الجـــــــــــــــــــمـهـوريـــــة الجـــــــــــــزائـريـة الديمـــــــقـراطيـة الـشـــــــــعبيـة

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

وزارة التـــــــــــــــــــــــــــعـليـم العـــــــالي و البـــــــــــحث العـــــلـمـــي

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

جـــــــــــــــامعة أبي بكـــر بـلـقـايد – تـلمســـــــــــــان –

Université Aboubakr Belkaïd – Tlemcen –

Faculté de TECHNOLOGIE

THESE

Présentée pour l'obtention du **grade** de **DOCTEUR EN SCIENCES**

En : Télécommunication

Spécialité : Systèmes et Réseaux de Télécommunications

Par : MESMOUDI Samira

Sujet

Vers une nouvelle approche intelligente pour la gestion de clés dans les réseaux de capteurs sans fils

Soutenue publiquement, le 09 /07 /2019 , devant le jury composé de :

Mr BORSALI Riadh	Professeur	Univ. Tlemcen	Président
Mr BENADDA Belkacem	Professeur	Univ. Tlemcen	Directeur de thèse
Mr MERZOUGUI Rachid	MCA	Univ. Tlemcen	Examinateur
Mr BENAISSA Mohamed	MCA	Centre Univ. Ain Temouchent	Examinateur
Mr SOUIR Mehdi	MCA	Ecole supérieure de management de Tlemcen	Examinateur

Remerciements

A l'issue de ces années de recherche, je retiens que l'achèvement d'une thèse c'est un travail qui pour émerger, se structurer et aboutir doit trouver son écho dans une foultitude de rencontres – c'est bel et bien dans l'interaction humaine qu'il s'élabore –cette thèse restera une étape lumineuse dans ma vie professionnelle et surtout personnelle.

C'est donc à tous ceux – professeurs, collègues doctorants, amis et famille – qui m'ont fait l'amitié de me dédier de leur temps et de me soutenir chacun à sa manière, que vont mes remerciements et ma gratitude.

Je tiens tout d'abord à remercier mon directeur de thèse, Pr. BENADDA Belkacem, pour son encouragement et ses conseils tout au long de ma thèse.

J'exprime tous mes remerciements à l'ensemble des membres de mon jury : Pr. BORSALI Riadh d'avoir accepté de présider le jury, Dr. MERZOUGUI Rachid, Dr. BENAISSA Mohamed et Dr. SOUIR Mehdi pour avoir accepté d'examiner cette Thèse.

Je tiens à remercier également le Dr. MESMOUDI Amin, pour leurs conseils pendant la thèse, qui m'ont été extrêmement précieux.

Je voudrais adresser un remerciement chaleureux à ma famille, et en particulier à mon marie, mes parents, mes frères et ma sœur qui, au cours de ces années de thèse, m'ont toujours soutenu et encouragé.

Je n'oublie pas non plus mes petits poussins et ma raison de vivre : Nadjib et Rofaïda.

Résumé

L'utilisation intensive des réseaux de capteurs sans fil (RCSF) dans des domaines très variés (médicale, militaire, environnemental, industriel) a soulevé plusieurs problèmes de sécurité. En effet, de nombreuses applications basées sur le RCSF nécessitent une communication sécurisée vu la sensibilité de certaines données collectées. Cette sécurité est généralement assurée par le cryptage des données transmis, ce qui nécessite l'établissement de nombreuses clés cryptographiques. La gestion de ces clés, dans un protocole, est une tâche importante qui garantit l'efficacité du mécanisme de sécurité. Le protocole doit être intelligemment adaptable non seulement aux événements d'intrusion, mais également au niveau de sécurité requis par certaines applications. Un protocole efficace optimise également l'énergie des capteurs et augmente par conséquent la durée de vie du réseau.
Dans cette thèse, nous proposons SKWN, un système de gestion de clés intelligent et dynamique pour les réseaux de capteurs sans fil hiérarchiques. Notre protocole propose trois sous-schémas pour l'établissement, le renouvellement et l'intégration de nouveaux nœuds. De plus, notre approche s'appuie sur une technique d'apprentissage automatique pour surveiller l'état du réseau et déterminer le niveau de sécurité approprié. Ainsi, SKWN ne fournit pas seulement des mécanismes de sécurité fiables, il optimise également la consommation d'énergie et les surcoûts liés à la communication et à l'utilisation de la mémoire. Les résultats présentés dans cette thèse sont issus de plusieurs simulations, qui démontrer la faisabilité et l'efficacité de notre proposition.

Mots clés : Réseau de capteurs sans fil hiérarchique, sécurité, gestion de clés, apprentissage automatique, l'efficacité d'énergie.

Abstract

The intensive use of wireless sensor networks (WSN) in a wide variety of areas (medical, military, environmental, industrial) has raised several security issues. Indeed, many WSN based applications require secure communication given the sensitivity of certain data collected. This security is generally ensured by the encryption of data transmitted by sensors, which requires the establishment of many cryptographic keys. Managing these keys, within a protocol, is an important task that guarantees the effectiveness of the security mechanism. The protocol should be intelligently adaptable not only to intrusion events but also to the security level needed by some applications. An efficient protocol optimizes also sensors energy and consequently increases the network life-cycle.
In this thesis, we propose SKWN, a Smart and dynamic Key management scheme for hierarchical Wireless sensor Networks. Our protocol offers three sub schemes to deal with key establishment, key renewal and new node integration. Our approach relies on a machine learning technique to monitor the state of the network and decide the appropriate security level. Thus, SKWN does not only provide reliable security mechanisms, but it also optimizes energy consumption and overheads related to the communication and memory usage. The results presented in this thesis are validated using several simulations, which demonstrate the feasibility and effectiveness of our proposal.

Key Words
Hierarchical wireless sensor network, security, key management, machine learning, energy-efficient.

ملخص

الإستخدام المكثف لشبكات الإستشعار اللاسلكية (WSN) في مجموعة واسعة من المجالات (الطبية، العسكرية، البيئية والصناعية) أثار العديد من المشكلات الأمنية. في الواقع، العديد من التطبيقات القائمة على شبكة WSN تتطلب اتصالًا آمنًا نظرًا لحساسية بعض البيانات التي يتم جمعها. هذا الأمن عادة ما يتم ضمانه من خلال تشفير البيانات التي تنتقل، الأمر الذي يتطلب إنشاء العديد من مفاتيح التشفير. إدارة هذه المفاتيح، ضمن بروتوكول، مهمة هامة تضمن فعالية آلية الأمن. حيث يجب أن يكون البروتوكول قابلاً للتكيف بذكاء ليس فقط مع أحداث التسلل ولكن أيضًا مع مستوى الأمن الذي تحتاجه بعض التطبيقات. يعمل البروتوكول الفعال على تحسين طاقة المستشعرات وبالتالي زيادة مدة حياة الشبكة.
في هذه الأطروحة، قمنا باقتراح بروتوكول جديد يدعى SKWN، وهو نظام إدارة مفاتيح ذكي وديناميكي لشبكات الإستشعار اللاسلكية الهرمية. هذا البروتوكول يقترح ثلاثة مخططات فرعية لإنشاء المفاتيح، تجديد المفاتيح ودمج العقد الجديدة. بالإضافة إلى ذلك، يعتمد منهجنا على أسلوب التعلم التلقائي لمراقبة حالة الشبكة وتحديد المستوى المناسب للأمن. وبالتالي لا يوفر البروتوكول SKWN آليات أمن موثوقة فحسب، بل إنها تعمل أيضًا على تحسين استهلاك الطاقة والتكاليف الإضافية المتعلقة باستخدام الاتصالات والذاكرة. النتائج المقدمة في هذه الأطروحة مستمدة من العديد من عمليات المحاكاة، والتي تثبت جدوى وفعالية اقتراحنا.

كلمات دلالية: شبكة الإستشعار اللاسلكية الهرمية، الأمن، إدارة المفاتيح، التعلم التلقائي، كفاءة الطاقة.

Table des matières

Table des figures

Liste des tableaux

Introduction générale

1. Introduction

Le développement de l'Internet des objets (Internet of Things, IoT) est principalement dû au déploiement massif des réseaux de capteurs sans fil (RCSF), qui sont de plus en plus utilisés dans différentes applications, notamment la gestion du trafic urbain, la surveillance des sites sensibles et l'étude de l'environnement naturel.

Généralement, un réseau de capteur sans fil est créé pour assurer les interactions entre le monde cybernétique et le monde physique. Ainsi, nous assistons à une production sans précédent de données, principalement collectées par des capteurs minuscules afin d'assurer l'interaction entre l'application et son environnement. Cet ensemble de capteurs est capable de récolter des mesures physiques liées à l'environnement (luminosité, son, pression barométrique, température, etc.) et transmettent par voie hertzienne vers un point centralisé. Certaines applications utilisent des réseaux de plusieurs centaines de capteurs sans fil. La figure 1 montre par ailleurs un exemple de cadre applicatif commun permettant d'exploiter les données collectées par des capteurs sans fil. Dans ce contexte, les données collectées sont agrégées et stockées à l'aide d'une infrastructure distribuée basée sur plusieurs serveurs de stockage et de calcul. La manière dont les données sont exploitées dépend des objectifs de l'application (par exemple, la régulation du climat, la surveillance de la santé et les diagnostics médicaux). D'ailleurs, dans ce contexte et vu la sensibilité de certaines données, il est nécessaire d'apporter une solution de sécurité du processus de collecte de données qui paraît importante voir cruciale. En effet, avec les contraintes liées à la miniaturisation, les capteurs (ou nœuds) sans fil sont dotés de faibles ressources en termes d'énergie, d'espace de stockage, et de calcul. Tout cela, en considérant de plus les environnements peu sûrs dans lesquels ils pourraient être déployés, rend ce genre de réseaux vulnérables à des nombreuses attaques malveillantes, telles que le brouillage, les attaques physiques et les attaques Sybil. Les capteurs sont en effet faciles à corrompre afin de récupérer les informations qu'ils collectent.

Un attaquant pourrait également récupérer les données en interceptant la communication entre des capteurs, par exemple par une attaque de trou noir. Il est nécessaire donc d'intégrer un mécanisme de sécurité qui non seulement gère les intrusions, mais garantit également un échange de données sécurisé.

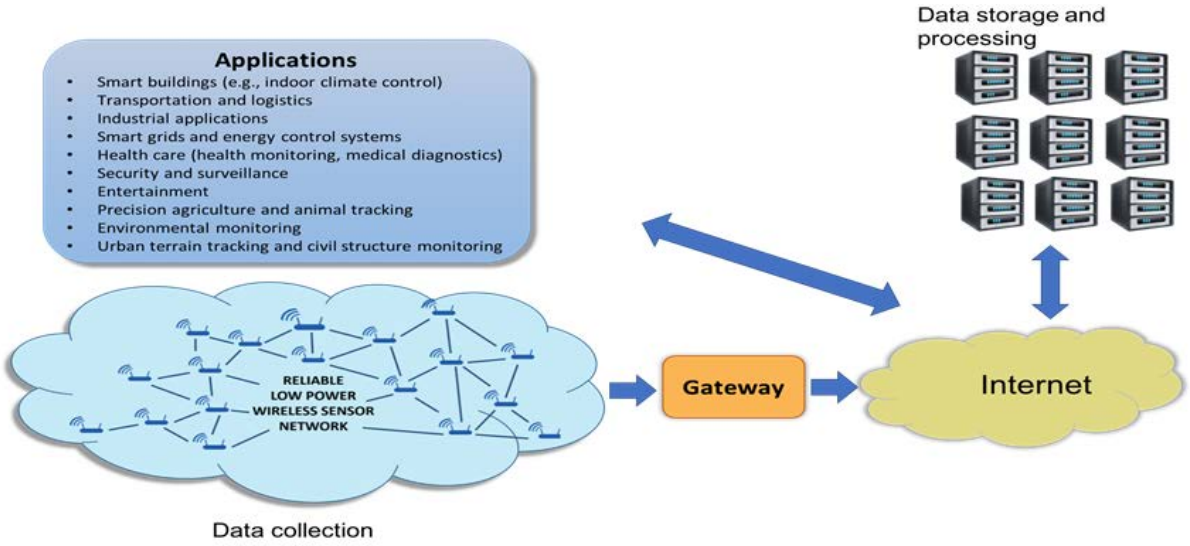

Figure.1. Vue d'ensemble sur un cadre applicatif d'exploitation de données avec les RCSF

Généralement, la cryptographie symétrique est l'une des mécanismes de sécurité les mieux adaptés aux réseaux de capteurs sans fils, et ce dans l'optique de sécuriser l'échange des données. En effet, avec un tel mécanisme, l'interception des communications ne permet pas de recouvrir les données envoyées par un nœud capteur. Un tel mécanisme s'appuie sur l'utilisation d'une clé secrète partagée entre l'expéditeur et le destinataire des données, ce qui permet de chiffrer tous les échanges. Toutefois, le point faible de ce mécanisme est la capacité d'intercepter et /ou de récupérer la clé cryptographique. Ainsi, il serait inutile d'intégrer des algorithmes cryptographiques dans un système de sécurité si la gestion de clés n'est pas prise en compte.

Généralement, la gestion de clés permet de pré-distribuer des clés cryptographiques, de révoquer les clés si les nœuds quittent le réseau, de renouveler des clés expirées, et d'assigner des nouvelles clés en cas d'une nouvelle intégration de nœud. D'ailleurs, dans cette thèse, nous nous intéressons aux méthodes de gestion de clés qui offrent non seulement une communication inter nœuds réduites mais aussi un niveau de sécurisation appréciable.

2. Problématique et motivation

La gestion de clés dans les réseaux de capteurs sans fil est devenue une tâche importante et cruciale qui permet d'assurer la fiabilité des mécanismes de sécurité (c.à.d. la cryptographie, l'authentification, ... etc.).

Plusieurs méthodes de gestion de clés ont été proposées afin d'avoir un schéma performant qui garantit un niveau élevé de sécurité et optimise les métriques de performances et conserve

l'énergie. En effet, il est difficile d'assurer un niveau de sécurité élevé avec une consommation d'énergie minimale.

À cette fin, quand on propose une méthode de gestion de clés, il est nécessaire de prendre en compte certains points.

-Un protocole de gestion de clé doit garantir une fiabilité en temps réel. C'est notamment par un ensemble de tâches dont le but est de pré-distribuer les clés cryptographiques, de révoquer les clés si les nœuds quittent le réseau, de renouveler les clés en cas de menace et d'attribuer des nouvelles clés lorsque certaines clés expirent.

-Un protocole doit également garantir la flexibilité et la scalabilité du réseau qui est liées au pouvoir de gérer les clés dans le cas d'une intégration d'un nouveau nœud.

-Un protocole de gestion de clés doit prendre en compte la durée de vie du réseau RCSF, qui sera réduite en raison de la consommation d'énergie. Cette consommation est principalement liée au coût de traitement de la CPU et au nombre de messages échangés par les capteurs. Par conséquent, un protocole de gestion de clés doit également optimiser la consommation d'énergie en utilisant des simples routines de calcul et un nombre réduit de messages.

-Un protocole de gestion de clés doit répondre efficacement aux exigences de sécurité. En effet, afin d'assurer la sécurité du protocole de gestion de clés, il est important de pouvoir effectuer diverses opérations cryptographiques, par exemple le cryptage et l'authentification. De plus, dans un RCSF, étant donné que certaines applications (par exemple, militaires) ont besoin de plus de sécurité que d'autres (agricultures), la complexité de la cryptographie utilisée (dans le protocole de gestion de clés) est liée à l'environnement de déploiement du réseau. Un protocole de sécurité doit donc être intelligent en s'appuyant sur des routines adaptatives afin de fournir le meilleur compromis entre la fiabilité et la consommation d'énergie.

En résumé, avec le déploiement intensif de RCSF, la nécessité d'un protocole de gestion de clés fiable, scalable et qui économise l'énergie reste un réel défi. Un tel protocole devrait donc être également intelligent en adaptant dynamiquement ses routines à l'environnement de déploiement du RCSF.

3. Contributions de cette thèse

Malgré que des études approfondies ont fait l'objet de plusieurs travaux de recherches ces dernières années afin de résoudre les problèmes de sécurité, notamment la gestion de clés liés aux RCSF, ce sujet reste ouvert.

Étant donné les perspectives applicatives prometteuses des RCSF et la nécessite de nouvelles solutions au problématique de sécurité, l'objectif de la thèse consiste à proposer une nouvelle approche de gestion de clés permettant d'apporter des solutions aux trois défis principaux : la fiabilité, la scalabilité et l'économie de l'énergie.

Nous proposons dans cette thèse un nouveau protocole nommé SKWN (Smart and dynamic Key management scheme for hierarchical Wireless sensor Networks) dont l'objectif est de surpasser les limites des protocoles de gestion de clés existants. Notre approche est destinée aux réseaux de capteurs sans fils hiérarchiques. SKWN est déterministe et repose sur la cryptographie symétrique. Afin d'offrir une performance globale optimisée, nous nous appuyions sur le composant ISA (Intelligent Security Agent). Ce composant nous permet de déclencher intelligemment uniquement les sous-routines de gestion de clés nécessaires en ce qui concerne l'environnement de déploiement et les menaces perçues.

4. Organisation de la thèse

Ce manuscrit est organisée en quatre chapitres en plus d'une introduction générale et d'une conclusion générale :

- Le premier chapitre présente une description générale sur le fonctionnement des réseaux de capteurs sans fil, leur application ainsi que les différentes topologies et organisations utilisées dans ce genre de réseau. Par la suite, nous avons décrit quelques défis et contraintes liés à la conception des RCSF.
- Dans le deuxième chapitre, nous présentons un aperçu sur les concepts de sécurité dans les RCSF qui diffèrent des autres réseaux. Nous commençons d'abord par les limites des réseaux de capteurs qui rendent la sécurité pour ce type de réseaux un véritable défi. Enfin, nous identifions une taxonomie des attaques et nous discutons les besoins des différents mécanismes de sécurité.
- Dans le troisième chapitre, nous présentons un état de l'art sur notre axe de recherche, à savoir, la gestion de clés. Nous survolons les principaux schémas proposés dans la littérature, en effectuant une classification et une analyse profonde, ainsi qu'une comparaison entre les différents schémas présentés en fonction des critères d'évaluation des performances d'un système de gestion de clés dans les RCSF.
- Le quatrième chapitre présente notre proposition liée au système de gestion de clés nommée SKWN (smart and dynamic key management scheme for hierarchical wireless sensor networks). Nous commençons d'abord par présenter les motivations de cette

proposition. Nous donnons par la suite les détails de notre protocole, et nous effectuons une évaluation de ses performances par rapport aux approches existantes.

- Dans la partie conclusion, nous résumons les idées et les résultats de notre proposition et nous proposons de nouvelles perspectives permettant à étendre ce travail.

.

1

Préliminaires– les réseaux de capteurs sans fil

Sommaire

1.1. Introduction

Les récents progrès réalisés dans les microprocesseurs miniatures, les conceptions de circuits à faible puissance et les technologies radio ont donné naissance à une nouvelle vision technologique appelée « réseaux de capteurs sans fil (RCSF) ». Un RCSF, ou *WSN* (pour *Wireless Sensor Networks* en anglais), sont des réseaux composés généralement d'un grand nombre de nœuds communicants dits capteurs. Chaque nœud/capteur est équipé de fonctionnalités de sensation avancées, il mesure ou détecte un évènement réel. La pression, la température, la présence d'un gaz ou l'humidité des exemples d'information qu'un nœud capteur est capable de recueillir. Toutefois, il s'agit d'un dispositif à capacité de calcul et de mémoire limitées.

Les capteurs forment des nœuds d'un réseau, ils coopèrent ensembles afin d'accomplir une tâche commune. Cette tâche peut monitorer ou agir sur environnement, par exemple la surveillance de sites naturels inaccessibles, pour construire une vue globale et la rendre accessible aux utilisateurs distants. Les données échangés par ces capteurs sont généralement envoyées vers un nœud de contrôle puissant appelé station de base (ou puits) qui va transmettre vers l'utilisateur final, via internet ou satellite (voir figure 1.1). En outre, le nœud de contrôle peuvent traiter les données collectées, disséminer des commandes de contrôle aux nœuds capteurs [1].

Nous allons retracer dans ce chapitre le fonctionnement général des réseaux de capteurs sans fil. Nous abordons d'abord les composants d'un capteur sans fil et ses fonctionnalités, leurs architectures protocolaire et les différentes topologies et organisation utilisées dans ce genre de réseau. Ensuite, nous présentons les domaines d'applications des RCSF. Et finalement, nous avons décrit quelques contraintes liés à la conception de ce type de réseaux.

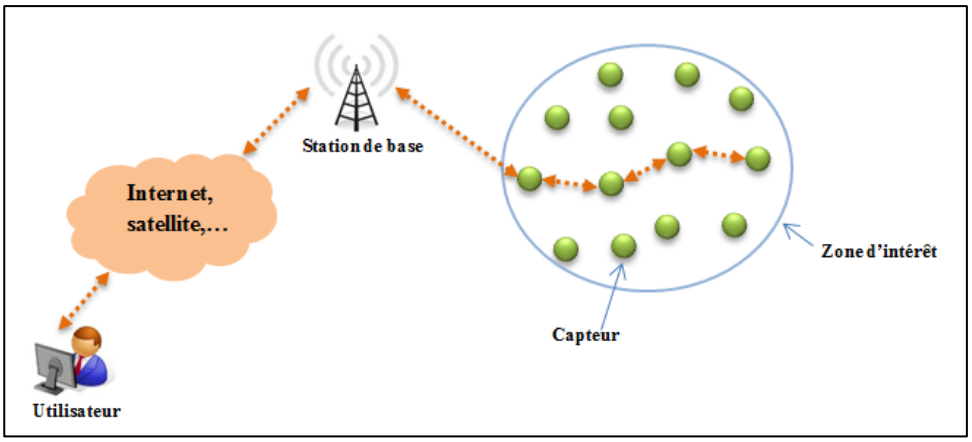

Figure1.1 : Architecture de base d'un réseau de capteur sans fil (RCSF)

1.2. Les nœuds de réseau de capteur sans fil

Les nœuds capteurs de RCSF sont des nœuds destinés à générer, acheminer et consommer l'information juxtaposés avec des capteurs, actionneurs et des circuits dédiés au traitement du signal. De plus, ils ne sont pas des dispositifs banalisés car chacun de leurs constituants a été conçu dans l'objectif d'une application bien spécifique. Dans Les deux prochaines sous-sections. Nous écrivons d'abord les différents modules matériels du nœud capteur. Puis, nous consacrons la suite au système d'exploitation qui commande les modules matériels.

1.2.1. Aspect matériel

Bien qu'il y a une large variété de capteurs, leur architecture matérielle reste identique. Un nœud capteur est composé de quatre modules principaux, qui nécessitent d'être regroupées dans un composant convenable [2] (figure 1.2) avec la contrainte de la taille qui doit rester petite (quelques centimètres). Il s'agit :

Unité de captage : composée de deux sous unités principales, la sous unité des capteurs chargés de réaliser des mesures sur les paramètres environnementaux, la sous unité de convertion analogique-numérique (CAN) qui convertit l'information relevée en un flot de valeurs numériques et la transmet par la suite à l'unité de traitement.

Unité de traitement : constitué d'un processeur et de mémoire (mémoire vive et mémoire non volatile). Cet ensemble est à la base de garantir le fonctionnement du système d'exploitation (TinyOs, par exemple), elle est chargée d'exécuter les protocoles de communications qui permettent au capteur de collaborer avec les autres nœuds capteurs pour accomplir la requête assignée, et elle peut aussi traiter les données récoltées.

Unité de communication : équipé généralement d'un transmetteur radio qui permet de communiquer avec les autres nœuds capteurs au sein d'un réseau. Celui-ci communique selon plusieurs modes, les plus fréquemment retenues sont : de type optique (exemple des capteurs *Smart Dust* [3]), ou de type radiofréquence (exemple des capteurs *Mica2* [4]). Le dernier type est basé sur les technologies sans fil à faible portée de communication, qui sont la technologie Zigbee [IEEE 802.15.4], Bluetooth [IEEE 802.15.1] ou WiFi [IEEE 802.11] [1].

Unité d'énergie : Il s'agit d'une source d'énergie se trouve généralement sous la forme de batterie de basse tension [5], accompagnée d'une unité de contrôle de l'énergie. Elle est chargée d'alimenter les autres modules du capteur. Cependant, à cause de capacité des batteries limitée et que, de plus, il n'est pas toujours possible de les remplacer lorsqu'elles sont déchargées, elle constitue une contrainte importante pour la durée de vie des capteurs.

Il peut contenir également, d'autre unités additionnelles suivant son domaine d'application, comme un mobilisateur chargé de déplacer le capteur pour accomplir la requête assignée. On peut même trouver des capteurs, dotés d'un système de localisation, dans le but d'attribuer leur position géographique. À titre d'exemple en utilisant un système de localisation (GPS). Certains applications aussi avoir besoin des capteurs, un peu plus volumineux, dotés d'un système générateur d'énergie (cellule solaire).

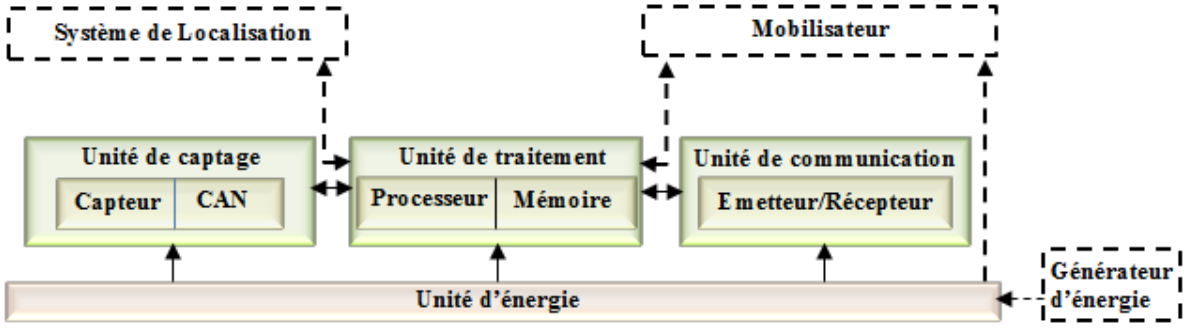

Figure1.2 : Composants d'un nœud capteur. Les quatre unités captage, traitement, communications et d'énergie sont indispensables, les unités restantes sont optionnelles.

L'électronique fournit des innovations au niveau du matériel. On constate que les composants électroniques disponibles sont régulièrement améliorés en taille et en performances. Aujourd'hui, divers exemples de capteurs existent sur le marché (figure 1.3) dont les caractéristiques dépendent du type d'application. Nous citerons MEMSIC (anciennement Crossbow), Cisco, Dalsa, EuroTherm, et Sens2B, sont les fabricants de capteurs les plus connu. Le Tableau 1.1 résume les principales caractéristiques matérielles de quelques capteurs sans fil, comme : le type de processeur, la mémoire Flash disponible pour les applications ou de la mémoire utilisable pour les calculs (RAM).

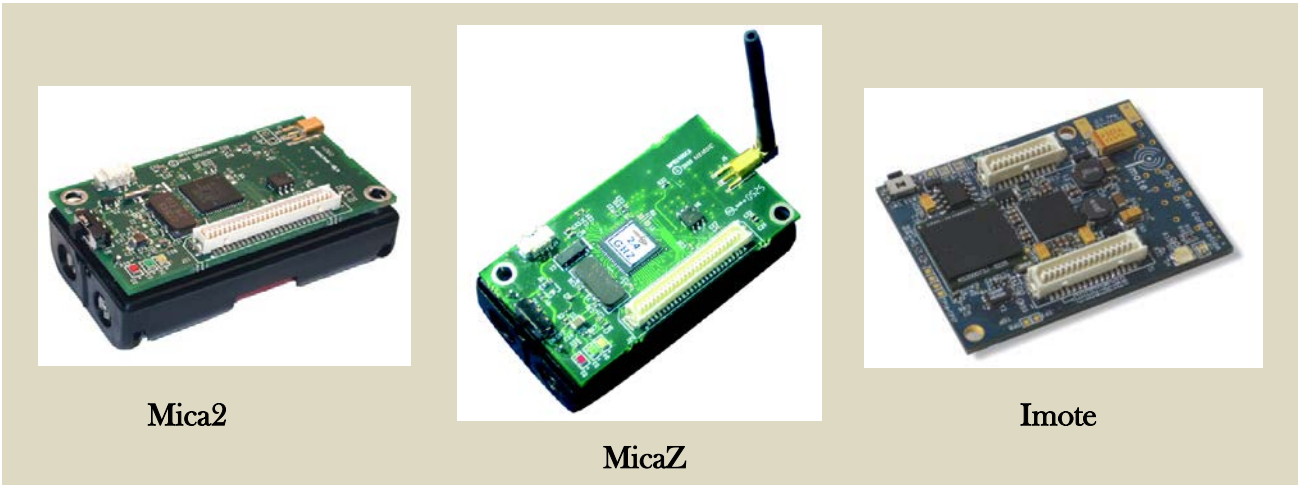

Figure 1.3 : Quelques modèles des capteurs sans fil

Tableau1.1 : Caractéristiques de quelques capteurs sans fil [6] [7]

Plate-forme	Imote2	Mica2	MicaZ	TELOSB	WSN430
Processeur	Intel PXA271 XScal	Atml ATmega128L	Atml ATmega128L	Texas Instruments MSP430	Texas Instruments MSP430
Mémoire	SDRAM 32 MB SRAM 256 KB flash 32 MB	RAM 128 KB EEPROM 4 KB flash 512 KB	RAM 128 KB EEPROM 4 KB flash 512 KB	RAM 10 KB EEPROM 16 KB flash 48 KB	RAM 10 KB flash 1 MB
Type radio	CC2420	CC1000	CC2420	CC2420	CC1101/CC2420
Fréquence	2400-2483,5 MHz	315/433/868/916 MHz	2400-2483.5 MHz	2400-2483.5 MHz	315/433/868/915 MHz
Débit de transmission	250 KB/s	38.4 KB/s	250 KB/s	250 KB/s	250 KB/s
Batterie	3×AAA	2×AA	2×AA	2×AA	PoLiFlex
Voltage	3,2 – 4,5 V	2.7 - 3.3 V	2.7 - 3.3 V	1.8 – 3.6 V	2.2 V

1.2.2. Le système d'exploitation

Un système d'exploitation pour les nœuds capteurs sans fil permet de jouer le rôle d'intermédiaire entre l'utilisateur et les périphériques matériels. Il est typiquement à une architecture qui permet l'exécution rapide, tout en diminuant au minimum le nombre d'instructions. Ceci à cause des contraintes de ressources des nœuds capteurs notamment en ce qui concerne l'espace de stockage, l'espace mémoire alloués aux systèmes d'exploitation et aux applications tournant dessus. Il existe plusieurs systèmes d'exploitation conçus pour les nœuds capteurs sans fil. Les plus utilisés sont TinyOS [8] et Contiki [9].

Tinyos est un système d'exploitation open-source dédié spécialement pour les applications embarquées fonctionnant en réseau et en particulier, pour les réseaux de capteurs sans fil. Cette plateforme logicielle a été développée au sein de l'université de Berkely en Californie. Il supporte les plateformes suivantes [10]: intelmote2, mica2, micaz, mica2dote, eyesIFXv2, telsob, tinynode et btnode3. La conception de Tinyos a été écrite entièrement en NesC (Network Embedded System C) [11], une extension de C. Il adopte une architecture basée sur une association de composants, qui réduit au minimum la taille du code nécessaire à sa mise en place. Cela due aux ressources très limitées de capteur (des contraintes de mémoires). Sa

bibliothèque de composants comprend les protocoles réseaux, les services de distribution, des pilotes de capteurs et les outils d'acquisition de données. Tinyos fournit également une solution permettant de développer des applications répondant à la diversité des caractéristiques existantes d'un réseau à l'autre.

1.3. Topologie et organisation de RCSF

La topologie détermine l'organisation des capteurs dans le réseau, une fois les nœuds capteurs déployés, ils s'auto-organisent et s'auto-configurent pour constituer un réseau [12]. Les topologies dans les réseaux de capteurs dépendent des applications et des techniques utilisées pour faire acheminer l'information des capteurs à la station de base. Il existe deux principales topologies dans les RCSF.

1.3.1. Topologie plate

Dans une topologie plate, le réseau est homogène, où tous les nœuds ayant les mêmes caractéristiques matérielles (même capacité de calcul, capacité énergétique, capacité de stockage, portée de communication, etc.). Cette architecture est utilisée pour une densité de capteurs élevée (plusieurs nœuds capteurs / m^2) [13]. Les capteurs peuvent communiquer directement avec la station de base (Figure 1.4) en utilisant une forte puissance, ou via un mode multi-sauts avec des puissances très faibles (c'est-à-dire que l'information envoyée par un nœud récolteur doit transiter par plusieurs nœuds intermédiaires avant d'atteindre sa destination finale sur le réseau et sans aucun traitement supplémentaire sur la donnée transportée).

Dans un très grand RCSF, il n'est pas possible de préserver une topologie plate (topologie adaptée aux petits réseaux dans laquelle tous les nœuds ont le même rôle et les mêmes caractéristiques) car plusieurs problématiques sont encore plus critiques avec le passage à l'échelle. Il s'agit entre autre de réduire [14]:

- la taille de la table de routage par nœud capteur,
- l'occupation de la bande passante,
- le nombre de transmissions et retransmission,
- la consommation d'énergie par nœud capteur.

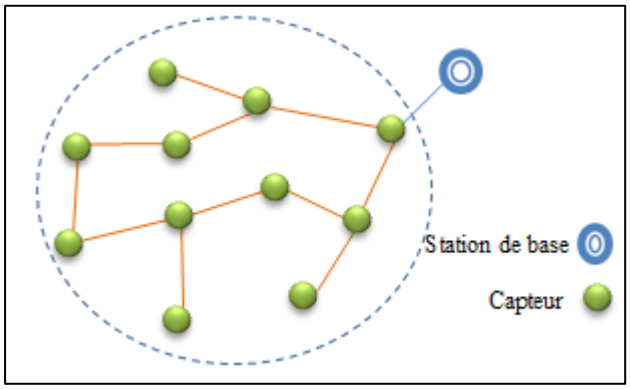

Figure 1.4 : Exemple d'une topologie plate d'un RCSF

1.3.2. Topologie hiérarchique

Il s'agit donc d'introduire une hiérarchie dans le réseau, donc en divisant les nœuds capteurs en plusieurs niveaux de responsabilité. Il consiste à diminuer la puissance de transmission des nœuds capteurs et donc à réduire leur portée de communication. Cette solution est retenue pour organiser un très grand RCSF.

L'une des méthodes hiérarchiques les plus utilisées est le clustering, ce genre de topologie est plus adapté à notre problématique. Nous aborderons ci-dessous les caractéristiques définissant les principes du clustering.

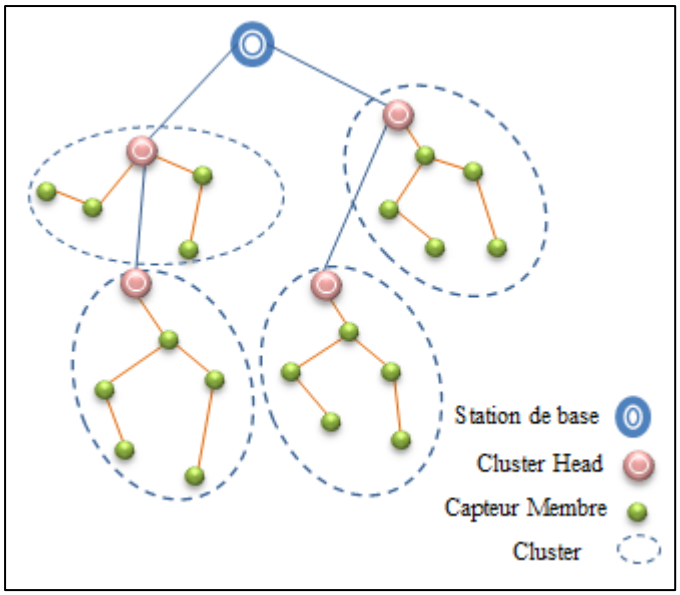

Figure 1.5 : Topologie hiérarchique par clustring d'un RCSF

Clustering

L'approche de clustering (ou la structuration) consiste à partitionner le réseau en un certain nombre de groupes appelés clusters ou grappes, plus homogènes selon une métrique particulière ou une combinaison de métriques, et former une topologie virtuelle (Figure 1.5). Un cluster est constitué d'un nœud particulier appelé chef de cluster ou "cluster-head" (CH), permettant de coordonner entre les membres de son cluster, d'agréger et /ou de traiter leurs données collectées et de les transmettre à la station de base [15]. Il est sélectionné soit par élection par les nœuds membres de son cluster, soit choisi et imposé par la station de base [12]. Le processus d'élection utilise une métrique spécifique pour chaque nœud telle que l'énergie restante, le degré de connectivité, la puissance de transmission, son emplacement géographique, le plus grand/petit ID dans son voisinage, etc., ou bien un poids qui représente une combinaison de certains métriques.

Un Cluster Head perdra son rôle et remplacé par un autre nœud capteur du groupe pour l'une des raisons suivantes : (1) CH compromis; (2) sa batterie sera épuisée; (3) éviter la présence des membres isolés, ou des raisons qui nécessite une sélection dynamique des cluster-heads.

De nombreux algorithmes de clustering ont été proposés dans la littérature. Elles peuvent être distingués et classés selon plusieurs paramètres tels que :

- ✓ le type du réseau, homogène : constitué de nœuds de mêmes ressources, hétérogène : constitué de nœuds à ressources non égales.
- ✓ Le mode de déploiement des nœuds capteurs : aléatoire ou déterministe,
- ✓ Le type d'algorithmes utilisés : centralisé où c'est le concepteur de l'algorithme ou la station de base qui désigne les chefs de clusters parmi les nœuds du réseau ou distribué où le chef du cluster est choisi suite à des interactions entre les nœuds, le processus d'élection des chefs de clusters, etc.

On cite les travaux : LEACH (Low-Energy Adaptive Clustering Hierarchy) [16], HEED (Hybrid Energy-efficient Distributed clustering protocol) [17], TEEN (Threshold sensitive Energy Efficient sensor Network protocol) [18], APTEEN (Adaptive Threshold sensitive Energy Efficient sensor Network protocol) [19]...etc. Ce sont des protocoles conçus pour la création et la gestion des topologies en clusters.

