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

Université de Batna 2
Faculté de Mathématiques et d'Informatique

Département d'Informatique

# Thèse

***En vue de l'obtention du diplôme de***

**Doctorat en Sciences en Informatique**

**Titre :**

# Protocoles pour la Sécurité des Réseaux de Capteurs Sans Fil

Présentée et soutenue par :

**ATHMANI Samir**

**15/07/2018**

**Devant le jury composé de:**

| | | |
|---|---|---|
| *Président :* | *MELKEMI Kamal Eddine,* | *Prof. Université de Batna 2.* |
| *Rapporteur :* | *BILAMI Azeddine,* | *Prof. Université de Batna 2.* |
| *Examinateur :* | *BACHIR Abdelmalik,* | *Prof. Université de Biskra.* |
| *Examinateur :* | *BENMOHAMMED Mohamed,* | *Prof. Université de Constantine 2.* |
| *Examinateur :* | *CHIKHI Salim,* | *Prof. Université de Constantine 2* |
| *Examinateur :* | *TITOUNA Faiza,* | *MCA. Université de Batna 2.* |

# *Remerciements*

Je souhaite tout d'abord remercier mon directeur de thèse Pr. BILAMI Azeddine, Professeur au département d'informatique de l'université de Batna 2, directeur du laboratoire Informatique LaSTIC, qui grâce à sa disponibilité, ses qualités pédagogiques et scientifiques et ses rigoureux conseils, j'ai pu entamer, développer et mener à termes ce travail. Au-delà de ses qualités humaines, sa sympathie et son sens de l'écoute, qu'il trouve ici l'expression de toute ma gratitude pour tout le temps qu'il a consacré au suivi de cette thèse.

J'exprime ma gratitude à Pr. MELKEMI Kamal Eddine, Professeur au département d'informatique de l'université de Batna 2, qui m'a fait l'honneur de présider le jury. J'adresse également mes sincères remerciements aux membres de jury :

- *BACHIR Abdelmalik, Professeur à l'université de Biskra ;*
- *BENMOHAMMED Mohamed, Professeur à l'université de Constantine 2 ;*
- *CHIKHI Salim, Professeur à l'université de Constantine 2 ;*
- *TITOUNA Faiza, Maitre de conférences à l'université de Batna 2 ;*

qui ont accepté de juger mon travail. Je leurs suis très reconnaissant pour l'intérêt qu'ils ont porté à mes travaux.

Je remercie vivement Dr. BOUBICHE Djallel Eddine pour sa collaboration tout le long de ce travail; ses remarques et suggestions ont contribué dans l'élaboration de ce travail. Des remerciements particuliers vont aussi à Mlle BOUBICHE Sabrina, Dr. DRID hamza et Dr. Gazouli Lyamine qui ont contribué à améliorer la qualité de ce manuscrit que vous avez entre les mains.

Je dédie cette thèse à ma mère, ma femme et à mes deux petits enfants Aridj et Mohamed Iyad. Merci de m'avoir supporté (dans tous les sens du terme) pendant ces années. Enfin, je dédie aussi ce travail à tous mes frères et sœurs.

Enfin, mes remerciements les plus chaleureux vont vers tous ceux qui m'ont toujours encouragé et soutenu depuis le début de cette thèse.

# *Résumé*

La sécurisation de la communication réseau représente l'un des défis les plus importants dans les réseaux de capteurs sans fil. La plupart des protocoles de sécurité sont construits autour d'algorithmes de cryptage et d'authentification puissants. Pour atteindre les objectifs de sécurité, la gestion des clés est la première fonction fondamentale puisque les nœuds de capteurs ont besoin d'une clé commune valide pour exploiter les mécanismes de cryptographie. Le problème de la distribution des clés a été largement abordé dans les RCSF homogènes et divers mécanismes ont été proposés. Malgré la variété des solutions efficaces proposées dans ces catégories, l'équilibre entre le niveau de sécurité et la consommation de ressources reste le problème majeur dans les RCSF homogènes. Les réseaux de capteurs sans fil hétérogènes (HWSN) ont ouvert une nouvelle direction de recherche pour le problème de sécurité, et ont offert plusieurs opportunités. Les nœuds HSN sont équipés d'un processeur puissant, d'un stockage mémoire important, d'une batterie de haute capacité et peuvent communiquer sur de grandes distances. L'architecture d'un réseau hétérogène est divisée en deux niveaux : les tâches qui exigent beaucoup de ressources sont attribuées aux nœuds avec de grandes capacités HSN et les tâches qui n'ont pas besoin de ressources importantes sont déléguées à des nœuds de capteurs simples (LSN). Les RCSF hétérogènes offrent des avantages beaucoup plus importants que les RCSF homogènes pour un ensemble varié d'applications de sécurité. Le schéma d'établissement de clés peut également bénéficier de ces RCSF hétérogènes en exploitant les capacités élevées des nœuds HSN. Dans cette thèse, un schéma d'authentification dynamique efficace et de gestion des clés est proposé pour les RCSF hétérogènes. L'idée principale est de fournir un seul protocole léger pour l'authentification et l'établissement de clés tout en optimisant le niveau de sécurité. L'algorithme de distribution de clé est basé sur des informations préexistantes pour générer des clés dynamiques et ne nécessite aucune phase de partage et de canal sécurisé, ce qui améliore considérablement le niveau de sécurité, l'efficacité énergétique et réduit la consommation de mémoire. Les résultats expérimentaux ont confirmé les performances de notre mécanisme par rapport à certains protocoles de sécurité existants.

***Mots clés:*** Réseau de capteurs sans fil, La sécurité, La gestion des clés, L'authentification.

# Abstract

Securing the network communication represents one of the most important challenges in wireless sensor networks. Usually, most of the security protocols are built around strong encryption and authentication algorithms. To achieve security objectives, key management is the first fundamental function since the sensor nodes need valid common key to exploit cryptography mechanisms. The key distribution problem has been widely addressed in homogeneous WSNs and various mechanisms were proposed. Despite the variety of efficient solutions proposed in these categories, the balance between the security level and the resources consumption remains the major problem in homogeneous WSNs. Heterogeneous Wireless Sensor Networks (HWSN) have opened a new research direction for the security problem and offered several opportunities. By deploying high resources capacity sensor nodes (HSN), HWSNs outperform the classical homogeneous WSNs. The HSN are equipped with powerful processor, high capacity memory storage and batteries and can communicate on large distances. The heterogeneous network architecture is divided into two levels where high resource tasks are attributed to HSNs and low resource tasks are delegated to simple sensor nodes (LSN). HWSNs offer much more significant benefits than homogeneous WSNs for a variety set of security applications. Key establishment scheme can also benefit from such heterogeneous WSNs by exploiting the capabilities of powerful HSN. In this thesis, an efficient dynamic authentication and key Management scheme is proposed for heterogeneous WSN. The main idea is to provide a single lightweight protocol for both authentication and key establishment while optimizing the security level. The key distribution algorithm is based on preexisting information to generate dynamic keys and does not require any secure channel and sharing phase which improves the security, energy efficiency and reduces the memory consumption. Experimental results have confirmed the performances of our mechanism compared to some of the existing security protocols.



*Keywords:* wireless sensor networks, security, key management, authentication.

# ملخص

يمثل تأمين الإتصال أحد أهم التحديات في شبكات المجسات اللاسلكية. بصفة عامة، يتم إنشاء معظم بروتوكولات الأمان بإستخدام خوارزميات التشفير والمصادقة القوية. تعد إدارة مفاتيح التشفير الوظيفة الأساسية الأولى لتحقيق الأهداف الأمنية، حيث تحتاج المجسات إلى مفتاح مشترك لإستغلال آليات التشفير. تم التعامل مع مشكلة توزيع مفاتيح التشفير على نطاق واسع في شبكات المجسات اللاسلكية المتجانسة و تم إقتراح العديد من الآليات المختلفة. على الرغم من تنوع الحلول الفعالة المقترحة, إلا أن التوازن بين مستوى الأمان واستهلاك الموارد يبقى المشكل الرئيسي في هذه الفئة من الشبكات المتجانسة. لقد فتحت شبكات الاستشعار اللاسلكية و الغير متجانسة إتجاهًا جديدًا للبحث حول مشكلة أمن الشبكات وقدمت العديد من الفرص المهمة, و ذالك من خلال نشر مجسات إستشعار ذات قدرة عالية تتفوق على المجسات التقليدية المستخدمة في الشبكات المتجانسة. حيث تم تجهيز هذه الأخيرة بمعالج قوي، ذاكرة تخزين معتبرة وبطاريات عالية السعة كما يمكنها التواصل على مسافات بعيدة. تنقسم بنية الشبكة غير المتجانسة إلى مستويين حيث تُعزى المهام التي تتطلب موارد معتبرة إلى المجسات ذات القدرة العالية, ويتم تفويض المهام المستهلكة لموارد منخفضة إلى مجسات الإستشعار البسيطة. تقدم شبكات الاستشعار اللاسلكية و الغير متجانسة ميزات أكثر أهمية من نظيرتها المتجانسة من خلال مجموعة متنوعة من التطبيقات الأمنية, كما يمكن لنظام توزيع مفاتيح التشفير أن يستفيد أيضًا من الإمتيازات التي توفرها هذه الشبكات و ذالك من خلال استغلال إمكانات المجسات ذات القدرة العالية. في هذه الأطروحة، نقترح مخططا ديناميكيا فعالا و موحدا للمصادقة وإدارة توزيع مفاتيح التشفير مع تحسين مستوى الأمان للشبكات الغير متجانسة. تعتمد الفكرة الأساسية للخوارزمية المقترحة على معلومات موجودة مسبقًا علي مستوى المجسات للإنتاج مفاتيح ديناميكية دون الحاجة لإنشاء أي قناة آمنة أو مرحلة تبادل للمفاتيح, ما يحسن مستوى الأمن والتقليل من استهلاك الطاقة و الذاكرة. لقد أكدت النتائج التجريبية أداءً فعالاً لآليتنا مقارنة ببعض بروتوكولات الأمن المقترحة في العديد من الأبحاث المنشورة في هذا المجال.

***كلمات البحث الأساسية*** : شبكة المجسات اللاسلكية, الأمن, إدارة مفاتيح التشفير, المصداقية.

# TABLE DES MATIERES

## Liste des Figures

# Liste des Tableaux

## Liste des Equations

# INTRODUCTION GENERALE

- **Contexte**

Les progrès et les innovations technologiques considérables et impressionnantes dans les domaines de micro-électronique, et la communication sans fil réalisés dans ces dernières décennies, ont abouti à la conception et la production de petits nœuds capteurs plus intelligents et à faible coût. Ainsi, ces capteurs peuvent exploiter des fréquences radio pour construire un réseau de communication sans fil (qu'on appelle réseau de capteurs sans fil ou RCSF) pour collecter et relayer des données environnementales. La technologie de miniaturisation, le faible coût de construction des nœuds capteurs ainsi que le support de communication sans fil, sont les principaux avantages qui ont permis aux RCSF d'être déployés d'une manière plus facile, plus rapide et moins coûteuse dans des zones inaccessibles, des environnements éventuellement hostiles sans surveillance et à grande échelle que d'autres types de réseaux. Ces opportunités ont encouragé l'utilisation des RCSF dans diverses applications de surveillance et de contrôle pour résoudre une variété de problèmes, allant des applications simples, telles que la gestion du trafic, la détection de la pollution dans les bâtiments et les usines, à des opérations plus complexes et sensibles, comme la surveillance des champs de bataille pour les militaires, le contrôle de processus dans les installations chimiques et nucléaires, qui étaient considérées auparavant comme coûteuses, complexes, ou même irréalisables.

Le rôle critique, la variété d'applications et la popularité des RCSF ont apporté une grande importance à ce type de réseaux. Cette importance émergente des réseaux de capteurs pourrait être entravée par leurs problèmes de sécurité inhérents. La nécessité d'intégrer des services de sécurité devient l'une des principales préoccupations pour la pérennité du succès des RCSF dans un certain nombre de domaines.

- **Problématique**

Les RCSF présentent des défis de sécurité supplémentaires par rapport à la sécurité des réseaux classiques et Ad-hoc, en raison des caractéristiques techniques des nœuds et les avantages apportés par ce type de réseaux. Les avantages offerts par les RCSF (communication sans fil, déploiement à grand échelle, etc.) sont eux même des sources de vulnérabilités (failles de sécurité) et posent de nouvelles contraintes de sécurité. Aussi, la miniaturisation et le faible coût des nœuds capteurs ont entrainé des limites sévères pour toute solution de sécurité, ce qui nous impose de penser à une sécurité mieux adaptée que pour ses équivalents traditionnels des réseaux filaires et Ad-hoc. Nous constatons que la sécurité dans les RCSF est compliquée à cause des problèmes suivants :

  - Le coût global du RCSF devrait être aussi bas que possible. En raison de leur faible coût :

- Les nœuds capteurs sont sensibles à la capture et la violation physique.
- Les nœuds capteurs sont très limités en termes de ressources. Cette limitation a écarté l'utilisation de techniques cryptographiques avancées pour répondre à la question de sécurité dans les RCSF.

- La technologie de la transmission radio combinée avec les contraintes de ressources, rendent les RCSF plus vulnérables aux attaques par déni de service et de brouillage.
- Une solution adéquate de sécurité doit trouver un compromis entre la minimisation de la consommation de ressources et la maximisation du niveau de sécurité, ce qui entraine un conflit d'intérêt.
- La plupart des protocoles de sécurité dédiés aux RCSF utilisent une cryptographie à clé symétrique. L'utilisation des clés est indispensable pour sécuriser les communications. Ainsi, une gestion efficace de la distribution des clés doit être mise en place. Cette gestion n'est pas spécifique aux RCSF, mais elle doit tenir compte des contraintes sévères posées par ce type de réseaux.
- La topologie du réseau de capteurs sans fil facilite les attaques pour différents types d'attaques de lien allant de l'écoute passive à l'interférence active. Les attaques sur un RCSF peuvent provenir de toutes les directions et cibler n'importe quel nœud conduisant à des fuites d'informations secrètes, à des messages interférents, à l'imitation de nœuds, etc.
- La plupart des protocoles de sécurité standards actuels n'ont pas été conçus pour un grand nombre de participants. Une solution de sécurité pour les RCSF doit également évoluer pour des déploiements à grande échelle.

- **Motivations**

Actuellement, la recherche dans le domaine de sécurité des RCSF se concentre généralement sur les trois axes suivants :

- La gestion des clés : En raison des limitations des ressources des capteurs, la cryptographie asymétrique est souvent trop chère pour qu'elle soit appliquée dans les RCSF par rapport à la cryptographie symétrique. Par conséquent, la plupart des protocoles de sécurité dédiés aux réseaux de capteurs utilisent une cryptographie à clé symétrique. Ainsi, un schéma de gestion des clés cryptographiques efficace est indispensable pour préserver l'efficacité du système cryptographique utilisé pour sécuriser la communication dans le réseau.
- Authentification : l'authentification est considérée comme l'un des mécanismes de base pour la sécurité dans les réseaux de capteurs sans fil. L'authentification est indispensable pour de nombreuses tâches administratives (par exemple, la reprogrammation du réseau, le cycle de service du nœud capteur puits, garantir la légitimité des nouveaux nœuds insérés, etc.).
- Mécanismes de sécurité spécialisés : certains progrès ont été réalisés dont l'objectif est de fournir des services de sécurité spécialisés, tels que le routage sécurisé, la

localisation sécurisée, l'agrégation sécurisée et la synchronisation temporelle sécurisée.

Les protocoles de sécurité sont construits autour d'algorithmes de cryptage et d'authentification puissants. Pour atteindre les objectifs de sécurité, la gestion des clés est la première fonction fondamentale puisque les nœuds capteurs ont besoin d'une clé commune valide pour exploiter les mécanismes cryptographiques. Le problème de distribution des clés a été largement abordé dans les RCSF homogènes et divers mécanismes ont été proposés. Malgré la variété des solutions efficaces proposées dans ces catégories, l'équilibre entre le niveau de sécurité et la consommation de ressources reste le problème majeur dans les RCSF homogènes. Les réseaux de capteurs sans fil hétérogènes (HWSN) ont ouvert une nouvelle direction de recherche pour le problème de sécurité, et ont offert plusieurs opportunités. En déployant des nœuds capteurs de haute capacité (HSN), les RCSF hétérogènes surpassent les RCSF homogènes classiques. Les nœuds HSN sont équipés d'un processeur puissant, d'un stockage mémoire important, d'une batterie de haute capacité et peuvent communiquer sur de grandes distances. L'architecture de réseau hétérogène est divisée en deux niveaux : les tâches qui exigent beaucoup de ressources sont attribuées aux nœuds HSN et les tâches qui n'ont pas besoin de ressources importantes sont déléguées à des nœuds capteurs simples (LSN). Les RCSF hétérogènes offrent des avantages beaucoup plus importants que les RCSF homogènes pour un ensemble varié d'applications de sécurité. Le schéma d'établissement des clés peut également bénéficier de ces RCSF hétérogènes en exploitant les capacités élevées des nœuds HSN.

En effet, la gestion des clés reste inexplorée dans les RCSF hétérogènes et seules quelques recherches ont abordé le problème. La plupart de ces recherches sont basées sur des schémas symétriques de pré-distribution qui souffrent de problèmes tels que la distribution des clés probabiliste entre les HSN et les LSN, la non-extensibilité après le déploiement, le manque de stockage de mémoire sur les LSN limités en ressources et les frais généraux de communication.

- **Contributions**

Dans cette thèse, un schéma efficace d'authentification dynamique et de gestion des clés appelé EDAK est proposé pour les RCSF hétérogènes. Notre principal objectif est de résoudre les principaux problèmes de sécurité introduits par les schémas de distribution des clés et d'optimiser le niveau de sécurité. En outre, l'algorithme de génération de clés allégées préserve les ressources des nœuds LSNs, exploite les capacités des capteurs HSNs et réduit considérablement les frais généraux de communication. Les principales contributions du système de gestion des clés proposé sont les suivantes:

- Un algorithme d'établissement de clés efficace est proposé sur la base des informations préexistantes pour créer des clés dynamiques par paire entre les LSNs, des clés de groupe entre les têtes de sous-groupe et une clé de groupe pour les têtes de cluster et BS.

- Pour optimiser le niveau de sécurité et empêcher la capture des clés, un processus de génération de clés dynamiques est introduit qui crée une nouvelle clé pour chaque message transmis sans nécessiter d'échange d'informations supplémentaires.
- Un mécanisme d'authentification est également proposé pour identifier le nœud capteur légitime sur la base des informations locales (clé de matrice dynamique DMK).
- Notre schéma est implémenté sur des nœuds capteurs réels, où nous évaluons les coûts de calcul et de stockage sur le protocole EDAK.

- **Organisation du manuscrit**

Le présent manuscrit de thèse est organisé comme suit :

➢ ***PARTIE 1 :***

- ***Chapitre 1 :*** **Introduction à la sécurité dans les réseaux de capteurs sans fil**

Dans ce chapitre, nous détaillons les objectifs de sécurité, qui sont généralement le résultat de la réunion des différentes conditions de sécurité à savoir les réseaux informatiques classiques, les réseaux ad hoc et les propres conditions posées par les RCSF. Ensuite, nous étudions les différentes contraintes de sécurité dans les réseaux de capteurs sans fil. Ces contraintes proviennent de deux types de vulnérabilités: vulnérabilité due aux nœuds capteurs et vulnérabilité liée à la technologie des réseaux sans-fil.

- ***Chapitre 2 :*** **Attaques de sécurité dans les RCSF**

Dans ce chapitre, nous proposons une taxonomie pour les attaquants et une vue d'ensemble sur toutes les attaques de sécurité qui peuvent nuire au réseau de capteurs sans fil. Tout d'abord, Les attaquants sont catégorisés selon de nombreux critères à savoir : émission, emplacement, motivation, quantité, rationalité et mobilité. Ensuite, les attaques de sécurité sont classées en deux grandes catégories: les attaques passives et actives. Dans attaques passives, les adversaires ne produisent aucune émission. Les attaques actives peuvent également viser l'accès non autorisé et l'utilisation des ressources ou la perturbation des communications. A la fin de ce chapitre, nous résumons les menaces de sécurité qui font l'objet de différents types d'attaques.

- ***Chapitre 3 :*** **Mécanismes de sécurité pour les RCSF**

Dans ce chapitre, nous classons les mécanismes de sécurité en deux classes : les mécanismes de base (cryptographie, la gestion des clés et l'authentification), et les mécanismes avancés (modèle de confiance, les systèmes de détections d'intrusions).

