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

Université A/Mira de Béjaia
Faculté des Sciences Exactes
Département d'Informatique

MÉMOIRE DE MASTER RECHERCHE

En
Informatique

Option
Réseaux et Systémes Distribués

Thème

Protocole de gestion de clés dans les réseaux de capteurs sans-fil

Présenté par : M^lle^ DJAMA Lynda
M^lle^ MEBARKI Soraya

Soutenu le 02 Juillet 2016 devant le jury composé de :

Rapporteur	**M. AISSANI Sofiane**	**M.A.A**
Président	**M. KHANOUCHE Mohamed-Essaïd**	**M.A.A**
Examinateur	**M. AKILAL Abdllah**	**M.A.A**

Promotion 2015-2016.

Remerciements

En préambule à ce mémoire nous remercions ALLAH le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

Nous souhaitons adresser nos remerciements les plus sincères à nos parents qui tout au long de ce travail, nous ont apporté leurs précieux soutien ainsi que leur encouragements.

Nous tenons à remercier sincèrement notre encadreur ***M. AISSANI Sofiane*** *pour sa disponibilité, son aide et le temps consacré qui ont constitué un apport considérable grâce auquel ce travail a pu être mené à bon port.*

Nos vifs remerciements vont également aux membres du jury qui ont accepter d'examiner notre travail et de l'enrichir par leurs propositions.

Nos sincères reconnaissances à tous nos enseignants pour les efforts fournis durant toute la période d'étude.

Enfin, nous tenons également à remercier toutes les personnes qui ont participé de près ou de loin à la réalisation de ce travail.

Dédicaces

A nos parents, pour leurs sacrifices déployés à notre égard, pour leur patience, leur amour et leur confiance. Qu'ils trouvent dans ce modeste travail, le témoignage de mes profondes affections et de mon attachements indéfectibles, nulle dédicace ne puisse exprimer ce que je leur doit.

A mes frères et à ma sœur et tous mes amis, ***Lynda, Dalia, Feriel, Zoubida, Selma, Lydia, Larbi, Sabrina et Samia*** pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.

Soraya

Dédicaces

A mes parents, pour leurs sacrifices déployés à mon égard, pour leur patience, leur amour et leur confiance. Qu'ils trouvent dans ce modeste travail, le témoignage de mes profondes affections et de mes attachements indéfectibles, nulle dédicace ne puisse exprimer ce que je leur doit.

A mon cher frère **Malek** qui a était d'une grande aide tout au long de cette période et tous mes amis et plus particulièrement à **Soraya**, **Feriel**, **Larbi**, **Sabrina** et **Massilia** pour chaque mot reçu, chaque geste d'amitié, à chaque main tendue et pour toute attention témoignée.

Lynda

Table des matières

Table des matières ii

Table des figures iii

Liste des tableaux iv

Liste des abréviations v

Introduction générale 1

1 Généralités sur les réseaux de capteurs sans fil et leurs sécurité 3

1.1 Capteur et ses composantes 3
1.1.1 Unité d'énergie 3
1.1.2 Unité de traitement 4
1.1.3 Unité de captage 4
1.1.4 Unité de communication 4
1.2 Présentation d'un RCSF 5
1.2.1 Architecture d'un RCSF 5
1.2.2 Objectifs de base des RCSF 6
1.3 Domaines d'application 7
1.3.1 Domaine militaire 7
1.3.2 Domaine médical 7
1.3.3 Application industrielle 7
1.3.4 Application écologique 7
1.3.5 Applications urbaine et domotique 8
1.3.6 Application environnementale 8
1.4 Facteurs et contraintes influençant l'architecture des RCSF . 9
1.4.1 Contraintes conceptuelles 9
1.4.2 Contraintes matérielles 10
1.5 Sécurité dans les RCSF 11

1.5.1 Objectifs de sécurité dans les RCSF 11
1.5.2 Vulnérabilités 12
1.5.3 Attaques sur les réseaux de capteurs 13

2 État de l'art sur la gestion de clés dans les RCSF 16
2.1 Problématique 16
2.2 Gestion de clés dans les RCSF 17
2.2.1 Objectifs de la gestion de clés dans les RCSF 17
2.2.2 Contraintes de mise en place et les phases d'un système de gestion de clés 17
2.3 Classification des schémas de gestion de clés pour les RCSF . . 19
2.3.1 Cryptographie hybride 20
2.3.2 Cryptographie Symétrique 22
2.3.3 Cryptographie Asymétrique 39
2.4 Comparaison 42
2.4.1 Critères de comparaison 42
2.4.2 Tableau comparatif 42
2.4.3 Discussion 44

3 Proposition et Simulation 46
3.1 Motivation 46
3.2 Schéma proposé 47
3.2.1 Hypothèses 47
3.2.2 Notation 48
3.2.3 Phases du LEAP-C 48
3.3 Exemple illustratif 50
3.3.1 Avant le déploiement 50
3.3.2 Après le déploiement 51
3.4 Simulation 52
3.4.1 Environnement de simulation 52
3.4.2 Résultats de simulation 53

Conclusion générale 59

Bibliographie 66

Table des figures

1.1 *Architecture matérielle d'un capteur [54].* 5
1.2 *Architecture d'un réseau de capteurs sans fil.* 6
1.3 *Quelques domaines d'applications pour les RCSF [54].* 8

2.1 *Contraintes de mise en place d'un système de gestion de clés [45].* 18
2.2 *Classification des schémas de gestion de clés pour les RCSF.* . 19
2.3 *Hierachie de la méthode [55].* 20
2.4 *Espace virtuel d'identifiant de noeuds d'un réseau de 100 noeuds [36].* 27
2.5 *Schéma probabiliste de base de gestion de clés [21].* 29
2.6 *Schéma probabiliste de q-composite de gestion de clés.* 30
2.7 *Le noeud **A** diffuse un message de négotiation de clés [30].* . . 38
2.8 *Addition de deux points sur une courbe elliptique [36].* 41

3.1 *Exemple illustratif de LEAP-C.* 50
3.2 *Réseau regroupé en cluster.* 51
3.3 *Énergie résiduelle des nœuds par rapport à la taille de la clé des trois protocoles.* 54
3.4 *L'espace de stockage par rapport à la taille de la clé des trois protocoles.* 54
3.5 *Énergie résiduelle des nœuds par rapport au round des trois protocoles.* 55
3.6 *La complexité en communication en nombre de messages émis par les nœuds dans chaque round des trois protocoles.* 56
3.7 *La complexité en communication en nombre de messages reçus par les nœuds dans chaque round des trois protocoles.* 57
3.8 *Passage à échelle en fonction du round* 58

Liste des tableaux

2.1 Tableau comparatif. 43

3.1 Les différentes notations utilisées dans la solution proposée . . 48
3.2 Les paramètres de simulation. 53

Liste des abréviations

CAN	**C**onvertiseur **A**nalogique **N**umérique.
CH	**C**luster **H**ead.
ECC	**E**lliptic **C**urve **C**ryptosystem.
INF	Key **INF**ection.
MAC	**M**essage **A**uthentification **C**ode.
MIT	**M**assachusetts **I**nstitute of **T**echnology.
PDA	**P**ersonal **D**igital **A**ssistant.
PKI	**P**ublic **K**ey **I**nfrastructure.
RAM	**R**andom **A**ccess **M**emory.
RCSF	**R**éseau de **C**apteur **S**ans **F**il.
ROM	**R**ead **O**nly **M**emory.
RSA	**R**ivest **S**hamir **A**delman.
SB	**S**tation de **B**ase.

Introduction générale

Au cours de ces dernières années le développement technologique des réseaux de capteurs sans fil (RCSF) a connu un essor important grâce aux avancées technologiques dans divers domaines, telles que la micro-électronique et la miniaturisation. C'est ainsi que de nouvelles voies d'investigation ont été ouvertes avec l'émergence des réseaux de capteurs sans fil. Les nœuds d'un RCSF apparaissent comme des systèmes autonomes à multifonctions, équipés d'une unité de traitement et de stockage de données, d'une unité de transmission sans fil et d'une batterie, organisés sous forme de réseau. Les capteurs sont capables de collaborer entre eux en vue de réaliser des tâches diverses et de communiquer les données collectées ensuite via le support sans fil à un centre de contrôle distant.

Par principe, les nœuds du réseau ont un mode d'organisation spontané ce qui constitue un atout qui les rend facilement intégrables dans une grande variété de domaines : militaire, environnemental, santé, habitat, éthologie, etc. Leur remarquable essor est dû à leur taille de plus en plus réduite, leurs prix de plus en plus faible ainsi que leur support de communication sans fil attrayant peu encombrant mais également peu sécurisant.

Toutefois, les nœuds d'un RCSF sont typiquement déployés dans des zones non surveillées, ce qui les rend vulnérables à plusieurs attaques dans lesquelles l'intrus peut prendre le contrôle d'un ou plusieurs nœuds pour perturber le bon fonctionnement du réseau en entier. Les ressources limités des capteurs (énergie limité,bande passante, capacité limité, etc.) et le support sans fil rendent les solutions de sécurité développées pour les réseaux filaires ou sans fil inapplicables aux RCSF. Par conséquent, plusieurs domaines de recherche sont apparus ces dernières années proposant des solutions de sécurité capables de remédier aux insuffisances des nœuds capteurs et aux vulnérabilités du médium de communication.

Afin de délivrer des services de sécurité, il est nécessaire que les nœuds communicants se partagent des clés cryptographiques pour le chiffrement et l'authentification des donnés échangées. Le nombre de nœuds élevés,

par conséquence le nombre de clés est potentiellement important, nécessite une gestion judicieuse afin de prendre en considération les contraintes de ressources imposées par ce type de réseaux.

Les techniques classiques de gestion de clés, qui utilisent une clé publique ou un centre de distribution de clé pour la cryptographie, sont inadéquates aux environnements de réseaux de capteurs. Pour cela, la plupart des solutions de gestion de clés proposées sont basées sur la cryptographie symétrique. De plus, afin d'achever l'établissement de clés entre les nœuds du réseau, ces solutions se basent sur la méthode de pré-distribution dans laquelle les clés sont chargées dans les nœuds capteurs avant leur déploiement. Les protocoles de gestion de clés proposés pour les réseaux de capteurs sans fil peuvent être classés dans plusieurs catégories selon la technique cryptographique, la façon dont les nœuds voisins partagent des clés communes, etc.

Dans ce mémoire, nous décrivons notre contribution qui consiste en une amélioration d'un protocole de gestion de clés dans les réseaux de capteurs sans fil, et ce, en terme de consommation d'énergie, d'espace mémoire et de communication.

Organisation du document :

Ce document est organisée comme suit :

Dans le chapitre 1, nous présentons les caractéristiques et l'architecture des RCSF et l'aspect de la sécurité .

Dans le chapitre 2, nous abordons la gestion de clés dans les réseaux de capteurs sans fil ensuite nous étudions quelques protocoles de gestion de clés.

Dans le chapitre 3, nous présentons notre protocole ainsi que les résultats de simulations qui nous permettent de comparer ce dernier avec d'autres protocoles de la littérature.

Chapitre 1

Généralités sur les réseaux de capteurs sans fil et leurs sécurité

Introduction

Les réseaux de capteurs sont un ensemble de dispositifs très petits, nommés "nœuds capteurs", formant un réseau sans infrastructure. Dans ces réseaux, chaque nœud est capable de détecter des énervements et de traiter l'information au niveau local ou de l'envoyer à un ou plusieurs points de collectes.

Ce chapitre est divisé en deux parties, la première étant la présentation des réseaux de capteurs sans fil, leurs architectures et domaines d'applications nous discuterons également les principaux facteurs et contraintes influençant leur conception. Dans la seconde partie nous définissons l'aspect de la sécurité dans RCSF.

1.1 Capteur et ses composantes

Définition 1.1.1. *[29] Un capteur est conçu de façon à transformer une grandeur physique observée (température, pression, humidité, etc.) sous la forme d'un signal électrique. Il est principalement composé d'un processeur, d'une mémoire, d'un émetteur/récepteur radio et d'une source d'énergie.*

Comme précédemment évoqué, plusieurs unités composent chaque nœud d'un RCSF, ce point sera détaillé dans se qui suit.

1.1.1 Unité d'énergie

Le capteur dispose de sa propre source d'énergie qui se trouve généralement sous la forme de batterie (ni rechargeable ni remplaçables dans la

plupart des cas). De ce fait, l'énergie est donc la ressource la plus précieuse du fait qu'elle influence sur la durée de vie d'un capteur et donc celle d'un réseau de capteurs [54].

1.1.2 Unité de traitement

Elle est généralement constituée d'un micro-contrôleur dédié et de la mémoire. Les micro-contrôleurs utilisés dans le cadre de réseaux de capteurs sont à faible consommation d'énergie. Une autre propriété est la taille de leurs mémoire qui est d'ordre de 10Ko de RAM pour les données et 10Ko de ROM pour les programmes. Outre le traitement de données, les [29].

1.1.3 Unité de captage

La fonction principale de cette unité est la collecte des données physique depuis l'objet cible. Elle se constitue d'un capteur qui transite les signaux analogique prélevés du phénomène observés au convertisseur (Analogique/Numérique (CAN)), à son tour il convertit ces signaux en données numérique interprétable par l'unité de traitement [35, 54].

1.1.4 Unité de communication

Cette unité dispose d'un transmetteur radio qui fournit au capteur la capacité de communiquer avec les autres au sein d'un réseau. Le système de transmission consomme environ 20 mW et possède une portée de quelques dizaines de mètres. Pour augmenter ces distances tout en préservant l'énergie, le réseau utilise un routage multi-sauts [25, 29].

Par ailleurs des composantes sont rajoutées en fonction du domaine d'application à titre d'exemple nous notons le système de localisation GPS, un générateur d'énergie (exemple : cellules solaires) ou un mobilisateur lui permettant de se déplacer. La Figure 1.1 illustre les composantes principale et additionnelles (représentés par des traits discontinus) d'un capteur.

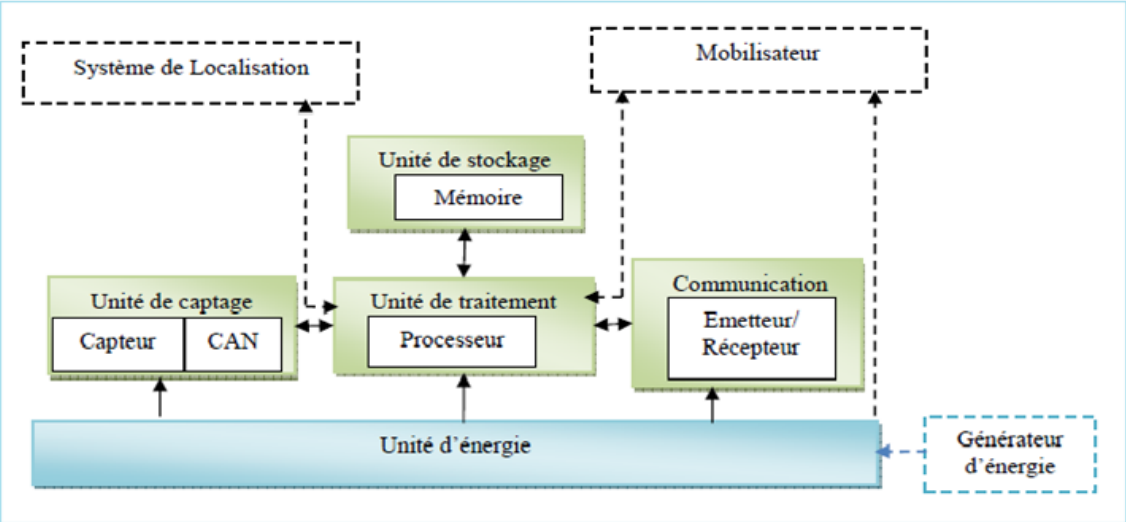


FIGURE 1.1 – *Architecture matérielle d'un capteur [54].*

1.2 Présentation d'un RCSF

Les réseaux de capteurs sans fil sont considérés comme un type spécial des réseaux *ad hoc* où l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle de hôtes et des routeurs [39].

1.2.1 Architecture d'un RCSF

Un réseau de capteurs sans fil usuellement abrégée en RCSF est un ensemble de capteurs déployés aléatoirement sur une zone géographique, dite zone d'intérêt, afin de surveiller un phénomène physique et de récolter leurs données d'une manière autonome. Les capteurs utilisent une communication sans fil pour router les données captées vers une station de base qui va transmettre, via internet ou satellite, ces informations à l'ordinateur central pour pouvoir les analyser [29, 40, 54]. La Figure 1.2 illustre cette architecture.

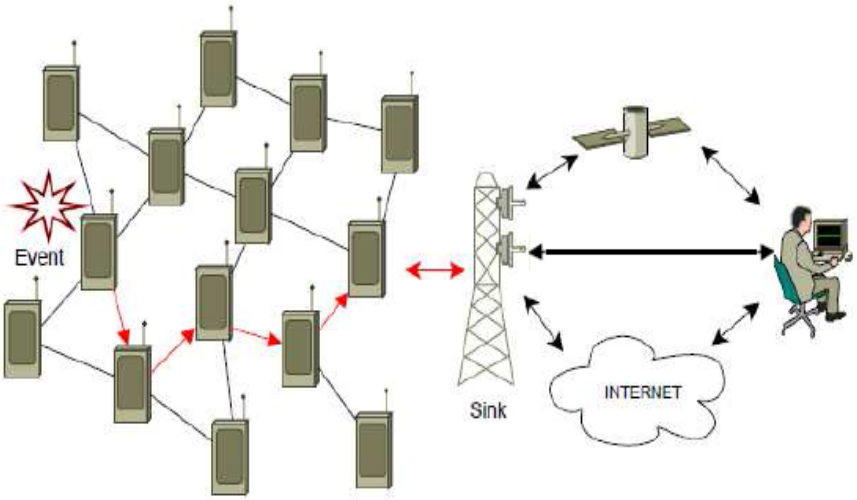


FIGURE 1.2 – *Architecture d'un réseau de capteurs sans fil.*

1.2.2 Objectifs de base des RCSF

Ces objectifs dépendent généralement des applications, cependant les taches suivantes sont communes à diverses applications :

— Recueillir des données de surveillance et d'agir en conséquence dans l'espace de leur déploiement [39].

