux de capteurs, les données peuvent être menacées par des événements extérieurs qui ne devraient pas arriver au cours du fonctionnement normal du réseau. En particulier, la confidentialité, l'intégrité et la disponibilité des données sont des fonctionnalités importantes que le réseau devrait pouvoir assurer. Garantir de telles caractéristiques est une tâche difficile à cause des caractéristiques spécifiques des réseaux de capteurs, à savoir, absence d'infrastructure, contrainte d'énergie, topologie dynamique, nombre important de capteurs, sécurité physique limitée, capacité réduite des nœuds. Le cas échéant, utiliser des protections physiques est, dans beaucoup de situations, quasiment impraticable. Déployés dans des environnements ouverts et hostiles, les réseaux de capteurs sans fil sont donc sujets à différents types de menaces et d'attaques telles que l'interception des données envoyées/reçues par le support sans fil et par la suite la possibilité de modifier et de rejouer les données. L'intrus peut également injecter, saturer ou endommager les équipements du réseau. Dans des applications critiques, de telles attaques peuvent être néfastes et peuvent engendrer des dégâts économiques et sécuritaires majeurs

Malheureusement, malgré la diversité des applications des réseaux de capteurs, leur succès dépend de leur propre sécurité. Leurs vulnérabilités face aux attaques malicieuses constituent leurs inconvénients majeurs freinant leur prolifération et se trouvent désormais à la hauteur de leurs promesses. En effet, les nœuds capteurs sont soumis à une forte contrainte de consommation d'énergie en raison de leurs dimensions très réduites ainsi qu'à l'environnement de déploiement. D'un autre côté, assurer des services de sécurité pour ces applications nécessite un surcoût en termes de consommation d'énergie. En fait, la consommation d'énergie des capteurs joue un rôle important dans la durée de vie du réseau qui est devenue le critère de performance prédominant dans ce domaine. De ce fait la sécurisation des réseaux de capteurs est à la source, aujourd'hui, de beaucoup de défis scientifiques et techniques. Un des enjeux principaux est donc de pouvoir trouver des solutions de sécurité pour les réseaux de capteurs adaptées à leurs caractéristiques

spécifiques. Dans cette optique, un protocole de sécurité doit pouvoir établir des sessions sécurisées avec peu d'influence sur la performance globale du réseau, tout en fournissant les différents services de sécurité pour chaque type d'application.

2. Contributions de cette thèse

Plusieurs travaux de recherches ont été menés pour résoudre les problèmes de sécurité liés aux réseaux de capteurs sans fil, tels que : l'établissement de clés de paires entre capteurs, la sécurité de l'agrégation de données, l'authentification d'une source de diffusion, la sécurité du routage et de la localisation, ainsi que le contrôle d'accès au réseau de capteur sans fil.

Nous nous sommes intéressés dans nos travaux à deux axes de recherche, à savoir, la sécurité des données agrégées et la sécurité de la localisation. Étant donné les perspectives applicatives prometteuses des réseaux de capteurs ainsi que la problématique de sécurité soulevée, l'objectif de la thèse consiste à étudier et à proposer de nouveaux algorithmes de sécurité permettant d'apporter des réponses aux deux problématiques notamment l'agrégation de donnée et la localisation.

Nous résumons ci-dessous les contributions présentées dans cette thèse.

- **La sécurité des données agrégées** : Une approche courante pour surmonter les limitations des réseaux de capteurs est d'agréger les données au niveau des nœuds intermédiaires**.** Ainsi les lectures individuelles de chaque capteur sont remplacées par une vue globale collaborative sur une zone donnée. L'agrégation des données dans les réseaux de capteur est relativement triviale, mais devient problématique lorsque l'on veut y ajouter de la sécurité. En effet, garantir la sécurité conjointement à des techniques d'agrégation est difficile parce qu'un nœud capturé pose un double problème. Il compromet la confidentialité des données (possibilité d'écoute) et leur disponibilité (possibilité d'attaque du type déni de service). Egalement, un nœud d'agrégation compromis met en danger l'intégrité de toutes les mesures qui font parti de l'agrégat dont le nœud est responsable. Dans notre travail, nous ne considérons pas la confidentialité des données comme un problème de premier plan. A la place, nous nous concentrons sur l'intégrité, pour lequel nous pensons qu'il manque des solutions efficaces. Notre étude effectuée dans cette problématique a décelé deux inconvénients principaux des solutions existantes focalisant sur l'intégrité des données de l'agrégation, à savoir, le *surcoût excessif* et *le rejet total des données*. A cette fin, nous proposons, un nouvel algorithme nommé

RAHIM (**R**obuste **A**daptive approach based on **HI**erarchical **M**onitoring) pour apporter des solutions à ces problèmes et améliorer la fiabilité et la disponibilité des données agrégées dans les réseaux de capteurs clustérisés.

- **La sécurité de la localisation** : La localisation est devenue une information à grande valeur ajoutée, que ce soit d'un point de vue économique ou militaire. Dans bon nombre d'applications des réseaux de capteurs, un événement détecté par un capteur n'est utile que si une information relative à sa localisation géographique est fournie. Cette information s'avère donc primordiale. C'est le cas de la surveillance des feux de forêt ou de troupes ennemies dans un contexte militaire. Sans cette information, ces applications n'auraient aucun sens. Il s'agit donc de déterminer pour chacun des capteurs sa position. Pour ces multiples raisons, il nous a semblé intéressant d'étudier plus en avant les méthodes permettant aux nœuds simples d'estimer leur position en fonction de leur environnement radio. La localisation est un domaine de recherche dont l'attrait est croissant ces dernières années et de nombreuses propositions ont été faites. Toutefois aucune d'entre elles nous paraissent robustes contre les manipulations abusives d'un adversaire malveillant, dont le but initial est de fausser les estimations de position des nœuds. Après avoir fait une étude sur les différents algorithmes de localisation ainsi que leur vulnérabilité envers certaines attaques, nous nous sommes focalisé sur la sécurisation de l'algorithme DV-Hop contre l'attaque wormhole. Nous avons choisi DV-Hop comme algorithme à cause de sa simplicité et de son coût bas en terme énergétique, quant au choix de l'attaque wormhole, il a été basé sur le fait que cette attaque est particulièrement sévère et difficile à détecter, car elle peut être déclenchée sans compromettre aucun nœud du réseau et sans avoir accès à aucune clé cryptographique. Il est donc clair qu'une solution qui dépend uniquement sur des techniques cryptographiques n'est pas suffisante pour s'immuniser contre l'attaque wormhole. Pour cela nous avons proposé WFDV (**W**ormhole-**F**ree **DV**-Hop localization scheme), un protocole de défense qui met en place une contre mesure proactive à l'algorithme de base DV-Hop.

3. Organisation du manuscrit

Ce manuscrit est organisé en cinq chapitres suivis d'une conclusion générale. Le positionnement de nos travaux est présenté sur les trois premiers chapitres et nos contributions sont détaillées dans les deux derniers.

Dans le premier chapitre, nous présentons les concepts de sécurité dans les réseaux de capteurs sans fil. Nous commençons d'abord par la définition des différents concepts gravitant autour de cette thématique, ensuite nous exposons les problèmes de sécurité dans les réseaux de capteurs sans fil, en mettant le point sur les vulnérabilités et les caractéristiques intrinsèques des réseaux de capteurs sans fil. Et enfin nous présentons une taxonomie des attaques et discutons les besoins de sécurité requis par les protocoles.

Dans le deuxième chapitre, nous poursuivons notre étude bibliographique sur le premier axe de recherche, à savoir, la sécurité de l'agrégation des données. Nous abordons tous les concepts et les challenges rencontrés lors de la conception d'un protocole sécurisé, et nous survolons les travaux existants qu'on a pu recenser dans la littérature, en effectuant une analyse profonde et une classification, ainsi qu'une évaluation et une comparaison entre les différentes solutions.

Dans le troisième chapitre, nous présentons l'investigation de notre deuxième axe de recherche, à savoir la problématique de la sécurité de la localisation dans les réseaux de capteurs sans fil. Tout d'abord, nous expliquons le principe des systèmes de localisation en général, ensuite nous abordons la localisation dans les réseaux de capteurs et la problématique de sécurité dans ce domaine. Nous présenterons les types d'attaques spécifiques au processus de localisation ainsi qu'un survol des travaux existants pour sécuriser la localisation.

Dans le quatrième chapitre, nous présentons notre première contribution liée à la sécurisation des données agrégées. Nous présentons les détails de notre protocole RAHIM, et nous effectuons une analyse de sécurité et une évaluation de performances ainsi qu'une comparaison par rapport à d'autres protocoles.

Dans le cinquième chapitre, nous présentons notre deuxième contribution liée à la sécurisation de la localisation. Notre attention porte sur la sécurisation de l'algorithme de localisation DV-Hop contre l'attaque wormhole. Nous effectuons une analyse profonde de sécurité et plusieurs simulations afin de consolider nos résultats et prouver l'efficacité de notre protocole nommé WFDV.

Nous finalisons ce manuscrit par une conclusion générale. Nous rappelons les différentes contributions réalisées tout au long de ce travail de recherche et nous présentons les perspectives de recherche de ce travail. Les travaux contenus dans ce document ont été publiés dans plusieurs conférences et journaux à facteur d'impact dont la liste est disponible dans la liste des publications.

PREMIERE PARTIE :

REVUE DE LITTÉRATURE SUR LA SÉCURITÉ DES RÉSEAUX DE CAPTEURS SANS FIL

On ne connaît pas complètement une science tant qu'on n'en sait pas l'histoire.

Auguste COMTE

Chapitre I

Concepts généraux de sécurité dans les réseaux de capteurs

SOMMAIRE

1. INTRODUCTION

Un capteur est un dispositif qui perçoit une propriété physique et qui mappe la valeur à une mesure quantitative (Gortz, et al., 2004). Après numérisation, cette dernière pourra être mémorisée, traitée, transmise pour être exploitée avec d'autres informations. On trouve par exemple des capteurs de position, de vitesse, d'accélération, de pression, de mouvement, de luminosité, et de température, pour n'en citer que quelques uns parmi les plus simples. Des capteurs plus complexes, comme des capteurs de son ou d'images sont aussi très largement utilisés. Dans les usines modernes d'aujourd'hui, les systèmes de production sont munis de capteurs qui surveillent et sécurisent les processus de fabrication. On y trouve par exemple des capteurs qui indiquent la position des matières premières, l'état des machines et la qualité du produit final, entre autres. Dans les voitures, on trouve des détecteurs de présence de passagers, d'ouverture des portes et de position (GPS).

Avant la révolution des télécommunications et le développement des technologies sans fil, l'acheminement de l'information relevée par un capteur se faisait par un système de câblage coûteux, encombrant et nécessitant la mobilisation d'efforts humains relativement importants. Le spectre d'utilisation des capteurs restait très limité. Pour justifier le déploiement d'un réseau de capteurs, il fallait un très grand enjeu sécuritaire ou des perspectives de profits économiques importants.

À présent, l'intégration de capteurs et de communications sans fil a conduit à la naissance d'une nouvelle gamme de dispositifs électroniques ouvrant la voie à de nouvelles applications basées sur des capteurs sans fil, dotés de circuits « radio » leur permettant de transmettre et de recevoir de l'information sans avoir besoin de connexions filaires rigides. De plus, ces capteurs sans fil disposent de capacité de mémorisation et d'une puissance de calcul permettant de réaliser le routage et l'acheminement des paquets d'informations. Ainsi, des réseaux de capteurs sans fil peuvent se former et s'auto-configurer de manière ad hoc sans infrastructure.

En conséquence, un grand nombre d'applications ont pu se développer en tirant profit de ce nouvel environnement de capteurs, et il y en aura certainement beaucoup d'autres dans le futur proche. Ces applications sont regroupées sous le terme de réseaux de capteurs ou sous le sigle WSN (« Wireless Sensor Networks »). Ils constituent un domaine en plein expansion et ayant des contraintes relativement différentes des systèmes de communications sans fil classiques.

De très nombreux états de l'art ont été proposés (Akyildiz, et al., 2002), (Holger, et al., 2005), (Isaac, et al., 2011), nous allons donc retracer dans le présent chapitre le fonctionnement général des réseaux de capteurs, le centre d'intérêt de cette thèse, en nous focalisant sur le problème de sécurité en général dans les WSN.

2. LES RESEAUX DE CAPTEURS SANS FIL

Un réseau de capteurs sans fil (WSN) (Akyildiz, et al., 2002) est un système distribué de grande échelle mettant en communication un grand nombre d'entités autonomes communément appelées « capteurs sans fil », ou simplement « capteurs ». Ces capteurs forment donc les nœuds du réseau. Dans un scénario d'application classique, plusieurs nœuds capteurs sont déployés dans une zone géographique appelée zone de captage afin de surveiller un phénomène et récolter ses données d'une manière autonome. Les nœuds capteurs utilisent une communication sans fil pour acheminer les données captées avec un routage multi-sauts vers un nœud collecteur appelé nœud puits (ou sink) qui va transmettre, via internet ou satellite, ces informations à l'utilisateur du réseau (voir Figure 1.1). Ainsi, l'usager peut adresser des requêtes aux autres nœuds du réseau, précisant le type de données requises, puis récolter les données environnementales captées par le biais du nœud collecteur.

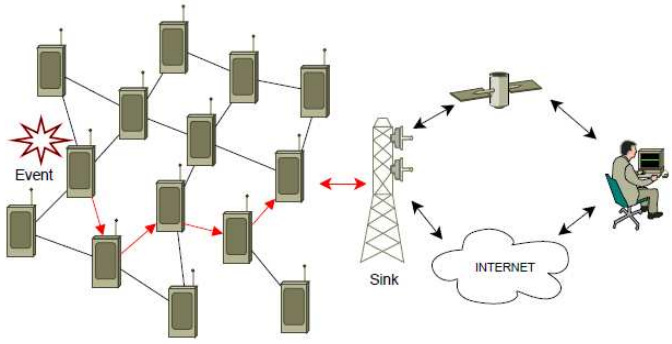


Figure 1. 1 : Architecture d'un réseau de capteurs sans fil.

2.1 Les composants d'un capteur sans fil

Les capteurs sans fil sont conçus comme de véritables systèmes embarqués, dotés de moyens de traitement et de communication de l'information, en plus de leur fonction initiale de relever des mesures. Ils représentent une révolution technologique des instruments de mesure, issue de la convergence des systèmes électroniques miniaturisés et des systèmes de communication sans fil. L'architecture d'un nœud est complètement dépendante de l'objectif

de son déploiement. Néanmoins, un capteur sans fil est composé fondamentalement de quatre unités élémentaires :

1. **Unité d'acquisition de données** : Ce composant est l'unité qui contient le ou les capteurs embarqués sur le nœud. Habituellement, un convertisseur analogique-numérique (CAN) convertit les signaux provenant des capteurs (signaux analogiques) en signaux interprétables par l'Unité de Traitement (signaux numériques).
2. **Unité de traitement des données** : Elle est généralement constituée d'un microcontrôleur dédié et de la mémoire. Les microcontrôleurs utilisés dans le cadre de réseaux de capteurs sont à faible consommation d'énergie. Leurs fréquences sont assez faibles, moins de 10 MHz pour une consommation de l'ordre de 1 mW. Une autre caractéristique est la taille de leur mémoire qui est de l'ordre de 10 Ko de RAM pour les données et de 10 Ko de ROM pour les programmes (Holger, et al., 2005). Cette mémoire consomme la majeure partie de l'énergie allouée au microcontrôleur, c'est pourquoi on lui adjoint souvent de la mémoire flash moins coûteuse en énergie. Outre le traitement des données, le microcontrôleur commande également toutes les autres unités notamment le système de transmission.
3. **Unité de transmission de données** : Elle est le plus souvent constituée d'un transmetteur radio qui fournit au capteur la capacité de communiquer avec les autres au sein d'un réseau. Les composants utilisés pour réaliser la transmission sont des composants classiques. Ainsi on retrouve les mêmes problèmes que dans tous les réseaux sans fil : la quantité d'énergie nécessaire à la transmission augmente avec la distance. Pour les réseaux sans fil classiques (LAN, GSM) la consommation d'énergie est de l'ordre de plusieurs centaines de milliwatts, et on se repose sur une infrastructure alors que pour les réseaux de capteurs, le système de transmission consomme environ 20 mW et possède une portée de quelques dizaines de mètres. Certaines technologies radio permettent de changer la fréquence et la puissance de transmission.
4. **Unité de puissance** : Comme il est souhaitable de s'affranchir de toute connexion par câble, le capteur doit disposer de sa propre source d'énergie qui est responsable de répartir l'énergie disponible aux autres modules et de réduire les dépenses en mettant en veille les composants inactifs par exemple. Cette unité se trouve généralement sous la forme de batteries standards de basse tension. A titre indicatif, ce sera souvent une pile AA normale d'environ 2.2-2.5 *Ah* fonctionnant à 1.5 V (Holger, et al., 2005).

En fonction des applications pour lesquelles ils sont conçus, les capteurs sans fil pourraient également avoir d'autres modules, comme *une unité de localisation*, afin d'identifier leur position géographique, par exemple en utilisant un récepteur GPS ou une technique de triangulation. Certaines applications pourraient aussi avoir besoin de capteurs équipés d'*un mobilisateur* pour qu'ils puissent se déplacer.

2.2 Les plates-formes existantes

Comme un certain nombre de technologies connues à ce jour, les nœuds de capteurs sans fil doivent être nés d'un projet militaire, ce qui entrave la mise en place d'une chronographie précise de leur développement. Cependant, le titre du premier prototype de nœuds de capteurs sans fil identifiable dans la bibliographie correspond sans aucun doute au module LWIM (Low-power Wireless Integrated Microsensors) développé dans le milieu des années 90 par l'Agence pour les Projets de Recherche Avancée de Défense (DARPA) des Etats-Unis et l'UCLA. Il s'agissait d'un géophone équipé d'un capteur de transmission radio fréquences et d'un contrôleur PIC. Depuis un peu plus de 10 ans, la technologie des capteurs sans fil a beaucoup évolué. Les modules deviennent de plus en plus petits et les durées de vie prévues augmentent. Aujourd'hui, le marché de nœuds a été ouvert à l'industrie. Le fournisseur le plus connu est *Crossbow Inc*., avec son offre de capteurs Mica2 et MicaZ (voir Figure 1.2).

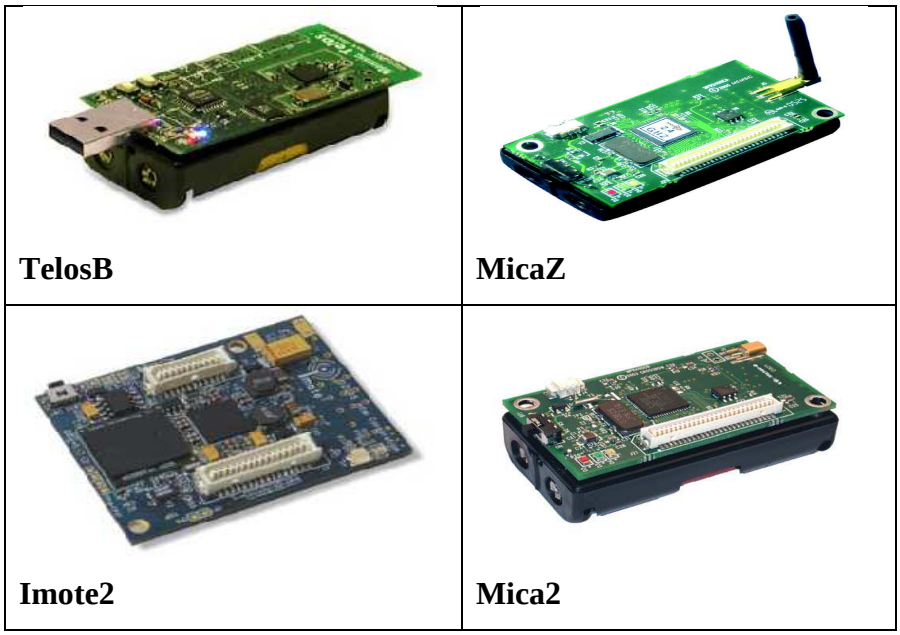


Figure 1. 2 : Quelques modèles de capteurs sans fil.

Le Tableau 1.1 recense les différents composants actuellement disponibles sur le marché. Parmi les modèles les plus courants, on trouve les capteurs MICA développés par l'université de Berkeley et commercialisés par Crossbow, les capteurs Imote commercialisés par Crossbow ainsi que les capteurs TinyNode développés, pour des applications réelles liées à l'industrie, par la compagnie Shockfish SA. Notons que bien qu'ils soient différents, ces modèles ont en commun les mêmes composants de base.

	Unité de traitement				**Unité de transmission**	**Unité de captage**	**Unité de puissance**
Modèle	**Micro-contrôleur**	**RAM**	**Flash**	**EEprom**	**Type radio**		
MICA2 (Crossbow)	**ATmega 128L**	**4 KB**	**128KB**	**4 KB**	**Chipcon CC1000** 38kbps	**Connecteur pour carte de capteurs externe**	2xAA
MICAZ (Crossbow)	**ATmega 128L**	**4 KB**	**128KB**	**4 KB**	**Chipcon CC2420** 250kbps	**Connecteur pour carte de capteurs externe**	2xAA
Imote2 (Crossbow)	**Intel PXA271**	**256KB**	**32 KB**	**32KB**	**Chipcon CC2420** 250kbps	**Connecteur pour carte de capteurs externe**	3xAAA
TeloSB (Crossbow)	**TI MSP 430**	**10KB**	**48KB**	**16KB**	**Chipcon CC2420** 250kbps	**Connecteur pour carte de capteurs externe**	2xAA
TinyNode (Shockfish SA)	**TI MSP 430**	**10KB**	**48KB**	**16KB**	**Semtech XE 1205** 153kbps	**Connecteur pour carte de capteurs externe**	2/3xAA
BTnode3 (ETH)	**ATmega 128L**	**64KB**	**128KB**	**4KB**	**Chipcon CC1000/Bluetooth**	**Connecteur pour carte de capteurs externe**	2xAA
Tmote Sky (Moteiv)	**TI MSP 430**	**10KB**	**48KB**	**128KB**	**Chipcon CC2420**	**Connecteur pour carte de capteurs externe**	2xAA

Tableau 1. 1 : Caractéristiques de capteurs existants actuellement.

Le concept prévalent dans le développement de nœuds de capteurs est la conception modulaire. En effet, tous les nœuds du Tableau 1.1 sont en fait des *cartes intégrées* qui regroupent l'unité de communication et l'unité de traitement, tandis que l'unité de captage est conçue comme une *carte distincte* qui peut être attachée sur l'unité principale. Cela permet bien sûr de pouvoir réutiliser les mêmes unités pour différentes applications. Par exemple, un nœud Mica2 peut être combiné avec une carte MTS310 qui comprend un capteur de température, un capteur lumière, un capteur de son, un capteur de champ magnétique, et un accéléromètre à deux axes. De même, nous pouvons combiner le nœud Mica2 avec une carte

MTS420 pour le doter d'un capteur d'humidité et d'un capteur de pression barométrique, et même d'un GPS pour le positionnement géographique. Une autre possibilité pour la même unité est l'ajout d'une carte d'acquisition MDA320. La Figure 1.3 et le tableau 1.2 montrent les cartes intégrées pouvant être combinées avec un mote.

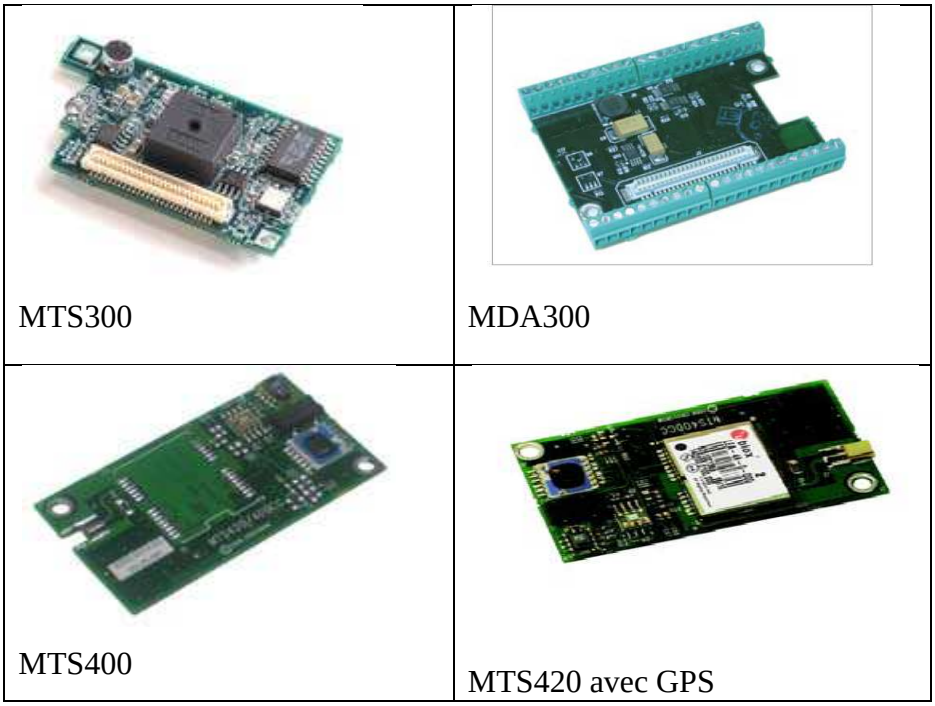


Figure 1. 3 : Quelques cartes De captage.

La plupart des fabricants adoptent des émetteurs RF à basse fréquence. Certains ont choisi de mettre en œuvre un protocole d'origine récente conçu pour les modules sans fil industriels et spécifié dans la norme IEEE 802.15.4. Ce protocole de transmission opère dans la bande de fréquences des 2.4GHz. Les microcontrôleurs choisis sont généralement d'une faible vitesse et de très faible consommation d'énergie. De même, la mémoire disponible pour les programmes et les données est très réduite en comparaison avec celle des équipements informatiques d'aujourd'hui.

Carte d'acquisition	Type de mesures captées
MTS300	Lumière, température et son.
MTS310	Lumière, température, son, accéléromètre X et Y, magnétomètre X et Y
MTS400	Pression barométrique, température, humidité relative, lumière et un accéléromètre X et Y.
MTS420	Pression barométrique, température, humidité relative, lumière, un accéléromètre X et Y et GPS.
MTS510	Lumière, son et un accéléromètre X et Y.
MDA300	Température et humidité.

Tableau 1. 2 : Cartes d'acquisition sensorielles.

2.3 Domaines d'application

Les capteurs ouvrent de nouveaux horizons à la gestion de l'information. Ils forment une "peau virtuelle" avec le monde réel qui nous informe des événements physiques se déroulant autour de nous. Cela a naturellement attiré plusieurs domaines d'applications qui ont mis en œuvre les facilités que les capteurs leur offrent. En voici une liste non exhaustive.

- **Le domaine militaire** : comme pour beaucoup d'autres domaines, les applications militaires ont été les locomotives de la recherche pour les réseaux de capteurs. Pour les militaires, un réseau de capteurs offre des avantages très précieux. Il s'agit d'un réseau qui s'installe rapidement, dynamiquement et sans aucune infrastructure. Ainsi, il offre un atout de taille pour surveiller les mouvements de l'ennemi, communiquer à bas coût entre les unités avec une logistique peu compliquée.
- **La surveillance environnementale** : la petite taille et les capacités relativement grandes au niveau de calcul et de communication des capteurs permettent de les placer aux endroits que les humains ne peuvent ou ne veulent pas accéder, comme par exemple les

grandes forêts, les volcans, les profondeurs des océans, les régions polaires, ou encore d'autres planètes que la terre (Mainwaring, et al., 2002). On peut aussi utiliser les WSN pour la surveillance du degré de maturité des récoltes (raisin), la mesure de la qualité de l'eau ou de l'air.

- **L'industrie** : les industriels s'intéressent au potentiel des capteurs pour diminuer les coûts du contrôle et de la maintenance des produits, de la gestion de l'inventaire, de la télésurveillance après vente, etc. (Bonivento, et al., 2006). En particulier, l'intégration de la technologie RFID avec les réseaux de capteurs est une des directions prometteuses de recherche dans l'industrie.
- **Les domaines urbains et domotique** : les capteurs entrent de plus en plus dans nos vies quotidiennes. Dans le milieu urbain, les capteurs sont déjà utilisés pour la localisation des bus, pour des tickets électroniques et pour la sécurité. Une des applications est la surveillance du trafic routier avec les réseaux de capteurs déployés sur les autoroutes (Cheung, et al., 2004). De plus, les maisons, les bâtiments, les bureaux équipés de capteurs intelligents permettent de construire des systèmes pervasifs (Estrin, et al., 2002) où l'information est omniprésente.
- **Le domaine médical** : la recherche sur l'usage des capteurs intelligents dans le domaine médical inclut les moyens d'hospitalisation à domicile, l'intégration des micro-capteurs "dans" le corps (e.g. construire un BAN - Body Area Network) et la gestion des urgences (Lorincz, et al., 2004). Parmi les applications les plus utiles, on cite la télésurveillance des signes vitaux et des niveaux d'activité à domicile des personnes âgées ou handicapées ainsi que le contrôle à distance des données physiologiques.

2.4 Facteurs influençant l'architecture des WSN

Un ensemble de métriques permet de déterminer le design d'un réseau de capteurs. Ces facteurs influencent sur l'architecture des réseaux de capteurs et le choix des protocoles à implémenter :

2.4.1 Tolérance aux pannes

Les nœuds peuvent être sujets à des pannes dues à leur fabrication (ce sont des produits de série bon marché, il peut donc y avoir des capteurs défectueux) ou plus fréquemment à un manque d'énergie. Les interactions externes (chocs, interférences) peuvent aussi être la cause des dysfonctionnements. La panne d'un nœud capteur ne doit pas affecter

le fonctionnement global de son réseau. La tolérance aux pannes est donc la capacité de maintenir les fonctionnalités du réseau sans interruption due à une panne d'un nœud capteur.

2.4.2 Passage à l'échelle

Le nombre de nœuds capteurs déployés dans un réseau peut être à l'ordre des centaines voire des milliers. Pour certaines applications, il peut atteindre quelques millions. Afin de garantir le bon fonctionnement du réseau, les nouveaux schémas de déploiement doivent être capables de travailler avec ce grand nombre de nœuds. Par ailleurs, ils doivent utiliser la propriété de haute densité dans les réseaux de capteurs ; et donc pouvoir déployer un grand nombre de nœuds dans une petite surface.

2.4.3 Contraintes matérielles

Un nœud doit être placé dans une petite surface n'excédant pas, généralement, un centimètre cube ($1cm^3$). En outre de cette contrainte de surface, un ensemble de conditions doit être satisfait. Un nœud capteur doit :

– Consommer le strict minimum d'énergie ;

– Fonctionner dans de fortes densités ;

– Avoir un faible coût de fabrication ;

– Etre autonome ;

– S'adapter à l'environnement.

2.4.4 Coût de production

Comme les WSN consistent en un grand nombre de nœuds capteurs, le coût d'un seul capteur est très important pour définir le coût total de son réseau. Si ce dernier est plus cher que le déploiement d'un ensemble de capteurs ordinaires, alors le coût du WSN n'est pas justifié. L'état de l'art définit le coût d'un réseau Bluetooth à 10$, et un nœud capteur à 1$ (Akyildiz, et al., 2002).

2.4.5 Topologie dynamique

La dynamicité du réseau découle des défaillances des nœuds ou des cassures des liens entre ceux-ci. La disparition d'un nombre de capteurs dans le réseau, ainsi que le déploiement de nouveaux capteurs, rend la topologie du réseau fréquemment instable. La maintenance d'un réseau est d'autant importante que le changement de sa topologie. On distingue généralement trois phases dans la mise en place et l'évolution d'un réseau :

- *Déploiement* : Le déploiement des capteurs est la première opération (phase) dans le cycle de vie d'un réseau de capteurs. On peut envisager plusieurs formes de déploiements selon les besoins des applications. Les nœuds peuvent être déployés aléatoirement d'un avion ou d'une roquette par exemple, ou bien ils peuvent être placés un par un d'une manière déterministe par un humain ou un robot. Dans un grand nombre d'applications, le déploiement manuel est impossible. De plus, même lorsque l'application permet un déploiement déterministe, le déploiement aléatoire est adopté dans la majorité des scénarios à cause de raisons pratiques tels que le coût et le temps. Cependant, le déploiement aléatoire ne peut pas fournir une distribution uniforme sur la région d'intérêt, ce qui déclenche de nouveaux problèmes dans les réseaux de capteurs. Les principaux problèmes engendrés sont la localisation, la couverture de la zone, la connexité et la sécurité.
- *Post-Déploiement - Exploitation* : Durant la phase d'exploitation, la topologie du réseau peut être soumise à des changements dus à des modifications de la position des nœuds ou bien à des pannes.
- *Redéploiement* : L'ajout de nouveaux capteurs dans un réseau existant implique aussi une remise à jour de la topologie.

Dans tous les cas, le réseau de capteurs doit pouvoir se réorganiser rapidement avec un coût énergétique réduit.

2.4.6 Environnement

Les capteurs sont souvent déployés en masse dans des endroits hostiles. Ils sont soumis à différentes conditions d'environnement; ils peuvent fonctionner sous haute pression au fond de l'océan, dans un environnement dur tel que les champs de bataille, dans des champs biologiquement ou chimiquement souillés ou même dans des milieux extrêmement froids. Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiquement éloignées ou inaccessibles

2.4.7 Agrégation des données

Dans les réseaux de capteurs, les données produites par les nœuds capteurs sont très reliées, ce qui implique l'existence de redondances de données. Une approche répandue consiste à agréger les données au niveau des nœuds intermédiaires afin de réduire la consommation d'énergie lors de la transmission de ces données.

2.4.8 Consommation d'énergie

L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner. En effet, un réseau de capteurs ne peut pas survivre si la perte de nœuds est très importante car ceci engendre des pertes de communications dues à une très grande distance entre les nœuds restants. Les réseaux de capteurs fonctionnant selon un mode de routage par saut, chaque nœud du réseau joue un rôle important dans la transmission de données. Le mauvais fonctionnement d'un nœud implique un changement dans la topologie et impose une réorganisation du réseau.

3. LA SECURITE DANS LES WSN

Comme nous l'avons déjà mentionné dans la section précédente, les réseaux de capteur sans fil ont un intérêt particulier pour les applications militaires, environnementales, domotiques, médicales, et bien sûr les applications liées à la surveillance des infrastructures critiques. La conception de ces applications suppose que tous les nœuds engagés sont coopératifs et dignes de confiance. Cependant, ceci n'est pas le cas dans les déploiements du monde réel, où les nœuds sont exposés à différentes types d'attaques qui peuvent carrément endommagé le bon fonctionnement du réseau. Ces attaques exploitent essentiellement l'incertitude du canal de communication et le déploiement aléatoire des nœuds capteurs dans des zones difficiles à surveiller. Garantir la sécurité de ce type de réseau est une tâche difficile, surtout quand les nœuds sont constitués d'engins électroniques peu onéreux avec des capacités matérielles limitées. Le cas échéant, utiliser des protections physiques est, dans beaucoup de situations, quasiment impraticable. Capturer des nœuds est alors une possibilité intéressante pour les attaquants.

Néanmoins, les WSN ne peuvent compter sur l'intervention humaine pour faire face aux tentatives d'un attaquant pour compromettre le réseau ou gêner ses propres opérations. Dans cette section, nous présentons un aperçu sur les problèmes de sécurité dans les réseaux de capteurs sans fil. Premièrement, nous présentons les défis de sécurité pour les WSN qui rendent la sécurité pour ce type de réseaux assez dure. Ensuite, nous présentons une taxonomie des attaques et discutons les besoins de sécurité requis par les protocoles.

3.1 Propriétés à impact majeur sur la sécurité

La sécurité des WSN peut être classifiées en deux grandes catégories (1) la sécurité opérationnelle et (2) la sécurité de l'information. L'objectif de la sécurité relative à l'opération est d'assurer la continuité de fonctionnement du réseau en entier même si une partie de ses composants a été attaquée (service de disponibilité). Quant à la sécurité relative à l'information, son objectif est que la confidentialité de l'information ne doit jamais être divulguée et que l'intégrité et l'authentification de l'information doivent toujours être assurées.

Alors qu'il peut sembler que la sécurité de l'information peut aisément être réalisée avec la cryptographie, ils existent néanmoins trois obstacles qui rendent l'achèvement des objectifs cités ci-dessus non trivial dans les réseaux de capteurs sans fil : les ressources très limitées, la communication sans fil et le couplage étroit avec l'environnement.

3.1.1 Les ressources très limitées

Comme précédemment indiqué, les nœuds capteurs sont dotés d'une mémoire très limitées. Ceci signifie qu'un mécanisme complexe de sécurité pourrait avoir un nombre d'instructions trop grand et donc consommer trop de mémoire, et ne laisser que très peu de mémoire ou presque pas pour d'autres opérations éventuelles pour le capteur. Ainsi, la taille du code doit être la plus petite possible et le nombre de clés de sécurité stockées dans le capteur doit être pris en considération afin qu'un mécanisme de sécurité ait peu d'incidence sur les performances d'un réseau de capteur (Walters, et al., 2005).

L'énergie limitée des capteurs est probablement la caractéristique la plus pénalisante. Le plus grand des défis dans le domaine des WSN reste de concevoir des protocoles, entre autre de sécurité, qui minimisent l'énergie afin de maximiser la durée de vie du réseau. En d'autres termes, l'énergie est sans aucun doute la ressource qui convient de gérer avec la plus grande attention. Les solutions de sécurité qui existent aujourd'hui ne sont pas utilisables car elles sont souvent trop coûteuses en termes de ressource. Par exemple, l'utilisation de la cryptographie à clés publiques est souvent proscrite de ce type d'environnement. De nouveaux algorithmes et protocoles de sécurité sont nécessaires.

3.1.2 La communication sans fil

La sécurité du réseau dépend du protocole de sécurité définit qui à son tour dépend de la communication. Dans la conception de mécanisme de sécurité, la perte, la modification de paquets et la latence doivent être pris en considération. Si le taux d'erreurs du canal est assez

élevé, alors un traitement d'erreur doit être effectué afin que les paquets de sécurité par exemple les clés cryptographiques ne soient pas endommagés (Walters, et al., 2005).

Le médium de communication sans fil est à son tour un obstacle à la sécurité, rendant les attaques sur le réseau de capteurs tels que l'accès au réseau, interception et déni de service plus faciles à achever que dans les réseaux filaires.

3.1.3 Couplage étroit avec l'environnement

La plupart des applications des WSN exigent un déploiement étroit des nœuds à l'intérieur ou à proximité des phénomènes à surveiller. Cette proximité physique avec l'environnement conduit à de fréquentes compromissions intentionnelles ou accidentelles des nœuds capteurs.

Comme le succès des WSN dépend également de leur faible coût, les capteurs ne peuvent pas se permettre une protection physique inviolable (tamper-proof).

Par conséquent un adversaire bien équipé peut extraire des informations cryptographiques des nœuds capteurs. Comme la mission d'un WSN est généralement sans surveillance, le potentiel d'attaquer des nœuds, de récupérer leur contenu ou d'injecter des données erronées est important. Du fait que les réseaux de capteurs soient contrôlés à distance, il est également très difficile de savoir si la nœud capteur a été physiquement manipulé ou reprogrammé (Walters, et al., 2005).

3.2 Les besoins de sécurité typiques aux WSN

Pour déterminer des objectifs de sécurité, il faudra connaitre ce qu'on doit protéger. Les réseaux de capteurs partagent certaines caractéristiques des réseaux mobiles ad hoc mais aussi possèdent des propriétés spécifiques aux WSN, discutées dans la section précédente. Donc les objectifs de sécurité englobent ceux des réseaux traditionnels et les objectifs issus des contraintes intrinsèques aux WSNs. Parmi les principaux objectifs de sécurité, nous citons :

3.2.1 L'authentification

Elle permet de coopérer au sein des WSN sans risque, en contrôlant et en identifiant les participants. Elle apparaît comme la pierre angulaire d'un réseau de capteur sans fil sécurisé. En effet, on ne peut assurer une confidentialité et une intégrité des messages échangés si, dès le départ, on n'est pas sûr de communiquer avec le bon nœud. Si l'authentification est mal gérée, un attaquant peut se joindre au réseau et injecter des messages erronés.

L'utilisation de Code d'Authentification de Message (CAM), ou MAC en anglais (Message Authentication Code), permet d'assurer à la fois l'authentification de l'origine et l'intégrité du message. Un exemple de MAC est : HMAC (Krawczyk, et al., 1997).

3.2.2 La confidentialité

Une fois les parties authentifiées, la confidentialité reste un point important, étant donné la communication sans fil des WSN. Elle consiste à préserver le secret des messages échangés et ne pas les révéler aux adversaires. La confidentialité peut être assurée par l'usage de la cryptographie à clé symétrique ou asymétrique.

3.2.3 L'intégrité

Elle assure que les données reçues n'ont pas été altérées durant leur transit dans le réseau de manière volontaire ou accidentelle. Elle peut être assurée par l'utilisation des fonctions de hachage cryptographiques qui permettent d'obtenir pour chaque message une empreinte numérique.

Les fonctions MD2 (Message Digest 2) (Kaliski, 1992), MD5 (Rivest, 1992), SHA-1 (Secure Hash Algorithm 1) (Eastlake, et al., 2001) sont des exemples de quelques fonctions de hash les plus utilisées.

3.2.4 La disponibilité

Elle signifie que le réseau est disponible pour assurer ses services et autoriser les parties communicantes lorsque ceci est nécessaire. Cette propriété reste difficile à assurer dans les WSN étant donné les contraintes qui pèsent sur ces réseaux, à savoir : topologie dynamique, ressources limitées des nœuds de transit, communications sans fil pouvant être facilement brouillées ou perturbées.

3.2.5 La fraîcheur

Ce dernier service permet de garantir que les données échangées sur le réseau sont actuelles et ne sont pas une réinjection de précédents échanges interceptés par un attaquant.

3.3 Classification des attaques

Dans les réseaux de capteurs, un attaquant peut effectuer une variété d'attaques n'ayant pas forcément le même objectif ou motivations. Ainsi le choix d'une stratégie de sécurité doit se baser sur une modélisation de l'attaque ; ceci afin d'éviter un déploiement excessif de moyens de protection conduisant à des solutions irréalistes. Selon (Yong, et al.,

2006), les attaques sur les réseaux de capteurs peuvent être classifiées dans les catégories suivantes :

3.3.1 Attaques passives VS attaques actives

Les attaques passives "eavesdropping" se limitent à l'écoute et l'analyse du trafic échangé. Ce type d'attaque est plus facile à réaliser (il suffit de posséder un récepteur adéquat) et il est difficile de le détecter puisque l'attaquant n'apporte aucune modification sur les informations échangées. L'intention de l'attaquant peut être la connaissance des informations confidentielles ou bien la connaissance des nœuds importants dans le réseau (chef de groupe "cluster head"). En analysant les informations de routage, l'attaquant va se préparer à mener ultérieurement une action précise.

Dans les attaques actives, un attaquant tente de supprimer ou modifier les messages transmis sur le réseau. Il peut aussi injecter son propre trafic ou rejouer d'anciens messages pour perturber le fonctionnement du réseau ou provoquer un déni de service.

3.3.2 Attaques externes VS Attaques internes

Dans le cas de l'attaque externe, le nœud attaquant n'est pas autorisé à participer dans le réseau de capteurs. Des techniques de cryptographie et d'authentification protègent l'accès au réseau à ce type d'attaquant. Cependant ce dernier peut uniquement déclencher des attaques passives tels que l'écoute clandestine, le brouillage radio, ou l'attaque par rejeu.

L'attaque interne est considérée comme la plus *dangereuse* du point de vue sécurité. Puisque l'attaquant qui capture un nœud, peut lire sa mémoire et avoir accès à son matériel cryptographique et par conséquent peut s'authentifier comme un nœud légitime et émettre des messages aléatoires erronés sans qu'il soit identifié comme intrus, puisqu'il utilise des clés valides. Les méthodes cryptographiques s'avèrent donc inefficace pour ce genre d'attaque. Il est donc nécessaire d'utiliser d'autres méthodes complémentaires telles que les systèmes de monitoring et les systèmes de réputations.

3.4 Modèle de l'attaquant

En général, plus l'attaquant dispose de ressources, plus la défense est coûteuse. La connaissance de la capacité de l'attaquant permet de définir au mieux la défense. La conception de réseaux de capteurs doit prendre en considération les menaces les plus fréquentes en énumérant les capacités des attaquants (leur nombre, leur coordination, leur capacité technique et leur intérêt d'influence).

Plusieurs attaquants peuvent menacer un réseau au même moment d'une manière autonome ou en coordination, afin de réaliser une attaque commune rendant la défense difficile. La définition des capacités techniques des attaquants est importante pour connaitre la nature de leur menace, par exemple un attaquant peut seulement recevoir la transmission de données, mais il peut aussi se présenter comme un capteur légal du réseau, et avoir accès à la totalité des services du réseau. Selon (Anderson, et al., 2004), un réseau de capteurs peut être attaqué par deux types d'attaquants : un attaquant puissant et un attaquant réaliste.

3.4.1 Attaquant puissant

L'adversaire est considéré comme présent avant et après le déploiement des nœuds. Il peut surveiller toutes les communications, n'importe où, et à tout instant. Il est aussi courant de le modéliser avec le potentiel de subvertir un sous-ensemble restreint de nœuds.

3.4.2 Attaquant réaliste

Anderson a déclaré dans (Anderson, et al., 2004) que la première catégorie est trop exigeante en termes de protocoles de sécurité et conduit à des solutions qui ne sont pas utilisées dans la pratique. Ces auteurs préfèrent modéliser l'attaquant comme étant capable de surveiller un pourcentage *fixe* des canaux de communication lors du déploiement du réseau. Pour justifier la vision « réaliste » de l'attaquant, et non pas le modèle de l'adversaire capable d'intercepter toute communication et d'injecter de nouveaux messages comme il le souhaite.

3.5 Les types de vulnérabilités des WSN

Les vulnérabilités sont les faiblesses d'un réseau que l'attaquant exploite afin de gagner des privilèges. Il y a deux types de vulnérabilités dans un réseau de capteurs WSN :

La vulnérabilité physique est un moyen d'attaque, qui permet à l'attaquant de changer en partie un capteur, en modifiant par exemple son code de programmation, ou en copiant les clés de protection afin de les réutiliser dans une nouvelle attaque. Un réseau de capteurs est vulnérable aussi aux modifications de son environnement, où un attaquant peut modifier les valeurs d'un capteur local, lui permettant ainsi d'avoir un accès aux commandes de contrôle du réseau WSN.

La vulnérabilité logique réside dans les programmes et les protocoles. Elle se présente sous quatre formes : (i) les défauts de conception, (ii) les défauts d'implémentation, (iii) les erreurs de configuration, et (iv) l'épuisement des ressources.

