République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université A/Mira de Béjaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin de cycle
en vue de l'obtention du diplôme de master recherche en Informatique
Option : Réseaux et Systèmes Distribués

Thème

Gestion de clés basée sur des clusters dans les réseaux de capteurs sans fil

---

Présenté par :
✓ M^{lle} **Boucheneb** Sonia.
✓ M^{lle} **Tahir** Ouissam.

Proposé et encadré par :
✓**M.Nafi** Mohammed.

Devant le jury composé de :

✓**Président** 1 **: Mr. Mir Faudil.**
✓**Examinateur** 2 **: M^{me}. Zidani .**

Promotion 2015/2016

Remerciment

Nous remercions en premier lieu le bon ***DIEU*** *de nous avoir donné les moyens, l'energie mais surtout la volonté nécessaire pour la réalisation de ce modéste travail.*

Nos vifs remerciments vont d'emblée à nos chers parents pour tous les sacrifices consentis à notre égard et leur énormes soutien.

Nous tenons à exprimer nos profondes gratitudes et nos sincères remerciments pour notre promoteur ,Monsieur NAFI MOHAMMED, pour nous avoir encadré et guidé tout au long de ce projet.

Nous témoignons toute notre reconnaissance à MR ***HAMOUDI SAMIR*** *pour son aide précieuse .*

Un grand merci à la secrétaire du département informatique madame LAHDIR pour son soutien moral et ses précieux conseils depuis notre deuxième année.

Nous addressons nos remerciments aux membres de jury qui ont fait l'honneur d'évaluer, examiner et enrichir notre modeste travail.

Merci à tous ceux qui ont contribué de loin ou de prés à la réalisation de ce travail.

Dédicaces

Avec un cœur rempli de joie je dédie ce modeste travail à :

L'amour de ma vie et à l'école de mon enfence PAPA. Aucune dédicace ne serait témoin de mon profond amour, mon immense gratitude et mon plus grand respect, car je ne pourrai jamais oublier la tendresse et l'amour dévoués par lesquels tu m'a toujours entourer depuis mon enfance. Sans toi je n'aurais jamais réuissi. Je t'aime énormément et je te souhaite une longue vie. Que dieu te garde pour moi.

A la plus belle femme au monde MAMAN,je ne te remercerie jamais assez tu es sans doute la meilleure des mamans au monde, que dieux te garde.

A mon grand frére ZAHIR et sa Femme BIBA, merci pour votre soutient.

A mes sœurs NORIA et SOUHILA et que je ne pourrais jamais remercier assez.

A mon meilleur ami garçon avec qui je m'amuse le plus mon frére Sofiane tu es le meilleur.

Au deux meilleurs cousines du monde HINDOUCHE et DABI vous illuminez ma vie.

A ma deuxiéme famille les les HAMMADENE.

A mes niéces ALICE et LENA.

A tout mes amis surtout : Lydia, Thiziri, Lamia, Katia, Amanda, Cherif, Lynda, Siham, Amel.

SONIA

Dédicaces

C'est avec profande gratitude et sincères mots, que je dédie ce modetse travail de fin d'étude à mes chers parents ; qui ont sacrifié leur vie pour ma réussite et nous ont éclairé le chemin par leur conseils judicieux.

J'espère qu'un jour, je pourrai leur un peu de ce qu'ils ont Fait pour nous, que dieu leur prete bonheur et longue vie .

Je dédie aussi ce travail à mes frères et sœurs, mes amis et à tous ceux qui nous sont chers.

OUISSAM

TABLE DES MATIÈRES

TABLE DES FIGURES

INTRODUCTION GÉNERALE

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

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

les lieux stratégiques (*aéroports, bâtiments critiques, etc...*). Ces noeuds capteurs qui opèrent dans des lieux difficiles d'accès, sans protection et sans possibilité de rechargement de batterie, peuvent être soumis à des actions perturbatrices et malveillantes susceptibles de compromettre l'essence même d'un RCSF. C'est pourquoi, il est primordial de pouvoir leur assurer un niveau de sécurité acceptable. Compte tenu de leurs spécificités contraignantes, la sécurité dans ce type de réseaux relève d'un véritable challenge. Comme l'objectif premier des noeuds d'un RCSF est de rassembler des donnés de surveillance et de les transmettre à un lieu de décision, cette opération doit se faire sans interférences malicieuses et avec un niveau de sécurité approprié.
La cryptographie permet de garder secrètes les informations transmises à travers les réseaux, mais elle nécessite des techniques assurant la gestion des clés cryptographiques utilisées dans l'opération de chiffrement /déchiffrement. Par le passé, divers protocoles basés sur la cryptographie asymétrique et d'autres sur la cryptographie symétrique ont été proposés dans la littérature. Basés généralement sur la méthode de pré-distribution de clés, ces protocoles essayent de fournir une gestion de clés plus ou moins légère et assurant une sécurité robuste. Toutefois, de part leurs contraintes spécifiques, les RCSFs sont réfractaires aux protocoles cryptographiques traditionnels, il y a lieu donc de leur adapter ou de concevoir des protocoles propres et conformes à leurs exigences Dans ce mémoire, nous nous proposons d'étudier les différents protocoles de gestion des clés proposés pour les RCSFs, d'examiner leurs capacités à résister à diverses attaques et leurs aptitudes à minimiser l'usage des ressources déjà limitées dans ce contexte. Après quoi, notre objectif est de pouvoir proposer un protocole de gestion des clés permettant de s'acquitter de sa tâche tout en garantissant une économie d'énergie, ressource très critique dont dépend essentiellement la durée de vie d'un RCSF.

Organisation du mémoire

Ce mémoire est organisé comme suit :
Dans le chapitre I, nous présentons un préambule des concepts généraux liés au réseaux de capteurs sans fil.
Dans le chapitre II nous ferons un état de l'art sur la gestion des clés dans les RCSFs ,ou nous ferons une étude bibliographique sur les différents protocoles de gestion de clés.
Apres cela dans le chapitre III nous proposerons un algorithme qui devise le réseau en cluster ,et un schéma de gestion de clés, ensuite nous presenterons les résultats de simulation qui nous permettront de comparer notre protocole à quelques protocoles proposés dans la litérature. Enfin nous finirons par une conculusion générale et des perspectives de notre travail.

CHAPITRE 1

GÉNÉRALITÉS SUR LES RÉSEAUX DE CAPTEURS SANS FIL

Introduction

Nous abordons d'abord dans ce chapitre une présentation sur les réseaux de capteurs. Nous allons principalement décrire leurs architectures, composants, types ainsi que l'architecture de communication, caractéristiques, domaines d'application, ensuite nous introduisons la notion de clustering et de sécurité .

1.1 Réseaux de capteurs sans fil (RCSFs)

1.1.1 Définitions

Un réseau de capteurs sans fil (RCSF) est un ensemble de dispositifs nommés nœuds capteurs, variant de quelques dizaines d'éléments à plusieurs milliers. Dans ces réseaux, chaque nœud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée vers un ou plusieurs points de collecte, à l'aide d'une connexion sans fil [1].

1.1.1.1 Qu'est-ce qu'un capteur ?

Les capteurs sont des dispositifs de taille réduite avec des ressources limitées, autonomes, capables de traiter des informations et de les transmettre, via les ondes radio, à une autre entité (*capteurs ou une unité de traitement*) sur une distance limitée à quelques mètres [3].

1.1.1.2 Architecture de base d'un capteur

Un noeud capteur est composé de quatre unités principales [3], qui sont présentées dans la (figure 1.1)

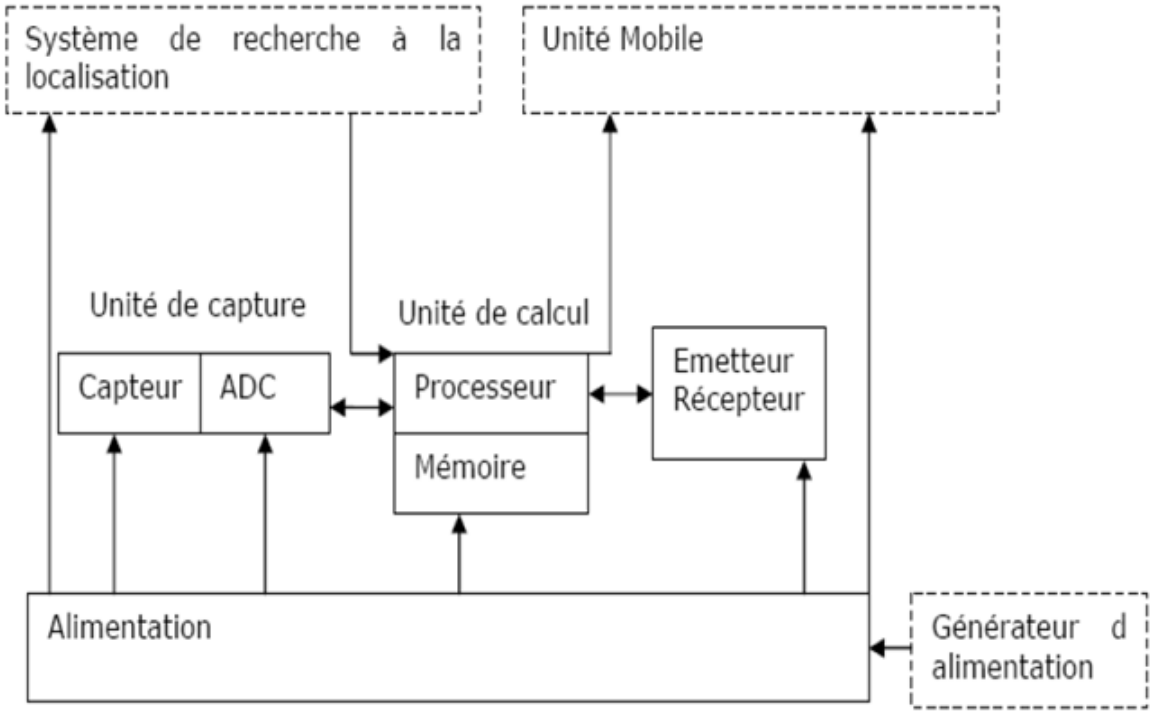

FIGURE 1.1 – Architecture d'un capteur [4].

- **Unité de capture (Sensing unit) :** Elle est composée de deux sous unités, un dispositif de capture physique qui prélève l'information de l'environnement et un convertisseur analogique appelé ADC (*Analog to Digital converts*) [1].
- **Unité de calcul (Processing unit) :**Le composant regroupe un processeur, et une unité de memoire reduite.
  Il permet de stocker les données, éxecute les tâches de perception qui lui sont assignées [1].
- **Unité de émetteur/récepteur :** Elle est composée d'un émetteur/récepteur (*module radio*) permettant la communication entre les différents nœuds du réseau [1].
- **Unité d'alimentation) :**c'est la batterie qui n'est généralement ni rechargeable ni remplaçable.la capacité d'énergie est limitée au niveau des capteurs et elle représente la contrainte principale lors de la conception des protocoles pour les réseaux de capteurs [1].

1.1.1.3 Caractéristiques d'un capteur

En analysant la gamme des composants disponibles sur le marché et les prototypes présentés dans la littérature, il est évident que la principale caractéristique d'un nœud capteur sans fil

est sa petite taille. Une deuxième caractéristique, évidente mais essentielle est l'autonomie (*pas seulement d'énergie élevée au point de vue de leur source évidente, mais aussi de leur fonctionnement*).
Ces deux premières particularités induisent plusieurs autres caractéristiques à considérer, en particulier la vitesse de calcul et la vitesse de transmission. Des performances élevés en termes de vitesse de traitement et de transmission impliquent une consommation élevée.
De manière générale, il est souhaitable que la durée de vie de la batterie des nœuds soit la plus grande possible, donc les différentes unités qui composent un nœud sont généralement très limitées en termes de ressources et de performances pour que leur consommation d'énergie soit extrêmement faible [1].

1.1.2 Objectifs des RCSFs

Les objectifs de base des réseaux de capteurs sans-fil dépendent généralement des applications, cependant les tâches suivantes sont communes à plusieurs applications :

- Déterminer les valeurs de quelques paramètres suivant une situation donnée. Par exemple, dans un réseau environnemental, on peut chercher à connaitre la température, la pression atmosphérique, la quantité de la lumière du soleil, et l'humidité relative dans un nombre de sites, etc.
- Détecter l'occurrence des évènements dont on est intéressé et estimer les paramètres des évènements détectés. Dans les réseaux de contrôle de trafic, on peut vouloir détecter le mouvement de véhicules à travers une intersection et estimer la vitesse et la direction du véhicule.
- Classifier l'objet détecté. Dans un réseau de trafic, un véhicule est-il une voiture, un bus, etc... [2].

1.2 Architecture des réseaux de capteur sans fil

Un réseau de capteurs est typiquement constitué d'un champ de captage, un ensemble de nœuds capteurs, une station de base et un centre de traitement de données [1]. Cette architecture est illustrée dans la figure 1.2.

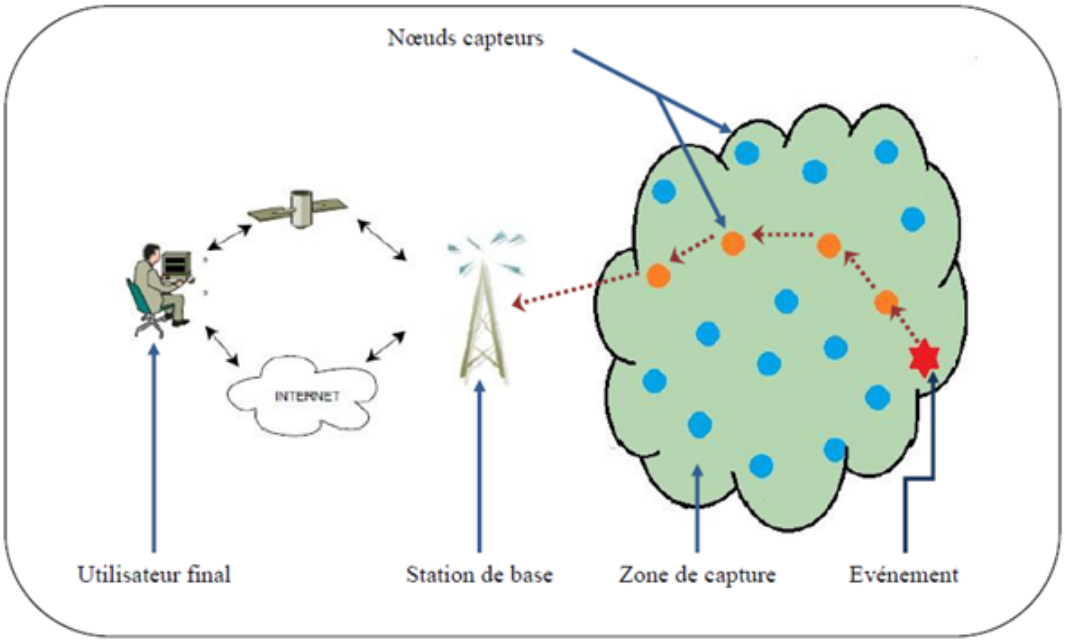

FIGURE 1.2 – Architecture de base d'un réseau de capteur sans fil [7].

- **Zone de captage (*éspace de collecte*) :** Il est considéré comme étant la zone d'intérêt pour le phénomène capté, là où les nœuds capteurs y sont placés.
- **Noeuds capteurs :** Ce sont le coeur du réseau, leur rôle est de collecter les données et de les router vers la station de base. Leur énergie est souvent limitée puisqu'ils sont alimentés par une batterie.
- **Station de base (*sink, puits*) :** C'est un noeud particulier chargé d'accueillir, stocker et traiter les données en provenance des autres noeuds et de diffuser les différentes requêtes sur le réseau. Sa source d'énergie est généralement illimitée puisqu'il faut qu'elle reste toujours active pour recevoir les données.
- **Utilisateur final (*gestionnaire des tâches*) :** il reçoit les données collectées par la station de base. Son rôle consiste à les regrouper et les traiter pour en extraire les informations utiles [1]. Selon [8] il existe deux types d'architectures pour les réseaux de capteur sans fil :

  -Les réseaux de capteurs sans fil plats : Un réseau de capteurs sans fil plat est un réseau homogène, où tous les nœuds disposent des mêmes capacités et fonctionnalités concernant le captage, la communication et la configuration du matériel. Seule la station de base échappe à cette règle puisqu'elle joue le rôle d'une passerelle chargée de la collecte des données issues des différents nœuds capteurs pour les transmettre à l'utilisateur.

  -Les réseaux de capteurs sans fil hiérarchiques : un réseau de capteur hiérarchique est un réseau hétérogène où les nœuds ont des capacités différentes. Par

exemple, certains nœuds peuvent disposer d'une source d'énergie plus importante ou une portée de communication plus large (*grande*).

1.3 Domaines d'application des RCSFs

La miniaturisation des micro-capteurs, le coût de plus en plus faible et la large gamme de capteurs disponibles (*thermique, optique, vibrations, etc.*) ainsi que le support de communication sans fil utilisé, permettent l'utilisation des réseaux de capteurs dans plusieurs domaines parmi lesquels : [13]

- **Domaine militaire :** Comme pour de nombreuses autres technologies, le domaine militaire a été le moteur initial pour le développement des réseaux de capteurs. Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine. Actuellement, les RCSF peuvent être une partie intégrante dans le commandement, le contrôle, la communication, la surveillance, la reconnaissance,... etc.
- **Domaine médical :** Les réseaux de capteurs sont également largement répandus dans le domaine médical. Cette classe inclut des applications comme : fournir une interface d'aide pour les handicapés, collecter des informations physiologiques humaines de meilleure qualité, facilitant ainsi le diagnostic de certaines maladies, surveiller en permanence les malades et les médecins à l'intérieur de l'hôpital.
- **Domaine architectural :** Transformation des bâtiments en environnements intelligents capables de reconnaitre des personnes, interpréter leurs actions et y réagir.
- **Domaine environnemental :** Dans ce domaine, les capteurs peuvent être exploités pour détecter les catastrophes naturelles (*feux de forêts, tremblements de terre, etc.*), détecter des émanations de produits toxiques (*gaz, produits chimiques, pétrole, etc...*) dans des sites industriels tels que les centrales nucléaires ou pétrolières.
- **Domaine commercial :** Parmi les domaines dans lesquels les réseaux de capteurs ont aussi prouvé leur utilité, on trouve le domaine commercial. Dans ce secteur nous pouvons énumérer plusieurs applications comme : la surveillance de l'état du matériel, le contrôle et l'automatisation des processus d'usinage, etc...[13]

1.3.1 Contraintes Conceptuelles

La conception des RCSF, leurs protocoles et algorithmes sont guidés par plusieurs facteurs :

- **La tolérance aux pannes :** La défaillance ou le blocage de certains nœuds dans un réseau de capteurs peut être engendrés par plusieurs causes, notamment l'épuisement d'énergie, l'endommagement physique ou les interférences liées à l'environnement. Ces

problèmes ne devraient pas affecter le reste du réseau. C'est le principe de la tolérance aux pannes.

