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

Mémoire de fin de Cycle

Pour l'obtention du diplôme de Master en Informatique
Option :*Réseaux et Système Distribués*

Thème

Etude des Mécanismes de Gestion des clés dans les réseaux de capteurs Sans Fils - Proposition d'un Protocole Hybride basé sur la Stéganographie -

MAKHLOUFI Yasmina & REZGUI Zahia

Devant le jury composé de :

Président du jury :	M^{r}. SAADI Mustapha.
Examinatrice :	M^{r}. MOKETFI Mohand.
Examinatrice :	M^{me}. DJENANE Amel.
Encadreur :	M^{r}. AISSANI Sofiane.

Promotion 2012/2013

Résumé

Un réseau de capteurs sans-fils ($RCSF_s$) est un ensemble de capteurs communicants par des liaisons sans-fils. Les contraintes matérielles de ces capteurs ainsi que les environnements hostiles dans lesquels ils pourraient être déployées, rendent ce type de réseaux très vulnérable et nécessite des mécanismes de sécurité efficaces et peu coûteux.

Dans ce mémoire, nous présentons une étude des problèmes de sécurité dans les $RCSF_s$ et un état de l'art des protocoles de gestion de clés utilisées dans les mécanismes cryptographiques ; comme conséquence, un nouveau protocole de gestion de clés est proposé.

Mots clés : Réseaux de capteurs sans-fils ($RCSF_s$), Sécurité, Cryptographie, Communication sans-fils, Gestion de clés.

Abstract

A wireless sensor network (WSN) is a set of communicating sensors by without physical links. Limitations of the sensors and hostile environments in which they could be deployed, make this type of network very vulnerable and requires efficient security mechanisms and inexpensive.

In this thesis, we present a study of wireless sensor networks, security problem in these networks, we present also a state-of-the-art of key management protocols, as consequence we propose a new protocol for key management.

Keywords : Wireless sensor networks, Security, Cryptography, Wireless communication, Key management.

Remerciements

Nous commencerons d'abord par adresser nos sincères remerciements à notre encadreur M^r :AISSANI Sofiane pour son aide précieuse, pour son immense patience, et sa disponibilité, pour la qualité de ses conseils, pour ses remarques critiques pertinentes et fructueuses, pour les nombreuses discussions que nous avons eu et pour avoir su nous faire confiance. Sans vous, Monsieur, la réalisation de ce mémoire n'aurait pas en lieu. Encore une fois, merci beaucoup.

Nous remercions M^r :SAADI Mustapha, pour l'honneur qu'il nous a fait en présidant notre jury de mémoire.

Nous remercions tout particulièrement M^r MOKATFI Mohand, et M^{me} DJENANE Amel d'avoir accepté d'examiner notre travail,sans oublier nos respects à tous nos professeurs.

Nos sincères remerciements s'adressent à nos parents, nos frères et nos soeurs pour leur soutien moral, leur encouragement inconditionnel, et surtout pour la confiance qu'ils nous accordent.

Nous remercions aussi M^r :HAMMOUDI Samir, de nous avoir aidés, orientés et conseillés.

Enfin, on remercie tous ceux qui ont collaboré de près ou de loin à l'élaboration de ce travail, en particulier tous nos ami(e)s pour leur soutien moral.

Yasmina & Zahia

Dédicaces

Je dédie ce modeste travail :

À *mes parents qui depuis mon plus jeune âge ont toujours fait leur maximum, en consacrant temps et argent, pour m'éveiller et m'encourager dans mes passions. C'est grâce a vous et pour vous que j'ai fait mon mémoire. Aucun mot sur cette page ne saurait exprimer ce que je vous dois, ni combien je vous aime. Qu'Allah vous bénisse, vous assiste, vous vienne en aide. Merci papa et merci mama.*

À *ma sœur «*Wassila *et son Mari* Nadir*» et ma sœur «* Yasmina *», et mes deux frères «* Kamel, Lyes *» pour avoir contribué à la réussite de ce travail d'une manière indirecte, d'y avoir apporté tant d'humeur et d'amour et pour tout le soutien moral prodigué dans les moments les plus difficiles.*

En ce qui te concerne Sofiane*, je vais faire bref : merci pour ta patience inconditionnelle et pour avoir été là à chaque fois que j'en ais eu besoin.*

À *ma collègue* Yasmine *qui m'a aidée dans les moments difficiles et avec qui j'ai eu le bonheur de partager des moments d'amitie. Je te remercie de tout mon cœur.Ainsi que toute sa famille*

À *ceux qui n'ont jamais cessé de se soucier de mon sort, ceux qui ont toujours été la pour m'encourager et me soutenir moralement, mon grand père* Lhadj Said *et tout sa famille.*

À *tous mes proches grands et petits.*

À *mes plus fidèles amies "* Amel, Lydia, Lamia, Meriem et Sassa *" et amis "* Juba, Zazi, Adel, Mohammed *et* Bilal *" . Je prends le temps de vous remercier tous et toutes individuellement pour votre soutien, votre aide et les services que vous m'avez rendus. Je vous en serais éternellement reconnaissant.*

À *tous mes enseignants du département informatique.*

Que Dieu les protège tous . . .

REZGUI Zahia

Dédicaces

Je voudrai dédier ce modeste travail en premier a ma chère grande mère qui ma soutenu durant ce projet, et qui ma appris à ne jamais baisser les bras pour réussir.

***À** mes chers parents symbole de sacrifice, de tendresse et d'amours ; sont les moindres sentiments que je puisse vous témoigner. Quoi que je fasse, je ne pourrais jamais vous récompenser pour les grands sacrifices que vous avez faits et continuer de faire pour moi. Aucune dédicace ne saurait exprimer mes grandes admirations, mes considérations et mes sincères affectations pour vous.*

***À** mes chères sœurs : Naima, Sabrina, Narimane et Meriem.*

***À** mes cousines et cousins, tentes et oncles.*

***À** tous mes enseignants du département d'informatique qui m'on offert un cadre de travail adapté a mes besoins.*

***À** mes amis : Juba, Mohammed, Farid, Adel, Bilal, qui n'ont pas hésité à être la quand j'avais besoins de leurs soutient.*

***À** toutes mes amies Meriem, Nissa, Lylia, Seltane, Rebiha, avec lesquels j'ai partagé mes moments de joie et de bonheur.*

***S**ans oublié ma binôme Zahia et son fiancé Sofiane ainsi que toute sa famille.*

Enfin je le dédie a tous mes amis que je n'ai pas cité, et à tous ceux que me connaissent, ainsi qu'a tous mes camarades de Master 2 (Académiques et professionnelle), Informatique.

MAKHLOUFI Yasmina

Table des matières

Table des figures

Liste des tableaux

Liste des Acronymes

ADC	**A**nalog to **D**igital **C**onverters.
AES	**A**dvanced **E**ncryption **S**tandard.
ASIC	**A**pplication **S**pecific **I**ntegrated **C**ircuit.
CAN	**C**onvertisseur **A**nalogique/**N**umérique.
CHs	**C**luster **H**eads.
CPU	**C**entral **P**rocessing **U**nit.
DCH	**D**ata **C**luster **H**eads.
DES	**D**ata **E**ncryption **S**tandard.
DoS	**D**enial **o**f **S**ervice.
DSA	**D**igital **S**ignature **A**lgorithm.
DSP	**D**igital **S**ignal **P**rocessors.
ECC	**E**lliptic **C**urve **C**ryptography.
ECDH	**E**lliptic **C**urve **D**iffie−**H**ellman.
ECDSA	**E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm.
ECIES	**E**lliptic **C**urve **I**ntegrated **E**ncryption **S**cheme.
EROM	**E**xtended **ROM**.
FPGA	**F**ield **P**rogrammable **G**ate **A**rray.
GPS	**G**lobal **P**ositioning **S**ystem.
IBPRF	**I**dentity **B**ased Key **P**re-Distribution using a **P**seudo **R**andom **F**unction.

IEEE	**I**nstitue of **E**lectronical and **E**lectronics **E**ngineers.
IETF	**I**nternet **E**ngineering **T**ask **F**orce.
LEACH	Low Energy Adaptive Clustering Hierarchy.
LEAP	**L**OCALIZED **E**NCRYPTIONAND **A**UTHENTICATION **P**ROTOCOL.
MAC	**M**essage **A**uthentification **C**ode.
MANET	**M**obile **A**d hoc **NET**work.
MHKMPS	**M**obile and **H**ybrid **K**ey **M**anagement **P**rotocol using **S**téganographie
MK	**M**achine **K**ey.
OTMK	**O**paque **T**ransitory **M**aster **K**ey Establishment.
PDA	**P**ersonnal **D**igital **A**ssitant.
PKC	**P**ublic **K**ey **C**ryptography.
PRF	**P**seudo **R**andom **F**unction.
QoS	**Q**ualité **o**f **S**ervice.
RAM	**R**andom Access Memory.
RCH	**R**outing **C**luster **H**eads.
RCSF	**R**éseau de **C**apteur **S**ans **F**il.
RC4	**R**ivest **C**ipher **4**.
RF	**R**adio **F**réquence.
RN	**R**eady **N**ode.
ROM	**R**ead **O**nly **M**emory.
RSA	**R**on Rivest, Adi **S**hamir et Leonard **A**dleman.
SB	**S**tation de **B**ase.
SNKM	**S**ecurity **N**ode-based **K**ey **M**anagement Protocol.
STKM	**S**panning **T**ree for **K**ey **M**anagement in Wireless Sensor Networks.
Wi-Fi	**W**ireless **F**idelity.
WSN	**W**ireless **S**ensor **N**etwork.

Introduction générale

Au cours de ces dernières années, le développement technologique des réseaux de communication sans fils, a connu un essor important grâce aux avancées technologiques dans divers domaines, telles que la micro-électronique et la miniaturisation. C'est ainsi que de nouvelles voies d'investigation ont été ouvertes avec l'émergence des réseaux de capteurs sans fils qui sont composés de petits dispositifs électroniques. Ces éléments sont autonomes, équipés de capteurs et capables de communiquer entre eux sans fils. En plus ils collaborent entre eux pour former un réseau de capteurs sans fils capable de superviser une région ou un phénomène d'intérêt, de fournir des informations utiles par la combinaison des mesures prises par les différents capteurs et de les communiquer ensuite via le support sans fils à un centre de contrôle distant.

Les réseaux de capteurs sans-fils sont de plus en plus utilisés par les applications de surveillance de grands systèmes dans une variété de domaines : le militaire, l'environnement, la santé, l'habitat, l'éthologie, etc. Leurs remarquable essor est dû à leurs taille de plus en plus réduite, leurs prix de plus en plus faible ainsi que leurs support de communication sans-fils attrayant peu encombrant mais également peu sécurisant.

La sécurité est une nécessité pour la majorité des applications qui utilisent les

$RCSF_s$, donc il est nécessaire que les nœuds communicants se partagent des clés cryptographiques pour le chiffrement et l'authentification. Le nombre de nœuds élevé, et par conséquent, le nombre de clés potentiellement important, nécessite une gestion judicieuse afin de prendre en considération les contraintes de ressources imposées par ce type de réseaux (énergie limitée, capacité de calcul et de stockage réduite, etc.).

Les techniques classiques de gestion des clés, qui utilisent une clé publique ou un centre de distribution de clés pour la cryptographie, sont inadéquates aux environnements de réseaux de capteurs. Pour cela, la plupart des solutions de gestion des clés proposées pour les réseaux de capteurs sans-fils, sont basées sur la cryptographie symétrique. De plus, afin d'achever l'établissement de clés entre les nœuds du réseau, ces solutions se basent sur la méthode de pré-distribution dans laquelle les clés sont chargées dans les nœuds capteurs avant leurs déploiement.

Organisation du mémoire

Ce mémoire est organisé comme suit :

Dans le premier chapitre, nous présentons les concepts généraux relatifs au domaine des réseaux de capteurs sans fils. Dans le deuxième chapitre, nous parlerons d'abord de la sécurité informatique en générale ensuite la sécurité dans les $RCSF_s$ ainsi que les attaques liées à ce dernier. Dans le troisième chapitre, nous ferons un état de l'art sur la gestion des clés dans les RCSFs où nous ferons une étude bibliographique sur les protocoles de gestion de clés dans les RCSFs .Après cela, nous proposerons notre protocole MHKMPS et des résultats de simulations qui nous permettront de comparer notre protocoles avec ceux de la litérature.

1

Généralité sur les réseaux de capteurs

Introduction

L'apparition récente des communications sans fils accessibles sur des portables, l'évolution des dispositifs de calcul et les progrès dans l'infrastructure de communication ont abouti à la croissance rapide des réseaux sans fils. De plus en plus, des objets s'équipent de processeurs et de moyens de communications mobiles, leurs permettant le traitement des informations mais également leurs transmissions. Le développement des réseaux sans fils et des réseaux mobiles ouvre une nouvelle ère dans le domaine de télécommunications. L'utilisation d'une interface sans-fils introduit des différences par apport à la communication par câble. Il n'y à pas si

longtemps, la seule solution pour acheminer les données du capteur jusqu'au contrôleur central était le câblage qui avait comme principaux défauts d'être couteux et encombrant.

Contrairement aux réseaux traditionnels qui se préoccupent de la garantie d'une bonne qualité de service, les réseaux de capteurs donnent une importance primordiale à la conservation d'énergie. Ils doivent intégrer des mécanismes qui permettent aux utilisateurs de prolonger la durée de vie du réseau en entier, car chaque nœud est alimenté par une source d'énergie limitée et généralement irremplaçable. Les reseaux de capteurs sans fils ($RCSFs$ [2]) représentent une révolution technologique des instruments de mesure, issue de la convergence des systèmes électroniques miniaturisés capables de mesurer certains phénomènes physiques dans l'environnement où ils sont déployés.

Dans ce chapitre, nous allons présenter un ensemble de généralités sur les réseaux de capteurs, leurs architectures, leurs caractéristiques ainsi que leurs domaines d'applications. Avant de conclure ce chapitre, Nous discuterons sur les principaux facteurs qui influencent la conception des réseaux de capteurs ainsi que les problématiques liées à ce dernier.

1.1 Les réseaux sans fils

Un réseau sans fils est un réseau informatique qui connecte différents postes ou systèmes entre eux par ondes radio [13]. Ces dernières sont plus exposés aux perturbations et aux interférences que ne le sont les communications filaires [50].
La norme IEEE [2]802.11, connu sous le nom de Wi-Fi est la norme la plus utilisée actuellement pour les réseaux sans fils.
Le rayonnement géographique des ondes est relativement limité, étant donné la faible puissance d'émission des solutions matérielles actuelles. Pour cette raison les réseaux

2. voir la liste des abréviations

sans fils se sont avant tout développés comme réseaux internes propre à un bâtiment, soit comme réseau d'entreprise, soit comme réseau domestique [13].

1.1.1 Les Réseaux ad-hoc

Un réseau sans fils ad hoc (ou MANET [1]) est formé par un ensemble d'hôtes (Fixes ou Mobiles) qui s'organisent seuls et de manière totalement décentralisée, formant ainsi un réseau autonome et dynamique ne reposant sur aucune infrastructure filaire. Aucune infrastructure n'étant disponible, ces objets ont donc à découvrir dynamiquement leurs environnement [12].

Cependant, l'IETF qui représente l'organisme responsable de l'élaboration de standards pour Internet, définit les réseaux ad hoc de la manière suivante :

"Un réseau ad hoc est un système autonome de plate-formes mobiles (par exemple un routeur interconnectant différents hôtes et équipements sans fils) appelées nœuds qui sont libres de se déplacer aléatoirement et sans contrainte. Ceci provoque des changements rapides et imprédictibles de la topologie du réseau. Ce système peut fonctionner d'une manière isolée ou s'interfacer à des réseaux fixes à travers des passerelles. Dans ce dernier cas, un réseau ad hoc est un réseau d'extrémité".

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

1.1.2 Les Réseaux de capteurs

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

1. voir l'Annexe

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

Bien que ces deux types de réseaux ont plusieurs points en commun, cependant on peut mentionnés quelques différences illustrées ainsi dans le tableau suivant [12] :

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

TABLE 1.1 – Comparaison entre les réseaux de capteurs et les réseaux ad-hoc.

1.2 Les réseaux de capteurs sans fils ($RCSF_s$)

1.2.1 Présentation d'un nœud capteur

1.2.1.1 Définition d'un capteur

Un capteur est un dispositif physique qui assure trois taches complémentaires : le relevé d'une grandeur physique, le traitement de l'information, et la communication avec d'autres capteurs. C'est un système qui représente les phénomènes physiques détectés (température, pression, humidité, etc.) sous forme de signal électrique. Ces dispositifs sont déployés de manière aléatoire à travers une zone géographique, appelée zone d'interêt [3].

Les capteurs sont des petits appareils dotés d'une batterie, capables de communiquer

entre eux et de détecter des événements s'ils se trouvent à l'intérieur de leurs rayon de perception, ainsi que des moyens qui leurs permettent de collecter, de stocker et de transmettre des données collectés au centre de collecte via le canal sans fils, du fait que un capteur possède le matériel nécessaire pour effectuer les communications sans fils par ondes radio [12].

1.2.1.2 Architecture matériel d'un capteur

Il existe divers types de capteur [1] sur le marché : les capteurs de température, d'humidité, de pression, etc. cependant, ils restent dotés d'une architecture matérielle similaire, malgré cette diversité apparente. Ainsi, on ne pourra pas décrire tous ces différents capteurs, et donc une liste exhaustive peut être trouvée sur le site The Sensor Network Museum [48].

Un capteur est composé principalement d'une unité de : captage, traitement, stockage, communication, et énergie. Selon le domaine d'application, des composants additionnels peuvent être rajoutés, comme par exemple un système de localisation tels qu'un GPS, un générateur d'énergie telle qu'une cellule solaire, ou un mobilisateur lui permettant de se déplacer. Ces composants principaux et optionnels (représentés par des traits discontinus) sont illustrés dans la figure 1.1 [3].

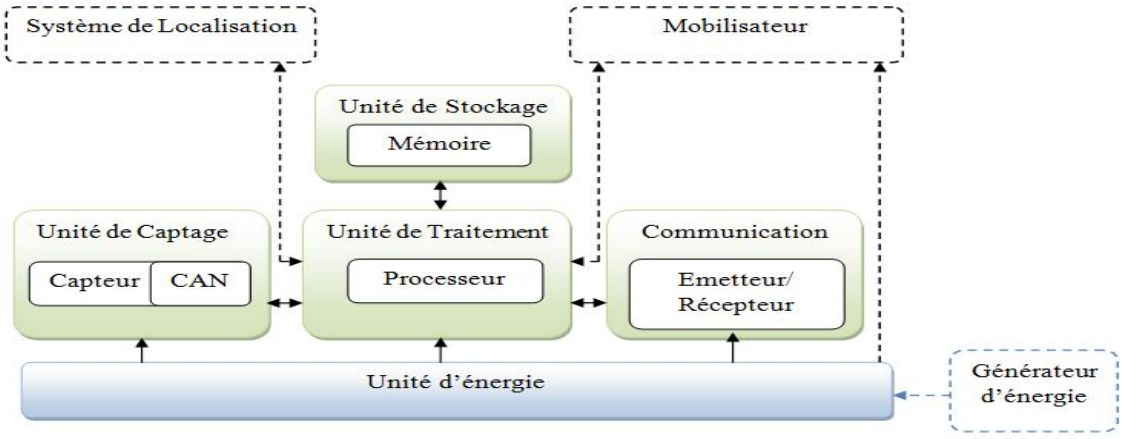


FIGURE 1.1 – Architecture matériel d'un capteur.

1. voir l'Annexe

1. **L'unité d'énergie (Power unit)**

   Un capteur est muni d'une source d'énergie, généralement une batterie, pour alimenter tous ces composants [3][4]. Cette batterie n'est généralement ni rechargeable ni remplaçable. Lors de conception de protocoles pour les réseaux de capteurs, la capacité d'énergie limitée au niveau des capteurs représente la contrainte principale puisqu'elle influe directement sur la durée de vie des capteurs et donc d'un réseau de capteurs [5].

2. **Unité de captage (Sensing unit)**

   La fonction principale de l'unité de captage consiste à capturer les données physiques à partir de l'objet cible [3]. Elle est composée de deux sous-unités : un dispositif de capture physique qui prélève l'information locale à capter (reconnaissent la grandeur physique à capter), et un convertisseur analogique/numérique appelé CAN [2]) (ADC [2])) qui convertit le signal analogique en signal électrique [5]. Le capteur envoie des signaux analogiques basés sur le phénomène observé au convertisseur CAN. Ce dernier transforme ces signaux en données numériques et les envoient à l'unité de traitement. Une ou plusieurs unités de captages peuvent appartenir à un même capteur [3][6].

3. **Unité de traitement (Processing unit)**

   Cette unité reçoit les données transmises par l'unité de captage, elle effectue un traitement sur ces données (si nécessaire) puis décide quand et où les envoyer. Elle exécute les programmes et les différents protocoles de communications, ainsi différents types de processeurs qui peuvent être utilisés dans un capteur incluent le Microcontrôleur, les DSP [2]), les FPGA [2]) et les ASIC [2]). Parmi toutes ces alternatives, le Microcontrôleur a été le processeur le plus utilisé pour les capteurs à cause de sa flexibilité à être relié aux autres composants (comme par exemple l'unité de communication), à son bon prix et sa faible consommation énergétique [3][6][7][8].

2. voir la liste des Abréviations

4. **Unité de communication (Transceiver unit)**

   Cette unité est composée d'un émetteur/récepteur permettant la communication entre les différents nœuds du réseau [5]. Un support de communication sans fils est nécessaire pour les émissions et réceptions de données, ainsi les différents choix de media de transmission incluent la Radiofréquence (RF), Le Laser et L'infrarouge.

5. **Unité de stockage (Mémoire)**

   Cette unité inclut généralement deux types de mémoires : la mémoire de programme (dont les instructions sont exécutées par le processeur) et la mémoire de données (pour conserver les données fournies par l'unité de captage et d'autres données locales). La taille de cette mémoire est souvent limitée par les considérations économiques et s'améliorera aussi probablement au fil des années [3].

1.2.1.3 Contraintes imposées par un capteur sans fils

Dans un réseau de capteurs, plusieurs contraintes sont imposées par les nœuds capteurs tels que : le type d'interface physique, la bande passante, et la résolution nécessaire (le débit de données à transférer, la taille des messages à transporter, etc.), l'architecture du réseau (capteur isolé, réseau local de capteurs, etc.), et l'environnement (capteur isolé, bruit externe, etc.),etc. Ainsi, leurs contraintes principales seront détaillées par la suite :

1. **Energie**

   Les capteurs sans fils sont des éléments indépendants les uns des autres, par conséquent, ils doivent disposer d'une alimentation autonome. La consommation d'énergie est un point très important pour les réseaux de capteurs. Souvent, dans les environnements sensibles, il est impossible de recharger ou changer une batterie, donc avoir une meilleure gestion de la consommation d'énergie est primordial pour augmenter la durée de vie du réseau. L'énergie

est consommée par divers fonctionnalités des réseaux qui sont donc par ordre décroissant de consommation d'énergie : Radio (pour la Communication), Protocoles (MAC, routage), Processeur (Calcul, agrégation). Le manque d'énergie sur certain nœud dégrade la fiabilité des transmissions et peut conduire à une partition du réseau (apparition des nœuds sans lien de communication), causant des changements topologiques cruciaux.

2. **Capacités limitées (processeur et mémoire)**

   En plus de l'énergie, les nœuds capteurs ont aussi une capacité de calcul (traitement) et de mémoire limitée [1].

1.2.2 Présentation d'un $RCSF_s$

1.2.2.1 Définition d'un réseau de capteurs

Un réseau de capteurs sans fils (RCSF), ou Wireless Sensor Network (WSN), se compose généralement d'un grand nombre de nœuds capteurs intelligents " smart sensor " qui vari de quelques dizaines d'éléments à plusieurs milliers, communiquant entre eux via des liens radio pour le partage d'informations et le traitement coopératif. Dans ces réseaux chaque nœud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs point de collecte. Les $RCSF_s$ sont une instance particulière de la classe des réseaux ad hoc, ils héritent leurs caractéristiques et partagent beaucoup de concepts existants avec ceux-ci.

La position des nœuds n'est pas obligatoirement prédéterminée. Ils sont dispersés à travers une zone géographique appelée champ de captage, les données captées sont acheminées grâce à un routage multi-saut à un nœud considéré comme un " point de collecte " appelé nœud puits (sink) [1]. On peut citer comme exemples un réseau détecteur de feu de forêt, ou un réseau de surveillance de solidité d'un pont après un tremblement de terre.

1.2.2.2 Architecture d'un RCSF

Les nœuds capteurs sont dispersés dans une zone de capture, chacun de ces nœuds a la possibilité de collecter les données et les router vers une ou plusieurs stations de base. Cette dernière est un point de collecte de données capturées appelé aussi puits (sink), elle peut communiquer les données collectées à l'utilisateur final à travers un réseau de communication, éventuellement Internet. L'utilisateur peut à son tour utiliser la station de base comme passerelle, afin de transmettre ses requêtes vers le réseau [5]. Ce processus est illustré dans la figure (Figure 1.2 [14]).

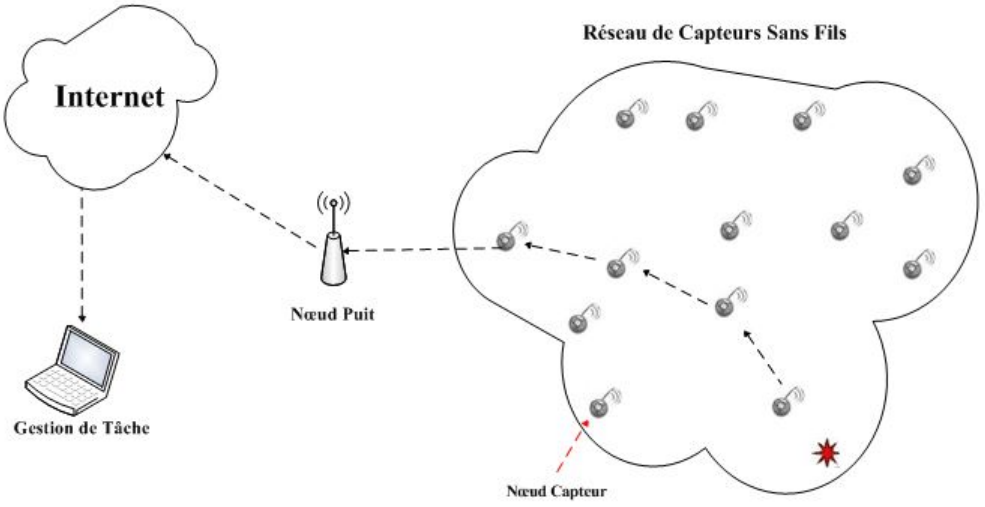


FIGURE 1.2 – Architecture générale d'un réseau de capteurs sans fils.

1.2.2.3 Caractéristiques des réseaux de capteurs sans fils

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

- **La durée de vie limitée**

  L'intervalle de temps qui sépare l'instant de déploiement du réseau de l'instant où le premier nœud tombe en panne énergétiquement. Les capteurs utilisent

leurs énergies à des fins de calcul et de transmission de données. Dans un RCSF chaque nœud joue le rôle d'émetteur et de routeur, ainsi, une défaillance énergétique d'un nœud capteur peut affecter des changements significatifs à la topologie du réseau et imposer une réorganisation couteuse de ce dernier [1].

- **Ressources limitées**

  Habituellement, les nœuds capteurs ont une taille très petite, ce facteur de forme limite la quantité de ressources qui peuvent être mises dans ces nœuds, par conséquence la capacité de traitement et de mémoire est très limitée [2].