— Déterminer les valeurs de quelques paramètres suivant une situation donnée. Par exemple, dans un réseaux environnemental, on peut chercher à connaitre la température, la pression atmosphérique, la quantité de la lumière du soleil et humidité relative dans un nombre de site, etc [39].

— Détecter l'occurrence des événements dont on est intéressé et estimer les paramètres des événements détectés. Dans les réseaux de contrôle de trafic, on peut vouloir détecter le mouvement de véhicules à travers une intersection et estimer la vitesse et la direction du véhicule [39].

— Classifier l'objet détecté. Dans un réseaux de trafic, un véhicule est-il une voiture, un bus, etc [39].

1.3 Domaines d'application

La miniaturisation, l'adaptation, la facilité du déploiement et la communication sans fil ont permis de généraliser l'utilisation des réseaux de capteurs. Aujourd'hui on trouve ce type de réseaux dans divers domaines, nous pouvons citer à titre d'exemple :

1.3.1 Domaine militaire

Comme beaucoup de technologie, le développement des RCSF a été suscité par des besoins militaires. En effet, les armées désirent être en mesure d'espionner discrètement les activités de leurs ennemis, de surveiller et analyser une zone de guerre (par exemple la détection d'agents chimiques, biologiques ou de radiations) ou de la position des troupes [29, 54].

1.3.2 Domaine médical

Les réseaux de capteurs peuvent être utilisés dans le domaine de la santé, pour assurer une surveillance permanente des organes vitaux de l'être humain (la détection de cancer, surveillance de glycémie, diabète, etc.) grâce à des micro-capteurs qui peuvent être implanté sous la peau. Ils peuvent prévenir de la dégradation de l'état de santé des patients et par conséquent d'être capable d'anticiper une hospitalisation en urgence et aussi de faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battement de cœur, etc. à l'aide des capteurs ayant chacun une tache particulière [41, 54].

1.3.3 Application industrielle

L'application de micro-capteurs dans le domaine industrielle permet de contrôler l'environnement dans les entreprises manufacturières, la gestion d'inventaire, de la télésurveillance après vente, etc. Ainsi une idée d'utilisation est d'installer ces organes sur des points d'usure des machines, ce qui permettrait de lancer des alertes quand leur état général se dégrade [39].

1.3.4 Application écologique

L'intégration des micro-capteurs dans le système de climatisation et de chauffage des immeubles. Ainsi, la climatisation ou le chauffage ne sont déclenchés qu'aux endroits où il y a du monde et seulement si nécessaire. Uti-

liser à grand échelle une telle application permettrait de réduire la demande mondiale en énergie [4].

1.3.5 Applications urbaine et domotique

Les capteurs entrent de plus en plus dans nos vies quotidiennes. Dans le milieu urbain, les capteurs sont déjà utilisés pour la surveillance du trafic routier. De même pour les maisons où ces capteurs peuvent être embarquées sur des appareils domestiques afin que l'utilisateur puisse les contrôler localement ou à distance [29, 54].

1.3.6 Application environnementale

Les capteurs dispersés dans les grandes forêts, les volcans, les profondeurs des océans, les régions polaires, ou encore d'autres planètes, permettent de surveiller une variété de paramètres ou des conditions environnementales telles que les incendies, les catastrophes naturelles, des conditions des animaux et des plantes[29, 39].

La Figure 1.3 ci dessous montre quelques domaines d'applications cités précédemment.

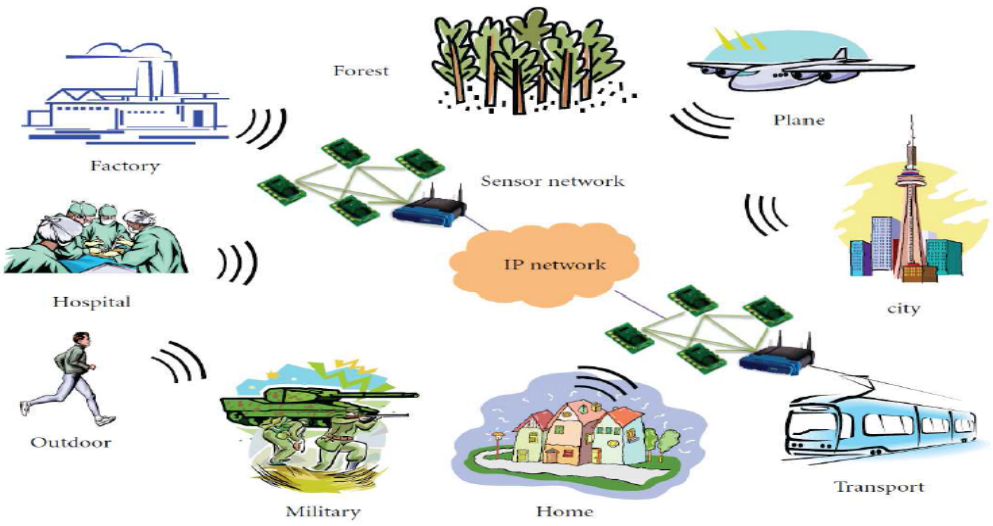


FIGURE 1.3 – *Quelques domaines d'applications pour les RCSF [54].*

1.4 Facteurs et contraintes influençant l'architecture des RCSF

La conception et la mise en place des RCSF sont influencées par plusieurs contraintes qui peuvent être conceptuelles ou matérielles

1.4.1 Contraintes conceptuelles

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

1.4.1.1 Tolérance aux pannes

Les nœuds d'un réseau de capteur peuvent être sujets à des pannes ou un blocage, qui peut être entraîné par plusieurs causes, notamment l'harassement d'énergie, dévastation physique, ou les interférences liées à l'environnement. Ces problèmes ne devraient pas affecter le bon fonctionnement du réseau car c'est le principe de la tolérance aux pannes [25, 54] .

1.4.1.2 Coût de production

Comme les RCSF consistent en un grand nombre de nœuds capteurs, le coût d'un seul capteur est très important pour l'évaluation du coût global de son réseau. Si ce dernier est plus cher que le déploiement d'un ensemble de capteurs classique, à ce moment l'usage de cette technologie ne serait pas rentable[29].

1.4.1.3 Passage à l'échelle

Du moment que les RCSF peuvent comporter des centaines de nœud ou plus, les nouveaux schémas de déploiement doivent être capables de travailler avec ce nombre élevé de nœuds. Par ailleurs, ils doivent utiliser la propriété de haute densité dans les réseaux de capteurs ; et donc pouvoir déployer un grand nombre de nœuds dans une petite surface[29].

1.4.1.4 Environnement

Les capteurs doivent être conçus d'une manière à résister aux différentes et sévères conditions de l'environnement : humidité, pluie, canicule [35].

1.4.1.5 Agrégation des données

Dans les RCSF, les données produites par les capteurs sont très reliées, ce qui engendre une redondance de données et l'épuisement d'énergie. L'une des techniques utilisées pour remédier à ces problèmes est l'agrégation des données, également appelée la fusion de données [54].

1.4.1.6 Média de transmission

Les nœuds communicants sont liés par un médium sans fil qui peut être un média optique, radio, infrarouge [54].

1.4.1.7 Consommation énergétique

Comme les nœuds capteurs sont des dispositifs micro-électronique, ils ne peuvent être équipés que par des sources limitées d'énergie ($<$0.5 Ah, 1.2 V). De plus, dans certain applications, il est impossible de réapprovisionner de l'énergie. Par conséquent, la durée de vie d'un capteur dépend fortement de la durée de vie de sa batterie associée. C'est pour cela que la bonne gestion et la conservation d'énergie est d'une importance primordiale dans les RCSF [39].

1.4.2 Contraintes matérielles

Parmi les contraintes matérielles liées aux RCSF, nous énumérons :

1.4.2.1 Dimension

La taille réduite des capteurs présente de nombreux avantages, elle permet un déploiement flexible et aisé du réseau. Néanmoins, la puissance des batteries utilisées pour alimenter les nœuds est limitée par leurs tailles [54].

1.4.2.2 Puissance de calcul

En plus de l'énergie les nœuds de capteurs ont aussi une capacité de calcul limités puisqu'ils utilisent des micro-contrôleurs de faibles fréquences et de mémoire restreinte dû à la taille des capteurs [54].

1.5 Sécurité dans les RCSF

1.5.1 Objectifs de sécurité dans les RCSF

Les réseaux de capteurs partagent certaines caractéristiques des réseaux ad hoc mais aussi possèdent des propriétés spécifiques aux RCSF. Conséquemment les objectifs de sécurité englobent ceux des réseaux traditionnels et les objectifs issus des contraintes intrinsèques aux RCFSs. Parmi les principaux objectifs de sécurité, nous citons [50] :

1.5.1.1 Confidentialité

La confidentialité constitue l'un des objectifs de sécurité les plus importants dans les réseaux de capteurs. Ce service désigne la garantie que l'information n'a pas été divulguée, et que les données ne sont compréhensibles que par les entités qui partagent le même secret . Le mécanisme qui permet d'obtenir ce service est le chiffrement des données à l'aide d'un algorithme cryptographique [13, 50].

1.5.1.2 Intégrité

Elle garantit que les données reçues n'ont pas été altérées durant leur transit dans le réseau de manière volontaire ou accidentelle. Elle peut être assurée par l'utilisation des fonctions de hachage cryptographiques qui permettent d'obtenir pour chaque message une empreinte numérique unique [9, 31].

1.5.1.3 Fraîcheur

Garantir que les données échangées sur le réseau sont récentes. Ce service permet de lutter contre la réinjection d'anciens messages interceptés par un attaquant [13].

1.5.1.4 Disponibilité

Elle signifie que le réseau est disponible pour assurer ses services aux parties communicantes lorsque ceci est nécessaire. Cette propriété reste difficile à assurer dans les RCSF étant donné les contraintes qui pèsent sur ces réseaux : topologie dynamique, ressources limitées des nœuds de transit et la communication sans fil [31].

1.5.1.5 Authentification

Elle permet de coopérer au sein des RCSF sans risque, en contrôlant et en identifiant les participants. En effet, on ne peut assurer une confidentialité et une intégrité des messages échangés si, dès le départ, on n'est pas sûr de communiquer avec le bon nœud [29].

1.5.1.6 Localisation sécurisée

Le besoin de se localiser et de connaître la position des autres nœuds peut être primordial dans de nombreux cas pour déjouer d'éventuelles attaques jouant sur les distances [31].

1.5.1.7 Collaboration et Auto-organisation

Après le déploiement, les capteurs doivent être capables, de fonctionner en collaboration et de s'auto-organiser pour se sécuriser eux-mêmes, en absence d'une sécurité physique et d'une tierce partie de confiance. Le développement des relations de confiance entre les nœuds capteurs, l'échange et établissement des clés de cryptages et la gestion des clés de session entre nœuds communiquant sont les fonctionnalités que doivent assurés les nœuds capteurs en s'auto-organisant et collaborant [45] .

1.5.2 Vulnérabilités

Quelques faiblesses sont inhérentes aux RCSF, d'autres liées à la technologie retenue. Nous distinguons deux catégories : les vulnérabilités physiques et les vulnérabilités technologiques.

1.5.2.1 Vulnérabilité physique

Parmi les vulnérabilités physiques, retenons le fait qu'un capteur est fréquemment installé dans un lieu peu sûr, tels que les lieux publics, les environnements naturels (forêt, région montagneuse, désert, etc.) ainsi que la plupart des bâtiments. De plus, l'attaquant peut facilement compromettre un nœud et obtenir le matériel cryptographique sauvegardé au niveau de sa mémoire, et cela dans le but de compromettre les liens de communication ou d'injecter du code pour détourner son utilisation [31].

1.5.2.2 Vulnérabilité technologique

La vulnérabilité est liés à la technologie sans fil sous-jacente, qui conque possédant un récepteur adéquat peut potentiellement écouter ou perturber

les messages échangés. Les messages de routage sont d'autant plus critique dans les RCSF puisque chaque nœud participe a l'acheminement des paquets à travers le réseau[39].

1.5.3 Attaques sur les réseaux de capteurs

Les attaques sur les réseaux de capteurs sans fil sont nombreuses, et leurs objectifs sont multiples. Dans ce qui suit, nous allons présenter une série des attaques (liste non exhaustive) les plus redoutables pour les RCSF :

1.5.3.1 Écoute passive

Cette attaque consiste à écouter le réseau et à intercepter les informations circulant sur le médium utilisant des ondes radio. Elle est facilement réalisable si les messages circulant sur le réseau ne sont pas cryptés. Par ailleurs l'écoute passive est difficile à détecter, car de par sa nature passive, elle ne modifie pas l'activité du réseau [37].

1.5.3.2 Analyse du trafic

Analyser le trafic peut permettre à un attaquant de connaître la position des nœuds d'agrégation de données ou des bases du réseau en repérant les chemins où le plus grand nombre de paquets transitent [37].

1.5.3.3 Nœud compromis ou nœud malicieux

Ce genre d'attaque physique peut permettre à un attaquant d'extraire par exemple les clés cryptographiques contenues dans le capteur, modifier ces circuits électroniques ou reprogramme le capteur, afin que ce dernier devienne ce que l'on appelle un nœud compromis ou nœud malicieux. Celui ci permet a l'attaquant de s'intégrer au réseau, de récupérer des informations ou de lancer d'autres attaques à partir d'un ou plusieurs de ces nœuds [37].

1.5.3.4 Attaque de trou noir

L'attaque du trou noir ou " black hole attack " consiste tout d'abord à insérer un nœud malicieux dans le réseau. Ce nœud, par divers moyens, va modifier les tables de routage pour obliger le maximum de nœuds voisins à faire transiter leurs informations par lui. Ensuite tel un trou noir dans l'espace, toutes les informations qui vont passer en son sein ne seront jamais retransmises [37].

1.5.3.5 Attaque de trou gri

Une variante de l'attaque précédente est appelée trou gris, dans laquelle seuls certains types de paquets sont ignorés par le nœud malicieux. Par exemple, les paquets de données ne sont pas retransmis alors que les paquets de routage le sont. Ce type d'attaque est ainsi plus difficile à détecter que l'attaque du trou noir, car le capteur malicieux tant qu'il se comporte de manière normale ne peut être facilement détecté [29, 37].

1.5.3.6 Attaque de trou de vers

L'attaque du trou de ver nécessite l'insertion dans le réseau de capteurs d'au moins deux nœuds malicieux qui ont pour but de tromper leurs voisins sur les distances les séparant, par conséquence les nœuds voisins vont principalement passer par ces nœuds malicieux pour faire circuler leurs informations. Ainsi les nœuds malicieux qui forment le trou de ver vont se trouver dans une position privilégiée qui va leur permettre d'avoir une priorité sur l'information circulant à travers leurs nœuds proches [37].

1.5.3.7 Altération de message

Un nœud malicieux va récupérer un message et l'altérer, en lui ajoutant des fausses informations (sur le destinataire, l'émetteur ou les données), en le modifiant ou bien en détruisant des paquets pour rendre incompréhensible le message [38].

1.5.3.8 Ralentissement

Un attaquant peut programmer des nœuds malicieux qui seront comme des agents dormant et qui n'auront que pour but d'augmenter la latence de distribution des données (par exemple avec une attaque de type trou gris) [38].

1.5.3.9 Attaque sybile

Cette attaque est définit comme un dispositif malveillant d'une manière illégitime prenant des identités multiples [50]. Dans cette attaque, un nœud malveillant peut revendiquer différentes identités dans le but de prendre de l'avantage sur les nœuds légitimes. L'attaque Sybil peut aussi mettre en péril les mécanismes comme l'agrégation des données [19], la sécurité, le routage [26], l'allocation de ressource [4], l'élection de cluster head ou la détection d'intrus [37].

1.5.3.10 Attaque par chantage

Un nœud malicieux fait proclamer qu'un autre nœud légitime est malicieux pour expulser ce dernier du réseau. Si ce nœud malicieux arrive à attaquer un nombre important de nœud, il pourra facilement perturber le fonctionnement du réseau [39].

1.5.3.11 Attaque spécifique aux capteurs

Dans ce type d'attaque l'attaquant va modifier de manière physique le comportement du capteur. Le but est de tromper le capteur, et ainsi d'envoyer ou d'enregistrer de fausses informations sur le réseau, ou bien tout simplement de faire réagir assez longtemps un noeud ou le réseau pour qu'ils consomment leurs énergies, comme dans le cas d'une attaque de type privation de mise en veille [31].

Conclusion

Dans ce chapitre nous avons procédé à l'étude des réseaux de capteurs sans fil. Nous avons posé les briques de base et fédéré quelques concepts généraux de sécurité nécessaires à la compréhension de nos problématiques dans la suite de ce rapport. Le chapitre suivant est dédié à l'étude de quelques protocoles de gestion de clés dans les réseaux capteurs sans fil.

Chapitre 2

État de l'art sur la gestion de clés dans les RCSF

Introduction

La gestion de clés constitue la fonction basique la plus critique dans la conception d'un système cryptographique. Dans ce chapitre nous présentons quelques techniques de gestion de clés dans les réseaux de capteurs sans fil, ce qui nous mène à définir ses objectifs, ses contraintes de mise en place et ses phases de déroulement. Nous allons aussi présenter quelques schémas de gestion de clés pour les RCSF et nous clôturons par une étude comparative de ces derniers.