Parmi les avantages du clustering, l'organisation d'un très grand RCSF. Pour la transmission d'une information sur le réseau, deux types de communication sont mises en œuvre : intra-cluster et inter-cluster. Ceci permet de réduire le nombre de nœuds participant à des

communications sur de longues distances. L'objectif de cette technique est d'exposer le meilleur procédé de conservation d'énergie par un traitement et une transmission contrôlée des données récoltées.

1.4. Architecture protocolaire dans les RCSF

Lors de l'activation, les nœuds de RCSF commencent de construire la topologie de communication. Ainsi, ils deviennent capables d'atteindre leurs objectifs de déploiement. Comme tous les types de réseaux, Pour la communication dans un RCSF une pile protocolaire est utilisée par la station de base (puits) et par tous les nœuds capteurs. Les architectures en couches classiques utilisées pour les réseaux de capteurs sans fil présentent des inconvénients en termes de performances et d'efficacité. Tandis que la grande majorité des solutions existantes sont basées sur l'approche classique des protocoles en couches. Les ressources limitées des nœuds capteurs sans fil, telles que la mémoire, la puissance de calcul et l'énergie, incitent à modifier les architectures en couches classiques à une architecture cross layer, cette terminologie anglophone signifie inter-couches ou multi-couches. Dans les deux sous-sections suivantes nous présentons les deux modèles d'architecture protocolaire.

1.4.1. Architecture en couche

Comme le montre la figure 1.6, la pile protocolaire est composée de cinq couches: une couche d'application, une couche de transport, une couche réseau, une couche de liaison de données et une couche physique. Chaque couche a son propre rôle et ses propres protocoles. Due à la forte contrainte de limitation de ressource des RCSF, trois plans de gestion doivent être ajoutés afin de gérer la consommation d'énergie, la mobilité des nœuds et l'ordonnancement des tâches [2].

1.4.1.1. Couches de la pile protocolaire

Une pile de protocole est une mise en œuvre particulière d'un ensemble de protocoles de communication réseau ou chaque couche a son propre protocole employé selon sa fonction. Elle est composée de:

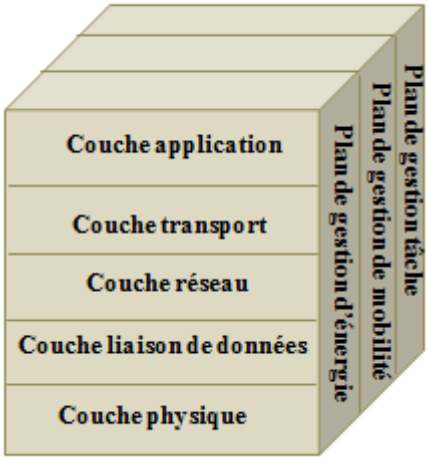

Figure 1.6 : La pile protocolaire des réseaux de capteurs

- **Couche physique :** Cette couche est responsable de la sélection de la fréquence, la génération de la fréquence porteuse, la détection du signal et la modulation/démodulation des données.
- **Couche liaison de données :** Elle spécifie comment les données sont transférées entre deux entités du réseau dans une distance d'un seul saut. Elle est responsable aussi, de la détection et la correction d'erreurs survenues sur la couche physique (en cas de perturbation ou dégradation du signal), du multiplexage des flux des données, de l'accès au media physique. Ainsi, elle assure la fiabilité du lien de canal radio point à point ou point à multi-points.
  Elle se décompose en deux sous-couches : la sous-couche de contrôle de la liaison logique (LLC : *Logical Link Control)* et la sous-couche du contrôle d'accès au support (MAC : *Media Access Control*). La première fournit la plupart des mécanismes de gestion d'erreur. Tandis que la sous-couche MAC gère tous les accès au canal radio physique [20].
  Parmi les protocoles de liaison de données, on peut citer: SMACS (Self-organizing Medium Access Control for Sensor networks) et EAR (Eavesdrop And Register) [21].
- **Couche réseau :** Elle s'occupe de l'acheminement des données via le réseau, c'est-à-dire elle gère l'adressage et le routage des données. Le protocole de routage est le principal acteur dans cette couche, il permet d'établir les routes entre les nœuds capteurs et le puits et sélectionne le meilleur chemin. Plusieurs métriques sont considérées dans l'optimisation des coûts des chemins dans les RCSF comme [20]:
  - L'énergie nécessaire pour transmettre le paquet.
  - Le temps nécessaire d'acheminer les paquets.
  - L'énergie disponible dans chaque nœud capteur.

Parmi les protocoles de routage conçu pour le RCSF, nous citons: SAR (Sequential Assignement Routing) [21] et LEACH (Low-Energy Adaptive Clustering Hierarchy) [22]

- **Couche transport :** Elle est chargée du transport des données fiable pour l'application. Elle manipule le découpage de données en paquets. Elle effectue le contrôle de flux de données, de la conservation de l'ordre des paquets et de la gestion des éventuelles erreurs de transmission. Pour les RCSF des versions allégées des protocoles UDP (User Datagram Protocol) [23] et TCP (Transmission Control Protocol) [24] utilisés dans cette couche. Ce dernier peut être nécessaire pour habiliter le réseau RCSF d'interagir avec les réseaux externes.
- **Couche application :** Il s'agit donc de la couche la plus proche des utilisateurs, géré directement par les logiciels. Elle assure un service propre à l'application déployée, Ces applications doivent fournir des mécanismes permettant à l'utilisateur d'interagir avec les réseaux de capteur. Les protocoles dans cette couche accomplissent les tâches administratives, tels que les échanges des informations qui déterminent l'endroit, des règles pour l'agrégation des données, la synchronisation des nœuds capteurs, la sécurité et la mobilité. Parmi les protocoles d'application, nous citons : SMP *(Sensor Management Protocol)* et TADAP *(Task Assignement and Data Advertisement Protocol)* [2].

1.4.1.2. Plans de gestions

On trouve aussi trois plans verticaux de gestions illustrés sur la figure 1.6 qui sont

Plan de gestion d'énergie : Sert à contrôler la manière d'utiliser l'énergie par le nœud capteur, et gérer l'énergie consommée selon le mode de fonctionnement employé (acquisition, calcul, et communication par radio). Par exemple, si le niveau d'énergie d'un nœud atteint un bas niveau, ce nœud annonce à ses voisins pour ne pas participer au routage des données et il conserver son énergie résiduelle pour les fonctionnalités d'acquisition.

Plan de gestion de mobilité : A pour rôle de détecter et enregistrer les mouvements des nœuds capteurs. De cette manière, chaque nœud peut déterminer leurs voisins afin d'équilibrer l'exécution des tâches et la consommation d'énergie. Il doit aussi maintenir continuellement une route vers l'utilisateur final.

Plan de gestion de taches : Il assure la coopération des efforts des nœuds capteurs, et ceci dans le but d'économiser de l'énergie sur le réseau. Pour cela, le plan de gestion des taches équilibre et ordonnance des tâches de capture dans une région d'acquisition spécifique. Les nœuds ne

sont pas obligés d'effectuer les tâches de capture à un même instant selon leur niveau d'énergie, et par conséquent, la durée de vie du réseau peut être prolongée.

1.4.2. L'architecture Cross-layer

Récemment, l'architecture Cross-layer est apparue comme une approche intéressante pour l'amélioration des performances des réseaux sans fil. Le concept de base de cette approche est consisté à un échange d'informations entre toutes les couches protocolaires, éventuellement non adjacentes pour les faire collaborer ensembles afin d'améliorer la flexibilité et d'augmenter les interactions inter-couches, tandis que l'architecture en couches interdit la communication directe entre les couches non adjacentes.

La mauvaise gestion des ressources par l'architecture en couche représente une autre motivation pour rendre l'approche Cross-layer particulièrement souhaitable pour les RCSF. En effet, l'architecture en couches engendre une certaine forme de redondance, qui peut gaspiller les ressources disponibles dans le réseau.

Par exemple, l'utilisation des protocoles efficaces en énergie au niveau de chaque couche de la plie protocolaire visent à diminuer au minimum la consommation d'énergie. Cependant, ça engendre une mauvaise gestion d'énergie. Pour cela et en raison de l'interdépendance des paramètres affectant la consommation, l'architecture en couches n'est pas la plus adaptée pour un réseau de capteurs sans fil. Les protocoles de communication basés sur l'architecture Cross-layer surpassent ce problème. Elle repose sur la réduction des coûts énergétiques occasionnés par le partage des informations entre les différentes couches protocolaires. Ainsi, un seul protocole sera utilisé afin de gérer efficacement les conserves d'énergie dans les différentes couches du modèle OSI [25].

On distingue trois types de base de l'architecture Cross-layer: architecture à base de communication directe, architecture à base de communication indirecte et architecture à base de nouvelles abstractions [26] [27].

Architecture Cross-layer à base de communication directe :

Le premier type d'architecture Cross-layer (voir figure 1.7(a)) consiste à préserver l'architecture en couche du modèle OSI, et à permettre la communication directe entre les protocoles au niveau des couches adjacentes et non adjacentes. Ainsi, les protocoles proposés pour cette architecture doivent être pouvoir manipuler les données Cross-layer échangées entre les différentes couches.

Architecture Cross-layer à base de communication indirecte :

Dans ce type d'architecture (voir figure1.7(b)), le concept de base est de conserver le fonctionnement normal de la pile protocolaire tout en permettant des communications entre les différentes couches protocolaires via une entité intermédiaire dont la dénomination et les fonctionnalités varient selon l'architecture [28] [29].

Architecture Cross-layer à base de nouvelles abstractions :

Le troisième type d'architectures Cross-layer (voir figure1.7(c)) est particulièrement différent des deux autres types, car il présente des nouvelles abstractions. Dans ce concept il ne plus considérer des couches individuelles, mais un système complet englobant toute la pile protocolaire de façon de construire une sorte de super couche, ce qui représente le noyau de communication.

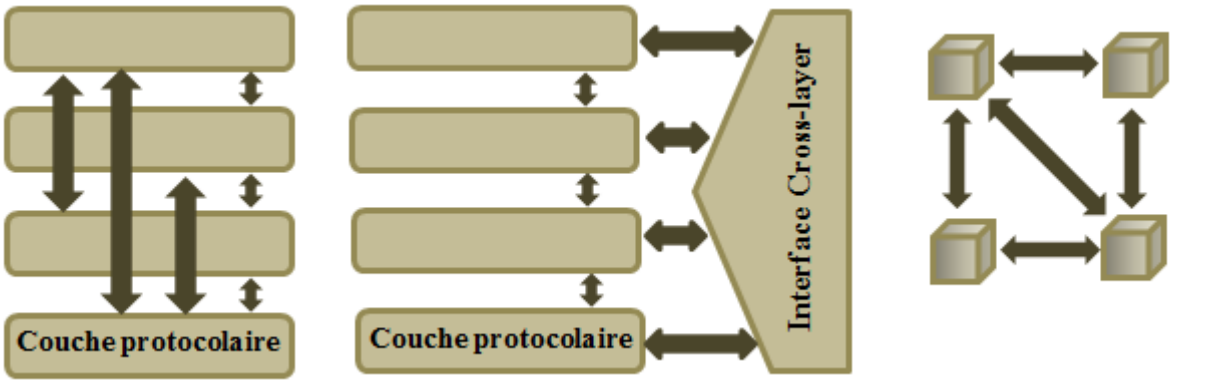

Figure 1.7 : Classification des architectures cross-layer [30]

1.5. Consommation et conservation d'énergie d'un nœud capteur (Efficacité énergétique)

Le nœud capteur sans fil, étant un dispositif micro-électronique, il ne peut être équipé que par une source limitée d'énergie. La batterie est considérée comme l'unique alimentation en ressources énergétiques des capteurs, dont la capacité est limitée étant donné sa petite dimension et son remplacement ou rechargement est souvent impossible. Au-delà de l'endroit hostile ou difficile d'accès avec moins de contrôle et d'existence humaine, la source d'énergie

joue un rôle critique dans la survie de nœud capteur. En effet, la durée de vie d'un réseau dépend essentiellement de la durée de vie de ses nœuds capteurs.

1.5.1. Le modèle de consommation d'énergie

Comme montre la figure 1.8, l'énergie consommée par un nœud capteur est due essentiellement aux trois opérations principales : l'acquisition, le traitement et la communication de données [31].

Acquisition : L'énergie d'acquisition est dissipée pour accomplir plusieurs tâches, notamment l'échantillonnage, la conversion des signaux physiques en signaux électriques, le traitement des signaux et la conversion analogique numérique. En général, elle représente un faible pourcentage de l'énergie totale consommé par un nœud. En revanche, elle varie en fonction du phénomène et du type de surveillance effectué, de sorte que les capteurs passifs (exemple : capteur de température) sont moins consommateurs d'énergie par rapport aux capteurs actifs (exemple : capteur d'image) [32].

Traitement : C'est l'énergie dissipée par l'unité de traitement lors de l'exécution des opérations relatives aux traitements effectués sur les données reçues des autres unités.

Généralement l'unité de traitement possède divers modes de fonctionnement avec différentes niveaux de consommation d'énergie: actif, " idle" ou écoute, "sleep" ou sommeil. En outre possède un état de commutation entre les modes de fonctionnement [20].

Ainsi, la quantité de consommation d'énergie des différents modes, les coûts de commutation entre les modes et le temps passé par l'unité de traitement dans chaque mode ont un impact important sur la consommation totale d'énergie d'un capteur.

Communication : Elle est destinée aux opérations d'émission et de réception des données. Elle est affectée par deux facteurs essentiels : la quantité de données à communiquer et la puissance de transmission (déterminée par la distance séparant les entités communicantes).

En général, On distingue 4 modes différents de fonctionnement dans les radios : transmission, réception, "idle"(écoute sans communication), "sleep" ou sommeil. En outre un état de transition entre les modes de fonctionnement.

Notons que la radio consomme beaucoup plus d'énergie dans les modes transmission et réception. Cependant, le mode "idle" induit une consommation d'énergie significative. Un autre facteur déterminant est que, la transition d'un mode à un autre implique un surplus d'énergie due à l'activité des circuits électroniques.

L'histogramme présenté par la figure 1.8, montre la consommation de l'énergie pour chaque tâche réalisée par le nœud capteur. On voit clairement que la consommation d'énergie pour la tâche de capture et celle du traitement sont peut être négligeables, alors que la tâche de transmission est la plus consommatrice de l'énergie suivie de celle de la réception.

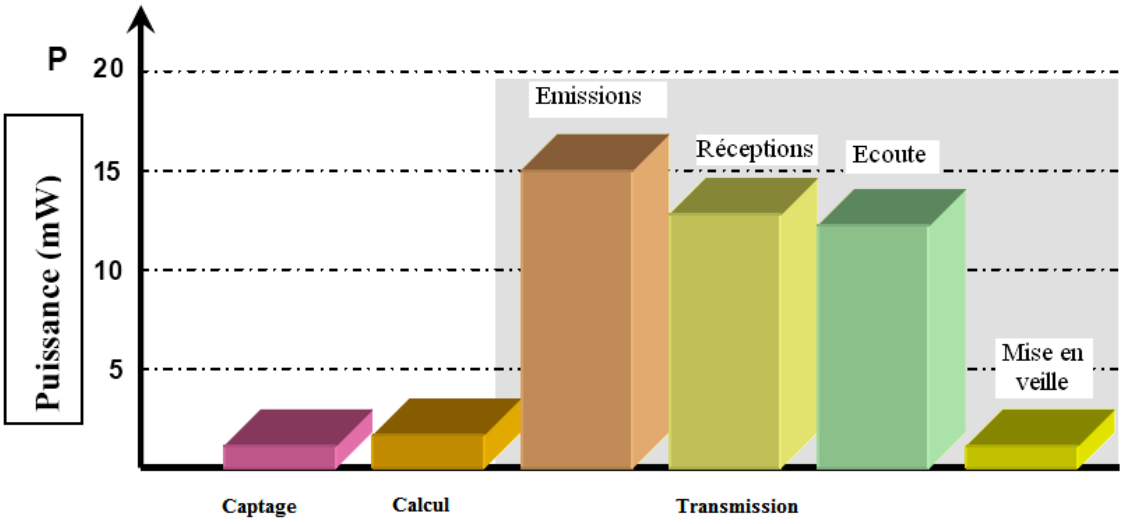

Figure 1.8 : Consommation d'énergie par un capteur sans fil [33]

1.5.2. L'importance de l'efficacité énergétique

L'efficacité en consommation d'énergie représente une métrique de performance spécifique, qui influence directement sur la durée de vie du réseau. Malgré les progrès qui ont été faits, la durée de vie du réseau continue d'être un défi majeur et un facteur clé, exigeant plus de recherches sur l'efficacité énergétique des plates-formes et des protocoles de communication [32].

L'optimisation de la consommation d'énergie exigeant la conservation de l'énergie à tous les niveaux de la pile de protocole le plus possible. Pour cela, les concepteurs au moment du développement des protocoles, négliger les autres métriques de performances au profit du facteur de consommation d'énergie. Les protocoles efficaces en énergie visent à minimiser la consommation d'énergie pendant l'activité du réseau.

Dans la littérature, il existe de nombreux mécanismes de conservation d'énergie et qui s'imposent comme les plus économes en énergie telles les techniques de routage efficace en énergie et ordonnancement de l'interface radio, agrégation de données spatio-temporelle [34].

Il existe bien évidemment beaucoup d'autres méthodes de conservation d'énergie. Par exemple, les mécanismes cross-layer et les paradigmes émanant de l'auto-organisation des systèmes [32].

1.6. Domaines d'applications des réseaux de capteurs

Actuellement, les réseaux de capteurs rencontrent une large palette d'applications. La miniaturisation des capteurs, leur coût de plus en plus faible, la diversité des types de capteurs de mesure disponibles (optique, thermique, chimique, etc.), et le support de communication sans fil utilisé ont permis d'élargir rapidement et continuellement leurs domaines d'application. Nous pouvons classifier les applications des RCSF en quatre classes d'applications [35]:

- application orientées temps (*time driven*) dans lequel chaque nœud capteur collecte les données et les envoie périodiquement à la station de base. L'acquisition des données capturées sont liées au temps : instant précis, période d'acquisition.
- application orientées événements (*event driven*) où chaque nœud capteur doit collecter les informations de façon continue et comparer à une valeur de seuil donnée. Le nœud capteur réagie immédiatement en cas des dépassements de cette valeur captée, il envoie la donnée à la station de base.
- application orientées requêtes (*query driven*) dans lequel un nœud capteur envoi de l'information intercepté suite à une demande explicite de la station de base.
- application hybride qui résulte de met en œuvre des deux types décrits précédemment. Par exemple, un réseau peut combiner entre un réseau de collecte de données par événements (*event driven*) et un réseau de surveillance (*time driven*).

Il a également été défini divers domaines d'application visé par les réseaux de capteurs :

1.6.1. Applications environnementales

La surveillance des paramètres environnementaux par les réseaux de capteurs peut donner création à plusieurs scénarios d'applications. En forêt, la dispersion des thermo-capteurs peut aider à détecter et de signaler un éventuel début de feu et ainsi faciliter la lutte contre les feux de forêt avant leur propagation. Dans les sites industriels, Le déploiement des capteurs sont généralement utilisés afin d'empêcher les risques industriels, comme des fuites de produits toxiques (produits chimiques, gaz, pétrole, éléments radioactifs, etc.). De même l'agriculture est susceptible d'avoir recours aux capteurs, ils peuvent être utilisés pour la réalisation de mesures afin de surveiller au mieux les conditions de développements tels que le processus d'irrigation lors de la détection de zones sèches dans un champ agricole. Il existe aussi différents applications pour surveiller les changements environnementaux et améliorer la prévision surtout pour les endroits difficilement accessible pour l'homme, comme par exemple les volcans, les régions polaires et les profondeurs des océans.

La détection des incendies de forêt dans le nord de l'Espagne est l'un de ces projets, Ce pays est touché chaque année par des incendies qui ciblent une zone de plus de 210 hectares. Les chercheurs ont installé donc dans les forêts (de cette zone) des capteurs Libelium (voir figure 1.9), permettant de détecter des incendies de forêt. L'objectif étant de fournir une infrastructure de surveillance de l'environnement, avec une capacité de gestion d'alerte rapide [36].

Figure 1.9 : Installation des capteurs Libelium en forêt pour détection des incendies (Le nord de l'Espagne)

1.6.2. Applications médicales

Certains usages des réseaux de capteurs tiennent du domaine médical, ces derniers connus par les réseaux de capteur corporels sans fil WBSN (Wireless Body Area Network). Ils sont conçus par plusieurs capteurs disposés sur ou à proximité du corps humain pour mesurer différents paramètres physiologiques (température, rythme cardiaque, rythme respiratoire, la glycémie,.. etc.) en différents endroits du corps. Les mesures effectuées sont grâce à des capteurs ayant chacun une tâche bien particulière. Ils peuvent faciliter la surveillance et de ce fait le diagnostic de quelques maladies. D'autre part, ces réseaux peuvent surveiller des activités à domicile chez les personnes dépendantes (handicapées ou âgées).

Le CGM (continuous glucose monitor) [37] peut être un dispositif de sauvetage pour les personnes avec n'importe quel type de diabète. Ils vérifient continuellement le glucose dans le sang de 24/24 heures. Les données sont transmis d'un capteur qui est insérer directement au-dessous de peau qui envoie les données au récepteur (voir figure 1.10).

Figure 1.10 : le capteur CGM (continuous glucose monitor) transmet l'information à un appareil d'enregistrement qui affiche les niveaux de glucose dans le sang [37]

1.6.3. Applications militaires

Les caractéristiques des réseaux de capteurs sont très adaptées pour les applications militaires. Il s'agit d'un réseau à : déploiement rapide, organisation autonome, faible coût et un faible taux de pannes. Ainsi, ce domaine pourra utiliser les réseaux de capteurs pour surveiller les mouvements des forces ennemies, surveillance des champs de bataille, Détection des attaques nucléaires, biologiques et chimiques. Plusieurs exemples intéressant de projets ont été lancés pour aider les unités militaires, comme le projet DARPA (Defense Advanced Research Projects Agency) [38], qu'il s'agit d'un réseau de capteur déployé sur les mines antichars, il est utilisé pour collecter des données distribuées afin de répondre aux attaques. Un exemple d'un projet militaire appelé JBREWS (Joint Biological Remote Early Warning System) [39], il est mis en place pour détecter et avertir les troupes dans le terrain de bataille des attaques biologiques. Un autre exemple appelé SPAN (Self-Powered Ad-hoc Network) [40].

Figure 1.11 : la disposition d'un capteur ISGS-SPAN-Sensor 460 sur le sol par un soldat [40]

Il est essentiellement un réseau de nœuds capteurs assez petit pour être placé dans des réceptacles aussi anodins que des petites pierres (voir figure 1.11). Ainsi, il constitue une solution rentable qui peut prendre en charge de nombreux types de missions, notamment la protection des frontières et la surveillance de zone.

1.6.4. Applications domotiques

Dans ces applications, le réseau de capteurs est déployé dans l'habitation où les capteurs sont intégrés dans des appareils domestiques (les fours à micro-ondes, les aspirateurs, les magnétoscopes, les réfrigérateurs,....) [41]. En effet, ces capteurs peuvent interagir entre eux et avec un réseau externe via internet afin de permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance. La surveillance de la luminosité, par exemple il s'éteint quand la chambre est vide, la surveillance de la température pour la climatisation et le chauffage, ou la détection de présence, par exemple quand un intrus veut accéder à la maison sont les essentielles applications visées par la domotique. Un tel réseau déployé doit permettre de créer une maison intelligente capable d'interpréter des situations suivant le comportement des occupants et d'en déduire des actions. Les maisons intelligentes portent aux utilisateurs un confort plus élevé en garantissant une vie plus simple et écologique (voir figure 1.12).

Figure 1.12 : le contrôle d'une maison grâce à un téléphone intelligent ou une tablette [42]

1.7. Défis et contraintes

La conception des protocoles et des techniques pour les réseaux de capteurs est influencée par de nombreux défis et contraintes, comme la tolérance aux pannes, l'environnement, la topologie dynamique de réseau ... etc. Ces facteurs rendent la conception et le développement

plus complexe. Dans la suite quelques défis et contraintes des réseaux de capteurs sans fil, peuvent être résumes comme suit :

1.7.1. Tolérance aux pannes

Le fonctionnement d'un nœud capteur peut être interrompu au cours du cycle de vie du réseau. Ce dernier peut échouer en raison du manque en ressources énergétiques, des dommages physiques, de problèmes de communication, compromission des nœuds, d'interférence environnementale ...etc. La tolérance aux pannes est donc la capacité de maintenir les fonctionnalités du réseau sans la moindre interruption due à dysfonctionnement d'un ou de plusieurs de ses nœuds capteurs. Les protocoles et les mécanismes peuvent être conçus pour évaluer le niveau de la tolérance aux pannes exigée par les réseaux de capteurs.

1.7.2. Le passage à l'échelle

Le nombre de nœuds déployés sur une zone de captage pour certaines applications peut atteindre des milliers. Dans ce cas, le réseau doit fonctionner avec des densités de capteurs très grandes. Ceci peut engendrer des problèmes de communication et de contrôle qui nécessite des protocoles capables de les gérer, ces protocoles doivent être capables de traiter un grand nombre d'évènements sans être saturés.

1.7.3. Topologie dynamique

Le changement de la topologie de réseau est l'un des aspects les plus importants dans les réseaux de capteurs sans fil. Ce changement topologique du réseau résulte à la défection d'un ou de plusieurs nœuds capteurs, la mobilité des nœuds capteurs, ainsi que l'ajout de nouveaux nœuds capteurs, ces raisons peut engendrer des difficultés de connectivité, on outre rend la topologie du réseau fréquemment instable. Dans ces cas il faut que les nœuds capteurs soient capables d'adapter leur fonctionnement dans le but de maintenir la topologie souhaitée. On distingue généralement trois phases dans la mise en place et l'évolution d'un réseau :

Déploiement : On peut distinguer plusieurs formes de déploiements d'un réseau de capteurs selon les besoins des applications. Les nœuds peuvent être déployés de manière prédéfinie, ou bien de manière aléatoire. Le déploiement aléatoire est adopté dans la majorité des scénarios à cause des raisons pratiques tels que le coût et le temps.

Post-Déploiement : la topologie du RCSF peut être affectée par des changements dus à la mobilité ou bien à des pannes des nœuds capteurs.

Redéploiement : L'addition des nouveaux nœuds capteurs dans un RCSF existant déjà implique aussi une remise à jour de la topologie.

1.7.4. La limitation des ressources

Les capteurs se caractérisent par une limitation de ressources dues à leur miniaturisation en termes de capacité de traitement, de mémoire, l'utilisation d'une communication sans fil et la réserve énergétique. L'énergie étant la contrainte la très forte, puisque dans la plupart des cas le remplacement ou bien la recharge des sources d'énergie des capteurs est quasiment impossible. C'est pour cela qu'il est nécessaire que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner plus longtemps.

1.7.5. L'environnement

Les capteurs peuvent être déployés en masse dans des endroits hostiles et sans aucune surveillance ni intervention humaine. Ils doivent être conçus pour résister aux différentes conditions climatiques telles que la pression, la chaleur, l'humidité, le froid...etc. Ainsi ils peuvent opérer dans des régions sous certaines contraintes, telles que: des tornades, des intersections encombrées, des surfaces contaminées biologiquement ou chimiquement, attachés à des animaux, dans un environnement dur tel que les champs de bataille, etc.

1.7.6. La sécurité

Pour les applications critiques des réseaux de capteurs, leur sécurité est une question essentielle qui demande un niveau de sécurité assez élevé, des mécanismes de confidentialité, d'authentification et d'intégrité doivent être mis en place au sein de leur communauté.
L'absence d'une protection physique des nœuds capteurs ainsi que la nature des liens sans fil, rend le réseau vulnérable aux attaques malveillantes. Les intrus peuvent espionner les communications et même modifier, reproduire et rejouer des messages. Par conséquent, il est nécessaire de mettre en œuvre des techniques de sécurité, qui tenir compte des ressources très limitées des nœuds capteurs.

1.8. Conclusion

Dans ce chapitre, nous avons présenté les réseaux de capteurs sans fil, les composants d'un capteur sans fil et ses fonctionnalités et les différentes topologies et organisation utilisées dans ce genre de réseau. Nous avons aussi décrit par la suite deux conceptions d'architecture

protocolaire dédiée aux RCSF : architecture en couche et architecture cross layer. Cependant, l'architecture cross-layer représente une solution intéressante pour remédier aux problèmes de ce réseau. Ainsi, elle a prouvé son efficacité par rapport à l'approche classique en couches.

Nous avons évoqué aussi le concept de consommation et de conservation d'énergie d'un nœud capteur, qui est indispensable à la compréhension des problématiques abordées dans le cadre de cette thèse. Ensuite, nous avons décrit quelques contraintes liés à la conception de ce type de réseaux.

Pour rappel, l'objectif de cette thèse est la sécurité des réseaux de capteurs sans fil, qui constitue l'une des défis majeurs et qui confrontent le bon fonctionnement de ces derniers. Ceci est dû principalement aux leurs ressources limitées, leur dispersion dans un environnement parfois hostile, leurs communications sans fil, et d'autres contraintes, etc. Dans le chapitre suivant, nous allons aborder le concept de sécurité dans les réseaux de capteurs.

2

La sécurité dans les réseaux de capteurs sans fil

Sommaire

2.1. Introduction

Les réseaux de capteurs sans fil (RCSF) peuvent être utilisés dans des applications critiques, particulièrement pour des applications du domaine médical, militaire, et autre, qui exigeant un aspect de sécurité strict sur les données échangées. Les caractéristiques inhérentes des RCSF les rendent particulièrement vulnérables aux différents types d'attaques qui peuvent compromettre le réseau ou endommagé son bon fonctionnement. Ces attaques exploitent le déploiement aléatoire des capteurs dans des zones généralement sans surveillance et la transmission des données par voie hertzienne. Assurer la sécurité des échanges au sein des RCSF est une tâche difficile, les capteurs qui les constituent comme mentionné précédemment, sont limités en termes de puissance, d'énergie, de capacité de communication et de calcul. Par conséquent, la mise en place des mécanismes de sécurité traditionnels sont inadaptables à ce type de réseau.

Dans ce chapitre, nous allons donner un aperçu sur les problèmes de sécurité dans les RCSF qui diffèrent des autres réseaux. Premièrement, nous présenterons les limites des réseaux de capteurs qui rendent la sécurité pour ce type de réseaux un véritable défi. Ensuite, nous allons identifier une taxonomie des attaques qui peuvent cibler ce type de réseaux. Puis, on va décrire les différents mécanismes de sécurité destinés aux RCSF, notamment en termes d'établissement et de gestion des clés.

2.2. Les objectifs de sécurité

Lorsque nous abordons le problème de sécurité dans les RCSF, nous visons à atteindre certains objectifs. En effet, les objectifs de la sécurité dans ce type de réseau ne sont pas différents de ceux dans les autres réseaux traditionnels. Elle vise à garantir que l'information provienne effectivement de la source légitime, soit correcte et qu'elle n'ait pas été modifiée. La sécurité associée avec les RCSF doit donc assurer les services de base suivants :

2.2.1. L'authentification

Ce service consiste à vérifier l'identité authentique des capteurs. Cependant, il permet aux capteurs de coopérer au sein des réseaux de capteurs sans risque. En effet, les capteurs utilisent un medium sans fil pour communiquer, qui est ouvert est confrontée aux menaces. Les mécanismes d'authentification sont essentiels pour détecter les paquets malicieusement falsifiés ou injectés, afin de s'assurer que les données ne parviennent pas d'un capteur malveillant.
Par conséquent, l'authentification est une solution qui permet de séparer les capteurs légitimes des ceux dits adversaires qui n'ont pas le droit d'intégrer le réseau. Elle est assurée grâce au Code d'Authentification de Message (CAM), ou MAC (pour Message Authentication Code en anglais) [43].

2.2.2. La confidentialité

Elle constitue l'un des objectifs de sécurité les plus importants dans les RCSF. Ce service désigne la garantie que l'information soit inaccessible aux adversaires. Cela signifie que l'information émise par chaque capteur n'est lisible que par le capteur auquel cette information est destinée. Un réseau fournissant la confidentialité des données dans leur sécurité empêchera toutes fuites d'information et fournira un support de communication sécurise pour ces capteurs afin de communiquer. Ceci est un point important pour les RCSF ou la

communication à travers le support sans fil est sensible à l'écoute. La cryptographie est la technique fondamentale utilisée pour assurer la confidentialité.

2.2.3. L'intégrité

Cette propriété permet d'assurer que le contenu d'un message n'a pas été modifié durant sa transmission dans le réseau, que ce soit volontairement ou accidentellement. En effet, la communication est fréquemment multi-sauts dans les RCSF. Ainsi un capteur intermédiaire compromis peut altérer un message entre l'émetteur et le récepteur. Garantir la confidentialité des données dans le réseau signifie bloquer toute essai d'injection de fausses données. Généralement, afin de vérifier l'intégrité le MAC (Message Authentication Code) et les signatures numériques sont utilisés.

2.2.4. La disponibilité

Elle désigne que le réseau est accessible pour assurer ses services et donc maintenir son bon fonctionnement. Ainsi en garantissant aux entités communicantes la présence et l'utilisation de la donnée au moment que l'on a souhaité. En effet, Il est impossible d'assurer de disponibilité des données à 100% dans un RCSF étant donné les contraintes qui pèsent sur ces réseaux, comme : la topologie dynamique, la communication sans fil qui peut être facilement brouillée ou perturbée par un attaquant, et les ressources limitées des capteurs de transit.

2.2.5. La fraicheur

Elle permet de garantir que les données sont fraîches. En effet, les données sont valides seulement dans un intervalle de temps limité. Par conséquent, lorsqu'un capteur reçoit un paquet de données, il doit être vérifié que les données transmises sont récentes ou non, et que l'adversaire n'a pas retransmis des vieux messages. Pour résoudre ce problème, un compteur ou bien un nombre pseudo-aléatoire peut être intégré aux paquets de données pour filtrer les vieux messages.

2.3. Les vulnérabilités de la sécurité dans les RCSF

Les RCSF possèdent plusieurs faiblesses que les exposent à différentes types d'attaques. En effet, ces faiblesses rendent aussi les mécanismes de sécurité utilisés pour les autres réseaux inapplicables à leur niveau. En conséquence, le développement de mécanisme de sécurité fiable doit être adapté aux caractéristiques de ce type de réseaux.

Certains faiblesses sont inhérentes aux RCSF et d'autres liées à la technologie retenue. Nous distinguons deux catégories : la vulnérabilité physique et la vulnérabilité technologique.

2.3.1. La vulnérabilité physique

Elle est liée à la nature de déploiement. En effet, le fait qu'un capteur est installé dans un lieu peu sûr notamment les lieux publics ou hostiles (forets, régions montagneuses) expose les liens de communication à des attaques. Ainsi que, les capteurs sont vulnérables à la capture physique et au vandalisme.

2.3.2. La vulnérabilité technologique

Est liée à plusieurs contraintes qui retour à la technologie des capteurs.

- **Limitation en énergie :** l'énergie est un facteur critique à considérer en concevant des mécanismes de sécurité, puisque les capteurs sont fréquemment déployés à des endroits hostiles, donc on ne peut pas changer les batteries ou les recharger. Alors, il est très important de minimiser la consommation d'énergie et de prolonger la durée de vie des batteries. Cette limitation impose la conception des mécanismes de sécurité à faible consommation énergétique.
- **Capacité de calcul limitée :** le capteur est doté d'un processeur d'une capacité de calcul très réduite ce qui empêche l'utilisation de mécanismes de protection cryptographiques qui exigeant plus de puissance de calcul.
- **Mémoire limitée :** Le capteur est un composant avec des ressources limitées en termes de mémoire. Par conséquent, il ne dispose pas suffisamment d'espace mémoire pour mémoriser le code de sécurité et les données relatives (tel que les clés de chiffrage). Ce qui signifie que n'importe quel mécanisme de sécurité conçue pour des RCSF devrait être très compacte en termes de taille du code.
- **Transmission/réception :** les capacités de transmission/réception du capteur sont limitées pour des besoins de conservation d'énergie. En effet, la transmission est particulièrement l'opération la plus couteuse d'un point de vu énergétique dans les RCSF (la transmission d'un bit est équivalent à environ 800 à 1000 opérations CPU [44]). C'est pourquoi dans la conception de mécanisme de sécurité il n'est pas possible d'utiliser des mécanismes compliqués, qui impliquent l'échange d'un nombre important de messages entre les capteurs. D'un autre point, le média de communication sans fil est un obstacle à la sécurité, il est ouvert et accessible à tout le monde. Par conséquent, un attaquant peut facilement altérer, retransmettre ou intercepter les transmissions issues du RCSF.

2.4. Classification des attaques dans les RCSF

Les différentes contraintes de sécurité des réseaux de capteurs sans fil les exposent à divers types d'attaques. Ainsi, le choix d'un mécanisme de défense doit se reposer sur une modélisation de l'attaque, mais doivent aussi tenir compte de la nature de l'attaquant et de ses caractéristiques. Les attaques sur les RCSF connaissent plusieurs classifications possibles dont les plus utilisées [45-48]. Ci-dessous nous citons les classifications les plus connues :

2.4.1. Attaques passives/actives

Les attaques passives se limitent uniquement à l'écoute et l'analyse du trafic, l'interception et l'espionnage des informations échangées. Ces attaques peuvent être facilement réalisées et ils sont difficilement détectés, ce qui les rend très dangereuses. Ainsi un attaquant passif ne fait que menacer la confidentialité des données ou bien la détermination des nœuds capteurs importants dans le réseau (exemple: chef de groupe). Une fois l'attaquant ayant obtenus suffisamment d'informations, il peut utiliser pour commencer d'autres types d'attaques. Tandis que les attaques actives, un attaquant essaye de supprimer ou modifier les paquets transmis sur le réseau. Il peut même injecter son propre trafic ou rejet de paquets afin de détruire le fonctionnement du réseau d'une manière totale ou bien partielle. Par conséquent, Un attaquant actif menace l'authenticité et la confidentialité des données aussi bien que leur intégrité.

