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

Université Abderrahmane Mira de Béjaïa

Faculté des Sciences Exactes

Département d'Informatique

Mémoire de Fin de Cycle

En vue de l'obtention du diplôme de Master recherche en Informatique

Spécialité: Réseaux et Systèmes Distribués

Thème

Gestion de clés dans les Réseaux de Capteurs Sans Fil

Réalisé par :

M^{lle} Ben Zerrouk Amel & M^{lle} Yaici Sara

Devant le jury composé de:

Président :	M.	ACHROUFENE	Achour
Examinateur :	M^{lle}	CHERFA	Hamida
Encadreur :	M.	AISSANI	Sofiane

Promotion 2015/2016

⁕*Remerciements* ⁕

***N**ous rendons grâce à Dieu le tout puissant et mésericordieux de nous avoir donné le courage et la patience de mener à bout ce modeste travail.*

***N**os remerciements s'adressent à notre promoteur en l'occurence M. AISSANI Sofiane, pour ses conseils, sa disponibilité et ses encouragements.*

***N**os vifs remerciements s'dressent aux membres du jury d'avoir accepté d'évaluer notre travail et pour l'intêret qu'ils y portent.*

***U**n grand merci à nos familles, pour leur soutien permanent, leur patience et leurs encouragements qui nous ont permis de trouver la force et la volonté cachées au plus profond de nous mêmes.*

***U**n merci à tous ceux qui ont contribué de près ou de loin à la concrétisation de ce travail.*

⁂*Dédicaces* ⁂

À mes très chers parents

À ma très chère soeur Miada et son mari Adel
À mon futur neveu à qui je souhaite un avenir lumineux

À mon merveilleux et unique frère Abdellah à qui je souhaite un avenir prospère

Vous vous êtes dépensés pour moi sans compter

En reconnaissance de tous les sacrifices consentis par tous et chacun pour me permettre d'atteindre cette étape de ma vie

À mes oncles, tantes, cousins et cousines

À tous ceux qui m'ont prêté leur attention, m'ont consacré leur temps et ont cru en moi et en ma force depuis toujours

À mon binôme Sara pour ces quatres dernières années qu'on a partagé ensemble

Je dédie ce travail.

Amel

À mes chers parents, pour leurs soutiens et encouragements, j'espère être à la hauteur de vos epérances

À ma soeur ainée Lydia en qui j'ai toujours pris exemple, à mes deux adorables soeurs Mounia, Lisa et mon seul et unique petit frère Reda pour leur présence et leur aide

À toute mon honorable et grande famille

À tous ceux qui m'ont fait confiance et ont été présents pour moi

À mon binôme Amel pour les moments inoubliables que nous avons passé ensemble

Je dédie ce travail.

Sara

Table des Matières

Liste des Acronymes

Ack	Acknolegement
AES	Advanced Encryption Standard
BS	Base Station
CH	Cluster Head
ECC	Elliptic Curve Cryptography
ECDSE	Elliptic Curve Digital signature Encryption
EG	Eschenauer-Gligor
EHKM	Efficient and Hybrid Key Management
IBKM	Identity-Based Key Management
ID	Identifiant
KDC	Key Distribution Center
LEAP	Light weight Extensible Authentication Protocol
LEKM	Low-Energy based Key Management
LKE	Location-Aware Key
LPKM	Light-weight Polynomial Key Management
MCA	Mobile Certification Authority
MD-5	Message Digest
RC-4	Rivest's Cipher 4
RC-5	Rivest's Cipher 5
RN	Random Number
RCSF	Réseaux de Capteurs Sans Fil
SHA	Secure Hash Algorithm
TESLA	Timed Efficient Streaming Loss Tolerant Authentication
VEBEK	Virtual Energy-Based Encryption and Keying
VDKM	Virtual Dynamic Keing Module
VLKM	Virtual-Location Key Management
WSN	Wireless Sensor Networks

Liste des figures

Liste des tableaux

Introduction Générale

Les avancées récentes dans la technologie, les communications sans-fil ont rendu possible le développement de capteurs multifonctionnels avec des coûts réduits, une consommation efficace de l'énergie et qui sont dotés de capacité à communiquer par diffusion radio à portée réduite. Ce concept de capteur miniature, composé d'unités de capture, de traitement de données et de communication a encouragé l'idée d'une nouvelle variante de réseaux ad hoc : les réseaux de capteurs sans fil, basés sur un effort collaboratif d'un grand nombre de noeuds. Les réseaux de capteurs représentent une grande avancée par rapport aux capteurs classiques et sont considérés comme une des technologies les plus importantes pour beaucoup d'applications à temps-réel.

Comme beaucoup de développements technologiques, les réseaux de capteurs sans fil ont émergé pour des besoins militaires tels que la surveillance sur le terrain de combat. Puis, ils ont trouvé leurs chemin pour des applications civiles. Aujourd'hui, les réseaux de capteurs sans fil sont devenus une technologie clé pour les différents types "d'environnements intelligents" . Ces réseaux sont d'une importance particulière quand un grand nombre de noeuds de capteurs doit être déployé, dans des situations dangereuses. Par exemple, pour une gestion des catastrophes, un grand nombre de capteurs peut être largué par un hélicoptère. Ces capteurs peuvent aider à réaliser des opérations de sauvetage en localisant les survivants, pour l'identification des zones à risque ou pour renseigner l'équipe de secours.

L'énergie limitée dans ce type de capteurs sans-fil et les environnements hostiles dans lesquels ils pourraient être déployés, sont des facteurs qui rendent ce type de réseaux très vulnérables. De même que, l'absence de sécurité physique pour ce type de capteurs et la nature vulnérable des communications radios sont des caractéristiques qui augmentent les risques d'attaques contre ce type de réseau. La confidentialité et l'intégrité des échanges sont des services de sécurité indispensables. Pour

certaines applications des réseaux de capteurs, notamment lorsqu'il s'agit de transporter des informations qui peuvent divulguer le secret médicale ou des informations sensibles, quand il s'agit de prévenir des accidents catastrophiques comme dans les réacteurs nucléaires, leur sécurité est primordiale pour mener à bien leurs opérations.

Problématique

Vu la sensibilité des informations transmises dans les réseaux de capteurs sans fil, la sécurité est une nécessité pour la majorité des applications qui utilisent les RCSFs (Réseaux de Capteurs Sans Fil), donc il est indisponsable que les noeuds de communication générent et partagent des clés cryptographiques entre eux, pour le chiffrement et l'authentification. Les systèmes de gestion de clés dans les réseaux filaires ne conviennet pas aux RCSFs car ils utilisent la cryptographie asymétrique qui nécessite beaucoup de calcul et qui utilisent des messages d'importante taille, d'où le besoin de systèmes de gestion de clés prenant en considération les contraintes de ressources que rencontrent ces RCSFs tels que la capacité de calcul et de stockage réduites ainsi que le problème majeur qui est l'énergie limitée des noeuds capteurs.

Organisation du mémoire

Le premier chapitre présente le concept des réseaux de capteurs sans fil, leurs architectures, caractéristiques, contraintes de conception ainsi que leurs domaines d'application.

Le deuxiéme chapitre quant à lui, décrit l'importance de la sécurité dans les RCSFs, il présente une classification des protocoles de gestion de clés, décrit également quelques protocoles existants dans la literrature, et enfin une étude comparative de ces derniers.

Le troisième chapitre décrit une amélioration d'un protocole existant, il montre également les résultats de simulation, une discussion montrant l'amélioration apportée.

1

Généralités sur les réseaux de capteurs sans fil

1.1 Introduction

Les récentes avancées technologiques, telles que la communication sans fil ou l'électronique digitale ont permis de mettre en marche le développement à faible coût de minuscules capteurs multifonctionnels, qui ont la capacité de communiquer avec fiabilité sur de courtes distances.
Ces capteurs ont influencé l'idée des réseaux de capteurs sans fil (RCSF), (WSN) pour Wireless Sensor Networks.

Dans ce chapitre nous présenterons les réseaux de capteurs sans fil, leurs architectures, leurs caractéristiques, leurs contraintes de conception ainsi que leurs domaines d'application.

1.2 Réseaux de capteurs sans fil

1.2.1 Définition du noeud capteur

C'est un composant éléctronique autonome à faible coût, capable d'effectuer des mesures simples sur son environnement immediat servant á la surveillence et au contrôle d'un phénomène donné. Ces petites entités électroniques, constituent les briques de base des réseaux de capteurs, dont l'objectif est de récolter des grandeurs physiques de leur environnement proche (luminosité, mouvement, température, pression barométrique, etc.), et éventuellement de les traiter et de les communiquer à leurs voisins ou vers un ou plusieurs points de collecte appelés station de base (SB) [1].

1.2.2 Définition d'un réseau de capteurs sans fil

Le déploiement des entités capteurs qui permet de collecter et de transmettre les données mesurées vers un ou plusieurs points de collecte, forme un réseau de capteurs sans fil. Ces réseaux sont composés de centaines, voire de milliers de capteurs avec une infrastrucure décentralisée: tous les noeuds participent au fonctionnement du réseau [2].
La figure ci-après illustre la schématisation d'un réseau de capteurs sans fil.

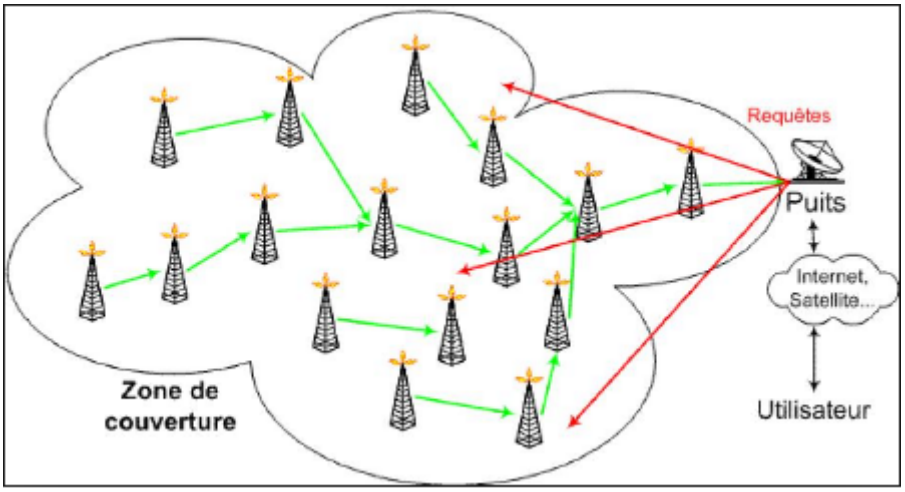

Figure 1.1: Schématisation d'un réseau de capteurs sans fil [2].

1.2.3 Architecture d'un réseau de capteurs sans fil

Un réseau de capteurs est constitué essentiellement de: **plusieurs** noeuds capteurs, **un** noeud Sink et **un** centre de traitement de données.
Les noeuds capteurs transmettent leurs données vers un noeud Sink. Et puisque le centre de traitement des données est éloigné, les données sont acheminées via Internet, une passerelle est utilisée pour adapter le type des données au canal.
Le centre de traitement de données a le rôle de regrouper les informations issues des noeuds et les traiter de façon à en extraire de l'information utile exploitable [3].
Il existe plusieurs critères pour classifier les réseaux de capteurs [4]. En effet, pour chaque type d'application, ces réseaux ont des caractéristiques différentes. Ils se distinguent par le modèle de mobilité dans le réseau, le mode de communication entre les capteurs et le puits et le type des noeuds du réseau, etc.

1.2.4 Caractéristiques d'un réseau de capteurs sans fil

Les réseaux de capteurs sans fil présentent de nombreuses caractéristiques, parmi ces dernières [1]:

- **Durée de vie limitée:** Les noeuds capteurs sont très limités par la contrainte d'énergie, ils fonctionnent habituellement sans surveillance dans des regions géographiques éloignées. Par conséquent, recharger ou remplacer leurs batteries devient impossible.

- **Ressources limitées:** Habituellement les noeuds capteurs ont une taille très petite, ce facteur de forme limite la quantité de ressources qui peuvent être mises dans noeuds, par conséquent, la capacité de traitement et de mémoire devient très limitée.

- **Topologie dynamique:** La topologie des réseaux de capteurs change d'une manière fréquente et rapide car: Les noeuds capteurs peuvent être déployés dans des environnements hostiles, la défaillance d'un noeud capteur est donc très probable.

- **Scalabilité:** Le nombre de noeuds capteurs peut être de centaines ou de milliers (selon l'application).

- **Déploiement aléatoire et dense:** Dans un réseau de capteurs sans fil, les capteurs sont généralement déployés d'une façon dense et plus ou moins aléatoire. La forte densité est souvent liée à des raisons de fiabilité.

- **Auto-organisation:** Pour remédier au problème de changement non prédictible de topologie, une auto-organisation du réseau est nécessaire. C'est-à-dire que les noeuds doivent savoir localiser leurs voisins et établir des routes pour que l'information puisse circuler à travers le réseau.

- **Sécurité limitée:** Les contraintes de limitations physiques font que le contrôle des données transférées doit être minimisé. De plus, les réseaux de capteurs sans fil sont plus sensibles aux attaques qui menacent les donnés transmises en raison de l'absence de l'infrastructure .

1.3 Contraintes de conception des RCSFs

La conception des RCSF est influencée par plusieurs contraintes. Ces facteurs importants servent comme directives pour le développement des algorithmes et protocoles utilisés dans les réseaux de capteurs, ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine [5].

1.3.1 Au niveau de la communication

- Perte de données à cause de la transmission radio : les RCSFs sont des réseaux sans fil, ce qui fait que la portée de la communication est limitée par la capacité de rayonnement des antennes utilisées, à quoi s'ajoute la limitation des renvois des paquets en raison du manque d'énergie.

- La bande passante est limitée et partagée par tous les noeuds du réseau de capteurs.

- Interférences : ce réseau travaille sur une bande de fréquences non propriétaire, ce qui rend leurs communications vulnérables aux problèmes d'interférences.

1.3.2 Au niveau du matériel

Bien qu'ils aient de nombreux avantages, les noeuds capteurs sont caractérisés par des ressources et des composantes plus limitées en raison de leurs tailles réduites, ce qui conduit à la génération de nombreuses contraintes, parmi lesquelles on retrouve [5]:

- Puissance de calcul limitée : fonctionnant dans la majorité des cas avec des registres 8 ou 16 bits, les processeurs des réseaux de capteurs sont différents

de ceux d'une machine classique, car ils utilisent souvent des micro-contrôleurs de faible fréquence.

- Mémoire limitée : 2 à 250 Ko de RAM et 1 à 32 Mo de mémoire flash.
- Environnement : Les noeuds capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement (forte chaleur, pluie, humidité, etc.).
- Consommation d'énergie : dans ces réseaux, les noeuds sont typiquement gérés par la durée de vie de leurs batteries qui est une réserve d'énergie limitée, minimiser la consommation d'énergie est d'une importance primordiale afin de maximiser la durée de vie du RCSF [6].

1.4 Domaines d'application des réseaux de capteur sans fil

Les réseaux de capteurs sans fil ont été classés parmi les 21 technologies les plus importantes du 21éme siècle. En effet, la recherche dans le domaine des capteurs est en train de vivre une révolution importante, ouvrant des perspectives d'impacts significatifs dans de nombreux domaines. Des exemples d'applications potentielles dans ces différents domaines sont exposés ci-dessous [7]:

1.4.1 Applications militaires

Le déploiement rapide, l'auto-configuration et la tolérance aux pannes des réseaux de capteurs sont des caractéristiques qui font de ce type de réseaux un outil appréciable dans un tel domaine. Le déploiement sur un endroit stratégique ou difficile d'accès, afin de surveiller toutes les activités des forces ennemies ou d'analyser le terrain avant d'y envoyer des troupes (pour la détection d'agents chimiques, biologiques ou de radiations, par exemple) [8].