- Les défauts de conception permettent l'utilisation d'un protocole qui viole le mode d'utilisation, tout en se conformant à la spécification du protocole. Par exemple, un manque d'authentification dans un protocole de gestion de puissance peut permettre de mettre n'importe quel capteur en sommeil à plusieurs reprises.
- Les défauts d'implémentation sont des erreurs dans la construction du matériel ou dans le codage du logiciel. Par exemple, une erreur de dépassement de mémoire, peut entraîner une violation d'accès et une mise en panne.
- Les défauts de configuration sont le résultat de défauts de paramétrages pour un attaquant.
- L'épuisement des ressources est possible même si la conception, l'implémentation, et la configuration sont correctes. Un attaquant générant de grandes quantités de trafic peut inonder un des liens réseau de la victime. Une mauvaise authentification de l'allocation de mémoire ou de l'exécution de code peut également permettre à un attaquant de consommer les ressources du capteur subissant l'attaque, et de causer un déni de service.

3.6 Les attaques contre les WSN

Les WSN peuvent faire l'objet d'un grand nombre d'attaques, chacune avec ses objectifs propres. Par exemple, certaines attaques visent à affecter l'intégrité des messages qui transitent dans le réseau, tandis que d'autres visent à réduire la disponibilité du réseau ou de ses composants. Les attaques se produisent souvent par l'insertion d'éléments intrus dans le réseau. Il existe aussi des attaques contre l'environnement extérieur au réseau, lesquelles provoquent des altérations ou des interférences sur les signaux transmis. Une bonne classification des attaques est présentée dans (Wood, et al., 2002). Nous présentons dans la suite les attaques les plus connues dans les WSN.

- Ecoute du réseau (eavesdropping) : Du fait que les transmissions se font en diffusion par les ondes radio, aucun contrôle d'accès au réseau n'est possible, ce qui est d'autant plus vrai que le réseau peut être déployé dans un environnement ouvert accessible à tout le monde. Il est donc très facile d'intercepter des données échangées sur un réseau de capteurs et d'accéder à leur contenu si aucun service de confidentialité n'est prévu.

- Attaque physique (tampering) *:* Comme les WSN sont très souvent déployés dans des zones sans aucune protection, ils sont très exposés aux attaques physiques qui peuvent être

considérées sous différents points de vue. L'un est lié au matériel qui n'est pas qualifié d'inviolable. Dans ces conditions, une attaque aura pour but de récupérer du matériel cryptographique comme les clés utilisées pour le chiffrement. Un autre objectif serait de reprogrammer le capteur pour perturber le réseau et l'application en provoquant volontairement un comportement anormal du nœud. La seconde attaque physique consisterait simplement à supprimer le capteur du réseau en le détruisant (on retombe sur la question de l'inviolabilité) ou en le subtilisant (Boyle, et al., 2008).

- Attaque de l'identité multiples (**sybil attack**) : Dans cette attaque, un nœud malveillant peut revendiquer différentes identités afin de participer à des algorithmes distribués tels que l'élection et de prendre de l'avantage sur les nœuds légitimes. Un nœud malveillant peut être capable de déterminer le résultat de n'importe quel vote en faisant voter toutes ses identités multiples pour une même entité. Les techniques d'authentification et de chiffrement peuvent empêcher un étranger de lancer une attaque Sybille sur le réseau de capteur.

- Attaque du trou noir (blackhole ou sinkhole) : Un nœud falsifie les informations de routage pour forcer le passage des données par lui-même. Sa seule mission est ensuite de ne rien transférer, créant ainsi une sorte de puits ou trou noir dans le réseau. L'intrus (nœud malveillant, qui s'introduit illégitimement) se place sur un endroit stratégique de routage dans le réseau et supprime tous les messages qu'il devrait retransmettre, causant la suspension du service de routage du réseau dans les routes qui passent par le nœud intrus.

- Attaque du trou gris (grey hole) : Une variante de l'attaque précédente est appelée trou gris, dans laquelle seuls certains types de paquets sont ignorés par le nœud malicieux. Par exemple, les paquets de données ne sont pas retransmis alors que les paquets de routage le sont.

- Brouillage radio (jamming) : L'intrus inonde avec du bruit les fréquences radio utilisées par le réseau de manière à empêcher les transmissions et/ou les réceptions de messages. Ce type d'attaque peut affecter tout ou une partie du réseau selon la portée radio de l'intrus. Dans ce cas-là, l'intention est de provoquer un déni de service.

- Relais sélective (selective forwarding) : L'intrus néglige son rôle de routeur et ne transmet pas certains messages qui sont choisis selon certains critères ou même aléatoirement.

- Attaque du trou de ver (wormhole) : L'intrus capture un message et, en utilisant un canal de faible latence, le retransmet vers un lieu distant dans le réseau. Le canal ainsi créé fait transiter un message à un endroit du WSN auquel il ne devrait normalement pas arriver. Cette attaque a une influence notable sur le routage dans le réseau.

- **Rejeu, Délai et Altération de Données** : L'intrus répète, retarde ou altère le contenu des messages en transit. Les messages peuvent contenir des données de perception prélevées et des données de configuration ou de routage. Ces types d'attaques visent entre autres à créer des boucles, attirer à lui ou éloigner du trafic, augmenter ou diminuer le nombre de routes, générer de fausses erreurs, partitionner le réseau, et augmenter la latence de distribution des données.

- **Attaque par chantage** : Elle est connue sous le nom anglais de "Blackmail attack". Un nœud malicieux fait annoncer qu'un autre nœud légitime est malicieux pour éliminer ce dernier du réseau. Si le nœud malicieux arrive à attaquer un nombre important de nœuds, il pourra perturber le fonctionnement du réseau.

- **Attaque de l'inondation de "HELLO"** : De nombreux protocoles de routage utilisent des paquets "HELLO" pour découvrir les nœuds voisins et ainsi établir une topologie du réseau. La plus simple attaque pour un attaquant consiste à envoyer un flot de tels messages pour inonder le réseau et empêcher d'autres messages d'être échangés.

- **Epuisement de la batterie (exhaustion)** : Cette attaque de déni de service est redoutable car elle vise à épuiser les batteries des nœuds composant le réseau de manière à réduire la durée de vie du réseau. Elle peut consister à injecter de nombreux messages dans le réseau qui conduisent les nœuds à gaspiller leur énergie en retransmissions inutiles.

3.7 Coût des protocoles de sécurité dans les capteurs

L'introduction de protocoles de sécurité dans un réseau de capteurs n'est pas anodine et peut même avoir des effets dévastateurs sur les capteurs, en particulier sur leur durée de vie du fait que la mise en œuvre de la sécurité est très consommatrice d'énergie.

D'une part, il faut considérer le problème du traitement à opérer au niveau des capteurs pour mettre en œuvre les fonctions de sécurité. Le choix de ces fonctions est extrêmement important puisqu'il est nécessaire que le code associé soit de taille réduite (ROM), le traitement pas trop gourmand en CPU et ce, de manière à s'intégrer aisément dans les capteurs (sans perturber leur fonctionnement de base). En particulier, on évitera les algorithmes cryptographiques à clés publiques qui sont très consommateurs en CPU et en mémoire. On conseillera plutôt l'usage des algorithmes symétriques de type RC5 (Rivest Cipher 5) ou Skipjack du fait de la taille réduite du code source, de sa rapidité d'exécution et de la mémoire réduite utilisée pendant son exécution (RAM).

Pour limiter la taille du code au sein des capteurs, l'idée généralement retenue sera d'utiliser les mêmes outils cryptographiques pour chiffrer les données émises (RC5 par exemple) et pour générer un MAC (Message Authentication Code) et protéger en intégrité les données. Ce MAC s'appelle classiquement CBC-MAC car il consiste à fragmenter les données à protéger en plusieurs blocs de données (voir Figure 1.4), et à faire intervenir l'opération XOR dans le chiffrement d'un bloc X_i, le bloc chiffré précédent H_{i-1} ; de la sorte, le MAC final obtenu correspond au dernier bloc chiffré ; il dépend bien de tous les blocs constituant les données à protéger et constitue une empreinte des données.

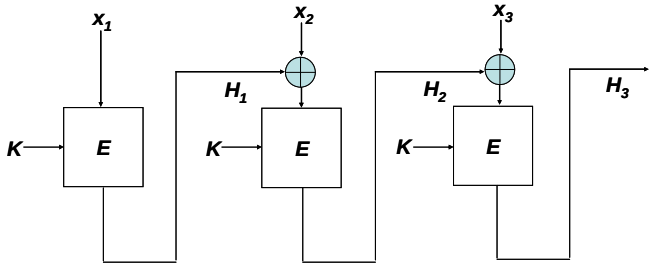


Figure 1. 4 : Authentification par CBC-MAC avec l'opérateur XOR.

D'autre part, comme le montre la Figure 1.5 (issue de (Perrig, et al., 2002.), l'activité la plus consommatrice en énergie dans un capteur n'est pas seulement le traitement de la sécurité qui ne représente que 3 à 4% de la consommation totale de son énergie, mais toutes les opérations de transmission qui représentent plus de 95% de l'énergie totale. Ainsi, il apparaît d'autant plus important que les solutions de sécurité considérées ajoutent un nombre d'informations très restreint dans un paquet.

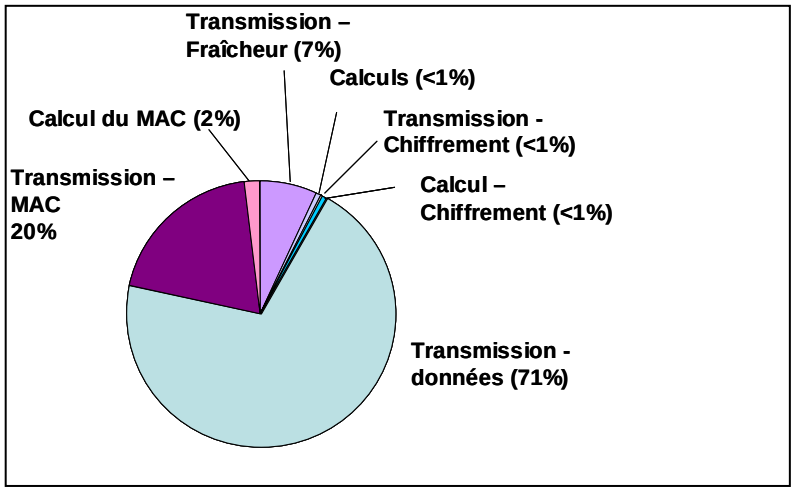


Figure 1. 5 : Energie consommée par la solution SNEP (Perrig, et al., 2002.)

Dans l'exemple de la Figure 1.5, le simple fait d'assurer la protection d'un paquet avec entre autre l'ajout d'un MAC allonge la taille du paquet de 6 octets et aura pour incidence que 20% de la batterie du capteur sera consommée par la simple transmission du MAC. Dès lors, la durée de vie du capteur sera réduite de plus de 27% par la simple introduction des mécanismes de sécurité : MAC et fraîcheur.

Ainsi, dans la présentation des solutions, plusieurs critères sont à considérer :

3.7.1 Coût en stockage

Il faut distinguer les mémoires ROM et RAM nécessaires à la mise en œuvre des solutions de sécurité. La mémoire ROM (non volatile) est destinée à contenir le système d'exploitation du capteur (généralement TinyOS), ainsi que tout autre code (programme) associé à la sécurité et à la gestion des communications. La mémoire RAM sert à contenir toutes les données en cours de traitement dans le capteur, telles que les résultats intermédiaires ou temporaires (ex : résultats des opérations cryptographiques).

3.7.2 Coût en énergie

Les explications précédentes démontrent que la consommation en énergie est un critère fondamental dans les réseaux de capteurs. En particulier, il faut retenir que l'opération de transmission de données est extrêmement gourmande en énergie et donc que le moindre ajout de MAC, de numéro de séquence, de vecteur d'initialisation…etc, dans les paquets de données a un coût très élevé qui va affecter la durée de vie des capteurs.

3.7.3 Failles de sécurité résiduelles

Les protocoles de sécurité ne permettant pas de résoudre tous les problèmes de sécurité, en particulier les attaques par épuisement de la batterie, il est intéressant de déterminer les failles les plus importantes qui persisteront même avec l'introduction de services de sécurité.

3.7.2 Fonctionnalités

Certaines fonctions habituellement utilisées sur les réseaux de capteurs ne sont pas compatibles avec certaines solutions de sécurité. Par exemple, la fonction d'agrégation a pour but de réduire le volume de données transmises par un capteur, mais elle suppose que le capteur soit en mesure d'accéder au contenu des paquets de données et de modifier ces

paquets, ce qui n'est pas toujours envisageable dans le cas d'une protection de paquets en confidentialité ou en intégrité.

4. ISSUES MAJEURES DE SECURITE

La sécurité est un domaine très vaste et représente un défi scientifique à cause des caractéristiques spécifiques des réseaux de capteurs. Les recherches dans cette problématique ont révélé plusieurs axes de recherche. Parmi ces axes, nous citons :

4.1 La sécurité du routage

Le problème du routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance comme la consommation énergétique. Une attaque simple de déni de service sur un protocole de routage consiste pour un nœud à refuser arbitrairement de transférer certains messages ou de supprimer un paquet en transit de façon aléatoire. L'attaque du trou de ver peut également faire croire à deux nœuds distants qu'ils sont très proches alors qu'en réalité ils sont éloignés de plusieurs sauts. En présence de telles attaques, les nœuds du réseau seront alors contraints de mettre à jour leur table de routage pour continuer d'assurer la fiabilité de leur service. Il est donc nécessaire de sécuriser les protocoles de routage conçus initialement pour un environnement sans risque ou même de concevoir de nouveaux algorithmes robustes afin de mener à bien l'opération de l'acheminement des données même en présence des nœuds malicieux. Cette problématique a été très largement étudiée par les chercheurs ces dernières années.

4.2 La sécurité de l'agrégation de données

Une approche courante pour surmonter les limitations des réseaux de capteurs est d'agréger les données au niveau des nœuds intermédiaires. Garantir la sécurité conjointement à des techniques d'agrégation est difficile parce qu'un nœud capturé pose un double problème. Il compromet la confidentialité des données (possibilité d'écoute) et leur disponibilité (possibilité d'attaque du type déni de service). Egalement, un nœud d'agrégation compromis met en danger toutes les mesures qui font partie de l'agrégat dont le nœud est responsable. Ce qui mène à déclencher de fausses alarmes ou même de dissimuler les événements d'exception. Dans les applications critiques, ceci peut avoir un impact néfaste.

Plusieurs chercheurs ont déjà étudié le problème de la sécurité de l'agrégation des données. Nous donnerons plus de détails dans le chapitre 2.

4.3 La sécurité de la localisation

La connaissance des positions des capteurs dans l'environnement surveillé est souvent indispensable pour une grande majorité des applications (militaires, suivis des animaux, ...), afin de pouvoir déterminer l'origine des événements détectés. « Où ? » est la question qui suit immédiatement la détection d'un événement (par exemple, où est le feu ?). En outre, la localisation peut être utilisée dans les protocoles de routage géographique dans les réseaux à grande échelle, en transmettant les données seulement dans la direction de la destination. Il est donc nécessaire de localiser, avec la meilleure précision possible, tous les nœuds du réseau. Cependant la plupart des capteurs ne peuvent être dotés d'un récepteur GPS et dépendent d'un certain type de capteurs nommés ancres pour estimer leur position. Les nœuds capteurs sont souvent éloignés de plusieurs sauts des ancres et calculent leurs coordonnées sur la base des coordonnées des ancres et le délai qui les séparent. Par conséquent la sécurisation des protocoles de localisation est nécessaire pour protéger le réseau des ancres malicieux et des attaquants qui tentent de perturber le processus de localisation. Cette problématique, malgré les nombreux travaux de recherche qui s'y étaient attachés ces dernières années, reste une problématique ouverte. Plus de détails seront donnés dans le chapitre 3

4.4 La gestion de clés

Dans le but de fournir les services de sécurité tels que : confidentialité, authentification, intégrité, sécurité de routage/agrégation/localisation etc, les capteurs ont besoin en premier lieu de partager/établir un certain nombre de clés cryptographiques secrètes. Ceci peut être effectué grâce à la gestion de clés qui fournit des mécanismes efficaces, sécurisés et stables de gestion de clés utilisées dans les opérations cryptographiques.

Par conséquence, la gestion de clés est un service primordial pour la sécurité de n'importe quel système basé sur la communication. Comme nous l'avons déjà mentionné, les nœuds capteurs sont potentiellement exposés aux attaques physiques et ne peuvent compter sur une intervention humaine. Un attaquant qui capture un nœud peut extraire toutes ses clés secrètes et déclencher tout type d'attaques sans qu'il soit identifié. Par conséquent, les protocoles de gestion de clés doivent être résistants aux attaques contre les capteurs. Une stratégie de distribution sécurisée des clés est également à prévoir afin de pouvoir assurer un certain

niveau de sécurité. Ce domaine de recherche est très critique, car sous les contraintes des WSN, la conception d'un système de gestion de clés est un grand défi. Sélectionner une solution cryptographique appropriée pour les WSN est un autre défi.

5. CONCLUSION

Dans ce chapitre nous avons procédé à l'étude des réseaux de capteurs sans fil. Nous avons posé les briques de base et fédéré quelques concepts généraux de sécurité nécessaires à la compréhension de nos problématiques dans la suite de ce manuscrit.

Cela fait des années que les réseaux de capteurs suscitent un engouement important dans la recherche et dans l'industrie en raison de la multiplicité de leurs applications faciles à déployer et à moindre coût. Ces applications ont souvent besoin d'un niveau de sécurité élevé. Or, de part de leurs caractéristiques très contraignante (plus particulièrement : contrainte d'énergie, sécurité physique limitée,...) la sécurisation des réseaux de capteurs est à la source, aujourd'hui, de beaucoup de défis scientifiques et techniques. Dans cette optique, un protocole de sécurité doit pouvoir établir des sessions sécurisées avec peu d'influence sur la performance globale du réseau, tout en fournissant les différents services de sécurité pour chaque type d'application.

Nous avons remarqué à travers nos lectures que minimiser la consommation d'énergie d'un nœud capteur « *est le cheval de bataille* » de toutes les solutions et protocoles proposés. En effet, lorsque ce n'est pas l'objectif principal, comme c'est le cas pour la problématique de sécurité, alors c'est sûrement un critère de performance capital.

Plusieurs travaux de recherches ont été menés pour résoudre les problèmes de sécurité liés aux WSN, tels que la sécurité de l'agrégation de données et la sécurité de la localisation. Dans nos travaux, nous avons investigué ces deux problématiques et pour chacune d'entre elles nous avons proposé une contribution que nous détaillerons dans les prochains chapitres.

Chapitre II

La sécurité de l'agrégation des données

Sommaire

1. INTRODUCTION

Dans beaucoup d'applications des réseaux de capteurs, les données peuvent être menacées par des événements extérieurs qui ne devraient pas arriver au cours du fonctionnement normal du réseau. En particulier, la confidentialité, l'intégrité, l'authentification et la disponibilité des données sont des fonctionnalités importantes que le réseau devrait pouvoir assurer. Garantir de telles caractéristiques est une tâche difficile, surtout quand les nœuds sont constitués d'engins électroniques peu onéreux avec des capacités matérielles limitées. Le cas échéant, utiliser des protections physiques est, dans beaucoup de situations, quasiment impraticable. Capturer des nœuds est alors une possibilité intéressante pour les attaquants.

Les contraintes inhérentes aux nœuds capteurs offrent de multiples possibilités d'attaques. Dans la mesure où le coût des communications radio est élevé en terme de consommation d'énergie (1 bit transmit est équivaut à l'exécution de 50 à 150 instructions (Peter, et al., 2007)), il est très important de réduire la charge des communications. A cette fin, une approche intéressante est d'*agréger les données*. Basé sur le principe que la station de base n'a pas nécessairement besoin de toutes les données collectées par chaque capteur en raison de leur redondance, mais seulement d'un résumé ou d'un agrégat de données effectué au niveau d'un nœud appelé agrégateur. L'agrégation de données peut considérablement aider à la conservation de l'énergie en éliminant les données redondantes (Rajagopalan, et al., 2006), et par conséquent à allonger la durée de vie du réseau des capteurs. Les fonctions d'agrégation typiques sont : la somme, la moyenne, la médiane, le max et le min, etc. (Xu, et al., 2004) (Mukherjee, et al., 2007).

Cependant, garantir la sécurité conjointement à des techniques d'agrégation est un défi du fait que les nœuds et les agrégateurs sont déployés dans des environnements hostiles et sont exposés à plusieurs menaces telles que la compromission de nœud, injection de fausses données, l'interception des données et agrégation des valeurs manipulées par des nœuds malicieux. Cependant un mécanisme d'agrégation de données doit résister aux attaques lorsque l'agrégateur et une partie de nœuds sont compromis.

Nous présenterons dans ce présent chapitre l'investigation de notre premier axe de recherche, à savoir la problématique de la sécurité de l'agrégation des données. Nous abordons tous les concepts et les challenges rencontrés lors de la conception d'un protocole sécurisé, et nous survolons les travaux existants en effectuant une analyse et une

classification. Cette étude approfondie, nous permettra de tracer les motivations pour la conception dans nouvel algorithme de sécurité des données agrégées.

Notre contribution dans ce chapitre est résumée en deux points :

- Présentation d'un état de l'art sur les algorithmes de la sécurité des données agrégées. Il existe (à notre connaissance) seulement deux états de l'art sur cet axe de recherche élaborés par Sang et al. (Sang, et al., 2006) et Alzaid (Alzaid, et al., 2008). Le premier classifie les algorithmes en deux catégories : les données agrégées chiffrées saut par saut et de bout en bout. Cependant, cette classification ne détaille aucune analyse de sécurité sur ces algorithmes ni leur performance. Le deuxième état de l'art classifie les algorithmes en deux groupes, selon le nombre de nœuds agrégateurs et selon la prise en considération de l'intégrité des données. Ce dernier est un état de l'art assez compréhensible car une étude conceptuelle pour un algorithme sécurisé a été proposée pour aider à la comparaison entre différentes solutions. Cependant, dans les deux états de l'art, les nouvelles solutions émergentes basées sur la confiance et les systèmes de réputation n'ont pas été abordées. Ces solutions représentent un complément attrayant à la cryptographie dans les réseaux de capteurs sans fil. Dans notre état de l'art, nous présentons un survol qui englobe la problématique de la sécurité particulièrement dans l'agrégation de données et une nouvelle taxonomie des solutions existantes.
- Présentation d'une étude comparative entre les solutions existantes en termes d'analyse de sécurité et de performances.

2. AGREGATION DE DONNEES

Dans un scenario typique d'un réseau de capteurs, plusieurs nœuds capteurs captent des données de l'environnement et les transmet à un nœud central ou sink (station de base ou encore puits) qui analyse et traite ces données afin de les transmettre à son tour à l'application. Cependant, dans beaucoup de cas les données produites de différents nœuds peuvent être conjointement traitées lors de leur transmission vers le sink. Par conséquent, l'agrégation s'occupe de ce traitement distribué des données dans le réseau.

L'agrégation de données dans les WSN consiste à remplacer les lectures individuelles de chaque capteur par une vue globale, collaborative sur une zone donnée (clustering). On peut utiliser par exemple de simples fonctions d'agrégat telles que MIN, MAX ou

MOYENNE, qui permettent à partir d'une série de *n* messages reçus par un « chef de zone » (capteur chef d'une zone) de ne renvoyer vers le sink qu'un seul message résumant l'information contenue dans ces *n* messages.

Elena Fasolo. (Fasolo, et al., 2007) a définit l'agrégation de données comme suit : «*In-network aggregation is the global process of gathering and routing information through a multi-hop network, processing data at intermediate nodes with the objective of reducing resource consumption (in particular energy), thereby increasing network lifetime* »

2.1 Les approches de l'agrégation

Ils existent deux approches dans l'agrégation de données: *avec réduction* de taille (*with size reduction*) et *sans réduction* de tailles (*without size reduction*).

L'agrégation *avec réduction* de taille se rapporte au processus de combiner et de compresser les paquets de données reçus par un nœud de ses voisins afin de réduire la longueur du paquet à transmettre ou à expédier vers le sink. Comme exemple, considérons la situation où un nœud capteur reçoit deux paquets ayant des données spatialement corrélées. Dans ce cas, il est inutile d'envoyer les deux paquets. Le nœud doit effectuer une fonction telle que la moyenne, le maximum ou le minimum et transmettre uniquement un seul paquet. Cette approche réduit de manière considérable la quantité de bits à transmettre dans le réseau et donc réduit la consommation d'énergie. Mais en contre partie, cette approche réduit *la précision* de la valeur de la donnée reçue.

Par contre, l'agrégation *sans réduction* de taille, se rapporte au processus de fusion des paquets de données reçus de différents voisins dans un paquet simple de données mais sans traiter la valeur des données. Par exemple, deux paquets peuvent contenir différentes quantités physiques (telles que température et humidité) et peuvent être fusionnés dans un paquet unique en préservant les deux valeurs intactes dans *un seul en-tête*. Cette approche contrairement à la première préserve les valeurs des données mais transmet *plus de bits* dans le réseau à cause de la longueur du paquet à transmettre. Néanmoins, elle réduire toujours le coût de transmission en gardant un en-tête unique.

L'utilisation de ces deux approches dépend de plusieurs facteurs tels que le type d'application, le taux des données et les caractéristiques du réseau. Il existe également un compromis entre consommation d'énergie et précision des données pour les deux approches.

La plupart des travaux effectués dans l'agrégation s'occupent particulièrement des problèmes d'acheminement des paquets de la source vers le sink, pour faciliter l'agrégation.

En fait l'idée principale est d'améliorer les protocoles de routage de façon à pouvoir agréger efficacement les données.

2.2 Classification des techniques d'agrégation

Les techniques d'agrégation sont classifiées en trois catégories :

2.2.1 Les techniques basées sur les arbres

La manière la plus simple pour agréger les données est d'organiser les nœuds de manière hiérarchique et de sélectionner quelques uns qui servent de point d'agrégation ou d'agrégateurs. Les techniques basées sur les arbres effectuent l'agrégation en construisant un arbre d'agrégation, qui pourrait être un arbre spanning-tree minimum, dont la racine est le sink et les nœuds de sources sont considérés en tant que feuilles. Chaque nœud a un nœud père qui lui expédie l'information. Le flux des données commence à partir des nœuds feuilles vers le sink.

Cependant cette technique possède certains inconvénients. Comme nous le savons les réseaux de capteurs sans fil ne sont pas exempts aux pannes. Dans le cas de perte de paquets à n'importe quel niveau de l'arbre, les données sont perdues non seulement pour un seul niveau mais aussi bien pour tout le sous-niveau correspondant. Malgré le coût élevé pour maintenir la structure arborescente dans un réseau à topologie dynamique, et le système peu robuste, cette technique est très appropriée pour concevoir une technique d'agrégation optimale et des techniques optimales en termes d'énergie.

2.2.2 Les techniques basées sur les chemins multiples

Un des inconvénients majeurs des techniques d'agrégation basées sur les arbres et que le système n'est pas robuste. Pour palier à cet inconvénient, une nouvelle technique a été proposée par plusieurs chercheurs. Au lieu de transmettre partiellement les données agrégées à un nœud père unique dans l'arbre, un nœud transmet ses données à travers plusieurs chemins (voire Figure 2.1). L'idée de base est que chaque nœud peut transmettre ses données à ces multiples voisins en exploitant la nature du médium radio. Ainsi les données sont acheminées à partir des sources vers le sink en traversant plusieurs chemins et l'agrégation peut être effectuée à chaque nœud intermédiaire. Il est clair que cette technique rend le système plus robuste puisque la perte de paquets aura peu d'influence sur l'agrégation. Mais en contre partie cette technique génère un *surcoût* supplémentaire. Une des structures d'agrégation qui s'adapte bien avec cette technique est la topologie en anneau, où le réseau

est divisé en cercles concentriques avec des niveaux définis selon leur distance (en sauts) avec le sink.

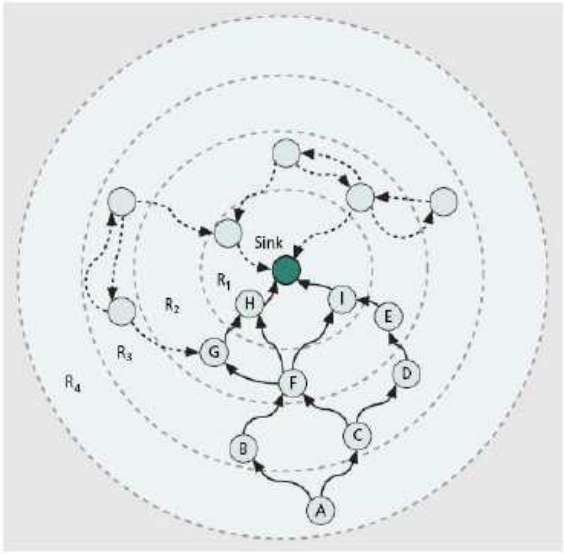


Figure 2. 1: Exemple d'agrégation basée sur les chemins multiples dans une topologie en anneau (Fasolo, et al., 2007).

2.2.3 Les techniques basées sur les clusters

Une autre manière d'organisation hiérarchique d'un réseau est l'organisation clustérisée. Dans cette organisation, le réseau en entier est divisé en plusieurs clusters (groupes) selon une métrique spécifique ou une combinaison de métriques. Chaque cluster est géré par un cluster-head (chef de groupe) qui est élu par les membres du cluster. Le cluster-head joue le rôle d'agrégateur qui agrège les données des membres du cluster locaux et transmet le résultat de l'agrégat au sink. Les avantages et les inconvénients des techniques basées sur les clusters sont similaires à ceux des techniques basées sur l'arbre.

Il existe également des techniques d'agrégation hybrides qui combinent les trois techniques citées ci-dessus.

2.3 Le modèle des données

La fonction principale des réseaux de capteur est de capter des données telles que température, humidité, localisation, accélération, vitesse, pression, etc. Généralement ces données sont exprimées comme des valeurs digitales ayant un certain intervalle $[d_{min} ; d_{max}]$. Par exemple, les données d'un compteur électrique sont normalement toujours comprises entre 0 et 100 kw/h par jour, selon la consommation. De même la température d'un milieu

ambiant normal est comprise entre 12 et 32°. Ces données sont périodiquement agrégées pour être transmise au sink toutes les heures ou toutes les demi-heures. De ce fait, l'intervalle de temps entre deux opérations d'agrégations doit être plus long que le temps utilisé par chaque opération d'agrégation.

2.4 Efficacité de l'agrégation des données

Les nœuds capteurs dans un WSN sont souvent limités en termes de ressources. Il est donc important de concevoir et de développer des techniques de traitement de données efficaces pour une bonne utilisation des données. L'agrégation des données (Madden, et al., 2002) est une technique efficace dans le traitement des requêtes dans laquelle les données sont traitées et agrégées dans le réseau. Le résultat de l'agrégat seul est transmit au sink. Dans cette configuration, les nœuds qui s'occupent de la fonction d'agrégat appelés agrégateurs, collectent les données individuelles de chaque nœud, les traitent localement et répondent aux requêtes d'un utilisateur distant.

En comparant avec l'approche centralisée, où toutes les données sont transmises, l'agrégation peut réduire de manière significative le coût de communication (l'overhead) et donc sauvegarder la consommation d'énergie et allonger la durée de vie du réseau de capteurs. Prenons l'exemple d'un réseau qui est déployé pour mesurer la température moyenne dans une zone géographique donnée. Ce réseau est structuré comme un arbre où les feuilles et les nœuds intermédiaires sont les capteurs et la racine est le sink. Chaque capteur envoie périodiquement ses données vers le sink. Chaque message est relayé nœud par nœud du capteur vers le sink. Par conséquent, si le réseau est constitué de *n* nœuds, le sink recevra à chaque période de mesure *n* messages. Pour réduire le nombre de messages et de bits transmis, chaque nœud intermédiaire peut additionner les données (températures) reçues de ces fils, ajouter la valeur de sa mesure, et envoyer le résultat à son père. Le sink recevra alors un *seul* message qui contiendra la somme des messages au lieu de *n* messages. Il pourra alors diviser cette somme par *n* et obtenir la moyenne de la température.

La Figure 2.2 illustre un réseau de 7 capteurs. Au total, 17 messages sont envoyés sur le réseau de capteurs. En utilisant le mécanisme d'agrégation de données, on obtient un total de 7 messages envoyés sur le réseau.

De plus en plus que la taille du réseau augmente (par exemple 2500 nœuds capteurs), la collecte de données sans agrégation consomme une bande passante extrêmement grande (Madden, et al., 2002). En effet, si les valeurs mesurées par chaque capteur sont codées sur *m*

bits, le sink recevra *log2(n) + m* bits au lieu de *n + m* bits. Le gain en bande passante est alors de *(log2(n)+ m)/(n + m).*

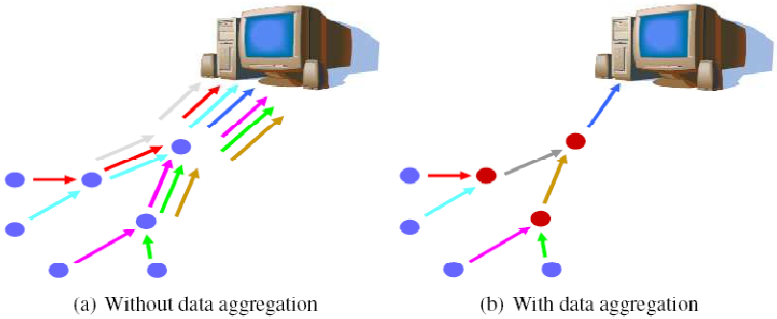


Figure 2. 2: Illustration de l'efficacité de l'agrégation de données (Labraoui, et al., 2011(a)).

3. PROBLEMATIQUE DE LA SECURITE DE L'AGREGATION DES DONNEES

L'agrégation des données dans les réseaux de capteur est relativement triviale, mais devient problématique lorsque l'on veut y ajouter de la sécurité et plus particulièrement de la confidentialité (chiffrement). Dans certaines applications, il est essentiel de s'assurer que les informations qui sont transmises sur le réseau ne puissent être interceptées et lues par des personnes non-autorisées. Elles doivent donc être chiffrées. Mais le chiffrement et l'agrégation sont deux concepts qui ne vont pas très bien ensemble.

Nous étudions dans cette section la problématique de sécurité dans l'agrégation de données, en précisant les besoins de sécurité, les attaques et les défis à relever lors de la conception d'un protocole sécurisé.

3.1 Besoins de sécurité dans l'agrégation des données

Le traitement des données dans le réseau (in-network data processing), tel que la fusion et l'agrégation (Rajagopalan, et al., 2006) a émergé ces dernières années comme un axe de recherche très actif dans les WSN. Une des préoccupations les plus importantes dans l'agrégation est de trouver un compromis réaliste entre coût de calcul, délai, résolution des données et fiabilité. Cette section décrit les primitives de sécurité requises pour renforcer la sécurité dans les algorithmes d'agrégation.

L'intégrité des données : cette propriété assure que le contenu du message ne soit pas altéré durant sa transmission. Un adversaire près du nœud agrégateur peut changer le résultat de l'agrégat et l'expédier à la station de base en ajoutant quelques paquets manipulés sans être détecté. Cependant, l'intégrité des données est très critique car même en l'absence d'un adversaire, les données peuvent être altérées ou perdues à cause du médium sans fil.

La confidentialité des données: il est aussi nécessaire d'empêcher la fuite des données sensibles. Par exemple dans les applications de surveillance militaire ciblée pour planifier des attaques surprises, il est nécessaire d'assurer la confidentialité de l'information pour la réussite de la mission. Pour cela, un réseau de capteur qui utilise l'agrégation des données requière aussi la confidentialité des données agrégées.

L'authentification des données : garantie que les données reportées sont les même que les données originales (données authentiques) et proviennent bien d'une source (identification) fiable. Dans la sécurité de l'agrégation de données, l'identification et l'authentification sont importantes pour assurer le transfert de données légitimes entre les nœuds capteurs.

Disponibilité et fraîcheur de données: sachant qu'un réseau de capteur est déployé généralement pour la surveillance des événements en temps réel, il est donc important d'assurer que les données fournies par le réseau soient courantes (fraîches) et disponibles tout le temps. Cela veut dire qu'un adversaire ne peut rejouer les anciens messages ultérieurement. Cependant, la sécurité de l'agrégation des données doit être implémentée prudemment pour éviter un excès dans la consommation d'énergie, car si l'énergie est épuisée, les données ne seront plus disponibles

3.2 Les attaques contre le processus d'agrégation de données

Les réseaux de capteurs sans fil sont vulnérables à différentes types d'attaque (Roosta, et al., 2006) du à la nature du medium de transmission (diffusion), le déploiement dans un environnement hostile et distant et le manque de sécurité physique dans chaque nœud capteur. Cependant, le dommage causé par ces attaques varie dans les applications selon le modèle d'attaque supposé. Donc l'agrégation des données doit s'effectuer de manière sécurisée pour empêcher une lecture erronée de l'état de l'environnement surveillé. Dans cette section, nous

discutons les attaques particulières qui peuvent affecter l'opération de l'agrégation des données dans les WSN.

Compromission de nœud : les capteurs à large utilisation ne peuvent pas se permettre une protection physique inviolable. Par conséquent, ils peuvent facilement être interceptés et corrompus. En effet, un adversaire peut facilement compromettre un nœud et obtenir le matériel cryptographique sauvegardé au niveau de sa mémoire, et cela dans le but de corrompre les liens de communication ou d'injecter du code pour détourner son utilisation. L'adversaire peut alors déclencher des attaques internes (insider attacks), en contournant le chiffrement et les mots de passe du système de sécurité. En considérant le scénario de l'agrégation des données, les nœuds compromis peuvent authentifier des données erronées à leurs voisins, qui n'ont aucun moyen de distinguer les données erronées des données légitimes (Perrig, et al., 2004).

L'attaque déni de service: cette attaque est standard dans les WSN, en transmettant des signaux radio qui interfèrent avec les fréquences radio utilisées par le réseau de capteurs. Souvent cette attaque est appelée « Jamming ». Plus les capacités d'un adversaire augmentent, plus il affecte de grandes portions du réseau. Dans le contexte de l'agrégation, un exemple du déni de service, sera qu'un agrégateur refuse d'agréger les données et empêche les données de traverser à travers lui pour aller aux niveaux supérieurs.

L'attaque Sybille: Dans l'agrégation des données, les capteurs d'un WSN, ont besoin de coopérer afin d'exécuter cette tâche. Dans l'attaque Sybille, de nouveaux capteurs (malicieux) peuvent prendre l'identité d'autres capteurs légitimes dans le réseau (Roosta, et al., 2006). Cette attaque affecte les algorithmes d'agrégation de différentes manières (Alzaid, et al., 2008). Premièrement, un adversaire pourra créer des identités multiples pour générer des votes additionnels dans l'élection d'un agrégateur. Deuxièmement, le résultat de l'agrégat peut être affecté si l'adversaire est capable de générer des entrées multiples avec des valeurs collectées différentes. Troisièmement, quelques algorithmes utilisent des nœuds témoins pour valider l'agrégation des données et les données sont valides si seulement *n* témoins acceptent le résultat de l'agrégat. Cependant, un adversaire peut déclencher une attaque Sybille et génère *n* identités de témoins ou plus pour modifier le vote et persuader la station de base pour qu'elle accepte le résultat de l'agrégat.

L'attaque du relais sélectif: dans le relais sélectif, un nœud malicieux agit comme un trou noir et refuse de relayer les parquets. L'adversaire utilise les nœuds compromis pour relayer les messages. Dans le contexte de l'agrégation, tout nœud intermédiaire compromis a la capacité de déclencher l'attaque du relai sélectif et par conséquent affecter le résultat de l'agrégation.

L'attaque par rejeu: dans ce cas, un adversaire enregistre quelques messages dans le réseau sans aucune compréhension de leur contenu et les rejoue ultérieurement pour tromper l'agrégateur et par conséquent affecter le résultat de l'agrégat.

L'attaque stealthy: le but de l'adversaire est de persuader un utilisateur d'accepter de faux résultats d'agrégation, sans être détecté. Ces résultats divergent de manière significative des résultats corrects déterminés par les valeurs collectées. Nous étudierons cette attaque en particulier dans le chapitre 4.

3.3 Critères de performances d'un protocole de sécurité

Il est clair que la sécurité de l'agrégation est un problème très difficile à résoudre et requière plus d'attention durant le processus de conception. Selon les propriétés requises par l'application et selon le type d'attaque et le type d'adversaire, un algorithme de sécurité efficace pour les données agrégées doit répondre aussi bien que possible aux propriétés suivantes :

3.3.1 Un surcoût en communication réduit

Le but de l'opération de l'agrégation est de réduire le surcoût en communication, nommé également overhead. Ainsi un protocole de sécurité doit maintenir ce but sans ça l'agrégation n'aura aucune utilité. Sachant très bien que la sécurité en elle-même engendre un surcoût en communication. Toute la difficulté réside dans le maintien d'un compromis entre niveau de sécurité et surcoût de communication.

3.3.2 Scalabilité

Les techniques de sécurité de l'agrégation doivent fournir un haut niveau de sécurité pour les réseaux de petites tailles, mais également permettent le passage à l'échelle tout en maintenant ces caractéristiques.

3.3.3 Flexibilité

Les techniques de sécurité de l'agrégation doivent être capables de bien fonctionner dans tout type d'environnement et supporter le déploiement dynamique des nœuds capteurs.

3.3.4 Exactitude

Dans un algorithme de sécurité des données agrégées, il est très important d'assurer l'exactitude du résultat final de l'agrégat, car des décisions sont prises en fonction de ce résultat.

3.3.5 Généralité

Un algorithme de sécurité doit être aussi général que possible et appliquer différentes fonctions d'agrégation telles que maximum/minimum, moyenne, somme, médiane, etc.

4. TAXONOMIE DES ALGORITHMES DE SECURITE DES DONNEES AGREGEES

Plusieurs solutions innovantes et intuitives ont été proposées pour résoudre le problème de sécurité de l'agrégation dans les réseaux de capteurs sans fil. Dans cette section, nous survolons ces solutions en les classifiant en deux grandes familles : les algorithmes basées sur la cryptographie et les algorithmes basés sur la confiance. Voir la Figure 2.3.

4.1 L'agrégation sécurisée basée sur la cryptographie

Les besoins de sécurité tells que la confidentialité et l'intégrité dans l'agrégation des données deviennent vitaux lorsqu'un réseau de capteur est déployé dans un environnement hostile. La plupart des recherches dans ce domaine se focalisent sur des algorithmes basés sur la cryptographie. Dans certaines applications, il est essentiel de s'assurer que les informations qui sont transmises sur le réseau ne puissent être interceptées et lues par des personnes non-autorisées. Elles doivent donc être chiffrées. Mais le chiffrement et l'agrégation sont deux concepts qui ne vont pas très bien ensemble.

Nous distinguons deux techniques de l'agrégation sécurisée: les techniques basées sur les données en clairs (confidentialité saut par saut) et les techniques basées sur les données chiffrées (confidentialité de bout en bout).

4.1.1 Les techniques basées sur les données en clair

Dans les techniques de l’agrégation sécurisée basées sur les données en clair, la fonction d’agrégation est effectuée par l’agrégateur sur des données en clair. Cela veut dire que l’agrégateur doit déchiffrer les données de ses fils (dans une architecture plate) ou des membres de son cluster (dans une architecture clustérisée) avant d’effectuer la fonction d’agrégat telle que la somme ou la moyenne. Pour mettre en œuvre cette technique, il existe deux *solutions simples* (mais naïves) à ce problème.

La première consiste à configurer tous les nœuds du réseau avec une clé de groupe. Chaque nœud chiffre alors ses données avec cette clé en utilisant un algorithme de chiffrement symétrique standard, comme AES ou RC5, et envoie le résultat à son père. Ce nœud déchiffre alors tous les messages reçus de ces enfants, ajoute sa mesure, chiffre le résultat avec la clé de groupe et envoie le message chiffré à son père. Les messages sont ainsi relayés et agrégés de nœud en nœud jusqu'à la station de base. La station de base peut alors déchiffrer le message et retrouver la somme des données. Cette solution a plusieurs inconvénients : (1) Elle est peu sûre car il suffit de compromettre un seul nœud pour découvrir la clé de groupe et déchiffrer l'ensemble des messages. (2) Elle est peu efficace car chaque nœud doit déchiffrer plusieurs messages et en chiffrer un.

Une deuxième solution consiste à utiliser, au lieu d'une clé de groupe, une clé différente lien par lien. En d'autres termes, dans cette solution, chaque nœud établit une clé secrète avec chacun de ses voisins. Il déchiffre alors les données reçues de ses fils, les agrège, puis chiffre le résultat avec la clé qu'il possède avec son père. Cette solution est meilleure, en terme de sécurité, que la précédente car la compromission d'un nœud ne révèle que les clés utilisées par ce nœud et non une clé globale. Cependant elle a les inconvénients suivants : (1) La compromission d'un nœud près de la station de base permet d'obtenir un agrégat qui est significatif car chaque nœud a accès aux données agrégées envoyées par ses fils. (2) Elle nécessite l'établissement de clé entre chaque nœud voisin, ce qui n'est pas trivial. (3) Comme la solution précédente, elle est relativement coûteuse.

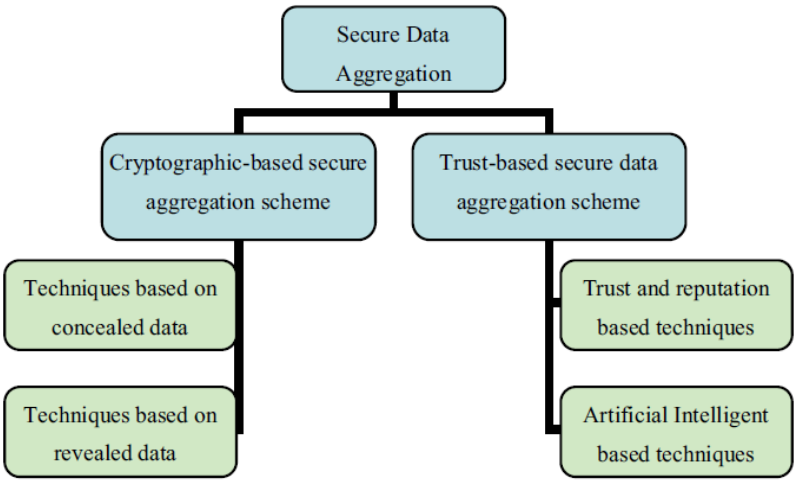


Figure 2. 3: Classification des algorithmes d'agrégation sécurisée (Labraoui, et al., 2011(a)).

Les travaux relatifs

Plusieurs algorithmes d'agrégation basés sur les données en clair fournissent une opération d'agrégation efficace et prennent en considération l'intégrité des données en particulier. Dans ce qui suit nous survoleront les travaux relatifs les plus significatifs.