2.1 Problématique

La cryptographie permet de garder secrètes les informations transmises à travers les réseaux, mais elle nécessite des techniques assurant la distribution, l'établissement et la gestion des clés cryptographiques utilisées dans l'opération de (chiffrement/déchiffrement). Par conséquent, la gestion de clés est un service primordial pour la sécurité de n'importe quel système basé sur la communication. Cependant les systèmes de gestion de clés dans les réseaux classiques ne sont pas applicables pour les RCSF, étant donné les caractéristiques des capteurs (énergie limité,bande passante, capacité limité, etc.). Pour cela, la gestion de clés doit tenir compte des ressources limitées des nœuds capteurs et de la possibilité de leur compromission, c'est pour cela que la plupart des solutions proposées reposent sur la cryptographie symétrique. Cela malgré les efforts déployés pour prouver qu'on peut utiliser le principe de cryptographie asymétrique ou combiner les deux types de chiffrement pour une sécurité meilleure et économe en ressources. Les résultats de ces efforts sont, pour le moment, peux convaincants.

2.2 Gestion de clés dans les RCSF

C'est le processus par lequel les clés sont générées, affectées, stockées, protégées, vérifiées, révoquées, renouvelées et détruites. c et l'authentification des entités communicantes dans le but de sécuriser le routage et de renforcer la coopération entre les nœuds du réseau [53].

Pour qu'un système soit entièrement sécurisé, chacune des entités du réseau doit disposer d'un ensemble de clés privées (dans un système à clés privées) ou de paire de clés publiques/privées (dans un système à clés publiques) afin de chiffrer (déchiffrer) les informations émises (reçues). Les clés sont générées et distribuées sur les capteurs de différentes manières [45] :

1. Pré-chargées sur les nœuds capteurs avant leur déploiement.
2. Générées sur les nœuds capteurs après le déploiement.
3. Générées par la station de base et distribuées sur les nœuds capteur après le déploiement.

2.2.1 Objectifs de la gestion de clés dans les RCSF

La gestion de clés dans les réseaux de capteurs sans fil permet de [45] :

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

2.2.2 Contraintes de mise en place et les phases d'un système de gestion de clés

I. Les contraintes qui découlent des propriétés des RCSF sont résumées sur la Figure 2.1 dans laquelle le premier niveau est consacré aux caractéristiques propre aux RCSF, le second niveau montre chaque contraintes que nous pouvons rencontré dans les RCSF et en fin le dernier niveau cite les solutions de mise en place d'un système de gestion de clés :

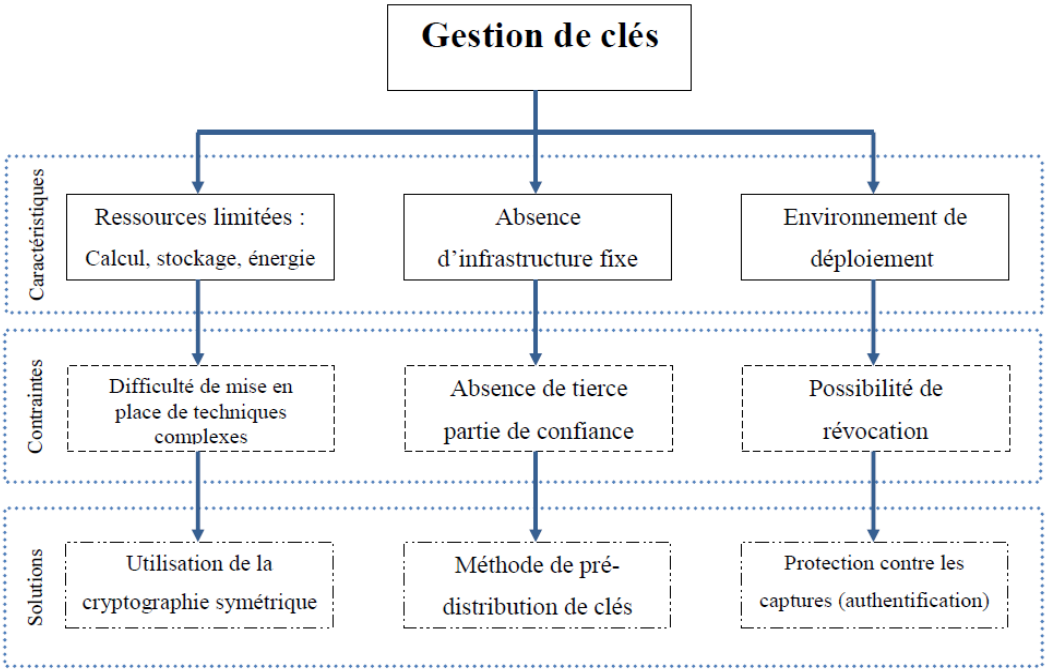


FIGURE 2.1 – *Contraintes de mise en place d'un système de gestion de clés [45].*

II. Le problème de gestion de clés dans les RCSF est décomposé par Hu et *al.* [23] en :

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

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

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

Mise en quarantaine des nœuds suspectés : un nœud suspect est un nœud qui ne fonctionne pas comme indiqué pour l'une des raisons suivantes :

i) Compromis par un attaquant,

ii) incapable d'assurer ses tâches car il a épuisé totalement son énergie. Les nœuds suspects doivent être isolés pour ne pas altérer le bon fonctionnement du système.

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

2.3 Classification des schémas de gestion de clés pour les RCSF

Nous avons choisi de faire une classification qui englobe l'ensemble des schémas de gestion et de distribution de clés en les regroupant en trois grandes familles. La première famille contient les schémas asymétriques, la deuxième regroupe les schémas symétriques et la troisième comporte les schémas hybrides. La Figure 2.2 illustre cette classification inspirée de celles présentées dans [36, 39]. Dans la suite, nous détaillons les schémas de cette figure.

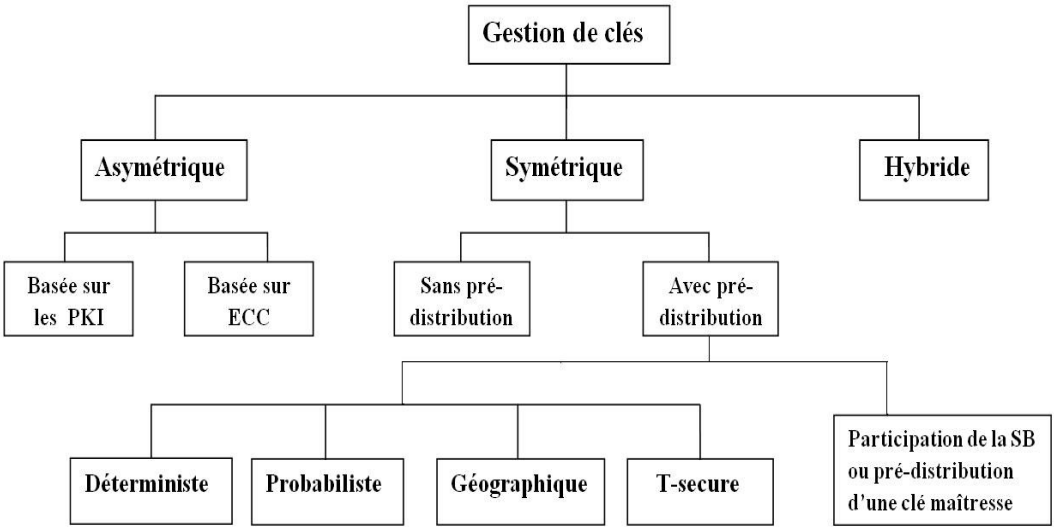


FIGURE 2.2 – *Classification des schémas de gestion de clés pour les RCSF.*

2.3.1 Cryptographie hybride

La cryptographie hybride est un système qui réunit les deux systèmes de cryptographie : asymétrique et symétrique. Zhang et Pengfei [55] proposent une méthode qui permet de combiner la cryptographie sur les courbes elliptiques avec la cryptographie symétrique afin d'améliorer la sécurité du réseau tout en prenant en considération la limitation de ressources des capteurs sans fil.

Cette dernière opte pour un modèle de réseau hierarchique et héterogène, de ce fait on distingue la **station de base (SB)** qui est dotée d'une capacité de calcul et d'énergie suffisante et que tous les nœuds lui font confiance, **High-end sensors (H-sensor)** qui ont de plus que l'énergie un matériels inviolables, ils sont considérés comme Cluster-Heads et les **Low-end sensors (L-sensor)** avec une énergie limitée. La Figure 2.3 illustre l'hiérarchie de cette méthode :

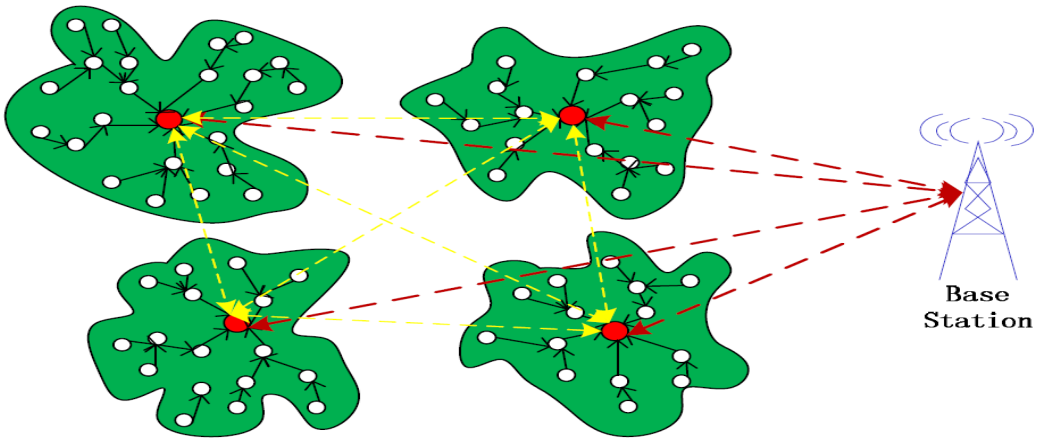


FIGURE 2.3 – *Hierachie de la méthode [55].*

2.3.1.1 Les phases de la méthode de Zhang et Pengfei

+ Pré-distribution de clés

Un serveur utilisant ECC se charge de munir les nœuds de clés avant leur déploiement. D'une part la station de base possède sa propre paire de clés (P^u_{BS}, P^r_{BS}) ainsi que la clé publique des Clusters Heads (P^u_H). D'autre part chaque H-sensor possède un identifiant unique id_{H_j}, la clé publique de la station de base (P^u_{BS}), sa propre paire de clés (P^u_H, P^r_H), la liste de clés publiques de tous les L-sensors (P^u_{Li}) et une fonction de hachage. De plus

chaque L-sensors sont chargés avec un identifiant unique id_{li}, leurs clés privées (P^r_{Li}) , la clé publique du Cluster-Head (P^u_H) et une fonction de hachage *hash()*.

+ Phase d'établissement de clusters

Les H-sensors diffusent un message m_{Hj} $(id_{Hj}, (x_{Hj}, y_{Hj}))$ contenant leur identifiant et leurs coordonnées . Les L-sensors choisissent leurs Head en fonction de l'intensité du signal, les L-sensors diffusent un message contenant leur identifiant et leurs coordonnées m_{Li} $(id_{Li}, (x_{Li}, y_{Li}))$ et construisent des informations (I_{Li}) sur l'ensemble de leurs voisins, ensuite ils envoient un message P_{Li} $(id_{Li}, I_{Li}))$ au H-sensors choisis. En se basant sur les P_{Li} reçus, les H-sensors construisent une table de routage.

+ Phase d'établissement de clés

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

Les H-sensors et les L-sensors utilisent l'algorithme de Diffie-Hellman pour établir une clé partagée entre eux. Supposons que le noeud H_j et le noeud L_i appartenant au même cluster veuillent établir une clé secrète alors :

Le noeud H_j calcule la clé ainsi :

$$K_{HjLi} = P^r_{Hj} \times P^u_{Li} = P^r_{Hj} \times P^r_{Li} \times G$$

Tandis que le noeud L_i calcule la clé ainsi :

$$K_{LiHj} = P^r_{Li} \times P^u_{Hj} = P^r_{Li} \times P^r_{Hj} \times G$$

Les clés obtenues sont équivalentes : $K_{HjLi} = K_{LiHj}$

G représente un point de base de la courbe. Les L-sensors communiquent entre eux via 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

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

Sinon

$$K_{uv} = hash(r \parallel id_v \parallel id_u)$$

Quant à la clé du cluster elle est calculée ainsi : $K_o = hash(r \parallel id_{Hj})$

+ Révocation des clés

Lorsqu'un L-sensor est capturé, le H-sensor du même cluster utilise un système de détection d'intrusions et identifie ce noeud. Il envoie un message de révocation contenant l'identifiant du nœud capturé 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 nœud compromis, dans ce cas les clés de session partagées avec le nœud sont révoquées.

+ Ajout de nouveaux nœuds

Si un nœud 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 de nouveau nœud. Le H-sensor lui envoie des informations sur le routage ainsi que le nombre r, qui sera utilisé par le nouveau nœud pour établir des clés de session avec les autres nœuds du cluster. Après la génération des clés le nombre r sera supprimé.

2.3.2 Cryptographie Symétrique

La plupart des schémas de gestion de clés traditionnelles ne sont pas conforme en raison de ressources et de l'énergie limitées des nœuds de capteurs. Les systèmes de gestion de clés symétriques sont les plus adaptés pour RCSF. Cette catégorie de mécanisme est utilisée dans le but d'établir une clé commune entre deux nœuds d'un réseau de capteurs [47].

De façon générale, les modèles de distribution de clés symétrique sont divisés en deux catégories : schémas avec la pré-distribution de clés et les schémas sans pré-distribution de clés [48].

2.3.2.1 Schéma d'absence de la pré-distribution

Ce mécanisme considère la réalité des RCSF. Pour cette raison, de nombreux systèmes de gestion de clés dynamiques pour les réseaux de capteurs ont été proposés. Pour un tel système, si un adversaire ne sait pas où et quand les nœuds sont déployés il serait difficile de lancer une attaque active [14].

◇ "Key Infection"

Anderson et *al.* ont proposé INF [2] qui suppose un déploiement de masse

et que les nœuds sont statiques. INF installe des clés symétriques entre les nœuds et leurs voisins d'un seul saut.

En premier temps, chaque nœud génère simplement une clé symétrique et l'envoie dans l'espace libre à ses voisins. Une approche de chuchotement de clés est employée, c'est-à-dire, la clé est au commencement transmise à un niveau de puissance bas. La puissance de transmission est alors augmentée jusqu'à ce que la clé soit entendue par au moins un voisin d'un seul saut et qu'une réponse soit reçue.

Dans l'hypothèse posée par les auteurs, l'adversaire a très peu de probabilité d'intercepter une telle communication locale et de faible portée, le nœud i, quand il ce met en mode veille, diffuse une clé k_i et peut-être une demi-douzaine d'autres nœuds sont devenu a portée. A son réveil, ils détectent la présence de chacun et commencer à s'organiser en un réseau . Les étapes du protocole s'effectuent comme suit :

1. A diffuse K_A
2. B et C entendent le message de A :
   a. B génère une clé de session K_{AB}
   b. C génère une clé de session K_{AC}
3. Les noeuds B et C renvoient leurs clés au nœud A :
   a. B envoi : $\{B, K_{AB}\}_{KA}$
   b. C envoi : $\{C, K_{AC}\}_{KA}$

L'avantage dans ce mécanisme est que la station de base ne participe pas a une installation des clés, ce qui implique moins d'énergie consommée. De plus, on n'a pas besoin de charger des clés dans les capteurs avant le déploiement. INF est simple et permet le passage à l'échelle. Cependant, la sécurité est faible. INF est vulnérable à l'écoute clandestine pendant le chuchotement de clés. En outre, il n'y a aucune authentification des parties communicantes, la capture des nœuds n'est pas prise en compte.

2.3.2.2 Schéma avec la pré-distribution

Ce schéma repose sur la distribution de clés statique, alors les clés sont chargées dans les nœuds de capteurs avant le déploiement de telle sorte que chaque nœud puisse calculer la clé commune qu'il partagera avec ses voisins pour pouvoir établir la communication entre eux. On peut la découper en cinq sous classes :

I. Déterministe

Les protocoles de gestion de clés déterministes assurent que chaque nœud est capable d'établir une clé par-paire avec ses voisins. Pour garantir le déterminisme, les protocoles, tels que [16] et [18], utilisent une clé commune, transitoire et pré-chargée dans tous les nœuds avant leurs déploiements. Dans ce qui suit nous allons présenté trois exemples de protocoles déterministes qui sont : LEAP, OTMK, shéma de Jolly et *al.* et PIKE.

◇ LEAP : "Localized Encryption and Authentication Protocol"

Zhu et *al.* propose *LEAP* [56] un protocole de gestion de clés à la fois localisé et d'authentification qui limite l'impact d'un nœud compromis sur son voisinage.

Le schéma de base

quatre types de clés sont employés pour chaque nœud : une clé individuelle unique partagée avec la station de base, pour sécurisé la communication entre eux ; une clé globale partagée par tous les nœuds du réseaux, employée par la station de base pour chiffrer les messages diffusés dans tout le réseau ; une clé de Cluster (groupe) partagée par un nœud et tous ses voisins du même groupe, et elle est employée pour sécuriser des messages localement diffusés et une quatrième clé partagée entre deux nœuds "Paire wise key" elle est employée pour envoyer des informations privées.

Une phase d'établissement de clés partagées entre les paires de nœuds, les clés de cluster sont effectuées après le déploiement. La clé individuelle est générée et pré-chargée dans chaque nœud avant son déploiement. Pour un nœud u, sa clé individuelle est $K_u^m = f_{K^m}(u)$, où f est une fonction pseudo-aléatoire et K^m est la clé maitresse qui n'est connue que par la station de base, chaque nœud emploie sa clé individuelle pour chiffrer les messages échangés avec la station de base. Les messages de diffusion sont envoyés par la station de base chiffrés avec la clé globale mais authentifié avec le protocole $\mu TESLA$. Quand un adversaire obtient un nœud, *LEAP* suppose que le nœud ne peut pas être compromis avant t_{min} de temps. Toutes les fois qu'un nœud est déployé dans un RCSF, il a besoin d'un temps (t_{est}) pour identifier ses voisins et pour établir des clés avec eux, on suppose que $t_{min} > t_{est}$.