- **L'extensibilité (*passage à l'échelle*) :** L'une des caractéristiques des RCSF est qu'ils peuvent contenir des centaines voire des milliers de nœuds capteurs. Suivant l'application, ce nombre peut encore augmenter jusqu'à des millions de capteurs. Les nouveaux schémas doivent pouvoir garantir un bon fonctionnement avec ce nombre élevé de capteurs. Ils doivent aussi exploiter la nature fortement dense des réseaux de capteurs.
- **L'environnement :** Les nœuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : forte chaleur, pluie, humidité...
- **Le média de transmission :** Les nœuds communicants sont reliés sans fil. Ce lien peut être réalisé par radio, signal infrarouge ou un média optique [14].
- **La Contrainte d'énergie, de stockage et de calcul :** La caractéristique la plus critique dans les réseaux de capteurs est la modestie de ses ressources énergétiques car chaque capteur du réseau possède de faibles ressources en termes d'énergie, de calcul et de stockage. Afin de prolonger la durée de vie du réseau, une minimisation des dépenses énergétiques est exigée chez chaque nœud [20]
- **L'agrégation de données :** Dans les RCSF, les données produites par les nœuds capteurs voisins sont corrélées spatialement et temporellement. Ceci peut engendrer la réception par la station de base d'information redondante. Réduire la quantité d'informations redondantes transmises par les capteurs permet de réduire la consommation d'énergie dans le réseau ainsi d'améliorer sa durée de vie. L'une des techniques utilisée pour réduire la transmission d'informations redondantes est l'agrégation des données, appelée aussi fusion des données.
- **Les contraintes matérielles** Parmi les contraintes matérielles liées aux RCSF, on peut citer :
- **La dimension :** La taille réduite des capteurs peut présenter de nombreux avantage, elle permet un déploiement flexible et simple du réseau. Cependant, la puissance des batteries utilisées pour alimenter les nœuds capteurs est limitée par la petite taille de ces derniers.
- **La puissance de calcul :** les processeurs des réseaux de capteurs sont différents de ceux d'une machine classique car ils utilisent souvent des micros contrôleurs de faibles fréquences [14].

1.4 Notion de cluster et de clustering

1.4.1 Définition

Un cluster est un sous-ensemble de nœuds connexe, et la structuration ou clustering est le processus de regroupement des nœuds en clusters donnant ainsi au réseau une structure

hiérarchique. Généralement et comme le montre la (Figure 1.4), les clusters comportent trois types de noeuds [10] :

- Un nœud particulier appelé chef de cluster ou "cluster-Head" (CH). Ce dernier permet de coordonner les membres de son cluster, d'agréger et /ou de traiter les données collectées et de les transmettre au collecteur de données. Le chef de cluster est choisi pour jouer ce rôle soit d'une manière déterministe (*chef de cluster prédéfini*) ou d'une manière aléatoire (*chef de cluster élu parmi les nœuds du réseau selon une métrique bien particulière ou une combinaison de métriques*).
- Un nœud passerelle ou "gateway" qui possède des liens inter-clusters et peut donc accéder à des clusters voisins et acheminer les données entre eux.
- Enfin un nœud ordinaire ne possédant pas de liens avec les autres clusters et quand il s'attache à un chef de cluster il en devient membre .

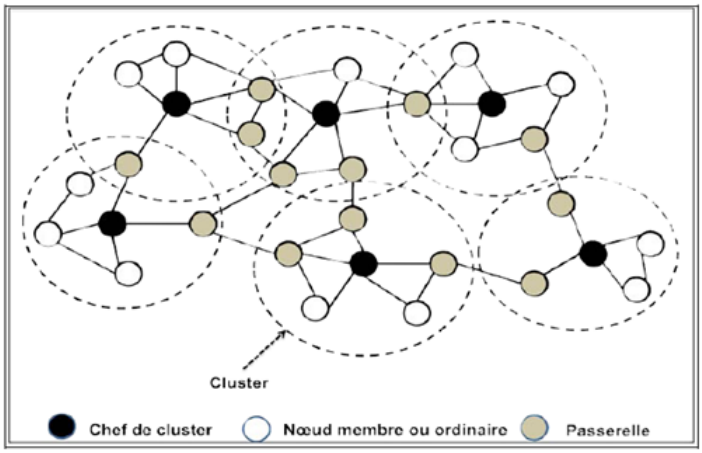

FIGURE 1.3 – Exemple de structure de clusters [10].

1.4.2 Les objectifs du clustering

La structuration du réseau en clusters a pour objectif de réduire les communications. Cet objectif général englobe plusieurs sous objectifs. Nous allons énumérer quelque uns des plus importants. Cependant, avant de lister ces objectifs, rappelons que dans [11], les auteurs précisent que le clustering comme tous les algorithmes d'infrastructure doit avoir un coût minimal. Cela est d'autant plus important que le processus de clustering peut être ramené à s'exécuter a plusieurs reprises pour réorganiser les structures suite à la mobilité des nœuds ou suite à d'autres pannes transitoires [10].

- **Equilibrer la charge :** le clustering cherche à répartir équitablement les tâches les plus coûteuses dans le réseau afin d'éviter des points de congestion ou une consommation de ressources déséquilibrée entre les nœuds du réseau [12].

- **Prolonger la durée de vie du réseau :** par la réduction des communications, le clustering vise à retarder l'épuisement des ressources des nœuds du réseau. Cela aura pour conséquence de prolonger la durée de vie du réseau [12].
- **Assurer une connectivité totale et une réduction des délais :** le clustering vise aussi à assurer une connectivité totale dans le réseau. Malgré le fait que les clusters constituent des zones de cloisonnement, n'importe quel couple de nœuds, se trouvant dans le même cluster ou dans des clusters différents, doit pouvoir communiquer. De même, la communication entre n'importe quel couple de nœuds doit être assurée avec un délai satisfaisant selon les besoins applicatifs [12] ;
- **Optimiser la bande passante :**en minimisant les communications et en évitant les duplications de messages et les retransmissions inutiles, le clustering vise à optimiser l'utilisation de la bande passante du réseau .
- **Assurer une qualité de service (QoS) :** la mobilité des nœuds ou d'autres fautes transitoires peuvent entrainer des ruptures de liens de communications entrainant ainsi un arrêt momentané du service fournit par l'applicatif du réseau. Il est donc nécessaire d'assurer une certaine qualité de service, même en présence de ruptures de liens de communication, afin d'assurer le service requis [12].

1.4.3 Cas d'utilisation possible du clustering

Les objectifs et les propriétés décrits ci-dessus, une fois satisfaits, permettent à la structuration en clusters d'offrir quelques cas d'utilisation qui ont favorisé l'attention portée sur elle ces dernières décennies [12].

- **Coordination des communications :** la structure hiérarchique offerte par le clustering permet de mieux coordonner les communications dans le réseau. En effet, il est possible, par exemple, de faire orchestrer les communications au sein des clusters par les cluster-heads mais aussi d'établir des politiques de communications entre clusters adjacents [12].
- **Routage hiérarchique :**dans les clusters, il est possible de mettre en place un routage hiérarchique en établissant une politique de routage au sein des clusters et d'autres schémas de routages spécifiques aux échanges d'informations entre clusters. Cela permet d'avoir un minimum d'informations à stocker dans les tables de routage conduisant ainsi à un routage plus efficace [12].
- **Agrégation de données :**la hiérarchisation permet aussi une réutilisation et une redistribution des ressources du réseau. En effet, les mêmes fréquences ou codes d'accès au médium peuvent être utilisés dans deux clusters différents à condition que les clusters soient non-recouvrant (*pour éviter les collisions et interférences*) [12].

1.4.4 Classification des solutions de clustering

Il existe dans la littérature une diversité d'algorithmes de formation de clusters qui peuvent être distingués et classés selon plusieurs paramètres tels que le type de réseau et le type d'algorithmes utilisés [11] :

- **Le type du réseau (réseau homogène :** constitué de nœuds de même ressources ou réseau hétérogène (constitué de nœuds à ressources non égales).
- **Le type d'algorithme utilisé :**
- **centralisé :**où c'est le concepteur de l'algorithme ou la SB ou le puits qui désigne les Chefs de clusters parmi les nœuds du réseau et ensuite, c'est le chef de cluster qui initie la construction de la structure virtuelle. Le nœud choisi peut être un nœud ordinaire où encore un super nœud ayant moins de contraintes de ressources (*réserve d'énergie, portée radio, capacité de traitement,etc...*) ;
- **Distribué :**où le chef de cluster est choisi suite à des interactions entre les nœuds pendant une durée déterminée ;
- Algorithme à 1-saut où chaque nœud à pour voisin un chef de cluster ;
- Algorithme à K-sauts qui permet de construire un nombre faible de clusters où les noeuds sont au maximum à une distance K de leurs chefs de clusters.

Plusieurs études bibliographiques ont essayé de faire des classifications des algorithmes de clustering selon différents angles de vue. Ainsi, les auteurs de (*Abbasi et Younis, 2007*) ont réalisé une taxonomie des différents algorithmes de clustering dédiés aux RCSFs. Ces algorithmes ont été comparés selon quelques métriques tels le taux de convergence, la stabilité de cluster, le recouvrement des clusters, la géolocalisation et la mobilité des noeuds. Alors que dans (*Arboleda et Nasser, 2006*) des algorithmes de clustering ont été comparés selon quelques concepts de base liés au processus de clustering, comme la structure des clusters, les types des clusters et les avantages du clustering. Tandis que (*Kumarawadu et al, 2008*) ont basé leur classification sur les paramètres de construction des clusters et les critères du choix d'un chef de cluster dans le mécanisme d'élection. Les auteurs ont discuté les algorithmes de clustering probabilistes basés sur les informations de voisinage et l'identité des noeuds [11].

1.4.5 Construction d'une topologie en cluster

De nombreuses techniques de clustering ont été proposées dans la littérature. Elles varient selon le mode de déploiement des nœuds (*déterministe ou aléatoire*), le processus d'élection des cluster-heads, la taille des clusters, le modèle de fonctionnement du réseau, etc. Le principe général de construction d'une structure auto-organisée en cluster est décrit sur les (Figures 1.5, 1.6, 1.7). Après une phase de découverte du voisinage (Figure 1.5), le RCSF construit sa structure en groupes de nœuds (Figure 1.7) ainsi qu'un chemin dominant de communication appelé dorsale (Figure 1.6). Notons que les deux dérnieres étapes sont chronologiquement in-

terchangeables et même peuvent être réalisées en même temps. Classiquement un algorithme simple de clustering peut se décrire ainsi [12] :

- Chaque nœud découvre son voisinage par le biais des messages "Hello" qu'il diffuse à son voisinage. Cela lui permettra de calculer sa métrique (Figure 1.5).

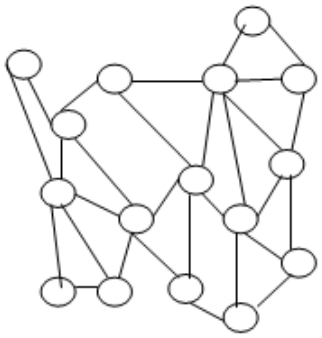

FIGURE 1.4 – Découverte du voisinage.

- Hors le cas d'une pré-désignation du cluster-head, un nœud détermine s'il est cluster-head ou pas en fonction de sa métrique et de celle de son voisinage (*immédiat ou non*) (Figure 1.6).

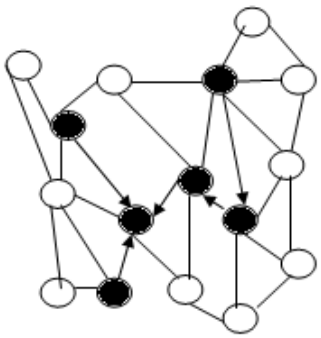

FIGURE 1.5 – construction de dorsale .

- Un nœud choisi comme cluster-head, diffuse son statut à son voisinage afin de notifier son désir de former un cluster et d'inviter ses voisins non affiliés à le rejoindre dans son cluster Figure 1.7.
- Tout changement de statut est notifié par une diffusion de message.

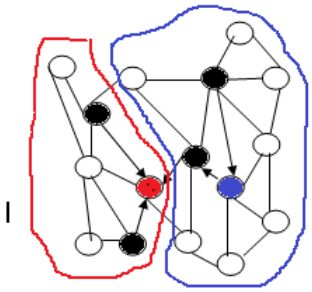

FIGURE 1.6 – construction des clusters.

La phase d'élection de cluster-heads appelée aussi la phase Set-up utilise une métrique spécifique ou une combinaison de métriques pour chaque noeud telle que le plus grand/petit ID dans son voisinage, le degré de connectivité, la puissance de transmission, l'énergie restante ou la mobilité, etc.... ou bien un poids qui représente une combinaison de quelques métriques. Les groupes formés peuvent avoir différentes caractéristiques selon la stratégie adoptée : clusters de tailles homogènes ou non, recouvrant ou non, passifs ou actifs, etc.
Si les clusters sont recouvrant, un nœud peut alors appartenir à plusieurs clusters. En général ces nœuds auront un rôle de passerelle dans la communication entre clusters. Dans le cas contraire, un nœud n'est associé qu'à un seul groupe. Dans un cluster, tout membre peut être soit au plus à 1 saut soit au plus à k sauts de son cluster-head. Dans un cluster à 1 saut, le cluster-head est directement connecté à tout nœud membre.

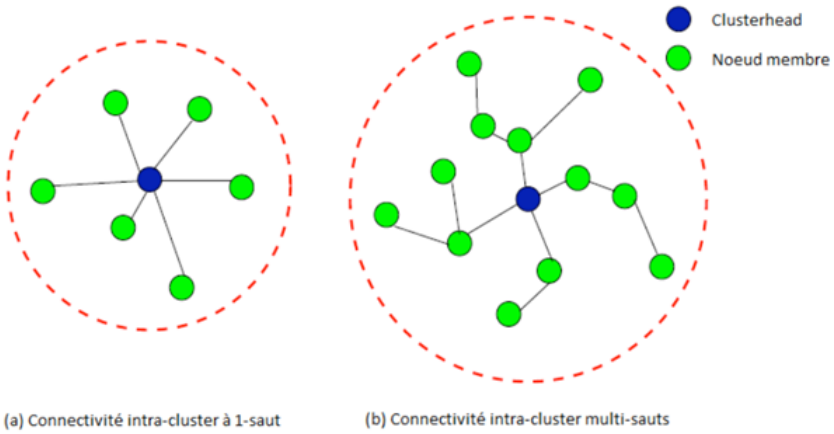

FIGURE 1.7 – Clusters à 1-saut ou à k-sauts [13].

1.5 La sécurité dans les RCSFs

Des contraintes parfois strictes et intrinsèques aux RCSFs imposent de penser à une sécurité mieux adaptée que son équivalent traditionnel des réseaux filaires.

1.5.1 Défis des RCSFs

Généralement l'objectif des RCSFs est de rassembler des donnés de surveillance et d'agir dans l'environnement. Les noeuds capteurs opèrent dans des lieux difficiles d'accès, sans protection et sans possibilité de rechargement de batterie. Le premier défi consiste alors à minimiser la consommation de l'énergie tout en maximisant les performances de sécurité. Un autre défi réside dans les caractéristiques spécifiques de la communication sans-fil rendant les politiques de sécurité appliquées dans les réseaux filaire impraticables [13].

1.5.2 Objectifs de la sécurité dans les RCSFs

- **Disponibilité :** La disponibilité donne une assurance sur la réactivité et le temps de réponse d'un système pour transmettre une information d'une source à la bonne destination. Cela signifie aussi que les services du réseau sont disponibles aux parties autorisées si nécessaire et assure les services de réseau en dépit des attaques de déni de service (DoS) pouvant affecter n'importe couche du réseau.
- **Intégrité des données :** C'est un service qui garantit que les données n'ont pas été altérées pendant la transmission. On peut distinguer les altérations accidentelles liées par exemple, à une mauvaise couverture des ondes, et les altérations volontaires d'un attaquant. Cela Concerne aussi la protection contre l'injection ou la modification des paquets.
- **Confidentialité :** La confidentialité est la garantie que l'information d'un noeud n'est rendue accessible ou révélée qu'à son destinataire. Dans notre cadre, il est important qu'aucun capteur étranger au système ne puisse être mis à proximité dans l'intention de surveiller les informations échangées.
- **Fraîcheur :** Elle concerne la fraîcheur de données et la fraîcheur des clés. Puisque tous les réseaux de capteurs fournissent quelques formes de mesures variables dans le temps, nous devons assurer que chaque message est frais. La fraîcheur de données implique que les données sont récentes, et elle assure qu'aucun adversaire n'a rejoué les vieux messages.
- **Authentification :** Un adversaire n'est pas simplement limité à modifier le paquet de données mais il peut également injecter des paquets supplémentaires. Ainsi le récepteur doit s'assurer que les données utilisées proviennent de la source correcte. D'autre part, en construisant le réseau de capteur, l'authentification est nécessaire pour beaucoup de tâches (*transmissions des mesures prélevées vers la station de base, synchronisation,...*).

- **Non répudiation :** Mécanisme destiné à prévenir que la source ou la destination désavoue ses actions ou nie qu'un échange ait eu lieu.
- **Contrôle d'accès :** Un service très important consiste à empêcher un accès au réseau à tout élément étranger au système. Le contrôle d'accès donne aux participants légitimes un moyen de détecter les messages provenant de sources externes au réseau [33].

1.5.3 Les attaques dans les RCSFs

Une classification des attaques consiste à distinguer les attaques passives des attaques actives.

Les attaques passives se limitent à l'écoute et l'analyse du trafic échangé. Ce type d'attaques 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 noeuds importants dans le réseau (*chef de groupe*). 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. Parmi les attaques actives les plus connues, on peut citer :

- **Attaque physique d'un noeud :** L'attaque physique peut être considérée 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 attaqué. 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 [33].
- **Attaque du trou noir (*black hole) :*** Un noeud 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 (*noeud malveillant, qui s'introduit illégitimement*), peut aussi se placer 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 noeud intrus. La nature des RCSFs où les informations sont routées vers une station de base rend ce type d'attaque plus réussi [33].
- **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 noeud malicieux. Par exemple, les paquets de données ne sont pas retransmis alors que les paquets de routage le sont [33].