1. Secure aggregation for wireless networks : 2003.

Hu et Evans ont été les pionnés dans le domaine de l'agrégation sécurisé. Ils ont proposé le premier algorithme sécurisé (SDA) capable de résister à un dispositif de compromission de clé (Hu, et al., 2003). Ils supposent que le réseau s'auto-organise sur un arbre où les nœuds internes, y compris le nœud racine, sont responsables de l'agrégation. Cet algorithme détecte les comportements malveillants des nœuds capteurs en exécutant deux phases: l'agrégation retardée (*delayed aggregation*) et l'authentification retardée (*delayed authentication*). Dans l'agrégation retardée, l'agrégation des données envoyées par les nœuds de niveau *k* de l'arbre n'est pas effectuée par les nœuds de niveau *k-1*, mais réalisée par des nœuds de niveau *k-2* (grand-père), où le niveau 0 est la racine de l'arbre. Une fois que la station de base reçoit les résultats d'agrégats partiels de la racine de l'arbre, elle révèle les clés d'authentification utilisée précédemment, pour permettre aux nœuds agrégateurs d'authentifier les messages qu'ils ont reçus de leurs enfants, et ainsi de détecter et de signaler toute fraude à la Station de base. Cette dernière, annule et rejette les résultats d'agrégation reçus en cas de détection de fraude. En reportant l'agrégation d'un saut plus loin, l'algorithme garantie l'intégrité des données lorsque deux nœuds consécutives sont compromis. Cependant cet algorithme ne

détecte pas la compromission si deux nœuds consécutifs (fils et père) sont compromis dans l'arbre.

2. **SIA : Secure information aggregation in sensor networks: 2003.**

Przydatek et al. ont proposé un algorithme (SIA) (Przydatek, et al., 2003) qui permet à l'agrégateur d'accepter les données avec une grande probabilité si le résultat de l'agrégat est dans une limite acceptable, ou de rejeter le résultat s'il est hors limite. En construisant un mécanisme d'échantillonnage aléatoire et une vérification interactive, cet algorithme propose plusieurs protocoles pour calculer de manière sécurisée les fonctions d'agrégations telles que la médiane, le minimum/maximum, le comptage et la moyenne.

3. **ESPDA: Energy Efficient and Secure Pattern-Based Data Aggregation for Wireless Sensor Networks: 2003.**

Çam et al. ont proposé un algorithme des données agrégées efficace en terme d'énergie (ESPDA) (Çam, et al., 2003). ESPDA est applicable dans les réseaux de capteurs hiérarchisés en clusters. Au départ, le cluster-head invite les nœuds capteurs de son cluster à émettre leur *pattern code* correspondant aux données collectées. Si plusieurs nœuds émettent le même *pattern code* que celui du cluster-head, alors seulement un d'autre eux est autorisé à envoyer la données au cluster-head. De ce fait le surcoût de communication dans le réseau est considérablement réduit.

4. **Secure A witness-based approach for data fusion assurance in wireless sensor networks: 2003.**

Du et al. ont proposé (WDA) (Du, et al., 2003) un protocole d'agrégation de données basée sur des témoins (*Witness*) pour assurer la validation des données envoyées du nœud agrégateur jusqu'à la station de base. Afin de prouver la validité du résultat de l'agrégat, l'agrégateur doit fournir des preuves de la part de plusieurs nœuds témoins. Un nœud témoin est un nœud dédié pour la surveillance et effectue lui aussi l'agrégation comme le fait l'agrégateur, mais ne transmet pas son résultat à la SB. Par contre chaque nœud témoin, calcule le code d'authentification du message (MAC) du résultat et le transmet au nœud agrégateur en guise de preuve.

5. **SRDA: Secure Reference-Based Data Aggregation Protocol for Wireless Sensor Networks: 2004.**

Sanli et al. (Sanli, et al., 2004) ont propose une nouvelle technique de sécurité des données agrégées en employant le concept des données différentielles (SRDA). L'idée de base, est que chaque nœud capteur transmet les données différentielles au lieu de données brutes. Les données brutes captées par le capteur sont comparées avec une donnée référence,

et donc seulement la différence est transmise. Par conséquent, l'agrégation différentielle a un potentiel de réduire la quantité d'information transmise entre les capteurs et le cluster-head. La motivation de base de cette idée, est qu'un changement significatif dans les mesures collecté n'arrive que si un événement important (exemple, la température du réseau lors d'un déclenchement de feu) se produit dans l'environnement. En général, ces évènements se produisent beaucoup moins fréquemment que les événements ordinaires dans un réseau de capteurs. Il serait donc inutile d'émettre des données semblables à chaque agrégation.

6. Resilient aggregation in sensor networks: 2004.

Wagner (Wagner, 2004) a étudié la sécurité inhérente de quelques fonctions d'agrégation, mais il a considéré seulement le niveau d'impact qu'un nœud compromis peut avoir sur le résultat final de l'agrégat. Pour cela, il a proposé un modèle mathématique (RA) pour évaluer la sécurité de plusieurs techniques d'agrégation et décrire les meilleures méthodes pour sécuriser l'agrégation de données. Dans son étude, il déclare que la fonction médiane est la meilleur pour résumer des statistiques. En outre, Wagner déclare que les fonctions de réglage et les fonctions qui permettent de tronquer les données peuvent être utilisées pour renforcer la sécurité de plusieurs primitives d'agrégation en éliminant les données aberrantes. Cependant, ce travail concerne la sécurité des fonctions d'agrégation et non pas la sécurité de l'agrégation en elle-même.

7. SDAP: a secure hop-by-hop data aggregation protocol for sensor network: 2006.

Yang et al. Ont proposé SDAP (Yang, et al., 2006), un algorithme de l'agrégation sécurisée saut par saut basé sur deux principes : « diviser et régner » et « garantir et certifier». Dans cet algorithme une nouvelle technique de groupement statistique est utilisée pour partitionner dynamiquement les nœuds dans une topologie d'arbre en sous-arbres. Une agrégation saut par saut basées sur la garantie est effectuée dans chaque sous-arbre pour générer un agrégat de groupe. La station de base identifie les sous-arbres suspects sur la base des ensembles d'agrégat de groupe. Finalement chaque sous-arbre suspecté participe à une procédure d'attestation pour prouver l'exactitude de son agrégat de groupe.

8. Secure hierarchical in-network aggregation in sensor networks :2006

Chan et al. (Chan, et al., 2006) a conçu un nouvel algorithme de vérification (SHDA) avec lequel la station de base peut détecter si l'agrégat calculé a été falsifié. Les auteurs ont fait une extension de l'algorithme SIA en appliquant « la preuve de garantie de l'agrégation » (aggregate-commit-prove framework) dans tout le réseau de manière distribuée au lieu de l'appliquer au model d'un agrégateur unique. En général, cet algorithme offre exactement les mêmes propriétés que celle de SIA, à savoir, intégrité, authentification et confidentialité des

données. Chaque capteur père effectue la fonction d'agrégation toutes les fois qu'il a reçu des données de ses nœuds fils. Il doit créer une garantie à l'ensemble des entrées utilisées pour calculer le résultat de l'agrégat, en utilisant un arbre de hachage de Merkle. Ensuite, il transmet le résultat de l'agrégat et la garantie au nœud père jusqu'à arrivé à la station de base (SB). Lorsque la SB reçoit les valeurs de garanties finales, elle les rediffuse au reste du réseau en utilisant un broadcast authentifié. Chaque nœud est responsable pour vérifier si ses contributions ont été ajouté aux données agrégées ou non. Lorsque ses valeurs collectées sont ajoutées, il envoie un code d'authentification à la SB. Pour une efficacité dans la communication, les codes d'authentification sont agrégés tout au long du trajet jusqu'à la SB. Cependant, l'oubli d'un seul code d'authentification pour une quelconque raison, mène la SB à rejeter le résultat de l'agrégat. En outre, il est apparent que cet algorithme, malgré l'efficacité de sa vérification, génère un délai et un surcoût en communication assez importants.

9. **Secure data aggregation without persistent cryptographic operations in wireless sensor networks: 2007.**

Kui et al. (Kui, et al., 2007) ont propose un protocole basé sur l'infrastructure des cliques. Une topologie en arbre est construite de telle sorte à ce que chaque nœud fils puisse être à un saut de son nœud père. Les nœuds fils et pères forment ainsi une clique. Ce protocole utilise la méthode des chiens de garde « watchdog » pour vérifier la valeur de l'agrégat du nœud père. Cependant les nœuds capteurs utilisent les algorithmes de cryptographie uniquement quant une activité malveillante a été détecté. Dans SAT chaque nœud fils surveille par écoute toutes les données acheminées vers son nœud père. Lorsqu'une agrégation de données est requise, un algorithme de vote pondéré est utilisé pour décider si l'agrégateur s'est bien comporté ou bien il a triché. Si un comportement malveillant de l'agrégateur a été détecté, alors SAT refait une reconstruction locale de la topologie de façon à exclure l'agrégateur de l'arbre de l'agrégation.

10. **FAIR: Fuzzy-based Aggregation providing In-network Resilience for real-time Wireless Sensor Networks: 2009.**

Emiliano et al. ont propose un algorithme Fuzzy-based FAIR pour une agrégation de données robuste en temps réel (Emiliano, et al., 2009). Comme le protocole de Du et al. (Du, et al., 2003), des nœuds témoins sont utilisés pour confirmer le résultat des nœuds agrégateurs afin d'assurer l'intégrité des données durant l'agrégation. Cependant ces nœuds témoins non seulement confirment le résultat mais aussi agrègent et expédient les données eux-mêmes. De ce fait le protocole est plus robuste et il n'y a pas de dépendance sur un agrégateur unique.

4.1.2 Les techniques basées sur les données chiffrées

Une solution de bout-en-bout de l'agrégation sécurisée serait préférable car la compromission d'un nœud ne fournirait aucune information sur l'agrégat ou les données envoyées par les autres nœuds du système. Cette solution est appelée agrégation des données cachée (concealed data aggregation : CDA). Contrairement aux techniques basées sur les données en clair, CDA consiste à effectuer la fonction d'agrégation sur des données chiffrées. La solution idéale serait que chaque nœud chiffrerait ses données avec une clé qu'il partagerait avec la station de base et avec laquelle les nœuds intermédiaires manipuleraient des données chiffrées sans jamais accéder aux données en clair. Seulement la SB peut alors déchiffrer les résultats.

Les bases fondamentales de CDA sont des méthodes qui fournissent la propriété d'homomorphisme qu'on appelle *privacy homomorphism* (PH). Un algorithme de chiffrement E() est homomorphique, si pour E(X) et E(Y) ; on peut obtenir E(X*Y) sans déchiffrer X et Y pour une certaine opération *.

Ce concept a été introduit par Rivest et al. (Rivest, et al., 1978) En 1978. Les deux variations les plus connues des PH sont le chiffrement homomorphique par l'addition et le chiffrement homomorphique par la multiplication. Ce dernier fournit la propriété suivante : $E(X\times Y)=E(X)\otimes E(Y)$.

Les travaux relatifs

1. **CDA: Concealed Data Aggregation for Reverse Multicast Traffic wireless Sensor Networks: 2005.**

Girao et al (Girao, et al., 2005) ont propose un protocole CDA (CDAM) base sur le chiffrement homomorphique (PH) proposé dans (Domingo, 2002). Ils ont réclamé que pour le scénario de l'agrégation dans les WSN, le niveau de sécurité est assez adéquat et la méthode PH proposée peut être utilisée pour le chiffrement.

2. **SecureDAV: A Secure Data Aggregation and Verification Protocol for Sensor Networks: 2004.**

Mahimkar et al. (Mahimkar, et al., 2004) ont développé un protocole sécurisé pour l'agrégation et la vérification (SDAV) qui a pour rôle d'assurer que la SB accepte le résultat avec une grande fiabilité. Cette fiabilité est assurée même si le cluster-head est compromis ou qu'il existe un nombre de collision moins que *t* nœuds compromis dans le cluster. La vérification de l'intégrité des données est effectuée en utilisant un arbre de hachage de Merkle

pour éviter la malveillance des cluster-heads. Les auteurs de ce travail, ont aussi présenté une structure de réseau hiérarchique pour établir la clé du cluster dans le réseau en utilisant les courbes elliptiques. La clé du cluster est secrète pour tous les nœuds capteurs pour éviter son interception. Une fois que l'agrégateur reçoit les valeurs collectées du même cluster, il les agrège et diffuse la moyenne des valeurs reçues à tous les membres du cluster. Chaque nœud dans le cluster compare sa mesure avec la moyenne reçue de l'agrégateur. Ensuite, il signe partiellement la moyenne si et seulement si la différence entre la moyenne reçue et sa mesure est au dessous d'un certain seuil. L'agrégateur, combine alors les signatures partielles pour former une signature complète du résultat de l'agrégat et l'expédie à la SB. Cependant, ce protocole génère un surcoût de communication très élevé lors de la validation des données, et supporte uniquement la fonction d'agrégation « moyenne ».

3. Efficient Aggregation of Encrypted Data Wireless Sensor Network: 2005.

Castellucia et al. Ont proposé un protocole simple basé sur le chiffrement par flot et l'homomorphique par l'addition (HSC) qui permet une agrégation de données efficace (Castelluccia, et al., 2005). Le nouveau chiffrement remplace l'opération du xor (OU-exclusif) par une addition modulaire qui est très appropriée pour les capteurs avec une capacité de calcul réduite. L'agrégation basée sur ce chiffrement peut être utilisée de manière efficace pour le calcul des fonctions statistiques telles que la moyenne, la variance et la déviation standard des données collectées ; tout en achevant un gain considérable de la bande passante. Les inconvénients de ce protocole sont l'overhead important généré dans un réseau non fiable et le passage à l'échelle.

4. PDA: Privacy-preserving Data Aggregation in Wireless Sensor Networks: 2007.

Wenbo et al. (Wenbo, et al., 2007) ont proposé deux solutions différentes : CPDA et SMART. La première proposition est une solution pour préserver la confidentialité contre d'autres nœuds capteurs dans l'agrégation de données pour la fonction d'addition. On remarque qu'une extension de CPDA peut être faite pour fournir aussi bien la confidentialité contre la station de base et la perte de données. Cependant, comme ça été mentionné par les auteurs de ce travail, ce protocole génère un surcoût très élevé. En effet, ils emploient les ensembles d'anonymisation, où chaque nœuds parmi *C* nœuds dans le cluster, envoie (et reçoit) *C-1* messages ; ce qui a pour résultat d'avoir $O(C^2)$ messages envoyés (et reçus) dans chaque cluster et pour chaque opération d'agrégation. En outre, chaque nœud doit chiffrer et déchiffrer *O(C)* messages, et le cluster-head doit calculer l'inverse de la matrice *C*C*, pour chaque procédure d'agrégation. La deuxième proposition, SMART, est plus efficace que CPDA. Cependant, SMART ne fournit pas de confidentialité contre la station de base, et

souffre du même problème de la perte des messages que dans le travail de (Castelluccia, et al., 2005).

5. Secure data aggregation with multiple encryptions: 2007.

Önen et al. (Onen, et al., 2007)] et Castellucia (Castellucia, 2007) ont propose un nouveau protocole qui combine le chiffrement homomorphique (PH) avec le chiffrement multiple, (PHM1 et PHM2). Ces deux travaux sont presque similaires mais ont été développés en parallèle et indépendamment. Le PH est une technique fondamentale de chiffrement qui permet aux capteurs d'agréger leurs mesures cryptées, alors que le chiffrement multiple assure que le résultat de l'agrégat et les mesures individuelles restent non dévoilés à tous les nœuds capteurs intermédiaires lors de leur transfert à la SB. Cette solution assure une confidentialité de bout en bout et permet le passage à l'échelle de manière efficace. Elle améliore également les performances de la bande passante de l'algorithme décrit en (Castelluccia, et al., 2005), et permet de résister à n nœuds compromis.

6. A Secure Data Aggregation Scheme for Wireless Sensor Networks: 2007.

Ren et al. (Ren, et al., 2007) ont proposé une agrégation sécurisée pour les réseaux de capteurs clustérisés (ECCM), en combinant la confidentialité de bout en bout en utilisant l'homomorphisme, et l'authentification saut par saut basée sur les courbes elliptiques (ECC-MAC). Le but de conception de cette solution est de protéger les massages sensibles contre la fuite, et de détecter l'injection, ou la fabrication et la suppression de messages aussi vite que possible. i.e avant l'arrivée à la station de base. Pour le premier but, les auteurs adoptent le chiffrement basé sur le PH pour éviter l'interception des données originales ; pour le deuxième but, l'authentification saut par saut est adoptée pour vérifier les messages à travers le code d'authentification de massage (MAC). Du moment que l'établissement de la clé partagée est effectué en utilisant les courbes elliptiques ECC, la difficulté de découvrir la clé est un problème de logarithme discret. Ce protocole est approprié aux réseaux de capteurs car le surcoût additionnel généré est réduit.

7. Concealed data aggregation in heterogeneous sensor networks using privacy homomorphism: 2007.

Ozedemir a proposé un protocole nommé CDAP (Ozdemir, 2007), qui possède les avantages du chiffrement homomorphique basé sur les clés symétriques, pour achever une confidentialité de bout en bout et l'agrégation des données en même temps. L'auteur, déclare que la confidentialité homomorphique basée sur les clés symétriques génère un surcoût en calcul très élevé, par rapport aux ressources limitées des nœuds de capteurs ordinaires. Pour surmonter ce problème, CDAP utilise un ensemble de capteurs ayant des plus de ressources

que les nœuds ordinaires, nommés AGGNODEs. Ces supers nœuds pourront effectuer le chiffrement homomorphique basé sur les clés symétriques. Dans ce protocole, après le déploiement du réseau, les AGGNODEs établissent les paires de clés avec les nœuds voisins pour assurer la confidentialité entre les nœuds, déchiffrent les données reçues de leurs voisins, agrègent les données et les transmettent à la SB après chiffrement. Seule la station de base peut déchiffrer le résultat final en utilisant une clé privée.

8. Securing wireless sensor networks against aggregator compromises: 2008.

Claveirole et al. (Claveirole, et al., 2008) ont propose trois nouvelles, nommées (a) *Agrégation Multi-chemins Secrète (SMA),* (b) *Agrégation Multi-chemins* Dispersée (DMA), et (c) Agrégation Multi-chemins Dispersée avec Authentification (A-DMA). L'idée principale de ces approches est d'exploiter plusieurs chemins jusqu'à la station de base. En fait, un capteur peut séparer ses mesures en n messages distincts tels que t messages soient nécessaires pour reconstruire les mesures. En envoyant chacun des messages sur des chemins disjoints, un capteur peut s'assurer que les nœuds intermédiaires n'auront pas une connaissance complète des données. Dans un tel scénario, SMA garantit la confidentialité des données grâce à la cryptographie à seuil. DMA et sa version authentifiée, DMA-A, s'intéressent à la disponibilité en dispersant l'information à travers les différents chemins. En fonction de la technique utilisée et de ses paramètres, il est possible d'obtenir différents degrés de résistance aux attaques de déni de service, aux écoutes passives, et aux falsifications de données. En utilisant l'agrégation multi-chemins secrète, on peut garantir qu'un sous-ensemble de nœuds compromis ne transmet aucune information au sujet des mesures. Cela possède un certain surcoût. En utilisant l'agrégation multi-chemins dispersée, on obtient un surcoût optimal, mais avec un niveau de confidentialité plus faible. En fonction de l'application ou du scénario, une approche peut présenter plus ou moins d'avantages sur l'autre.

4.1.3 Discussion

Le champ de la cryptographie dans le traitement de données dans les réseaux est un axe de recherche très prometteur, et introduit beaucoup de challenges. Pour l'instant, l'approche la plus simple dans les WSN est celle qui achève une confidentialité saut-par-saut et une intégrité des données. Cependant, les nœuds capteurs peuvent être capturés et la révélation de leur matériel cryptographique peut mener à la révélation des données originales et des résultats partiaux des agrégats. En particulier, le package des nœuds capteur à large

utilisation est affecté par son bas coût, et ne peut se permettre une résistance aux attaques physiques. Ce qui facilite à l'attaquant la tâche pour corrompre le nœud capteur.

Pour surmonter ce problème, des techniques ont été proposées qui exploitent la confidentialité de bout en bout conjointement aux distributions de clés particulières, au chiffrement homomorphique ou au chiffrement à clé publique. Cependant il ne faut pas oublier qu'un nœud capteur a des ressources très limités en énergie, en mémoire et en calcul CPU. Ce qui contraint les concepteurs des techniques de sécurité, à mettre en œuvre des solutions « lights » qui prennent en considération ces contraintes techniques.

Pour trouver un compromis réaliste entre le surcoût du calcul, le délai et la crédibilité des données (trustworthiness), la sélection de méthodes cryptographiques appropriées est fondamentale. Par conséquent, divers services cryptographiques sont requis pour certaines applications et l'utilisation commune des algorithmes à clé symétriques tels que : AES et MAC n'ont pas uniquement imposé des problèmes tels que la gestion et la protection des clés, mais peuvent être en même temps bien plus coûteux.

La cryptographie des courbes elliptiques (ECC) a émergé comme un cryptosystème à clé publique *attractif* pour les réseaux de capteurs sans fil. En comparaison aux cryptosystèmes traditionnels comme RSA, ECC offre une sécurité équivalente avec des clés à petites tailles, ce qui a comme conséquence d'accélérer les calculs, de minimiser la consommation d'énergie, de sauvegarder la mémoire et la bande passante. Le gouvernement américain a récemment approuvé la cryptographie des courbes elliptiques.

Malgré la diversité et l'efficacité prouvée des solutions basées sur la cryptographie, la majorité des solutions supposent que les nœuds capteurs sont dignes de confiance et rapportant des données de manière correcte. Cependant dans la pratique, les capteurs sont déployés dans des environnements ouverts sans surveillance, et donc sont exposé aux attaques physiques. Lorsqu'un nœud est compromis, l'attaquant peut injecter des données erronées dans le réseau. Cependant, nous déduisons que la vue conventionnelle de la sécurité basées dur la cryptographie seule est *insuffisante* à cause des caractéristiques spécifiques et des nouveaux comportements malveillants rencontrés dans des réseaux ouverts. Quoique, la cryptographie peut assurer l'intégrité, la confidentialité et l'authentification, elle échoue face aux attaques internes (insider attacks). Cela nécessite un système complémentaire qui peut faire face à des attaques internes.

4.2 L'agrégation sécurisée basée sur la confiance

Comme nous l'avons déjà cité, les réseaux de capteurs sans fil sont déployés dans des territoires sans surveillance et souvent hostiles, et sont sujets aux captures physiques par des attaquants. Du moment que le camouflage des capteurs n'est pas une solution viable (Ganeriwal, et al., 2004), les capteurs peuvent être modifiés pour avoir un comportement malveillant et perturber le réseau en entier. Ceci permet à l'attaquant d'accéder au matériel cryptographique du capteur et de déclencher des attaques de l'intérieur du réseau, en surpassant le chiffrement et les mots de passe des systèmes de sécurité. Le nœud ainsi compromis, peut authentifier ses données erronées à ses voisins, qui n'ont aucun moyen de distinguer les fausses données des données légitimes (Perrig, et al., 2004). Les systèmes de confiance et de réputations ont été proposés comme un complément attractif à la cryptographie pour sécuriser les WSN. Ils fournissent l'habilité de détecter et d'isoler les nœuds malicieux qui se comportent de manière inappropriée dans le contexte des réseaux de capteurs spécifiques.

Récemment, une attention a été donnée au concept de la confiance pour augmenter la sécurité et la fiabilité des réseaux Ad Hoc (Gursel, et al., 2008) et les réseaux de capteurs (Ganeriwal, et al., 2004). La notion de confiance utilisée dans ce travail, est brièvement définie comme : « la confiance est le degré de croyance au sujet du futur comportement des autres entités, qui est basé sur ceux des expériences antérieures avec l'observation des autres actions ». La réputation est une autre notion complexe à envergure multi disciplinaires. Elle est assez différente mais facilement confondue avec la confiance. La base mathématique pour la gestion de la réputation est fondée sur les statistiques et la probabilité (Peter, et al., 2007). En outre, la réputation est basée sur la collection de l'évidence du bon et mauvais comportement entrepris par d'autres entités. Elle est basée sur les expériences antérieures pour une entité donnée, alors que la confiance n'est pas limitée à cela.

Dans cette section, nous allons étudier les protocoles proposés basés sur la confiance et la réputation pour sécuriser les données agrégées dans les WSN. Nous les classifions en deux catégories : les protocoles basés sur la confiance et la réputation, et les protocoles basés sur l'intelligence artificielle.

4.2.1 Les protocoles basés sur la confiance et la réputation

Parmi les travaux qui sont basés sur la confiance et la réputation, nous citons les travaux suivant :

1. **Trust-based aggregation in wireless sensor networks: 2005**

Hur et al. Ont proposé dans (Hur, et al., 2005) un protocole d'agrégation basée sur l'évaluation de la confiance locale (LTE) dans les WSN. Ce mécanisme d'évaluation de confiance locale est approprié aux réseaux de capteurs ayant des ressources limités. Le degré de confiance d'un nœud est calculé sur la base de plusieurs facteurs d'évaluation de confiance, tels que la durée de vie de la batterie, le taux de communication, le résultat du captage et le niveau de consistance. Chaque nœud capteur calcule seulement la confiance cumulée des nœuds voisins. Avant l'agrégation des données, les nœuds capteurs élisent par vote majoritaire un nœud agrégateur ayant la plus haute valeur de confiance dans une région définie. Un processus d'accord de confiance est nécessaire, car la valeur de confiance d'un nœud est évaluée par ses nœuds voisins, et que n'importe quel nœud ignore sa propre valeur de confiance. Les données captées par différents nœuds sont agrégées selon les valeurs de confiances convenues pour chaque capteur membre de chaque zone. Une donnée erronée d'un nœud malicieux ou compromis dont la valeur de confiance est inférieure à celle des nœuds légitimes sera exclue de l'agrégation. Ainsi un filtrage de données est effectué en fonction des valeurs de confiance avant toute opération d'agrégation. L'inconvénient de ce protocole est qu'il considère que l'évaluation de la confiance est calculée uniquement par des nœuds dignes de confiance.

2. **A trust based framework for secure data aggregation on wireless sensor networks : 2006**.

Dans (Zhang, et al., 2006), les auteurs ont proposé un protocole (TKL) basé sur la confiance pour sécuriser l'agrégation des données dans les WSN. Ce protocole est basé sur le modèle bayésien et la probabilité de la béta distribution. Ils ont premièrement évalué la confiance dans des capteurs individuels basée sur la distance Kullback-Leibler (KL) et l'entropie relative. L'idée était de calculer la distance entre le comportement idéal d'un nœud et le comportement d'un nœud actuel. Les auteurs ont assigné une valeur de confiance pour agréger les données d'un capteur. Basé sur la confiance des données, le protocole calcule une opinion concernant la croyance et l'incertitude de l'agrégation au moyen de l'opération de consensus (Audun, 2001). Néanmoins, cette approche met du temps pour établir une réputation stable sur des nœuds de capteurs. La réputation d'un nœud basée sur le carré inverse de sa distance KL, souffre d'oscillations sévères pour la première évaluation de réputation. L'algorithme met donc beaucoup de temps pour converger.

3. Secure aggregation in sensor networks using neighbourhood watch: 2007.

Rabinovich et al. (Rabinovich, et al., 2007) ont proposé un mécanisme pour détecter et atténuer l'attaque stealthy contre les réseaux de capteurs en utilisant la validation distribuée des contraintes localisées et le routage randomisé (RTM). Plus spécifiquement, ils considèrent les applications des réseaux de capteurs où les mesures sont spatialement corrélées. Cette corrélation est exprimée sous la forme de contraintes de mesures. Les nœuds capteurs observent leurs voisins durant la transmission et font des plaintes si les données de leurs voisins ont violé les contraintes en comparant avec leurs propres valeurs. La protection contre les agrégateurs compromis est assurée par la construction d'un arbre de délivrance randomisée. A chaque fois, les nœuds capteurs envoient les données à la SB, en incluant la mesure courante et la plus petite des mesures récentes. Cela permet à la SB de détecter une attaque lorsqu'une petite fraction des capteurs sont compromis. Basé sur les plaintes, la station de base utilise le système de béta réputation (Jsang, et al., 2002) pour identifier les nœuds compromis. Pour tester l'exactitude de leur protocole, les auteurs ont développé un simulateur pour analyser le temps de détection des nœuds compromis, le nombre de nœuds compromis détectés avec succès et l'overhead de communication induit par le protocole de sécurité.

4. Secure and Reliable Data Aggregation for Wireless Sensor Networks: 2007.

Ozdemir (Ozdemir, 2007) a proposé un protocole d'agrégation sécurisée et fiable nommé SELDA, basé sur le degré de confiances des nœuds capteurs et des données agrégées. L'idée de base de SELDA est que les nœuds capteurs observent les actions des nœuds voisins pour développer des niveaux de confiance pour l'environnement en utilisant la fonction de la béta distribution. Les nœuds capteurs échangent leurs niveaux de confiances avec leurs voisins pour former un enchainement de confiance qui leur permet de déterminer des chemins fiables et sûrs pour les données agrégés. Basé sur les niveaux de confiance, les nœuds capteurs transmettent leurs mesures à l'agrégateur à travers un ou plusieurs chemins sûrs. Durant l'agrégation des données, l'agrégateur pondère la donnée sur la base des niveaux de confiance du nœud émetteur. Pour prévenir les attaques de fabrication et de relai sélectif des nœuds compromis, les auteurs ont proposé un algorithme de transmission des données à travers des chemins sécurisés pour délivrer les données aux agrégateurs. Cet algorithme sélectionne secrètement certains chemins sur la base de la fiabilité des chemins et sauvegarde l'identité des chemins sélectionnés. L'importance de SELDA provient du fait que le mécanisme de surveillance peut détecter si la donnée agrégée a été affectée par une attaque de déni de

service ; la simulation a démontré que SELDA augmente la fiabilité de l'agrégation des données aux dépens d'un overhead en communication tolérable.

5. **RSDA: Reputation-based Secure Data Aggregation in Wireless Sensor Networks:2008.**

Alzaid et al. (Alzaid, et al., 2008) ont proposé un protocole de sécurité de l'agrégation basé sur la réputation (RSDA) qui se focalise sur l'amélioration de la disponibilité des données et l'exactitude du résultat de l'agrégat. Des nœuds capteurs ordinaires sont regroupés dans des cellules et dans chaque cellule, un capteur C_{rep} est élu pour représenter sa cellule. Initialement, C_{rep} est choisi aléatoirement puisque tous les nœuds ont la même valeur de réputation au début. Le C_{rep} est responsable de la confirmation de la mesure de sa cellule C_{read} (reportée par les autres membres de la cellule), il est également responsable d'agréger cette valeur C_{read} avec d'autres valeurs (si la cellule est une cellule intermédiaire), et de transmettre le résultat à la cellule supérieure. En surveillant les activités du voisinage, chaque nœud capteur évalue le comportement de son représentant C_{rep} et des membres de sa cellule pour filtrer les données inconsistantes en présence de plusieurs nœuds compromis (< t dans chaque cellule). RSDA est sensé détecter les nœuds compromis et les mettre en liste noire ; ce qui permet d'étendre la durée de vie du réseau et protéger l'exactitude du résultat d'agrégat. RSDA utilise la fonction de la densité de la béta probabilité (PDF) pour mettre à jour la valeur de réputation de chaque nœud capteur. Cette fonction est utilisée pour sa flexibilité, ses bases statistiques robustes et surtout sa simplicité par rapport à des capteurs à ressources limitées.

4.2.2 Les protocoles bases sur l'intelligence artificielle

L'intérêt d'appliquer des techniques d'intelligence artificielle pour sécuriser les réseaux de capteurs a commencé à émerger ces dernières années mais de manière timide. Les auteurs du travail (Mukherjee, et al., 2007) ont utilisé les réseaux de neurones basés sur les techniques d'apprentissage pour modeler spatialement les données collectées. Dans (Servin, et al., 2007) les auteurs ont appliqué des techniques d'apprentissage pour la détection d'intrusions. Dans (Wu, et al., 2006) et (Ruairi, et al., 2007), les auteurs ont basé leur système de détection sur les systèmes multi-agents. Cependant, la problématique de l'agrégation n'a pas été prise en considération par aucune de ces recherches.

1. **Robust Trust Mechanisms for Monitoring Aggregator Nodes in Sensor Networks: 2008.**

Dans (Gursel, et al., 2008), les auteurs ont proposé le premier mécanisme qui combine les statistiques et les techniques d'intelligence artificielles (AIF) pour la détection robuste des nœuds malicieux dans les réseaux de capteurs sans éliminer inutilement des nœuds honnêtes, par exemple les descendants des nœuds malicieux. En particulier, les mécanismes des tests d'hypothèses sont utilisés par les nœuds capteurs fils pour estimer la probabilité d'erreur rapportée par un nœud fils durant une époque. Alors que des algorithmes d'apprentissage sont utilisés pour mettre à jour la réputation durant plusieurs époques successives. L'utilisation de techniques statistiques n'est pas suffisante pour surveiller les erreurs dans les réseaux de capteurs, à cause des variations de l'environnement et les caractéristiques des erreurs de mesures intrinsèques. Donc, des tests significatifs doivent être effectués pour observer les déviations entre les mesures individuelles reportées et les valeurs des agrégats reportées par les nœuds pères. Pour créer un système robuste, la réputation doit être cumulée durant des époques successives ; et si des déviations consistantes sont observées, alors le nœud père est étiqueté comme nœud malicieux. L'inconvénient de ce protocole est qu'un taux d'apprentissage élevé peut augmenter les chances de détection mais en contre partie introduit un taux de faux positifs inacceptable.

4.2.3 Discussion

Les systèmes basés sur la confiance et la réputation ont récemment été suggérés comme mécanismes efficaces de sécurité pour les environnements ouverts tels qu'Internet. Une recherche considérable a été menée pour modeler et gérer le concept de confiance et la réputation. Quelques travaux de recherche ont démontré que l'estimation de la confiance et la réputation d'un nœud est une approche efficace dans les environnements distribués pour améliorer la sécurité, soutenir la prise de décision et promouvoir la collaboration des nœuds. L'évaluation du degré de confiance des nœuds capteur est basée par exemple sur la théorie des probabilités. L'information de confiance est donc utilisée pour déterminer si la donnée reportée par un capteur est correcte ou bien erronée. D'une manière primordiale, il n'ya pas de calcul ou de stockage centralisé pour l'estimation de la réputation. Chaque nœud maintient l'estimation de réputation des autres nœuds avec lesquels il interagit. Cette estimation est alors utilisée pour supprimer les contributions des nœuds malicieux dans l'agrégation des données finales. Pour accélérer la convergence, i.e la construction de la réputation à travers le temps, les nœuds capteurs partagent leurs observations sur d'autres nœuds avec le reste du

réseau. Cependant, l'échange périodique des valeurs de réputation entre les nœuds induit un overhead élevé en transmission. Le passage à l'échelle de ces approches est étroitement lié au nombre de nœuds impliqués dans le WSN. Ce de fait, l'utilisation des systèmes de confiance et de réputation pour sécuriser les réseaux de capteurs exige de prêter une attention particulière à la bande passante et au surcoût encouru, qui jusqu'ici, ont été négligé par la plupart des travaux de recherche.

D'un autre côté, la construction de systèmes de confiance et de réputation robustes est affrontée à plusieurs défis importants en elle-même (Ganeriwal, et al., 2004). Le plus crucial est la participation d'un nœud malicieux dans le système de réputation qui peut l'empêcher de fonctionner correctement à cause de son mensonge. En effet, un nœud compromis peut accuser un nœud digne de confiance d'avoir entrepris des actions malveillantes ou bien plaider en faveur d'un nœud malveillant. Pour maintenir son intégrité, un système de réputation doit être capable de faire face à ce genre d'attaque. Sun et Liu (Sun, et al., 2006) ont classifié les attaques sur la confiance en trois catégories: les attaques bad mouthing, les attaques on-off et les attaques conflicting behaviour.

L'attaque Bad mouthing : est une attaque lorsqu'un nœud malicieux fournit un vote malhonnête pour augmenter la réputation d'un autre nœud malveillant. La diffamation (bad mouthing) implique également qu'un nœud ignore de sauvegarder les valeurs de confiance positives pour un nœud légitime ou encore sauvegarde des valeurs négatives, menant à une mauvaise réputation.

L'attaque On-off : dans ce type d'attaque, les nœuds malicieux se comportent tantôt bien et tantôt mal, en compromettant le réseau dans l'espoir que leur mauvais comportement ne soit pas détecté. Cette attaque peut rendre un nœud légitime comme étant un nœud malicieux, et rendre un nœud malicieux comme étant digne de confiance.

L'attaque Conflicting behaviour *:* les nœuds malicieux peuvent endommager la réputation des nœuds dignes de confiance en ayant un comportement conflictuel ; bien se comporter pour certains nœuds et se mal comporter pour d'autres nœuds. A cause de comportement contradictoire, les nœuds d'un groupe peuvent avoir des valeurs de réputation très basses par rapport aux nœuds d'un autre groupe.

Un autre défi important à relever lors de la construction d'un système de réputation est de déterminer quand est ce qu'un nœud a effectué une action malicieuse et d'être capable de distinguer cette action d'une panne du capteur. A cause de la nature incertaine de l'environnement des WSN telle que les collisions sur le canal sans fil, il n'est pas toujours possible de distinguer ces deux types de comportements incorrectes.

Tous les protocoles de sécurité mentionnés utilisant les mécanismes de confiance et de réputation sont basés sur des hypothèses particulières concernant la nature de l'attaque. Si le modèle de l'attaquant considéré est réaliste, (i.e un attaquant faible), le protocole achève ses buts de sécurité avec succès. Cela veut dire que l'attaquant est empêché de mener à bien ses objectifs pour perturber le bon fonctionnement du réseau de capteurs. Par contre si le modèle considéré est un attaquant puissant, il ya une probabilité non négligeable que l'adversaire puisse arriver à s'introduire dans le réseau. Car à cause de ressources limitées des nœuds capteurs, ces derniers ne peuvent faire face aux attaquants puissants. Pour cette raison, il est nécessaire d'entreprendre une autre voie de défense : les systèmes de détection d'intrusion (IDS) qui peuvent éventuellement détecter les tentatives d'exploiter les insécurités possibles, et de prévenir les attaques malveillantes, même si ces attaques n'ont pas été éprouvées avant.

5. COMPARAISON DE PERFORMANCE

Après avoir survolé les travaux existants et proposé une taxonomie des protocoles, nous allons dans cette section faire une comparaison entre tous les protocoles cités dans la section précédente. La comparaison des protocoles de sécurité est assez difficile car les concepteurs ont résolu l'agrégation sécurisée sous différents angles. Pour cette raison, nous allons les comparer selon trois paramètres : *les services de sécurité assurés* (confidentialité, intégrité, authentification, fraîcheur et disponibilité), *buts de conception* (passage à l'échelle, overhead, *flexibilité, exactitude* et *généralité*) et enfin la *vulnérabilité* de ces protocoles aux attaques citée dans la section 2. Nous résumons cette comparaison dans les tableaux 2.1 et 2.2

Du moment que le modèle de l'attaquant varie d'un protocole à un autre, chaque solution proposée a ses propres besoins en sécurité. Dans les protocoles basés sur la cryptographie, la confidentialité des données semble être le minimum des besoins en sécurité. Comme nous le constatons dans le tableau 2.1, les techniques basées sur les données en clair assurent une opération d'agrégation plus efficace en termes d'overhead et considèrent de manière considérable l'intégrité des données. Cependant, ces techniques représentent un modèle faible pour les perspectives de confidentialité des données par rapport aux techniques basées sur les données chiffrées. Cependant, dans les protocoles basés sur la confiance, la disponibilité et la durée de vie du réseau semblent être leurs soucis principaux, et doivent être assurés aussi bien que possible.

Il faut noter que tous les protocoles proposés basés sur la cryptographie sont vulnérables aux attaques déni de service et aux attaques physiques, excepté les protocoles

SMA, DMA et A-DMA proposés par Claveirole et al. (Claveirole, et al., 2008). Ce sont les seuls protocoles qui prennent en charge l'attaque de déni de service. D'un autre côté, il faut noter que tous les protocoles basés sur la confiance, sont vulnérables aux attaques de déni de service et aux attaques Sybille.

	Security Services					**Attacks existence**						**Design Goal**				
Concealed Data	Conf	Integ	Auth	Fresh	Avai	NC.	DoS.	SFo	Rep	Syb	Ste	Scal	Over	Fle	Eff	Gen
CDAM	⊗					⊗	⊗	⊗			⊗		High	⊗		⊗
HCS	⊗					⊗	⊗						High			⊗
PHM1	⊗				⊗	⊗	⊗	⊗				⊗	High	⊗		⊗
PHM2	⊗					⊗	⊗	⊗				⊗	High	⊗		⊗
SDAV	⊗	⊗	⊗			⊗	⊗	⊗	⊗		⊗		High		⊗	
CPDA	⊗					⊗	⊗						High			
SMART	⊗					⊗	⊗						Med			
SMA	⊗					⊗			⊗				Med			⊗
DMA	⊗					⊗			⊗				High			⊗
A-DMA	⊗		⊗			⊗							High			⊗
ECCM	⊗		⊗	⊗		⊗	⊗	⊗					Med			
CDAP	⊗					⊗	⊗	⊗	⊗			⊗	Med			
Revealed Data	Conf	Integ	Auth	Fresh	Avai	NC.	DoS	SFo	Rep	Syb	Ste	Scal	Over	Fle	Eff	Gen
SDA		⊗	⊗	⊗		⊗	⊗	⊗			⊗		High			⊗
SIA	⊗	⊗	⊗	⊗		⊗	⊗	⊗					High		⊗	
SDAP	⊗	⊗	⊗			⊗	⊗	⊗			⊗		High		⊗	
SHDA	⊗	⊗	⊗			⊗	⊗	⊗					High		⊗	⊗
ESPDA	⊗					⊗	⊗			⊗		⊗	Low			⊗
SRDA	⊗			⊗			⊗					⊗	Low			⊗
WDA		⊗				⊗	⊗	⊗	⊗	⊗	⊗		Med		⊗	⊗
FAIR		⊗	⊗			⊗	⊗	⊗					Med		⊗	⊗
SAT		⊗	⊗			⊗	⊗	⊗	⊗	⊗	⊗		Low		⊗	⊗

Conf: Confidentiality - **Integ**: Integrity – **Auth**: Authentication – **Fresh**: Freshness – **Avai**: Availability

NC: Node Compromise – **DoS**: Deni of Service – **Sfo**: Selective Forwarding – **Rep**: Replay – **Syb**: Sybil – **Ste**: Stealthy.

Scal: Scalability – **Over**: Overhead – **Fle**: Flexibility – **Eff**: Effectiveness – **Gen**: Genarality

Tableau 2. 1 : A summary of comparison between cryptography-based secure aggregation schemes (Labraoui, et al., 2011(a)).

Les métriques utilisées pour évaluer les buts de conceptions sont : le passage à l'échelle, l'overhead, la flexibilité et l'exactitude.

5.1 Le passage à l'échelle

Comme nous l'avons déjà cité dans la section 2.3, un protocole est scalable s'il peut assurer un haut niveau de sécurité aussi bien pour les réseaux de moyenne taille que pour les réseaux de grandes tailles. Alzaiz et al. (Alzaid, et al., 2008) a classifié l'agrégation des données sécurisée en deux modèles : avec agrégateur unique et avec des agrégateurs multiples. Dans le modèle avec agrégateur unique, toutes les données individuelles dans le réseau traversent un seul point qui est représenté par un agrégateur unique, pour arriver à la SB. Ce modèle peut s'avérer utile dans des réseaux de petites tailles, cependant dans les réseaux à grande taille, il n'est pas approprié d'implémenter ce modèle spécialement lorsque la redondance des données à un niveau inférieur est assez importante. C'est le cas des protocoles SIA et SHDA. Dans le modèle avec agrégateurs multiples, les données collectées dans le réseau sont agrégées plus d'une fois avant d'arriver à la SB. Ceci nous laisse penser que ce modèle est approprié pour les réseaux de grande taille car il achève une plus grande réduction du nombre de bits transmis. Cependant, cette affirmation n'est crédible que si le protocole ne génère pas d'overhead élevé en communication et en bande passante. Nous citons l'exemple de PHM1 et PHM2, qui malgré leur overhead élevé (causé par le chiffrement multiple qui consomme beaucoup d'énergie), arrivent à passer à grande échelle de manière efficace car ils améliorent fortement le gain en bande passante. Alors que le protocole HCS ne permet pas le passage à l'échelle plus particulièrement si le réseau est non fiable, car il génère un trafic important en bande passante lors de la transmission des identificateurs des nœuds impliqués (ou non) dans la fonction de l'agrégation. Dans le protocole ESPDA, uniquement un seul nœud capteur par modèle est autorisé à émettre ses données au cluster-head et dans le protocole SRDA chaque capteur transmet la donnée différentielle au cluster-head à la place de la donnée originale, réduisant ainsi l'overhead de la transmission. Ces deux protocoles sont indépendants de la taille du réseau et peuvent assurer efficacement le passage à l'échelle. Dans le protocole CDAP, des super-nœuds capteurs nommés AGGNODEs sont utilisés pour aggréger les données et permettent ainsi le passage à grande échelle sans aucun problème. Le protocole SRDA, basé sur la confiance, est lui aussi scalable car dans le cluster un seul capteur est élu pour transmettre uniquement la donnée représentative du cluster.

5.2 Overhead

Un protocole induit un overhead (ou surcoût) s'il consomme de la bande passante et de l'énergie. Cet overhead est calculé par la taille des messages et la consommation d'énergie

lors de grand calculs. Dans les techniques basées sur les données chiffrées, le chiffrement des données est *très coûteux*, et génère une consommation en énergie importante spécialement si les données sont chiffrées plusieurs fois (chiffrement multiple). C'est le cas des protocoles CDAM, HCS, PHM1, PHM2, SDAV et CPDA. Y compris l'énergie consommée dans le calcul CPU, chaque primitive cryptographique requiert un temps différent et un nombre de cycles CPU différents pour son exécution, résultant à des valeurs de consommation d'énergie différentes. Par exemple, l'algorithme de chiffrement Skipjack requiert 22,044 cycles CPU et consomme 71,76 μjoules pour calculer un MAC d'un paquet de 29 octets (Wander, et al., 2005). Dans le cas du protocole WDA et FAIR, pour prouver la validité du résultat de l'agrégat, l'agrégateur doit fournir des preuves de plusieurs nœuds témoins. Il doit donc transmettre les preuves à la SB en les embarquant dans le paquet contenant le résultat de l'agrégat. Ceci a pour effet de générer une consommation additionnelle en bande passante. L'overhead croît également à cause de la phase de vérification interactive imposée par la vérification de l'intégrité et l'exactitude. D'ailleurs, le processus de vérification est mis en place par différentes manières, telles que l'utilisation des protocoles interactifs (le cas dans SIA, SHDA, SDAP), l'authentification diffusée par la SB (le cas de SDA, SIA, SHDA, SDAP), ou le système de vote (le cas de WDA, FAIR).

Les protocoles RTM et SELDA basés sur la confiance, rendent le réseau plus fiable mais introduisent un overhead élevé à cause du fait que RTM utilise les arbres de délivrances randomisés et que SELDA utilise la transmission de données à travers des chemins multiples. SDAV améliore la vulnérabilité de l'intégrité des données du protocole WDA en utilisant les signatures digitales. Il induit alors un surcoût en communication pour la validation des données.