2.4.2. Attaques internes/externes

Dans le cas des attaques externes, le capteur attaquant n'est pas reconnue comme faisant partie du réseau. Un attaquant extérieur n'a pas forcément à une connaissance de la façon dont fonctionne le réseau. Il peut mener à des attaques passives sans ces informations telles que le brouillage radio, l'attaque par rejeu ou l'écoute clandestine. Par contre, les attaques internes sont effectuées par des capteurs légitimes (appartenant au réseau). Cependant, les attaques internes sont considérées comme les menaces les plus dangereuses qui peuvent perturber le fonctionnement du réseau de capteur. Elle pourra être effectuée en utilisant des capteurs du réseau qui ont été capturés physiquement et par conséquent l'attaquant peut lire sa mémoire et avoir accès à son matériel cryptographique. Ce dernier est plus subtil, plus délicat à détecter si aucune méthode de détection d'intrusion n'a été implémentée.

2.4.3. Attaques orientées selon les couches protocolaires

Une autre méthode de classification catégorise les attaques en se basant sur les couches protocolaires. L'architecture en couches des RCSF rend vulnérables à divers types d'attaques.
Ce classement permet une revue efficace, couche par couche, de la plupart des attaques connues. C'est donc selon ce critère que nous allons maintenant présenter les principales attaques connues dans les réseaux de capteurs.
Cependant, on peut les classer selon la couche ciblée.

2.4.3.1. Les attaques ciblant la couche physique

La couche physique correspond au médium physique utilisé afin de transmettre des données entre deux nœuds. Une attaque qui cible cette couche vise généralement à créer des interférences pour occuper les canaux et empêcher les nœuds capteurs de communiquer normalement. Un attaquant peut transmettre en continu des signaux radio sur un canal sans fil. Il peut aussi envoyer des signaux à haute énergie afin de bloquer efficacement le support sans fil et d'empêcher les capteurs de communiquer.

2.4.3.2. Les attaques ciblant la couche de liaison de données

Les attaquants peuvent exploiter les comportements de protocole prédéfinis au niveau de la couche liaison pour lancer des attaques contre cette couche. Donc, ils peuvent provoquer des collisions afin de causer une interférence, provoquer un épuisement des ressources énergétiques des capteurs par des retransmissions répétées des paquets, ou intercepter des messages afin d'acquérir des informations.

2.4.3.3. Les attaques ciblant la couche réseau

La couche réseau des RCSF est vulnérable aux différents types d'attaques, telles que les attaques DoS [49] qui visent à perturber complètement les informations de routage, et donc l'ensemble du fonctionnement du réseau. Une attaque de Sinkhole [50] tente d'achemine presque tout le trafic vers le capteur malveillant. L'attaquant va convaincre ses voisins comme étant la station de base ou du cluster-head. Par conséquent tous les paquets reçus seront modifiés et envoyer à la station de base. Les informations de routage falsifiées, altérées ou rejoué sont les attaques les plus directes lancées contre un protocole de routage afin de perturber le trafic sur le réseau.

2.4.3.4. Les attaques ciblant la couche transport

Des attaques peuvent profiter les spécifications de la couche transport : par exemple, un attaquant peut émettre un nombre considérables d'informations, telle qu'une demande d'une nouvelle connexion. Ainsi épuiser les ressources d'un capteur, qui atteignent rapidement la limite maximale (saturation). L'attaque de désynchronisation est un autre exemple d'attaque dans cette couche [49]. Le but de ce dernier est de perturber le protocole de communication en modifie les numéros de séquence des paquets.

2.4.3.5. Les attaques ciblant la couche application

Différents types d'attaques peuvent être effectuées dans cette couche, telles que Overwhelm[9], la répudiation, la corruption de données. En cas d'attaque Overwhelm[51], un attaquant amène le réseau à acheminer de gros volumes de trafic vers la station de base. Ce type d'attaque consomme de la bande passante de réseau et épuiser l'énergie des capteurs.

Chacune de ces méthodes de classification aide à voir les attaques selon différentes visions. Nous avons illustré dans la figure 2.1 les attaques les plus connues dans les RCSF selon trois classifications parmi celles citées ci-dessus.

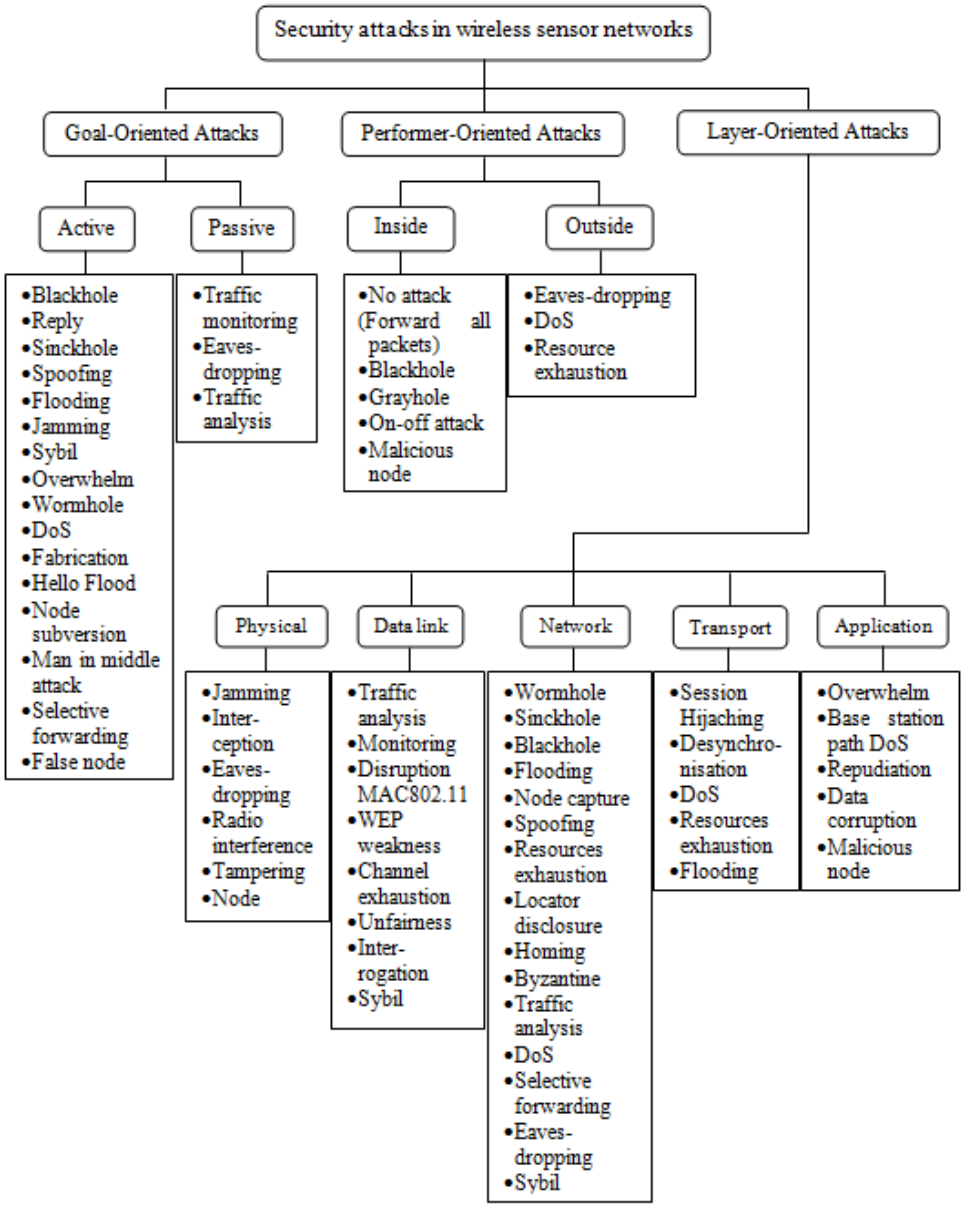

Figure 2.1 : Classification des attaques dans les RCSF [52]

2.5. Description de quelques attaques

En général, la probabilité d'attaque contre les RCSF est supérieure à celle de tous les autres types de réseaux, et cela en raison de leurs ressources limitées et l'environnement de déploiement. Les attaques se créent fréquemment par l'insertion d'éléments adversaires dans le réseau. Il existe aussi des attaques extérieur au réseau (contre l'environnement), les quelles provoquent des interférences ou des altérations sur les signaux envoyer. Dans cette section, nous présentons les attaques les plus connues dans les RCSF.

Ecoute du réseau (eavesdropping)

Du fait de la nature de support de transmission dans les RCSF, aucune vérification d'accès au réseau n'est possible. Il est donc très facile qu'un nœud malveillant peut intercepter des messages échangés et accéder à leur contenu. Ainsi, l'intrus peut espionner et capter des données stratégiques qui peuvent aider au lancement d'attaques plus dangereuses.

Brouillage radio (jamming)

C'est une attaque qui vise les médias de communication utilisés dans les RCSF. L'attaquant cherche en général à émettre un signal parasite (un bruit) d'une fréquence proche de celle utilisée par les nœuds capteurs dans le réseau, de façon à ce que le nœud attaqué ne puisse plus recevoir de manière correcte les paquets qui lui sont transmises par les nœuds légitimes, et provoque ainsi l'indisponibilité des canaux de communication [53]. L'intrus peut lancer des attaques de brouillage stratégiques en affectant des zones sensibles du réseau.

Attaque de collision (Collision attack)

L'attaquant cherche à produire des collisions en transmettant un signal en même temps qu'un nœud capteur légitime, afin de causer une interférence. Autrement dit, le récepteur ne puisse pas recevoir correctement le paquet qui lui est destinée. En particulier, le nœud malveillant ne respecte pas les conditions d'accès au média de communication en vérifiant le medium de communication pour assurer qu'il est occupé (réception des paquets RTS et CTS). Si c'est le cas, il émet un signal afin de chevaucher avec les communications existantes [54].

Attaque de l'identité multiple (Sybil attack)

Dans cette attaque, un nœud malicieux peut revendiquer un nombre important d'identités afin d'altérer le fonctionnement des autres nœuds du réseau. Chaque identité peut être dupliquée d'une identité d'un nœud de capteur légitime qui existe déjà, ou générée aléatoirement [55]. Dans la figure 2.2 le nœud malveillant prend une place dans le réseau et reçoit les paquets de plusieurs nœuds, ici A, B, C et D. Il prétend être ces derniers auprès de X après avoir usurpé leur identité. Si X ne possède pas le moyen de vérifier l'identité des expéditeurs, le nœud malveillant jouera le rôle des autres nœuds.

De cette façon, ce type d'attaque a un effet important sur les protocoles de routage géographique qui nécessitent les informations de la localisation d'un nœud de capteur pour lui router efficacement les paquets. En outre, le nœud capteur attaquant peut exploiter les multiples identités pour être élue comme chef de groupe.

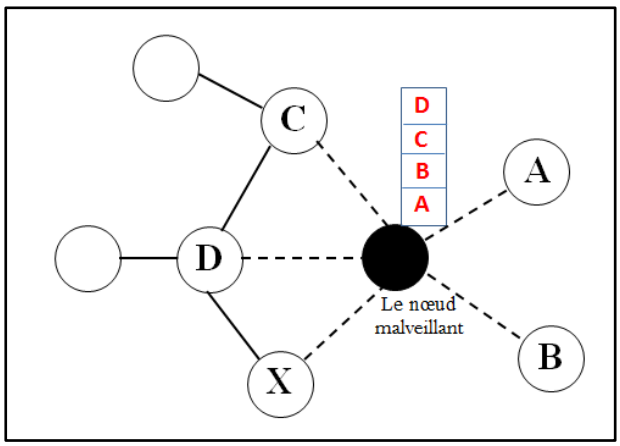

Figure 2.2 : Attaque de l'identité multiple (Sybil attack)

Attaque du trou de ver (wormholes)

Cette attaque consiste à intercepte les paquets en un point donné du réseau pour le réinjecter en un autre point. Il faut donc au moins deux nœuds malveillants complices, l'un qui capture et l'autre qui injecte les paquets, et qui communiquent l'un avec l'autre à l'aide d'un canal auxiliaire distinct des canaux légitimes utilisés sur le réseau (un tunnel, ou « trou de ver ») [50]. La figure 2.3 illustre un exemple d'attaque du trou de ver. Ainsi, les nœuds malveillants permettent d'atteindre un endroit éloigné avec un seul saut. Cela trompera les autres nœuds capteurs sur les distances qui séparent les deux nœuds, mais va surtout forcer les nœuds voisins à passer par les nœuds malveillants pour faire circuler les paquets.

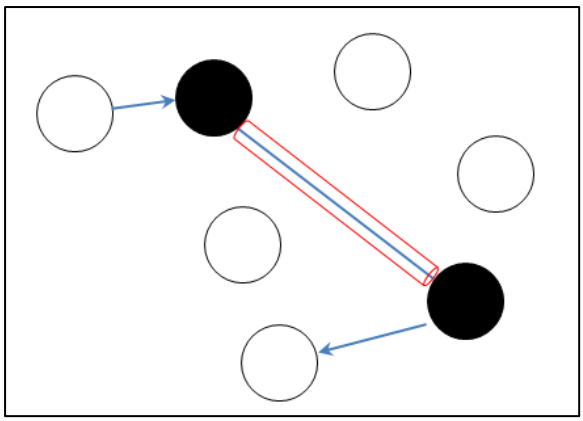

Figure 2.3 : Attaque du trou de ver (wormholes)

L'attaque de trou de puits (Sinkhole attack)

Dans ce cas, l'attaquant essaye d'attirer le maximum de paquets possible vers le nœud compromis afin de contrôler la majorité des informations circulant dans le réseau. En effet, le nœud compromis va convaincre ses voisins que c'est le nœud le plus proche de la station de

base. Ainsi, il peut supprimer, rejeter ou modifier les paquets reçus, ce qui altéré le bon fonctionnement du réseau [49-50] [55]. Par exemple, la figure 2.4 montre que le nœud malveillant se comporte comme un voisin de la station de base afin d'absorber tout le trafic.

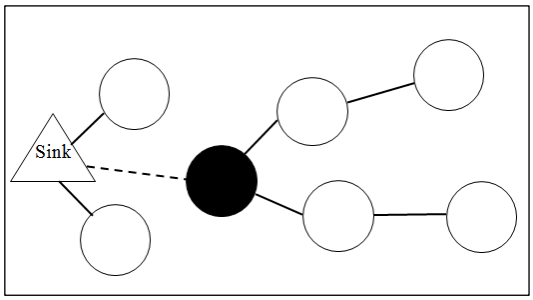

Figure 2.4 : Attaque de trou de puits (Sinkhole attack)

Attaque du trou noir (black hole) :

Ce type d'attaque consiste pour un nœud compromis à falsifier les informations de routage pour forcer le passage des paquets par lui-même. Ensuite, le nœud compromis n'effectuer aucun transfert des paquets qui lui sont envoyés, créant ainsi une sorte de «trou noir» dans le réseau, qui aspire toutes les données. Le nœud compromis peut aussi s'installer dans un emplacement de routage stratégique, mène donc à la suppression des paquets en cours de transit et par conséquent la mise hors service de tout le réseau [56].

Attaque d'inondation par paquet de Hello (Hello flood attack)

Les nœuds capteurs utilisent des paquets Hello pour découvrir leurs voisins et ainsi déterminer une topologie du réseau, qui n'est pas établie au préalable. En plus, l'échange de ce type de paquet périodiquement est très important pour la réorganisation et la localisation de la majorité des protocoles de communication. L'attaquant peut exploiter les paquets Hello, il peut utiliser un transmetteur de haute puissance pour tromper un nombre important de nœuds capteurs en leur faisant croire qu'ils sont dans son voisinage. Ainsi, les nœuds capteurs recevant les paquets Hello vont envoyer leurs données à cet attaquant. Dans ce cas l'attaquant va inonder le réseau et empêcher d'autres paquets d'être échangés.

Attaque par rejeu (Replay attack) :

Dans ce cas, les paquets échangés entre les nœuds capteurs sont enregistrés et retransmis par l'attaquant afin d'épuiser l'énergie des nœuds récepteurs. D'une autre coté, les paquets peuvent contenir des données de routage et données de perception prélevées ou de configuration. L'attaquant peut modifier ces données et les réutiliser pour générer des faux messages, et par conséquent il détourner le réseau de son but initial.

L'attaque de désynchronisation (De-synchronization)

L'attaquant fait croire à un nœud capteur (victime) qu'il n'a pas reçu un certain nombre de paquets en faussant la séquence des paquets intercepter qui est réellement destinée à un autre nœud capteur dans le réseau. L'attaquant réalise ce type d'attaque en augmentant le numéro de séquence des paquets passants par lui plus que prévu auprès du nœud victime. Ainsi, il incite le destinataire à demander la retransmission des paquets manquants auprès des expéditeurs. Par conséquent, l'attaquant pousse sa victime à dissiper son énergie en tentant de réparer les erreurs de transmission qui n'ont pas vraiment existées [51].

2.6. Mécanismes de sécurité

Plusieurs mécanismes, sont mis en place afin de répondre aux problèmes de sécurités dans les RCSF. En effet, dans le cadre du développement d'un mécanisme de sécurité, il faut toujours assurer un compromis entre la sécurité garantie et le surcoût imposé par le mécanisme appliqué.

Dans un contexte de réseau de capteurs, les mécanismes de sécurité existants tentent de protéger le matériel des nœuds capteurs, le canal de communication et les protocoles et services. En protégeant le matériel des nœuds capteurs, il est possible de détecter et/ou prévenir les attaques qui tentent de compromettre un nœud capteur. Un canal de communication sécurisé ne peut pas être affecté par la plupart des attaques courantes (écoute, modification, rejeu et injection) qui affectent l'échange des paquets entre les nœuds capteurs. Enfin, avec le support adéquat, les protocoles et services utilisés dans le réseau peuvent tolérer des perturbations de service et des attaques complexes. Nous citons dans ce qui suit ces axes de mécanismes de sécurité proposés contre les attaques ou les comportements malicieux

2.6.1. Protection matérielle

Les RCSF sont très souvent déployés dans des zones hostiles et sans aucune protection. Par conséquent, les nœuds capteurs sont très exposés aux attaques d'altération physique qui causent généralement d'autres types d'attaques. En effet, un attaquant aura pour but d'extraire les clés cryptographiques, reprogrammer le nœud capteur pour perturber le réseau ou même remplacer le nœud capteur par un nœud malicieux. Par conséquent, il devrait y avoir certains types de protection matérielle pour éviter de telles attaques. Parmi ces protections, nous citons :

TPD (Tamper-Proof Device)

Un dispositif considéré comme inviolable, il permet de stocker les informations sensibles (les informations confidentielles, les clés privées) de manière sécurisée afin d'empêcher un attaquant de récupérer ces informations lorsque le nœud capteur est compromis. Une fois qu'une altération de la puce est détectée, le TPD détruit toutes les informations stockées [57].

L'obfuscation de code

En général, l'obfuscation de code consiste à cacher les détails d'implémentation d'un programme à un adversaire. Les programmeurs obscurcissent leur code dans le but de rendre difficile la détermination et l'extraction des données du code. L'obscurcissement du code et des données, cause une augmentation dans le temps nécessaire pour qu'un attaquant analyse les nœuds compromis, Il sera donc plus difficile de déduire les secrets du contenu extrait des mémoires : flash, l'EEPROM ou SRAM [57].

2.6.2. Des canaux de communication sécurisés

La caractéristique la plus évidente d'un RCSF est que la communication se passe sur un canal sans fil, le milieu sans fil est habituellement un canal radio. Ainsi, il est ouvert et accessible à tout le monde. Pour Cela, le support de communication sans fil est à son tour un obstacle à la sécurité, si on considère l'existence d'un attaquant, le canal peut subir toute forme d'attaque de communication telle que l'interception, accès au réseau, et déni de service. Par conséquent, il est nécessaire d'établir un canal de communication sécurisé entre les nœuds capteurs, où aucun attaquant ne peut endommager l'échange des messages. Afin de créer ce canal, il est nécessaire d'utiliser des primitives cryptographiques, et il est également essentiel d'établir les informations de sécurité (clés secrètes) nécessaires à ces primitives.

2.6.2.1. Primitives cryptographiques

Les différentes problématiques de sécurité discutées précédemment nécessitent des solutions basées sur l'utilisation des primitives cryptographiques, qui sont les briques servants à construire les protocoles de sécurité. Pour cela nous allons aborder dans ce qui suit les diverses primitives cryptographiques destinées aux réseaux de capteurs sans fil.

(i) La cryptographie

L'une des premières solutions de sécurité qui répond à l'ensemble des problèmes liés à la sécurité des informations est la cryptographie. Cette dernière permet de chiffrer et d'authentifier les messages échangés entre les nœuds capteurs. Les techniques cryptographiques produisent sont par nature un nombre élevé de calculs. Par conséquent, les algorithmes les plus complexes ne peuvent pas souvent être implémentés sur les nœuds d'un RCSF. Ainsi, les algorithmes cryptographiques doivent être légers pour s'adapter à la nature des ressources limitées de ce réseau. On distingue deux types de cryptographies permettant de garantir chacune un certain nombre de propriétés :

- Cryptographie symétrique

La cryptographie symétrique également dite à clé secrète. Elle se base sur l'utilisation d'une seule clé de chiffrement. Elle nécessite pour fonctionner que la même clé est utilisée entre deux nœuds capteurs communicants pour chiffrer et déchiffrer les informations en utilisant un algorithme de chiffrement symétrique (voir la figure 2.5). Il existe deux catégories d'algorithmes de chiffrement symétrique :

- **Le chiffrement par bloc :** Il se caractérise par une fragmentation des données en bloc de taille fixe généralement comprise entre 64 et 512 bits. Les blocs seront chiffrés les uns après les autres selon différents modes, les plus utilisés : CBC (Cipher Block Chaining), ECB (Electronic Code Book) ou OFB (Output Feedback) [58]. Les algorithmes de chiffrement par bloc les plus connus et utilisés sont le DES (Data Encryption Standard) [59], l'AES (Advanced Encryption Standard) [60] et RC5 (Rivest Cipher 5) [61].
- **Le chiffrement en flot :** Il est fait bit à bit sans attendre la réception totale des données, avec lequel on opère un XOR (ou exclusif) entre un bit d'un flux de bits aléatoire et un bit provenant des données. En cas de déchiffrement, les bits des données chiffrées sont combinés par l'opération XOR avec le même flux de bits aléatoire pour retrouver les bits des données. Dans ce type de chiffrement, le flux de bits aléatoire est générer par un générateur pseudo aléatoire, qui est initialisé avec la clé partagée de chiffrement. Un exemple de chiffrement à flot est l'algorithme RC4 (Rivest Cipher 4) [62].

Les solutions de chiffrement à clés symétriques sont implémentées au sein des RCSF et apportent une réelle solution pour la sécurité du réseau de capteur.

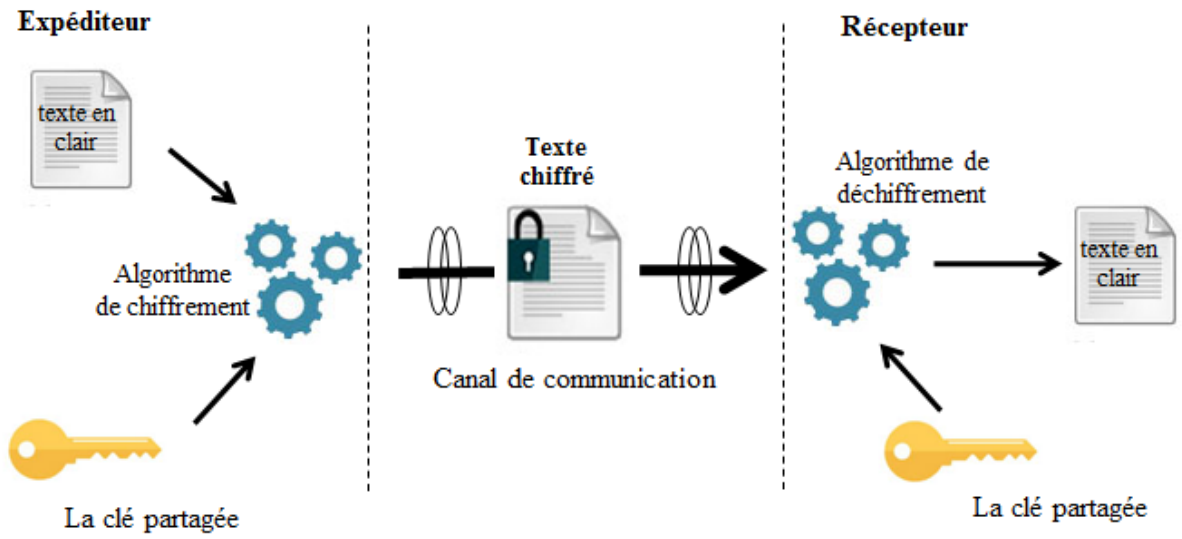

Figure 2.5 : Cryptographie symétrique

- **Cryptographie asymétrique**

La cryptographie asymétrique ou à clé publique se base sur l'utilisation de deux clés différentes : une clé publique diffusée à tous les nœuds servants au chiffrement de données qu'ils vont émettre au récepteur, et une clé privée maintenue secrète chez le récepteur servant pour le déchiffrement de ces données lorsque ce dernier les reçoit. Ainsi, l'émetteur peut utiliser la clé publique du récepteur pour chiffrer un message que seul le récepteur (en possession de la clé privée) peut déchiffrer. Elle permet non seulement de garantir la confidentialité des données en chiffrant et déchiffrant des messages mais aussi garantir l'authentification de l'auteur d'un message en utilisant la signature numérique, dans ce cas l'émetteur peut utiliser sa propre clé privée pour signer un message et le récepteur peut vérifier la signature du message à l'aide de la clé publique correspondante. La figure 2.6 présente un mécanisme de chiffrement basé sur la cryptographie asymétrique.

La construction de ce type de cryptographie repose sur différents problèmes mathématiques, qui se présentent à cause des calculs utilisés pour déchiffrer les données.

En effet, la complexité de chiffrement asymétrique n'exige que le nœud capteur à une capacité de traitement et de stockage plus élevée et une consommation d'énergie plus haute.

Parmi les exemples d'algorithmes de chiffrement à clé publique nous citons : le RSA (Rivest Shamir Adleman) [63] et l'ECC (elliptic curve cryptography)[64].

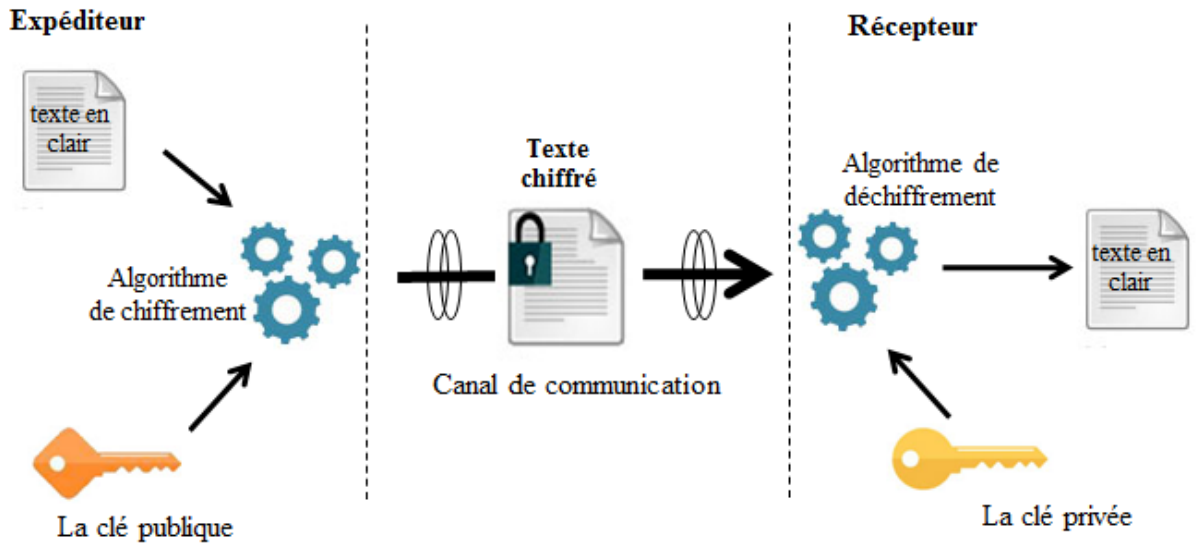

Figure 2.6 : Cryptographie asymétrique

(ii) Fonction de hachage

Une fonction de hachage cryptographique consiste à calculer une courte empreinte de taille fixe à partir d'un bloc de données de taille arbitraire. En général, Il est facile de calculer l'empreinte à partir du contenu du message à transmettre, tandis qu'il est difficile de trouver le contenu du message à partir de l'empreinte (c.à.d. très difficile à inverser). C'est pourquoi la fonction de hachage est dite à sens unique. Cette empreinte est recalculée par le destinataire afin qu'il la compare à celle calculée par l'expéditeur. Si elles sont différentes, alors les données ont été modifiées pendant leur transmission. Les fonctions de hachage sont souvent utilisées comme un mécanisme qui vérifie l'intégrité d'un message ou bien pour générer des signatures numériques.

Une fonction de hachage possède les trois propriétés suivantes :

- Sens unique : quand la valeur hachée h (tel que $h = H(M)$) pour un message M est donnée, il est impossible de trouver le message M tel que $H(M) = h$;
- Résistante faible aux collisions : à partir d'un message M, il est impossible de trouver un message M' tel que $H(M') = H(M)$;
- Résistance forte aux collisions : il est impossible de trouver deux messages distincts M et M' tel que $H(M) = H(M')$.

Parmi les fonctions de hachage les plus utilisées sont: SHA-1 (*Secure Hash Algorithm*) [65] et MD5 (*Message Digest 5*) [66].

(iii) Le code d'authentification de message

Le code d'authentification de message (CAM), ou MAC (pour Message Authentication Code en anglais) [43] permettant d'assurer d'une part l'intégrité du message transmis et d'autre part l'authenticité de l'expéditeur. Un MAC consiste à utiliser une fonction cryptographique de hachage combinée à une clé secrète (symétrique) connue uniquement par les deux entités communicantes (échangeant le message). Autrement dit, le MAC est un algorithme qui prend en entrée un message *M* à transmettre et une clé secrète *K* et qui produit un condensé. En effet, ce condensé est par la suite transmis avec les données. Le destinataire calcule à son coté le condensé MAC avec la même clé partagée avec l'expéditeur et le compare au condensé qu'il a reçu. S'ils sont identiques, alors l'expéditeur du message est authentique et les données n'ont pas été modifiées. Dans la figure 2.7, le MAC est illustrée par un scénario d'authentification.
Un exemple de MAC utilisé dans la pratique est le HMAC (Hash Message Authentication Code) [67]. Ce dernier (c.à.d. le HMAC) peut utiliser n'importe quelle fonction de hachage, comme SHA-1 ou MD5.

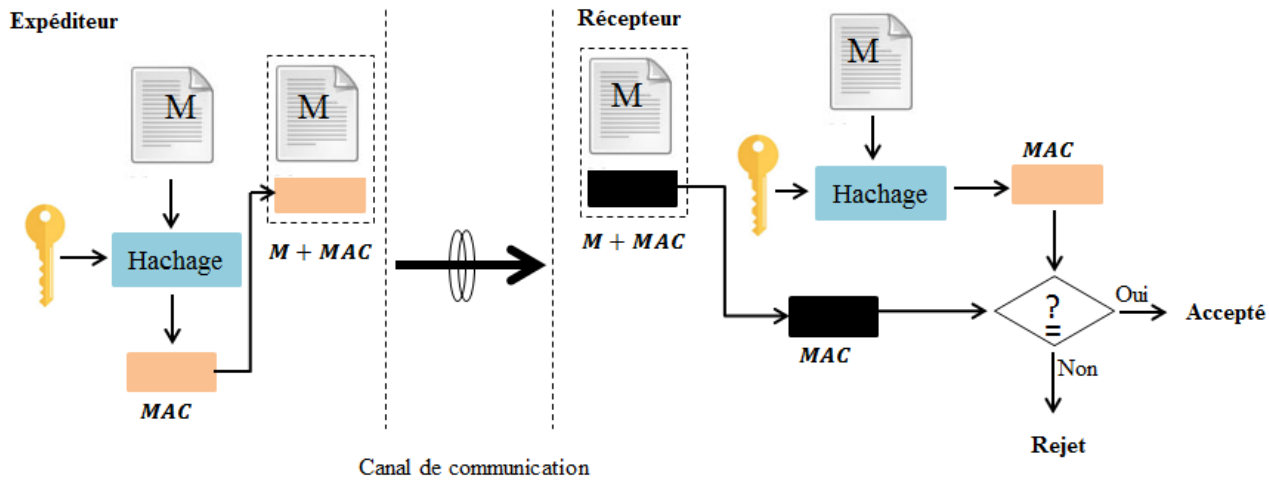

Figure 2.7 : Le code d'authentification de message MAC

2.6.2.2. La gestion de clés

Comme nous l'avons vue dans la partie précédente, la sécurité des communications dans un RCSF commence par la protection des liens entre chaque paire de nœuds de ce réseau. Elle pourra être assurée à l'aide de primitives cryptographiques (c.à.d. l'utilisation du chiffrement, déchiffrement, de la signature, etc.), pour cela les nœuds de capteur doivent partager certaines informations d'identification de sécurité. Par exemple, dans la cryptographie symétrique,

lorsqu'un nœud *A* crypte l'information avec une clé secrète *K*, l'autre nœud *B* aura besoin de la même clé secrète *K* pour obtenir l'information originale par décryptage. La tâche d'établissement et de maintien (protection, sauvegarde, distribution, renouvellement, chargement, etc.) de ces clés doit être effectuée par un système de gestion de clés [68].
Il reste difficile à réaliser dans les RCSF en raison des contraintes de limitation de ressources telles que: la bande passante, l'autonomie des batteries, la mémoire disponible, la portée des transmissions radio et le déploiement aléatoire. Ainsi, la gestion de clés doit assurer le couplage des caractéristiques propres aux RCSF aux exigences de sécurité telles que la confidentialité, la disponibilité, l'intégrité, et l'authentification des nœuds communicantes.

2.6.3. Protocoles et services: protocoles de base

Bien que la protection du canal de communication protège les RCSF contre certaines attaques, elle ne garantit pas entièrement que d'autres attaques ne les affectent pas. Par exemple, une attaque DoS [49] peut conduire à une dégradation d'un ou plusieurs services réseau et delà compromettre la disponibilité du réseau, et d'autres attaques spécifiques aux protocoles élaborées peuvent perturber, détruire ou corrompre un réseau. Cependant, elles peuvent être n'importe quel événement qui diminue ou élimine la capacité du réseau d'exécuter ses fonctions attendues. Par conséquent, il est nécessaire de créer des protocoles et services spécialisés capables de soutenir adéquatement la protection des données.
Nous concentrons dans ce qui suit sur les principaux mécanismes qui sont utilisés pour sécuriser les protocoles de base du réseau RCSF à savoir : les mécanismes permettant de sécuriser le routage, l'agrégation de données, la localisation et la synchronisation temporelle.

2.6.3.1. La sécurité du routage

Les protocoles de routage destinés aux RCSF doivent satisfaire certaines exigences et résoudre certains problèmes afin de garantir le bon fonctionnement de la découverte de route entre la source et la destination même en présence des nœuds malicieux. Le problème du routage consiste à trouver un chemin optimal pour envoyer des paquets à travers le réseau au sens d'un certain critère de performance comme la consommation énergétique. Pour cela, il faut être capable de déterminer une route qui consomme moins d'énergie. La sécurité est un autre facteur qui ne peut être ignoré dans la conception des protocoles de routage. Il existe différents menaces aux protocoles de routage: par exemple, une attaque simple du trou de ver peut faire croire à deux nœuds éloignés qu'ils sont très proches alors qu'en réalité ils sont distants de

plusieurs sauts. Autrement dit, en présence de telles attaques, les nœuds du réseau cessent de continuer d'assurer la fiabilité de leur service. Il est donc nécessaire de sécuriser les protocoles de routage pour mener à bien l'opération de l'acheminement des données.

2.6.3.2. La sécurité de l'agrégation de données

Les réseaux de capteurs se composent de milliers de capteurs capables de générer une quantité importante des données produites de leurs mesures. Dans la plupart des cas, toutes ces données doivent être envoyées à la station de base, ce qui entraîne un coût important, en termes d'utilisation de la bande passante et de consommation d'énergie, lors du transit de toutes les données des nœuds vers la station de base. Cependant, étant donné que les nœuds sont physiquement proches les uns des autres, il y aura une certaine redondance des données. Le rôle de l'agrégation est d'exploiter cette redondance en collectant les données d'une certaine région et en les résumant en un seul rapport, diminuant ainsi le nombre de paquets envoyés à la station de base. Agréger les données au niveau des nœuds capteurs intermédiaires est une approche courante pour surpasser les limitations des RCSF.

La sécurité des techniques d'agrégation a rencontré plusieurs difficultés lors de la mise en œuvre parce que les données agrégées peuvent être facilement attaquées par un adversaire malveillant, même si les communications sont protégées contre toute attaque par injection de données ou attaque d'intégrité des données. Le risque majeur de l'agrégation est quand un nœud d'agrégation est compromis, ce qui met en danger toutes les données mesurées qui font partie de l'agrégat dont le nœud capteur est responsable. En effet, si un nœud agrégateur est surveillé par un adversaire, il peut facilement ignorer les données reçues de ses voisins et créer un faux rapport. Les agrégateurs de confiance peuvent toujours recevoir de fausses données des nœuds défectueux ou des nœuds surveillés par un adversaire. Plusieurs solutions ont été proposées pour sécuriser l'agrégation, une solution possible consiste à essayer de découvrir si les rapports envoyés par un agrégateur malveillant sont falsifiés ou non. Une autre approche consiste à interroger l'agrégateur lui-même sur les données utilisées pour créer le rapport. D'autres approches tirent parti de la densité des réseaux de capteurs en utilisant comme témoins les nœuds situés au voisinage de l'agrégateur.

2.6.3.3. La sécurité de la localisation

Dans les RCSF, la localisation est un facteur très important pour assurer la fiabilité de leur fonctionnement. Dans un nombre important d'applications des réseaux de capteurs, les nœuds

sont généralement déployés aléatoirement. La plupart de ces applications (militaires, suivis des animaux, ...) exigent la connaissance de la position physique des nœuds capteurs afin de pouvoir localiser l'origine des événements détectés. En effet, l'utilité d'un RCSF se fondera sur ses capacités de localiser automatiquement chaque capteur dans le réseau. Ainsi, un réseau de capteurs conçu afin de détecter des événements aura besoin d'informations précises sur l'endroit pour repérer exactement la position de ces derniers. En outre, la localisation peut être utilisée aussi pour d'autres aspects tels que dans l'identification des données, et dans les protocoles de routage géographique dans les réseaux à grande échelle. Par conséquent la sécurisation des mécanismes de localisation est nécessaire pour protéger le réseau des adversaires malicieux qui tentent de compromettre les informations de localisation afin de perturber le fonctionnement du réseau.