1.4.2 Applications liées à la sécurité

les réseaux de capteurs peuvent être utilisés dans de nombreuses applications de sécurité et de surveillance. 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, sans alimentation électrique ou autres connexions filaires. Les capteurs doivent s'activer périodiquement et peuvent ainsi fonctionner

durant des années, voire des décennies. Un réseau de capteurs de mouvements peut constituer un système d'alarme distribué qui servira à détecter les intrusions sur un large secteur [8].

1.4.3 Applications médicales

Surveillance permanente des patients et une possibilité de collecter des informations physiologiques de meilleure qualité facilitant ainsi le diagnostic de maladies grâce à des micro-capteurs qui pourront être ingèrés ou implantés sous la peau. (i) Les micro-cameras qui peuvent être ingérées sont capables, sans avoir recours à la chirurgie, de transmettre des images de l'intérieur d'un corps humain. (ii) La création d'une rétine artificielle composée d'une centaine de micro-capteurs pour améliorer la vision de l'oeil [9].

1.4.4 Applications commerciales

Il est possible d'intégrer des noeuds capteurs au processus de stockage et de livraison. Le réseau ainsi formé, pourra être utilisé pour connaître la position, l'état et la direction d'un paquet ou d'une cargaison. Il devient alors possible pour un client qui attend la réception d'un paquet, d'avoir un avis de livraison en temps réel et de connaître la position actuelle du paquet. Pour les entreprises manufacturières, les réseaux de capteurs permettront de suivre le procédé de production à partir des matières premières jusqu'au produit final livré [9].

1.5 Conclusion

Dans ce chapitre nous avons présenté de manière générale les réseaux de capteurs sans fil, leurs architectures, leurs principales contraintes de conception ainsi que leurs domaines d'application. Pour pouvoir mettre en oeuvre et mener à bien ces réseaux de capteurs sans fil, la sécurité joue un rôle crucial, l'un des aspects de cette dernière dans ce domaine est la gestion de clés. Dans le chapitre suivant nous étudierons quelques protocoles proposés, portants sur les gstion de clé dans les RCSF, existants dans la littérature.

2

Etat de l'art sur la gestion de clés dans les réseaux de capteurs sans fil

2.1 Introduction

Les résaux de capteurs sans fil sont constitués d'une multitude de petits capteurs, capables de collecter des informations sur l'environnement dans lequel ils sont déployés, et de les envoyer à une station de base.

L'isolement et l'insécurité des terrains où sont déployés les capteurs ont poussé les chercheurs à renforcer la sécurité des données communiquées afin de diminuer le risque d'interception et d'altération. En tenant compte de la contrainte des ressources, il est devenu primordial de personnaliser les mécanismes de sécurité existants afin de permettre une consommation d'énergie minime tout en assurant une sécurité infaillible.

Dans ce chapitre nous parlerons de l'importance de la sécurité dans les RCSFs, de la gestion de clés dans les RCSFs qui est un aspect de cette dernière, nous reprendrons une classification des protocoles de gestion de clés, quelques protocoles existants dans la literrature et nous terminerons par un tableau et d'une discussion comparatifs de

ces protocoles.

2.2 Sécurité et gestion de clés dans le réseaux de capteurs sans fil

2.2.1 Sécurité

La pertinence de la sécurité dans les réseaux de capteurs est étayée par de nombreuses menaces existantes qui peuvent entraver plusieurs fonctionnalités majeures des réseaux mondiaux. En raison des canaux sans fil et les capacités limitées des noeuds capteurs, il peut être relativement facil pour l'adversaire de contrôler ou même prendre le contrôle du comportement d'un RCSF non protégé. Un réseau de capteurs doit être prêt pour prévenir ou minimiser l'effet de ces attaques en utilisant divers mécanismes possibles, tels que la communication sécurisée (canaux sécurisés, protocoles sécurisés: par exemple le routage, l'agrégation, synchronisation de l'heure) etc.
Les primitives de sécurité, telles que la cryptographie à clé symétrique et la cryptographie à clé publique, permet le construction d'une communication sécurisée entre deux ou plusieurs dispositifs, assurer la confidentialité, l'intergritét l'authentification.

2-2-1-1 Mécanismes de sécurité

Plusieurs mécanismes, basés généralement sur la notion de cryptographie, sont mis en place afin de répondre à la question de la sécurité dans les RCSF. La cryptographie est l'étude des techniques mathématiques qui permettent d'assurer certains services de sécurité. Elle permet de convertir des informations en clair en informations cryptées (codées), c'est à dire non compréhensibles, et puis, à partir de ces informations cryptées, de restituer les informations originales [34].

- **Chiffrement**
  Le chiffrement est le système cryptographique assurant la confidentialité. Pour cela,il utilise des clés. Selon cette utilisation, on distingue deux classes de primitives : symétrique ou asymétrique [34].

  1. **Le chiffrement symétrique**

     Une même clé est utilisée entre deux noeuds communicants pour chiffrer et déchiffrer les données en utilisant un algorithme de chiffrement symétrique [34].

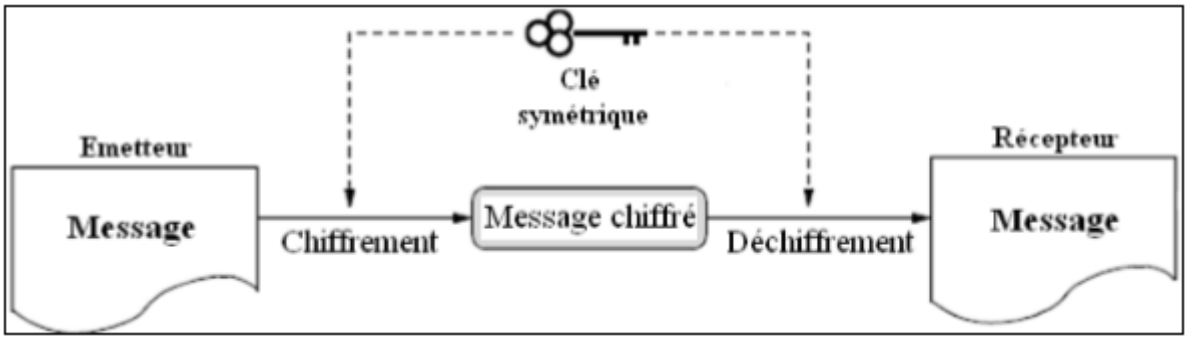

Figure 2.1: Chiffrement Symétrique [34].

2. **Le chiffrement asymétrique**

   Deux clés différentes sont générées par le récepteur: une clé publique diffusée à tous les noeuds 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. 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 [34].

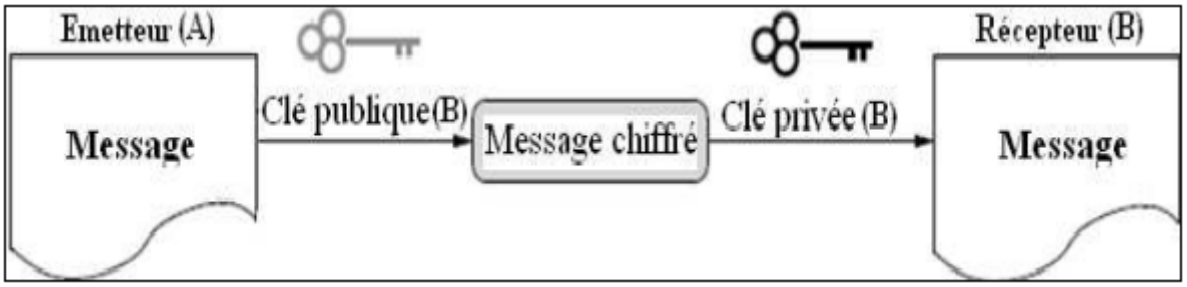

Figure 2.2: Chiffrement Asymétrique [34].

- **La signature digitale**
  La signature digitale est un système cryptographique assurant la non-répudiation de la source. Elle repose sur les 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 avec 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-à-dire, les données proviennent bien de leur émetteur légitime qui ne pourra pas nier l'émission de ces données dans le futur [34].

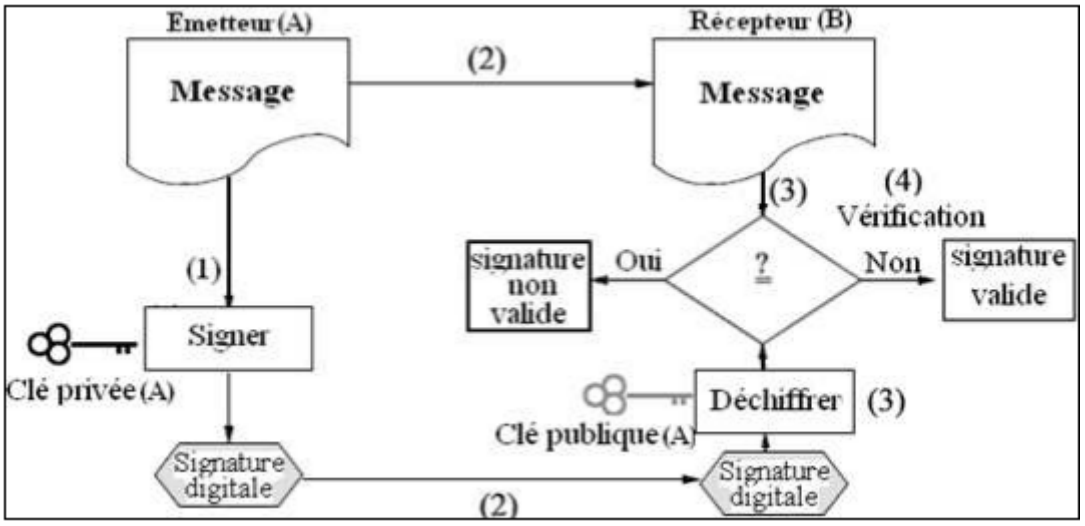

Figure 2.3: Signature Digitale [34].

- **La fonction de hachage**
  C'est le mécanisme qui assure l'intégrité de données. Cette fonction calcule une courte empreinte de taille fixe à partir d'une donnée de taille arbitraire. Cette empreinte est recalculée par le récepteur afin qu'il la compare à celle calculée par l'émetteur. Si elles sont différentes, alors les données ont été altérées pendant leur transmission.

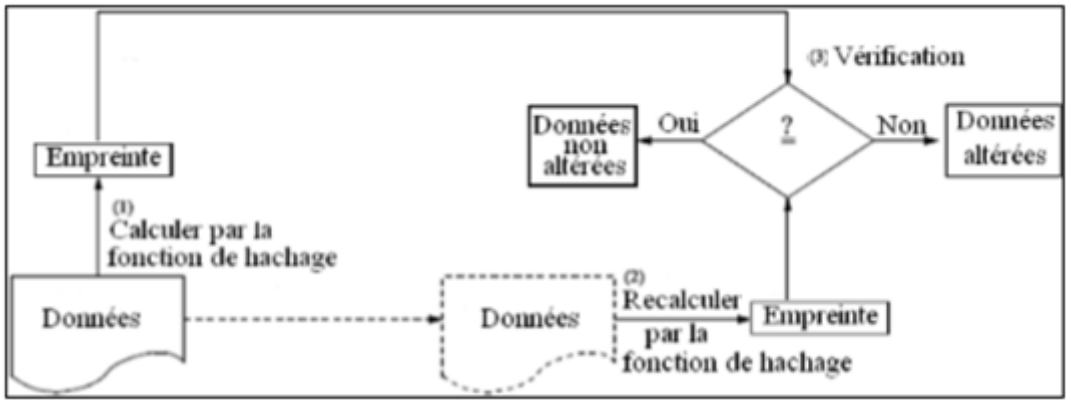

Figure 2.4: Fonction de Hachage [34].

2.2.2 Gestion des clés

La gestion de clés est l'un des aspects les plus difficiles lors de la configuration d'un système cryptographique de sécurité. Pour qu'un tel système fonctionne et soit

sécurisé, chacun des utilisateurs doit disposer d'un ensemble de clés secrètes (dans un système à clés secrètes), ou de paire de clés privée/publique (dans un système à clés publiques). Cela implique de générer les clés et de les distribuer de manière sécurisée aux utilisateurs ou d'offrir à l'utilisateur le moyen de les générer. Il doit aussi pouvoir enregistrer et gérer ses clés publiques et privées de manière sûre.

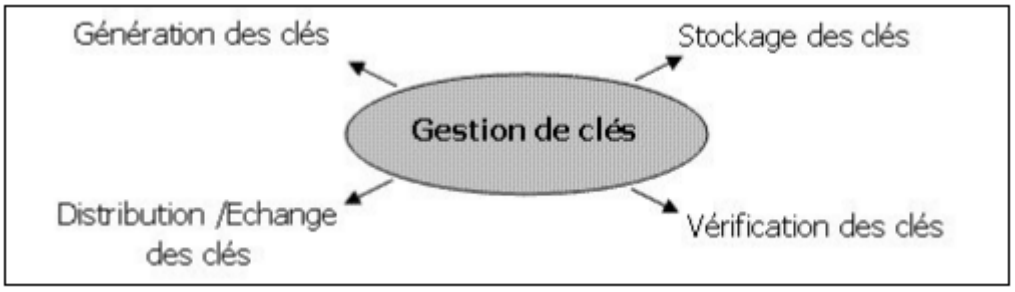

Figure 2.5: Gestion de Clé [34].

2.3 Classification des systèmes de gestion de clés dans les RCSFs

Les réseaux de capteurs sans fil étant devenus très répandus, plusieurs chercheurs se sont penchés sur la necessité de l'utilisation de la cryptographie dans la communication inter-noueuds aboutissant à des protocoles de gestion de clés de cryptage.

Dans cette partie nous reprendrons un article où les auteurs [10] ont regroupé les dernières recherches menées dans ce domaine affirmant qu'un état de l'art est necessaire pour définir les nouveaux défis rencontrés.
Une étude des techniques de gestion de clés fait l'objet de cette partie en considérant les diffrentes étapes impliquées lors de l'établissement des clés; l'authentification, la génération, la distribution et la gestion des clés.

2.3.1 Systèmes d'authentification

L'authentification est l'aspect fondamental d'une communication sécurisée. Selon qu'elle soit unicast, multicast ou broadcast, plusieurs techniques d'authentification ont été développées dont **TESLA** *Timed Efficient Streaming Loss tolerant Authentication* et **LEAP** *Light weight Extensible Authentication Protocol.*

TESLA [11] est une technique de cryptographie asymétrique pour une communication multicast, qui ajoute une signature numérique de 24 bits à chaque message sortant. L'authentifcation est ainsi assurée, l'utilisation de clés asymétriques conduit, néanmoins à un épuisement rapide de l'énergie finie des capteurs. **uTESLA** [12][13], utilisée en broadcast, a été introduite pour remédier à ce problème, en se basant sur un envoi retardé des clés et une restriction du nombre d'authentifications. Plusieurs améliorations, dites multiniveaux, de **uTESLA** ont aussi été introduites pour contrer le problème de lenteur lors de l'authentification.

LEAP+ [14], une extension du protocole **LEAP** [38], est une méthode assurant l'authentification et est adaptée pour les réseaux de capteurs homogènes. **LEAP+** utilise une fonction pour générer des clés pseudo aléatoirement qui seront communiquées à la station de base, et les noeuds communiquent entre eux via des paires de clés calculées en se basant sur leurs identités. Ce protocole considère les réseaux comme étant sécurisés pour un certain intervalle de temps ce qui peut laisser places à plusieurs attaques sur le routage.