5.3 Efficacité

Un protocole qui assure l'exactitude du résultat final de l'agrégat, peut contenir une phase de vérification pour aider la station de base à distinguer entre les mesures agrégées valides et non valides, comme c'est le cas des protocole SIA, SHDA, SDAP, WDA et FAIR basés sur la cryptographie. Un protocole peut également assurer l'efficacité de l'agrégation en surveillant l'agrégateur lorsqu'il envoie le résultat de l'agrégat à la SB. Si les nœuds dédiés pour la surveillance détectent que l'agrégateur a envoyé des données altérées, alors ils déclenchent des alarmes à la SB pour annoncer la triche. C'est le cas du protocole SAT basé sur la cryptographie et les protocoles basés sur la confiance TKL, LTE, RTM, RSDA et SELDA. Cependant le protocole IAF basé sur les techniques de l'intelligence artificielle ne

peut assurer complètement l'exactitude de l'agrégation car ce protocole ne peut détecter les erreurs non biaisées générées par le nœud agrégateur.

	Security Services					Attacks existence						Design Goal				
Trust and Reputation	Conf	Integ	Auth	Fresh	Avai	NC.	DoS	SFo	Rep	Syb	Ste	Scal	Over	Flex	Eff	Gen
TKL				⊗	⊗		⊗	⊗		⊗			low		⊗	⊗
LTE				⊗	⊗		⊗	⊗		⊗			low		⊗	⊗
RTM				⊗	⊗		⊗	⊗		⊗			High		⊗	⊗
RSDA				⊗	⊗		⊗			⊗		⊗	low		⊗	⊗
SELDA		⊗	⊗	⊗	⊗		⊗			⊗			Med		⊗	⊗
Artificial Intelligent	Conf	Integ	Auth	Fresh	Avai	NC.	DoS	SFo	Rep	Syb	Ste	Scal	Over	Flex	Eff	Gen
AIF			⊗	⊗	⊗		⊗	⊗		⊗			low			⊗

Tableau 2. 2 : A summary of comparison between trust-based secure aggregation schemes (Labraoui, et al., 2011(a)).

5.4 Flexibilité

Tous les protocoles proposés dans ce manuscrit sont désignés pour des WSN statiques. Ces protocoles sont mis en service selon un déploiement de nœuds prédéfini avant la mise en service. Désormais, ils n'ont pas pris en compte la mobilité des nœuds. Cependant, la topologie des WSN peut souvent varier, des nœuds peuvent disparaitre à cause d'une panne ou un événement extérieur (exemple vol ou destruction un animal). Un protocole flexible doit prendre en considération cette caractéristique et considérer l'ajout de nouveaux nœuds ou leur disparition après le déploiement du réseau, et ce en assurant le même niveau de sécurité. Il est vrai que cette propriété est très difficile à assurer compte tenu de la complexité de la mise en œuvre de mesures de sécurité dans un milieu variable. Jusqu'à présent, les seuls protocoles flexibles qui ont pris en considération le changement de topologie sont : CDAM, PHM1 et PHM2.

5.5 Généralité

Un protocole de sécurité pour l'agrégation des données est dit général, s'il peut appliquer plusieurs fonctions d'agrégation telles que la somme, la moyenne, la médiane, le max/min et le comptage. La majorité des protocoles proposés dans la littérature assurent cette propriété excepté les protocoles SDAV, CPDA, SMART, SIA et SDAP qui sont conçu uniquement pour une fonction d'agrégation bien définie.

Finalement, il n'y a aucun protocole qui est parfait. Chaque solution proposée a ses avantages et ses limitations. Un compromis entre niveau de sécurité et performances du réseau doit être soigneusement équilibré.

6. CONCLUSION

Les techniques d'agrégation sont très utiles dans les réseaux de capteurs. Elles contribuent de manière significative dans la minimisation des transmissions redondantes et donc dans l'économie de l'énergie et la longévité de la durée de vie du réseau. Cependant, Les réseaux de capteurs sont vulnérables aux attaques externes et internes plus que les autres réseaux sans fil pour les raison discutées dans le chapitre 1. Dans le contexte de l'agrégation, le nœud agrégateur est une cible potentielle pour les attaquants, puisqu'il est considéré comme *la pierre angulaire* du processus de l'agrégation. La compromission de quelques nœuds agrégateurs assure la falsification complète du résultat final de l'agrégat dans tout le réseau. Lors de la conception d'un protocole de sécurité, il est important de comprendre les effets dangereux causés par les différentes attaques afin de faire face et de prévenir tout dommage contre l'application déployée dans le réseau. L'agrégation des données sécurisée est une problématique assez critique, puisque l'agrégation est la base de toute opération dans un réseau de capteurs. Le champ de recherche dans cet axe est très actif et n'a cessé de croître ces dernières années. Les solutions proposées sont classifiées en deux catégories : les solutions basées sur la cryptographie et les solutions basées sur le concept de confiance. Des résultats prometteurs ont été récemment achevés dans l'agrégation des données sécurisée tels que le chiffrement homomorphique et les courbes elliptiques. Malgré l'efficacité des techniques cryptographique pour assurer l'intégrité, la confidentialité et l'authentification, elles sont inefficaces contre les attaques internes. D'autres techniques relatives au concept de confiance et de réputation ont émergé comme solutions supplémentaires à celles basés sur la cryptographie. Cependant ces techniques sont à l'état primaire et il n'existe pas de modèles clairs pour l'évaluation de confiance qui soient appropriés aux réseaux de capteurs sans fil.

Dans le chapitre 4, nous allons proposer un nouveau protocole nommé RAHIM: **R**obust **A**daptive Approach Based on **Hi**erarchical **M**onitoring Providing Trust Aggregation for Wireless Sensor Networks. Ce protocole assure l'intégrité et la disponibilité de l'opération d'agrégation dans un réseau de capteurs sans fil clustérisé.

Chapitre III

La sécurité de la localisation

Sommaire

1. INTRODUCTION

Dans bon nombre d'applications, un événement détecté par un capteur n'est utile que si une information relative à sa localisation géographique est fournie. C'est le cas de la surveillance des feux de forêt ou de troupes ennemies dans un contexte militaire. Sans cette information, ces applications n'auraient aucun sens. La question qui suit immédiatement la détection d'un événement est « où se passe-t-il ? », sans connaître leurs positions, les nœuds seront incapables de répondre à une telle question. La localisation des capteurs est un des principaux problèmes dans ce type de réseaux et nombreuses sont les solutions qui ont été proposées pour le résoudre, chacune faisant des hypothèses diverses sur les capacités des capteurs.

Plusieurs techniques de localisation sont disponibles. La première et la plus développée est l'utilisation du GPS (*Global Positioning System*), i.e, équiper chaque capteur d'un module GPS. Cette technique n'est pas applicable à l'ensemble du réseau de capteurs, car elle est bien trop coûteuse du point de vue financier comme du point de vue énergétique. Pour réduire ce coût, d'autres approches ont été proposées qui consistent à équiper une partie des capteurs d'un module GPS, permettant de se localiser grâce aux coordonnées terrestres (longitude et latitude). Une fois leurs coordonnées absolues récupérées, ces nœuds appelés « ancres » ou « beacons » émettent leur position autour d'eux, qui servira ensuite de repères aux autres nœuds « ordinaires » (ceux n'étant pas équipés de module GPS) pour qu'ils puissent à leur tour se localiser. Une deuxième technique de localisation consiste à déployer un seul nœud mobile au lieu de plusieurs équipés par un GPS. Une fois déployé, le mobile traverse toute la zone en diffusant des informations autour de lui pour aider les nœuds à trouver leurs positions.

La problématique de la localisation dans les WSN a largement été étudiée, et plusieurs approches dites range-based et range-free ont été proposées. Cependant les réseaux de capteurs sans fil de part leur nature, sont exposés à plusieurs types d'attaques, et presque toutes les solutions précédemment proposées peuvent être trivialement manipulées par un adversaire malveillant. Par exemple, la compromission des nœuds ancres, cibles potentiels pour les attaquants, peut mener à un calcul de positionnement tout à fait faux pour les nœuds ordinaires, puisque ces derniers se basent entièrement sur les informations des ancres pour estimer leur position. D'autre attaques peuvent être déclenchées dans le but de falsifier les calculs de localisation et donc d'empêcher la connaissance des positions exactes des nœuds

capteurs. Les dégâts survenus suites à ces attaques sont de degrés différents selon le type d'application, et l'impact de la localisation sur cette application.

Du moment que l'information de positionnement fait partie de la plupart des services des réseaux de capteurs, tels que le routage géographique et la surveillance des infrastructures critiques, il est donc primordiale de concevoir un algorithme de localisation qui soit résistant aux attaques qui peuvent *empoisonner* le calcul. Trois métriques sont associées à la localisation : *efficacité énergétique*, *exactitude* et *sécurité*. Bien que les deux premières métriques soient largement investiguées, la métrique de la sécurité, une métrique clé, n'a attiré l'attention des chercheurs que récemment. Pour cela ce champ de recherche est encore vierge, et plusieurs contributions pourraient émerger afin de trouver des solutions de sécurité pour la localisation dans les réseaux de capteurs sans fil. Cependant, il faut toujours garder à l'esprit que la sécurité est coûteuse en termes de ressources (calcul, mémoire, énergie) pour les capteurs dont les ressources sont généralement limitées (Labraoui, et al., 2011(a)). Un compromis entre niveau de sécurité et performance doit donc être prudemment équilibré (Labraoui, et al., 2011(a)).

Nous présenterons dans ce présent chapitre l'investigation de notre deuxième axe de recherche, à savoir la problématique de la sécurité de la localisation dans les réseaux de capteurs sans fil. Tout d'abord, nous expliquons le principe des systèmes de localisation en général, ensuite nous abordons la localisation dans les WSN et la problématique de sécurité dans ce domaine. Nous présenterons les types d'attaques spécifiques au processus de localisation ainsi qu'un survol des travaux existants pour sécuriser la localisation. Cette étude, nous permettra de tracer les motivations pour la conception d'une stratégie de sécurité pour immuniser un algorithme de localisation de type range-free contre l'attaque wormhole, qu'on verra dans le chapitre 5.

2. PRESENTATION GENERALE DES SYSTEMES DE LOCALISATION

Le terme *localisation* est utilisé pour faire référence à un système permettant de déterminer l'emplacement d'un objet. Pour situer un objet dans l'espace, il faut être capable de le placer dans un plan bidimensionnel (latitude, longitude) ou tridimensionnel (latitude, longitude, altitude).

Historiquement, les systèmes de localisation ont été surtout utilisés et développés pour la navigation. En effet, sur terre, les repères étaient la position par rapport aux cartes, à au moins deux points visibles et connus puis à la boussole.

Aujourd'hui, les systèmes de localisation n'ont plus comme unique intérêt la navigation. On utilise toujours ces systèmes dans les transports maritimes, aériens et terrestres mais également pour de nombreuses autres applications. Le principe de base reste celui de la triangulation ou la trilatération, que nous verrons par la suite. Les systèmes listés ci-dessous sont les plus communs :

- Positionnement par satellites
- Positionnement par réseau de téléphonie mobile
- Positionnement par réseau Wi-Fi
- Positionnement par puces électroniques
- Positionnement par adresse IP

2.1 Les principaux systèmes de localisation par satellite

Si autrefois les systèmes de localisation par satellite étaient exclusivement réservés aux applications militaires, ils se sont ouverts depuis les années 90 au monde civil. Les grandes puissances ont toutes cerné les impacts politiques et économiques de ces systèmes. Si l'on parle aujourd'hui de système de navigation par satellite accessible au grand public, le seul mot qui vient à l'esprit est le GPS, acronyme de « *Global Positioning System* », lancé par le gouvernement américain pendant la guerre froide. Pourtant, des systèmes concurrents sont déjà en cours de construction, dont le système européen Galileo fortement poussé ces dernières années par une partie de l'Europe mais également certains autres pays dont la Chine.

Le GPS (Parkinson et al, 1996) est le système de localisation américain. Opérationnel depuis les années 1980, il a été développé pour fournir à l’armée américaine un système de repérage à couverture mondiale et de très grande précision. Son rôle consiste, par exemple, à guider un missile sur des centaines de kilomètres. C’est à la fin de l’année 1993 que le département américain de la défense a ouvert l’accès gratuit au GPS pour les utilisateurs civils.

Le système russe GLONASS (IAC), développé de 1976 à 1982, n’est plus pleinement opérationnel et ce dû aux conditions politiques et économiques du pays. Toutefois, la Russie a entrepris sa remise à niveau et GLONASS est devenu fonctionnel en novembre 2011.

De son côté, l'Inde met en œuvre son système IRNSS (Indian Regional Navigational Satellite System). Il offrira une précision au sol inférieure à 20 mètres et devrait être prêt en 2012.

Le système Beidou développé par la Chine en est à une version expérimentale (Beidou- 1). Elle est composée de quatre satellites ayant des fonctionnalités limitées. La version Compass (ou Beidou-2) devra compter 35 satellites opérationnels d'ici 2013.

Enfin, Galileo (CNES et ESA) est le système de localisation européen. Il est composé de 27 satellites et atteindra une précision inférieure au mètre pour les applications du domaine civil.

Il faut savoir également que dans certains cas (notamment militaires) la précision de ces systèmes atteint le millimètre.

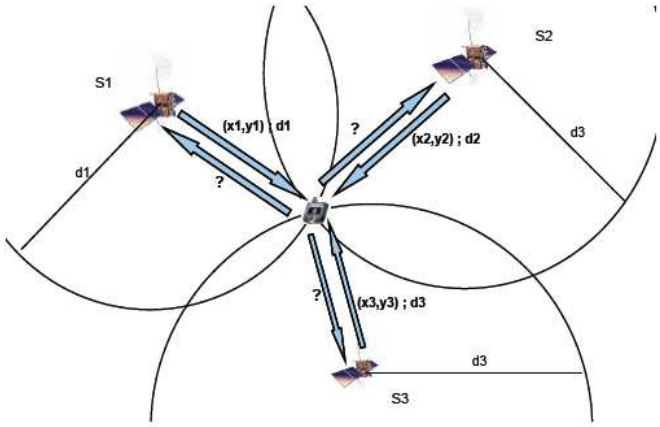


Figure 3. 1 : Principe des méthodes de localisation par satellite (en 2D).

2.2 Principe de la localisation par GPS

Le principe de localisation est en lui même très simple. Il est illustré dans la Figure 3.1. Des satellites dédiés gravitent autour de la terre. Lorsqu'un module de localisation demande sa position, chaque satellite lui communique ses coordonnées et la distance qui le sépare de lui. En effet, si on imagine vouloir localiser un point M, de la surface du globe terrestre, il suffit d'entrer en contact avec ces satellites qui lui communiquent leurs coordonnées. Au moins trois satellites sont nécessaires pour une localisation dans la deuxième dimension, alors qu'au moins quatre le sont pour la troisième dimension.

Le point M applique alors la *multilatération* : en deux dimensions, il s'agira de définir le point d'intersection des cercles dont les centres sont les positions des satellites et les rayons sont les distances entre le point M et les satellites. En trois dimensions, les cercles sont remplacés par des sphères dont le point d'intersection correspond à la position du point M.

3. LA LOCALISATION DANS LES WSN

La localisation dans les réseaux de capteurs déployés de manière aléatoire consiste à déterminer les coordonnées géographiques des différents capteurs. La localisation des nœuds est nécessaire, non seulement pour localiser les différents évènements survenus dans la zone surveillée, mais aussi pour le développement de protocoles de routage de l'information récoltée, pour la couverture de la zone d'intérêt, pour l'agrégation des données, etc. Elle est la première tâche exécutée par les nœuds après leur déploiement.

La localisation dans les réseaux de capteurs pose néanmoins plusieurs problématiques en raison de la quasi absence de tout dispositif d'auto positionnement (comme par exemple les dispositifs GPS). Comme mentionné dans l'introduction, équiper tous les nœuds par des dispositifs GPS est tout simplement techniquement difficile et économiquement non viable. Dans ce cas, les nœuds doivent eux mêmes, au travers de techniques de coopération entre eux, déterminer leur position respective. Sans perte de généralité, les techniques de coopération peuvent être regroupées en deux grandes familles : les approches centralisées et les approches distribuées. Une troisième voie a été récemment étudiée et qui consiste à assister les nœuds du réseau par une ancre mobile leur permettant de se localiser. Cette approche a de nombreux avantages en termes d'économie d'énergie et de précision de localisation.

3.1 Le processus de la localisation dans les WSN

De la problématique de la localisation découle trois problèmes sous-jacents. Les deux premiers sont directement liés au matériel utilisé (définition d'un système de coordonnées et estimation des distances), tandis que le troisième concerne les techniques logicielles utilisées.

– **Définition d'un système de coordonnées** (un repère) : en connaissant les positions de quelques nœuds du réseau (appelés ancres ou « *beacons* ») dans un certain système de coordonnées et les positions relatives des autres nœuds par rapport à ces ancres, il est possible au travers d'un «*mapping* » de retrouver les positions absolues des nœuds dans le même système. Toute la question demeure de bien « sélectionner » les points repères (les ancres).

– **Estimation des distances** : ce procédé est fortement dépendant du matériel de communication utilisé. En d'autre terme, en collectant des indicateurs de la qualité des communications, les différents nœuds peuvent estimer les distances les séparant les uns des autres.

– **Algorithme de localisation** : les algorithmes de localisation sont utilisés à fin de calculer les positions finales en se basant sur les positions des ancres d'une part et d'autre part sur les estimations inter-nœuds.

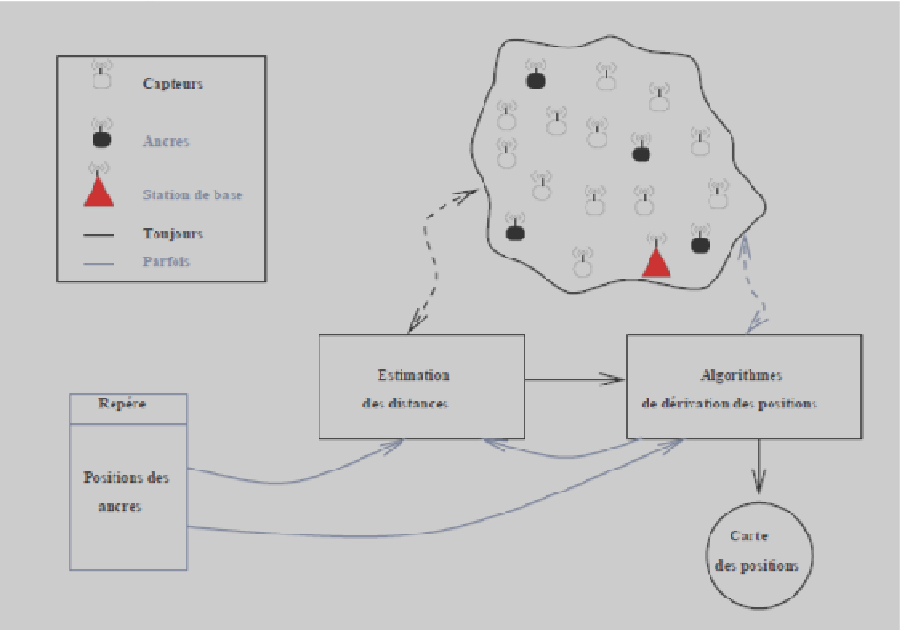


Figure 3. 2 : Une vue du processus de localisation dans un WSN.

Le processus de localisation dans les réseaux de capteurs de la façon la plus générale possible est illustré dans la figure 3.2. Les parties noires sont communes à toutes les méthodes de localisation. Les parties grises en revanche dépendent des différentes méthodes de localisation utilisées.

D'une manière générale, ce processus consiste à définir un repère de coordonnées, une technique d'estimation des distances inter-nœuds et un algorithme pour la dérivation des positions.

3.1.1 Ancres statiques

Les ancres (souvent appelées beacons) sont au préalable nécessaires pour localiser les nœuds d'un réseau de capteurs dans un système de coordonnées global. Les ancres sont simplement des nœuds capteurs ordinaires qui connaissent leurs coordonnées *à priori*. Cette connaissance pourrait être difficilement codée, ou bien facilement acquise par un certain matériel supplémentaire comme un récepteur GPS. Au minimum, trois ancres non-colinéaires sont nécessaires pour définir un système de coordonnées en deux dimensions.

Les coordonnées peuvent être *globales*, c'est à dire qu'elles sont alignées avec un système extérieur comme le système GPS par exemple, ou bien *relatives*, ce qui signifie

qu'elles forment une transformation rigide (rotation, réflexion, translation) des coordonnées du système global. Dans le deuxième cas, on n'a pas besoin de la position des nœuds pour fonctionner, une carte relative est suffisante. Les méthodes qui créent une carte relative des coordonnées sans recours aux ancres sont appelées « *anchor-free* » (Priyantha, et al., 2003). Par contre d'autres méthodes ne fonctionnent pas sans connaître la position d'un certain nombre d'ancres *à priori*, sont appelées « *anchor-based* » (Costa, et al., 2006) (He, et al., 2003).

Les ancres peuvent être utilisées de plusieurs façons. Certains algorithmes de localisation (Costa, et al., 2006) trouvent une carte arbitraire relative pour les coordonnées des nœuds, puis ils utilisent les ancres pour déterminer une transformation rigide des coordonnées relatives vers les coordonnées globales. D'autres algorithmes (He, et al., 2003), partant des positions des ancres, calculent les positions des nœuds ordinaires dans un système global.

Le placement des ancres peut souvent avoir un impact significatif sur la localisation. On a constaté que la précision de la localisation s'améliore si les ancres forment un polygone convexe autour du réseau (Langendoen, et al., 2003). De plus, d'autres ancres supplémentaires placées au centre du réseau peuvent être également utiles.

Inconvénients : L'utilisation des ancres permet de grandement simplifier la tâche d'attribuer les coordonnées aux nœuds ordinaires. Par contre, elle présente des inconvénients inhérents. Les récepteurs GPS sont d'une part chers et d'autre part ne sont pas utilisables pour des applications à l'intérieur. Ils peuvent également être obstrués par de grands immeubles ou d'autres obstacles environnementaux. En outre, Les récepteurs GPS consomment aussi une quantité non négligeable de l'énergie de la batterie, qui peut être un problème pour le pouvoir énergétique limité des nœuds capteurs. Une alternative au GPS est la pré-programmation des nœuds avec leurs positions ou bien l'emplacement manuel. En revanche, ceci peut être non pratique (par exemple lors du déploiement de 10000 nœuds avec 500 ancres), ou même impossible (par exemple lors du déploiement à partir d'un avion).
Pour pallier à certains de ses inconvénients, des travaux dans la littérature ont proposé l'utilisation d'une seule ancre mais cette fois ci il s'agit d'une ancre mobile à la place de plusieurs fixes.

3.1.2 Ancre mobile

Le grand nombre des ancres nécessaires, leur coût, leur faible précision et leur forte consommation d'énergie, motivent l'utilisation d'une seule ancre mobile. Au lieu d'avoir plusieurs ancres statiques, une seule mobile est déployée avec les nœuds, puis elle traverse la

zone de surveillance en communiquant avec les autres nœuds afin de les aider à s'auto-localiser comme montre la figure 3.3. L'ancre mobile diffuse des informations tout au long de sa trajectoire. Elle peut être un opérateur humain, un robot déployé avec le réseau de capteurs, ou dans le cas d'un déploiement d'un avion, l'avion lui-même.

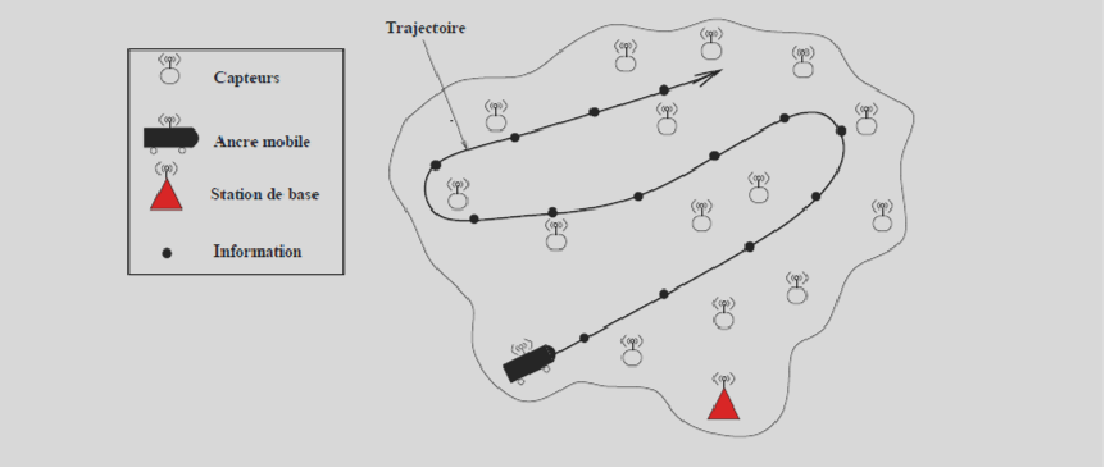


Figure 3. 3 : Une seule ancre mobile pour aider les nœuds à s'auto-localiser.

Inconvénient : Le principal inconvénient de l'utilisation d'une ancre mobile est cependant l'absence d'une trajectoire bien définie, ainsi que la détermination des instants durant lesquels l'ancre va diffuser les informations. Notons que ce problème est très difficile car les positions des autres nœuds ne sont pas connues *à priori.*

3.1.3 Les techniques pour l'estimation des distances

Parmi les méthodes de localisation dans les réseaux de capteurs nous distinguons entre deux catégories, les méthodes qui ne sont pas basées sur la distance inter-nœuds et d'autres qui y sont. Les premières sont celles qui ne calculent pas de distances entre voisins. Elles utilisent d'autres informations telles que la connectivité pour estimer la position des nœuds. Les deuxièmes sont des méthodes qui estiment les distances entre les nœuds pour calculer les positions. Plusieurs techniques sont développées pour les estimations des distances entre les nœuds voisins. Parmi lesquelles nous trouvons celles qui sont basées sur les dispositifs radio, comme la méthode de la force du signal reçu « *RSSI* » et celles qui sont fondées sur l'utilisation d'autres matériels (microphones, etc) comme la technique de la différence entre les temps d'arrivée de deux signaux « *TDoA* » et celle qui estime l'angle d'arrivée du signal« *Angle of Arrival (AoA)* ».

1. Temps d'arrivée

La technologie ToA (Time of Arrival) suppose que les nœuds du réseau sont synchrones. La distance qui sépare deux capteurs se déduit de la vitesse de propagation du signal et de la différence entre les dates d'émission et de réception du message. Cette technologie est celle utilisée par le système GPS (Global Positioning System). Lorsque les nœuds ne sont pas synchrones, l'envoi d'un message aller-retour est nécessaire. En fonction de son horloge, de la vitesse de propagation du signal et du temps de traitement du signal reçu, un capteur récepteur obtient la distance qui le sépare du capteur émetteur en calculant la différence entre les dates d'émission et de réception, en y soustrayant le temps de traitement du signal, puis en divisant le résultat par deux. Cela suppose que les nœuds du réseau ont un temps de traitement du signal identique.

2. Différence des temps d'arrivée

La technologie TDoA (Time Difference of Arrival) requiert d'autres matériels et elle n'est pas basée sur le signal radio seulement (Savvides, et al., 2001). En effet, chaque nœud devrait être équipé d'un haut-parleur et d'un microphone. TDoA se base sur la différence des dates d'arrivée d'un ou plusieurs signaux et suppose également que la vitesse de propagation des signaux est connue. Cette technologie s'applique dans les cas suivants :

– un émetteur envoie des signaux de natures différentes (par exemple, l'ultrason, l'onde radio,...) à un récepteur ;

– un récepteur reçoit des signaux d'une même nature d'au moins trois émetteurs ;

– un émetteur envoie un signal reçu par au moins trois récepteurs (dans ce dernier cas une vue globale des signaux sera connue).

Dans chacun des cas, les récepteurs mettent en corrélation leurs informations et en déduisent les distances qui les séparent des émetteurs. Il s'agit d'une simple résolution d'un système d'équations dont les distances sont les inconnues.

3. Puissance du signal

Dans les réseaux de capteurs sans fil, chaque capteur est équipé d'une radio. La question est de savoir comment utiliser la radio pour aider à localiser le réseau ? La technologie RSSI (Received Signal Strength Indicator) (Bahl, et al., 2000) propose une solution élégante pour l'estimation des distances dans les réseaux de capteurs. Elle considère la perte de puissance d'un signal entre son émission et sa réception. Cette perte varie en fonction de la distance entre les deux capteurs : plus les capteurs sont éloignés (resp. proches), plus la perte est importante (resp. faible). Cette perte sera alors traduite en une distance.

En pratique, les mesures par RSSI contiennent des erreurs de l'ordre de quelques mètres (Bahl, et al., 2000). Ce bruit se produit parce que la propagation des ondes radio tend à être fortement non-uniforme dans des environnements réels. Par exemple, la radio se propage différemment sur l'asphalte que sur l'herbe. Des obstacles physiques tels que les murs, meubles, etc, reflètent et absorbent les ondes radio. Par conséquent, la précision sur la distance en utilisant la force du signal n'est pas bien démontrée par rapport à d'autres techniques comme la « TDoA ».

4. Angle d'arrivée

La technologie AoA (Angle of Arrival) (Niculescu, et al., 2003) consiste à calculer l'angle formé entre deux capteurs. La direction (l'angle) est généralement recueillie par la radio et un ensemble de microphones, qui permettent à un nœud écouteur de déterminer sa direction par rapport à l'émetteur. Il est également possible de la recueillir par le moyen d'une communication optique.

Dans cette technique, on a besoin de plusieurs (3-4) microphones spatialement séparés qui entendent un seul signal transmis. En analysant la phase ou la différence entre les temps d'arrivée du signal aux différents microphones, il est possible de découvrir l'angle d'arrivée du signal. Ces méthodes peuvent obtenir une précision de l'ordre de quelques degrés (Priyantha, et al., 2001). Malheureusement, elles exigent plus de matériels (un haut-parleur et plusieurs microphones) que la technique « TDoA », donc elle est plus coûteuse et les nœuds tendent à être plus volumineux. Il existe peu d'algorithmes de localisation qui sont basés sur la technique « AoA», même si plusieurs d'entre eux sont capables de l'utiliser quand elle est présente.

3.1.4 Dérivation des positions

La dérivation des positions consiste à calculer les positions finales de chaque nœud capteur en utilisant un des algorithmes de localisation. Chaque algorithme utilise une méthode de calcul qui dépend de la technique d'estimation de distance utilisée. Nous classifions ces méthodes en trois catégories :

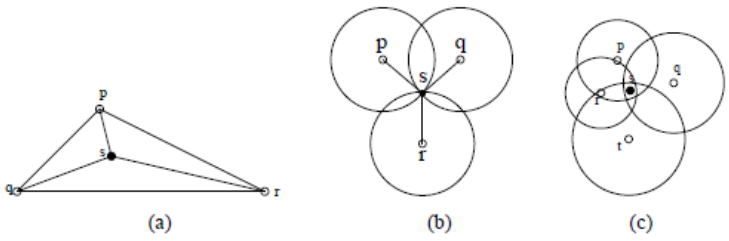


Figure 3. 4 :(a) triangulation, (b) trilatération, (c) multilatération.

1. **la trilatération** (Langendoen, et al., 2003) est la méthode la plus simple pour déterminer les positions des capteurs en utilisant la géométrie des triangles. Elle est fondée sur le même principe qu'un système GPS. Elle consiste à s'appuyer sur trois points de référence, c'est à dire des nœuds dont on connaît la position (les ancres), et sur les distances qui les séparent du nœud dont on cherche à estimer la position. Cette dernière correspond alors au point d'intersection des trois cercles. Voir la Figure 3.4.
2. **La triangulation** est une technique permettant de déterminer la position d'un point en mesurant les angles entre ce point et d'autres points de référence dont la position est connue (les ancres), et ceci plutôt que de mesurer directement la distance entre les points. Ce point peut être considéré comme étant le troisième sommet d'un triangle dont on connaît deux angles et la longueur d'un côté. Voir la Figure 3.4.
3. **La multilatération** (Langendoen, et al., 2003) a le même principe que la trilatération, en utilisant plus que trois points de référence (ancres). La position d'un nœud est calculée en résolvant l'intersection de plusieurs hyperboles basées sur la différence des temps d'arrivée TDoA.

   Soit une cible *i*, connaissant les positions (x_a, y_a) de *m* ancres ($1 \leq a \leq m$) ainsi que les distances d_{ia}, où d_{ia} représente la distance euclidienne entre *i* et l'ancre *a*.

Ayant ces informations et pour calculer la position (x_i, y_i)[1] de la cible *i* nous formons le système suivant :

$$\begin{aligned}
(x_1 + x_i)^2 + (y_1 - y_i)^2 &= d_{i1}^2 \\
(x_2 + x_i)^2 + (y_2 - y_i)^2 &= d_{i2}^2 \\
(x_3 + x_i)^2 + (y_3 - y_i)^2 &= d_{i3}^2 \\
&\vdots \\
(x_m + x_i)^2 + (y_m - y_i)^2 &= d_{im}^2
\end{aligned}$$

Ce système peut être linéarisé en soustrayant la dernière équation des $m-1$ équations précédentes. En réordonnant les termes, nous obtenons un système d'équations linéaires. Ayant des erreurs dans les estimations de distances, il paraît qu'une solution exacte pour un tel système d'équations est presque impossible. La solution la plus proche de la solution exacte c'est au sens des moindres carrés. Cette solution est plus détaillée dans (Langendoen, et al., 2003).

[1] Pour des raisons de simplicité nous avons considéré uniquement deux coordonnées.

3.2 Classification des approches de localisation

De nombreuses techniques sont proposées pour permettre aux nœuds d'estimer leur position. Nous pouvons distinguer deux types de stratégie de localisation : les stratégies directes (localisation absolue) et les stratégies indirectes (localisation relative). Les localisations absolues détermineront de manière précise les coordonnées du nœud dans le réseau tandis que les localisations relatives, ou grossières, spécifieront une surface ou des coordonnées virtuelles, etc...

3.2.1 Les approches directes

Connues également sous le nom de localisation absolue. L'approche directe elle-même peut être classifiée en deux types : configuration manuelle et localisation basée sur le GPS. La méthode de configuration manuelle est très encombrante et couteuse. Elle n'est ni pratique ni adaptée pour les réseaux de capteurs à grande échelle et en particulier, ne s'adapte pas bien pour les réseaux de capteurs mobiles. D'un autre côté, la méthode de localisation basée sur le GPS, permet de résoudre en théorie le problème de localisation de chaque nœud du réseau, et s'adapte bien pour les nœuds mobiles. Toutefois, équiper chacun des capteurs d'un récepteur GPS constitue souvent une solution irréalisable en pratique, à cause du coût prohibitif d'un tel équipement pour un réseau constitué de millier de capteurs, de la réserve énergétique limitée des capteurs et du mauvais fonctionnement de cette technologie en intérieur.

3.2.2 Les approches indirectes

L'approche indirecte est également connue sous le nom de localisation relative, dans laquelle les positions des nœuds sont dérivées par rapport aux positions d'autres nœuds dans leur proximité. Les approches indirectes de localisation ont été introduites pour surpasser les inconvénients des techniques de localisation basées sur le GPS. Dans ces techniques, quelques nœuds capteurs nommés Beacon ou ancres sont équipés de récepteurs GPS et servent de repères pour les autres nœuds ordinaires qui vont calculer leur position selon des méthodes appropriées. Ces approches s'avèrent moins couteuses que les approches directes et Dans les approches indirectes, le processus de localisation est classé en deux catégories: les méthodes *range-based* et les méthodes *range-free*.

1. **Les méthodes range-based**

Les méthodes Range-based utilisent les technologies ToA, RSSI, AoA et autres afin de mesurer les distances ou les angles entre deux capteurs voisins. Grâce à cette capacité de mesure, un capteur pourra, sous certaines conditions, obtenir sa position exacte. Autrement, une position estimée lui sera attribuée. Parmi les algorithmes range-based, nous citons : **Dynamic fine-grained** (Savvides, et al., 2001), **APS** using AoA (Niculescu, et al., 2003) et **MDS** (Ji, et al., 2004).

2. **Les méthodes range-free**

Ces méthodes ne calculent jamais de distances entre voisins. Elles utilisent d'autres informations telles que la connectivité pour identifier la position des nœuds. Dans un objectif de simplicité et de réduction du coût, ces méthodes supposent que le déploiement des nœuds respecte certaines contraintes et proposent des calculs plus ou moins complexes pour évaluer la position. Elles semblent donner de bons résultats dans les réseaux denses et réguliers. Dans notre travail, nous nous intéresserons aux méthodes range-free. Parmi les algorithmes range-free, nous citons : **APS** (Niculescu, et al., 2001) avec ses trois méthodes (Dv-Hop, Dv-Distance et Euclidian distance), **GPS-less** (Bulusu, et al., 2000), **Convex position** (Doherty, et al., 2001) et **APIT** (He, et al., 2003).

3.3 Implémentation du processus de localisation dans les WSN

Nous distinguons plusieurs façons d'implémenter le processus de localisation :

3.3.1. Les méthodes centralisées

Tous les nœuds communiquent avec leurs voisins et renvoient à l'ordinateur central soit des informations sur le signal, soit directement les distances. L'ordinateur central s'occupe si nécessaire d'estimer les distances à partir des informations sur le signal et ensuite de localiser les nœuds (Bulusu, et al., 2000).

3.3.2. Les méthodes distribuées

Ici tous les nœuds communiquent avec leurs voisins pour estimer les distances et échangent leurs informations de voisinage. Ils dérivent ensuite de façon distribuée la position de tous les nœuds dans le réseau. C'est-à-dire qu'à la fin du processus de localisation, chaque

nœud doit connaître sa position ainsi que celles de ses voisins et ce sans l'aide d'un ordinateur central qui effectuerait les calculs. Pour les grands réseaux, on considère qu'une méthode distribuée est nécessaire car les méthodes centralisées demanderaient trop de communication pour l'acheminement des informations vers l'unité centrale et consommeraient donc trop d'énergie (Costa, et al., 2006).

3.4 Critères de localisation

Un algorithme de localisation est évalué selon une liste de critères dont nous citons :

Précision de la localisation : L'erreur de la localisation est souvent définit comme étant, la distance euclidienne entre les vraies positions des nœuds et celles estimées par l'algorithme. L'objectif d'un algorithme de localisation est de minimiser cette erreur pour augmenter la précision de localisation. Généralement, cette imprécision vient de l'imprécision des méthodes d'estimation de la distance. Les obstacles environnementaux et les terrains irréguliers peuvent influencer la précision des algorithmes de localisation. Des obstacles comme de gros rochers peuvent interférer avec les ondes radios, et empêcher l'utilisation de «TDoA » du fait qu'on n'a plus une ligne droite.

Contraintes de ressources : Les nœuds capteurs possèdent généralement des ressources très limitées. Ils possèdent de faibles processeurs et de petites mémoires, ce qui rend les grands calculs irréalisables. Par conséquent, un algorithme de localisation doit être simple et non complexe et son développement n'exige pas de grands calculs ni de grande capacité de stockage de mémoire. De plus, nous ajoutons la rapidité de l'algorithme. Avec quelle rapidité le système de localisation renvoie-t-il les positions des nœuds ? Ceci est particulièrement important, surtout lors du traçage d'un chemin d'une cible.

Contraintes énergétiques : La seule source d'énergie d'un nœud capteur est sa batterie. Pour cela, dans les réseaux de capteurs, une gestion de l'énergie très économique est nécessaire. Comme le facteur dominant de la consommation d'énergie est la communication radio, il faut trouver un algorithme de localisation qui communique le moins possible via la radio.

Passage à l'échelle : Les réseaux de capteurs sont généralement envisagés à large échelle, avec des centaines voir des milliers de nœuds. La question qui se pose, est-ce qu'un algorithme de localisation fonctionne sur un réseau de plusieurs milliers de nœuds ? Et si oui, est-il toujours aussi efficace ? Ce critère est en rapport avec le fait qu'un algorithme soit implémenté de façon distribuée ou non.

4. PROBLEMATIQUE DE LA SECURITE DE LA LOCALISATION

La localisation est un processus très important dans un réseau de capteurs et peut être nécessaire dans plusieurs applications dites *location-aware,* et contribue également au développement de protocoles de routage de l'information récoltée, pour la couverture de la zone d'intérêt, pour l'agrégation des données, etc. L'utilisation d'un système de localisation basé sur le GPS est une solution non praticable à cause de son coût prohibitif, les nœuds capteurs doivent donc calculer leur position en utilisant des algorithmes de localisation.
La recherche récente qui aborde le problème de localisation dans les réseaux de capteurs a, toutefois, conduit à des algorithmes non complexes, à moindre coût et qui assurent une bonne précision de positionnement. Néanmoins, les nombreux travaux de recherche relatifs à cette problématique, ont investigué le problème d'estimation de positions dans un environnement sans adversaires. Cependant dans la majorité des cas, un réseau de capteurs est déployé dans des zones hostiles et sans surveillance. Les nœuds capteurs peuvent donc être exposés aux attaques conventionnelles mais aussi à de nouvelles attaques (Hu, et al., 2003) menant à l'interruption des fonctionnalités des applications dite « location-aware » et des fonctions du réseau en exploitant les vulnérabilités du processus d'estimation de positions. Il est donc nécessaire que la localisation des nœuds capteurs puisse être estimée de manière robuste en présence d'adversaires.

La recherche dans la sécurité de la localisation est encore à un stade embryonnaire, et reste une problématique ouverte. Nous étudions dans cette section la sécurisation de la localisation dans les réseaux de capteurs, en précisant les attaques spécifiques au processus de localisation ainsi que les mécanismes de sécurité utilisés. Nous survolerons également les solutions déjà proposées pour sécuriser les protocoles de localisation dans les réseaux de capteurs sans fil.

4.1 Les attaques contre le processus de localisation

Un processus de localisation se divise en trois composants fortement connectés comme illustré dans la figure 3.5: l'estimation de distances, le calcul des positions et l'algorithme de localisation. Ce processus peut être attaqué par différentes manières. Le moindre comportement anormal sur l'un de ces composants, peut sérieusement affecter le système de localisation. Par exemple, une estimation de distance erronée malicieuse peut

causer un faux calcul de position qui sera propagé à l'algorithme de localisation et causera probablement une erreur de localisation majeure pour les nœuds capteurs. A cause de la relation étroite entre les composants, chacun d'eux peut être une cible potentielle pour un attaquant, ce qui rend ces systèmes de localisation très fragiles et difficiles à sécuriser. Dans ce qui suit, nous discutons la vulnérabilité de chaque composant de localisation.

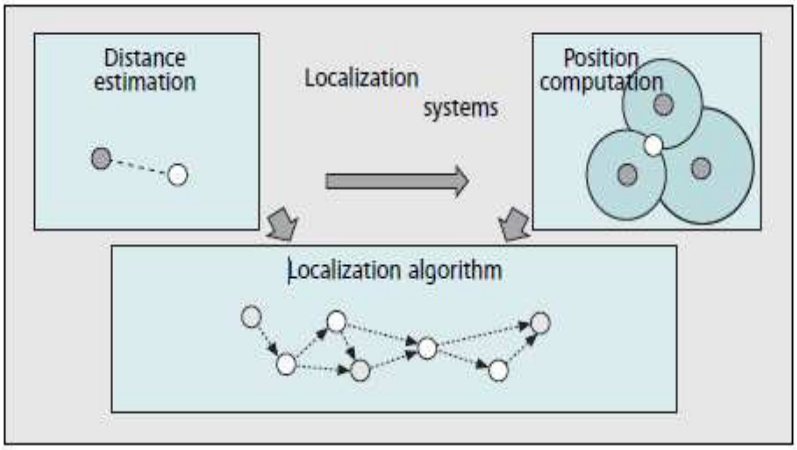


Figure 3. 5 : La division des systèmes de localisation en 3 composants distincts (Boukerche, et al., 2008).

4.1.1 Les attaques contre l'estimation de distance

L'estimation de distance peut être basée sur la force du signal, le temps d'arrivée ou l'analyse du nombre de saut (hop count). Dans le premier cas, un nœud compromis peut envoyer un paquet avec une puissance de transmission plus grande ou plus petite pour faire croire aux nœuds voisins qu'il est plus proche ou plus loin qu'il est réellement.

Dans le second cas, le temps de transmission d'un paquet peut être retardé, causant des problèmes aux systèmes basés sur ToA et TDoA ;

L'estimation de distance basée sur le nombre de sauts peut être faussée par des nœuds compromis qui annoncent des nombres de sauts erronés. En fait, puisque c'est un algorithme multi-sauts, l'estimation du nombre de saut peut aussi être affectée par les attaques sur l'algorithme de localisation.

Dans un environnement compromis, la puissance du signal et les techniques ToA peuvent être ciblées en changeant le médium physique, par exemple, en introduisant du bruit, des obstacles ou une fumée. Egalement les systèmes basés sur AoA peuvent être compromis en déployant des aimants dans le champ du capteur (Boukerche, et al., 2008).

4.1.2 Les attaques contre le calcul des positions

Pour calculer sa position, un nœud capteurs a besoin d'au moins trois estimations de distance et de trois positions connues (celles des nœuds ancres). Comme ça été mentionné dans la section précédente, la moindre attaque sur l'estimation de distance affectera surement le calcul de position. Cependant, certaines attaques peuvent affecter le calcul de position directement en annonçant des positions incorrectes des nœuds ancres. Ceci a pour effet de fausser le calcul des positions même si l'estimation des distances est correcte. Dans ce cas, un nœud compromis peut envoyer non seulement ses propres messages contenant des positions erronées, mais aussi envoyer des messages additionnels simulant l'existence de plusieurs nœuds différents dans différentes localisations.

Dans un environnement compromis, un signal GPS peut être brouillé (jammed) pour le rendre erroné ou empêcher les nœuds ancres d'estimer leurs positions. (Boukerche, et al., 2008).

4.1.3 Les attaques contre les algorithmes de localisation

Les attaques sur les composants d'estimation des distances et de calcul des positions, sont des attaques spécifiques aux systèmes de localisation. Cependant, le troisième composant d'un système de localisation, à savoir l'algorithme de localisation partage le même genre de vulnérabilités liés aux systèmes distribués, car l'algorithme est distribué et généralement multi-sauts. Par exemple, certaines attaques incluent les attaques Sybille, les attaques de rejeu et l'attaque du trou de ver (wormhole).

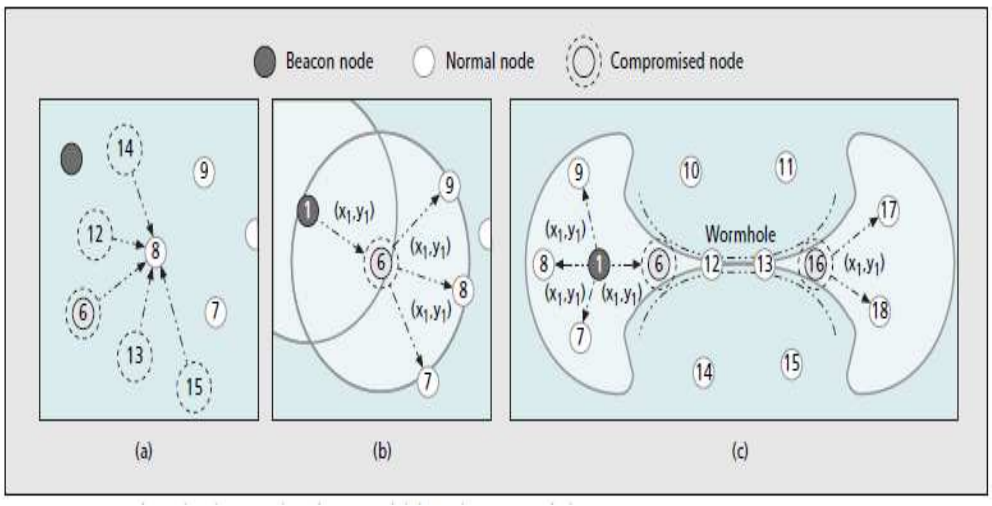


Figure 3. 6 : Les attaques sur les algorithmes de localisation : (a) sybille ; (b) rejeu ; (c) wormhole (Boukerche, et al., 2008).