Le protocole *LEAP* est un peu coûteux en mémoire, du fait que chaque nœud doit stocker quatre types de clés. Si le nombre de voisins d'un nœud est

d, il doit stocker une clé individuelle, d clés pour ses voisins, d clés de cluster, et une clé de groupe. En outre, un nœud doit stocker une chaîne de clé L à sens unique pour le broadcast local et pour l'authentification des messages de la station de base (utilisation de $\mu TESLA$). Le tout à stocker sera $3d+2+L$. De plus, la quantité en mémoire du $LEAP$ dépend de la densité du réseau. Cette solution fournit l'authentification et la confidentialité mais avec un coût élevé de communication et la sécurité du système dépend de la clé de groupe qui peut être compromise.

◇ **OTMK : "Opaque Transitory Master Key Establishment"**

Dans [17] les auteurs ont proposé une solution de gestion de clés basée sur le concept de la clé initiale transitoire de $LEAP$. Ce protocole déterministe, comme $LEAP$, permet l'établissement des clés par-paires entre des noeuds voisins. Pour alléger les conséquences de la compromission de la clé initiale K_{IN} par un adversaire, les auteurs ont défini deux propriétés qui doivent être vérifier dans le développement des solutions tolérantes de gestion de clés : (1) *la propriété d' opacité* - Un adversaire ne peut pas déduire la plupart des clés utilisées dans le réseau par la compromission d'un petit nombre de nœuds ; (2) *la propriété d'inoculation*-Un adversaire ne peut pas aider un nœud étranger de joindre le réseau par la compromission d'un petit nombre de nœuds. Contrairement au protocole $LEAP$, dans $OTMK$, même si la clé initiale K_{IN} est compromise, la propriété d'opacité reste vérifiée. Cependant, la propriété d'inoculation est non vérifie dans les deux protocoles.

Le schéma de base

Chaque nœud est pré configuré, avant le déploiement, avec une même clé initiale K_{IN}. Pour établir des clés par-paires avec ses voisins, chaque nœud u ; diffuse un message $JOIN$ comme suit :

$u \longrightarrow * : JOIN \parallel E_{K_{IN}}(ID_u; nonce)$

où ID_u : IDentité du nœud u

$nonce$: Un nombre aléatoire

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

$v \longrightarrow u : REPLAY \parallel E_{K_{IN}}(ID_v, nonce + 1 \parallel K_{v\,u})$.

A la réception au niveau de nœud u reçoit ce message, il le décrypte puis vérifiée le $nonce$. Si c'est vérifiée, il enregistre le nœud v comme étant voisin vérifiée. La clé par-paire entre u et v est soit K_{vu} générée par le nœud v, ou bien K_{uv} générée par le nœud u. Si $ID_u < ID_v$, les nœuds u et v utilisent K_{uv}

comme clé par-paire entre eux sinon ils utilisent K_{vu}. Dans le but de réduire les chances de compromettre la clé initiale K_{IN} chaque nœud, après un temps suffisant pour l'établissement des clés par-paires, détruit la clé initiale de sa mémoire.

◇ **Schéma de G. Jolly, M. Kusku, P. Kokate et M. Younis**

Jolly et *al.* [24] ont proposé un protocole déterministe de gestion de clés pour le domaine militaire basé sur la méthode de pré-distribution de clés ayant une architecture hiérarchique et reprenant les notions du clustering appliquées au routage de données dans les RCSF. En effet, les nœuds capteurs sont regroupés selon des centres d'intérêt et liés entre eux par des passerelles est caractérisées par des ressources énergétiques et des capacités de calcul et de stockage élevées. Dans cette architecture, on utilise également des nœuds de commande responsables des missions de sécurité du réseau et représentent la tierce partie de confiance de tous les nœuds.
Les auteurs supposent que les nœuds et les passerelles n'ont aucune connaissance à postériori de la topologie du réseau et sont aléatoirement déployés.

Les phases du protocole :

- **Distribution de clés :** chaque nœud stocke deux clés secrètes, une partagée avec la passerelle et l'autre avec le nœud de commande. Les passerelles partagent des clés entre elles et avec le nœud de commande et les clés des capteurs de son groupe ce dernier stocke toutes les clés du réseau. Les clés sont distribuées avant le déploiement et aucune action supplémentaire de distribution, immédiate ou après le déploiement, n'est envisagée et un gain d'énergie sur les opérations d'émission / réception de clés est tiré profit.

- **Construction de groupes :** après le déploiement, chaque nœud diffuse un message « Hello » de découverte de voisins contenant son identificateur et l'identificateur de la passerelle qui contient la clé partagée, chaque nœud reçoit une réponse de la passerelle correspondante.

- **Révocation de clés :** si un nœud est compromis, le nœud de commande et la passerelle l'expulsent du groupe en ignorant les routes qui passent par lui. Si une passerelle est compromise, le nœud de commande l'expulse et choisit une autre passerelle pour la remplacer.

- **Renouvellement de clés :** le nœud de commande produit les nou-

velles clés et les transmettent aux passerelles. Chaque passerelle transmet à son tour une clé pour chaque nœud de son groupe.

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

◇ **PIKE : " Peer Intermediaries for Key Establishment "**

Chan et Perrig [11] propose $PIKE$ dédié à l'établissement de clés. $PIKE$ utilise des grilles pour faire la pré-distribution de clés. $PIKE$ propose une méthode d'établissement de clés déterministe qui pourra se réaliser en stockant des clés d'ordre $O(\sqrt{N})$, N étant le nombre de nœuds du réseau. Cette méthode utilise des nœuds ordinaires du réseau comme intermédiaires de confiance, les clés sont pré-distribuées pour que n'importe quel couple de nœuds A et B puisse toujours avoir la possibilité de trouver un nœud C du réseau qui partage une clé unique avec A et B.

Les identifiants (ID) des nœuds sont arrangés dans une structure en grille carrée de dimension $\sqrt{N} \times \sqrt{N}$. Chaque nœud aura un ID de la forme (x, y) où $x, y \in \{0, 1, 2, ..., \sqrt{N-1}\}$, chaque nœud sera seulement chargé d'une clé secrète unique avec chacun des nœuds des deux listes : (i, y) où $\forall i \in \{0, 1, 2, ..., \sqrt{N-1}\}$ et (x, j) où $\forall j \in \{0, 1, 2, ..., \sqrt{N-1}\}$. La Figure 2.4 présente une grille d'identifiants de 100 nœuds. Chaque nombre présent dans la grille est l'identifiant d'un nœud du réseau. Cela permet de constater que les nœuds 91 et 14 peuvent partager une clé unique avec chacun des nœuds qui appartiennent à leur colonne ou à leur ligne. Ainsi, les nœuds 11 et 94 peuvent jouer le rôle d'intermédiaires entre eux.

00	01	02	03	04	...	09
10	11	12	13	14	...	19
20	21	22	23	24	...	29
30	31	32	33	34	...	39
.	.	.	.	.		.
.	.	.	.	.		.
.	.	.	.	.		.
90	91	92	93	94	...	99

FIGURE 2.4 – *Espace virtuel d'identifiant de noeuds d'un réseau de 100 noeuds [36].*

Chaque nœud de la Figure 2.4 sera chargé avec 18 clés (9 clés pour les nœuds appartenant à sa ligne et 9 clés pour les nœuds appartenant à sa colonne). Pour généraliser, chaque nœud d'un réseau de taille N est chargé avec $2(\sqrt{N}-1)$ clés distinctes.

II. Probabiliste

Dans le schéma probabiliste, chaque nœud est pré-chargé, avant le déploiement, avec un sous ensemble de clés prélevées à partir d'un grand ensemble de clés.

L'idée maîtresse est que deux nœuds voisins peuvent communiquer si ils ont une certaine probabilité d'avoir partagé une clé commune qui appartient aux deux sous ensembles de ces voisins.

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

Eschenauer et Gligor dans [21] propose un schéma de base de gestion de clés basé sur la clé probabiliste partagée entre les nœuds d'un graphe aléatoire sans possibilité substantielle de calcul et de communication. Le schéma complet est le suivant :

• **Avant le déploiement**, un grand ensemble P de clés est générer (entre 2^{17} et 2^{20} clés). Pour chaque noeud, m clés sont choisies au hasard à partir de l'ensemble P. 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 probabilité p d'avoir au moins une clé en commun.

• Après le déploiement, une **phase de découverte de clés partagées** est effectuée, les noeuds découvrent leurs voisins et plus particulièrement ceux avec qu'ils sont en mesure de communiquer de façon sécurisée car ils possèdent une clé identique dans leur trousseau de clés respectif, cette clé devient la clé de session de lien entre eux. La Figure 2.5 illustre cette phase.

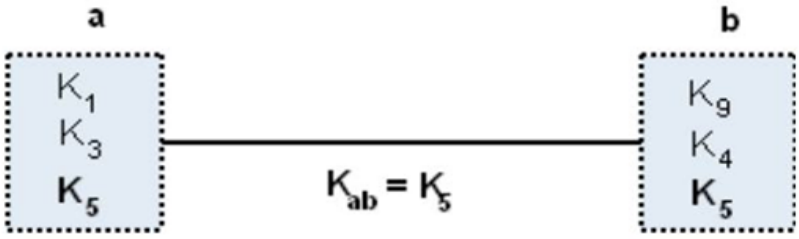


FIGURE 2.5 – *Schéma probabiliste de base de gestion de clés [21].*

• A la fin de cette phase, **une phase d'établissement de chemin de clé** aura lieu, les nœuds utilisent les liens existants pour mettre en place des clés partagées avec leurs voisins qui ne partageaient pas de clé en commun avec eux.

• **La révocation** d'un nœud compromis se fait par l'élimination de son trousseau de clés. Pour cela, un nœud contrôleur annonce un message de révocation contenant une liste signée de m identificateurs des clés pour que ces clés soient retirées des trousseaux des autres nœuds. La liste des identités est signée par une clé de signature K_e générée par le nœud contrôleur et envoyée en unicast à chaque nœud en la chiffrant avec la clé K_{ci} (Noter que les clés K_{ci} sont partagées entre le i^{me} contrôleur et chaque nœud capteur pendant la phase de pré-distribution de clés). Quelques liens disparaitrons à cause de la suppression des clés du nœud compromis ce qui nécessite une reconfiguration de ces liens (par la découverte de clés partagées ou l'établissement de chemin de clé).

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

◇ **Schéma de Chan, Perring et Song**

Le schéma proposé par Chan et *al.*[12] est un ensemble d'amélioration au schéma de Eschenauer et Gligor et porte sur les corrections suivantes : i) un nœud doit partager q clés ($q > 1$) avec un autre nœud pour établir un chemin de clés sécurisé au lieu d'une seule, ii) a pour but de renforcer la sécurité d'un lien entre deux nœuds a et b en changeant la clé k issue de l'ensemble de clés P par une valeur aléatoire $k^{'}$, iii) introduire un protocole d'authentification nœud à nœud.

- **Schéma q-composite**

Une paire de nœuds (A, B) doit partager q clés pour établir un lien sécurisé. La nouvelle clé entre A et B est le hash de clés communes tout cela est représenté dans la Figure 2.6.

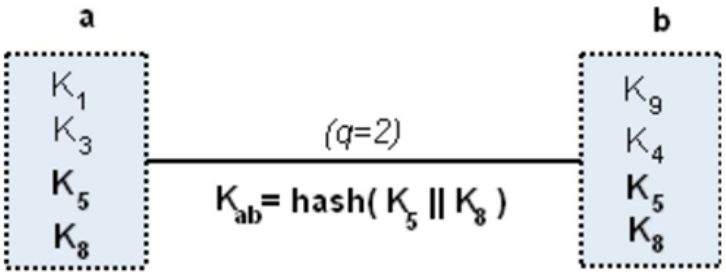


FIGURE 2.6 – *Schéma probabiliste de q-composite de gestion de clés.*

Lorsque la quantité de chevauchement de clés exigée augmente, il devient exponentiellement plus difficile à un attaquant avec un ensemble donné de clés de casser un lien. Cependant, pour préserver une probabilité donnée p' que deux nœuds partageant des clés, il est nécessaire de réduire la taille de l'ensemble de clés P, ceci permet à un attaquant de gagner un plus grand échantillon de P en cassant peu de nœuds.

- **Renforcement de la sécurité des liens**

Si un attaquant compromet un nœud C et récupère sa liste de clés $m_C = K_1, K_2, \ldots, K_m$, il se peut que deux nœuds A et B établissent un lien en utilisant une clé $K_{AB} \in m_C$. Dans ce cas, l'attaquant utilise la clé K_{AB} pour déchiffrer les messages échangés entre A et B. Pour éviter un tel scénario, les auteurs suggèrent que le nœud A connait les différentes routes qui mènent vers le nœud B avec un certain nombre de sauts h et les utilisent pour envoyer j valeurs aléatoires $v_1, v_2, \ldots, v_j$ sur les j routes qui mènent vers B. La nouvelle clé $K^{'}$ sera calculée par B comme suit : $K^{'} = K \,\Phi\, v_1 \,\Phi\, v_2 \,\Phi. \ldots ... \Phi\, v_j$. Par conséquent, plus le nombre de chemin entre deux nœuds augmente, plus le degré de sécurité entre eux augmente.

- **Protocole d'authentification nœud à nœud**

Un protocole a la propriété de l'authentification nœud à nœud si n'importe quel nœud peut assurer l'identité des nœuds avec lesquels il communique, cette propriété est utile en soutenant beaucoup de fonctions de sécurité :

- A la détection d'un nœud compromis, il est essentiel qu'un nœud soit certain de l'identité du nœud compromis avant la prise de n'importe quelle action.

- Permettre aux différents nœuds de capteurs de résister à une attaque de réplique de nœud en maintenant que des identités de nœuds qui ont été déjà insérés dans le réseau et de rejeter la connexion additionnelle établie par cette identité.

- L'authentification nœud à nœud peut se débrouiller des fonctions de sécurité loin de la station de base en permettant à des nœuds d'exécuter d'une façon autonome des révocations sur des nœuds compromis, cela améliore le temps de réaction à des intrusions détectées par le réseau.

◇ **" key management using deployement knowledge "**

Du et *al.* [20] ont fait une extension de la gestion de clés développée par [21]. Leur schéma exploite des connaissances sur le déploiement. Cette connaissance préalable de déploiement est utile pour la pré-distribution de clés. Quand les capteurs voisins sont connus, la pré-distribution principale devient facile et exige simplement que pour chaque nœud n de stocker des paires de clés entre n et chacun de ses voisins. Ceci garantit que chaque nœud peut établir un canal sécurisé avec chacun de ses voisins après déploiement. La connaissance de déploiement dans ce schéma est modélisée en utilisant les fonctions de densité de probabilité (pdf : probability density function). Quand la pdf est uniforme, aucune information ne peut être obtenue sur la résidence d'un nœud après déploiement [53].

III. Géographique

Les schémas de gestion de clés géographique utilisent la position des nœuds pour augmenter la connectivité du réseau et gérer convenablement et aisément les clés entre les nœuds voisins. Ils nécessitent des composants spéciaux comme le GPS ou des algorithmes de localisation pour localiser les nœuds après le déploiement. Plusieurs variantes sont développées pour associer la position des nœuds et la création des clés entre eux. Exemples de protocoles géographiques : LBK [33], LKE [34].

◇ **LBK : "Location-based pairwise key establishments for static sensor networks" :**

Liu et Ning [33] ont proposé LBK, ils supposent que les nœuds du réseau

sont statiques, de nouveaux nœuds peuvent être ajoutés au réseau à tout moment et qu'il est souvent possible de déterminer approximativement les positions des capteurs qui permettra aux nœuds d'utilisés les informations sur leur emplacement pour partager des clés par paire avec leurs voisins.

Déroulement du protocole :

Pré-distribution des clés : les nœuds sont pré-chargés par le serveur d'instalation (setup server) avec des clés par paire suivant les coordonnées (x, y) choisies pour les endroits prévus. Plus précisément, pour chaque nœud capteur u, le serveur découvre l'ensemble S de ses voisins les plus proches de son endroit prévu. Pour chaque nœud $v \in S$, le serveur génère une clé aléatoire unique $K_{u,v}$ si aucune autre clé n'est assignée à ce couple (u, v), et distribue ensuite la clé $K_{u,v}$ aux nœuds capteurs u et v.

Etablissement de clés directe : après le déploiement, si deux nœuds u et v souhaitent établir une communication sécurisée entre eux, il suffit de vérifier s'ils partagent une clé pré-distribuée. L'algorithme pour identifier une telle clé commune est trivial, parce que chaque clé par pair dans un capteur particulier a été associé avec l'ID de capteur.

Établissement de clés indirecte : après le déploiement, si deux nœuds u et v ne partagent aucune clé pré-distribuée, ils peuvent s'associer à un voisin intermédiaire qui partage des clés par paire avec chacun d'eux pour établir une clé de session. Pour cela, l'un des deux nœuds (supposons le nœud u) diffuse un message Broadcast avec son identificateur et l'identificateur du nœud v, le nœud y intercepte le message, et vérifie s'il partage une clé $K_{u,y}$ avec u et une clé $K_{y,v}$ avec le nœud v. Si oui, le nœud y transmet au nœud u un message contenant les informations $E_{K_{u,y}}(K)$ et $E_{K_{y,v}}(K)$. La clé de session est calculée à partir des clés $K_{u,y}$ et $K_{y,v}$. A la réception du message le nœud u obtient la clés de session en déchiffrant $E_{K_{u,y}}(K)$, il transmet à son tour l'information $E_{K_{y,v}}(K)$ au nœud v.

Addition de noeud : Pour ajouter un nouveau capteur, le serveur exécute le processus de pré-distribution pour ce dernier, puis informe les capteurs déployés choisis (le serveur peut connaître les emplacements réels des capteurs déployés, dans ce cas, il utiliseras leurs localisations au lieu de leurs emplacements prévus pour sélectionner les nœud voisins) sur les clés par paire correspondantes au nouveau capteur à travers des liens sécurisés.

révocation de noeud : Pour révoquer un capteur, les autres capteurs qui

ont une clé par pair partagée avec celui-ci ont seulement besoin de retirer cette clé de leur mémoire.