- ***Chapitre 4 :*** **Gestion des clés dans les RCSF**

Dans ce chapitre, nous classons les schémas de gestion des clés en deux catégories à savoir : symétrique et asymétrique. Ensuite, nous discutons les schémas de gestion des clés dynamiques et nous terminons ce chapitre par l'étude de quelques schémas de gestion des clés proposés pour les RCSF hétérogènes.

- ***PARTIE 2:***

  - ***Chapitre 5 :*** **Un mécanisme efficace dédié pour la gestion des clés et l'authentification**

Ce chapitre est consacré à la présentation de notre nouveau mécanisme de sécurité, nommé EDAK ou « Efficient Dynamic Authentication and Key management mechanism », dédié pour la gestion et la distribution efficace des clés ainsi que la garantie d'authentification des nœuds légitimes dans le réseau. L'idée principale est de fournir un seul protocole léger pour l'authentification et l'établissement de clés tout en optimisant le niveau de sécurité.

  - ***Chapitre 6 :*** **Evaluation des performances**

Dans ce chapitre, nous analysons notre mécanisme de sécurité par rapport à quelques attaques de sécurité les plus dévastatrice dont l'objectif est de nuire les schémas de gestion des clés et d'authentification. Ensuite nous présentons une étude des performances de notre protocole, dans laquelle ce dernier est évalué en fonction de consommation en mémoire, et surtout de communication et de temps de calcul.

# Partie 1 :

# Etat de L'art

# CHAPITRE 1 :

# INTRODUCTION A LA SECURITE DANS LES RESEAUX DE CAPTEURS SANS FIL

# 1 Introduction

Grâce aux nombreux avantages et les nouvelles opportunités offertes par les réseaux de capteurs sans fil (RCSF), ces derniers figurent de nos jours dans presque tous les aspects de la vie et couvrent différents domaines d'application à savoirs : applications militaires, applications liées à la sécurité, applications environnementales, applications médicales, applications commerciales, applications écologiques, applications de traçabilité et de localisation. Cette diversité d'utilisation exige des RCSF de garantir un niveau de sécurité important. Un nombre considérable d'objectifs de sécurité s'impose (selon le domaine d'application des RCSF) pour répondre à la question de sécurité dans les RCSF.

La technologie de miniaturisation, le faible coût de construction des nœuds capteurs ainsi que le support de communication sans fil utilisés dans les RCSF, constituent les principales caractéristiques qui permettent une facilité de déploiement de ces derniers dans des environnements hostiles sans surveillance et à grand échelle. Ces opportunités ont encouragé l'utilisation des RCSF dans diverses applications de surveillance et de contrôle qui étaient considérées auparavant comme coûteuses, complexes, ou même irréalisables. Malheureusement, ces caractéristiques intéressantes et attirantes ainsi que les nouvelles opportunités des RCSF constituent en elles même des contraintes sévères pour rependre aux différentes conditions de sécurité dans les RCSF, où la satisfaction de ces derniers exige l'utilisation de techniques et d'outils coûteux en termes de ressources, ce qui dépasse les capacités des RCSF et limite l'application de ces techniques.

Dans ce chapitre nous détaillons les objectifs de sécurité, qui sont généralement le résultat de la réunion de différentes conditions de sécurité à savoir les réseaux informatiques classiques, les réseaux ad hoc et les propres conditions posées par les RCSF. Ensuite nous étudions les différentes contraintes de sécurité dans les réseaux de capteurs sans fil. Ces contraintes proviennent de deux types de vulnérabilités: vulnérabilités dues aux nœuds capteurs et vulnérabilités liées à la technologie des réseaux sans-fil.

# 2 Objectifs de sécurité dans les RCSF

Dans cette section, nous présentons les objectifs de sécurité (appelés aussi conditions de sécurité), où tous protocoles ou approches de sécurité destinés aux réseaux de capteurs sans fil sont conçus pour satisfaire un ou plusieurs de ces objectifs. Ces objectifs sont le résultat de la réunion des différentes conditions de sécurité à savoir les réseaux informatiques classiques, les réseaux ad hoc et les conditions posées par les RCSF. Les principales conditions de sécurité dans les RCSF sont énumérées ci-dessous:

## 2.1 Authentification

En raison de la nature sans fil des médias et de la nature non surveillée des réseaux de capteurs, l'authenticité de la communication est extrêmement indispensable dans ces conditions. Un service d'authentification fiable doit garantir l'authentification à deux niveaux [1], [2] :

- Authentification d'entité : avant de permettre à n'importe quelles entités ou participant d'accéder aux services ou à des informations dans le réseau, une vérification d'entité doit être effectuée pour assurer que le demandeur soit une entité légitime et autorisée à récupérer ce service ou information. L'authentification peut être effectuée entre deux nœuds communiquant ou un nœud (par exemple, une tête de grappe) et plusieurs autres nœuds autour de ce nœud (c'est-à-dire une authentification de diffusion)
- Authentification de l'origine des données: Les capteurs doivent s'assurer que les données reçues proviennent d'une source identifiée. Un adversaire n'est pas limité simplement à modifier le paquet de données. Il peut aussi changer complètement le trafic en injectant de faux paquets supplémentaires. Ainsi, le récepteur doit s'assurer que les données utilisées dans n'importe quel processus décisionnel proviennent d'une source correcte.

Un certain nombre de schémas d'authentification pour les RCSF ont été proposés dans la littérature. La plupart de ces schémas se basent sur un routage sécurisé pour délivrer des paquets fiables. Certains de ces schémas seront examinés dans la section 2.2 du chapitre 3.

## 2.2 Contrôle d'accès

Les services de contrôle d'accès fournissent des règles selon lesquelles les capteurs ou les utilisateurs peuvent rejoindre le réseau, envoyer des requêtes et accéder à une ressource du réseau. L'ensemble de ces règles varie souvent en fonction des tâches fournies par le réseau de capteurs et la façon dont il est exploité. Le contrôle d'accès devient particulièrement difficile en présence d'un nœud compromis et des attaques de type déni de service (DoS). Le contrôle d'accès peut être divisé en deux services de sécurité : l'authentification et l'autorisation[3].

## 2.3 Confidentialité

La confidentialité des données est une des pierres angulaires de la sécurité du réseau. La confidentialité est le fait de s'assurer que le contenu échangé (par exemple, des données collectées, des rapports, des commandes) entre les nœuds capteurs n'est accessible et compréhensible qu'à ceux dont l'accès est autorisé. Donc, même si un adversaire possède le privilège d'accéder au contenu, il ne devrait pas pouvoir décoder les messages échangés dans le réseau [1]. Dans les réseaux de capteurs, la confidentialité devrait répondre aux exigences suivantes [4], [5] :

- Un nœud capteur ne doit pas divulguer ses données aux nœuds voisins. En particulier dans une application militaire, les données stockées dans les nœuds capteurs peuvent être très sensibles.

- Dans de nombreuses applications, les nœuds communiquent des données hautement sensibles, par exemple la distribution de clés, il est donc extrêmement important de construire un canal sécurisé dans un réseau de capteurs sans fil.

- Les informations publiques sur les capteurs, telles que les identités des capteurs et les clés publiques, doivent également être cryptées dans une certaine mesure pour les protéger contre les attaques d'analyse de trafic.

L'approche standard pour sécuriser le transfert des données et assurer la confidentialité est de crypter les données avec une clé secrète partagée uniquement entre l'émetteur et le récepteur.

## 2.4 Auto-Organisation

Généralement, un réseau de capteurs sans fil est déployé aléatoirement, à grand échelle et sans infrastructure fixe pour la gestion du réseau. Pour ces raisons, un nœud capteur doit être indépendant et assez flexible à l'auto organisation. Cette caractéristique apporte un grand défi à la sécurité du réseau de capteurs sans fil. Donc, les nœuds capteurs doivent également s'auto-organiser pour gérer les clés et établir une relation de confiance entre les capteurs. Si l'auto-organisation fait défaut dans un réseau de capteurs, les dommages résultant d'une attaque ou même de l'environnement dangereux peuvent être dévastateurs [5].

## 2.5 Intégrité

L'intégrité des données dans les réseaux de capteurs est nécessaire pour assurer la fiabilité des données. Un service d'intégrité de données doit fournir des mécanismes aux nœuds communicants pour que ces derniers puissent détecter qu'un paquet n'a pas été falsifié, altéré ou modifié pendant la transmission. Même avec la mise en œuvre de la confidentialité, cela ne signifie pas que les données sont sécurisées. Un nœud malveillant peut ajouter quelques fragments ou manœuvrer les données dans un paquet. Ce nouveau paquet peut alors être envoyé au récepteur original. La perte ou les dommages de données peut même se produire sans présence d'un nœud malveillant mais aux conditions instables dues au canal de communication sans fil [1], [2], [5].

## 2.6 La sécurité de localisation

Dans quelques approches de sécurité, les informations de position des nœuds représentent un paramètre très important pour élaborer les protocoles de sécurité afin de pouvoir localiser et écarter les nœuds intrus. De plus, un réseau de capteurs dédié pour la surveillance d'environnement ou pour le suivi du développement d'un phénomène doit fournir l'endroit exact des événements et les anomalies détectées (ce qui correspond à l'endroit du nœud capteur lui-même) afin de localiser l'emplacement d'un défaut ou de prendre les décisions appropriées. Par conséquent, l'efficacité d'un réseau de capteurs repose sur sa capacité de localiser précisément, de manière sécurisée et automatiquement chaque nœud capteur dans le réseau.

Si les informations de localisation ne sont pas sécurisées correctement, un adversaire potentiel peut facilement manipuler et fournir de fausses informations de localisation par l'envoi d'un signal puissant erroné, rejouant des messages, etc. Plusieurs schémas de sécurité dont l'objectif est de sécuriser les informations de localisation ont été proposés tels que : le SeRLoC (appelé localisation sécurisée indépendante de la distance) et le VM (appelé multilatération vérifiable)[4].

## 2.7 Non-répudiation

La non-répudiation consiste à s'assurer qu'un capteur accepte la réception de tous les paquets qui lui sont destinés, et qui proviennent d'une source légitime, ainsi que l'envoi de tous les paquets destinés à l'autre partie impliquée dans la communication [1].

## 2.8 Fraîcheur

Même si la confidentialité et l'intégrité des données sont assurées, un adversaire peut facilement envoyer des paquets périmés dans le réseau. A la réception de ce genre de paquets, ces derniers peuvent être authentifiés et décryptés par un nœud sans détecter leur nature, ce qui implique des perturbations au niveau de plusieurs tâches dans les RCSF, parmi lesquelles on note : le résultat d'une fonction d'agrégation, quand il y a des stratégies de partage de clés utilisées dans la conception ou lorsque des tâches administratives et décisives se basent sur des anciens paquets. Pour résoudre ce problème, un compteur relatif au temps différent peut-être ajouté dans le paquet pour assurer la fraîcheur des données [6]. Ainsi, la fraicheur des données consiste à s'assurer que les paquets soient récents et qu'aucun vieux paquet n'a été rejoué dans le réseau.

## 2.9 Disponibilité

La disponibilité est la capacité permanente d'avoir accès à tous les services ou les fonctionnalités fournis par le réseau pour chaque membre du réseau. En effet, la disponibilité peut être menacée à cause des deux raisons suivantes :

- La présence d'attaques internes ou externes telles qu'une attaque par déni de service (DoS) [2].
- Les approches utilisées pour satisfaire les autres objectifs de sécurité telles que la confidentialité ou l'intégrité des données. Malheureusement, toutes ces approches affaiblissent la disponibilité d'un réseau (service, fonctionnalité, capteurs) pour les raisons suivantes[5] :
    - Les mécanismes cryptographiques exécutent des algorithmes supplémentaires. Le calcul supplémentaire consomme une énergie supplémentaire. S'il n'y a plus d'énergie, les ressources ou les nœuds ne seront plus disponibles.

- Certaines approches tentent d'utiliser une communication supplémentaire pour atteindre un objectif de sécurité. Une communication supplémentaire consomme également plus d'énergie. De plus, en mesure de l'augmentation de la communication, la possibilité d'un conflit de communication augmente également.

- Une défaillance ponctuelle sera introduite si un schéma de point central est utilisé. Ce qui menace considérablement la disponibilité du réseau.

# 3 Contraintes de sécurité dans les RCSF (sources de vulnérabilités dans les RCSF) :

Les RCSF présentent des défis de sécurité supplémentaires par apport la sécurité des réseaux classiques et Ad-hoc, en raison des caractéristiques techniques des nœuds et les avantages apportés par ce type de réseaux. Les avantages offerts par les RCSF (communication sans fil, large déploiement, etc.) représentent en eux même des sources de vulnérabilité (failles de sécurité) et posent de nouvelles contraintes de sécurité. Ainsi, la miniaturisation et le faible cout des nœuds capteurs entraînent des limités sévères pour toute solution de sécurité, ce qui nous pousse à envisager une sécurité mieux adaptée que celle des réseaux filaires et Ad-hoc.

Malgré les caractéristiques intéressantes des RCSF, ces derniers possèdent des contraintes strictes et intrinsèques. Ces contraintes proviennent de deux types de vulnérabilités: vulnérabilité due aux nœuds capteurs et vulnérabilité liée à la technologie des réseaux sans-fil [7].

## 3.1 Vulnérabilités du nœud capteur

Les ressources limitées du capteur nécessite que toute approche de sécurité ne soit pas couteuse en termes de ressources, pour prolonger la durée de vie du nœud et du réseau. Les ressources limitées des capteurs imposent des exigences strictes sur la conception d'un mécanisme de sécurité efficace pour les RCSF [8].

La particularité de fonctionnement du réseau de capteurs exige que les nœuds capteurs puissent être déployés dans des environnements non-protégés et laissés sans surveillance pendant de longues périodes, ce qui déclenche des nouveaux avertissements de sécurité pour les nœuds capteurs sans surveillance [7].

### 3.1.1 Protection physique faible

A cause de leurs faibles coûts, Les capteurs peuvent être déployés dans des environnements non-protégés (montagnes, forêts, champs de bataille, etc.).Ainsi, ils utilisent rarement des composants électroniques anti-corruption (tamper-résistant devices).

Par conséquent, ces réseaux sont vulnérables aux catastrophes naturelles (tremblements de terre, les tornades ou les inondations) et peuvent facilement être interceptés et corrompus ou encore subir des attaques physiques (destruction définitive des capteurs de telle sorte que les pertes soient irrécupérables) dans un tel environnement. Il est bien clair qu'aucun protocole de sécurité ne peut résister à ce type d'attaques physiques, mais des techniques de sécurité peuvent être conçues afin de fournir des capacités d'auto-réparation au réseau[7], [9], [10], [11].

### 3.1.2 Ressources extrêmement limitées de nœuds capteurs

La mise en œuvre de toute approche de sécurité nécessite une certaine capacité de ressources, y compris la mémoire des données, l'espace de code, puissance de calcul et l'énergie pour alimenter le capteur. Cependant, en raison du faible coût et de miniaturisation, ces ressources sont très limitées dans ce type de nœuds capteurs sans fil. Les principales limitations dues aux caractéristiques des nœuds capteurs sont [5]:

#### *3.1.2.1 Limitation de mémoire et d'espace de stockage*

Un capteur est un petit dispositif avec une capacité limitée de mémoire et d'espace de stockage pour le code. Par exemple un capteur de type *Mica mote*, possède un processeur Atmel ATMEGA103 4 MHz avec 128 Ko de mémoire d'instructions, 512 Ko de mémoire flash, et seulement 4 Ko de RAM pour les données. Donc avec une telle limitation, il est indispensable de limiter la taille du code de l'algorithme de sécurité afin de construire un mécanisme de sécurité efficace [7], [9].

#### *3.1.2.2 Limitation de la puissance énergétique*

L'énergie est un autre défi dans les RCSF; elle est considérée comme la contrainte la plus sévère aux capacités des capteurs sans fil. C'est l'une des principales raisons pour lesquelles les nœuds sont sujets à des défaillances en raison de l'épuisement des batteries. Une fois les nœuds capteurs déployés dans un réseau de capteurs, ils ne peuvent pas être facilement remplacés (coût d'exploitation élevé) ou rechargés [7].

Pour transmettre des données, Un nœud capteur doit allumer son antenne radio ce qui consomme beaucoup d'énergie (la transmission est particulièrement coûteuse en termes de puissance énergétique). Si des nœuds stratégiques et importants subissent une attaque de privation de sommeil, ou l'attaquant transfère généralement des paquets inutiles vers le nœud cible afin de garder sa radio allumée, ce qui consomme leur batterie afin de l'épuiser complètement. Par conséquent, le nœud capteur devient incapable de prendre part au processus de communication, ce qui dégrade sérieusement les performances du réseau [12].

Ainsi, la consommation d'énergie doit être minimisée pour prolonger la durée de vie des capteurs; cela nécessite à la fois l'efficacité énergétique du matériel ainsi que l'efficacité de la sécurité et d'autres protocoles de routage [13].

#### *3.1.2.3 Limitation de puissance de calcul*

En raison de la petite taille des nœuds et le faible coût, les nœuds capteurs disposent d'un microcontrôleur à faible capacité de calcul. Par exemple, les capteurs de type *Telos B* contiennent un processeur RISC 16 bits avec 8 MHz. De telle contrainte sur la puissance de calcul exige des algorithmes de sécurité extrêmement compétents en termes de complexité de calcul. Ainsi, ceci réduit également la faisabilité de certaines techniques de cryptage efficace.

## 3.2 Vulnérabilités technologiques du réseau

Malgré les améliorations matérielles et logicielles apportées aux RCSF ces dernières décennies, les principaux obstacles de sécurité dans les RCSF émergent à partir des caractéristiques du réseau qui les rendent efficaces et attirants:

### *3.2.1* Communication non fiable

Une communication non fiable est une autre source de vulnérabilité pour la sécurité des RCSF. La sécurité du réseau dépend fortement d'un protocole bien défini, qui à son tour dépend de la communication. Les principaux paramètres influant la sécurité de communication dans les RCSF sont définis dans[7].

#### *3.2.1.1 Support sans fil*

La nature elle même du support de communication sans fil est l'une des principales menaces de sécurité des RCSF, Contrairement aux réseaux filaires où un périphérique doit être physiquement connecté au support, le support de communication sans fil est ouvert et accessible à tout le monde. Cela conduit à plus de soucis de sécurité dans les RCSF, ce qui constitue l'une des primordiales menaces à la sécurité des capteurs. Avec le moindre effort, un adversaire qui pénètre dans la zone de couverture peut commodément capturer, Falsifier ou rejouer tous les messages échangés.

Un intrus ayant un émetteur puissant peut rendre les nœuds capteurs incapables de transmettre des paquets par la production d'un bruit sur le canal. Donc, le média peut apparaître comme occupé en permanence. Les médias sans fil permettent facilement aux intrus d'interceptés de détruire des paquets valides, et injecter des malveillants ou encore corrompus [9], [12]–[14].

#### *3.2.1.2 Transfert non fiable*

Dans les RCSF, le routage des paquets est sans connexion, ce qui est intrinsèquement peu fiable. Dans le cas d'une erreur de canal ou l'abondant des nœuds hautement congestionnés, des paquets peuvent être corrompus et par conséquent, des paquets critiques de sécurité peuvent être endommagés ou perdus [7], [15].

#### *3.2.1.3 Latence*

Une valide synchronisation entre les nœuds est indispensable pour tout mécanisme de sécurité qui se base sur la distribution de clés cryptographiques et les rapports d'événements. Une synchronisation correcte entre les nœuds capteurs dans les RCSF est presque impossible en raison d'encombrement du réseau, le routage multi-sauts et le traitement des nœuds, ce qui introduit une latence importante dans le réseau [7], [15].

#### *3.2.1.4 Conflits*

Une communication fiable ne peut être assurée à cause de la propriété de diffusion dans les RCSF. Au milieu du transfert des paquets, des conflits peuvent se produire en raison des collisions des paquets, ce qui provoque l'échec du transfert. Un intrus puissant peut facilement exploiter cette faiblesse, afin de perturber le réseau par la production d'interférences dans la zone de couverture [7], [13], [15].

#### *3.2.1.5 Environnement multi-sauts:*

Afin de réduire le coût de déploiement, et pour un déploiement facile et rapide, une architecture multi-sauts est indispensable pour les réseaux de capteurs sans fil, dans laquelle les nœuds ont la possibilité d'auto-guérison, d'auto-configuration et d'auto-ajustement. Ce type d'architectures permet aux adversaires de menacer la sécurité par l'exploitation de quelques attaques comme : l'attaque de trou noir, l'attaque de routage sélectif, l'attaque sybil, ainsi que les attaques qui permettent la création de chemins erronés ou inexistants entre la source et la destination [12], [16].