- **Relais sélectif de paquets :** Un noeud néglige son rôle de routeur et décide de ne pas transmettre les données de certains noeuds choisis selon certains critères ou d'une façon aléatoire. La raison peut être aussi bien d'ordre énergétique, que liée à une attaque [33].
- **Attaque par chantage :** Un noeud malicieux fait annoncer qu'un autre noeud légitime est malicieux pour éliminer ce dernier du réseau. Si le noeud malicieux arrive à attaquer un nombre important de noeuds, il pourra perturber le fonctionnement du réseau [33].
- **Attaque de l'inondation de HELLO :** De nombreux protocoles de routage utilisent des paquets HELLO pour découvrir les noeuds 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 [33].
- **Brouillage radio (*jamming*) :** Une attaque bien connue sur la communication sans-fil, est celle qui consiste à perturber le canal radio en envoyant des informations inutiles sur la bande de fréquences utilisées. Ce brouillage peut être temporaire, intermittent ou permanent [33].
- **Wormholes :** Connu aussi sous le vocable de tunneling, dans cette attaque, un adversaire peut recevoir des messages et les rejouer dans différentes parties à l'aide d'un tunnel entre les noeuds malicieux [33].

Conclusion

Les réseaux de capteurs restent une nouvelle technologie peu accessible au grand publique. Elle est principalement répandue dans les laboratoires de recherches.

Des progrès sont encore à réaliser dans ce domaine. Néanmoins ils correspondent à une certaine vision du futur et permettront des améliorations dans d'innombrables domaines de la vie quotidienne.

Dans ce chapitre, nous avons défini ce qu'est un RCSF. Nous avons évoqué brièvement des notions que nous allons développer dans les chapitres à venir de ce mémoire.

Dans le chapitre suivant, nous ferons un état de l'art sur les protocoles de gestion de clés déterministe basé sur une topologie hiérarchique dans les RCSFs.

CHAPITRE 2

ETAT DE L'ART SUR LES PROTOCOLES DE GESTION DE CLÉS DANS LES RCSFS

Introduction

La majorité des études et recherches dans les réseaux da capteurs sont basées sur la capacité de les rendres faisables et utiles. Des solutions ont été proposées par des chercheurs aux menaces liées à la sécurité des RCSFs.

Confidentialité, intégrité, et authentification sont des services critiques permettant d'empêcher un adversaire de compromettre la sécurité d'un RCSF ; la gestion et l'établissement des clés est nécessaire pour assurer cette protection dans les RCSFs. Cependant, fournir une gestion efficace des clés est difficile en raison de la nature ad hoc du réseau, connectivité intermittente et limitations des ressources du réseau de capteur.

Par conséquence, la gestion des clés est un service primordial pour la sécurité de n'importe quel système basé sur la communication. Parmi les raisons qui nous ont motivées a travailler dans le domaine de gestion des clés dans les RCSFs, nous citons :

- Chaque système cryptographique est fondé sur la gestion des clés.
- De 'Bonnes' clés cryptographiques doivent se présenter pour qu'elles soient utilisées par la cryptographie, la signature numérique ou MAC.
- La sécurité des clés implique la sécurité du réseau entier.
- La confiance accordée aux informations reçues.
- Une rupture dans le schéma de distribution des clés a souvent comme conséquence l'échec de sécurisation d'une communication sans fil.
- La rénovation des clés est très importante et essentielle dans les RCSFs.
- Le problème de gestion de clés est l'un des problèmes les plus délicats de la cryptographie.

2.1 Problématique

La gestion des clés est un processus par lequel des clés cryptographiques sont produites, enregistrées, protégées, transférées, chargées, employées, et détruites. Où la problématique se pose :
-Le problème de pré-distribution des clés : Quel est le nombre de clés nécessaire et comment sont-elles distribuées avant le déploiement des nœuds ?
- Le problème d'ajout de nœud : Comment un nœud ajouté au réseau peut établir et fixer une clé avec des nœuds existant dans le réseau ? .
-Le problème de l'établissement de clé : Comment une paire de nœuds, ou un groupe de nœuds établissent une clé ?
-Le problème d'isolation des nœuds anormaux : Comment un nœud "expulsé" du réseau ne pourra plus établir de clés avec n'importe quel nœud existant dans le réseau, et il ne sera plus capable de déchiffrer le trafic d'information dans le réseau ?

2.2 Gestion des clés dans les réseaux de capteurs

La gestion des clés est un des aspects les plus difficiles de la configuration d'un système cryptographique de sécurité. Pour qu'un tel système fonctionne et soit sécurisé, chacun des utilisateurs doit disposer d'un ensemble de clés secrètes (*dans un système à clés secrètes*) ou de paire de clés publiques/privés (*dans un système à clés publiques*). Cela implique de générer les clés et de les distribuer de manière sécurisée aux utilisateurs ou d'offrir à l'utilisateur le moyen de les générer. Il doit aussi pouvoir enregistrer et gérer ses clés publiques et privées de manière sure [15] comme le montre la figure 2.1.

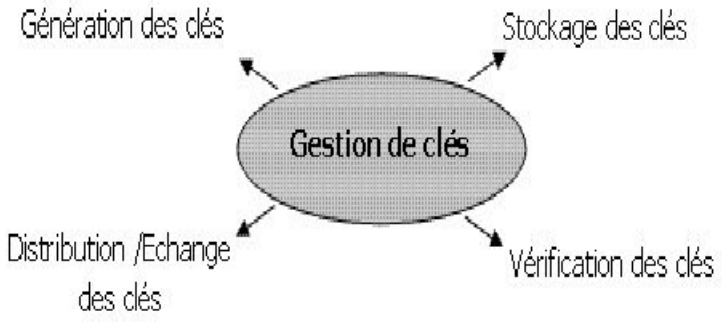

FIGURE 2.1 – Fonctions de la gestion de clés [14].

2.2.1 Pourquoi la gestion de clés dans les RCSF ?

Après leur déploiement, les capteurs ont besoin d'établir des clés cryptographiques avec leurs voisins pour assurer des services de sécurité.

2.2.2 Phases de la gestion des clés

2.2.2.1 Pré-distribution de clés

il s'agit de pré-charger des clés dans chaque noeud avant le déploiement afin de sécuriser la transmission de paquets [18].

2.2.2.2 Découverte de voisinage

Chaque noeud doit découvrir ses voisins dans sa porté sans-fil de communication avec les quels il partage des clés. Un lien existe entre deux noeuds de capteur seulement s'ils partagent une clé. Un bon schéma de découverte des voisins ne donnera à un attaquant aucune occasion de découvrir les clés partagées, et l'attaquant peut seulement faire l'analyse de trafic [18].

2.2.2.3 Etablissement de clés de chemin

Etablir des clés entre les noeuds non liés directement. Pour n'importe quelle paire de noeuds qui ne partagent pas une clé mais sont reliés par un chemin multi saut doivent fixer une clé de chemin "path key" pour sécuriser la communication bout à bout, cette clé de chemin ne peut pas être celle déjà employée entre les noeuds voisins [18].

2.2.2.4 Isolation des noeuds anomaux

Identifier et isoler les noeuds anormaux qui agissent comme des noeuds intermédiaires est important pour continuer l'opération du RCSF. Un noeud peut cesser de fonctionner comme prévu pour les raisons suivantes :

- - Il a épuisé sa source de puissance.
- - Il est endommagé par un attaquant.
- - Un noeud intermédiaire a été compromis et il corrompt la communication en modifiant les données.
- - Un noeud a été compromis et il communique l'information factice à la station de base [18].

2.2.2.5 Renouvellement des clés

De nouvelles clés doivent être mises en service et cela dans le cas où la vie des clés expirent. La rénovation des clés re-keying est un défi puisque de nouvelles clés doivent être produites

d'une manière efficace et conforme à une consommation et conservation d'énergie [18].

2.2.2.6 Latence d'établissement des clés

Réduire la latence résultante des communications et conserver de l'énergie constitue un objectif primaire dans le processus de gestion des clés. Tout schéma de gestion des clés devrait prendre la réduction de latence comme facteur crucial [18].

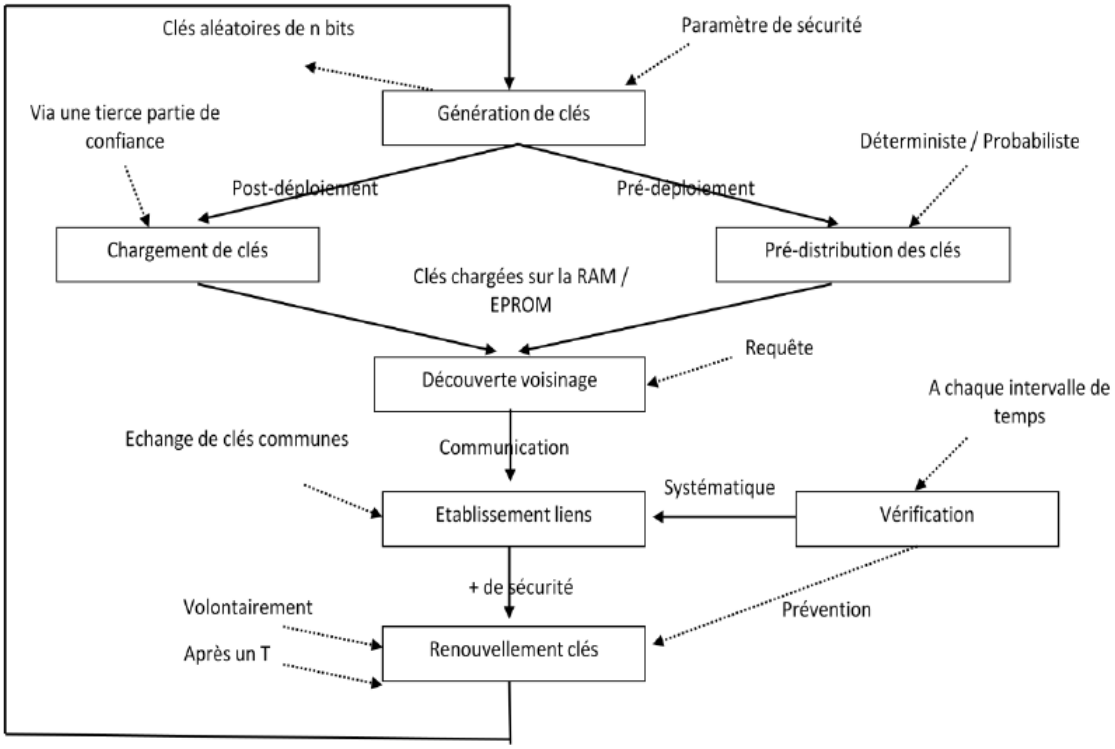

FIGURE 2.2 – Schéma de gestion de clés [7].

2.3 classification des protocoles de gestion de clés

La gestion des clés dans les RCSFs est habituellement décrite par le procédé de pré-distribution des clés qui exige un chargement d'information secrète dans les nœuds capteurs avant leurs déploiement dans le réseau, cette dernière peut être une clé secrète, ou une information auxiliaire qui aide les nœuds à dériver la clé secrète réelle.La plupart des approches de gestion des clés disponibles actuellement tombent dans une des classes suivantes : approche utilisant la cryptographie symétrique ou asymétrique. En se basant sur ce critère, nous avons classifié les protocoles de gestion des clés dans les réseaux de capteurs en deux classes : ceux utilisant la cryptographie symétrique et ceux utilisant la cryptographie asymétrique, ainsi la (figure 2.3) illustre cette classification.

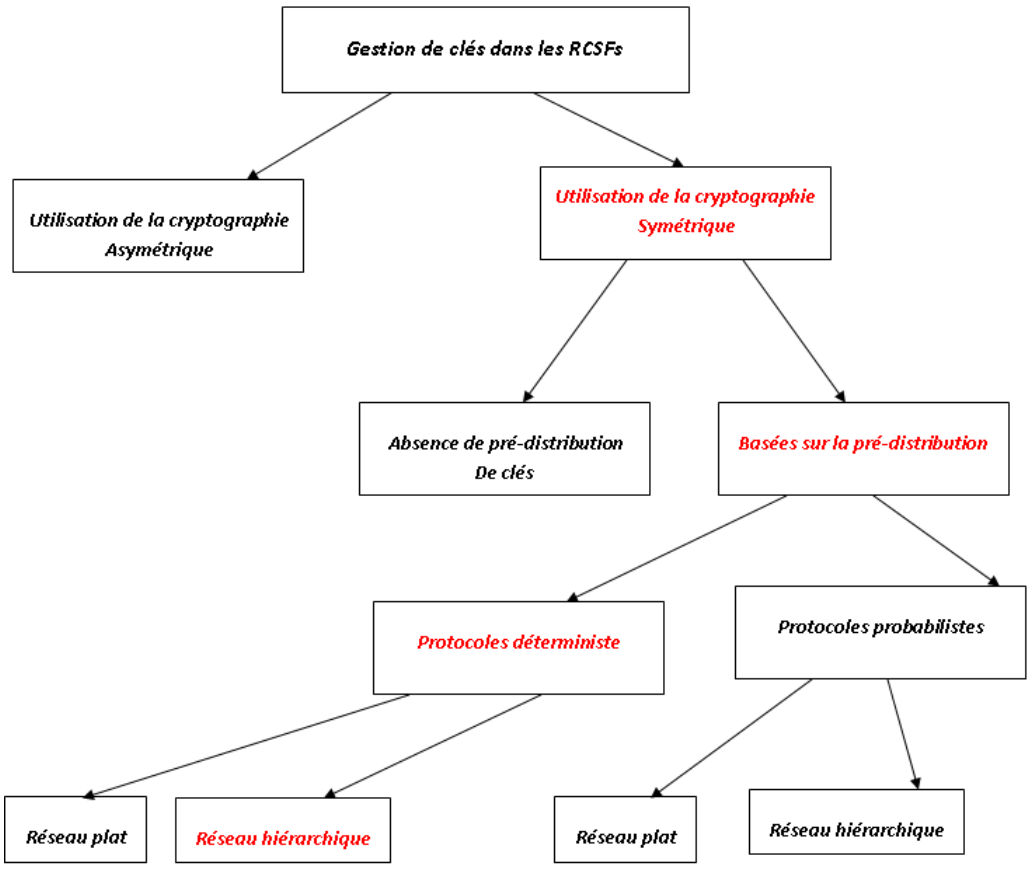

FIGURE 2.3 – classification des approches de gestion de clés .

2.3.1 Utilisation de la cryptographie asymétrique

Principe : avant le déploiement, chaque noeud a la clé maitresse publique et privée (KM, KM^{-1}), puis chaque noeud A génère sa paire de clés (KA, KA^{-1}). Après le déploiement, les noeuds échangent les clés : échange des clés publiques et une signature par la clé maitresse pour la vérification des clés publiques reçues. Par la suite, une clé symétrique peut être générée et échangée entre les noeuds encryptée par leurs clés publiques, Carmen, Kruus et Matt ont présenté une analyse d'efficacité de la cryptographie asymétrique dans les RCSFs [2].

Avantages :

- Résistance contre la capture de noeud.
- Permettre la "scalabilité".

Inconvénients :

- Exigence sur le hardware et le software des noeuds (hardware : augmentation du coût des noeuds, software : demande beaucoup de calcul, consommation d'énergie).
- Vulnérabilité vis-à-vis des attaques du déni de service : l'emploi de la cryptographie asymé-

trique implique un calcul qui peut prendre quelques secondes (*voir quelque minutes*), les noeuds sont vulnérables à un déni de service d'épuisement de batterie par un attaquant inondant le réseau par des signatures illégales.
- Pas de résistance contre la réplication de noeud.
La plupart des algorithmes asymétriques sont adéquats pour l'usage dans les MANETs (*réseaux ad hoc mobiles*) mais ils ne sont pas adéquats pour les RCSFs parce que ce genre de réseaux utilise des dispositifs faibles en énergie (*low-power*) [2].

2.3.2 Utilisation de la cryphtographie symétrique

Bien que la cryptographie asymétrique comporte des avantages certains par rapport à la cryptographie à clés symétriques et malgré les recherches qui visent à les appliquer aux RCSFs, la cryptographie à clé symétrique possède ses propres qualités qui la rende toujours la plus préférée pour les RCSFs. Pour cette raison la plupart des schémas de gestion des clés proposés pour les réseaux de capteurs sont basés sur la cryptographie symétrique [20].

2.3.2.1 Absence de pré-distribution de clés

Ce mécanisme ne prend en considération aucune pré-distribution de clés. Par ailleurs si un adversaire ne sait pas où et quand les nœuds sont déployés, il serait difficile pour lancer une attaque active. Cependant il est souvent présent après ou avant la phase de déploiement des nœuds, il peut surveiller toutes les communications à tout moment. Un tel adversaire n'est souvent pas réaliste, dans la plus part des applications, il n'est capable de surveiller qu'une certaine petite portion du trafic pendant la phase de déploiement initial [21].

2.3.2.2 Protocoles de gestion basés sur la pré-distribution

La méthode de pré-distribution a été proposée comme solution au problème d'établissement des clés entre les nœuds. Elle consiste à échanger les clés entre les nœuds avant leurs déploiement.

Nous allons maintenant étudier quelques protocoles de gestion de clés déterministe utilisant des mécanismes cryptographiques symétrique et avec pré-distribution de clés. Les protocoles sont classifiés selon une topologie hiérarchique.

A.les protocoles Probabilistes

L'ensemble M des clés utilisées est choisi au hasard à partir d'un ensemble de clés générées aléatoirement où $|M| >> |N|$. N étant le nombre de noeuds du réseau. Chaque capteur est pré-chargé avec un sous ensembles R de clés de l'ensemble M où $|R| < |M|$. La probabilité que

deux noeuds A et B partagent une clé K parmi les M générées est donnée par :

P = (|M| !/(|M|-m)!) * (1/|M|m)

Pour que la probabilité P soit importante ($P > 0.7$), on doit augmenter l'espace mémoire réservé aux clés du sous ensemble R. Les inconvénients sont, entre autres, l'absence d'authentification entre chaque paire de noeuds et le besoin d'augmenter l'espace mémoire réservé aux clés stockées sur le capteur pour les réseaux de grande taille. Beaucoup de travaux supposent connaitre la position des noeuds voisins (*avec une certaine probabilité*) pour réduire le nombre de clés stockées en mémoire. D'après l'étude faite , les protocoles probabilistes ont peu d'intérêt par rapport aux protocoles déterministes [14].