- **Bande passante limitée (Média de transmission)** Different Médium sans fils (radio, infrarouge, optique), sont utilisées par les nœuds capteurs pour se communiquer. Le médium utilisé doit être compatible avec l'environnement de l'application ; cependant, la majorité des capteurs communiquent par l'utilisation d'un circuit RF " Radio Fréquence ". En raison de la puissance limitée, les nœuds capteurs ne peuvent pas supporter des débits très élevés [1].

- **Scalabilité**

  Le nombre des nœuds capteurs dans un réseau de capteurs peut être de l'ordre de centaines ou milliers (selon l'application) [1].

- **Réseau auto-organisé**

  Une configuration manuelle d'un réseau est en pratique impossible à réaliser á cause du grand nombre de nœuds et leurs déploiements dans des environnements hostiles. Par ailleurs, des nœuds peuvent quitter le réseau en tombant en panne (manque d'énergie, panne physique,etc.), et d'autres peuvent l'intégrer. Par conséquent, il est essentiel que le réseau s'auto-organise [1] .

- **Topologie dynamique**

  La topologie des réseaux de capteurs change d'une manière fréquente et rapide du fait que les nœuds peuvent être déployés dans des milieux difficiles (par exemple un champ de bataille), ainsi que la défaillance très probable des nœuds capteurs. Cependant, les nœuds capteurs et les nœuds finaux (les nœuds de

destination) où ils doivent envoyer l'information capturée peuvent être mobiles, et donc la transmission de messages en provenance ou vers un nœud mobile est un autre défi. Ainsi, la capture peut être aussi bien statique que dynamique dépendant de l'application[1].

- **L'agrégation des données**

  Les techniques d'agrégation des données concernent le traitement des données par le réseau, permettent de réduire le nombre de messages et par conséquent réduire la consommation en énergie. Dans les RCSFs, les données produites par les nœuds capteurs sont très corrélées, ce qui implique l'existence de redondances de données. Les utilisateurs sont intéressés par le phénomène qui est saisi par les données générées par chaque nœud et par conséquent, ces réseaux fournissent la possibilité d'agréger les données afin de réduire la largeur de la bande passante [1].

1.3 Domaine d'application

La miniaturisation, l'adaptabilité, le faible coût et la communication sans fils permettent aux réseaux de capteurs d'envahir plusieurs domaines d'applications. Parmi elles nous citons :

1.3.1 Applications militaires

Le déploiement rapide, le coût réduit, l'auto-organisation et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui rendent ce type de réseaux appréciable, voire indispensable, ce qui rend les applications militaires des $RCSF_s$ multiples. Plusieurs projets ont été lancés pour aider les unités militaires dans un champ de bataille et protéger les villes contre des attaques, telles que les menaces terroristes[3].

1.3.2 Applications médicales

Les réseaux de capteurs peuvent être utilisés dans le domaine de la médecine, pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro-capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers, . . .).Ils peuvent effectuer des mesures physiologiques telles que : la tension artérielle, battements du cœur, etc [3].

1.3.3 Applications commerciales

Les capteurs peuvent être utilisés pour le contrôle environnemental des bâtiments, pour permettre une meilleure gestion des ressources à faibles coûts. Un autre exemple est celui de l'utilisation des capteurs dans des musées scientifiques pour un apprentissage plus rapide des visiteurs. L'application des nœuds de capteurs dans le domaine commercial permet de contrôler l'environnement dans les usines et les bureaux, ainsi que le contrôle d'inventaire, de qualité des produits, et des robots dans les environnements de fabrications automatiques et la surveillance de routes[3].

1.3.4 Applications environnementales

Les micro-capteurs dispersés à partir d'un avion dans une zone difficile d'accès peuvent permettre de détecter des incendies, surveiller des catastrophes naturelles (inondations, séismes, éruptions volcaniques), surveiller des phénomènes météorologiques, de détecter la pollution (qualité des eaux, taux d'ensoleillement, fuite du pétrole . . .) [3].

1.3.5 Applications liées à la sécurité

Les $RCSF_s$ peuvent être utilisés dans plusieurs domaines liés à la sécurité dont on peut citer : les altérations dans la structure d'un bâtiment, suite à un séisme ou au vieillissement, pourraient être détectées par des capteurs intégrés dans les murs ou

dans le béton, la surveillance de voies ferrées pour prévenir des accidents matériels ou humaines peut être une application intéressante des réseaux de capteurs. Cependant, l'application des réseaux de capteurs dans le domaine de la sécurité pourrait diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux (forêt, usines, etc.) et à la protection des êtres humains[9].

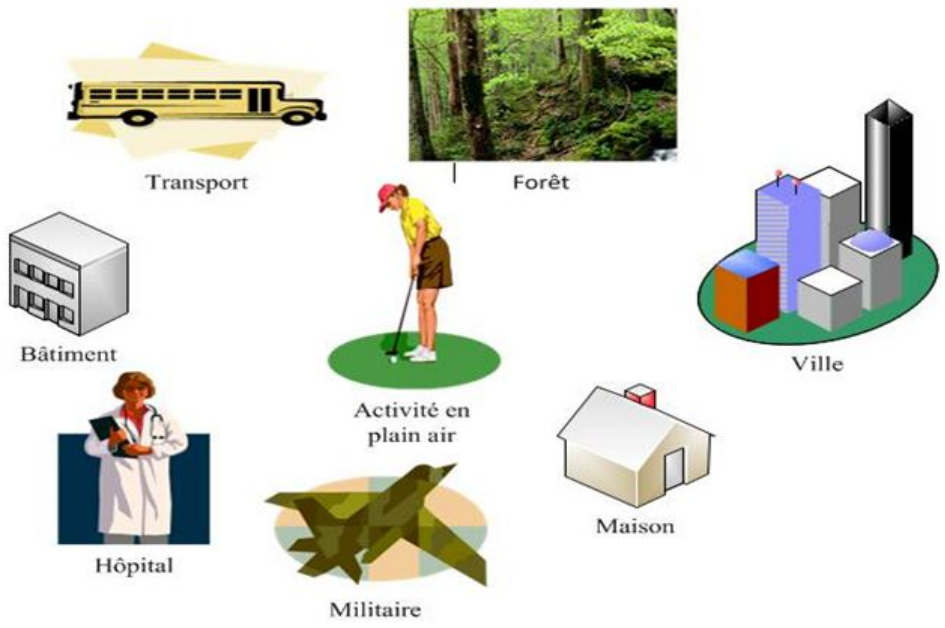


FIGURE 1.3 – Quelques domaines d'applications pour les RCSFs.

1.4 Facteurs et contraintes de conception d'un RCSF

Un réseau de capteurs possède plusieurs contraintes qui influencent sur la conception et la mise en place des $RCSF_s$, parmi lesquels : la tolérance aux pannes, la scalabilité, le coût de production, les contraintes matérielles, l'environnement, le support de transmission et la consommation énergétique, la topologie du réseau. Ces facteurs importants servent comme directives pour le développement des algorithmes et protocoles utilisés dans les réseaux de capteurs ; ils sont considérés également comme métrique de comparaison de performances entre les différents travaux dans le domaine [2].

1.4.1 La tolérance aux pannes

Quelques nœuds capteurs peuvent être en panne ou être bloqués à cause du manque d'énergie, de dommage physique ou d'interférence environnementale, La défaillance des nœuds ne devrait pas affecter la tâche globale du réseau. Ce qu'on appelle la fiabilité ou la tolérance aux pannes qui consiste à pouvoir maintenir le fonctionnement du réseau de capteurs sans fils en cas de défaillance d'un nœud même s'il fonctionne en mode dégradé[2].

1.4.2 L'extensibilité (passage a l'échelle ou la Scalabilité)

Le nombre de nœuds capteurs déployés dans une région de capture peut être dans l'ordre des centaines ou milliers, ou plus. Par conséquent, le réseau doit être capable de fonctionner avec un grand nombre de capteurs tout en permettent l'évaluation de ce nombre. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter-nœuds et nécessite que la station de base soit équipée de suffisamment de mémoire pour stocker les informations reçues[2].

1.4.3 Le coût de production

Les réseaux de capteurs doivent être à faible coût afin de pouvoir les déployer à grande échelle, c'est-à-dire, que le coût de réseau ne doit pas dépasser le coût de déploiement de capteurs. Le coût de production d'un seul micro-capteur est très important pour l'évaluation du coût global du réseau[2].

1.4.4 Les contraintes matérielles

Parmi les contraintes matérielles liées aux $RCSF_s$, on peut citer :

- **La dimension**

  La taille réduite des nœuds capteurs peut présenter de nombreux avantages, elle permet un déploiement flexible et simple du réseau. Cependant, la puis-

sance de batteries utilisées pour alimenter les nœuds capteurs est limitée, par la petite taille de ces derniers.

- **Puissance de calcul**

  Les processeurs des réseaux de capteurs sont différents de ceux d'une machine classique, car ils utilisent souvent des microcontrôleurs de faibles fréquences.

1.4.5 Environnement

Les nœuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : forte chaleurs, pluie, humidité,. . . [3].

1.4.6 Support de transmission

Dans un réseau de capteurs multi-sauts, les nœuds communicants sont liés par un médium sans fils qui peut être de l'un des trois types suivants : radio, infrarouge, ou optique. Cependant, il faut s'assurer de la disponibilité du moyen de transmission choisi dans l'environnement de capture afin de permettre au réseau d'accomplir la totalité de ses tâches [2].

1.4.7 Consommation énergétique

Comme les nœuds capteurs sont des composantes microélectroniques, ils peuvent être équipés seulement d'une source d'énergie limitée (<0,5 Ampère-heure, 1,2 V). Dans certains scénarios d'applications, il est impossible de réapprovisionner de l'énergie, par conséquent la durée de vie d'un nœud capteur dépend fortement de la durée de vie de sa batterie. D'autres parts, la retransmission des données, la réorganisation du réseau ainsi que le changement de sa topologie rendent la gestion et la conservation d'énergie d'une haute importance. Cette énergie est consommée par les différentes unités du capteur afin de réaliser les taches de captage, traitement de données et communication. Cette dernière est l'opération qui consomme le plus d'énergie [3].

Dans les réseaux de capteurs, l'efficacité en consommation d'énergie représente une métrique de performance significative, qui influence directement sur la durée de vie du réseau en entier [2].

1.4.8 La topologie du réseau

Le maintien de la topologie d'un réseau de capteurs est une tâche complexe vu aux caractéristiques de déploiement aléatoire, le fonctionnement autonome et la fréquence élevé de panne [2]. En effet, le déploiement d'un grand nombre de nœuds exige une bonne gestion de la maintenance de la topologie du réseau déployé. Cette maintenance consiste en trois phases :

- **Pré-déploiement et déploiement :** Les nœuds capteurs peuvent être soit éparpillés en masse ou bien placés un par un dans le champ de perception. Ils peuvent être déployés en les jetant d'un avion, délivrés dans un obus d'artillerie, fusée, ou missile, et placés soit par un humain ou un robot.
- **Post-Déploiement :** Après la phase de déploiement, la topologie du réseau peut subir des changements dû aux changement de position des nœuds , non-accessibilité à cause du brouillage ou des obstacles en mouvement, épuisement d'énergie, et le mal fonctionnement d'un ou de plusieurs nœuds.
- **Redéploiement des nœuds additionnels :** Des nœuds capteurs additionnels peuvent être redéployés à n'importe quel moment pour remplacer les nœuds mal fonctionnant. Cette addition entraîne la réorganisation du réseau et le changement de sa topologie.

1.5 Les problématiques liées aux $RCSF_s$

Derrière les RCSFs se cachent une multitude de problématiques, Les travaux et pistes de recherche dans ce domaine viennent en réponse aux nombreux problèmes relatifs aux réseaux de capteurs. Un nombre important d'entre eux sont issus des

réseaux sans fils Ad Hoc et d'autres sont propres aux $RCSF_s$.

1.5.1 La gestion des ressources

Le point crucial à considérer dans les $RCSF_s$ est la limitation des ressources qui sont : l'énergie disponible, la mémoire et la puissance de calcul. Pour les $RCSF_s$, la complexité se situe au niveau des restrictions de ressources à considérer dans, par exemple, l'implémentation d'un algorithme de cryptographie ou de contrôle des erreurs dans un tel capteur.

1.5.2 La gestion des données collectées

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

1.5.3 La mise à l'échelle

Au niveau des réseaux sans fils Ad Hoc, le problème de mise à l'échelle ou d'extensibilité existe bien mais celui-ci est plus important dans les $RCSF_s$ et le sera encore plus dans les années à venir. Les capteurs sans fils du futur sont destinés à avoir des dimensions microscopiques permettant, par exemple, de les disperser sur la zone à étudier à partir d'un hélicoptère. Des milliers voire des dizaines de milliers de capteurs pourront ainsi être disséminés dans un périmètre restreint.

1.5.4 L'adressage

Le problème d'adressage est lié à celui de la mise à l'échelle. Le nombre important de capteurs sans fils déployé oblige à se poser des questions sur l'adressage à utiliser. Deux tendances ressortent au sujet de cette problématique :

- Identification unique des capteurs ;
- Identification par localisation.

La première tendance consiste à offrir à chaque capteur un identifiant unique. Quand le nombre de capteurs est limité à une cinquantaine, des solutions simples à mettre en place sont possibles. La deuxième tendance est de ne pas fournir d'identification aux nœuds mais de se centrer sur les données en leurs associant un repère spatial et temporel. De cette manière, on sait d'où provient la donnée ainsi que l'heure à laquelle elle a été collectée .

1.5.5 La tolérance aux pannes

La robustesse des capteurs sans fils est un élément très important. Lors de l'utilisation de ses capteurs à l'extérieur, il est très rare de disposer d'un atelier électronique mobile adapté à une réparation des capteurs in-situ. Les causes de pannes dans les $RCSF_s$, sont nombreuses et on cite, par exemple, des dommages liés à l'environnement, un manque d'énergie ou des interférences liées à l'environnement.

1.5.6 L'organisation et le fonctionnement du réseau

Une simplification trop grande du mécanisme de tolérance aux pannes serait de penser que plus le nombre de capteurs dédiés à une zone est important plus le réseau est robuste. Or, les interférences de communication augmentent avec le nombre de capteurs au mètre carré. En outre, dans un réseau très dense, le routage est fortement lié à la gestion de l'énergie. Ainsi, les routes peuvent être rallongées en terme de nombre de sauts c'est-à-dire de nœuds rencontrés pour préserver certains nœuds affaiblis. A l'inverse, la mise en place du réseau, les routes les plus courtes doivent être préférées surtout celle qui relie le nœud à la station centrale de collecte de données.

1.5.7 La sécurité

Les aspects liés à la sécurité avaient été peu dans les premiers travaux sur les $RCSF_s$, mais la faute n'était pas à la motivation des chercheurs mais aux ressources

disponibles au sein des capteurs. Pour mettre en place une application de réseau de capteurs, il faut résoudre essentiellement les problématiques de routage, de gestion de données puis vient ensuite la sécurité [10].

Conclusion

Les réseaux de capteurs restent une nouvelle technologie peu accessible au grand publique. La flexibilité, la tolérance aux fautes, le prix réduit et les moyens rapides de déploiement des réseaux de capteurs annoncent un futur prometteur à cette technologie.

Cependant, la mise en place d'une application de RCSF doit prendre en considération les contraintes qui caractérisent les nœuds capteurs et qui sont principalement : La consommation d'énergie, les contraintes physiques, le changement de la topologie et l'adaptation à l'environnement.

En outre, pour assurer la fiabilité du système, de nouveaux mécanismes et mesures de sécurité doivent être mis en place afin d'éliminer les vulnérabilités dans les $RCSF_s$ qui sont sujets à de nombreuses menaces et attaques.

Dans le prochain chapitre, nous aborderons de façon plus complète l'aspect de la sécurité dans les $RCSF_s$.

2

La Sécurité des réseaux de capteurs

Introduction

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

Cependant l'agrégation de données dans un RCSF implique des problèmes de sécurités, tels que l'intégrité, la confidentialité ou bien la fraîcheur des données. Des

techniques doivent alors être déployées pour limiter les risques de sécurité tout en respectant les contraintes imposées par l'architecture des $RCSF_s$.
Dans ce chapitre, nous avons présenté d'abord la sécurité informatique en général ensuite nous parlons sur la sécurité dans les $RCSF_s$ ainsi que les attaques liées aux $RCSF_s$.

2.1 La Sécurité informatique

2.1.1 Définition

La sécurité informatique est l'ensemble des techniques qui assurent que les ressources du système d'information (matérielles ou logicielles) d'une organisation sont utilisées uniquement dans le cadre où il est prévu qu'elles le soient. On peut dire aussi que la sécurité informatique est un ensemble des moyens mis en œuvres pour réduire la vulnérabilité d'un système contre les menaces accidentelles ou intentionnelles [44].

2.1.2 Objectifs de la sécurité

Lorsque nous abordons le problème de sécurité, nous visons à atteindre certains objectifs, dont les principaux sont les suivants :

a) **L'authentification :** Elle permet de coopérer au sein des $RCSF_s$ sans risque, en contrôlant et en identifiant les participants. En effet, on ne peut assurer une confidentialité et une intégrité des messages échangés si, dès le départ, on n'est pas sûûr de communiquer avec le bon nœud. Si l'authentification est mal gérée, un attaquant peut se joindre au réseau et injecter des messages erronés [14].

b) **L'intégrité :** Elle assure que les données reçues n'ont pas été altérées durant leurs transit dans le réseau de manière volontaire ou accidentelle [14].

c) **La confidentialité :** Une fois les parties authentifient, la confidentialité reste un point important, étant donné la communication sans-fils des $RCSF_s$. Elle consiste à préserver le secret des messages échangés et ne pas les révéler aux

adversaires[14].

d) **La disponibilité :** L'objectif de la disponibilité est de garantir l'accès à un service ou à des ressources. Cette propriété reste difficile à assurer dans les $RCSF_s$ [14].

e) **La fraicheur :** Ce service permet de garantir que les données échangées sur le réseau sont actuelles et ne sont pas une réinjection de précédents échanges interceptés par un attaquant [14] .

f) **Le contrôle d'accès :** Ce service consiste à empêcher un accès au réseau à tout élément étranger au système. Le contrôle d'accès donne aux participants légitimes un moyen pour détecter les messages provenant de sources externe du réseau [9].

g) **La non répudiation :** Signifie que l'expéditeur d'un message ne peut pas plus tard nier l'envoie de l'information et le récepteur ne peut pas nier la réception [9].

2.1.3 La vulnérabilité

Les vulnérabilités sont les faiblesses d'un système ou d'un logiciel. Ces faiblesses sont exploitées par des hackers pour obtenir un accès à une ressource (CPU [2], réseaux, etc.) ou à une information. Les faiblesses se trouvent au niveau des services, des applications et des utilisateurs :

a. **Les services**

En général, ces services sont l'e-mail, le Web ou toute autre application qui communique avec une autre application.

b. **Les applications**

Les vulnérabilités des applications nécessitent souvent une intervention humaine : par exemple l'utilisateur qui active un virus par un clic de souris. Quoi de plus alléchant que des mails avec le sujet "I love you" ou avec un contenu

2. voir la liste des abréviations

informant que vous êtes l'heureux gagnant d'une énorme somme d'argent ? Un simple clic et on se retrouve victime d'un virus.

c. **Les actions des utilisateurs**

Les utilisateurs peuvent engendrer des vulnérabilités sur des systèmes ou des logiciels par des configurations mauvaises ou incorrectes. Un utilisateur peut reconfigurer un système ou un logiciel et ouvrir ainsi des portes à des hackers. Dans ce cas, même le système de sécurisation le plus performant, s'il est mal configuré, offre une brèche de sécurité [64].

2.1.4 Les risques

En sécurité informatique, le risque est lié à l'éventualité menace informatique volontaire ou involontaire, interne ou externe au système d'information. Le risque informatique est lié à la connaissance du métier, au niveau de maîtrise de l'outil informatique ainsi qu'à l'utilisation des moyens de contrôles. D'autre part, le risque informatique dans une entreprise est étroitement lié à la dépendance de cette entreprise par rapport à son système d'information [65].

2.1.5 Les Menaces

Les menaces contre la sécurité informatique regroupent les actions susceptibles de nuire et de porter atteinte, partiellement ou totalement, à un système informatique. Dans ce contexte, les menaces peuvent être soustraites du cadre purement virtuel, propre au système numérique, et ainsi venir de l'extérieur ou de l'intervention humaine [66].

2.1.6 Les attaques de sécurité

Les attaques portées à la sécurité d'un ordinateur ou d'un réseau sont mieux caractérisées en considérant le système en tant que fournisseur d'information. Il existe quatre catégories d'attaques : interruption, interception, modification et fabrication.

- **interruption :** Un atout du système est détruit ou devient indisponible ou inutilisable. C'est une attaque portée à la disponibilité. La destruction d'une pièce matérielle (tel qu'un disque dur), la coupure d'une ligne de communication, ou la mise hors service d'un système de gestion de fichiers en sont des exemples.
- **Interception :** Une tierce partie non autorisée obtient un accès à un atout. C'est une attaque portée à la confidentialité. Il peut s'agir d'une personne, d'un programme ou d'un ordinateur. Une écoute téléphonique dans le but de capturer des données sur un réseau, ou la copie non autorisée de fichiers ou de programmes en sont des exemples.
- **Modification :** Une tierce partie non autorisée obtient accès à un atout et le modifie de façon (presque) indétectable. Il s'agit d'une attaque portée à l'intégrité. Changer des valeurs dans un fichier de données, altérer un programme de façon à bouleverser son comportement ou modifier le contenu de messages transmis sur un réseau sont des exemples de telles attaques.
- **Fabrication :** Une tierce partie non autorisée insère des contrefaçons dans le système. C'est une attaque portée à l'authenticité. Il peut s'agir de l'insertion de faux messages dans un réseau ou l'ajout d'enregistrements à un fichier [67].

2.1.7 Les outils de cryptographie

Pour réaliser la sécurité dans n'importe quel modèle de communication, il est important de chiffrer les messages transmis aux nœuds selon un arrangement de gestion des clés convenues.

- **Le chiffrement [13]**

  Le chiffrement est un système cryptographique assurant la confidentialité. Pour cela, il utilise des clés. Selon cette utilisation, on distingue deux classes de primitives : symétrique et asymétrique.

1. **Le chiffrement symétrique**

   Une même clé est utilisée entre nœuds communicants pour chiffrer et déchiffrer les données en utilisant un algorithme de chiffrement symétrique, comme illustré dans la figure 2.1 [68]. Les algorithmes de chiffrements symétriques sont décomposés en deux catégories :

   – Le chiffrement en chaine est fait bit à bit sans attendre la réception entière des données. comme RC4 [2][61].

   – Le chiffrement par bloc consiste à fractionner les données en blocs de taille fixe. Chaque bloc sera ensuite chiffré une fois qu'il atteint la taille envisagée. Comme DES [2][69], AES [2] [69].

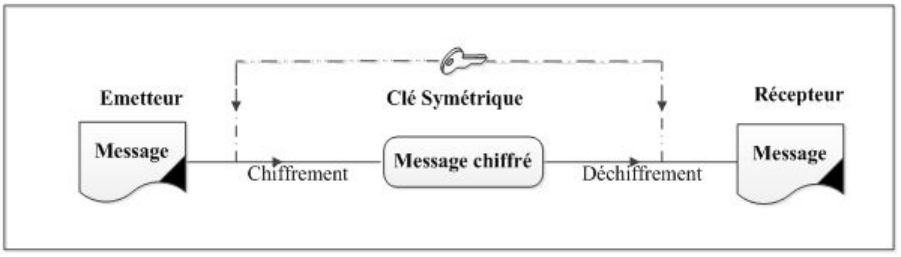


FIGURE 2.1 – Le chiffrement symétrique.

2. **Le chiffrement asymétrique**

   Deux clés différentes sont générées par le récepteur : une clé publique diffusée à tous les nœuds servant au chiffrement de données qu'ils vont émettre au récepteur, et, une clé privée maintenue secrète chez le récepteur servant pour le déchiffrement de ces données lorsque ce dernier les reçoit, comme illustré dans la figure 2.2 [68]. Le point fondamental sur lequel repose la sécurité du chiffrement asymétrique est l'impossibilité de déduire la clé privée à partir de la clé publique. L'algorithme le plus connu est : RSA [1][70].

2. voir la liste des abréviations
1. voir l'Annexe

FIGURE 2.2 – Le chiffrement asymétrique.

- **La signature digitale**

La signature digitale est un système cryptographique assurant la non-répudiation de la source. Elle repose sur des clés asymétriques. L'émetteur (A) signe les données à transmettre avec sa clé privée (A) en produisant une signature digitale (1). Ce dernier est par la suite envoyé avec les données (2). Si elle peut être déchiffrée, la clé publique (A) par le récepteur (B) et si son résultat est identique aux données reçues alors la signature est valide (4), c'est-a-dire, les données proviennent de leur émetteur légitime qui ne pourra pas nier l'émission de ces données dans le future[13]. Ce processus est expliqué dans la figure 2.3 [68] :

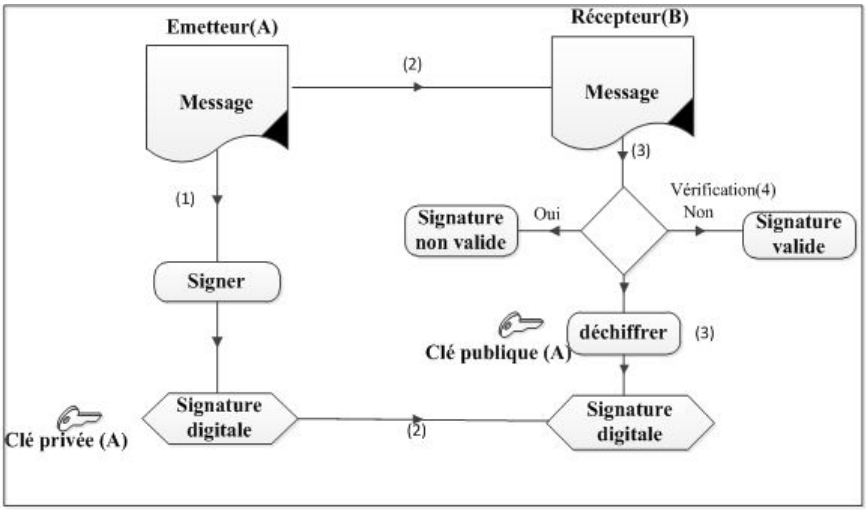


FIGURE 2.3 – La signature digitale.

- **La fonction de hachage**

Le principe de cette fonction est qu'un message clair de longueur quelconque doit être transformé en un message de longueur fixe inférieure à celle de départ. Le message

réduit portera le nom de "Haché", de " résumé ", ou de "Condensé". L'intérêt est d'utiliser ce condensé comme empreinte digitale du message original afin que ce dernier soit identifié de manière univoque. La fonction de hachage doit être telle qu'elle associe un et un seul haché à un texte en clair (cela signifie que la moindre modification du document entraîne la modification de son haché). D'autre part, il doit s'agir d'une fonction à sens unique afin qu'il soit impossible de retrouver le message original à partir du haché. S'il existe un moyen de retrouver le message en clair à partir du haché, la fonction de hachage est dite " à brèche secrète ". Le hachage est aussi employé pour les signatures numériques. Ce processus est expliqué dans la figure 2.4 [68] :

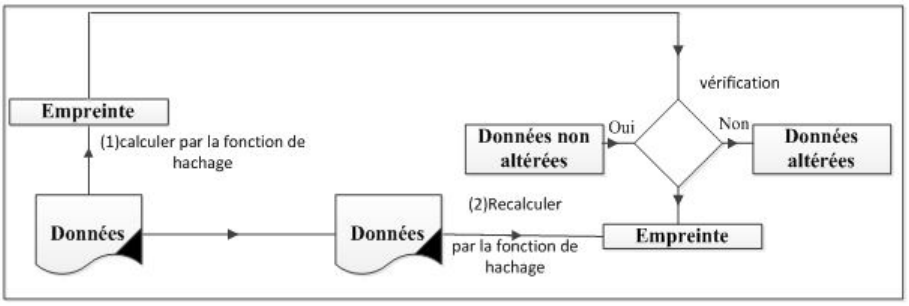


FIGURE 2.4 – La fonction de hachage.