L'attaque Sybille: dans ce type d'attaque, un nœud malicieux prend l'identité de différents nœuds et commence à envoyer des informations erronées. Ces informations erronées peuvent être soit des estimations de distance, des positions, des nombres de sauts parvenus par des nœuds ordinaires ou des nœuds ancres non existants. La figure 3.6 (a) illustre cette attaque lorsque le nœud 6 déclare être les nœuds 12 et 15.

L'attaque de rejeu : dans cette attaque, un nœud compromis sauvegarde les messages reçus (par exemple d'un nœud ancre) et retransmets les même messages plus tard. Du moment que c'est une copie d'un message original, les nœuds voisins déduisent de manière incorrecte que le nœud malicieux est un nœud qui a envoyé un message original (figure 3.6 (b)). Dans ce cas, du moment que l'estimation des distances est effectuée sur la base des nœuds compromis, alors que la position dans le message est basée sur un nœud légitime, le calcul de la position est affecté. La puissance du signal ainsi que l'estimation de distance basée sur le temps sont également affectées, car le massage rejoué par le nœud compromis aura une puissance de signal différente et un temps de propagation différent.

L'attaque wormhole : cette attaque est l'une des attaques les plus sévères, dans laquelle, l'information reçue par un nœud malicieux dans un côté du réseau est relayée et répliquée par un autre nœud malicieux de l'autre côté du réseau, en faisant apparaitre que le message provient d'un nœud proche. Cette attaque est illustrée dans la figure 3.6 (c). Ce type d'attaque peut considérablement perturber un système de localisé non sécurisé en introduisant des points de référence totalement différentes et erronées dans le calcul des positions.

4.2 Les techniques de sécurité de la localisation

Des solutions de sécurité pour les systèmes de localisation ont été proposées ces dernières années afin d'assurer un positionnement correcte des nœuds dans les applications critiques et militaires des réseaux de capteurs sans fil. La plupart de ces solutions sont achevées soit en utilisant la cryptographie, soit en détectant et en bloquant les nœuds compromis ou les informations erronées, soit en prenant des décisions statistiques, ou en filtrant les positions utilisées dans les calculs.

4.2.1 Techniques de cryptographie

La plupart des attaques de sécurité sont effectuées par un nœud malicieux qui tente de se faire passer pour une entité qu'il n'est pas ou de modifier les valeurs des données dans les messages. Ces problèmes peuvent être résolus par cryptographie en utilisant l'*authentification*

et *l'intégrité des messages*. Il est aussi possible d'assurer la *confidentialité* des positions pour empêcher l'acquisition des informations du réseau par des nœuds malicieux.

La cryptographie peut être utilisée pour se protéger des attaques externes déclenchées par des nœuds externes au réseau. Mais lors de la présence de nœuds compromis de l'intérieur du réseau local, les attaquants peuvent prendre le contrôle du matériel cryptographique des nœuds légitimes (clés et mots de passe) et s'introduire dans le réseau sans se faire détectés, puisqu'ils peuvent s'authentifier en utilisant les clés légitimes, c'est le cas des attaques internes. Pour cette raison, la plupart des algorithmes de localisation n'utilisent pas de méthode de sécurité basées sur la cryptographie, comme nous allons le voir dans la section suivante. Par contre ils utilisent la cryptographie comme deuxième ligne de défense. C'est le cas des algorithmes HiRLoc (Lazos, et al., 2006), SeRLoc (Lazos, et al., 2004), et ROPE (Lazos, et al., 2005) dans lesquels ils utilisent des primitives cryptographiques efficaces pour sécuriser la transmission des messages envoyés par les nœuds ancres. Dans l'algorithme SPINE (Capkun, et al., 2005) la cryptographie est utilisée pour authentifier l'estimation des distances, et dans (Liu, et al., 2005(a)) pour assister la détection des nœuds ancres malicieux.

Dans la majorité des cas, il est supposé que les nœuds du réseau peuvent établir des pairs de clés secrètes. Cependant à cause de son coût en termes de mémoire et de calcul CPU, la cryptographie est souvent évitée, car les nœuds capteurs ont des ressources limitées. Mais dans la plupart des cas, si nous avons besoin de sécuriser la localisation, nous avons aussi besoin de sécuriser l'accès au médium, le routage ainsi que la synchronisation du temps. Les techniques cryptographiques peuvent aussi constituer une couche de sécurité pour les trois composants d'un système de localisation (estimation des distances, calcul des positions et algorithme de localisation) en assurant un contrôle d'authentification, d'intégrité pour l'échange des messages.

4.2.2 Détection des comportements anormaux

Lorsque les techniques cryptographiques sont compromises par des attaques internes, une des solutions pour se protéger de ces attaques est d'observer les comportements des nœuds dans le temps pour décider s'ils sont dignes de confiance ou s'ils sont compromis. Ces techniques peuvent être employées principalement pour protéger le composant de calcul de position, car les informations collectées par des nœuds compromis sont simplement ignorées lors du calcul de la position des nœuds.

Liu et al. (Liu, et al., 2005(a)) ont proposé un ensemble de techniques pour la détection des nœuds ancres malicieux. Une des techniques compare la distance estimée en

utilisant les informations de position fournies par ces nœuds ancres avec la distance estimée par le signal reçu (exemple, RSSI, TDoA, AoA). Une autre technique évalue le temps d'aller retour RTT (round-trip time) entre deux nœuds voisins, en se basant sur l'observation que le rejeu d'un signal beacon (message envoyé par un nœud ancre) introduit un délai plus grand. La station de base utilise ces informations sur les nœuds ancres malicieux pour raisonner sur la suspicion de chaque nœud ancre et filtrer les nœuds malicieux. Srinivasan et al. (Srinivasan, et al., 2006) ont proposé une extension des techniques de Liu et al. en utilisant une balance continue et un mécanisme basé sur la réputation et la confiance. Le résultat a été le protocole DRBTS (distributed reputation-based beacon trust system), qui est un protocole de sécurité distribué pour exclure les nœuds ancres malicieux. Dans DRBTS chaque nœud ancre surveille son voisinage pour les nœuds ancres suspectés et fournit des informations en maintenant et en s'échangeant les tables de réputation entre nœuds voisins. De cette façon, chaque nœud capteur peut choisir les nœuds ancres dignes de confiances sur la base d'une approche de vote.

4.2.3 Calcul de position robuste

Une autre manière pour traiter les nœuds malicieux est d'accepter leur présence dans le réseau et de proposer un calcul de position qui soit robuste même en présence d'informations erronées. Ceci peut être possible grâce aux techniques statistiques et de filtrage des valeurs aberrantes (outliers filtering). Dans ces cas il est supposé que le nombre des nœuds bénins est plus grand que le nombre des nœuds malicieux. Ces techniques sont utilisées pour se protéger (pour être plus robustes) des attaques sur les deux composants du processus de localisation, estimation des distances et calcul de position.

Li et al (Li, et al., 2005) a utilisé le principe de la technique des moindre carrés de la fusion des données pour proposer un estimateur de position des moindres carrés et des moindre médianes adaptatifs. L'idée est d'utiliser la technique des moindres carrés en l'absence d'attaques, et la technique des moindres médianes en présence d'attaque, car cette dernière tolère un pourcentage de valeurs aberrantes de 50 pour cent et achève une estimation correcte.

Liu et al. (Liu, et al., 2005(b)) ont proposé une méthode qui utilise l'estimation MMSE (minimum mean square estimation) qui est une technique de fusion de données pour obtenir une estimation améliorée, et identifier et supprimer les informations de position malicieuses. Dans cette méthode, les positions de capteurs sont estimées en utilisant la méthode basée sur l'estimation MMSE. Cette méthode vérifie si l'estimation de la position a

été estimée à partir d'un ensemble consistant de références de positions. Dans le cas contraire, les références inconsistantes sont identifiées et supprimées et la position d'un nœud est estimée de nouveau. Ce processus est réitéré jusqu'à suppression de toutes les références inconsistantes. L'erreur quadratique moyenne (mean square error) de la mesure de distance est utilisée comme indicateur d'inconsistance. Une seconde méthode proposée par Liu et al. (Liu, et al., 2005(b)) est la technique d'estimation de position basée sur le vote. Dans cette technique, la zone de déploiement des capteurs est divisée en cellules dans lesquelles chaque nœud ancre vote sur la cellule dans laquelle se trouve le nœud ordinaire. Ensuite la méthode sélectionne les cellules dont le nombre de vote est le plus grand et utilise le centre de ces cellules comme estimation de position. Les résultats de vote peuvent être affinés de manière interactive pour améliorer l'exactitude.

4.2.4 Vérification de position

Certaines solutions se focalisent sur la fiabilité du résultat final du calcul de position au lieu d'éviter ou de détecter les nœuds compromis et les attaques. La détection des anomalies de localisation (LAD) (Du, et al., 2005) utilise la connaissance du déploiement avec un modèle de déploiement basé sur le groupe, pour vérifier si les positions calculées des nœuds sont consistantes avec le modèle connu et les observations. Dans (Sastry, et al., 2003) un algorithme est proposé pour la vérification à l'intérieur d'une région dans laquelle un certain nœud peur vérifier si un autre nœud est réellement à l'intérieur d'une région particulière dans laquelle il prétend être. Le protocole proposé nommé Echo, utilise les propriétés physiques connues des fréquences radio et ultrason pour calculer les distances et vérifier si le nœud est réellement dans la région prétendue. Ces techniques peuvent être utilisées pour fournir une couche de sécurité pour les trois composants du processus de localisation, car elles vérifient seulement le résultat du système global de localisation.

4.2.5 Algorithmes simples et sécurisés

Les systèmes de localisation sont vulnérables la plus part du temps à cause du nombre de composants disponibles pour être attaqués. Une autre manière de sécuriser un système de localisation est d'utiliser des algorithmes de localisation simples et moins dépendant, tels que GPS-free, range-free et/ou des algorithmes à un seul saut. Un exemple est l'algorithme SeRLoc (Lazos, et al., 2004) (secure range-independent localization), dans lequel les nœuds ancres sont équipés d'un ensemble d'antennes directionnelles de haute puissance. Ces nœuds envoient des paquets en utilisant une transmission asymétrique. Ces paquets contiennent leur

position et le secteur de l'antenne dans lequel le paquet a été envoyé. Comme c'est un algorithme à un seul saut et range-free, il est protégé contre les attaques qui visent à altérer les mesures et contre les nœuds compromis. Cependant, il n'est pas immunisé contre l'attaque wormhole, qui est évitée en vérifiant les propriétés du réseau telles que l'unicité de secteur et la portée de communication. Une technique similaire est utilisée dans HiRLoc (Lazos, et al., 2006) (high-resolution robust localization), qui possède une exactitude plus grande mais génère une plus grande complexité de calcul et de communication. De telles techniques peuvent être utilisées pour protéger le troisième composant du processus de localisation, à savoir l'algorithme de localisation.

4.3 Comparaison des solutions existantes

Dans la sécurité des réseaux, il est connu qu'aucun système n'est totalement sûr. Il ya toujours des failles, et la question est simplement si elles sont acceptables. Dans les WSN cette problématique devient un peu plus compliquée à cause des limitations de ressources. Dans ce cas, nous devons décider du niveau de sécurité requis, qui dépend entièrement de l'application, et combien de ressources sont dépensées pour assurer le niveau de sécurité. Selon l'analyse coûts-avantage, nous pouvons décider quelles solutions ou techniques de sécurité seront utilisées pour sécuriser le réseau de capteur. Dans la table 3.1, nous comparons chacune des solutions étudiées selon le type de sécurité utilisé, et nous faisons des observations sur elles et leurs potentielles limitations.

Comme nous pouvons le remarquer, les solutions de sécurité proposées se basent sur un certain genre de cryptographie *light* comme seconde ligne de défense combiné avec des techniques de sécurité telles que la détection de comportement anormaux, le calcul de position robuste, la vérification de position et les algorithmes simple combinés avec un matériel hard additionnel.

Algorithme	cryptographie	Détection de comportements anormaux	Calcul de position robuste	Vérification de position	Algo. simples sécurisés	observations
HirLoc (Lazos, et al., 2006)	Chiffrement/authentification des ancres. Pré-chargement global des clés	-	-	-	Oui	Requiert un matériel spécial (antenne directionnelle) dans les ancres
SeRLoc (Lazos, et al., 2004)	Chiffrement/authentification des ancres. Pré-chargement global des clés	-	-	-	Oui	Requiert un matériel spécial (antenne sectorielle) dans les ancres. Ne considère pas les ancres hostiles
ROPE (Lazos, et al., 2005)	Chiffrement/authentification des ancres. Pré-chargement global des clés	-	-	Vérification des distances par les ancres	Oui	Requiert un matériel spécial (antenne directionnelle) dans les ancres
SPINE (Capkun, et al., 2005)	Cryptographie symétrique ou à clé publique pour authentifier les estimations de position	-	Multilatération vérifiable	-	-	Horloges nanosecondes. Utilise les ultrasons. Grand nombre d'ancres
DRBTS (Srinivasan, et al., 2006)	Chiffrement en utilisant les clés de groupe	Basé sur la réputation et la confiance	-	-	-	Réseau dense. Les observations bénignes doivent former la majorité
LAD (Du, et al., 2005)	-	-	-	Connaissance de déploiement	-	Requiert la connaissance de déploiement
Echo (Sastry, et al., 2003)	-	-	-	propriétés physiques des fréquences radio et ultrason	-	La majorité bénigne. Pas d'ancres hostiles. Utilisation des ultrasons
Li et al. (Li, et al., 2005)	-	-	Méthodes statistiques robustes	-	-	Les observations bénignes doivent former la majorité
Liu et al. (Liu, et al., 2005(a))	Authentification des messages beacons en utilisant les clés de pairs partagées	Calcul de distance et RTT	-	-	-	Requiert des ancres redondantes
Liu et al. (Liu, et al., 2005(b))	Authentification avec établissement de clés de pairs	-	Basé sur le vote	-	-	Ancres bénins doivent former la majorité

Tableau 3. 1 : comparaison des systèmes de localisation sécurisés.

5. CONCLUSION

La localisation dans les réseaux de capteurs est essentielle à la fois pour les protocoles de communication (routage géographique) que pour certaines applications (suivi de véhicules). De nos jours, les techniques de localisation sont multiples. Une bonne maîtrise et connaissance de ces diverses méthodes sont nécessaires afin de judicieusement dimensionner sa propre solution de localisation. Vu les caractéristiques spécifiques des réseaux de capteurs, cette solution ne doit pas être surdimensionnée, sinon elle entraîne un surcoût en termes d'énergie et d'overhead ce qui peut la rendre impraticable.

La localisation est un domaine de recherche dont l'attrait est croissant ces dernières années et de nombreuses propositions ont été faites. Toutefois aucune d'entre elles nous paraissent robustes contre les manipulations abusives d'un adversaire malveillant, dont le but initial est de fausser les estimations de position des nœuds.

Dans ce chapitre, nous avons présenté le contexte dans lequel s'inscrit le problème de la localisation dans les réseaux de capteurs. Nous avons également étudié les systèmes de localisations du point de vue sécurité, leur vulnérabilité face aux différentes attaques qui peuvent compromettre le fonctionnement entier du réseau de capteurs. En premier lieu, nous avons présenté les trois composants d'un processus de localisation : estimation des distances, calcul des positions et algorithme de localisation. Ensuite nous avons décrit chacun de ces composants, et les différentes méthodes pour les compromettre. Finalement nous avons présenté les mécanismes de défense utilisés pour sécuriser les systèmes de localisation.

DEUXIEME PARTIE :

LES CONTRIBUTIONS A LA RECHERCHE

Toutes les preuves conduisent inévitablement à des propositions qui n'ont pas de preuve ! Toutes choses sont connues parce que nous voulons croire en elles !

Frank Herbert, Les enfants de Dune.

La science, dans ses résultats, est plus magique que la magie : c'est une magie à preuves !

Jean-Marie Adia, La carte d'identité.

Chapitre IV

PROPOSITION D'UN PROTOCOLE DES DONNEES AGREGEES SECURISEES : RAHIM

Sommaire

1. INTRODUCTION

Les réseaux de capteurs sont typiquement déployés dans des zones non surveillées, ce qui les rend vulnérables à plusieurs attaques dans lesquelles l'intrus peut prendre le contrôle d'un ou plusieurs nœuds capteurs pour perturber le bon fonctionnement du réseau. La capture physique des capteurs peut révéler toutes les informations et les primitives de sécurité à l'attaquant, qui va facilement déclencher diverses attaques telles que l'altération des données, la négligence de messages, le jamming, etc (Maarouf, et al., 2009), (Ning, et al., 2005). Dans le contexte de l'agrégation des données, un nœud capteurs compromis peut authentifier avec succès des données erronées à ses voisins, qui n'ont aucun moyen de distinguer entre les données erronées et les données légitimes (Perrig, et al., 2004). Il peut altérer le résultat partiel ou final de l'agrégat afin de fabriquer de faux événements pour tromper les décideurs, ou endommager une partie du réseau. Dans les applications critiques, l'utilisation de données corrompues peut avoir des conséquences désastreuses.

Dans ce chapitre, nous présenterons notre première proposition nommée RAHIM (Labraoui, et al., 2011(a)), pour sécuriser l'agrégation des données dans les réseaux de capteurs à architecture hiérarchique. Nous commencerons d'abord par présenter les motivations de cette proposition, ensuite nous présenterons les détails de notre algorithme et l'analyse de sécurité ainsi que l'évaluation de ses performances.

2. MOTIVATION

Dans notre travail, nous nous somme intéressés à l'intégrité des données pour faire face aux altérations du résultat final de l'agrégat causées par les nœuds agrégateurs ou par les nœuds ordinaires du réseau.

Notre étude de l'art effectuée dans le chapitre II a décelé deux inconvénients principaux des solutions existantes focalisant sur l'intégrité des données de l'agrégation, à savoir, le *surcoût excessif* et *le rejet total des données*.

Le problème du surcoût excessif est du à la génération d'un overhead en communication et en calcul CPU ainsi que le coût induit par la phase de vérification interactive souvent nécessaire entre la station de base et les capteurs.

Le second problème, qui est le plus crucial est le rejet total des données. En effet, la violation de l'intégrité des données à n'importe quel niveau du réseau, oblige la station de base à rejeter

le résultat final de l'agrégat reçu même si les résultats partiels sont corrects, ce qui implique l'annulation de toutes les étapes du processus d'agrégation. Une grande partie des données correctes sont perdues, et l'effort fourni par les nœuds pour contribuer à cette agrégation est aussi perdu et par conséquent, les ressources précieuses de ces capteurs sont donc inutilement gaspillées. Puisque l'annulation d'un processus d'agrégation implique forcement un recommencement dès le début.

Dans ce chapitre, nous présentons un nouvel algorithme nommé RAHIM (**R**obust **A**daptive approach based on **HI**erarchical **M**onitoring) pour résoudre les problèmes cités ci-dessus et améliorer la fiabilité et la disponibilité des données agrégées dans les réseaux de capteurs clustérisés. La pierre angulaire de notre proposition est la gestion d'un nouveau mécanisme de surveillance nommé *surveillance hiérarchique*. Cette surveillance est effectuée au sein de chaque cluster et permet de vérifier l'intégrité et l'exactitude des résultats d'agrégation selon deux niveaux de surveillance, mais seulement en cas de besoin, i.e. uniquement lorsqu'une fraude a été détectée. Ce système permet à la station de base de recevoir le résultat correct même en présences de nœuds compromis. Contrairement aux solutions existantes, qui n'ont qu'une seule règle de gestion de sécurité, notre proposition possède plusieurs règles de gestion et adapte sa réaction en fonction du scénario d'attaque. L'exactitude de l'agrégation ainsi que l'efficacité énergétique ont été les buts principaux pour la conception de notre algorithme.

3. SPECIFICATIONS GENERALES

Dans cette section, nous spécifions les hypothèses sur le réseau ainsi que le type d'attaque considérée.

3.1 Modèle du réseau

Dans notre étude, nous considérons un réseau de capteurs à architecture clustérisée constituée de n nœuds capteurs statiques et une station de base (SB) statique. Chaque nœud capteur a un identificateur Id_i unique dans le réseau, où $1 \leq i \leq n$. Le réseau est divisé en groupes nommés clusters, et chaque cluster est géré par un cluster-head (CH) nommé également chef de groupe, qui jouera le rôle d'agrégateur. Nous avons opté pour l'algorithme de formation de clusters proposé par Sun et al. (Sun, et al., 2006), dans lequel les clusters (nommés cliques) sont formés de façon à ce que chaque nœud capteur membre du cluster est relié directement avec tous les autres membres du cluster par un seul saut. Par conséquent,

lorsqu'un nœud capteur envoie un message au *CH*, ce message peut être écouté et reçu simultanément par tous les autres capteurs du cluster, comme dans le système « watchdog » dans (Maarouf, et al., 2009).

Nous supposons que tous les nœuds peuvent atteindre directement la *SB* comme ça été supposé dans le protocole LEACH (Handy, et al., 2002). Néanmoins, pour minimiser l'overhead de communication dans le réseau, uniquement les *CHs* peuvent communiquer directement avec la *SB*, les autres nœuds ordinaires communiquent seulement avec les nœuds capteurs de leur cluster correspondant. Les nœuds capteurs utilisent donc deux niveaux de puissance pour la communication, une puissance minimale P_{min} lorsqu'ils communiquent entre eux au sein du même cluster, et une puissance plus forte P_{max} lorsque le cluster-head communique avec la station de base. Nous supposons que tous les capteurs sont physiquement identiques (par exemple des Mica2), alors que la station de base est supposés être robustes et dotée de ressources inépuisables.

Nous supposons également, qu'il existe un canal de communication fiable que les nœuds capteurs peuvent utiliser pour alerter la *SB* en cas de fraude, et que la limite de sa latence est connue, i.e. nous considérons la disponibilité d'une méthode pour les nœuds de capteurs pour communiquer (de manière fiable) avec la *SB* sans passer par l'agrégateur. Ce canal d'alarme est plus couteux que le lien entre l'agrégateur et la station de base ; cependant, puisqu'il n'est utilisé qu'en cas de fraude, son surcoût n'est pas un facteur dans des conditions normales du processus d'agrégation.

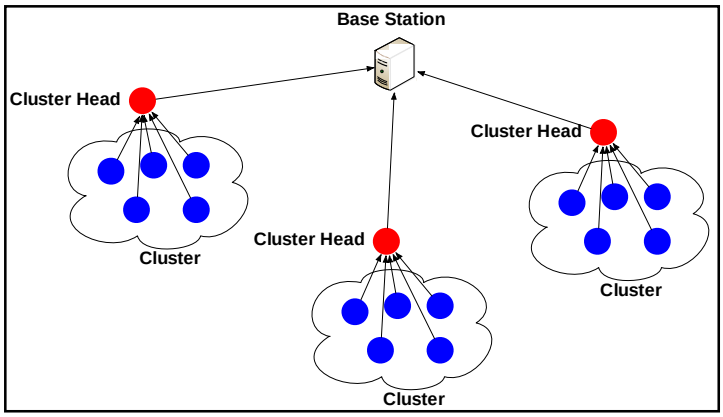


Figure 4. 1 : le modèle du réseau.

3.2 Modèle d'attaque

Nous supposons que l'attaquant prend le contrôle d'un nombre arbitraire de capteurs, y compris la connaissance de toutes leurs clés secrètes. Le seul but de l'attaquant est de déclencher l'attaque stealthy (Pai, et al., 2010), i.e. persuader la station de base pour qu'elle accepte un faux résultat d'agrégation qui soit largement différent du résultat correct de l'agrégat. Cette attaque peut être effectuée soit par *injection directe* de mesures erronées ou bien par *altération de la valeur* de l'agrégation.

Dans notre travail, nous considérons le modèle de l'attaquant réaliste, i.e que l'attaquant peut compromettre au plus t parmi n nœuds capteurs à l'intérieur du cluster $(t < 2)$.

Finalement, on suppose que la station de base est robuste et digne de confiance (crédible). En outre on admet qu'un attaquant ne peut la compromettre en un temps limité.

La table 4.1 résume la notation utilisée dans ce travail.

Notation	Description
BS	Station de base
CH	Cluster-Head qui joue le rôle d'agrégateur
PSUP_L1	Moniteur principal du 1er niveau
PSUP_L2	Moniteur principal du 2ème niveau
$MONIT_i$	Moniteur secondaire n°i
Id_i	Identificateur du capteur i
K_i^{BS}	Clé symétrique partagée entre le capteur *i* et la *SB*
$MAC_{K_i^j}(m)$	Code d'authentification du message *m* avec la clé partagée entre *i* et j
AGG_i	Résultat de l'agrégat calculé par le capteur *i*
N_a	un nonce envoyé par la *BS* lors de sa requête

Tableau 4. 1 : Notation.

4. OBJECTIFS DE CONCEPTION

Dans les conditions mentionnées ci-dessus, nous avons proposé un mécanisme de sécurité qui puisse être capable de faire face aux altérations de l'agrégation des données,

causées par des nœuds compromis. De ce fait, nous avons visé dans la conception de notre mécanisme de défense d'atteindre aussi bien que possible les objectifs suivants:

Exactitude : le résultat de l'agrégation doit résister aux nœuds compromis et aux manipulations des données. Ainsi, le résultat accepté par la *SB* ne devrait pas être trop différent de la valeur correcte.
Disponibilité : aussi longtemps que l'attaque persiste, la *SB* peut toujours obtenir un résultat d'agrégation correcte même si tous les agrégateurs et une partie des capteurs ordinaires ont été compromis dans le cluster.
Efficacité : l'algorithme doit assurer les objectifs de sécurité d'une manière *light*, i.e. générer un coût bas en communication et consommer peu d'énergie.

5. LE PROTOCOLE SECURISE PROPOSE : RAHIM

Dans cette section, nous présentons notre algorithme pour sécuriser l'agrégation des données. Nous commencerons par donner une vue d'ensemble sur l'algorithme ensuite nous le détaillons.

5.1 Vue d'ensemble du protocole proposé

La conception de RAHIM est basée sur deux principes: *agrégation indépendante et exactitude basée sur une surveillance hiérarchique et adaptative*. Il est également bâti sur un concept principal : « aucune confiance n'est supposée dans les nœuds capteurs ». Pour cela, nous désignons deux niveaux de surveillance hiérarchiques pour assurer l'intégrité et l'exactitude du résultat de l'agrégation. Dans le premier niveau de surveillance, nous dédions un nœud capteur qui joue le rôle d'un superviseur principal (nommé *PSUP_L1*). Ce *PSUP_L1* surveille le comportement du *CH*. Alors que dans le second niveau, le reste des nœuds capteurs membres d'un cluster jouent le rôle de superviseurs secondaires et surveillent en même temps le comportement du superviseur principal du niveau 1 (*PSUP_L1*) ainsi que le comportement du *CH*. Pour des raisons d'efficacité, nous dédions un deuxième superviseur principal de niveau 2 (*PSUP_L2*) parmi ces superviseurs secondaires. Ce *PSUP_L2* gère la tâche de surveillance dans le second niveau. Cependant, dans une situation normale (sans attaque), le *CH* effectue sa fonction d'agrégation et la transmet à la *SB* qui l'acceptera sans aucun overhead additionnel en communication.

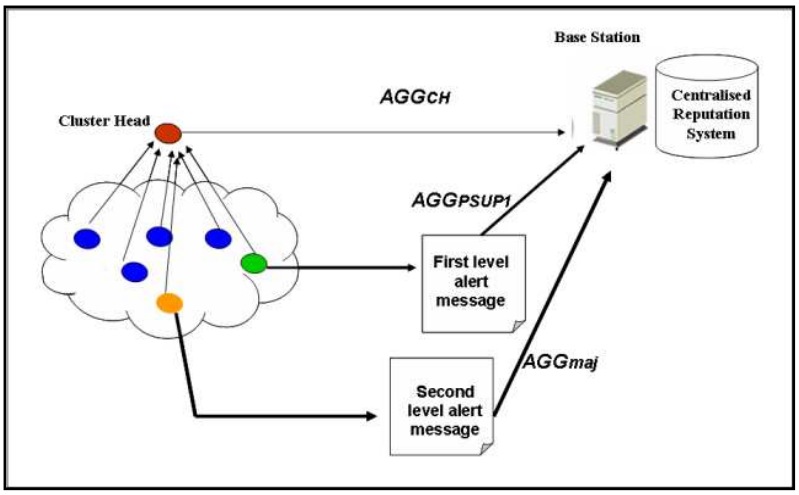


Figure 4. 2 : Architecture du protocole proposé.

5.2 Détails du protocole

Notre algorithme de sécurité des données agrégées évolue en *trois étapes régulières* et éventuellement en *deux étapes spéciales*. Lorsque le *CH* et le *PSUP_L1* agissent de manière correcte (ne sont pas corrompus), le processus d'agrégation se termine au bout de trois étapes régulières. Cependant, si la compromission du *CH* et/ou du *PSUP_L1* a été détectée, le protocole exécute l'étape 4 et/ou l'étape 5 comme étapes spéciales supplémentaires, en fonction du scénario d'attaque.

5.3 Les étapes régulières

1- Initialisation*:* cette étape inclus la phase d'initialisation avant le déploiement des nœuds capteurs, dans laquelle la *SB* assigne chaque capteur *i* par un identifiant unique Id_i et une clé symétrique k_i^{SB} que la *SB* partage avec le capteur i. En outre, on suppose qu'un capteur peut initialiser des clés paires avec chacun de ses voisins de manière sécurisée lors du déploiement du réseau.

La formation des clusters s'effectue après le déploiement, dans laquelle les capteurs s'auto-organisent en cliques disjointes. Lorsque les clusters (cliques) sont formés, les nœuds de chaque cluster élisent un d'entre eux comme cluster-head (*CH*) qui va jouer le rôle de l'agrégateur. Chaque *CH* envoie à la *SB* la liste des capteurs membres de son propre cluster.

Le processus de l'agrégation peut être initialisé suite à une requête de la *SB*. La *SB* diffuse un message de requête aux *CHs*. Dans chaque requête, la *SB* élit *dynamiquement* un superviseur principal du premier niveau (*PSUP_L1*) et un superviseur principal du second

niveau (*PSUP_L2*) dans chaque cluster. Elle embarque ces deux identificateurs dans le message de requête diffusé. Cependant, il est important de mentionner que le choix de *PSUP_L1* et *PSUP_L2* n'est pas trivial, et qu'il n'est pas fait de manière aléatoire. Nous supposons que la *SB* a la capacité d'avoir son opinion sur le comportement des capteurs, en maintenant un système de réputation centralisé. Donc le *PSUP_L1* et *PSUP_L2* sont élus parmi les nœuds ayant un score de réputation élevé. Les détails de ce système de réputation ne relève de notre travail. Lorsque le *CH* reçoit la requête de la *SB*, il la diffuse à son tour à tous les capteurs membres de son cluster.

2- Filtrage des données et agrégation*:* notre algorithme exploite la nature à diffusion de la transmission radio pour distribuer la tâche de l'agrégation à tous les membres du cluster, i.e. tous les voisins de chaque agrégateur, participent à la fonction de l'agrégation et collectent les données (les mesures) à travers l'écoute passive. Cependant, malgré la participation de tous les nœuds à l'opération d'agrégation, seul le *CH* qui s'occupera d'envoyer le résultat effectué par lui-même à la *SB*. Les autres nœuds jouent le rôle de superviseurs pour assurer l'exactitude du résultat de l'agrégat et réagissent selon leur rôle si cette exactitude a été altérée. Nous supposons que le *CH* n'incluse pas sa propre mesure dans la fonction de l'agrégation.

Le processus de l'agrégation s'effectue en époque (round), à l'intérieur de chaque cluster, comme c'est le cas dans tous les protocoles de la littérature (la synchronisation est bien entendu requise). Le l^{ème} round de l'agrégation dans le cluster Cl_i mené par le cluster-head CH_i est effectué comme suit :

$$i \rightarrow *: Id_i, S_i \qquad (1)$$

Chaque nœud capteur $i \in Cl_i$ excepté CH_i, diffuse ses mesures collectées S_i. Il faut noter qu'un attaquant ne peut pas personnifier un nœud *i*. en effet, la communication dans le cluster est à un seul saut seulement et les messages ne passent pas par des nœuds intermédiaires où ils peuvent être altérés de manière malicieuse. Par conséquent, nous n'avons pas besoin d'utiliser un MAC (code d'authentification de message) pour garantir l'intégrité du message. Cependant, pour détecter les altérations non malicieuses de l'environnement, nous utilisons le mécanisme du contrôle d'erreur CRC (Cyclic Redundancy Check) (Ning, et al., 2005).

Chaque nœud $x \in Cl_i$, reçoit (de manière passive) touts les messages diffusés, envoyés par les membres du cluster.

Néanmoins, afin d'effectuer toute fonction d'agrégation, nous ajoutons une étape préliminaire au modèle de l'agrégation, dans laquelle, après la réception des mesures de tous les nœuds capteurs, chaque nœud (y compris l'agrégateur) effectue localement *un filtrage* des données reçues avant l'agrégation, et tente d'identifier les mesures erronées potentielles pour les éliminer et ne pas les introduire dans le calcul de la fonction de l'agrégation. Cette étape préliminaire est *très importante* avant d'effectuer l'agrégation. En effet, si l'adversaire modifie les mesures des capteurs en manipulant directement l'environnement (par exemple, mettre une source de chaleur près d'un capteur pour faire augmenter la mesure de la température), il pervertira sûrement les résultats d'agrégation.

Pour vérifier la fiabilité des données, une technique statistique robuste doit être appliquée pour identifier les mesures aberrantes. Un très bon algorithme de détection des valeurs aberrantes doit détecter la plupart des fautes et le nombre de faux positifs (un faux positif est la détection d'une vrai valeur comme étant une valeur aberrante) doit être petit. RAHIM utilise la médiane qui est classée statistiquement parmi les fonctions robustes pour détecter les valeurs aberrantes (Wagner, 2004). Elle est basée sur des règles et donc ne requière pas de comparaison avec les déviations standards estimées (qui sont affectées par la présence des valeurs aberrantes) des mesures pour décider si une valeur est aberrante ou non (Kumar, et al., 2009).

Pour chaque nœud capteur dans le cluster, la médiane des mesures des nœuds voisins est calculée. Si une mesure s'éloigne de la médiane de plus d'un seuil, elle est déclarée comme valeur aberrante. L'algorithme est défini dans *Algorithm 1*. Il est supposé que l'écart type moyen de la mesure d'erreur (la calibration de l'erreur) du capteur utilisé est fournie par le fabricant du capteur. Le seuil est pris comme deux fois l'erreur maximum (Kumar, et al., 2009).

Après le filtrage des mesures erronées et le calcul de la fonction d'agrégation localement par chaque nœud capteur du cluster, seulement le *CH* est autorisé à envoyer le résultat (AGG_{CH}) à la *SB*.

$$CH \rightarrow BS:\ Id_{CH}, AGG_{CH} || MAC_{K_{CH}^{BS}}(AGG_{CH}, N_a) \quad (2)$$

S'il existe des valeurs erronées (aberrantes), le *CH* incluse l'identificateur des nœuds concernés dans le message envoyé à la *SB*. Ce message est directement transmit du *CH* à la *SB* comme dans le protocole LEACH (Handy, et al., 2002).

Algorithm 1: Data filtering and aggregation algorithm

Input: S *set of received readings from the sensors in the cluster*
Output: aggregation result
$S_1 = \phi$
$MED = median\_of\_readings$
For each reading i *of* S *do*
  If $abs(i - MED) < threshold$ *then*
    $S_1 = S_1 \cup \{ i \}$
  EndIf
EndDo
Compute aggregation function on subset S_1

3- Validation de l'agrégation validation: lorsque la *SB* reçoit le résultat envoyé par le *CH*, elle calcule le MAC de la valeur d'agrégation reçue AGG_{CH} pour vérifier l'intégrité de la donnée. Si la *SB* ne reçoit pas d'alarme dans une limite de temps donné, elle suppose qu'il n'y a aucun nœud capteur qui désapprouve le résultat de l'agrégation et conclut que le résultat reçu AGG_{CH} est correct, et qu'il n'y a eu aucune activité malicieuse lors du processus d'agrégation. Cela veut dire que le superviseur du premier niveau ainsi que les superviseurs du second niveau approuvent la valeur AGG_{CH}. La limite de temps dans laquelle la *SB* attend les l'arrivées des alarmes dépend du niveau d'urgence de l'application déployée dans le réseau de capteur.

Cependant, si la *SB* reçoit un message d'alarme du premier niveau de la part du *PSUP_L1*, qui contient la valeur de l'agrégation AGG_{PSUP_L1} (calculée par *PSUP_L1*), et n'a pas reçu de message d'alarme du second niveau, elle conclut que les superviseurs secondaires approuvent le résultat AGG_{PSUP_L1}. Elle accepte donc AGG_{PSUP_L1} à la place de AGG_{CH}. Mais si par contre, la *SB* reçoit un message d'alerte du second niveau avec la nouvelle valeur d'agrégation AGG_{maj}, elle conclut donc que les superviseurs secondaires désapprouvent soit la valeur AGG_{CH} envoyée par le *CH* ou bien la valeur AGG_{PSUP_L1} envoyée par *PSUP_L1*. Cette alarme de second niveau est *prioritaire* par rapport à celle du premier niveau, puisqu'elle est déclenchée par la majorité des superviseurs secondaires par vote majoritaire.

Finalement, la *SB* calcule le résultat total de l'agrégation $AGG = f(AGG_i | \forall i, Cl_i)$, à partir des résultats partiaux générés par chaque cluster.

5.4 Les étapes spéciales

4- Surveillance de premier niveau : Le superviseur principal de premier niveau *PSUP_L1* surveille le résultat de l'agrégation (AGG_{CH}) envoyé par le *CH* à la *SB*, par écoute passive. Il compare ce résultat avec le sien AGG_{PSUP_L1}. Dans le meilleur des cas, lorsque AGG_{CH} est correcte, le *PSUP_L1* ne déclenche aucune alarme de premier niveau. Cela veut dire que PSUP_L1 approuve le résultat de l'agrégation.

Cependant, si le PSUP_L1 désapprouve la valeur AGG_{CH}, i.e. détecte que l'agrégateur a falsifié le résultat, il déclenche un message d'alarme qui contient son propre résultat AGG_{PSUP_L1}.

$$PSUP_{L1} \rightarrow BS:\ Id_{PSUP\_L1}, AGG_{PSUP_{L1}} || MAC^{BS}_{K_{SUP_{L1}}}(AGG_{PSUP_{L1}}, N_a) \quad (3)$$

Comme pour le CH, s'il existe des valeurs erronées (valeurs aberrantes), le *PSUP_L1* inclut les identificateurs des nœuds capteurs correspondants dans le message envoyé à la *SB*.

5- Surveillance de second niveau : comme nous n'avons supposé aucune confiance ni sur le *CH* ni sur le *PSUP_L1*, une surveillance supplémentaire est prévue et effectuée par le reste des nœuds capteurs nommés *superviseurs secondaires* ($MONIT_i$). Ces $MONIT_i$ sont responsable de surveiller le comportement du *CH* et du *PSUP_L1* lors de leur transmission de leur résultat à la *SB*. Sans la compromission de ces deux capteurs importants *CH* et *PSUP_L1* (puisqu'ils représentent une cible potentielle pour un attaquant), aucune action n'est envisagée, et donc aucun message d'alerte n'est envoyé à la *SB*.

Cependant, si les superviseurs $MONIT_i$ detectent la fraude de *PSUP_L1* seul ou la fraude du *CH* et du *PSUP_L1*ensemble, ils coopèrent pour déclencher un message d'alarme de second niveau à la *SB*. Cette alarme contient la valeur d'agrégation AGG_{maj} basée sur le vote majoritaire. Si nous supposons que le nombre des superviseurs $MONIT_i$ est *n*; il n'est pas judicieux en terme d'efficacité d'envoyer *n* messages d'alertes à la *SB*. Contrairement aux protocoles existants, nous désignons un superviseur principal parmi ces superviseurs secondaires nommé *PSUP_L2*, qui collecte les messages de plaintes de chaque $MONIT_i$ qui désapprouvent le résultat de l'agrégat, et effectue un vote majoritaire pour générer un message d'alerte.

$$MONIT_i \rightarrow PSUP_{L2}:\ Id_{MONIT_i}, H(AGG_{MONIT_i}) || MAC^{BS}_{K_{MONITi}}(AGG_{MONITi}, N_a) \quad (4)$$

Amélioration: il est évident que la surveillance du second niveau est plus couteuse que celle du premier niveau, à cause de la transmission des messages de plaintes dans le cluster. Cependant, du moment que le résultat de l'agrégation peut avoir différentes tailles, chaque superviseur $MONIT_i$ envoie juste l'empreinte de son résultat $H(AGG_i)$ (le hachage de AGG_i) à la place de la valeur du résultat AGG_i, et cela dans le but de réduire l'overhead de transmission. Normalement, puisque tous les nœuds capteurs du cluster écoutent les mêmes messages (les mêmes mesures), tous les nœuds dignes de confiance devront avoir la même valeur de l'agrégation AGG_i. Par conséquent, ils envoient la même empreinte $H(AGG_i)$ du résultat d'agrégation. Nous supposons que tous les capteurs utilisent la même fonction de hachage *H*. Après avoir collecté un nombre suffisant de messages de plaints, le superviseur *PSUP_L2* calcule le XOR des MAC reçus et envoie ce message d'alerte de second niveau à la SB:

$$PSUP\_L2 \rightarrow BS: Id_{PSUP_{L2}}, AGG_{maj} || \oplus MAC^{BS}_{K_{MONITi}}(AGG_{MONITi}, N_a) \qquad (5)$$

Si un nœud capteur *x* d'un cluster n'a pas réussi à envoyer son résultat AGG_i, le *PSUP_L2* inclut son identificateur Id_x dans le message d'alerte de second niveau envoyé à la *SB*, pour notifier que le calcul du XOR (le ou éxclusif) des MAC n'a pas été calculé sur la contribution du nœud *x*. Dans le cas de valeurs de hachage différentes (et donc des valeurs d'agrégation contradictoires), *PSUP_L2* choisit la valeur de hachage majoritaire $H(AGG_{maj})$ comme valeur de hachage du résultat d'agrégation du cluster. Dans le cas où $H(AGG_{PSUP\_L2}) \neq H(AGG_{maj})$, *PSUP_L2* demande à chaque capteur parmi ceux qui ont reporté la valeur majoritaire du hachage, de lui envoyer le résultat d'agrégation AGG_i. Dans tous les cas, *PSUP_L2* calcule le XOR des MAC uniquement sur les MACs relative à la valeur de hachage majoritaire, et reporte l'identificateur *Id* de chaque nœud capteur donc le résultat de son agrégation est différent du résultat majoritaire AGG_{maj}.

Comme nous l'avons mentionné dans la section 3.2, le nombre de nœuds capteurs compromis est plus petit que celui des nœuds dignes de confiance au sein d'un cluster. Donc, le superviseur *PSUP_L2* ignore tout message d'alerte, s'il reçoit moins que *n/2* messages d'alerte. Cela veut dire que les nœuds compromis ne peuvent pas déclencher d'alerte contre un résultat correct puisqu'ils ne constituent pas la majorité dans le cluster. Et c'est là toute la puissance de ce second niveau de surveillance.

6. ANALYSE DE SECURITE

L'analyse de sécurité de notre algorithme RAHIM repose sur :

- *La résistance contre l'attaque d'injection de données erronées:* est ce qu'un attaquant peut altérer avec succès le résultat de l'agrégation en fabricant une mesure erronée ?
- *La résistance contre l'attaque de falsification d'agrégation*: est ce qu'un attaquant peut persuader la *SB* à accepter un faux résultat d'agrégation après falsification du résultat ?
- *Resistance contre le rejet de données:* est ce que la disponibilité est assurée même lors de la persistance d'activités malveillantes dans le réseau ?
- *Tolérance aux pannes des agrégateurs:* est ce que le protocole peut assurer l'exactitude du résultat d'agrégation en cas de panne du nœud agrégateur.

6.1 Resistance contre l'attaque d'injection de données erronées

L'attaque d'injection de données erronées se produit lorsqu'un attaquant modifie des mesures collectées par des nœuds dont il détient le contrôle (Can, et al., 2006). Il est très difficile de détecter cette attaque, puisqu'elle fait partie des attaques internes (insider attack).

Cependant, la majorité des solutions existantes dans les données agrégées sécurisées, supposent le plus souvent que les mesures collectées par les capteurs sont correctes (non manipulées) (Labraoui, et al., 2011(a)) ou bien acceptent uniquement des mesures dont les valeurs appartiennent à un intervalle limité par une valeur minimale et une valeur maximale, selon l'application déployée (Bagaa, et al., 2007). Cependant, cette dernière supposition, réduit l'impact d'injection de données erronées mais il est très difficile de différencier entre les événements d'urgences détectés par des nœuds légitimes et les événements malicieux. D'autres protocoles relatifs au concept de confiance, ont récemment émergé. Ils sont inspiré par la vie sociale et utilisent le nouveau paradigme de réputation inspiré par les comportements humains afin d'isoler les données erronées injectées dans le réseau (Maarouf, et al., 2009), (Kumar, et al., 2009), (Junbeom, et al., 2005). Néanmoins, ces approches sont vulnérables aux attaques *bad mouthing* dans lesquelles un nœud compromis peut accuser un nœud digne de confiance d'avoir entrepris des actions malicieuses ou bien plaider en faveur d'un nœud malicieux. En outre, un surcoût très élevé est généré par les échanges périodiques des valeurs de réputation entre les nœuds. Dans notre protocole, nous avons fait face avec l'attaque d'injection de données erronées de manière *light* en ajoutant une étape préliminaire

au modèle d'agrégation des données, dans lequel un algorithme de filtrage de donnée est effectué localement avant de calculer la fonction d'agrégation.

Pour prouver l'efficacité de l'algorithme de filtrage de données basé sur la médiane, nous l'avons simulé en utilisant l'environnement Matlab.

Imaginons un scenario d'une application typique pour la collecte de la température: un groupe de nœuds capteurs tels que Micas sont déployés pour collecter des échantillons de température dans une zone d'intérêt. Supposons que chaque groupe est constitué de *n* nœuds qui s'auto-organisent dans un cluster. Chaque minute, ils mesurent la température et la transmettent au cluster-head. Il est clair que les mesures des capteurs telles que la température sont très corrélées dans une région géographique assez restreinte. Cette corrélation dans les échantillons est un phénomène naturel.

L'échantillon des mesures est généré par la fonction *randn*. Un attaquant est simulé par une fonction qui remplace ces éléments d'échantillon par une valeur erronée, qui correspond à une proportion déterminée par *k*. La valeur erronée est largement différente de la valeur prévue réelle de l'échantillon. Pour obtenir l'objectif de l'attaquant, i.e., une distorsion maximale, nous avons effectué 50 simulations pour différentes valeurs de *k* (i.e., différentes proportions de nœuds compromis).