### 3.2.2 Déploiement à grande échelle :

Afin de couvrir des zones immenses et de fournir une redondance, le nombre de nœuds capteurs dans un RCSF peut être très important (de dizaines à des centaines de milliers de nœuds). Ainsi, les capteurs peuvent être largués par voie aérienne et leur emplacement géographique exact peut donc être imprédictible. Ces deux facteurs offrent des avantages aux attaquants qui incitent le réseau à les accepter comme des nœuds légitimes, ou plus encore d'utiliser leur propre formule pour capturer ou reprogrammer les nœuds légitimes dans le réseau.

L'échelle étendue dans les RCSF pose des problèmes de recherche sérieux et complexes pour le développement d'un mécanisme de sécurité qui prend en charge un grand nombre de nœuds répartis sur une grande surface, en maintenant l'énergie , l'espace mémoire et la puissance de calcul du nœuds capteurs [9], [16].

### 3.2.3 Topologie de réseau dynamique

De nombreux facteurs rendent la topologie des RCSF dynamique et non déterministe. Donc, aucune topologie fixe ne peut être définie à l'avance en raison de quelques facteurs, tels que le

réapprovisionnement périodique du réseau (la révocation de nœud épuisé ou défaillant et l'ajout de nouveaux nœuds), les mouvements de nœud, etc. cette topologie instable qui aura une grande incidence sur les performances des protocoles de sécurité, rend les mécanismes de sécurité traditionnels, basés sur des configurations statiques, impossibles à appliquer.

Il est donc nécessaire que tout mécanisme de sécurité conçu pour ces réseaux puisse fonctionner dans cet environnement dynamique et évolutif. Des techniques de sécurité plus robustes doivent être élaborées, et qui peuvent s'adapter dynamiquement en fonction de la modification de la topologie du réseau. Pour cela, une transparence en ce qui concerne l'ajout et la révocation des nœuds sur le réseau, ainsi que des informations concernant la reconfiguration de la topologie du réseau est nécessaire [9], [13], [17].

### 3.2.4 Manque d'identifications globales

En raison de l'échelle et la dynamicité du RCSF, il est impossible d'affecter une identification globale pour l'ensemble des nœuds. Par conséquent, il est impossible d'appliquer les protocoles de sécurité existants, en se basant sur une identification unique des nœuds au sein du réseau [17].

## 4 Conclusion

Dans ce chapitre, nous avons décrit en premier lieu les objectifs de sécurité standards ainsi que les objectifs de sécurité spéciaux pour les RCSF. Ensuite, nous avons présenté les différentes contraintes de sécurité ainsi que les failles de sécurité. Ces contraintes freinent considérablement le déploiement de techniques avancées pour satisfaire les objectifs de sécurité. Par conséquent, plusieurs types d'attaques peuvent être facilement montés par des attaquants plus ou moins sophistiqués, exploitant ces failles de sécurité et les faibles techniques employées afin de nuire au fonctionnement du réseau. Le chapitre suivant sera consacré à la présentation des différents types d'attaquants et d'attaques de sécurité dans les RCSF.

Nous constatons qu'une préalable étude détaillée sur le domaine d'application permet une restriction sur les objectifs de sécurité nécessaires pour assurer le bon fonctionnement des tâches affectées au réseau. Cette restriction nous permet de maîtriser les contraintes sévères liées aux RCSF durant la conception et le choix du système cryptographique et les techniques avancées pour sécuriser le réseau, en développant des mécanismes équilibrés en termes de ressources et de niveau de sécurité.

# CHAPITRE 2 :

# ATTAQUES DE SECURITE DANS LES RCSF

# 1 Introduction

Tout système ou réseau informatique peut être sujet à divers types d'attaques qui ciblent généralement les conditions de sécurité standards à savoir la confidentialité, la disponibilité, l'authentification et l'intégrité. Ces attaques peuvent être classées en quatre catégories :

- L'interception est une attaque contre la confidentialité. Le réseau de capteurs peut être compromis par un adversaire pour obtenir un accès non autorisé au nœud capteur ou aux données stockées dans celui-ci.
- L'interruption : est une attaque qui tente de rendre les services et les ressources du réseau indisponibles, par exemple la capture physique des nœuds, la corruption de messages, l'insertion de code malveillant, etc.
- La fabrication est une attaque contre l'authentification. En fabrication, un adversaire injecte de fausses données et compromet la fiabilité de l'information relayée.
- La modification est une attaque contre l'intégrité. Une modification signifie qu'une partie non autorisée accède non seulement aux données, mais les altère, par exemple en modifiant les paquets de données en cours de transmission ou en provoquant une attaque par déni de service telle que l'inondation du réseau avec des données erronées.

Les réseaux de capteurs sans fil sont aussi vulnérables aux types d'attaques listés ci-dessus et à plusieurs autres types d'attaques qui peuvent être réalisées par l'exploitation de failles de sécurité particuliers aux RCSF. Ces attaques peuvent être effectuées de diverses manières, notamment par des attaques par déni de service, par l'analyse du trafic, la violation de la confidentialité, les attaques physiques, etc., et par l'utilisation d'un nœud plus puissant qui peut facilement compromettre le réseau, perturber la bande passante ou capturer un nœud capteur afin d'empêcher effectivement le réseau de capteurs de remplir son rôle.

Dans ce chapitre, nous proposons une taxonomie pour les attaquants et une vue d'ensemble sur toutes les attaques de sécurité qui peuvent nuire au réseau de capteurs sans fil. Tout d'abord, les attaquants sont catégorisés selon de nombreux critères à savoir : émission, emplacement, motivation, quantité, rationalité et mobilité. Ensuite, les attaques de sécurité sont classées en deux grandes catégories: les attaques passives et actives. Dans les attaques passives, les adversaires ne produisent aucune émission. Les attaques actives peuvent également viser l'accès non autorisé et l'utilisation des ressources ou la perturbation des communications. A la fin de ce chapitre, nous résumons les menaces de sécurité qui font l'objet de différents types d'attaque.

# 2 Les attaquants

Une attaque peut être définie comme une tentative d'accès non autorisé à un service, une ressource ou une information, ou la tentative de compromettre l'intégrité, la disponibilité ou la confidentialité d'un système. Les attaquants, les intrus ou les adversaires sont à l'origine d'une attaque. La faiblesse d'une conception, d'une implémentation, d'une configuration ou d'une limitation de la sécurité du système qui pourrait être exploitée par des attaquants est connue sous le nom de vulnérabilité ou faille. Toute circonstance ou tout événement (tel que l'existence d'un attaquant et les vulnérabilités) susceptibles d'avoir un impact négatif sur un système à travers une faille de sécurité est appelé menace et la probabilité qu'un attaquant exploite une vulnérabilité particulière, causant des dommages à un actif du système est connue sous le nom de risque [14].

Les attaquants peuvent également être catégorisés selon de nombreux critères. La Figure 1 illustre les différentes caractéristiques utilisées pour présenter une classification détaillée des attaquants.

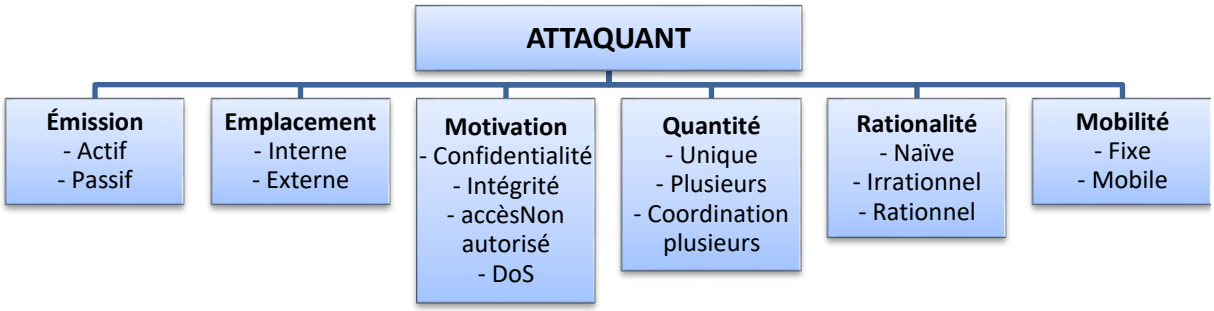


**Figure 1 : Taxonomie des Attaquants.**

Un attaquant peut être un nœud interne ou externe du réseau. Un attaquant interne est un nœud qui a été compromis et fait partie du réseau attaqué. Par conséquent, l'attaquant peut connaître toutes les informations cryptographiques appartenant au nœud compromis. Donc les attaques actives peuvent être organisées par des attaquants internes. En d'autres termes, un attaquant interne peut être vu comme un nœud qui a été enregistré légalement ou un nœud autorisé à accéder au réseau. Les attaques externes peuvent être passives ou actives. Un attaquant externe est généralement un nœud ou plusieurs qui ne sont pas les bienvenus sur le réseau. Lorsqu'il y a plusieurs attaquants, ils peuvent collaborer les uns avec les autres, ce qui peut être considéré comme un cas plus difficile à défendre [18].

# 3 Attaques de sécurité

Vues les sources de vulnérabilité des RCSF, des nombreuses attaques peuvent être fabriquées afin d'altérer les messages transférés, perturber le bon fonctionnement du réseau, menacer la confidentialité des données et dans le pire des cas paralyser complètement le réseau ou le mettre hors service.

Toute tentative d'altérer l'intégrité, la disponibilité ou la confidentialité d'un système, la tentative d'accès non autorisé à un service, une ressource ou une information est considérée comme une attaque[14].

En fonction de l'emplacement de l'attaquant, le niveau de dommages apportés et les dispositifs d'attaque utilisés, les attaques dans le réseau de capteurs sans fil sont discutées et classées de différentes manières dans la littérature. Nous suivons dans cette section la classification des attaques selon les deux grandes catégories: les attaques passives et actives. Ainsi, dans chaque catégorie, on présente une classification des attaques dans les différentes couches. Une classification générale des attaques est donnée comme suit:

## 3.1 Attaques passives

Dans ce type d'attaques, les attaquants sont typiquement camouflés, c'est-à-dire cachés, ils se limitent à l'écoute, la collecte des données et l'analyse du trafic échangé, ce qui permet à l'attaquant d'intercepte, de contrôler et d'observer les données entre les nœuds communicants. Puisque l'attaquant n'apporte aucune modification aux paquets échangés, ce type d'attaques sont faciles à réaliser et en même temps difficiles à détecter.

L'objectif de ce type d'attaques est de comprendre l'échange d'informations entre les nœuds communicants et les nœuds d'agrégation des données, les informations de routage (en utilisant l'analyse du trafic), trouver des informations utiles (en analysant les en-têtes des paquets, leur taille et la fréquence de transmission). Par conséquent, ces information peuvent être utilisées ensuite pour effectuer des attaques plus sophistiquées. Les attaques de cette catégorie peuvent être regroupées dans les types suivants[17], [18] :

### 3.1.1 Camouflage d'adversaires

Suite à une attaque active, un adversaire peut être compromis ou inséré dans un chemin de routage comme étant un nœud légitime pour attirer des paquets afin d'analyser le trafic dans une région.

### 3.1.2 Écoute (Eavesdropping)

L'écoute passive est définie comme l'acte d'écouter discrètement une conversation privée. Un attaquant s'insère dans un chemin actif, afin d'écouter passivement tout le trafic envoyé sur le

support de diffusion et extraire les données collectées par l'ensemble du réseau (données agrégées). Si aucun mécanisme cryptographique n'est utilisé pour protéger les messages, l'adversaire pourrait facilement comprendre le contenu de la conversation, ce qui menace la confidentialité des données. Dans le cas où un mécanisme cryptographique est utilisé, l'écoute entre deux nœuds capteurs ne peut pas aider l'attaquant dans la compréhension approfondie de l'ensemble du réseau, mais elle peut être exploitée pour préparer une attaque plus sophistiquée ou une attaque active[17], [19].

### 3.1.3 Analyse du trafic

En raison de l'analyse approfondie du trafic, un adversaire combine l'écoute et l'analyse de trafic pour réaliser une attaque efficace. Par conséquent, il est possible d'obtenir des informations utiles sur la topologie de réseau, comprendre les rôles des nœuds ou d'identifier la station de base. Par exemple, les contacts d'un nœud peuvent être déterminés en filtrant le trafic du réseau[17], [19].

L'analyse du trafic [18], [20] peut également être utilisée pour cibler des informations confidentielles. Par exemple, dans une communication tactique, l'augmentation du trafic fait comprendre à l'adversaire qu'il y a des événements qui vont se produire, ainsi que le silence qui peut indiquer une préparation à une attaque, une pénétration ou un mouvement tactique. L'analyse du trafic peut être réalisée par l'une des techniques suivantes:

#### *3.1.3.1 Analyse du trafic sur la couche physique*

Cette attaque ne peut être réalisée que sur une topologie bien déterminée ou les nœuds capteurs sont à un emplacement fixé, elle se limite à l'analyse des débits de trafic dans le réseau et la détection de porteuse.

#### *3.1.3.2 Analyse du trafic dans les couches MAC et supérieures*

Un adversaire peut extraire des chemins de routage et des informations utiles concernant la topologie du réseau, par l'application d'une analyse approfondie du modèle de trafic ainsi que les entêtes des trames.

#### *3.1.3.3 Analyse du trafic par corrélation d'événements*

Un adversaire peut cibler la détection des nœuds sources pour certains paquets de données, ce qui lui permet de détecter l'emplacement des événements. Cette information peut être corrélée avec le trafic ainsi que d'autres informations pour détecter le type ou le rôle de certains nœuds tels que les chefs de clusters. Par conséquent une attaque de déni de service peut avoir lieu[18].

## 3.2 Attaques actives

Les attaques actives n'agissent pas seulement sur la confidentialité des données, mai peuvent également influer sur la disponibilité, la fraîcheur et l'intégrité des données, ainsi que d'autres mesures de sécurité. Pour cela, un attaquant actif participe à tout le trafic quelques soient

les paquets de contrôle ou de données, ce qui lui permet de faire injecter son propre trafic, modifier, supprimer des message ou rejouer d'autres anciens. Afin de perturber le fonctionnement du réseau, un attaquant actif peut également falsifier les informations de routage, viser l'accès non autorisé et l'utilisation des ressources ou encore provoquer des attaques d'épuisement de ressources.

Contrairement aux attaques passives, ce type d'attaques peut être détecté par l'élaboration de mécanismes de sécurité avancés, mais la question la plus difficile dans ce type de réseaux c'est de protéger le réseau contre tous types d'attaques actives afin d'assurer le bon fonctionnement et la vivacité du réseau. On regroupe les attaques actives les plus connues selon les classes suivantes[18], [20].

### 3.2.1 Attaques de la couche physique

Le rôle principal de cette couche est de moduler les données et les acheminer dans le media physique tout en choisissant les bonnes fréquences porteuses, la détection du signal et le cryptage des données. Deux attaques sont explorées dans cette couche.

#### *3.2.1.1 Attaque de brouillage (Jamming)*

Vu la sensibilité du canal sans fil, un attaquant utilise un puissant dispositif de brouillage pour interférer le canal de communication entre deux interlocuteurs. Cette attaque peut être très dangereuse si elle cible des nœuds puits tels que la station de base ou le chef de cluster pour isoler toute une région ou la station de base afin de paralyser l'ensemble du réseau.

Afin de défendre les RCSF contre le brouillage de communication, les techniques de transmission de signaux en commutant rapidement une porteuse parmi de nombreux canaux de fréquence sont employées, telles que l'étalement de code et le saut de fréquence, ce qui empêche l'attaquant de détecter le canal de fréquence utilisé entre l'émetteur et le récepteur. D'autres techniques sont utilisées pour bloquer les attaques de ce type. Ces techniques sont couteuses en termes de complexité et d'énergie, ce qu'exige à la communauté de proposer d'autres approches de sécurité pour maintenir les exigences des nœuds capteurs telles que la faible puissance des batteries et le faible coût des dispositifs de détection. Une défense logique consiste à mettre les capteurs dans un mode de veille à long terme et les réveiller périodiquement pour tester le canal[4], [6], [21].

#### *3.2.1.2 Attaque d'altération (Tampering)*

Généralement, Les RCSF sont déployés aléatoirement dans des environnements non-protégés sans aucune surveillance. Dans un tel environnement, la manière la plus simple d'attaquer est d'endommager ou de modifier physiquement les capteurs, ce qui permet d'arrêter ou de modifier leurs services. L'impact négatif sera plus dangereux si les stations de base ou les points d'agrégation sont attaqués. Cependant, l'efficacité de ces attaques contre les capteurs physiques est très limitée en raison de la densité et la redondance élevée des nœuds dans la plupart des RCSF. Sauf si une grande quantité de capteurs est compromise[22]. L'altération physique est une autre façon

d'attaquer. En effet, un attaquant peut capturer un nœud capteur ce qui lui permet d'obtenir facilement des informations sensibles (clés cryptographiques), des données qui le traversent, altérer les circuits électroniques, reprogrammer le nœud, ce qui cause généralement des dégâts irréversibles [21].

### 3.2.2 Les attaques de la couche liaison

La couche liaison est responsable du multiplexage des flux de données, de la détection des trames de données, du contrôle d'accès au support et du contrôle des erreurs[4]. Les attaques sur cette couche provoquent des collisions intentionnellement, pour perturber la communication entre l'ensemble des nœuds dans le réseau.

Les algorithmes de la couche liaison, en particulier les schémas MAC, présentent de nombreuses opportunités permettant l'exploitation des attaques de type DoS. Par exemple, entre autres, les attaques de type DoS de la couche MAC qui peuvent bloquer un canal en permanence sont :

- Chaque fois qu'un signal RTS est reçu, un signal qui entre en collision avec le signal CTS est transmis. Puisque les nœuds ne peuvent pas commencer à transmettre des données avant de recevoir le signal CTS, ils continuent d'envoyer des signaux RTS.
- Si le schéma MAC est basé sur des périodes actives et inactives, le blocage uniquement des périodes actives peut bloquer en permanence le canal.
- De faux signaux RTS ou CTS avec de longs paramètres de transmission de données sont envoyés en continu, ce qui fait que les autres nœuds qui effectuent la détection de porteuse virtuelle attendent indéfiniment.
- L'usurpation d'accusé de réception, où un adversaire envoie des accusés de réception erronés.

Dans la sous-section suivante, on examine trois catégories d'attaques qui peuvent cibler la couche liaison de données.

#### *3.2.2.1 Collisions*

Une collision se produit lorsque deux nœuds tentent d'émettre simultanément sur la même fréquence. Les paquets qui entrent en collision sont rejetés et doivent être retransmis. Un adversaire peut provoquer de manière stratégique des collisions dans des paquets spécifiques tels que des messages de contrôle ACK ainsi que dans des périodes critiques tels que les périodes de réveil ou les périodes actives. Les résultats possibles de telles collisions peuvent être entre autres, l'épuisement des ressources, l'injustice dans l'allocation, et peuvent conduire à des retards exponentiels coûteux dans certains protocoles de contrôle d'accès aux médias. Par conséquent, le fonctionnent des

applications en temps réel, qui s'exécutent sur d'autres nœuds se dégrade à cause de la perturbation par l'interruption de leurs transmissions de trames[4].

Plusieurs techniques ont été proposées pour défendre le réseau contre les collisions. Au niveau des collisions qui se produisent en raison d'erreurs environnementales, la technique la plus adaptée est les codes de correction d'erreur. Malheureusement, ce type de codes nécessite des frais supplémentaires de traitement et de communication pour surmonter les collisions. Mais, nous devons accepter le fait que nous ne serons pas capables de corriger plus que ce qui a été corrompu. Même s'il n'est pas impossible de détecter ces collisions malveillantes, jusqu'à présent, il n'y a pas de technique de défense appropriée pour surmonter complètement ces attaques[19].

#### *3.2.2.2 Épuisement:*

Des collisions répétées peuvent également être utilisées par un attaquant pour provoquer l'épuisement des ressources. Cette attaque domine les ressources de puissance des nœuds en les obligeant à retransmettre le message même en l'absence de collision ou de collision tardive [12]. Une solution possible consiste à appliquer des limites de débit au contrôle d'admission MAC de sorte que le réseau puisse ignorer les demandes excessives, évitant ainsi la perte d'énergie causée par les transmissions répétées. Une seconde technique consiste à utiliser le multiplexage temporel où chaque nœud se voit attribuer un créneau temporel dans lequel il peut transmettre. Ceci élimine le besoin d'arbitrage pour chaque trame et peut résoudre le problème d'ajournement indéfini dans un algorithme de back-off. Cependant, il est toujours sensible aux collisions[23].