2.6.3.4. Les systèmes de détection d'intrusions

Les mécanismes de sécurité présentés précédemment seuls ne suffisent pas pour assurer une sécurité optimale du réseau de capteur. En effet, un nœud adversaire peut compromettre un nœud légitime afin d'accès non autorisé dans le réseau. Le nœud adversaire utilisera toutes les informations capturées du nœud cible pour surmonter les contrôles d'authentification et déchiffrer toutes les données codées. Ainsi, il est généralement nécessaire d'ajouter à ces mécanismes des systèmes de détection d'intrusions pour compléter les fonctions de sécurité et garantir un niveau élevé de sécurité.

Un système de détection d'intrusion (SDI ou IDS pour Intrusion Detection System) [69] se charge de détecter les activités anormales ou suspectes sur la cible analysée et déclenchera une alarme lorsqu'une tentative malveillante se produite.

Une fois le réseau déployé, et le système de détection d'intrusion mis en œuvre, on surveille et analyse le comportement du système cible que l'on veut sécuriser. Au moment d'analyse, les SDI doivent être en mesure de faire la distinction entre les comportements normaux et anormaux afin de découvrir n'importe quelle tentative malveillante. Selon la classe de comportements prise en considération, on fait le choix d'une technique de détection qui est une fonction cruciale qui affecte le niveau de sécurité. On a de manière générale deux techniques principales pour l'analyse et la détection des comportements malicieuses:

- **Technique de détection à base de signature :** Cette approche consiste à comparer l'action observée d'un nœud avec un ensemble de signatures (actions habituellement effectuées par des adversaires) d'adversaires répertoriés par le système de détection. On détecte que le nœud

analysé est défini comme étant un adversaire quand on parvient à trouver une signature parmi les actions analysées.

- **Technique de détection à base d'anomalies:** Cette approche est focalisée d'abord sur la modélisation d'activité normale d'un nœud et puis identifier tout ce qui s'éloigne de ce modèle de référence comme étant une anomalie. L'avantage principal de cette technique est de pouvoir détecter les attaques inconnues.

Le système de détection d'intrusion (SDI) reste une tâche importante qui complété les fonctions de sécurité par leur détection et prévention de toutes les attaques malveillantes. Dans le chapitre 4, Nous donnerons plus de détails sur le système de détection d'intrusion utilisées dans le cadre de notre thèse.

2.7. Conclusion

Dans ce deuxième chapitre, nous avons traité le problème de sécurité dans les RCSF, à savoir les différentes vulnérabilités qui peuvent être rencontrées les mécanismes de sécurité dans ce réseau, les attaques qui menacent les RCSF, ainsi que les différents mécanismes de sécurité existants qui tentent de protéger le matériel des nœuds capteurs, le canal de communication et les protocoles et services.

Comme nous l'avons vue dans la partie 2.6.2, Les RCSF sont menacés aisément à cause de l'utilisation de l'air comme médium de transmission et ont nécessité d'être protégés. La protection des liens de communication pourra être assurée à l'aide des primitives cryptographiques ; c.à.d. l'utilisation du chiffrement, déchiffrement, de la signature, etc., pour cela les nœuds capteur doivent partager certaines informations de sécurité (clés). L'établissement d'une clé secrète entre deux nœuds ou plusieurs est l'un des services de sécurité le plus important qui assure la confidentialité et l'intégrité des échanges dans un RCSF. Afin d'atteindre ce but d'une façon sécurisée, nous avons besoin d'un protocole de gestion de clés. La gestion de clés dans un RCSF représente un point très important pour de nombreux services de sécurité, que nous introduirons dans le chapitre suivant.

3

La gestion de clés dans les réseaux de capteurs sans fil

Sommaire

3.1. Introduction

Les réseaux de capteurs sont menacés facilement à cause de l'utilisation d'un canal sans fil comme médium de transmission. L'établissement d'une communication sécurisée entre les nœuds capteurs revêt un caractère primordial. Par conséquent, il est nécessaire d'utiliser des primitives cryptographiques comme les algorithmes de chiffrement pour protéger les communications, il est également essentiel d'établir des clés secrets avec les voisins des nœuds, ce qui est nécessaire aux primitives de sécurité qui permettent d'assurer des services comme la sécurité de l'agrégation, la sécurité du routage, la coopération (authentification), etc.

L'établissement d'une clé secrète entre les paires ou les groupes de nœuds est l'un des services de sécurité le plus important qui assure avec les primitives cryptographiques la confidentialité, l'authentification, la disponibilité, l'intégrité des échanges des données dans un RCSF. Afin de garantir l'efficacité de ces fonctionnalités, nous avons besoin d'un mécanisme de gestion de clés.

Dans ce chapitre, nous commençons par un aperçu sur les composants d'un système de gestion de clés dans les RCSF. Nous présentons ensuite les principaux schémas de gestion de clés, suivis d'une discussion sur les critères importants pour l'évaluation des performances d'un système de gestion de clés dans les RCSF. Enfin, nous concluons par une comparaison entre les différents schémas présentés.

3.2. Composants de la gestion de clés

La gestion de clés est un mécanisme essentiel pour assurer la sécurité des applications et des services réseau dans les RCSF. L'objectif de la gestion de clés consiste à définir les clés utilisées entre les nœuds de manière sécurisée et fiable après le déploiement. En outre, ce système doit prendre en charge le renouvellement et la révocation des clés pendant tout le cycle de vie du réseau. De ce fait, un système de gestion de clés inclut les trois composants suivants (voir figure 3.1) [70] :

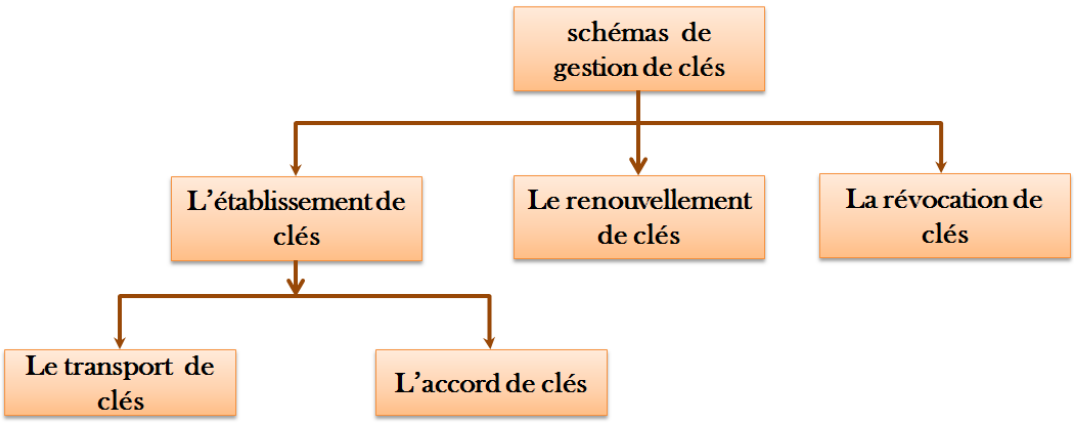

Figure 3.1: Schéma global montrant les composants d'un protocole dédié à la gestion de clés au sein d'un réseau de capteurs.

3.2.1. L'établissement de clés

L'établissement de clé est un processus ou protocole par lequel une clé secrète partagée devient disponible pour deux ou plusieurs entités, pour une utilisation cryptographique ultérieure. Ainsi, l'établissement de clé consiste à créer une clé de session entre les entités qui ont besoin de communiquer en toute sécurité les unes avec les autres.

L'établissement de clés dans les réseaux de capteurs peut également être réalisé avec des protocoles où les nœuds définissent une clé secrète partagée après le déploiement, soit par le biais du transport de clé ou d'un accord de clé [43].

(i) Un schéma de transport de clé (key transport schema) est un protocole dans lequel une entité crée ou obtient une clé secrète et la transfère de manière sécurisée à l'autre entité (ou aux autres entités) [71].

(ii) Un schéma d'accord de clé (key agreement schema) est une technique d'établissement de clé dans lequel toutes les entités participantes fournissent une entrée aléatoire qui est utilisée pour dériver une clé secrète partagée[70].
L'avantage d'un schéma de l'accord de clé par rapport au transport de clé est qu'aucune entité ne peut prédéterminer le résultat clé car cela dépend de la contribution de tous les participants. Les protocoles d'accord de clé (ou d'échange de clé) permettent à deux entités d'établir directement une clé en échangeant des messages sur un canal de communication non sécurisé.

Il existe trois types de schémas généraux d'accords de clés [72] [73]:

- **Le schéma du serveur de confiance (The trusted-server schema)**

Dans cette catégorie, l'établissement de clés s'effectue via des serveurs sécurisés centralisés, généralement de nature statique. Dans le RCSF, la station de base ou le sink peut agir en tant que centre de distribution de clés (KDC, Key distribution center).Habituellement, des clés symétriques uniques sont partagées entre la station de base et les nœuds ordinaires. Si deux nœuds devaient communiquer entre eux, ils s'authentifieraient d'abord avec la station de base, après la station de base génère une clé de liaison et l'envoie de manière sécurisée aux deux parties [74].
SPINS [75] et Kerberos [76] sont deux exemples de protocole d'accord de clé pour sécuriser les réseaux de capteurs sans fil. Dans Kerberos, le serveur de confiance partage des clés de longue durée avec chaque nœud du réseau et transmet les clés de session aux nœuds de capteurs sur demande. Cette méthode est extrêmement coûteuse pour le relais de message, elle n'est donc pas adaptée aux réseaux de capteurs. Dans SPINS (Security Protocols for Sensor Networks), seulement une clé unique est pré-chargée dans chaque nœud du réseau. Par conséquent, une capture de nœud n'entraînera pas la compromission totale du réseau. L'inconvénient principal de ce système est que la station de base représente un seul point de compromis pour les informations de sécurité et peut également induire une charge de communication centrée sur la station de base, ce qui peut entraîner un épuisement prématuré

de la batterie pour les nœuds les plus proches de la station de base. Une autre préoccupation est que certains réseaux ne disposent pas d'un dispositif approprié, hautement fonctionnel et inviolable qui peut être utilisé comme KDC sécurisé.

- **Le schéma d'auto-application (self-enforcing scheme) :** Le schéma d'auto-application dépend de la cryptographie asymétrique, telle que l'accord de clé utilisant des certificats de clé publique. Cependant, les ressources de calcul et d'énergie limitées des nœuds capteurs rendent indésirable l'utilisation des algorithmes à clé publique [77], Un exemple bien connu de protocole fournissant un accord de clé est le protocole Diffie-Hellman [78].
- **Le schéma de pré-distribution de clés :** Le troisième type de schéma d'accord de clé est la pré-distribution de clés, où les clés sont chargées dans les nœuds capteurs avant le déploiement, ce qui facilite l'établissement de clés entre les nœuds. Des recherches récentes sur les réseaux de capteurs suggèrent que les schémas de pré-distribution de clés sont les seules méthodes pratiques pour la distribution des clés aux nœuds du RCSF dont la topologie est inconnue avant le déploiement.

3.2.2. Le renouvellement de clés ("re-keying")

Pour éviter ou rendre une telle situation plus difficile pour l'adversaire, le système de gestion de clés doit permettre le renouvellement de clés. Il constitue un défi majeur pour le système de gestion de clés puisque des nouvelles clés doivent être créées d'une manière efficace et conforme à une consommation et conservation d'énergie. Plusieurs raisons justifient le renouvellement de clés du réseau RCSF:

-Renouvellement périodique : Ce dernier se fait volontairement après une période de temps fixe. Une clé ne doit pas être utilisée pendant une longue durée car elle offre des possibilités de compromis. Selon Abdalla et M. Bellare [79], si la clé utilisée est de longueur de k bits, donc doit être changé après $2^{2k/3}$ nombre de cryptage.

-Renouvellement à cause d'une compromission de nœud : Cette phase est déclenchée par la station de base lorsqu'elle détecte une anomalie dans le réseau, elle se fait soit : préventivement : lors d'une tentative d'accès illégale par un attaquant ou bien obligatoirement : après la compromission d'un ou de plusieurs nœuds capteurs du réseau.

-Renouvellement en cas du changement de la fonction du nœud : Cette phase concerne seulement les clusters d'une topologie hiérarchique. Pour prolonger la durée de vie de l'ensemble du réseau, il est nécessaire de changer le chef de groupe. Ainsi, l'un de ses nœuds

membres devient un chef de groupe. Par conséquent, certaines clés doivent être renouvelées pour assurer la sécurité.

3.2.3. La révocation de clés

La révocation de clé est un élément important de la gestion de clé car elle permet d'évincer les nœuds compromis du réseau. Ce processus consiste à supprimer des clés avant leur expiration prévue à l'origine, pour des raisons telles que la capture de nœud. Parfois, il ne peut pas être possible d'empêcher complètement les clés d'être compromises. Dans ces circonstances, le système de gestion de clés devrait fournir des mécanismes permettant de révoquer les clés compromises des nœuds identifiés de manière dynamique. La révocation garantit qu'un nœud expulsé n'est plus en mesure de déchiffrer les messages sensibles transmis sur le réseau. Donc, ces mécanismes sont utiles pour empêcher un nœud compromis de modifier le comportement du réseau en injectant de fausses données ou en modifiant des données des nœuds sécurisés.

3.3. Les phases d'établissement de clés

Dans les RCSF, les protocoles de gestion de clés sont basés sur des fonctions cryptographiques symétriques ou asymétriques. Traditionnellement, l'établissement de clés se réalise en se basant sur un système de cryptographie asymétrique, aussi appelée à clé publique, étant donné que ce dernier offre des mécanismes plus sûrs et fiables pour l'authentification et la distribution de clés. Par contre, la cryptographie asymétrique exige un espace de stockage assez grand et de haute capacité de calcul, ce qui rend son utilisation inappropriée pour les RCSF.

En effet, l'emploie d'un système cryptographique à base de clés symétriques pour l'établissement de clés diminue considérablement la consommation d'énergie des nœuds capteurs et l'espace de stockage réservé pour ces clés.

Ainsi, la difficulté de ces systèmes est de mise en œuvre de ces schémas en tenant compte des limitations : d'énergie, de la bande passante du réseau, et de mémoire de stockage des nœuds capteurs.

Bien que la cryptographie asymétrique comporte certains avantages comparé à la cryptographie symétrique et malgré les recherches qui visent à quantifier le coût d'énergie des algorithmes de cryptographie a clé publique, la cryptographie symétrique a ses propres qualités qui la rend généralement la plus préférée pour les RCSF. Pour cette raison la majorité des approches de gestion de clés proposés pour les RCSF sont basés sur la cryptographie symétrique.

Le problème majeur avec la cryptographie symétrique est de pouvoir trouver une méthode qui facilite l'établissement de clés entre les nœuds. La solution commune est d'utiliser une méthode de pré-distribution, dans laquelle les clés sont chargées dans les nœuds capteurs avant le déploiement.
Les RCSF utilisent un mécanisme à clé symétrique pour l'établissement de clé basée sur la pré-distribution de clés, qui comprend les trois étapes suivantes [80]:

3.3.1. Pré-distribution de clés (Key pre-distribution)

Dans un RCSF, La majorité des mécanismes basés sur les systèmes symétriques, asymétriques ou hybrides résolvent le problème d'établissement de clés en passant par une phase de pré-distribution. La pré-distribution des clés cryptographiques est le fait de charger ces clés dans la mémoire des nœuds capteurs avant le déploiement. Ainsi, S'il existe une clé commune entre deux nœuds capteurs, ils peuvent alors créer un lien sécurisé entre eux. Dans la pré-distribution de clés, un gros problème est de savoir comment charger un ensemble de clés (appelé porte-clés) dans la mémoire limitée de chaque capteur.

3.3.2. Découverte de clé partagée

Après le déploiement, chaque nœud tente de découvrir ses voisins dans sa portée de communication avec laquelle il partage des clés. En effet, un protocole de communication après le déploiement est chargé de découvrir la clé commune entre deux nœuds voisins. Ainsi, si un nœud partage une clé commune avec un nœud particulier, il peut utiliser cette clé pour une communication sécurisée.
Le bon schéma de découverte des voisins ne donnera pas à un attaquant l'occasion de découvrir les clés partagées et l'attaquant ne peut donc faire que l'analyse du trafic.

3.3.3. Établissement de clés de chemin

Si une clé commune n'existe pas entre deux nœuds voulants communiquer, et qui sont relies par un chemin multi-saut, alors un chemin sécurisé doit être trouvé entre eux. Ce chemin est composé des liens sécurisés entre des nœuds. Autrement dit, des nœuds partageant des clés communes. Une fois ce chemin établi et la clé de chemin (path key) générée, les deux nœuds peuvent l'utiliser pour communiquer en toute sécurité.

3.4. Classification de méthodes et protocoles

Comme les nœuds ont des ressources limitées; la gestion de clés dans les RCSF est un problème. Pour cette raison, les protocoles de gestion de clés pour ces réseaux doivent être extrêmement légers. Dans la littérature, les protocoles de gestion de clés sont basés sur des fonctions cryptographiques symétriques ou asymétriques. Tandis que, la plupart des protocoles de gestion de clés existants pour les RCSF sont basés sur la cryptographie à clé symétrique car les techniques de cryptographie à clé publique nécessitent en général des calculs intensifs.

Il existe plusieurs façons de classer les schémas de gestion de clés dans le RCSF en considérant différents référentiels. Divers chercheurs ont présenté différentes taxonomies [81-84]. Les schémas de gestion de clés dans le RCSF peuvent être classés de manière générale en solutions utilisant des clés pré-distribuées ou générées dynamiquement par paire, par groupe ou par réseau durant la communication. Les solutions basées sur des techniques de la pré-distribution de clés peuvent utiliser l'une des deux approches: probabiliste ou déterministe.

Ces schémas sont également classés en solutions dynamiques et statiques, dans une gestion statique toutes les clés sont pré-distribuées aux nœuds avant le déploiement et le renouvellement de clés n'est pas appliqué. Par contre, dans une gestion dynamique, un sous ensemble de clés est redistribué aux nœuds après le déploiement, et le renouvellement de clés périodique ou à la demande est effectué, exigeant ainsi un nombre restreint de messages et faire face aux nœuds compromis. Un autre critère de classement concerne le rôle des nœuds de réseau dans le processus de gestion de clés, avec lequel les schémas proposés sont classés en réseaux homogènes ou hétérogènes. Les schémas basés sur un réseau homogène supposent généralement un modèle de réseau plat, tandis que les schémas basés sur un réseau hétérogène sont destinés à la fois aux réseaux plats et aux réseaux en grappe (cluster). Certaines d'autres, sont classées selon les différents styles de communication tels que : (i) par pair (unicast) entre une paire de nœuds, (ii) par groupe (multicast) dans un groupe de nœuds de capteurs, et (iii) par réseau (broadcast) à partir de la station de base aux nœuds capteurs.

Nous considérons la taxonomie la plus courante de la gestion de clés comme montre la figure 3.2. Dans cette taxonomie, les solutions basées sur la cryptographie symétrique sont classées dans deux catégories selon la topologie du réseau (plate ou hiérarchique) et selon les techniques de la pré-distribution de clés : déterministe ou probabiliste. Dans cette section, nous détaillerons les principales solutions de cette figure.

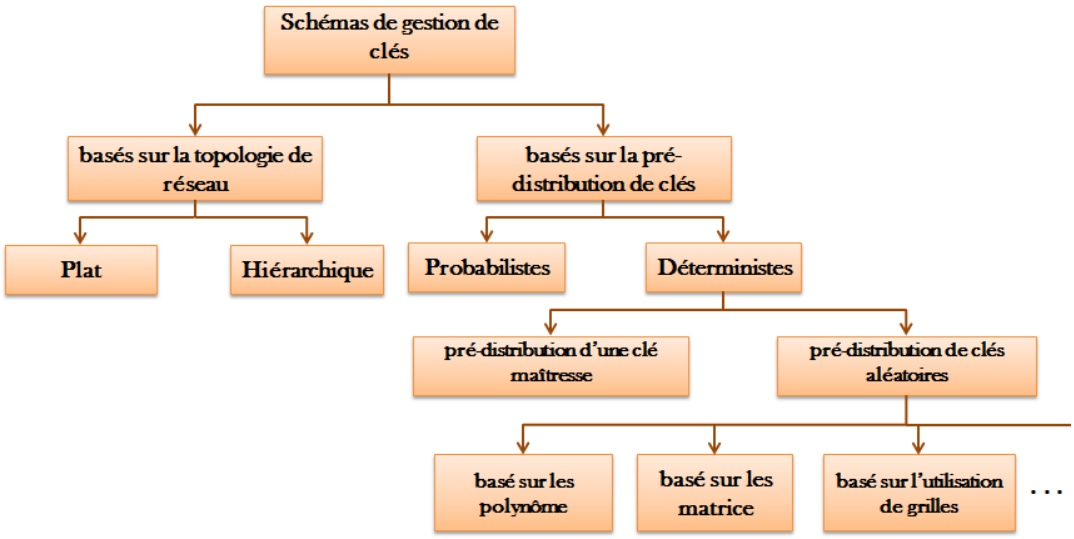

Figure 3.2 : Classification des schémas de gestion de clés dans le réseau de capteur sans fil.

3.4.1. Schémas basés sur la pré-distribution de clés

La pré-distribution de clé dans les RCSF peut être effectuée de l'une des deux méthodes :
(i) les méthodes probabilistes, dans ce cas un sous ensemble de clés prélevées à partir d'un grand ensemble de clés et placés dans les nœuds capteurs. L'idée de cette méthode est que deux nœuds communiquent entre eux ont une certaine probabilité d'avoir partagé une clé commune qui appartient aux deux sous-ensembles de ces communicants. (ii) les méthodes déterministes, assurent que chaque nœud est capable d'établir une clé par paire commune (elle est connue par « pairwise key ») avec n'importe quel autre nœud du réseau.

3.4.1.1. Schémas probabilistes

Eschenauer et Gligor [85] ont été parmi les premiers à introduire le concept de pré-distribuer ou de stocker dans les nœuds capteurs. Cette méthode est considérée comme le schéma basique des méthodes probabilistes. Il adresse l'établissement de clés, leur renouvellement et leur révocation. Dans ce schéma, trois phases sont nécessaires pour installer les clés secrètes entre les nœuds capteurs.
(i) La phase de pré-distribution de clés : Cette phase est effectuée avant le déploiement et elle est réalisée en plusieurs étapes. Au début, un large ensemble de clés P (Pool) est génère et chaque clé est associée à un identificateur. Ensuite, m clés sont choisies aléatoirement dans l'ensemble P servant de porte-clés (Key ring) à chaque nœud. Ces porte-clés sont stockés

ensuite dans la mémoire des nœuds. L'association entre la liste d'identifiant des porte-clés (Key ring) et l'identifiant des nœuds capteurs est enregistrée dans un nœud de contrôle.

Afin d'assurer l'établissement d'une clé commune entre deux nœuds après le déploiement, Le nombre de clés $|P|$ est choisi de telle façon que deux sous-ensembles aléatoires de P de taille m auront une certaine probabilité d'avoir au moins une clé en commun. Par exemple pour une probabilité d'appairage de 0,5 seulement 75 clés sont choisies aléatoirement à partir de l'ensemble P avec $|P| = 10000$

(ii) la phase de découverte de clés communes : Après le déploiement, chaque nœud essaie de découvrir ses voisins avec lesquels il partage des clés communes. Le moyen le plus simple est que les nœuds diffusent leurs listes d'identifiants des clés stockées dans sa mémoire à d'autres nœuds. Si un nœud découvre qu'il partage une clé commune avec un nœud particulier, il peut utiliser cette clé pour une communication sécurisée. Cette approche ne donne à l'adversaire aucune opportunité d'attaque s'il découvre la liste d'identifiant des clés et laisse seulement la possibilité de lancer une attaque d'analyse de trafic.

Un exemple de la méthode d'Eschenauer et Gligor [85] est illustré à la figure 3.3, dans laquelle un groupe de 20 clés et un certain nombre de porte-clés, chacun contenant trois clés, sont présentés. Comme on peut le constater à la figure 3.3, le réseau en termes de partage de clé n'est pas parfaitement connecté. Cependant, ils atteignent une connectivité suffisante. Les processus permettant aux nœuds sans partage de clé direct de trouver leur clé partagée sont décrits ci-dessous.

(iii) La phase d'établissement de clé de chemin :

Un lien existe entre deux nœuds seulement s'ils partagent une clé, mais l'étape d'établissement de la clé de chemin facilite la mise en place du lien entre deux nœuds s'ils ne partagent pas une clé commune. Les nœuds capteurs peuvent alors utiliser les liaisons existantes pour mettre en place des clés partagées avec leurs nœuds voisins qui ne partageaient pas de clé en commun avec eux, dans laquelle il sélectionne un nœud intermédiaire agissant comme un centre de distribution de clé ou un médiateur entre les deux nœuds de capteur afin d'établir une clé de session commune. Cette clé est ensuite utilisée comme clé de chemin pour les paires de nœud capteur sélectionnées.

(iv) La révocation de clé: Lorsqu'un nœud de capteur compromis, son porte-clés doit être supprimées. Pour cela, un nœud contrôleur (qui pourra être mobile) diffuse un message simple de révocation signé contenant la liste des identifiants du porte-clés pour que ces clés soient éliminées des porte-clés des autres nœuds révoqué. La liste est signée par une clé de signature générée par le nœud contrôleur et envoyée individuellement (unicast) à chaque nœud en la

chiffrant avec la clé partagée entre tous les nœuds et le nœud de contrôle pendant la phase de pré-distribution de clés. Une fois que chaque nœud révoqué supprimé une clé du nœud compromis, des liens créés peuvent disparaitre et affectent l'envoi des données ce que nécessite une reconfiguration de ces liens (en commençant de nouveau la phase d'établissement de clés avec leurs voisins ou l'établissement de clé de chemin).

Notons que dans ce schéma, la procédure de renouvellement de clés d'un nœud est équivalente à une révocation de la clé partagée avec un autre nœud.

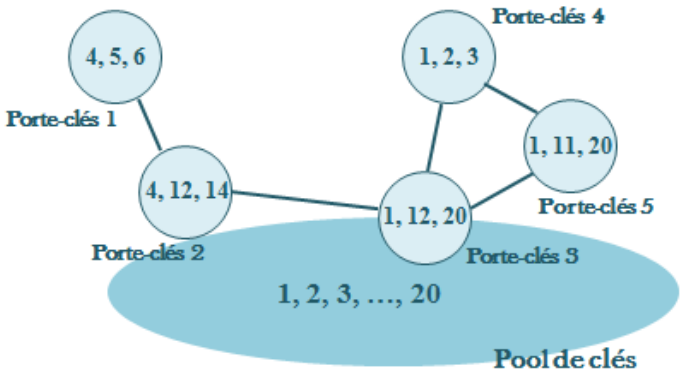

Figure3.3 : Un exemple du schéma d'Eschenauer et Gligor

Chan, Perrig et Song [86] ont introduit un ensemble d'amélioration au schéma de base (Eschenauer et Gligor) et de deux aspects différents. Deux variantes sont proposées : la pré-distribution de clés aléatoires Q-Composite et le renforcement de clés par trajets multiples. Chacun de ces propositions vient avec un type différent de compromis. Par exemple, le nouveau schéma de pré-distribution Q-Composite augmente la résistance contre la capture de nœuds car l'attaquant aura plus de difficulté à découvrir la clé partagée avec un nœud voisin. Nous détaillerons dans les prochains sous sections les deux mécanismes proposés dans ce schéma.

Schéma Q-composite

Dans ce schéma, deux nœuds voisins doit partager q clés avec $q > 1$ pour établir un lien sécurisé. La clé utilisée pour sécuriser la communication entre ces deux nœuds est le hash de toutes les clés partagées. Un exemple de schéma de pré-distribution de clé Q-composite est illustré à la figure 3.4. On peut observer que la figure 3.4 est assez similaire à la figure 3.3. Cependant, une fois que deux nœuds ne peuvent pas trouvez q clés (dans cet exemple q=2) communes dans leur porte-clés respectifs, elles ne sont pas connectées.

En effet, plus la quantité de chevauchement des clés entre deux nœuds augmente plus la résilience contre la capture du nœud augmente. Ainsi, il devient plus difficile pour un adversaire avec un ensemble donné de clés de casser un lien de communication.

La taille du pool de clés ($|S|$) est le paramètre critique à calculer pour que le schéma Q-Composite soit efficace. Ainsi, $|S|$ est calculée en fonction de la contrainte de la probabilité que deux nœuds partagent au moins q clés et le nombre de clés qu'un nœud peut contenir m (voir tableau 3.1).

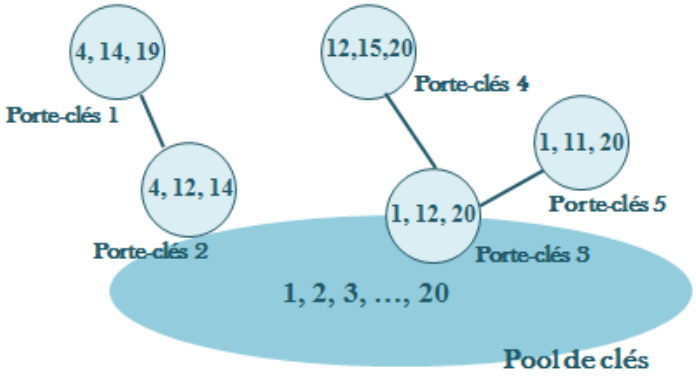

Figure 3.4 : Un exemple du schéma de q-composite

Renforcement de clé par chemins multiples

Ce mécanisme peut être appliqué en conjonction avec le schéma aléatoire de base (Schéma de Eschenauer et Gligor) ou deux nœuds voisins partagent une seule clé pour établir un lien de communication sécurisé. Supposons que A a un lien sécurisé avec B après la phase de découverte de clé partagée et l'établissement de clé de chemin utilisant une clé partagée k. Étant donné que k peut résider dans le porte-clés (Key ring) de certains autres nœuds du réseau. Dans ce cas, si un nœud est compromis, il est possible que la liaison entre les nœuds A et B non compromis soient également compromises.

Afin de résoudre ce problème, les auteurs ont proposé que les clés utilisées pour la sécurité des liens de communication entre les nœuds non compromis, soient actualisées. Si les deux nœuds A et B ont h chemins disjoints (c'est-à-dire les chemins qui mènent de nœud A vers le nœud B avec un certain nombre de sauts), le nœud A génère h valeurs aléatoires et envoie chacun d'eux à B via un chemin séparé disjoint. Ensuite, les deux nœuds A et B calculent une clé k' à l'aide de la clé k et de toutes les h valeurs aléatoires. Et donc, plus le nombre de chemins entre les deux nœuds augmente, plus le niveau de sécurité augmente.

Cependant, afin de minimiser le risque qu'un adversaire puisse écouter et déchiffrer les messages circulant sur un chemin, la longueur des chemins disjoints doit être réduite.

Du et autres. [87] a combiné un schéma de pré-distribution de clé aléatoire avec la connaissance du déploiement de nœud. Dans ce schéma, la connaissance de déploiement est modélisée à l'aide de fonctions de densité de probabilité non uniformes (non uniform probability density functions (pdfs)), ce qui signifie qu'ils supposent que les positions des nœuds de capteurs se situent à certaines zones. Ils ont décrit un modèle de déploiement basé sur les groupes dans lequel la zone de déploiement est divisée en grilles bidimensionnelles et les nœuds sont classés en groupes. Ce schéma comporte trois phases. La première phase est la pré-distribution de clés, qui consiste à diviser l'ensemble de clés (Pool) à des sous-ensembles de clés. Chaque groupe de nœuds choisit ses clés dans un sous-ensemble de clés correspondant et ils sont censés être déployés dans une grille fixe. Sous cette forme, il est plus facile de partager plus de clés. Les deux autres phases (découverte de clé partagée et l'établissement de clé de chemin) sont exactement les mêmes que le schéma de base [85].

La majorité des approches probabilistes de gestion de clés sont simples dans leur implémentation. Par contre, elles souffrent de quelques désavantages. Xu et autres [88] ont montré que la gestion de clés probabiliste n'avait que des avantages limités par rapport aux approches déterministes. En ce qui concerne les performances, la taille de l'ensemble de clés pré-distribuées va augmenter avec la taille du réseau et donc une utilisation importante de l'espace mémoire. De plus, l'approche ne peut pas protéger le réseau contre des nombreuses menaces tel que la capture physique de nœud. Une fois que l'ensemble de clés d'un nœud est extrait, l'attaquant pourra découvrir les clés communes établies avec ces nœuds voisins.

3.4.1.2. Schémas déterministes

Il existe plusieurs schémas pour l'établissement de clés symétriques d'une manière déterministe. Ils peuvent être classés aux : méthodes de pré-distribution de clés aléatoires et méthodes de pré-distribution d'une clé maîtresse. La première classe elle-même peuvent être classés dans les cinq types suivants selon la construction mathématique sur laquelle elles sont basées: méthodes basées sur l'utilisation de grilles [89][90], méthodes polynomiales [91], méthodes basées sur des structures combinatoires [92][93], méthodes matricielles [94], et méthodes basées sur le système de base d'exclusion (EBS) [95]. Nous avons limité la synthèse bibliographique de cette partie à quelques méthodes pour quelques catégories.

3.4.1.2.1. Schémas de pré-distribution de clés aléatoires

Dans ce type des méthodes déterministes, la liste de clés ou les porte-clés sont générés d'une façon déterministe pour s'assurer de l'établissement de certains liens entre les nœuds capteurs

Blom [94] a proposé un schéma déterministe pour établir une clé symétrique distincte entre chaque paire de nœuds du réseau par des calculs matriciels. Premièrement, la station de base construit une matrice symétrique D de la taille $(\lambda + 1)$ $(\lambda + 1)$ et une matrice publique G de la taille $(\lambda + 1)$ N sur un corps fini $GF(q)$, où N est la taille du réseau. Toutes les clés par-paires de ces N nœuds sont stockées dans une matrice symétrique $K = AG$, sachant que $A = (DG)^T$, et il est appelée matrice secrète. Chaque élément k_{ij} du K est la clé du nœud i pour sécuriser le lien avec le nœud j. Ensuite, chaque nœud i est préchargé avec la i-ème rangée de la matrice secrète et la i-ème colonne de la matrice publique. Ce processus est illustré à la figure 3.5.

Après le déploiement, deux nœuds quelconques i et j peuvent calculer individuellement leur clé par-paire $k_{ij} = k_{ji}$ en échangeant uniquement leurs colonnes, car la clé est un produit de leur propre ligne et la colonne de l'autre. Ce schéma peut attribuer à chaque paire de nœuds une clé, et il tolère la compromission de λ nœuds. En d'autres termes, si pas plus de λ nœuds ne sont compromis, toute la matrice secrète est parfaitement sécurisée. Donc, λ est le seuil prévu pour compromettre la sécurité de réseau. Ainsi, ce schéma conforme à la propriété de λ ("λ-secure property").

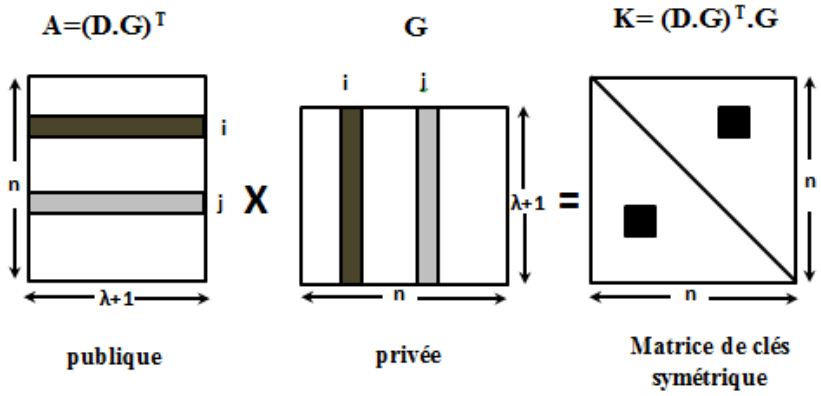

Figure 3.5 : Les matrices dans le schéma de Blom [94]

Blundo et autres [91] ont proposé une approche conforme également à la propriété de λ du fait qu'elle utilise un polynôme symétrique bivariant de degré λ, $P(x, y) = \sum_{i=0}^{\lambda} \sum_{j=0}^{\lambda} a_{ij} x^i y^j$, qui est généré sur un corps fini $GF(q)$, où nous avons $P(x, y) = P(y, x)$ en choisissant $a_{ij} = a_{ji}$.

Chaque nœud i est pré-distribué avec un polynôme $\mathrm{P}(\mathrm{i},\mathrm{y})$. Après le déploiement, Pour établir une clé par paire, chaque nœud évalue le polynôme à l'ID de l'autre nœud de capteur. Par exemple, dans le cas de deux nœuds i et j, La clé commune entre eux est $K_{ij} = \mathrm{P}(\mathrm{i},\mathrm{j}) = \mathrm{P}(\mathrm{j},\mathrm{i})$. La sécurité dans ce schéma polynomial est parfaite quand pas plus de λ nœuds sont compromis. D'une autre côté, le coût de communication est réduit pendant le processus d'établissement de clés. Tandis que, le coût de stockage du polynôme est relatif au λ.

Chan et Perrig [89] ont présenté un schéma déterministe appelé PIKE (Peer Intermediaries for Key Establishment), dans lequel les clés secrète pour les communications dans le réseau sont pré-chargées pour que n'importe quel couple de nœuds A et B puisse toujours avoir la possibilité de trouver un nœud C du réseau qui ayant une clé commune avec A et B. Alors A pourra utiliser C comme nœud intermédiaire de confiance afin de transmettre son message d'établissement de clé à B.