2.2 La sécurité des réseaux de capteurs

2.2.1 Définition

Un réseau de capteurs est un type spécial de réseaux. Il partage quelques vulgarisations avec un réseau informatique typique, mais pose également des conditions uniques de ses propres caractéristiques [13]. En plus des objectifs de sécurité cités en hauts qui concerne la sécurité des réseaux informatique en général , un protocole de sécurité pour un réseau de capteurs doit satisfaire les conditions suivantes :

a) **Auto-organisation :** Les capteurs du réseau doivent être capables, après avoir été déployés, de s'auto-organiser et surtout de se sécuriser eux-mêmes, sans autres interventions extérieures. Ce besoin d'auto-organisation se retrouve dans l'établissement automatique de la distribution des clés de cryptages entre les nœuds du réseau et la gestion de ses clés ou bien encore dans le développement des relations de confiance entre capteurs du réseau (principalement dans l'utilisation de sécurité utilisant les principes des réseaux de confiance). Pour cela les capteurs doivent avoir été munis au préalable des outils qui leurs permettent de telles fonctionnalités [49].

b) **Localisation sécurisée :** Le besoin de se localiser et de connaître la position des autres nœuds peut être primordial dans de nombreux cas, pour déjouer d'éventuelles attaques jouant sur les distances [49].

c) **Temps synchronisé :** De nombreuses solutions de sécurité nécessitent des capteurs synchronisés pour qu'elles soient effectives. Il faut ainsi s'assurer que les capteurs du réseau ou des sous-réseaux du réseau ont une horloge commune afin par exemple d'éviter des attaques de type rejeter des paquets [49].

2.2.2 Analyse de vulnérabilité

Quelques faiblesses sont inhérentes à la nature des $RCSF_s$ et d'autre à la technologie retenue pour leurs mise en œuvre et leurs déploiement. Nous distinguons deux catégories : la vulnérabilité physique et la vulnérabilité technologique.

1. **Vulnérabilité physique :** La vulnérabilité physique est le fait qu'un capteur est fréquemment installé dans un lieu peu sûr. Parmi les vulnérabilités physiques on trouve : les lieux public, les environnements naturels (forêt, région montagneuse, désert, etc.) ainsi que la plupart des bâtiments, maisons intelligentes et musées [14].
2. **Vulnérabilité technologique :** Les vulnérabilités technologiques sont liées

à plusieurs facteurs. Comme par exemple la technologie sans fils sous jacente, quiconque possédant un récepteur adéquat peut potentiellement écouter ou perturber les messages échangés, cependant les mécanismes de routage sont d'autant plus critiques dans les $RCSF_s$ que chaque nœud participe à l'acheminement des paquets à travers le réseau [12][14]. Nous pouvons donc faire le bilan suivant :

- La limitation de ressources restreint les mécanismes de sécurité.
- La puissance de calcul limitée empêche l'utilisation de mécanismes de protection cryptographiques résistants.
- L'absence de SB impose un fonctionnement autosuffisant des nœuds .

2.2.3 Contraintes influençant la sécurité dans un RCSF

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

- **Puissance d'énergie basse :** L'énergie des nœuds capteurs est limitées, et généralement irremplaçable. Les réseaux ad hoc visent à réaliser une haute qualité de service (QoS [2]) tel que minimiser le temps d'attente et la réservation de débit, alors que les protocoles des réseaux de capteurs doivent se concentrer principalement sur la conservation d'énergie .
- **Espace mémoire et capacité de calcul limité :** Dans la majorité des $RCSF_s$, les nœuds n'ont pas la capacité de mémoriser des clés de taille importante ou d'exécuter des protocoles cryptographiques complexes [52].

2.2.4 Défi de la sécurité

Le premier défi de la sécurité consiste à minimiser la consommation d'énergie toute en maximisant les performances de sécurité. En effet, les capacités et les contraintes du nœud capteur influencent directement sur les performances et les

2. voir la liste des abréviations

mécanismes de sécurité utilisés. Ainsi, la plus grande partie d'énergie consommée par un nœud pour assurer la sécurité est liée :

- Au calcul requis pour les fonctions de sécurité, tel que le chiffrage, déchiffrage, signature de données, vérification de la signature.
- A l'énergie requise pour la transmission des données de sécurité.
- A l'énergie requise pour le stockage des paramètres de sécurité, tel que le stockage de la clé de chiffrement.

Le deuxième défi concerne la topologie des $RCSF_s$, ces derniers sont exposés aux attaques, qu'elles soient passives ou actives grâce à la topologie qui change fréquemment du fait des pannes des nœuds capteurs ou leurs mobilités.

Le troisième défi est celui de la communication sans fils qui caractérise les $RCSF_s$, ce type de communication rend les schémas de sécurité utilisés dans les réseaux filaires impraticables [14].

2.2.5 Les attaques dans les $RCSF_s$

Les différentes spécificités citées précédemment (énergie limitée, faible puissance de calcul, utilisation des ondes radio, etc.) exposent les réseaux de capteurs à de nombreuses menaces. 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. Un attaquant peut chercher à récupérer les informations du réseau en écoutant le médium, si le réseau n'encrypte pas ses données. Ce type d'attaque est plus facile à réaliser (il suffit de posséder un récepteur adéquat) et il est difficile de le détecter puisque l'attaquant ne porte aucune modification sur les données échangées [49].

Dans les attaques actives, un attaquant cherche à modifier ou supprimer des informations, ou bien encore à empêcher le réseau de fonctionner correctement. Il peut aussi injecter son propre trafic, ou rejouer d'anciens messages pour perturber le fonctionnement du réseau ou provoquer un déni de service.

Dans la suite de cette section, nous décrivons une liste non exhaustive mais représentative des attaques les plus courantes et connues, actives ou passives, que nous pouvons trouver dans les réseaux de capteurs sans fils.

- **Ecoute passive :** Cette attaque consiste à écouter le réseau et à intercepter les informations circulant sur le médium. Elle est facilement réalisable si les messages circulant sur le réseau ne sont pas cryptés. Par ailleurs l'écoute passive est difficile à détecter, car de par sa nature passive, elle ne modifie pas l'activité du réseau.
- **Analyse du trafic :** Cette attaque met en jeu des mécanismes d'écoute passive et de surveillance du réseau. L'attaquant en analysant uniquement les chemins empruntés par les paquets sur le réseau pourra récupérer des informations précieuses sur les vulnérabilités de ce réseau [49].
- **Brouillage Radio :** Du fait qu'il est quasiment impossible de restreindre l'accès à un médium utilisant des ondes radio. Un attaquant peut donc envoyer des ondes sur la même fréquence que le réseau de capteurs pour brouiller les ondes radio [53]. Les nœuds du réseau n'ont alors plus accès au medium et ne peuvent plus communiquer du fait de ce brouillage radio. Or un réseau sans accès au médium est un réseau hors service.
- **Attaque sur la couche de lien :** La fonction principale de la couche de liens dans les réseaux de capteurs est de gérer l'accès au medium de communication et contrôler les erreurs. Ainsi, l'attaque sur cette couche [53] est un autre type d'attaque de déni de service, qui consiste à provoquer des collisions lors de l'envoi de message afin, d'une part d'empêcher la transmission d'un paquet, et d'autre part obliger le capteur émetteur à retransmettre le paquet. Et donc si le protocole de communication de la couche de lien spécifie à un capteur de réenvoyer le paquet jusqu'à ce qu'il soit acquitté, le capteur va épuisée sa batterie en renvoyant continuellement le même paquet.
- **HELLO Flooding :** Les protocoles de découvertes sur les $RCSF_s$ utilisent

des messages de type HELLO pour s'insérer dans un réseau et pour découvrir ses nœuds voisins. Dans une attaque dite de HELLO Flooding, l'attaquant va utiliser ce mécanisme pour consommer l'énergie des capteurs et empêcher leurs messages d'être routés.

Considérants,un nœud malicieux X avec une connexion radio puissante qui lui permet d'envoyer à un grand nombre de nœuds des messages de type HELLO, de manière continue. Les nœuds voisins V vont alors considérer le nœud malicieux comme un voisin, même s'ils sont situés à des distances qui ne permettent pas de l'atteindre. Lorsqu'ils chercheront à envoyer des données, les nœuds V vont passer par le nœud X qu'ils considèrent comme leur voisin, mais leurs messages ne pourront jamais l'atteindre. Comme X est inaccessible, ils vont utiliser leur antenne radio au maximum de sa puissance, consommant alors leurs énergie, et leurs messages ne seront jamais transmis car jamais reçus.

- **Réplication de données :** Si un attaquant a la possibilité de lire et enregistré les paquets envoyer sur le réseau, il pourra alors renvoyer ces mêmes paquets à une date ultérieure pour tromper le réseau. Cette attaque peut être illustrée si on prend pour exemple un réseau de capteurs qui a pour objectif de détecter un incendie. Si un premier incendie est détecté et qu'un capteur envoie un paquet pour informer la base, l'attaquant pourra enregistrer ce paquet même s'il est chiffré et qu'il ne peut le déchiffré, puis l'émettre à une autre date en se faisant passer pour ce capteur et faire croire à un nouvel incendie. Cette attaque est réalisable si le paquet ne contient d'informations concernant la date de l'envoi ou si cette date est accessible et facilement modifiable par un attaquant (information non chiffré par exemple).
- **Destruction ou vol :** Les plus élémentaires des attaques actives dans les réseaux de capteurs sans fils sont le vol et la destruction. Le fait que les capteurs sont le plus souvent déployés dans des zones qui ne peuvent être surveillées ou difficile d'accès, ces attaques sont avérés très facile.

Une personne physique seule peut subtiliser un ou plusieurs capteurs, voire les détruire. Dans ce cas, le réseau doit être capable de s'adapter à la situation, sous peine de ne plus fonctionner, ou d'être coupé en plusieurs sous-réseaux incapables de communiquer entre eux car les nœuds qui faisaient le pont entre sous réseaux sont détruits ou subtilisés. De plus, un nœud volé, peut divulguer certaines informations à un attaquant [49].

- **Nœud compromis ou nœud malicieux :** Cette attaque physique peut permettre à un attaquant d'extraire par exemple les clés cryptographiques contenues dans le capteur, modifier ces circuits électroniques ou modifier le programme qu'il contient pour le remplacer par un autre, afin que le capteur devient ce que l'on appelle un nœud compromis ou nœud malicieux (malicious node). Ce dernier contrôlé par l'attaquant va lui permettre de s'intégrer au réseau, de récupérer des informations ou de lancer d'autres attaques [55]. Des travaux de recherche récents ont pour objectif de créer des capteurs résistants aux attaques physiques avec des mécanismes tels que la suppression des clés cryptographiques lors de la détection d'une atteinte physique du capteur. Cependant la plupart des capteurs utilisés aujourd'hui sont très vulnérables aux attaques physiques, comme démontré par [54] qui à prouvé qu'un capteur de type Mica2 peut être compromis dans un temps inférieur à 1 minute.
- **Attaque du trou noir (Black Hole Attack) :** Cette attaque consiste tout d'abord à insérer un nœud malicieux dans le réseau [56].Ce nœud, par divers moyens, va modifier les tables de routage pour obliger le maximum de nœuds voisins à faire transiter leurs informations par lui. Ensuite tel un trou noir dans l'espace, toutes les informations qui vont passé en son sein ne seront jamais retransmises.
- **Attaque du trou de ver (Worm hole attack) :** Cette attaque nécessite l'insertion d'au moins de deux nœuds malicieux dans le réseau de capteurs [57]. Ces deux nœuds sont reliés entre eux par une connexion puissante qui

peut être filaire ou radio. Le but de cette attaque est de tromper les nœuds voisins sur les distances les séparant. Dans le cas d'une attaque du trou de ver, les deux nœuds malicieux permettent d'atteindre un lieu éloigné en un saut unique. Cette possibilité va tromper les autres nœuds sur les distances réelles qui séparent les deux nœuds, mais va surtout avoir pour conséquence que les nœuds voisins vont principalement passer par ces nœuds malicieux pour faire circuler leurs informations. Ainsi les nœuds malicieux qui forment le trou de ver vont se trouver dans une position privilégiée qui va leurs permettre d'avoir une priorité sur l'information circulant à travers leurs nœuds proches.

- **Attaque sybile (Sybil Attack) :** Une attaque sybile [58] consiste à ce qu'un capteur malicieux se fasse passer pour plusieurs capteurs en utilisant l'identité d'autres capteurs légitimes du réseau.

  L'attaque sybile va alors pouvoir tenter de mettre en péril les mécanismes comme l'agrégation des données, la sécurité, le routage, l'allocation de ressource ou la détection d'intrus. Un nœud malicieux qui peut se faire passer pour plusieurs nœuds peut gagner un avantage important pour une élection de cluster Head. Fort d'un nombre de votes plus important, il pourra tromper ses nœuds voisins pour par exemple inciter le cluster à l'élire comme cluster Head. Si le nœud malicieux obtient cette distinction, ses décisions au sein du cluster auront une incidence plus forte (refus de routage des informations en dehors du cluster, envoi d'information tronquée sur les clusters voisins, etc.).
- **Altération des messages :** Un nœud malicieux va récupérer un message et l'altérer, en lui ajoutant des fausses informations (sur le destinataire, l'émetteur, l'information en elle-même), en le modifiant ou bien en détruisant des paquets pour rendre incompréhensible le message[49].
- **Privation de mise en veille :** Cette attaque active à pour but de priver un capteur de se mettre en veille par différents moyens [59]. Le capteur s'il ne peut plus se mettre en veille va consommer très rapidement sa batterie, jusqu'à se

retrouver hors service[49].

- **Attaque spécifique au type de capteur :** Cette attaque concerne le capteur lui même, où un attaquant modifie d'une manière physique son comportement. Il peut par exemple allumer une flamme devant un capteur thermique ou bien allumer une lampe devant un capteur de luminosité. Le but est de tromper le capteur, et ainsi d'envoyer ou d'enregistrer de fausses informations sur le réseau, ou bien tout simplement de faire réagir assez longtemps un nœud ou le réseau pour qu'ils consomment leurs énergie, comme dans le cas d'une attaque de type privation de mise en veille [49].

2.2.6 Modèles de l'attaquant

- **Attaquant puissant (Strong attacker) :** L'adversaire est considéré comme présent avant et après le déploiement des nœuds. Il peut surveiller toutes les communications, n' importe où et à tout instant [12].
- **Un modèle réaliste d'attaquant :** L'attaquant est capable de surveiller un pourcentage fixe des canaux de communication lors du déploiement du réseau [43]. Voici comment ce modèle est exprimé dans [43] : "The hostile surveillance is not ubiquitous during the deployment phase of the network and only fraction of the established link keys can be obtained by the attacker".

2.2.7 Mécanismes de sécurité pour les $RCSF_s$

Les attaques présentées précédemment montrent l'étendue des possibilités d'attaque d'un réseau de capteurs et les différents points de sécurité mis en danger (intégrité du réseau, authentification, confidentialité des données, etc.).
Dans ce qui suit nous présentons quelques solutions de sécurité et conjointement une description des moyens mis à disposition pour les atteindre.

1. **Etablissement et gestion des clés :** Un aspect de sécurité qui suscite beaucoup d'attention dans un RCSF est le domaine de la gestion des clés. L'éta-

blissement des clés peut se faire une fois que les capteurs sont déployés. C'est le cas lorsqu'un grand nombre de capteur est déployé de façon aléatoire, par exemple à partir d'un hélicoptère. Dans ce type de scénario, chaque nœud doit établir une clé secrète avec chacun de ces voisins, ces derniers n'étant pas connus avant le déploiement. Ce problème serait trivial à résoudre si l'utilisation des protocoles utilisant la cryptographie à clé publique était possible. Les nœuds auraient pu alors s'échanger leurs certificats et établir une clé secrète [1].

Le problème avec la cryptographie asymétrique, dans un RCSF est qu'elle est trop intensive en calcul pour les différents nœuds du réseau. C'est vrai dans le cas général, cependant, Flane et al ont prouvé dans [62] qu'il est faisable avec un bon choix d'algorithme. La gestion des clés consiste à générer des clés, les stocker, les distribuer, les mettre à jour, les révoquer, les supprimer et les archiver. L'anonymat des nœuds et la limitation des ressources disponibles en termes de mémoire et de capacité de calcul dans les $RCSF_s$ rendent la gestion des clés très difficile.

2. **Authentification nœud - station de base :** Le problème d'authentification entre un nœud et la SB [2] est connu sous le nom d'authentification initiale. Un capteur devra après le déploiement du réseau être en mesure de transmettre ses informations à la SB , cet échange nécessite que les données soient authentifiées et présupposé dans le cadre de la cryptographie symétrique que le capteur et la SB partagent un secret. L'authentification initiale est le moyen de mettre en place ce secret [63].

3. **Prévention de déni de service :** Une attaque de DoS [1] peut être définie en tant que n'importe quel événement qui diminue ou élimine une capacité de réseau d'exécuter les fonctions prévues. Pannes de matériel, erreurs de pro-

2. voir la liste des abréviations
1. voir l'Annexe

grammation, épuisement de ressource, conditions environnementales, ou toute interaction compliquée entre ces facteurs peut causer un DoS [2] [12].

Conclusion

A cause de leurs vulnérabilités, les $RCSF_s$ sont sujets à de nombreuses Menaces et attaques. Pour faire face à la plupart de ces menaces, la cryptographie est un moyen très puissant. Mais, pour assurer la sécurité dans les RCSFs les systèmes cryptographiques doivent être associés à une gestion de clés efficaces. La gestion de clés sera le sujet de notre prochain chapitre ainsi que l'étude de quelques protocoles de gestion de clés dans le cadre des $RCSF_s$.

2. voir la liste des abréviation

3

La Gestion des clés dans les réseaux de capteurs 'Etat de l'Art'

Introduction

La majorité des études et recherches dans les réseaux de capteurs sont basées sur la capacité de les rendre faisables et utiles. Dans le domaine de la sécurité, des solutions ont été proposées par des chercheurs aux menaces liés à la sécurité des $RCSF_s$.

La gestion des clés et les algorithmes cryptographiques assurent, la disponibilité, la confidentialité, l'intégrité, et l'authentification pour les protocoles de transmission, ils fournissent ainsi des mécanismes efficaces et sécurisés . 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 à travailler dans le domaine de la gestion des clés dans les reseaux de capteurs sans-fils, on cite :

- Chaque système cryptographique est fondé sur la gestion des clés.
- Des 'Bonnes' clés cryptographiques doivent se présenter pour qu'elles soient utilisées par la cryptographie, les signatures numériques ou MAC.
- La sécurité des clés implique la sécurité du réseau entier.
- La confiance accordée aux informations reçus.
- Une rupture dans le schéma de distributions 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 $RCSF_s$.
- Le problème de gestion des clés est l'un des problèmes les plus délicats de la cryptographie.

Problématique

La gestion des clés est le processus par lequel des clés cryptographiques sont produites, enregistrées, protégées, transférées, chargées, employées, et détruites.

- Quel est le nombre de clés nécessaire et comment elles sont distribuées avant le déploiement des nœuds ? C'est le problème de pré-distribution des clés.
- Comment une paire de nœuds, ou un groupe de nœuds établissent une clé ? C'est le problème de l'établissement de clé.
- Comment un nœud ajouté au réseau peut établir et fixer une clé avec des nœuds existant dans le réseau ? C'est le problème d'ajout de nœud.
- 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 ? C'est le problème d'isolation des nœuds anormaux.

3.1 La gestion des clés

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 il est important que, chacun des utilisateurs doit disposer d'un ensemble de clés secrètes (dans un système à clés secrètes) ou de paire de clés publiques/privées (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. Ce dernier doit aussi pouvoir enregistrer et gérer ses clés publiques et privées de manière sûre[11].Ainsi, la figure 3.1 [1] illustre les différentes fonctions de la gestion des clés.

FIGURE 3.1 – Fonctions de la gestion des clés.

Les protocoles de gestion des clés ont pour but de permettre la pré-distribution des clés cryptographiques, le renouvellement des clés expirées, ainsi que de retirer les clés quand les nœuds quittent le réseau, et assigner des nouvelles clés aux nœuds joignant le réseau [9].

3.1.1 La gestion des clés dans les réseaux de capteurs

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

- Sécuriser le routage
- Sécuriser l'agrégation

– Coopération (authentification), etc. [11].

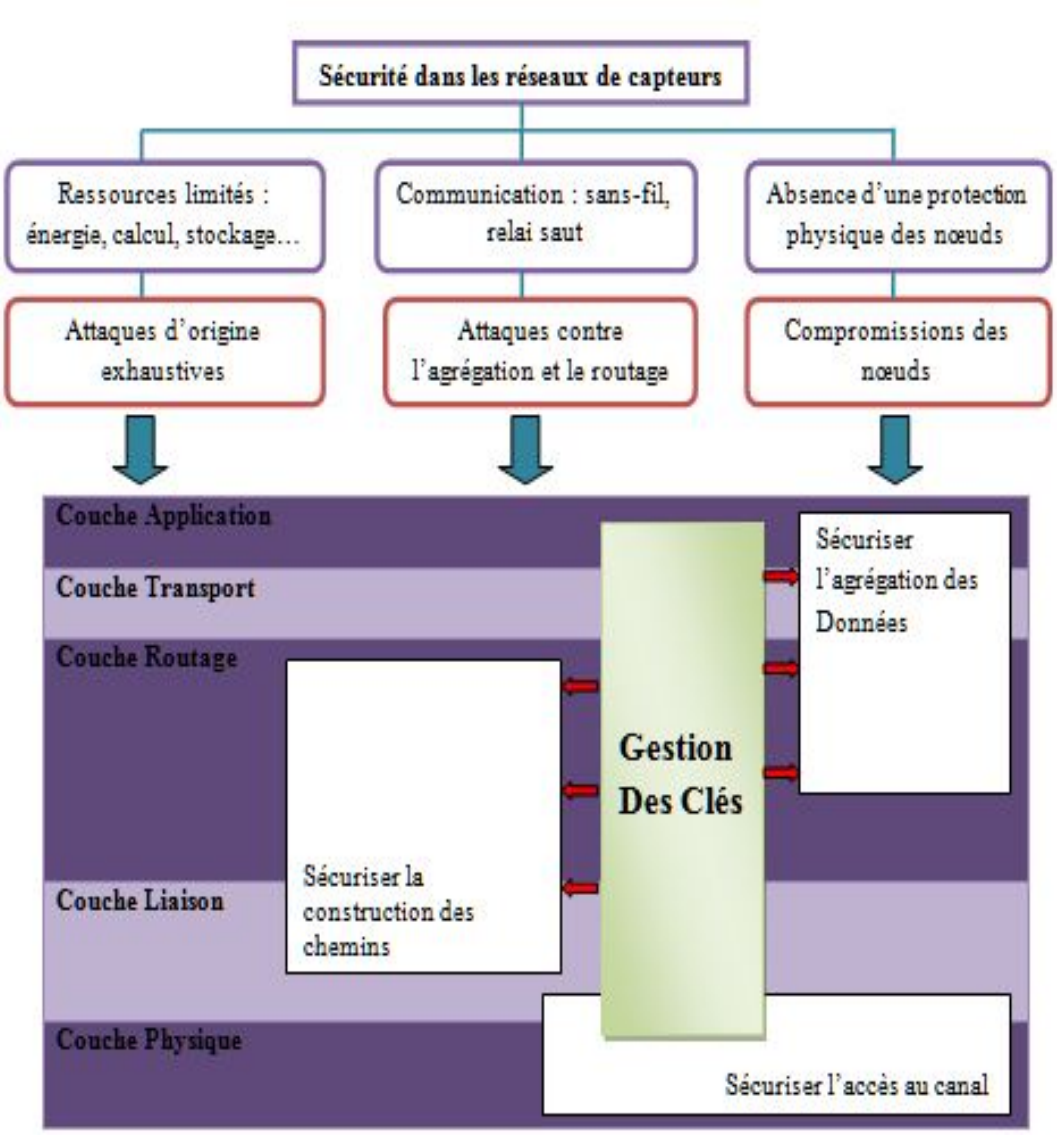


FIGURE 3.2 – Positionnement de la gestion des clés dans un RCSF sécurisé [14].

3.1.2 Propriétés d'une clé

La confidentialité est la principale propriété d'une clé, elle consiste à garantir qu'un utilisateur étranger qui n'appartient pas au réseau au moment présent, ne peut ni calculer ni découvrir la clé. Elle est basée essentiellement sur les deux propriétés suivantes :

- **Confidentialité passé (Backward Secrecy)**

  Consiste à garantir qu'un intrus qui connait un ensemble de clés courantes ne peut déduire les clés antérieures du réseau. Cette propriété permet d'assurer qu'un nouveau membre du réseau ne pourra pas déchiffrer les messages envoyés dans le réseau avant son adhésion.

- **Confidentialité futur (Forward Secrecy)**

  Consiste à garantir qu'un intrus qui connait un ensemble d'anciennes clés du réseau, ne pourra pas déduire les nouvelles clés du réseau. Cette propriété permet d'assurer qu'un ancien membre ne pourra pas déchiffrer les messages destinés au réseau après l'avoir quitter[1].

3.2 Phases de la gestion des clés

3.2.1 Pré-distribution de clés

Chaque nœud doit avoir une clé avant le déploiement. La seule méthode pratiques pour la distribution des clés aux nœuds de RCSF dont la topologie est inconnue avant le déploiement devra compter sur la pré-distribution des clés, des clés doivent être installées dans des nœuds à fin de sécuriser la transmission des paquets comme illustré dans la figure3.3.([16]) [12].

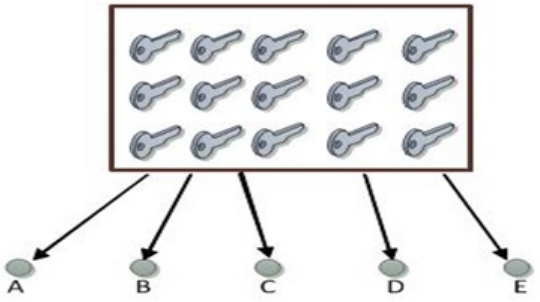


FIGURE 3.3 – Phase principale de pré-distribution de clés.

3.2.2 Découverte de voisinage

Chaque nœud doit découvrir ses voisins dans sa porté sans-fil de communication avec lesquelles il partage des clés. Un lien existe entre deux nœuds capteurs seulement s'ils partagent une clé. Le bon schéma de découverte de voisins ne donnera à un attaquant aucune occasion de découvrir les clés partagées et l'attaquant peut seulement faire l'analyse du trafic [12].

3.2.3 Etablissement de clés de chemin

Une clé de chemin "path key" est importante entre les nœuds non liés directement mais sont reliés par un chemin multi saut pour sécuriser la communication bout à bout, cette clé de chemin ne peut pas être celle déjà employée entre les nœuds voisins. La figure 3.4 [16] illustre les deux phases précédentes :

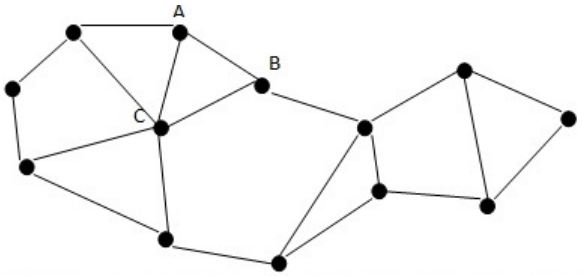


FIGURE 3.4 – Découverte de voisinage et établissement des clés de chemin.