B.les protocoles déterministes

C'est la première solution proposée dans la littérature. La génération de clés entre deux voisins se fait d'une manière déterministe. Une clé commune est pré-chargée sur tous les capteurs avant le déploiement. Après le déploiement, chaque noeud prendra connaissance de ses voisins et générera une clé par paires avec eux et la clé commune sera effacée après cette génération. Les inconvénients sont, entre autres, le nombre élevé de messages échangés entre les noeuds pour l'établissement des liens sécurisés, et la possibilité de récupération par un attaquant de la clé maitre qui lui permet de générer toutes les autres clés du réseau [14].

B.1 LEAP : Localized Encryption and Authentication Protocol [24]

LEAP est un protocole de gestion de clés déterministe pour les RCSFs. Le mécanisme de gestion de clés fourni par LEAP supporte le traitement interne "in network processing" tout en limitant l'impact de sécurité d'un noeud compromis sur son voisinage immédiat dans le réseau. LEAP support l'établissement de quatre type de clés pour chaque noeud : clé individuelle, clé par paire, clé de groupe et clé globale.

- **Phase1 :Hypothèse de fonctionnement**
  LEAP est basé sur une clé initiale transitoire KIN chargée dans chacun des noeuds du réseau. Les auteurs de LEAP supposent que pour compromettre un noeud, l'adversaire nécessite un temps minimal Tmin : c'est le temps de brancher un câble série et le temps de copier le contenu de la mémoire du noeud compromis. LEAP exploite ce temps (de confiance) pour permettre à deux noeuds voisins d'établir, d'une manière sécurisée, une clé symétrique de session à partir de la clé initiale transitoire KIN. Après Tmin, la clé KIN est supprimée de la mémoire du noeud.
- **Phase2 :Chargement de la clé initiale**

La (SB) génère une clé initiale KIN et charge chaque noeud avec cette clé. Chaque noeud u dérive une clé principale (*Master Key*) Ku=fKIN(u), fk étant une fonction pseudo-aléatoire.

- **Phase3 :Découverte des voisins** Immédiatement après son déploiement, le noeud u essaye de découvrir ses voisins en diffusant un message HELLO qui contient son id. Aussi, il initie un timer qui sera déclenché après le temps Tmin. Le noeud u attend un ACK de chacun de ses voisins v qui contient l'identificateur de v. l'ACK est authentifié en utilisant la clé principale Kv, qui est dérivée comme suit : Kv=fKIN(v). Comme le noeud u a la clé KIN, il pourra aussi dériver Kv, ainsi il pourra vérifier l'authenticité du ACK reçus :
  u ==> *, u
  v ==> u, v | MAC(Kv, u|v)
- **phase4 :Etablissement de la clé par-paire**
  Le noeud u calcule sa clé par paire Kuv avec v, comme suit : Kuv=fKv(u).
  Le noeud v peut de même calculer Kuv de la même manière.
  Kuv sert comme clé entre u et v.

Avantage :

A la fin de ces quatre étapes, le noeud U aura établi une clé par paire partagée avec chacun de ses voisins. Cette clé sera utilisée pour sécuriser les données échangées entre eux. Aucun noeud dans le réseau ne possède la clé KIN.

Un adversaire peut écouter clandestinement tout le trafic dans cette phase, mais sans la clé KIN il ne peut injecter des informations incorrectes ou déchiffrer les messages.

Un adversaire compromettant un nœud après Tmin, obtient seulement les clés du noeud compromis.

Quand un noeud compromis est détecté, ses voisins suppriment simplement les clés qui ont été partagée avec lui.

Évolutivité (*adaptabilité, scalabilité*).Capable d'effectuer les communications de cluster.

Inconvénients :

Un peu couteux en memoire .

B.2 Lanying Li, and Xin Wang : "A high security dynamic secret key management scheme for Wireless Sensor Networks" [25]

Ce système essaye de trouver une approche de gestion de clés avec une consommation d'énergie moindre, il divise le réseau en modèles à trois niveaux comme illustré dans la (figure 2.4).

Dans le premier niveau, Le réseau est organisé avec des Nœuds Normaux (NN), le deuxième niveau avec les clusters Heads (CH) et au niveau supérieur la station de base (SB).

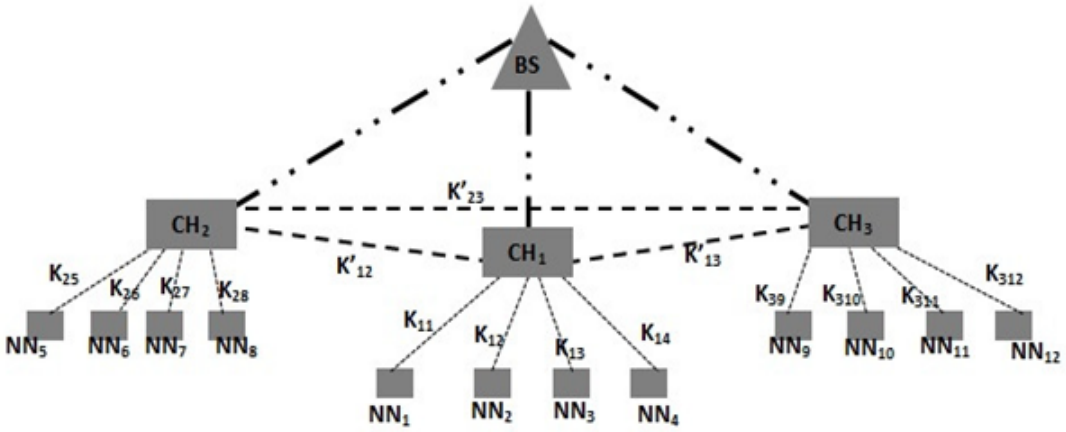

FIGURE 2.4 – Modèle du réseau [25].

Les communications dans ce réseau se font entre les nœuds normaux (NN) et leus CH correspondant, entre les CHs voisins , et entre les CHs et la SB.

- **Phase de pré-distribution**
  La SB gère le réseau comme suite :
  chaque capteur reçoit un identifiant ID unique.
  Tous les nœuds de ce réseau, y compris la SB, sont pré-chargés avec une règle uniforme .
- **Phase d'etablissement de clés par-paire**
  Cela permet d'obtenir les clés qui sont nécessaires pour les trois types de communications dans le réseau. La SB génére une matrice Ckp au hasard avec N nœuds qui est égal au nombre de colonnes de la matrice générée, et où p est le nombre de clés générées, et k est la longueur des clés nécessaires, tandis que P - N représente le nombre maximum de nouveaux capteurs qui pourraient être ajoutés à ce réseau, cette phase comprendra les étapes suivantes :

– **Etablissement de clés entre la SB et CH**
  Après avoir terminé le calcul pour la SB et le CH, ce processus commence, avec les éléments suivants cette séquence :
  Les clusters doivent être formés, puis la SB diffuse un message à tous les CHs dans le réseau, les informant pour démarrer le processus d'établissement de clé.Les CHs envoient leur ID à la station de base (SB).
  La SB utilisera les valeurs des ID qu'elle reçoit pour se référer à la matrice Ckp et obtient

la valeur de la colonne, puisque les IDs représentent les numéros de colonnes dans cette matrice, Ensuite, la SB utilisera la valeur de KM et la règle uniforme pour calculer la clés par paires.

Comme réponse de la SB, elle envoie la valeur de KM et la liste des IDs générés aux CHs correspondant.

Chaque CH vérifie si son ID est inclus dans la liste ou non.

Les CHs calculent leurs clés sur la valeur de la colonne qu'ils ont et le KM qu'ils ont obtenu de la SB, et enregistrent les clés.

- **Établissement de clés entre deux CH**

  Les CHs ont déjà une chaîne binaire qui représente un état initial utilisé pour générer les clés qui sont nécessaires pour être calculés avant ;les étapes de ce procédé sont comme suit :

  Chaque CH possède une valeur de colonne pré-chargée sur lui ;

  Les CHs voisins s'envoient leurs valeur, de sorte que chaque CH va avoir la valeur de sa colonne C_i et celle de son voisin C_j.

  Les deux CHs qui ont fait l'échange précédent calculent C_i et C_j en conséquence, et le resultat de ce calcul est utilisé comme valeur de l'état initial de la règle uniforme.

  Les CHs calculent respectivement, selon la règle uniforme et la valeur KM comme nombre d'itérations, puis enregistre les nouvelles clés.

- **Établissement de clés entre le CH et les nœuds capteurs normaux**

  Cette étape est un peu semblable à celle qui se passe entre les CHs et la SB, mais ici il faut procéder en deux étapes, une entre le nœud normal (NN) et le cluster-head (CH), et la seconde entre le cluster-head (CH) et la station de base (SB), comme suit :

  Les nœuds de capteurs normaux au sein d'un cluster envoient leur identifiant (ID) au CH correspondant.

  A son tour, le CH à ce stade passe juste la liste des IDs reçus à la SB.

  La SB interroge la matrice Ckp, et envoie ensuite la valeur de la colonne en fonction de la liste des IDs et la fixation de la valeur de KM avec elle pour le CH.

  Après avoir reçu les informations précédentes de la SB, le CH fait son calcul, et en même temps, diffuse les KMs et la liste des IDs pour les nœuds capteurs au sein du cluster.

  Les nœuds capteurs normaux vérifient si leurs IDs sont dans la liste reçue, et aussi calculent en utilisant KM et la règle uniforme leur clés, et sauvegardent le résultat.

- **Rénovation de clés**

  Cette étape consiste à ressaisir et mettre à jour les clés entières dans le réseau, et dans cette approche, cela se fait quand un CH est capturé par un adversaire. La SB est celle qui détecte que le CH a été capturé. Ainsi, lorsque la SB détecte la capture d'un CH, elle doit notifier les autres CHs dans le réseau pour rafraîchir les clés, et elle génére un nouveau nombre aléatoire en tant que valeur pour KM. Donc, l'ensemble des étapes

d'établissement de clé seront répétées, tout en utilisant la nouvelle valeur de KM et la même chaîne binaire qui a été attribué à chaque capteur.

- **Ajout et suppression de capteurs**
  Lorsque les nœuds du réseau sont déployés pour la première fois, le nouveau nœud qui va rejoindre le réseau sera pré-chargé avec les mêmes informations nécessaires, tels que l'ID de nœud qui représente le numéro de colonne dans la matrice Ckp, la valeur de la colonne et une règle uniforme. Lorsque le nœud est déployé dans le réseau, il diffuse tout d'abord son ID ; le CH le plus proche qui intercepte cet ID va le passer à la SB, qui le vérifie. Après cela, Le processus d'établissement de clés est débuté et il est comme celui qui s'est passé entre le CH et les noeuds normaux.
  La suppression de nœuds dans ce modèle de réseau se fait pour les nœuds normaux seulement, Parce qu'il a un niveau d'énergie moindre par rapport aux CHs. Donc quand la puissance du noeud normal est approximativement épuisée, il informe son CH correspondant qui, à son tour, va effacer l'ID et la clé associée à ce nœud .

B.3 LEKMP :A Low-Energy Key Management Protocol for Wireless Sensor Networks [26]

Les noeuds capteurs sont regroupés selon des centres d'intérêt et liés entre eux par des CHs. Des noeuds de commande responsables des missions de sécurité du réseau et représentent la tierce partie de confiance de tous les noeuds sont également utilisés. Dans ce schéma les noeuds d'un groupe utilisent un mode de communication direct avec le CH, lors des opérations de gestion de clés, et les CHs utilisent également le même mode et sont atteignables en 1-saut. Les noeuds et les CHs n'ont aucune connaissance à postériori de la topologie du réseau et sont aléatoirement déployés.

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

- **Distribution des clés :** chaque noeud stocke dans sa mémoire deux clés secrètes : une clé partagée avec le CH ,et une autre avec la SB. Les CHs partagent des clés entre eux et avec la SB. Les CHs peuvent stocker toutes les clés des noeuds de leurs groupes ,mais cela est moins sécurisé pour effectuer le stockage de toutes ces clés et les clés partagées avec les autres CHs. Toutes les clés sont distribuées et chargées sur les capteurs avant le déploiement, et aucune action supplémentaire de distribution, immédiate ou après le déploiement, n'est envisagée.
- **Construction des groupes :** après le déploiement, chaque noeud diffuse un message Hello de découverte de voisins contenant son ID et l'ID du CH qui contient la clé partagée. Chaque CH exécute l'algorithme de formation de groupe en se basant sur les clés partagées

avec les noeuds capteurs. A la fin de cette étape, chaque noeud reçoit une réponse du CH.

- **Révocation des clés :** si un noeud est compromis, la SB et le CH l'expulsent du groupe en ignorant les routes qui passent par lui. Si un CH est compromis, la SB l'expulse et choisit un autre CH pour le remplacer. Le nouveau CH reçoit les IDs des noeuds de son groupe ainsi que les clés partagées avec eux. Les noeuds de son nouveau groupe ainsi que les autres CHs seront informés par ce remplacement en acceptant ses messages et en ignorant les messages du CH compromis.
- **Renouvellement de clés :** La SB produit les nouvelles clés et les transmet aux CHs. Chaque CH transmet à son tour une clé pour chaque noeud de son groupe.
- **Ajout de nouveaux capteurs :** Le nouveau capteur sera pré-chargé avec deux clés secrètes. La SB transmet un message contenant l'identificateur et la clé du nouveau capteur à un CH sélectionnée au hasard. Le CH procède à l'intégration du noeud capteur après l'exécution de l'algorithme de reformation de groupes .

  - Un gain d'energie sur les opérations d'emission et reception de clés est tiré profit.

B.4 CMKMS : Schéma de gestion de clés moblies axées sur le cluster [27]

B.4.1 Methodologie utilisée

Le réseau est divisé en clusters. Le nœud avec la capacité de confiance et une efficacité maximales est sélectionné comme CH, il agit comme un gestionnaire de clé, il regroupe les informations de tous les autres nœuds du cluster. La communication entre deux noeuds capteurs est une communication intra-cluster, qui est faite en utilisant le lien SN-à-SN. La transmission entre deux CHs et entre un CH et une SB est une communication inter-cluster, qui est faite en utilisant un lien CH-à-CH OU CH-à-SB. nous supposons que les noeuds peuvent passer d'une position à une autre ,mais le CH et la BS sont fixés à une position.

Cet algorithme considère les deux scénarios suivants :

(i) scénario 1 : Le CH est statique ,et les autres noeuds capteurs sont mobiles .Le CH agit en tant que Key Manager (KM) qui gère les clés de tous les nœuds du cluster.A chaque fois qu'un nœud change de position le scénario de gestion de clés est envisagé.

(ii) scénario 2 : Le CH et les nœuds sont mobiles (*des key manager*).

Le CH devrait transférer la responsabilité de la gestion des clés aux autres noeuds dans le cluster, à savoir faire de nouveaux KM ou CH dans le réseau. Chaque fois que le CH est à la limite du cluster, il transfère les responsabilités de gestion des clés à d'autres CHs en exécutant l'algorithme de sélection de CH.

Le nouveau CH est sélectionné au hasard à l'aide d'algorithme de sélection de CH et cela en se basant sur sur les mêmes critéres.

Le mécanisme de travail est divisé en deux parties :

(i) Phase de mise en place :

Dans cette phase, un seul cluster est établi dans le réseau et les clés du cluster sont mises en place. La phase de mise en place est divisée en deux parties :

- L'organisation du réseau en clusters.
- La mise en place de clés de clusters.

Un ID unique est attribué à chaque nœud capteur qui l'identifie distinctement dans le réseau. Chaque nœud maintient deux clés : la clé Kh, qui est la clé de maison (*au sein du même cluster*) et la clé Kf, qui est une clé étrangère (*pour les nœuds se déplacant d'un cluster à un autre*) .

Les étapes de l'algorithme sont les suivantes :

- ***Etape 1 :*** Lancement du programme.
- ***Etape 2 :***Les noeuds de capteur attendent un laps de temps aléatoire.
- ***Etape3 :*** Les noeuds envoient un message HELLO.
- ***Etape 4 :***Les noeuds reçoivent le message HELLO.
- ***Etape 5 :*** Si le noeud décide le rôle , il rejette tous les messages de devenir CH ou membres en même temps et envoie le message ACK.
- ***Etape 6 :*** Rejoindre le cluster du nœud qui envoie le message.

- ***Etape 7 :*** Définir les Kh et Kf comme clé de cluster.

B.4.2 Maintien des clés

La phase de maintenance est de pouvoir maintenir et gérer les clés de maison Kh et les clés étrangères Kf au cours de la mobilité du nœud, et cela dans les différentes situations suivante :

Cas1 : Quand un nouveau noeud rejoint le groupe

- ***Étape 1 :*** Le nouveau nœud envoie un message de balise au cluster.
- ***Étape 2 :*** Les nœuds capteurs récoltant le message de balise le transmettent au CH.
- ***Étape 3 :***Le CH ajoute le nouveau nœud avec le Cluster.
- ***Étape 4 :*** Le Nouveau noeud exécute la phase de configuration du cluster (et obtient la clé de maison et le clé étrangère).

Cas 2 : Lorsqu'un noeud se déplace d'un cluster à l'autre

- ***Etape 1 :*** Le noeud quittant envoie un message de balise au CH du cluster d'accueil .
- ***Etape 2 :***Le nouveau CH informe de cela ses membres et le CH voisin.
- ***Etape 3 :*** Le CH comprend cela et communique avec le noeud entrant avec la clé étrangere.
- ***Etape 4 :*** Enfin, le nouveau noeud exécute la phase de configuration du cluster (*et obtient la clé de maison et la clé étrangère*).

Avantages :

- La gestion de la mobilité est améliorée.
- Augumentation de la durée de vie du réseau ainsi que son efficacité.

B.5 Gestion de clés basée sur un arbre M-ary [28]

Les capteurs d'un cluster sont organisés sous forme d'un arbre équilibré comme indiqué dans la figure 2.5 . Cet arbre est maintenu par un leader, qui est désigné par H noeud.