Il existe plusieurs autres techniques d'authentification basées sur la cryptographie symétrique telles que **TinySec** [15][16], **RC5**, **SHA-1**, **AES** et **MD5** [17-20].

2.3.2 Approches pour la génération de clés de communication

La génération des clés est une partie importante dans la gestion des clés, pour se faire il existe plusieurs approches, parmi elles; l'approche basée sur les polynomes et l'approche basée sur la conception combinatoire.

2-3-2-1 Approche basée sur les polynômes

Cette approche basée sur la génération de clés symétriques est couramment utilisée dans les réseaux de capteurs. Dans la méthode **E-G** (Eschenauer-Gligor) [21], un serveur génère aléatoirement un polynôme bi-varié et symétrique ***f(x, y)*** de degré **t** sur un corps fini de cardinalité **q** noté **Fq**.

$f(x,y) = \sum_{i,j=0}^{t} a_{i,j} x^i y^j$ où $f(x,y)=f(y,x)$.

Chaque noeud se voit attribuer une partie du polynôme. Par exemple, le noeud **k** possède la partie $f(k,y)$ tandis que **l** aura la partie (l,y)

$f(k,l)$ et $f(l,k)$ avec $f(k,l) = f(l,k)$

2-3-2-2 Approche basée sur la conception combinatoire

La théorie de la conception traite de l'organisation d'éléments d'ensemble fini en sous-ensembles; (blocs), en satisafaisant plusieurs contraintes. Une conception combinatoire régulière peut être exprimée sous forme d'un quadruplet **(v, b, r, k)** avec **b * k = v * r** où, **v** éléments de **S** sont agencés en blocs **b** tel que chaque bloc contient **r** éléments et que chaque élément doit apparaître **k** fois dans tous les blocs de **b**.

Cette technique est utilisée dans [22] où les noeuds sont identifiés par blocs. La clé (x, y, z) sera ainsi assignée au noeud **(a,b,c)** seulement si $ax + by + cz = 0$.

2.3.3 Schémas de distribution et gestion des clés

En général, les modèles de distribution des clés sont divisés en deux catégories: la distribution statique et la distribution dynamique.

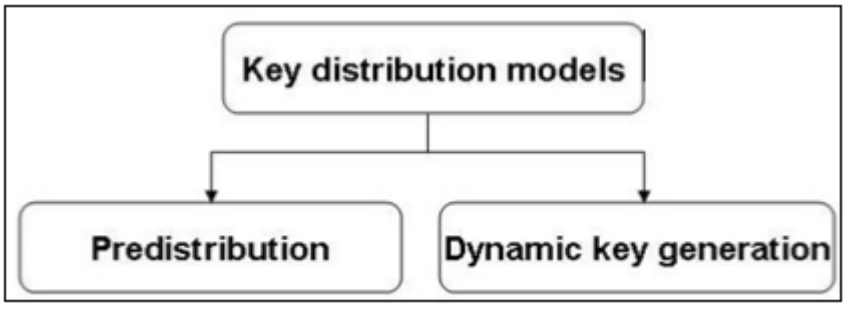

Figure 2.6: Modèles de distribution de clés [10].

Dans la distribution de clés statique, les clés sont chargées dans les noeuds capteurs avant le déploiement, de telle sorte que chaque noeud puisse calculer la clé commune qu'il partagera avec ses voisins. la communication ne sera établie qu'après le calcul de cette clé. Par contre, dans le cas dynamique les clés sont générées dynamiquement durant la communication.

Lors de la distribution des clés, on distingue 3 modèles de chiffrement :

- **Chiffrement par paires:** Chaque noeud doit posséder **(N-1)** clés dans un réseau de **N** noeuds.
- **Chiffrement par clusters :** Les noeuds appartenant à un même cluster partagent une seule clé.

- **chiffrement par réseaux :** Une seule clé est partagée entre tous les noeuds du réseau. Ce dernier modèle s'avère être celui dont la consommation d'énergie est la plus faible.

2-3-3-1 Approches pour la distribution des clés

Dans la gestion des clés, une étape cruciale étant la distribution des clés précédemment générées. Assurer la non-interception des clés lors de leur diffusion est l'objectif principal des quelques approches citées ci-dessous.

- **Approche basée sur la localisation**
  Dans la génération de clés aléatoires, les noeuds requièrent beaucoup de temps et d'énergie pour calculer les clés partagées. Une approche basée sur la localité des noeuds survient afin d'alléger ce problème. Elle est souvent utilisée dans des applications comme la détection de l'humidité où le noeud connait sa localité en permanence. **LKE** [23] , **FRP** [24], **DDHC** [25] sont des protocoles utilisant les coordonnées des noeuds pour établir une connexion sécurisée.

- **Approche basée sur l'échange**
  Sushmita Ruj et all [26] ont proposé un concept de distribution dans lequel chaque trois noeuds du réseau peuvent partager une clé commune. Ils ont aussi introduit une distribution d'un unique couple de clés partagé entre **k** noeuds voisins. II est inspiré par un protocole proposé par *Joux* [27] où 3 noeuds peuvent s'échanger une clé partagée en une seule diffusion.

- **Approche basée sur les matrices**
  Dans [28], une approche matricielle a été introduite pour augmenter la connectivité du réseau, sécuriser les liens établis, augmenter la résilience ainsi que minimiser l'espace mémoire utilisé par les noeuds pour stocker les clés. Une matrice **K** contenant toutes les clés du réseau est divisée en deux matrices **L** représentant la matrice triangulaire inférieure et **U** représentant la matrice triangulaire supérieure. Les noeuds utilisent des parties de ces matrices pour s'authentifier entre eux.

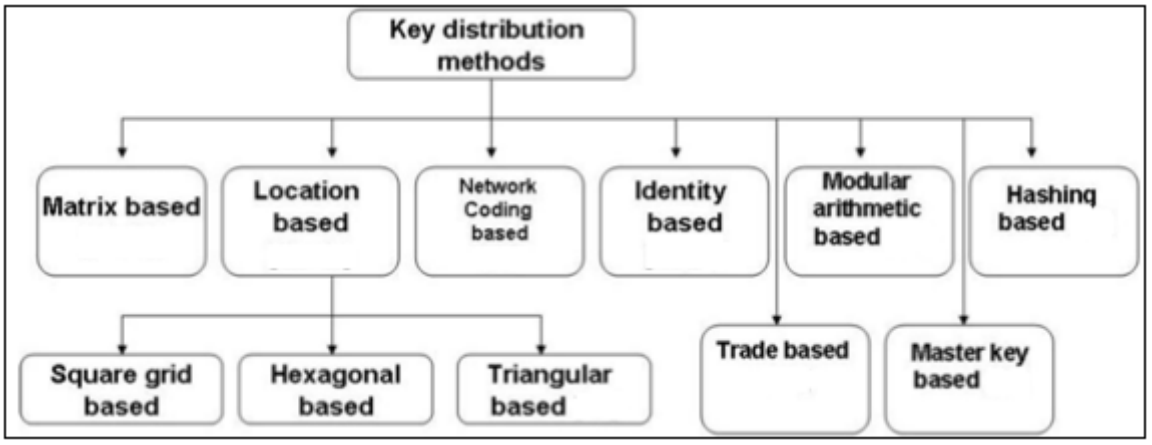

Figure 2.7: Méthodes de distribution de clés [10].

2.4 Description de quelques protocoles de gestion de clés dans les RCSFs

Mettre en oeuvre des systèmes de gestion de clés économes en énérgie et assurant la sécurité des RCSFs est un défi pour les chercheurs à cause des contraintes de ressources que rencontrent les noeuds capteurs dans les RCSFs.
Dans cette partie nous étudierons quelques protocoles de gestion de clés pour les RCSFs proposés dans la littérature.

2.4.1 VLKM: Virtual Location-Based Key Management Scheme for Wireless Sensor Networks

VLKM (*Virtual Location-Based Key Management Scheme for Wireless Sensor Networks*) est un protocole de gestion de clés basé sur la localisation virtuelle, qui sera utilisée pour la génération de clés pour chaque tour [29].

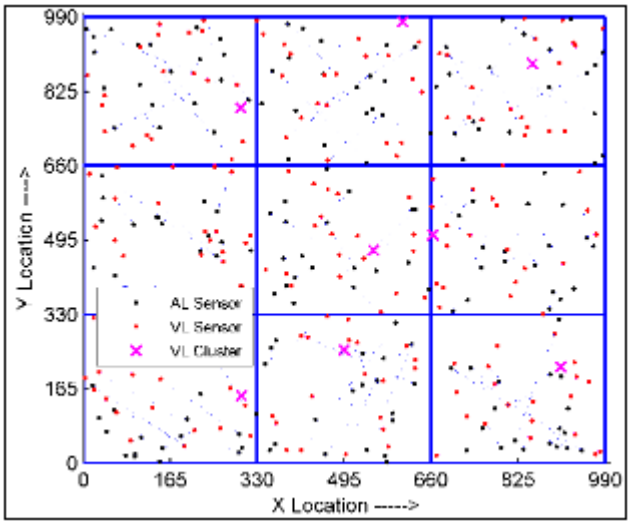

Figure 2.8: Déploiement aléatoire des noeuds et localisation virtuelle [29] .

2-4-1-1 Type de clés utilisées

- **Cluster key**
  C'est une clé partagée par tous les noeuds du même cluster, elle est également connu par la station de base. Cette clé est utilisée lors de la communication des noeuds avec le Cluster Head du même cluster.

- **Sensor key**
  Cette clé est partagée entre le noeud capteur et la station de base. Le Cluster Head utilise cette clé pour communiquer avec la station de base.

2-4-1-2 Génération de clés

- **Pour les clusters**
  Chaque cluster possède des paramètres fixes propres à lui, tels que, la localisation virtuelle initiale, frontières ou limites de sa zone virtuelle, vitesse, angle et direction de mouvement virtuels.
  Tous les noeuds du cluster bougent de la même manière, ce qui implique le changement de leur localisation virtuelle. Tous les noeuds calculent la localisation virtuelle courrante à laquelle ils appliquent une fonction de hachage. Le résultat de cette fonction sera la clé du cluster.

- **Pour les noeuds**
  De même que pour les clusters, les noeuds possèdent aussi des paramètres

propres à chacun: localisation virtuelle initiale, limites de sa zone, vitesse et direction de mouvement. Tous les noeuds sont positionnés par rapport à l'origine virtuelle du cluster.
Après que chaque noeud ait effectué un mouvement virtuel dans sa zone limitée il calcule sa localisation virtuelle courrante à laquelle il applique une fonction de hachage pour produire sa clé.

Le **VDKM** *Virtual Dynamic Keying Module* est chargé de la génération de clés en appliquant une fonction de hachage entre la localisation virtuelle initiale et la localisation virtuelle courrante.

2-4-1-3 Mise à jour des clés

Que ce soit pour les clusters ou pour les noeuds, la mise à jour des clés est effectuée en fonction du changement de la localisation virtuelle. Càd si un noeud souhaite changer sa clé il effectue un mouvement virtuel, de ce fait il change sa localisation courrante à partir de laquelle il produit une nouvelle clé.

2.4.2 Energy Efficient key Management Scheme for Wireless Sensor Networks

Les deux auteurs de cet article Sughanthi et Sumathy ont proposé un système de gestion de clés économe en énergie. Ce système proposé se focalise sur l'établissement et la maintenance des paires de clés entre les noeuds voisins et la clé du réseau. Chaque noeud capteur génére la clé du réseau et la paire de clé en utilisant des fonctions polynômiales. La fonction polynômiale est identifiée par un ID.
Pour accroitre la sécurité du message dans le réseau lors de la phase d'initialisation, les noeuds utilisent leur clé individuelle. La station de base calcule les clés individuelles de tous les noeuds en utilisant les clés et les identifiants uniques qu'elle a stocké [30].

2-4-2-1 Type de clés utilisées

Chaque noeud du réseau dispose de trois clés:

- Une clé partagée avec la station de base.
- Une clé partagée avec chaque noeud voisin.
- Une clé partagée avec tous les noeuds du réseau.

2-4-2-2 Construction de l'arbre couvrant

Avant le déploiement, chaque noeud est équipé d'une pseudo-fonction aléatoire et d'une clé initiale. Cette foction et cette clé sont utilisées par les noeuds pour le calcul de leur propre clé individuelle qui est utilisée pour la communication initiale avec la station de base.

Le message Hello qui est utilisé pour construire l'arbre couvrant est aussi crypté avec la clé individuelle de chaque noeud.
La station de base diffuse le message Hello, les noeuds qui répondent à ce message deviennet ses fils, ces noeud diffusent à leur tour le message Hello pour les autres noeuds, ceux qui répondent deviennent ainsi leurs fils.
L'opération s'arrêtte lorsque les noeuds ne reçoivent plus de rṕonse au message Hello.

2-4-2-3 Etablissement de clés

- **Clé individuelle**
  Avant le déploiement, chaque noeud est préchargé d'une fonction pseudo-aléatoire (Rf) partagée par tous les noeuds du réseau, ainsi que d'une clé initiale. La fonction pseudo-aléatoire et la clé initiale sont utiliséespar chaque noeud dans le réseau pour calculer sa clé individuelle.

$$K_a = R_f(K_I, Id_a)$$

- **Paire de clés entre noeuds voisins**
  La station de base envoie l'identifiant de la fonction et un nombre aléatoire, le tout crypté avec la clé individuelle de chaque noeud.

$$E_{ka}(P_{fid}, R_n)$$

  Après le déploiement des noeuds capteurs, chaque noued communique avec son voisin à l'aide d'une paire de clé.

$$K_{ab} = P_{fid}(R_n, Id_a)$$

  La foction polynômiale prend le nombre aléatoire communiqué auparavant par la station de base ainsi que l'identifiant du noeud initiateur (le noeud A dans ce cas), et calcule la paire de clé. Après que cette dernière soit calculée, la

station de base envoie le message crypté avec cette clé suivi de l'identifiant du noeud A au noeud B, qui désormais dispose de tous les paramètres qui lui faut pour le calcul de la paire de clé qu'il utilisera pour décrypter le message de A.

- **Clé du groupe**.
  La clé du groupe (de tous le réseau) est préchargée dans chaque noeud. Le générateur de clé du groupe qui est présent dans chaque noeud capteur est utilisé pour la génération de la clé en question.
  La clé du groupe est clculée comme suit: $K_g = Pf_{id}(R_n, G_K)$ tel que K_g est la clé du groupe, Pf_{id} la fonction polynomiale, R_n un nombre aléatoire et G_K le générateur de la clé du groupe.
  Cette clé est utilisée pour la communication entre la station de base et les noeuds de manière directe.

2-4-2-4 Mise à jour des clés

Un délai est fixé pour le réetablissement des paires de clé ainsi que la clé du réseau. Lorsque ce délai est atteind la mise à jour des clés est effectuée. Cette procédure est réralisée en changeant uniquement le coefficient de la foction polynômiale, ainsi les clés calculées anterieurement sont annulées.

2.4.3 An Efficient and Hybrid Key Management for Heterogeneous Wireless sensor Networks

Dans cet article la méthode proposée par Zhang et Pengfei concerne les réseaux hétérogènes, constitués de noeuds ayant des capacités énergétiques différentes et équipés de dispositifs de detection variés qui collectent et transmettent des typologies de données différentes [31].

La méthode opte pour un modèle de réseau hiérarchique et hétérogène, de ce fait on distingue:

- **Station de base**: Dotée d'une grande capacité de calcul, les différentes informations récoltées sont traitées à son niveau, du fait qu'elle ait suffisamment d'énergie et que tous les noeuds lui font confiance.
- **High-end sensors**: Dotés d'une grande capacité énérgétique, d'une largeur de bande passante, d'espace de stockage et de capacité de calcul importants. Ils sont par défaut Cluster-Heads.
- **Low-end sensors**: Disposent d'une capacité énérgétique inferieure de celle des H-sensors.