D'autres travaux ont été proposés dans [3], basés sur ce schéma et hybridés avec d'autre tel que le schéma de base [21].

IV. T-secure

L'objectif est de résister aux attaques visant à compromettre tous les nœuds du réseau. Le principe est qu'un nombre inférieur ou égal à t de nœuds compromis de l'ensemble n des nœuds du réseau ($t < n$) ne permettra pas de compromettre tous le réseau. Ils s'appuient en majorité sur deux schémas cryptographiques particuliers : le schéma de Blom [7] et le schéma de Blundo et *al.* [8]. Pour construire les clés partagées, le schéma de Blom se base sur des matrices spéciales tandis que le schéma de Blundo et al se base sur l'évaluation des polynômes symétriques.

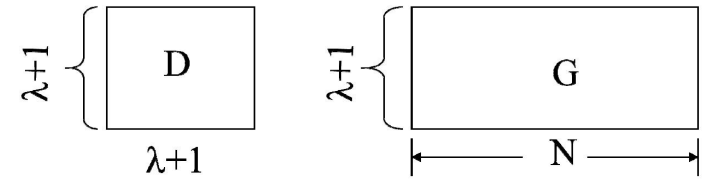
Plusieurs travaux ont combiné le modèle probabiliste avec les schémas de Blom et de Blundo et al. pour générer des clés secrètes aux moindres coût de chargement sur les nœuds capteurs. Les inconvénients sont, entre autres, l'impossibilité des nouveaux nœuds de vérifier l'identité des nœuds précédemment déployés, et dans les cas ou ces derniers sont compromis. Exemples de protocoles : [6], [15].

◇ Schéma de Blom

Toutes les clés possibles entre deux nœuds dans un réseau de la taille N peuvent être représentées dans une matrice de clés de taille $N \times N$. Blom [7] a proposé une méthode pour établir une clé symétrique distincte entre chaque paire de nœuds du réseau par des calculs matriciel. Une matrice à deux dimensions G de la taille $(\lambda + 1) \times N$, et une matrice symétrique D de la taille $(\lambda + 1)(\lambda + 1)$ sont construites à l'avance. La matrice D contient l'information privée et elle est maintenue secrète pendant toute la durée de vie du réseau. N est le nombre de nœuds et λ est le seuil prévu pour compromettre la sécurité de réseau, "λ-secure property" : si l'adversaire arrive à compromettre un nombre de nœuds inférieur ou égale à λ, le reste du réseau est sécurisé. Le cas où l'adversaire arrive à compromettre avec un taux supérieur à λ nœuds, toutes les paires de clés du réseau peuvent être calculées.

Une nouvelle matrice A est produite par $A = (D \times G)^T$ (transposé de $D \times G$), A a n lignes et $(\lambda + 1)$ colonnes.

Chaque nœud i stocke la i^{me} ligne de la matrice secrète A et de la i^{me} colonne de la matrice publique G. Après déploiement, chaque paire de nœuds i et j peuvent individuellement calculer la clé partagée entre eux $k_{ij} = k_{ji}$ en échangeant seulement leurs colonnes en claire, parce que la clé est le produit scalaire de leur propre ligne et les colonnes reçus des autres. Par exemple, le nœud i stock la i^{me}ligne de A et la i^{me} colonne de G, le nœud j stock j^{me}ligne de A et la j^{me} colonne de G.



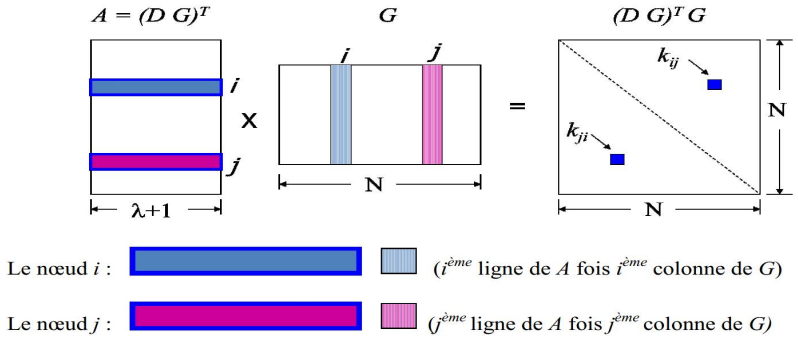
Matrice secrète $D_{(\lambda+1)(\lambda+1)}$ Matrice publique $G_{(\lambda+1)(N)}$

$A = (D \times G)^T$, et $K = D \times G$

Alors :

$K = D \times G = (D \times G)^T = G^T \times D^T \times G = G^T \times D \times G = G^T \times A^T = (A \times G)^T = K^T$

La matrice K est symétrique, l'entrée (i, j) de la matrice égale à l'entrée (j, i).



Après le calcul, le nœud i obtient Le schéma de Blom peut attribuer à chaque paire de nœuds dans un RCSF une clé (symétrique) directement, et il tolère la compromission de λ nœuds. Par conséquent, il exige $\lambda + 1$ d'espace mémoire et une diffusion d'un message de taille $\lambda + 1$, et une multiplication coûteuse de deux vecteurs de $\lambda + 1$ éléments.

V. Schémas de pré-distribution basé sur la station de base ou la distribution d'une clé maîtresse

V.1 L'emploie d'une station de base confiante pour envoyer les clés de session dans le but d'établir une communication entre deux nœuds quelconques, on a plus besoin que chaque nœud dans le réseau stocker $n-1$ clés.

◇ SPINS : " Security Protocols for Sensor Networks"

Perrig et *al.* [44] ont proposés SPINS, pour une architecture centralisée qui assume un arbre comme topologie du réseau. La racine de l'arbre est une station de base, et les nœuds de capteur forment le reste de l'arbre. SPINS comprend deux modules : Secure Network Encryption Protocol (SNEP) et la version micro du "Timed Efficient Streaming Loss-tolerant Authentication Protocol" (μTESLA).

○ SNEP : "Secure Network Encryption Protocol"

Le protocole SNEP [44] s'intéresse à la protection des communications entre un capteur et une station de base ou entre deux nœuds capteurs dans le réseau. Par hypothèse, chaque nœud i initialement partage une clé principale symétrique k_i avec la station de base ; à partir de cette clé sont déduites les clés ke_i, ka_i, la première clé est pour le chiffrement, et la deuxième est pour l'authentification. De plus, chaque noeud i partage un compteur CPT_i avec la station de base (un compteur pour chaque direction de communication) ; ce compteur est incrémenté à chaque envoi ou réception de paquet. L'utilisation du compteur permet au récepteur d'avoir une garantie quant à l'ordre des paquets reçus. Enfin, les capteurs ne partagent initialement aucun secret entre eux. Supposons que la station de base envoie une requête R au noeud i, le message envoyé est le suivant :

$$Station\,de\,base \rightarrow i : R, MAC(Ka_i, CPT_i||R)$$

L'utilisation de CPT_i protège le noeud i contre les rejeux de paquets ; car à chaque émission de paquet, le compteur est incrémenté des deux cotés. L'utilisation de MAC garantit au capteur i l'intégrité des paquets et leurs origines. Le capteur i émet la réponse R_i suivante :

$$i \rightarrow station\,de\,base : \{R_i\}_{Kei,CPT_i}, MAC(Ka_i, CPT_i||\{R\}_{Kei,CPT_i})$$

L'utilisation du compteur CPT_i dans le chiffrement de R_i offre une sé-

curité sémantique, c'est-à-dire le chiffrement du même paquet avec la même clé donne deux différents paquets chiffrés parce que le compteur change de valeur à chaque réception (émission), et rend difficile la tâche de l'adversaire qui aimerait avoir le texte clair à partir d'un texte chiffré. Si de plus, la station de base exige de tester l'état de fraîcheur du résultat, c'est-à-dire que le résultat retourné par un capteur vient en réponse à sa requête, alors il est possible d'intégrer dans la requête R un nombre aléatoire N généré par la station de base, il suffira de tester que la réponse fournie a bien pris en compte ce même nombre N. Du fait du caractère aléatoire de N, une réponse émise par un capteur prenant en compte N prouve que la réponse a bien été générée après la réception de la requête R_i. L'état de fraîcheur est donc bien garantie de la sorte. Les échanges deviennent donc :

$$station\,de\,base \rightarrow i : N, R, MAC(ka_i, N||CPT_i||R)$$
$$i \rightarrow \{R_i\}_{kei,CPT_i}, MAC(ka_i, N||CPT_i||\{R_i\}_{kei,CPT_i})$$

Lorsque deux nœuds i et j veulent communiquer en toute sécurité, il est tout d'abord nécessaire de mettre en place un secret principale partagé entre ces deux capteurs. Pour cela, la station de base joue le rôle du tiers de confiance en générant une clé k_{ij} et en la communiquant d'une manière protégée à chacun des capteurs.

SNEP offre la confidentialité, l'authentification, l'intégrité, et la fraîcheur des données, et nécessite peu de mémoire. Par contre, la station de base sur laquelle toutes communications reposent peut aussi faire l'objet d'une attaque de déni de service, ce qui aboutira à la paralysie du réseau. La taille du compteur CPT_i doit être suffisamment grande pour éviter sa répétition, sinon, le risque existe qu'un attaquant déduise des informations concernant le texte en clair à partir du texte chiffré.

$\circ\mu$TESLA : " Micro Time Efficient Streaming Loss-Tolerant Authentication"

Le protocole μTESLA [44] s'appuie sur le protocole TESLA est adapté aux capteurs. Il assure l'authentification des paquets émis par la station de base en diffusion sur le RCSF.

La station de base partage avec l'ensemble des capteurs une clé de groupe k_g. Cependant, μTESLA vise à authentifier l'origine des paquets émis par la station de base et à éviter qu'un nœud du groupe devenu malveillant n'usurpe l'identité de la station de base lors de l'émission d'un message.

Une liste chaînée de clés est générée k_g^n, k_g^{n-1},....., k_g^1, k_g^0, de telle sorte que $k_g^{k+1} = F(k_g^k)$, où F est une fonction de hachage irréversible. Chaque capteur est initialisé avec la clé k_g^0 avant le déploiement du réseau. Cette clé est encore connue comme clé de base "commitment key". De plus, chaque capteur i partage une clé symétrique principale k^i avec la station de base qui permet de s'authentifier mutuellement (avec la clé k_i^a).

Les capteurs authentifient les paquets en deux étapes ; pour cela, le temps est décomposé en intervalles T. Dans la première étape, la station de base diffuse les paquets authentifiés P1, P2,... avec la clé k_g^k (k correspondant à l'intervalle de temps choisi pour émettre), ces paquets sont conservés sans traitement par les capteurs qui ne peuvent pas encore vérifier leur provenance car ils ne possèdent pas la clé d'authentification k_g^k. En effet, ils ne connaissent que la clé k_g^{k-1} et du fait de la propriété de la fonction F irréversible, ils ne peuvent pas déduire k_g^k.
Dans une seconde étape, la station de base diffuse la clé d'authentification k_g^k dans l'intervalle de temps k + $\delta(\delta \geq 1)$, les capteurs vérifient alors que $k_g^{k-1} = F(k_g^k)$ puis ils vérifient que les paquets précédemment envoyés dans l'intervalle k sont correctement authentifiés. Notons que la station de base doit être sûre que tous les paquet sont arrivés à destination des capteurs avant de divulguer la clé d'authentification, sinon un nœud malveillant bien positionné pourrait forger des paquets signés avec cette clé et inonder le réseau, et les capteurs n'auraient alors aucun moyen de distinguer les informations en provenance de la station de base de celles forgées par un noeud malveillant.

Comme pour SNEP, la solution de protection des échanges entre capteurs s'appuient sur la relation de confiance existante entre la station de base et chacun des capteurs. Un capteur diffuse ces paquets à travers la station de base qui se charge alors de les diffusés dans le réseau comme décrit précédemment. Cette diffusion consomme de l'énergie car les capteurs sont beaucoup sollicités. μTESLA exige un espace mémoire supplémentaire dans les capteurs pour stocker les paquets non authentifiés jusqu'à la réception de la clé d'authentification.

V.2 Le pré-chargement d'une clé maîtresse qui est utilisé pour la génération des clés de réseau

◇ **BROSK : " BROadcast Session Key Negotiation Protocol "**

Brosk [30] est un nouveau protocole : chaque nœud peut négocier une

clé de session avec ses voisins par la diffusion d'un message de négociation de clé. Brosk utilise un système entièrement ad-hoc pour négocier la clé de session et peut effectuer ce processus de négociation de clé efficacement.De plus l'évolutivité de Brosk est significative en particulier lorsqu'il est appliqué à un réseau dense.

Hypothèses

Les nœuds sont statiques ou ont une faible mobilité : en fait, dans de nombreuses applications, les nœuds sont fixés dans une position pour l'ensemble de durée de vie du réseau.

Les nœuds partagent une clé maîtresse : chaque nœud du même réseau partage une clé maîtresse qui ne sera jamais communiquée. C'est la clé avec laquelle le nœud peut dire si un autre nœud est dans le même réseau ou non. Cette clé sera utilisée pour authentifier les autres nœuds et négocier la clé de session. La clé maîtresse doit être conservée en secret et elle ne pourra jamais être compromise malgré la capture du nœud.

Diffusion de message de négociation de clé : un nœud capteur va essayer de négocier une clé de session par la diffusion de message de négociation de clé . Chaque nœud tente de diffuser le message suivant : ID_A $|N_A$ $||$ MAC_K $(ID_A \mid N_A)$.

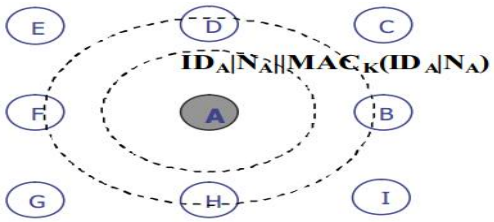


FIGURE 2.7 – *Le noeud* ***A*** *diffuse un message de négotiation de clés [30].*

Une fois qu'un nœud reçoit le message d'introduction diffusé par son voisin, il peut construire la clé de session partagée en générant le MAC de deux nonces. Pour Ainsi, sur la Figure 2.8, le nœud B reçoit le message diffusé à partir du nœud A. le nœud A reçoit aussi le message de diffusion du nœud B (l'équation 2.1). Ils peuvent utiliser K_{AB} comme clé de session partagée (l'équation 2.2).

$$ID_B|K_B||MAC_K(ID_B|N_B) \tag{2.1}$$

$$K_{AB} = MAC_K(N_A|N_B) \tag{2.2}$$

Renégocier la clé : lorsque le réseau de capteurs travaille depuis un certain temps, les nœuds risque de manquer de clés de session. Par conséquent, les nœuds de réseau de capteurs ont besoin de renégocier de nouvelles clés de session.

2.3.3 Cryptographie Asymétrique

Le schéma de gestion de clés asymétriques est une technique composée d'une paire de clés (privée, publique) qui assure une communication sécurisée. Son principe étant : qu'avant le déploiement, chaque nœud a la clé maitresse publique et privée (K_M, K_M^{-1}), puis chaque nœud A génère sa paire de clés (K_A, K_A^{-1}). Après le déploiement, les nœuds échangent les clés : échangent des clés publiques et une signature par la clé maitresse pour la vérification des clés publiques reçues. Par la suite, une clé symétrique peut être générée et échangée entre les nœuds encryptées par leurs clés publiques. Cette technique requière des clés de taille importante pour garantir le maintien de la sécurité, c'est pour cela qu'elle ne convient pas pour les RCSF [39, 47].

2.3.3.1 Schémas de gestion de clés basé sur PKI

Munivel et Ajit [43] propose micro-PKI (Micro Public Key Infrastructure), une version simplifiée des PKI conventionnelles. La station de base possède une clé publique et une autre privée. La clé publique est utilisée par les nœuds du réseau pour authentifier la station de base, et la clé privée est utilisée par la station de base pour déchiffrer les données envoyées par les nœuds.

Avant le déploiement, la clé publique de la station de base est stockée dans tous les nœuds. Les auteurs incluent dans leur méthode deux types d'authentification. Le premier type d'authentification se fait entre un nœud du réseau et la station de base, le nœud génère une clé symétrique de session et la chiffre avec la clé publique de la station de base. La clé chiffrée est transmise à la station de base sans être déchiffrée en chemin puisque les nœuds ne connaissent pas la clé privée de la station de base. À la réception, la station de base déchiffre la clé de session et la stocke dans une table. Le deuxième type d'authentification se déroule entre tous couples de nœuds du réseau en passant par la station de base qui joue le rôle de l'authentificateur entre eux. L'un des deux nœuds envoie une requête à la station de base

contenant l'identifiant de nœud destinataire. À la réception, la station de base génère une clé aléatoire et la chiffre avec la clé de session correspondante au nœud émetteur de la requête [36].

2.3.3.2 Schémas de gestion de clés basé sur ECC

ECC est une méthode d'échange de clés proposée indépendamment par Koblitz [27] et Miller [42]. Afin d'assurer un niveau de sécurité appréciable en utilisant un système ECC (Elliptic Curve Cryptosystem), il faut trouver un problème difficile à résoudre comme le problème de la factorisation d'un produit en ses facteurs premiers, utilisé par le système RSA [46]. Pour cela, nous considérons l'équation suivante : $Q = [k]\ P$ où $Q, P \in E_p$ et $k < p$. Il est aisé de calculer Q connaissant k et P mais il est très ardu et voire impossible pour un k assez grand, de déterminer k sachant P et Q. Ce problème est le problème du logarithme discret, très difficile à résoudre dans le contexte des courbes elliptiques. La cryptographie sur les courbes elliptiques se base sur ce problème pour l'échange de clés, le chiffrement et même la signature des messages entre deux entités communicantes. ECC reposant sur des clés de tailles réduites, des temps de traitement raisonnables et des capacités de stockage moins importantes [45].