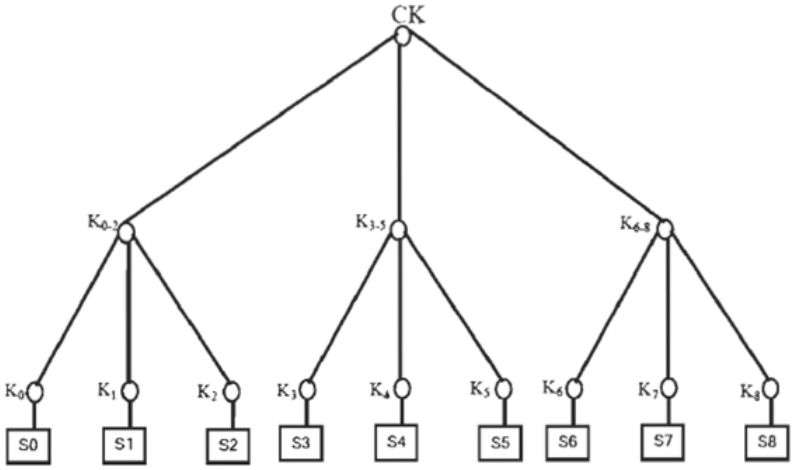

FIGURE 2.5 – Arbre M-ary des clés [28].

Les feuilles sO, s2, ... , sn représentent les noeuds dans le cluster.
Chaque noeud partage avec le leader sa clé privée utilisée pour communiquer avec le lui.
Les noeuds kO, kl ,... kn dans l'arbre correspondent aux clés privées.
Les clés kO-2, k3-5, k(n-2)-n représentent les clés qui sont partagées par un certain sous ensemble de noeuds (dites des clés intermédiaires).
CK est la clé de groupe qui est partagée par tous les noeuds dans le cluster.
Chaque capteur stockera toutes les clés le long du chemin de la feuille à la racine de l'arbre.
Tous les noeuds leaders dans le réseau forment un autre arbre qui est géré par la SB. La clé partagée par tous les leaders est dite CCHK. Elle est utilisée pour communiquer entre eux.

A.5.1 Établissement des clés

Initialement :

- Chaque capteur est pré chargé d'une clé privée avec laquelle il échange avec le leader avant le déploiement.
- Tous les leaders sont pré chargés de toutes les clés qui sont assignées aux noeuds.

Après le déploiement :

- Tous les leaders émettent le message Hello aux noeuds normaux.
- Chaque noeud normal choisit son leader LD dont le message Hello a la meilleure proportion de bruit du signal.
- Après la réception de la réponse des noeuds, chaque leader supprimera les clés des noeuds qui ne sont pas dans son cluster.
- Chaque leader construit un arbre et assigne des clés dans son arbre pour chacun des noeuds qui a répondu à son message Hello.

- Le leader distribue toutes les clés le long du chemin de la feuille à la racine de l'arbre en les chiffrant grâce aux clés privées des noeuds.
- En recevant l'ensemble des clés, les noeuds peuvent communiquer avec leur leader en utilisant la clé du cluster CK aussi bien qu'avec les autres noeuds dans le cluster en utilisant les clés intermédiaires.

B.5.2 Suppression des noeuds compromis

Aussitôt qu'un noeud est compromis, le leader changera toutes les clés le long du chemin allant de la position du noeud compromis à la racine de l' arbre. Les clés changées sont distribuées de façon sécurisée aux autres noeuds en utilisant les clés intermédiaires.

B.5.3 Ajout d'un Nouveau nœud

- Un nouveau noeud est pré-chargé d'une clé privée qu'il partage avec le leader. La SB chiffre cette clé privée en utilisant la clé CCHK, et l'envoie à tous les leaders.
- En recevant le message, chaque leader a toute l'information concernant le nouveau noeud.
- Chaque leader envoie le message **Hello** pour ajouter le nouveau noeud capteur.
- Le nouveau noeud choisit son leader LD.
- Le leader sélectionne une position pour le nouveau noeud dans l'arbre et mis a jour l'arbre .
- Le leader distribue la nouvelle clé du cluster à tous les noeuds sauf le nouveau noeud, le message est chiffré en utilisant l'ancienne clé CK.
- Le restant des clés changées sont chiffrées en utilisant la vieille clé intermédiaire respective, puis ces clés seront distribuées à tous les autres noeuds du cluster.
- Pour distribuer toutes les clés changées au nouveau noeud, le leader les chiffre avec la clé privée du nouveau noeud.

B.5.4 Rafraichissement des clés

La clé du cluster CK est changée à CK' par les leaders, et elle est distribuée en la chiffrant avec l'ancienne clé CK.

De la même façon la SB changera CCHK à CCHK' et la distribue à tous les leaders en chiffrant CCHK' avec CCHK .

Dans cette méthode, chaque sous-ensemble de noeuds a une seule clé partagée et elle est utilisée pour la communication avec les noeuds ; alors si un seul noeud est compromis, un adversaire peut écouter tous les messages entre tous les noeuds du sous-ensemble.

B.6 CBKM : Cluster based key management in wireless sensor network [29]

Dans ce protocole ,le réseau est divisé en clusters, mais pas nécessairement de même taille. Chaque cluster a un contrôleur secondaire (CH) en plus d'un dispositif de commande principal (SB), qui appartient à l'ensemble des clusters .

B.6.1 Hypothèses

Dans un cluster de n noeuds, il y a n*(n - 1)/2 paires de clés. Le CH distribue les clés entre ses

membres et la SB fait un travail semblable entre les CHs.
Le CH est dans la portée radio de son propre cluster et la SB peut atteindre tous les CHs.
La SB peut atteindre un CH dans le réseau, mais le contraire ne peut pas être vrai.
Le regroupement des neouds en clusters se fera par l'algorithme de clustering K-means++ [35].

- **B.6.2 Distribution de clés**

- **B.6.2.1 Distribution de clés entre les nœuds**

  Dans ce schéma, le problème d'anniversaire est utilisé dans la distribution de clés entre les noeuds du réseau. Selon le problème d'anniversaire, nous devons trouver un nombre de personnes avec une certaine probabilité, de sorte qu'au moins deux d'entre eux partagent un anniversaire commun.

  Soit p(n) la probabilité avec laquelle nous devons trouver le nombre de personnes n dans une chambre et d est le nombre de jours. Ainsi, selon le problème anniversaire $p(n) = 1 - n!(dn)/d^n$ l'équation peut être approximé par :

  $p(n) = 1 - (1 - 1/d)puissancec(n, 2)$ Nous pouvons établir une analogie se rapportant au scénario présent comme suit :

- 1 : Le nombre total de nœuds dans un cluster peut être comparé au nombre de jours.
- 2 : Le nombre de personnes dans une pièce peut être associé au nombre de noeuds partageant des clés avec chaque noeud dans un cluster.

Supposons que le nombre total de nœuds est de m dans le réseau et le réseau est divisé en k clusters. Choisir au hasard un cluster avec n nœuds avec une probabilité p(s), donnée par : p(s)=1-(1-1/n) puis c(s,2)
Ici, s est le nombre de noeuds avec lesquels tout noeud partage une clé.
Comme la valeur de p est en fonction de s alors chaque fois que s change de valeur p change aussi de valeur, et qui nous donnent le nombre total de noeuds où deux nœuds partagent une clé unique. On considère un noeud aléatoire i dans le cluster et on choisi aléatoirement les noeuds s qui partagent une clé unique avec le noeud i dans le cluster.
Sur n (n - 1)/ 2 clés au total, les clés des noeuds avec lesquels le noeud i partage une clé sont stockées dans la mémoire. Ce processus est répété jusqu'à ce que tous les nœuds du cluster stockent les clés.

B.6.2.2 Distribution de clés entre les sous-contrôleurs

Ce n'est pas chaque CH qui peut interagir avec tous les autres CHs, en raison de ses capacités de communication limitées. tous ces sous contrôleurs qui sont dans la portée de communication doivent être identifiés en premier. Soit L le nombre de clusters identifiés à une portée de communication de la Distance de d.

Du problème d'anniversaire, nous pouvons identifier le nombre de CHs partageant des clés entre eux. Un CH S_i doit partager des clés avec SS CHs qui sont à sa portée sur le nombre total de L CHs. La probabilité p (ss) peut être déterminée comme suit :
$P(SS) = 1 - (1 - 1/L)puissc(SS, 2)$
Ici nous obtenons la valeur de SS i.e, le nombre total de CHs avec lequel le CH S_i, partage une clé.

1. Chemin d'identification

Après la phase de distribution de clé, les chemins doivent être établis.

l'algorithme du plus court chemin de Dijkstra [36] est utilisé . Tous les poids sont supposées être égales à 1 .

Les étapes suivantes sont à suivre afin d'identifier le chemin d'un CH à l'autre.

- 1. Assigner initialement un CH et identifier tous les CHs qui peuvent interagir avec lui.
- 2. Attribuer à tous les CHs un état comme non visité.
- 3. Lorsque tous les nœuds voisins du nœud actuel sont visités, marquer le nœ ud actuel à visité.

- 4. Si par la 3ième étape tous les noeuds sont visités la recherche se termine là sinon, marquer le prochain noeud comme noeud courant et répétez le processus de l'étape 2. Ainsi , par cette procédure tous les chemins entre les différents sous contrôleurs sont identifiés et stockés dans leur mémoire respective.

2 chemin entre les nœuds

Le chemin est établi entre deux noeuds lorsque cela est nécessaire. Autrement dit, quand un message doit être transféré d'un noeud à l'autre pour la première fois, un chemin d'accès est identifié à l'aide des ensembles de clés et ce chemin est enregistré pour une utilisation prochaine. De cette façon, le chemin est identifié entre les différents nœuds et stocké dans le CH.

Les deux cas qui doivent être traités sont les suivants :

Les deux nœuds source et destination appartiennet au même groupe.

Les deux appartiennent à différent clusters.

- **Le premier cas :** Supposons qu'un chemin d'accès doit être identifié entre les noeuds i et j qui sont dans le même cluster. le nœud i envoie le message, si i et j partagent une clé, le message est diffusé directement, sinon, i le passe pour le nœud n, où n
  n | si n appartient ensemble de clés de i et quelque soit n où $d_in < d_jn$, où
  d_in est la distance entre le nœud i et n

d_jn est la distance entre le nœud j et n

Lorsque le message est diffusé à n, le noeud suivant le plus proche de j est recherché, et ce processus est répété jusqu'à ce qu'il existe un chemin direct à partir de n à j.

– **2eme cas :** quand un noeud dans un cluster doit communiquer avec un noeud d'un autre cluster, ce noeud envoie un message à son CH. En se basant sur les informations de l'autre CH, ce CH envoie un message au CH de la destination par le chemin qui a déjà été défini. Ainsi, après avoir atteint le CH du cluster de destination, il enverra un message au premier nœud de son cluster, puis le procédé tel que décrit dans le cas 1 est répété .

B.7 EECBKM : ENERGY EFFICIENT CLUSTER BASED KEY MANAGEMENT TECHNIQUE [30]

Après que les noeuds sont déployés, ils signalent d'abord à la SB leur emplacement physique et le réseau commence à sélectionner les CHs ,selon l'algorithme de sélection de CH ,chaque noeud décide si il est capable de servir de CH sur la base des critères de sélection suivants :

- Ressources énergétiques élevés.
- Gamme de communication large.
- Haute capacité de traitement.

Pour le processus d'authentification, le mécanisme de chiffrement est exploité.
Lorsque les critères de sélection sont satisfaits par un nœud, il est capable d'être le CH. Donc , ce noeud N_i diffuse un paquet balise (CH_{BEACON}). Le paquetCH_{BEACON}) est chiffré avec une clé appelée clé primaire, K_{pri} :
N_i : diffuse $K_{pri}(CH_{BEACON})$.
Lorsque les nœuds voisins de S_i reçoivent ce message, un message de réponse (CH_{REPLY}) est envoyé vers le nœud N_i par les nœuds qui ont l'intention de rejoindre le cluster. Le message de réponse contient l'ID et le contenu de la réponse Ack. Si le nombre de messages de réponse reçus par N_i est supérieure à un seuil R_{th}, alors N_i peut être sélectionnée comme le CH .
Enfin ,le CH assigne des IDs à tous ses nœuds membres qui ont l'intention de rejoindre le cluster .

B.7.1 Communication de cluster

la (figure 2.7) montre que chaque CH est connecté à la SB. Dans cette figure , le réseau possède trois clusters. Chaque cluster possède un CH à savoir, CH1, CH2 et CH3 . Le CH1 contient les éléments de 1 à 7,CH2 contient les éléments de 8 à 14 et le CH3 contient les éléments de 15 à 21 . Après que les clusters sont formés, le CH envoie à la SB : <$cluster_{id}$, $membre_{id}$> .
X1, X2 et X3 sont les informations de cluster envoyées par le CH_1, le CH_2 et le CH_3 vers la SB, donné par :

– X1 = <C1,1>, <C1,2>,... , <C1,7>
– X2 = <C2,8>, <C2,9>,..., <C2,14>
– X3 = <C3,15>, <C3,16>,..., <C3,21>

Le SB attribue une clé de cluster, KCH_i à chaque cluster dans le réseau .
Après avoir obtenu la clé de cluster de la SB, chaque CH reçoit l'ensemble de clés par paire qui est basé sur le système (EBS)[34] . Le jeu de clés EBS comprend les clés par paires, P_{ij} pour la communication entre le CH et ses membres ainsi que les clés par paires, $PH_{ii'}$ pour la communication entre les CHs, chiffrée par la clé de cluster.

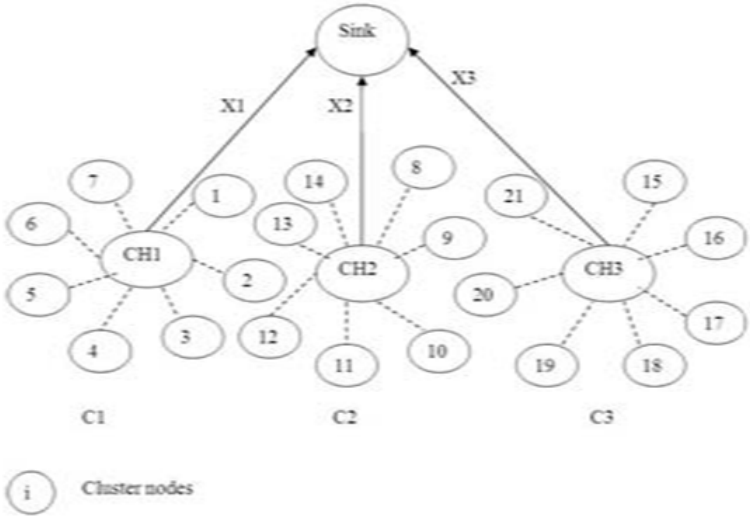

FIGURE 2.6 – Organisation du réseau.

B.7.1.1 communication intra-cluster :

Le CH déchiffre les clés appariées envoyées par la SB, avec sa clé de cluster KCH et les distribue à ses membres :

- $CH_i => CM_j$
  Où :
  - i= 1 —> j=1 à 7
  - i= 2 —> j=8 à 14
  - i= 3 —> j=15 à 21
- $CH_1 => K_{11}\ CM_1$
- $CH_1 => K_{15}\ CM_5$

Ensuite un chemin sécurisé est établi entre les deux nœuds , le nœuds 1 et le noeud 5 après l'échange de message hello et le message d'acquitement :

- $CH_1 => CM_1$: Message HELLO
- $CH_1 => CM_5$: Message HELLO
- $CM_1 => CH_1$: Message ACK
- $CM_5 => CH_1$: Message ACK

– CM_5 => CH_1 : Chemin sécurisé

Après réception du message d'accusé de réception , un canal sécurisé est établi entre le noeud et le CH . Ainsi, qu'un chemin sécurisé est établi entre les deux nœuds qui veulent communiquer entre eux.

B.7.1.2 communication inter cluster :

Si le nœud 10 de C_2 veut communiquer avec le nœud 15 de C_3, alors :
Initialement le CH_2 distribue les clés par paire K_{210} au nœud 10 et le CH_3 distribue les clés par paire K_{315} au nœuds 15, et puis un canal sécurisé est établi dans C_2 entre le CH_2 et le noeud 10, et dans C_3 entre le CH_3 et le noeud 15.
Afin d'établir un canal sécurisé entre C_2 et C_3, les étapes suivantes sont respectées :
CH_2 => K_{23} CH3
Ensuite le message HELLO est envoyé deC_2 à C_3 :

– CH_2 => CH_3 : Message HELLO
– CH_2 <= CH_3 : Message ACK

À la réception du message d' acquitement, un canal sécurisé est établi entre C_2 et C_3 :
CH_2 <==> CH_3 : lien sécurisé

A chaque fois qu'un noeud à l'intérieur d'un cluster souhaite communiquer avec un noeud appartenant à un autre groupe alors la

communication inter cluster a lieu dans le réseau. Pour la communication entre deux groupes, le CH utilise les clés par paires, $PH_{ii'}$ provenant de de l'ensemble de clés EBS [34]. Après la distribution des clés par paire entre les CHs, les canaux sécurisés sont mis en place entre les CHs. Initialement la source CHi envoie un message HELLO au $CH_{i'}$ avec lequel elle veut communiquer. A la réception du message d'acquitement de la cible $CH_{i'}$, la source CHi établit un canal entre elle-même et la cible $CH_{i'}$:

– CH_i —> $CH_{i'}$: Message HELLO

I=1, 2, 3 ; i'=1, 2, 3, et i =! i'

– CHi' —> CHi : Message ACK
– CH_2 <==> CH_3 : lien sécurisé

B.8 SKMSH : A Secure Key Management Scheme for Hierarchical WSN [31]

T.Laskar et D.Jena ont proposé un protocole de gestion de clés où le réseau est divisé en cluster. La communication se fait entre les nœuds capteurs et le CH, ainsi qu'entre les CHs et la SB.