3.2.4 Isolation des nœuds anormaux

On dit qu'un nœud est anormal s'il ne fonctionne pas comme spécifié et indiqué. Identifier et isoler les nœuds anormaux est important pour continuer l'opération du RCSF car ils agissent comme des nœuds intermédiaires. Parmi les raisons pour lesquelles un nœud ne fonctionne plus on trouve : l'épuisement de sa source d'énergie, l'endommagement par un attaquant, la compromission d'un nœud intermédiaire et la corromption de la communication en modifiant les données, ou la compromission du nœud et la communication de l'information factice à SB.

3.2.5 Renouvellement des clés

La vie des clés expire, donc des nouvelles clés doivent être mises en service. 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 [12].

3.2.6 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[12].

3.3 Classification des protocoles de gestion des 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 [12] et ceux utilisant la cryptographie asymétrique, ainsi la figure 3.5 illustre cette classification.

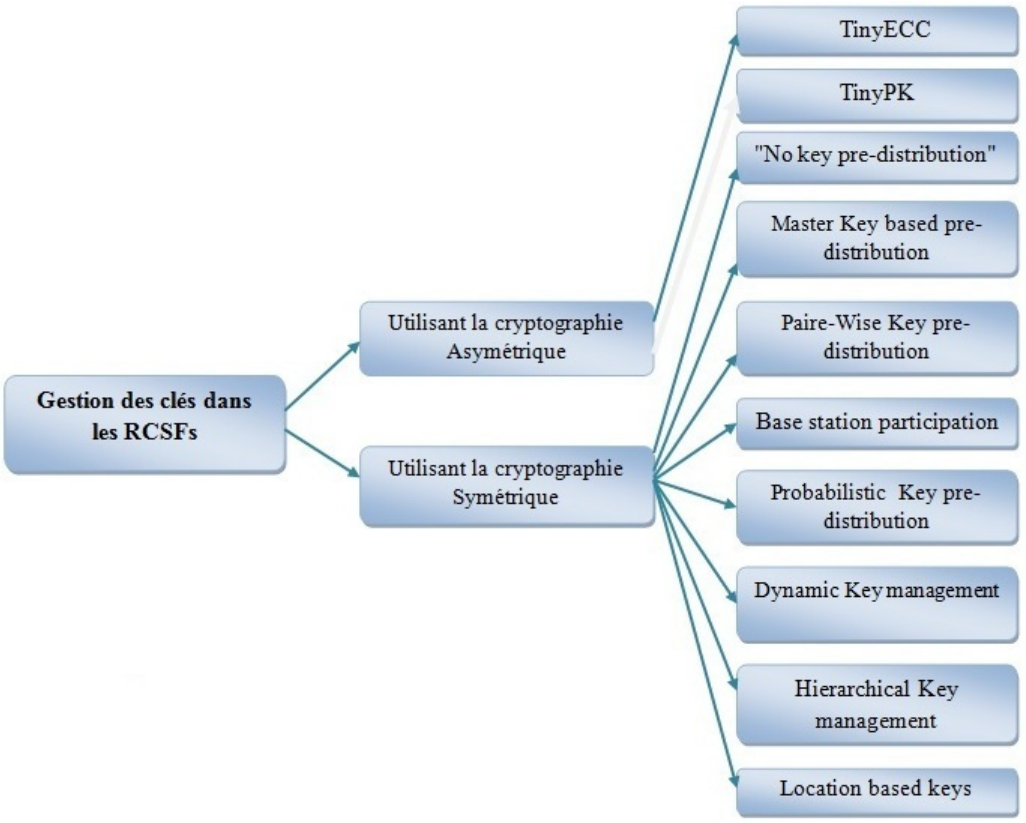


FIGURE 3.5 – Classification des approches de gestion de clés.

3.3.1 Utilisation de la cryptographie asymétrique

Avant le déploiement, chaque nœud du réseau possède la clé maitresse publique et privé (KM, KM^{-1}), puis chaque nœud A génère sa paire de clé (KA, KA^{-1}). Apres le déploiement, les nœuds échangent les clés (échange des clés publiques et une signature par la clé maitresse pour la vérification des clés publiques reçues). En suite une clé symétrique peut être généré et cryptées par leurs clés publiques[12]. L'utilisation de la cryptographie asymétrique dans les réseaux de capteurs sans fils permet la scalabilité et la résistance contre la capture de nœud malgré qu'elle ait été presque universellement considéré comme trop consommatrice de ressources pour l'usage dans les RCSFs et vulnérable vis-à-vis les attaques de déni de service provoquant un calcul qui peut prendre quelques secondes (voir quelques minutes), les

nœuds sont vulnérables à un déni de service d'épuisement de batterie par un attaquant inondant le réseau par des signatures illégales.
La plupart des algorithmes asymétriques sont utilisable dans les MANETs mais pas dans les RCSFs car ce genre de réseau utilise des dispositifs faible en énergie, cependant il y a eu quelques efforts d'adapter les techniques publiques de cryptographie aux dispositifs de capteur.

3.3.1.1 TinyECC

La cryptographie à clé publique (PKC) a été la technologie qui sous-tend les nombreux services de sécurité et des protocoles dans les réseaux traditionnels tels que Internet. Dans le cadre de réseaux de capteurs sans fils, la cryptographie à courbe elliptique (ECC [1]) [19], est l'un parmi les plus efficaces types de PKC [2].ECC [2] est étudié pour fournir un support de PKC dans les applications de réseau de capteurs ainsi que les solutions axées sur la PKC existantes peuvent être exploitées. TinyECC fournit un certain nombre de commutateurs d'optimisation, qui peut activer les optimisations spécifiques ou désactiver selon les exigences des développeurs. Le TinyECC prend en charge seulement les schémas ECC bien étudiés comme ECDSA [1] est une variante de l'algorithme de signature digitale (DSA [2]) [25], ECDH [1] est une variante du Diffie-Hellman protocole principal d'accord [26] relatif aux groupes elliptiques de courbe et ECIES [1] un plan de cryptage à clé publique qui fournit la sécurité sémantique contre un adversaire [27], lesquels sont définis dans les normes de cryptographie efficace [18]. En outre, TinyECC inclut également les paramètres de courbe elliptique [17].
Les courbes elliptiques Utilisé dans les cryptographies sont généralement définis sur deux types de champs finis : champs F_p, où p est un grand nombre premier et un champ binaires d'extension F_{2m}.
Une courbe elliptique E sur F_p est définie par une équation cubique :

1. voir l'Annexe
2. voir la liste des abréviations

$y^2 = x^3 + ax + b$, où a, b $\in F_p$ sont des constantes telles que $4a^3 + 27b^2 \neq 0$[19] [17] Les nœuds A et B se mettent d'accord (publiquement) sur une courbe elliptique E (a, b, p), c'est-à-dire qu'ils choisissent une courbe elliptique. Ils se mettent aussi d'accord (toujours publiquement) sur un point P situe sur la courbe. Secrètement, le nœud A choisit un entier d_A, et B un entier d_B, A envoie à B le point d_A P, et B envoie à A d_B P. Chacun est capable de calculer d_A (d_B P) = (d_A d_B) P qui est un point de la courbe, et constitue leurs clé secrète commune. Si un adversaire a espionné leurs échanges, il connait E (a, b, p), P, d_A P, d_B P, Pour pouvoir calculer d_A d_B P, il faut pouvoir calculer d_A connaissant P et d_A P. C'est ce que l'on appelle résoudre le logarithme discret sur une courbe elliptique. Or, actuellement si les nombres sont suffisamment grands, on ne connait pas la méthode efficace pour résoudre ce problème en un temps raisonnable [20][12].

L'inconvénient est que l'application de cette méthode de la cryptographie asymétrique reste toujours complexe car TinyECC offre la possibilité de chiffrer et d'authentifier des données avec des algorithmes à base de courbes elliptiques (ECDSA [2], ECIES [2], ECDH [2]). Le temps d'exécution de cette solution dépasse la minute à quelques millisecondes selon le type d'architecture utilisé qui rend cette solution impraticable sur tout type de capteur.

Malgré tout ils ont prouvés que l'utilisation d'ECC est un excellent choix pour faire la cryptographie asymétrique dans les RCSFs puisque ils nécessitent des tailles de clés inférieures à RSA [1] [70].

3.3.2 Utilisation de la cryptographie symétrique

Bien que la cryptographie asymétrique comporte des avantages certains par rapport à la cryptographie à clé symétriques et malgré les recherches qui visent à les appliquer aux RCSFs, la cryptographie à clé symétrique possède ses propres qualités

2. voir la liste des abréviations
1. voir l'Annexe

qui la rend toujours la plus préféré pour les RCSFs. Pour cette raison la plupart des schémas de gestion des clés proposées pour les réseaux de capteurs sont basés sur la cryptographie symétrique [13].

3.3.2.1 Absence de Pré-distribution de clés "No Key pre-distribution"

Ce mécanisme ne prend en considération aucune pré-distribution de clés. Par ailleurs si un adversaire ne sait pas ou 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 est n'est souvent pas réaliste, dans la plupart des applications, il n'est capable de surveiller qu'une certaine petite portion du trafic pendant la phase de déploiement initial.

- **'Key infection'**

  Anderson et autres ont proposés "INF" [15] un schéma de gestion de clés prévu pour les RCSFs. Le protocole est fondé sur l'hypothèse que, pendant la phase de déploiement, l'attaquant peut surveiller seulement un pourcentage fixe des voies de transmission. INF suppose un déploiement de masse et que les nœuds sont statiques, chaque nœud du réseau génère et partage une clé symétrique avec son voisin à un saut. La sécurité est basée sur la surprise : elle est fondée sur le modèle réaliste [43] de l'attaquant. En premier lieu, chaque nœud génère simplement une clé symétrique et la diffuse sur le réseau à ses voisins. Une approche de chuchotement de clé est employée, c.-à-d, la clé est au commencement transmise à un niveau de puissance bas. La puissance de transmission est alors augmentée jusqu'à ce que la clé soit entendue par au moins un voisin d'un seul saut et qu'une réponse soit reçue. Dans l'hypothèse posée par l'auteur, un attaquant a très peu de probabilité d'intercepter une telle communication locale et de faible porté, il ne peut pas surveiller tout les nœuds déployés.

  Le protocole se déroule en trois phases, au départ chaque nœud A diffuse sa

clé sur le réseau vers ces voisins, si on suppose que les nœuds B et C entendent le message de A (B et C sont les voisins de A à un saut), alors chacun d'eux va générer une clé de session K_{AB} (K_{AC} respectivement) , cette clé est employée pour protéger le trafic entre le nœud A et le nœud B (rentre le nœud A et le nœud C respectivement), en suite les nœuds B et C envoient une réponse au nœud A avec un message chiffré par la clé symétrique de A constitué de leurs identités ainsi que la clé de session déjà générée, ainsi le nœud B envoie {B, $K_{AB}\}K_A$ et le nœud C envoie {C, $K_{AC}\}K_A$.

L'avantage de ce protocole est que la station de base est absente dans la phase d'installation des clés, ce qui diminue la consommation d'énergie, de plus l'absence de la phase de déploiement des clés dans le réseau. INF est simple mais il n'assure pas une bonne sécurité du réseau.

3.3.2.2 Les protocoles de gestion de clés basée 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 charger les clés dans les nœuds avant leurs déploiement. La figue 3.6 illustre une taxonomie des solutions de gestion des clés basées sur la pré-distribution.

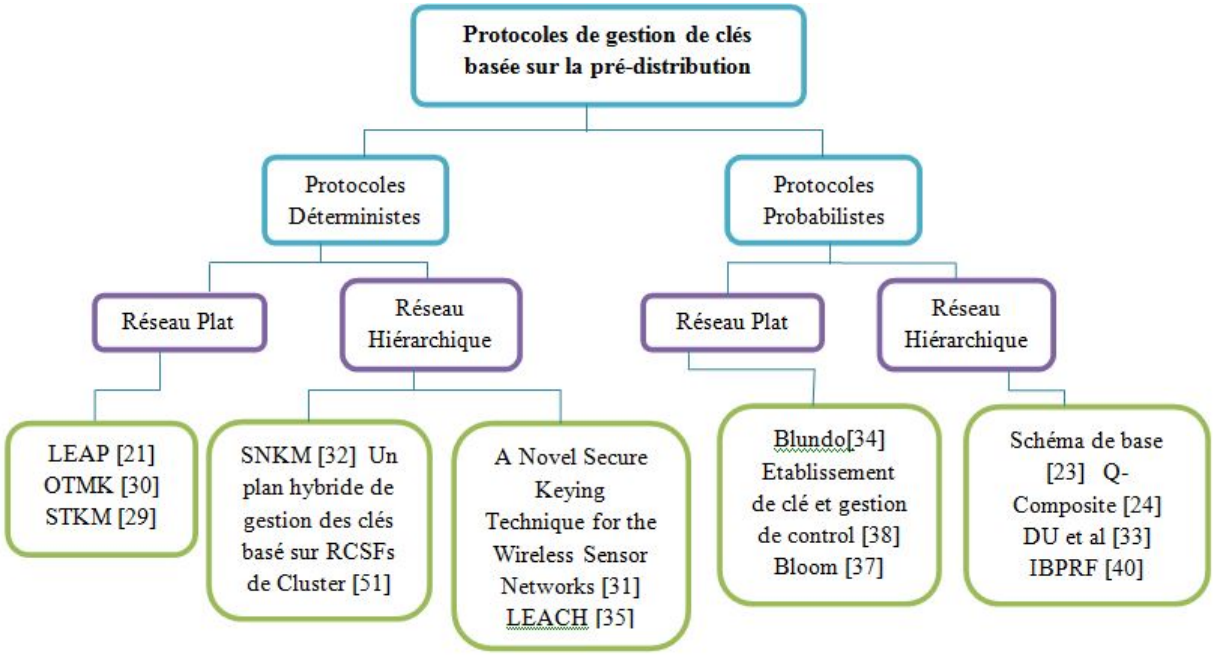


FIGURE 3.6 – Taxonomie de pré-distribution de clé pour les $RCSF_s$.

Nous allons maintenant étudier quelques protocoles de gestion de clés basés sur la pré-distribution. Les protocoles sont classifiés dans plusieurs catégories selon la topologie du réseau (hiérarchique [1] ou plate [1]) et la façon avec laquelle les nœuds voisins partagent des clés communes (probabiliste ou déterministe).

A. Les protocoles déterministes

Les protocoles de gestion de clés déterministes assurent que chaque nœud est capable d'établir une clé par-paire avec ses voisins. Pour garantir le déterminisme, les protocoles, tels que LEAP [2] [21] et OTMK [2] [30], utilisent une clé commune, transitoire et pré-chargée dans tous les nœuds avant leurs déploiements. Cette clé est utilisée en vue de générer des clés par-paires entre chacun des deux nœuds voisins. Pour sécuriser les nœuds contre les attaques de captures, la clé transitoire est éffacée du nœud après la génération des clés par-paires [14].

1. voir l'Annexe
2. voir la liste des abréviations

A.1. **LEAP**

LEAP[21] est un protocole déterministe de gestion des clés pour les $RCSF_s$. Le mécanisme de gestion des clés fourni par LEAP support le traitement interne «in-network processing [1]» tout en limitant l'impact de la sécurité d'un nœud compromis sur son voisinage immédiat dans le réseau. LEAP support l'établissement de quatre types de clés pour chaque nœud : Une clé individuelle partagée avec SB, Une clé par-paire partagée avec les autres nœuds du réseau, Une clé de groupe partagée entre tous les nœuds du réseau et une clé globale partagée avec les voisins multiple des nœuds. Il est conçu pour assurer la sécurité des communications dans les RCSFs, donc il fourni les services de bases de sécurité comme : la confidentialité et l'authentification. Cependant il doit répondre aux différentes exigences de sécurité et d'exécution qui sont considérablement plus provoquant aux $RCSF_s$.

- **La clé individuelle :** Chaque nœud possède une clé unique qu'il partage avec SB. Cette clé est employée pour sécuriser la communication entre le nœud et SB.
- **La clé du groupe :** C'est une clé globalement partagée, elle est utilisée par SB pour chiffrer les messages et les envoyer aux membres du groupe.
- **La clé globale :** C'est une clé partagée par un nœud avec tous ces voisins, elle est principalement utilisée pour sécuriser les messages diffusés.
- **La clé par paire :** Chaque nœud partage une clé principale avec chacun de ses voisins immédiats.

  a) Hypothèses de fonctionnement

  LEAP se base sur les hypothèses suivantes :

  – Le réseau est statique.

  – Les nœuds capteurs sont homogènes (réseau plat).

  – Le déploiement est aléatoire.

  – La révocation des clés (ou l'expulsion des nœuds) est exécutée après la

détection des nœuds compromis.

- Le nœud de commande est informé par le mécanisme de détection d'intrusion des nœuds compromis.
- Un adversaire peut capturer tout le trafic, injecter des paquets et répondre aux anciens messages dans le réseau.
- La compromission d'un nœud implique la compromission de toutes les informations contenues dans ce nœud.
- SB ne peut être compromise.

b) Notation

La notation suivante est utilisée dans LEAP :

Notation	Description
N	Nombre de nœuds dans le réseau.
u, v	Identificateurs des nœuds communiquants.
$\{f_k\}$	Une fonction pseudo-aléatoire [22].
$\{s_k\}$	Message s crypté avec la clé K.
MAC (K,s)	Authentification du message s en utilisant la clé symétrique K.

TABLE 3.1 – Les différentes notations utilisées dans LEAP.

c) Principe du schéma

Les auteurs de LEAP exploitent la propriété statique des $RCSF_s$, c'est à dire que l'ensemble des voisins d'un nœud est relativement statique. Un nœud dans le réseau peut découvrir la plus part de ces voisins dans le temps initial de déploiement. Cependant, l'établissement de clés entre un nœud et ses voisins (pair-wise key) se fait en quatre étapes, et qui sont :

- **Pré-distribution de clé**

  Le contrôleur (SB) génère une clé initiale K_{IN} et charge chaque nœud avec cette clé. Chaque nœud u dérive une clé principale (Maser Key) $K_u = f_{KIN}(u)$, f_k étant une fonction pseudo-aléatoire.

- **Découverte des voisins**

  Immédiatement après son déploiement, le nœud u essaye de découvrir

ses voisins en diffusant un message HELLO qui contient son id. Aussi, il initie un timer. Le nœud 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 K_v, qui est dérivée comme suit : $K_v = f_{KIN}(v)$. Comme le nœud u a la clé K_{IN}, il pourra aussi dériver K_v, ainsi il pourra vérifier l'authenticité de l'ACK reçus :

u = = > *, u

v = = > u, v | MAC (K_v, u|v)

- **Etablissement de la clé par-pair**

  Le nœud u calcule sa clé par paire K_{uv} avec v, comme suit : $K_{uv} = f_{k_v}(u)$. Le nœud v peut de même calculer K_{uv} de la même manière. K_{uv} sert comme clé entre u et v.

- **Effacement des clés**

  Lorsque le timer expire, le nœud u efface K_{IN} et toutes les clés principales (K_v) de ses voisins. Il est à noter que le nœud u n'efface pas sa clé principale K_u.

A la fin de ces quatre étapes, le nœud u aura établi une clé par paire partagée avec chacun de ses voisins. Cette clé sera utilisée pour sécuriser les données échangées entre eux. Ainsi pendant une communication sécurisée, deux nœuds n'ont pas besoin d'établir une clé par-pair pour une direction et une autre clé par-pair dans la direction inverse. De plus, aucun nœud dans le réseau ne possède la clé K_{IN}. Un adversaire peut écouter clandestinement tout le trafic dans cette phase, mais sans la clé K_{IN} il ne peut injecter des informations incorrectes ou déchiffrer les messages. Un adversaire compromettant un nœud après T_{min}, obtient seulement les clés du nœud compromis. Quand un nœud compromis est détecté, ses voisins suppriment simplement les clés qui ont été partagée avec lui. Cependant, le message HELLO est non authentifié, et donc un adversaire peut exploiter ceci pour lancer des attaques de consommation

de ressources (DoS [1]) en injectant un grand nombre de messages HELLO.

A.2. **OTMK**

Les auteurs d'OTMK [30] ont proposés une solution de gestion de clé basé sur le concept de la clé initiale transitoire de LEAP. OTMK est un protocole déterministe qui permet l'établissement des clés par-paires entre les nœuds voisins. Pour réduire les conséquences de la compromission de la clé initiale K_{IN} par un adversaire, deux propriétés doivent être vérifiées dans le développement des solutions tolérantes de gestion de clés.

(1) **La propriété d'opacité :** Un adversaire ne peut pas déduire la plupart des clés utilisées dans le réseau par la compromission d'un petit nombre de nœuds.

(2) **La propriété d'inoculation :** Un adversaire ne peut pas aider un nœud étranger de joindre le réseau par la compromission d'un petit nombre de nœuds. Contrairement au protocole LEAP, dans OTMK, même si la clé initiale K_{IN} est compromise, la propriété d'opacité reste vérifiée. Cependant, la propriété d'inoculation est non vérifiée dans les deux protocoles.

a) Notation

La notation suivante est utilisée dans OTMK :

Notation	**Description**
u $\rightarrow$ v	Le nœud u envoi un message au nœud v.
u $\rightarrow$ *	Emission du nœud u à tous les voisins.
a || b	Le message a concaténé avec le message b
$\{E_k\}$(p)	Le message p chiffré avec la clé
$\{M_K\}$(p)	Produire du MAC pour le message p avec la clé K.
ID_u	Identité du nœud u.
nonce	Un nombre aléatoire.

TABLE 3.2 – Les différentes notations utilisées dans OTMK.

1. voir l'Annexe

b) Le schéma de base

Ce protocole consiste à préconfiguré chaque nœud du réseau avant la phase de déploiement avec une même clé initiale K_{IN}. Pour établir des clés par-paires avec ses voisins, chaque nœud u diffuse un message JOIN comme suit :

u→* : JOIN || EK_{IN} (ID_u ; nonce + 1).

Lorsque le nœud v reçoit ce message, il génère un nombre aléatoire $K_{v;u}$, et envoie le message REPLY suivant à u :

v → u : REPLY || E_{KIN} (ID_v, nonce + 1 ||$K_{v;u}$).

Lorsque le nœud u reçoit ce message, il le décrypte puis vérifie le nonce. Si c'est vérifié, il enregistre le nœud v comme étant voisin vérifié. La clé par-paire entre le nœud u et le nœud v est soit $K_{v,u}$ généré par le nœud v, ou bien $K_{u,v}$ généré par le nœud u. si $ID_u < ID_v$ alors les nœuds u et v utilisent la clé $K_{u,v}$ comme clé par-paire entre eux, dans le cas contraire ils utilisent la clé Kv,u comme clé par-paire.

A fin de réduire les chances de compromettre la clé initiale K_{IN}. Chaque nœud, après un temps suffisant pour l'établissement des clés par-paires, détruit la clé initiale de sa mémoire.

Dans le cas de la compromission de la clé initiale K_{IN}, OTMK préserve la propriété d'opacité et non celle d'inclusion. La clé initiale est utilisée seulement pour authentifier des nœuds légitimes, et ne contribue pas aux calculs des clés par-paires. Si un adversaire compromet la clé initiale K_{IN} après qu'un nœud a établi des clés par-paires avec ses voisins, l'adversaire ne peut pas déduire ces clés. Par contre, si un nœud adversaire compromet la clé initiale avant l'établissement des clés par-paires, il peut intercepter toutes les clés par-paires qui sont entrain d'être échangées en observant les messages REPLY. Cependant, cet adversaire peut déduire les clés par-paires d'une petite partie du réseau. Ainsi l'opacité est préservée avant et après la compromission de la clé initiale. Contrairement, dans LEAP, la compromission de la clé initiale à n'importe

quel moment permet à un adversaire de déduire toutes les clés par-paires installées dans le réseau.

A.3. **STKM**

Ce schéma de gestion de clé STKM [2] [29] est basé sur la cryptographie symétrique, qui utilise une méthode de pré-distribution de clé avec renouvellement périodique ou à la demande. Ce schéma se base sur les hypothèses suivantes :

- Le réseau de capteurs est statique (Les nœuds ne sont pas mobiles).
- Les nœuds capteurs sont homogènes : les nœuds capteurs sont similaires dans leurs capacité de traitement, de communication, d'énergie et de stockage.
- Le déploiement est aléatoire : les voisins de n'importe quel nœud ne sont pas connus avant le déploiement.
- Un attaquant peut écouter tout le trafic, renvoyer d'anciens messages, ou injecter ses propres messages.
- La compromission d'un nœud implique que toutes les informations stockées dans sa mémoire sont connues par l'attaquant.
- SB n'a pas de contraintes sur les capacités de calcul, de stockage, et ne peut être compromise.
- 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.

a) Notation

La notation suivante est utilisée :

2. voir la liste des abréviations

Notation	Description
S_i	L'i-ème nœud capteur dans le réseau,S_i dénote l'identificateur (unique) du nœud.
M_K	Le cryptage du message M par la clé K.
SB $\rightarrow$ * : M	La station de base diffuse le message M. tout nœud dans le rayon de perception de SB reçoit le message M.
MAC_K(M)	Le Message d'authentification de M avec la clé symétrique k.
A || B	La concaténation de l'information A avec l'information B.
N_i	Un nonce généré par le nœud S_i.
H_K(M)	Une fonction de hachage à sens unique appliquée à la chaine de caractères M utilisant la clé k.

TABLE 3.3 – Les différentes notations utilisées dans STKM.

b) Idée de base

L'idée de base de ce schéma est de construire un arbre couvrant de manière sécurisée et conservant l'énergie après un déploiement aléatoire des nœuds, cet arbre est par la suite utilisé pour le renouvellement de clés. SB est l'initiateur de l'algorithme, et chaque nœud tire profit des messages reçus, même si le message n'est pas destiné à ces nœuds, ce qui permet la réduction de nombre de messages transmis, et par conséquant, minimiser la consommation d'énergie.

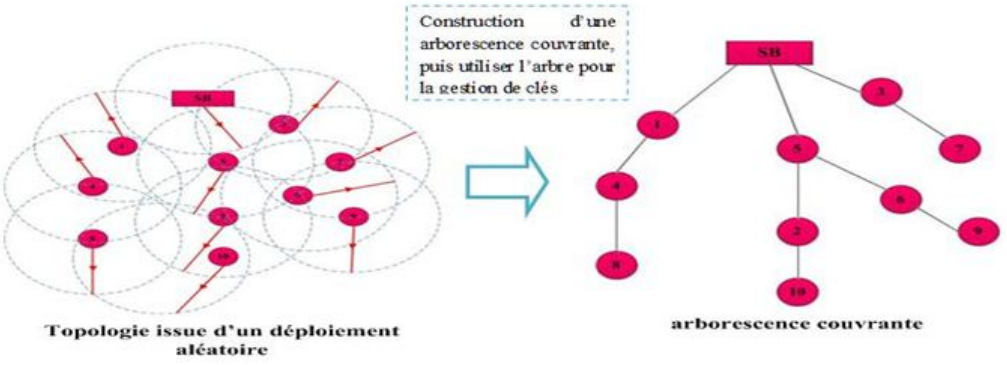


FIGURE 3.7 – Construction d'une arborescence couvrante.

c) Schéma proposé

Les différentes phases de ce schéma sont :

- **Pré-distribution de clé :** Cette phase est exécutée par SB avant le déploiement :

  Etape 1 : pour chaque nœud capteur S_i du réseau est assigné un identificateur unique.