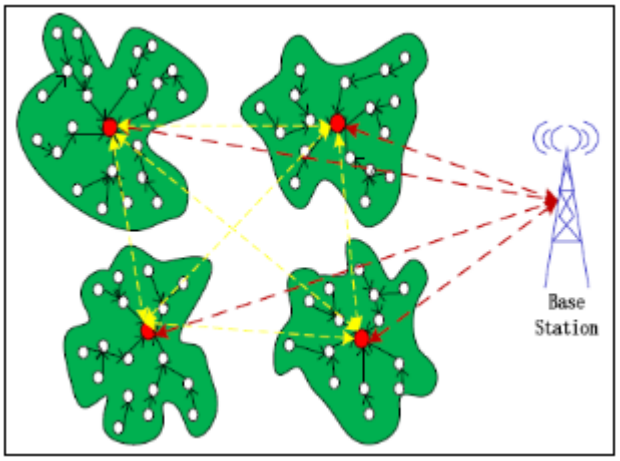

Figure 2.9: Hiérarchie de la méthode EHKM [31].

Tous les noeuds sont statiques et connaisent leurs propres coordonnées.

2-4-3-1 Prédistribution des clés

- Un serveur utilisant *ECC* (*Elliptic Curve Cryptography*) se charge de munir les noeuds des clés avant leur dempoiement. Les *H-sensors* étant munis de dispositifs inviolables, l'extraction des données est supposée impossible, de ce fait tous les *H-sensors* utilisent la même paire de clés.

- La station de base posséde sa propre paire de clés ainsi que la clé publique des clusters heads.

- Chaque H-sensor posséde un ID unique, la clé publique de la station de base, sa propre paire de clés, la liste des clés publiques des L-sensors ainsi qu'une fonction de hachage.

- Les L-sensors possèdent un ID unique, leur clé privée, la clé publique du cluster head et une fonction de hachage.

2-4-3-2 Génération des clés

La sécurisation de la communication entre les noeuds suit un système hybride, combinant des clés symétriques et des clés asymétriques.

La communication entre la station de base et les H-sensors se fait soit directement soit par l'intermédiaire d'autres H-sensors. La communication entre les deux entités se fait en utilisant les paires de clés générées avec *ECC* avant le déploiement

des noeuds.

Les H-sensors et les L-sensors utilisent l'algorithme de l'échange de clés de *Diffie-Hellman* pour établir une clé partagée entre les deux noeuds, cette clé est K_{HL}.

Les L-sensors communiquent entre eux en utilisant une clé de session obtenue en utilisant une fonction de hachage. Le cluster Head se charge de générer un nombre aléatoire **r** qu'il enverra après l'avoir chiffré avec les clés K_{HL} à tous les noeuds du cluster. Si deux noeuds **u** et **v** veulent établir une connexion, ils s'échangent leurs identifiants et calculent la clé partagée comme suit: **r** est supprimé à la fin de l'opération.
Si

$$id_u > id_v \text{ alors } K_{uv} = hash(r||id_u||id_v)$$

Sinon

$$k_{uv} = hash(r||id_v||id_u)$$

Quant à la clé du cluster elle est calculée ainsi: $k_0 = hash(r||id_{H_j})$

2-4-3-3 Mise à jour des clés

L'utilisation de clés symétriques pour l'établissement de liens sécurisés entre les L-sensors est due à la faible énergie qu'ils possèdent. En effet, l'utilisation de clés symétriques consomme moins d'énergie que celle des clés asymétriques, le niveau de sécurité octroyé est cependant plus bas. C'est pourquoi il est nécessaire de mettre à jour régulièrement les clés partagées entre les L-sensors.

Après un intervalle de temps T, les H-sensor génèrent un nouveau nombre aléatoire **r´**, qui est supprimé par la suite.

Si aucun noeud n'a pas été capturé durant la période T, le H-sensor chiffre **r´** avec la clé du cluster avant de le diffuser vers les autres noeuds du cluster. Une fois arrivé, les L-sensors utilisent K_0 pour retrouver **r´**.

$$k_{uv} = hash(r'||k_{uv})$$
$$k_0 = hash(r'||k_0)$$

Si un noeud a été capturé durant la période T, K_0 n'est plus fiable dans ce cas, le H-sensor génère **r´** et le chiffre avec les clés K_{HL} avant de l'envoyer vers les noeuds du cluster. Une fois arrivé, les L-sensors procèdent comme suit :

$$k_{uv} = hash(r'||k_{uv})$$
$$k_0 = hash(r'||id_{H_j})$$

2-4-3-4 Révocation des clés

Lorsqu'un L-sensor est capturé, le H-sensor du même cluster utilise un système de détection d'intrusions et identifie le noeud. Il envoie un message de révocation contenant l'identifiant du noeud capturé et signé en utilisant **ECDSA** (*Elliptic Curve Digital Signature Encryption*). A l'arrivée de ce message, les L-sensors vérifient s'ils sont en communication avec le noeud compromis, dans le cas affirmatif les clés de session partagées avec le noeud et avec le cluster sont révoquées.

2-4-3-5 Ajout de nouveaux noeuds

Si un noeud rejoint le réseau après avoir été chargé avec sa clé privée, la clé publique des H-sensors et une fonction de hachage, la station de base informe le H-sensor de l'arrivée d'un nouveau noeud. Le H-sensor lui envoie des informations sur le routage ainsi que le nombre **r**, qui sera utilisé par le nouveau noeud pour établir des clés de session avec les autres noeuds du cluster. Après la génération des clés le nombre **r** sera supprimé.

2.4.4 Large Scale Wireless Sensor Networks with Multi-Level Dynamic Key Management Scheme

Le système que propose l'auteur est un système de distribution de clés qui a pour rôle d'affecter des clés aux noeuds dans toute la zone du réseau de capteurs. Il supporte les deux modèles de réseau connus pour les WSNs: le modèle hiérarchique et le modèle distribué. Il a pour autorité de certification le **MCA** *Mobile Certification Authority* qui est également le centre de coordination du système [32].

2-4-4-1 Distribution des clés

Grâce à la structure hybride du système proposé, chaque noeud enregistre sa clé privée et la clé publique du **MCA**.
Pour que deux noeuds s'échangent des messages de manière sécurisée ils doivent d'abord s'échanger leurs identités. Un noeud peut établir une clé partagée avec un noeud voisin suivant le protocole suivant:
Supposons que le noeud **A** et le noeud **B** soient les deux noeuds communiquants.

- **Etape1**: Chacun des noeuds diffuse un message contenant son identifiant (ID_A) pour son voisin B et (ID_B) pour son voisin A .
- **Etape2**: Après que B ait reçu (ID_A), il doit obtenir la clé publique de A du MCA, cette requête est cryptée avec la clé publique du MCA.
- **Etape3**: Après que le MCA ait trouvé la clé demandée dans sa base de données, il l'envoie au noeud qui l'a demandé (dans ce cas le noeud B), cryptée avec la clé publique de B.
- **Etape4**: Le noeud B utilise la clé publique de A pour crypter le message qui contient son ID et un nombre aléatoire (Rn) pour identifier la transaction.
- **Etape5**: Le noeud A décrypte le message reçu de B ainsi que le nombre aléatoire, puis il selectionne une clé secrète K_{AB} et la retourner à B avec le (Rn), le tout crypté avec la clé publique de B.
- **Etape6**: Après ce processus, les deux parties vérifient chacune l'autre et établissent une paire de clés pour la communication entre elles.

2-4-4-2 Mise à jour des clés

Dans le système proposé, à part le méchanisme de mise à jour de clés périodique, un méchanisme de mise à jour basé sur le volume des informations peut être favorisé. Après que le flux de messages atteind un certain seuil, la mise à jour de clés est lancée. Les clusters heauds peuvent aussi déclencher l'opération de mise à jour. Si un cluster head détecte un noeud malicieux dans le réseau, il envoie un message de mise à jour pour tous les noeuds du cluster et ils mettent tous à jour leurs clés partagées.
Après qu'un noeud décide de mettre à jour une clé partagée, il recommence la phase de découverte de ses voisins comme décrit précédemment.

2-4-4-3 Ajout de nouveaux noeuds

Le système possède une structure réseau dynamique dans laquelle l'addition et la révocation de noeuds est possible. Les nouveaux noeuds capteurs déployés ont besoin de mettre en place des clés secrètes avec leurs voisins de manière efficace et autonome.

2.4.5 VEBEK: Virtual Energy-Based Encryption and Keying for Wireless Sensor Networks

VEBEK est un protocole de communication sécurisé où les informations sont codées en utilisant un protocole basé sur un code de permutation généré par RC4. La clé fourni au méchanisme de cryptage RC4 change dynamiquement en fonction de l'énergie résiduelle des noeuds capteurs. Ainsi une clé à la fois est utilisée pour un seul paquet et différentes clés sont utilisées pour les prochains paquets.
VEBEC est composé de trois modules: le module d'établissement de clés à base d'énérgie virtuelle, le module de cryptage et le module de transmission. il est aussi capable de détecter et filtrer de fausses informations injectées par des adversaires dans le réseau, il supporte ainsi deux modes opérationnels: *VEBEK-I* et *VEBEK-II* [33].

2-4-5-1 Architecture du système

- **Module d'établissement de clés à base d'énérgie virtuelle**
  Ce module implique la création de clés dynamiques, contrairement aux autres systèmes de génération de clés, il n'y a pas beaucoup de messages échangés pour l'établissement des clés. En effet le noeud capteur calcule ses clés en utilisant son énérgie virtuelle résiduelle, qui est une énérgie que possède chaque noeud lors de son déploiement dans le réseau. Chaque action effectuée par le noeud capteur vaut un certain coût en terme d'énérgie, d'où sa décrémentation.

- **Module de cryptage**
  Une fois la clé générée dans le module précédent, elle est fourni au module de cryptage. Le but de ce dernier est d'assurer la confidentialité, l'authentification et l'intégrité des informations détéctées. Les paquets dans VEBEK ont comme champs: ID, Type, DATA, l'algorithme de cryptage RC4 prend la clé et les champs des paquets pour produire un résultat.

- **Module de transmission**
  Ce module est résponsable de la transmission des paquets (rapports après la detection d'inforamtions), le paquet traverse les noeuds du réseau jusqu'à destination. Les noeuds intermediaires entre la source et la destination sont capables de vérifier l'authentcité et l'intégrité des paquets transitant en utilisant la valeur de la clé générée grâce à l'énérgie virtuelle du noeud expediteur.

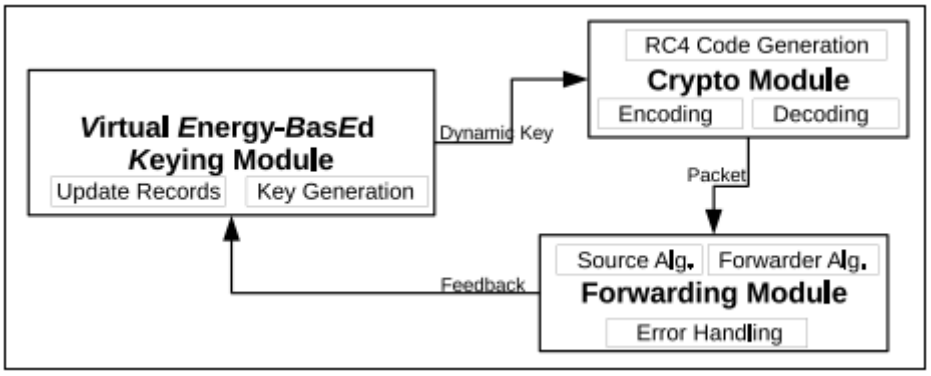

Figure 2.10: Structure modulaire de VEBEK [33].

2-4-5-2 Modes opérationnels de VEBEK

Le protocole *VEBEK* assure trois services de sécurité: authentification, intégrité et non répudiation. La notion fondamentale derrière l'assurance de ces services est un méchanisme de surveillance. Le méchanisme de surveillance exige à ce que les noeuds stockent une ou plusieurs données pour le calcul de clés dynamiques, utilisées par le noeud capteur pour décoder les paquets et détecter les paquets érronés.
VEBEK prend en considération ces besoins de sécurité, ainsi il s'appuie sur deux modes opérationnels: *VEBEK-I* et *VEBEK-II*.

- **VEBEK-I**
  Dans le mode opérationel *VEBEK-I* chaque noeud capteur surveille ses voisins se situant à un saut. Lorsqu'un événement est détecté le rapport établie est codé à l'aide de la clé générée aupparavant en fonction de l'énérgie virtuelle. Quand un paquet arrive d'un noeud voisin, il est décodé , son authenticité et son intégrité sont vérifiées, seulement les paquets ligitimes sont transmis à destination, tous les noeuds sur le passage effectuent le même traitement.
  VEBEK-I réduit la transmission, comme il est capable de détecter les paquets malicieux, néaumoins le taux de traitement est élevé à cause du codage et du décodage à chaque saut.

- **VEBEK-II**
  Dans ce deuxième mode opérationnel, chaque noeud surveille d'autres noeuds choisis aléatoirement. Lorsqu'un paquet quitte le noeud source, il passe par les noeuds surveillés uniquement par ce dernier, de ce fait VEBEK est une approche statique de filtrage. Si le noeud sur le passage est concerné, il décode le paquet comme dans VEBEK-I, sinon il est transmis au noeud prochain.
  Ce mode opérationnel a plus de transmissions à effectuer car les paquets

provenant des noeuds malicieux peuvent ne pas être detectés et atteindre la destination, par contre le traitement est réduit par rapport à VEBEK-I.

2-4-5-3 Mise à jour des clés

Dans *VEBEK*, une seule clé est utilisée à la fois pour un seul paquet, les prochains paquets seront codés et décodés par d'autres clés. En effet dans ce protocole les clés sont calculées à partir de l'énérgie virtuelle de laquelle disposent les noeuds capteurs. Après chaque action (envoi ou reception d'informations), cette énérgie est décrémentée, et c'est à partir de cette nouvelle valeur qu'une nouvelle clé est produite et calculée.

2.4.6 IBKM: An Efficient Identity-Based Key Management Scheme for Wireless Sensors Networks using the Bloom Filter

IBKM est un modèle de réseau hiérarchique ,Il dispose de trois types de dispositifs sans fil différents :

- La station de base (BS).
- Le cluster Head (CH).
- Le noeud de capteur (N).

IBKM se compose de trois phases : initialisation des paramètres, enregistrement du noeud et génération et partage de la clé secrète entre deux noeuds [36].

2-4-6-1 Phase d'initialisation des paramètres

La station de base (BS) sélectionne deux nombres premiers p, q et génère une courbe elliptique aléatoire E sur un champ fini Fp. Un point P sur la courbe E est sélectionné et utilisé comme générateur pour construire un groupe G1 additif, et $e : G1 \times G1 - - > Fp*$ est une application bilinéaire. $H1 : 0, 1 * - - > E(Fp)$ sont deux fonctions de hachage cryptographique.

- BS sélectionne un nombre aléatoire s et calcule P_{pub} = sPeG1 comme clé publique, BS diffuse les paramètres publics (G1,E(Fp),p,q,e,P,P_{pub},H1,H2).
- BS génère l'ID de chaque noeud et calcule la paire de clés publique et privée du noeud. Ensuite, les BS pré-charge dans le noeud.La clé publique est $Q_N =$

$H_1(ID_N)$, et la clé privée est $S_N = sQ_N$, où **N** est un noeud dans le réseau de capteur.