Hypothèses

- Les nœuds capteurs ont un identifiant unique (ID).
- La SB a une table d'ID et les clés secrètes correspondantes à ces capteurs.
- La table est mise à jour chaque fois qu'un nœud entre ou quitte le réseau.
- La SB a une immense capacité d'alimentation, de stockage, de mémoire, de calcul et elle est située dans un endroit bien protégé.
- La SB dispose d'un système d'authentification qui fonctionne pour n'importe quel nœud du réseau. On suppose que l'adversaire a besoin au moins d'un temps Tcap pour capturer un nœud, puis collecter les informations à partir du nœud

Notations

- SB : station de base.
- CH_j : Jième cluster-head.
- ID_i : Identité du nœud S_i.
- Ek(Msg) : chiffré du message 'Msg' avec la clé K.
- MACk(Msg) : Le code d'authentification du message 'Msg', généré en utilisant la clé K.
- nonce : Un numéro de chaîne aléatoire.
- adv : message généré par le CH.
- data1 : Données du nœud 1.
- F(data1, data2,..., datan) : Fonction d'agrégation de données.
- Network Key (KN) : Une clé globale qui est partagée par tous les nœuds du réseau et la SB.
- Sensor Key (Ks_i) : clé Générée par la SB, pré-déployée dans chaque nœud capteur et partagée par les nœuds capteurs et la SB.
- Initial Key (K_i) : clé par-paire partagée entre le nœud capteur Si et la SB. Elle est utilisée une seule fois pour authentifier les nœuds au prés de la SB.
- Tepoch : c'est un temps qui se réfère à la période de temps qui est inférieure au temps requis pour capturer un nœud.
- || : Opération de concaténation.
- **B.8.1 Schéma de gestion de clés**
  Les nœuds utilisent une technique de chiffrement symétrique pré-déployée. Ce système se base sur l'authentification et le partage de clé secrète.

– **B.8.1.1 Phase de déploiement de nœuds**
Dans cette phase, les nœuds transmettent un message Hello à leur voisins ainsi que leur ID. les nœuds qui sont à portée de celui ci répondent avec leurs IDs. Le nœud demandeur crée sa base de données composée de table de voisins.

– **B.8.1.2 Phase d'initialisation du réseau**
Les étapes suivies dans cette phase peuvent être énoncées comme suit :
– **Etape1 :** Le nœud 'Si' envoie le paquet d'identification à la SB en insérant son ID, génére un nonce, chiffre le message et calcul la valeur de MAC en utilisant sa clé initial Ki.
S_i => **BS :** [ID_i**, nonce, Eki (**ID_i **,nonce),** $MACk_i$ **(IDi ,nonce)**]
La SB authentifie le nœud en vérifiant le cryptage (chiffré) et le MAC à l'aide de la clé initiale de ce nœud.
– **Etape 2 :** la SB génère la nouvelle clé KN du réseau, crypte avec la clé initiale Ki, et la diffuse à tous les autres nœuds dans le réseau. Ici, elle la transmet au nœud de capteur S_i.
BS => S_i **:** [Ek_i **(KN)**]
– **B.8.1.3 Phase d'initialisation de cluster** Dans cette phase, les clusters dans le réseau sont formés et authentifiés par la SB.
– **Etape1 :** Le CHj qui peut être sélectionné ou élu s'authentifie à la SB en envoyant un paquet composé de son ID et du nounce. L'ID du CH et le nonce sont aussi cryptés avec la clé du capteur CH_j : KCH_j et la valeur de MAC calculée par la clé du capteur : KCH_j. **CHj => BS :** [$IDCH_j$**, nonce, adv, E** KCH_j**.(**$IDCH_j$**,nonce,)** **,**$MACKCH_j$**(**$IDCH_j$**,nonce)**]
– **Etape2 :** la SB authentifie le CH en vérifiant la valeur chiffrée et MAC à l'aide de la clé du capteur CH_j : KCH_j.
– **Etape 3 :** la SB diffuse l'ID du Cluster-head et 'adv 'chiffré avec la clé du réseau.
BS => * : [$IDCH_j$ **, E KN (adv)**]
– **Etape 4 :** les nœuds recevant 'adv' prennent note de la puissance du signal de la diffusion reçue et les IDs de CHs correspondant. Les nœuds envoient un message d'adhésion au CH dont la puissance du signal est la plus forte. Pour l'adhésion au groupe du CH_j, le noeud capteur S_i génère un message M constitué de son propre identifiant Id_i, de l'identificateur du CH prévu $IDCH_j$ et de sa propre clé, Ks_i.
M= (Id_i || $IDCH_j$ || KS_i) Le noeud Si crypte alors le message M avec la clé du réseau KN, avec la valeur de MAC calculée en utilisant la clé du réseau KN. Le nœud Si envoie également une requête de jointure join-req avec son identifiant et celui de son cluster-head.
S_i => **CH :** [ID_i, $IDCH_j$**, nonce, join-req, EKN (M), MACKN(M)**]
A la fin de cette phase, le CH recueille toutes les clés secrètes de ses membres. Le CH génère une table comprenant les IDs de ses membres et de leurs clés secrète.

- **Etape5 :** Phase d'authentification de cluster : Le CH calcule le code d'authentification pour lui-même et de ses membres en utilisant la fonction de hachage à sens unique comme suit :
- **H1 = H (Ks1,** KCH_j**)**
- **H2 = H (Ks2, H1)**
- **.....**
- **Hn = H (Ksn,** H_{n-1}**)**

  où KCHj est la clé du CH et Ks1, Ks2 .. ... Ksn sont les clés des nœuds S1, S2, ... Sn.
  Le CH crypte ses identifiants et les valeurs de hachage avec sa clé secrète KCHj et l'envoie à la BS.
- **CHj => BS : [IDCHj, E KCHj(**$IDCH_j$ **, ID1„ ID2,... IDn...H1, H2,.... Hn)]**
- **etape1 :** les nœuds membres cryptent les paquets de données à l'aide de leur propre clé, qui est à nouveau chiffrée par la clé du réseau KN. Chaque nœud S l'envoie alors à son Cluster-Head CH_j.
- **Si => CHj : [IDi ,**$IDCH_j$**, E KN.(E** KS_i**(dataSi))]**
- **Etape 2 :** le CH décrypte toutes les données cryptées envoyées à partir de ces membres, les réunis et les cryptent avec sa clé ,puis par la clé du réseau et envoi la donnée à la SB.
  CHj => BS : [$IDCH_j$**, E KN (E** KCH_i**(F(data1 , data2, data3..., datan)]**
- **B.8.1.4 Mise à jour des clés :**
- **Etape 1 :** Pour la clé du réseau KN, la SB diffuse périodiquement une nouvelle clé de réseau NewKN en la cryptant avec la clé actuelle du réseau (KN). Les nœuds capteurs décryptent avec la clé actuelle du réseau et obtiennent la nouvelle clé.
  BS => * : [E KN (NewKN)]
- **Etape 2 :** Pour la clé Ks_i de chaque nœud capteur S_i, la SB diffuse également une nouvelle clé du capteur NewKSi chiffrée avec la clé actuelle du capteur.
  Le nœud capteur Si déchiffre avec la sa clé actuelle et obtient la nouvelle clé.
  BS => S_i **: [E** KS_i **(**$NewKS_i$**)]**
- **B.8.1.5 Phase d'ajout d'un nœud :** Les nouveaux nœuds qui entrent dans le réseau sont d'abord authentifiés et puis obtienne la clé du réseau actuel, en se basant sur sa distance par rapport à la SB, la SB diffuse également l'Id du CH le plus proche pour le nouveau nœud.
  Le nouveau nœud envoie alors au CH, une demande de jointure composée de son ID, nonce, MAC et le cryptage effectué sur ces deux valeurs en utilisant la clé du réseau actuelle.
- **B.8.1.6 Phase de Re-clustering du réseau :** Après la durée du cluster Tch, la SB diffuse un message aux cluster-heads actuels pour effacer le numéro du membre de leurs table. Le nouveau processus de formation de cluster démarre et le processus se poursuit.

B.9 CKP : A Cluster based Key Management Scheme for Underwater Wireless Sensor Networks [32]

Dans l'architecture proposée, les nœuds capteurs forment des clusters dynamiques autour de nœuds très compétents (Cluster Head). Le scénario de réseau proposé comprend également une station de base (SB) qui recueille des données directement à partir des CHs et les transmet au monde extérieur. La SB est également équipée d'un transmetteur (d'émetteur-récepteur) acoustique (*pour communiquer avec les capteurs*) et d'un transmetteur radio (*L'émetteur-récepteur de radiodiffusion*) (*pour communiquer avec le monde extérieur*). Les Nœuds capteurs sont déployés avec une courte portée de modem acoustique et chaque fois qu'ils veulent transmettre quelque chose, ils l'envoient directement à leurs CHs c'est-à-dire la communication entre les capteurs et leurs CHs est a un saut. Ainsi, la compromission d'un nœud n'affecte pas les autres nœuds.

Ainsi , afin de minimiser la réaction de réseau aux changements topologiques , utiliser les ressources énergétiques de manière efficace , augmenter la durée de vie du réseau et fournir une couverture optimale, les CHs fixe et puissants sont deployés dans les eaux peu profondes (*noeuds en eau peu profonde ont des taux de mobilité élevée en raison de diverses activités de surface de mer par rapport aux noeuds en eau profonde.*) à des endroits prédéfinis, et les nœuds mobiles sont déployés en eaux peu profonde, ainsi que dans les aux profondes.

CKP est un protocole de gestion des clés qui offre la confidentialité, l'intégrité, l'authentification, la fraîcheur et l'attaque de collusion. En outre, il répond à divers exigences de performances et de sécurité.

Hypothèses

la SB est à l'abri de tout type d'attaques. les CHs portent des informations complètes sur les clusters, alors ils sont plus sujettes à des attaques. Leur déploiement avec un matériel résistant aux falsification (*viole*) ne peut pas augmenter beaucoup le coût parce qu'ils comprennent une très petite partie du réseau. Les Nœuds capteurs ne se font pas confiance mutuellement.

CKP prend en charge trois types de clés à des fins différentes :

- **Clé du réseau :** tous les nœuds du réseau partagent cette clé .elle est utilisée pour chiffrer les messages que la BS diffuse à tous les nœuds du réseau. Pour des raisons de sécurité, cette clé doit être effacée des nœuds avant le temps minimum requis pour capturer un nœud et extraire des informations de lui.
- **Clé de groupe :** Cette clé est partagée entre le CH et ses membres. Elle est utilisé pour sécuriser des messages de multidiffusion que le CH envoie à ses membres. Les nœuds utilisent cette clé pour chiffrer les messages qui ne porte pas des informations extrêmement

sensibles pour, par exemple,les messages de requête de jointure.

– **Clé par paires :** Plus tard, la SB envoie ces clés de capteurs à leur CH respectif. Chaque capteur chiffre les lectures/informations sensibles avec sa paire de clé et l'envoie au CH. Ce phénomène ne permet pas à un nœud compromis de récupérer des informations à partir d'autres nœuds et limite l'impact d'une capture de nœud à lui-même.

Notations

– **Nonce :** est une chaîne aléatoire utilisée pour obtenir la fraîcheur.
– CH_i /S_i **:** désigne le ième Cluster Head/noeud .
– **idCHi/**idS_i **:** désigne l'id du ième Cluster-Head/noeud .
– **MACk(Msg) :** est le Code d'authentification du message Msg avec la clé de chiffrement k.
– **Ek(Msg) :** est le chiffré du message Msg avec la clé de chiffrement k.
– **idlistsensors/idlistauthentic-sensors :** désigne une liste qui comprend des ids de capteurs/capteurs authentiques.
– **listpairwise-keys :** désigne une liste de clés par paires correspondant aux capteurs authentiques.
– c_i **:** Désigne un compteur initialisé à une valeur aléatoire par la SB pour le ièm CH.
– x_i **:** désigne un compteur initialisé à une valeur aléatoire par le ième CH pour ses membres.
– $Slots_i$ **:** représente un intervalle de temps assigné au ième capteur.

A.9.1.1 Description du protocole

- **Phase de génération et distribution de clés :** la SB génère la clé globale, elle précharge chaque noeud avec un id unique, une clé globale et une clé par paires. Elle assigne à chaque CH une clé de groupe unique.
- **Phase d'installation (configuration) du cluster :** Chaque CH diffuse périodiquement un message chiffré avec la clé globale (*parce que l'énergie n'est pas un problème majeur pour les CHs*) qui comprend l'id du CH, la clé de groupe, le nonce (pour la fraicheur) et le MAC générée à l'aide de la clé globale (*utilisée à des fins d'authentification et d'intégrité*). Tous les nœuds décryptent les messages envoyés par le CH pour récupérer les clés de groupe et suppriment immédiatement leur clé globale. Ensuite , les nœuds choisissent leur plus proche CH en se basant sur la force des signaux reçus et envoient un message de requête de jointure chiffré avec la clé de goupe du CH élu. Les identificateurs de nœuds sont envoyés au cours du message de requête de jointure, les nœuds envoient leurs ids sous forme cryptée pour empêcher les ids de capteurs authentiques contre l'espionnage. Autrement, un attaquant peut insérer un nouveau capteur dans le réseau et se le faire passer avec l'id du capteur authentique. Pour assurer la fraîcheur , et réduire la taille des messages, le capteur intègre le nonce envoyé précédemment dans le MAC .Les CHs forment une liste contenant les ids des capteurs qui ont envoyé un message de requête et

l'envoi à la SB sous une forme cryptée. Si la SB détecte des ID de capteur malveillants alors elle les supprime à partir de idlistsensors (*liste des ids de capteurs*). La SB envoie la liste des n ids de capteur authentique (n < = m) avec leurs clés préchargées, un compteur (*ci est une valeur aléatoire qui est unique pour chaque CH*) et MAC au CH. Le compteur ci est utilisé pour garantir la fraîcheur des messages entre le CH et la SB et sa valeur est incrémenté de 1 a chaque fois que le CH envoie un paquet à la SB. Le CH diffuse un message de réponse de jointure à ses capteurs authentiques qui comprend les ids de capteurs , leur ordonnancement et X_i (*éviter une attaque par répétition parmi les capteurs et leurs CH*) . xi est une valeur aléatoire générée par le CH. Ce message comprend divers sous-messages. Chaque sous-message est crypté avec une paire de clé séparée. Le CH assigne des slots et les membres (ids) du cluster envoi la donnée dans le slot attribué pour éviter les collisions et minimiser la communication/énergie.

- **Phase de collecte de données :** puisque nous travaillons sur des applications de surveillance de la qualité de l'eau et les noeuds à proximité mesurent des valeurs similaires donc au lieu d'envoyer la valeur redondante dans une bande passante limitée ,les nœuds du UWSN restent en état de sommeil pour la plupart de la période et se réveille à des intervalles déterminés pour recueillir des informations et envoient au CH par tour en utilisant TDMA. Cette faible opération de cycle de tâche consomme moins d'énergie et améliore la durée de vie du réseau considérablement. Dans un premier temps , a est 0 et sa valeur est incrémenté à chaque fois qu'un capteur envoie un message au CH .
- **Phase de transfert de données :** Lors de la réception des données à partir de ses membres, le CH effectue le transfert sélectif. Ils déterminent si la valeur reçue est dans la gamme (*portée, standard ou non*). Si la valeur reçue n'est pas dans la gamme standard alors le CH attend une réponse de deux membres en plus. Si chacun d'entre eux envoient des valeurs dévié De la gamme standard alors il traite et agrège leurs valeurs pour récupérer les informations significative et transmet immédiatement à la SB. comme pour a, b est initialisé à 0 et est incrémenté à chaque fois que le CH envoie un message à la SB. Si la valeur reçue est dans la gamme standard ,les CHs ne transmettent aucune donnée. Cette méthode utilise efficacement les ressources limitées du UWSN et améliore la durée de vie du réseau.
- **Ajout d'un nœd :** Pendant toute la durée de vie du RCSF il peut être nécessaire d'ajouter quelques nouveaux nœuds dans le réseau. Ici, nous présentons la façon d'ajouter de nouveaux nœuds dans le réseau existant . Les nouveaux nœuds sont déployé aléatoirement dans le réseau et ignorant leur CHs . la SB pré-charge le nouveau noeud avec une paire de clé. Le Nouveau nœud détermine son CH le plus proche en fonction des émissions périodiques des CHs et envoie un message de demande de jointure à son plus proche CH cryptée avec sa Paire de clé. S'il est authentique, la SB envoie la paire de clé du nouveau noeud et un nouveau compteur au CH. Maintenant, le CH décrypte l'id du nœud transi-

toire à partir du message envoyé précédemment et envoi un message qui est composé de la nouvelle valeur de temps (slot) et x_i. Ce message sert également d'accusé de réception à la demande de jointure pour le capteur récemment ajouté. Maintenant, le nouveau nœud peut transmettre de manière sécurisée des données à son CH élu.

- **Suppression d'un nœud :** Chaque fois qu'un nœud est supprimé en raison d'un comportement malveillant ou de batterie en général, le CH supprime son identifiant et sa paire de clé et informe la SB. Le SB supprime l'id du noeud de sa base de données ou l'assigne à de nouveaux noeuds.
- **Transition de nœud :** Chaque fois qu'un nœud fait une transition d'un groupe à l'autre en raison de la faible force du signal il notifie son CH actuel sur son départ. Le CH actuel supprime l'id du nœud et la paire de clé de sa base de données et transmet ses informations de départ à la SB. Le nœud partant transmet un message de jointure à son CH le plus proche d'une manière similaire à celle de l'ajout d'un nœud.

2.4 Comparaison

Des métriques sont employées pour comparer les différents protocoles de gestion des clés, ces métriques sont :

- **Efficacité :** les limitations de mémoire, les communications, et le traitement des noeuds doivent être considérées,
  1. Complexité en mémoire : quantité de mémoire nécessaire pour enregistrer les clés.
  2. Complexité en communication : nombre de messages échangés pour la gestion des clés.
  3. Complexité en traitement : quantité de cycles de processeur nécessaires pour établir une clé.
- **Connectivité en terme de clé 'key connectivity' :** probabilité que deux noeuds (ou plus) partagent une clé.
- **Scalability :**cette métrique consiste en qualité d'être flexible avec la taille du réseau même après le déploiement de noeuds.
  Il est intéressant qu'un réseau ait une bonne capacité de "scalability" par ce que ceci implique que le réseau peut être facilement augmenté et qu'il n'y a aucun problème quand nous voulons ajouter un nouveau noeud au réseau. D'un autre côté, il est intéressant que le réseau puisse supporter un grand nombre de noeuds.
- **Résilience contre la capture de noeud :** ou résistance contre la capture de noeud, cette métrique mesure comment le RCSF est compromis quand un noeud est compromis, et l'influence de ce noeud sur la sécurité du réseau [19].

Conclusion

Pour conclure ce chapitre, la gestion des clés est l'un des secteurs les plus importants dans la

sécurité des RCSFs, beaucoup de travaux ont été effectués afin d'avoir un schéma performant qui assure un niveau élevé de sécurité et optimise les métriques de performances et conserve l'énergie.

Dans ce chapitre, nous avons présenté un état de l'art sur la sécurité dans les RcSF. Ou nous avons etudier quelques protocoles de gestion de clés se basant sur les clusters. Dans ce qui suit, nous allons proposer un systeme de gestion de clé dans un réseau devisé en clusters.