Les identifiants (ID) des N nœuds sont organisés dans une structure en grille à deux dimensions$(\sqrt{N} \times \sqrt{N})$. Chaque ID du nœud a une coordonnée unique (x, y), où $x, y \in \{0, 1, 2, \dots, \sqrt{N} - 1\}$. Chaque nœud partage des clés secrète uniques avec $2(\sqrt{N} - 1)$ nœuds qui ont la même coordonnée x ou y. Si deux nœuds ne possèdent pas de coordonnées x ou y communes, ils doivent choisir un nœud intermédiaire possédant une coordonnée x ou y commune avec les deux pour les aider à établir une clé par paire en toute sécurité.

La figure 3.6 illustre l'idée de base du schéma PIKE. Des lignes sombres connectent les nœuds qui partagent une clé unique avec le nœud A, et Les lignes lumineuses connectent les nœuds qui partagent une clé unique avec le nœud B. Il existe 2 nœuds qui partagent une clé unique avec les nœuds A et B. Ainsi, ils peuvent jouer le rôle d'intermédiaires de confiance dans l'établissement de clés entre A et B. Une fois le nœud intermédiaire C élu, A lancé le processus d'établissement de clés. Cependant, Chaque message échangé est accompagné d'un message d'authentification MAC (code) afin de garantir l'intégrité.

Par rapport à d'autres méthodes, PIKE minimise le coût de stockage de clés avant le déploiement. Cependant, les échanges de messages consomment du temps et de l'énergie pendant le processus d'établissement de clés.

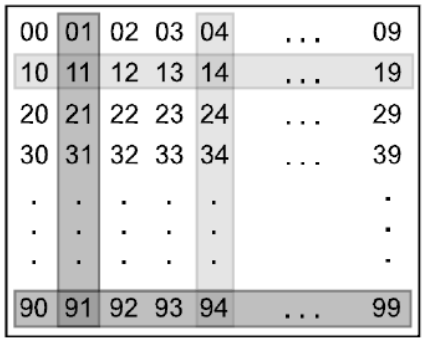

Figure 3.6 : Un exemple d'Espace virtuel d'identifiants de nœuds d'un réseau de 100 nœuds pour le schéma PIKE [89].

3.4.1.2.2. Schémas basés sur la pré-distribution d'une clé maîtresse

Pour garantir le déterminisme, Une clé commune est pré-chargée sur tous les nœuds capteurs avant leurs déploiements. Cette dernière est utilisée afin de générer des clés par-paires entre chacun des deux nœuds, et qui sera effacée après l'établissement de clés.

Lai et autres [95] ont proposé le protocole BROSK (Broadcast Session Key). Dans ce schéma, une seul clé est pré-distribuée dans chaque nœud de capteur avant le déploiement. Une paire de nœuds (S_i, S_j) peut être établie une clé de session à l'aide de cette clé principale K_m (master key) et d'un nombre aléatoire échangé entre les deux nœuds N_i et N_j. Ce schéma proposé a l'avantage que chaque capteur n'a besoin que de très peu de mémoire. Cependant, l'inconvénient est évident. Lorsque la clé principale est compromise, toutes les clés de session sont exposées. Par conséquent, ce schéma n'a aucune résilience.

Ce schéma et tous les schémas déterministes décrits ci-dessus reposent sur une hypothèse commune, à savoir que les adversaires peuvent compromettre les nœuds capteurs tant qu'ils sont déployés. Cependant, cette hypothèse peut-être trop forte. Bien que les nœuds capteurs ne soient pas inviolables, les adversaires ont besoin d'au moins une courte période de temps pour trouver, casser et contrôler un nœud capteur. Sur la base de ce modèle de menace faible, les schémas de gestion de clés peuvent être conçus de manière plus efficace et efficiente.

Un parmi les solutions proposées est le protocole LEAP (Localized Encryption and Authentication Protocol) proposé par Zhu et autres [97]. Dans ce schéma amélioré, la clé principale est effacée après que les clés de session soient établies. Dans ce cas, la résilience est améliorée.

Zhang et autres [98] a proposé un schéma de gestion des clés déterministes distribuées à efficacité énergétique réduite s'appelle EDDK (energy-efficient distributed deterministic key management scheme), qui visait à résoudre les attaques d'épuisement des ressources et les attaques par déni de service. Dans le schéma EDDK, chaque nœud est pré-chargé avec une fonction pseudo-aléatoire f et une clé initiale K_I qui peut utiliser pour calculer sa clé individuelle. De plus, chaque nœud de capteur stocke une table qui maintient les informations des nœuds voisins tels que ID de voisin, clé par paire, numéro de séquence. Il partage également une clé de cluster locale avec ses nœuds voisins et stocke également cette clé dans la table. Ce schéma comprend trois phases: l'établissement de clés, transfert des données et maintenance de clés.

Dans la phase d'établissement de la clé par paire, un nœud A calcule d'abord sa clé individuelle K_A par $K_A = f_{K_I}(ID_A)$, puis génère une séquence numérique aléatoire SN_A et diffuse le message $JOIN$ aux nœuds voisins. Le message $JOIN$ contient $ID_A \parallel E_{K_A}(SN_A \parallel K_G) \parallel MAC_{K_A}(ID_A \parallel E_{K_A}(SN_A \parallel K_G))$. Lorsque les deux nœuds A et B reçoivent le message $JOIN$ l'un de l'autre, ils vérifient l'exactitude du message $JOIN$. Après vérification, les clés par paires sont générées: $K_{AB} = f_{K_I}(K_A \oplus K_B, SN_A \oplus SN_B)$. Comme LEAP, une fois que le temporisateur d'établissement de clé a atteint sa valeur de seuil prédéfinie, un nœud supprime toutes les clés individuelles de ses voisins, les nombres aléatoires, la fonction pseudo-aléatoire et la clé initiale pour améliorer la sécurité et économiser l'espace de stockage dans le nœud.

Le principal avantage du schéma EDDK réside dans le fait que les clés par paire sont décentralisées et que la compromission d'un nœud de capteur n'affecte pas les autres liaisons de communication. Il est également résistant aux attaques par rejeu, à identité multiple (Sybil) et à la réplication de nœud. Le principal inconvénient d'EDDK est qu'il n'est pas applicable dans les réseaux denses, car chaque nœud capteur doit stocker une table, qui inclut les informations de tous ses voisins.

3.4.2. Schémas basés sur la topologie de réseau

Selon la topologie de réseau, les protocoles de gestion de clés dans un RCSF peuvent être hiérarchiques ou plats. Un RCSF hiérarchique comprend une ou plusieurs stations de base robustes en calcul. Les nœuds capteurs sont déployés dans un voisinage à un ou deux sauts autour des stations de base ou des nœuds capteurs riches en ressources (appelés chef de groupe), comme illustré à la figure 1.5 (voir chapitre1). Les stations de base sont généralement considérées comme fiables et utilisées comme centres de distribution de clés. Dans un réseau

hiérarchique, des clés par paire, par groupe et par réseau sont nécessaires pour sécuriser le flux de données, qui peuvent divisée en trois types :(i) par paire (unicast) entre des paires des nœuds capteurs et des nœuds capteurs à la station de base, (ii) par groupe (multidiffusion) au sein d'un groupe des nœuds capteurs, (iii) par réseau (diffusion) des stations de base aux nœuds capteurs. Dans un RCSF plat, il n'y a pas de membre riche en ressources et les nœuds de capteur ont des capacités équivalentes. Pour le flux de données, il est similaire au flux de données dans un réseau RCSF hiérarchique, à la différence que des messages réseau (diffusion) peuvent être envoyés par tous les nœuds capteurs. Les nœuds capteurs utilisent directement des clés pré-distribuées ou utilisent des clés matériaux pour générer de manière dynamique des clés par paires et par groupes.

La plupart des schémas de gestion de clés proposés sont des schémas plats. Ces systèmes entrent également dans des catégories probabilistes et déterministes qui sont détaillées dans la sous-section 3.4.1.1 et la sous-section 3.4.1.2 respectivement.

Dans l'intention d'économie d'énergie, l'utilisation d'une topologie hiérarchique peut simplifier et améliorer l'évolutivité et l'efficacité de la procédure de gestion de clés. Plusieurs systèmes de gestion de clés déterministes, ont été proposés pour établir des communications sécurisées entre les nœuds capteurs. Cependant, peu de travaux envisagent un schéma de gestion de clés qui tire parti de la topologie hiérarchique des RCSF. Dans la section suivante, nous présentons des protocoles de gestion de clés hiérarchiques.

3.4.2.1. Schémas hiérarchiques

Zhu et autres [97] proposent le protocole déterministe LEAP (Localized Encryption and Authentication Protocol). Ce protocole supporte l'authentification et la gestion de clés destiné aux réseaux de capteurs hiérarchiques. LEAP prend en charge l'établissement de quatre types de clés pour chaque nœud de capteur (i) clé individuelle partagée avec la station de base, (ii) clé par-paire partagée avec un autre nœud, (iii) clé de cluster partagée avec ses nœuds voisins du même cluster et, (iv) clé de groupe partagée avec tous les nœuds du réseau.

La station de base génère une clé initiale K_I et la stocke dans la mémoire de chaque nœud avant le déploiement. Lorsque la phase de déploiement est terminée, chaque nœud A tente de trouver ses voisins après qu'il dérive sa propre clé individuelle $K_A = f_{K_I}(ID_A)$. Tout d'abord, le nœud A diffuse un message HELLO contenant son ID à ses voisins et attend que ses voisins répondent. Lorsque le nœud voisin du nœud A reçoit ID_A, le nœud B répond de son ID_B et $MAC_{K_B}(ID_A \parallel ID_B)$. Le nœud A utilise la clé initiale K_I pour calculer $K_B = f_{K_I}(ID_B)$ et

vérifier l'identité du nœud B. Ensuite, le nœud A calcule sa clé par paire partagée avec lui $K_{AB} = f_{K_B}(ID_A)$. Le nœud B pourra calculer cette clé de la même façon. Après que le temporisateur d'établissement de clé atteint sa valeur de seuil T_{min}(fixé au début), le nœud A efface la clé initiale et toutes les clés individuelles (exemple K_B) de ses voisins. Il garde seulement sa clé individuelle et les clés par paire partagées avec ses voisins.

L'établissement de la clé de cluster suit la phase d'établissement de la clé par paire. Supposons qu'un nœud A soit maintenant un chef de cluster veuille établir une clé de cluster avec tous ses nœuds voisins $b_1, b_2, b_3, \dots, b_m$. Il génère d'abord une clé aléatoire K_A^c, puis la chiffre avec la clé par paire partagée avec chaque nœud voisin b_i $(1 \leq i \leq m)$ avant de le transmet en unicast à b_i $(A \rightarrow b_i : (K_A^c)_{K_{Ab_i}})$.

Cependant, la clé de groupe est pré-chargée à chaque nœud avant le déploiement. Un problème important qui se pose immédiatement est la nécessité de mettre à jour la clé de groupe de manière sécurisée une fois qu'un nœud compromis est détecté.

Dans le schéma de base (LEAP), Etant donne l'hypothèse qu'un nœud de capteur ne peut pas être compromis dans T_{min}, le schéma de base est sécurisé parce que K_I n'est pas compromise. Les auteurs de LEAP ont proposés un schéma étendu s'appelle LEAP+ [99]. Ils considèrent des attaques plus puissantes en supposant que K_I peut être compromise. L'idée essentielle du schéma étendu est d'enlever la dépendance de la simple clé initiale K_I, au lieu de cela un ordre des clés initiales est établi pour dériver les clés principales des différents nœuds du réseau. Dans ce cas, la durée de vie totale du réseau est divisée en intervalles de temps $T_1, T_2, T_3, \dots, T_m$. Le contrôleur de réseau produit aléatoirement M clés (clés initiales) : $K_I^1, K_I^2, K_I^3, \dots, K_I^m$ et chaque clé initiale est considérée comme valide uniquement dans un intervalle de temps T_i.

Zhang et Wang [100] ont proposé un système de gestion de clés hiérarchique efficace et sécurisé, appelé SEHKM. Ce système prend en charge l'établissement de trois types de clés pour chiffrer les messages envoyés entre les nœuds de capteurs. Ce sont : (i) la clé de réseau est utilisé afin de chiffré les messages diffusés et authentifie les nouveaux nœuds, (ii) la clé de groupe est partagée par tous les nœuds du même cluster, et (iii) la clé par paire est partagée avec une paire de nœuds spécifique. Dans un RCSF hiérarchique, les Cluster Heads (CH) sont très importantes en raison de la structure du réseau. Afin d'améliorer la sécurité et de réduire le coût des ressources et le risque après la compromission de CH, un nœud assistant est introduit

dans ce schéma. Dans SEHKM, tous les nœuds doivent pré-distribuer de deux éléments de clé: la clé de réseau et un numéro initial IN.

Dans le processus de construction des groupes (clustering), après qu'un CH connaît tous les ID de ses nœuds membres, il calculer la clé par paire jetable (disposable pairwise key) partagée avec chaque nœud membre($DPK_{A,i} = f_{K_I}(ID_A)$) et le nœud A pourra calculer cette clé de la même façon. A la fin de cette phase, chaque nœud efface IN et ne reste que la clé par paire jetable (disposable pairwise key).

La station de base et tous les CH constituent un groupe de niveau supérieur, chaque CH et ses nœuds membres dans le même cluster formant un groupe. Le CH génère la clé de groupe et la diffuse dans leur groupe crypté par une clé par paire jetable (disposable pairwise key) $(CH_i \rightarrow A : (GK_i)_{DPK_{A,i}})$. Une fois la clé de groupe utilisée, la station de base génère un nombre aléatoire en tant que une nouvelle clé de réseau. Elle chiffre la clé par la clé de groupe et la diffuse à tous les CH, puis chaque CH diffuse la clé chiffrée par sa clé de groupe vers ses propres nœuds membres.

Il existe trois types de clés par paires dans ce schéma qui se distinguent par leur génération et leur stockage. Le premier type est constitué des clés par paires associées à un groupe, partagées par les CH avec leurs nœuds membres et les clés par paires partagées par SB avec des CH. Le deuxième type de clés par paires est associé à un nœud assistant. Cette clé par paire est partagée par le nœud assistant avec les nœuds membres du même cluster et la clé partagée par le nœud assistant avec SB. Le dernier type est une clé par paire jetable et elle sera effacée après l'établissement de toutes les clés. Dans le premier type là où la génération de clé par paire associée à un groupe, les auteurs utilisent l'algorithme de Diffie-Hellman [78]. Dans le deuxième type, la clé par paire associée au nœud assistant est générée par une fonction pseudo-aléatoire. Pour les nœuds u et v la clé par paire partagée entre eux est $PK_{u,v} = f_m(ID_u)$ tel que m est un nombre uniquement connu par le nœud générateur v.

Ensuite, le nœud générateur v envoie $PK_{u,v}$ au nœud u. Le nœud u stocke $PK_{u,v}$, tandis que le nœud v ne stocke que l'ID du nœud u. Il générera $PK_{u,v}$ s'il est nécessaire. Pour une clé partagée par le nœud assistant et un nœud membre, le nœud assistant joue le rôle de générateur. Cependant, SB est le générateur de la clé par paire partagée entre SB et un nœud assistant. Après la mise en place de toutes les clés par paires, les clés par paires jetables doivent être effacées.

Messai et autres [101,102] ont proposé une nouvelle approche appelée EAHKM (Energy Aware Hierarchical Key Management in WSNs) pour sécuriser la formation du cluster et

assurer la gestion de clés dans les RCSF hiérarchiques. Cette approche comporte deux phases: une phase de pré-distribution de clés et une phase de formation de cluster et d'établissement de clés.

EAHKM assure la mise en place d'une clé par paire entre chaque nœud capteur et son Cluster Head, établissant ainsi une clé de diffusion dans chaque cluster du réseau. Dans la première phase, chaque nœud de capteur S_i est pré-distribué avec trois clés : K_{BS,S_i}, $K_{S_i,BS}$ et K_N. K_{BS,S_i} , $K_{S_i,BS}$ sont deux clés partagées entre S_i et la BS pour le cryptage des messages. S_i utilise $K_{S_i,BS}$ pour chiffrer ses messages envoyés à la BS. De même, la BS utilise K_{BS,S_i} pour chiffrer les messages envoyés à S_i. K_N est la clé de réseau partagée par tous les nœuds capteurs. Dans la deuxième phase, la BS initie la construction des clusters après le déploiement, en diffusant un message HELLO $(BS \rightarrow \{Hello, BS, Level = 0, Energy = \infty, MAC_{K_N}(BS, 0, \infty)\}_{K_N})$. Lorsqu'un nœud de capteur reçoit le message HELLO de la BS, il la choisit comme CH. Si le nœud reçoit des messages HELLO de ses voisins, il choisit comme CH le nœud avec le niveau le plus bas ayant la valeur d'énergie la plus élevée. Après la sélection des CH, chaque nœud membre calculer la clé par paire partagée avec son CH ($K_{S_i,CH_i} = H_{K_N}(S_i \parallel CH_i \parallel Level_i)$, où $H()$ est une fonction de hachage à sens unique).

Après la phase de formation du cluster et l'établissement de clés, chaque CH génère une clé de cluster K_{C_i} et l'envoie à chaque nœud membre de son cluster chiffré à l'aide des clés par paires établies. Ainsi, chaque nœud de capteur du réseau stocke une clé de cluster K_{C_i} et une clé par paire utilisée pour sécuriser la communication avec son CH, en plus des deux clés pré-distribuées. Cependant, la clé K_N sera supprimée.

Mamun et autres [103] présentent KMP, un système de gestion de clé sécurisé adopté pour les réseaux de capteurs sans fil hiérarchique. Ce système utilise la pré-distribution de clés partielles dans les nœuds avant le déploiement. Dans un premier temps, un pool des clés partielles est généré par la SB, puis chaque nœud de capteur est pré-distribué avec une clé de réseau N_K (partagée par tous les nœuds), un pool des clés partielles P, une liste d'index L (des clés partielles) et un identifiant unique ID. Dans un deuxième temps, pour chaque cluster i, la station de base envoie à chaque CH_i une liste d'index des clés partielles identifiée par LPK_i afin d'utiliser dans son cluster. Le cluster Head (CH) diffuse ensuite le LPK_i à tous les nœuds membres du cluster i. Une fois qu'un nœud de capteur a enseignai quelles clés partielles il utilisera avec son CH, il supprime le reste des clés partielles de P qui a été inséré.

À ce point, les nœuds de capteurs sont prêts à établir les clés de communication. Pour chaque membre du cluster i, le CH_i envoie une liste d'ordres unique O_{CH_i} contenant la liste ordonnée des numéros d'index de q clés partielles sélectionnées à partir de LPK_i. En réponse, chaque nœud membre A_i crée également une liste d'ordres O_{A_i} avec un ordre différent des index et envoie l'O_{A_i} à CH_i. Les nœuds capteurs A_i et le cluster Head CH_i peuvent construisez maintenant leurs clés de communication secrètes pour chaque cycle. Pour plus de simplicité, les auteurs ont utilisé une simple fonction de concaténation pour créer une clé de communication à partir des deux clés partielles ($K_{A_iCH_i}^t = L(O_{A_i}[t]) \parallel L(O_{CH_i}[t])$, où O_{Ai} $[t]$ renvoie le $t^{ème}$ index de A_i). Après chaque cycle, le CH et les membres peuvent régénérer des nouvelles listes d'ordres pour créer des nouvelles clés de communication.

3.5. Métriques d'évaluation

Un système de gestion de clés doit répondre aux exigences de sécurité traditionnelles suivantes: confidentialité, authentification, fraicheur, intégrité et non-répudiation. De plus, en fonction des fonctionnalités et de l'environnement d'application de la gestion de clés, certaines métriques sont généralement prises en compte. Par conséquent, cette section définit les métriques les plus couramment utilisées pour évaluer les différentes méthodes de gestion de clés proposées pour les RCSF.

3.5.1. Efficacité des ressources

Comme les nœuds de capteur sont limités en ressources, un bon schéma de gestion de clés ne doit pas consommer une grande quantité de ressources. Les ressources ici pourraient être :

- La puissance de calcul : est mesuré en termes de quantité de cycles de processeur nécessaires pour l'établissement de clés. Par exemple, la cryptographie à clé asymétrique n'est pas prise en compte en raison de la forte exigence de calcul.
- La capacité de communication : détermine le nombre de messages échangés requis pour la gestion de clés. Étant donné que la communication domine la consommation d'énergie des nœuds capteurs. Par conséquent, le nombre de messages doit être réduit que possible.
-L'espace de stockage : est la quantité de mémoire nécessaire pour stocker les informations de sécurité, telles que des clés (par exemple clés publique / privée, clés par paire) et un certificat d'utilisateur (par exemple, ID). La mémoire dans les nœuds capteurs n'est que des dizaines de kilo-octets, ce qui implique que le schéma de gestion de clés ne peut pas stocker trop de clé dans les nœuds capteurs.

3.5.2. Résilience contre la capture de nœud

Une autre métrique qui doit être respecté est la résilience contre la capture de nœud. L'adversaire peut physiquement compromettre les nœuds capteurs, dans ce cas il peut utiliser les informations stockées dans les nœuds capteurs compromis pour lancer des nouvelles attaques. Dans le contexte d'établissement de clés, l'adversaire peut essayer de déduire la clé partagée entre les nœuds capteurs non compromis. Ainsi, la résilience est définie pour évaluer la capacité à protéger la confidentialité des données générées et échangées entre les nœuds capteurs non compromis.

3.5.3. La connectivité

La connectivité de clé est définie comme la probabilité qu'une paire de nœuds puissent établir une clé commune entre eux. La connectivité locale prend en compte la connectivité entre toute paire de nœuds voisins, tandis que la connectivité globale fait référence à la connectivité de l'ensemble du réseau. Comme nous avons étudié dans la partie 3.4.1.1, dans un grand nombre des schémas de gestion de clés probabiliste des paires des nœuds capteurs ne peuvent pas avoir une clé partagée, cela permet de limiter la connectivité du réseau. Pour assurer la continuité de la sécurité, La méthode de gestion de clés doit être capable d'assurer une bonne connectivité du réseau.

3.5.4. Passage à l'échelle (scalability)

Le nombre de nœuds de capteurs déployés dans la zone de détection peut atteindre plusieurs centaines, voire plusieurs milliers. De plus, pendant toute la durée de vie du réseau de capteurs, des nœuds peuvent rejoindre ou quitter. Par conséquent, les solutions de gestion de clés doivent pouvoir s'adapter à différentes tailles de réseau. Dans le même temps, les fonctionnalités de sécurité et d'efficacité des petits réseaux doivent être conservées lorsqu'elles sont appliquées aux réseaux plus grands.

Schémas			Le Type de nœud	Critères de comparaison					
				Coût de ressources utilisées					
Basé sur l'architectur	Auteurs des schémas	Type		Espace mémoire (stockage des clés)	Coût de communication	Coût de calcul	Connectivité	Résilience	Passage à l'échelle
Plate	Eschenauer et autres [85]	Prob	-	$2\times m$	$d+1$	Recherche	p ($voir$[85])	Moyenne	Juste
	Chan, Perrig et Song [86]	Prob	-	$2\times m$	$d+1$	Recherche	p' ($voir$[86])	Moyenne	Limité
	Du et autres. [87]	Prob	-	$d-1$	$d+1$	Recherche	$p"$ ($voir$[87])	Moyenne	Juste
	Blom [94]	Déter	-	$2(\lambda+1)$	$d+1$	$MulVec(\lambda+1)$	1	Moyenne	Limité
	Blundo et autres [91]	Déter	-	$\lambda+1$	$d+1$	$EvalPoly(1)$	1	Moyenne	Oui
	Chan et Perrig [89]	Déter	-	$2(\sqrt{N}-1)$	$d+1$	Recherche	$1/\sqrt{N}$	Moyenne	Limité
	Lai et autres [95]	Déter	-	1	$2\times d$	PRF	1	faible	Oui
	Zhang et autres [98]	Déter	-	$2 + tableau\ de\ voisins$	$d+1$	$1.Enc + d.Dec + (2d+1).PRF + 1.MAC$	1	élevée	Limité
Hiérarchique	Zhu et autres [97]	Déter	-	$3d + 2 + la\ chaine\ de\ clés$	$2d+1$	$d.Enc + d.MAC + (3d+1).PRF$	1	élevée	Oui
	Zhang et Wang [100]	Déter	CH	$n+4$	$2n+10$	$(n+6).Enc + (n+4).Dec + (n+1).PRF + (n+1).DH$	1	Moyenne	Oui
			CM	4	5	$1.Enc + 4.Dec + 2.PRF + 1.DH$			
			ASS	$n+3$	$n+4$	$5.Dec + (n-1).Enc + 1.DH + (n+1).PRF$			
	Messai et autres [101]	Déter	CH	$n + 4$	$n+d+2$	$(d+1).Dec + (n+1).Enc + 1.MAC + n.H$	1	Moyenne	Oui
			CM	5	$d+2$	$(d+1).Dec + 1.Enc + 1.MAC + 1.H$			
	Mamun et autres [103]	Déter	CH	$pool\ de\ clés + la\ liste\ d'index\ des\ clés + 1 + 2.n\ (la\ liste\ d'ordres\ des\ numéros\ index)$	$2 + 2.q.x.n + (1+m.x).n$	$(1 + m.x + q.x.n).Dec + (1 + m.x.n + q.x.n).Enc$	1	Moyenne	Oui
			CM	$pool\ de\ clés + la\ liste\ d'index\ des\ clés + 1 + 2\ (la\ liste\ d'ordres\ des\ numéros\ d'\ index)$	$1 + 2.q.x + m.x$	$(m.x + q.x).Dec + (1 + q.x).Enc$	1	Moyenne	Oui

Prob : schéma probabiliste. **Déter :** schéma déterministe. **CH:** le nœud cluster-head. **CM:** le nœud membre de cluster. **Ass:** le nœud assistant. **d:** le nombre de voisins. **n:** le nombre de membres de cluster. **C:** le nombre de clusters. **N:** la taille de réseau. **λ :** le seuil prévu pour compromettre la sécurité de réseau. **Enc:** la fonction de cryptage. **Dec:** la fonction de décryptage. **Recherche :** rechercher une ou plusieurs clés dans une chaîne. **MulVec(x):** multiplication de deux vecteurs de taille x. **EvalPoly(x) :** évaluation polynomiale sur x points. **MAC:** le code d'authentification de message. **PRF:** la fonction pseudo aléatoire. **DH:** l'algorithme de Diffie-Hellman. **H:** la fonction de hachage. **m:** le nombre des clés partielles. **q:** le nombre d'index dans la liste d'ordres. **x= index_taille /packet_taille. index_taille**: le nombre de bits requis pour identifier chaque clé partielle. **packet _taille :** la taille(en bits) de données échangées dans le paquet.

Tableau 3.1: Comparaison des schémas proposés pour la gestion de clés dans un RCSF

3.6. Comparaison

Après avoir étudié différents types de schémas de gestion de clés basé sur la pré-distribution : probabiliste et déterministe, nous sommes maintenant en mesure de les comparer. Dans le tableau 3.1, nous comparons : le passage à l'échelle, la connectivité de clé, la résilience et le coût de ressources utilisées pour l'établissement de clés. Nous précisons également la nature de la topologie de réseau avec laquelle ces schémas sont destinés: plat ou hiérarchique.

Nous quantifions la résilience avec les trois valeurs suivantes: (i) fort: signifie que le nœud compromis ne peut pas affecter les nœuds non compromis. (ii) moyenne: fait référence au nœud compromis affecte moins de nœuds non compromis. (iii) faible: indique que la compromission d'un nœud entraîne la compromission de l'ensemble du réseau. Alors que la colonne connectivité peut prendre des valeurs différentes de probabilité que deux nœuds (ou plus) partagent une clé. En ce qui concerne le passage à l'échelle (Scalability). Pour le quantifier, nous utilisons les valeurs suivantes: (i) Oui: le protocole n'entraîne pas de coûts supplémentaires lorsque le nombre de nœuds dans le réseau augmente, (ii) Juste: le protocole induit un coût raisonnable lorsque le nombre de nœuds augmente, et (iii) limité: le coût du protocole dépend du nombre de nœuds.

Notons que le stockage en mémoire évalué dans le tableau prend en considération le nombre exact de clés, le nombre des identificateurs de clés, ..., etc. La communication est l'opération la plus consommatrice d'énergie réalisée par un nœud de capteur. Le coût de communication est mesuré par le nombre des paquets envoyés et reçus par un nœud de capteur. Le coût de traitement est déterminé par le nombre de fonctions requises pour l'établissement de clés ; «PRF», «H», «MAC», «Enc», «Dec», «Recherche», «MulVec(x)» et «EvalPoly(x)» désignent respectivement les fonctions pseudo-aléatoire, hachage, Code d'Authentification de Message, cryptage, décryptage, recherche (rechercher une ou plusieurs clés dans une chaîne), multiplication de deux vecteurs de taille x et évaluation polynomiale sur x points.

Les schémas [85] [86] [87], représentant des schémas probabilistes qui pré-chargent les nœuds avec un trousseau de clé. Nous observons, qu'ils ne demande pas beaucoup de coût de calcul et consomment peu d'énergie. Cependant, l'espace mémoire utilisé pour stocker les grandes tailles des trousseaux de clés rendent ces schémas plus coûteux en termes d'occupation d'espace mémoire. Pour la connectivité dans ces derniers, elle est variante et dépend de la taille du trousseau de clés stocké dans les nœuds capteurs. En effet, ils ne peuvent pas résister aux attaques telles que les captures physiques des nœuds. Alors que les schémas déterministes qui

basé sur une pré-distribution de clés aléatoire ([89] [91] [94]) assure une meilleure connectivité entre les nœuds capteurs du réseau, mais ils passent difficilement à l'échelle.

Les schémas BROSK et EDDK utilisent une clé maîtresse dans l'établissement de clés. Ce qui réduit le stockage de clés dans la mémoire des nœuds. En outre, ces schémas déterministes utilisent des algorithmes efficaces (en termes de communication et de calcul) et achèvent une connectivité totale. Si la compromission des nœuds aura lieu après la phase d'établissement de clés c.à.d. après T_{min} , la résilience pour ces derniers est parfaite. Dans le cas contraire, la sécurité de tout le réseau devient en danger.

En effet, les schémas déterministes risquent de ne pas être suffisamment souples pour compenser les paramètres importants. Par exemple, nous voyons qu'EDDK présente des problèmes de passage à l'échelle.

D'après l'étude faite, vu que ces protocoles se basent sur une topologie plate. Cependant, peu de travaux [97,99] [100] [101,102] [103] envisagent un schéma de gestion de clés qui tire parti de la topologie hiérarchique des RCSF.

Nous pouvons constater d'après le tableau que les schémas basés sur une topologie hiérarchique (LEAP, SEHKM, EAHKM et KMP), présentent une bonne résilience parce qu'ils centralisent la tâche d'établissement de clés au niveau des chefs de groupes (CH), qui sont supposés être sécurisés. Nous remarquons aussi dans le tableau de comparaison que l'utilisation d'une topologie hiérarchique peut simplifier et améliorer le passage à l'échelle et l'efficacité de la procédure de gestion de clés.

3.7. Conclusion

La gestion de clés est le mécanisme de sécurité fondamental dans les réseaux de capteurs sans fil. Il s'agit des technologies de base pour établir des communications sécurisées entre les capteurs d'un réseau sans fil. En raison de la nature limitée des ressources sur les nœuds de capteurs sans fil, de nombreux chercheurs ont utilisé différentes techniques pour proposer différents types de mécanismes de gestion de clés. Dans ce chapitre, nous avons présenté un état de l'art qui détaille les composants d'un protocole de gestion de clés destiné aux RCSF en particulier. Ensuite, quelques solutions existantes sont classées et décrites. Afin de faire une comparaison entre les différentes solutions décrites, nous résumons les métriques d'évaluation pour la gestion de clés dans les réseaux de capteurs sans fil. Pour conclure, Toutes les méthodes que nous avons étudiées dans ce chapitre possèdent de grands avantages. Cependant, il est difficile d'assurer un niveau de sécurité élevé avec une consommation d'énergie minimale.

Par conséquent, un protocole de sécurité doit être intelligent en s'appuyant sur des routines adaptatives afin de fournir le meilleur compromis entre la fiabilité et la consommation d'énergie.

4

Une gestion dynamique et intelligente de clés dédiée aux RCSF hiérarchiques

Sommaire

4.1. Introduction

Malgré les prouesses et avancés technologiques, il est actuellement évident de constater que les nœuds capteurs possèdent une faible capacité en termes de calcul, de stockage et d'énergie, ce qui les rend vulnérables et faciles à corrompre afin de récupérer les informations qu'ils possèdent. Dans ce contexte, un mécanisme de sécurité est en effet nécessaire pour la majorité des applications basées sur le RCSF, en particulier lors de l'utilisation des nœuds capteurs dans un lieu peu sûr. La gestion de clés constitue la pierre angulaire des autres mécanismes de sécurité, car presque tous les mécanismes de sécurité reposent sur le cryptage ou sont liés à celui-ci.

Dans le chapitre 3, toutes les méthodes de gestion de clés que nous avons étudiées ont été proposés dans le but d'avoir un schéma fiable, qui garantit un niveau élevé de sécurité, et optimise les métriques des performances liées à l'énergie. Toutefois, il est extrêmement difficile d'assurer un niveau de sécurité élevé avec une consommation d'énergie minimale.

Afin d'économiser l'énergie, un RCSF est généralement organisé sous forme d'une topologie basée sur des clusters qui forme un réseau hiérarchique. L'utilisation d'une topologie hiérarchique peut simplifier et améliorer la scalabilité et même améliorer l'efficacité de la procédure de gestion de clés. Cependant, peu de travaux envisagent un schéma de gestion de clés dédiée à une topologie hiérarchique des RCSF.

Dans ce chapitre, nous exposons notre système développé baptisé SKWN (**S**mart and dynamic **K**ey management scheme for hierarchical **W**ireless sensor **N**etworks), dédié pour une gestion de clés innovante associée aux réseaux de capteurs ayant une topologie hiérarchique. Nous commencerons d'abord par présenter la motivation derrière cette conception, ensuite nous présenterons les détails de cette dernière. Nous évaluons par la suite les performances par rapport aux approches existantes. En effet, plusieurs aspects liés à la scalabilité, aux performances et à la fiabilité doivent être prises en compte.

4.2. Motivation

Généralement, la cryptographie symétrique est utilisée pour sécuriser l'échange de données. C'est en effet l'un des mécanismes de sécurité les mieux adaptés aux RCSF. Avec un tel mécanisme, l'interception des communications ne permet pas de récupérer les données envoyées par un capteur. Un tel mécanisme repose sur l'utilisation d'une clé secrète afin de chiffrer tous les échanges. Cette clé est partagée entre l'expéditeur et le destinataire des données. Toutefois, le point faible de cette stratégie est la capacité d'intercepter et / ou de récupérer la clé cryptographique. Par conséquent, l'établissement de clés est une tâche importante et cruciale qui permet d'assurer la fiabilité de ce type de mécanisme de sécurité. Une telle tâche est généralement gérée au sein d'un protocole (également appelé schéma) dont le but est de pré-distribuer les clés cryptographiques, de révoquer les clés si les nœuds quittent le réseau, de renouveler les clés en cas de menace et d'attribuer des nouvelles clés lorsque certaines clés expirent. Un protocole doit également pouvoir gérer les clés dans le cas d'une intégration d'un nouveau nœud, afin de garantir la flexibilité et la scalabilité du réseau. En résumé, le premier défi pour un protocole de gestion de clés consiste à garantir une fiabilité en temps réel.

Les protocoles de gestion de clés doivent également prendre en compte d'autres aspects ayants un impact sur les performances du réseau RCSF. En effet, en s'appuyant sur les routines de sécurité dites complexes (par exemple, la cryptographie asymétrique), la durée de vie du réseau sera réduite en raison de la consommation d'énergie. Cette consommation est principalement liée au coût de traitement de l'unité de calcule (Central Processing Unit CPU) et au nombre de

messages échangés par les capteurs. Par conséquent, un protocole de gestion de clé doit également optimiser la consommation d'énergie en utilisant des simples routines de calcul et un nombre réduit de messages.

D'autre part, pour assurer la sécurité du protocole de gestion de clés, il est important de pouvoir effectuer diverses opérations cryptographiques, par exemple le cryptage et l'authentification. De plus, dans un RCSF, étant donné que certaines applications (par exemple, militaires) ont besoin de plus de sécurité que d'autres (par exemple, agricultures), la complexité de la cryptographie (utilisée dans le protocole de gestion de clés) est liée à l'environnement de déploiement du réseau. Un protocole de sécurité doit donc être intelligent en s'appuyant sur des routines adaptatives afin de fournir le meilleur compromis entre la fiabilité et la consommation d'énergie.

Ce compromis pourrait être trouvé en s'appuyant sur une décision dynamique de la complexité de la cryptographie par rapport à l'environnement de déploiement. Un niveau de cryptage très élevé nécessite un effort de calcul important et par conséquent une consommation d'énergie élevée. Le deuxième défi d'un protocole de gestion de clés est de répondre efficacement aux exigences de sécurité.

Bien que la gestion de clés et plus généralement des mécanismes de sécurité aient fait l'objet des études approfondies ces dernières années, ce sujet reste ouvert. En effet, avec le déploiement intensif de RCSF, la nécessité d'un protocole fiable, scalable et économe en énergie reste un réel défi. Un tel protocole devrait donc être intelligent en adaptant dynamiquement ses routines à l'environnement de déploiement du RCSF.

4.2. Spécifications générales

Dans cette partie, nous spécifions les hypothèses sur le réseau ainsi que le système de détection d'intrusion utilisée pour développer notre proposition.

4.2.1. Modèle du réseau

Dans notre étude, nous nous intéressons aux réseaux RCSF hiérarchiques en clusters en raison de leur capacité à optimiser la consommation d'énergie. Cette capacité est obtenue en distinguant trois types de rôles de nœud: station de base (BS), cluster-head (CH) et membre de cluster (CM). Un membre du cluster assure la collecte des données, qui doivent être transférées à la station de base via le nœud CH. Ce nœud effectue des opérations d'agrégation et de

filtrage. La station de base gère le réseau et communique avec le monde extérieur. La figure 4.1 montre un exemple de RCSF hiérarchique.