- BS génère l'ID du CH et calcule la paire de clés publique et privée du CH. Ensuite, BS stocke dans le CH, dans lequel la clé publique est $Q_{CH} = H_1(ID_{CH})$; la clé privée est $S_{CH} = sQ_{CH}$, où CH est le Cluster Head dans les réseaux de capteurs.

- BS tient une liste des identifiants de tous les noeuds et leurs paires de clés public-privée. BS conserve également les identifiants de tous CH et les clés publiques pour les prochaines étapes.

2-4-6-2 Phase d'enregistrement des noeuds

Dans cette phase, tous les noeuds de capteurs enregistrent les têtes de munitions et une clé de session est générée entre chaque noeud et son cluster head.

- Le CH diffuse un message qui contient sa propre identité et une clé publique à tous les noeuds capteurs voisins.

$$CH \xrightarrow{E_{S_{CH}}(ID_{CH} \| Q_{CH})} All\ Nodes$$

- Lors de la réception des messages du CH, chaque noeud capteur envoie son ID et la clé publique au CH avec lequel il veut se joindre.

$$All\ Nodes \xrightarrow{E_{Q_{CH}}(ID_N \| Q_N)} CH$$

- Après avoir reçu l'ID et la clé publique d'un noeud, le CH calcule la clé de session KC2

$$K_{C2} = e(S_{CH}, Q_N)$$

- Le noeud calcule la même clé de session avec le CH.

$$K_{C1} = e(S_N, Q_{CH})$$

- le CH génère un Bloom filter de tous les identifiants des noeuds et des clés publiques au sein de son groupe et envoie le Bloom filter chiffré par la clé de session générée auparavant à tous les noeuds du cluster. La figure montre la génération du Bloom filter.

$$CH \xrightarrow{E_{K_C}(Bloom\ Filter)} All\ Nodes$$

2-4-6-3 Partage de clés secrètes entre deux noeuds

- Le noeud capteur A choisit un nombre aléatoire r1 et diffuse un message qui contient son ID, la clé publique et un cachet de temps qui sont chiffrés par sa propre clé privée pour les noeuds voisins aprés enregistrement dans le CH.

$$A \xrightarrow{ID_A \| E_{S_A}(r_1 Q_A \| T)} Neighbor\ Nodes$$

- Lorsque le noeud voisin B reçoit le message, il vérifie l'authenticité de A en vérifiant si le haché du mappage de (IDA,QA) est contenu dans le Bloom filter obtenu à partir du CH. Une réponse négative signifie l'échec de l'authentification. Si l'authentification est adopté, B choisit son numéro aléatoire r2 et renvoie son ID, la clé publique et un horodatage qui seront chiffrés par sa propre clé privée. Puis, B calcule la clé de session KS1.

$$B \xrightarrow{ID_B \| E_{S_B}(r_2 Q_B \| T)} A$$
$$K_{S1} = e(\mathrm{r}_2 S_B, \mathrm{r}_1\, H_1(ID_A))$$

- A décrypte le message avec sa propre clé privée et obtient l'ID de B et la clé publique et vérifie l'authenticité de B en utilisant le bloom filter obtenu à partir de CH. Si B est authentifié, A calcule la clé de session KS2. Il est possible de prouver que KS1 = KS2 comme suit:

$$K_{S1} = e(r_2 S_B, r_1 H_1(ID_A)) = e(r_2 s Q_B, r_1 H_1(ID_A))$$
$$= e(r_2 s H_1(ID_B), r_1 H_1(ID_A)) = e(r_1 s H_1(ID_A), r_2 H_1(ID_B))$$
$$= e(r_1 s Q_A, r_2 H_1(ID_B)) = e(r_1 S_A, r_2 H_1(ID_B)) = K_{S2}$$

- Par la suite, les noeuds A et B peuvent communiquer les uns avec les autres en utilisant la clé de session partagée. La clé secrète partagée entre deux noeuds peut être décidée conformément à la Figure.

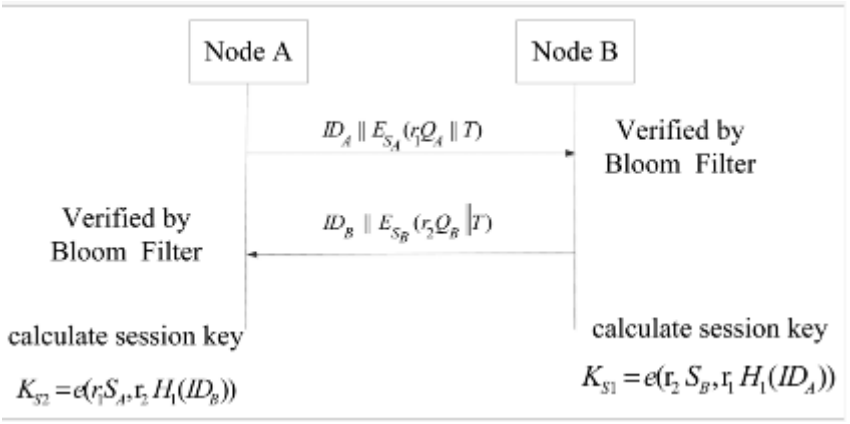

Figure 2.11: Génération de la clé secrète entre deux noeuds [36]

2.4.7 LEKM: A Low Energy Key Management Protocol for Wireless Sensor Networks

LEKM est un modèle hiérarchique, le réseau de capteur se compose d'un grand nombre de capteurs regroupés selon un centre d'intéret. Il y a un noeud de commande responsable de la mission du réseau et assumé pour être sûr et a la confiance de tous les noeuds du réseau. Le modèle présente également de superbe-noeuds, appelés passerelles ,en plus des noeuds de capteurs. Les passerelles ont des ressources énergétiques considérablement élevées comparées aux noeuds capteurs , sont équipées par des processeurs à grande performance et ont plus de mémoire. les passerelles divisent les capteurs en un ensemble de groupes (clusters) distincts. Ce protocole déterministe de gestion de clés est basé sur la méthode de pré-distribution. Il consiste à définir comment les clés sont distribuées, ajoutées, révoquées et renouvelées pendant la durée de vie du réseau de capteurs [37].

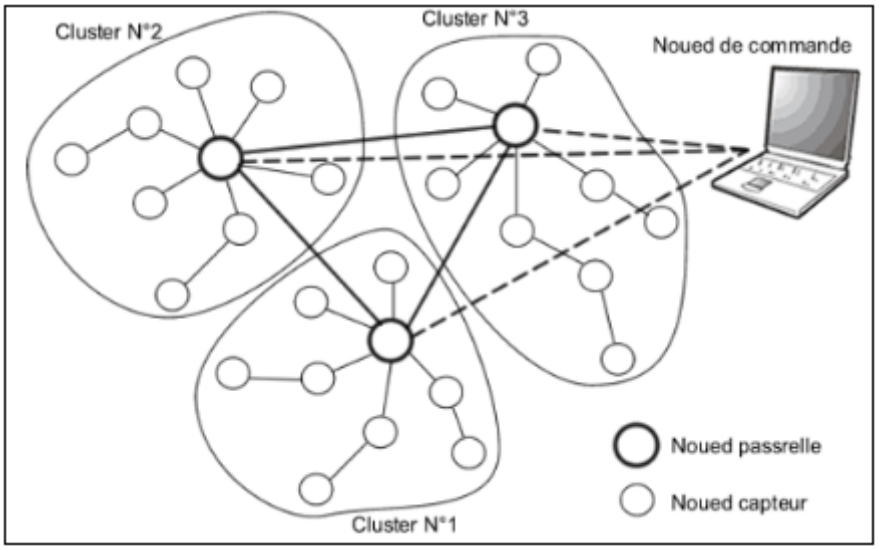

Figure 2.12: Réseau de capteurs hiérarchique avec milti-passerelles [37].

Notation	**Description**
C	Noeud de commande
G_i	Passerelle i
S_i	Noeud capteur i
G	Ensemble de toutes les passerelles dans le réseau
S	Ensemble de tous les capteurs
Id_i	Identité du noeud i
Nonce	Valeur aléatoire de nonce
sdada	Données sur la location et le niveau d'énergie de capteur
K_{AB}	Clé partagée entre A et B (A et B peuvent êtres des S_i, G_i ou C)
E(K,...)	Fonction symétrique de cryptage utilisant la clé K
$\|\|$	Opérateur de concaténation
G_h	Passerelle principale (utilisée pour la révocation)

Table 2.1: Terminologie utilisée dans le protocole *LEKM*.

Le tableau précédent illustre la terminologie qui sera utilisée dans les sous-sections suivantes.

2-4-7-1 types de clés utilisées

Chaque noeud de capteurs stocke deux clés secrètes.

- Une est partagée avec la passerelle
- Une est partagée avec le noeud de commande

Chaque passerelle stocke les clés qu'elle partage avec les capteurs de son groupe et la clé partagée avec le noeud de commande. Elle partage également une clé avec une autre passerelle dans le réseau. Le noeud de commande peut stocker toutes les clés secrètes dans le réseau. Le nombre total de clés stockées dans le noeud de commande est égal à :|S|+|G|.

2-4-7-2 Phase d'initialisation

Au moment du déploiement chaque passerelle est aléatoirement assignée |S|/|G| clés. Alors chaque passerelle forme des groupes en utilisant un algorithme de formation de groupe, et acquiert ensuite les clés des capteurs de son groupe à partir des autres passerelles. Les étapes du protocole d'initialisation sont les suivantes :

- Chaque noeud capteur est pré-chargé avec l'identificateur de la passerelle id_{Gi} qui contient sa clé partagée. Après le déploiement le noeud capteur diffuse un message Hello contenant l'identificateur id_{Gi}.

$$S- > G : id_{Gi}||id_{si}||E(K_{Si,Gi}, nonce||sdata) \text{ (i1)}$$

- Processus de clustering (construction des groupes) (i2)
- Chaque G_i identifie l'ensemble des capteurs id_i qui sont dans son groupe et les diffuse aux autres passerelles

$$G_i- > G : id_{Gi}||E(K_G, nonce||id_i) \text{ (i3)}$$

- Chaque G_js répond à Gi avec l'ensemble de ces clés $\{(K_{si,Gi}, id_{Si})\}_i$, où id_i est un sous ensemble de id_j

$$G_i- > G_j : E(K_{Gi,Gj}, nonce||(K_{Sk,Gj}, id_{Sk})i) \text{ (i4)}$$

- chaque capteur Si dans le groupe de Gi reçoit un message de G_i qui lui assigne la passerelle G_i

$$S_i- > G_i : id_{Gi}||E(K_{Si,Gj}, nonce||d_{Gi}||msg) \text{ (i5)}$$

2-4-7-3 Mise à jour des clés

Afin d'accomplir le renouvellement des clés des noeuds capteurs, le noeud de commande produit de nouvelles clés, et les pousse aux passerelles, comme dans le cas de la révocation cité plutard. L'intervalle de temps entre des renouvellements successifs peut dépendre du volume de trafic des données, de la puissance des primitifs cryptographiques et de la charge de traitement supplémentaire encourue aux passerelles.

2-4-7-4 Révocation des clés

- Si un noeud capteur est compromis, un mécanisme de détection d'intrusion informe le noeud de commande et il sera expulsé du cluster par la passerelle (La passerelle ne fera pas passer des messages ultérieurs du noeud compromis). Egalement ignoré par les autres noeuds du même cluster.

-Dans le cas de la révocation d'une passerelle (G_j), le noeud de commande choisit une passerelle G_h comme passerelle principale et l'ordonnera d'expulser G_j de G ainsi de distribuer ses noeuds sur les passerelles restantes. Par la suite, G_h distribue les clés des noeuds sur leurs passerelles respectives. Chaque passerelle contacte les noeuds affectés à son cluster. Elle transmet le message contenant son ID et aussi transmet le message à partir du noeud de commande au noeud crypté avec la clé partagée entre le noeud et le noeud de commande.

2-4-7-5 Ajout de nouveaux noeuds

1. Les nouveaux capteurs sont déployés arbitrairement (non pré-assignés à un groupe) et sont chargés avec deux clés comme les autres capteurs.

2. Le noeud de commande transmet la liste (Identificateur, Clé) à une passerelle G_h sélectionnée aléatoirement (pas à toutes les passerelles) pour réduire le risque de compromis. Celle-ci devient la passerelle qui partage les clés des nouveaux capteurs.

$$C- > G_{iE} : E(K_{Gi,C}, nonce \| \{K_{SK}, id_{Si})\}_i)$$

3. Ensuite toutes les étapes précédentes sont exécutés ((1), (2), (3), (4), (5)).(phases d'initialisation).

2.4.8 LEAP: Light Weight Extensible Authentication Protocol

LEAP est un protocole déterministe de gestion de clés pour les réseaux de capteurs sans fil. Le mécanisme de gestion de clés fourni par LEAP supporte le traitement "in-network processing" tout en limitant l'impact de sécurité d'un noeud compromis sur son voisinage immédiat dans le réseau.LEAP supoorte l'établissement de quatre types de clés pour chaque noeud capteur:**clé individuelle,clé paarpaire,clé de groupe et clé qlobale** [38].

2-4-8-1 Types de clés utilisées

- **Clé individuelle :**

  Chaque noeud possède une clé unique qu'il partage avec SB.Cette clé est employée pour sécuriser la communicationn entre le noeud et SB.

- **Clé par-paire :** chaque noeud partage une clé principale avec chacun de ses voisins immédiats.

- **Clé de groupe :** est une clé globalement partagée,elle est utilisée par SB pour chiffrer les messages et les envoyer aux membres du groupe.

- **Clé globale:** Elle est partagée par un noeud avec tous ses voisins,elle est principalement utilisée pour sécuriser les méssages diffusés.

2-4-8-2 Pré-distribution de la clé

Le contrôleur (SB) génère une clé initiale k_{IN} et charge chaque noeud avec cette clé. Chaque noeud u dérive une clé principale (Maser Key)k_u=$\mathbf{f}_{KIN}\mathbf{(u)}$, fk étant une fonction pseudo-aléatoire.

2-4-8-3 Découverte des voisins

Immédiatement après son déploiement, le noeud u essaye de découvrir ses voisins en diffusant un message Hello qui contient son id. Aussi, il initie un timer qui sera déclenché après le temps Tmin. Le noeud u attend un ACK de chacun de ses voisins v qui contient l'identificateur de v. L'ACK est authentifié en utilisant la clé principale f_{Kv}, qui est dérivée comme suit : k_v=f_{KIN}(v). Comme le noeud u a la clé k_{IN} , il peut aussi dériver Kv, ainsi il pourra vérifer l'identité de v.

u –> * : u.

v –> u : v,MAC(Kv,u|v).

2-4-8-4 Génération de la clé par-paire

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

2-4-8-5 Effacement des clés

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

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

Eschenauer et Gligor ont proposé un schéma de gestion de clés basé sur la probabilité de partager une clé entre les noeuds d'un graphe aléatoire. Il fournit des techniques pour la pré- distribution de clés, la découverte de la clé partagée, l'établissement de chemin de clé ainsi que la révocation de clés. L'idée maîtresse de ce schéma est de distribuer aléatoirement un certain nombre de clés, issues d'un ensemble fini à chaque noeud du réseau avant son déploiement. Deux noeuds quelconques seront en mesure de s'échanger des messages sécurisés s'ils possèdent une clé commune [39].

2-4-9-1 Phase de pré-distribution de clés

Un grand ensemble P de clés est généré (entre $2^{17} et 2^{20}$ clés). Pour chaque noeud, m clés sont choisies au hasard à partir de l'ensemble P ($P = (\{(kid1, key1), (kid2, key2), ...\})$. Ces m clés sont stockées dans la mémoire du noeud et forment le trousseau de clés du noeud. 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.