#### *3.2.2.3 Injustice:*

L'injustice peut être considérée comme une forme faible d'une attaque DoS. Un attaquant peut provoquer une injustice dans un réseau en utilisant de manière intermittente les attaques de couche liaison ci-dessus[23]. Les protocoles MAC de la couche de liaison administrent les communications dans les réseaux en contraignant les schémas de priorité pour une corrélation transparente. Il est possible d'utiliser ces protocoles affectant ainsi les schémas de préséance, ce qui aboutit finalement à une diminution du service[12].

### 3.2.3 Les attaques au niveau de la couche réseau (Routing Attacks)

La fonction principale de la couche réseau [24]est de router les messages d'un nœud capteur vers un autre. Ceci joue un rôle important pour améliorer les performances du réseau (l'efficacité énergétique, l'adressage et la localisation des nœuds). La plupart des protocoles de routage sont vulnérables à une foule d'attaques parce qu'ils sont conçus pour assurer la fonctionnalité opérationnelle et la convivialité du réseau mais n'incluent pas la sécurité comme objectif.

Les attaques qui agissent sur la couche réseau peuvent être classées en quatre catégories selon leurs cibles. Dans la première catégorie, le mécanisme d'établissement de route est ciblé directement pour affecter la structure topologique du réseau. Les attaques de la deuxième catégorie

tentent de manipuler un lien ou un chemin établi entre deux ou plusieurs capteurs pour désactiver une partie ou la totalité du réseau. Dans la troisième catégorie l'altération de l'identité des nœuds est l'objectif de l'attaque Sybil, le clonage ou la fabrication de nœuds. La quatrième catégorie regroupe les attaques qui affectent les opérations à l'échelle du réseau telles que l'attaque d'inondation par paquet Hello[17], [19].

#### *3.2.3.1 Usurper (falsifier), modifier et relier les Informations de routage (Spoofed, altered and replayed routing information):*

C'est l'attaque la plus courante contre un protocole de routage. Dans un routage ad hoc non protégé, les informations de routage peuvent être affectées parce que chaque nœud agit comme un routeur. Donc, cette attaque cible les informations de routage échangées entre les nœuds et par conséquent les adversaires peuvent être en mesure de générer de fausses informations de routage afin de partitionner le réseau, créer des boucles de routage, attirer ou repousser le trafic réseau, étendre ou raccourcir les routes sources, générer de faux messages d'erreur et augmenter la latence de bout en bout[17], [19].

L'authentification est la solution standard pour contrer ce type d'attaques. Ainsi, seulement les informations de routage provenant de routeurs authentifiés sont acceptées.

#### *3.2.3.2 Attaque de routage sélectif des paquets et de trou noir (Selective Forwarding/Black hole)*

Les RCSF utilisent généralement une communication à plusieurs bonds pour collecter les données. Dans ce type de communication, on suppose que l'ensemble des nœuds participants à la communication sont légitimes. Cependant, un nœud malveillant sur le chemin de transmission de données peut acheminer des paquets et refuser de transférer d'autres afin de les empêcher d'atteindre leur destination.

Ce type d'attaques peut être plus dangereux lorsque l'attaquant cible des nœuds importants dans le réseau tels que les chefs de groupes ou les nœuds d'agrégation des données. Donc un nœud malveillant rejette uniquement les paquets des nœuds ciblés, et transfère normalement les paquets des autres nœuds ou les paquets non intéressants vers leur destination, ce qui lui permet de réduire la possibilité de détection de son comportement malicieux. Ainsi, un attaquant peut produire des collisions sur les canaux de transmission pour réunir les paquets ciblés qui passent au niveau de ses nœuds voisins, ce qui réussit effectivement une attaque sélective. Dans une attaque de trou noir, le nœud malveillant bloque le transfert de tous les paquets qu'il reçoit. Cette attaque est particulièrement efficace lorsque le nœud du trou noir est aussi un trou de puits. Une telle combinaison d'attaques peut arrêter tout le trafic de données autour du trou noir[18].

Plusieurs techniques ont été proposées, basées sur l'utilisation de multiples chemins pour assurer l'acheminement des données à leur destination [25]. D'autres techniques de défense

consistent à ajouter un numéro de séquence sur l'en-tête de chaque paquet de données, la vérification correcte et continue permet de détecter ce type d'attaques. Ainsi, les nœuds malveillants peuvent être écartés et ignorés le plutôt possible pour le transfert de messages en considérant un itinéraire alternatif[17], [19].

#### *3.2.3.3 L'attaque Sybil (Sybil Attack)*

Les attaques Sybil sont très typiques dans les RCSF si elles sont combinées avec d'autres attaques. Dans l'attaque Sybil, un nœud malveillant se comporte comme plusieurs nœuds légitimes, soit en fabriquant ou en volant les identités des nœuds légitimes (le nœud malveillant crée plusieurs identifiants légaux dans un emplacement ou forge plusieurs identifiants à différents endroits). Par conséquent, un adversaire peut être à différents endroits en même temps, ce qui cause de graves dommages à la pluparts des protocoles de routage et particulièrement aux protocoles de routage géographique qui se basent sur la localisation, où les nœuds échangent des informations de coordonnées avec leurs voisins pour construire le réseau.

Puisque le fraude d'identité mène à l'attaque Sybil, l'authentification qui nous permet la vérification d'identité des nœuds est indispensable pour remédier aux dégâts apportés par ce type d'attaques, mais les limitations de calcul et de stockage dues aux RCSF posent des contraintes sévères pour appliquer cette solution[17], [19].

#### *3.2.3.4 L'attaque de trou de puits (Sinkhole Attack)*

Dans ce type d'attaques, un nœud malveillant essaye de sembler le plus attrayant dans une zone cible pour tromper ses nœuds voisins afin d'être sélectionné comme nœud relais sur leur route. Par conséquent, les nœuds compromis peuvent attirer presque tous les flux de données dans des zones spécifiques (généralement plus proche de la station de base), empêchant ainsi les paquets correspondants d'atteindre leurs véritables destinations.

Ce type d'attaques est capable d'être combiné avec d'autres attaques telles que le routage sélectif, puisque il peut attirer tout le trafic provenant d'une zone importante dans le réseau. Ainsi, le nœud compromis ne transmet que les paquets sélectionnés.

L'attaque de trou de puits reste toujours difficile à défendre surtout dans les protocoles de routagequi se basent sur les informations de l'énergie ou l'estimation de la fiabilité de bout en bout pour construire une topologie de routage dans le réseau[17], [19].

#### *3.2.3.5 L'attaque de trou de ver (Wormhole Attack)*

Dans une attaque de trou de ver, deux nœuds malveillants sont connectés avec un lien direct à faible latence appelé lien wormhole. Avec le lien wormhole, l'adversaire peut capturer des transmissions de données d'un nœud voisin, les envoyer rapidement à l'autre nœud via le lien wormhole et rejouer ces transmissions de données []. Le cas le plus simple de cette attaque est

d'avoir un nœud malveillant transmettant des données entre deux nœuds légitimes. Les trous de ver convainquent souvent les nœuds éloignés qu'ils sont voisins, conduisant à l'épuisement rapide de leurs ressources énergétiques [].

Les solutions utilisées pour contrer ce type d'attaques se basent principalement sur la synchronisation temporelle et les informations de position pour préciser la localisation des nœuds. Avec des capteurs à ressources limitées, l'obtention de ce genre d'informations est couteuse, ce qui limite l'application de ce type de solutions[17], [19].

#### *3.2.3.6 L'attaque d'inondation par paquet Hello (Hello Flood Attack)*

De nombreux protocoles de communication exigent aux nœuds capteurs l'échange périodique des paquets HELLO pour la découverte de voisins. Ainsi, chaque nœud recevant un tel paquet peut supposer qu'il se trouve dans la couverture radio de l'expéditeur. Un nœud malveillant doté d'un émetteur puissant peut exploiter cette exigence pour tromper un grand nombre de nœuds en leur faisant croire qu'ils sont dans son voisinage. Ainsi, les nœuds capteurs tentent de transmettre leurs données au nœud attaquant qui peut être en dehors de leur portée radio. Par conséquent, il y aura un gaspillage d'énergie et une perte de données, ce qui perturbera le bon fonctionnement du réseau.

De nombreuses recherches ont été effectuées pour empêcher cette attaque en examinant la bidirectionnalité des liens locaux avant de les utiliser. Une autre défense partielle à cette attaque consiste à utiliser des protocoles de diffusion authentifiés par un tiers[17], [19].

#### *3.2.3.7 Usurpation d'accusé de réception (Acknowledgement spoofing)*

Cette attaque peut être montée contre les algorithmes de routage qui reposent sur des accusés de réception de la couche liaison. La transmission de paquets sur des liaisons de communication de faible qualité et multi-sauts permet à un adversaire d'usurper facilement les accusés de réception en encourageant le nœud cible à transmettre des paquets sur ces liens. Par conséquent, l'adversaire peut efficacement lancer une attaque de routage sélectif ou provoquer une perte de paquets.

L'utilisation d'une authentification correcte ainsi qu'un algorithme de cryptage efficace permet de contrer ce type d'attaques et renforcer les liens de communication faibles[14], [17].

### 3.2.4 Les attaques au niveau de la couche transport

La couche de transport fournie une gestion des connexions de bout en bout entre la source et la destination. Les protocoles qui conservent les informations de connexion sont vulnérables aux attaques d'inondation et de désynchronisation. Par contre, les protocoles de transport sans connexion sont protégés contre ce type d'attaques. Nous discutons ci-dessous deux types d'attaques possibles sur cette couche [26]:

#### *3.2.4.1 Inondation (Flooding)*

L'objectif de ce type d'attaques est de gaspiller les ressources des nœuds capteurs et d'arrêter la communication entre les nœuds. Pour ce faire, un adversaire envoie par exemple plusieurs demandes de connexion à un nœud victime sans connexion, écrasant ainsi le tampon de connexion, ce qui mène le nœud victime à refuser toute demande, y compris les requêtes authentiques de tous les nœuds du réseau.

La défense contre les attaques d'inondation consiste à limiter le nombre de demandes de connexion de chaque nœud, ainsi que de coder les paquets de type TCP SYN pour éviter de conserver l'état de connexion sur le serveur, mais les calculs et les frais généraux de ces techniques sont indésirables dans les RCSF[19], [26].

#### *3.2.4.2 Désynchronisation (De-synchronization)*

Pour réaliser cette attaque, un attaquant essaie de perturber une connexion active entre deux nœuds par la transmission de faux messages qui comportent des numéros de séquence modifiés ou des drapeaux de contrôle, en raison de la désynchronisation, les nœuds concernés retransmettent les paquets falsifiés, ce qui gaspille considérablement leur énergie.

Pour remédier l'effet de cette attaque, il est conseillé d'activer tous les champs de contrôle dans l'en-tête du paquet de transport, ainsi qu'une authentification des paquets échangés entre deux nœuds peut vaincre une telle attaque[19], [26].

### 3.2.5 Les attaques au niveau de la couche application

Cette couche est responsable de la collecte, le regroupement, la gestion et le traitement des données pour obtenir des résultats fiables, ainsi que d'assurer un flux d'informations fluide vers les couches inférieures. Un attaquant pourrait tenter de submerger les nœuds du réseau avec des stimuli de capteurs, ce qui amènerait le réseau à transférer de gros volumes de trafic vers une station de base. Une autre attaque de la couche application consiste à injecter des paquets parasites ou rejoués dans le réseau. Les objectifs des attaques contre cette couche est de consommer la bande passante du réseau et drainer l'énergie des nœuds[26].

Parmi les solutions proposées, on peut citer la révocation des nœuds compromis par l'utilisation d'un mécanisme de clé efficace tel que le protocole LEAP. D'autres solutions utilisent des techniques de limitation du débit et des algorithmes d'agrégation des données afin de réduire les effets de ces attaques. Aussi, la combinaison de l'authentification par paquets et la protection anti-rejoue (antireplay) empêchent ces attaques[19], [26].

# 4 Objectifs des attaques

## 4.1 Contre la confidentialité et l'authentification

Il existe plusieurs types d'attaques [27] dont l'objectif est de nuire aux techniques cryptographiques pour menacer le secret des données et l'authentification des nœuds. On peut classer ces attaques en deux catégories :

### 4.1.1 Attaque des nœuds répliqués (Node replication attack)

Dans l'attaque de réplication de nœuds, l'adversaire essaie de voler l'identité d'un nœud légitime existant déjà par la réplication de son identificateur, ce qui lui permet d'insérer un nœud malveillant dans l'ensemble du réseau. Par conséquent, un nœud malveillant bien placé peut facilement intercepter les clés cryptographiques, ce qui provoque de graves perturbations à l'ensemble du réseau[7].

L'utilisation d'un nœud central fiable pour la collecte des données tandis que la vérification des identités des nœuds sert à empêcher ce type d'attaques[4], [19].

### 4.1.2 Attaque sur la confidentialité (Attacks on privacy)

Garantir la confidentialité d'un grand volume d'informations sensibles qui circulant sur un support de communication sans fil est un défi particulièrement difficile dans les RCSF, ce qui permet aux intrus de collecter facilement des informations importantes, sans être physiquement présents grâce à l'accès à distance. Par conséquent, des données sensibles sur lesquelles se basent des décisions importantes peuvent être interprétées, ce qui menace la confidentialité et la vie privée du réseau. les attaques les plus communes contre la confidentialité des capteurs sont déjà présentées dans la section 3.1 des attaques passives, telles que le camouflage d'adversaires, l'écoute passive et l'analyse du trafic[4], [19], [28].

## 4.2 Contre l'intégrité des données

Pour menacer l'intégrité des données, un intrus utilise un nœud puissant pour compromettre, écouter et modifier les paquets transmissent entre les nœuds capteurs pour les rendre incomplets ou incorrectes, ce qui altère la sémantique des données. Ce type d'attaques peut aussi erroné les informations de routage pour perturber le bon fonctionnement du réseau et le rend éventuellement inutile, ce qui qualifie ce type d'attaques comme une d'attaque par déni de service. Comme la couche application est responsable des services visibles aux utilisateurs, les attaques contre l'intégrité des données ciblent généralement cette couche.

Le système de clés asymétriques et la signature numérique sont les mécanismes les plus efficaces pour lutter contre les attaques menaçant l'intégrité des données. Malheureusement, ces

mécanismes nécessitent beaucoup de message de contrôle supplémentaire (surcharger le réseau) qui les rendent non adaptés aux RCSF[14], [21].

## 4.3 Contre la disponibilité

L'ensemble des attaques qui menacent la disponibilité sont les attaques de type déni de service (DoS : Denial of Service). Elles font généralement référence à la tentative de perturber, corrompre ou détruire un réseau. Le déni de service peut être causé par un simple évènement empêchant ou diminuant le bon fonctionnement du réseau. Le déni de service le plus simple est d'empêcher le fonctionnement normal du capteur victime en lui envoyant un nombre important de messages inutiles, pour qu'il ne soit plus accessible par les autres utilisateurs[21].

Les contraintes physiques d'un capteur, son environnement de déploiement la nature du support de communication sans fil, rendent les RCSF plus vulnérables aux attaques de déni de service en comparaison aux autres types de réseau.

La plupart des solutions proposées pour protéger le réseau contre les attaques de type déni de service exigent beaucoup de ressources, ce qui dépasse les capacités des RCSF et limite l'utilisation de ces solutions.

Au niveau de toutes les couches du modèle OSI, deux catégories d'attaques peuvent être cités, à savoir l'attaque passive et l'attaque active. Ainsi, chaque couche possède ses propres DoS :

- le DoS peut se présenter au niveau de la couche physique comme une attaque "flooding" ou "Tampering".
- le DoS peut se présenter au niveau de la couche liaison comme une collision ou un "Jamming".
- le DoS peut se présenter au niveau de la couche réseau comme une attaque de trou de ver ou une attaque de trou noir.
- le DoS peut se présenter au niveau de la couche transport comme une attaque "flooding" malveillante ou une désynchronisation.

## 4.4 Attaque de drain d'énergie

Afin de drainer l'énergie des nœuds capteurs[29] l'attaquant doit être doté d'une antenne radio puissante ainsi que d'une source d'énergie permanente. Puisque les nœuds dans les RCSF sont alimentés par des batteries faibles (quantité d'énergie limitée), le déploiement est aléatoire et la topologie est dynamique. Ainsi, il est difficile de remplacer ou de recharger les batteries des nœuds capteurs. La raison pour laquelle, un attaquant peut utiliser des nœuds puissants pour injecter des rapports fabriqués dans le réseau ou générer une grande quantité de trafic sur le réseau. Les

rapports fabriqués provoquent de fausses alarmes qui vont gaspiller les efforts de réponse et drainer l'énergie des capteurs, ce qui rend toute une zone hors service [14]. La prise en conscience de la localisation et de l'adressage, peuvent constituer des solutions adéquates pour améliorer l'efficacité énergétique du réseau.

## 4.5 Contre l'agrégation des données

L'agrégation des données[30], [31] est une technique qui permet de réduire la quantité des données stockées ainsi que les transmissions redondantes dans les réseaux. Cette technique utilise une fonction d'agrégation sur les mesures provenant de nombreux nœuds capteurs, ce qui améliore l'efficacité énergétique et l'utilité de la bande passante. L'agrégation des données affecte certaines métriques de performance, elle peut augmenter le délai de transmission des données, dégrader l'exactitude des données collectées et augmenter la vulnérabilité de l'ensemble du réseau. L'agrégation des données dans les réseaux de capteurs est relativement triviale, mais devient problématique lorsque l'on veut y ajouter de la sécurité et plus particulièrement de la confidentialité[17].

Les objectifs de ce type de menaces est de falsifier les données des capteurs dans une zone ciblée, soit en injectant de fausses données ou en falsifiant le résultat d'une opération d'agrégation.

Généralement, Les nœuds agrégateurs doivent décrypter les paquets reçus pour les agréger et les crypter avant de les transmettre à leur destination, ce qui affecte la confidentialité des données ainsi que la procédure de décryptage-agrégation-cryptage, augmente la latence et entraine des surcharge de calcul supplémentaires. Si un nœud agrégateur est compromis, tous les résultats d'agrégation peuvent être modifiés ou altérés. Par conséquent, les performances et l'efficacité du réseau sont dégradées considérablement [21].

# 5 Conclusion

D'après ce qui a été présenté dans les deux chapitres précédents, nous constatons que La communication sans fil ouverte (les paquets diffusés sont accessibles pour tout le monde), les ressources limitées des nœuds capteurs (énergie, traitement, mémoire), l'environnement hostile (non surveillé), le déploiement aléatoire (aucune information sur l'infrastructure du réseau, la position et l'identification des nœuds n'est connue avant le déploiement), sont les principaux caractéristiques qui opposent les réseaux de capteurs sans fil à multiples types d'attaques de sécurité. Par conséquent, un adversaire bien équipé peut facilement exploiter une ou plusieurs sources de vulnérabilité pour monter une attaque passive ou active au niveau d'une ou plusieurs couches. Ces attaques partagent des objectifs à savoir : menacer l'intégrité des données, menacer la disponibilité, menacer l'agrégation des données ou drainer l'énergie des nœuds afin de nuire au bon fonctionnement du réseau. La sécurité d'un tel réseau est un aspect très important permettant d'élargir et varier l'utilisation des RCSF.

Plusieurs mécanismes de sécurité ont été proposés dans la littérature pour remédier à l'impact de ces attaques et assurer le bon fonctionnement du réseau afin d'accomplir les tâches affectées à ce dernier. Dans le chapitre suivant, nous discuterons les principaux mécanismes de sécurité dédiés aux RCSF.

# CHAPITRE 3 :

# LES MECANISMES DE SECURITE POUR LES RCSF

# 1 Introduction

Les réseaux de capteurs sans fil ont des applications étendues dans les domaines de la médecine, de l'armement militaire, la surveillance de l'environnement, la détection d'intrusions, etc. En raison de la variété et la popularité des RCSF, la nécessite d'intégrer des services de sécurité devient l'une des principaux préoccupations afin de poursuivre le succès des RCSF dans un certain nombre de domaines. Cependant, fournir des services de sécurité dans de tels réseaux s'avère être une tâche difficile en raison des limitations et les différentes vulnérabilités des RCSF présentées dans le premier et le deuxième chapitre respectivement, ces vulnérabilités font que les RCSF souffrent d'un nombre incroyable de menaces de sécurité. Ces caractéristiques et défis motivent une grande communauté de chercheurs à s'orienter vers la proposition des mécanismes de sécurité pour les réseaux de capteurs sans fil.

La recherches dans la sécurité a permis le développement de nombreux outils pour répondre à la question de sécurité dans les RCSF, ces derniers sont devenus indispensable pour développer n'importe quel protocole ou approche de sécurité, parmi lesquelles nous citons la cryptographie, la gestion de clés et l'authentification. En effet la réponse à la question de sécurité dépendent du besoin de savoir ce que nous allons protéger, en d'autres termes, on doit spécifier quels sont les objectifs de sécurité qu'on doit les satisfaire.