  Etape 2 : chaque nœud possède une clé partagée avec SB $K_{SB,i}$ pour chiffrer les messages du nœud vers SB, et une autre clé $K_{SB,i}$ pour les messages du sens inverse. Ces deux clés sont utilisées pour sécuriser les communications entre les nœuds et SB et vis versa. Une clé K_r est partagée par tous les nœuds du réseau, elle est utilisée pour chiffrer les messages justes après le déploiement
- **Construction de l'arbre :** Après le déploiement, chaque nœud copie sa clé K_r dans sa RAM (mémoire volatile), et la supprime de la mémoire non volatile (EROM). Si un attaquant capture un nœud quelques secondes après le déploiement, il n'aura pas accès à la clé K_r. SB initie l'algorithme par la diffusion du message suivant :

  SB $\rightarrow$ * :{ SB, HELLO, 0,- -, MAC_{Kr} (SB, CPT)}K_r

  Le but de ce message est de découvrir les nœuds voisins de SB, CPT est un compteur initialisé à zéro et reflète le niveau dans l'arbre. Le MAC du compteur et de l'identificateur de la source du message est calculé, le tout est chiffré par la clé K_r.
- **Maintenance de l'arbre et rafraichissement de clés :** A la fin de la phase précédente, chaque nœud partage :une clé symétrique avec SB, une autre clé symétrique avec son nœud père, et la clé K_r partagée par tout le réseau. Si un nœud père détecte qu'un de ses fils est malveillant, il ignore ses messages, et il le supprime de la liste des fils. Dans le cas inverse, où un fils détecte que son nœud père est malveillant, et si sa liste de voisins n'est pas vide, il choisi un de ses voisins comme père en lui envoyant un message pour l'informer.

Un renouvellement de clés est lancé par SB, soit d'une manière périodique ou à la demande. Pour chaque nœud voisin S_i, SB envoie le message :

SB, REFRESH, N_{SB}, MAC_{Kr} (SB, N_{SB})$K_{SB,i}$;

Un nœud fils de la station de base reçoit le message, rafraîchit la clé K_r par : $K_r := HK_r$ (K_r || N_{SB}), et envoie à chacun de ses fils un message (crypté par la clé symétrique partagée entre père et fils) de rafraîchissement de clé avec le même nonce N_{SB} (pour avoir la même clé K_r).ainsi, le message de rafraîchissement est propagé dans l'arbre et la clé renouvelée est par la suite utilisée.

A.4. **A Novel Secure Keying Technique for the Wireless Sensor Networks**

Les Différentes techniques de gestion de clés ont leurs propres forces et faiblesses comme énuméré dans [31]. Ainsi un bon mécanisme de gestion de clés devrait se composer d'une combinaison de variété de clés comme, les clés du réseau (in-network generated keys), les clés de pré-distribution (pre-deployed keys), et les clés de diffusion (broadcast keys). Ce schéma est motivé par le fait que les différents types de messages échangés entre les nœuds capteurs ont différentes exigences de sécurité.

a) Schéma proposé

Une technique de gestion simple n'est pas appropriée à sécuriser tous les types de communications dans les RCSFs. Les différentes clés utilisées dans ce schéma sont :

- **Kb (Buddy Key) :** Elle est calculée par tous les nœuds capteurs après que la phase de découverte de voisins soit terminée.
- **Ko (My-Own-Key) :** Chaque nœud capteur est pré chargé avec son identificateur, ce dernier est utilisé pour le calcul de Ko.
- **Kn (Network Key) :** Quand un nœud veut joindre le réseau il envoie une

demande à SB pour acquérir le Kn principal de réseau. Seulement le nœud qui est authentifié acquiert Kn.

- **Kc (Cluster Key) :** Elle est calculée par le CHs.
- **Kbro (Broadcast Key) :** Elle est publiée par SB après qu'elle soit authentifiée comme CHs.

b) Les différentes phases

Ce schéma est divisé en plusieurs phases comme illustré ci-dessous :

- **Phase 1 : la pré-distribution de clés :** Avant le déploiement, chaque nœud possède son propre identificateur (id-nœud), qui est utilisé pour le calcul de la clé Ko (My-Own-Key).
- **Phase 2 : la formation de topologie et la découverte des voisins :** Cette phase est détaillée dans [28]. Pour récapituler, une topologie groupée hiérarchique est adaptée. Le réseau est divisé en plusieurs secteurs, plus de détailles sont disponibles dans [41]. Après cette phase le choix du cluster Head (CHs) est fait. SB va sélectionner en suite le DCH [2] et le RCH [2] de chaque secteur, les détails concernant DCH/RCH sont disponibles dans [28].
- **Phase 3 : le calcul de la clé Ko :** La clé Ko est calculée comme suit : f(nœud-id, secteur-id, niveau d'énergie restant).
- **Phase 4 :** Authentification de nœuds par les SB.

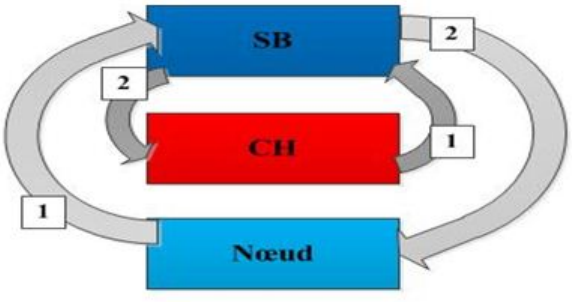


FIGURE 3.8 – Authentification d'un nœud.

2. voir la liste des abréviations

– **Phase 5 :** Elle concerne l'acquisition de la clé du réseau et la clé de diffusion. Le calcul principal de la clé du réseau se compose de (a) l'établissement de la clé Kb entre les voisins et (b) l'établissement de la clé Kc entre le DCH et le RCH. L'acquisition de la clé de diffusion Kb se compose de (a) la communication entre SB et tout les nœuds du réseau en prenant en considération le rôle du CH, et (b) une requête pour acquérir la clé Kbro par le CHs.

– **Phase 6 : Clé de Rafraîchissement/Maintenance :** La clé Kbro calculé par le RCH est rafraichie à chaque fois que le CH change. Aussi, la clé Kn est rafraichie par SB dans un intervalle régulier du temps. Le rafraîchissement principal s'assure que les nœuds appartenant au réseau sont bien authentifiés du temps en temps. Également chaque nœud dans le réseau maintient la table de toutes les clés mentionnées dans la section précédente.

A.5. **LEACH**

Le RCSF est caractérisé par le fait que les ressources sont limitées notamment l'énergie des nœuds capteurs. L'organisation de Cluster-base a été proposée pour fournir une manière efficace d'économiser l'énergie pendant la communication. Dans ce genre d'organisation, les nœuds sont organisés en Clusters. Les Cluster Heads (CHs [2]) passent des messages entre un groupe de nœuds (groupe pour chaque CH) et SB. Selon cette organisation, LEACH [35] a ajouté une autre issue intéressante à ce genre de réseau qui est la sécurité.

Ce schéma a été proposé pour la première fois dans le but de réduire la consommation d'énergie totale dans les $RCSF_s$. L'auteur de LEACH [2] suppose que chaque nœud peut directement communiquer avec les SB en employant une haute puissance de transmission, ainsi, pour avoir un équilibrage de consommation d'énergie une hiérarchie groupée est appliquée. Ce processus a comme conséquence l'économie d'énergie pour les nœuds qui ne sont pas impliqués

2. voir la liste des abréviations

dans CHs puisqu'ils peuvent transmettre par une puissance minimum de transmission, mais en même temps l'énergie des CHs est consommés. Pour résoudre ce problème, LEACH propose un CH dynamique rotationnelle avec changement du CHs à chaque ronde. A chaque ronde, un nouveau nœud deviendra un CH, le réseau choisit ce dernier en utilisant un algorithme distribué.

a) Schéma proposé

LEACH se compose de deux phases contenant cinq étapes à chaque ronde. Les deux phases sont :

- **La phase d'installation (phase initiale) :** Chaque nœud décide la probabilité qu'il soit un CH pour le ronde courant en prenant en considération l'énergie et la connaissance du pourcentage désiré du CHs. Appelons ces derniers des RNs, alors ils diffusent des messages d'annonce pour le réseau tout entier. Ainsi, lorsque le reste des nœuds reçoivent tous les messages d'annonce, ils choisissent un CH selon les signaux les plus élevés reçus de RNs ; puis chacun de ces nœuds envoie une requête au CH désiré pour le joindre. Ainsi, quand le CH reçoit les messages, il commence à diffuser des confirmations pour les nœuds acceptés avec un bilan de tranche de temps pour chacun d'eux. Ce dernier sert à informer chaque membre du groupe de l'heur de transmission des messages.
- **La phase d'état stable (une vraie phase de transmission) :** Cette phase concerne la vraie transmission de données. selon le programme fourni par le CH aux autres nœuds, chacun commence à envoyer ses données au CH approprié. Les CHs alors rassemblent les messages des membres, les analysent et manipulent, puis envoient les résultats à SB. Dans LEACH, la possibilité pour que le réseau soit attaqué est très petite, parce que le CH changent à chaque ronde de communication, le rendant dur pour que les intrus connaissent le CH prévu pour chaque ronde de sorte qu'elles puissent perturber les points critiques du réseau [36].

A.6. **Un plan hybride de gestion des clés basé sur les réseaux sans fils de Cluster**

Ce schéma [51] est basé sur la construction d'un arbre de clé de d dimension entre le cluster Head et SB comme illustré dans la figure 3.9. Il considère que le cluster Head dans le RCSF en cluster a une capacité plus élevée en traitement et en stockage d'informations que les nœuds normaux.

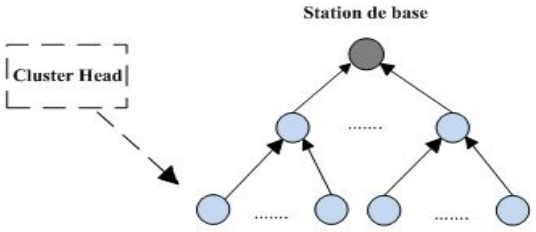


FIGURE 3.9 – Arborescence de la clé.

a) Notation

Les notations suivantes sont utilisées dans ce schéma :

Notation	Description
K_i	Une clé privée.
ID_u, ID_v	L'identificateur local du nœud u/v.
r	Le rayon de correspondance.
PK (r)	La clé de session.
R	Le nembre aléatoire qui est produit par la station de base.
K_m	La clé principale pré-chargée.
S_u, S_v	Présente des informations de localisation de u/v.

TABLE 3.4 – les différentes notations utilisées dans [51].

b) Schéma proposé

Ce schéma se déroule en trois phases, expliquées ci-dessous :

– **Phase de Pré-distribution de clé :** Avant le déploiement, chaque nœud charge k_M à l'avance et produit aléatoirement son identificateur. Et chaque

Cluster Head sera stocke également une clé privée, et une clé de session PK(r) qui est initialement égale au k_M.

– **Phase d'établissement de clé :** Cette phase est divisée en deux types :

1) **Construire un arbre principal entre cluster Head et la station de base :** Chaque membres de l'arbre stocke une clé privée K_i,SB et le cluster Head partagent une fonction PK(r) = E(k_M, R). SB chiffre R avec chaque clé privée K_i et l'envoie alors à chaque cluster Head. Selon la clé k_M, le cluster Head produit une clé de session en utilisant la fonction précédente.

2) **Etablir la clé de communication entre les membres du même Cluster :** Selon la clé k_M, le nœud génère une clé unique et une paire de clé adjacente basées sur l'emplacement. Comme le nœud u, basé sur son emplacement et l'identification locale il aura donc une clé unique : K_u=FKm(ID_u,S_u). Afin de communiquer avec des nœuds voisins, ce message doit être diffusé : u→* :,S_u,MAC_{km}(ID_u,S_u). Quand u et v se sont trouvés, ils utilisent la clé k_M selon l'équation : $K_{u,v}$=FK_m(ID_u,S_u,ID_v,S_v) qui donne accès direct à la clé de session.

– **Phase de maintenance de clé :** Dans cette phase, l'auteur traite quelques situations des nœuds telle que :

1) **Une augmentation dans le Cluster Head :** Le nœud u a récemment joint le nœud v. Le nœud v pendant la phase initiale fait une diffusion d'un message pour annoncer ses propres moyens de localisation et sa position. Quand le nœud u reçoit ce message, vérifie si le nouveau nœud jointif est à côté de lui, en employant l'équation suivante :$|Su - Sv| \leqslant r$ pour confirmer. Si la condition ne satisfait pas avec cette équation, directement le nœud u reçoit des informations sur le nœud v . Puisque le nœud v conné k_M et l'information de position d'inscription locale aussi bien que de deux nœud, peut directement obtenir la clé de session partagée avec le nœud voisin u (équations PK (r)).En conclusion, le nœud v chiffre la clé partagé en utilisant la clé du nœud u (K_u),

puis l'envoyer au nœud u pour établir la clé de session voisine.

2) **Quitte et trahison dans le Cluster :** La trahison du nœud aura seulement effet avec les connexions associées, d'autres connexions dans les réseaux de capteurs sont sûre. SB envoie l'information du nœud de révolte chiffrée avec PK (r) à chaque Cluster Head. Ce dernier fait suivre l'information reçu à tous les membres du même Cluster, les membres du Cluster vérifient si la clé est partagée avec le nœud de révolte, si c'est le cas ils suppriment alors la connexion.

3) **Échange de Cluster Head :** Si le Cluster Head est remplacée, les clés correspondantes sont mises à jour également en même temps. Après l'authentification de nouveaux membres, SB met à jour la clé PK (r) au PK (r+ 1) (c.-à-d., le PK (r+ 1) = E (PK (r), R)), distribuant toutes les clés et partageant la fonction E(PK, R) du nœud de feuille au nœud de racine, SB envoie le PK (r+ 1) qui est chiffrée par le PK (r) à l'autre membre de l'arbre par l'émission multiple. Au contraire, quand le Cluster Head quittE le réseau, SB utilise la clé privée du nœud pour chiffrer R, et l'envoie à chaque Cluster Head par l'émission multiple, et selon la fonction PK (r+ 1), les nœuds obtiennent la nouvelle clé de session PK (r+ 1) pour accomplir la mise à jour.

A.7. **SNKM**

SNKM [32] est un schéma de gestion de clé basé sur la sécurité des nœuds pour les clusters dans les $RCSF_s$. Ce schéma utilise plusieurs genres de clés. Ainsi, Selon différents genres de paquets de données, les nœuds peuvent choisir différentes clés pour le chiffrage et l'authentification. SNKM [2] a été proposé afin d'améliorer le degré de sécurité des clusters Head et réduire l'énergie consommé pour l'établissement des clusters.

2. voir la liste des abréviations

a) **Hypothèses**

Ce schéma repose sur les hypothèses suivantes :

- Le sink possède suffisamment d'énergie.
- Les nœuds capteurs sont statiques, et ils possèdent les mêmes communications et les mêmes capacités de calcul.
- Chaque paquet de données est sur 36 bits.
- Chaque nœud possède suffisamment d'espace pour mémoriser des informations sur la clé.
- Avant le déploiement, les nœuds ne connaissent pas leurs voisins.
- Avant le temps T_{test} quand la phase de pré-distribution des nœuds soit terminée, un attaquant ne pourra pas obtenir des informations sur les nœuds.
- Après T_{min} secondes, des informations sur les nœuds capturés peuvent être obtenus.
- Le sink est absolument en sécurité et il ne pourra pas être capturé.

b) **Schéma proposé**

Selon les différents niveaux de sécurité des nœuds, les auteurs de SNKM adoptent différents schémas de sécurité et différents types de clés. Le Cluster Head joue un rôle important dans les $RCSF_s$, ainsi sa sécurité doit être assurée. Une fois qu'un comportement anormal du cluster Head a été constaté, un nouveau Cluster Head doit être initié immédiatement. Cependant les différents systèmes de sécurité et clés utilisées dans ce schéma sont les suivantes :

- **Les nœuds de sécurité :** Selon une fonction aléatoire, le nœud calcule un nombre aléatoire. Si ce nombre est plus grand que T, alors ce nœud peu être un nœud de sécurité. Une fois qu'un nœud détecte un comportement anormal de ses voisins, il envoie un rapport au nœud de sécurité. La clé du nœud K_u est une clé entre un nœud et le sink. Elle est utilisée par un nœud pour chiffrer les informations à envoyer vers le sink.

- **La clé par-paire :** Avant le déploiement, les nœuds ne connaissent pas leurs voisins. Ainsi, lorsqu'un nouveau nœud u joint le réseau, il essaye de découvrir ses voisins, il diffuse donc des paquets Hello contenant son ID et il se met en attente des réponses de ses voisins. Lorsqu'un nœud voisin v reçoit le paquet Hello, il lui répond par un ACK, et chiffre l'information avec la clé publique K_g. En suite, lorsque le nœud u reçoit l'ACK il calcul alors la clé par-paire $K_{u;v}$ entre eux.
- **La clé du Cluster :** Cette clé est négociée par les nœuds de sécurité. Au début chaque nœud de sécurité produit une clé aléatoire K_u^c et l'envoie vers tous les autres nœuds de sécurité avec un timestamps. Les nœuds de sécurité comparent les timestamps et prend la clé aléatoire avec un timestamps minimum comme une nouvelle clé du cluster. Ainsi, cette dernière est envoyée vers le cluster Head ; quand ceci reçoit la même clé du cluster d'au moins un nœud de sécurité, il confirme alors cette nouvelle clé, enfin il chiffre la clé du cluster avec la clé par-paire pour informer tout les autres nœuds.
- **La clé publique :** Elle est utilisée par le sink pour chiffrer les informations diffusées, et il est nécessaire qu'elle soit régulièrement mise à jour.

B. **Les protocoles probabilistes**

Dans le schéma de base de pré-distribution de clés purement probabiliste, chaque nœud est pré-chargé, avant le déploiement, avec un sous ensemble de clés prélevées à partir d'un grand ensemble de clés. L'idée dans ce genre de schéma est que deux nœuds voisins ont une certaine probabilité d'avoir partagé une clé commune qui appartient aux deux sous ensembles de ces voisins. Bien que ce schéma à l'avantage d'être simple et complètement distribué, il a deux principaux inconvénients : Plus la probabilité de partager une clé entre des nœuds voisins augmente plus l'espace mémoire, pour contenir un grand sous ensemble de clés, est important. De plus, si le nombre de nœuds compromis

augmente, la sécurité fournie devient insuffisante [14].

B.1. Schéma aléatoire de pré-distribution de clés de L.Eschenauer et D.Gligor

Eschenauer et Gligor dans [23] ont proposés un schéma de base de gestion de clés basé sur la probabilité de partager une clé entre les nœuds d'un graphe aléatoire. Il fournit des techniques pour la pré-distribution de clé, la découverte de clé partagée, l'établissement de clé de chemin , la révocation de clé et la résilience à la capture de nœud.

a) Principe du schéma

L'idée maitresse du ce schéma est la distribution aléatoire d'un certain nombre de clés, avant le déploiement de chaque nœud du réseau, un ensemble fini des clés est associés lors de sa mise en marche. Deux nœuds quelconques peuvent s'échanger des messages sécurisés s'ils possèdent au moins une clé en commun. Le schéma complet est le suivant :

- Une phase de pré-distribution de clés est effectuée avant le déploiement. Un grand ensemble P de clés est généré (entre2^{17} et 2^{20} clés). Pour chaque nœud, m clés sont choisies au hasard à partir de l'ensemble P
  (P = ({(kid1 ; key1) ; (kid2 ; key2) ; . . . . }). Ces m clés sont stockés dans la mémoire du nœud et forment le trousseau de clés du nœud. Le nombre de clés |P| de l'ensemble P est choisi de telle manière que deux sous-ensembles aléatoires de P de taille m auront une certaine probabilité p d'avoir au moins une clé en commun, par exemple pour une probabilité p = 0.5 on a besoin d'un sous ensemble de taille m = 75 clés de l'ensemble P de taille |S| = 10 000 clés.
- Après que les nœuds ont été déployés, une phase de découverte de clés partagées est effectuée. Les nœuds découvrent leurs voisins et plus particulièrement ceux avec qu'ils sont en mesure de communiquer de façon sécurisée

car ils possèdent une clé identique dans leurs trousseau de clés respectif. Le protocole est de diffuser la liste des identités k_{idi} des clés possédées. La clé partagée devient la clé de session de lien entre les deux nœuds, la figure 3.10 [14] illustre cette phase.

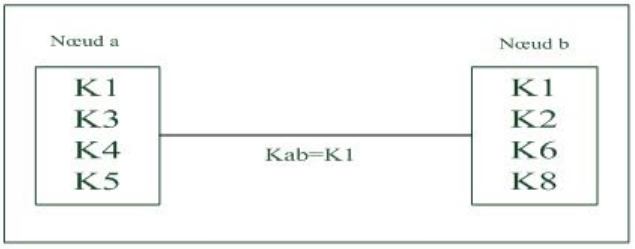


FIGURE 3.10 – la phase de découverte de clés partagées.

– A la fin de cette phase, une phase d'établissement de clé de chemin aura lieu. En effet, le réseau est un graphe connecté formé de liens sécurisés. Les nœuds peuvent alors utiliser les liens existants pour mettre en place des clés partagées avec leurs voisins qui ne partageaient pas de clé en commun avec eux. La figure 3.11 [11] illustre cette phase :

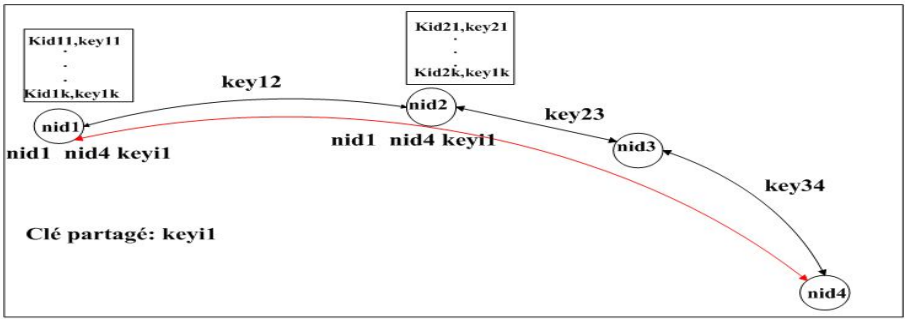


FIGURE 3.11 – Phase d'établissement de clé de chemin.

– La révocation d'un nœud compromis se fait par l'élimination de son trousseau de clés. Pour cela, un nœud contrôleur (qui est mobile et possède une grande connectivité) annonce un message simple de révocation contenant une liste signée de m identificateurs des clés (k_{idi}) pour que ces clés soient retirées des trousseaux de clés des autres nœuds. La liste des identités est signée par une clé de signature k_e générée par le nœud contrôleur et envoyée

en unicast à chaque nœud en la chiffrant avec la clé k_{ci} (Noter que les clés k_{ci} sont partagées entre le $i^{éme}$ contrôleur et chaque nœud capteur pendant la phase de pré-distribution de clés). Quelques liens disparaitrons à cause de la suppression des clés du nœud compromis ce qui nécessite une reconfiguration de ces liens (par la découverte de clés partagées ou l'établissement de clé de chemin). La figure 3.12 [11] illustre cette phase :

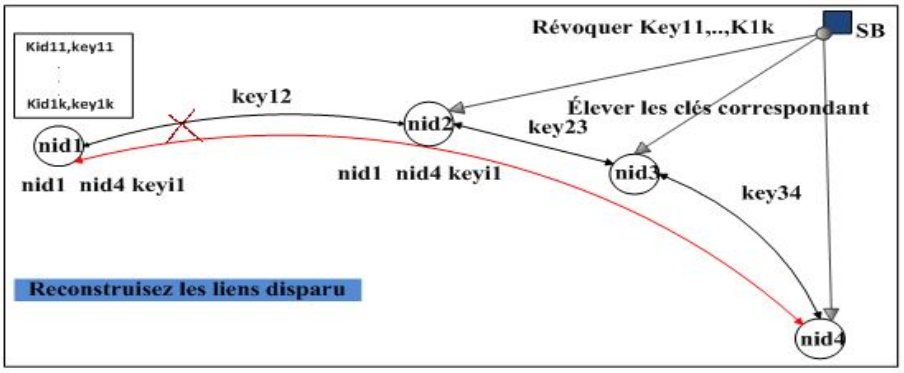


FIGURE 3.12 – La phase de la révocation de clé.

– Il est possible que dans certains cas la vie des clés expirent et leurs renouvellement doit avoir lieu. Le renouvellement de clé est équivalent à une révocation de clé effectuée par le nœud lui même. Après la suppression de clé révoquée, le nœud affecté lance une phase de découverte de clé partagée et probablement une phase d'établissement de clé de chemin pour rétablir le lien cassé.

b) Schéma des clés q-composite

Ce schéma [24] est identique à celui de Eschenauer et Gligor sauf qu'à la place d'une clé partagée exigée pour la communication, une paire de nœud doit partager q clés avec q > 1 pour établir un lien sécurisé. La nouvelle clé utilisée pour la communication entre ces deux nœuds est le hash de toutes les clés partagées entre eux, par exemple pour deux nœuds quelconques qui partage q' clés

(q'>= q), la clé utilisée pour la communication est K = hash (k1 ||k2 || . .. kq'). Les clés sont hachées dans un certain ordre canonique basé sur l'ordre qu'elles se produisent dans l'ensemble de clés P. La figure 3.14 [14] illustre un exemple de partage de clés avec q=2.

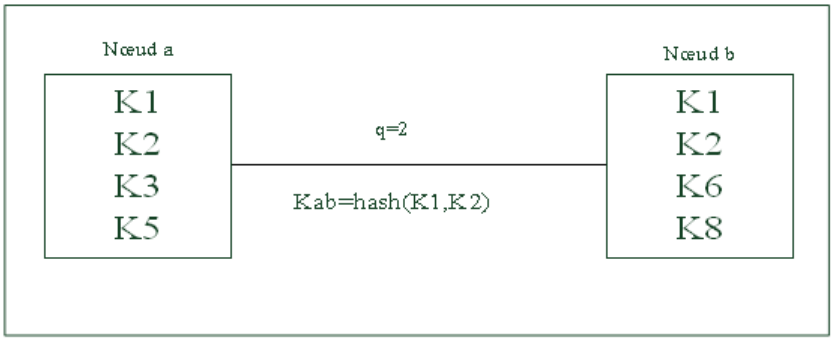


FIGURE 3.13 – Schéma probabiliste de q-composite de gestion de clés.

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

B.2. **key management using deployement knowledge**

Du et autres [33] ont fait une extension de la gestion de clés développée par Eschenauer et Gligor [23]. Ce schéma exploite des connaissances sur le déploiement, cette connaissance est utile pour la pré-distribution de clés. Quand les capteurs voisins sont connus, la pré-distribution principale devient facile et exige simplement pour chaque nœud n de stocker des paires de clés entre n et chacun de ses voisins. Ce qui garanti que chaque nœud est capable d'établir un canal sécurisé avec chacun de ses voisins après déploiement. Ce système comprend trois phases : la phase de pré-distribution de clés, la phase de découverte

de la clé partagée et en fin, la phase de création de chemin de clé d'accès. Les deux dernières phases sont exactement les mêmes que [23], mais en raison de la connaissance de déploiement, la première phase diffère considérablement du régime de base.

Les avantages de considérer des connaissances sur le déploiement réduit au minimum le nombre de clés, et aide à augmenter la résilience ou la résistance à la capture de nœuds et à réduire la complexité en communications. L'inconvénient de ce schéma est la complexité [71].

B.3. **Etablissement De Clés Et Gestion De Contrôle D'Accès**

Ce schéma est une amélioration du protocole Blundo [34]. L'auteur du [38] définit un polynôme semi-symétrique tri-varié de degré t,

p(x,y,z)=f(x,y,z)=$\sum_{i,j=0}^{t}_{k=0} aijx^i y^j z^k$ sur un corps fini F_s. Ce polynôme est construit de telle manière qu'il vérifie la propriété suivante p (x, y, z) = p (y, x, z).

a) Hypothèses

- SB possède des ressources illimitées, dignes de confiance et responsables de la configuration des nœuds avant le déploiement.
- l'attaquant peut être passif ou actif durant le fonctionnement du réseau.
- Une fois qu'un nœud légitime est compromis, toutes les informations stockées sont accessibles par l'attaquant. Les adversaires possèdent des ressources plus importantes (énergie, communication, calculs,...) que les nœuds du réseau.

b) Notations :

La notation suivante est utilisée dans ce schéma :

Notation	Description
Id_i	Identité du nœud n_i.
a_i, $a_{i'}$	Deux valeurs de deux chaines de hachage différentes.
N_i	Un nonce aléatoire généré par n_i.
z_i	L'ordre du nœud n_i.
s_i^j	Un secret aléatoire généré par n_i pour n_j.
X_i	Paramètre d'authentification du nœud n_i.
μij	Une valeur calculée par n_i pour caché s_i^j
K_{ij}	La clé secrète établie entre n_i et n_j.
h	Une fonction de hachage.
MAC_k(m)	Code authentification de message m utilisant la clé k.
a || b	a concaténé avec b.
$\bigoplus$	L'opérateur XOR.