2-4-9-2 Phase de découverte de clés partagées

Les noeuds découvrent leurs voisins et plus particulièrement ceux avec qui ils sont en mesure de communiquer de façon sécurisée, car ils possèdent une clé identique dans leur trousseau de clés respectifs. Le protocole est de diffuser la liste des identités

kid_i des clés possédées, la clé partagée devient la clé de session de lien entre les deux noeuds.

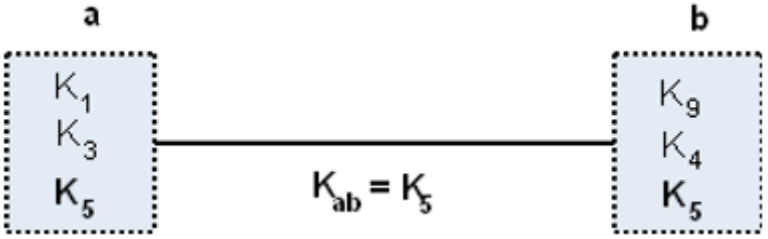

Figure 2.13: Schéma probabiliste de base de gestion de clés [38].

2-4-9-3 Phase d'établissement de chemin de clé

Après la phase de découverte des clés partagées, le réseau devient un graphe connecté formé de quelques liens sécurisés. Les noeuds 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.

2-4-9-4 Mise à jour des clés

Elle est équivalente à une révocation de clé effectuée par le noeud lui-même. Après la suppression de clé révoquée, le noeud affecté lance une phase de découverte de clé partagée et probablement une phase d'établissement de chemin de clé pour rétablir le lien cassé.

2-4-9-5 Révocation des clés

La révocation d'un noeud compromis se fait par l'élimination de leur trousseau de clés. Pour cela, un noeud contrôleur (qui a une grande connectivité et peut être mobile) annonce un message simple de révocation contenant une liste signée de m identificateurs des clés (kid_i) pour que ces clés soient retirées des trousseaux de clés des autres noeuds.
La liste des identités est signée par une clé de signature K_e générée par le noeud contrôleur et envoyée en unicast à chaque noeud i en la chiffrant avec la clé K_{ci}(la clé K_{ci} est partagée entre le contrôleur et le ième noeud pendant la phase de pré-distribution de clés). Quelques liens disparetront à cause de la suppression de clés du noeud compromis ce qui nécessite une reconfiguration de ces liens (par la découverte de clés partagées ou l'établissement de chemin de clé).

2.4.10 LPKM: Light Weight Polynomial-Based Key Management Protocol for Distributed Wireless Sensor Networks

LPKM est un système de gestion de clés pour les RCSFs distribués. LPKM permet aux noeuds capteurs de former un groupe et à établir une clé partagée. Trois types de clés peuvent être établis pour chaque noeud capteur. LPKM comprend également un protocole d'authentification de diffusion locale probabiliste qui prend en charge l'authentification de la source grâce à la collaboration entre les noeuds voisins [40].

2-4-10-1 Types de clés utilisées

Trois types de clés peuvent être établies pour chaque noeud capteur

- Une clé par-paire partagée avec un noeud voisin à un saut.
- Une clé par-paire partagée avec un noeud non voisin à plusieurs sauts.
- Une clé de cluster partagée avec les tous les noeuds du même cluster.
- Une clé de groupe partagée par l'ensemble du réseau.

2-4-10-2 Phase d'initialisation

Le **KDC** *Key Distribution Center*, étant le centre de distribution de clés effectue les étapes suivantes pour initialiser un réseau de capteurs.

1. Le KDC génère les paramètres du système (p, m, h, b, f (x)). Bien que les paramètres (p, m, h) sont connues publiquement, les paramètres (b, f (x)) sont des informations secrètes.
2. Pour un noeud capteur avec un identifiant i unique, le KDC calcule une part de f(x), qui est un (k + 1) moins le vecteur dimensionnelle, $v_i = (by_i a_0, by_i a_1, ..., by_i a_k)$.
3. Le KDC recharge les paramètres publics (p, m, h) et la part de f(x) qui est v_i dans le noeud capteur i.

2-4-10-3 Génération de clés

1. **Génération de la clé par-paire**
   Dans ce RCSFs, une clé par-paire est utilisée pour sécuriser des communications one-to-one , qui sont soit entre noeuds capteurs et leurs voisins immédiats (à savoir, les voisins à un saut) ou entre deux noeuds capteurs qui sont

à plusieurs sauts de distance. Nous discutons de ces deux cas séparément ci-dessous.

-Clé par-paire pour les noeuds voisins

Avant que les noeuds capteurs commencent l'établissement de la clé par-paire avec leurs voisins immédiats, ils ont besoin d'exécuter une phase de découverte de voisinage. Le noeud i diffuse localement un message BONJOUR contenant son identité. Chaque noeud j recevant le message de BONJOUR répond par un message d'accusé de réception qui inclut l'identité j. Le message ACK du noeud j peut être authentifié par la clé par-paire $K_{i,j}$ qui est calculée comme suit:

$$K_{i,j} = f_i(j)f_i(0)^{m-1+\delta} = f_j(i)f_j(0)^{m-1+\delta} = b^{m-1}f(0)^{m-1+\delta}f(i)f(j).$$

-Clé par-paire pour les noeuds non-voisins

La mise en place des clés par-paire pour les noeuds non-voisins est souhaitable dans des applications telles que l'agrégation de données, où un noeud capteur doit se rendre compte de ses lectures à un noeud d'agrégation, qui est habituellement à plusieurs sauts de distance, de manière sécurisée. Une clé par-paire multi-sauts entre un noeud capteur **i** et un noeud d'agrégation **a** peut être suffisamment calculée comme suit:

$$K_{i,a} = f_i(a)f_i(0)^{m-1+\delta} = f_a(i)f_a(0)^{m-1+\delta} = b^{m-1}f(0)^{m-1+\delta}f(i)f(a).$$

2. **Génération de la clé de cluster**

   La clé cluster est utile pour la sécurisation des messages de diffusion transmis par un noeud capteur à ses voisins. En supposant qu'un noeud i et ses voisins immédiats sont désignés par un ensemble $I = \{i_1, i_2, ..., i_t\}$, la clé de cluster $K_{i,i_1,i_2,...,i_t}$ peut être calculée comme suit:

$$K_{i,i_1,i_2,...,i_t} = \prod_{j\in I} f_i(j)f_i(0)^{m-t+\delta} = b^{m-1}f(0)^{m-t+\delta}\prod_{j\in I\cup\{i\}} f(j)$$

3. **Génération de la clé de groupe**

   Dans les RCSFs, une clé de groupe est utile dans les applications où une station de base ou un noeud puits mobile diffuse des requêtes (commandes) à une certaine zone ou même l'ensemble du réseau. Même si une clé de groupe partagée par une station de base et tous les noeuds du réseau peut être pré-calculée et pré-chargée dans chaque noeud , une clé de groupe partagée par un noeud puits mobile et un petit groupe de noeuds peut être générée. L'établissement d'une

clé de groupe entre un noeud puits mobile g et un groupe de noeuds capteurs { g1 , g2 , ... , g t } suit la même procédure que l'établissement d'une clé de cluster. Habituellement, le noeud puits mobile g diffuse d'abord un message GROUP FORMATION contenant les identités du groupe cible $g_1, g_2, ..., g_t$ et un MAC calculé avec la clé de groupe $K_{g_1,g_2,...,g_t}$. Après que le noeud puits mobile ait reccu des messages GROUP-DONE de tous les membres du groupe, la clé de groupe est établie et prête à l'emploi .

2-4-10-5 Mise à jour des clés

Dans ce système, la partie polynômiale pour un noeud i est de la forme $(by_i a_0, (by_i a_1, ..., (by_i a_k)$, où b appartenant à Fp est un masque aléatoire. Dans le but de mettre à jour les partie polynômiales, un générateur de nombres pseudo aléatoire sécurisé à poids léger pour générer de nouveaux masques aléatoires périodiquement peut être utilisé.

2-4-5-5 Révocation des clés

Soit $K_{g_1,g_2,...,g_t}$ la clé partagée entre le noeud puits mobile g et un groupe de noeuds capteurs indexés par un ensemble $G = \{g_1, g_2, ..., g_t\}$, le résultat si-après est obtenu:

$$K_{g_1,g_2,...,g_t} = \prod_{j\in I} f_g(j) f_g(0)^{m-(t+1)+\delta} = b^{m-1} f(0)^{m-(t+1)+\delta} \prod_{j\in G\cup\{g\}} f(j)$$

Supposons que les l premiers noeuds capteurs ont été compromis par un adversaire et ont été détecté par tous les autres membres du groupe à un instant donné. Pour sécuriser les communications ultérieures entre le noeud puits mobile g et le reste des noeuds capteurs une nouvelle clé de groupe $g_l, g_{l+1}, ..., g_t$ devrait être établie.

$K_{g,g_{l+1},...,g_t} = K_{g,g_{l+1},...,g_t} \cdot (\prod_{j\in\{g1,...,gl\}})^{-1} \cdot f_g(0)^l = b^{m-1} f(0)^{m-(t-l+1)+\delta} (\prod_{j\in\{g,gl+1...,gt\}}) f_j$

le reste des noeuds capteurs effectuent des calculs similaires pour générer la nouvelle clé de groupe. De cette façon , les noeuds compromis sont expulsés du groupe automatiquement.

2.5 Etude comparative des protocoles

2.5.1 Présentation des paramétres utilisés

Pour l'évaluation des solutions de gestion de clés proposés pour les RCSF, les paramétres suivants sont utilisés:

- **Coût de communication** détermine le nombre de messages échangés dans le réseau dans le but d'établir, compromettre et de mettre à jour les clés.
- **Coût de calcul** est mesuré en terme de nombre de cryptage et de décryptage nécessaires pour le changement des clés lorsqu'un noeud est compromis ou lorsq'un nouveau noeud rejoint le réseau.
- **Espace mémoire** occupé par les informations dont chaque noeud a besoin.
- **Prédistrubution de clés** est une méthode pour la distribution de clés aux noeuds du RCSF dont la topologie est inconnue avant le déploiement.

2.5.2 Tableau comparatif des protocoles

Le tableau ci-après résume une comparaison de propriétés entre les différents protocoles de gestion de clés que nous avons étudié préalablement.

Protocoles	**Cout de communication**	**Espace mémoire**	**Cout de calcul**	**Pré-distribution de clés**
EHKM	Moyen	Elevé	Moyen	Oui
IBKM	Elevé	Moyen	Elevé	Non
Energy Efficient Key Management Scheme	Moyen	Faible	Faible	Oui
LEKM	Moyen	Moyen	Moyen	Oui
Large Scale WSN with Mlti-level Dynamic KMS	Elevé	Elevé	Moyen	Non
LEAP	Moyen	Moyen	Moyen	Oui
VEBEK	Faible	Faible	Elevé	Non
L.E et D.G	moyen	Elevé	Faible	Oui
VLKM	Faible	Faible	Elevé	Non
LPKM	Moyen	Moyen	Elevé	Non

Table 2.2: Tableau comparatif des protocoles étudiés.

2.5.3 Discussion

D'après l'étude faite, et qui est résumé dans le tableau présenté préalablement, nous avons constaté que les protocoles de gestion de clés basés sur la méthode de pré-distribution sont les plus appropriés aux RCSFs et ceci pour leurs faible coût. Ils ont un coût de communication moyen et sont moins couteux en espace mémoire, à l'éxception des protocoles qui stock un grand nombre de clés tel que EHKM et LE/DG. Par contre, les protocoles sans pré-distribution ont un coût de communication élevé et moyennement couteux en espace mémoire, à l'éxception de VLKM et VEBEK oû les clés sont calculées à partir des paramètres virtuels tel que l'énergie résiduelle ou la localisation.

2.6 Conclusion

Nous avons étudié quelques protocoles qui ont répondu aux problèmes de gestion de clés dans les RCSF. Ces protocoles fournissent le service de base pour sécuriser les différentes communications.
Dans ce chapitre, nous avons repris une classification déjà proposée concernent les

systèmes de gestion de clés dans les RCSF, une description du fonctionnement de quelques protocoles proposés dans la littérature est également établie, enfin une comparaison de certains critères et une discussion viennent clôturer le chapitre.
Dans ce qui suit, nous allons présenter une amélioration du protocole "Energy Efficient Key Management Scheme For Wireless Sensor Networks", ainsi que sa simulation sur *Matlab*.

3

Proposition et simulation

3.1 introduction

Etant donnés les différents domaines sensibles auxquels peuvent s'appliquer les RCSF, la sécurisation des applications développées pour ces derniers devient un élément essentiel et indispensable, tout en tenant compte de l'économisation de l'énergie. Les applications des RCSF sont basées sur la communication, qui nécessite des protocoles de gestion de clés.

Après avoir abordé quelques protocoles et solutions de gestion de clés proposés pour les RCSF, nous avons opté pour le protocole intitulé "Energy Efficient Key Management Scheme For Wireless Sensor Networks" et proposé une amélioration dans le but de réduire le coût de communication et de renforcer la sécurité au niveau de l'établissement de la paire de clé entre deux noeuds voisins. Ce protocole est comparé au protocole "Large Scale Wireless sensor network with multi-level dynamic Key managment Scheme". La comparaison est faite à travers des analyses de sécurité et des résultats de simulation. Pour cela, nous les avons implémenté en utilisant l'environnement *MATLAB*.

3.2 Solution proposée

3.2.1 Motivation du choix du protocole

La gestion des clés dans les réseaux de capteurs sans fil est l'un des aspects assurant la sécurité de la communication des messages transitant dans le réseau. Ce pendant, les travaux portant sur le problème d'énérgie dans ce domaine ont essentiellement pris le dessus. Toutefois, trois critères sont principalement pris en considération: le coût de la communication, le coût de calcul ainsi que l'espace mémoir nécessaire pour le stockage d'informations au niveau de chaque noeud capteur.

Après l'étude de certains protocoles existants dans la littérature, nous avons décidé d'aporter une amélioration sur le protocole intitulé "Energy Efficient Key Management Scheme For Wireless Sensor Networks". Les raisons pour lesquelles on en a opté pour ce protocole sont par rapport au préchargement des clés initiales ainsi que la clé du réseau dans les noeuds capteurs avant leur déploiement, ce qui épargne leur calcul plutard, de plus les noeuds n'ont pas besoin d'un grand espace mémoire pour stocker les clés après leur déploiement, en ce qui concerne la communication, le nombre de messages communiqués est moyen par rapport à ceux qui s'échangent un nombre considérable de messages pour l'établissement des paires de clés entre voisins par exemple comme dans le protocole "Large Scale Wireless sensor network with multi-level dynamic Key managment Scheme", ou aux protocoles qui s'échangent un nombre hyper réduit de messages comme dans les deux protocoles VLKM et VEBEK.

3.2.2 Rappel du fonctionnement du protocole

On précise que la partie du protocole sur laquelle la proposition est faite concerne celle de l'établissement de la paire de clé entre deux noeuds voisins.
On concidère que les deux noeuds **A** et **B** sont les noeuds souhaitant établir la paire de clé K_{ab}, tel que **A** représente le noeud initiateur de la communication. Le principe de cette partie est comme suit:

- **Etape1:** La station de base transmet l'identifiant de la fonction polynômiale Pf_{id} et un nombre aléatoire R_n pour chaque noeud i du réseau, cryptés avec la clé individuelle de chaque noeud.

$$E_{ki}(Pf_{id}, R_n)$$

- **Etape2:** Au niveau du noeud initiateur **A**, la foction polynômiale prend le nombre aléatoire ainsi que l'identifiant de **A** comme entrées pour calculer la paire de clé.

$$K_{ab} = P_{fid}(R_n, Id_a)$$

- **Etape3:** Après le calcul de la paire de clé, la station de base transmet le message ***M*** provenant du noeud initiateur (**A**), crypté avec la paire de clé calculée, le tout concaténé à l'identifiant de **A**.

$$K_{ab}(M) \| Id_A$$

- **Etape4:** Le second noeud **B**, disposant de tous les paramètres nécessaires, calcule la paire de clé de la même manière que le noeud **A** , puis décrypte le message.