Plusieurs mécanismes, basés généralement sur la notion de cryptographie, sont mis en place afin de répondre à un ou plusieurs objectifs de sécurité dans les RCSF. Dans ce chapitre nous allons classer les mécanismes de sécurité on deux classes, à savoir les mécanismes de base (cryptographie, la gestion de clés et l'authentification) et les mécanismes avancés (modèle de confiance, les systèmes de détections d'intrusions).

# 2 Outils de base pour sécuriser les RCSF

## 2.1 La Cryptographie dans les RCSF

La cryptographie [32]–[35] est définie comme l'étude des techniques mathématiques dont l'objectif est de satisfaire des aspects de la sécurité de l'information tels que la confidentialité, l'intégrité des données, l'authentification des entités et l'authentification de l'origine des données. Les objectifs de la cryptographie sont la confidentialité, l'authentification, l'intégrité des données et la non-répudiation [34]. L'établissement [21] d'un système cryptographique basé sur des clés sécurisées est l'une des premières contre-mesures de sécurité dans les RCSF, ce qui permet aux nœuds capteurs de chiffrer et d'authentifier les messages communiqués entre eux. Les méthodes cryptographiques utilisées dans les réseaux RCSF doivent répondre aux contraintes des nœuds capteurs et la nature des communications sans fil, ainsi que d'être évaluées en fonction de la taille du code, de la taille des données, du temps de traitement et de la consommation d'énergie. Mais comme les nœuds capteurs sont limités dans leurs capacités de calcul et de mémoire, les

techniques cryptographiques traditionnelles bien connues ne peuvent pas être simplement appliqués aux RCSF sans les adapter. Par conséquent, pour satisfaire les contraintes de sécurité mentionnées ci-dessus, soit les techniques existantes doivent être adaptées, soit des techniques nouvelles doivent être développées. Sur la base des techniques cryptographiques existantes, Les systèmes cryptographiques sont généralement classés selon les trois aspects suivants: les techniques cryptographiques symétriques, les techniques cryptographiques asymétriques et les techniques cryptographiques hybrides. En raison des contraintes des nœuds capteurs, la sélection de la technique cryptographique appropriée est une tache essentielle et vitale dans les RCSF [33], [35].

Plusieurs mécanismes, basés généralement sur la notion de cryptographie, sont mis en place afin de répondre à la question de la sécurité dans les RCSF.

### 2.1.1 Cryptographie symétrique

C'est la plus ancienne technique et la seule disponible avant la publication de la cryptographie asymétrique en 1976.La cryptographie à clé symétrique est également connue sous le nom de cryptographie à clé partagée, à clé unique et à clé secrète. Deux primitives sont utilisées dans le cryptage symétrique la substitution et la transposition (permutation). En raison de sa facilité d'implémentation sur un matériel qui dispose de ressources (mémoire, calcul, énergie) limitées, La plupart des schémas de sécurités destinées aux RCSF utilisent uniquement la cryptographie à clé symétrique. Dans ce type de cryptage, une phase de pré-distribution de clés secrètes est obligatoire avant le déploiement des nœuds capteurs, donc l'expéditeur et le destinataire doivent partager la même clé secrète pour commencer à crypter et décrypter les messages entre eux en utilisant cette clé. Deux types de chiffrements symétriques sont utilisés [36]:

- Le chiffrement en chaîne est fait bit à bit sans attendre la réception entière des données. L'algorithme le plus connu est : RC4/RC5 (*Rivest Cipher 5*).
- Le chiffrement par bloc consiste à fractionner les données en blocs de taille fixe (64 bits, 128 bits). Chaque bloc sera ensuite chiffré une fois qu'il atteint une taille envisagée. Les algorithmes les plus utilisés sont : DES (*Data Encryption Standard*), AES (*Advanced Encryption Standard*) [6].

Le processus de pré-distribution de clés, qui permet de préserver la confidentialité de la clé secrète dans le réseau est une tache très difficile dans un environnement hostile où les RCSF sont utilisés. La cryptographie à clé symétrique ne peut pas garantir la non-répudiation. Etant donné que l'expéditeur et le destinataire utilisent la même clé, les messages provenant d'un utilisateur particulier ne peuvent pas être vérifiés.

Les contraintes sur le calcul et la consommation d'énergie dans les RCSF favorisent l'application de la cryptographie à clé symétrique. La plupart des études et recherches faites sur les algorithmes à clé symétrique dans les réseaux de capteurs se concentrent essentiellement sur les

schémas de cryptage populaires, RC4, RC5, IDEA, DES, 3DES (Triple DES), AES. L'évaluation de ces schémas permis de conclure les principaux facteurs de performance de la cryptographie à clé symétrique suivants:

- Largeur du bus de données embarqué: de nombreux algorithmes de chiffrement préfèrent l'arithmétique des mots de 32 bits, mais la plupart des processeurs embarqués utilisent généralement un bus de données de 8 ou 16 bits [23].
- Jeu d'instructions: l'instruction ISA (Instruction Set Architecture) a des effets spécifiques sur certains algorithmes. Par exemple, la plupart des processeurs embarqués ne supportent pas le bit variable instruction de rotation comme ROL (rotation des bits à gauche) de la L'architecture Intel, qui améliore considérablement les performances de RC5 [23].

#### *2.1.1.1 DES (Data Encryption Standard)*

DES est l'approche traditionnel de la cryptographie symétrique, inventé en 1977 par le NBS (National Bureau of Standards). Il s'agit d'un système de chiffrement symétrique par blocs qui permet de chiffrer des mots de 64 bits à partir d'une clef de 56 bits (56 bits servant à chiffrer + 8 bits de parité servant à vérifier l'intégrité de la clef en réalité). Chaque bit de parité de la clé (1 tous les 8 bits) sert à tester un des octets de la clé par parité impaire, c'est-à-dire que chacun des bits de parité est ajusté de façon à avoir un nombre impair de '1' dans l'octet à qu'il appartient. La clé possède donc une longueur « utile » de 56 bits, ce qui signifie que seuls 56 bits servent réellement dans l'algorithme.

Généralement les grands étapes de l'algorithme DES sont les suivantes [37] :

- Division du texte en blocs de 64 bits (8 octets) ;
- Permutation initiale des blocs ;
- Fractionnement des blocs en deux parties: gauche et droite G et D ;
- Etapes rondes sert de répétées le processus de permutation et substitution 16 fois ;
- Recollement des parties gauche et droite puis permutation initiale inverse.

Le déchiffrement DES est identique au chiffrement, à condition de prendre dans l'ordre inverse les sous clés.

L'utilisation du DES est assez limitée en raison du fait qu'il peut être cassé facilement. La recherche exhaustive ($2^{56}$) devient réaliste, L'Electronique Frontier Foundation exhibe en 1998 une machine craquant le DES en 9 jours maximum, 64 bits est devenu court et présente des risques d'attaque. Même la solution double DES na pas pu résister lentement, ainsi que le 3DES offre une

sécurité convenable, mais l'algorithme est trois fois plus lent que le DES et la taille du bloc subsiste pose aussi un autre problème [5].

#### *2.1.1.2 AES (Advanced Encryption Standard)*

En octobre 2000 la NIST (National Institute of Standards and Technology) élit Rijndael comme nouveau standard qu'on nomme aussi AES (Advanced Encryption Standard), après un appel d'offre international lancé en janvier 1997 pour remplacer le vieillissant DES. L'algorithme belge Rijndaël (à prononcer raindal), proposé par Vincent **Rij**men Joan **Dae**men et, est un algorithme de chiffrement par blocs à plusieurs tours similaire à DES mais avec une taille de blocs et de clefs supérieures et variables, choisis entre 128, 196 et 256 bits [38].

L'AES opère sur des blocs rectangulaires de 4 lignes et *Nc* colonnes, dont chaque terme $x_{i;j}$ (appelé octet ou byte) est composé de 8 bits ($b = b_7b_6b_5b_4b_3b_2b_1b_0$), et peut être vu algébriquement comme un polynôme de degrés inferieur ou égale à 7 avec des coefficients dans $F_2$ (0 ou 1) [38] :

$$b = b_7X^7 + b_6X^6 + b_5X^5 + b_4X^4 + b_3X^3 + b_2X^2 + b_1X + b_0$$

**Équation 1 : Polynôme de degré inférieur ou égale à 7**

Le nombre de tours dans l'AES varie suivant la taille des blocs et de la clef. Chaque tour utilise une sous-clef ki différente, Nous décrivons ainsi les principaux étapes de chaque opération de ce chiffrement [38], [39] :

- SubBytes (passage dans un S-box) : La fonction SubBytes effectue une inversion dans le groupe $GF(2^8)$, suivie d'une application affinée, définie par :

$$b_i = b_i \oplus b_{(i+4) \bmod 8} \oplus b_{(i+5) \bmod 8} \oplus b_{(i+6) \bmod 8} \oplus b_{(i+7) \bmod 8} \oplus c_i \qquad \text{Pour } 0 < i < 8$$

**Équation 2 : Fonction SubBytes**

- ShiftRows (décalage de ligne) : Cette opération effectue une simple permutation circulaire à gauche par ligne de chacun des octets du bloc. La première ligne sera inchangée. Dans la seconde, chaque octet sera décalé d'un cran à gauche (le premier prenant la place du dernier). Dans la troisième, chaque octet sera décalé de 2 crans à gauche, et dans la quatrième, chaque octet sera décalé de 3 crans à gauche.

- MixColumns (mélange des colonnes) : Cette procédure effectue un "mélange" à l'intérieur de chaque colonne. Chaque octet de la colonne (sous forme polynômiale) est multiplié par le polynôme $a(X) = 3X^3 + X^2 + X + 2$, modulo $X^4 + 1$. Les coefficients sont ensuite réduits modulo 2.

- XorRoundKey : L'opération XorRoundKey effectue simplement l'addition modulo 2 avec la section de la clé étendue correspondant au tour *r* durant lequel s'effectue la transformation.

- Calcul de la clé étendue : La clé étendue est de la forme : k0k1k2k3k4…où chaque kc est une matrice 4 lignes et 1 colonne. Pour le XorRoundKey, on prendra Nc colonnes kc différentes à chaque tour.

Avant de commencer le chiffrement, un calcul de la clé étendue se fait une fois pour toute. Un XorRoundKey est effectué avec le message clair et la clé de chiffrement avant d'effectuer le premier tour. De plus, le dernier tour ne contient pas l'opération MixColumns. Ainsi que les mêmes étapes sont suivies pour le déchiffrement [39].

L'AES offre un gain de sécurité sans perte de performances ce qui lui permet d'être totalement sûr et opérationnel sur tout type d'environnement. AES prouve sa résistance à la cryptanalyse linéaire et différentielle, puisqu'une recherche exhaustive de la clé n'est absolument pas envisageable en un temps limité (on parle de près de 149 milliards d'années) et aucune attaque ne lui est connue à ce jour. Il est très efficace en termes de ses besoins en ressources (mémoires, traitement, énergie) qui sont également très faibles. Cela induit la possibilité d'utilisé AES dans une grande variété de plateformes et d'applications. L'implémentation d'AES peut être aussi bien sous forme logicielle que matérielle (câblé) [39].

Le temps d'exécution (de l'ordre de plusieurs millisecondes) et la limite de surcoût énergétique de chiffrement et déchiffrement de données ont poussé l'algorithme AES d'être proposé comme l'algorithme de cryptage le plus adapté aux RCSF, pour sécuriser les données transmises au niveau de la couche MAC.

AESCTR (mode compteur d'opération cryptographique avec AES) signifie que le mode CTR utilise AES comme code de bloc; et fournit un contrôle d'accès, un cryptage de données et une fraîcheur séquentielle optionnelle. L'authentification est effectuée en utilisant le bloc de chiffrement chaîné avec le code d'authentification de message (CBC-MAC) [40].

### 2.1.2 Cryptographie asymétrique

La cryptographie asymétrique [36], également appelée cryptographie à clé publique, a été introduite en 1976 par Diffie et Hellman. Dans les systèmes asymétriques, le principe de base est d'utiliser deux clés séparées non reconstructibles l'une à partir de l'autre pour rendre les clés de chiffrement et de déchiffrement différentes, non reconstructibles l'une à partir de l'autre. La première clé est secrète, elle peut être utilisée pour déchiffrer et signer des données, donc elle doit être gardée privée tandis que la deuxième clé est publique (connue publiquement), elle peut être utilisée pour crypter et vérifier des données. Le point le plus délicate sur lequel repose la sécurité du chiffrement asymétrique est que toute opération effectuée avec la clé secrète ne peut être inversée qu'avec la clé publique, et inversement, donc la connaissance d'une clé ne permettrait pas à une personne de trouver l'autre. Les exigences pour la réalisation de la cryptographie à clé publique [18], [34] données par Diffie et Hellman sont:

- Un algorithme qui peut facilement calculer et générer une paire de clés, c'est-à-dire une clé privée KS et une clé publique KP;
- Un message clair M peut être facilement crypté en un message chiffré MC en utilisant la clé publique KP;
- Le message chiffré MC peut être facilement déchiffré au message clair M en utilisant la clé privée KS;
- Le message chiffré MC ne peut pas être déchiffré au message clair M en utilisant la clé publique KP;
- Un adversaire connaissant la clé publique KP ne peut pas déterminer la clé privée KS.

La cryptographie à clé publique repose sur l'arithmétique des nombres entiers tel que le modulo la multiplication le décalage la soustraction et l'addition, ce qui exige beaucoup de ressources pour réaliser les opérations de chiffrement et de déchiffrement. Les techniques cryptographiques asymétriques peuvent en outre être classées en trois classes: techniques de base RSA, techniques basées sur ECC et techniques basées sur l'appariement [33].

Dans ces algorithmes, différents problèmes mathématiques se présentent à cause des calculs intensifs utilisés pour chiffrer et déchiffrer les données. La complexité de telles opérations est indésirable dans les RCSF à cause de la capacité de traitement plus élevée et la plus haute dissipation d'énergie. En utilisant le chiffrement asymétrique, chaque nœud capteur souffre d'un autre problème dû au stockage de clés publiques de tous les nœuds restant du réseau [6]

Cela provoque une forte occupation des mémoires au niveau de chaque nœud. Cependant, la distribution de clés est moins pénible car leur échange est fortement simplifié. En effet, avec un système asymétrique, chaque nœud a besoin d'une paire de clés. Si on considère que le nombre de nœuds dans le réseau est égal à n, il faudra donc gérer 2.n clés [6].

L'efficacité d'un l'algorithme de cryptage est principalement déterminée par le nombre de cycles d'horloge requis pour exécuter une instruction de multiplication sur un microprocesseur. Les algorithmes de clé publique tels que RSA exécutent généralement des milliers ou même des millions d'instructions de multiplication pour effectuer une seule opération de sécurité. Ainsi que les algorithmes à clé publique requièrent généralement de l'ordre de quelques dizaines de secondes et jusqu'à quelques minutes pour effectuer des opérations de chiffrement et de décryptage dans des dispositifs sans fil [23].

Malgré que la cryptographie à clé publique à été écarté et considéré comme étant irréalisable dans les RCSF pendant un certain nombre d'années, à cause de la taille du code, la taille des données, le coût de traitement et la consommation d'énergie [32]. Des études récentes ont montré qu'il est possible d'appliquer la cryptographie à clé publique aux réseaux de capteurs en

basant sur la bonne sélection d'algorithmes avec les paramètres associés, l'optimisation et les techniques de faible puissance de calcul et d'énergie. La plupart des études dans la littérature se concentrent sur les algorithmes RSA et ECC. L'attirance d'ECC est qu'il semble d'offrir un niveau de sécurité pareil aux : schéma de Rabin, Ntru-Encrypt et RSA mais avec une taille de clé beaucoup plus petite, réduisant ainsi les frais généraux de traitement et de communication. Par exemple, RSA avec 1024 bits (RSA-1024) fournit un niveau de sécurité actuellement accepté pour de nombreuses applications est équivalent à ECC avec des clés de 160 bits (ECC-160) [23].

[41] Gura et al. rapportent que la cryptographie RSA et la cryptographie à courbe elliptique sont possibles en utilisant des CPU 8 bits avec ECC, ce qui démontre un avantage de performance par rapport à RSA. Un autre avantage est que les clés de 160 bits d'ECC résultent en des messages plus courts pendant la transmission par rapport aux clés RSA de 1024 bits. En particulier Gura et al, ont démontré que les opérations de multiplication ponctuelle dans ECC sont d'un ordre de grandeur plus rapide que les opérations à clé privée au sein de RSA, et qu'elles sont comparables (quoique plus lentes) à l'opération de clé publique RSA [5], [41].

#### *2.1.2.1 RSA (Rivest-Shamir-Adleman)*

RSA [42] est le premier système cryptographique à clé publique, Il a été inventé par Ron **R**ivest, Adi **S**hamir et Leonard **A**dleman en 1977.RSA est l'une des technologies de cryptage à clé publique les plus populaires actuellement disponibles. RSA s'appuie sur sa force due à la facilité de multiplier deux grands nombres mais il est difficile de factoriser le produit de grands entiers, et une fonction "puissance" à sens unique. Etant donné deux grands nombres entiers *p* et *q*, il est facile de trouver *N*, tel que N est le résultat de multiplication de *p* et *q* (*N*=*p.q*). A l'inverse, étant donné *N*, il très difficile de factorisé *N* pour retrouver *p* et *q* [3].

Pour faire une explication imagée, imaginons que Bob souhaite recevoir d'Alice des messages en utilisant RSA, L'algorithme fonctionne de la manière suivante [38] :

- **génération des clefs** :
  - p et q, deux grands nombres premiers sont générés au hasard grâce à un algorithme de test de primalité probabiliste, avec n = pq.
  - Un nombre entier e premier avec (p-1)(q-1) est choisi. Deux nombres sont premiers entre eux s'ils n'ont pas d'autre facteur commun que 1.
  - L'entier d est l'entier de l'intervalle [2, (p-1)(q-1)[ tel que ed soit congrue à 1 modulo (p-1)(q-1), c'est-à-dire tel que ed-1 soit un multiple de (p-1)(q-1).
- **distribution des clefs** : le couple (n, e) constitue la clef publique de Bob. Il la rend disponible à Alice en lui envoyant ou en la mettant dans un annuaire. Le couple (n, d) constitue quand à lui sa clef privée.

- **chiffrement du message** : Pour crypter le message Alice représente le message sous la forme d'un ou plusieurs entiers M compris entre 0 et n-1. Elle calcule C= Me mod n grâce à la clef publique (n, e) de Bob et envoie C à Bob.

- **déchiffrement du message** : Bob reçoit C et calcule grâce à sa clef privée Cd mod n.

Il obtient ainsi le message initial M.

Le système RSA permet de préserver la taille du message, ce qui le favorise d'être pratique pour assurer la confidentialité et l'authentification sur les liaisons de communication par le biais de signatures numériques. Pendant plusieurs années, certains chercheurs pensent que nombreuses techniques de sécurité sont lourdes pour les réseaux de capteurs et que de nouvelles alternatives doivent être développées, ce qui conduit à de nouvelles recherches intéressantes. Westoff et al. [43] démontrent qu'avec une conception soignée, le système de cryptage à clé publique RSA peut être utilisée même sur les dispositifs de réseau de capteurs les plus limités en ressources [44].

[45], Watro et al. Prouvent que nous pouvons appliquer avec succès certaines parties du système cryptographique RSA à des capteurs sans fil réels, en particulier les MICA2 [46]. Ils ont mis en œuvre les opérations publiques sur les capteurs eux-mêmes, ainsi que les opérations privées sont délocalisé vers des dispositifs mieux adaptés aux tâches de calcul intensifs [5].

#### *2.1.2.2 ECC (Elliptic Curve Cryptography)*

ECC a été inventé en 1985 par Koblitz et Miller, il fait partie des systèmes cryptographie à clé publique, et basée sur les mathématiques des courbes elliptiques. Les courbes elliptiques utilisées en cryptographie sont typiquement définies sur deux types de champs finis: les champs premiers Fp, où p est un grand nombre premier, et les champs d'extension binaires F2m [47]. ECC est calculé par la multiplication ponctuelle sur des courbes elliptiques sur des champs entiers premiers ou des champs polynomiaux binaires. ECC tire sa puissance de sécurité du problème de logarithme discret de la courbe elliptique (ECDLP). Soit *P* un point sur une courbe elliptique, et *Q=dP* (où *d* est un entier) aussi un point de la courbe elliptique, alors *d* est le logarithme discret de *Q* en base *P*. Etant donnés *Q* et *P* deux points d'une courbe elliptique, il est extrêmement difficile de trouver *d* tel que *Q=dP* : c'est le problème de logarithme discret elliptique [3].