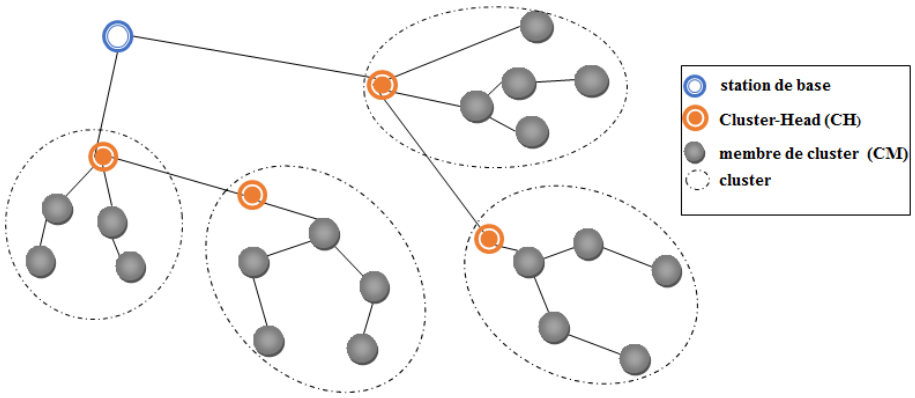

Figure 4.1. Modèle d'architectures hiérarchique pour un RCSF

Tous les nœuds du réseau, à l'exception de la station de base, ont les mêmes ressources. La station de base comprend des installations informatiques et de communication haut de gamme. Les cluster-heads sont uniformément réparties dans le réseau et sont choisies en fonction de la fonction de regroupement, dont la valeur dépend de divers critères tels que la localisation, la portée de communication, les capacités en ressources et en énergie [104]. L'énergie résiduelle de cluster-head est réduite lors des calculs et des communications. Il est donc nécessaire de disposer des mécanismes de rotation du CH qui permettent de prolonger la durée de vie du CH et par conséquent du réseau.

En termes de sécurité, l'organisation des nœuds en clusters doit être prise en compte. En effet, dans les réseaux plats, tous les nœuds ont le même rôle et donc la même importance. Par conséquent, compromettre un nœud n'affectera pas nécessairement les autres nœuds du réseau. D'autre part, dans un réseau hiérarchique, compromettre un cluster-head implique de compromettre tous les nœuds membres du cluster. La rotation du CH devrait également être prise en compte. En fait, lorsqu'un CH est remplacé, il est nécessaire de renouveler les clés des nœuds de la partie du réseau liée au CH remplacé.

4.2.2. La détection d'intrusion

Le système de détection d'intrusion (SDI) est un système qui vérifie le comportement du réseau et trouve les nœuds qui ne fonctionnent pas normalement [105], ce qui peut jouer un rôle important dans la détection et la prévention des attaques de sécurité. Dans un tel système, chaque nœud RCSF est équipé d'un agent SDI [106]. L'une des approches les plus

prometteuses en termes de prévision est l'approche de l'agent de sécurité intelligent nommée ISA (Intelligent Security Agent) [107], qui est utilisé pour faciliter les interactions cross-layer. L'architecture dans cette approche est classée dans le type d'architectures cross-layer à base de communication indirecte (voir chapitre 1), où l'entité intermédiaire et intelligente (ISA) est chargée de l'interaction entre les couches protocolaires.

L'ISA de chaque nœud surveille plusieurs paramètres importants qui aident à détecter les tentatives d'intrusion. En tant que paramètres, nous pouvons citer la consommation d'énergie, la force du signal des nœuds voisins et les ID des nœuds voisins. Au total, l'ISA s'appuie sur onze paramètres intégrés dans un arbre de décision. Basé sur l'apprentissage des règles de décisions, le composant ISA détecte non seulement les intrusions en temps réel, mais fournit également une recommandation sur le "niveau de sécurité", ou "précepte de sécurité" [107], en ce qui concerne l'application et l'environnement de déploiement.

Pour le réseau hiérarchique en clusters, une nouvelle approche de gestion de clés qui place l'ISA au cœur de son fonctionnement représente une opportunité très intéressante qui permet de répondre efficacement aux exigences de sécurité et d'optimiser les ressources utilisées par les capteurs, et par conséquent, il prolonge durée de vie du réseau.

4.3. L'agent de sécurité intelligent

Dans cette section, nous donnons quelques détails sur l'ISA que nous utilisons comme composant principal pour la détection des intrusions. Ce composant est intégré, en tant que composant séparé, dans l'architecture de nœud (voir la figure 4.2). L'ISA prend ses responsabilités de toutes les décisions de sécurité. C'est l'équivalent du module de plateforme sécurisée (TPM) [108] sur un ordinateur personnel.

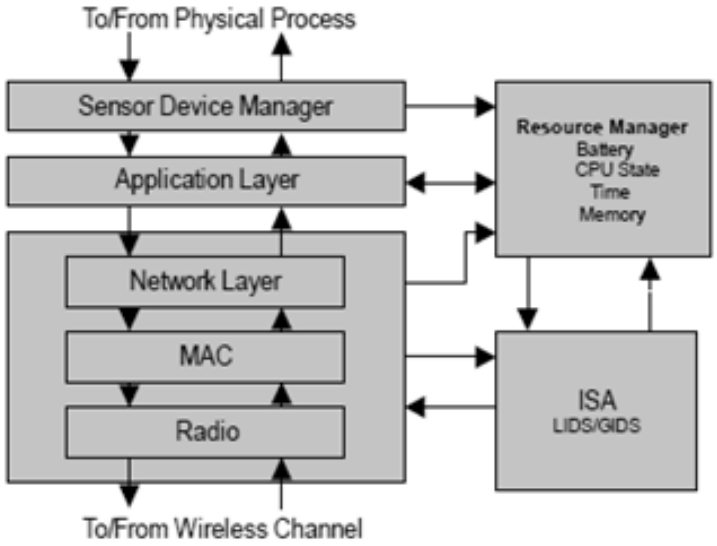

Figure 4.2. L'intégration du composant ISA dans l'architecture du nœud [107]

L'ISA peut échanger des paramètres avec toutes les couches de protocole, comme le gestionnaire de ressources (Ressource Manager). De plus, comme indiquer par Sharma et Ghose dans [109], l'ISA a été introduite pour réduire au minimum les surcoûts créés par les architectures cross-layer. En effet, la sécurité cross-layer introduit un surcoût important dans le maintien d'une interface entre différentes couches de protocole afin d'échanger plusieurs paramètres importants.

Ce composant est le seul contrôleur de toutes les options de sécurité. Chaque fois qu'un paquet est envoyé ou qu'un canal de communication est ouvert, l'ISA est consultée en ce qui concerne la sécurité. L'ISA de chaque nœud surveille plusieurs paramètres importants permettant de détecter les tentatives d'intrusion locales et globales. Un ISA signale à la station de base toute anomalie liée à ces paramètres.

Il offre deux fonctionnalités liées à la détection d'intrusion: 1) le système de détection d'intrusion local (LIDS : local intrusion detection system), qui est lié uniquement au nœud sur lequel ISA est installé, et 2) le système de détection d'intrusion globale (GIDS : global intrusion detection system), qui permet de surveiller les voisins du nœud sur lequel ISA est installé.

Parmi les paramètres surveillés pour le LIDS figurent les valeurs détectées, le taux de collision des paquets, les paquets non valides et le nombre de paquets retransmis. D'autre part, le sous-système GIDS surveille des paramètres tels que la consommation d'énergie, la force du signal des nœuds voisins et les identifiants des nœuds voisins.

Outre la détection d'intrusion, chaque composant ISA stocke un «niveau de sécurité» ou un «percept de sécurité». Ce paramètre change lorsqu'une anomalie est détectée afin de répondre à la menace perçue. Cela permet au réseau de fournir uniquement le niveau de sécurité nécessaire. En effet, quand il n'y a pas des attaquants, l'utilisation d'un mécanisme de sécurité coûteux n'a pas de sens.

En fonction de la perception actuelle, l'ISA déterminera une réaction adaptative au niveau de sécurité. Cette décision peut être prise en s'appuyant sur de nombreuses politiques et recommandations qui sont données au moment du déploiement ou après. Les informations percept sont collectées à partir de différentes couches à l'aide d'interactions entre couches et à partir du gestionnaire de ressources. Cette information regroupe:

- Type d'information.
- Mémoire disponible.
- Energie disponible.
- Niveau de confiance des nœuds voisins (si une menace pour la sécurité est détectée).
- Stratégies et recommandations prédéfinies.

Une arborescence binaire de décision est gérée par l'ISA afin de recommander le niveau de sécurité en ce qui concerne les informations de perception. Cette structure doit être mise à jour de temps en temps à l'aide des nouvelles entrées. Il convient de souligner que le bon fonctionnement d'ISA dépend en grande partie aux politiques et recommandations prédéfinies correctes.

4.4. Le système de gestion de clés proposé

Dans cette section, nous abordons d'abord l'interaction entre nos sous-schémas de gestion de clés et le composant ISA. En effet, nous proposons trois sous-schémas pour gérer respectivement l'établissement de clés, le renouvellement de clés et l'intégration d'un nouveau nœud. Nous donnons ensuite une vue d'ensemble sur le schéma proposé puis nous le détaillons dans les sections suivantes.

4.4.1. Intégration du composant ISA

Un protocole de gestion de clés nécessite le transfert de plusieurs messages entre les différents nœuds du réseau. Ces messages sont principalement liés à la génération de clés cryptographiques. Afin d'éviter la fuite de ces messages, nous nous appuyons sur l'algorithme RC5 pour assurer le cryptage / décryptage [110]. Cet algorithme assure également la génération du code MAC (Message Authentication Code) garantissant l'authenticité du message.

De plus, RC5 est un algorithme de cryptage paramétré qui repose sur une taille de bloc variable, un nombre de tours variable et une longueur variable de clé secrète. RC5 est hautement configurable et dépend du nombre de tours qui varie de 0 à 255. Il est évident que les messages chiffrés avec 255 tours sont plus difficiles à déchiffrer qu'un chiffrement avec 100 tours. Le nombre de tours a un effet proportionnel sur la sécurité du RC5, qui influe sur la consommation d'énergie.

Dans notre approche, nous distinguons trois niveaux de sécurité différents: le niveau 0, destiné aux applications nécessitants une sécurité faible, ou dans le cas d'une nouvelle élection d'un CH. Le niveau 1, un niveau intermédiaire utilisé lorsqu'une menace (pas très importante) est détectée ou dans le cas où une application nécessite un tel niveau de sécurité. Finalement, le niveau 2, c'est le niveau le plus élevé utilisé lorsqu'une menace majeure est détectée et / ou que l'application nécessite une sécurité renforcée. En ce qui concerne RC5, les trois niveaux nécessitent les paramètres suivants:

- Niveau 2: RC5 utilisent une clé de 64 bits et 12 tours
- Niveau 1: RC5 utilisent une clé de 64 bits et 8 tours
- Niveau 0: RC5 utilise une clé de 64 bits et 4 tours

Ce niveau est un paramètre géré par l'ISA. Cela changera en fonction de la menace perçue et de l'environnement de déploiement. Chaque sous-routine s'appuie sur ce paramètre dans les opérations cryptographiques. De plus, le niveau sera interprété pour déduire la taille des clés et le nombre de tours à utiliser par le RC5. Il est important de noter que l'augmentation du niveau de sécurité implique la consommation de plus d'énergie. D'un autre côté, le niveau est également utilisé pour établir les clés. Dans ce cas, sa valeur est choisie par l'utilisateur en fonction des besoins de l'application.

Dans notre approche dite SKWN, un ISA est utilisé en tant que composant distinct dans l'architecture du nœud (voir la figure 4.2). Ainsi, un ISA peut échanger des paramètres avec

toutes les couches de protocole telles que le gestionnaire de ressources. Dès qu'une menace est détectée, l'ISA prend une décision sur le niveau de sécurité garantissant le bon fonctionnement du système. Ce niveau est utilisé par l'algorithme de chiffrement (RC5 dans notre cas).
Le composant ISA est intégré à chaque nœud. Cependant, comme nous considérons un réseau hiérarchique, ce composant est activé uniquement pour le nœud CH, ce qui permet de limiter la consommation d'énergie. En effet, un ISA pourrait entraîner une consommation d'énergie importante, notamment si plusieurs nœuds utilisent ce composant. Même si un ISA est activée uniquement sur le nœud CH, le fait de disposer d'un réseau hiérarchique permet au nœud CH de détecter et d'isoler les nœuds membres malveillants. De plus, si l'un des paramètres liés à un autre CH change brusquement, la station de base est avertie par le nœud CH détecteur de cette anomalie afin de l'isoler. Dans une telle situation, une alerte sera déclenchée, ce qui permettra également de déclencher le processus de renouvellement de clés.

4.4.2. Vue d'ensemble du protocole proposé

SKWN est un nouveau schéma de gestion de clés intelligent et dynamique pour les réseaux de capteurs sans fil hiérarchiques. En effet, l'objectif d'un protocole de gestion de clé est : (1) de pré-distribuer des clés cryptographiques, (2) de révoquer des clés si des nœuds quittent le réseau, (3) de renouveler des clés en cas de menace et (4) d'attribuer des nouvelles clés aux nœuds rejoignant le réseau ou à l'expiration de certaines clés. SKWN est scalable et permet de répondre en temps réel à différents types d'intrusion. Nous sommes en mesure de prendre des décisions dynamiques en fonction de la partie ciblée du réseau. Pour cela, notre approche repose sur le composant ISA qui permet de détecter les intrusions et de donner une recommandation sur le niveau de sécurité ou le précepte de sécurité. SKWN comprend trois sous-schémas:
1. Etablissement de clés: permet de gérer les clés cryptographiques avant et pendant le déploiement du réseau. En s'appuyant sur des clés pré-chargées, notre sous-schéma d'établissement de clés est entièrement distribué. En effet, aucune clé secrète ne sera échangée via le réseau.
2. Renouvellement de clés: cela permet de gérer les clés en cas d'intrusion ou de rotation de CH. Nous réagissons par rapport au rôle du nœud impliqué.
3. Intégration du nouveau nœud: permet de gérer les clés dans le cas d'une intégration du nouveau nœud. Notre objectif est de disposer d'un protocole de sécurité garantissant la flexibilité et la scalabilité du réseau.

SKWN permet l'établissement sécurisé de clés sur la base d'un mécanisme cryptographique adaptatif prenant en compte l'environnement de déploiement et les menaces perçues.

4.5. Les sous-schémas de protocole de gestion de clés proposé

Notre proposition repose sur trois sous-schémas: établissement de clés, renouvellement de clés et intégration des nouveaux nœuds capteurs. Dans le premier sous-schéma, différentes clés secrètes sont générées pour les nœuds capteurs. Dans le deuxième, trois stratégies de renouvellement de clés sont proposées dans trois cas, lorsqu'un nœud CH ou un nœud CM sont compromis et lorsqu'un nouveau nœud CH est élu. Enfin, un nouveau mécanisme est proposé afin de gérer les clés liées à l'intégration d'un nouveau nœud. Nous présentons au tableau 4.1 un résumé des notations que nous avons utilisées pour détailler chaque sous-schéma.

4.5.1. L'établissement de clés

Avant le déploiement du réseau de capteurs, deux clés sont pré-distribuées à chaque capteur. Ces clés permettent de sécuriser la phase de déploiement. L'une de ces clés est utilisée pour sécuriser les communications pendant la phase d'installation de clés et sera effacée après le déploiement de clés. Dans ce qui suit, nous avons détaillé chaque étape liée au sous-schéma d'établissement de clés:

4.5.1.1. La pré-distribution de clés

Plusieurs nœuds capteurs sont pré-chargés avec plusieurs informations avant d'être livrés dans la zone de détection. La SB doit pré-charger certain matériel cryptographique dans chaque nœud pour générer des autres clés. Ces matériaux incluent:

- Une clé K_{in} partagée avec la station de base pour chiffrer / déchiffrer les messages du nœud vers la station de base et les messages en sens inverse.
- Une clé K_r partagée par tous les nœuds du réseau, utilisée pour chiffrer / déchiffrer les messages juste après le déploiement.

Une fois les nœuds déployés, ils signalent d'abord leur emplacement physique à la station de base, puis le réseau commence à sélectionner les cluster-heads à l'aide des algorithmes de sélection des cluster-heads [111] [22]. Chaque nœud de capteur reçoit alors l'identifiant de leur CH.

Tableau 4.1 : Acronymes définition

Notation	Explication
id_{CM_j}	Identificateur de membre de cluster j
id_{CH_α}	Identificateur de cluster-head α
id_{BS}	Identificateur de la station de base
L_id_{CM}	Liste contenant les identifiants des nœuds membres du cluster
$E_K(M)$	Chiffrement du message M avec la clé K
$MAC_K(M)$	Code d'authentification de message du message M avec la clé symétrique K
N_S	Nonce généré par le nœud de capteur S
${H_K}^i()$	$i^{ème}$ fonction de hachage avec la clé symétrique K
Lev	Niveau de sécurité pour chaque application
CPT	Compteur reflète le nombre de renouvellement de clés
$S \rightarrow *: M$	Le nœud S diffuse le message M
$A \parallel B$	Concaténation de l'information A avec l'information B
$\oplus$	opération XOR au niveau du bit

4.5.1.2. L'étape d'installation de clés

Dans SKWN, deux fonctions sont appliquées aux messages afin d'assurer les objectifs de sécurité des communications. La première est la fonction $MAC_{K_r}\{\}$ (code d'authentification du message), utilisée pour authentifier les données envoyées. La deuxième est la fonction $E_{K_r}\{\}$, utilisée pour chiffrer les données envoyées. Tandis que, RC5 est utilisé comme algorithme de chiffrement dans ces deux fonctions (voir section 4.5.1). Nous utilisons également le nonce (N_S), utilisé non seulement pour vérifier l'intégrité du paquet envoyé, mais également pour calculer les clés partagées. De plus, le paramètre Lev est utilisé pour déterminer le niveau de sécurité par rapport aux besoins de sécurité. Le type du message est également envoyé dans le paquet afin de déterminer son objectif.

Dans notre approche, l'établissement de clés est considéré pendant une courte période, notée T_{min} [112]. En effet, la probabilité de compromettre un nœud pendant cette période est négligeable. Les étapes suivantes sont effectuées par les nœuds capteur (nœud CH ou nœud CM) afin de s'assurer que des clés distinctes sont établies sur tous les nœuds.

Étape 1. Les cluster-heads lancent la phase d'établissement de clés en diffusant un message $HELLO$ contenant leur identifiant. Ils initient un Timer qui sera déclenché après le temps T_{min}.

$$CH_\alpha \rightarrow *: \ id_{CH_\alpha} \parallel Lev \parallel E_{K_r}\{HELLO, N_{CH_\alpha}\} \parallel MAC_{K_r}\left\{id_{CH_\alpha} \parallel Lev \parallel E_{K_r}\{HELLO, N_{CH_\alpha}\}\right\}$$

Étape 2. Après avoir reçu le message d'initiation de leur cluster-head, le nœud CM authentifie le message $HELLO$ (en vérifiant le MAC) et calcule la clé par paire à l'aide de l'équation suivante:

$$K_{CM_j-CH_\alpha} = H_{K_r}\left(ma\,x\left(id_{\,CM_j}, id_{CH_\alpha}\right) \| N_{CM_j} \oplus N_{CH_\alpha}\right) \quad (4.1)$$

Ils répondent ensuite par le message $HELLO\_REP$ qui contient leur identifiant

$$CM_j \rightarrow *:$$

$$id_{CM_j} \| id_{CH_\alpha} \| E_{K_r}\left\{HELLO\_REP, N_{CM_j}\right\} \| MAC_{K_r}\left\{id_{CM_j} \| id_{CH_\alpha} \| E_{K_r}\left\{HELLO\_REP, N_{CM_j}\right\}\right\}$$

Étape 3. Après la réception du message $HELLO\_REP$, chaque nœud CH commence par vérifier l'authenticité du message (en vérifiant le MAC) et identifie ensuite l'ensemble des capteurs qui se trouvent dans le même cluster. Ils diffusent ensuite un message de requête contenant la liste L_id_{CM} aux autres CH.

$$CH_\alpha \rightarrow *: id_{CH_\alpha} \| Lev \| E_{K_r}\{HELLO\_REQ, N_{CH_\alpha}, L_{\_id\,CM}\}$$

$$\| MAC_{K_r}\left\{id_{CH_\alpha} \| Lev \| E_{K_r}\{HELLO\_REQ, N_{CH_\alpha}, L\_id_{CM}\}\right\}$$

Où $L\_id_{CM} = \{id_{CM_1}, id_{CM_2}, id_{CM_3}, \dots, id_{CM_n}\}$

Ici, n est le nombre de nœuds CM dans chaque cluster.

Étape 4. Chaque nœud CM, après avoir vérifié l'authenticité du message $HELLO\_REP$ des autres nœuds CM, identifie les nœuds membres appartenant au même cluster. Il calcule ensuite la clé par paire partagée avec chaque nœud membre.

$$K_{CM_i-CM_j} = H_{K_r}\left(\min\left(id_{CM_i}, id_{CM_j}\right) \| N_{CM_i} \oplus N_{CM_j}\right) \quad (4.2)$$

Étape 5. Chaque cluster-head, après avoir reçu les messages $HELLO\_REQ$ des autres nœuds CH, vérifie l'authenticité (en vérifiant le MAC) et calcule les clés par paires partagées entre eux.

$$K_{CH_\alpha-CH_\beta} = H_{K_r}(max(id_{CH_\alpha}, id_{\,CH_\beta}) \| min(id_{CH_\alpha}, id_{\,CH_\beta}) \| N_{CH_\alpha} \oplus N_{CH_\beta}) \quad (4.3)$$

Pour chaque interaction dans la phase d'installation, un MAC de message nécessite 4 octets, 2 octets pour les identifications de source et de destination et 1 octet pour le niveau de sécurité. Nous considérons également 8 octets pour les données cryptées. Lorsque les données cryptées correspondent au type de message et au nonce, nous avons respectivement besoin de 1 et 4

octets. Dans le cas où le message diffusé contient la liste d'identification L_id_{CM} de l'ensemble des nœuds membres, la taille des données cryptées dépend non seulement du type de message et du nonce, mais également du nombre de nœuds CM dans chaque cluster. Par conséquent, la taille du message pour $HELLO$ et $HELLO\_REP$ est respectivement de 15 et 16 octets.

Selon les étapes décrites ci-dessus, tous les nœuds CH et CM du réseau établissent des clés distinctes. Dans cette phase, chaque taille de clé est de 8 octets. La figure 4.3 résume le processus d'établissement de clés.

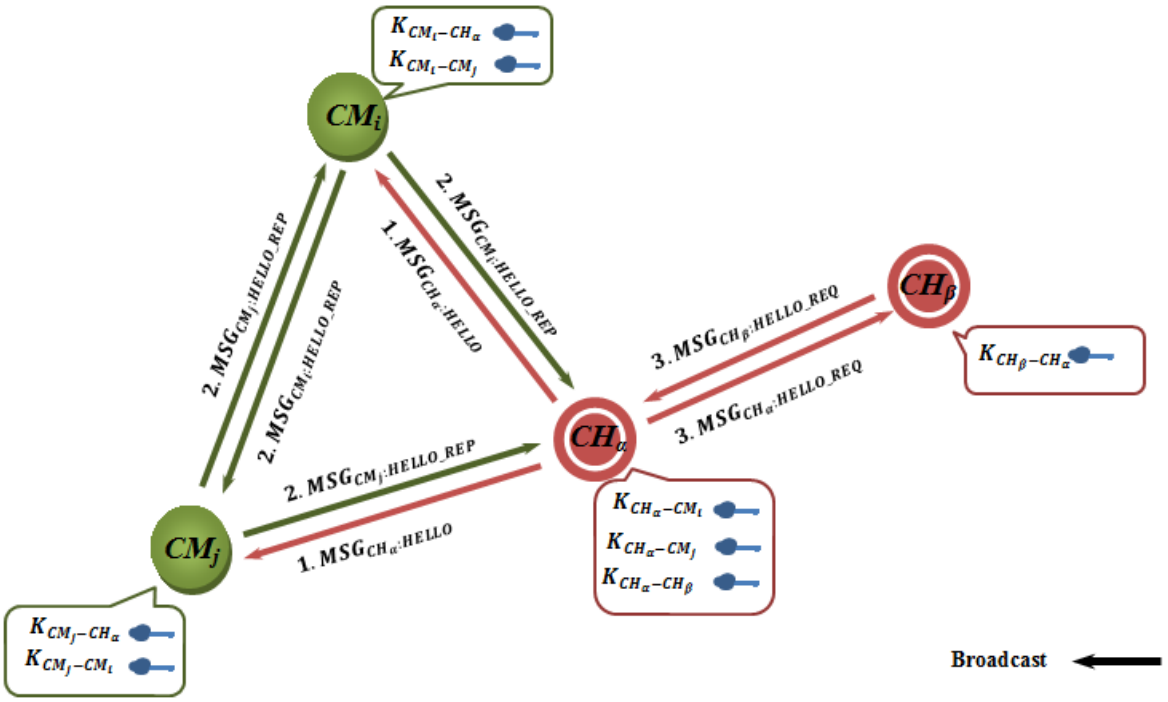

Figure 4.3. Le processus d'établissement de clés

4.5.1.3. Effacement de clés

À la fin de la phase d'installation de clés, les clés K_r et K_{in} seraient supprimées de la mémoire du nœud. Dans ce cas, des nouvelles clés seront calculées avant d'effacer les clés précédentes:

$$K_{in} = H_{K_{in}}{}^{CPT}(K_{in}) \tag{4.4}$$

$$K_r = H_{K_r}{}^{CPT}(K_r) \tag{4.5}$$

CPT est un compteur, initialisé à zéro, qui reflète le nombre de renouvellements de clés. Le CPT dans ce cas est égal à 1.

4.5.2. Renouvellement de clés

Les schémas de gestion de clés pour RCSF doivent être dynamiques en renouvelant les clés établies entre les nœuds capteurs. Dans ce sous-schéma, un renouvellement de clés est effectué à la demande. Il y a trois raisons pour changer une clé cryptographique:

- Détection d'un adversaire qui a compromis un CH.
- Détection d'un adversaire qui a compromis un CM.
- Élection d'un nouveau CH.

Nous sécurisons le processus de renouvellement de clés en utilisant différents niveaux de sécurité. En effet, le niveau de sécurité change par rapport à la raison du renouvellement de clés.

Au fait, compromettre un nœud CH implique de compromettre tous les nœuds du cluster et les nœuds CH associés, le premier cas nécessite un niveau de sécurité supérieur à celui du second.

Trois niveaux de sécurité différents sont utilisés en fonction des trois cas de renouvellement (voir section 4.5.1). Notre objectif principal est de garantir l'utilisation efficace de l'énergie par les nœuds capteurs. Dans ce qui suit, nous discutons des processus de renouvellement pour chaque cas.

4.5.2.1. Processus de renouvellement de clés pour le cluster-head compromis (Comp_CH)

Dans ce cas, nous présentons le processus de génération et de modification de certains types de clés en raison de la détection d'un adversaire compromettant un cluster-head. Après avoir compromis un nœud CH, un adversaire ne peut pas récupérer les informations de l'ensemble du réseau. En effet, seuls les nœuds directement liés (c'est-à-dire les membres de cluster et les CH associés) sont compromis. La procédure de renouvellement de clé est décrite comme suit:

Étape 1. La station de base commence par élire un nouveau CH individuellement. Ensuite, il envoie un message contenant le type de message pour en déterminer le but.

$$BS \rightarrow CH_{New}: id_{BS} \parallel E_{K_{in}}\{REFRESH\_CHcomp, CPT, N_{BS}\}$$

$$\parallel MAC_{K_{in}}\left\{id_{BS} \parallel E_{K_{in}}\{REFRESH\_CHcomp, CPT, N_{BS}\}\right\}$$

Étape 2. Après avoir reçu le message de rafraîchissement, le nouveau nœud CH diffuse le message $REFRESH\_CHcomp$ avec le niveau de sécurité requis.

$$CH_{New} \rightarrow *: id_{CH_{New}} \parallel Lev \parallel E_{K_r}\left\{id_{CH_{compromise}}, REFRESH\_CHcomp, CPT, N_{CH_{New}}\right\}$$

$$\parallel MAC_{K_r}\left\{id_{CH_{New}} \parallel Lev \parallel E_{K_r}\{id_{CH\_compromise}, REFRESH\_CHcomp, CPT, N_{CH_{New}}\}\right\}$$

Étape 3. Après avoir reçu le message $REFRESH\_CHcomp$, chaque nœud CH efface la clé partagée avec le CH compromis et calcule une nouvelle clé par paire partagée avec le nouveau nœud CH.

$$K_{CH_\alpha - CH_{New}} = H_{K_r}^{CPT}(\max(id_{CH_\alpha}, id_{CH_{New}}) \parallel \min(id_{CH_\alpha}, id_{CH_{New}}) \parallel CPT) \quad (4.6)$$

Le message $REFRESH\_CHcomp\_REP$ est ensuite envoyé au nouveau CH. Ce message contient la liste d'identification L_id_{CM} de l'ensemble des nœuds membres.

$$CH_\alpha \rightarrow CH_{New}: id_{CH_\alpha} \parallel Lev \parallel E_{K_r}\{REFRESH\_CHcomp\_REP, CPT, N_{CH_\alpha}, L\_id_{CM}\}$$

$$\parallel MAC_{K_r}\left\{id_{CH_\alpha} \parallel Lev \parallel E_{K_r}\{REFRESH\_CHcomp\_REP, CPT, N_{CH_\alpha}, L\_id_{CM}\}\right\}$$

Étape 4. Chaque nœud CM (membre du groupe du nouveau CH), recevant le message $REFRESH\_CHcomp$, efface la clé partagée avec le nœud CH compromis et calcule une nouvelle clé où

$$K_{CM_j - CH_{New}} = H_{K_r}^{CPT}(K_{CM_j - CM_l} \parallel id_{CH_{compromise}}) \quad (4.7)$$

$K_{CM_j - CM_l}$ dans la formule est la clé précédente partagée entre le nœud CM_j et le noeud $CM_l = CH_{New}$

En réponse, le nœud CM diffuse un message $REFRESH\_CHcomp\_REP$.

$$CM_j \rightarrow *: id_{CM_j} \parallel id_{CH_{New}} \parallel E_{K_r}\left\{REFRESH\_CHcomp\_REP, CPT, N_{CM_j}\right\}$$

$$\parallel MAC_{K_r}\left\{id_{CM_j} \parallel id_{CH_{New}} \parallel E_{K_r}\left\{REFRESH\_CHcomp\_REP, CPT, N_{CM_j}\right\}\right\}$$

Étape 5. Le nouveau CH reçoit deux types de messages $REFRESH\_CHcomp\_REP$. Le premier est diffusé par tous les membres de leur cluster pour vérifier son authenticité et rafraîchir la clé partagée entre eux comme la formule (4.7). Le second est envoyé par les autres nœuds CH afin de rafraîchir la clé à l'aide de la formule (4.6).

Étape 6. Après avoir reçu le message $REFRESH\_CHcomp\_REP$ diffusé par les autres nœuds CM du même cluster, le nœud CM vérifie l'authenticité et rafraîchit les clés précédentes avec:

$$K_{CM_i-CM_j} = H_{K_r}^{CPT}(K_{CM_i-CM_j}) \quad (4.8)$$

Une fois ces étapes terminées, le nœud CH compromis sera isolé, chaque clé partagée avec ce nœud sera effacée et toutes les clés nécessaires seront actualisées. Un aperçu du processus de renouvellement est présenté à la figure 4.4.

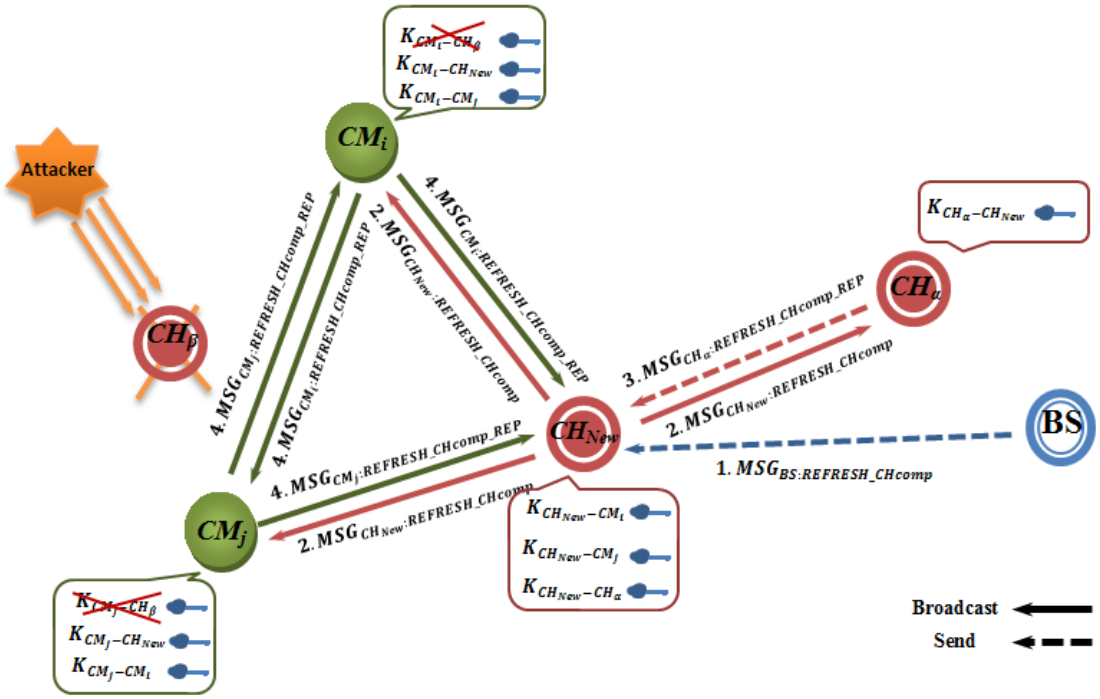

Figure 4.4. Le processus de renouvellement de clés pour un cluster-head compromis (Comp_CH)

4.5.2.2. Processus de renouvellement de clés pour un membre de cluster compromis (Comp_CM)

La compromission d'un nœud CM n'affecte pas le réseau comme pour compromettre un nœud CH. En effet, un nœud CH est plus important qu'un nœud CM. La procédure détaillée pour le processus de renouvellement de clés dans ce cas est décrite comme suit:

Étape 1. Lorsqu'un nœud CH détecte un adversaire, qui compromet un membre du cluster, il diffuse un message $REFRESH\_CMcomp$ pour notifier les autres nœuds et rafraîchir les clés. Le message de rafraîchissement doit contenir l'ID de nœud compromis et le niveau de sécurité requis.

$$CH_\alpha \rightarrow *: id_{CH_\alpha} \parallel Lev \parallel E_{K_r}\{id_{CM\_compromise}, REFRESH\_CMcomp, CPT, N_{CH_\alpha}\}$$

$$\parallel MAC_{K_r}\{id_{CH_\alpha} \parallel Lev \parallel E_{K_r}\{id_{CM\_compromise}, REFRESH\_CMcomp, CPT, N_{CH_\alpha}\}\}$$

Étape 2. Chaque nœud CH recevant le message $REFRESH\_CMcomp$ efface l'identification du nœud CM compromis de la liste appropriée L_id_{CM} et rafraîchit la clé par paire avec le CH notifié avec:

$$K_{CH_\beta - CH_\alpha} = H_{K_r}^{CPT}(K_{CH_\beta - CH_\alpha}) \tag{4.9}$$

Étape 3. Chaque nœud CM (membre de CH émetteur), qui reçoit le message $REFRESH\_CMcomp$, efface l'identification et la clé du nœud CM compromet et rafraîchit les clés avec:

$$K_{CM_j - CH_\alpha} = H_{K_r}^{CPT}(K_{CM_j - CH_\alpha} \parallel id_{CM_{compromise}}) \tag{4.10}$$

Le message de réponse de rafraîchissement est diffusé par le nœud CM:

$$CM_j \rightarrow *: id_{CM_j} \parallel id_{CH_\alpha} \parallel E_{K_r}\left\{REFRESH\_CMcomp\_REP, CPT, N_{CM_j}\right\}$$

$$\parallel MAC_{K_r}\left\{id_{CM_j} \parallel id_{CH_\alpha} \parallel E_{K_r}\left\{REFRESH\_CMcomp\_REP, CPT, N_{CM_j}\right\}\right\}$$

Étape 4. Comme pour l'étape 6 du cas précédent, le message $REFRESH\_CMcomp\_REP$ reçu est utilisé pour vérifier l'authentification des autres nœuds CM de leur cluster et rafraîchit les clés avec la formule (4.8).

Étape 5. Après avoir reçu le message $REFRESH\_CMcomp\_REP$ diffusé par les autres nœuds CM de leur cluster, le nœud CH vérifie l'authenticité et rafraîchit les clés précédentes avec la formule (4.10). Ensuite, les clés précédentes partagées avec les nœuds CH sont actualisées avec la formule (4.9). Dès que ces étapes ont été effectuées, le nœud CM compromis sera isolé, chaque clé partagée avec ce nœud sera effacée et toutes les clés nécessaires seront rafraîchies. La figure 4.5 présente un aperçu du processus de renouvellement de clés dans ce cas.