La Figure 4.3 illustre la déviation d'erreur du calcul de la médiane pour une application typique de collecte de température. L'erreur de déviation est très significative en dessous de 50 pourcent de nœuds compromis. Mais pour une valeur de *k* plus élevée, le résultat du calcul de la médiane décline rapidement.

Dans la Figure 4.4, nous remarquons que la valeur d'agrégation après le filtrage des données erronées est très proche de la moyenne réelle de l'échantillon originale. Dans les deux figures 4.3 et 4.4, la médiane a un pic (breakdown point) de 50.

En conclusion, les résultats de simulation de l'attaque d'injection de données erronées démontrent que le calcul de la médiane est très simple et n'induit pas d'overhead de calcul important et produit toujours une estimation précise jusqu'à 50 pourcent de nœuds compromis dans le cluster. La médiane est donc une méthode statistique robuste en présence de plusieurs valeurs erronées (aberrantes) et produit zéro faux positifs en dessous de ce seuil. Ce qui représente un résultat intéressant par rapport à nos hypothèses de départ. Ainsi, notre protocole de sécurisation des données agrégées est immunisé contre l'attaque d'injection de données erronées.

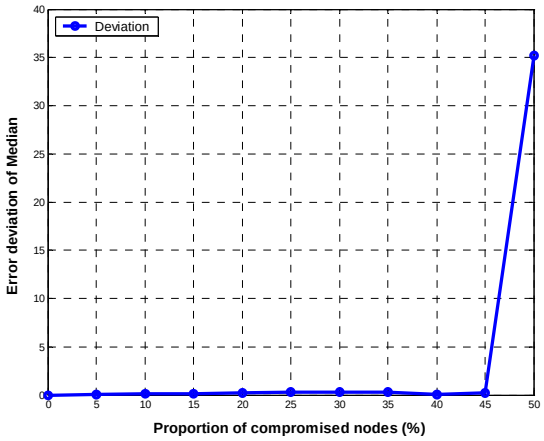


Figure 4. 3 : Erreur de déviation de la médiane.

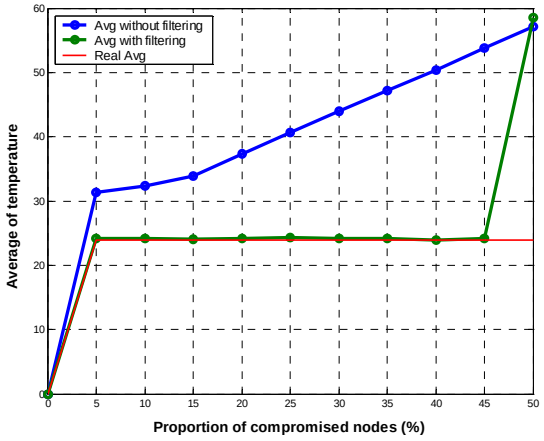


Figure 4. 4 : Comparaison de la fonction d'agrégation moyenne.

6.2 Résistance contre l'attaque de falsification d'agrégation

L'agrégateur représente *le nerf central* du processus d'agrégation, et sa compromission mène au succès de l'attaque. Il est donc très important de vérifier le comportement correct des nœuds agrégateurs. Pour cette raison, nous utilisons l'approche basée sur la surveillance hiérarchique pour assurer l'exactitude du résultat d'agrégation. Cependant, puisqu'aucune confiance n'est supposée en aucun capteur dans le cluster,

plusieurs scénarios d'attaques peuvent se produire. Nous expliquons ces scénarios dans ce qui suit :

- **Compromission du cluster head** : si le *CH* est compromis, il peut fabriquer (forger) des résultats d'agrégation arbitraires et générer les MAC correspondants de ces résultats erronés. Dans notre protocole, une telle attaque est bien défendue, puisque nous utilisons un premier niveau de surveillance. Le *PSUP_L1* déclenche une alerte contre les faux résultats envoyés par le CH et fournit à la *SB* son propre résultat d'agrégation.

- **Attaque sélective sur le superviseur principal de premier niveau :** Une idée évidente pour l'attaquant est de compromettre en même temps le *CH* et le *PSUP_L1*. En effet, le *PSUP_L1* peut être complice avec le *CH* et ne dénonce pas la fraude tout simplement en s'obstinant d'envoyer un message d'alerte à la *SB*. Cependant, dans notre protocole, cette attaque est également bien défendue, puisque un deuxième de niveau de surveillance est entrepris dans lequel le *PSUP_L2* déclenche une alerte sur la base des messages de plaintes et fournit à la *SB* le résultat correcte. .

- **Compromission du superviseur principal du second niveau**: si *PSUP_L2* est compromis, il tente de fabriquer un message d'alerte pour persuader la *SB* d'accepter sa valeur (falsifiée) et de rejeter la valeur réelle reçu soit par le *CH* ou par le *PSUP_L1*. Cependant, le *PSUP_L2* ne peut agir seul et forger un MAC légal pour générer le vote majoritaire, et donc il ne peut pas générer un message d'alerte valide pour discréditer le *CH* ou le *PSUP_L1*.

6.5 Resistance contre le rejet de données

Le rejet des données est un problème assez crucial dans les protocoles de sécurité des données agrégées. Un protocole qui souffre de ce type de problème ne peut empêcher les données erronées d'infecter le résultat global d'agrégation, et par conséquent toutes les étapes du processus d'agrégation sont annulées. Notre protocole RAHIM surmonte le rejet total en stoppant localement les données non valides durant la phase d'agrégation (par l'algorithme de filtrage) et en utilisant le concept de surveillance hiérarchique à deux niveaux. Le rôle des moniteurs est de fournir un résultat d'agréation valide à la *SB*, en évitant le rejet de données

lorsque l'intégrité des données a été altérée. Donc notre protocole assure plus de disponibilité que les autres solutions.

6.6 Tolérance aux pannes des agrégateurs

Puisque la tâche de l'agrégation est effectuée de manière distribuées, et que le modèle de notre réseau est basé sur les cliques, il est plus tolérant aux pannes des nœuds agrégateurs que les autres protocoles comme (Du, et al., 2003), (Hu, et al., 2003) et (Przydatek, et al., 2003). Du moment que tous les nœuds du cluster participent au calcul du résultat de l'agrégation, si le *CH* tombe en panne (pour une quelconque raison) durant le processus d'agrégation ; notre protocole peut s'adapter pour récupérer l'échec et continuer l'agrégation à partir du point d'échec.

7. EVALUATION DES PERFORMANCES

Le raisonnement principal de RAHIM est de conserver l'énergie en n'exigeant aucune opération cryptographique et en n'induisant aucun overhead en communication lorsque les nœuds capteurs ont un comportement correcte. Ce raisonnement est légitime uniquement si RAHIM ne consomme pas plus d'énergie dans la transmission des données que les autres protocoles d'agrégation, et si l'énergie consommée par RAHIM lors de la surveillance est moins que celle consommée par les opérations cryptographiques. Dans la section suivante, nous démontrons que ces conditions sont vérifiées dans RAHIM.

7.1 L'overhead de transmission

Le but principal de l'agrégation est la réduction de l'overhead de communication. Or, les mécanismes de sécurité génèrent de part leurs primitives un surplus d'overhead. Notre protocole de sécurité a été conçu de sorte à maintenir aussi bien que possible ce but en introduisant un overhead de transmission assez bas, tout en assurant un niveau de sécurité acceptable sans aucune dégradation des performances du réseau. En se basant sur deux niveaux hiérarchiques de surveillance, la densité des nœuds superviseurs secondaires n'augmente pas la contention pour l'accès au médium. Le protocole est donc indépendant de la taille du réseau contrairement au travail de (Emiliano, et al., 2009) et (Du, et al., 2003). Le choix du modèle de réseau supposé a été inspiré du protocole de formation de cluster proposé par Sun et al. (Sun, et al., 2006). Ce protocole réduit de manière significative l'overhead car l'élection périodique du *CH* à l'intérieur du cluster ne change pas les nœuds membres du

cluster. Alors que dans les approches telles que LEACH (Handy, et al., 2002), TEEN (Manjeshwar, et al., 2001) et APTEEN (Manjeshwar, et al., 2002), commencent par élir d'abord un *CH* et c'est par la suite que les clusters sont formés. Par conséquent, un changement périodique de *CH* implique forcément la formation de nouveaux clusters, et donc implicitement une sur-consommation d'énergie à cause des messages échangés.

Afin de facilité l'analyse et la comparaison des protocoles, nous supposons que dans chaque message transmis, la longueur des données, l'identificateur des nœuds ainsi que les MAC sont de tailles similaires dans la plupart des protocoles. Nous considérons le nombre de messages transmis comme notre métrique pour l'overhead de la transmission. Nous considérons également une transmission fiable dans le cluster avec *n* nœuds capteurs, qui collectent leurs mesures. Pour la deuxième étape, chaque nœud capteur envoie ses mesures au *CH* local. Nous utilisons *m* pour représenter la longueur de la mesure (data reading), *c* pour la longueur de l'identificateur du nœud et du MAC ensemble, *w* pour la longueur de l'identificateur du nœud et du CRC ensemble, et *p* pour la longueur de la valeur de hachage et le MAC ensemble, avec $w < c$.

Dans l'étape suivante, chaque *CH* retransmit le MAC de la valeur d'agrégation. La sortie de la fonction d'agrégation a la même longueur que celle des mesures originales. Différents scénarios d'attaques sont détaillés dans ce qui suit.

- **Scénario 1** : Lorsque les nœuds capteurs se comportent correctement, i.e. sans aucune attaque. Le nombre total des bits transmis lors du processus d'agrégation est égal à $(n+1)m + nw + c$. Pour faire une comparaison avec une méthode d'agrégation sans sécurité (TAG) (Madden, et al., 2002), *n* messages sont agrégés en un seul message au niveau de chaque nœud agrégateur. Ainsi, chaque nœud a besoin de transmettre $n + w$ bits. Ce qui induit une transmission total de $(n+1)m + (n+1)w$ bits. Dans ce scénario, notre protocole implique uniquement la phase d'agrégation, et ne génère pas de messages additionnels. Par rapport à une agrégation non sécurisées, notre protocole génère un overhead de 4 octets seulement.

- **Scenario 2**: si l'agrégateur est le seul nœud compromis dans le cluster, alors l'étape 4 du protocole (surveillance de premier niveau) est exécutée. Dans ce cas, notre protocole génère seulement un message additionnel de $c + m$ bits au processus d'agrégation. Ainsi, le nombre total de bits transmis dans ce scénario est égal à $(n+2)m + nw + 2c$. C'est

un overhead insignifiant par rapport à la réaction des autres protocoles envers la compromission des nœuds agrégateurs.

- **Scenario 3**: Lorsque le superviseur principal de premier niveau *PSUP_L1* est compromis et le *CH* est non compromise, l'étape 5 du protocole (surveillance de second niveau) est exécutée. C'est le *pire des cas* dans lequel l'overhead généré est égal à $(n+3)m + nw + tp + 3c$. t représente le nombre de messages de plaintes avec $t < n$.

- **Scenario 4***:* Dans l'attaque par complicité, lorsque le *PSUP_L1* et le *CH* sont compromis, le *PSUP_L1* s'abstient de générer un message d'alerte contre la fraude de l'agrégateur car les deux sont complices. L'overhead dans ce scénario est donc égal à $(n+2)m + nw + tp + 2c$. .

Selon le protocole de Hu et Evans (Hu, et al., 2003), le nombre total de bits transmit par leur protocole avec un nombre *bd* de nœuds feuille (dans l'arbre) est égal à : $m(2bd+1 - b2 - b) / (b - 1) + c(2bd+1 + bd - b2 - 2b) / (b - 1)$. Lorsque les nœuds feuilles sont éloignés de la *SB* de *d* sauts (hop) et que chaque nœud a *b* nœuds fils.

Pour donner un sens de ce que signifient ces nombres pour des applications typiques, nous choisirons *m*=22 octets, *c*=14 octets, *w*=10 octets et *p=22* octets, en se basant sur les hypothèses présentées dans (Perrig, et al., 2002.) (pour les messages ne contenant pas de MAC, 2 octets sont requis pour le CRC assurant l'intégrité du message).

Soit un réseau constitué de *n* nœuds capteurs avec *n*=16 (*b*=4 et *d*=2), l'overhead total de transmission lorsque chaque nœud transmet sa mesure est égal à 544 octets dans une methode d'agrégation non sécurisée (TAG) contre 1352 octets dans le protocole de Hu et Evan.
Cependant, dans notre protocole l'overhead total de transmission est égal à 548 octets dans le scénario1, 584 octets dans le scénario2, 1060 octets dans le scénario3 et 1024 octets dans le scenario4, si on suppose que le nombre de nœuds légitimes est *t*=10 (en considérant 40% de nœuds compromise dans le cluster).

En résumé, à travers l'analyse et la comparaison, comme illustré dans le tableau 2, on remarque clairement que notre protocole RAHIM ne génère pas un grand overhead de transmission par rapport à un protocole d'agrégation non sécurisé, et avec un overhead assez acceptable.

Leaf Nodes		16	32	64	128
TAG		4.3 KB	8.4 KB	16.6 KB	33 KB
Hu and Evans [Hu, 03]		10.8 KB	38.4 KB	49.4 KB	159.8 KB
Our Scheme	Scenario1	4.3 KB	8.4 KB	16.6 KB	33 KB
	Scenario2	4.6 KB	8.7 KB	16.9 KB	33.3 KB
	Scenario3	8.4 KB	12.5 KB	24.1 KB	47.1 KB
	Scenario4	8.1 KB	12.2 KB	23.8 KB	46.8 KB

Tableau 4. 2*: Comparaison Du surcoût de transmission avec 40% de nœuds compromis.*

7.2 L'Overhead de calcul

La cryptographie consomme beaucoup d'énergie, essentiellement à cause de l'overhead des messages, ce qui par conséquent réduit la durée de vie du réseau (Perrig, et al., 2002.), (Karlof, et al., 2004). Y compris l'énergie consommée par le calcul CPU, chaque primitive cryptographique requiert un temps différent de cycle CPU pour son exécution, menant à une consommation d'énergie différente d'une primitive à une autre. Par exemple, l'algorithme Skipjack requiert 22,044.60 cycles CPU et consomme 71.76 µjoules pour le calcul du MAC d'un paquet de 29 octets (Wander, et al., 2005).

Cependant, la majorité des protocoles proposés pour les données agrégées sécurisées, qui se focalisent sur l'intégrité des données dans les WSN, se basent systématiquement sur des opérations cryptographiques comme preuve d'approbation. Chaque capteur transmet sa mesure à l'agrégateur avec son emprunte MAC. Par conséquent, nous pouvons noter que les protocoles de (Emiliano, et al., 2009) et (Du, et al., 2003) induisent un overhead en communication et en calcul très élevé menant à une consommation d'énergie excessive même en l'absence d'attaque.

Contrairement à ces protocoles, notre solution se base sur les preuves de fraude à la place de preuve d'approbation. De ce fait, tous les nœuds capteurs dans le cluster sauf le cluster-head, jouent le rôle de superviseurs durant le processus d'agrégation. Dans une situation normale, nous n'avons pas besoin de MAC pour garantir l'intégrité des messages lorsque les capteurs transmettent (par diffusion) leur mesure, car la communication entre eux est à un seul saut (one hop), et les messages ne passent pas par des nœuds intermédiaires où ils peuvent être potentiellement corrompus (altérés ou modifiés) par des nœuds malicieux. Cependant le *CH* doit calculer le MAC du résultat de l'agrégation avant sa transmission à la *SB* (puisque cette dernière peut être éloignée et que l'attaquant pourra éventuellement

corrompre le résultat lors de son transfert). De ce fait, nous évitons l'exécution de plusieurs nombre de cycles CPU. Nous évitons donc d'ajouter des octets additionnels aux messages originaux, et sauvegardons l'énergie qui peut être dépensée lors de la transmission de ces octets.

7.3 Cout d'énergie pour la surveillance

L'écoute passive est aussi considérée comme une cause qui gaspille l'énergie (Iima, et al., 2009). Cependant, les superviseurs secondaires ne sont pas soumis à l'écoute durant de longues périodes. Ils écoutent seulement durant le processus de l'agrégation qui s'effectue par époque en réponse à une requête de la *SB*. La structure des clusters basée sur une communication à un seul saut entre les capteurs, tient pleinement de l'avantage de la diffusion du canal radio et donc aucune surconsommation n'est requise pour recevoir les messages si les nœuds capteurs sont en mode écoute (promiscuous listening mode). C'est le même principe du mécanisme " watchdog" (Maarouf, et al., 2009).

D'un côté, notre proposition atténue le fardeau du coût de surveillance pour les capteurs en les déchargeant du calcul systématique des preuves basées sur des primitives cryptographiques imposées par la vérification de l'intégrité des données. D'un autre côté, les superviseurs secondaires sont dédiés pour calculer des fonctions d'agrégations simples n'impliquant pas beaucoup de cycle CPU, telles que la moyenne, le min et le max. comme ça été rapporté dans (Wu, et al., 2006), le nombre des opérations basiques dans les fonctions min/max et moyenne est égal à 23 opérations contre 4192 opérations dans le chiffrement symétrique RC5 pour un paquet de 16 octets. Il est évident que les opérations d'agrégation sont plus simples que des opérations cryptographiques.

7.4 Comparaison des caractéristiques

Dans le tableau 4.3, nous résumons les caractéristiques (features) de notre protocole par rapport à d'autres protocoles existants dans la littérature.

La caractéristique de « type d'agrégation » indique le nœud responsable de l'agrégation : hop-by-hop signifie qu'un modèle à agrégateurs multiples est utilisé et dans lequel chaque nœud ajoute sa propre valeur à l'agrégation, alors que CH signifie que l'agrégation est effectuée par un cluster-head. Cependant, dans le protocole SIA, un modèle à un seul agrégateur est utilisé dans lequel toutes les données individuelles du réseau traversent seulement un seul nœud agrégateur avant l'arrivée à la *SB*. La caractéristique « résistance aux attaques internes » indique la résistance contre l'injection des données erronées, i.e.,

lorsqu'un attaquant manipule les mesures collectées. Nous pouvons remarquer que toutes les solutions précédemment existantes ne prennent pas en charge cette attaque. Le tableau 4.3 indique aussi si le protocole est résistant contre les agrégateurs malicieux et les pannes d'agrégateurs dans la colonne 4 et 5 respectivement. La colonne 6 indique la résistance contre le rejet des données, l'inconvénient majeur de la plupart des solutions existantes focalisant sur l'intégrité des données agrégées. La dernière colonne dénote la politique de gestion adoptée par les protocoles. Par « règle unique », nous désignons l'utilisation systématique des primitives cryptographiques même en l'absence d'attaque. Par « règle adaptative », nous désignons la réaction adaptative selon le scénario d'attaque rencontré. Dans ce dernier cas, les primitives de cryptographiques ne sont utilisées qu'en cas de besoin, i.e., lorsque des activités malicieuses ont été détectées.

	Type d'aggregation	**Résistance aux attaques internes**	**Résistance aux agrégateurs malicieux**	**Tolérance aux pannes d'agrégateurs**	**Résistance aux rejets de données**	**Politique de gestion**
SDA [Hu, 03]	Hop-by-hop	Non	Oui	Non	Non	Unique
SIA [Przydatek, 03]	Agrégateur unique	Non	Oui	Non	Non	Unique
WDA [Du, 03]	Hop-by-hop	Non	Oui	Non	Non	Unique
SDAP [Yang, 06]	Hop-by-hop	Non	Oui	Non	Non	Unique
FAIR [Emiliano,09]	Hop-by-hop	Non	Oui	Oui	Oui	Unique
RAHIM Notre solution	CH	Oui	Oui	Oui	Oui	Adaptative

Tableau 4. 3 : Protocoles d'agrégation sécurisée : comparaison de caractéristiques.

7.5 Résultats de simulation

Dans cette section, nous effectuons une étude de simulation pour démontrer l'aspect pratique et l'efficacité de notre protocole de sécurité des données agrégées. Nous évaluons les performances de notre protocole en termes de *latence*, *exactitude d'agrégation* et efficacité

énergétique. Le protocole est implémenté avec le simulateur NS2. L'extension Mannasim a été utilisée pour rajouter de nouveaux modules pour le développement et l'analyse des différentes applications des WSN dans NS2.

Nous avons utilisé l'algorithme Skipjack pour le calculer des MACs. La capacité du canal est supposée constante et égale à 10 Kbps sur le lien sans fil. Nous considérons que le canal est idéal.

Les nœuds capteurs sont déployés dans une zone de 100 mètre sur 100 mètres. Puisque notre protocole est exécuté dans chaque cluster, nous effectuons la simulation dans le cluster et nous varions le nombre de nœuds de 6 à 36 pour changer la densité du cluster. Le rayon de transmission de chaque nœud capteur est de 40 mètres. La tableau 4.4 résume les paramètres de simulation sur des capteurs de type Crossbow mica2.

La puissance de transmission (Pt_) est la puissance avec laquelle le signal est transmit. C'est cette puissance qui décide du rayon de transmission du nœud capteur. La puissance de transmission (txPower) est la puissance consommée par le transceiver pour transmettre un paquet de données. La puissance de réception (rxPower) est la puissance consommée pour recevoir un paquet de données.

Paramètre	**Valeur**
Nombre de noeuds dans le cluster	6, 16, 26 et 36
Nombre d'époques	10
Puissance de transmission (Pt_)	8.564E-4 mW
Puissance de transmission (txPower)	0.036 mW
Puissance de réception (rxPower)	0.024 mW
Energie initiale	10 J
Zone couverte	100m x 100m
Rayon de transmission	40 m

Tableau 4. 4 : Paramètres de transmission.

La simulation est exécutée en utilisant plusieurs scenarios d'attaques et 40% de nœuds compromis sont insérés dans le cluster. Dix requêtes sont initiées par la station de base. La simulation est obtenue en calculant la moyenne de toutes les exécutions.

Pour faire des comparaisons, nous avons également implémenté le protocole d'agrégation non sécurisé (TAG) et un protocole d'agrégation sécurisé classique dans lequel la violation de l'intégrité des données induit un rejet des données.

1. Latence: nous signifions par *latence*, le délai moyen entre la requête initiée par la *SB* et la délivrance des résultats d'agrégation à la *SB*. La Figure 4.5 illustre l'intérêt de l'utilisation du mécanisme de surveillance pour délivrer le résultat correct à la *SB* sans avoir recours à l'annulation du processus d'agrégation lorsqu'une fraude a été détectée. Par rapport à l'agrégation non sécurisé (TAG), la rapidité de délivrance dans notre protocole est *constante* et *très proche* de celle de TAG dans le scénario1 et le scénario2. Cependant, dans le scenario3 et le scénario4, ce délai croît relativement lorsque le nombre de nœuds augmentent dans le cluster. Ceci est expliqué par le fait que dans ces deux derniers scénarios, l'envoi des messages de plaintes allonge le délai de délivrance.

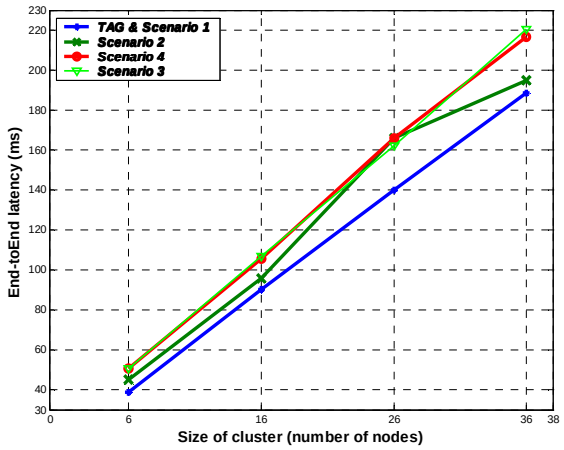


Figure 4. 5 : Délai de délivrance.

2. exactitude : dans une situation idéale, lorsqu'aucun nœud capteur n'est compromis dans le réseau, RAHIM atteint une exactitude du résultat d'agrégation de 100%. Cependant, puisque les nœuds capteurs sont déployés dans des environnements peu sûrs, et peuvent être compromis, cette exactitude est affectée. Nous définissons la métrique d'exactitude pour la fonction « moyenne », comme le rapport entre la moyenne estimée par le protocole utilisé et la moyenne réelle de toutes les mesures individuelles de chaque capteur. Une valeur d'exactitude élevée signifie que la moyenne estimée en utilisant un protocole d'agrégation spécifique est plus exacte. Une valeur d'exactitude de 1.0 représente la situation idéale.

La Figure4.6 illustre l'exactitude du protocole TAG et RAHIM après simulation dans laquelle, nous avons considéré un cluster constitué de 26 nœuds capteurs. Nous observons que cette exactitude décroît en fonction de l'augmentation des nœuds compromis dans le protocole d'agrégation non sécurisé TAG, puisque ce dernier est très sensible à un

environnement non sûr. Par contre dans tous les scénarios d'attaque, notre protocole RAHIM atteint une meilleure exactitude que celle de TAG.

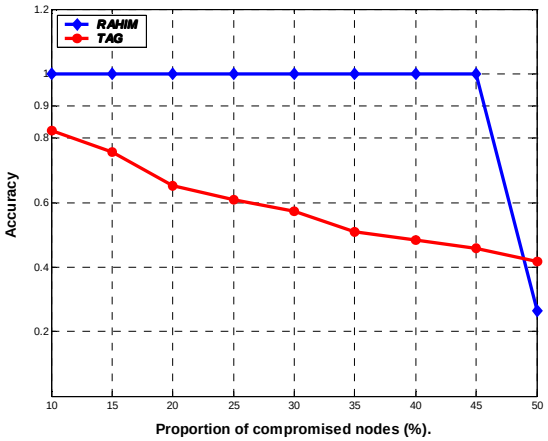
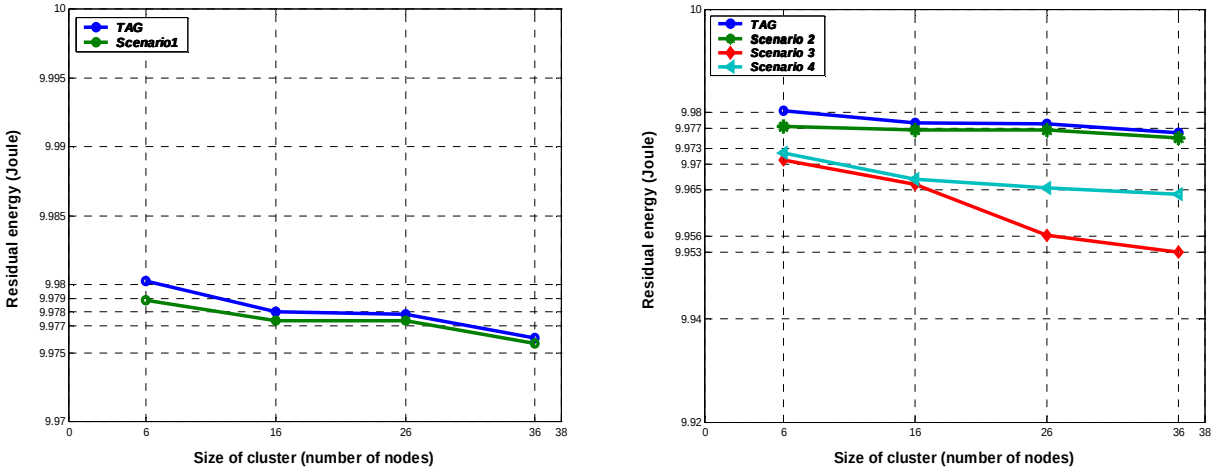


Figure 4. 6 : Comparaison d'exactitude entre TAG et RAHIM.

3. Efficacité énergétique*:* RAHIM utilise le mécanisme de surveillance pour protéger l'intégrité de l'agrégation. Par ce mécanisme, des messages d'alertes sont déclenchés lors de détection de fraude. Ceci induit une consommation d'énergie. Donc pour investiguer l'efficacité énergétique de notre protocole, nous avons d'abord étudié l'énergie résiduelle du protocole, ensuite le gain en énergie de RAHIM par rapport à un protocole de sécurité classique.

- **Energie résiduelle**: nous analysons la moyenne de *l'énergie résiduell*e en fonction de la variation du nombre de capteurs dans le cluster et ce, dans les quatre scenarios d'attaques. La Figure 4.7 (a) et (b) démontre l'effet de l'augmentation du nombre de capteurs sur la moyenne de l'énergie résiduelle dans une époque d'agrégation. Initialement, chaque nœud capteur a une énergie de 10 joules. Nous remarquons que dans la Figure 4.7 (a) que la consommation d'énergie de notre protocole est *très proche* de celle du protocole TAG dans une situation normale (sans attaque). Cependant, en cas d'attaque, notre protocole adapte sa réaction en fonction du scénario d'attaque et consomme un peu plus d'énergie qu'en situation normale (c'est le prix de la sécurité !) mais pas trop par rapport au protocole TAG. Notre protocole a donc maintenu le but principal de l'agrégation en termes d'efficacité énergétique.



(a) (b)

Figure 4. 7 : Energie résiduelle (a) situation normale ; (b) en présence d'attaque.

- **Gain en énergie**: dans notre protocole, lorsqu'un résultat d'agrégation falsifié a été envoyé, la *SB* n'annule pas le processus d'agrégation, puisqu'on lui délivre systématiquement le résultat correct (réel) de l'agrégation par le biais du message d'alerte. Dans cette métrique, nous analysons l'impact du rejet de données sur la consommation d'énergie en variant le nombre de rejet. Nous simulons un protocole de sécurité des données agrégées classique, dans lequel le processus d'agrégation est annulé et donc toutes les étapes sont ré-exécutées. La Figure 4.8 illustre clairement l'énergie dépensée avec une, deux et trois rejets de données. Alors que les Figures 4.9, 4.10 et 4.11, illustrent le gain en énergie par le protocole RAHIM par rapport à un protocole classique respectivement avec une, deux et trois rejets.

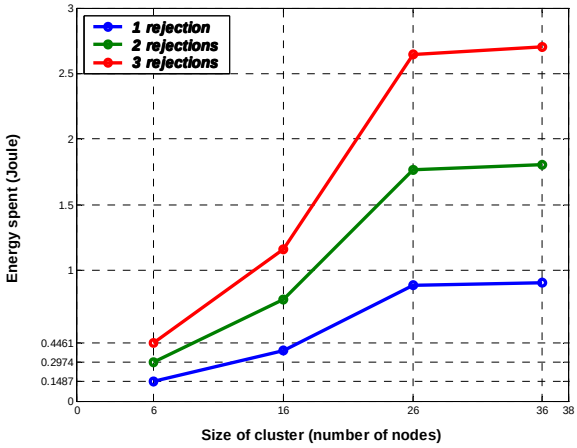


Figure 4. 8 : Energie consommée par le rejet de données.

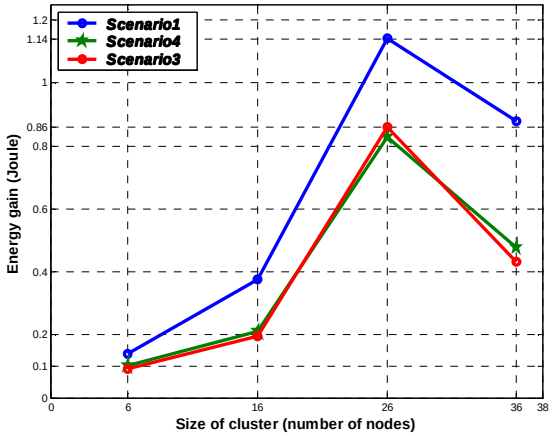


Figure 4. 9 : Gain en énergie par RAHIM (avec 1 rejet).

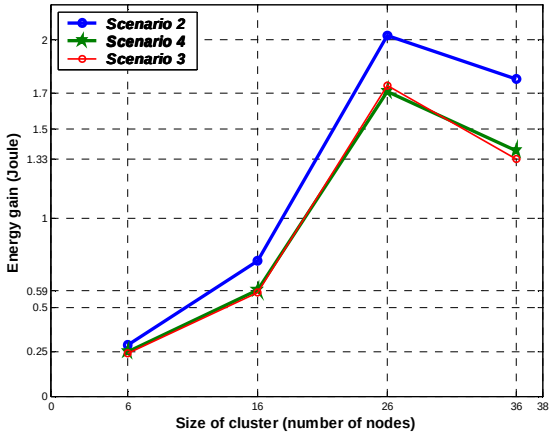


Figure 4. 10 : Gain en énergie par RAHIM (avec 2 rejets).

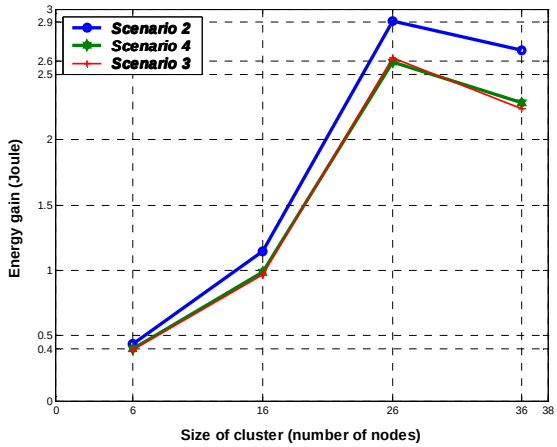


Figure 4. 11 : Gain en énergie par RAHIM (avec 3 rejets).

En conclusion, notre protocole RAHIM surpasse de manière significative les protocoles classiques de sécurité des données en termes de consommation d'énergie sous plusieurs scénarios d'attaques.

8. CONCLUSION

Dans ce chapitre, nous avons proposé un nouvel algorithme appelé « RAHIM » pour sécuriser l'agrégation des données dans les réseaux de capteurs clustérisés. RAHIM est basé sur l'application d'un niveau de surveillance *hiérarchique* et *adaptatif* qui assure l'exactitude du résultat de l'agrégat d'une manière *light*, même si tous les nœuds agrégateurs et une partie des nœuds membres d'un cluster sont compromis.

Nous avons visé avec RAHIM la création d'un système de surveillance hiérarchique de l'opération de l'agrégation et qui s'adapte à différentes situations selon le type d'attaque. Pour cela nous avons conçu un système qui ne réagit qu'en cas de besoin. En effet, contrairement aux solutions existantes, notre algorithme se base sur le mécanisme de « preuve de fraude» et non pas sur le mécanisme de « preuve d'approbation », cela veut dire que si le processus d'agrégation a été effectué normalement et sans aucune fraude, alors notre algorithme n'entreprend aucune mesure de sécurité. Par contre, si l'un des membres du cluster y compris l'agrégateur a été détecté comme malveillant (i.e qu'il a manipulé le résultat de l'agrégat ou injecter des données erronées), des mesures de sécurité adaptatives sont mises en place.

RAHIM est également un algorithme robuste contre le rejet total des données. En effet le rejet total des données est le problème principal de la majorité des solutions existantes ; un rejet d'un résultat de l'agrégation de la part de la station de base, implique l'annulation de tout le processus de l'agrégation. Toute l'énergie consommée par les nœuds capteurs pour contribuer à ce processus sera donc perdue puisqu'il faudra réinitialiser l'agrégation à partir du début. Dans notre algorithme ce rejet total est évité par l'utilisation des nœuds moniteurs qui joueront leur rôle au temps opportun pour expédier immédiatement le résultat exact de l'agrégat à la station de base sans aucune réinitialisation. Ceci a pour effet d'éviter la phase de vérification interactive, qui génère un surcoût en communication et un délai non négligeable pour la délivrance des résultats finaux.

Comme le réseau considéré est structuré sous forme de cliques (cluster), où les membres d'une clique sont tous reliés entre eux par un seul saut, la tâche de l'agrégation est

distribuée, ce qui rend l'algorithme tolérant aux pannes des nœuds agrégateurs durant une époque d'agrégation, puisque ces derniers sont la pierre angulaire du processus de l'agrégation et une panne d'un agrégateur entraîne surement la rupture partielle du service.

RAHIM s'adapte et réagit différemment à chaque situation afin d'éviter de consommer inutilement de l'énergie, et d'assurer plus de disponibilité de service que les autres algorithmes de sécurité.

Chapitre V

Proposition d'un protocole de localisation sécurisée : WFDV

Sommaire

1. INTRODUCTION

Les réseaux de capteurs sans fil (WSNs) sont particulièrement attrayants pour plusieurs applications de surveillance d'infrastructures critiques ou de zones difficiles à atteindre. Une grande majorité de ces applications utilise un déploiement aléatoire d'un grand nombre de capteurs, en raison soit de l'hostilité de la zone à surveiller, soit de son immensité. La phase de localisation est donc nécessaire non seulement au fonctionnement du réseau (routage géographique par exemple), mais également à l'exploitation des données récoltées («où» est la question qui suit immédiatement l'avènement d'un évènement dans la zone surveillée). Il est donc nécessaire de localiser, avec la meilleure précision possible, tous les nœuds du réseau. L'idéal serait d'équiper chaque capteur d'un récepteur GPS pour obtenir sa position exacte. Cependant cette solution est trop coûteuse du point de vue financier comme du point de vue énergétique. Pour réduire ce coût, d'autres approches ont été proposées qui consistent à équiper une partie des capteurs d'un module GPS, permettant de récupérer leurs coordonnées absolues, ces nœuds appelés « ancres » ou « beacons » émettent leur position autour d'eux, qui servira ensuite de repères aux autres nœuds « ordinaires » (ceux n'étant pas équipés de module GPS) qui vont à travers des techniques de coopération entre eux, estimer leur position respective.

La localisation dans les réseaux de capteurs sans fil a attiré l'attention de plusieurs chercheurs ces dernières années, et plusieurs approches dites range-based et range-free ont été proposées (Zhao, et al., 2005) (Bahl, et al., 2000). Cependant les réseaux de capteurs eux-mêmes sont sujets à des attaques de sécurité (Zahariadis, et al., 2010) et presque toutes les solutions de localisations précédemment proposées peuvent être trivialement manipulées par un adversaire malveillant et les résultats seront donc erronés. Il est important de noter que l'information de position fait partie de la plupart des services des réseaux de capteurs sans fil et ne doit en aucun cas être falsifiée surtout pour les applications critiques. Il est donc primordial de concevoir des solutions de localisation qui résistent aux empoisonnements de positionnement. Cette problématique, malgré les nombreux travaux de recherche qui s'y étaient attachés, reste une problématique ouverte.

Dans ce chapitre, nous présenterons notre deuxième proposition nommée WFDV :**W**ormhole-**F**ree **DV**-hop localization scheme (Labraoui, et al., 2011(c)), pour sécuriser l'algorithme de localisation DV-Hop contre l'attaque wormhole. Nous commencerons d'abord par présenter les motivations de notre proposition, ensuite nous

présenterons une étude sur la vulnérabilité de l'algorithme DV-Hop afin de bien cerner la problématique et enfin nous présenterons les détails de notre algorithme et l'analyse de sécurité ainsi que l'évaluation de ses performances.

2. MOTIVATION

A cause des ressources limitées des WSN, les solutions de localisation range-free représentent une alternative bien moins coûteuse que les solutions range-based (Bahl, et al., 2000) tout en fournissant une estimation acceptable. Cependant le problème de sécurité reste le même pour les deux approches.

Le but de notre travail n'est pas de proposer une nouvelle technique de localisation, mais d'analyser l'algorithme DV-Hop, une approche typique range-free, afin d'améliorer sa sécurité contre l'attaque wormhole. Nous avons choisi DV-Hop comme algorithme à cause de sa simplicité et de son coût bas en terme énergétique, quant au choix de l'attaque wormhole, il a été basé sur le fait que cette attaque est particulièrement sévère et difficile à détecter, car elle peut être déclenchée sans compromettre aucun nœud du réseau et sans avoir accès à aucune clé cryptographique. Il est donc clair qu'une solution qui dépend uniquement sur des techniques cryptographiques n'est pas suffisante pour s'immuniser contre l'attaque wormhole.

L'idée principale de notre approche de sécurité est de mettre en place une *contre-mesure proactiv*e à l'algorithme de base DV-Hop, nommée *prévention d'infection*, qui est constituée de deux phases pour détecter l'attaque wormhole. La première phase, utilise deux techniques peu coûteuses sur la base d'informations locales disponibles durant les opérations normales des nœuds capteurs. Quant à la deuxième phase, une technique plus avancée est appliquée uniquement si une attaque wormhole a été suspectée pour ignorer les messages délivrés par un lien wormhole. Cependant s'il n'y a aucune attaque wormhole, les nœuds capteurs n'ont pas besoin de gaspiller inutilement leurs ressources.

3. DEFINITION DU PROBLEME

Dans cette section, nous allons tenter de bien cerner le problème qui a motivé notre travail, et ce en décrivant l'algorithme DV-Hop, sa vulnérabilité contre l'attaque wormhole, et l'impact de cette attaque sur la précision de positionnement.

3.1 L'algorithme de localisation DV-Hop

Niculescu (Niculescu, et al., 2001) a proposé l'algorithme range-free DV-Hop, qui est un algorithme de localisation distribué multi-sauts. Il est facilement implémenté et ne nécessite pas de grandes ressources (Wenfeng, 2009). Il est donc adapté pour des capteurs sans fil dont les ressources sont particulièrement limitées.

DV-Hop s'exécute en trois étapes :

Dans *la première étape*, chaque nœud ancre diffuse le message beacon (message contenant ses coordonnées x_i, y_i avec un nombre de saut initialisé à zéro, h_i=0). Chaque nœud qui reçoit ce message maintient le nombre de saut minimum par ancre de tous les messages reçus, incrémente le nombre de saut de 1 et diffuse à son tour le message beacon.

Dans *la deuxième étape*, chaque nœud ancre qui reçoit le nombre de sauts le séparant d'un autre nœud ancre, va estimer la taille moyenne du saut, qui va être diffusée à tous les nœuds. La taille moyenne du saut (Hop-size) est estimée en utilisant la formule suivante :

$$HopSize_i = \frac{\sum_{j \neq i} \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}}{\sum_{j \neq i} h_{ij}} \qquad (1)$$

Où (x_i , y_i), (x_j , y_j) sont les coordonnées des nœuds ancres i et j, h_{ij} est le nombre de sauts entre l'ancre i et l'ancre j. Les nœuds ordinaires reçoivent la valeur du *Hop-Size*, et sauvegardent la première valeur reçue. En même temps, ils transmettent cette valeur à leurs voisins. A la fin de cette étape, les nœuds ordinaires calculent la distance les séparant des autres nœuds ancres sur la base de la taille du saut et le nombre de sauts.

$$d_i = hopcount_i \times HopSize_i \qquad (2)$$

Dans *la troisième étape*, après que chaque nœud ordinaire ait obtenu trois distances (ou plus) des nœuds ancres, il peut calculer sa position physique en utilisant une des méthodes telle que la trilatération (Langendoen, et al., 2003).

3.2 Illustration de l'algorithme DV-HOP

L'exemple suivant illustre le calcul de DV-Hop :

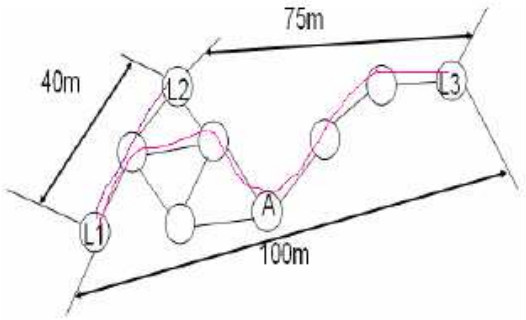


Figure 5. 1 : Algorithme DV-Hop.

Soit dans la Figure 5.1, les nœuds L1, L2 et L3 représentent des ancres. Et A le nœud voulant calculer sa position. Chaque ancre calcule la correction (Hop-Size) et la diffuse.

$$L_1 \rightarrow \frac{(100+40)}{(6+2)} = 17.5$$

De même pour L2 et L3 :

$$L_2 \rightarrow \frac{(75+40)}{(5+2)} = 16.42$$

$$L_3 \rightarrow \frac{(100+75)}{(6+5)} = 15.90$$

Un nœud ordinaire du réseau obtient une mise à jour de l'ancre la plus proche. La diffusion des corrections à travers le réseau est contrôlée : quand un nœud reçoit ou émet une mise à jour, il doit tout d'abord supprimer les anciennes. Lorsque le réseau est étendu, un champ TTL (Time To Leave) est utilisé pour localiser les mises à jour au voisinage de l'ancre.

Dans l'exemple de la Figure 5.2, le nœud A peut estimer la distance le séparant à L1, L2, et L3 à travers la mise à jour de correction reçue de L2 (i.e le Hop-Size=16.42) ainsi A calcule la distance qui le sépare des ancres L1, L2 et L3:

$$A \rightarrow L_1 = 16.42 \times 3$$

$$A \rightarrow L_2 = 16.42 \times 2$$

$$A \rightarrow L_3 = 16.42 \times 3$$

A partir de ces trois valeurs (distances entre A el les ancres), le nœud A peut déterminer sa position relative en appliquant la multilatération. En reprenant l'exemple de la Figure 5.1, A obtiendra sa position en résolvant le système suivant :

$$\begin{cases} {d^2}_{L1L2} = (x_A - x_{L1})^2 + (y_A - y_{L1})^2 \\ {d^2}_{L2L3} = (x_A - x_{L2})^2 + (y_A - y_{L2})^2 \\ {d^2}_{L1L3} = (x_A - x_{L3})^2 + (y_A - y_{L3})^2 \end{cases}$$

3.4 Impacts négatifs de l'attaque wormhole sur DV-Hop

Les attaques wormhole (Hu, et al., 2003) sont relativement faciles à monter, mais difficile à détecter et à prévenir. Dans l'attaque wormhole typique, lorsqu'un attaquant reçoit (capture) des messages dans un point du réseau, il crée un tunnel avec un autre attaquant de l'autre côté du réseau pour lui transmettre les messages reçus. Dans cette attaque, l'attaquant rejoue des messages sauvegardés, originaux, il n'a donc pas besoin de compromettre des nœuds capteurs ou l'intégrité et l'authenticité de la communication. Ce qui rend sa détection très difficile.

Dans l'algorithme DV-Hop, l'attaque wormhole peut causer deux impacts négatifs :

1) Erreur d'estimation : l'attaque wormhole peut détériorer de manière significative la procédure de localisation DV-Hop. Elle peut affecter la première étape en falsifiant le nombre de saut ; par conséquence, la seconde étape est aussi affectée et l'algorithme de localisation en entier est donc carrément faussé. Comme illustré dans la Figure 5.2, un lien wormhole entre les nœuds malicieux A1 et A2 existe constituant un tunnel. A1 reçoit le message beacon de la part du nœud B1 avec un nombre de saut égale à 1, puis le retransmet via le tunnel à A2. A2 rejoue le message beacon et le retransmet au nœud S2. Normalement le nombre de saut entre les nœuds ancres B1 et B2 est égal à 5, mais avec l'existence du lien wormhole, le nombre de saut est égal à 2, ce qui mène B2 à faire une fausse estimation de la moyenne de la taille du saut (Hop-Size). En même temps les nœuds proches de B2 se croient donc plus proches de B1 et l'estimation de leur position en utilisant la multilatération génère une grande erreur d'estimation.