Comme les nœuds capteurs dans les RCSF sont dotés par des processeurs lents, l'implémentation de l'ECC sur les RCSF est principalement basée sur les champs de nombres entiers car les mathématiques de champ polynomiales binaires ne sont pas supportées par ce type de processeurs [34], [48]. De la discussion ci-dessus il est clair que la cryptographie à courbe elliptique est complexe, ce qui nécessite de résoudre un certain nombre de problèmes pratiques avant l'intégration de la cryptographie à courbe elliptique dans un système de sécurité des réseaux de capteurs sans fil. Il faut tenir compte des frais généraux liés aux performances en termes de temps,

de mémoire et d'occupation de bande passante pour l'utilisation de l'authentification et du chiffrement / déchiffrement dans les applications RCSF.

En raison de la petite taille de clé et la possibilité d'implémentation pratique dans les dispositifs à ressources limitées, ECC a attiré l'attention des chercheurs pour intégrer ce dernier dans les RCSF. Une étude d'ECC sur les RCSF est présentée dans [49]. Leurs résultats indiquent également que les algorithmes de clé publique sont un bon choix pour une utilisation dans la mise en réseau de capteurs sans fil, et que les avantages des clés et des certificats ECC plus petits seront importants pour améliorer la conservation de l'énergie. ECC est utilisé pour réaliser l'authentification [50] et la gestion des clés. Une implémentation d'ECC basée FPGA pour les RCSF est donnée dans [51].

Cependant la multiplication modulaire est la partie la plus critique dans ECC ainsi que dans tous les algorithmes cryptographiques de clé publique. Malgré que les mathématiciens ont apporté quelques modifications pour réduire le temps de calcul mais la multiplication modulaire reste l'élément individuel qui consomme beaucoup de temps [48].

### 2.1.3 Fonctions de hachage

Cette fonction calcule une courte empreinte de taille fixe à partir d'une donnée de taille arbitraire. Une fonction de hachage cryptographique est basée sur l'exécution d'une fonction de compression unidirectionnelle à partir d'un bloc de données de n'importe quelle taille vers une sortie de longueur fixe n. Cela se produit dans plusieurs tours, et la taille de sortie n implique le niveau de sécurité fourni par la fonction de hachage. Une fonction de hachage devrait être facilement calculable et publiquement connue, ce qui rend les implémentations matérielles et logicielles pratiques. Etant donnée une fonction de hachage f, et un message à transmettre m. La fonction f doit remplir les conditions suivantes [6], [18] :

- Il est facile de calculer f(m), c'est-à-dire, de calculer l'empreinte à partir du contenu du message.
- Il est difficile de calculer m tel que f(m) = f, c'est-à-dire, de trouver le contenu du message à partir de l'empreinte. C'est pourquoi la fonction f est dite « à sens unique ».
- Il est difficile de trouver un autre message m2 tel que f(m) = f(m2), c'est-à-dire, il est difficile de trouver deux messages aléatoires qui donnent la même empreinte et cela mène à la résistance aux collisions. Cette empreinte est recalculée par le récepteur (2) afin qu'il la compare à celle calculée par l'émetteur. Si elles sont différentes (3), alors les données ont été altérées pendant leur transmission [6].

Le but d'utilisation d'une fonction de hachage est de rendre difficile le retour en arrière, donc un adversaire ne puisse pas déduire le message en claire à partir de son empreinte. Une fonction de

hachage efficace est une fonction qui permet d'obtenir un changement significatif de l'empreinte lors d'un petit changement du texte initial [52].

Les fonctions de hachage peuvent utiliser un chiffrement par bloc ou des opérations modulaires comme fonction de compression. Cependant, l'utilisation d'un chiffrement par bloc est moins efficace en raison du traitement clé - sous clé. Les fonctions de hachage les plus courantes sont: la famille MD (message digest), représentée par MD2, MD4 et MD5; la famille de l'algorithme de hachage sécurisé (SHA), représentée par SHA-0, SHA-1, SHA-256, SHA-384 et SHA-512 ; la famille RIPEMD (primitive d'évaluation des primitives d'intégrité RACE), représentée par RIPEMD, RIPEMD-128 et RIPEMD-160; et d'autres comme HAVAL et Whirlpool. MD5 effectue un seul passage sur les données et génère un résumé de 128 bits. Puisque l'empreinte est supposée d'être unique, MD5 n'est plus considéré comme sûr, Il a été prouvé plusieurs fois que MD5 ne peut pas satisfaire la propriété de résistance aux collisions, c'est-à-dire deux messages au contenu aléatoire et différents ayant la même empreinte. En 2006, un algorithme permettant de trouver des collisions pour le MD5 en quelques minutes en utilisant un simple ordinateur portable a été publié (Klima, 2006) [18], [52].

## 2.2 Authentification

Afin de fournir une construction sûre du réseau de capteurs, l'authentification [53] est indispensable pour de nombreuses tâches administratives (par exemple, la reprogrammation du réseau, le cycle de service du nœud capteur puits, garantir la légitimité des nouveaux nœuds insérés) et de nombreuses applications dans les réseaux de capteurs, de ce qui précède, l'authentification est considérée comme l'un des mécanismes de base pour la sécurité dans les réseaux de capteurs sans fil. Un adversaire peut changer le flux de paquets entier en injectant des paquets supplémentaires ou usurper une adresse source puis injecter du contenu contrefait ou de renvoyer le même contenu dans le réseau afin de camoufler son originalité. Le destinataire doit donc s'assurer que les données utilisées dans tout processus de prise de décision proviennent de la source correcte. L'authentification est une assurance sur les identités des nœuds ou des principaux communicants dans n'importe quel réseau, donc elle peut prouver au nœud récepteur que les données ou les paquets de contrôle (les informations de routage, de localisation et de gestion de clés) proviennent bien de la bonne source [5], [21].

L'authentification fonctionne à deux niveaux: au niveau de l'entité appelée authentification de l'identité et l'autre au niveau des données appelé authentification des messages, également appelée intégrité des données. Dans les réseaux de capteurs sans fil, l'authentification peut être effectuée entre deux nœuds communiquant ou un nœud (par exemple, un chef de groupe) et plusieurs autres nœuds qui participent à la communication autour de ce nœud (c'est-à-dire une authentification de diffusion). Les propositions précédentes [23] pour une diffusion authentifiée ne sont pas pratiques dans les RCSF pour les raisons suivantes:

- La plupart des propositions reposent sur la cryptographie à clé publique pour l'authentification. Cependant, la cryptographie à clé publique n'est pas pratique pour les RCSF;

- Même les schémas de signature à usage unique basés sur la cryptographie à clé symétrique ont trop de frais généraux.

µTESLA et ses extensions ont été proposées pour fournir une authentification de diffusion pour les réseaux de capteurs.

Plusieurs approches d'authentification ont été proposées dans la littérature. L'approche la plus [1] simple se base sur un mot de passe. Le nœud envoie un mot de passe avec ses informations de connexion. Le récepteur vérifie que le nœud est un nœud légitime en vérifiant que le mot de passe est associé au nœud expéditeur. L'autre approche se basée sur la cryptographie. Dans les RCSF, l'authentification est principalement basée sur ce que vous savez. Les connaissances secrètes seront liées aux clés ou à tout matériel cryptographique utilisé par les identités pour prouver leur caractère unique. Ainsi, la génération de clés et leur distribution constituent l'étape la plus importante et la base de tous les services de sécurité. Une technique classique pour fournir l'authentification consisterait à utiliser le code d'authentification de message MAC (*Message Authentication Code) qui* fait partie des fonctions de hachage à clé symétrique assurant l'intégrité de données comme toute autre fonction de hachage, en plus, l'authenticité de la source de données. Cette clé est utilisée pour calculer le code MAC par l'émetteur (1). Ce code est par la suite envoyé avec les données (2). Le récepteur calcule à son tour le code MAC avec cette même clé et le compare au code qu'il a reçu (3). S'ils sont bien identiques (4), alors la source est authentique et les données n'ont pas été altérées. Dans la pratique, HMAC *(keyed-Hash Message Authentication Code)* est utilisé [6], [54].

Dans ce qui suit nous présentons les principaux protocoles d'authentification proposés dans la littérature :

### 2.2.1 Le protocole SPIN (Security Protocols for Sensor Networks)

SPINS est considéré comme le premier protocole de sécurité qui répond aux différentes exigences de sécurité sans fil. SPINS propose deux protocoles de sécurité pour sécuriser les canaux de communication. Ces deux protocoles sont "SNEP et µTESLA.

#### *2.2.1.1 Le protocole µTESLA (Micro Timed Efficient Stream Loss-tolerant Authentication)*

µTESLA est un protocole de diffusion authentifié. µTESLA est le résultat de l'amélioration et l'adaptation du protocole TESLA [55] pour assurer l'authentification des communications par diffusion dans les réseaux de capteurs. µTESLA est utilisé pour sécuriser les informations de routage, les messages d'agrégation de données, etc. µTESLA suppose que les récepteurs sont tous synchronisés avec l'expéditeur et que chaque nœud connaisse une limite supérieure de l'erreur de

synchronisation maximale. Le schéma adopte une fonction de hachage à sens unique et utilise les pré-hachages comme clés dans un code d'authentification de message (MAC) [23], [56]. µTESLA introduit l'asymétrie à travers une divulgation retardée des clés symétriques. Cette technique de divulgation retardée est utilisée pour toute la chaîne de hachage et exige donc des horloges vaguement synchronisées entre les nœuds puits et capteurs. µTESLA est ensuite amélioré pour surmonter la limite de longueur de la chaîne de hachage. Pour prendre en charge le scénario multiutilisateur µTESLA est également étendu. même dans le scénario mono-utilisateur les schémas de type µTESLA présentent généralement les défauts suivants:

- Tous les récepteurs doivent mettre en tampon mémoire tous les messages reçus dans un intervalle de temps;
- Ils sont soumis à des attaques Wormhole, en raison du délai de propagation des clés divulguées des messages pourraient être falsifiés. En outre, dans des réseaux de capteurs multi sauts nous pouvant signaler une vulnérabilité plus sérieuse des schémas de type µTESLA [57].

µTESLA peut être très efficace et fournit des solutions pratiques pour les réseaux de capteurs, en utilisant uniquement des primitives cryptographiques symétriques avec une précision de synchronisation dans la plage de µs.

#### *2.2.1.2 Le protocole SNEP*

Le protocole SNEP [58] fournit: la confidentialité des données, l'authentification des communications unicast (*entre une paire de nœuds)*, l'intégrité, la protection contre la répétition, la faible fraîcheur des messages.

Pour crypter les messages et générer les codes d'authentification de message (MAC) SNEP utilise une version allégée de RC5. Pour le cryptage, SNEP utilise une fonction F pseudo-aléatoire FK (X) = MAC (K, X) pour générer des clés séparées et indépendantes, et une fonction unidirectionnelles pour amorcer les clés, par conséquents si une clé est compromise, les deux parties peuvent dériver une nouvelle clé sans transmettre des informations confidentielles.

SNEP [58] présente un certain nombre d'avantages intéressants qui le rend efficace et attirant. Premièrement, il ajoute seulement 8 octets par message ce qui lui permet de réduire le coût de communication; Deuxièmement, un compteur maintenu à la fois par l'expéditeur A et le récepteur B. Après chaque transmission réussie, le compteur est incrémenté à la fois par A et B. Alors l'état aux deux extrémités est conservé sans transmettre la valeur du compteur. Troisièmement, SNEP atteint une sécurité sémantique, une propriété de sécurité forte qui empêche les intrus de déduire le contenu du message crypté. Ce protocole simple et efficace offre aussi l'authentification des données, il permet d'éviter de rejouer les messages et propose une faible fraîcheur des messages.

### 2.2.2 Le protocole TinySec *(Tiny security)*

TinySec est considéré comme le premier protocole de sécurité destiné a la couche liaison de données introduit par Karlof et al. L'objectif était de fournir un protocole de sécurité qui permet de garantir l'authenticité, la confidentialité et l'intégrité des données, tout en préservant la puissance de calcul, l'espace de stockage et la bande passante. Pour se faire, TinySec se base sur l'utilisation du code d'authentification de message (MAC). TinySec opère en deux modes d'opérations différents : le premier mode de fonctionnement appelé TinySec_Auth qui fournis un mécanisme dédié uniquement pour l'authentification de message basé sur le code d'authentification de message (MAC), ainsi que le deuxième mode TinySec_AE qui prend en charge l'authentification et la confidentialité des messages via le cryptage en mode CBC (Cipher Block Chaining). Du point de vue architectural, TinySec a également pour objectif d'utiliser une conception plug-and-play pour permettre de modifier les éléments de base de l'architecture de sécurité, à savoir le chiffrement utilisé ou le mode de fonctionnement chiffré utilisé [59], [60].

Le seul point de défaillance de Tinysec se réside dans la gestion des clés puisque il se base sur une seul clé pour générer des codes MAC.

### 2.2.3 Le protocole MiniSec *(Mini security)*

MiniSec [61] est un protocole de couche réseau qui fournit un niveau de sécurité élevé a un faible coût d'énergie. MiniSec offre deux modes de fonctionnement, le premier est destiné à l'authentification de la communication à source unique (unicast) qui s'appelle MiniSec-U, et le second est destiné à l'authentification de communication multi-sources (multicast). MiniSec est basé sur le chiffrement par bloc OCB (Offset Code Book) comme méthode de chiffrement, car il offre un chiffrement authentifié avec un seul passage sur les données du message ainsi que le texte chiffré à la même longueur que le texte en clair, ce qui est considéré particulièrement adapté aux contraintes d'énergie strictes des nœuds capteurs. En plus de l'OCB, MiniSec utilise également le Bloom filtres et une synchronisation temporelle pour minimiser la consommation d'énergie.

Les choix de configuration optée par les concepteurs de MiniSec on permet de réussir à atteindre le tiers de l'énergie consommée par le protocole TinySec-AE tout en atteignant un niveau de sécurité correspondant à celui de Zigbee. Ce qui qualifie MiniSec d'être parmi les protocoles d'authentification les plus efficaces et adapter aux RCSF.

## 2.3 La gestion de clés dans les RCSF

La gestion des clés fournit des mécanismes fiables, sécurisés et efficaces par lesquels les clés cryptographiques sont générées, stockées, protégées, transférées, chargées, utilisées et détruites. Par conséquence, la gestion de clés est un service primordial pour la sécurité de n'importe quel système basé sur la communication. Sous les contraintes strictes et sévères posés par les RCSF, la

conception d'un système de gestion de clés est un grand défis. Sélectionner une solution cryptographique appropriée pour les RCSF est un autre défis [62].

D'une manière générale, la gestion des clés dans les RCSF peut être décomposée selon les phases suivantes:

### 2.3.1 Phase d'établissement des clés (key establishement)

Bien que la cryptographie à clé publique comporte des avantages certains par rapport à la cryptographie à clé symétrique et malgré qu'elle offre une meilleure résistance aux attaques de compromission de nœud ainsi que les recherches qui visent à les appliquer aux RCSF, la cryptographie à clé symétrique possède ses propres qualités qui la rend toujours la plus préférée pour les RCSF. Pour ce la et principalement en raison de sa consommation d'énergie raisonnable la plupart des solutions de gestion de clés existantes sont basées sur la cryptographie symétrique.

En cryptographie symétrique, la source et la destination utilisent la même clé pour crypter et décrypter les messages. La cryptographie asymétrique implique l'utilisation d'une paire de clés (clé publique et clé privée) pour crypter et décrypter les messages. La cryptographie asymétrique offre une meilleure résistance aux attaques et permet une évolutivité, mais nécessite une partie supplémentaire sur le logiciel et le matériel des nœuds. Certains chercheurs ont étudié des outils cryptographiques asymétriques et proposent des solutions adaptées telque le Tiny Public Key (TinyPK) [9] et Tiny Elliptic Curve Cryptosystem (Tiny ECC) [63].

### 2.3.2 Phase de pré distribution des clés (Key predistribution)

Dans le cas des RCSF, où la topologie du réseau est inconnue qu'après le déploiement des nœuds, une phase de pré distribution des clés qui consiste de chargées les clés dans les nœuds capteur avant le déploiement, est le seul moyen sûr et efficace qui permet aux nœuds communicants de partager les clés secrètes d'une manière sécurisée.

Les clés stockées dans la mémoire avant le déploiement constituent le porte-clés (Key ring). S'il existe une clé commune entre deux nœuds, ils peuvent créer une connexion sécurisée entre eux. La solution optimale en termes de ressources est de pré chargé tous les nœuds par une seule clé secrète. Ce qui provoque une très faible résilience, alors un adversaire puissant peut capturer un nœud et compromettre la clé très facilement. Pour surmonter ce problème, une autre solution consiste à utiliser une paire de clés distinctes pour toutes les paires de nœuds possibles dans le réseau. Cependant, deux problèmes qui se posent dans les schémas traditionnels de pré distribution de clés, dont le premier est de savoir comment charger un ensemble de clés dans la mémoire limité de chaque capteur. Le second problème inclue la sauvegarde de l'identifiant clé parmi un ensemble de clés et l'association de l'identifiant du nœud avec un nœud de contrôleur de confiance [64].

Le problème de distribution de clés est traité de manières déférentes selon le type du réseau : les réseaux de capteurs sans fil distribué, les réseaux de capteurs sans fil hiérarchique et les réseaux de capteurs sans fil hétérogène.

Dans les réseaux de capteurs sans fil distribués [65], les nœuds capteurs utilisent directement des clés pré-distribuées, ou utilisent des matériaux de chiffrement (les secrets partagés, les clés partiels, etc.) pour générer dynamiquement des clés par paire et par groupe. Le défi consiste à trouver un moyen efficace de distribuer les clés et les matériaux de chiffrement aux nœuds capteurs avant le déploiement. Les solutions au problème de distribution de clés dans les RCSF distribué peuvent utiliser l'une des trois approches: probabiliste, déterministe ou hybride.

Dans les RCSF hiérarchiques [65], il existe une ou plusieurs stations de base qui peuvent agir comme un centre de distribution de clés. Il existe trois approches pour garantir la distribution de clés :

- La distribution à base d'une clé par réseau (Network keying).
- La distribution à base d'une clé par paire de nœuds (Pair-wise keying).
- La distribution à base de groupe (Group keying).

### 2.3.3 Phase de découverte des clés (Shared-Key decouvert)

Après le déploiement, le protocole de communication est responsable de la découverte de la clé commune entre deux nœuds voisins. Selon la portée de l'antenne radio, un nœud doit découvrir ses voisins parmi lesquelles il partage une clé. Ainsi, si deux nœuds partagent la même clé donc un lien peut être établé entre les deux nœuds. Un système efficace de découverte ne doit pas permettre à un adversaire de connaitre les clés partagées entre les nœuds [64], [66].

### 2.3.4 Phase d'établissement d'une clé de chemin sécurisé (Path-key establishement) :

Après la phase de découverte de clés partagées, le réseau peut être vu comme un graphe connecté par des liens sécurisés. Ainsi toute paire de nœuds qui ne partagent pas une clé commune mais sont connecté par plusieurs sauts et souhaitent communiquer peuvent chercher un chemin sécurisé entre eux. Ce chemin passe par un ensemble de nœuds qui présente déjà des liens sécurisés. Une fois le chemin établi, les deux nœuds peuvent l'exploiter pour commencer une communication sécurisée de bout en bout [66], [67].

### 2.3.5 Phase d'isolement des nœuds incohérents et mise à jour des clés (revocation and re-keying):

Un nœud incohérent est celui qui ne fonctionne pas comme spécifié [66]]. Identifier et isoler les nœuds aberrants est important pour la poursuite du bon fonctionnement du réseau de capteurs. Un nœud peut cesser de fonctionner pour les raisons suivantes :

- Il a épuisé sa source d'énergie.
- Il est endommagé par un attaquant.
- Isolé parce que le nœud intermédiaire a été compromis.
- Un nœud a été compromis et communique des informations fictives à la station de base.

Dans un schéma de base la révocation d'un nœud compromis se fait par le nœud contrôleur (qui a une grande connectivité et peut être mobile). Lorsque le nœud contrôleur détecte un nœud compromis dans le réseau il diffuse un message de révocation à tous les nœuds du réseau, ce nœud contrôleur contient une liste signée de k identificateurs des clés ($k_i d_i$) pour que ces clés soient retirées des listes de clés des autres nœuds. Une fois que les clés correspondantes sont complètement supprimées au niveau des listes de tous les nœuds, il peut y avoir des liens manquants entre les différents nœuds et ils doivent alors se reconfigurer à partir de l'étape de découverte de clé partagée pour que de nouveaux liens puissent être formés [64].