TABLE 3.5 – Les différentes notations utilisées dans [38].

Ce schéma permet de générer des paires de clés et de contrôler l'accès au réseau pour empêcher l'authentification des nœuds malicieux. Il est composé des phases suivantes :

– **Pré-distribution :** Avant le déploiement des nœuds dans le réseau, SB construit un polynôme semi-symétrique tri-varié de degré t. le réseau est supposé qu'il soit composé de r nœuds. SB choisit deux nombres aléatoires w_0 et w_1 (connus seulement par SB) et une fonction de hachage h. Ensuite elle effectue les calculs suivants ; elle construit deux chaines de hachage : La première est une chaine basique et la deuxième est une chaine de type Lamport [39].

A chaque phase de déploiement, chaque nœud n_i est configuré avec :

- la valeur z_i qui indique l'ordre de déploiement et deux paramètres secrets de sécurité $a_i = h^{zi}$ (w_0) et $a_{i'} = h^{l-zi}$ (w_1) des deux chaines de hachages.
- le polynôme secret f_i (y, z) = P $(Id_i$, y, z) évalué en x.
- la fonction de hachage h.

– **Etablissement de la clé secrète et gestion du contrôle d'accès :** La phase d'authentification permet de garantir le droit d'accès au réseau et l'exactitude de l'ordre de déploiement du nœud concerné.

1) La première étape est la découverte de voisinage et l'envoi de demande de

construction de paires de clés. Le nœud n_i génère deux nombres aléatoires : s_i^j et N_i pour garantir le non-rejeu de la clé K_{ij}. Ensuite, n_i diffuse une demande d'établissement de clé à l'aide d'un message Hello.

2) Après cette étape, le nœud n_j compare z_i avec son propre ordre z_j ; n_j calcule le paramètre d'authentification X_j à l'aide du polynôme. n_j calcule ensuite le paramètre de dissimulation μji .Finalement, n_j détermine le paramètre de contrôle d'accès A_j et envoie une réponse au nœud n_i.

3) Lorsque le nœud n_i reçoit la réponse du nœud n_j, il calcule le paramètre de contrôle d'accès A_i. le nœud n_i peut extraire le secret s_j^i à partir de γji avec une soustraction modulaire, et enfin détermine h (s_j^i $||\mu$ij). Il vérifie ensuite si cette valeur est égale à A_j, si c'est faux, n_i rejette simplement la demande et déclare le nœud n_j comme étant un nœud attaquant. Dans l'autre cas, n_j est bien authentifié. La clé partagée ainsi entre n_i et n_j sera K_{ij}.

4) Recevant le message de confirmation, n_j vérifie l'identité du nœud n_i en calculant la valeur h (s_i^j $||\mu$ij) et en la comparant avec A_i. La valeur secrète s_i^j est extraite de γij à l'aide d'une soustraction modulaire de μij. Si l'authentification est réussie, n_j calcule la clé partagée K_{ji}.

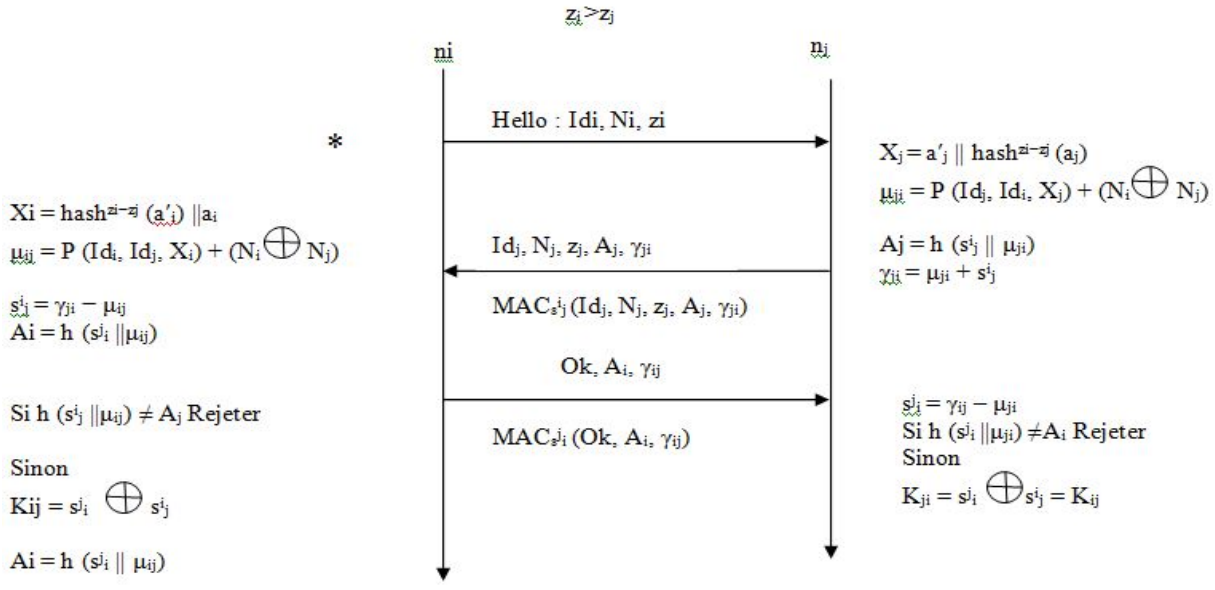


FIGURE 3.14 – Protocole d'établissement de clés.

– **Ajout de nouveaux nœuds par groupe :** Durant le fonctionnement du réseau, un certain nombre de nœuds sont ajoutés par groupe au réseau soit pour maintenir une bonne connectivité ou pour remplacer les nœuds épuisés. Quand un nouveau nœud n_u est déployé, la station de base génère un nouvel ordre de déploiement z_u et génère les deux valeurs des chaines de hachages et le nouveau nœud est configuré avec tous les paramètres existants.

– **Etablissement de clé de groupe :** La clé du groupe permet de garantir que seuls les membres du groupe sont en mesure de déchiffrer les messages échangés. Si n_i décide de créer une clé de groupes avec les nœuds n_j, n_l et n_m (après une découverte du voisinage, Après la réception des réponses). Le nœud n_i construit les clés partagés qui sont individuels K_{ij}, K_{il} et K_{im} respectivement avec les trois nœuds en utilisant le mécanisme précédent, La suite du mécanisme est illustré dans la figure suivante [38] :

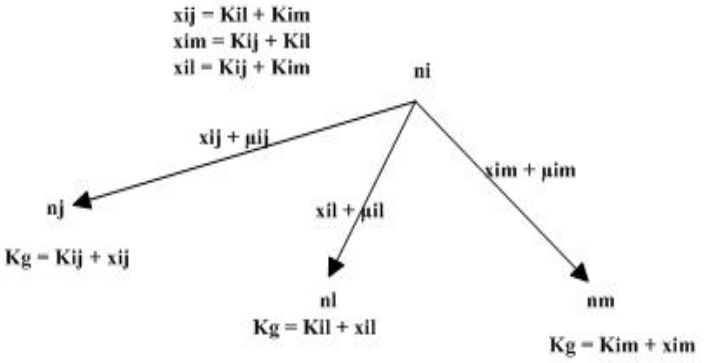


FIGURE 3.15 – Protocole d'établissement de clé de groupe (K_{xy} est la clé partagé entre le nœud n_x et le nœud n_y) .

– **Mise à jour des clés :** Il est très important de pouvoir changer régulièrement de clés afin de limiter le nombre de liens compromis dans le réseau à un instant donné. L'auteur suppose que chaque nœud du réseau peut recevoir une demande de mise à jour " update request " de la part de SB, Cette diffusion peut être sécurisée en utilisant des mécanismes de diffusion classiques. En-

suite chaque nœud qui reçoit cette demande va hacher le polynôme uni-varié $\sum_{k=0}^{t} h^j(aijk)z^k$sachant que la fonction de hachage h est la même pour tous les nœuds. Après cette phase, chaque nœud dans le réseau va obtenir un nouveau polynôme tri-varié semi-symétrique et va reconstruire les clés précédemment construites avec ses voisins permettant ainsi une durée de vie limitée pour chacune des clés du réseau.

B.4. **IBPRF**

Ce schéma [40] appelé la clé de l'identité basée sur la pré-distribution de clé à l'aide de fonction pseudo aléatoire possède les propriétés suivantes :

- Il n'y a aucune communication aérienne pendant la phase d'établissement de clé directe les capteurs.
- Il n'y a aucune communication aérienne pendant l'addition de nouveaux nœuds.
- Lorsque les nœuds sont mobiles, le schéma établi facilement les paires des clés directes entre les nœuds mobiles et leurs voisins physiques.

Cela fonctionne pour n'importe quelle topologie de déploiement, mais ce schéma se base essentiellement sur les deux propriétés suivantes :

– Une pseudo fonction aléatoire (PRF [2]) proposée par Goldreich et autres en 1986 [41].

– Une clé partagée entre chaque nœud de capteur (MK [2]) et le serveur de clé d'installation.

a) Le schéma proposé

Le schéma d'IBPRF [2] se déroule en cinq différentes phases citées ci-dessous :

– **Pré-distribution de clé :** Pour chaque nœud capteur u, le serveur de clé d'installation génère aléatoirement une clé MK_u. Ainsi, pour chaque nœud, le serveur de clé d'installation sélectionne un ensemble S de taille m qui génère

2. voir la liste des abréviations

aléatoirement les id_sdes nœuds de l'ensemble N qui sont considérées comme id du probable voisin physique. Pour chaque id du nœud ∈ S, le serveur de clés d'installation génère une clé symétrique.

– **Etablissement de clé directe :** Après déploiement des nœuds dans un secteur de déploiement, La phase d'établissement de clé directe à les étapes suivantes :

- Chaque nœud localise d'abord tous ses voisins physiques. Après identification des voisins physiques par un nœud u, il peut donc vérifier quels id_s des voisins physiques existent dans son trousseau de clés K_u. Si u constate une clé symétrique par paires avec le nœud v alors il informe v qu'elle a une telle clé. Cela ce fait par l'envoi d'un message court contenant l'identification de u.
- Lors de réception d'un tel message par le nœud v, il peut facilement calculer une clé symétrique, en employant sa propre clé machine et l'id du nœud u. Ainsi, les nœuds u et v peuvent établir une clé par paires directe partagée entre eux très facilement et utiliser cette clé pour leurs futures communications.

– **Etablissement de clé de chemin :** Les nœuds u et v peuvent établir la clé directe entre eux comme suit :

- u trouve d'abord un chemin (u =u0, u1, u2,. . . .uh-1, uh = v).
- u produit un nombre aléatoire k comme clé partagée entre u et v, et le chiffre en utilisant la clé partagée $SK_{u,u1}$ et l'envoie au nœud u1.
- u1 récupère k en le déchiffrant en utilisant la clé partagé $SK_{u,u1}$, et le chiffre en utilisant la clé partagé $SK_{u1,u2}$ et l'envoie à u2.
- Ce processus est continué jusqu'à ce que la clé k portées au nœud v.

– **Mobilité des nœuds :** Dans le nouvel emplacement, u trouve les *ids* de ses nouveaux voisins physiques avec lesquels il ne partage actuellement aucune clé. Si v est un voisin physique, alors u l'informe qu'il à une clé par paires

avec lui en envoyant un message de demande à v contenant l'id du nœud u à l'exclusion de la valeur exacte de la clé. Lors de réception de ce message, v calcule immédiatement la clé par paires partagée entre eux par l'exécution de PRF et à l'aide de la clé principale MK_v et l'id du nœud u. Ainsi, u et v emploient cette clé pour leurs futures communications.

– **Addition des nœuds :** Afin d'ajouter un nouveau nœud u, le serveur de clé d'installation choisit un ensemble S de taille m aléatoirement et produits des ids des nœuds à partir de l'ensemble N. Il produit aléatoirement une clé MK_u pour le nœud u. Pour chaque nœud idv∈S, Le serveur de clé d'installation prend la clé MK_v et calcule la clé symétrique $SK_{u,v}$ comme clé par paires partagée entre les nœuds u et v, et distribue la combinaison de clé ($SK_{u,v}$, v) à u.

3.4 Comparaison

3.4.1 Les métriques d'évaluation

La comparaison entres les différents protocoles de gestion des clés se base selon les métriques suivantes :

– **Efficacité**

Un RCSF est dit efficace par rapport à la quantité de mémoire nécessaire pour enregistrer les clés (*Complexité en mémoire*), le nombre de messages échangés pour la gestion des clés (*Complexité en communication*), la quantité de cycles de processeur nécessaire pour établir une clé (*Complexité en traitement*).

– **Révocation**

Cette propriété sert à savoir si le RCSF a la possibilité du retrait d'un nœud en panne ou quand son opération n'est pas correcte [12].

– **Scalabilité**

On dit qu'un réseau a une bonne scalabilité quand il peut être facilement

augmenté le nombre de nœud et qu'il n'y a aucun problème lors de l'ajout d'un nouveau nœud au réseau car il est intéressant que le réseau assure le bon fonctionnement quel que soit le nombre de nœuds.

- **Résilience contre la capture de nœud**

  Ou résistance contre la capture de nœud, cette métrique mesure comment le RCSF est compromis quand un nœud est compromis, et l'influence de ce nœud sur la sécurité du réseau [12].

- **Connectivité (local/global)**

  Connectivité locale est la probabilité que deux nœuds en communication (nœuds voisins) partagent au moins une clé, tandis que la connectivité globale se réfère au ratio le nombre de nœuds pour pouvoir communiquer dans le graphique de la post-keys-réglage à la taille de l'ensemble réseau [16].

3.4.2 Tableau comparatif

Dans cette partie, nous allons faire une comparaison entre les différents protocoles de gestion de clés dans les RCSFs étudiés dans ce chapitre, on se basant sur les métriques d'évaluations cité ci-dessus, afin d'extraire les différences majeurs existantes et de déterminer le protocole qui repont plus aux besoins des RCSFs.

Schémas	Complexité en Mémoire	Complexité en Communication	Connectivité	Résilience contre la capture de nœud	Scalabilité	Révocation
Key Infection[15]	Dépond du nombre de voisins à un saut(d)	Pour chaque nœud : 2 x d	100 %	Faible	Bien	Faible
LEAP[21]	(3xd)+2+la chaine de clés pour μTESLA	(2 x d)+1	100 %	Très bien avant T_{min}	Bien	Bien
OTMK[30]	d + 1	d + 1	100%	Parfaite avant et après T_{min}	Bien	Bien
Q-Composite [24]	2*m	d+1	p'< p	Dépond de m	Moyen	Trés Bien
Blom [37]	$2(\lambda + 1)$	d + 1	100 %	λ- secure	Moyen	Bien

Schéma de base de Blundo[34]	t+1	d + 1	100 %	t- secure	Très Bien	Bien
Etablissement De Clés Et Gestion De Contrôle D'Accès[38]	t+1	d + 1	100 %	t-secure	Très Bien	Bien
Key management using deployment knowledge [33]	d - 1	d + 1	p'	Dépend de d et de p'	Bien	Bien
Un plan hybride de gestion des clés basé sur les réseaux sans fil de Cluster[51]	un peu complexe	relativement petites	100%	2nc/N	Bien	Trés Bien
IBPRF [40]	m+1	<d+1	Dépend de la taille du réseau	Parfaite	Bien	Bien
SNKM[32]	O(N)	< O(N)	100%	En raison des nœuds de sécurité,la résistance monte effectivement.	Très Bien	Trés Bien

A Novel Secure Keying Technique for the Wireless Sensor Networks [31]	44	Nœud simple stocke au moins 3clés. RCH/DCH stocke les 5 types de clé.	Dépend de CHs	Parfaite	Bien	Bien
STKM[29]	3+nombre de fils	d+1	100%	Bien	Bien	Bien
LEACH[35]	Basée sur CHs	Basée sur CHs	Dépend de CHs	Bien	Bien	Bien

TABLE 3.6: Tableau comparatif entre les protocoles de gestion des clés.

A partir de ce tableau, on constate que les protocoles de gestion de clés déterministes qui sont basés sur une clé initiale et qui achèvent une connectivité totale (100%), sont moins couteux en espace mémoire utilisée. Cependant, les protocoles probabilistes qui pré-charge les nœuds avec un trousseau de clé, sont plus couteux en terme d'espace mémoire. La connectivité dans ces derniers est variante et dépend de la taille du trousseau de clés pré-chargé dans les nœuds capteurs. Les protocoles déterministes, basés sur une topologie plate, présentent une résilience parfaite si la compromission des nœuds aura lieu après la phase d'établissement de clés (après T_{min}). Les protocoles probabilistes, à l'exception du protocole de base d'Echenauere et Gligor, essayent d'atteindre une meilleure résilience par la combinaison d'autres mécanismes au protocole de base. Par contre, les protocoles de gestion de clé sans pré-distribution sont faibles en termes de résilience contre la capture des nœuds.

Conclusion

Dans ce chapitre, nous avons étudié quelques protocoles et solutions de gestion des clés proposées pour les $RCSF_s$. Nous avons vu que les protocoles basés sur la méthode de pré-distribution sont les plus appropriés aux RCSFs, pour leurs faible coût. On conclut que la gestion des clés est l'un des secteurs les plus importants dans la sécurité des $RCSF_s$ et pour cela 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 le chapitre suivant on donnera une description détaillée de notre protocole de gestion de clés dans les $RCSF_s$.

4

Proposition et Simulation

Introduction

La communication dans les RCSFs est basée sur différents paradigmes de communication ; un−à−un, plusieurs−à−un et un−à−plusieurs. La communication un-à-un est utilisée dans des réseaux de type évènement, un nœud capteur détecte une activité qui doit être signalée à une entité lointaine. Dans le paradigme plusieurs-à-un, les nœuds capteurs collectent des données à partir de leurs environnement et les transmettent vers un centre de traitement appelé PUITS. L'acheminement de ces données utilise des routes construites à l'aide de protocoles de routage qui se basent généralement sur un paradigme de communication un−à−plusieurs.

Le problème consiste à trouver la manière d'établir une communication sécurisée dans des différentes applications des $RCSF_s$ et comment établir des clés cryptogra-

phiques entre les différents nœuds capteurs.

Afin de sécuriser les communications entre les nœuds du réseau, il est utile d'intégrer des algorithmes cryptographiques pour la gestion des clés. Dans les $RCSF_s$, la gestion des clés est cruciale pour sécuriser les communications des nœuds. D'autre part la stéganographie [72][73] permet de cacher et d'altérer l'information de base afin d'empêcher un attaquant de l'intercepter. Selon David MARTIN [49], les champs contrôle de trame, numéro de séquence et Adressage de la trame de données de la couche MAC [2] nous offrent des possibilités de cacher des informations.

Dans ce chapitre, nous proposons un schéma de gestion de clés basé sur une combinaison entre la cryptographie et la technique de stéganographie, utilisant une méthode de pré-distribution de clés. L'étude réalisée dans le chapitre précédant montrait que la pré-distribution de clés est la méthode la plus adéquate dans les RCSFs.

4.1 Le protocole proposé

Notre schéma Mobil and Hybrid Key Management Protocole using Stéganographie (MHKMPS) est caractérisé par le faite d'utiliser la technique de stéganographie pour cacher la clé, ce qui permet d'assurer la sécurité des informations circulant sur le réseau ; ainsi, diminuer le nombre de messages échangés entre les nœuds capteurs. Cependant, notre approche utilise quatre types de clés : une clé initiale partagée entre chaque nœud et SB et elle est identique pour tous les nœuds du réseau, une clé de voisins (appelé clé par−paire) partagée entre un nœud et ces voisins et une clé privée unique pour chaque nœud, ainsi qu'une clé publique identique pour tous les nœuds utilisé pour la mise à jour de la clé privée. La clé initiale est mise à jour régulièrement et d'une façon très fréquente par SB et sans engendrer des messages supplémentaires pour assurer plus de sécurité. Ainsi, dans le cas où un adversaire a

2. voir la liste des abréviations

pu intercepter la clé, il pourra l'utiliser que pour une période fixe du temps, puisque cette clé est mise à jour fréquemment.

Aussi, notre schéma est basé sur des nœuds mobiles c'est−à−dire que chaque nœud doit exécuter la phase de découverte de voisins à chaque fois qu'il se déplace, ce qui nécessite la mise à jour des clés par−paire.

La clé privée est attribué à chaque nœud avant son déploiement, elle est utilisée pour chiffrer le nonce qui permet de calculer la clé par-paire. Si un attaquant a pu écouter le trafique entre les différents nœuds, il ne pourra déduire que la clé initiale K_{IN} puisque c'est la même pour tous les nœuds du réseau, mais il ne pourra pas obtenir la clé privée des nœuds que dans le cas où l'attaquant a accès physique au nœud capturé.

a. **Hypothèses**

Ce schéma est fondé sur les hypothèses suivantes :

- Les nœuds sont mobiles, cette propriété permet à tous les nœuds du réseau de se déplacer à tout moment et de changer leurs clés à chaque déplacement. Ce paramètre n'est pas pris en charge par la plus part des protocoles proposés pour la gestion des clés.
- Les nœuds capteurs sont homogènes (c'est-à-dire les nœuds sont similaires dans leurs capacité de traitement, d'énergie et de stockage.
- Le déploiement est aléatoire (les voisins de n'importe quel nœuds ne sont pas connu avant le déploiement).
- La compromission d'un nœud implique que toutes les informations stockées dans sa mémoire sont connues par l'attaquant.
- La SB n'a pas de contrainte sur la capacité de calcul, de stockage et ne peut être compromise.
- Avant que la phase de pré-distribution de clé soit terminée, un attaquant ne pourra pas obtenir des informations sur les nœuds.

– Après la phase de déploiement des nœuds, les informations sur les nœuds capturés peuvent être obtenues.

b. **Notation**

La notation suivante est utilisée dans ce schéma :

Notation	**Description**
n	Nombre de nœuds .
K_{IN}	Clé initiale.
K_{AB}	Clé par-paire entre les nœuds A et B.
Id	Identifiant d'un nœud.
f	Fonction pseudo aléatoire.
MAC (K_{IN}, A|B)	Produire du MAC pour le message A|B avec la clé K_{IN}.
$A \longrightarrow B$	Le nœud A envoi un message au nœud B.
A||B	Le message A concaténé avec le message B.
Nonce	Nombre aléatoire.
T_{min}	Le temps qu'il faut pour les nœuds avant d'établir des clés par-paires entre eux.
K_{pr}	Clé privée.
K_{SB}	La clé privée de la station de base.
K_p	Clé publique.

TABLE 4.1 – les différentes notations utilisées dans MHKMPS.

c. **Idée de base**

MHKMPS [2] se compose de six phases importantes qui permettent une bonne gestion des clés échangées et aussi une meilleure sécurité des données et des clés. L'utilisation de la technique de stéganographie permet de cacher des clés dans les différents champs de la trame de donnée, ainsi de tromper l'attaquant de l'existence d'une clé dans la trame envoyée. Les différentes phases de notre protocole sont bien détaillées ci-dessous

1) **La phase de pré-distribution**

– SB attribue à chaque nœud du réseau un Id unique, une clé initiale utilisée pour chiffrer le message HELLO envoyés par les nœuds pour la découverte de leurs voisins, et une clé privée ainsi qu'une clé publique.

2. voir la liste des abréviations

– SB possède une clé privée K_{SB} utilisée pour déchiffrer les messages envoyés par les autres nœuds du réseau pour informer SB du changement de leurs clés privées.

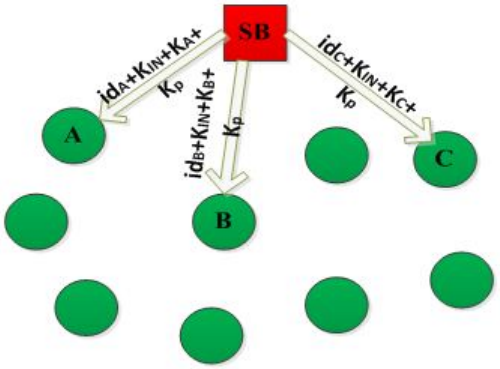


FIGURE 4.1 – la phase de pré-distribution dans MHKMPS.

2) **Phase de découverte de voisins**

Immédiatement après son déploiement, le nœud A essaye de découvrir ses voisins en diffusant un message HELLO qui contient son Id, la clé privée de A (K_A) et le nonce généré aléatoirement par le nœud A caché avec K_A dans la trame de donnée et chiffré avec K_A, cette clé est supprimée immédiatement juste après l'établissement de la clé par−paire, ces deux derniers ($(nonce)_{KA}$ et K_A) sont chiffrés avec la clé initiale K_{IN} : $A \longrightarrow * : id_A +$ (K_A, (nonce)K_A) K_{IN}, et il se met en attente des réponses de la part de ses voisins. Supposant que le nœud B entende le message de A (B étant le voisin de A à un saut) alors il lui repond par une autre trame de donnée (on le considère comme un ACK) contenant son Id, sa clé privée (K_B) et le nonce envoyé par A caché aussi dans la trame de donnée et chiffré avec K_B, ces deux derniers sont chiffrés avec la clé K_A. En plus de ça, le nœud B envoie le MAC des deux identifiant (id_A, id_B) chiffré avec K_{IN} pour assurer l'authentification : $B \longrightarrow A : id_B$, MAC ($K_{IN}$, $id_A||id_B$) + (K_B, (nonce)K_B) K_A. Lorsque le nœud A reçoit l'ACK envoyé par B, il récupère l'Id de B et exécute la fonction d'authentification MAC sur les identités des nœuds A et B pour authentifier le nœud B. De cette manière le

nonce est partagé entre les deux nœuds d'une manière plus sûre et sécurisée et donc chacun de sa part va calculer la clé pars-paire à partager entre eux sans que les attaquants l'intercepte. La clé par-paire est calculée en utilisant la fonction pseudo aléatoire f et le nombre aléatoire nonce. Ainsi, chacun des nœuds A et B calcule sa clé par-paire comme suit : K_{AB}= fK_A (id_B||nonce).

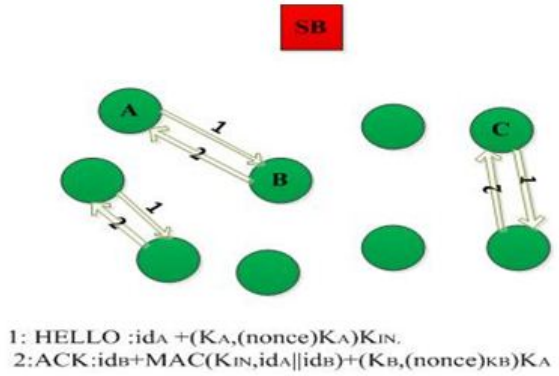


FIGURE 4.2 – la phase de découverte des voisins dans MHKMPS .

Algorithme de génération de la clé par-paire entre les nœuds voisins :

Paramètre d'entrée : n nœud, K_{IN} clé initiale partagée entre chaque nœud et SB, f fonction pseudo aléatoire.
Paramètre de sortie : la clé par-paire entre les voisins K_{AB}.
Début
Pour (chaque nœud A) faire
Diffuser un message HELLO: id_A+ (K_A, (nonce) $_{KA}$) K_{IN};
Fixer un minuteur et attendre ;
Si (ACK reçu par le voisin B avant l'expiration du minuteur) alors
Obtenir l'id du nœud B ;
Obtenir K_B et le nonce chiffré avec K_B ,
Générer la clé par-paire K_{AB} en utilisant la fonction pseudo aléatoire f :
K_{AB}= f_{KA} (id_B||nonce),
Finsi ;
Finpour ;
Fin.