2) Gaspillage d'énergie : mise à part l'erreur d'estimation causée par l'attaque wormhole, les nœuds capteurs vont transmettre plus de messages rejoués qui n'ont aucune utilité et donc consommeront plus d'énergie que dans un environnement bénin. Ce qui peut être fatal pour un capteur à ressources limitées.

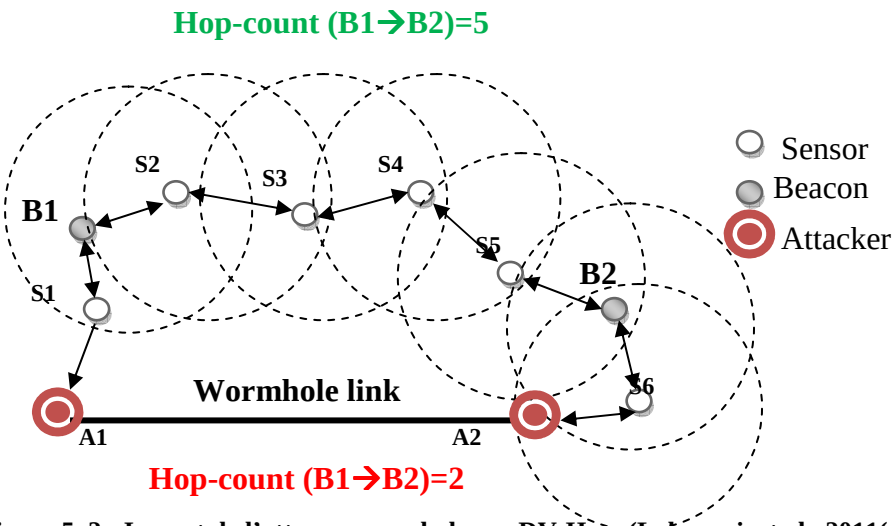


Figure 5. 2 : Impact de l'attaque wormhole sur DV-Hop (Labraoui, et al., 2011(c)).

4. MODELE DU SYSTEME

Cette section illustre notre modèle du système qui inclut les modèles de la communication, du réseau et de l'adversaire.

4.1 Modèle simplifié de l'affaiblissement de propagation

L'affaiblissement de propagation, aussi connu comme *affaiblissement de parcours* ou par son nom anglais de *path-loss* caractérise l'affaiblissement que subit une onde électromagnétique lorsqu'elle parcourt une distance (Goldsmith, 2005), (Rappaport, 2001). Cet affaiblissement est dû à la dispersion de la puissance, mais également aux obstacles rencontrés sur le chemin : édifices, montagnes, précipitations et autres bloquant le signal. Le path-loss est un terme utilisé pour quantifier la différence (en dB) entre la puissance du signal transmis, Pt, et la puissance du signal reçu Pr(d) à une distance d. cependant, généralement, dans la conception d'un système, nous utilisons le modèle simplifié du path-loss. Le modèle prévoit que le path-loss moyen $\overline{PL}(d)$, mesuré en dB, à une distance de séparation d entre l'émetteur et récepteur, sera de :

$$\overline{PL}(d) = \overline{PL}(d_0) + 10\gamma log_{10}\left(\frac{d}{d_0}\right) \qquad (3)$$

Où, $\overline{PL}(d_0)$ est le path-loss moyen en dB à une distance de référence d0 (très proche), qui dépend des caractéristiques de l'antenne et de l'atténuation moyenne du canal, et γ est l'exposant du path-loss. L'exposant du path-loss varie selon l'environnement de la

propagation radio. Lorsque γ=2 le path-loss mentionné prévoit le comportement du signal dans un environnement d'espace libre. La distance de référence d0 est choisie à une distance dans laquelle la propagation est considérée comme assez proche de l'émetteur et la diffraction est négligeable et le lien est considéré comme un espace libre.

Typiquement d0 est choisie entre 1 et 10 mètres dans un environnement *indoor*, et entre 10 et 100 mètres dans un environnement *outdoor*. Lorsque le modèle simplifié est utilisé pour rapprocher des mesures empiriques, la valeur de $\overline{PL}(d_0)$ est considérée dans un espace libre à une distance de référence d0 :

$$\overline{PL}(d_0) = 20log_{10}\left(\frac{4\pi d_0}{\lambda}\right) \qquad (4)$$

Où, $\lambda = {}^{c}/_{f}$ est la longueur d'onde du signal transmis (c est la vitesse de lumière, $3x10^8$ m/s, et *f* est la fréquence du signal transmis en Hz). Les affaiblissements du signal (path-loss) à des différentes positions géographiques et à une distance d (pour d>d0) séparant l'émetteur et le récepteur, montrent une variation normale due à l'environnement. Il suit une distribution gaussienne avec une déviation standard σ dB sur la distance qui dépend du path-loss moyen $\overline{PL}(d)$. Finalement, la puissance du signal reçu à une distance de séparation *d* basée sur le signal transmis en dB est :

$$P_r(d) = P_t - \overline{PL}(d_0) - 10\gamma log_{10}\left(\frac{d}{d_0}\right) + \sigma \qquad (5)$$

Le standard IEEE 802.15.4 (Shon, et al., 2008) est un protocole de communication destiné aux réseaux sans fil de la famille des LR WPAN (Low Rate Wireless Personal Area Network) du fait de leur faible consommation, de leur faible portée et du faible débit des dispositifs utilisant ce protocole. Récemment, la majorité des plateformes de capteurs sont équipés de Chip RF (Radio Frequence) spécifiques qui peuvent fournir les caractéristiques physiques d'IEEE 802.15.4. Le chip CC2420 est un de ces RF transceivers utilisé dans plusieurs plateformes de capteurs. Le module CC2420 RF peut mesurer la puissance du signal reçu comme un RSSI (Received Signal Strength Indicator). Sur la base de cette valeur, ayant le niveau de puissance de transmission, le récepteur peut estimer la distance qui le sépare de l'émetteur.

4.2 Modèle du réseau

Dans notre étude, nous considérons un réseau de capteurs statique constitué de plusieurs motes (nœuds capteurs) distribués uniformément dans un champ d'intérêt. Tous les nœuds du réseau sont identiques et équipés de deux radios : une radio ordinaire RF et une radio avec des capacité de saut de fréquence (FH : Frequency Hopping). Nous supposons que le réseau est constitué d'un ensemble *S* de nœuds ordinaires, ne connaissant pas leur position, et d'un ensemble *B* de nœuds ancres connaissant leur position absolue soit par GPS ou par configuration manuelle.

Nous supposons que la portée de communication R de chaque nœud capteur est la même dans tout le réseau. Nous supposons également que chaque paire de nœuds partage deux clés cryptographiques K1 et K2 par la découverte de leur voisinage.

Concernant le protocole d'accès au medium utilisé dans le réseau, nous considérons l'accès basé sur la contention, et il existe au moins une période de temps RTS/CTS/DATA/ACK que chaque paire de nœuds peut communiquer. Nous supposons que durant une exécution de RTS-CTS-Data-ACK l'environnement est stable, et la perte de messages peut être ignorée. Par conséquent, si l'émetteur réussi à envoyer un RTS au récepteur, tous ces nœuds voisins reçoivent le RTS et ne contesteraient pas pour le canal. Donc, le CTS devrait être reçu correctement par l'émetteur.

4.3 Modèle de l'adversaire

Dans le modèle de l'attaque, nous supposons que le lien wormhole est bidirectionnel entre deux ou quatre points finaux (wormhole ends). La longueur du lien wormhole est supposée plus grande que la portée de transmission *R* pour éviter les boucles sans fin de transmission de paquets causés par les attaquants. Cependant, nous ne considérons pas le cas où les attaquants peuvent intentionnellement supprimer, ou modifier les paquets reçus. Nous traitons les attaquants externes qui agissent de manière passive sur le réseau.

Pour décrire notre solution propose de manière Claire, nous donnons les définitions suivantes:

Définition 1. *Voisin local*: les voisins locaux d'un nœud sont tous les voisins qui se trouvent à un seul saut, i.e dans la portée de communication du nœud.

Définition 2. *Faux voisin* : un nœud est dit faux voisin si on peut communiquer avec lui via le lien wormhole.

Dans les sections suivantes de ce chapitre, nous utilisons les notations dans le tableau5.1:

Notation	*Description*
S_i	Nœud capteur *i*
$RTT_{(S1,S2)}$	RTT entre le nœud *S1* et le nœud *S2*
$RTT_{wormhole}$	RTT du lien sous l'attaque wormhole
$AvgRTT_{S1}$	Moyenne RTT de tous les liens entre *S1* et ses voisins
w	Temps pour passer le message via le tunnel
n	Nombre de voisins d'un nœud
N_i	un nonce
P	Délai de propagation d'un lien légitime
Pt	Puissance de transmission du signal
Pr	Puissance de réception du signal
E(K,M)	Chiffrement du message M avec la clé secrete K
HMAC(K,M)	Emprunte du message M en utilisant la fonction de hachage et la clé K

Tableau 5. 1 : Notations.

5. NOTRE PROPOSITION: WFDV

Dans cette section, nous décrivons notre protocole de localisation nommé : WFDV: « Wormhole-Free DV-Hop based localization ». WFDV permet aux capteurs de déterminer leur position et de résister contre l'attaque wormhole en même temps. Du moment que l'algorithme DV-Hop est bien connu, nous focalisons notre attention principalement sur l'amélioration de la sa robustesse contre les menaces wormholes. Le succès de l'attaque wormhole dans la première étape de DV-Hop mène à infecter sa seconde étape et donc, de fausser l'estimation de positionnement.

Le protocole WFDV inclut deux phases, prévention de l'infection et la localisation sécurisée basée sur DV-Hop. Premièrement une contre-mesure proactive nommée *prévention de l'infection* est mise en place pour prévenir la contamination wormhole via les liens wormhole. Après l'élimination des connexions (liens) illégaux, la procédure de localisation Dv-Hop peut être effectuée avec succès.

Le diagramme de notre protocole est illustré dans la figure 2.

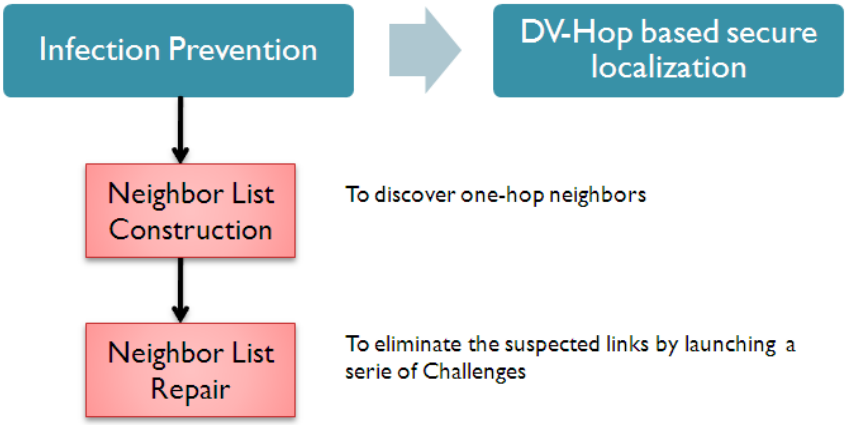


Figure 5. 3 : Le diagramme de WFDV (Labraoui, et al., 2011(c)).

5.1 La prévention de l'infection

La prévention de l'infection est effectuée avant la première étape de l'algorithme DV-Hop pour éliminer les fausses connexions (faux voisins) produites par l'attaque wormhole. Ces connexions infectent la procédure de localisation en relayant et en reportant de faux nombres de sauts (false hop-counts).

Le but de l'attaquant est de réduire la distance entre deux voisins éloignés en relayant les messages envoyés par les nœuds ancres ou par des nœuds ordinaires dans la première étape de l'algorithme DV-Hop. Il est très difficile de distinguer un voisin local d'un faux voisin, car l'attaquant relaye des messages *originaux*. Dans notre approche, chaque nœud capteurs construit la liste de ses proches voisins et tente de détecter les liens suspects faisant partie d'une attaque wormhole. Cette prévention est très utile, car le nœud peut détecter les messages rejoués et les supprimer immédiatement, en évitant de les transmettre. Par conséquent, les nœuds capteurs préservent plus d'énergie et de bande passante et évitent d'infecter d'autres nœuds. Cette mesure de défense tente de stopper l'attaque et d'éviter sa propagation dans le réseau.

Dans ce qui suit, nous présentons les deux phases constituant cette prévention :

– Phase I – Construction de la liste des voisins (CLV): dans cette étape, un nœud S1, découvre ses proches voisins (situés à un saut) en diffusant dans sa portée de communication un message de requête de voisin (NREQ) et en sauvegardant le temps de transmission de NREQ : TREQ. Le nœud qui reçoit NREQ, répond à S1 avec un message de réponse de voisin (NREP), dans lequel il embarque la puissance de transmission de son signal Pt. Le nœud demandeur S1 sauvegarde le temps de réception de chaque NREP : TREP.

Dans la phase CLV, nous utilisons deux techniques pour découvrir si le lien doit être suspecté. La première technique est basée sur la technologie RSSI se trouvant dans la majorité des plateformes des capteurs. En tenant compte des avantages des capacités de communication des réseaux de capteurs sans fil, les techniques basées sur la technologie RSSI sont peu coûteuses en énergie et possèdent des caractéristiques à moindre coût. Notre protocole WFDV utilise la technologie RSSI pour assister la construction de la liste des voisins, pour détecter les faux liens et les supprimer. La deuxième technique utilisée est basée sur la technique du RTT.

- **Technique 1 : Vérification de la propriété d'atténuation du signal**

En se basant sur le modèle simplifié du path-loss présenté dans la section 4.1, la puissance du signal reçu n'importe où plus loin que la distance de référence doit être plus petite que la puissance reçue à la distance de référence ($(\forall d > d0: Pr(d) < Pr(d0)$). Nous appelons ça, la propriété du signal d'atténuation. Dons si nous supposons que la distance entre chaque deux nœuds est plus grande que la distance de référence, aucun nœud ne peut recevoir un message avec une puissance plus grande que $Pr(d0)$.

Le but de l'attaquant est de rejouer le message avec une fausse puissance de signal. Dans le cas de l'attaquant aveugle (blind attacker) le message est rejoué en modifiant la puissance du message aléatoirement. Ceci peut mener à la violation du modèle du path-loss. Lorsque le nœud S1 reçoit le message de réponse NREP, la propriété de l'atténuation du signal est systématiquement vérifiée par S1. Si la connexion ne suit pas cette propriété, le nœud S1 supprime cette connexion et la mets en liste noire.

Algorithm 1. Neighbor list Construction

```
LocalNs=Ø;  SuspectNs=Ø;  TotalRTT=0;  n=0;
1. S1→*: NREQ: IDS1,N1;
2. Si→S: NREP: IDSi,N1,Pt;
3. for each reply from node Si Do
     if (Pt-Pr) < PL(d0)  {Signal attenuation property}
        then Si is a fake neighbor {Si is blacklisted}
        else
              SuspectNS1=SuspectNS1 ∪ Si
              TotalRTT=TotalRTT+ RTT(S1,Si)
              n=n+1
        endif
   End do
4.  If SuspectNS1 ≠ Ø     {RTT detection}
      then     AvgRTTS1=Total RTT/n
        For each node Si ∈ SuspectNS1  Do
          if  RTT(S1,Si)≥ k * AvgRTTS1
           then  Confirm the link (S1,Si) is suspicious
                 Execute Neighbor list repair.
           else LocalNs1= LocalNs1 ∪Si
          end if
        End Do
     end if
```

- **Technique 2: Détection basée sur le RTT**

Si nous supposant que l'attaquant est assez intelligent pour modifier la valeur du RSSI et rejouer le message avec une puissance *ajustée* de manière à ne pas violer la propriété d'atténuation du signal, alors la technique de vérification de la propriété du signal devient inefficace puisque l'attaquant va faire passer ses messages rejoués sans être détecté. Dans ce cas, une deuxième technique est utilisée, basée sur le délai de transit aller retour d'un lien (RTT :Round Trip Time). RTT est une mesure du temps consommée par un paquet pour aller d'un nœud à un autre à travers le réseau sans fil et revenir. (Temps d'aller retour). Le RTT peut être calculéé comme suit : $RTT=T_{REP} - T_{REQ}$.

Soit le nœud *S1* qui communique avec son voisin *S2*. Dans un environnement bénin, le RTT entre *S1* et *S2* est égal à *2p*. si le lien direct (S1,S2) est formé via un lien wormhole, alors le RTT devrait être $RTT_{wormhole}=2(p+w+p)=2(2p+w)$, où *w* est le temps pour relayer le paquet entre les deux points finaux de l'attaque wormhole.

Il est donc évident que le RTT d'un lien wormhole doit être au moins deux fois le RTT d'un lien normal, même si *w* est plus petit que p. dans la section 6, nous effectuons des simulations pour confirmer ce fait.

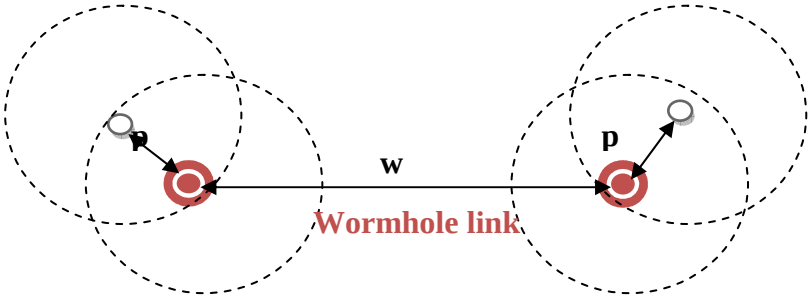


Figure 5. 4 : le RTT d'un lien normal et d'un lien wormhole (Labraoui, et al., 2011(c)).

Pour chaque message NREP, le nœud *S1* mesure le RTT avec tous les voisins de présomption. S'il trouve un nœud *S* tel que RTT (S1, S) est au moins *k* fois la moyenne de tous les RTT des voisins de *S1*, alors le lien est (*S1*,*S*) peut être un lien wormhole, et donc *S* peut être un faux voisin. La valeur de *k* est un paramètre du système qui dépend sur *n* et *w*. dans la section 6 nous expliquerons comment nous avons déterminé la valeur de *k*.

La détection basée sur le RTT est similaire à celle du protocole proposé dans (Tran, et al., 2007). Cependant, la différence entre ce protocole et notre solution, c'est que dans notre protocole nous définissons une valeur de seuil de manière déterministe alors que dans (Tran, et al., 2007) la valeur du seuil est déterminée sur la base des simulations. Le pseudo-code de la phase CLV est présenté dans l'algorithme 1.

– Phases II – Réparation de la liste des voisins: après avoir suspecté un lien wormhole dans le réseau, WFDV va lancer une séries de challenges pour être sûr que le lien wormhole est correctement identifié. Cette phase va également éviter les faux négatifs, i.e éviter de détecter les liens légaux comme des liens wormhole. Pour cela, une technique de saut de fréquences est utilisée pour confirmer l'existence d'un lien wormhole. Le pseudo-code est présenté dans l'algorithme 2.

Nous illustrons dans la Figure 5.5, l'implémentation de l'algorithme 2 en utilisant le mécanisme RTS/CTS pour les protocoles d'accès au médium (MAC) basés sur la contention tels que S-MAC, T-MAC ou B-MAC. Dans le premier message, *S1* envoie un RTC et un nonce *N1* (chiffré avec la clé K1) à *S2* en utilisant la fréquence *f1*. (*f1* est présumée connue

entre les deux nœuds). Lorsque le nœud *S2* reçoit ce message, il répond toujours dans la fréquence *f1* avec un message CTS contenant la fréquence *f2* (*f2* est choisie aléatoirement parmi l'ensemble des fréquences partagées par *S1* et *S2*), le nonce *N1* envoyé par *S1* et le nouveau nonce *N2* chiffré avec la clé K_1. Pour protéger l'intégrité du paquet, *S2* peut optionnellement calculer une signature du message en utilisant la fonction HMAC avec la clé K_2.

Algorithm2: Frequency Hopping Challenge(S1,S2)

```
1: S1 →S2: RTS, Enc(K1,N1); (frequency f1)
2: S2 →S1: CTS,Enc(K1,f2,N1,N2),HMAC(K1,f2,N1,N2); (f1)
3: S2 switches its receiver to f2 and waits for 2*RTT(S1,S2)
   time;
4: After receiving the CTS,
   S1 →S2: RTS,Enc(K1,N2),HMAC(K2,N2);(f2)
5: if S1 receives ACK from S2 in frequency f2 within duration of
   2*RTT(S1,Si) time
       then  LocalNS1=LocalNS1 ∪ S2
       else  S2  is fake neighbor  {S2 is blacklisted};
   end if
```

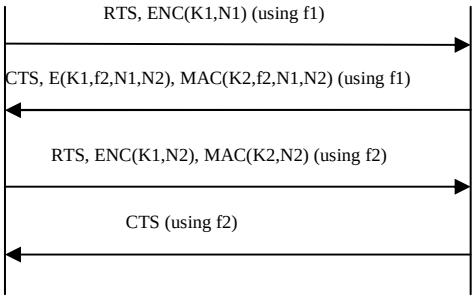


Figure 5. 5 : Le challenge basé sur le saut de fréquence (Labraoui, et al., 2011(c)).

Après avoir répondu à *S1* avec le message CTS, *S2* fait un saut de fréquence et commute son récepteur vers la fréquence *f2* et attend un paquet de *S1* sur cette fréquence. Dans ce cas, nous présumons que le paquet CTS ne soit pas perdu et qu'il arrive toujours à destination dans des conditions d'environnement stable. Plus tard, dans la section d'analyse, nous discutons cette supposition en détail. Immédiatement après avoir reçu le message CTS, *S1* commute ses récepteurs vers la fréquence *f2* et envoie à nouveau un message RTS à *S2* qui contient le nonce *N2* pour assurer l'authentification du message. Finalement, *S2* répond avec un message CTS pour terminer le challenge.

Si *S1* et *S2* sont des nœuds réellement éloignés et deviennent des voisins par un lien wormhole, alors en commutant vers la nouvelle fréquence, normalement ils ne pourront plus échanger de messages entre eux. Ceci parce que l'attaquant ne peut connaître la nouvelle fréquence et donc ne peut relayer les messages entre *S1* et *S2*. L'utilisation du nonce *N1* et *N2* est utile pour éviter les attaques de rejeu. Sans ces nonces, l'attaquant peut déclencher l'attaque comme suit : supposons que l'attaquant a capturé un message CTS qui contient la fréquence *f2* non connue par l'attaquant. Ce dernier peut sauvegarder ce message et tenter de scanner toutes les fréquences pour trouver celle que *S1* et *S2* vont utiliser. Si la fréquence a été correctement identifiée par l'attaquant, alors il va rejouer le même message pour chaque nouveau challenge entre la même paire *S1* et *S2*, ce qui peut effectivement rompre la solution de sécurité mise en place. Cette attaque n'est pas possible en utilisant les nonces qui aident à détecter les messages rejoués. Nous pouvons également améliorer la sécurité de ces messages en incluant un temps d'expiration pour chaque message.

5.2 La localisation sécurisée basée sur DV-Hop

Après avoir greffé l'étape de la prévention de l'infection, et après son exécution, chaque nœud *S1* du réseau va obtenir une liste de voisins locaux nommé $LocalN_{Si}$. Donc une fois que chaque nœud a éliminé les faux liens de sa liste de voisins, la procédure de localisation DV-Hop va s'effectuer. Dans la première et la seconde phase de DV-Hop, chaque nœud va simplement ignorer les messages reçus des nœuds n'appartenant pas à sa liste de voisins locaux. Avec cette stratégie, les impacts de l'attaque wormhole sur la localisation ainsi que sur les ressources des capteurs, seront évités avec succès. Et donc, l'algorithme de localisation DV-Hop sera sécurisé et immunisé contre l'attaque wormhole grâce à notre contre-mesure proactive.

6. ANALYSE DE SECURITE

Dans cette section nous présentons une analyse de sécurité de notre protocole de sécurité WFDV. Nous montrons que les impacts de l'attaque wormhole sur l'estimation de la position des nœuds sont prévenus de manière proactive et que la procédure de localisation DV-Hop peut être effectuée avec succès.

6.1 Analyse de la phase de construction de liste de voisins

Dans cette section, nous allons analyser la sécurité de la phase de construction de liste de voisins.

A. Violation de « la propriété d'atténuation du signal »

Considérant le scénario simple, illustré dans la figure 5.6, dans lequel l'attaquant veut fabriquer quatre faux liens, S_1-D_1, S_1-D_2, S_2-D_1 et S_2-D_2. Nous définissons la *topologie victime* comme deus ensembles de nœuds correspondants aux deux côtés de l'attaque. Chaque nœud est membre d'un seul ensemble et son path-loss est représenté sur le lien. Dans notre scénario, nous supposons que la topologie victime est {{45,70},{50,80}} qui signifie qu'il ya 2 nœuds du côté droit (gauche) de l'attaque avec ces valeurs de path-loss à savoir 50 et 80 dB(45 et 70 dB). Nous supposons également que le niveau de puissance maximal est 0dB, et le path-loss à une distance référence est de 40dB. Les nœuds M1 et M2 sont deux points de relai de l'attaque et les nœuds S_i et D_i sont victimes.

L'attaquant doit changer la puissance du signal avant de le relayer. Si on considère que le niveau de puissance que l'attaquant utilise pour relayer le message est ΔP plus la puissance reçue, le path-loss de bout en bout entre deux nœuds proches doit respecter la « propriété de l'atténuation du signal » ; i.e le path-loss de bout en bout doit être plus grand que 40 dB (celui de la distance de référence). Pour maximiser la chance de création de faux lien, l'attaquant doit minimiser le ΔP. cependant, le minimum de ΔP que l'attaquant peut utiliser pour rendre ces 4 faux liens est de 60 dB. Par conséquent, lorsqu'il relaye le message des nœuds proches il peut être détecté par un nœud proche de d'autre côté du lien wormhole car le path-loss de bout en bout entre les nœuds est plus petit que 40 dB ce qui est impossible car dans ce cas la propriété d'atténuation du signal est non respectée.

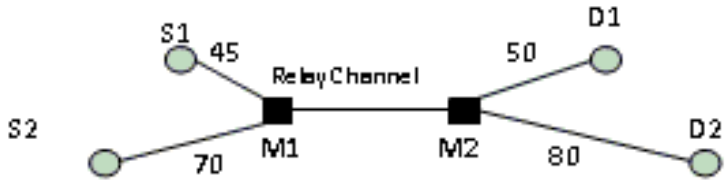


Figure 5. 6 : Un canal de relai simple.

B. Attaque de la détection basée sur le RTT

Dans l'algorithme 1 nous avons annoncé que le $RTT_{(S1,S2)}$ doit être au moins *k* fois la moyenne des RTT $AvgRTT_{S1}$ pour que le nœud S1 suspecte le lien (S_1,S_2) comme étant un lien wormhole. Nous verrons dans ce qui suit comment chaque nœud va déterminer la valeur de ce seuil *k*.

Soit *n* le nombre de voisins du nœud, nous assumons qu'au sein de ces *n* voisins il existe au plus *m* lien wormhole (m < *n*). Nous avons :

$$RTT_{S1,S2} = 2(2p + w)$$

$$AvgRTT_{S1} = \frac{(n-m)2p + 2(2p+w)m}{n} \quad (6)$$

$$Test = \frac{RTT_{(S1,S2)}}{AvgRTT_{S1}} = \frac{2(2p+w)n}{(n-m)2p + 2(2p+w)m} \geq k \quad (7)$$

Nous observons que la valeur de *Test* augmente lorsque la valeur de *w* diminue. Cependant, *w* est toujours supérieur à 0. Donc, si nous calculons la valeur du seuil *k* avec *w*=0 alors l'attaquant sera très probablement détecté. Dans ce cas, $k = \frac{2n}{n+m}$ et peut être calculé par chaque nœud capteur du réseau. Par exemple, si *n* =6 et *m*=1, alors le seuil *k* sera égal à 12/7 = 1.7.

C'est une valeur déterministe, contrairement à celle calculée dans (Tran, et al., 2007), où la valeur du seuil varie dans différents réseaux.

6.2 Analyse de la phase de réparation de la liste des voisins

L'attaquant possède deux options pour répondre au challenge: soit en supprimant le paquet RTS ou en laissant passer ce paquet vers S_i. Nous verrons que l'utilisation de ces deux options ne soit d'aucune utilité à l'attaque wormhole et que dans les deux cas, cette attaque sera détectée.

A. Suppression du paquet RTS

Dans notre solution si S_1 n'a pas reçu de message CTS dans un temps fini, il retransmet un autre paquet RTS. Dans le standard 802.15.4 chaque nœud tente r fois (typiquement r =3) avant de déclarer une erreur de transmission (Shon, et al., 2008). Si une erreur de transmission arrive, notre solution considère que le challenge a été perdu. Si un lien possède *M* défis perdus continus, notre solution déclare que le lien est malicieux.

Si le lien S_1 a envoyé un message RTS alors la probabilité qu'une collision arrive est donné par :

$$P[collision] = 1 - (1 - \tau)^{n-1} \quad (8)$$

Où τ est la probabilité de transmission à un moment *t* de chaque nœud et *n* est le nombre de nœuds voisins d'un nœud. Si S_1 n'a pas reçu de message CTS au bout d'un certain temps fini, il retransmet son message RTS. Si tous les *r* RTS messages ont cause une collision, avec les autres nœuds alors la probabilité que cela arrive est:

$$P[Losing\ r\ RTS] = [1 - (1 - \tau)^{n-1}]^r \quad (9)$$

La probabilité que *M* challenges échouent à cause du canal sans fil et non à cause d'une attaque wormhole est :

$$P[Failing\ M\ challenges] = [1 - (1 - \tau)^{n-1}]^{rM} \quad (10)$$

En utilisant $M = 6$, $r = 3$, $n = 10$ et $\tau = 0.1$ nous avons :

$$P[Failing\ M\ challenges] = 1.4 \times 10^{-4} \quad (11)$$

La probabilité que *M* challenges échouent sans l'existence d'un lien wormhole est donc négligeable. Ce qui montre que la stratégie de suppression de paquets RTS ne peut aider l'attaquant à réussir son attaque wormhole.

B. Permettre le passage de paquets RTS

La deuxième option pour l'attaquant est de permettre le passage des parquets RTS. Nous supposons que (1) qu'il est très coûteux pour l'attaquant d'écouter tous les canaux disponibles et (2) il est impossible de casser le chiffrement pour obtenir la fréquence *f2* dans un temps rapide. Par conséquent, en laissant passer les paquets RTS, l'attaquant va deviner la fréquence *f2* car le contenu du message est chiffré et son intégrité est protégée.

La probabilité de deviner correctement la bonne fréquence est de *1/N*, où *N* est le nombre de canaux. Si nous forçons chaque nœud à passer le défi δ fois, cette probabilité de deviner la bonne fréquence chaque fois est réduite de $1/N\delta$. En utilisant une valeur appropriée de δ et *N*, cette probabilité peut être très petite.

Par exemple, si N= 27 (standard 802.15.4) et δ = 2 la probabilité est inférieure à 2%. L'attaque wormhole a donc peu de chance pour réussir son attaque contre la phase de réparation de liste de voisin.

7. ANALYSE DE COUT

Dans cette section, nous évaluons le coût de calcul et le coût de stockage de notre protocole proposé.

7.1 Coût de calcul

Soit N_{NB} le nombre moyen des voisins. Soit N le nombre total des nœuds dans le réseau. Soit N_{Susp} le nombre moyen des voisins suspectés.

Le coût de calcul de la phase de construction de la liste des voisins est égal à $N_{NB}*N$. Le coût de calcul de la phase de réparation de la liste des voisins est égal à $N*N_{Susp}$. Donc le coût total de calcul de WFDV est de *O(N)* puisque N_{NB} et N_{Susp} sont des constantes, qui sont bien plus petite que *N*. par conséquent, nous constatons que le coût de calcul de WFDV est linéaire au nombre de nœuds dans le réseau de capteur.

7.2 Coût de stockage

Chaque nœud dans le réseau a besoin de stocker la liste des proches voisins (voisins à un saut) et deux paires de clés partagées avec ses voisins, ainsi que la liste des fréquences partagées. Soit S_{ID} l'identificateur du nœud capteur. Soit S_{KEY} la clé du nœud capteur. Soit S_{FREQ} la fréquence du nœud capteur, et N_{FREQ} le nombre des fréquences partagées entre deux nœuds capteurs.

Le coût de stockage pour sauvegarder la liste des voisins est estimé à $S_{ID}*N_{NB}$. Le coût de stockage pour sauvegarder les deux clés partagées avec ses voisins est estimé à $2*S_{KEY}*N_{NB}$. Le coût de stockage de la liste des fréquences partagée est estimé à $S_{FREQ}*N_{NB}*N_{FREQ}$. par conséquent, le coût de stockage de chaque nœud est égal à $\{S_{ID}*N_{NB}+ 2*S_{KEY}*N_{NB}+ S_{FREQ}*N_{NB}* N_{FREQ}\}$. si nous supposons que S_{ID} est égal à 4 octets, , S_{KEY} est à 8 octets [19], S_{FREQ} à 4 octets, N_{FREQ} à 5 et N_{NB} à 10 octets, alors le coût de stockage de chaque nœud est estimé à 400 octets.

Dans un réseau de capteur sans fil, le stockage d'un nœud capteur inclut 4ko de RAM pour les données, et 512 Ko pour la mémoire flash (Karlof, et al., 2004). Nous constatons clairement que notre protocole WFDV utilise une petite partie de la mémoire et donc peut être très adapté pour les réseaux de capteurs sans fil compte tenu de leurs ressources.

Le coût de stockage total pour tous les nœuds du réseau est donc estimé à $\{S_{ID}*N_{NB}+ 2*S_{KEY}*N_{NB}+ S_{FREQ}*N_{NB}* N_{FREQ}\}*N$, qui est de complexité $O(N)$ car S_{ID}, S_{KEY}, S_{FREQ}, N_{FREQ} et N_{NB} sont bien plus petite que N.

8. RESULTATS DE SIMULATION

Pour investiguer l'impact de l'attaque wormhole et l'habilité de WFDV pour détecter les attaques, nous avons effectué une série de simulations en utilisant le simulateur ns2. Tout d'abord, nous définissons les paramètres de nos scénarios, ensuite nous présenterons nos résultats de simulation.

8.1 Paramètres de simulation

La simulation est effectuée en utilisant le simulateur ns-2 version 2.29 avec la couche MAC 802.15.4 et les extensions sans fil CMU. Le tableau 5.2 ci-dessous résume la configuration utilisé pour ns-2.

Nombre de nœuds	2, 4, 200
Portée de communication	20-60 m
Propagation	TwoRayGround
Antenne	Omni Antenna
Couche Mac	802.15.4
Temps de simulation time	5 minutes

Tableau 5. 2 : Configuration de la simulation**.**

L'attaque wormhole a été implémentée comme une connexion filaire qui a moins de latence qu'une connexion sans fil. La position de l'attaque, i.e la position de ces deux points (wormhole ends) est complètement aléatoire dans le réseau. Pour des raisons de comparaison, nous avons également implémenté la version basic de l'algorithme DV-Hop ainsi que l'algorithme dit label-based (Wu, et al., 2010).

Pour faciliter la comparaison, nous désignons l'algorithme proposé dans (Wu, et al., 2010) par LBDV, l'algorithme de base DV-Hop par DV-Hop et notre protocole par WFDV.

8.2 Résultats de simulation

Dans cette sous-section, nous présentons les impacts pratiques de notre protocole de détection. Nous nous sommes intéressés à deux propriétés, à savoir (1) les performances de détection de l'attaque wormhole, et (2) les performances de localisation.

8.2.1 Les performances de détection de l'attaque wormhole

Pour évaluer l'exactitude (accuracy) de notre protocole de détection, quatre métriques importantes sont utilisées : *l'impact de l'attaque wormhole sur les valeurs RTT, efficacité de la détection basée sur le RTT, l'impact de la longueur du lien wormhole sur la détection* et enfin *l'impact du degré des nœuds sur la détection.*

- **Impact de l'attaque wormhole sur les valeurs RTT**

Nous avons effectué la simulation pour étudier l'impact des liens wormhole sur les valeurs RTT. Dans le premier scénario de simulation, nous avons mis en place un réseau de capteur simple constitué de deux nœuds capteurs. Nous avons mesuré la moyenne des RTT

lors de l'envoi d'un paquet Ping d'un nœud à un autre et la réception d'un accusé de réception du même paquet.

Dans le second scenario de la simulation, nous avons mis en place un réseau de capteurs constitué de quatre nœuds capteurs, dont deux nœuds sont légitimes et deux nœuds sont compromis. Nous avons simulé l'attaque wormhole lorsqu'un paquet envoyé par un nœud légitime est capturé par le premier attaquant (premier point wormhole), transmis via le lien wormhole vers le deuxième attaquant, et relayé vers le second nœud légitime. Le lien wormhole a été implémenté comme une connexion filaire et avec une longueur plus grande que la portée de communication de deux fois. Dans ce scénario, nous confirmons que le RTT d'un lien wormhole est deux fois que celui d'un lien normal.

Nous avons effectué la simulation des deux scénarios pendant 5 minutes et nous avons calculé la moyenne des résultats. La figure 5.7 démontre que le RTT d'un lien wormhole est plus grand que celui d'un lien normal, et cela malgré que le lien wormhole est réellement plus rapide qu'un lien sans fil. La moyenne des RTT lors de l'envoi de paquets via le lien wormhole et via un lien légitime est respectivement égal à 15.22 ms et 7.37 ms. Après ce résultat, nous pouvons donc confirmer que l'utilisation du délai aller retour est un indicateur efficace pour suspecter tout lien dans le réseau.

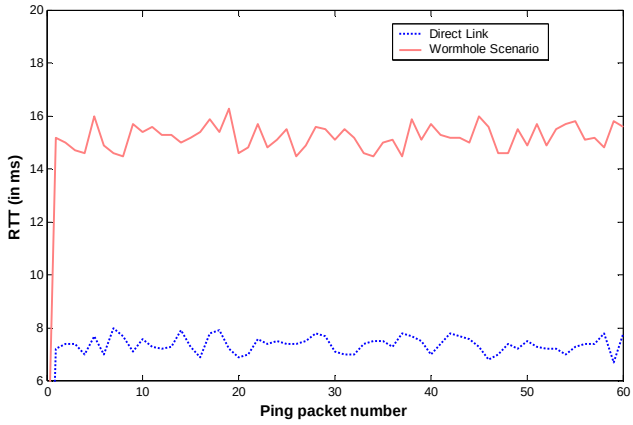


Figure 5. 7 : RTT (lien wormhole et lien normal).

- **Efficacité de la détection basée sur le RTT**

Nous avons implémenté la phase de construction de liste des voisins, pour étudier l'efficacité de la valeur seuil. Pour cela, nous avons créé une topologie de réseau constituée de 200 nœuds capteurs déployé aléatoirement dans une surface de 1000 mètre x 1000 mètres. La portée de communication est de 20 mètres pour chaque nœud. Les nœuds sont statiques et

le trafic est généré aléatoirement par le générateur de trafic de ns-2. Des connexions CBR avec 4 paquets par seconde sont crées et la taille du paquet est de 512 octets.

Dans la simulation, nous désignons aléatoirement un nœud *S1*. Nous créons le lien wormhole entre *S1* et un nœud distant *S2*. En répétant l'expérience plusieurs fois, nous pouvons sélectionner *S1* en variant le degré de ses voisins. Nous calculons par la suite le RTT des voisins de *S1* et nous calculons le seuil *k* comme nous l'avons décrit dans la section 6.1. Nous effectuons la simulation pendant 5 minutes.

La comparaison des valeurs de simulations avec les valeurs analytique est illustrée dans la figure 5.8. Nous observons clairement que le rapport des RTT d'un lien wormhole sur la moyenne des RTT est toujours au dessous du seuil calculé et donc nous confirmons que la valeur du seuil que nous avons suggéré est efficace.

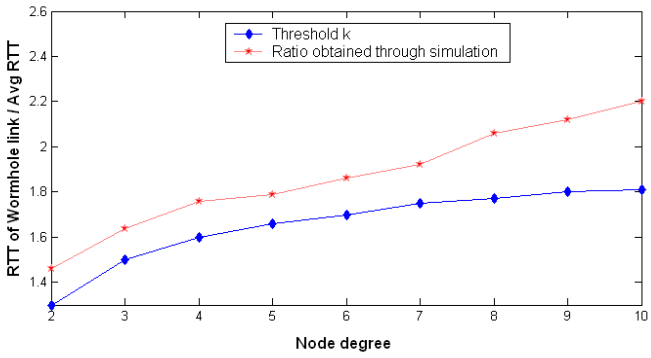


Figure 5. 8 : RTT: Théorique vs Simulation.

- **Impact de la longueur du lien wormhole sur la détection**

L'impact de la longueur du lien wormhole sur le processus de détection est illustré dans la figure 5.9 qui démontre la comparaison de performance de la probabilité de détection de l'attaque wormhole entre notre protocole et le protocole LBDV. Dans cette figure, le rapport des nœuds ancres sur les nœuds capteurs du réseau est égal à 30% et la moyenne des degrés de nœuds (nombre de voisins) est égale à 5. Les nœuds ancres sont aléatoirement placés à l'intérieur du réseau. Nous considérons différentes valeurs de longueur de l'attaque wormhole en commençant par un saut jusqu'à 6 sauts. Il est clair que notre protocole obtient de bonnes performances avec des probabilités qui dépassent 98% pour chaque lien wormhole dont la longueur est supérieure à 2 sauts. D'un autre côté, le processus de détection dans le protocole LBDV (Wu, et al., 2006) peut atteindre un taux de détection de 96% pour tout lien wormhole dont la longueur est supérieure à 2 sauts, et un taux qui descend légèrement lorsque

la longueur du lien wormhole augmente. Nous pouvons noter que pour un lien wormhole dont la longueur est inférieure à 2 sauts, la probabilité de détection de notre protocole est relativement basse par rapport au protocole LBDV. Ceci est expliqué par le fait que le temps pour faire passer le paquet via le tunnel wormhole est presque nul lorsque les deux extrémités des points de l'attaques sont très proches. Par conséquent, certaines attaques wormhole peuvent passées sans être détectées. Cependant, nous pouvons noter que ceci n'est pas un problème car généralement l'attaque wormhole a une longueur plus grande que la portée de communication pour maximiser la distorsion de positionnement dans le cas d'un algorithme de localisation (autrement l'impact de l'attaque est minime). Nous pouvons donc déclarer que notre protocole WFDV peut détecter l'attaque wormhole avec une grande probabilité et surpasse le protocole LBDV.

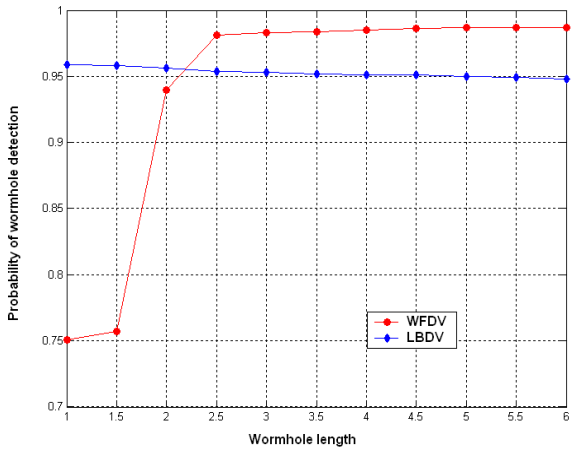


Figure 5. 9 : Probabilité de détection de l'attaque wormhole Vs longueur de l'attaque .

- **Impact du degré des nœuds sur la détection.**

Pour étudier l'impact du degré des nœuds sur la performance de la détection de l'attaque wormhole, nous avons simulé des réseaux avec divers degrés moyens de nœuds. Le degré moyen d'un nœud est changé en modifiant la portée de transmission des nœuds entre 20 et 60 mètres. Evidemment, plus la portée est grande, plus le nombre des nœuds atteints par un nœud donné est grand et donc plus le degré moyen de ce nœud est grand.

L'impact du degré des nœuds sur la détection de l'attaque wormhole est illustré dans la Figure 5.10. Dans cette figure, le rapport des nœuds ancres sur les nœuds capteurs du réseau est fixé à 30% et la longueur de l'attaque wormhole est fixée à 4 sauts. Nous observons que le processus de détection de notre protocole peut détecter l'attaque wormhole avec succès (un taux élevé) même dans les réseaux dont le degré des nœuds est bas. Un degré moyen de

nœuds de 3 et 3.5 est suffisant pour achever un taux de détection de 98%. Avec un degré moyen de nœud moins de 3, la connectivité du réseau peut devenir un problème (issue). Nous pouvons donc conclure que notre protocole WFDV peut sécuriser le réseau de manière efficace contre l'attaque wormhole.

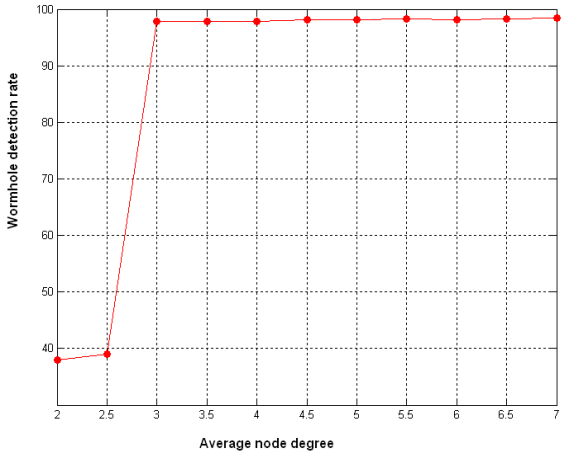


Figure 5. 10 : Taux de détection Vs degré moyen des nœuds.

8.2.2 Performance de localisation

Dans cette sous-section, nous étudions les performances de localisation en termes *d'erreur de localisation* et *d'efficacité énergétique* pour démontrer l'efficacité de notre protocole.

- **Erreur de localisation**

Un des facteurs d'évaluation le plus important dans la technologie de localisation est l'exactitude de la localisation, qui se réfère au degré de précision de l'information calculée par un nœud capteur ordinaire obtenue par l'algorithme de localisation ou le système. Dans les WSN, l'erreur de localisation est généralement utilisée comme une description quantitative de l'exactitude de la localisation.

Pour mesurer l'ampleur de l'augmentation de l'erreur de localisation en présence d'attaque, nous mesurons l'erreur de localisation avec et sans l'existence de l'attaque wormhole dans la version basic de l'algorithme DV-Hop, dans l'algorithme LBDV (Wu, et al., 2010) ainsi que dans notre protocole WFDV.

Nous avons simulé ces trois algorithmes sous deux configuration de l'attaque: (1) avec un seul lien wormhole avec deux points finaux (deux faux liens) sans se soucier du nombre

total de nœuds capteurs dans le réseau, et (2) un seul lien wormhole avec quatre points finaux (quatre faux liens). Dans ce cas de figure, l'erreur de localisation est calculée comme la moyenne de la différence absolue entre la position réelle et la position estimée par chaque nœud du réseau.