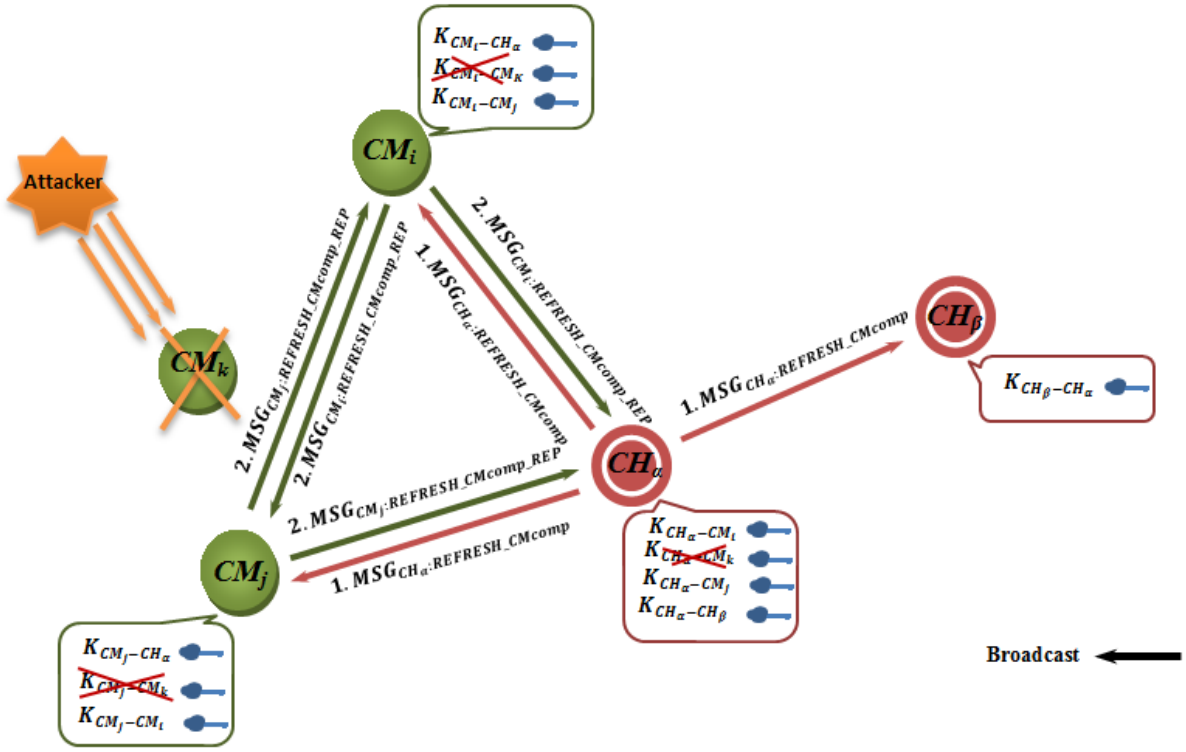

Figure 4.5. Le processus de renouvellement de clés pour un membre de cluster compromis (Comp_CH)

4.5.2.3. Processus de renouvellement de clés pour l'élection d'un nouveau cluster-head (ELEC)

Pour prolonger la durée de vie de l'ensemble du réseau, il est nécessaire de changer le cluster-head. Par conséquent, certaines clés doivent être renouvelées pour assurer la sécurité. Il est évident que l'absence d'un nœud compromis (nœud CH ou CM) réduit dans ce cas le niveau de sécurité requis. Le processus de renouvellement de clés est décrit comme suit:

Étape 1. Lorsque le nouveau cluster-head est élu, il lance le processus de renouvellement de clés en diffusant le message de rafraîchissement. Ce message contient l'ID du nœud CH précédent

$$CH_{New} \rightarrow *: id_{CH_{New}} \parallel Lev \parallel E_{K_r}\{id_{CH\_previous}, REFRESH\_ELEC, CPT, N_{CH_{New}}\}$$

$$\parallel MAC_{K_r}\left\{id_{CH_{New}} \parallel Lev \parallel E_{K_r}\left\{id_{CH_{previous}}, REFRESH\_ELEC, CPT, N_{CH_{New}}\right\}\right\}$$

Étape 2. Chaque nœud CH, après avoir vérifié l'authenticité du message de demande de renouvellement de clés du nouveau cluster-head, remplace l'identification du nouveau nœud CH par l'identification du nœud CH précédent dans la liste L_id_{CM}. Il calcule ensuite la nouvelle clé partagée entre eux en utilisant la formule (4.6). Il envoie ensuite le message

$REFRESH\_ELEC\_REP$ au nouveau CH, qui contient l'identification de l'ensemble de ses membres L_id_{CM}.

$$CH_\alpha \rightarrow CH_{New}: id_{CH_\alpha} \parallel Lev \parallel E_{K_r}\{REFRESH\_ELEC\_REP, CPT, N_{CH_\alpha}, L\_id_{CM}\} \parallel MAC_{K_r}\left\{id_{CH_\alpha} \parallel Lev \parallel E_{K_r}\{REFRESH\_ELEC\_REP, CPT, N_{CH_\alpha}, L\_id_{CM}\}\right\}$$

Étape 3. Chaque nœud CM (membre de CH émetteur), qui reçoit le message $REFRESH\_ELEC$, remplace la clé partagée avec le nœud CH précédent par la clé suivante:

$$K_{CM_j-CH_{New}} = H_{K_r}^{CPT}\left(K_{CM_j-CM_l} \parallel id_{CH_{New}}\right) \quad (4.11)$$

Tels que $K_{CM_j-CM_l}$dans la formule est la clé précédente partagée entre le nœud CM_j et le nœud $CM_l = CH_{New}$

Un rafraîchissement de toutes les clés partagées est effectuée avec les autres nœuds CM de leur cluster et la clé partagée avec le CH précédent (revenu membre) avec la formule (4.8).

Étape 4. Après avoir reçu le message $REFRESH\_ELEC$, le CH précédent calcule la clé partagée avec le nouveau nœud CH avec la formule (4.11) et rafraîchit les clés précédentes partagées avec les nœuds CM avec la formule (4.8).

Étape 5. Après avoir reçu le message $REFRESH\_ELEC\_REP$ diffusé par les autres nœuds CH, le nouveau nœud CH calcule la clé avec chaque CH émetteur suivant la formule (4.6) et rafraîchit les clés précédentes partagées avec les nœuds CM avec la formule (4.11).

À la fin de ces étapes, le nouveau cluster-head rafraîchit non seulement la clé partagée avec ses membres de cluster, mais calcule également la clé partagée avec chaque nœud CH et enregistre la liste de ses membres. De l'autre côté, seuls les nœuds CM, avec le cluster affecté, renouvellent les clés nécessaires. Les mêmes tailles de message sont prises en compte pour la phase de renouvellement de clés. De plus, le paramètre CPT prend 1 octet dans une interaction de message.

Le processus de renouvellement de clés dans le cas de l'élection d'un nouveau CH est illustré à la figure 4.6.

Il convient de noter que tous les nœuds capteur rafraîchissent les clés K_r et K_{in} (formules (4.4) et (4.5)) pour utiliser les mêmes clés dans les autres cas de renouvellement. En effet, le calcul de ces nouvelles clés est lié au changement de la valeur du CPT. De plus, le nœud CH dominant (c'est-à-dire le nouveau nœud CH dans le renouvellement ELEC et le

renouvellement Comp_CH ou le nœud CH qui a détecté un nœud CM compromis dans le renouvellement Comp_CM) initié dans chaque cas de renouvellement de clés le Timer qui sera déclenché après T_{min}.

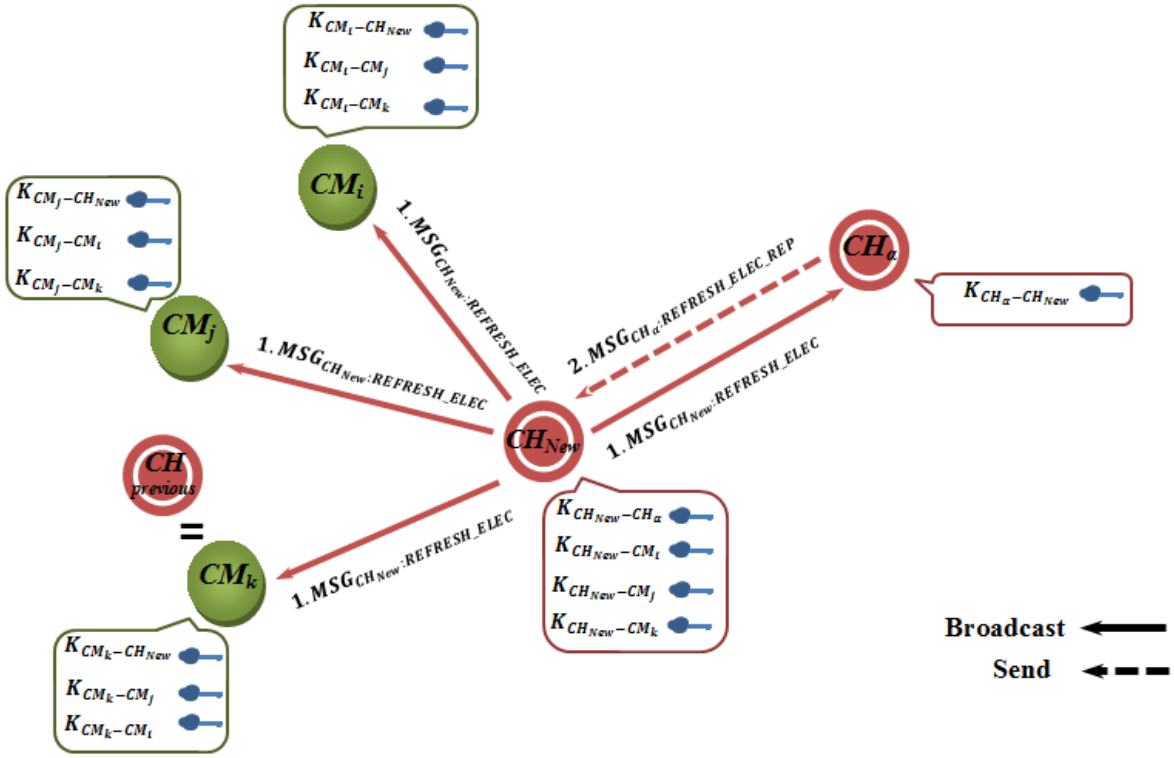

Figure 4.6. Le processus de renouvellement de clés pour une élection d'un nouveau cluster-head (ELEC)

4.5.3. Intégration des nouveaux nœuds capteurs

Pour ajouter un nouveau nœud de capteur CM_m dans le réseau, la SB génère des nouvelles clés K_r et K_{in}. Ces clés sont stockées dans la mémoire de CM_m. Avant déployer ce nœud, la SB envoie les nouvelles clés, ce qui permet à chaque nœud CH de les transmettre à ses voisins. Le processus détaillé est décrit comme suit:

Étape 1. Lorsqu'un nouveau nœud CM_m est déployé, il diffuse le message:

$$CM_m \rightarrow *: id_{CM_m} \parallel E_{K_r}\{JOIN, N_{CM_m}\} \parallel MAC_{K_r}\left\{id_{CM_m} \parallel E_{K_r}\{JOIN, N_{CM_m}\}\right\}$$

Pour être sûr que les données ne sont pas réinjectées par les échanges précédents, nous nous référons au nonce (N_{CM_m}).

Étape 2. Tout nœud CH situé dans le rayon de perception du nouveau nœud répond par le message suivant:

$$CH \rightarrow CM_m: id_{CH} \parallel Lev \parallel E_{K_r}\{JOIN\_REP, N_{CH}\} \parallel MAC_{K_r}\left\{id_{CH} \parallel Lev \parallel E_{K_r}\{JOIN\_REP, N_{CH}\}\right\}$$

Étape 3. Le nouveau nœud déclare la source du premier message reçu comme cluster-head, puis il diffuse le message:

$$CM_m \rightarrow *:$$

$$id_{CM_m} \parallel id_{CH} \parallel E_{K_r}\{JOIN\_REQ, N_{CM_m}\} \parallel MAC_{K_r}\left\{id_{CM_m} \parallel id_{CH} \parallel E_{K_r}\{JOIN\_REQ, N_{CM_m}\}\right\}$$

La clé par paire partagée avec leur cluster-head est calculée à l'aide de la formule (4.1)

Étape 4. Après avoir reçu le message $JOIN\_REQ$, le cluster-head du nouveau nœud vérifie l'authenticité, calcule la clé par paire partagée entre eux (formule (4.1)) et diffuse un message pour informer les autres nœuds CH, qui contient l'identification du nouveau nœud.

$$CH \rightarrow * :$$

$$id_{CH} \parallel Lev \parallel E_{K_r}\{id_{CM_m}, JOIN\_REQ, N_{CH}\} \parallel MAC_{K_r}\left\{id_{CH} \parallel Lev \parallel E_{K_r}\{id_{CM_m}, JOIN\_REQ, N_{CH}\}\right\}$$

Étape 5. Chaque nœud CM (dans le même cluster du nouveau nœud), qui reçoit le message $JOIN\_REQ$ du nouveau nœud, vérifie l'authenticité, calcule la clé par paire partagée entre eux (formule (4.2)) et envoie un message au nouveau nœud, qui contient leur identification et le nonce pour calculer la clé partagée.

$$CM_j \rightarrow CM_m:$$

$$id_{CM_j} \parallel id_{CH} \parallel E_{K_r}\left\{JOIN\_REQ\_REP, N_{CM_j}\right\} \parallel MAC_{K_r}\left\{id_{CM_j} \parallel id_{CH} \parallel E_{K_r}\left\{JOIN\_REQ\_REP, N_{CM_j}\right\}\right\}$$

Étape 6. Après avoir vérifié l'authenticité du message $JOIN\_REQ\_REP$ de chaque nœud CM, le nouveau nœud CM identifie l'ensemble des capteurs qui se trouvent dans le même cluster et calcule la clé partagée entre eux (formule (4.2)).

Notez que lorsque la station de base génère des nouvelles clés K_r et K_{in}, la valeur CPT est initialisée à 0.

4.6. Analyse des performances théoriques et de la sécurité

Dans cette section, nous présentons une analyse comparative de SKWN par rapport à cinq schémas existants: LEAP + [98], l'un des premiers schémas qui permet de prendre avantage de la topologie hiérarchique des RCSF, EDDK [99], qui offre des meilleures performances que

LEAP+. SEHKM [100], EAHKM [101] et KMP [103], les trois approches les plus récentes qui peuvent prendre avantage du clustering.

4.6.1. L'Overhead

Nous nous appuyons sur deux critères d'analyse des performances, à savoir le coût de calcul et le coût de stockage. Une analyse théorique des coûts liés au calcul et au stockage est présentée dans le tableau 4.2.

En ce qui concerne les systèmes LEAP + et EDDK, SKWN est plus performant en termes de calcul et de stockage. En effet, dans ces approches, un nœud doit non seulement traiter les messages provenant de tous ses voisins, mais également stocker les clés associées en mémoire. Dans notre cas, dû l'avantage de la mise en clusters, seuls les messages provenant des nœuds du même cluster seront traités.

Dans le cas d'EAHKM, nos coûts de calcul sont inférieurs à ceux de cette approche. En effet, dans cette approche, un nœud doit déchiffrer les messages de tous ses voisins. De l'autre côté, le coût de stockage de cette approche et de SKWN est très proche.

En ce qui concerne SEHKM, le coût de stockage de cette approche et de SKWN est très proche. De l'autre côté, nos coûts de calcul sont inférieurs à ceux de cette approche. En effet, cette approche repose sur l'algorithme Diffie-Hellman, qui nécessite un coût de calcul très élevé.

Enfin, KMP nécessite une capacité de stockage importante par rapport à SKWN et nécessite plus de communication en raison de la transmission de la liste d'index de clés.

4.6.2. Résilience contre la capture de nœud

Nous supposons qu'un adversaire peut augmenter le nombre des attaques physique sur les nœuds capteurs après le déploiement du réseau. Une fois qu'un nœud capteur est capturé, l'adversaire peut lire des informations secrètes de sa mémoire. Une telle attaque peut compromettre non seulement les liens adjacents de ceux compromis, mais également les liens externes indépendants des nœuds compromis. Ainsi, la sécurité fournie par les schémas de gestion de clés peut être mesurée sur la base de leur résilience à la capture des nœuds.

Cette métrique est calculée en estimant la fraction de la communication sécurisée totale qui est compromise par une capture de c nœuds n'incluant pas la communication dans laquelle les nœuds compromis sont directement impliqués.

En d'autres termes, nous voulons trouver la probabilité qu'une connexion sécurisée entre deux nœuds quelconques CM_i et CM_j, ne soit pas compromise lorsque c nœuds (différents de CM_i et CM_j) sont capturés.

Nous quantifions cette métrique avec les trois valeurs suivantes: (i) Résilience élevée: le nœud compromis ne peut pas affecter le réseau malgré le nombre des nœuds compromis dans le réseau. (ii) Résilience moyenne: le nœud compromis affecte moins des nœuds non compromis. (iii) Résilience faible: la compromission d'un nœud conduit à la destruction de l'ensemble du réseau.

Dans notre proposition, tous les nœuds CM communiquent directement avec le CH au sein du même cluster, des communications existent entre les nœuds CM dans le même cluster d'une part, et entre les nœuds CH d'une autre part. Chaque paire de nœuds partage une clé symétrique unique. Ainsi, la capture d'un membre ou d'un cluster-head n'a aucun impact sur les liens sécurisés entre des autres nœuds non compromis.

Utilisons $\boldsymbol{P_R}(c)$ pour représenter la probabilité de résilience d'un lien de communication entre deux nœuds lorsque c autres nœuds sont capturés. Cette probabilité est calculée comme suit:

$$\boldsymbol{P_R}(c) = 1 - \boldsymbol{P_c}(c)$$

Où $\boldsymbol{P_c}(c)$ indique la probabilité d'une communication totalement sécurisée compromise entre deux nœuds après la capture de c nœuds capteurs. Dans notre cas, comme expliqué précédemment, nous avons $\boldsymbol{P_c}(c) = 0$. De cette manière, nous avons montré que notre schéma proposé est inconditionnellement sécurisé contre la capture des nœuds et offre par conséquent une résilience élevée.

Une hypothèse critique est formulée par le schéma LEAP + en considérant qu'aucun nœud n'est capturé pendant la phase d'installation de clés. La résilience n'est faible que pendant cette phase. En conséquence, LEAP + offre une sécurité parfaite contre les attaques par capture de nœud si les nœuds ne sont pas capturés au cours de cette phase. Dans la phase d'installation, LEAP + disperse les dommages résultant de la divulgation de la clé initiale à un attaquant. La connaissance de cette clé par un attaquant permet d'établir facilement toutes les clés par paires du réseau.

Dans le schéma EDDK, chaque nœud capteur est pré-chargé par une clé initiale principale et une fonction pseudo-aléatoire partagée par tout le réseau. Chaque nœud capteur est capable de calculer sa clé individuelle en utilisant sa clé initiale et sa fonction pseudo-aléatoire. Ainsi, il peut calculer toutes les clés par paires partagées avec ses nœuds voisins. Une fois que le Timer

d'établissement de clés atteint sa valeur de seuil prédéfinie, un nœud supprime les informations secrètes telles que toutes les clés individuelles de ses voisins et la clé initiale pour améliorer la sécurité. Comme pour LEAP +, EDDK a une résilience élevée du fait que les clés par paire sont décentralisées et que la compromettre d'un nœud capteur n'affecte pas les autres liaisons de communication.

Dans le schéma KMP, comme il fait utilise un pool de clés de taille S et que chaque nœud contient m clés partielles sélectionnées dans le pool de clés par la SB, la probabilité de compromettre une liaison sécurisée entre deux nœuds non compromis lorsque on a un nombre c de nœuds qui n'a pas été compromis est

$$p(not\_compromised) = 1 - c \times \frac{P_{m,2}}{P_{S,2}}$$

Où $P_{m,2}$ est le nombre de clés complètes pouvant être établies (P est la fonction de permutation).

Dans EAHKM, tous les nœuds capteurs communiquent directement avec le cluster-head et aucune communication n'existe entre les nœuds capteurs appartenant au même cluster. De plus, chaque nœud capteur du réseau stocke une clé de cluster et une clé par paire employée pour sécuriser la communication avec son CH. À l'intérieur d'un cluster, tous les nœuds dépendent de la même clé de cluster. Si un nœud compromis est un membre du cluster, tous les membres peuvent être compromis lorsque la clé du cluster est divulguée. La capture d'un nœud membre a un impact sur les liens sécurisés entre des autres nœuds non compromis (c'est-à-dire entre n'importe quel nœud membre et le cluster-head). Cependant, cela n'affecte pas les liens de communication liés aux autres clusters. Dans EAHKM, si un nœud compromis est CH, les liaisons de communication affectées sont uniquement liées au même cluster.

Dans SEHKM, trois types de clés sont fournis pour chiffrer les messages: 1) Une clé de réseau partagée par tous les nœuds du réseau, elle est utilisée par SB pour chiffrer les messages de diffusion et d'authentification. 2) Une clé de groupe partagée par tous les nœuds du même groupe (un cluster peut être un groupe), elle sert à sécuriser les messages diffusés dans un groupe. 3) une clé par paire, partagée par une paire de nœuds spécifique. La clé par paire sécurise les messages en monodiffusion (unicast) et peut être utilisée pour l'authentification. Dans ce schéma, la capture de n'importe quel nœud CM peut compromettre la totalité des nœuds CM. Il disperse les dommages résultant de la divulgation de la clé de groupe. Également, si un nœud compromis est un CH, tous les CH du réseau peuvent être compromis et ceci si la clé de groupe est divulguée.

Tableau 4.2 Comparaison des performances de notre schéma (SKWN) avec d'autres schémas [113]

Schéma	Le type de nœud	Le coût de calcul	Le coût de stockage	Le coût de communication	Resilience
LEAP+ [98]		$d.Enc + d.MAC + (3d + 1).PRF$	$3d + 2 + la\ chaine\ de\ clés$	$2d + 1$	élevée
EDDK [99]		$1.Enc + d.Dec + (2d + 1).PRF + 1.MAC$	$2 + tableau\ de\ voisins$	$d + 1$	élevée
SEHKM [100]	nœud CH	$(n + 6).Enc + (n + 4).Dec + (n + 1).PRF + (n + 1).DH$	$n + 4$	$2n + 10$	moyenne
	nœud CM	$1.Enc + 4.Dec + 2.PRF + 1.DH$	4	5	
	nœud Assistant	$5.Dec + (n - 1).Enc + 1.DH + (n + 1).PRF$	$n + 3$	$n + 4$	
EAHKM [101]	nœud CH	$(d + 1).Dec + (n + 1).Enc + 1.MAC + n.H$	$n + 4$	$n + d + 2$	moyenne
	nœud CM	$(d + 1).Dec + 1.Enc + 1.MAC + 1.H$	5	$d + 2$	
KMP [103]	nœud CH	$(1 + m.x + q.x.n).Dec + (1 + m.x.n + q.x.n).Enc$	$pool\ de\ clés + la\ liste\ d'index\ des\ clés + 1 + 2.n\ (la\ liste\ d'ordres\ des\ numéros\ index)$	$2 + 2.q.x.n + (1 + m.x).n$	moyenne
	nœud CM	$(m.x + q.x).Dec + (1 + q.x).Enc$	$pool\ de\ clés + la\ liste\ d'index\ des\ clés + 1 + 2\ (la\ liste\ d'ordres\ des\ numéros\ d'\ index)$	$1 + 2.q.x + m.x$	
SKWN (notre schéma)	nœud CH	$(n + C - 1).Dec + 2.Enc + 2.MAC + (C + n - 1).H$	$n + C + 1$	$n + C + 1$	élevée
	nœud CM	$n.Dec + 1.Enc + 1.MAC + n.H$	$n + 2$	$n + 1$	

d: le nombre de voisins. **n**: le nombre de membres de cluster. **C**: le nombre de clusters. **Enc**: la fonction de cryptage. **Dec:** la fonction de décryptage. **MAC:** le code d'authentification de message. **PRF:** la fonction pseudo aléatoire. **DH:** l'algorithme de Diffie-Hellman. **H:** la fonction de hachage. **m:** le nombre des clés partielles. **q:** le nombre d'index dans la liste d'ordres. $x = index\_taille\ /packet\_taille$. **index_taille**: le nombre de bits requis pour identifier chaque clé partielle. **packet _taille:** la taille(en bits) de données échangées dans le paquet.

Pour la résilience contre la capture de nœud, nous soulignons que notre schéma, LEAP + et EDDK fournissent une résilience parfaite (élevée). Toutefois, dans KMP, EAHKM et SEHKM, lorsque le nombre de nœuds compromettants augmente, la fraction du total des communications réseau compromises augmente. Par conséquent, la résilience de ces schémas est moyenne.

4.7. Simulation et résultats

Dans cette section, nous présentons l'évaluation des performances de SKWN à travers un ensemble d'expériences. En effet, nous proposons des simulations approfondies pour vérifier les métriques de performances telles que le coût de communication, le coût de stockage et la consommation d'énergie. Dans le premier ensemble de simulations, nous comparons SKWN avec EDDK [98] et KMP [103], qui servent également de base afin de faire des comparaisons avec d'autres schémas tels que LEAP +[99], SEHKM[100], et EAHKM [101].

D'autre part, le deuxième ensemble de simulations est consacré à l'évaluation de la performance de différents processus de renouvellement de clés.

À cette fin, nous avons implémenté SKWN en utilisant le langage de programmation NesC [11] afin de l'intégrer à TinyOS. Les simulations sont effectuées à l'aide de l'environnement TOSSIM [114]. Nous avons également utilisé TinySec, qui est implémenté dans TinyOS, en tant que bibliothèque cryptographique. Dans nos simulations, nous utilisons plusieurs réseaux d'une taille variant de 30 à 150 nœuds de type MICA2. Parmi ces nœuds, 10% des nœuds sont cluster-head (CH). Les nœuds sont répartis uniformément et de manière aléatoire sur une surface de 150 × 150 m. La portée de transmission d'un capteur est de 22 m, la taille d'un paquet est de 31 octets et le taux d'erreur de transmission est de 0.

4.7.1. Le coût de communication

En ce qui concerne EDDK, dans l'étape d'établissement de clés, chaque nœud capteur envoie un message et reçoit autant de messages d'après le nombre de ses voisins. Le coût de communication dans KMP dépend linéairement du nombre d'index des clés partielles (m) et du nombre d'index dans la liste ordonnée (q) crée par le CH ou le CM.

Chaque nœud membre envoie un message à son CH, reçoit $m \times x$ messages permettant d'échanger la liste d'index de clés du CH et reçoit $q \times x$ messages permettant d'échanger la liste d'ordres crée par son CH pour déterminer les clés communes entre eux.

En réponse, chaque nœud membre envoie une liste d'ordres unique, qui doit envoyer $q \times x$ messages. Après chaque CH nœud reçoit la liste d'index des clés de la SB, il envoie environ $m \times x$ messages et environ $q \times x$ messages pour transmettre une liste d'ordres unique à chaque nœud membre. Il reçoit également les listes d'ordres de leurs nœuds membres. Ainsi, pour un cluster comprenant n nœuds membres, $n \times q \times x$ messages au total sont reçus.

De l'autre côté, dans SKWN, pour un réseau de N nœuds comportant C clusters de n membres, chaque CH diffuse un message, reçoit n messages de ses membres et $(C-1)$ messages des autres nœuds CH, puis diffuse un message contenant la liste d'identification de l'ensemble des nœuds membres. Chaque nœud CM reçoit également un message de leur CH, diffuse un message en réponse, puis reçoit $(n-1)$ messages des autres membres du même cluster.

Comme la montre la figure 4.7, comme le nombre des nœuds augmente, le coût de communication augmente pour EDDK. Pour KMP, ce coût n'est pas affectée par l'évolution du nombre des nœuds et ceci pour CH et CM. Alors que dans SKWN, les nœuds CH nécessitent moins de coût de communication. Comparé au CH, le nœud CM nécessite moins de coût de communication et a une valeur fixe par rapport à N.

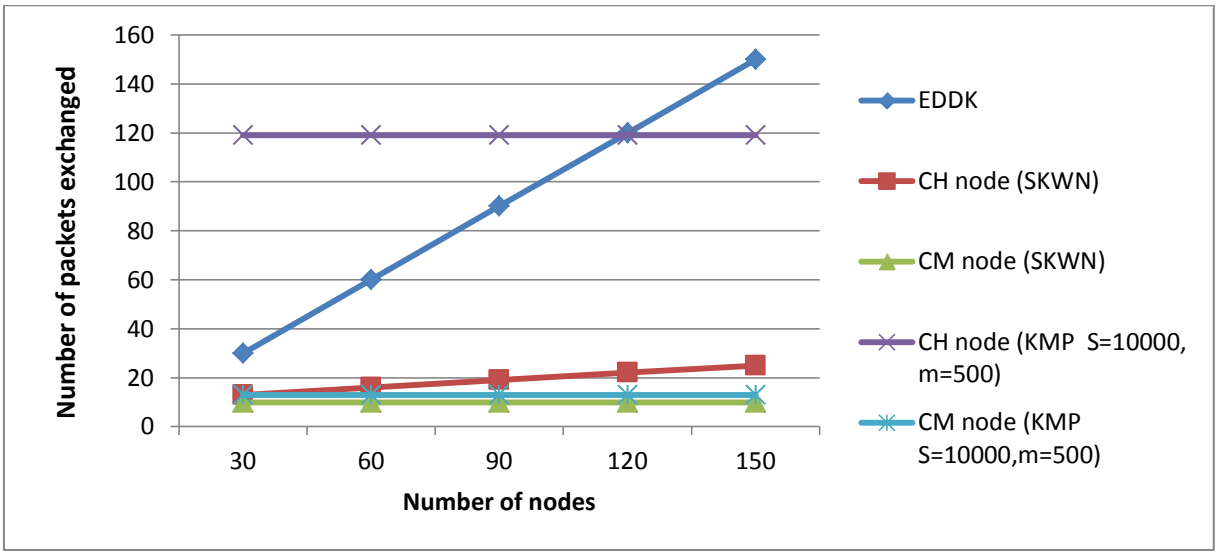

Figure 4.7 Comparaison du nombre de paquets échangés.

4.7.2. Le coût de stockage

La figure 4.8 présente le totale de stockage requise par rapport à la taille du réseau. À partir de cette figure, on peut constater qu'EDDK et KMP utilisent plus de mémoire que SKWN. En effet, pour EDDK, le nombre de clés stockées est lié au nombre de voisins.

Cependant, la mémoire de stockage requise par KMP est principalement liée aux besoins du nœud capteur afin de stocker les m clés partielles (de 64 bits chacune), la liste d'index (de clés partielles), la clé de réseau (doit être de 128 bits de longueur), les deux listes index ordonnés de q clés partielles (la première est créé par le nœud membre et la seconde est envoyé par leur CH). En cas de CH, 2n listes ordonnée est stockée. Pour un pool de clés contenant S clés partielles, $log_2 S$ bits sont requis pour chaque index utilisé.

Dans notre schéma, chaque nœud de capteur ne doit stocker que deux clés dans sa mémoire avant le déploiement. Après le déploiement, chaque nœud CH est pré-chargé avec $(C-1)$ clés partagées avec les autres nœuds CH et n clés partagées avec ses nœuds CM. Lorsque la taille du réseau augmente, l'espace mémoire total augmente linéairement pour le nœud CH. En effet, le nœud CM utilise moins de mémoire pour stocker les clés. Il suffit de stocker n clés.

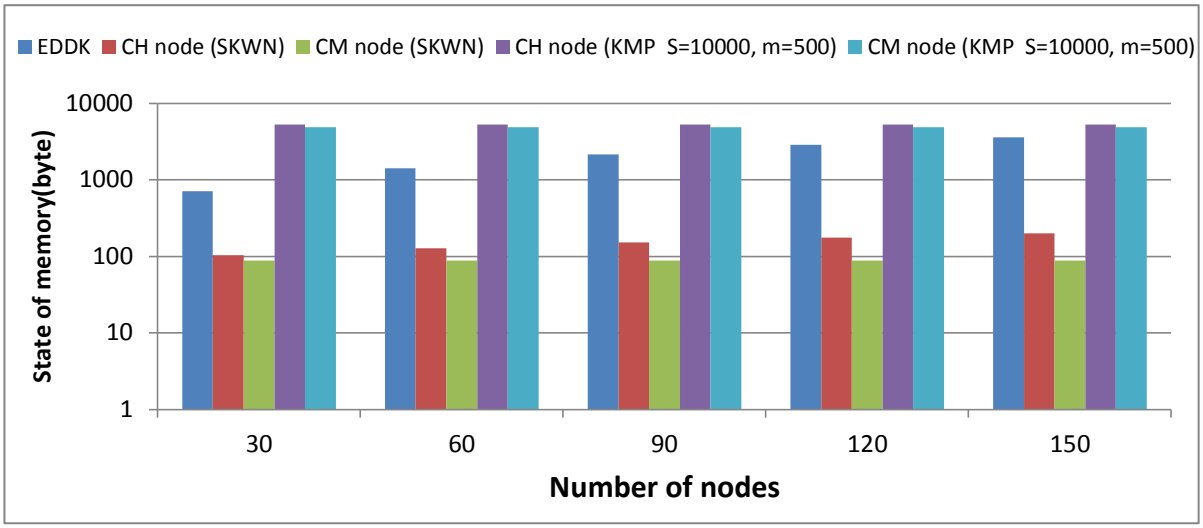

Figure 4.8 La consommation de mémoire

4.7.3. La consommation d'énergie

La consommation d'énergie est un paramètre important pour tout schéma d'établissement de clés. Par conséquent, nous avons utilisé le plugin PowerTOSSIM dans TinyViz pour analyser l'énergie des trois approches. La figure 4.9 présente l'énergie totale consommée par SKWN, EDDK et KMP avec respect plusieurs tailles de réseau.

Il est évident que le schéma proposé nécessite moins de consommation d'énergie par rapport à EDDK et KMP. En effet, dans SKWN, le nœud CM (et le nœud CH) échange moins de paquets, et la taille des messages échangés pour la construction des clés par paires est plus petite que celle d'EDDK. Cette figure montre également que l'énergie consommée par notre approche est négligeable par rapport à celle liée au KMP.

Comme illustré, notre approche plus performante que KMP de 91% à 94%. En comparant avec les résultats présentés par Messai et al dans [102], Qui montrent qu'EAHKM plus performante que KMP de 35% à 40%, nous pouvons en déduire que notre approche (SKWN) plus performante qu'EAHKM.

À travers les résultats des simulations, nous montrons que SKWN plus performante qu'EDDK de 17% à 25%. Comme Zhang et al [98] montrent qu'EDDK plus performante que LEAP de 80% et que Zhang et al [100] montrent que SEHKM plus performante que LEAP de 90% à 93%, nous pouvons conclure que notre approche plus performante à la fois de LEAP et de SEHKM. Par conséquent, nous pouvons conclure que notre approche est plus performante que KMP, LEAP, EDDK, SEHKM et EAHKM.

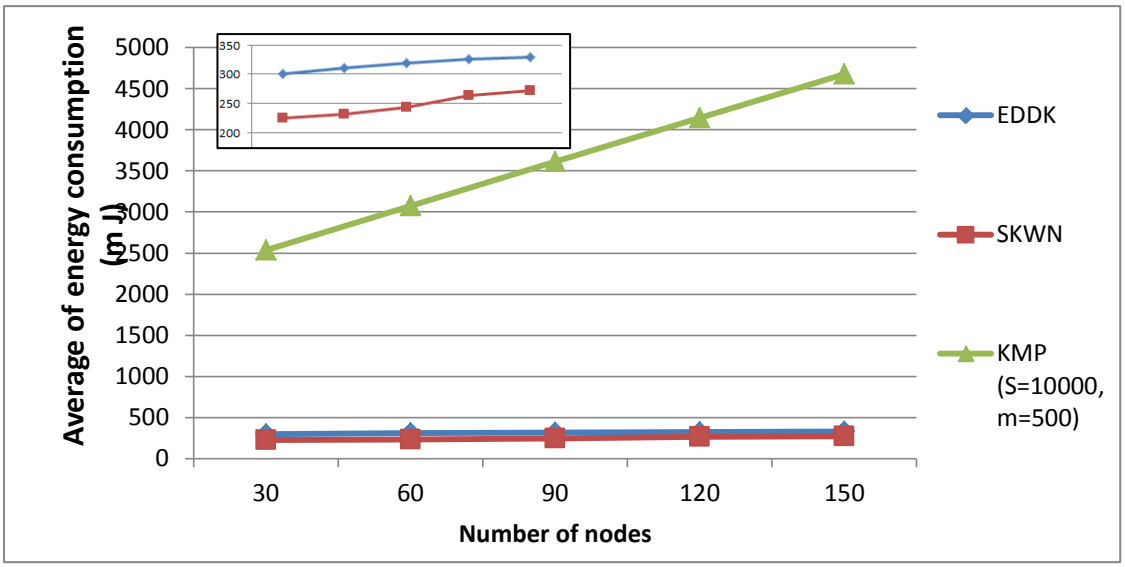

Figure 4.9 Moyenne de consommation d'énergie

4.7.4. Analyse et évaluation des mécanismes de renouvellement de clés

Afin de démontrer l'efficacité de l'utilisation de plusieurs niveaux de sécurité dans différents cas de renouvellement de clés, nous avons effectué des simulations sur un réseau composé de 150 nœuds. La valeur CPT est fixée à 2. Dans la phase de renouvellement de clés, pour les trois approches, nous avons considéré trois entités (c.-à-d. nœud CH dominant, nœud CH et nœud CM) dans le réseau avec des rôles différents. Nous avons évalué la consommation d'énergie des trois entités lors de l'exécution des algorithmes de renouvellement de clés, ce qui permet de montrer l'impact de l'utilisation de la fonction RC5 à différents niveaux. On peut voir que

1. Pour un nœud CM, le nombre de paquets échangés et le nombre d'utilisation de la fonction RC5 sont les mêmes pour les deux approches Comp_CH et Comp_CM. De plus, ce nombre ne dépend pas de la taille du réseau. La figure 4.10 montre la variation de l'énergie consommée par un nœud CM pour les trois approches. On peut voir que l'approche Comp_CM permet un gain de consommation d'énergie significatif par rapport à l'approche Comp_CH. Cela est dû aux tours utilisés dans la fonction RC5. De l'autre côté, le nœud CM dans l'approche ELEC génère moins de paquets et fait moins des appels à la fonction RC5 comparé aux autres méthodes. Cette approche permet aux nœuds de consommer moins d'énergie.
2. Pour un nœud CH, les deux approches ELEC et Comp_CH ont non seulement le même nombre de paquets échangés, mais également le même nombre des appels de la fonction RC5. La figure 4.10 montre que l'approche ELEC consomme moins d'énergie. En effet, cela est dû au fait que son tours utilisé dans la fonction RC5 est inférieur à celui utilisé par Comp_CH. Cette figure montre également que l'approche Comp_CM consomme moins d'énergie que Comp_CH et ne provient pas seulement des tours de la fonction RC5 mais également du nombre réduit de paquets échangés et du nombre des appels de la fonction RC5.
3. Pour le nœud CH dans Comp_CM et le nouveau CH choisi pour Comp_CH et ELEC, le nœud dominant (c.-à-d. le nœud CH) fournit le niveau de sécurité approprié dans chaque cas. Pour ce nœud, nous ne pouvons pas montrer l'influence de différentes utilisations de tours pour chaque approche. En effet, chaque méthode a un nombre différent de paquets échangés et un nombre différent d'utilisation de la fonction RC5. Pour cette raison, nous avons également fait varier le nombre de nœuds de 30 à 150. La figure 4.11 illustre la consommation d'énergie de chaque méthode pour différentes tailles de réseau. Il est évident que l'approche ELEC nécessite moins de consommation d'énergie par rapport à l'approche Comp_CH. De l'autre côté, l'approche Comp_CM nécessite moins d'énergie par rapport aux approches Comp_CH et ELEC. En effet, le nœud CM (ou le nœud CH) échange moins de paquets dans le processus de renouvellement de clés.