Nous présentons la cryptographie basée sur les courbes elliptiques comme suit : soient F_q un corps fini à q éléments et $\overline{F_q}$ la clôture algébrique de F_q , c'est-à-dire que tout polynôme de degré supérieur ou égal à 1, à coefficients dans $\overline{F_q}$ admet au moins une racine dans $\overline{F_q}$. Une courbe elliptique E est l'ensemble des couples ou points $(x, y) \in \overline{F_q} \times \overline{F_q}$ vérifiant l'équation 2.1.

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\ ,\ avec \quad a_i\ \in\ F_q \tag{2.3}$$

L'équation 2.1 est connue sous le nom de l'équation de Weierstrass sur le corps F_q. Pour leur usage en cryptographie a_1, a_2 et a_3 doivent avoir une valeur nulle. Avec les cryptographes a_4 est devenu a et a_6 est devenu b, d'où l'équation 2.2 assortie d'une condition portant sur les coefficients.

$$y^2 = x^3 + ax + b\ ,\ avec \quad 4a^3 + 27b^2\ \neq\ 0 \tag{2.4}$$

La courbe elliptique utilisée dans la cryptographie est constituée d'un ensemble de points (x, y) avec un point particulier nommé le point à l'infini ϑ, $E = \{(x, y) \in \overline{F_q}\ \times\ \overline{F_q}\}\ \cup\ \{\vartheta\}$. L'ensemble des points de (E) forme un groupe abélien (groupe dont la loi de composition interne est commutative) par rapport à la loi d'addition, qui est une loi spécifique pour les courbes elliptiques, expliquée notamment dans [28]. Le point va servir comme l'élément

d'identité ou d'élément neutre de (E). Notons que souvent le corps est choisi d'une façon telle que $q = p^m$, p étant un nombre premier ($p = 2$, typiquement) et m étant la taille de la clé de chiffrement ($m = 160$, typiquement).

La Figure 2.10 montre l'addition de deux points sur une courbe elliptique de la forme (E). Pour additionner les deux points p_1 et p_2 , nous traçons la droite qui passe par ces deux points. La droite coupe la courbe en un point p_3 . Puisque les trois points sont alignés alors nous obtenons p_1 + p_2 + p_3 = ϑ . Alors, p_1 + p_2 = -p_3. Il reste donc à calculer le point p_4, l'opposé de p_3 , pour obtenir l'addition. Dans le cas où p_1=$p_2$la droite qui passe par p_1 et p_2 est bien la tangente à p_1. Une autre opération importante est la multiplication scalaire. Par exemple, pour calculer 2.p_1 nous faisons l'addition p_1 + p_1 .

Soit k un entier positif, le calcul de k.p_1 est égale à $\underbrace{p_1 + ... + p_1}_{k}$. La constante k est considérée comme une clé privée et le point obtenu sera considéré comme la clé publique après avoir vérifié qu'il appartient à la courbe.

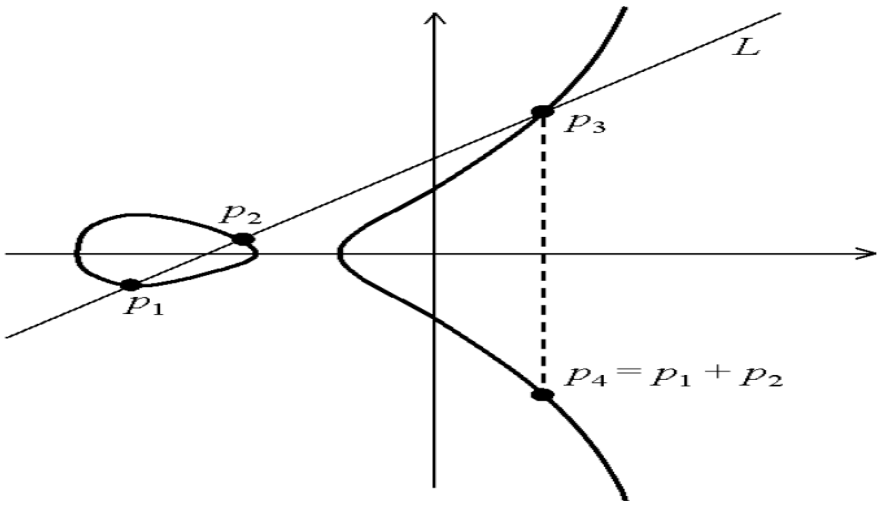


FIGURE 2.8 – *Addition de deux points sur une courbe elliptique [36].*

TinyECC [32] est une cryptographie sur les courbes elliptiques, ces dernières peuvent être utilisées pour des opérations asymétriques comme des échanges de clés sur un canal non sécurisé.

2.4 Comparaison

2.4.1 Critères de comparaison

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

- **Efficacité** : les limitations de mémoire et des communications des nœuds doivent être considérées
  - — Complexité en mémoire : quantité de mémoire nécessaire pour enregistrer les clés cryptographique.
  - — Complexité en communication : nombre et taille des messages échangés pour la gestion des clés.
- **Connectivité** : en terme de clé "key connectivity", probabilité que deux nœuds (ou plus) partagent une clé.
- **Scalabilité** : On dit qu'un réseau a une bonne scalabilité quand il peut être facilement augmenté le nombre de nœud et qu'il n'y aucun problème lors de l'ajout d'un nouveau nœud au réseau car il est intéressent que le réseau assure le bon fonctionnement quel que soit le nombre de noeuds.
- **Résilience** : ou résistance contre la capture de nœud, cette métrique mesure comment le RCSF est compromis quand un nœud est compromis, et l'influence de ce nœud sur la sécurité du réseau en entier.
- **Révocation** : cette possibilité d'enlever un nœud s'il est inaccessible (panne,énergie esquissée) ou compromis.

2.4.2 Tableau comparatif

Établissement de clé est la principale primitive cryptographique dans toutes les applications où la sécurité est la principale préoccupation. La gestion de clés est la méthode la plus efficace pour assurer une meilleure sécurité contre plusieurs types d'attaques. Toutes les clés ont leurs propres limites et avantages selon la zone dans laquelle elles sont déployées. Le tableau 2.1 résume une comparaison des différentes méthodes de gestion de clés proposés dans la littérature, pour la construction de ce dernier, nous nous sommes basés sur les travaux [1, 5, 10, 49, 51].

Schémas de gestion de clés	Complexité en mémoire	Complexité en communication	Connectivité	Résilience	Scalability
Key infection	Dépend du nombre de voisins (d) à un saut	Pour chaque nœud : $2 \times d$	Forte	Faible	Bien
LEAP	$3 \times d + 2 + L$	$2 \times d + 1$	Forte	Très bien avant T_{min}	Bien
PIKE	$2(\sqrt{N} - 1)$	Proportionnel à $\sqrt{N}$	Forte	Faible	Non
Jolly et *al.*	• Nœud : 2 • Passerelle : $\mid S \mid +1$ • Nœud commande : $\mid G \mid + \mid S \mid$	Proportionn-elle aux passerelles et aux nombre de nœuds du cluster	• $S-S$: communication à travers la passerelle • $G-G$: communication à travers un nœud	Faible	Non
Eschenauer et Gligor	$m + ID$	$d + 1$	p	Dépend de m et p	Bien
Chan et al	$m' > m$	$d + 1$	$p' < p$	Dépend de m' et p'	Moyen
Key management	$d - 1$	$d + 1$	Dépend de pdf	Moyen	Bien
LBK	$5(\lambda + 1)$	$2 \times d$	p_r	λ-secure	Bien
Blom	$2(\lambda + 1)$	$d + 1$	Forte	λ-secure	Moyen
SPINS	k	3*(N/2)	Forte	Faible	Bien
BROSK	1	$2 \times d$	Forte	Très Faible	Très bien
Micro-PKI	• $SB : 2 + N$ • $noeud : 2$	• $SB : 2 \times N$ • $noeud : 2 \times Nbcom$	Faible	Moyen	Bien

TABLE 2.1 – Tableau comparatif.

N, S : Nombre de nœuds dans le réseaux.
d : Nombre de voisins d'un noeud.
L : Chaine de clés a sens unique pour $\mu TESLA$.
G : Ensemble de passerelles dans le réseaux.
m, m' : Tailles du trousseaux de clés.
p, p', pr : Une probabilité.
λ : Nombre de nœuds compromis dans le réseaux.
ID : Identificateur de clés.
$Nbcom$: Nombre de communications.

2.4.3 Discussion

A partir de ce tableaux, en constate que le défi dans les protocoles de gestion de clés probabiliste, est de trouver un compromis entre la taille de l'ensemble P et la taille du trousseau de clés pré-chargé dans la mémoire de chaque nœuds, dont le but est d'assurer une connectivité et une résilience meilleure et optimise en mémoire. On a déduit ainsi que le protocole géographique LBK permet le passage à l'échelle, néanmoins le coûts des composants supplémentaires (GPS, etc.) influence leur application, de plus, la sécurité produite dépend du nombre de nœuds compromis. Concernant le protocole blom qu'a l'éloge d'être scalable et a une connectivité totale, cependant la résilience contre les attaques de capture dépend du nombre de nœuds compromis. On note aussi que les protocole déterministes ont l'avantage d'être simple à implémenter et offrent une connectivité total du réseau en raison de sa certitude, toutefois ils présentent deux inconvénients majeurs, l'espace mémoire important pour stocker les clés de cryptages et une faible résilience contre les différentes attaques, en particulier, le protocole LEAP caractérisé par une bonne résistance aux attaques, une forte connectivité, et permet l'ajout de nouveaux nœuds au réseau sans omettre qu'il est coûteux en terme d'espace mémoire et nécessite l'échange d'un grand nombre de messages pour établir toutes les clés cryptographique ce qu'épuisent la batterie des capteurs .

Conclusion

Dans ce chapitre nous avons passé en revue quelques protocoles de gestion de clés dans les RCSF, en les classant suivant la technique cryptographique utilisée, la façon avec laquelle les nœuds voisins partagent des clés communes (probabiliste ou déterministe, etc.avec), etc. Ensuite, nous avons comparé et discuté les protocoles présentés selon des métriques bien définies. Nous avons

jugé que le protocole LEAP proposé par Zhu et *al.* est un protocole trés interessant c'est pour cela que nous l'avons selectionné pour mieux l'étudier et lui apporter d'éventuelles amélioration.

Le chapitre suivant sera dédié à notre contribution.

Chapitre 3

Proposition et Simulation

Introduction

Notre étude de l'art effectuée dans le chapitre précédent a décelé que l'approche la plus appropriée pour assurer des communications sécurisées au sein d'un réseau de capteurs sans fil est l'établissement d'un protocole de gestion de clés basé sur la cryptographie symétrique et plus précisément sur la pré-distribution d'une clé maitresse.

Dans ce chapitre, nous allons présenter notre contribution pour la gestion de clés dans les réseaux de capteurs sans fil. Nous commencerons d'abord par présenter nos motivations, ensuite nous énonçons les détails de notre contribution ainsi que l'évaluation de ses performances.

3.1 Motivation

Dans notre travail, nous nous sommes intéressés à la gestion de clés dans les RCSF. Le nombre et la taille des paquets de données échangés durant les phases de déroulement du protocole (découverte de voisinage, installation des clés, renouvellement des clés et insertion des nouveaux nœuds) influence explicitement sur la consommation d'énergie par les nœuds capteurs qui est le concept majeur déterminant la durée de vie d'un réseau de capteurs sans fil. C'est pour cette raison que la plupart des solutions proposées essayent davantage de réduire le nombre de messages émis et reçus et la quantité d'énergie consommée dans les opérations cryptographiques durant tout le cycle de vie du réseau.
Après la discussion qui a été faite, nous avons vu que le protocole *LEAP* proposé par Zhu et *al.* est particulièrement une solution très intéressante puisque il assure une forte connectivité des nœuds du réseaux ainsi qu'une

bonne résilience.

3.2 Schéma proposé

Dans ce qui suit, nous allons présenter notre contribution qui consiste en une amélioration du protocole Localized Encryption and Authentication Protocol nommé Localized Encryption and Authentication Protocol based on Cluster ($LEAP-C$). Notre idée de base est d'essayer de réduire la complexité en mémoire et en communication liés à ce dernier, c'est pour cela qu'on a opté à la suppression de la clé par pair, c'est vrai qu'elle est la solution cryptographique la plus sécurisée, mais aussi la plus coûteuse en terme d'énergie et d'après ce qu'on a vu dans le chapitre 1, le but premier des capteurs n'est pas de comprendre les messages reçus mais bel et bien de collecter et de transmettre ces dernières vers une station de base.

Notre protocole se déroule en deux parties distinctes. Dans la première chaque nœud est pré-chargé avec deux clés, une clé individuelle partagée avec la SB et une commune à tout les nœuds du réseau. Dans la deuxième partie, après le déploiement, on commence par l'établissement d'une clé pour chaque groupe (*cluster*), par la suite la clé globale est régénérée et retransmise à chaque nœud par l'intermédiaire de la clé de *cluster*. Les détails sur la manière de construction et de renouvellement de clés sont définis dans ce qui suit.

3.2.1 Hypothèses

LEAP-C est fondé sur les hypothèses suivantes :

- La station de base dispose d'une capacité de calcule et de stockage illimité.
- Le réseau est statique (c'est-à-dire les nœuds ne peuvent changer leurs coordonnées initial a aucun moment).
- Les nœuds capteurs sont homogènes (c'est-à-dire les nœuds sont similaire dans leurs capacité de traitement, d'énergie et de stockage) et peuvent être compromis (les informations stockés dans leurs mémoire sont connu par l'attaquant).
- Déploiement aléatoire (les voisins de n'importe quel nœuds ne sont pas connu avant le déploiement).
- Les nœud sont regroupés en cluster.

3.2.2 Notation

On utilise la notation suivante dans notre protocole LEAP-C :

Notation	Description
S_i	Dénote l'identificateur unique d'ième nœud capteur dans le réseau.
K_{SB,S_i}	La clé individuelle partagée entre la station de base est l'ième capteur de réseaux.
f_K	Une fonction pseudo-aléatoire f est chiffrée avec la clé symétrique K.
$MAC_K(M)$	Le Message Authentification Code du message M avec la clé symétrique K.
CPT	Un compteur incrémenté à chaque émission ou réception d'un message de gestion de clés, et il est pré-chargé.
N_i	Nonce générer par l'iéme nœud capteur.
$hash_K(M)$	Une fonction de hachage à sens unique appliquée à la chaine de caractère M en utilisant la clé K.

TABLE 3.1 – Les différentes notations utilisées dans la solution proposée

3.2.3 Phases du LEAP-C

La solution proposée se compose de quatre phases importantes qui permettent une bonne gestion de clés échangées et aussi une meilleur sécurité des données et des clés. Les phases sont bien détaillées ci-dessous.

3.2.3.1 Établissement de la clé individuelle

Chaque nœud S_i du réseau possède une clé individuelle partagée avec la SB. Cette clé est pré-chargée dans la RAM de chacun des nœuds avant le déploiement. La clés individuelle K_{SB,S_i} est générée en utilisant une fonction pseudo-aléatoire de la manière suivante : K_{SB,S_i}=$f_{K_{IN}}(S_i)$.

3.2.3.2 Établissement de la clé globale

Avant le déploiement, une clé similaire K_g^i est pré-chargée dans la mémoire de chaque nœud. Cette dernière est employée pour envoyer un message à tout le réseau d'une manière sécurisée. C'est la solution la moins coûteuse en terme d'énergie et de l'éspace mémoire, cependant la sécurité fournie est insuffisante. C'est pour cela qu'on doit faire appelle à un système efficace de renouvellement de clés périodique ou à la demande (re-keying).

3.2.3.3 Établissement de la clé de cluster

Après le déploiement et l'execution de l'algorithme de formation de cluster, qui consiste à partitionner le réseau en un certain nombre de clusters (groupes), plus homogènes selon une métrique spécifique ou une combinaison de métriques, et former une topologie virtuelle[22], le C génère une clé de cluster $K^j_{Cluster}$ et il diffuse un message contenant cette clé à tout les membres de son cluster en utilisant la clé du réseau. Le message diffusé est authentifie en utilisant le MAC (Message Authentification Protocole).

$$S_{CH} \rightarrow * : \{S_{CH}, K^j_{Cluster}, MAC_{K_g}(K^j_{Cluster}, CPT + 1)\}_{K_g}$$

3.2.3.4 Renouvellement de clés

Il est important de pouvoir changer régulièrement les clés afin de limiter le nombre de liens compromis à un instant donné.

+ Renouvellement de la clé globale

La station de base génère une nouvelle clé global K^{i+1}_g, et cela est périodiquement ou après la compromission d'un nœud, cette clé est produite en utilisant une fonction pseudo-aléatoire, la SB se serve des CH_s comme distributeur de la nouvelle clé. La SB utilise les clés individuelles quelle partage avec les CH_s pour leurs diffuser un message contenant K^{i+1}_g , à leurs tours, chaque CH emploie sa clé de cluster pour re-diffuser le message provenant de la SB a tout les membres de son cluster.

+Renouvellement de la clé de cluster

Le cluster-head génère un nonce N_{CH} et calcule la nouvelle clé du cluster de la manière suivante :

$K^{j+1}_{Cluster} = hash_{K^j_{Cluster}}(K^j_{Cluster}||N_{CH})$.

Et en fin authentifie le résultat en utilisant le MAC comme suit :

$S_{CH} \rightarrow *\{S_{CH}, K^{j+1}_{Cluster}, MAC_{K^j_{Cluster}}(K^{j+1}_{Cluster}, CPT{+}1)\}_{K^j_{Cluster}}$ et il l'envoie à tout les nœuds de son groupe. A la réception du message par un nœud du même groupe que le cluster-head, ce dernier déchiffre et vérifie le message reçu en utilisant sa clé de cluster $K^j_{Cluster}$.