# 3 Mécanismes avancés pour la sécurité dans les RCSF

En raison de leur nature de déploiement et d'application, les RCSF sont ouverts à des problèmes uniques, en plus des attaques trouvées dans les réseaux traditionnels, les limitations et les contraintes due aux nœuds capteurs dans les RCSF interdit l'utilisation des mécanismes sophistiquées pour garantir certaines objectifs de sécurité. Certaines méthodes utilisées, telles que la cryptographie, l'authentification et d'autres mécanismes, ne résolvent pas entièrement le problème. D'autres mesures de sécurité différentes des approches traditionnelles doivent être mises en place pour améliorer la sécurité du réseau. Récemment, l'attention a été portée sur le concept de confiance et les systèmes de détection d'intrusions pour renforcer la sécurité et la fiabilité dans les réseaux de capteurs sans fil.

## 3.1 Modèles de confiance

Dans les RCSF un système de gestion de confiance devrait calculer la confiance et la réputation séparément. Sur la base de confiance, vient celle de la réputation, qui est parfois considérée par certains auteurs comme une confiance. Parfois la réputation est confondue avec la confiance par certaines auteur : la première n'affecte que partiellement la confiance. Les valeurs de réputation sont référées au comportement des différentes entités du réseau (la réputation est l'opinion d'une personne à propos de l'autre, et par construction, d'un nœud capteur à un autre). Ces valeurs pourraient être utilisées comme entrées afin de déterminer les valeurs de confiance. Alors la confiance est une dérivation de la réputation d'une entité. Basé sur la réputation, un niveau de confiance est accordé à une entité. En ne calculant pas la confiance directement à partir du comportement d'un nœud, il est possible de mieux gérer des aspects tels que l'évolution du nœud, le

vieillissement, etc. La réputation elle-même a été construite au fil du temps en fonction de l'histoire de comportement de cette entité, et peut refléter une évaluation positive ou négative. Par exemple, un nœud malveillant et non fiable qui devient soudainement bon ne devrait pas être immédiatement approuvé. Certains systèmes de gestion de confiance ne tiennent pas compte de la réputation et obtiennent directement les valeurs de confiance sous la forme d'une combinaison pondérée de confiance directe et de confiance indirecte [68], [69].

Un modèle d'évaluation de confiance se base principalement sur la construction d'un mécanisme de confiance pour chaque nœud à l'intérieur du réseau sur la base d'expériences d'interactions passées. Le modèle évalue le comportement de communication des nœuds, calcule la valeur de confiance des nœuds, ainsi l'établissement de la confiance entre les nœuds permet d'évaluer la fiabilité des autres nœuds, car l'amélioration de la durée de vie des réseaux dépend de la nature coopérative et confiante de ses nœuds. En connaissant, de la réputation des nœuds dans leur voisinage et leur comportement réel, il est possible que les nœuds choisissent un plan d'action approprié lorsqu'ils prennent des décisions opérationnelles (savoir quel est le meilleur partenaire pour démarrer une collaboration). Le modèle peut reconnaître les nœuds suspects (mauvais nœuds) à l'intérieur du réseau et réduire leur impact à l'acquisition de données et à la communication au maximum [68]–[70].

La confiance dans les réseaux de capteurs joue un rôle important dans la construction du réseau et l'ajout et / ou la suppression des nœuds défaillants et peu fiables. L'auto configuration n'est pas le seul avantage: un système de gestion de confiance peut également aider et / ou tirer parti d'autres protocoles et mécanismes de sécurité, par exemple, protection du matériel, un système de détection d'intrusions, gestion des clés, confidentialité, mesure pour le routage, par exemple, les nœuds capteurs sont par fois besoin de connaître les autres nœuds auxquels faire confiance pour le transfert d'un paquet. En outre, des services complexes tels que la localisation sécurisée et les systèmes de détection d'intrusion peuvent bénéficier de l'existence d'un système de gestion de confiance, soit en utilisant la sortie du système, ou en tant qu'assistant dans leur processus de prise de décision, ou en fournissant des informations de confiance utiles qui pourraient être utiles pour tout autre service. L'existence d'un système de gestion de la confiance peut également aider les activités des systèmes de gestion des clés et des mécanismes liés à la vie privée [69], [71].

### 3.1.1 Types de modèles de confiance

Le modèle de confiance [71] peut être conçu en considérant les composantes principales suivantes : la collecte d'informations, la modélisation de l'information, la diffusion de l'information, la notation parasite, la détection et la réponse. Chaque étape a plusieurs problèmes qui peuvent être considérés avec précaution dans la conception du modèle de confiance dans les réseaux de capteurs sans fil. Selon les informations de confiance stockées, les modèles de confiance peuvent être classées

généralement en trois grands catégories : modèle centralisé, distribué et hybride comme le représente la Figure 2.

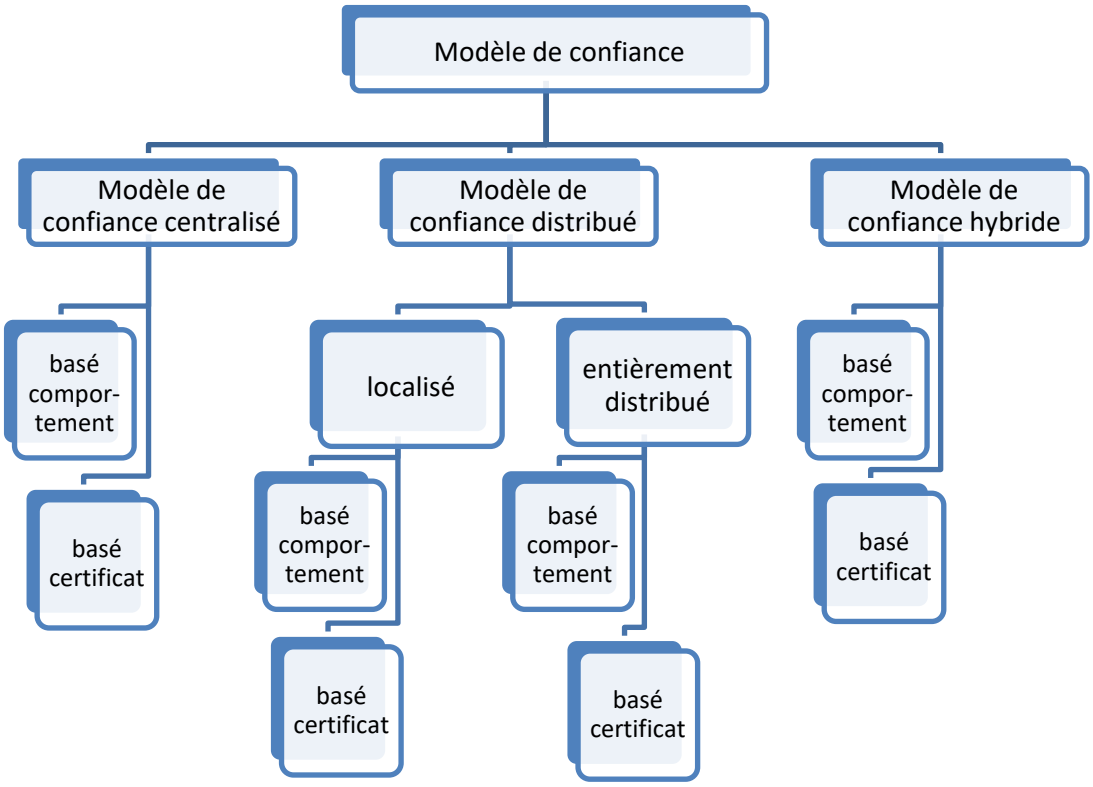


**Figure 2 : Classification des modèles de confiances.**

Le modèle de confiance centralisée consiste en un seul serveur globalement approuvé qui est généralement la station de base, ce dernier permet de détermine les valeurs de confiance pour chaque nœud du réseau. Par conséquent, ce modèle permet de préserver au niveau des nœuds le temps de calcul et l'espace mémoire, mais présente aussi les inconvénients suivants : plus de temps de communication, moins fiable, manque d'évolutivité. Dans le modèle de confiance distribuée, chaque nœud calcule localement les valeurs de confiance de tous les autres nœuds du réseau. Chaque nœud doit également maintenir un enregistrement à jour des valeurs de confiance de l'ensemble du réseau sous la forme d'une table. Malgré que ce modèle augmente le coût de calcul au niveau des nœuds capteurs mais il est considéré comme le modèle le plus fiable et le plus évolutif. Le modèle de confiance hybride combine les propriétés des approches de gestion de confiance centralisée et distribuée. L'objectif principal de cette approche est de réduire le coût associé à l'évaluation de la confiance par rapport aux approches distribuées. Ce schéma est utilisé avec les schémas de cluster dans lesquels la tête de cluster agit en tant que serveur central pour ce cluster. Par apport au modèle centralisé, ce modèle permet de réduire les frais généraux de communication,

mais présente une surcharge de calcul et de mémoire au niveau des chefs de clusters. Ce modèle est moins fiable et évolutif par rapport au modèle distribué [72].

## 3.2 Système de détection d'intrusions

### 3.2.1 Notions préliminaires

#### *3.2.1.1 Les intrusions*

Les intrusions sont des actions qui tentent de contourner les mécanismes de sécurité des systèmes informatiques. Il s'agit donc d'un ensemble d'actions qui menace les conditions de sécurité standards qui sont : l'intégrité, la disponibilité ou la confidentialité d'une ressource réseau [27]. Les intrusions dans un réseau peuvent se produire avec différentes manières [73] :

- Tentative d'effraction: une tentative d'avoir un accès non autorisé au réseau.
- Masquerade: un attaquant utilise une fausse identité pour gagner accès non autorisé au réseau.
- Pénétration: L'accès non autorisé au réseau.
- Fuite: un flux d'informations indésirable provenant du réseau.
- DoS: blocage des ressources réseau (c'est-à-dire, communication bande passante) aux autres utilisateurs.
- Utilisation malveillante: nuire délibérément aux ressources du réseau.

#### *3.2.1.2 La détection d'intrusion*

La détection d'intrusion [27] est le processus de surveillance des événements se produisant dans un système informatique ou un réseau et de les analyser pour déceler des signes d'intrusion, comme une entrée non autorisée, une activité ou une modification de fichier. Le processus de détection d'intrusion comporte trois étapes:

- surveiller et analyser le trafic;
- Identifier des activités anormales;
- Évaluer la gravité et déclencher une alarme.

#### *3.2.1.3 Système de détection d'intrusions (SDI)*

Un SDI est un logiciel qui automatise le processus de détection d'intrusion et détecte les intrusions possibles. Les systèmes de détection d'intrusion servent trois fonctions de sécurité

essentielles: ils surveillent, détectent et répondent aux activités non autorisées des initiés de l'entreprise et à l'intrusion extérieure. Un SDI est composé de plusieurs composants:

• des capteurs qui génèrent des événements de sécurité;

• Console pour surveiller les événements et les alertes et contrôler les capteurs;

• Moteur central qui enregistre les événements consignés par les capteurs dans une base de données et utilise un système de règles pour générer des alertes à partir des événements de sécurité reçus [27].

Un SDI en cours de conception doit satisfaire les conditions suivantes [73] :

- ne pas introduire de nouvelles faiblesses dans le système,
- besoin de peu de ressources système et ne doit pas dégrader la performance globale du système en introduisant des frais généraux,
- fonctionner en continu et rester transparent pour le système et les utilisateurs,
- utiliser les normes pour être coopératif et ouvert,
- être fiable et minimiser les faux positifs et les faux négatifs dans la phase de détection.

### 3.2.2 Système de détection d'intrusions pour les RCSF

Due aux caractéristiques sévères des réseaux de capteurs sans fil, ces derniers sont susceptibles de nombreuses formes d'intrusion. Dans les réseaux filaires, le trafic et le calcul sont typiquement surveillés et analysés pour détecter les anomalies à divers points de concentration. Ceci est souvent coûteux en termes de mémoire et de consommation d'énergie du réseau, ainsi que de sa bande passante intrinsèquement limitée. Dans le contexte spécifique des RCSF à savoir : ressources de communication et de calcul limitées, courte portée radio, protection physique faible ce qui permet à un adversaire de compromettre facilement le réseau. Les réseaux de capteurs sans fil nécessitent une solution entièrement distribuée qui se base sur la coopération de nœuds et peu coûteuse en termes de communication, d'énergie et de mémoire. Afin de rechercher des anomalies, les applications et les modèles de menaces typiques doivent être compris. Il est particulièrement important pour les chercheurs et les praticiens de comprendre comment des adversaires coopérants pourraient attaquer le système. L'utilisation de groupes sécurisés peut être une approche prometteuse pour la détection d'intrusion décentralisée. En particulier, un système de détection d'intrusion dédié aux RCSF doit satisfaire les propriétés suivantes [74] :

- Localisez l'audit : Un SDI pour les réseaux de capteurs doit fonctionner avec des données d'audit localisées et partielles. Dans les réseaux de capteurs, il n'existe pas de points centralisés (en dehors de la station de base) capables de collecter des

données d'audit globales. Cette approche correspond donc au paradigme du réseau de capteurs.

- Minimiser les ressources : Un SDI pour les réseaux de capteurs doit utiliser une petite quantité de ressources. Le réseau sans fil ne dispose pas de connexions stables et les ressources physiques du réseau et des périphériques, telles que la bande passante et l'alimentation, sont limitées. La déconnexion peut survenir à tout moment. En outre, la communication entre les nœuds à des fins de détection d'intrusion ne devrait pas prendre trop de la bande passante disponible.

- Ne faites confiance à aucun nœud : Un SDI ne peut pas supposer que n'importe quel nœud est sécurisé. Contrairement aux réseaux câblés, les nœuds de capteurs peuvent être très facilement compromis. Par conséquent, dans les algorithmes coopératifs, l'SDI doit supposer qu'aucun nœud ne peut être entièrement approuvé.

- Soyez vraiment distribué : Cela signifie que la collecte et l'analyse des données sont effectuées sur plusieurs sites. L'approche distribuée s'applique également à l'exécution de l'algorithme de détection et à la corrélation d'alertes.

L'auto sécurité : Un SDI devrait être capable de résister à une attaque hostile contre lui-même. La compromission d'un nœud de surveillance et le contrôle du comportement de l'agent SDI intégré ne devraient pas permettre à un adversaire de révoquer un nœud légitime du réseau, ou de garder un autre nœud d'intrus non détecté [74].

### *3.2.2.1 Techniques de détection d'intrusions*

La tache de détection d'intrusion est considéré comme le cœur d'un système de détection d'intrusion, cette tache permet à un SDI de détecter les activités anormales dans un système. Les SDIs fonctionnels doivent remplir plusieurs objectifs liés à la détection d'intrusion en utilisant divers ingrédients comme [75] :

- Les points de contrôle d'intrusion représentent les états observables du SDI et analysent l'activité du capteur qui prédit la transition de la forme normale à l'état d'intrusion.

- Création d'un profil d'activité qui identifie l'activité anormale des états observables en mesurant l'écart du capteur par rapport au comportement normal.

- Dérive conceptuelle qui mesure le changement de comportement de l'utilisateur sur une période donnée.

- Boucle de contrôle qui adopte le déclencheur sur la base de la somme pondérée des mesures de capteur proportionnelles, moyennes et dérivées sur la fenêtre de temps dérivée et intégrale

Les SDIs existants dans les réseaux ad hoc ne peuvent pas être adaptés aux RCSF. La recherche actuelle se concentre sur la façon de détecter et d'éliminer les fausses informations injectées. Notez que les nœuds compromis peuvent toujours injecter de fausses informations dans un réseau de capteurs. Ainsi, la coopération entre les capteurs, en particulier les nœuds voisins, est nécessaire pour décider de la validité d'un rapport. Plusieurs techniques sont employées pour détecter les intrus à savoir : la détection des anomalies, la détection de signature, la surveillance de la cible, la détection de mauvaise utilisation, etc. Mais quels que techniques ne sont praticables dans les RCSF comme la détection de signature et la détection de mauvaise utilisation, qui exiges une capacité de stockage importantes pour le sauvegarde et la mise à jour des bases de signature, Dans cette section, nous discutons des techniques de détection d'intrusion qui peuvent être utilisés dans les RCSF.

- ***Détection d'anomalies :*** Cette technique est conçue pour détecter les comportements anormaux, pour ce faire un SDI décrit d'abord les caractéristiques réelles d'un comportement normal afin d'établir une base de référence des modes d'utilisation normaux, ainsi toutes les activités qui s'écartent de ces comportements sont signalées en tant qu'intrusions possibles. Ces techniques peuvent identifiés des nouveaux types d'intrusion comme des déviations par rapport à un usage normal. Si un nœud capteur [76] n'agit pas selon les spécifications définies d'un protocole particulier, l'SDI aura une confiance élevée pour décider que le nœud est malveillant. Les mauvaises décisions prises par SDI en termes d'alarmes fausse positives et fausses négatives affectent la précision de la détection. Par conséquent, l'inconvénient de cette méthodologie est que le système peut présenter un comportement légitime mais invisible.

- ***Détection à base de spécification :*** Les techniques de détection d'intrusion basées sur des spécifications [73] combinent les avantages de la détection à la fois par mésusage et par anomalie techniques, en utilisant des spécifications développées manuellement et contraintes pour caractériser le comportement légitime du système. Les techniques de détection d'intrusion basées sur des spécifications sont similaires aux techniques de détection basées sur l'anomalie, dans le sens de détecter les attaques comme les écarts par rapport à un profil normal. La définition du modèle est réalisée à l'aide d'un algorithme d'apprentissage formé sur les données étiquetées, où chaque instance d'un ensemble de données est étiquetée comme «normale» ou «intrusive». Ces techniques sont capables de recycler automatiquement les modèles de détection d'intrusion sur différentes données d'entrée qui incluent de nouveaux types d'attaques; aussi longtemps qu'ils ont été

étiquetés de manière appropriée. Contrairement à l'SDI basé sur les signatures, les modèles de mauvaise utilisation sont créés automatiquement et peuvent être plus sophistiqués et précis que les signatures créées manuellement. Cela permet de simplifier le système de détection, et réduit significativement le taux de fausses détections négatives et ils ont un degré élevé de précision dans la détection des attaques connues et de leurs variantes. Leurs inconvénients sont qu'ils ne peuvent pas détecter les intrusions inconnues et ils s'appuient sur des signatures extraites par des experts humains. Comparée à la détection à base d'anomalies, cette technique semble être la mieux appropriée aux limitations des réseaux de capteurs [21], [73].

#### *3.2.2.2 Architecture des systèmes de détection d'intrusions*

Généralement , Les architectures du SDI sont classées selon plusieurs façons, la première classification divise les SDIs en deux catégories de base selon le mécanisme de collecte des données [77] : basées sur l'hôte et sur le réseau, selon le mécanisme de collecte des données. De plus, les architectures du SDI peuvent être classées en fonction de la technique de détection. La classification des architectures du SDI proposées pour les réseaux ad hoc sans fil peut être adaptée aux besoins des RCSF en trois catégories. Cette classification divise les SDIs en trois catégories.

- ***SDI à base d'architecture décentralisée :*** Dans cette catégorie, chaque nœud fonctionne d'une façon autonome comme un SDI indépendant et est responsable de la détection des attaques uniquement pour lui-même. Un tel SDI ne partage aucune information ou ne coopère pas avec d'autres systèmes. Par conséquent, certaines attaques ne peuvent être détectées, étant donné que chaque nœud ne possède qu'une vision partielle du réseau. L'architecture décentralisée est très simple, et offre une grande résilience aux attaques de compromission. Cette architecture implique que tous les nœuds du réseau sont capables d'exécuter un SDI. Cependant, la limitation des nœuds capteurs ne permet pas l'utilisation des puissantes méthodes d'analyse et de détection. Cependant, certaines attaques ne peuvent être détectées, étant donné que chaque nœud ne possède qu'une vision partielle du réseau. En outre, la limitation des nœuds capteurs ne permet pas l'utilisation des puissantes méthodes d'analyse et de détection [21].

- ***SDI à base d'architecture centralisée :*** Dans une architecture centralisée le principe de base consiste à commander à partir d'une console centrale (en général la station de base) la détection, la surveillance et l'audit. Alors toutes les informations pertinentes pour la détection d'intrusions et la surveillance doivent être transférées vers la station de base, qui est généralement la plus puissante en termes de ressources. Cette centralisation, à permet de perfectionné le SDI par l'utilisation de méthodes de détection plus sophistiquées, et plus fiable. Ainsi une vision globale sur l'état du réseau peut être réalisée facilement puisque toutes les informations nécessaires sont à la disposition de la station de base. Comme toutes architectures centralisées, cette architecture présente les

mêmes problèmes tels que la surcharge du réseau, la consommation élevée d'énergie et de ressources, ainsi la compromission d'un point central affecte l'ensemble du réseau.