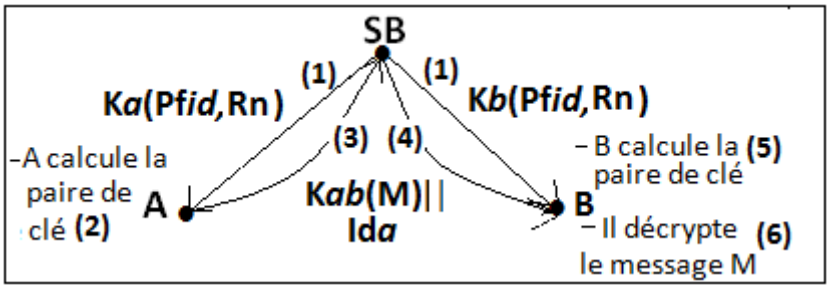

Figure 3.1: Shéma illustratif de l'établissement de la paire de clé.

3.2.3 Idée de base

À partir des étapes précédentes ainsi que de la figure 3-1, on remarque que la station de base se charge de transmettre l'identifiant de la fonction polinômiale et le nombre aléatoire à tous les noeuds du réseau, ce qui veut dire qu'ils disposent tous du même nombre aléatoire.
On rappelle aussi que la paire de clé se calcule en utilisant les deux paramètres précédents ainsi que l'identifiant du noeud initiateur de la communication, ce qui induit au partage de la même paire de clé entre l'initiateur et tous ses voisins. De plus la station de base communique trois messages pour l'établissement de la paire de clé.
Dans le but de minimiser le coût de communication et de renforcer la sécurité, nous

avons pensé à réduire le nombre et la taille de messages échangés et de placer un générateur de nombres aléatoires au niveau de chaque noeud capteur. Ceci sera détaillé dans la partie qui suit.

3.2.4 Hypothèses

L'amélioration que nous avons proposé pour ce protocole se repose sur les hypothèses suivantes:

- Chaque noeud capteur i est préchargé d'une clé individuelle K_i.
- Une clé de groupe K_g partagée par tous les noeuds du réseau est préchargée dans chaque noeud avant leur déploiement.
- Cette clé est utilisée lorsque la station de base distribue un message sécurisé (une instruction confidentielle par exemple), et dans notre cas, cette clé de groupe est également utilisée lors de la transmission des paramètres nécessaires, voire le nombre aléatoire et l'identifiant du noeud initiateur de la communication, qui serviront au second noeud de calculer de la paire de clé.
- La paire de clé K_{ab} calculée entre deux noeuds voisins **A** et **B** est utilisée pour sécuriser les messages échangés entre eux.
- Chaque noeud du réseau dispose d'un générateur de nombres aléatoires.

3.2.5 Description de la proposition

Étant donnés deux noeuds **A** et **B** souhaitant établir une paire de clé pour pouvoir communiquer en toute sécurité. **A** étant le noeud initiateur de la communication. Les étapes suivantes montrent

- La station de base transmet l'identifiant de la fonction polynômiale Pf_{id} au noeud **A** et au noeud **B** , crypté avec la clé individuelle de chacun des noeuds K_a et K_b respectivement.

$$K_a(Pf_{id}),\ K_b(Pf_{id})$$

- **A** génère un nombre aléatoire R_{n1}. La fonction polynômiale prend le nombre aléatoire ainsi que l'identifiant de **A** comme entrées pour calculer la paire de clé.
  L'identifiant de la foction polynômiale est utilisé pour indiquer quelle fonction polynômiale sera choisie parmi le groupe des autres fonctions.

- Après que le noeud **A** ait calculé la paire de clé, il envoie le nombre aléatoire qu'il a génèré ainsi que son identifiant au noeud **B**, le tout crypté avec la clé du groupe K_g, ce message sera concaténé au message **M** que **A** souhaite transmettre à son voisin **B**, crypté avec la paire de clé K_{ab} qu'il vient de calculer.

$$K_g(R_{n1}, Id_a) \| K_{ab}(M)$$

- Le noeud **B** calcule la paire de clé K_{ab} puisqu'il dispose de toutes les informations essentielles, et enfin décrypte le message **M** envoyé par **A**.

on rappelle que chaque noeud calcule la paire de clé en connaissant l'identifiant du noeud initiateur.

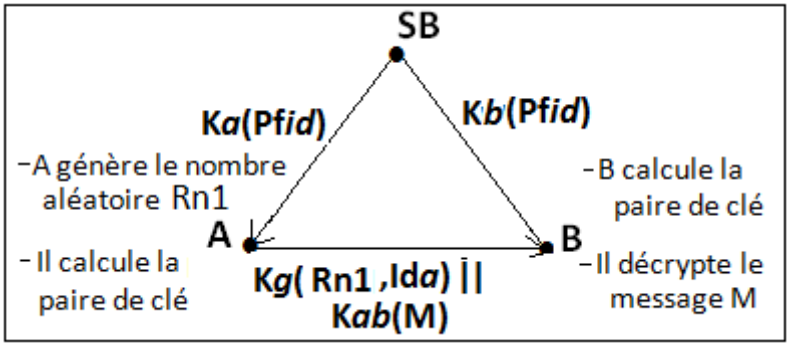

Figure 3.2: Shéma illustratif de la solution proposée.

3.2.6 Analyse théorique de la proposition

Dans la proposition faite on peut extraire les améliorations suivantes:

- La station de base transmet que l'identifiant de la fonction polynômiale pour le noeud initiateur de la communication ainsi que pour le noeud communiquant, ce qui par conséquent réduit la taille du message à transmettre.

- Le noeud initiateur génère un nouveau nombre aléatoire à chaque fois qu'il souhaite établir une paire de clé avec un de ses voisins, ce qui fait que le nombre aléatoire est différent à chaque fois, d'où les paires de clés qu'il établie sont toutes différentes entre lui et ses voisins.

- C'est le noeud initiateur de la communication qui transmet à son voisin les paramètres nécessaires (R_{n1} et Id_a) pour le calcul de la clé au lieu que ce soit la station de base qui le fasse. La station de base étant à une distance plus grande pourra consommer plus d'énérgie, contrairement aux noeuds voisins qui ne sont pas très loin.

3.3 Simulation

3.3.1 Présentation de l'environnement MATLAB

MATLAB est une abréviation de *Matrix LABoratory*. Écrit à l'origine, en Fortran, par C. Moler. *MATLAB* est un environnement puissant, complet et facile à utiliser, convivial aux calcul numérique et de visualisation graphique. Il apporte aux ingénieurs, chercheurs et à tout scientifique un système interactif intégrant centaines fonctions mathématiques et d'analyse numérique (calcul matriciel , le MAT de Matlab, traitement de signal, traitement d'images, visualisations graphiques, etc.).C'est un environnement performant, ouvert et programmable qui permet de remarquables gains de productivité et de créativité. *MATLAB* permet le travail interactif soit en mode commande, soit en mode programmation ; tout en ayant toujours la possibilité de faire des visualisations graphiques. Considéré comme un des meilleurs langages de programmations (C ou Fortran), *MATLAB* possède les particularités suivantes par rapport à ces langages [41] :

- La programmation facile.
- La continuité parmi les valeurs entières, réelles et complexes.
- La gamme étendue des nombres et leurs précisions.
- La bibliothèque mathématique très compréhensive.
- L'outil graphique qui inclus les fonctions d'interface graphique et les utilitaires.
- La possibilité de liaison avec les autres langages classiques de programmations (C ou Fortran).

3.3.2 Environnement de simulation

Notre modèle d'experimentation est établie sur 100 noeuds capteurs, dépoyés d'une manière aléatoire sur une surface carrée de $100m^2$. Nous supposons que les paramètres qui serviront au calcul de la paire de clé sont échangés après chaque

intervalle de temps (itération).
Les paramètres de notre simulation sont résumé dans le tableau ci-après:

Paramètres	**Valeur**
Nombre de noeuds	100
Energie initiale de chaque noeud	0.5 Joul
Energie initiale du réseau	50 Joul
Valeur d'énergie de transmission	50*0.000000001
Valeur d'énergie de reception	50*0.000000001
Valeur d'énergie d'amplification	100*0.000000000001
Valeur d'agrégation de données	5*0.000000001
Nombre d'itérations	1000

Table 3.1: Paramètres de simulation

3.3.3 Résultats de simulation

Durant chaque intervalle de temps (itération), nous avons mesuré l'énergie moyenne restante dans chaque noeud capteur, nous avons également mesuré l'énergie moyenne consommée dans tous le réseau, ainsi que le nombre de noeuds morts.
L'énergie initiale de chaque noeud capteur est de 0.5 Joul. A la fin de la simulation on aura trois graphes représentant l'énergie moyenne restante dans chaque noeud, lénergie moyenne consommée dans tous le réseau, le nombre de noeuds morts, le tous en fonction du nombre d'itérations.
La figure 3.3 illustre un réseau de 100 noeuds capteurs déployés aléatoirement dans une surface de 50*50 $metres^2$.

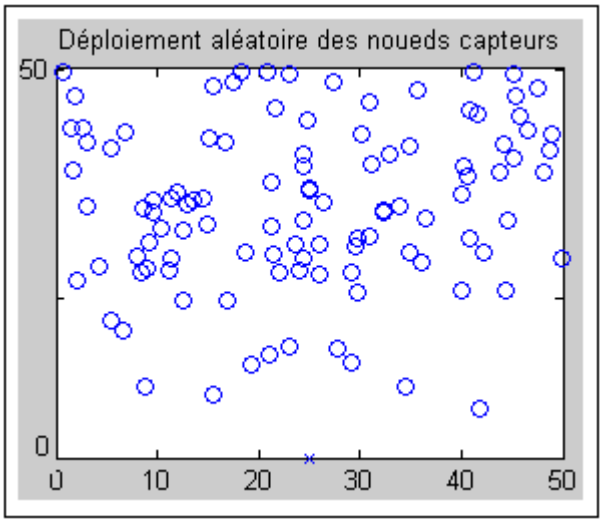

Figure 3.3: Déploiement aléatoire des noeuds capteurs.

3.3.4 Analyse et évaluation des performances de notre proposition

Afin d'évaluer les performances de la proposition faite, nous nous sommes basés sur la métrique d'énergie moyenne dans le réseau (restante dans chaque noeud et consommée dans tous le réseau) étant le facteur déterminant de la durée de vie du réseau capteur.
Nous avons comparé notre proposition à deux autres protocoles; le protocole initial [30] et le protocole multi-niveau dynamique de O.K.Sahingoz [32]. dans ces trois protocoles, l'élément déterminant l'énergie consommée du réseau est la taille des messages échangés entre les noeuds.

- **Consommation d'énergie**
  La ressource énérgétique détermine la durée de vie du réseau et doit être soigneusement prise en compte dans la conception de n'importe quel réseau de capteurs sans fil.
  Après la simulation on a obtenu les résultats suivants:

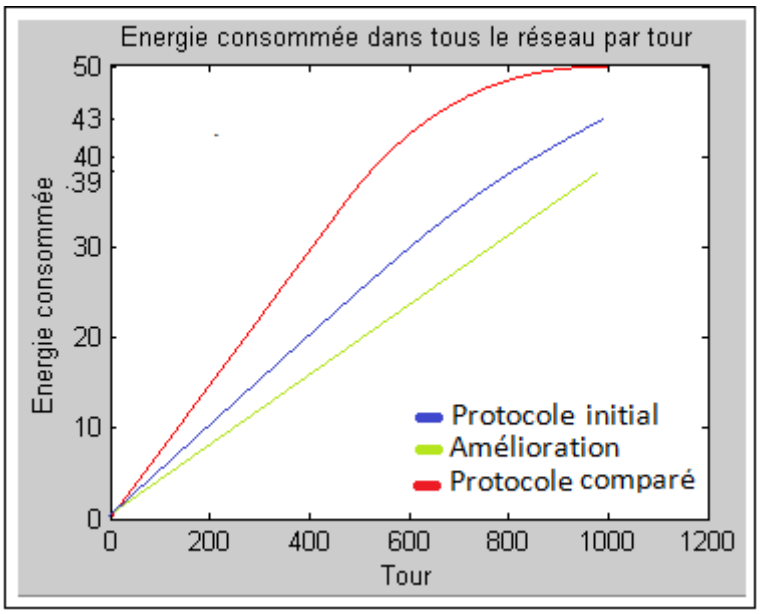

Figure 3.4: Consommmation d'énergie moyenne dans tous le réseau de notre approche et des deux autres protocoles.

La figure 3.4 montre l'énérgie moyenne consommée dans tous le réseau dans le cas du protocole initial "Energy Efficient Key Management Scheme For Wireless Sensor Networks" [30], de notre approche et du protocole comparé "Large Scale Wireless sensor network with multi-level dynamic Key managment Scheme" [32], respectivement.
On remarque que la consommation d'énergie est plus petite dans notre proposition par rapport aux deux autres protocoles. La simulation illustre qu'au bout de 1000 itérations la consommation d'énergie a diminué de 8% dans notre approche par rapport au protocole initiale (de 43 Joul à 39 Joul), tandis qu'elle est complétement consommée à la 950ème itération dans le protocole [32]. Ceci est dû à la taille réduite des messages.

- **Energie restante**

  La figure 3.5 représente l'évaluation de l'énergie moyenne restante dans chaque noeud du réseau en fonction du nombre d'itérations, ceci pour les trois protocoles; l'initial "Energy Efficient Key Management Scheme For Wireless Sensor Networks" [30], notre approche et celui auquel on a comparé "Large Scale Wireless sensor network with multi-level dynamic Key managment Scheme" [32] respectivement.

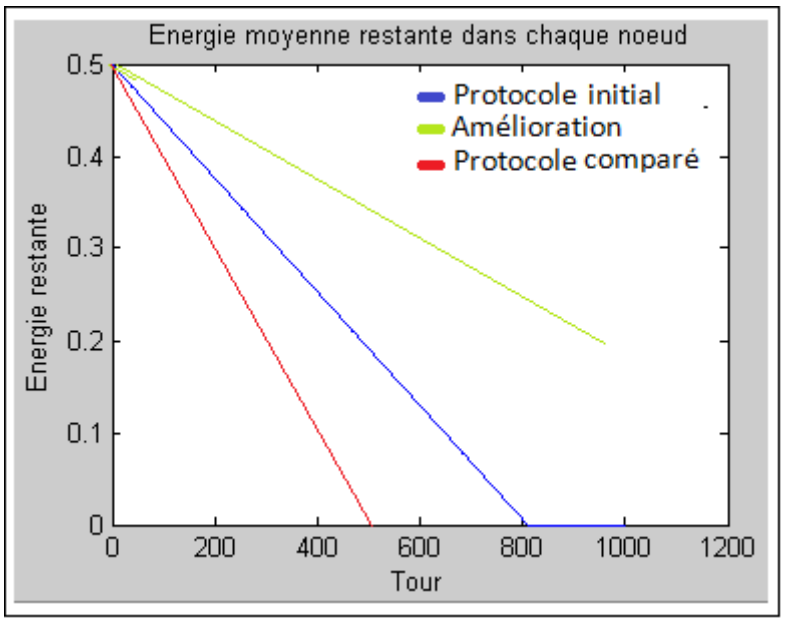

Figure 3.5: Energie moyenne restante dans chaque noeud de notre approche et des deux autres protocoles.