3.3 Exemple illustratif

Pour mieux expliquer et illustrer le fonctionnement de notre schéma on considère l'exemple de la Figure3.1 : soient les noeuds S_1, S_2et une station de base (SB)

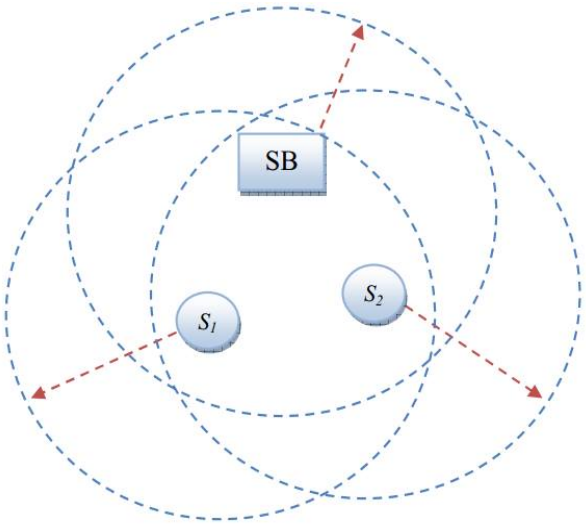


FIGURE 3.1 – *Exemple illustratif de LEAP-C.*

3.3.1 Avant le déploiement

3.3.1.1 Établissement de clés individuelles

Avant le déploiement la station de base génère une clés maitresse K_{IN} puis calcule sa clé individuelle partagée avec S_1 (respectivement S_2) comme suit :

S_1 :
$K_{SB,S_1} = f_{K_{IN}}(S_1)$

S_2 :
$K_{SB,S_2} = f_{K_{IN}}(S_2)$

Par la suite, la clé K_{SB,S_1} (respectivement K_{SB,S_2}) est pré-chargée dans la mémoire de S_1 (respectivement S_2) .

3.3.1.2 Établissement de la clé globale

La SB s'occupe de la génération d'une clé K_g^i qui est commune à tout les nœuds du réseaux, ensuite cette dernière est pré-chargée dans la mémoire des nœuds S_1 et S_2.

3.3.2 Après le déploiement

Après avoir établi toutes les clés individuelles et la clé globale, les capteurs emploient un algorithme de clustering et ceci afin de réunir les capteurs en groupes et d'élire un CH. Compte tenu des problèmes d'efficacité et de sécurité, nous avons adopter l'algorithme proposée dans [22]. Après les CHs sont élus, chaque CH stocke les identités de tout les nœuds de son cluster en plus de la clé global et sa clé individuelle.

3.3.2.1 Établissement de a clés de cluster

Après avoir appliquer un algorithme de formation de cluster on obtient le réseau présenté dans la figure suivante : Prenant un cluster de ce réseau on le dénote par CH_1.

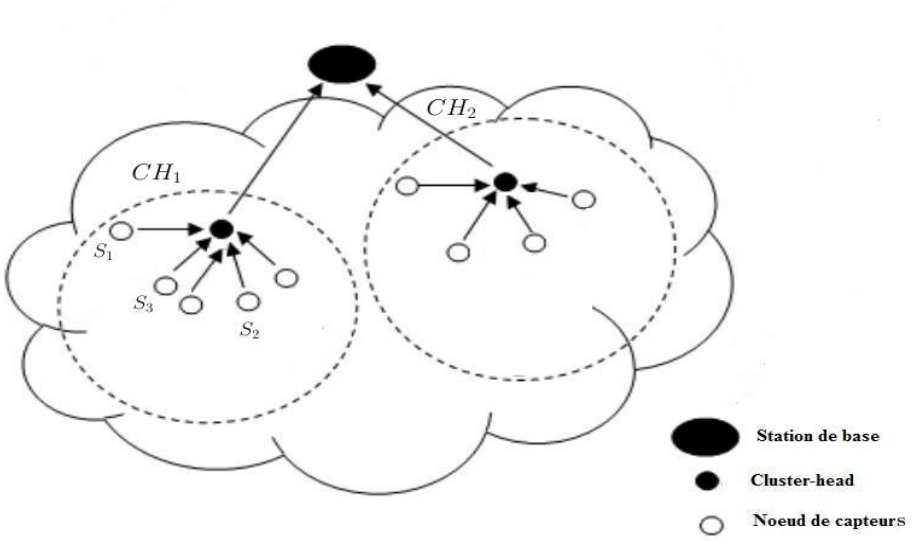


FIGURE 3.2 – *Réseau regroupé en cluster.*

Pour établir la clé de cluster on suit les phases ci-dessous :

- **Phase 1 :** soient les nœuds S_1, S_2, S_3 appartenant au CH_1 possèdent un compteur CPT.
- **Phase 2 :** le cluster head CH de CH_1 génère une clé $K^j_{cluster}$.
- **Phase 3 :** le CH effectue le calcul suivant : $CPT = CPT + 1$.
- **Phase 4 :** le CH calcul le MAC de message ($K^j_{cluster}$, $CPT + 1$) et de la clé global K^i_g.

- **Phase 5 :** le CH diffuse un message au membre de son groupes contenant trois champs : son identifiant, la clé $K^j_{cluster}$ et le MAC calculé, et tout le message est chiffré avec la clé K^i_g :

$$S_{CH} \rightarrow *\{S_{CH}, K^j_{cluster}, MAC_{K^i_g}(K^j_{cluster}, CPT+1)\}_{K^i_g}$$

- **Phase 6 :** quand un nœud reçoit ce message, tout d'abord il déchiffre avec sa clé global donc il obtient l'identificateur de CH, la clé $K^j_{cluster}$ et le MAC. Pour qu'il puisse authentifier le message reçu il calcul a son tours le MAC pour ce faire il doit incrementer son CPT (CPT+1) ensuite il calcule $MAC_{K^i_g}$ $(S_{CH}, CPT+1)$ puis il compare le résultat obtenu avec celui qu'il viens de recevoir, s'ils sont égaux alors il considère $K^j_{cluster}$ comme sa clé de cluster.

3.4 Simulation

La simulation informatique, ou simulation numérique, est une série de calculs effectués sur un ordinateur et reproduisant un phénomène physique. Elle aboutit à la description du résultat de ce phénomène, comme s'il était réellement déroulé. Cette représentation peut être une série de données, une image ou même un film vidéo. un simulateur peut réagir a des modification de paramètres et modifier ces résultats en conséquence [52].

Afin de simuler notre protocole, nous avons utiliser l'environnement MATLAB, qui est un langage de haut niveau et un environnement interactif pour le calcul numérique, la visualisation et la programmation. En utilisant MATLAB, on peut analyser les données, développer des algorithmes, et créer des modèles et des applications. La lange, les outils et les fonctions intégrées de mathématique permettent d'explorer des approches multiples et parvenir à une solution plus rapide qu'avec des tableurs ou des langages de programmation traditionnels,tel que C/C++ ou java. MATLAB est utilisé pour une gamme d'applications, y compris le traitement du signal et de la communication, l'image et le traitement video, les systèmes de controle, de test e de mesure, finance computationnelle, et la biologie computationnelle [35].

3.4.1 Environnement de simulation

Notre modèle d'expérimentation est établi sur 100 nœuds et une station de base, les nœuds sont dispersés aléatoirement sur une surface carré de 100 * 100 m. Les paramètres de notre simulation sont résumés dans le tableau suivant :

Paramètre	valeur
Localisation de SB	(x =50, y=175).
Le nombre de nœuds	100.
Énergie initial du réseaux	1 Joul.
La taille des paquets	128 Bytes.
La valeur d'énergie électronique	0.00000005 Joul.
La valeur d'énergie d'amplification	0.0000000000000013 Joul.
La valeur de rayon de couverture	183.
La valeur de rayon de communication	100.

TABLE 3.2 – Les paramètres de simulation.

3.4.2 Résultats de simulation

Afin d'évaluer les performances de LEAP-C, LEAP, et OTMK nous avons utilisé les métriques suivantes :

3.4.2.1 Énergie résiduelle des nœuds par rapport à la taille de la clé

La taille de la clé est un critère très important dans l'évaluation de performance d'un protocole de gestion de clés, d'où on a évalué l'énergie résiduelle du réseau tout en variant la taille de la clé.

La Figure 3.3 montre que LEAP-C est mois coûteux en terme d'énergie que le protocole LEAP et le protocole OTMK quelque soit la taille de clés et cela est évident puisque notre protocole réduit le nombre de clés qui implique une réduction au niveau de messages échangés.

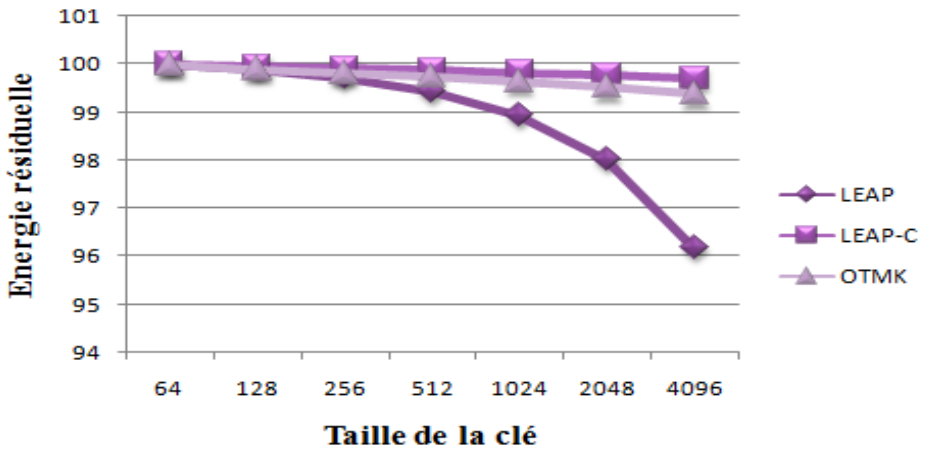


FIGURE 3.3 – *Énergie résiduelle des nœuds par rapport à la taille de la clé des trois protocoles.*

3.4.2.2 Espace mémoire

La mémoire est une ressource critique dans les réseaux capteur sans fil donc on est dans l'obligation de faire très attention et de réduire son utilisation le plus possible. Nous avons calculé le stockage de clés en faisant varier la taille des clés cryptographiques.

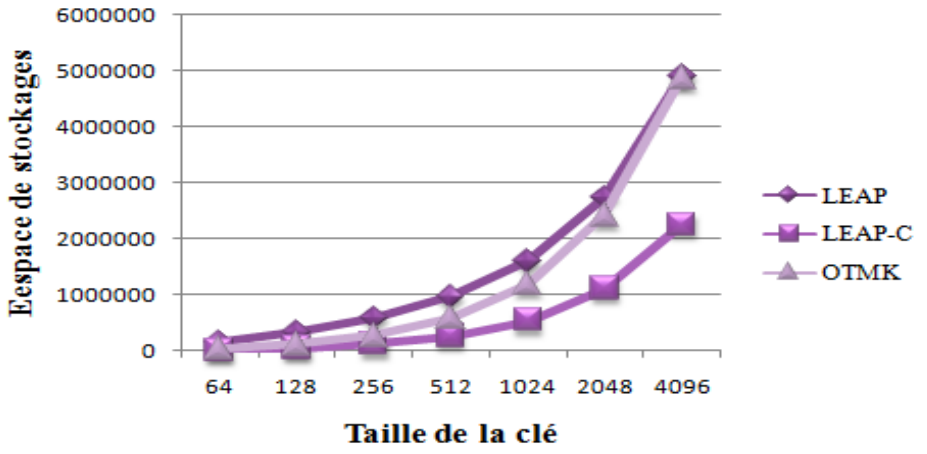


FIGURE 3.4 – *L'espace de stockage par rapport à la taille de la clé des trois protocoles.*

D'après les résultats obtenu dans la Figure 3.4 nous remarquons que l'espace mémoire requis pour le stockage de toutes les clés par LEAP-C est nettement beaucoup moins que celui du protocole LEAP et OTMK, et ce, grâce à la suppression de la clé par paire.

3.4.2.3 Énergie résiduelle des nœuds par rapport aux rounds

La ressource énergétique détermine la durée de vie du réseau et doit être soigneusement prise en compte dans la conception de n'importe quelle application dans les RCSF. Pour cette raison, nous avons comparé l'énergie résiduelle de LEAP-C, LEAP et OTMK en fonction de rounds comme l'illustre la figure suivante :

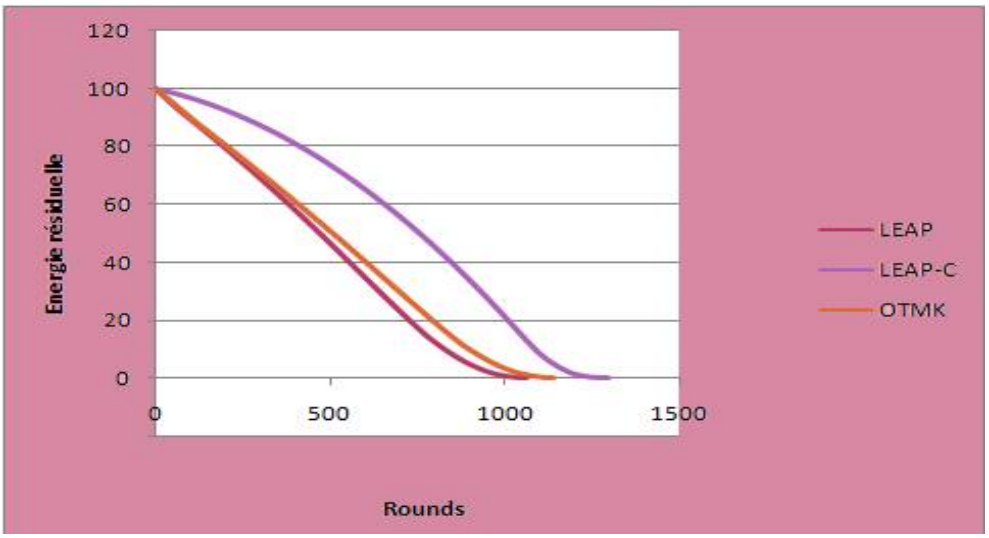


FIGURE 3.5 – *Énergie résiduelle des nœuds par rapport au round des trois protocoles.*

De la Figure 3.5 nous remarquons que l'énergie résiduelle du réseau dans le cas d'utilisation de LEAP-C est supérieur à l'énergie résiduelle du réseau dans le cas d'utilisation du protocole LEAP ou du protocole OTMK. Donc on peut dire que LEAP et OTMK requière plus d'énergie et cela est dû au nombre de messages qui doivent être émis/reçus pour que tout les nœuds du réseaux puissent avoir les quatre types de clés et pouvoir communiquer en toute sécurité, de ce fait la durée de vie du réseau en utilisant notre protocole est beaucoup plus importante par rapport à LEAP.

3.4.2.4 Complexité en communication

Nous avons calculé le nombre de messages émis et reçus des trois protocoles :

Messages émis

L'histogramme ci-dessous illustre le nombre de messages émis par les nœuds pour chaque round de simulation de notre protocole comparé avec LEAP et OTMK.

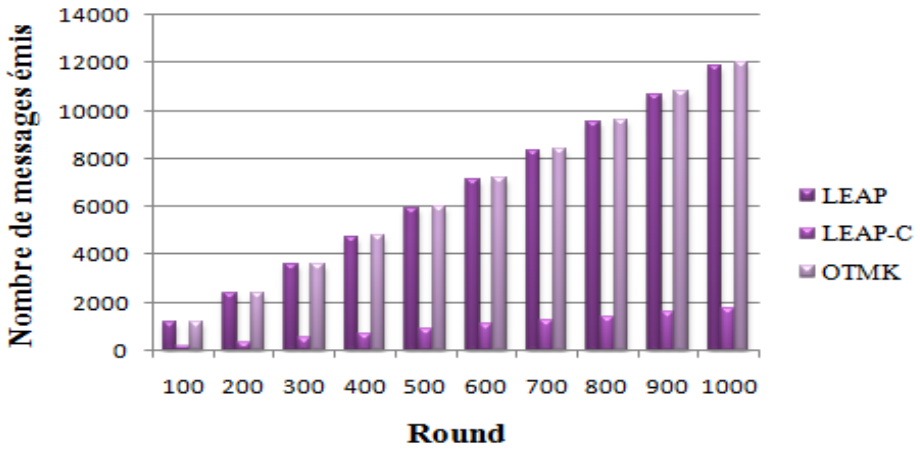


FIGURE 3.6 – *La complexité en communication en nombre de messages émis par les nœuds dans chaque round des trois protocoles.*

Messages reçus

Le nombre de messages reçus par les nœuds pour chaque round de simulation de notre protocole comparé avec LEAP et OTMK.

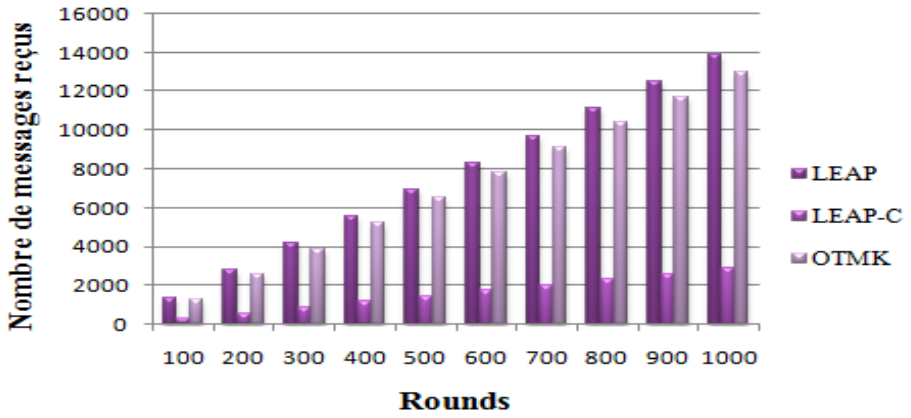


FIGURE 3.7 – *La complexité en communication en nombre de messages reçus par les nœuds dans chaque round des trois protocoles.*

Comme nous remarquons, le nombre de messages échangés dans le réseau en utilisant notre protocole est beaucoup moins important que celui en utilisant le protocole LEAP.