CHAPITRE 3

PROPOSITION ET SIMULATION

Introduction

Les RCSFs sont considérés comme des réseaux ad hoc sans fil sans infrastructure fixe [3]. Les nœuds doivent donc collaborer pour organiser l'échange d'informations de contrôle et permettre l'acheminement du trafic. Ces réseaux doivent posséder la capacité de s'auto-organiser, sans intervention humaine.

Plusieurs travaux préalables, notamment ont montrés que toute architecture de communication dans un RCSF se basant sur une topologie plate (communication multi sauts sans clusterisation) entrainera une dégradation significative de ce réseau, voire même un échec de communication et de surveillance au sein d'un réseau à large échelle. Pour cette raison plusieurs travaux ont porté ou portent toujours sur le problème de clustering avec un mécanisme de sécurité au sein des groupes. Nous allons offrir avant tout une organisation de réseau en clusters, où le cluster-head de chaque groupe est chargé de gérer les nœuds membres de son groupe. Par la suite nous proposons un mécanisme de gestion de clés dans ce réseau.

3.1 Protocole proposé

Dans ce qui suit, nous proposons un algorithme qui devise le réseau en clusters inspiré de [14],et un schéma de gestion de clés hybride ,basé sur la cryptographie symetrique et asymétrique pour sécuriser les communications. Notre objectif principal est de sécuriser le processus de transfert des données vers la SB, l'utilisation de la cryptographie asymétrique nous permet de chiffrer les paquets échangés entre la SB et les CHs, et la cryptographie symétrique pour chiffré les paquets entre les CHs et leurs membres. Dans notre approche nous utilisons deux types de clés :

- **Une paire de clés publique :**(Ksb,Ksb^{-1})de la SB où la clé publique Ksb est connue

par tout les noeuds du réseau.

- **Une paire de clé publiques :** (KCH_i,$KCHi^{-1}$) du ième CH, .
- **Une clé symétrique :** Ks_{ij} la clé privée du ième noeud du jième cluster .

3.1.1 Hypothèses :

Notre schéma se base sur les hypothèses suivantes :

- Le réseau de capteur est statique (*Les nœuds ne sont pas mobiles*).
- Les nœuds capteurs sont homogènes : les nœuds capteurs sont similaires dans leur capacité de traitement, de communication, d'énergie et de stockage.
- Le déploiement est aléatoire.
- Le réseau est organisé en cluster a l'aide d'un algorithme de clustering .
- Un attaquant peut écouter tout le trafic, renvoyer d'anciens messages, ou injecter ses propres messages.
- La compromission d'un noeud implique que toutes les informations stockées dans sa mémoire sont connues par l'attaquant.
- La station de base n'a pas de contraintes sur les capacités de calcul, de stockage, d'énergie et ne peut être compromise.
- La génération des clés asymétriques se fait avec les courbes élliptiques.
- Les canaux de communications sont bidirectionnels, si un nœud u peut recevoir un message du nœud v alors u peut envoyer un message à v.
- La SB possède une paire de clé (Ksb,Ksb^{-1}).
- Chaque capteur est pré-chargé avec un ID unique et la clé publique de la SB Ksb avant le déploiement.

3.1.2 Notations :

- S_i : L'i-ème noeud capteur dans le réseau.
- SB : La station de base .
- CH_i : L'i-ème Cluster Head dans le réseau.
- IDS_i : Identifiant du noeud S_i dans le réseau.
- ChS_i : La charge résiduelle du ième nœud
- DSi : Le degré de connectivité du ième nœud .
- PS_i : le poids du ieme noeud .
- KS_{ij} : la clé du ieme noeud S du jiéme CH.
- KCH_i :la clé publique du ième cluster head.
- KCH^{i-1} :la clé privée du ième cluster head.
- Ksb : la clé publique de la SB.
- Ksb^{-1} : la clé privé de la SB.

3.1.3 Phase de pré-distribution :

La station de base effectue les opérations suivantes :

- Charger chaque capteur avec les informations suivantes : ID, K_{sb}.
- Elle diffuse un message pour initier l'algorithme de clustering .

3.1.4 L'algorithme de clustering

Après un déploiement aléatoire des nœuds, le réseau suit un algorithme de clustering. Cet algorithme permet de former des clusters à un seul saut, où chaque membre est voisin direct de son CH. Il considère une phase de formation des Clusters. Pendant cette phase, les nœuds procèdent à la connaissance de leurs voisins et déroulent entre eux l'algorithme de formation des clusters. Cet algorithme se base sur l'approche multicritères d'aide à la décision pour le choix des CHs, les critères sont : la charge résiduelle et le degré de connectivité. Cet algorithme associe un poids à chaque nœud. Ce poids est représenté par une somme pondérée des différentes métriques impliquées dans son calcul comme montré dans l'équation :
$\sum_1^2(w_i * p_i)$ avec wi > 0.
où : Où W_i sont les coefficients de chaque critère.

La charge résiduelle et le degré de connectivité doivent être maximale puisque le CH a plusieurs tâches à effectués et plus de clés à stocker.
Le degré de connectivité représente le nombre de voisin d'un nœud .

3.1.4.1 Les étapes de l'algorithme de clustering

L'algorithme introduit la notion de poids pour la sélection des Cluster-head(s).

Entrée :

- (S_1,S_2, ...,S_n) l'ensemble des nœuds dans le réseau, (n : nombre de nœuds dans le réseaux).
- Déploiement aléatoire des nœuds.

Sortie :

- L'organisation de réseaux en clusters.
- L'élection des Cluster-head(s).

1) Chaque nœud S_i envoie des messages Hello et son ID afin de définir ses voisins.
2) Chaque nœud s calcule ses métriques (critères) qui sont les suivantes : Chs, Ds.
- ChS_i : La charge résiduelle du nœud S_i.
- DS_i : Le degré de connectivité du nœud S_i.
3) Chaque nœud Si calcule son poids selon la méthode de sommes pondérée :
Ps = W_1***Chs**+W_2***Ds**.
Puisque l'objectif est d'élire les nœuds avec les meilleurs capacités comme CHs nous prenons

des coefficients élevés pour la charge, comme suit :
Le coefficient pour Chs : W_1=1.
Le coefficient pour Ds : W_2=0.7.
4) Chaque nœud S_i envoie un message contenant son poids et son ID à ses voisins .
5) Chaque nœud S_i choisit parmi ses voisins le nœud qui a le plus grand poids (maximal) comme CH en envoyant un message de jointure contenant son ID et celui du CH .

- Si on a plusieurs nœuds qui ont le même poids maximal le CH sera le nœud qui a les meilleurs critères selon leurs importances : Chv puis Ds ,sinon si tous les critères des nœuds sont égaux, le choix est aléatoire.
- Si parmi les voisins d'un nœud S_i on a le nœud j qui a le poids max et qui appartient à un autre cluster alors on choisit le nœud k avec le poids max suivant ($P_j < P_i$) et ainsi de suite sinon le nœud S_i deviendra un CH.
- Si le nœud S_i est isolé (n'a aucun voisin) il deviendra un CH.

6) Chaque CH envoie à la SB la liste des identifiants de ses membres .

3.1.4.2 Maintenance des clusters

Il existe deux situations qui nécessitent la maintenance des clusters, et qui sont :

- **Suppression d'un nœud s' après le clustering**
  si le noeud est compromis ou sa charge est épuiséealors :
  - Si le nœud s' est un membre d'un cluster alors il va être supprimé de ce cluster et le CH correspondant informe la SB.
  - Si le nœud s' est un CH alors le clustering sera répété.
- **Changement de la structure des clusters après chaque round**
  Pour prolonger la durée de vie du réseau et pour équilibrer les charges énergitiques entre les differents noeuds nous répètons le processus de clustering à chaque round.

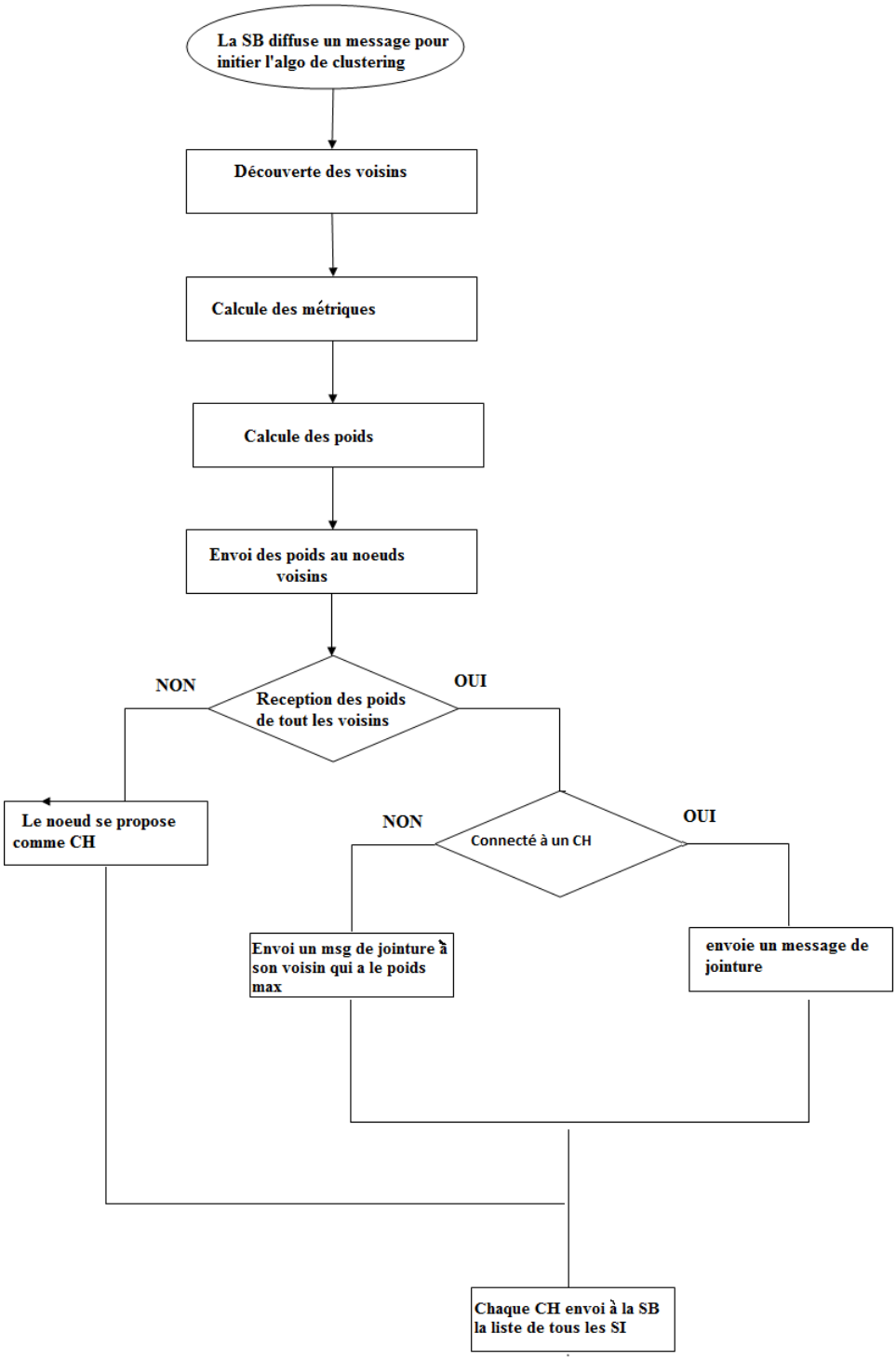

FIGURE 3.1 – Organigramme de la phase de clustering.

3.1.5 Etablissement et distribution des clés

Après l'étape de clustering le processus d'établissement de clés commence sachant que la SB contient tout les identifiants de tous les nœuds de chaque cluster. Nous supposons que pour compromettre un noeud, l'adversaire nécessite un temps minimal T_{min} :c'est le temps necessaire de brancher un cable série et le temps de copier le contenu de la memoire du noeud compromis, nous exploitons ce temps pour permettre l'etablissement et la distribution des clés dans le réseau . Nous supposons que chaque nœud interagit seulement avec son CH et il n'y a pas d'interaction directe avec les nœuds membres. Nous supposons aussi que la SB peut atteindre tous les CHs et vis-versa, donc la distribution des clés devrait être envisagée dans deux cas : entre les nœuds normaux et leurs CH, entre la SB et les CHs.
La SB génère et envoie une paire de clé (*asymétrique*) pour chacun CH_i et une clé privée pour chaqu'un de leur membre S_i le tout chiffrés avec sa clé privée Ksb^{-1} .
SB $=>$ CH_i **:** Ksb^{-1} [$IDCH_i$,(KCH_i,KCh_i^{-1}) ,Ksb^{-1}(IDS_{ij},KS_{ij})]
Chaque CH déchiffre le message et récupère sa paire de clé ,il diffuse le reste du message à ses noeuds membres et sa clé publique aux clusters heads voisins. Chaque noeud membre récupère sa clé privée KS_{ij} lui correspendant. Tous les nœuds membres S_i suppriment la clé Ksb de leur mémoire pour des raisons de sécurité.

3.1.5.1 Maintenance des clés

- **Suppression d'un nœud normal :** Dans ce cas ce nœud sera supprimé du cluster auquel il appartient, son CH corrependant supprime sa clé et son ID de sa liste et informe la SB de ce changement, cette dernière le supprime de sa liste aussi.
- **Supression d'un CH :** Dans ce cas la SB supprime le CH de sa base de données et initie de nouveau l'algorithme de clustering et tout le processus sera répété pour le cluster corespendant.

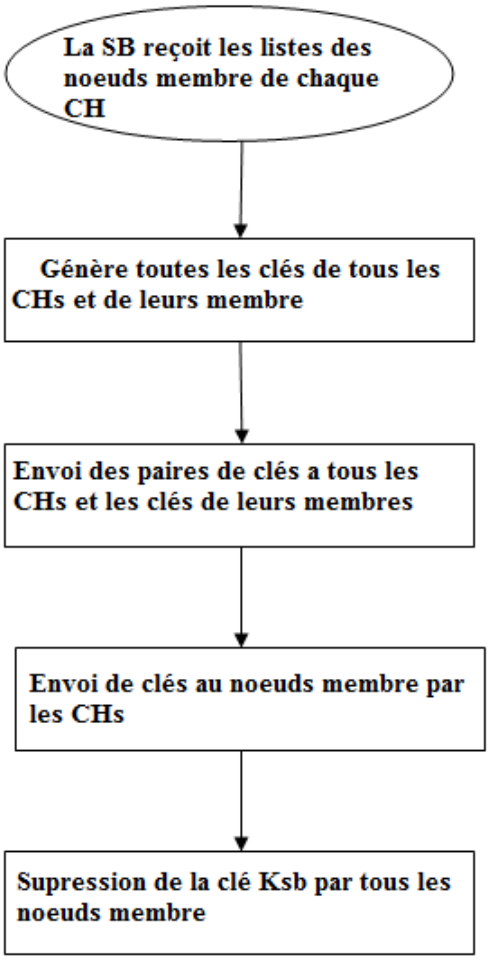

FIGURE 3.2 – Organigramme de la phase de génération et distribution de clés.

3.1.6 Communications

Après avoir terminé avec la phase d'établissement et de distribution de clés nous pouvons établir des communications sécurisées dans le réseau.

3.1.6.1 Communication intra-cluster

S_i et S_j appartiennent au même cluster :

- S_i veut communiquer avec le nœud S_j il envoi le message chiffré avec sa clé KS_{ij} à son

CH correspondant celui-ci déchiffre le message et le chiffre avec la clé qu'il piartage avec S_j.
S_i=>$CH_{i'}$: ($KS_{ii'}$[**Msg**],IDS_j, IDS_i)
$CH_{i'}$=>S_j :($KS_{ji'}$[**Msg**], IDS_i, IDS_j).

- Si le nœud S_i veut communiquer avec son CH_k alors il envoi un message chiffré avec sa clé KS_{ik} au CH_k.
  S_i => CH_k : [IDS_i , KS_{ik}(**Msg**)].
  Le CH_k décrypte le message reçu avec la clé KS_{ik} qu'il partage avec le CH_k .

3.1.6.2 Communication inter-cluster

- Si le nœud S_i appartenant au CH_k souhaite communiquer avec le nœud S_j appartenant au CH_v alors le nœud S_i chiffre son message avec la clé KS_{ik} et l'envoie au CH_k.
  S_i => CH_k : [IDS_i , $IDCH_k$, IDS_j, KS_{ik}(**Msg**)].
  Le CH_k déchiffre le message, et le chiffre avec la clé publique de la SB (Ksb) et l'envoie à la SB.
  CH_k => **SB** : [$IDCH_k$,IDS_i, IDS_j, K_{sb}(**Msg**)].
  La SB déchiffre le message reçu avec sa clé privée, le chiffre avec la clé publique de CH_v et l'envoie à celui-ci .
  SB=>CH_v : [IDS_i, IDS_j,KCH_v(**Msg**)].
  Le CH_v déchiffre le message avec sa clé privée et l'envoie au noeud de destinataire chiffré avec la clé privé du noeud.
  CH_v => S_j :[IDS_i, IDS_j, KS_{jv}(**Msg**)].
- Si le CH_i veut communiquer avec le CH_j alors il chiffre son message avec la clé publique de CH_j (KCH_j), si ils sont à portée, et l'envoi au CH_j, ce dernier décrypte le message avec sa clé privée.
  CH_i =>CH_j : [$IDCH_i$, $IDCH_j$, KCH_j(**Msg**)].
  Aprés la récolte de toutes les données envoyées à partir de ses membres, le CH décrypte toutes ces données les réunis et les crypte avec la clé publique de la SB ,cette dérniere déchiffre avec sa clé privée.
  CH_j=> **SB** : [$IDCH_j$, K_{sb} (***data1 , data2, data3..., datan***)].

3.2 Simulation

Afin d'évaluer les performances de notre mécanisme de sécurité, nous avons simulé son fonctionnement à l'aide de MATLAB ,nous avons simuler les déploiements aléatoires des RCSFs, La figure (3.3) illustre un réseau de 100 noeuds capteurs déployés aléatoirement dans une surface de 100 * 100 mètres, avec une énergie initiale de 1 Joule pour chaque noeud .