Après cette phase, chaque nœud aura établi une clé par-paire avec chacun de ses voisins, il est à noter que chaque nœud supprime toutes les clés privées de ses voisins pour éviter qu'elles soient interceptées par un attaquant et pour ne pas surcharger leur espace mémoire. En outre, tous les nœuds du réseau seront obligés de changer leurs clés privées juste après la phase de découverte des voisins pour qu'elles restent toujours secrètes. Pour cela, les nœuds informent SB de ce changement en envoyant un message contenant l'Id du nœud et sa nouvelle clé privée, le tout chiffré avec la clé public K_p :

$A \longrightarrow SB$: $(id_A,\ K_{A'})\ K_p$; ainsi, pour le déchiffrer SB utilise sa clé privée K_{SB}.

Puisque les nœuds sont mobiles, la découverte des voisins est indispensable à chaque fois qu'un nœud se déplace, ce qui nécessite une mise à jour régulière de la clé par-paire. Ainsi, chaque nœud doit exécuter à nouveau la phase de découverte de voisins comme expliqué ci-dessus.

3) **La phase de communication**

Cette phase permet l'établissement des communications soit entre les nœuds voisins eux-mêmes, soit entre les nœuds et SB. Ainsi, il existe deux types de communications :

– Les nœuds voisins peuvent communiquer en utilisant la clé par-paire partagée entre eux, et donc chiffrer les informations envoyées avec cette clé.
– SB peut communiquer avec les autres nœuds du réseau en utilisant les clés privées de ces derniers.

4) **La phase de mise à jour**

Il est très important de pouvoir changer régulièrement de clés afin de limiter le nombre de liens compromis dans le réseau à un instant donné. On suppose que chaque nœud du réseau puisse recevoir une demande de mise à jour de la clé K_{IN} " update request " de la part de SB, Cette diffusion peut être sécurisée en utilisant des mécanismes de diffusion classiques. Cependant, SB cache la

nouvelle clé K_{IN}, en utilisant la stéganographie dans le champ de contrôle de la trame de données, cryptée avec la clé privée de chacun des nœuds du réseau. Ainsi, chaque nœud peut récupérer cette nouvelle clé envoyée par SB. Il existe deux cas de mise à jour de la clé initiale K_{IN} :

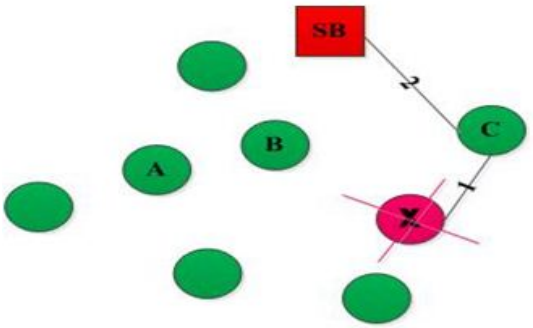
- SB décide de mettre à jour à chaque ronde.
- Quand SB remarque un comportement anormal d'un des nœuds du réseau, ou bien lorsqu'elle reçoit un certain nombre de message l'informant qu'un tel nœud est compromis ; dans ce cas, elle envoie une requête chiffré avec la clé privée vers tous les nœuds pour les informer, cette dernière contient l'Id du nœud compromis.

5) **La suppression d'un nœud**

Parmi les raisons pour lesquelles un nœud ne fonctionne plus on trouve : l'épuisement de sa source d'énergie, l'endommagement par un attaquant, la compromission d'un nœud intermédiaire . . .

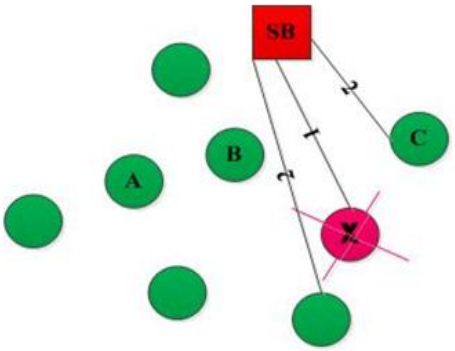
Dans notre approche deux cas de figure se présentent, le premier cas c'est quand un nœud détecte qu'un de ses voisins est malveillant, alors il ignore ses messages, supprime la clé par-paire partagée avec lui et il le supprime définitivement de sa liste des voisins, ensuite il envoie un message vers SB pour l'informer, ce dernier comporte l'Id du nœud compromis.

Le deuxième cas c'est quand SB détecte le comportement anormal d'un des nœuds du réseau, alors elle envoie des messages vers les nœuds avec qui il partage des clés par-paire pour les informer, puis le supprimer définitivement du réseau.



1: ignorer les messages, supprimer les clés partagées, et supprimer le nœud compromis de sa liste des voisins.
2:informer SB.

FIGURE 4.3 – la suppression des nœuds compromis −cas1−.



1: détecter le comportement anormal, et supprimer le nœud compromis.
2:informer les nœuds avec qui il partage des clés par-paire.

FIGURE 4.4 – la suppression des nœuds compromis −cas2−.

6) **L'arrivée d'un nœud**

Durant le fonctionnement du réseau, un certain nombre de nœuds sont ajoutés par groupe au réseau soit pour maintenir une bonne connectivité ou pour remplacer les nœuds épuisés. Quand un nouveau nœud N est déployé, SB génère un nouvel identifiant, une clé initiale, et une clé privée unique, ainsi qu'une clé publique.

Après son déploiement, le nœud N essaye de découvrir ces voisins à un saut, il exécute donc la phase de découverte de voisins pour établir des clés par-paire avec ses voisins, et commence à échanger des messages et des trames de données avec eux.

4.2 Discussion

Le schéma proposé est un protocole déterministe avec une topologie plate car tous les nœuds du réseau ont les mêmes ressources et les mêmes caractéristiques à l'exception de la station de base. La compromission de la clé K_{IN} par un attaquant lui permet de récupérer toutes les informations sur le nœud compromis et de les utiliser pour perturber les communications avec le voisinage. Mais, cette attaque ne

pourra pas durer longtemps car si l'un des voisins de ce nœud compromis le détecte, il supprime alors la clé par-paire partagée avec lui et informe SB, et dans le cas où SB reçoit plus d'un message informant qu'un tel nœud est compromis elle informe les autres nœuds du réseau, puis elle le supprime. Cependant, puisque dans notre cas les nœuds sont mobiles, donc si un attaquant est présent et il a déjà compromis une partie du réseau, alors il pourra aussi se déplacer pour découvrir un nouveau voisinage et compromettre une autre partie du réseau et ainsi de suite jusqu'à qu'il arrive à compromettre le réseau tout entier, mais dans notre approche cela n'est pas possible, car à la présence d'un attaquant (ou un nœud compromis) SB informe tout les nœuds du réseau et donc aucun d'eux n'acceptera d'établir une clé par-paire avec lui. Comme conclusion à ce point, on pourra dire que la présence d'un attaquant pourra compromettre qu'une partie du réseau mais pas tout le réseau.

Notre protocole assure la propriété d'authentification entre les nœuds voisins avec l'utilisation de la fonction MAC, la confidentialité des données transmises avec une combinaison entre la stéganographie et la cryptographie. MHKMPS assure aussi la scalabilité puisque il permet de rajouter des nœuds sans aucune contrainte sur le bon fonctionnement du réseau, et aide à augmenter la résilience ou la résistance à la capture de nœuds, la connectivité et la révocation des nœuds, et aussi à réduire la complexité en communications avec l'utilisation de la stéganographie qui permet d'envoyer la clé cachée dans un message et donc au lieu d'envoyer deux messages : le premier c'est un message d'information et le deuxième pour transmettre la clé, on envois un seul message qui fait les deux choses ; ce qui permet donc de réduire le nombre de message échangés entre les nœuds capteurs. Cependant, la mobilité des nœuds est un point très important dans un schéma de gestion de clés dans les $RCSF_s$. L'inconvénient de ce protocole est la complexité en mémoire, puisque chaque nœud doit stocker dans sa mémoire les trois clés : K_{IN}, K_p, K_{pr}, plus les clés par-paire qui correspond aux nombre de voisins de chaque nœud.

4.3 Tableau des résultats des métriques de comparaisons de MHKMPS

Selon les Métriques qu'on à définis dans le chapitre précédent on présente le tableau suivant qui illustre la valeur des différentes métriques d'évaluation de notre approche :

Métrique	**Valeur**
Complexité en mémoire	3+ nombre de voisins.
Complexité en communication	d+1.
Connectivité	100%.
Résilience contre la capture de nœud	Parfaite avant T_{min},et bien après T_{min}.
Scalabilité	Bien.
Révocation	Très Bien.

TABLE 4.2 – Le tableau des résultats du protocole MHKMPS.

4.4 Exemple illustratif

Pour éxpliquer et illustrer le fonctionnement de MHKMPS en considère l'éxemple suivant : Soient les deux nœuds A et B et une SB.

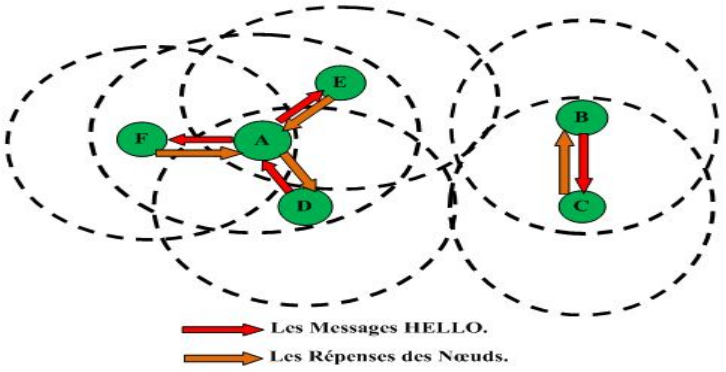


FIGURE 4.5 – Exemple illustratif.

Dans la phase de pré-distribution de clés, les nœuds A et B possèdent un identifiant et une clé privé unique, ainsi qu'une clé initiale connu par les deux plus la clé publique (K_p) ; et SB qui possède sa propre clé privée (K_{SB}).

Après leurs déploiements, chacun d'eux va essayer de découvrir ces voisins comme suit :

Le nœud A : $A \longrightarrow * : id_A + (K_A, (\text{nonce})K_A)K_{IN}$

Le nœud B : $B \longrightarrow * : id_B + (K_B, (\text{nonce})K_B)K_{IN}$

Supposant que les nœuds F, E et D sont les voisins de A à un saut, et le nœud C le voisin de B à un saut. Le déroulement est le suivant :

$D \longrightarrow A : id_D + \text{MAC}\ (K_{IN}, id_A||id_D) + (K_D, (\text{nonce})K_D)K_A.$

$C \longrightarrow B : id_C + \text{MAC}\ (K_{IN}, id_B||id_C) + (K_C, (\text{nonce})K_C)K_B.$

$E \longrightarrow A : id_E + \text{MAC}\ (K_{IN}, id_B||id_E) + (K_E, (\text{nonce})K_E)K_B.$

$F \longrightarrow A : id_F + \text{MAC}\ (K_{IN}, id_B||id_F) + (K_F, (\text{nonce})K_F)K_B.$

Après cette phase, chacun des nœuds A, B, C, D, E, F établir une clé par-paire avec chacun de ces voisins comme suit :

$K_{AD}= f_{KA}\ (id_D||\text{nonce}).$

$K_{BC}= f_{KB}\ (id_C||\text{nonce}).$

$K_{AE}= f_{KA}\ (id_E||\text{nonce}).$

$K_{AF}= f_{KA}\ (id_F||\text{nonce}).$

Les clés par-paires sont établies entres les nœuds, donc c'est le moment du changement des clés privées, ainsi les nœuds A, B, C, D, E, F envoient les messages suivant vers SB pour l'informer :

$A \longrightarrow SB : (id_A, K_{A'})K_p$

$B \longrightarrow SB : (id_B, K_{B'})K_p$

$C \longrightarrow SB : (id_C, K_{C'})K_p$

$D \longrightarrow SB : (id_D, K_{D'})K_p$

$E \longrightarrow SB : (id_E, K_{E'})K_p$

$F \longrightarrow SB : (id_F, K_{F'})K_p$

SB déchiffre ensuite les messages reçus en utilisant sa clé privée.

L'arrivée d'un nœud au réseau

Lorsque le nœud U arrive au réseau, SB lui attribut un identifiant Id_U, une clé initiale K_{IN}, une clé privé K_U et une clé publique K_p. Le nœud U à son rôle, exécute la phase de découverte de voisin comme suit : $U \longrightarrow * : id_U + (K_U, (\text{nonce})K_U)K_{IN}$. On suppose que les nœuds A et D sont les voisins de U à un saut, Le déroulement est le suivant : $A \longrightarrow U : id_A + \text{MAC}\ (K_{IN}, id_U||id_A) + (K_A, (\text{nonce})K_A)K_U$.

$D \longrightarrow U : id_A + \text{MAC}\ (K_{IN}, id_U||id_D) + (K_D, (\text{nonce})K_D)K_U$.

Après cette phase, chacun des nœuds U, A, D établii une clé par-paire avec chacun de ces voisins comme suit : $K_{UC}= f_{KU}\ (id_C||\text{nonce})$.

$K_{UA}= f_{KU}\ (id_A||\text{nonce})$.

Après l'établissement de la clé par-paire entre le nœud U et ces voisins, la mise à jour de sa clé privé est très importante. Il envoi alors le message suivant vers la SB pour l'informer : $U \longrightarrow SB : (id_U, K_{U'})K_p$.

Le déplacement d'un nœud

On suppose que le nœud B se déplace dans le réseau comme illustré dans la figure 4.6 alors il exécute une autre fois la phase de découverte de voisin comme expliqué ci-dessus où les nœuds A, D, E sont ses voisin à un saut.

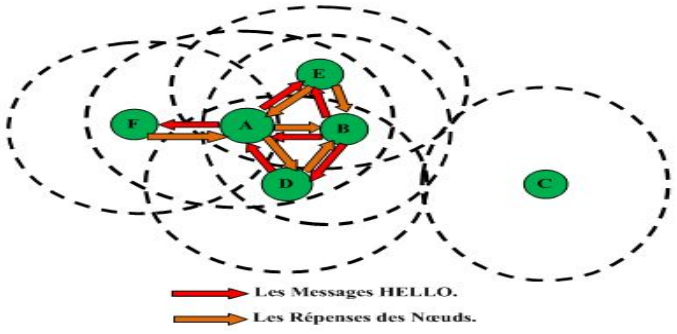


FIGURE 4.6 – le déplacement du nœud B dans un réseau.

4.5 Simulation

La simulation informatique, ou simulation numérique, est une série de calculs effectués sur un ordinateur et reproduisant un phénomène physique. Elle aboutit à la description du résultat de ce phénomène, comme s'il s'était réellement déroulé. Cette représentation peut être une série de données, une image ou même un film vidéo [75].

Un simulateur peut réagir à des modifications de paramètres et modifier ses résultats en conséquence. Un simulateur de vol, par exemple, modifie la trajectoire calculée de l'avion en fonction des commandes transmises par l'utilisateur [74].

A fin de simuler notre protocole, nous avons utilisé MATLAB qui est un langage de haut niveau et un environnement interactif pour le calcul numérique, la visualisation et la programmation. En utilisant MATLAB, on peut analyser les données, développer des algorithmes, et créer des modèles et des applications. La langue, les outils et les fonctions intégrées de mathématiques permettent d'explorer des approches multiples et parvenir à une solution plus rapide qu'avec des tableurs ou des langages de programmation traditionnels, tels que C/C++ ou Java. MATLAB est utilisé pour une gamme d'applications, y compris le traitement du signal et de la communication, l'image et le traitement vidéo, les systèmes de contrôle, de test et de mesure, finance computationnelle, et la biologie computationnelle [75].

4.5.1 Environnement de simulation

Notre modèle d'expérimentation est établi sur 100 nœuds, dispersés aléatoirement sur une surface carrés de $100m^2$. Nous supposons que les nœuds peuvent se déplacer après chaque 10 secondes en essayant de trouver d'autres voisins et d'établir d'autres clés. Les paramètres de notre simulation sont résumés dans le tableau suivant :

1. voir l'Annexe

Paramètres	**Valeur**
Localisation de SB	(x=50, y=50)
Le nombre de nœuds	100
Energie initiale du resaux	5 Joul
La taille des paquets	512 ø
La valeur d'énergie électronique	0.0000005 Joul
la valeur d'énergie d'amplification	0.000000013 Joul
La valeur du rayon de couverture [1]	15 Mètres
La valeur du rayon de communication [1]	30 Mètres
Minuteur (timer)	10 Secondes

TABLE 4.3 – Paramètres de simulation.

4.5.2 Résultats de simulation

Le temps de simulation est divisé en plusieurs intervalles de 10 secondes. Durant chaque intervalle de temps, nous avons mesuré le nombre de messages échangés entre les nœuds, ainsi que l'énergie consommés. Au début de la simulation, l'énergie est initié à 5 Joul, à chaque intervalle de temps le réseau consomme de l'énergie, ce qui implique que l'énergie initiale diminue à chaque fois, ainsi le critère d'arrêt de notre simulation est quand l'énergie descend sous le seuil de 1 Joul.

A la fin de la simulation, on aura deux graphes : le premier illustre l'énergie moyenne consommée par tous le réseau durant tous les intervalles de temps (l'énergie moyenne consommée par tous le réseau) ; le deuxième illustre l'énergie résiduelle du réseau.

La figure suivante (figure 4.7) illustre un réseau de 100 nœuds capteurs déployés aléatoirement dans une surface de 100 * 100 $mètres^2$. Chaque nœud du réseau a une portée de signal de 15 mètres (R_c=15).

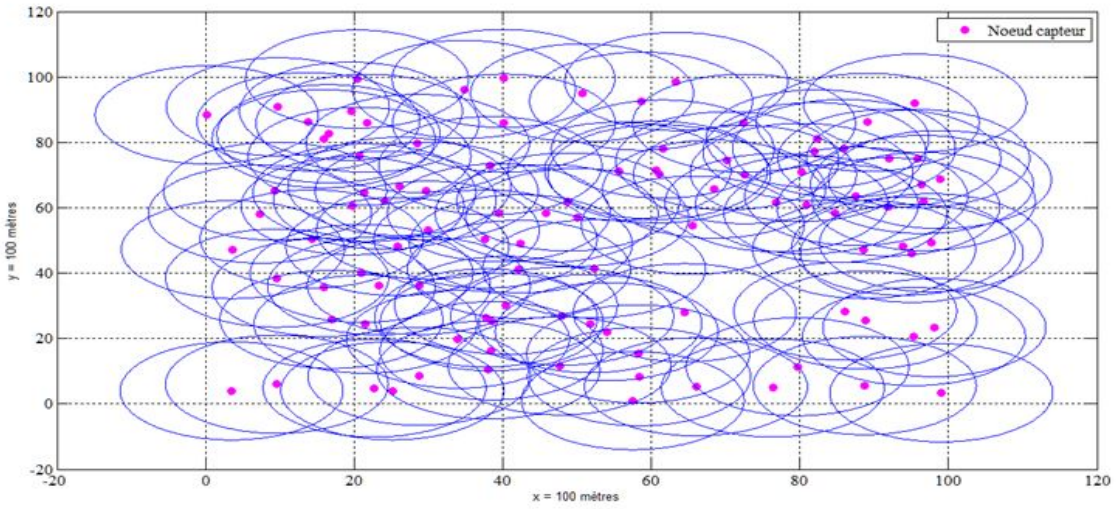


FIGURE 4.7 – le déploiement aléatoire de 100 nœuds.

4.5.2.1 Analyse et évaluation des performances de notre approche

Afin d'évaluer les performances de notre protocole MHKMPS, nous avons utilisé les métriques suivantes :

- Nous avons utilisé la consommation d'énergie car c'est le facteur le plus important.
- Nous avons utilisé le nombre de messages échangés entre les nœuds capteurs comme métrique pour montrer l'apport essentielle de notre protocole qui se base sur l'utilisation de la stéganographie, qui permet ainsi de minimiser le nombre de message échangés qui implique moins d'énergie consommée. Nous avons mesuré cette métrique en faisant varier le nombre de nœuds.
- Nous avons utilisé la scalabilité qui permet de mesurer la capacité de notre apprche.

a) **La consommation d'énergie**

La ressource énergétique détermine la durée de vie du réseau et doit être soigneusement prise en compte dans la conception de n'importe quelle application de RCSF. Pour cette raison, on a mesuré et comparé la consommation d'énergie

de notre approche (MHKMPS) avec les protocoles LEACH [35] et le protocole hybride de gestion des clés basé sur les réseaux sans fils de Cluster [51].

Heinzelman et al.[60] proposent un modèle radio de consommation d'énergie. Ainsi, les énergies nécessaires pour émettre ET_x(S, d) et recevoir ER_x(S) des messages sont données par :

- Pour émettre un message de S bits vers un récepteur loin de d mètres, l'émetteur consomme : $ET_x(S,\ d) = (Eelec\ *\ S) + (Eamp\ *\ S\ *d2)$
- Pour recevoir un message de S bits, le récepteur consomme :

$$ER_x(S) = Eelec\ *\ S$$

Eelec et Eamp représentent respectivement l'énergie de transmission électronique et d'amplification.

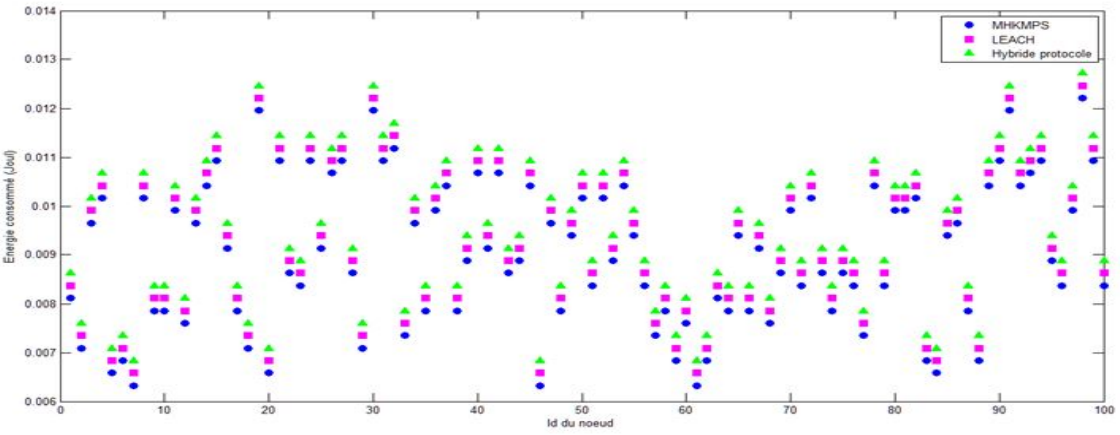


FIGURE 4.8 – la consommation d'énergie de chaque nœud du réseau à un intervalle de temps de notre approche comparé avec deux autres protocoles.

La figure 4.8 montre l'énergie consommée par chaque nœud du réseau des trois protocoles : MHKMPS, LEACH [35] et le protocole hybride [51]. La simulation illustre que la consommation d'énergie du MHKMPS est plus petite par rapport à celle du[35] et[51]. Ceci est dû à l'utilisation de la stéganographie qui réduit au maximum le nombre de message échangés.

La figure 4.9 illustre l'énergie moyenne consommée par tous le réseau durant tous les intervalles du temps en fonction du nombre d'itération effectué (c.à.d. le nombre

d'intervalle de temps avant l'expiration d'énergie initiale) pour les trois protocoles : MHKMPS, LEACH et le protocole hybride. Ainsi, la figure 4.10 illustre l'énergie résiduelle des approches citées ci-dessus en fonction du nombre d'itération aussi.

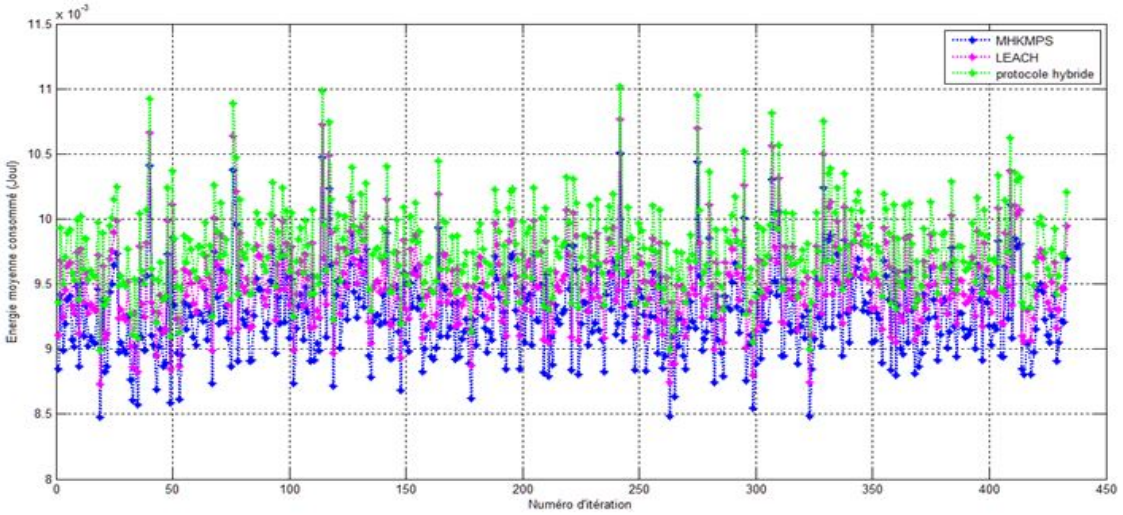


FIGURE 4.9 – la consommation énergétique moyenne pour les trois protocoles.

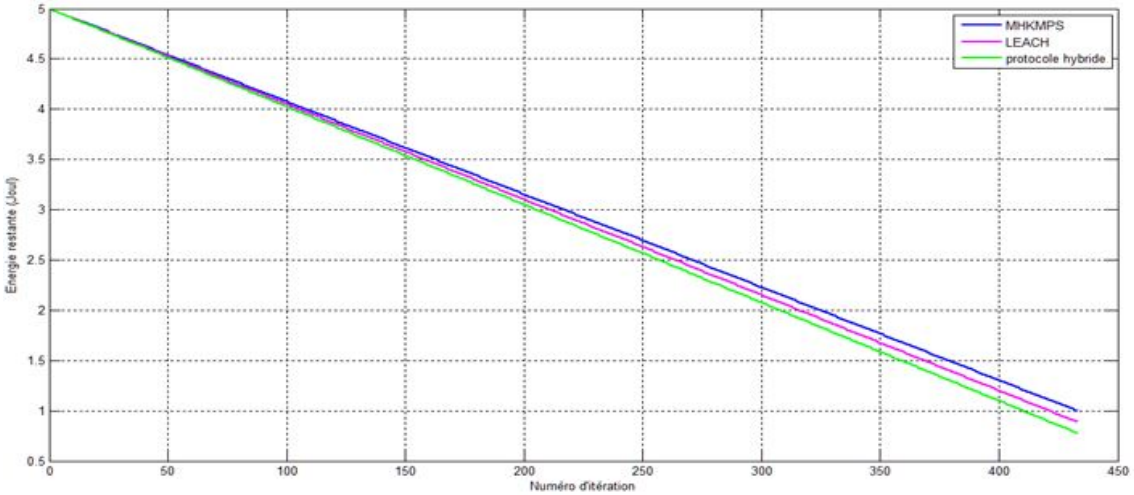


FIGURE 4.10 – Energie résiduelle des trois approches.

b) **La complexité en communication**

L'histogramme suivant montre la complexité en communication (nombre de messages) par chaque nœud capteur en fonction de nombre de voisins.