D'après les résultats de la figure, nous constatons que l'énergie moyenne restante dans les noeuds dans le cas de notre approche est prolongée d'avantage comparant au protocole initial [30] et au comparé [32] ce qui permet au noeud de rester vivant plus longtemps et cela implique une durée de vie plus longue du réseau. Comme illustré sur les graphes, l'énergie moyenne restante est à 0.2 Joul à la 1000ème itération dans notre approche, à 0 dans le protocole [30] et [32] à la 801ème et à la 500ème itération respectivement .

- **Nombre de noeuds morts**

  Tant qu'il y a des noeuds capteurs qui disposent d'une énergie résiduelle, le réseau reste en activité, une fois lénergie de tous les noeuds est épuisée le réseau de capteurs devient sans utilité.

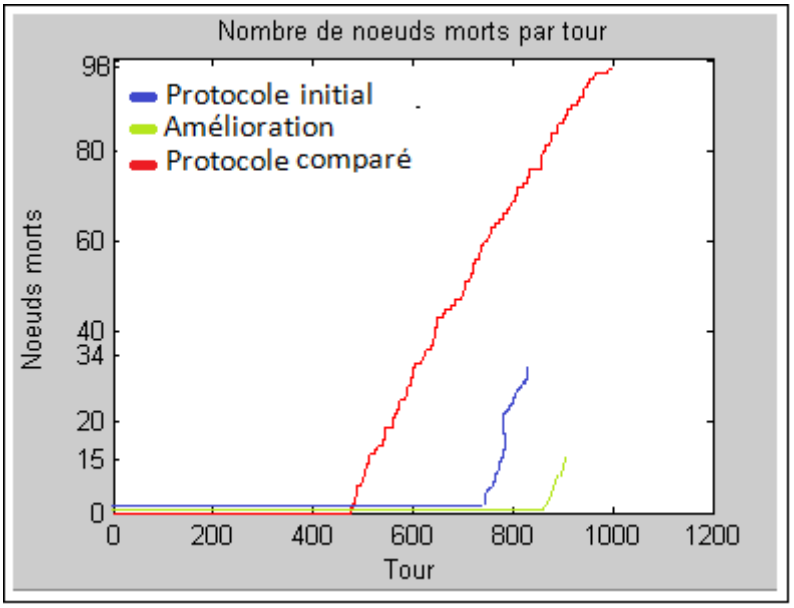

Figure 3.6: Nombre de noeuds morts des trois approches.

La figure 3.6 illustre le nombre de noeuds morts après chaque itération. On remarque que la durée de vie du réseau est plus longue dans notre proposition par rapport aux deux autres protocoles.
Le nombre d'itérations est fixé à 1000, on remarque que le nombre de noeuds morts dans le protocole initiale est de 34 noeuds tandis qu'il est de 15 dans notre proposition, sur le dernier graphe on voit que presque tout les noeuds sont morts, il reste 2 noeuds sur 100 dans tout le réseau.

3.4 Conclusion

Dans ce dernier chapitre, nous avons présenté l'amélioration proposée à l'un des protocoles existants, nous avons évalué ses performances tout en la comparant à deux autres protocoles. Les résultats de la simulatuion ont montré que la taille des messages échangés entre les noeuds capteurs du réseau joue un rôle important dans la consommation d'énergie; plus la taille des messages est réduite, plus l'énergie consommée est moins importante, par conséquent la durée de vie du réseau est plus longue.

Conclusion Générale

Les réseaux de capteurs sans fil constituent un développement technologique majeur ces dérnières années, apportant des solutions aux différents problèmes dans plusieurs domaines d'applications liés à la sécurité, la santé, l'agronomie, etc.

Dans le cadre de ce mémoire, nous avons étudié le problème de gestion de clés dans les réseaux de capteurs sans fil, le but principal des protocoles de gestion de clés est d'assurer la sécurité des informations transitant dans le réseau et cela en générant, partageant et distribuant des clés cryptographiques, tout en tenant compte des contraintes de ressources des RCSFs et de trouver un moyen pour consommer moins d'énergie, ce qui permettera de prolonger la durée de vie du réseau. De nos jours beaucoup de protocoles de gestion de clés sont disponibles pour ce type de réseaux.

Après une étude de quelques protocoles de gestion de clés proposés dans la littérature, et un étude comparative de ces protocoles, notre choix s'est porté sur le protocole intitulé "Energy Effcient Key Management Scheme For Wireless Sensor Networks", ce système permet l'établissement et la maintenance des paires de clés entre les noeuds voisins qui sont générées en utilisant l'identifiant d'une fonction polynômiale, un nombre aléatoire et l'identifiant du noeud initiateur de la communication.

Nos contributions peuvent être énumérées comme suit:

- Une étude synthétique des travaux de recherche qui ont été faits dans le domaine de la gestion de clé dans les RCSFs.
- Présentation de notre proposition qui consiste à réduire le coût de communication en phase d'établissement de la paire de clé.

- Simulation du protocole lui même, de notre proposition ainsi que du protocole intitulé "Large Scale Wireless Sensor Networks with Multi-level Dynamic Key Management Scheme", et cela afin d'évaluer et comparer les performances de la proposition.
- Montrer que notre proposition aporte une amélioration par rapport à la version originale du protocole "Energy Effcient Key Management Scheme For Wireless Sensor Networks".

Comme perspectives, on peut envisager la mobilité de la station de base dans le réseau de capteurs, cela pourra mener à une basse consommation d'énergie, car en se déplaçant, la station de base sera plus proche des noeuds à qui elle souhaite transmettre les paramètres nécessaires pour le calcul des clés, comme deuxième perspective, on propose d'appliquer notre amélioration sur un réseau où les noeuds capteurs seront mobiles.

Bibliographie

[1] L.Khelladi, N.Badache, *Les Réseaux de Capteurs: Etat de l'Art*, Rapport de recherche. Laboratoire des Systèmes Informatiques, Faculté d'Informatique et Eléctronique, Bab Ezzouar, Alger, Février 2004.

[2] K.Heurteufeux, *Protocoles Localisés pour Réseaux de Capteurs*, Thèse pour obtenir le grade de docteur, Ecole Doctorale Informatique et Mathématique, Lyon, 2009.

[3] Y.Romdhane, *Evaluation des Performances des Protocoles S-MAC dans les Réseaux de Capteurs*, Rapport de projet de fin d'étude, Ecole Superieure des Communications, Tunis, 2007.

[4] I.F.Akyildiz, W.Su, Y.Sankarasubamaniam, E.Cayirci, *IEEE Communications Magazine*, Computer Networks, Vol. 40, N. 8, pp. 102-114, August 2002.

[5] R.Kaur, D.Sharma, N.Kaur, *Comparative Analysis Of Leach And Its Descendant Protocols In Wireless Sensor Network*, International Journal of P2P Network Trends and Technology, Vol. 3, Issue 1, pp. 51-55, 2013.

[6] W.Heinzelman, A.Chandrakasan, H.Balakrishnan, *Energy-Efficient Communication Protocol for Wireless Microsensor Networks*, International Conference on Systems Science, vol. 8, pp. 8020, January 2000.

[7] C.F.Garcia-Hernandez, P.H.Ibarguengoytia-Gonzales, J.Garcia-Hernandez , J.A.Perez-Diaz, *Wireless Sensor Networks and Applications : A Survey*, IJCSNS International Journal of Computer Science and Network Security, Vol. 7, N. 3, pp. 264-273, 2007.

[8] S.Sentilles, *Architecture logicielle pour capteurs sans-fil en réseau*, Rapport de recherche, Université de Pau et des Pays de l'Adour, Juin, 2006.

[9] N.Lasla, *La Gestion de Clés dans Les Réseaux de Capteurs sans-fil*, Mémoire de magister, Institut National de Formation en Informatique (I.N.I), Oued-Smar, Alger.

[10] P-R.Vamsi, K.Kant, *A Taxonomy of Key Management Schemes of Wireless Sensor Networks*, Fifth International Conference on Advanced Computing and Communication Technologies, pp. 690-696, 2015.

[11] A.Perrig, R.Canetti, D.Song, D.Tygar, *Efficient and Secure Source Authentication for Multicast*, In proceedings of the Networks and Distributed System Security Symposium, February 2001.

[12] D.Liu, P.Ning, *Effcient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks* In Proceedings of The 10th Annual Network and Distributed System Security Symposium, pp. 263-276, February 2003.

[13] D. Liu, P. Ning. *Multi-level mTESLA : Broadcast Authentication For Distributed Sensor Networks*, ACM Transactions in Embedded Computing Systems, Vol. 4(3), pp. 800-836, 2004.

[14] S.Zhu, S.Setia, S.Jajodia, *LEAP+ : Effcient Security Mechanisms for Large-Scale Distributed Sensor Networks*, ACM Transactions on Sensor Networks, Vol. 2, N. 4, pp. 500-528, November 2006.

[15] TinyOS : http ://www.tinyos.net/

[16] TinySec : http ://webs.cs.berkeley.edu/tos/tinyos-1.x/doc/tinysec.pdf

[17] K.JinPaek, J.Kim, C-S.Hwang, S.Lee, *Group-Based Key Management Protocol For Energy Effciency in Long-Lived And Large-Scale Distributed Sensor Networks*, Journal of Computing and Informatics, Vol. 27, pp. 743-756, 2008.

[18] H.DAI, H.XU, *Triangle-based Key Management Scheme for Wireless Sensor Networks*, Front Electron, Vol. 4(3), pp. 300-306, China, 2009.

[19] F.Delgosha, F.Fekri, *A Multivariate Key-Establishment Scheme for Wireless Sensor Networks*, IEEE Transactions on Wireless Communications, Vol 8, N. 4, April 2009.

[20] P-F.Oliveira, J.Barros, *A Network Coding Approach to Secret Key Distribution*, IEEE Transactions on Information Forensics and Security, Vol. 3, N. 3, September 2008.

[21] M-F.Younis, K.Ghumman, M.Eltoweissy, *Location-Aware Combinatorial Key Management Scheme for Clustered Sensor networks*, IEEE Transactions on Parallel and Distributed Systems, Vol. 17, N. 8, August 2006.

[22] M-K-R.Syed, H.Lee, S.Lee, Y-K.Lee, *MUQAMI+ : A Scalable and Locally Distributed Key Management Scheme for Clustered Sensor Networks*, Springer Ann. Telecommun, pp. 101-116, 2010.

[23] F.Liu, X.Cheng, *LKE : A Self-Configuring Scheme for Location-Aware Key Establishment in Wireless Sensor Networks*, IEEE Transactions on Wireless Communications, Vol. 7, N. 1, January 2008.

[24] K.Taekyoung, J.Lee, J.Song, *Location-Based Pairwise Key Redistribution for Wireless Sensor Networks*, IEEE Transactions on Wireless Communications, Vol. 8, N. 11, pp. 5436-5442, November 2009.

[25] A-K.Das, *Improving Identity-based Random Key Establishment Scheme for Large-scale Hierarchical Wireless Sensor Networks*, International Journal of Network Security, Vol. 14, N. 1, pp. 1-21, January 2012.

[26] S.Ruj, A.Nayak, I.Stojmenovic, *Pairwise and TripleKeyDistribution in Wireless Sensor Networks with Applications*, IEEE Transactions on Computers, Early Access Article, Likely to be published in the month and year of march 2013.

[27] A.Joux, *A One Round Protocol for Tripartite Diffie-Hellman*, W. Bosma (Ed.) : ANTS-IV, LNCS 1838, pp. 385-393. 2000.

[28] H.Dai, H.Xu, *Triangle-Based Key Management Scheme for Wireless Sensor Networks*, Front. Electr. Electron. Eng. DOI 10.1007/s11460-009-0034x, Vol. 4(3), pp. 300-306, China 2009.

[29] R.Vaid, V.Katiyar, *VLKM: Virtual Location-Based Key Management Sheme for Wireless sensor Networks*, International Conference on Parallel, Distributed and Grid Computing, 2014.

[30] N.Sughanthi, V.Sumathy, *Energy Efficient Key Management Scheme for Wireless Sensor Networks*, Int J Compute Commun, pp. 71-78, February, 2014.

[31] Y.Zhang, J.Pengfei, *An Efficient and Hybrid Key Management Scheme for Heterogeneous Wireless Sensor Networks*, The 26th Chinese Control and Decision Conference, May 31 2014-June, 2014.

[32] O-K.Sahingoz, *Large Scale Wireless Sensor Networks with Multi-Level Dynamic Key Management Scheme*, Journal of Systems Architecture, Vol 59, pp. 801-807, 2013.

[33] A-S.Uluagac, R-A.Beyah, Y.Li, J-A.Copeland, *VEBEK: Virtual Energy-Based Encryption and Keying for Wireless Sensor Networks*, IEEE Transactuions on Mobile Computing, Version accepted for publication by IEEE.

[34] D.Baker, H.X.Mel, *La Cryptographie Décryptée*, Campus Press edition, Référence Collection, pp. 414, Juillet 2001.

[35] A.Ouadjaout, M.Bagaa, A.Bachir, Y.Challal, N.Lasla, *Information Security in Wireless Sensor Networks*, Encyclopedia on Ad Hoc and Ubiquitous Computing, World Scientific, pp. 427-472, 2009.

[36] Z.Qin, X.Zhang, K.Feng, Q.Zhang,et J.Huang, *IBKM: An Efficient Identity-Based Key Management Scheme for Wireless Sensor Networks Using the Bloom Filter*, Sensor Journal, vol. 14, pp. 17937-17951, October 2014.

[37] G.Jolly, M.C.Kuscu, P.Kokate, et M.Younis, *LEKM: A Low-Energy Key Management Protocol for Wireless Sensor Networks*, IEEE Symposium on Computers and Communications, vol.1, pp. 335-340, Juillet 2003.

[38] S.Zhu, S.Setia, S.Jajodia, *LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor networks*, Proceedings of the 10th ACM Conference on Computer and Communications Security (New York, NY, USA), pp. 62-72, 2003.

[39] L.Eschenauer et V. D.Gligor, *A Key Management Scheme for Distributed Sensor Networks*, In Proceedings of the 9th ACM Conference on Computer and Communications Security, November 2002.

[40] X.Fan et G.Gong, *LPKM: A Lightweight Polynomial-Based Key Management Protocol for Distributed Wireless Sensor Networks*, Ad Hoc Network, vol 111, pp. 180-195, 2013.

[41] S.Balac, *Débuter avec Matlab*, Centre de Mathématiques INSA, Lyon, France.

Résumé

Un Réseau de Capteurs Sans Fil (RCSF) est un ensemble de capteurs communiquants par des liaisons sans fil. Leur déploiement dans des zones hostiles rend ce type de réseau très vulnérable, d'où la nécessité de mécanismes de sécurité efficaces et peu coûteux.

Au sein de ce mémoire, nous nous concentrons sur la gestion de clés dans les RCSFs, une étape très délicate lors de leur conception, non seulement la sécurité des messages échangés entre les noeuds capteurs doit être assurée, mais aussi la consommation d'énergie doit être réduite. Cependant, une étude de quelques protocoles existants dans la littérature portants sur la sécurité et la gestion de clés dans les RCSFs est établie. Par conséquent une amélioration de l'un de ces protocoles est proposée et évaluée.

Mots clés **:** RCSF, Gestion de clés, Sécurité.

Abstract

A Wireless Sensor Network (WSN) is a set of communicating sensors via wireless links. Their deployement in hostile areas makes this type of network very vulnerable, thus, it requires efficient and inexpensive security mechanisms.

In this thesis, we focus on key management in WSNs, a very delicate step at their design, not only the security of exchenged messages between sensor nodes must be assured, but energy consumption must also be reduced. However, a study of some protocols that already exist in literature about security and key management schemes in WSNs is done. By consequent, an improvement of one of these protocols is proposed and evaluated.

Keywords**:** WSN, Key management, Security.