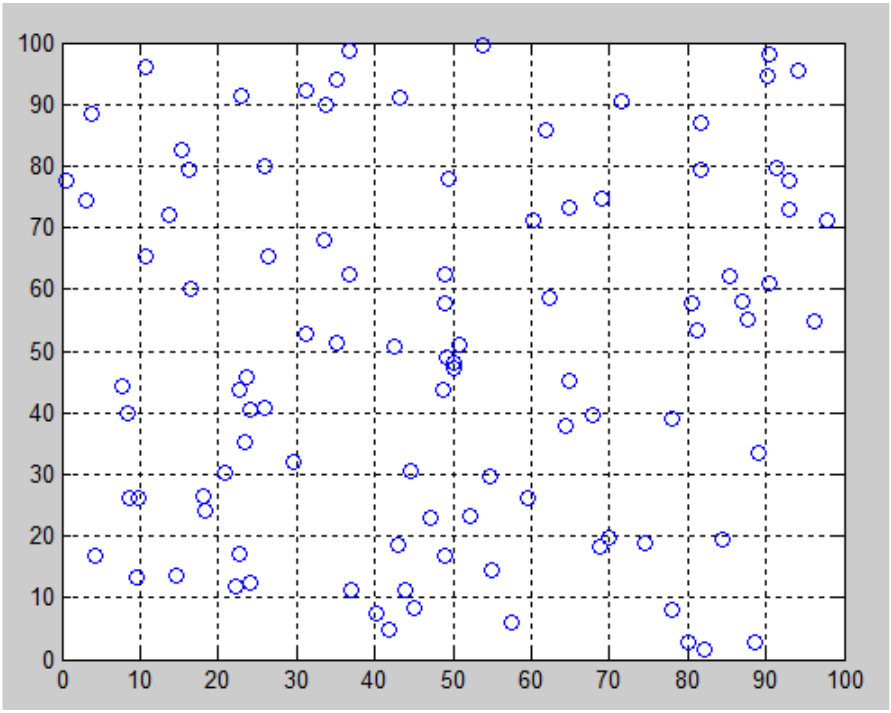

FIGURE 3.3 – Déploiement aléatoire de 100 noeuds.

Chaque noeud du réseau a une portée de signal de 20 mètres.

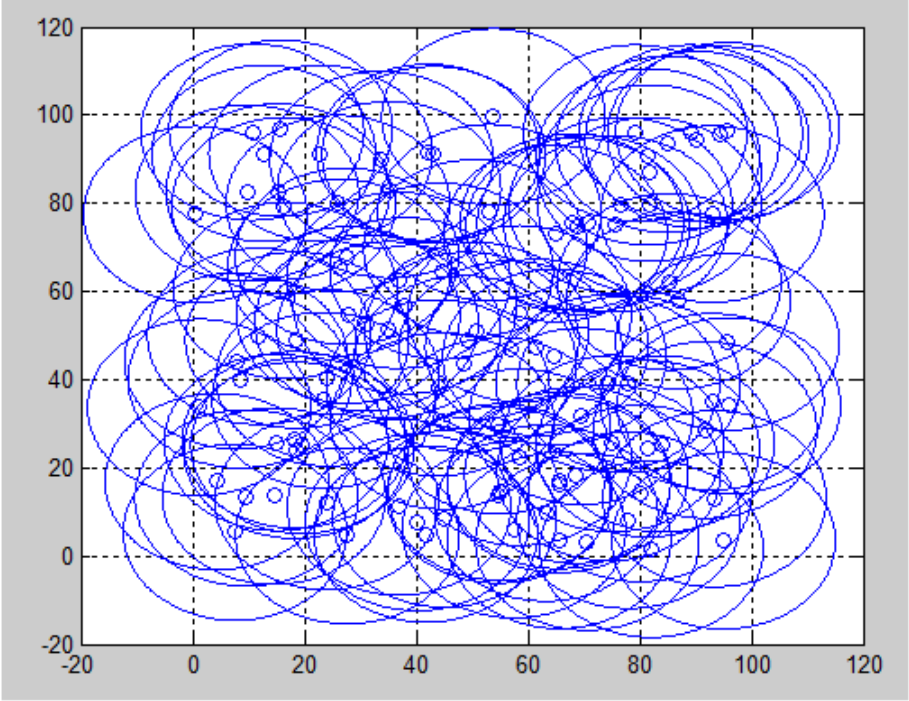

FIGURE 3.4 – Portée des capteurs.

L'algorithme de clustering proposé permet de formés des cluster à un seul saut, comme

illustré sur la Figure 3.5. Nous joignons à l'aide d'une ligne les noeuds memebres avec leurs CH.

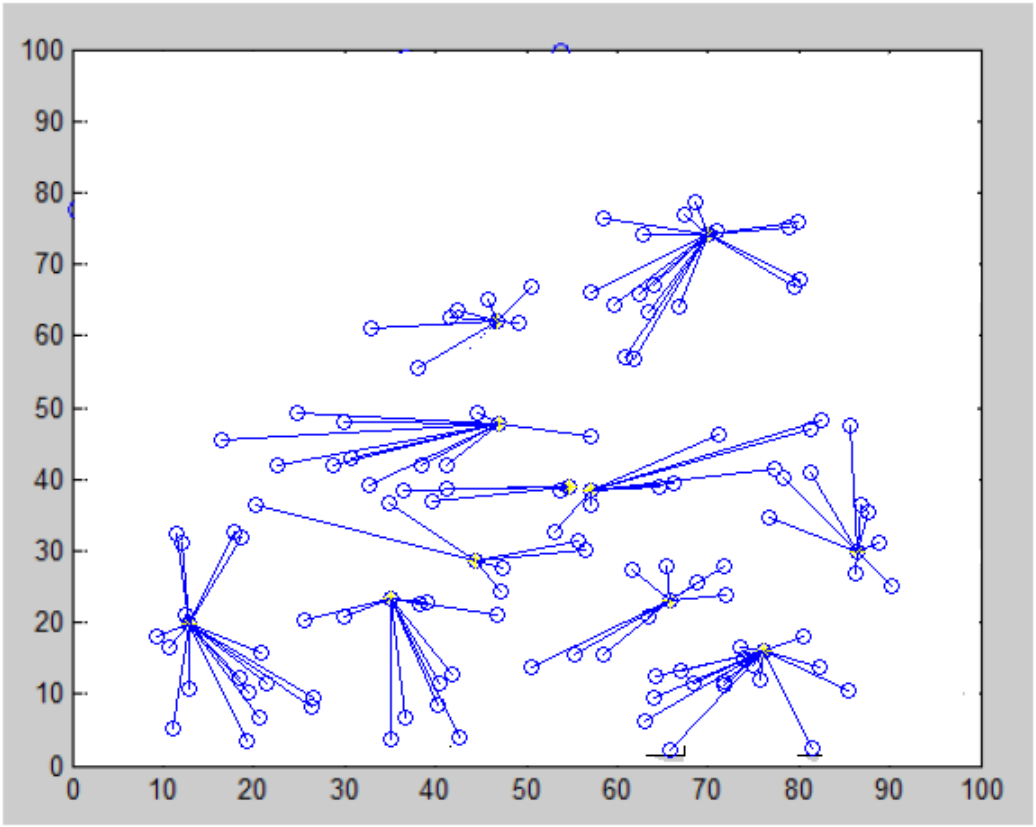

FIGURE 3.5 – Formation de cluster.

3.3 Analyse des performences

Les métriques auquel nous nous sommes intéressés sont :
- Le nombre de clés stockées sur chaque nœud.

Dans le tableau 3.6 nous avons calculé le nombre de clés stockées par chaque CH et chaque noeud normal (noeuds membre). Soit :

- **M :**le nombre de CH sans le réseau
- **N :** le nombre de noeuds normaux dans le réseau
- **NC :** le nombre de noeuds normaux dans le cluster
- $NVCH_i$ **:** le nombre de CH voisin du Ch_i

Nœud	Nombre de clés stockées sur chaque nœud
CHi	NC(Ksi)+NVCH(Kchj)+KPBsb+Kchi
Nœuds membre (normal)	Ksi

FIGURE 3.6 – Nombre de clés stockées sur chaque capteurs.

Tel que :

- Ks_i **:** est la clé privée du nœud membre s_i appartenant au Ch_i.
- **Kch :**la paire de clé public du CHj voisin du Ch_i.
- **KPBsb :** est la clé public de la SB.
- **Kchi :** est la paire de clé public du Ch_i.

Conclusion

La sécurité permet d'utiliser les RCSFs avec confiance. Sans sécurité, l'utilisation des RCSFs dans n'importe quel domaine d'application aurait des conséquences indésirables. Etablir une communication sécurisée implique l'établissement et la distribution des clés pour crypter et authentifier les messages. La gestion des clés est le problème le plus délicat de la cryptographie. Notre contribution fournit un mécanisme de distribution de clés sécurisé et efficace, permettant un simple établissement de clés pour les RCSFs.

CONLUSION GÉNÉRALE

Le travail consigné dans ce mémoire a été le fruit d'une étude menée dans le contexte des RCSFs en particulier et ce, relativement au problème de sécurité.
Nous avons étudié les caractéristiques essentielles et les notions fondamentales des réseaux de capteurs sans fil. Nous avons étudié plus particulièrement les notions de sécurité et d'énergie par la définition complète des contraintes, des besoins, des défis et des moyens de chacune d'elles mis à la disposition des noeuds capteurs du réseau pour un acheminement correcte, sécurisé et économe en énergie.
Nous avons étudier plusieurs protocoles de gestion de clés et nous avons mis une classification. Nous nous sommes intéressé très particulièrement par les protocoles déterministes basé sur la cryptographie symétrique et avec une pré-distribution et cela dans le but d'achever l'établissement de clés entre les entités communicantes dans le réseau. Après avoir étudier les solutions précédentes et les avoir critiqué, nous avons constaté que le défi dans la conception des schémas de gestion de clés est de trouver un compromis entre un système efficace et les contraintes caractérisant les RCSFs.
De cette étude, résulte notre contribution consistant en une proposition d'une solution qui permet :

- Diviser le réseau en clusters.
- Gérer la génération et la distribution des clés dans le réseau.

Nous avons trouvé un compromis entre le niveau de sécurité et le respect des contraintes posées par les RCSFs.

Concevoir un protocole efficace de gestion de clés demeure encore un domaine de recherche ouvert. Il serait donc plausible, comme perspective de notre travail, d'adapter notre proposition à une mobilité des noeuds (*pour une meilleure durée de vie du réseau*), et le comparé à d'autres protocoles proposés dans la litérature pour montrer son efficacité.

BIBLIOGRAPHIE

[1] O.Mawloud , Conception et implémentation d'un système de gestion de clés pour les réseaux de capteurs sans fil . Mémoire ingénieur d'état en génie informatique Université A. Mira de Bejaia, 2011.

[2] M. Mohamed Lamine, sécurité dans les réseaux de capteurs sans fil , mémoire de magister université A. Mira Bejaia, 2008.

[3] F.Abdefatah. développement d'une bibliothèque de capteurs, Rapport de recherche université monpellier2, 2008.

[4] I. F. Akyildiz, W. Su, Y. Sankara subramaniam, E. l. Cayirci. A survey on sensor networks. IEEE Communications Magazine, Vol. 40, No. 8, pp. 102-116, Aout 2002.

[5] G.Chalhoub , les réseaux de capteurs sans fil support de cours , université d'Auvergne, 2015.

[6] Y.Wang , G.Attebury et B.Ramamurthy, Surveu of Security Issues in Wireless , 2005.

[7] R.Mohamed, Problème de sécurité dans les réseaux de capteurs avec prise en charge de l'énergie Mémoire de Magister, uuniversité de SAAD DAHLAB DE BLIDA, 2013.

[8] Y.Rromdhan, Evaluationde performances des prtocoles SMAC et Directed Diffusion dans les réseaux de capteurs .Rapport de projet fin d'étude école supérieure des communications de tunnis , 2007.

[9] C. T. Kone, Conception de l'architecture d'un réseau de capteurs sans fil de grande dimension , thèse de doctorat, Université Henri Poincaré - Nancy I, 2011.

[10] S.Harchi, Un protocole de session dans les réseaux de capteurs sans Fil , thèse de doctorat, Université de Lorraine, 2013.

[11] I. G. Shayeb, A. H. Hussein et A. B. Nasoura, A Survey of Clustering Schemes for Mobile Ad-Hoc Network (MANET), American Journal of Scientic Research, 2011 .

[12] M.BA, Vers une structuration auto-stabilisante des réseaux ad hoc : cas des réseaux de capteurs sans fil , thèse de doctorat, Université de Reims Chanpagne-Ardenne, 2014.

[13] G.Labouret, Introduction à la cryptographie, Supports de cours, Cabinet Hervé Schauer Consultants-HSC, 09 Février 2001.

[14] B.Nadia, A.Nacira, Approche Décentralisée pour la sécurité d'un Réseau de Capteurs Sans Fil (RCSF), Memoire d'ingérieur d'etat en informatique, université de Bechar, 2010.

[15] N. Merani, N.Khimoun. Simulation et évaluation de protocoles de gestion de clés dans les réseaux de capteurs. Memoire d'ingénieur d'etat en informatique.Bejaia 2009.

[16] Z.Benchabane, K.Oussalah, Conception et implémentation d'un systeme de gestion de clés pour les reseaux de capteurs sans fil, Memoire d'ingérieur d'ingérieur d'etat en génie informatique, Université de Bejaia 2011.

[17] Y.Challal, Réseaux de capteurs sans fil, Système intelligents pour de transfert, Université de Technologie de Compiègne, Heudiasuc, France.17/11/2008.

[18] N.Lasla, La gestion de clés dans les réseaux de capteurs sans fil, Memoire de magister, Institut national en Informatique (I.N.I) Ouad-Smar, Alger.2007

[20] S.Atmani, Protocoles de sécurité pour les réseaux de capteurs sans fil, Mémoire de magister en informatique option : ingénierie des systèmes d'information, Université Hadj lakhdar Batna, 15/07/2010.

[21] R.Anderson, H.Chan, and A.Perrig, key Infection : Smart Trust for Smart Dust.In proceedings of the 12th IEEE international, conference on network protocols, octobre 2004.

[22] W.Znaidi,Thèse : Quelques propositions de solutions pour la sécurité des réseaux de capteurs sans fil, L'Institut National des Sciences Appliquées de Lyon, Octobre 2010,

[23] F. Hu , N. K. Sharma. Security considerations in ad hoc sensor networks. Ad Hoc Networks 3, Elsevier Science, pp. 69Ű89, 2005.

[24] S. Zhu, S. Setia, S. Jajodia, LEAP : Efficient Security Mechanisms for Large-Scale distributed Sensor Networks, In Proc of The 10th ACM Conference on Computer and Communications Security . pp. 62-72. 2003.

[25] L.Lanying, X.Wang, A high security dynamic secret key management scheme for Wireless sensor networks, third internantional symposium on intelligent information technology and security informatics,2010 , pp.507-5010

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

[27] M.Shainika, C.Hema, National Conference on Recherche Advences In communication ,computation ,Electrical Science And structures NCRACC

[28] A.S.Poornima, B.B.Amberker, Tree-based Key Management Scheme for Heterogeneous Sensor Networks, 1 6th IEEE International Conference, ICON 2008, ISBN 978- 1-4244-3805-1 , NewDelhi, 2008.

[29] A.Aileni, Cour universitaire, Cluster based Key Management in Wireless Sensor Networks, Oklahoma State University Stillwater.

[30] S. Jayapraba , A.F.Sheik Hakkani, Secutity key management ant authentication sheme for wireless sensor networks, ARPN Journal of Engineering and Applied Sciences, Mars 2015.

[31] .Laskar et D.Jena, A Secure Key Management Scheme for Hierarchical WSN , Computer Engineering and Intelligent Systems, Vol.6, No.2, 2015.

[32] S.Verma, M.Prachi ,A Cluster based Key Management Scheme for Underwater Wireless Sensor Networks,conference : Computer Network and Information Security, 2015.

[33] D. Carman, P. Kruus, and B. Matt. Constraints and approaches for distributed sensor network security. 2000.

[34] L.Shen, X.Shi, A Dynamic Cluster-based Key Management Protocol in Wireless Sensor Networks,INTERNATIONAL JOURNAL OF INTELLIGENT CONTROL AND SYSTEMS,VOL. 13, NO. 2, JUNE 2008.

[33] [35] Arthur and S. Vassilvitskii, k-means++ : the advantages of careful seeding. Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms. pp. 1027Ű1035, 2007.

[36] D. MacKay, Information Theory, Inference and Learning Algorithms. Cambridge University Press. pp. 284-292, 2003.

bibliography

Résumé

Un réseau de capteurs sans fil (RcSF) est un réseau ad hoc particulier. Il est utilisé, en général, pour contrôler un environnement particulier. Il est constitué d'un ensemble de capteurs communicants par des liaisons sans fil. Les RcSFs interviennent dans des applications particulières : militaires, médicales, environnementales, pour la surveillance des infrastructures critiques dans des zones sinistrées et hostiles. Une des contraintes principales dans les réseaux de capteurs sans fil est la protection des communications, pour cela, les RcSF nécessitent des mécanismes de sécurité efficaces et peu couteux en énergie. La plupart des protocoles de gestion de clés proposés dans la littérature se basent sur des mécanismes de chiffrement symétrique. Une grande partie de ces protocoles utilise la méthode de pré-distribution de clés. Dans le présent travail, nous avons étudié, dans un premier temps, les techniques cryptographiques proposées dans la littérature ainsi que les différents protocoles de gestion de clés existants et nous avons proposé par la suite une solution qui pourrait satisfaire conjointement aux deux contraintes majeures liées à l'établissement de communications sécurisées et à la gestion efficace de l'énergie.
Mots clés :Energie, Protocoles de gestion de clés, Réseaux de capteurs.

Abstract

A Wireless Sensors Network (WSN) is a special ad hoc network. It is used in general to monitor a particular environment. It consists of a plurality of sensors communicating with wireless links. The WSNs are involved in specific applications : military, medical, environmental, for the monitoring of critical infrastructure in the affected and hostile areas. One of the major constraints in wireless sensor networks is the protection of communications. For this, the WSN security mechanisms require efficient and inexpensive energy. Most key management protocols proposed in literature are based on symmetric encryption mechanisms. Most of these protocols use the pre-distribution of keys. In this project, we studied, in a first time, cryptographic techniques proposed in the literature and various protocols of management key existing and we subsequently proposed a solution that satisfies both the major constraints linked to establishment of secure communications and efficient energy management.
keyword :Energy, key management protocols , sensor networks.