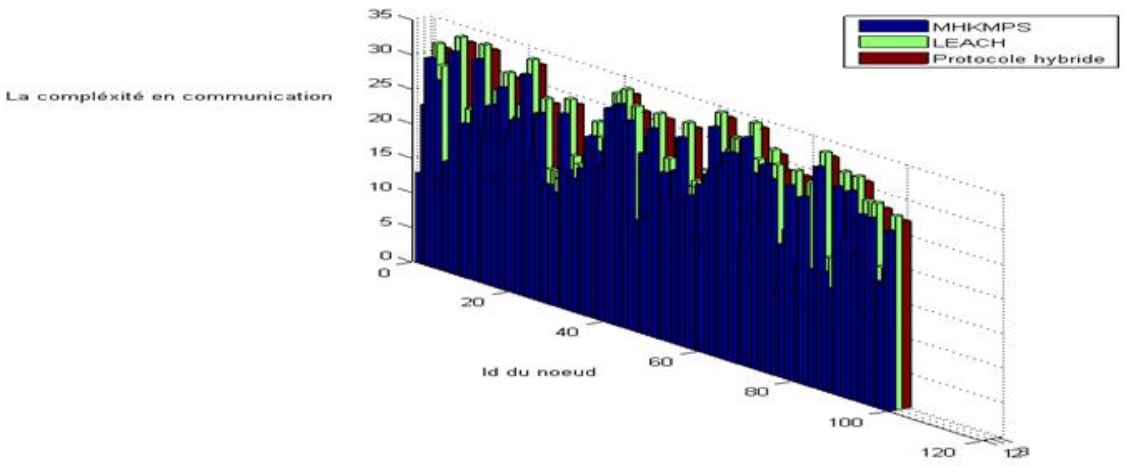


FIGURE 4.11 – La complexité en communication des trois approches.

c) **Le passage à l'échelle**

La figure 4.12 illustre la capacité de notre approche, le réseau fonctionne même si le nombre de nœuds dépasse les 4000.

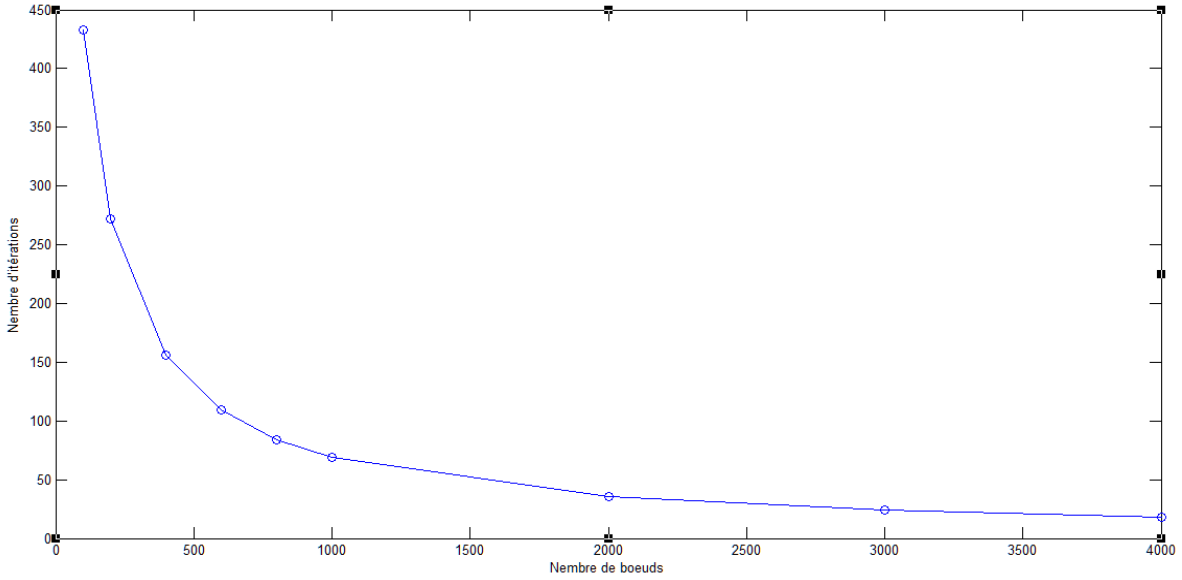


FIGURE 4.12 – La scalabilité de MHKMPS.

Conclusion

La sécurité permet d'utiliser les $RCSF_s$ avec une certaine confiance. Sans sécurité, l'utilisation des $RCSF_s$ 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.

Dans ce chapitre nous avons évalué les performances de notre protocole MHKMPS, puis on l'à comparé avec les deux protocoles de gestion de clé, basé sur la prédistribution présentés dans le chapitre 3, LEACH et le plan hybride. Les résultats ont montré l'intérêt d'utilisé la stéganographie dans les messages échangés. A partir de l'étude faite, nous avons constaté que l'ensemble des protocoles de gestion de clés dans les $RCSF_s$ essayent d'atteindre une connectivité et une résilience meilleures tout en minimisant la consommation de la mémoire et d'énergie. MHKMPS présente une meilleure résilience avant T_{min}, il réduit aussi au maximum le nombre de messages échangés avec l'utilisation de la stéganographie, et comme les protocoles déterministes il atteint une connectivité totale.

Conclusion et Perspectives

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

L'ensemble des protocoles de gestion de clés proposés pour les $RCSF_s$ se basent principalement sur la cryptographie à clé symétrique et la méthode de pré-distribution de clés, afin d'achever l'établissement de clés entre les entités communicantes dans le réseau. Nous avons étudié un ensemble de ces protocoles de gestion de clés qui sont classés dans plusieurs catégories selon la topologie du réseau (hiérarchique ou plate) et la façon dont les nœuds voisins partagent des clés communes (probabiliste ou déterministe). Après l'étude de ces solutions, nous avons constaté que le défi dans la conception des schémas de gestion de clés est de trouver un compromis entre un système efficace et les contraintes caractérisant les $RCSF_s$.

Nommé MHKMPS (Mobil and Hybrid Key Management Protocol using Stéganographie), notre solution montre à travers les résultats de l'évaluation et de la simulation qu'elle peut fournir plus de sécurité avec moins d'exigence que d'autres solutions. Comme perspective, nous étudions une solution qui permet d'intégrer l'idée proposée dans un algorithme de routage pour la recherche du plus court chemin et de minimiser la complexité en mémoire.

A

Annexe

– **MANETs :** Un MANET est un type de réseau ad hoc qui peut changer des emplacements et se configurer en marche. Puisque MANETs sont mobiles, ils emploient les connexions sans fil pour se relier à de divers réseaux. Ceci peut être une connexion standard de Wifi, ou un milieu différent, tel qu'une transmission par satellite cellulaire.

– **RSA** [45] **:** Le cryptage RSA, du nom de ses concepteurs, Ron Rivest, Adi Shamir et Leonard Adleman, est le premier algorithme de chiffrement asymétrique. Il a été découvert en 1977 au Massachusetts Institute of Technology. Le RSA est basé sur la théorie des nombres premiers, et sa robustesse tient du fait qu'il n'existe aucun algorithme de décomposition d'un nombre en facteurs premiers. Alors qu'il est facile de multiplier deux nombres premiers, il est très

difficile de retrouver ces deux entiers si l'on en connaît le produit .

– **Réseau plat** [14] **:** Un réseau de capteurs sans fil plat est un réseau homogène, où tous les nœuds sont identiques en termes de batterie et de complexité du matériel, excepté le nœud PUITS qui joue le rôle d'une passerelle et qui est responsable de la transmission de l'information collectée à l'utilisateur final. Selon le service et le type de capteurs, une densité de capteurs élevée (plusieurs nœuds capteurs/m2) ainsi qu'une communication multi-sauts peut être nécessaire pour l'architecture plate. En présence d'un très grand nombre de nœuds capteurs, le passage à l'échelle devient critique. Le routage et le contrôle d'accès au médium (MAC) doivent gérer et organiser les nœuds d'une manière très efficace en termes d'énergie .

– **Réseau hiérarchique** [14] **:** Une architecture hiérarchique a été proposée pour réduire la complexité de la plupart des nœuds capteurs et leur déploiement, en introduisant un ensemble de nœuds capteurs plus puissants. Ceci permet de décharger la majorité des nœuds simples à faible coût de plusieurs fonctions du réseau. L'architecture hiérarchique est composée de plusieurs couches : une couche de capteurs, une couche de transmission et une couche de point d'accès.

– **Le traitement interne " in-network processing "** [21] **:** C'est un mécanisme de sécurité, qui permet des transformations dans le réseau comme l'agrégation de données et la participation passive. Ce traitement permet de réduire d'une manière significative l'énergie consommée dans les $RCSF_s$.

– **Attaques de consommation d'énergie (DoS)** [44] **:** Le déni de service, ou DoS (Denial of Service), est une attaque qui vise à rendre indisponible un

service, un système ou un réseau. Ces attaques s'appuient généralement sur une faiblesse d'implémentation, ou bogue, ou sur une faiblesse d'un protocole.

– **ECC [46] :** Est une approche de cryptographie à clé publique basée sur les aspects mathématiques des courbes elliptiques. Son avantage par rapport a un algorithme comme RSA base les nombres premiers est d'utiliser des clés de taille bien plus petite.

– **ECDSA [46] :** Cet algorithme assure l'authentification de l'utilisateur. Les fonctions ECDSA sont définies dans huecc.h. La génération de signature et de vérification doivent êêtre appliquée à un résumé du message. Un objet de paramètres ECC est requis pour générer ou vérifier une signature à l'aide d'ECDSA. La fonction hu_ECCParamsCreate () crée ces objets. Il faut aussi un objet clé ECC. Un contexte RNG est obligatoire si la génération de clés sera effectuée.

– **Types de capteurs :** Il existe actuellement un grand nombre de capteurs, avec des fonctionnalités diverses et variées. Cependant, on ne pourra pas décrire tous ces différents capteurs, et donc une liste exhaustive peut être trouvée sur le site The Sensor Network Museum [48].La plupart des capteurs dépendent de l'application pour lesquels ils ont été conçus (capteurs aquatiques, sous-terrain, etc.). Bien qu'ils soient différents, ces modèles ont en commun les mêmes composants de base illustrés dans la figure 1.1, ainsi les différents composants de chaque modèle et leurs caractéristiques sont donné dans le tableau suivant [49] :

Nom du capteur	**MCU**	**RAM**	**Flash**	**Stockage**	**Radio**	**Dimension**
Spec node (2003)	AVR RISC 8 bit	3KB	0KB	0KB	RF	2 x 2.5mm
pParticle V2/2x	PIC 18f6270	4KB	128KB	512KB	TR1001	45 x 27mm
Mica (2001)	ATMega 128	4KB	128KB	512KB	TR1000	
Mica2 (2002)	ATMega 128	4KB	128KB	512KB	CC1000	58 x 32 x 7mm (x2AA)
Mica2Dot (2002)	ATMega 128	4KB	128KB	512KB	CC1000	25 x 6mm (x2AA)
MicaZ (2004)	ATMega 128	4KB	128KB	512KB	CC2420	58 x 32 x 7mm (x2 AA)
Rene2 (2000)	ATMega 163	1KB	16KB	32KB	TR1000	
TelosA (2004)	TI MSP430v	2KB	60KB	512KB	CC2420	
TelosB (2004)	TI MSP430	10KB	48KB	1MB	CC2420	65 x 31 x 6mm (x2AA)
Tmote Sky (2004)	TI MSP430	10KB	48KB	1MB	CC2420	3.2 x 8 x 1.3cm
BTnode3 (2004)	ATMega 128	64KB	128KB	180KB	CC1000 /ZV4002 Bluetooth	58.15 x 33mm (x2AA)
Imote (2003)	ARM7	64KB	512KB	0KB	ZV4002 Bluetooth	
Imote2 (2007)	Intel PXA271 Xscale	256KB	32KB	0KB	CC2420	36 x 48 x 9mm
Iris (2008)	ATmega 1281	8KB	128KB	512KB	CC2420	58 x 32 x 7mm (x2AA)
Stargate (2003)	Intel PXA255 Xscale	64MB	32MB		Bluetooth ou IEEE 802.11	9.53 x 6.33 x 1.86m

TABLE A.1 – Caractéristiques des capteurs les plus courants.

– **ECIES** [46] **:** Le plan de chiffrage intégré par courbe elliptique (ECIES) est un plan de chiffrage de clé publique. Les fonctions d'ECIES sont définies dans huecc.h. Un objet de paramètres de CCE est exigé pour chiffrer ou déchiffrer utilisant ECIES. La fonction de hu_ECCParamsCreate () crée ces objets.

– **ECDH** [46] **:** L'algorithme basé sur courbe elliptique d'accord de clé de Diffie−Hellman (ECDH) permet à deux parties de partager une valeur secrète commune. Les fonctions d'ECDH sont définies dans huecc.h. Un objet de paramètres de CCE est exigé pour exécuter l'accord principal d'ECDH. La fonction de hu_ECCParamsCreate () crée ces objets .

– **EROM :**Cette partie est la base du logiciel. Il contrôle l'interface de Flash, le chargeur de démarrage, etc.

– **La zone de couverture** [42] **:** Un nœud capteur permet de surveiller une zone appelée zone de couverture, cette zone est souvent considérée comme un disque de rayon R_c. Un nœud est capable de détecter n'importe quel événement qui se passe dans sa zone de couverture.

– **La zone de communication** [42] **:** La vision d'un capteur dépend du rayon de réception de son module de communication Rcom. Un nœud ne peut pas communiquer avec un deuxième sauf si ce dernier se trouve dans sa zone de communication, c'est à dire si la distance Euclidienne entre les deux nœuds est plus petite ou égale à Rcom. Souvent le rayon de communication (R_{com}) est considéré plus grand que le rayon de couverture (R_c).

Bibliographie

[1] MERRANI Nassima, KHIMOUM Nadia. ' Simulation et évaluation de protocoles de gestion de clés dans les réseaux de capteurs'. Mémoire d'ingénieur d'état en informatique. Bejaia 2009.

[2] Samra BOULFEKHAR. ' Approches de minimisation d'énergie dans les réseaux de capteurs'. Mémoire de Magistère En Informatique. Bejaia 2006.

[3] YACINE YOUNES. ' Minimisation d'énergie dans un réseau de capteur'. Mémoire de magister en informatique option systèmes informatiques . Université Mouloud Mammeri Tizi-Ouzou. 2012.

[4] A.Montoya, D.C..Restrepo et D.A.Ovalle. 'Artificial Intelligence for wirless sensor networks Enhacement'. Smart Wireless Sensor Networks . 2010.

[5] Sofiane MOAD.'La consommation d'énergie dans les réseaux de capteurs sans fils', Mémoire Master Recherche 2 en Informatique, IFSIC-Rennes .2007/2008.

[6] Qinghua Wang et Ilangko Balasingham.' Wireless Sensor Network_ an Introduction'. Dept. of Electronics and Telecommunications Norwegian University of Science and Technology .Norway .2010.

[7] Ben L.Titzer et J.Palsberg. 'Nonintrusive Precision Instumentation of Microcontroller Software'. ACM, New York, ETATS-UNIS, Volume 40 Issue 7, Pages 59−68, juillet 2005.

[8] H.Karl et A.Willig. 'Protocols and Architectures for Wireless Sesor Networks'. John Wiley & Sons Ltd, 2005.

[9] Z.BENCHAABANE, K.OUSSALAH. ' Conception et Implémentation d'un Système de Gestion de clés pour les Réseaux de Capteurs Sans Fils '.mémoire d'ingénieur d'Etat en Génie Informatique. Université de Bejaia 2011.

[10] Gil DE SOUSA. ' Etude en vue de la réalisation de logiciels bas niveau dédiés aux réseaux de capteurs sans fil : microsystème de fichiers' .Thèse de doctorat, Université Blaise Pascal. 2008.

[11] Yassine CHALLAL. 'Réseaux de capteurs sans fils'. Systèmes intelligents pour de transfert.Université de Technologie de Compiègne ,Heudiasuc , France.17/11/ 2008.

[12] Messai Mohamed Lamine. 'Sécurité dans les réseaux de capteurs sans-fils'. Mémoire de magister en informatique, option réseau et système distribués. Universite Abderrahmane Mira de Bejaia .2008.

[13] Samir ATMANI. ' Protocoles de sécurité pour les réseaux de capteurs sans fils '. Mémoire de magister en informatique option Ingénierie des systèmes d'information. Université Hadj lakhder Batna. 15/07/2010.

[14] Nouredine LASLA. 'La gestion de clé dans les réseaux de capteur sans fils'. Mémoire de magister, Institut National de formation en Informatique (I.N.I) Ouad-Smar, Alger.2007.

[15] 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, pp. 206−215, Octobre 2004.

[16] SUN Dong-Mei, HE Bing. 'Review of Key Management Mechanisms in Wireless Sensor Networks'. Vol. 32, No. 6 .ACTA AUTOMATICA SINICA.Novembre 2006.

[17] An Liu, Peng Ning. 'TinyECC : A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks'. Department of Computer Science North Carolina State University Raleigh, NC 27695.

[18] Certicom Research. 'Standards for efficient cryptography −SEC 1 : Elliptic curve cryptography'. http ://www.secg.org/download/aid−385/sec1_final.pdf, Septembre 2000.

[19] D. Hankerson, A. Menezes, and S. Vanstone. 'Guide to Elliptic Curve Cryptography'. Springer, 2004.

[20] http ://www.apprendre-en-ligne.net/crypto/menu/index.html.(dernier consultation juin 2013.)

[21] S. Zhu, S. Setia, and S. Jajodia, 'Leap : efficient security mechanisms for large-scale distributed sensor networks'. CCS 03 : Proceedings of the 10th ACM conference on Computer and communications security (New York, NY, USA).2003.

[22] O. Goldreich, S. Goldwasser, and S. Micali. 'How to Construct Random Functions'. Journal of the ACM, Vol. 33, No. 4, pp 210−217 ,1986.

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

[24] Haowen Chan ,Adrian Perrig , Dawn Song . 'Random Key Pré-distribution Schemes for Sensor'. Université de Carnegie Mellon . 1−1−2003.

[25] http :www.scholar.google.com/scholar_urlhl=fr&q=http :/citeseerx.ist.psu.edu /viewdoc /download%3Fdoi%3D10.1.1.140.7266%26rep%3Drep1%26type%3D pdf&sa=X &scisig=AAGBfm1MpFbOeISgYMhKsuOJbz4Q2q−jbw&oi=scholarr.

[26] W. Diffie and M.E. Hellman. 'New directions in cryptography'. IEEE Transactions on Information Theory, IT−22 :644−654,Journal & Magazines. Novembre 1976.

[27] V. Gayoso Martínez, F. Hernández Álvarez, L. Hernández Encinas. 'Analysis of ECIES and Other Cryptosystems Based on Elliptic Curves'. Department of Information Processing and Coding Applied Physics Institute, CSIC, Madrid, Spain.

[28] S. Zhu, S. Setia, and S. Jajodia, 'LEAP+ : Efficient Security Mechanisms for Large−Scale Distributed Sensor Networks'.Journal ACM Transactions on Sensor Networks TOSN, vol. 2, no.4, pp. 500−528,New York, NY, USA .2006.

[29] Messai Mohamed Lamine 1, Aliouat Makhlouf 2 . 'Protocole Efficace de Gestion des Clés dans les Réseaux de Capteurs Sans-fils'. (UAMB, Ecole Doctorale en informatique ReSyD Bejaia, Algérie, messai.amine@gmail.com) 1, (Département informatique, UFAS Sétif, Algérie, aliouat_m@yahoo.fr) 2 .18−11−2009.

[30] J. Deng, C. Hartung, R. Han, and S. Mishra, 'A Practical Study of Transitory Master Key Establishment for Wireless Sensor Networks'. Computer Science Department University of Colorado at Boulder. Boulder.Septembre 2005.

[31] Kalpana Sharma, M. K. Ghose, Jr, and V. K. Singh. 'A Novel Secure Keying Technique for the Wireless Sensor Networks'. International Journal of Information and Electronics Engineering, Vol. 2, No. 4. India. Juillet 2012.

[32] Bi Jiana, E Xu. 'An Energy−efficient Security Node−based Key Management Protocol for WSN'. the 2nd International Symposium on Computer, Communication, Control and Automation (ISCCCA−13). Université de Bohai Jinzhou, China .2013.

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

[34] C. Blundo et al, "Perfectly−secure key distribution for dynamic conferences". In Proceedings of the 12th Annual International Cryptology Conference on Advances in Cryptology, Berlin : Spring−Verlag. 1992.

[35] Mohammed A. Abuhelaleh and Khaled M. Elleithy. 'SECURITY IN WIRELESS SENSOR NETWORKS : KEY MANAGEMENT MODULE IN SOOAWSN'. School of Engineering University Of Bridgeport, Bridgeport. Octobre 2010.

[36] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. 'Energy-efficient communication protocol for wireless microsensor networks'. Conf. on System Sciences, pages 4-7,In IEEE Hawaii Int. Janvier 2000.

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

[38] Mr ZNAIDI Wassim. ' Quelques propositions de solutions pour la sécurité des réseaux de capteurs sans fil'. Informatique et Mathématiques de Lyon. L'Institut National des Sciences Appliquées de Lyon.Lyon 10 Octobre 2010.

[39] L. Lamport, 'Password authentication with insecure communication'. Technical Note Operating Systems. ACM, vol. 24, no. 11, pp. 770-772, SRI International. 1981

[40] Ashok Kumar Das1, Debasis Giri2. 'An Identity Based Key Management Scheme in Wireless Sensor Networks'.Indian Institute of Technology, Kharagpur, India. 24 Mar 2011.

[41] O. Goldreich, S. Goldwasser, and S. Micali. 'How to construct random functions'. Journal of the ACM,33(4) :792−807. Octobre 1986.

[42] Samira ALLAM. 'Approche Multi agents pour contrôler l'inondation dans un réseau de capteurs'. Mémoire d'ingénieur en informatique, Option Systèmes Informatiques. Ecole nationale supérieure d'informatique (ESI). Oued-Smar, Alger.2009.

[43] D. Cvrcek and P. Svenda. 'Smart Dust Security − Key Infection Revisited'. Electronic notes in Theoretical Computer Science 157, pp. 11−25, Elsevier 2006.

[44] Cédric Llorens, Laurent Levier, Denis Valois, Benjamin Morin. ' Tableau de bord de la Sécurité Réseau '.www.editions−eyrolles.com.

[45] DESTREE Lucile, MARCHAL Mickaël. 'Mini- RSA : Programme d'initiation au chiffrement RSA'.

[46] http ://developer.blackberry.com/native/reference/bb10/crypto_libref/topic /ecc_ecies.html(dernier consultation juin 2013)

[47] Yasser ROMDHANE. ' Evaluation des performances des protocoles S-MAC et Directed Diffusion dans les réseaux de capteurs '. Cycle de formation des ingénieurs en Télécommunication, Option : Réseaux et Services mobiles. Ecole supérieur des communications de Tunis.2007.

[48] The Sensor Museum. In http ://www.btnode.ethz.ch/Projects/ SensorNetwork-Museum/.

[49] David MARTINS. ' Sécurité dans les réseaux de capteur sans fils Stéganographie et réseaux de confiance'. Mémoire de Doctorat. Spécialité informatique. Université de FRANCHE−COMTE. Besançon .2010.

[50] SAYAD MAYA. ' Energy Efficient Protocol (EEP) : Un Protocole de Routage Efficace en Energie pour Réseaux de capteurs sans fils '. Mémoire d'ingénieur d'état en Informatique option : Système Informatique. Oued-Smar, Alger 2008.

[51] Pengcheng Zhao, Yong Xu, Min Nan. 'A Hybrid Key Management Scheme Based on Clustered Wireless Sensor Networks'. Department of Mathematics and Computer Science, Anhui Normal University,China. 2012.

[52] F. Hu and N. K. Sharma. 'Security considerations in ad hoc sensor networks'. Ad Hoc Networks 3, pp. 69−89, Elsevier Science 2005.

[53] A.D. Wood and J.A. Stankovic. 'Denial of services in sensor networks'. IEEE Computer, Octobre 2002.

[54] C. Hartung, J. Balasalle, and R. Han. 'Node compromise in sensor networks : The need for secure systems'. Technical Report CU-CS-990- 05, Department of Computer Science,University of Colorado at Boulder, Jan 2005.

[55] Xun Wang, Wenjun Gu, Wei Yu, and Dong Xuan. 'Search-based physical attacks in sensor networks'. The Department of Computer Science and Engineering. USA 2005.

[56] Chris Karlof , David Wagner. ' Secure routing in wireless sensor networks : attacks and countermeasures'. Ad Hoc Networks :293_315, University of California at Berkeley, Berkeley, CA 94720, USA. 2003.

[57] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. 'Wormhole attacks in wireless networks'. IEEE Journal on Selected Areas in Communications :370_380, 2006.

[58] J. Newsome, E. Shi, D. Song, and A. Perrig. 'The sybil attack in sensor networks : analysis & defenses'. Université de Carnegie Mellon. 2004.

[59] Frank Stajano and Ross J. Anderson. 'The resurrecting duckling : Security issues for ad hoc wireless networks'. Security Protocols Workshop, volume 1796 of Lecture Notes in Computer Science, pages 172_194. Springer, 1999.

[60] W. Heinzelman, A. Chandrakasan, H. Balakrishnan. 'Energy-Efficient Communication Protocol for Wireless Micro sensor Networks'. In proc of the Hawaii International Conférence on Systems Science, vol. 8, pp. 8020,Janvier 2000.

[61] Emmanuel. Bresson. 'Cryptographie : chiffrement par flot', Séminaire de la cryptographie, Page(s) :22−34, Laboratoire de cryptographie, Université de parus XII 2001/2002.

[62] Gunnar Gaubatz, Jens−Peter Kaps, and Berk Sunar.' public key cryptography in sensor networks_Revisited'. Department of Electrical & Computer Engineering, Worcester Polytechnic Institute, U.S.A 2001.

[63] X. Perseguers. 'la sécurité dans les réseaux de capteurs sans fils'. Mémoire Master, Ecole Polytechnique Fédérale de Lausanne (EPFL) et Centre Suisse d'Electronique et de Microtechnique (CSEM). Fevrier 2005.

[64] http ://www.awt.be/web/sec/index.aspx ?page=sec,fr,100,010,001.(Dernière consultation juin 2013).

[65] Didier BALLOY.' Le RISQUE INFORMATIQUE Comment y remédier ? '. Mémoire d'ingénieur C.N.A.M. en Informatique Réseaux Système Multimédia. PARIS. 21 janvier 2002.

[66] http ://www.anti-cybercriminalite.fr/article/les-menaces-contre-la-s%C3%A9 curit%C3%A9-informatique .(Dernière consultation juin 2013).

[67] Laurent Poinsot. 'Introduction à la sécurité informatique'. Université Paris 13 − Institut Galilée.

[68] A.Bachir, A.Ouadjaout, L.Khelladi, M.Bagaa, N.Lasla, Y.Challal. 'Information Security in Wireless Sensor Networks'.On Ad hoc and Ubiquitous Computing, edited by : Agrawal Dharma P., and XIE Bin .,World Scientific, 2009.

[69] http ://www.repo.zenk-security.com/Cryptographie%20.%20Algorithmes%20.%20 Steganographie/Data%20Encryption%20Standard%20(DES).pdf.(dernier consultation juin 2013).

[70] H.Schyns. 'Cours de Mathématiques :Chiffrement RSA'.ENSIGNEMENT DE PROMOTION SOCIALE. Mai 2008.

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

[72] Bénoni Martin. 'Stéganographie : techniques'. France. 2007.

[73] Johann Barbier. 'La Stéganographie Moderne : d'Hérodote à nos Jours'. Département de Cryptologie du CELAR (France), laboratoire de Virologie et Cryptologie de l'ESAT. 2007.

[74] http ://www.futura-sciences.com/fr/definition/t/informatique−3/d/simulation informatique_11319/.(Dernière consultation juin 2013.)

[75] http ://www.mathworks.com/products/matlab/.(Dernière consultation juin 2013.)