3.4.2.5 Passage à l'échelle

D'après le graphe suivant qui spécifie le passage à l'échelle d'un RCSF, en évoluant le nombre de nœuds, on observe que dans chaque ajout de nœuds la durée de vie du réseau diminue systématiquement, cela est causé par le nombre croissants de messages échangés entre les capteurs. Cependant LEAP-C reste au dessus des deux autres ceci montre que ce dernier peux être utilisé même si le nombre de nœuds dépasse 1000 nœuds.

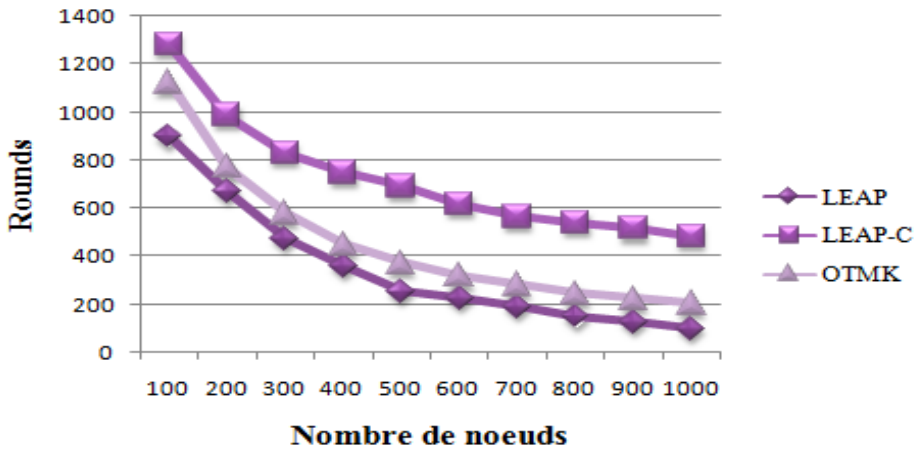


FIGURE 3.8 – *Passage à échelle en fonction du round*

Conclusion

Dans ce chapitre nous avons présenté notre contribution qui consiste en un protocole de gestion de clés dans les réseaux de capteurs sans fil ainsi l'évaluation de ces performance. Les résultats de simulation qui ont démontré que notre protocole est meilleur que le protocole LEAP et OTMK, et ce, en terme d'énergie, de stockage et de communication.

Conclusion générale

Depuis quelques années, les avancées technologiques en termes de miniaturisation des machines et des supports de communication y afférant ont rendu envisageable le déploiement et l'exploitation de milliers de capteurs, organisés en réseau ad hoc. D'ailleurs, selon le MIT, les réseaux de capteurs ont été identifiés comme l'une des dix technologies clefs de l'avenir et ce en raison de l'incroyable potentiel applicatif qu'elle renferme.

Si les perspectives d'utilisation des réseaux de capteurs sont claires et attrayantes, les problématiques qu'engendrent ces réseaux n'en sont pas moins nombreuses. A priori, ils ne dépendent d'aucune infrastructure et les capteurs n'ont aucune information relative au réseau auquel ils appartiennent. De plus, étant construits de façon ad hoc, ces réseaux doivent être auto-organisés. Dans ce mémoire, nous avons mis en avant les caractéristiques essentielles des réseaux de capteurs sans fil, ainsi que les besoins et les défis de la sécurité dans ces derniers. Nous avons étudiée aussi quelques schémas de gestion de clés qui permettent d'offrir le service de sécurité de base pour n'importe quel système basé sur la communication. L'ensemble des protocoles de gestion de clés proposés pour les RCSF, se basent principalement sur la cryptographie à clé symétrique et la méthode de pré-distribution de clés afin d'achever l'établissement de clés entre les entités communicantes dans le réseau. Nous avons étudié un ensemble de ces protocoles de gestion de clés que nous avons classé selon trois grandes famille et cela on nous basons sur quelques références.

En effectuant cet état de l'art détaillé nous avant fini part déceler les manques et delà, résulte notre contribution consistant en une amélioration d'une solution de gestion de clés pour les RCSF. Notre idée de base est de trouver un compromis entre le niveau de sécurité à assurer et le respect des contraintes posées par ces réseaux. Notre solution montre à travers les résultats de la simulation qu'elle peut fournir un bon niveau de sécurité avec moins d'exigence que la solutions de base .Comme la plupart des solutions

de gestion de clés proposées ont été conçues pour des RCSF statiques, les RCSF mobiles connaissent actuellement une certaine fébrilité de recherche et de nombreuses applications s'y sont développées. Concevoir un protocole efficace de gestion de clés pour de telles applications demeure encore un domaine de recherche ouvert. Il serait donc plausible, comme perspective de notre travail, d'adapter notre proposition à une mobilité des nœuds.

Bibliographie

[1] S. Akhbarifar and A.M. Rahmani. A survey on key pre-distribution schemes for security in wireless sensor networks. *International Journal of Computer Networks and Communications Security*, Volume 2, Numéro 12, PP. 423–442, Decembre 2014.

[2] R. Anderson, H. Chan, and A. Perrig. Key infection : Smart trust for smart dust. *Proceedings of the 12th IEEE International Conference on Network Protocols (ICNP), IEEE*, PP. 206–215, 2004.

[3] F. Anjum. Location dependent key management using random key-predistribution in sensor networks. In *Proceedings of the 5th ACM workshop on Wireless security*, PP. 21–30, 2006.

[4] S. Athmani. Protocole de sécurité pour les réseaux de capteurs sans fil. *Mémoire de Magister en Informatique, Université Hadj Lakhder de Batna, Algérie*, Juillet 2010.

[5] S. Bala, G. Sharma, and A.K. Verma. Classification of symmetric key management schemes for wireless sensor networks. *International Journal of Security and Its Applications*, Volume 7, Numéro 2, PP. 117–138, 2013.

[6] C. Bekara and M. Laurent-Maknavicius. A new resilient key management protocol for wireless sensor networks. In *Information Security Theory and Practices. Smart Cards, Mobile and Ubiquitous Computing Systems. Springer*. PP. 14–26, 2007.

[7] R. Blom. Non-public key distribution. *Advances in Cryptology, Springer*, PP. 231–236, 1983.

[8] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung. Perfectly-secure key distribution for dynamic conferences. *Advances in cryptology—CRYPTO'92, Springer*, PP. 471–486, 1992.

[9] D.E. Boubiche. Une approche inter-couches (cross-layer) pour la sécurité dans les rcsf. *Thèse de Doctorat en Science en Informatique, Université Hadj Lakhder de Batna, Algérie*, 2013.

[10] S.A. Camtepe and B. Yener. Key distribution mechanisms for wireless sensor networks : a survey. *Rensselaer Polytechnic Institute, Troy, New York, Technical Report*, PP. 05–07, 2005.

[11] H. Chan and A. Perrig. Pike : Peer intermediaries for key establishment in sensor networks. In *INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings, IEEE*, Volume 1, PP. 524–535, November 2005.

[12] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *Symposium on Security and Privacy. In Proceedings, IEEE*, PP. 197–213, 11-14 Mai 2003.

[13] O. Cheikhrouhou. Sécurité des réseaux ad hoc. *Diplôme National d'ingénieur en Génie Informatique, Université de Sfax*, Tunisie, Juillet 2005.

[14] C. Chen, Z. Huang, Q. Wen, and Y. Fan. A novel dynamic key management scheme for wireless sensor networks. *4th IEEE International Conference on Broadband Network and Multimedia Technology (ICBNMT), IEEE*, PP. 549–552, 2011.

[15] Y. Cheng and D.P. Agrawal. Efficient pairwise key establishment and management in static wireless sensor networks. In *IEEE International Conference on Mobile Adhoc and Sensor Systems Conference, IEEE*, PP. 7, November 2005.

[16] Y. Cheng and D.P. Agrawal. An improved key distribution mechanism for large-scale hierarchical wireless sensor networks. *Ad Hoc Networks, Elsevier*, Volume 5, Numéro 1, PP. 35–48, 2007.

[17] J. Deng, C. Hartung, R. Han, and S. Mishra. A practical study of transitory master key establishment forwireless sensor networks. In *First International Conference on Security and Privacy for Emerging Areas in Communications Networks (SECURECOMM'05), IEEE*, PP. 289–302, 2005.

[18] J. Deng, C. Hartung, R. Han, and S. Mishra. A practical study of transitory master key establishment forwireless sensor networks. *First International Conference on Security and Privacy for Emerging Areas in Communications Networks, Secure Communication, IEEE*, PP. 289–302, Septembre 2005.

[19] M.M. Diouri. Réseaux de capteurs sans fil : routage et sécurité. *Diplôme d'ingénieur en Informatique de l'INSA, Université de Lyon*, France, 2010.

[20] W. Du, J. Deng, Y.S. Han, S. Chen, and P.K. Varshney. A key management scheme for wireless sensor networks using deployment knowledge.

In *INFOCOM 2004. Twenty-third Annual Joint conference of the IEEE computer and communications societies, IEEE*, Volume 1, 2004.

[21] L. Eschenauer and V.D. Gligor. A key-management scheme for distributed sensor networks. *Proceedings of the 9th ACM conference on Computer and communications security, ACM*, PP. 41–47, November 2002.

[22] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor network. In *Proceedings of 33rd Annual Hawaii, international conference on system science, IEEE, Big Island, USA*, 2000.

[23] F. Hu, J. Ziobro, J. Tillett, and N.K. Sharma. Secure wireless sensor networks : Problems and solutions. *Cybernetics And Informatics, Rochester Institute of Technology, Rochester, New York, USA*, Volume 1, Numéro 4, PP. 90–100, 2004.

[24] G. Jolly, M.C. Kuşçu, P. Kokate, and M. Younis. A low-energy key management protocol for wireless sensor networks. In *Eighth IEEE International Symposium on Computers and Communication(ISCC 2003). Proceedings, IEEE*, PP. 335–340, 2003.

[25] R. Kacimi. Techniques de conservation d'énergie pour les réseaux de capteurs sans fil. *Thèse de Doctorat en Réseaux et Télécommunications, Institut National Polytechnique de Toulouse, France*, 28 Septembre 2009.

[26] C. Karlof and D. Wagner. Secure routing in wireless sensor networks : Attacks and countermeasures. *In Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications*, 2003.

[27] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, Volume 48, Numéro 177, PP. 203–209, 1987.

[28] N. Koblitz, A. Menezes, and S. Vanstone. The state of elliptic curve cryptography. *Towards a quarter-century of public key cryptography, Springer*, PP. 103–123, 2000.

[29] N. Labraoui. La sécurité dans les réseaux sans fil ad hoc. *Thèse de Doctorat en Informatique, Université Abou Bekr Belkaid de Tlemcen, Algérie*, 2012.

[30] B. Lai, S. Kim, and I. Verbauwhede. Scalable session key construction protocol for wireless sensor networks. In *IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES)*, page 7. Citeseer, 2002.

[31] N. Lasla. La gestion de clés dans les réseaux de capteurs sans-fil. *Tèse de Doctorat en Informatique, Ecole Nationale Supérieure d'Informatique, Algerie*, Juin 2008.

[32] A. Liu and P. Ning. Tinyecc : A configurable library for elliptic curve cryptography in wireless sensor networks. *in International Conference on Information Processing in Sensor Networks, IEEE*, PP. 245–256, 2008.

[33] D. Liu and P. Ning. Location-based pairwise key establishments for static sensor networks. *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks, (CCS'03), ACM*, PP. 72–82, 2003.

[34] F. Liu and X. Cheng. Lke : A self-configuring scheme for location-aware key establishment in wireless sensor networks. *IEEE Transactions onWireless Communications, IEEE*, Volume 7, Numéro 1, PP. 224–232, 16 Janvier 2008.

[35] Y. Makhloufi and Z. Rezgui. Etude de mécanisme de gestion des clés dans les réseaux de capteurs sans fils proposition d'un protocole hybride basé sur la stéganographie. *Mémoire de Master en ReSyD, Université Abderrahmane Mira de Béjaia*, 2013.

[36] I. Mansouri. Contribution à la sécurité des communications des réseaux de capteurs sans fil. *Thèse de Doctorat en Informatique, Université Blaise Pascal - Clermont-Ferrand II*, France, 5 Juillet 2013.

[37] D. Martins. Sécurité dans les réseaux de capteurs sans fil-stéganographie et réseaux de confiance. *Thèse de Doctorat en Informatique, L'UFR des Sciences et Techniques de l'université de Franche-Comté*, France, 29 Novembre 2010.

[38] D. Martins and H. Guyennet. Etat de l'art - sécurité dans les réseaux de capteurs sans fil. *SAR-SSI 2008 : 3rd conference on Security of Network Architectures and Information Systems*, France,PP. 167–181, 2008.

[39] M.L. Messai. Sécurité dans les réseaux de capteurs sans-fil. *Mémoire de Magister en Informatique, Université Abderrahmane Mira de Béjaia, Algérie*, 2008.

[40] M.L. Messai and M. Aliouat. Protocole efficace de gestion des clés dans les réseaux de capteurs sans-fil. *UAMB, Ecole Doctorale en Informatique, ReSyD Béjaia, Algérie*, 2009.

[41] A. Milenković, C. Otto, and E. Jovanov. Wireless sensor networks for personal health monitoring : Issues and an implementation, elsevier. *Computer communications*, Volume 29, Numéro 13, PP. 2521–2533, 2006.

[42] V.S. Miller. Use of elliptic curves in cryptography. *Advances in Cryptology—CRYPTO'85 Proceedings, Springer*, PP. 417–426, 1985.

[43] E. Munivel and G. M. Ajit. Efficient public key infrastructure implementation in wireless sensor networks. *in International Conference on Wireless Communication and Sensor Computing, IEEE, Chennai*, PP. 1–6, Janvier 2010.

[44] A. Perrig, R. Szewczyk, J.D. Tygar, V. Wen, and D.E. Culler. Spins : Security protocols for sensor networks. *Wireless networks, Springer-Verlag New York, Inc*, Volume 8, Numéro 5, PP. 521–534, 2002.

[45] M. Ramdani. Problémes de sécurité dans les réseaux de capteurs avec la prise en charge de l'énergie. *Mémoire de Magister en Informatique Répartie et Mobile, Université de Saad Dahlab de Blida, Algérie*, Novembre 2013.

[46] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, Volume 21, Numéro 2, PP. 120–126, 1978.

[47] R. Sharmila, P.C. Gopi, and V. Vijayalakshmi. A survey of key management schemes in wireless sensor networks. *International Journal of Computer Organization Trends*, Volume 3, Numéro 9, PP. 444–450, Octobre 2013.

[48] P.R. Vamsi and K. Kant. A taxonomy of key management schemes of wireless sensor networks. *Fifth International Conference on Advanced Computing And Communication Technologies (ACCT), IEEE*, PP. 690–696, 2015.

[49] P.R. Vamsi and K. Kant. A taxonomy of key management schemes of wireless sensor networks. In *2015 Fifth International Conference on Advanced Computing & Communication Technologies (ACCT), IEEE*, PP. 690–696, 2015.

[50] J.P. Walters, Z. Liang, W. Shi, and V. Chaudhary. Wireless sensor network security : A survey. *Security in distributed, grid, mobile, and pervasive computing, CRC Press, FL, USA*, Volume 1, PP. 367–370, 2007.

[51] Y. Wang, G. Attebury, and B. Ramamurthy. A survey of security issues in wireless sensor networks. *Communications Surveys and Tutorials, IEEE*, Volume 8, Numéro 2, PP. 2–23, 2006.

[52] www.futura-sciences.com/magazines/hightech/infos/dico/d/informatique-simulation-informatique 11319/. Dernière consultation le 28/06/2016.

[53] Y. Xiao, V.K. Rayi, B. Sun, X. Du, and M. Galloway. A survey of key management schemes in wireless sensor networks. *Computer Communications*, Elsevier, Volume 30, Numéro 11, PP. 2314–2341, 10 September 2007.

[54] Y. Younes. Minimisation d'énergie dans un réseau de capteurs. *Mémoire de Magister en Informatique, Université Mouloud Mammeri de Tizi-Ouzou*, Septembre 2012.

[55] Y. Zhang and J. Pengfei. An efficient and hybrid key management for heterogeneous wireless sensor networks. *The 26th Chinese Control and Decision Conference (2014 CCDC), IEEE*, PP. 1881–1885, 2014.

[56] S. Zhu, S. Setia, and S. Jajodia. Leap : efficient security mechanisms for large-scale distributed sensor networks. In *Proceedings of the 10th ACM conference on Computer and communications security, ACM. Washington, DC, USA*, PP. 62–72, Octobre 2003.

Résumé

Un réseau de capteur sans fil est un ensemble de capteurs communicants par des liaison sans fil. Les contraintes matérielles de ces capteurs ainsi que les environnement de déploiement hostile rend ce type de réseau vulnérable et nécessite des mécanismes de sécurité efficaces et peu coûteux.

Dans ce mémoire nous présentons une étude des problèmes de sécurité dans les RCSF ainsi qu'un état de l'art sur les protocoles de gestion de clés existants. Ce travail s'achève par une amélioration d'un protocole de gestion de clés et l'analyse de ces performance, part la suite nous allons comparé notre solution avec deux protocoles que nous allons simulé en utilisant l'environnement MATLAB. Les résultats de simulation démontrent que le protocole proposé permet de faire des économies significatives dans la consommation d'énergie, espace mémoire et calcul.

Mots clés : Réseau de capteurs sans fil, Cryptographie, Gestion de clés, Communication sans fil.

Abstract

A wireless sensor network is a set of sensors communicating by wireless link. The physical constraints of these sensors and the hostile deployment environment makes this type of network vulnerable and requires effective and inexpensive security mechanisms.

In this paper we present a study of safety issues in WSNs and a state of the art on existing key management protocols This work concludes with an improvement of a key management protocol and analysis of these performance share later we will compare our solution with two protocols we simulated using the MATLAB environment. Simulation results demonstrate that the proposed protocol allows for significant savings in energy consumption, memory and calculation.

Keywords : Wireless sensors networks, Cryptography, Key management , Wireless communication.