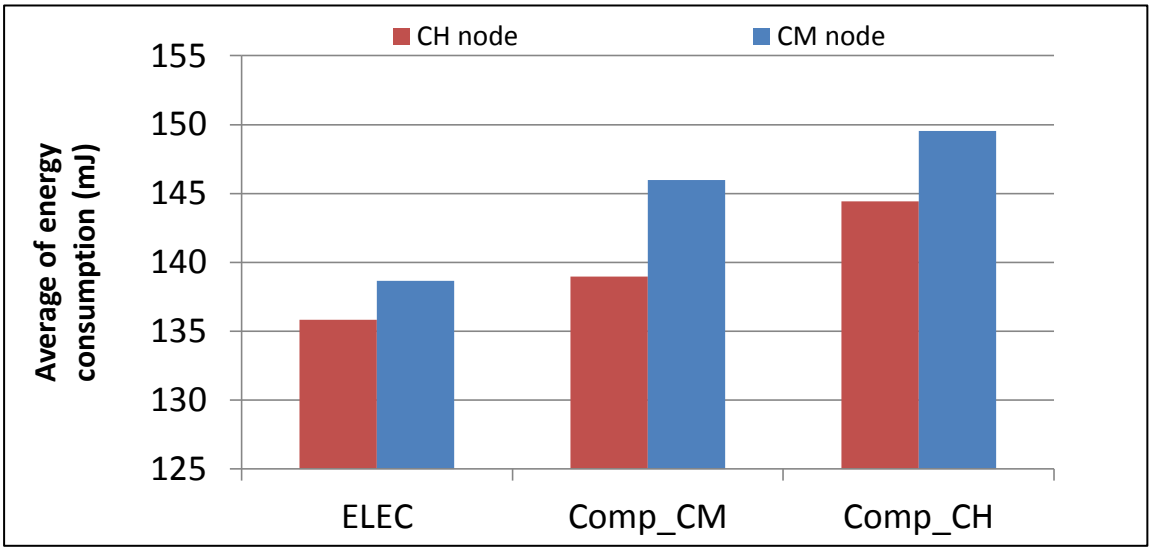

Figure 4.10 Moyenne de la consommation d'énergie dans les schémas de renouvellement de clés

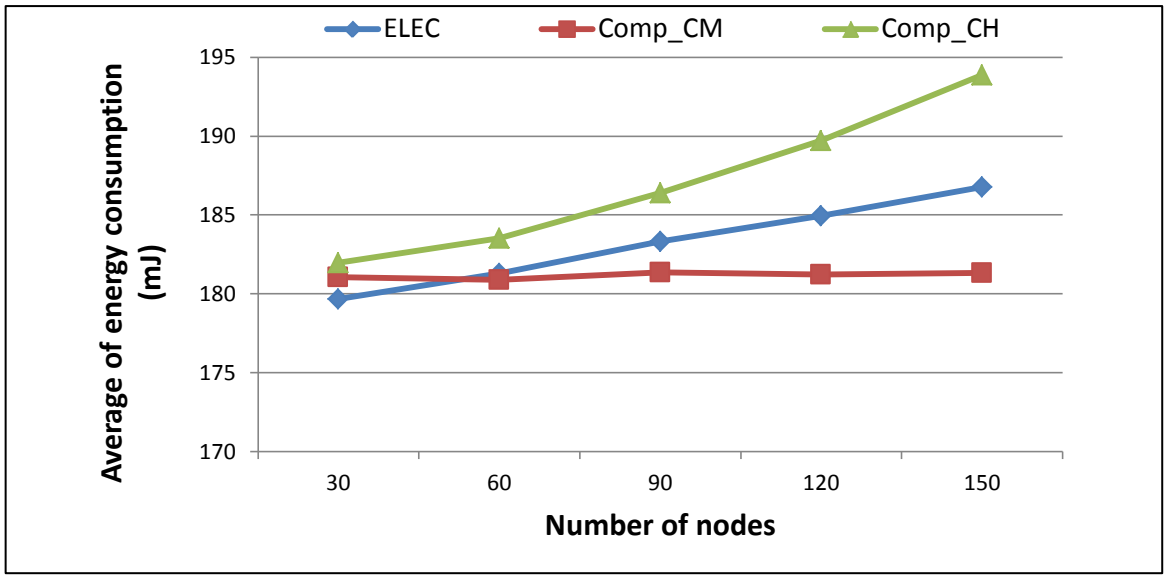

Figure 4.11 Moyenne de la consommation d'énergie par CH dominant dans les schémas de renouvellement de clés

4.8. Conclusion

Dans ce chapitre, nous avons proposé un schéma de gestion de clés intelligent et dynamique pour les réseaux de capteurs sans fil hiérarchiques. Notre schéma propose trois sous-schémas pour l'établissement, le renouvellement de clés et l'intégration des nouveaux nœuds, ce qui permet de garantir la scalabilité et la flexibilité du réseau. Nous avons montré comment notre approche est scalable et permet de prendre en compte différents types d'intrusion. L'approche

proposée appelée SKWN (**S**mart and dynamic **K**ey management scheme for hierarchical **W**ireless sensor **N**etworks) permet de rendre la décision dynamique en temps réel basée sur la partie ciblée du réseau. En effet, SKWN s'appuie sur le composant ISA, qui implémente certaines règles d'auto-apprentissage, pour adapter le niveau de sécurité en fonction des besoins de l'application. Ce composant permet par conséquent d'assurer une consommation d'énergie efficace et l'utilisation non redondante des opérations de sécurité. Par comparaison avec les schémas existants, SKWN fournit non seulement des mécanismes de sécurité fiables, mais optimise également la consommation d'énergie et les coûts liés à la communication et à l'utilisation de la mémoire. Nous avons validé notre proposition en fournissant une étude théorique et une étude expérimentale, ce qui permet une analyse approfondie de notre schéma. Par des simulations, nous avons montré également qu'une consommation d'énergie efficace pouvait être atteinte en utilisant un niveau de sécurité variable pour chaque scénario de renouvellement de clés.

Conclusion générale

À l'ère de l'Internet des objets (IoT), nous assistons à une production de données sans précédent en raison du déploiement massif des réseaux de capteurs sans fil (RCSF). Ces derniers peuvent être utilisés dans diverses applications critiques liées à des domaines variant du médical jusqu'au militaire. Ces applications ont de fortes exigences en matière de sécurité et ce afin de les protéger contre des attaques qui exploitent non seulement le déploiement des capteurs dans des zones généralement hostiles et sans surveillance mais aussi la transmission des données par voie hertzienne. Par conséquent, assurer la sécurité des échanges des données au sein des RCSF est une tâche importante et en même temps difficile. En effet, les capteurs qui les constituent les RCSF sont limités en termes de ressources énergétiques et de capacités physiques. De ce fait beaucoup de recherches ont été consacré à cette problématique en proposant des mécanismes de sécurité adaptés aux nœuds capteurs. Cependant, nous avons pu constater que la gestion de clés constitue la pierre angulaire des autres mécanismes de sécurité, qui s'appuient en général sur le cryptage.

Dans cette thèse, notre objectif a été de proposer une solution efficace au problème de la gestion de clés dans les RCSF. Pour cela, nous avons commencé par étudier profondément les schémas existants et avons montré leurs limites. Enfin nous avons proposé des solutions plus adaptées à l'environnement des RCSF. D'ailleurs, nous nous sommes fixés comme objectif la proposition d'un protocole de gestion de clés fiable, scalable et économe en énergie. D'autre part, ce protocole est intelligent en adaptant dynamiquement ses routines à l'environnement de déploiement de RCSF.

Dans le cadre de cette thèse, nous avons proposé un protocole de gestion de clés intelligent et dynamique pour les réseaux de capteurs sans fil hiérarchiques nommé SKWN. Ce dernier repose sur trois sous-schémas pour l'établissement, le renouvellement de clés et l'intégration des nouveaux nœuds. Notre proposition permet d'assurer la scalabilité et la flexibilité du réseau et permet aussi de prendre en compte différents types d'intrusion.

SKWN permet de prendre des décisions dynamiquement et en temps réel en se basant sur la partie ciblée du réseau. En effet, SKWN repose sur le composant ISA, qui implémente certaines règles d'auto-apprentissage, afin d'adapter le niveau de sécurité en fonction des

besoins de l'application. Ce composant permet par conséquent d'assurer une consommation d'énergie efficace et l'utilisation non redondante des opérations de sécurité.
Par rapport aux schémas existants et connus dans la littérature, SKWN fournit non seulement des mécanismes de sécurité fiables, mais optimise également la consommation d'énergie et les coûts liés à la communication et à l'utilisation de la mémoire.
Les performances de notre schéma ont été évaluées en fournissant une étude théorique et une étude expérimentale, ce qui a permis de fournir une analyse approfondie de notre schéma.
Dans le premier ensemble de simulations. Nous avons présenté une comparaison de notre sous-schéma lié à l'établissement de clés par rapport aux approches existantes. Ces simulations ont également confirmé que notre approche garantit la scalabilité.
D'autre part, le deuxième ensemble de simulations a permis l'évaluation de la performance de différents processus de renouvellement de clés. Nous avons montré également qu'une consommation d'énergie efficace pouvait être atteinte en utilisant un niveau de sécurité variable pour chaque scénario de renouvellement de clés.

- **Perspectives**

Les travaux présents au cours de la thèse traitent les problèmes liés à la gestion de clés dans les RCSF. Par ailleurs, nombreuses perspectives peuvent être envisagées afin d'améliorer plus ces travaux.
Il est clair que les travaux effectués ont montré que notre système de gestion de clés était efficace dans les environnements RCSF statiques. Cependant, il existe des applications liées aux RCSF avec des périphériques mobiles comme par exemple dans les océans, qui requièrent le même niveau de sécurité. Par conséquence, notre système devrait être fiable et optimisé les métriques de performances pour le réseau RCSF mobile. Les considérations à prendre en compte incluent l'amélioration de l'efficacité du système. Nous prévoyons principalement d'introduire l'impact de la localisation sur la gestion de clés. En effet, avec une grande mobilité des nœuds de capteurs, nous devons assurer la génération et l'expiration rapides des clés lorsque les nœuds se déplacent à l'intérieur et à l'extérieur de la portée de communication.

Aussi, nous avons pu montrer l'efficacité de l'optimisation de la consommation énergétique lorsqu'on combine un mécanisme de gestion de clés avec une topologie hiérarchique pré-établie dans le réseau RCSF. Cependant, nous avons également montré l'influence du nœud cluster-head dans le cas où il est compromis. Une des solutions pour faire disparaître ce type de problème semble donc de créer un mécanisme pour sécuriser la formation des clusters et

assurer la gestion de clés dans la même phase. Ainsi, en plus d'un établissement de clés sécurisé, cette solution nous permet de faire confiance aux cluster-heads élus.

Bibliographie

[1] Q. Monnet, '' Modèles et mécanismes pour la protection contre les attaques par déni de service dans les réseaux de capteurs sans fil '', thèse de doctorat en informatique, université de Paris-Est, École doctorale MSTIC, France, 2015.

[2] I. F. Akyildiz, Su. Weilian, Y. Sankarasubramaniam, and E. Cayirci, ''A Survey on Sensor Networks'', Published online in Wiley Online Library, pp.102–114, 2002.

[3] https://robotics.eecs.berkeley.edu/~pister/SmartDust/ [En ligne; accédé Avril 2017].

[4] http://www.snm.ethz.ch/snmwiki/Projects/Mica2 [En ligne; accédé Avril 2017].

[5] S. Sudevalayam, P. Kulkarn, "Energy Harvesting Sensor Nodes: Survey and Implications", IEEE Communications Surveys & Tutorials, vol. 13(3), pp. 443-461, September 2011.

[6] http://www.snm.ethz.ch/ [En ligne; accédé Avril 2017].

[7] https://www.iot-lab.info/hardware/wsn430/ [En ligne; accédé Avril 2017].

[8] J. Hill, R. Szewczyk, A. Woo, et al, "System Architecture Directions for Networked Sensors", ACM SIGOPS, vol. 35(11), pp. 93-104, November 2000.

[9] A. Dunkels, B. Gronvall and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors", In Proceedings of 29th IEEE International Conference on Local Computer Networks (LCN 2004), pp. 455-462, Washington, USA, November 2004,

[10] F. Khadar, " Contrôle de la topologie dans les réseaux de capteurs : de la théorie à la pratique", thèse de doctorat en informatique, université de Lille 1, France, 2009.

[11] D. Gay, P. Levis, R.V. Behren, M. Melsh, E. Brewer, and D. Culler, "The nesC Language: A Holistic Approach to Networked Embedded Systems", In Proceedings of the ACM SIGPLAN conference on Programming language design and implementation (PLDI '03), pp. 1-11, San Diego, California, USA, June 2003.

[12] S. Harchi, " Un protocole de session dans les réseaux de capteurs sans fils", thèse de doctorat en Automatique, Traitement du Signal et des Images, Génie Informatique, université de Lorraine, France, 2013.

[13] M. GAYE, "Etat de l'art sur les WSN (Wireless Sensor Network)", université Cheikh Anta Diop de Dakar, Juin 2014.

[14] C.T. Kone, "Conception de l'architecture d'un réseau de capteurs sans fil de grande dimension", thèse de doctorat en Automatique, Traitement du Signal et des Images, Génie Informatique, université de Henri Poincaré, Nancy I, France, 2011.

[15] M. LEHSAINI, "Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique", thèse de doctorat en informatique, université de Tlemcen, Algérie, 2009.

[16] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks". IEEE Transactions on Wireless Communications, vol. 1(4), pp. 660-670, 2002.

[17] O. Younis and S. Fahmy, "HEED: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks", IEEE Transactions on Mobile Computing, vol. 3(4), pp. 366–379, 2004.

[18] A. Manjeshwar and D. P. Agrawal, " TEEN : a routing protocol for enhanced efficiency in wireless sensor networks", In Proceedings of the 15th International Parallel and Distributed Processing Symposium (IPDPS01), pp. 2009–2015, San Francisco, USA, 2001.

[19] A. Manjeshwar and D. P. Agrawal, "APTEEN: A hybrid protocol for efficient routing and comprehensive information retrieval in wireless sensor networks", In Proceedings of the 16th International Parallel and Distributed Processing Symposium (IPDPS02), pp.195–202, Fort Lauderdale, Florida, USA, 2002.
[20] A. Darif, "Contributions à l'amélioration des performances des Réseaux de Capteurs sans fil à base d'IR-UWB ", thèse de doctorat en informatique et télécommunications, université de Mohammed V, Maroc, 2009.
[21] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, "Protocols for self-organization of a Wireless Sensor Network", IEEE Personal communications, vol. 7(5), pp.16–27, 2000.
[22] W.R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences, January 2000.
[23] N. Kushalnagar, G. Montenegro, and C. Schumacher, "RFC 4919: IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals", Network Working Group, Request for Comments, August 2007.
[24] T. Braun, T. Voigt, and A. Dunkels, "TCP support for sensor networks", In Proceedings of the Fourth Annual Conference on Wireless on Demand Network Systems and Services (WONS 2007), pp. 162-169, Obergurgl, Austria, January 2007.
[25] D. Dessales, "Conception d'un réseau de capteurs sans fil, faible consommation, dédié au diagnostic in-situ des performances des bâtiments en exploitation", thèse de doctorat en Optoélectronique et micro-ondes, université de Poitiers, France, 2009.
[26] V. Srivastava and M. Motani, "Cross-Layer Design: A Survey and the Road Ahead", IEEE Communications Magazine, vol. 43(12), pp.112-119, 2005.
[27] V. T. Raisinghani and S. Iyer, "Cross layer design optimizations in wireless protocol stacks", Computer Communications, vol. 27(8), pp.720-725, Mai 2004.
[28] R. Winter, J. Schiller, N. Nikaein, and C. Bonnet, "Crosstalk: Cross-layer decision support based on global knowledge", IEEE Communications Magazine, vol. 44, pp. 2-8, January 2006.
[29] V. T. Raisinghani and S. Lyer, "Eclair : An efficient cross layer architecture for wireless protocol stacks", In Proceedings of 5th World Wireless Congress, San Francisco, USA, Mai 2004.
[30] D.E. Boubiche, "Une approche Inter-Couches (cross-layer) pour la Sécurité dans les R.C.S.F ", thèse de doctorat de sciences en informatique, université de Batna, Algérie, 2013.
[31] A. Salhieh, J. Weinmann, M. Kochhal, and L. Schwiebert, "Power Efficient Topologies for Wireless Sensor Networks", International Conference on Parallel Processing, Valencia, Spain, September 2001.
[32] R. Kacimi, "Techniques de conservation d'énergie pour les réseaux de capteurs sans fil ", thèse de doctorat en réseaux et télécommunications, université de Toulouse, France, 2009.
[33] A. Berrachedi, et A. Diarbakirli, " Sécurisation du protocole de routage hiérarchique LEACH dans les réseaux de capteurs sans fil", mémoire de fin d'étude pour l'obtention du diplôme d'ingénieur d'état en informatique, Ecole nationale Supérieure d'Informatique (E.S.I), Algérie, Juin 2009.
[34] I. Amadou, G. Chelius, F. Valois, "PFMAC: Routage sans connaissance du voisinage efficace en énergie", CFIP 2011 – Colloque Francophone sur l'ingénierie des protocoles, Sainte Maxime, France, 2011.
[35] M. Ilyas and I. Mahgoub. "Handbook of sensor networks Compact wireless and wired Sensing Systems", ISBN 08493196864. CRC PRESS LLS, USA, 2005.

[36] http://www.factorysystemes.fr/solutions/m2m/experiences-clients/smart-cities-detection-incendies.html [En ligne; accédé Novembre 2017].
[37] https://www.thediabetescouncil.com/continuous-glucose-monitoring-everything-you-need-to-know/ [En ligne; accédé Novembre 2017].
[38] G.E. Rolader, J. Rogers, and J. Batteh, "Self-healing minefield", In Proceedings of the International Society for Optics and Photonics, vol. 5441, pp.13-24, 2004.
[39] M. J. Brown, "Users Guide Developed for the JBREWS Project", Technical Report LA-UR-99-4676, Los Alamos National Laboratory of California University, 1999.
[40] https://www.lockheedmartin.com/content/dam/lockheed/data/isgs/documents/LM%20SPAN%20fact%20sheet.pdf [En ligne; accédé Novembre 2017].
[41] E. M. Petriu, N. D. Georganas, D. C. Petriu, D. Makrakis, and V.Z. Groza, "Sensor based information appliances", IEEE Instrumentation Measurement Magazine, vol.3(4), pp.31-35, 2000.
[42] http://ipadwiki.com/what-is-smart-home-automation-designer-homes-perth/[En ligne; accédé Novembre 2017].
[43] A. J. Menezes, S. A. Vanstone, and P. C. V. Oorschot, "Handbook of Applied Cryptography", Boca Raton, FL: CRC Press, June 1996.
[44] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D.Culler, and K. Pister, " System architecture directions for networked sensors", ACM SIGOPS operating systems review, vol. 34, pp. 93–104, 2000.
[45] T. Roosta, S. Shieh and S. Sastry, "Taxonomy of Security Attacks in Sensor Networks and Countermeasures", In Proceedings of First IEEE International conference on System integration and Reliability improvements, 2006.
[46] D. Martins and H. Guyennet, "Wireless Sensor Network Attacks and Security Mechanisms – A short survey", In Proceedings on 13-th International conference on Network-Based Information Systems (NBiS'10), Japan, 2010.
[47] Y. Wang, G. Attebury, and B. Ramamurthy, "A Survey of Security Issues In Wireless Sensor Networks", IEEE Communications Surveys & Tutorials, vol. 8(2), 2006.
[48] H.K.D. Sarma and A. Kar, "Security Threats in Wireless Sensor Networks", In Proceedings of 40th Annual 2006 International Carnahan Conference on Security Technology, Lexington, KY, USA, October 2006.
[49] A.D. Wood and J.A. Stankovic, "Denial of service in sensor networks," IEEE Computer, vol. 35(10), pp. 54-62, 2002.
[50] 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, pp. 113-127, Mai 2003.
[51] D. R. Raymond and S. F. Midkiff, "Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses", IEEE Pervasive Computing, vol. 7(1), pp.74-81, January 2008.
[52] K. Chelli, "Security Issues in Wireless Sensor Networks: Attacks and Countermeasures", In Proceedings of the World Congress on Engineering (WCE 2015), London, U.K, July 2015.
[53] E. Shi and A. Perrig, "Designing secure sensor network", Wireless Communication Magazine, vol. 11(6), pp. 38-43, December 2004.
[54] Y.W. Law, P. Hartel, J. den Hartog and P. Havinga, "Link-layer Jamming Attacks on S-MAC", In Proceedings of the Second European Workshop on Wireless Sensor Networks, pp. 217-225, Istanbul, Turkey, July 2005.
[55] J. Newsome, E. Shi, D. Song, and A. Perrig, "The Sybil attack in sensor networks: Analysis and defenses," In Proceedings of the 3rd International Symposium on Information Processing in Sensor Networks, pp. 259-268, April 2004.

[56] B.K. Mishra, M.C. Nikam, and P. Lakhadwala, "Security against black hole attack in wireless sensor network - a review", In Proceeding of Fourth International Conference on Communication Systems and Network Technologies, pp. 615-620, Bhopal, India, April 2014.
[57] R. R. Castro," Application-Driven security in Wireless sensor Networks", PhD thesis in computer science, university of Malaga, Spain, 2008.
[58] D. Stinson, "Cryptography: Theory and Practice", Second Edition. CRC/C&H, 2002.
[59] W. Stallings, "Cryptography and Network Security Principles and Practices", Fourth Edition, Publisher: Prentice Hall, Pub Date: November 16, 2005.
[60] "Specification for the Advanced Encryption Standard (AES)", Federal Information Processing Standards Publication 197 (FIPS PUB 197), 2001.
[61] R. L. Rivest, "The RC5 encryption algorithm", In Proceeding of the 2nd Workshop on Fast Software Encryption, Springer, pp.86-96, Berlin, Heidelberg, June 1995.
[62] B. Schneier, "Applied Cryptography", 2nd edition Wiley, ISBN 0-471-12845-7, 1996.
[63] R. Rivest, A. Shamir and L. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, vol. 21(2), pp.120-126, 1978.
[64] I. Blake, G. Seroussi and N. P. Smart, "Elliptic Curves in Cryptography", Cambridge University Press, ISBN 0-521-65374-6, 2000.
[65] D. Eastlake and P. Jones, "US Secure Hash Algorithm 1 (SHA1)", IETF Request for Comments 3174, September 2001.
[66] R. Rivest, "The MD5 Message-Digest Algorithm ", IETF Request for Comments 1321, 1992.
[67] Nist Publication, "The Keyed-Hash Message Authentication Code (HMAC)", 2002.
[68] V. Kumar,"Hash Chain Based Key Management for Heterogeneous Sensor Network", Master memory of technology in computer engineering, National Institute of Technology Haryana, India, July 2013.
[69] R. Bace, "Intrusion Detection", New Riders, ISBN 1-57870-185-6, 2000.
[70] S. Bela,"En efficient key management scheme for wireless sensor network", PhD thesis in philosophy, university of Thapar, Patiala, India, July 2014.
[71] J. Grobschädl, A. Szekely and S. Tillich, "The energy cost of cryptographic key establishment in wireless sensor networks", In Proceedings of the 2nd ACM symposium on Information, computer and communications security (ASIACCS '07), pp. 380-382, Singapore, March 2007.
[72] A. Price, K. Kosaka and S. Chatterjee, "A Key Pre-Distribution Scheme for Wireless Sensor Networks", In proceedings of Wireless Telecommunications Symposium, Pomona, CA, USA, April 2005.
[73] W. Du, J. Deng, Y.S. Han and P.K. Varshney, "A pairwise key predistribution scheme for wireless sensor networks", ACM Transactions on Information and System Security (TISSEC), vol. 8(2), pp.228-258, May 2005.
[74] A. Faquih, P. Kadam and Z. Saquib,"Cryptographic techniques for wireless sensor networks: A survey", In Proceedings of IEEE Bombay Section Symposium (IBSS), Mumbai, India, September 2015.
[75] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler,"SPINS: security protocols for sensor networks", Wireless Networks Journal, vol. 8 (5),pp. 521–534, September 2002.
[76] J. T. Kohl and B. C. Neuman, "The Kerberos Network Authentication Service (Version 5) Internet Engineering Task Force", Networking Group, Internet Draft RFC 1510, September 1993.

[77] R. kuchipudi and N. M. J. Basha, "Key Distribution Approaches for Wireless Sensor Networks", International Journal of Computer Science and Information Technologies (IJCSIT), vol. 3 (1), pp. 3187 – 3190, 2012.
[78] W. Diffie and M. E. Hellman, "New directions in cryptography", IEEE Transactions on Information Theory, vol. 22(6), pp. 644–654, November 1976.
[79] M. Abdalla and M. Bellare, "Increasing the Lifetime of a Key: A Comparative Analysis of the Security of re-Keying Techniques", In Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology (ASIACRYPT '00), pp. 546-559, December 2000.
[80] F. Hu, J. Ziobro, J. Tillett and N. Sharma, "Wireless Sensor Networks: Problems and Solutions", Systemics, Cybernetics and informatics, vol. 1(4), pp. 90-100, 2004.
[81] M. Simplicio, P. S. L. M. Barreto, C.B. Margi, and T.C. Carvalho, "A survey on key management mechanisms for distributed wireless sensor networks", Computer Networks, vol. 54(15), pp. 2591–2612, 2010.
[82] C.Y. Chen and H.C. Chao, "A survey of key distribution in wireless sensor networks", Security and Communication Networks, Wiley Online Library, July 2011.
[83] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy", Journal of Network and Computer Applications, vol. 33(2), pp.63–75, 2010.
[84] S.A. Çamtepe, B. Yener, "Key distribution mechanisms for wireless sensor networks: a survey", Technical Report, Rensselaer Polytechnic Institute, Troy, New York, March 2005.
[85] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks", In Proceedings of the 9th ACM conference on Computer and communications security, New York, NY, USA, pp. 41–47, November 2002.
[86] H. Chan, A. Perrig and D. Song, "Random key predistribution schemes for sensor networks", In Proceedings of the 2003 IEEE Symposium on Security and Privacy (SP'03), pp. 197–213,Washington, USA, May 2003.
[87] W. Du, J. Deng, Y. Han, S. Chen and P. K. Varshney, "A key management scheme for wireless sensor networks using deployment knowledge", In Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'04), pp. 586–597, Hong Kong, China March 2004.
[88] D. Xu, J. Huang, J. Dwoskin, M. Chiang and R. Lee, "Re-examining probabilistic versus deterministic key management", In Proceedings of International Conference on Information Theory (ISIT), pp. 2586–2590, Nice, France, June 2007.
[89] H. Chan and A. Perrig, "PIKE: peer intermediaries for key establishment in sensor networks", In Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM 2005), pp. 524 – 535, Miami, FL, USA, March 2005.
[90] D. Liu, P. Ning and W. Du, "Group-based key predistribution for wireless sensor networks ", ACM Transactions on Sensor Networks , vol. 4(2), March 2008.
[91] C. Blundo, A. Santis, A. Herzberg, S. Kutten, U. Vaccaro and M. Yung, "Perfectly-secure key distribution for dynamic conferences", In proceedings of 12th Annual International Cryptology Conference, Springer , pp. 471-486, Santa Barbara, California, USA. August 1992.
[92] S. A. Çamtepe and B. Yener, "Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks", IEEE/ACM Transactions on Networking, vol. 15(2), pp. 346-358, April 2007.
[93] J. Lee and D. R. Stinson, "A combinatorial approach to key predistribution for distributed sensor networks", In proceedings of IEEE Wireless Communications and Networking Conference, New Orleans, LA, USA, March 2005.

[94] R. Blom, "An optimal class of symmetric key generation systems", In proceedings of Workshop on the Theory and Application of Cryptographic Techniques (EUROCRYPT 84) , pp. 335-338, Paris, France, April 1984.
[95] M. Eltoweissy, M. Younis, and K. Ghumman, " Lightweight key management for wireless sensor networks", In Proceedings of the IEEE International Conference on Performance, Computing, and Communications, pp. 813–818, Phoenix, AZ, USA, April 2004.
[96] B. Lai, S. Kim, and I. Verbauwhede, "Scalable session key construction protocol for wireless sensor networks", In Proceedings of the IEEE Workshop on Large Scale Real-Time and Embedded Systems (LARTES), Washington, USA, 2002.
[97] S. Zhu, S. Setia, and S. Jajodia, "LEAP: efficient security mechanisms for large-scale distributed sensor networks", In Proceedings of International Conference on Computer and Communications Security (CCS'03), pp. 62-72, Washington, USA, October 2003.
[98] X. Zhang, J. He, and Q. Wei, "EDDK: energy-efficient distributed deterministic key management for wireless sensor networks", EURASIP Journal on Wireless Communications and Networking, vol. 2011(12), pp. 1-11, 2011.
[99] S. Zhu, S. Setia, and S. Jajodia, "LEAP+: Efficient security mechanisms for large-scale distributed sensor networks", ACM Transaction on Sensor Networks (TOSN), vol. 2(4), pp. 500–528, 2006.
[100] X. Zhang and J. Wang, "An efficient key management scheme in hierarchical wireless sensor networks", In Proceedings of International Conference on Computing, Communication and Security (ICCCS), IEEE , Pamplemousses, Mauritius, December 2015.
[101] M. L. Messai, H. Seba, and M. Aliouat, "A New Hierarchical Key Management Scheme for Secure Clustering in Wireless Sensor Networks", In Proceedings of 13th International Conference on Wired and Wireless Internet Communications (WWIC 2015), Springer, pp. 411-424, Malaga, Spain, May 2015.
[102] M. L. Messai and H. Seba, "EAHKM+: energy-aware secure clustering scheme in wireless sensor networks", International Journal of High Performance Computing and Networking, vol. 11(2), pp. 145-155, 2018.
[103] Q. Mamun, R. Islam, and M. Kaosar, "Secured Communication Key Establishment for Cluster based Wireless Sensor Networks", International Journal of Wireless Network and Broadband Technologies (IJWNBT), vol. 4(1), pp. 29-44, 2015.
[104] S. K. Gupta, N. Jain, and P. Sinha, "Clustering protocols in wireless sensor networks: A survey", International Journal of Applied Information System, vol. 5(2), pp. 41-50, 2013.
[105] A. H. Farooqi, F. A. Khan, "Intrusion detection systems for wireless sensor networks: A survey", In Proceedings of Communication and networking, Springer, pp. 234-241, Berlin, Heidelberg. December 2009.
[106] R. Roman, Z. Jianying, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks", In Proceedings of 3rd IEEE Consumer Communications and Networking Conference (CCNC), pp. 640-644, Las Vegas, NV, USA, January 2006.
[107] K. Sharma and M. K. Ghose, "Complete Security Framework for Wireless Sensor Networks", International Journal of Computer Science and Information Security, vol. 3(1), pp. 196-202, 2009.
[108] T. Morris, "Trusted platform module", Encyclopedia of cryptography and security, MA: Springer, pp. 1332-1335, Boston, 2011.

[109] K. Sharma and M. K. Ghose, "Cross layer security framework for wireless sensor networks", International Journal of Security and Its Applications, vol. 5(1), pp. 39-52, 2011.
[110] B.S. Kaliski Jr, Y.L. Yin, "On the Security of the RC5 Encryption Algorithm", http://ftp.arnes.si/security/crypto-tools/rsa.com/rsalabs/rc5/rc5-report.pdf.gz. [En ligne; accédé Avril 2018].
[111] O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks", IEEE Transactions on Mobile Computing, vol. 3(4), pp. 366–379, 2004.
[112] J. Deng, C. Hartung, R. Han, and S. Mishra, "A practical study of transitory master key establishment for wireless sensor networks", In Proceedings International Conference on Security and Privacy for Emerging Areas in Communications Networks (SecureComm'05), IEEE, pp. 289–299, Washington, USA, 2005.
[113] S. Mesmoudi, B. Benadda, A. Mesmoudi, "SKWN: Smart and Dynamic Key management scheme for wireless sensor networks, International Journal of communication Systems (IJCS), vol. 32 (7), 2019.
[114] N. Lee, M. Welsh, and D. Culler, " TOSSIM: Accurate and Scalable Simulation of Entire TinyOS Applications", In Proceedings of 1st International Conference on Embedded networked sensor systems (SenSys '03), ACM, pp. 126-137, Los Angeles, California, USA, November 2003.

Liste des publications

Revues Internationales

1- S. Mesmoudi, B. Benadda, A. Mesmoudi, "SKWN: Smart and Dynamic Key management scheme for wireless sensor networks, International Journal of communication Systems (IJCS), vol. 32 (7), 2019. DOI: 10.1002/dac.3930.

2- S. Mesmoudi, M. Feham, '' BSK-WBSN: Biometric Symmetric Keys to Secure Wireless Body Sensors Networks '', International Journal of Network Security & Its Applications (IJNSA), vol. 3(5), pp. 155-166, 2011. DOI: 10.5121/ijnsa.2011.3512.

Conférence nationale

1- Samira Mesmoudi, Asma Mesmoudi and Mohammed Feham, "Using Biometric Method to Secure Symmetric Key Establishment in Wireless Body Sensors Networks", Conférence Nationale sur les Technologies de l'Information et les Télécommunications CNTIT'13, 10-11 Décembre 2013.

Résumé

L'utilisation intensive des réseaux de capteurs sans fil (RCSF) dans des domaines très variés (médicale, militaire, environnemental, industriel) a soulevé plusieurs problèmes de sécurité. En effet, de nombreuses applications basées sur le RCSF nécessitent une communication sécurisée vu la sensibilité de certaines données collectées. Cette sécurité est généralement assurée par le cryptage des données transmis, ce qui nécessite l'établissement de nombreuses clés cryptographiques. La gestion de ces clés, dans un protocole, est une tâche importante qui garantit l'efficacité du mécanisme de sécurité. Le protocole doit être intelligemment adaptable non seulement aux événements d'intrusion, mais également au niveau de sécurité requis par certaines applications. Un protocole efficace optimise également l'énergie des capteurs et augmente par conséquent la durée de vie du réseau.
Dans cette thèse, nous proposons SKWN, un système de gestion de clés intelligent et dynamique pour les réseaux de capteurs sans fil hiérarchiques. Notre protocole propose trois sous-schémas pour l'établissement, le renouvellement et l'intégration de nouveaux nœuds. De plus, notre approche s'appuie sur une technique d'apprentissage automatique pour surveiller l'état du réseau et déterminer le niveau de sécurité approprié. Ainsi, SKWN ne fournit pas seulement des mécanismes de sécurité fiables, il optimise également la consommation d'énergie et les surcoûts liés à la communication et à l'utilisation de la mémoire. Les résultats présentés dans cette thèse sont issus de plusieurs simulations, qui démontrer la faisabilité et l'efficacité de notre proposition.

Mots clés : Réseau de capteurs sans fil hiérarchique, sécurité, gestion de clés, apprentissage automatique, l'efficacité d'énergie.

Abstract

The intensive use of wireless sensor networks (WSN) in a wide variety of areas (medical, military, environmental, industrial) has raised several security issues. Indeed, many WSN based applications require secure communication given the sensitivity of certain data collected. This security is generally ensured by the encryption of data transmitted by sensors, which requires the establishment of many cryptographic keys. Managing these keys, within a protocol, is an important task that guarantees the effectiveness of the security mechanism. The protocol should be intelligently adaptable not only to intrusion events but also to the security level needed by some applications. An efficient protocol optimizes also sensors energy and consequently increases the network life-cycle.
In this thesis, we propose SKWN, a Smart and dynamic Key management scheme for hierarchical Wireless sensor Networks. Our protocol offers three sub schemes to deal with key establishment, key renewal and new node integration. Our approach relies on a machine learning technique to monitor the state of the network and decide the appropriate security level. Thus, SKWN does not only provide reliable security mechanisms, but it also optimizes energy consumption and overheads related to the communication and memory usage. The results presented in this thesis are validated using several simulations, which demonstrate the feasibility and effectiveness of our proposal.

Key Words
Hierarchical wireless sensor network, security, key management, machine learning, energy-efficient.

ملخص

الإستخدام المكثف لشبكات الإستشعار اللاسلكية (WSN) في مجموعة واسعة من المجالات (الطبية، العسكرية، البيئية والصناعية) أثار العديد من المشكلات الأمنية. في الواقع، العديد من التطبيقات القائمة على شبكة WSN تتطلب اتصالًا آمنًا نظرًا لحساسية بعض البيانات التي يتم جمعها. هذا الأمن عادة ما يتم ضمانه من خلال تشفير البيانات التي تنتقل، الأمر الذي يتطلب إنشاء العديد من مفاتيح التشفير. إدارة هذه المفاتيح، ضمن بروتوكول، مهمة هامة تضمن فعالية آلية الأمن. حيث يجب أن يكون البروتوكول قابلاً للتكيف بذكاء ليس فقط مع أحداث التسلل ولكن أيضًا مع مستوى الأمن الذي تحتاجه بعض التطبيقات. يعمل البروتوكول الفعال على تحسين طاقة المستشعرات وبالتالي زيادة مدة حياة الشبكة.
في هذه الأطروحة، قمنا باقتراح بروتوكول جديد يدعى SKWN، وهو نظام إدارة مفاتيح ذكي وديناميكي لشبكات الإستشعار اللاسلكية الهرمية. هذا البروتوكول يقترح ثلاثة مخططات فرعية لإنشاء المفاتيح، تجديد المفاتيح ودمج العقد الجديدة. بالإضافة إلى ذلك، يعتمد منهجنا على أسلوب التعلم التلقائي لمراقبة حالة الشبكة وتحديد المستوى المناسب للأمن. وبالتالي لا يوفر البروتوكول SKWN آليات أمن موثوقة فحسب، بل إنها تعمل أيضًا على تحسين استهلاك الطاقة والتكاليف الإضافية المتعلقة باستخدام الاتصالات والذاكرة. النتائج المقدمة في هذه الأطروحة مستمدة من العديد من عمليات المحاكاة، والتي تثبت جدوى وفعالية اقتراحنا.

كلمات دلالية: شبكة الإستشعار اللاسلكية الهرمية، الأمن، إدارة المفاتيح، التعلم التلقائي، كفاءة الطاقة.