Supposons que le réseau est constitué de *N* nœuds, que la position réelle de chaque nœud est $\{(x,y)_a | a \in N\}$, et que la position estimée par chaque nœud est $\{(\hat{x},\hat{y})_a | a \in N\}$; alors l'erreur moyenne de localisation est :

$$LocError = \frac{1}{N}\sum_{a\in N}\sqrt{(x-\hat{x})^2+(y-\hat{y})^2} \quad (12)$$

Si *R* représente la portée de communication alors l'erreur de localisation normalisée est donnée par :

$$NLocError = \frac{LocError}{R} \quad (13)$$

Nous savons, qu'avec une portée de communication *R* identique, une basse erreur signifie de meilleures performances de l'algorithme de localisation.

La figure 5.11 et 5.12 illustrent la relation entre *NLocError* et le rapport (ou le taux) des nœuds ancres lorsque le nombre de faux liens dans le lien wormhole est égal à 2 et 4 respectivement. La courbe avec le label « DV-Hop » indique l'erreur de localisation pour la version basic de DV-Hop dans un environnement bénin (sans attaque). Cette courbe est donc utilisée comme référence lorsqu'une attaque wormhole existe. La courbe avec le label « 2ends/4ends » indique l'erreur de localisation pour l'algorithme DV-Hop sous l'attaque wormhole. Nous pouvons remarquer que lors d'une attaque wormhole, l'erreur de localisation de DV-Hop augmente rigoureusement spécialement lorsqu'un attaquant augmente le nombre de faux liens, ce qui démontre clairement les impacts négatifs de l'attaque wormhole sur l'algorithme de localisation DV-Hop.

Cependant pour le protocole proposé, représenté par la courbe avec le label « WFDV », l'erreur de localisation est très proche de celle de DV-Hop sans attaque, et ce dans les deux configurations de l'attaque wormhole, i.e, avec deux faux liens et quatre faux liens. Nous pouvons constater que les performances de WFDV sont tout à fait stables et insensibles au nombre de faux liens. L'exactitude de notre protocole spécialement dans la Figure 5.11 peut être expliquée en utilisant la technologie RSSI qui a sensiblement aidé

l'algorithme de localisation DV-Hop à supprimer les faux liens. Le but de l'attaquant est de rejouer les messages avec une puissance de signal falsifiée pour maximiser la chance de création de faux liens dans les deux côtés du lien wormhole et donc de maximiser la distorsion de la localisation. Cependant, en utilisant la vérification de la propriété de l'atténuation du signal, la première technique dans la phase de construction de la liste des voisins, certains faux liens peuvent être détectés et supprimés immédiatement.

D'un autre côté, l'algorithme Label-based (Wu, et al., 2010) représenté par la courbe « LBDV », l'erreur de localisation est graduellement proche de celle de celle du DV-Hop sans attaque pour la configuration de deux points finaux (2 ends wormhole), et l'impact de l'attaque n'est pas très grand. Par contre pour la configuration de quatre points finaux (4 ends wormhole), l'algorithme LBDV est complètement neutralisé alors que WFDV peut conquérir les impacts négatifs de l'attaque wormhole.

Nous pouvons conclure que notre protocole achève de très bonne performance de localisation et surpasse de loin l'algorithme label-based.

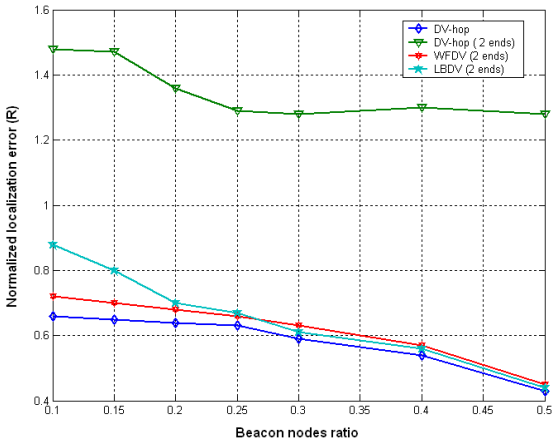


Figure 5. 11 : Erreur de localisation (wormhole avec 2 faux links).

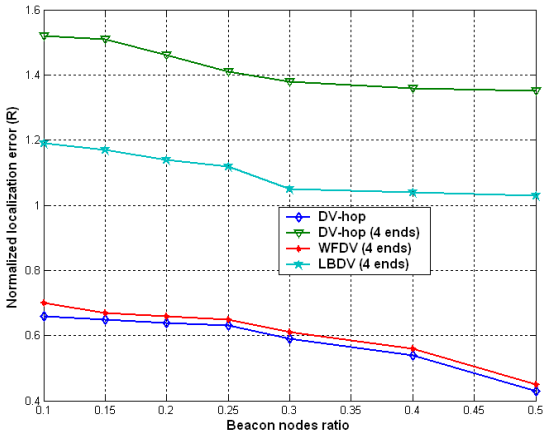


Figure 5. 12 : Erreur de localisation (wormhole avec 4 faux links).

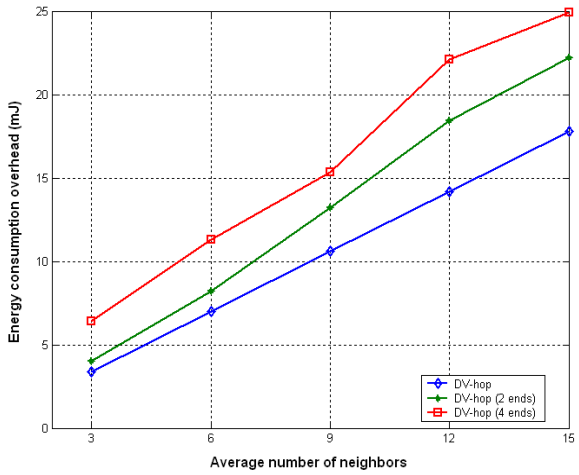
- **Efficacité énergétique**

Pour investiguer l'efficacité énergétique de notre protocole, nous étudions tout d'abord l'impact de l'attaque wormhole sur l'énergie dans l'algorithme DV-Hop. Ensuite, nous étudions la consommation de WFDV pour démontrer l'avantage de notre solution dans la conservation d'énergie.

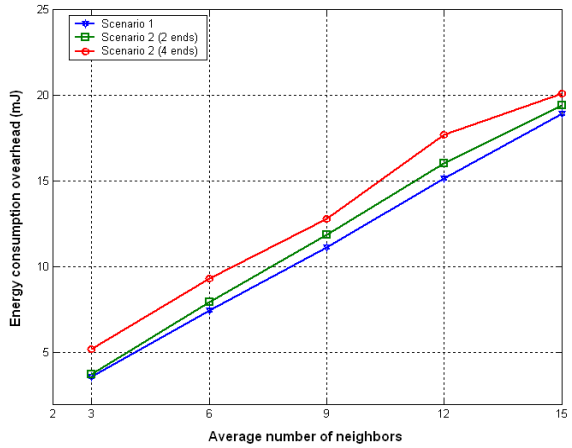
La Figure 5.13(a) illustre la consommation moyenne en énergie pour chaque nœud dans l'algorithme de localisation DV-Hop sans attaque et dans l'algorithme DV-Hop avec deux-points finaux et quatre points finaux de l'attaque wormhole. Par rapport à DV-Hop dans un environnement bénin, l'attaque wormhole cause un gaspillage d'énergie de 15% avec 2-fins et un gaspillage de 36% avec 4-fins.

La figure 5.13(b) illustre la consommation d'énergie de chaque nœud pour la phase de prévention de l'infection selon le nombre moyen de voisins dans le réseau. Nous avons simulé la phase de prévention d'infection sous deux scénarios d'attaque :

- ✓ **Scénario 1** : représenté par un attaquant aveugle (blind attaquer) qui rejoue le message avec une puissance de signal inappropriée. Dans ce cas, il peut être dans la plupart du temps détecté seulement par la première technique de la phase de construction de la liste des voisins.
- ✓ **Scenario 2**: représenté par un attaquant intelligent (smart attaquer) qui rejoue les messages avec une puissance ajustée de telle sorte à ce qu'il ne soit pas détecté par la première technique. Par contre il peut être détecté par la deuxième technique de la phase de construction de la liste des voisins et par la phase de réparation de la liste des voisins.



(a)



(b)

Figure 5. 13: Energie moyenne consommée pae chaque noeud.

Comme c'est illustré dans la figure 5.13(b), la consommation moyenne d'énergie de chaque nœud capteur durant la phase de prévention d'infection est au dessous de 25 mJ. L'overhead additionnel dans DV-Hop causé par les mesures de sécurité dans le protocole WFDV n'est pas très élevé par rapport à celui causé par l'attaque wormhole, et achève un gain en énergie de 10% dans le scénario 1, et un gain de 5% et 15% dans le scénario 2 avec les deux configurations (2 ends et 4 ends respectivement).

En résumé, nous pouvons confirmer que notre protocole WFDV peut conquérir les impacts négatifs de l'attaque wormhole en termes de consommation d'énergie sous les deux scénarios d'attaque.

9. CONCLUSION

L'attaque wormhole est considérée comme une attaque sévère qui est effectuée pour altérer le bon fonctionnement des réseaux sans fil. Elle peut être facilement déclenchée même dans les réseaux assurant la confidentialité et l'authentification. La détection d'une telle attaque représente un challenge.

Nous avons présenté dans ce chapitre le protocole WFDV, un protocole efficace pour détecter et prévenir de manière proactive l'attaque wormhole dans l'algorithme de localisation DV-Hop. La solution proposée est facilement déployée car elle ne nécessite aucune synchronisation ni un matériel spécial.

L'idée principale de WFDV est de mettre en place une *contre-mesure proactiv*e à l'algorithme de base DV-Hop, nommée *prévention d'infection*, qui est constituée de deux phases pour détecter l'attaque wormhole. La première phase, utilise deux techniques peu coûteuses sur la base d'informations locales disponibles durant les opérations normales des nœuds capteurs. Quant à la deuxième phase, une technique plus avancée est appliquée uniquement si une attaque wormhole a été suspectée pour ignorer les messages délivrés par un lien wormhole. Cependant s'il n'y a aucune attaque wormhole, les nœuds capteurs n'ont pas besoin de gaspiller inutilement leurs ressources.

Les résultats de simulation ont démontré l'habilité de WFDV à détecter les attaques wormhole, tout en maintenant une bonne performance de localisation et une efficacité énergétique satisfaisante.

CONCLUSION GENERALE

A la fin d'une longue discussion, nous arrivâmes à conclure qu'au fond il n'y rien de plus particulier qu'une idée générale.

Jules Renard, Journal.

Le danger qui menace les chercheurs aujourd'hui serait de conclure qu'il n'y a plus rien à découvrir.

Pierre Joliot, La recherche passionnément.

Depuis quelques années, les avancées technologiques en termes de miniaturisation des machines et des supports de communication y afférant ont rendu envisageable le déploiement et l'exploitation de milliers de capteurs, organisés en réseau ad hoc. D'ailleurs, selon le MIT, les réseaux de capteurs ont été identifiés comme l'une des dix technologies clefs de l'avenir et ce en raison de l'incroyable potentiel applicatif qu'elle renferme. Cependant, en raison de la jeunesse de cette technologie, le domaine de réseaux de capteurs soulève d'importantes problématiques de recherche en termes d'exploitation des données récoltées, de localisation et de sécurité. Les travaux présentés dans cette thèse s'inscrivent dans ce cadre là.

▪ Synthèse

Dans la première partie de cette thèse consacrée à la revue de littérature, nous avons débuté par une étude générale sur la sécurité dans les réseaux de capteurs, en mettant en relief les différents concepts et techniques mis en œuvre. Cette étude a mis en lumière les différentes facettes des réseaux de capteurs ainsi que leurs caractéristiques intrinsèques (limitation en énergie, topologie dynamique, mobilité, etc.) et leurs vulnérabilités aux différentes attaques. Nous avons également accordé une part importante à la dimension applicative (médicale, domotique, militaire, etc.) des réseaux de capteurs en raison du lien étroit qui les lie. En effet, avec la grande diversité des applications liées à ce type de réseaux, il est très difficile de trouver des solutions (protocoles, algorithmes, etc.) génériques dont les spécifications sont totalement indépendantes des applications.

Si les perspectives d'utilisation des réseaux de capteurs sont claires et attrayantes, les problématiques qu'engendrent ces réseaux n'en sont pas moins nombreuses. A priori, ils ne dépendent d'aucune infrastructure et les capteurs n'ont aucune information relative au réseau auquel ils appartiennent. De plus, étant construits de façon ad hoc, ces réseaux doivent être auto-organisés. Dans ce manuscrit, nous avons traité deux problématiques importantes dans les réseaux de capteurs : la sécurité des données agrégées et la sécurité de la localisation. Nous avons effectué un état de l'art détaillé afin de déceler les manques et proposer par la suite dans la deuxième partie de cette thèse des solutions originales pour les résoudre.

Dans la deuxième partie de cette thèse consacrée aux contributions, nous avons décrit deux propositions : l'une dans les données agrégées sécurisées et l'autre dans la localisation sécurisée.

Dans la sécurité des données agrégées, nous avons proposé un nouvel algorithme nommé RAHIM : Adaptive approach based on HIerarchical Monitoring, pour résoudre le problème du coût excessif et du rejet total des données des solutions déjà existantes et améliorer la fiabilité et la disponibilité des données agrégées dans les réseaux de capteurs clustérisés. La pierre angulaire de notre proposition est la gestion d'un nouveau mécanisme de surveillance nommé *surveillance hiérarchique*. Cette surveillance est effectuée au sein de chaque cluster et permet de vérifier l'intégrité et l'exactitude des résultats d'agrégation selon deux niveaux de surveillance, mais seulement en cas de besoin, i.e. uniquement lorsqu'une fraude a été détectée. Ce système permet à la station de base de recevoir le résultat correct même en présences de nœuds compromis. Contrairement aux solutions existantes, qui n'ont qu'une seule règle de gestion de sécurité, notre proposition possède plusieurs règles de gestion et adapte sa réaction en fonction du scénario d'attaque. L'exactitude de l'agrégation ainsi que l'efficacité énergétique ont été les buts principaux pour la conception de notre algorithme.

Dans la sécurité de la localisation, nous avons proposé WFDV : Wormhole- Free DV-hop based localization, un protocole pour sécuriser l'algorithme de localisation DV-Hop contre l'attaque wormhole. Notre approche de sécurité consiste à mettre en place une *contre-mesure proactiv*e à l'algorithme de base DV-Hop, nommée *prévention d'infection*, qui est constituée de deux phases pour détecter l'attaque wormhole. La première phase, utilise deux techniques peu coûteuses sur la base d'informations locales disponibles durant les opérations normales des nœuds capteurs. Quant à la deuxième phase, une technique plus avancée est appliquée uniquement si une attaque wormhole a été suspectée pour ignorer les messages

délivrés par un lien wormhole. Cependant s'il n'y a aucune attaque wormhole, les nœuds capteurs n'ont pas besoin de gaspiller inutilement leurs ressources.

- **Perspectives**

Les réseaux de capteurs constituent un axe de recherche très fertile et ont de nombreuses perspectives d'application dans des domaines très variés : domotique, surveillance industrielle et environnementale, etc. Il reste encore de nombreux problèmes à résoudre dans ce domaine afin de pouvoir les utiliser dans les conditions réelles. En outre, chaque application a ses propres contraintes. De ce fait, la conception d'un réseau de capteurs est une tâche très difficile parce qu'elle devra combiner les contraintes propres aux systèmes distribués et aux systèmes embarqués. Pour cette raison, les perspectives ouvertes par ces travaux sont nombreuses et variées.

Les travaux de recherche dans le domaine de la sécurité des réseaux de capteurs ont, jusqu'à présent, principalement porté sur la sécurité des réseaux statiques, c'est à dire non mobiles. En effet la communauté scientifique n'a pas donné trop d'intérêt à la mobilité des nœuds, et pratiquement toutes les contributions concernent uniquement un réseau statique où les nœuds sont supposé immobiles. Mais qu'on est-il si les nœuds étaient mobiles comme par exemple dans un océan ? Est-ce que la mobilité rendra la sécurité plus accrue ou au contraire contribuera à son amélioration ?

Un autre point essentiel, est le passage à l'échelle, pratiquement toutes les solutions de sécurité proposées sont praticables uniquement pour des réseaux dont la taille ne dépasse pas 1000 capteurs. Cela est sûrement du au surcoût induit par les primitives de sécurité. Des approches telles que le codage et la compression des données pourraient certainement contribuer à l'amélioration et à l'élaboration de solutions de sécurité pour des réseaux à large échelle.

Enfin, il est essentiel, de prêter plus d'attention pour l'obtention de protocole de sécurité plus robustes qui assurent en même temps confidentialité, intégrité, authentification et disponibilité. Les solutions bio-inspired pourraient être une voie très prometteuse, comme par exemple s'inspirer des colonies de fourmis ou d'abeilles afin de doter les réseaux de capteurs de plus d'intelligence dans leur gestion de collecte d'information. L'avenir est sûrement dans ce qu'on peut appeler SMART SENSORS !

LISTE DES PUBLICATIONS

Cette thèse à donné lieu à plusieurs publications dans des journaux scientifiques et à des articles de conférences. En voici la liste :

1- Journaux avec comité de lecture et indexés par Thomson

1- Labraoui N., Gueroui M., Aliouat M., Zia T., "Data Aggregation Security Challenge in Wireless Sensor Networks: A Survey", Ad hoc & Sensor Wireless Networks, International Journal, vol. 12, no. 3-4, pp. 295-324, 2011. **(Impact Factor= 0,302).**

2- Labraoui N., Gueroui M., Aliouat M., Petit J., "RAHIM: Robust Adaptive Approach Based on Hierarchical Monitoring Providing Trust Aggregation for Wireless Sensor Networks**",** Journal of Universal Computer Science (J.UCS), Vol. 17, Issue 11, pp. 1550-1571, 2011. **(Impact Factor= 0,572).**

3- Labraoui N., Gueroui M., Aliouat M., "Secure DV-Hop Localization Scheme against Wormhole Attacks in Wireless Sensor Networks", European Transaction on Telecommunications, doi: 10.1002/ett.1532. Published online by Wiley. 2011. **(Impact Factor= 0,448)**.

2- Conférences internationales avec comité de lecture

1- Labraoui N., Gueroui M., "Secure Range-free Localization Scheme in Wireless Sensor Networks", 10th IEEE International Symposium on Programming and Systems (ISPS'2011), 2011. Alger.

2- Labraoui N., Gueroui M., Aliouat M., "Proactive defense-based Secure Localization Scheme in Wireless Sensor Networks"**,** Digital Information and Communication Technology and Its Applications (DICTAP), Dijon, France. 2011.

3- Labraoui N., Gueroui M., Aliouat M., Petit J., '' Adaptive Security Level for Data Aggregation in Wireless Sensor Networks", 5th IEEE International Symposium on Wireless Pervasive Computing, ISWPC 2010, pp. 325 – 330, Modène, Italie.

4- Labraoui N. "Taxonomy of Secure In-Network Data Aggregation Schemes in Wireless Sensor Networks". Colloque International NTICRI 2009, Oran.

5- Labraoui N., Gueroui M., Tanveer, Z. "Data Aggregation Security Challenges in Wireless Sensor Networks", ISPS 2009. Alger.

6- Labraoui N., Gueroui M., Tanveer Z. “Secure In-Network Data Aggregation in Wireless Sensor Networks: Issues and Solutions”, COSI'2009, Annaba.

3- Chapitre dans un livre

Proactive Defense-Based Secure Localization Scheme in Wireless Sensor Networks

Nabila Labraoui, Mourad Gueroui and Makhlouf Aliouat

Communications in Computer and Information Science, 1, Volume 166, Digital Information and Communication Technology and Its Applications, Part 5, Pages 603-618. Edited by Springer-Verlag Berlin Heidelberg.

BIBLIOGRAPHIE

1. **Akyildiz Ian F [et al.],** A Survey on Sensor Networks, Published online in Wiley Online Library, 2002. pp. 102–114.
2. **Alzaid H, Foo E and Gonzalez J M,** secure data aggregation in wireless sensor networks: a survey, 6th Australasian Information Security Conference (AISC), Australia, L. Brankovic and M.Miller, editors, 2008, Vol. 81, pp. 93–105.
3. **Alzaid H, Foo E and Nieto J G,** RSDA: Reputation-based Secure Data Aggregation in Wireless Sensor Networks, 9th International Conference on Parallel and Distributed Computing, Applications and Technologies, 2008, pp. 419–424.
4. **Anderson R, Chan H and Perrig A,** Key Infection: Smart Trust for Smart Dust, IEEE International Conference, 2004.
5. **Audun J,** A logic for uncertain probabilities, International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 2001, Vol. 9(3), pp. 279–311.
6. **Bagaa M [et al.],** SEDAN: Secure and Efficient Protocol for Data Aggregation in Wireless Sensor Networks, 32nd IEEE Conference on Local Computer Networks, 2007, pp. 1053-1060.
7. **Bahl P and Padmanabhan V N,** RADAR:An In-building RFbased User Location and Tracking System, IEEE INFOCOM, 2000.
8. **Bahl P and Padmanabhan V N,** RADAR: An In-Building RF-Based User Location and Tracking System, Proceeding of the IEEE INFOCOM'00, 2000.
9. **Bonivento Alvise, Carloni Luca P and Sangiovanni A,** Platform-based design of wireless sensor networks for industrial applications, Conference on Design, automation and test in Europe, Belgium, 2006, pp. 1103-1107.
10. **Boukerche A [et al.],** Secure Localization Algorithms for Wireless Sensor Networks, IEEE Communications Magazine, 2008, Vol. 46(4), pp. 96 - 101.
11. **Boyle D and Newe T,** Securing Wireless Sensor Networks: Security Architectures, Journal of Networks, 2008, VOL. 3(1).
12. **Bulusu N, Heidemann J and Estrin D,** GPS-less low-cost outdoor localization for very small devices, Personal Communications, IEEE, 2000, Vol. 7(5), pp. 28-34.
13. **Çam H, Muthuavinashiappan D and Nair P,** ESPDA: Energy Efficient and Secure Pattern-Based Data Aggregation for Wireless Sensor Networks, Proceeding IEEE Sensors, Toronto, 2003, pp. 732–36.
14. **Çam H, Muthuavinashiappan D and Nair P,** Security Protocol for Wireless Sensor Networks, IEEE Vehicular Technology Conference, Orlando, 2003, pp. 2981–2984.
15. **Can H, Perrig A and Song D,** Secure Hierarchical In-network Aggregation in Sensor Networks ,13th ACM Conference on Computer and Communications Security, 2006, pp. 278–287.

16. **Capkun S and Hubaux J P,** Secure Positioning of Wireless Devices with Application to Sensor Networks , INFOCOM ’05, Miami, 2005.
17. **Castelluccia C, Mykletun E and Tsudik G,** Efficient Aggregation of Encrypted Data Wireless Sensor Network ,Proceeding ACM/IEEE Mobiquitous, San Diego, 2005.
18. **Castellucia C,** Securing very dynamic groups and data aggregation in wireless sensor networks, IEEE International Conference on Mobile Adhoc and Sensor Systems, (MASS), Piza, 2007, pp. 1-9.
19. **Chan H, Perrig A and Song D,** Secure hierarchical in-network aggregation in sensor networks, 13th ACM conference on Computer and communications security, 2006, pp. 278–287.
20. **Cheung Coleri S and Varaiya P,** Sensor networks for monitoring trafic, 42nd Allerton Conference on Communication, Control and Computing, 2004.
21. **Claveirole T [et al.],** Securing wireless sensor networks against aggregator compromises, IEEE Communications Magazine, 2008.
22. **Costa J A, Patwari N and Hero O A,** Distributed multidimensional scaling with adaptive weighting for node localization in sensor networks, IEEE/ACM Trans. Sensor Networks, 2006.
23. **Doherty L, Pister K and El Ghaoui L,** Convex position estimation in wireless sensor networks, IEEE Conference on Computer (INFOCOM), 2001.
24. **Domingo F J,** A Provably Secure Additive and Multiplicative Privacy Homomorphism, Lecture Notes in Computer Science, 2002, Vol. 2433, pp. 471-483.
25. **Domingo F J,** A Provably Secure Additive and Multiplicative Privacy Homomorphism, 2002.
26. **Du W [et al.],** A witness-based approach for data fusion assurance in wireless sensor networks, IEEE Global Communications Conference (GLOBECOM), 2003, pp. 1435–1439.
27. **Du W, Fang L and Ning P,** Lad: Localization Anomaly Detection for Wireless Sensor Networks, 19th IPDPS, 2005, pp. 1-14.
28. **Eastlake D and Jones P,** US Secure Hash Algorithm 1 (SHA1) // RFC 3174, September 2001.
29. **Emiliano D C, Jens M B and Dirk Westho,** FAIR: Fuzzy-based Aggregation providing In-network Resilience for real-time Wireless Sensor Networks, 2nd ACM conference on Wireless network security, Zurich, 2009, pp. 253–260.
30. **Estrin Deborah [et al.],** Connecting the physical world with pervasive networks, IEEE Pervasive Computing, 2002, VOL. 1(1), pp. 59-69.
31. **Fasolo E [et al.],** In-Network Aggregation Techniques for Wireless Sensor Networks: A Survey, IEEE Wireless communication, 2007.
32. **Ganeriwal S and Srivastava M,** Reputation-based framework for high integrity sensor networks, 2nd ACM workshop on Security of ad hoc and sensor networks, 2004, pp. 66-77.
33. **Girao J, Westhoff D and Schneider M,** CDA: Concealed Data Aggregation for Reverse Multicast Traffic wireless Sensor Networks, IEEE International Conference on Communication, Seoul, 2005.

34. **Goldsmith A,** Wireless Communications. Cambridge, University Press: New York, USA, 2005.
35. **Gortz Manuel [et al.],** Context-aware communication services : A framework for building enhanced IP telephony services, International Conference on Computer Communications and Networks (ICCCN), 2004, pp. 535–540.
36. **Gursel A, Mistry O and Sandip S,** Robust Trust Mechanisms for Monitoring Aggregator Nodes in Sensor Networks, International Workshop on Agent Technology for Sensor Networks, Estoril, 2008.
37. **Handy M, Haase M and Timmermann D,** Low Energy Adaptive Clustering Hierarchy with Deterministic Cluster-Head Selection, IEEE Mobile and Wireless Communications Networks, Stockholm, 2002.
38. **He T, Huang C and Blum B M,** Range-free localization schemes for large scale sensor networks, 9th annual international conference on Mobile computing and networking, 2003, pp. 81-95.
39. **Holger Karl and Willig Andreas,** Protocols and architectures for wireless sensor networks, John Wiley and Sons, 2005.
40. **Hu L and Evans D,** Secure aggregation for wireless networks, Workshop on Security and Assurance in Ad hoc Networks, Orlando, 2003.
41. **Hu Y, Perrig A and Johnson D,** Packet leashes: A defense against wormhole attacks in wireless ad hoc networks, INFOCOM, 2003.
42. **Hur J H [et al.],** Trust-based aggregation in wireless sensor networks, 2005.
43. **Iima Y [et al.],** An Evaluation of Overhearing-Based Data Transmission Reduction in Wireless Sensor Networks, 9th International Conference on Mobile Data Management: Systems, Services and Middleware, 2009, pp. 519-524.
44. **Isaac S.J. [et al.],** A survey of wireless sensor network applications from a power utility's distribution perspective, AFRICON 2011, 2011, pp. 1-5.
45. **Ji X and Zha H,** Sensor positioning in wireless ad-hoc sensor, IEEE INFOCOM'04, 2004.
46. **Jsang A and Ismail R,** The Beta Reputation System, 15th Bled Conference on Electronic Commerce, Slovenia, 2002, pp. 17-19.
47. **Junbeom J H [et al.],** Trust-based aggregation in wireless sensor networks, International Conference on Computing, Communications and Control Technologies, 2005.
48. **Kaliski B,** The MD2 Message-Digest Algorithm // RFC 1319, April 1992.
49. **Karlof C, Sastry N and Wagner D,** TinySec: A Link Layer Security Architecture for Wireless Sensor Networks, 2nd International Conference on Embedded Networked Sensor Systems, Baltimore, 2004, pp. 162-175.
50. **Krawczyk H, Bellare M and Canetti R,** HMAC : Keyed-Hashing for Message Authentication // RFC 2104, 1997, February 1997.
51. **Kui W [et al.],** Secure data aggregation without persistent cryptographic operations in wireless sensor networks, Ad hoc Networks, 2007, Vol. 5(1), pp. 100-111.

52. **Kumar A and Ribeiro V J,** REEF: a reliable and energy efficient framework for wireless sensor networks, 1st International Communication Systems and Networks and Workshops, 2009.
53. **Labraoui N [et al.],** Data Aggregation Security Challenge in Wireless Sensor Networks: A Survey, Ad hoc & Sensor Wireless Networks, International Journal, 2011(a), Vol. 12(3-4), pp. 295-324.
54. **Labraoui N, Gueroui M and Aliouat M,** Secure DV-Hop Localization Scheme against Wormhole Attacks in Wireless Sensor Networks, European Transaction on Telecommunications, 2011(c), doi: 10.1002/ett.1532. Published online by John Wiley &sons.
55. **Labraoui N, Gueroui M and Aliouat M, Petit, J,** RAHIM: Robust Adaptive Approach Based on Hierarchical Monitoring Providing Trust Aggregation for Wireless Sensor Networks, Journal of Universal Computer Science (J.UCS), 2011(b), Vol. 17(11), pp. 1550-1571.
56. **Langendoen K and Reijers N,** Distributed localization in wireless sensor networks : a quantitative comparison, Computer Networks, 2003, Vol. 43(4), pp. 499–518.
57. **Lazos L and Poovendran R,** Hirloc: High-Resolution Robust Localization for Wireless Sensor Networks, IEEE JSAC, 2006, pp. 233-246.
58. **Lazos L and Poovendran R,** Serloc: Secure Range-Independent Localization for Wireless Sensor Networks, Proceeding of WiSe '04, 2004, pp. 21–30.
59. **Lazos L, Poovendran R and Capkun S,** Rope: Robust Position Estimation in Wireless Sensor Networks, Proceeding of IPSN, 2005, pp. 324–331. .
60. **Li Z [et al.],** Robust Statistical Methods for Securing Wireless Localization in Sensor Networks, 4th Int'l. Symp. Info. Processing in Sensor Networks, 2005.
61. **Liu D, Ning P and Du W,** Attack-Resistant Location Estimation in Sensor Networks, 4th Int'l. Symp. Info. Processing in Sensor Networks, 2005(b).
62. **Liu D, Ning P and Du W,** Detecting Malicious Beacon Nodes for Secure Location Discovery in Wireless Sensor Networks, 25th ICDCS, 2005(a), pp. 609-619.
63. **Lorincz Konrad [et al.],** Sensor networks for emergency response : Challenges and opportunities, IEEE Pervasive Computing, 2004, VOL. 3(4), pp.16-23.
64. **Maarouf I, Baroudi U and Naseer A R,** Efficient monitoring approach for reputation system-based trust-aware routing in wireless sensor networks, IET communications, 2009, Vol. 3, pp. 846-858.
65. **Madden S, Franklin M J and Hellerstein J M,** TAG: A Tiny AGgregation Service for Ad-Hoc Sensor Networks, ACM SIGOPS Operating Systems Review, 2002, Vol. 36, pp. 131–146.
66. **Mahimkar A and Rappaport T S,** SecureDAV: A Secure Data Aggregation and Verification Protocol for Sensor Networks, Global Telecommunications Conference, 2004, pp. 2175–2179.
67. **Mainwaring Alan [et al.],** Wireless sensor networks for habitat monitoring, 1st ACM international workshop on Wireless sensor networks and applications, New York, 2002, pp. 88-97.

68. **Manjeshwar A and Agrawal D,** APTEEN: A Hybrid Protocol for Efficient Routing and a Comprehensive Information Retrieval in WSN, IEEE International Symposium on Parallel and Distributed Processing, Florida, 2002, pp. 195–202.
69. **Manjeshwar A and Agrawal D,** TEEN: A Protocol for Enhanced Efficiency in WSN, IEEE International Symposium on Parallel and Distributed Processing, San Francisco, 2001, pp. 2009–2015.
70. **Mukherjee P and Sen S,** Detecting malicious sensor nodes from learned data patterns ,Proceedings of the Workshop on Agent Technology for Networks, 2007, pp. 11–17.
71. **Niculescu D and Nath B,** Ad Hoc Positioning System (APS), IEEE GLOBECOM, 2001, pp. 2926–2931.
72. **Niculescu D and Nath B,** Ad Hoc Positioning System (APS) using AoA, IEEE INFOCOM '03, 2003.
73. **Niculescu D and Nath B,** DV Based Positioning in Ad Hoc Networks, Journal of Telecommunication Systems. - 2003.
74. **Ning P and Sun K,** How to Misuse AODV: A Case Study of Insider Attacks against Mobile Ad-hoc Routing Protocols, Ad Hoc Networks, 2005, Vol. 3(6), pp. 795-819.
75. **Onen M and Molva R,** Secure data aggregation with multiple encryption, 4th European conference on Wireless Sensor Networks, The Netherlands, 2007.
76. **Ozdemir S,** Concealed data aggregation in heterogeneous sensor networks using privacy homomorphism, IEEE International Conference on Pervasive Services, Istanbul, 2007, pp. 167-168.
77. **Ozdemir S,** Secure and Reliable Data Aggregation for Wireless Sensor Networks, Lecture Notes in Computer Science, Springer Berlin/Heidelberg, 2007, Vol. 4836, pp. 102–109.
78. **Pai H T, Deng J and Han Y S,** Time-slotted voting mechanism for fusion data assurance in wireless sensor networks under stealthy attacks, Computer communications, 2010, Vol. 33, pp. 1524-1530.
79. **Perrig A [et al.],** SPINS: Security Protocols for Sensor Networks, Wireless Networks Journal, 2002, Vol. 8 (5).
80. **Perrig A, Stankovic J and Wagner D,** Security in wireless sensor networks, Communication of the ACM, 2004.
81. **Peter S, Piotrowski K and Langndoerfer P,** On concealed data aggregation for aggregation for wireless sensor networks, IEEE Consumer Communications Conference, 2007.
82. **Priyantha N [et al.],** Anchor-free distributed localization in sensor networks, MIT Laboratory for Computer Science, Tech. Rep. 892, 2003.
83. **Priyantha N [et al.],** The cricket compass for context-aware mobile applications, 7th annual international conference on Mobile computing and networking, Italy, 2001, pp. 1-14.

84. **Przydatek B, Song D and Perrig A,** SIA : Secure information aggregation in sensor networks, 1st international conference on Embedded networked sensor systems, Los Angeles, 2003.
85. **Rabinovich P and Simon R,** Secure aggregation in sensor networks using neighbourhood watch, IEEE International Conference, Glasgow, 2007, pp. 1484–1491..
86. **Rajagopalan R and Varshney P K,** Data aggregation techniques in sensor networks: A survey, Communications Surveys & Tutorials, IEEE, 2006, Vol. 8.
87. **Rappaport T,** Wireless Communications, Principles and Practice, Prentice Hall PTR: Upper Saddle River, USA, 2001.
88. **Ren S Q and Kim D S. Park, J S,** A Secure Data Aggregation Scheme for Wireless Sensor Networks, Frontiers of high performance computing and networking Workshop, 2007, pp. 32-40.
89. **Rivest R L, Adleman L and Dertouzos M L,** On Data Banks and Privacy Homomorphisms ,Foundations of Secure Computation, New York, 1978, pp. 169–179.
90. **Rivest R,** The MD5 Message-Digest Algorithm // RFC 1321, April 1992.
91. **Roosta T, Shieh S and Sastry S,** Taxonomy of security attacks in sensor networks, 1st IEEE International Conference on System Integration and Reliability Improvements,Washington, 2006.
92. **Ruairi R M and Keane M T,** An energy-efficient, multi-agent sensor network for detecting diffuse events, 20th international joint conference on Artifical intelligence, 2007, pp. 1390–1395.
93. **Sang y. Shen, H [et al.],** Secure data aggregation in wireless sensor networks: A survey, 7th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT ’06)’, IEEE Computer, Washington, 2006, pp. 315–320.
94. **Sanli H, O, Ozdemir S and Cam H,** SRDA: Secure Reference-Based Data Aggregation Protocol for Wireless Sensor Networks, Vehicular Technology Conference, 2004, pp. 4650–4654.
95. **Sastry N, Shankar U and Wagner D,** Secure Verification of Location Claims, WiSe ’03, 2003.
96. **Savvides A, Han C C and Strivastava M B,** Dynamic fine-grained localization in ad-hoc networks of sensors, ACM SIGMOBILE, 2001.
97. **Servin A L and Kudenko D,** Multi-agent reinforcement learning for intrusion detection, Adaptive Learning Agents and Multi Agent Systems, 2007, pp. 158-170.
98. **Shi E and Perrig A,** Designing Secure Sensor Networks, Wireless Communication Magazine, 2004, Vol. 11(6), pp. 38–43..
99. **Shock,** http ://www.shock_sh.com/ site Internet de Shock_sh SA.
100. **Shon T and Choi H** Towards the Implementation of Reliable Data Transmission for 802.15.4-Based Wireless Sensor Networks ,5th international conference on Ubiquitous Intelligence and Computing. - Oslo : [s.n.], 2008.
101. **Srinivasan A, Teitelbaum J and Wu J,** DRBTS:Distributed Reputation-Based Bzacon Trust System, 2nd IEEE DASC, 2006, pp. 277-283.
102. **Sun K [et al.],** Secure Distributed Cluster Formation in Wireless Sensor Networks, 22nd Annual Computer Security Applications Conference, 2006.

103. **Sun Y L, Han Z and Liu K J R,** Attacks on Trust Evaluation in Distributed Networks, 40th Annual Conference on Information Sciences and Systems, Princeton, 2006.

104. **Tran P V [et al.],** TTM: An Efficient Mechanism to Detect Wormhole Attacks in Wireless Ad-hoc Networks, 4th IEEE Consumer Communications and Networking Conference, Las Vegas, 2007.

105. **Wagner D,** Resilient aggregation in sensor networks, Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks, Washington, 2004.

106. **Walters J P [et al.],** Wireless Sensor Networks Security: A Survey, Tech. Report, Department of computer science Wayne State University, 2005.

107. **Wander A S [et al.],** Energy Analysis of Public-Key Cryptography on Small Wireless Devices, 3rd IEEE International on Pervasive Computing and Communication, 2005.

108. **Wenbo H [et al.],** PDA: Privacy-preserving data aggregation in wireless sensor networks, 26th IEEE International Conference on Computer Communications, (INFOCOM), Anchorage, 2007, pp. 2045–2053.

109. **Wenfeng L,** Wireless sensor networks and mobile robot control, Science Press, 2009.

110. **Wood A D and Stankovic J A,** Denial of service in sensor networks, IEEE Computer, 2002, pp. 48-56.

111. **Wu J, Chen H and Lou W,** Wang Z. Label-Based DV-Hop Localization Against Wormhole Attacks in Wireless Sensor Networks, 5th IEEE International Conference on Networking, Architecture, 2010.

112. **Wu J, Wang C J and Chen S F,** Dynamic hierarchical distributed intrusion, IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology, Washington, 2006, pp. 89–93.

113. **Wu K [et al.],** Secure Data Aggregation without Persistent Cryptographic Operations in Wireless Sensor Networks, 21st IEEE International Conference on Performance Computing and Communications, 2006.

114. **Xu N [et al.],** A Wireless Sensor Network for Structural Monitoring, ACM Conference on Embedded Networked Sensor Systems, Baltimore, 2004.

115. **Yang Y [et al.],** SDAP: a secure hop-by-hop data aggregation protocol for sensor networks, ACM International Symposium on Mobile Ad Ad Hoc Networking and Computing, Florence, 2006.

116. **Yong W, Garhan A and Byrav R** A, survey of security issues in wireless sensor networks, IEEE Communications Surveys & Tutorials, 2006.

117. **Zahariadis T [et al.],** Trust management in wireless sensor networks, European Transactions on Telecommunications, 2010, Vol. 21, pp. 386–395.

118. **Zhang W, Das S and Liu Y,** A trust based framework for secure data aggregation on wireless sensor networks, 3rd Annual IEEE Communications Society, 2006, pp. 60-69.

119. **Zhao M and Servetto SD,** An Analysis of the Maximum Likelihood Estimator for Localization Problems, 2nd IEEE International Conference on Broadband Networks, Boston, 2005.

Ad Hoc & Sensor Wireless Networks, Vol. 12, pp. 295–324
Reprints available directly from the publisher
Photocopying permitted by license only

Data Aggregation Security Challenge in Wireless Sensor Networks: A Survey

NABILA LABRAOUI[1], MOURAD GUERROUI[2], MAKHLOUF ALIOUAT[3] AND TANVEER ZIA[4]

[1]*STIC University of Tlemcen, Algeria*
labraouinabila@yahoo.fr
[2]*PRISM University of Versailles, France*
mourad.guerroui@prism.uvsq.fr
[3]*University of Setif, Algeria*
aliouat_m@yahoo.fr
[4]*Charles Sturt University, Australia*
tanzir@it.usyd.edu.au

Received: February 27, 2010. Accepted: October 14, 2010.

Data aggregation in wireless sensor networks (WSN) is a rapidly emerging research area. It can greatly help conserve the scarce energy resources by eliminating redundant data thus achieving a longer network lifetime. However, securing data aggregation in WSN is made even more challenging, by the fact that the sensor nodes and aggregators deployed in hostile environments are exposed to various security threats. In this paper, we survey the current research related to security in data aggregation in wireless sensor networks. We have classified the security schemes studied in two main categories: cryptographic based scheme and trust based scheme. We provide an overview and a comparative study of these schemes and highlight the future research directions to address the flaws in existing schemes.

Keywords: Data Aggregation, Security, Sensor Networks, Survey.

1 INTRODUCTION

Advancements in micro electro mechanical systems (MEMS) [9] and wireless networks have made possible the advent of tiny sensor nodes called "smart dust" which are low cost small tiny devices with limited coverage, low power, smaller memory sizes and low bandwidth [2]. A wireless sensor network (WSN) [5] is an ad-hoc network consisting of a large number of sensor nodes deployed to sense their surrounding environment. Sensor nodes are

EUROPEAN TRANSACTIONS ON TELECOMMUNICATIONS
Eur. Trans. Telecomms. (2011)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/ett.1532

RESEARCH ARTICLE

Secure DV-Hop localization scheme against wormhole attacks in wireless sensor networks

Nabila Labraoui[1]*, Mourad Gueroui[2] and Makhlouf Aliouat[3]

[1] STIC, University of Tlemcen, Tlemcen, Algeria
[2] PRISM, University of Versailles, Versailles, France
[3] University of Setif, Setif, Algeria

ABSTRACT

Localization is an important topic in mobile wireless ad hoc and sensor networks, which has received considerable attention from the research community during the past few decades. In many sensor networks applications, location awareness is useful or even necessary. However, because of their key role in wireless sensor networks, localization systems can be the target of an attack that could compromise the entire functioning of a wireless sensor network. In this paper, we present a novel defense mechanism against wormhole attacks in DV-Hop localization algorithm. The main idea of our approach is to plug in a proactive countermeasure to the basic DV-Hop scheme called *Infection prevention*. We choose the wormhole attack as our defending target because it is a particularly challenging attack that can be successfully launched without compromising any nodes or having access to any cryptographic keys. Using analysis and simulation, we show that our solution is effective in detecting and defending against wormhole attacks with a high detection rate. Copyright © 2011 John Wiley & Sons, Ltd.

KEY WORDS
range-free localization; secure localization; WSN

***Correspondence**
Nabila Labraoui, STIC, University of Tlemcen, Tlemcen, Algeria.
E-mail: labraouinabila@yahoo.fr

Received 3 March 2011; Revised 30 September 2011; Accepted 3 October 2011

1. INTRODUCTION

Wireless sensor networks (WSNs) have gained worldwide attention academically and industrially because of its great potential for many applications in various scenarios such as military target tracking and surveillance, natural disaster relief, biomedical health monitoring, hazardous environment exploration, and seismic sensing [1]. These networks are especially attractive for scenarios where it is not feasible or expensive to deploy significant networking infrastructure. The sensor network is used to collect information from the sensors distributed in a field to support various (monitoring) applications [2]. In many of these applications, location awareness is useful or even necessary. For example, when a sensor detects an event-driven emergency, its location information should be quickly and accurately determined; sensing data without knowing the sensor's location is meaningless [3]. A straightforward solution is to equip each sensor with a GPS receiver that can accurately provide the sensors with their exact location. Unfortunately, the high costs of GPS technology are at odds with the desire to minimize the cost of individual nodes. Thus, it is only feasible to fit a small portion of all sensor nodes with GPS receivers. These GPS-enabled nodes called *anchor* or *beacon nodes* provide position information, as a beacon message, for the benefit of *non-beacon* or *blind nodes* (i.e., nodes without GPS capabilities). Blind nodes can use the location information furnished by multiple nearby beacon nodes to estimate their own positions, thus amortizing the high cost of GPS technology across many nodes [4].

Localization in WSNs has drawn growing attention from the researchers and many range-based and range-free approaches [5, 6] have been proposed. Because of the hardware limitations of WSN devices, solutions in range-free localization are being pursued as a cost-effective alternative to more expensive range-based approaches [6]. However, the wireless sensor networks themselves, are prone to security attacks [7] and almost all previously proposed localization can be trivially abused by a malicious adversary. Because location information is a part of most wireless sensor networks services, such as geographical

Journal of Universal Computer Science, vol. 17, no. 11 (2011), 1550-1571
submitted: 30/9/10, accepted: 21/6/11, appeared: 1/7/11 © *J.UCS*

RAHIM: Robust Adaptive Approach Based on Hierarchical Monitoring Providing Trust Aggregation for Wireless Sensor Networks

Nabila Labraoui
(STIC University of Tlemcen, Tlemcen, Algeria
labraouinabila@yahoo.fr)

Mourad Gueroui
(PRISM University of Versailles, Versailles, France
mourad.gueroui@prism.uvsq.fr)

Makhlouf Aliouat
(University of Setif, Setif, Algeria
m_aliouat@yahoo.fr)

Jonathan Petit
(University of Twente, Enschede, The Netherlands
j.petit@utwente.nl)

Abstract: In-network data aggregation has a great impact on the energy consumption in large-scale wireless sensor networks. However, the resource constraints and vulnerable deployment environments challenge the application of this technique in terms of security and efficiency. A compromised node may forge arbitrary aggregation value and mislead the base station into trusting a false reading. In this paper, we present RAHIM, a *reactive defense* to secure data aggregation scheme in cluster-based wireless sensor networks. The proposed scheme is based on a novel application of adaptive hierarchical level of monitoring providing accuracy of data aggregation result in lightweight manner, even if all aggregator nodes and a part of sensors are compromised in the network.

Keywords: Accuracy, Availability, Data aggregation, Monitoring mechanism, Wireless sensor networks, Security
Categories: C.2, C.2.3

1 Introduction

Wireless sensor networks (WSN) are rapidly emerging technologies with potentials for many different distributed applications, such as detection of chemical or biological agents, fire detection or tracking of enemy vehicles, which renders them a hot research topic over the past few years. However, sensor network has extremely constrained resources like energy, bandwidth and capabilities of processing and storing data. The current version of sensors such as mica2 [Corporation, 07] uses a 16 bits, 8 MHz Texas Instruments MSP430 microcontroller with only 10 KB RAM, 48 KB Program space, 1024 KB External flash, and is powered by two AA batteries.