- ***SDI à base d'architecture hybride :*** Cette architecture est applicable sur le réseau hiérarchique, où le réseau est divisé en clusters avec des nœuds chefs de cluster. Ces nœuds [77] sont responsables du routage au sein du cluster et acceptent tous les messages d'accusations des autres membres du cluster indiquant quelque chose de malveillant. De plus, les nœuds chef de cluster peuvent également détecter des attaques contre les autres nœuds de tête de cluster du réseau, car ils constituent l'épine dorsale de l'infrastructure de routage. Cette architecture [21] est une combinaison d'architectures centralisées et décentralisées. Par conséquent, la détection d'intrusions est centralisée au niveau des clusters et décentralisée entre les chefs des clusters (la détection d'intrusions est centralisée localement et décentralisée globalement). En effet, chaque nœud dans le cluster dispose d'un agent de détection d'intrusions. Ce dernier transfert ses alarmes au chef du cluster afin de vérifier et confirmer l'intrusion détectée. L'architecture de détection hybride offre un bon compromis entre les deux architectures précédentes en se servant de leurs avantages tout en minimisant leurs inconvénients.

# 4 Conclusion

La sécurité dans les réseaux de capteurs sans fil a attiré beaucoup d'attention des chercheurs dans ces dernières années. Ainsi plusieurs approches de sécurités ont été développés pour répondre à la question de sécurité dans les RCSF, nous constatons que les métriques suivantes soient indispensables pour l'évaluation de n'importe quelle approche de sécurité proposé :

- Résilience: un système de sécurité devrait être capable de protéger le réseau contre les attaques même avec la présence de quelques nœuds compromis,
- Efficacité énergétique: un système de sécurité doit fournir des solutions de sécurité tout en préservant l'énergie des nœuds capteurs afin de prolonger la durée de vie du réseau.
- Flexibilité: la gestion des clés doit être flexible afin de permettre différentes méthodes de déploiement réseau, telles que la diffusion aléatoire des nœuds et le placement prédéterminé des nœuds.
- Évolutivité: un système de sécurité doit pouvoir évoluer sans compromettre les exigences de sécurité.
- Tolérance aux pannes: un schéma de sécurité devrait continuer à fournir des services de sécurité en présence de défauts tels que des nœuds défaillants.

- Auto-organisation: les capteurs peuvent être défectueux ou leurs énergies soit épuisés. Les capteurs restants doivent être capable de se réorganisés pour maintenir le niveau de sécurité.

- Assurance: un système de sécurité doit fournir la capacité de garantir la disponibilité et la diffusion de toutes informations à différents niveaux aux utilisateurs finaux.

Plusieurs mécanismes, conçus pour sécuriser les RCSF ont été développé ces dernières décennies, Malgré ces efforts fournis la sécurité est apparu toujours comme le problème le plus sévère et la recherche dans ce domaine reste toujours ouvert pour répondre à la question de sécurité tout en respectant les contraintes exigés par les RCSF. Puisque la cryptographie, est la solution la plus efficace qui nous permet d'assurer plusieurs conditions de sécurité en même temps, nous constatons qu'un schéma de gestion des clés cryptographique efficace est indispensable pour renforcer le niveau de sécurité fourni par les systèmes cryptographiques. Dans le chapitre suivant nous allons présenter une étude détaillée sur les schémas de gestion des clés dans les RCSF.

# CHAPITRE 4 :

# LA GESTION DES CLES DANS LES RCSF

# 1 Introduction

La plupart des protocoles de sécurité proposés pour sécuriser les réseaux de capteurs sans fil se basent sur des systèmes cryptographiques. La cryptographie est utilisée pour pouvoir chiffrer les messages échangés entre les nœuds capteurs. Quelque soit le système cryptographique (symétrique/asymétrique), les nœuds capteurs ont besoin de générer, charger, utiliser, transférer, stocker, protéger, et détruire des clés avec leurs voisins de manière sécurisée afin de répondre à une ou plusieurs conditions de sécurité, telles que la confidentialité, l'authentification, la disponibilité ou l'intégrité des données. Par conséquent, la gestion des clés est l'un des problèmes le plus délicats de la cryptographie dans les RCSF. Un schéma de gestion de clés efficace doit fournir les tâches suivantes :

- Tâches d'établissement des clés (key establishement) ;
- Tâches de pré distribution des clés (Key predistribution) ;
- Tâches de découverte des clés partagées (Shared-Key decouvert) ;
- Tâches d'établissement d'une clé de chemin sécurisé (Path-key establishement) ;
- Phase d'isolement des nœuds incohérents et mise à jour des clés (revocation and re-keying).

La gestion de clés constitue la pierre angulaire de la sécurité dans les RCSF. Ainsi que le processus de distribution des clés constitue l'épine dorsale des schémas de gestion des clés cryptographiques dans les RCSF, cette tâche a été étudiée de manière intensive récemment dans le contexte des RSCFs.

La spécificité des réseaux de capteurs sans fil par rapport aux réseaux traditionnels et Ad Hoc exige d'adapter de nouveaux schémas de gestion des clés qui tiennent compte cette spécificité. Cette dernière est due à la nature du support de communication sans fil, la capacité de bande passante, le déploiement aléatoire des nœuds, la densité des nœuds et la grande échelle des RCSF, aux contraintes de ressources telles que la capacité limitée de calcul, de mémoire et d'énergie des nœuds capteurs. Les schémas de gestion des clés conçus pour les réseaux de capteurs sans fil doivent également garantir un niveau de sécurité élevé tout en tenant compte des contraintes liées aux nœuds capteurs. Plusieurs schémas de gestion des clés ont été proposés dans la littérature. Ces derniers sont classifiés en plusieurs catégories.

# 2 La distribution des clés dans les RCSF

Le principal problème des méthodes cryptographiques utilisées pour sécuriser les réseaux de capteurs est la distribution des clés. Ce problème a été étudié de manière intensive récemment. Dans le cas des RCSF, où la topologie du réseau est inconnue qu'après le déploiement des nœuds, une phase de pré distribution des clés. Une phase de pré distribution des clés (qui consiste de charger les clés dans les nœuds capteur avant le déploiement) est le seul moyen sûr et efficace qui

permet aux nœuds communicants de partager les clés secrètes d'une manière sécurisée. Il existe trois approches pour garantir la distribution des clés :

- ***La distribution à base d'une clé par réseau (Network keying) :*** C'est le système de distribution de clés le plus simple à utilisé dans les réseaux de capteurs sans fil. Le principe de ce système consiste à charger une seule clé unique au niveau de tous les nœuds capteurs du réseau avant leur déploiement. Tous les nœuds capteurs communiquent en utilisant cette clé unique. Ce système est très léger en termes de mémoire, de calcul et d'échange de données mais il est également très vulnérable. Comme tous les capteurs du réseau utilisent la même clé pendant leurs communications, la probabilité d'une attaque crypto analytique sur cette clé est très élevée. Autrement dit, si un nœud est compromis, capturé ou si la clé est révélée d'une autre manière, tout le réseau est compromis.
- ***La distribution à base d'une clé par paire de nœuds (Pair wise keying) :*** La distribution d'une clé entre une paire de nœuds dans un réseau de capteurs est le système de gestion de clés le plus sûr pour les réseaux de capteurs sans fil. cette solution consiste à utiliser des clés distinctes pour toutes les paires de nœuds possibles dans le réseau. Pour un réseau de taille N, chaque nœud est pré-chargé avec N-1 clés secrètes. Chacune de ces clés est connue seulement par ce nœud et un des N-1 autres nœuds. Les schémas de distribution de clés par paire opèrent généralement en trois phases :
    - Configuration de la clé avant le déploiement du réseau ;
    - Découverte de la clé partagée après le déploiement ;
    - Etablissement de la clé de chemin si deux nœuds capteurs ne partagent pas une clé.

  Ce schéma est probablement le plus sécurisé pour les réseaux de capteurs sans fil. L'établissement de clés par paires fournit non seulement la confidentialité et l'authenticité dans un réseau, mais fournit également un mécanisme efficace pour la révocation d'un nœud capteur compromis dans le réseau. Cependant, Si le nombre n devient trop grand ce schéma n'est pas efficace en termes d'évolutivité et de besoins en mémoire.
- ***La distribution à base de clé par groupe (Group keying) :*** Selon différentes échelles, il existe deux types de communication de groupe. Le premier type de communication se base sur la diffusion globale dans le réseau. Généralement, il est effectué par une station de base. Le deuxième type de communication est basé sur la diffusion locale, où chaque nœud collabore avec ses voisins pour remplir divers objectifs. Les deux types nécessitent une clé de groupe pour chiffrer les communications. Ce modèle de distribution des clés est un modèle hybride qui consiste à combiner les deux précédents modèles. Au niveau d'un cluster, les nœuds partagent une clé unique pour établir une communication sécurisée au sein du cluster *(distribution à base de clé par réseau)*. Ainsi, les communications entre les groupes se basent sur l'utilisation des clés partagées entre chaque paire de groupes *(distribution à base de clé par paire de nœuds)*.

# 3 Classification des schémas de gestion des clés dans les RCSF

Dans les réseaux de capteurs, Il existe plusieurs façons de classer les schémas de gestion de clés et divers taxonomies sont présentées par plusieurs chercheurs. L'une des classifications consiste à classer les protocoles de gestion des clés selon les catégories suivantes:

- Déterministe: dans ce cas, les processus qui génèrent les pools de clés et les chaînes clés sont déterministes.
- Probabiliste: les chaînes de clés sont sélectionnées de manière aléatoire à partir d'un pool de clés donné et réparties entre les nœuds.
- Hybride: cette approche utilise une combinaison de solutions probabilistes et déterministes pour augmenter la résilience et l'évolutivité.

Un autre critère est constitué par les réseaux de capteurs hiérarchiques et distribués basés sur des modèles de réseau. Les autres critères de classification peuvent être basés sur le modèles de réseau à savoir : dynamiques et statiques, réseaux homogènes ou hétérogènes, hiérarchiques et distribués. Nous classons les schémas de gestion des clés selon le système cryptographique utilisé à savoir symétrique ou asymétrique. Cette classification nous permet d'englober tous les autres critères cités ci-dessus.

## 3.1 Schéma de gestion de clés symétriques

Malgré que cette approche est peu pratique pour l'implémentation en temps réel et à grande échelle, car elle n'assure pas une bonne évolutivité du réseau. Mais la simplicité et l'efficacité de ces systèmes favorisent l'utilisation de la cryptographie à clé symétrique dans la plupart des schémas de gestion des clés proposés dans les réseaux de capteurs. Par conséquent, cette technique nécessite un mécanisme de pré-distribution de clé efficace. Nous classons les schémas de gestion des clés symétriques en trois grandes catégories: schéma de participation à une station de base, schéma de troisième nœud à base de confiance et schémas de pré-distribution.

### 3.1.1 Système de gestion basé sur la participation de la station de base

Ce schéma est également appelé approche de distribution de clés (KDC) centralisée. Dans ce type de schémas la station de base est chargée de fournir une clé unique qui sera partagée par tous les nœud réseau, ainsi de générer, crypter et envoyer en utilisant la clé partagée une clé de liaison pour chaque paire de nœuds qui ont besoin d'une clé par paire pour communiquer entre eux. Ce schéma consomme moins de mémoire et permet une réplication et une résilience de nœud parfaitement contrôlée. Mais comme la station de base joue le rôle d'un centre de distribution de clés, par conséquent cette dernière représente une faille de sécurité et devient la cible de différents types d'attaques, Le problème majeur de ce schéma est au niveau de l'évolutivité du système. On

outre, la station de base doit garantir un envoi sécurisé des clés de liaison aux nœuds capteurs associés, ce qui pose un autre défi de sécurité pour ce type de schémas.

Les schémas qui se basent sur la participation de la station de base sont les suivants:

- SPINS : est le premier protocole de sécurité qui répond aux différentes exigences de sécurité sans fil. SPINS propose deux protocoles de sécurité pour sécuriser les canaux de communication. Ces deux protocoles sont : SNEP qui permet de fournir la confidentialité des données, l'authentification des communications unicast (*entre une paire de nœuds)*, l'intégrité, la protection contre la répétition, la faible fraîcheur des messages et µTESLA qui permet d'assurer l'authentification des communications par diffusion dans les réseaux de capteurs.

- LKHW : est un schéma de communication de groupe sécurisé, qui combine entre la diffusion dirigée et le schéma LKH (Logical Key Hierarchy). Il est utilisé pour protéger le protocole de diffusion dirigée. Il se compose d'un ensemble de clés structurées et arborescentes avec des nœuds source en tant que feuilles et un nœud puits en tant que racine, où chaque nœud feuille contient les clés de tout le chemin allant du nœud feuille au nœud racine. LKHW intègre la sécurité et le routage dans le même protocole.

### 3.1.2 Schéma basé sur un tiers de confiance

Dans ce schéma, un troisième nœud est impliqué dans l'établissement d'une clé entre deux nœuds capteurs, ce dernier est considéré comme un nœud tiers de confiance et qui est utilisé en tant qu'intermédiaire de confiance pour l'établissement d'une clé partagée entre des nœuds. Ce type de systèmes permet de diminuer les frais généraux de communication par rapport à l'approche centralisée de distribution de clés (KDC). Le protocole le plus populaire qui se base sur un troisième nœud tiers de confiance est appelé PIKE proposé par Chan et Perrig (2005).Il repose sur l'utilisation d'un ou de plusieurs nœuds intermédiaires de confiance pour établir des clés partagées entre deux nœuds communicants.

### 3.1.3 Schémas de pré-distribution de clés

Le principe de base de cette méthode consiste à charger les nœuds capteurs par des clés ou des informations secrètes avant leur déploiement dans la zone de détection. Ce schéma consiste se déroule en trois phases : phase de pré-distribution de clés, phase de découverte de clés partagées et phase d'établissement de clés de chemins. Plusieurs solutions de gestion de clés basées sur la pré-distribution existent. Dans ce qui suit, nous présentons une taxonomie de ces solutions.

#### *3.1.3.1 Schéma de pré-distribution basé sur une clé maîtresse*

Dans ce schéma, Une seule clé appelée clé principale ou maîtresse est chargée dans tous les nœuds du réseau avant le déploiement. Après le déploiement, tous les nœuds partagent la même clé pour le cryptage et le décryptage des messages. Ce schéma est simple à cause de l'absence du processus de découverte et de partage de clés, ce qui répond aux exigences des RCSF. En plus, comme chaque nœud capteur n'a besoin de stocker qu'une seul clé, ce qui occupe très peu d'espace mémoire, il permet une bonne évolutivité du système. Cependant, ce schéma ne garantit aucune résilience et il y a toujours une probabilité que la clé principale soit compromise. Dans ce cas, tout le réseau est compromis. Par conséquent,. En outre, il n'y a pas d'authentification car tous les nœuds capteurs partagent la même clé.

Lai et al. [78], ont proposé un schéma de pré-distribution basé sur une clé maîtresse. Dans ce schéma, une clé par paire peut être établie en utilisant cette clé principale et un nombre aléatoire échangé entre chaque nœud capteur. Zhu et al [79], ont proposé un schéma amélioré qui permet de renforcer la résilience des nœuds.

#### *3.1.3.2 Schémas de pré-distribution de clés par paire*

La solution la plus simple et la moins couteuse en termes d'utilisation des ressources est de déployer une seule clé principale pour tous les capteurs. Depuis, un adversaire peut capturer un nœud et compromettre la clé très facilement, alors il a une très faible résilience. La solution la plus efficace pour étainier ce type d'attaques consiste à utiliser des clés paires distinctes pour toutes les paires possibles dans le réseau. Pour un réseau de taille N, chaque nœud doit stocker N-1 clés dans sa mémoire. Les schémas de distribution de clés par paire opèrent généralement en trois phases : (1) configuration de la clé avant le déploiement, (2) découverte de la clé partagée après le déploiement, et (3) établissement de la clé de chemin si deux nœuds ne partagent pas une clé.

Ce schéma peut être considéré comme le plus sécurisé et assure une très bonne résilience des clés pour les réseaux de capteurs sans fil. L'établissement de clés par paires fournit non seulement la confidentialité et l'authenticité dans un réseau, mais fournit également un mécanisme efficace pour la révocation d'un nœud capteur compromis dans le réseau. Cependant, ce schéma n'est pas efficace en termes d'évolutivité et de besoins en mémoire. Si le nombre N devient trop grand comme dans de nombreuses applications de réseaux de capteurs sans fil, ce schéma devient irréalisable. En effet, la réplication des nœuds nécessite un temps supplémentaire pour que chaque nœud établisse N - 1 clés uniques avec tous les autres nœuds du réseau et pour maintenir les clés dans la mémoire. De même, la communication entre chaque paire de nœuds capteurs n'est pas nécessaire dans les réseaux de capteurs sans fil, ce qui crée une charge de stockage inutile au niveau des nœuds capteurs.

### *3.1.3.3 Schémas de pré-distribution des clés pures probabilistes*

Pour qu'un réseau de capteurs sans fil fonctionne, il est important que le nœud qui a des données à transmettre (ou à relayer) ait une bande passante suffisante et un (des) nœud(s) voisins qui lui permettent de relayer, à travers différents chemins, ses messages vers la station de base. Donc, il n'est pas nécessaire que des clés soient établies entre chaque paire de nœuds capteurs. En outre, l'ajout ou la suppression d'un ou plusieurs nœuds capteurs seraient à la fois coûteuses et complexes. Pour surmonter les inconvénients présentés ci-dessus, qui sont liée aux schémas basés sur une clé maîtresse et de clé par paire, une solution qui se base sur une certaine probabilité que deux nœuds quelconques puissent communiquer en utilisant une clé par paire a été proposée.

L'idée initiale a été proposée par Eschenauer et Gligor [80]. Elle se base sur un partage de clé probabiliste entre les nœuds d'un graphe aléatoire. Ce schéma ne garantit toutefois pas que deux nœuds soient toujours capables de calculer une clé par paire pour une communication sécurisée. Ce schéma fournit des techniques pour la pré-distribution de clés, découverte de la clé partagée, l'établissement de clé de chemin, et la révocation de clé. Dans la phase de pré-distribution de clé, un grand groupe de clés P est généré, puis k clés distinctes de P et leurs identifiants sont chargé dans chaque nœud avant le déploiement. Chaque paire de nœuds partage une clé avec une certaine probabilité. Dans phase de découverte, chaque nœud diffuse ses identifiants de clé et la clé partagée devient la clé de session du lien entre les deux nœuds. Ainsi si certaines clés restent inutilisées après la phase de découverte, elles peuvent être utilisées pour établir des clés entre des nœuds qui ne partagent pas une clé commune.

### *3.1.3.4 Systèmes de pré-distribution à base polynomiale*

Un schéma de pré-distribution de clé à base polynomiale est basé sur des schémas de pré-distribution de clés par paire. Ainsi, ces schémas surmontent quelques inconvénients des schémas de pré-distribution probabilistes. Dans ce schéma, pour pré-distribuer les clés par paires, un serveur de clé génère aléatoirement un polynôme bi varié de degré *t* :

$$f(x,y) = \sum_{i,j=0}^{t} a_{ij} X^i Y^j$$

**Équation 3 : Polynôme bivarié de degré t.**

sur le champ approprié *Fq*, où *q* est un nombre premier qui est assez grand pour accueillir une clé cryptographique, et la propriété de $(x,y) = f(y,x)$ . Dans le réseau de capteurs, on suppose que chaque capteur a un identifiant unique. Pour chaque capteur *i*, le serveur calcule un partage polynomial de $f(x,y)$, c'est-à-dire, $f(i,y)$. Pour deux nœuds capteurs *i* et *j*, le nœud *i* peut calculer la clé commune $f(i,j)$ en évaluant $f(i,y)$ au point **j**, et le nœud *j* peut calculer la clé commune avec *i*

en évaluant $f(j,y)$ à *i*. Dans ce schéma, chaque nœud capteur doit stocker un polynôme de degré $tf(i,x)$ , qui occupe $(t+1)\log q$ d'espace mémoire. Pour établir la clé d'accès, les deux nœuds capteur doivent évaluer le polynôme pour l'ID de l'autre nœud capteur. Ce schéma prouve sa certitude et une résistance à *t* collusion [81].

Liu et. al [82], ont proposé un schéma polynomial de pré-distribution de clés à partir d'un pool de clés, pour chaque nœud capteur un serveur génère un polynôme de degré *t* (alors la taille mémoire requis pour ce schéma est directement proportionnelle à la valeur de *t*). Ces polynômes possèdent la propriété$f(x,y) = f(y,x)$. Par exemple, si le nœud i reçoit un polynôme $f(i,j)$ et le nœud $j$ reçoit un polynôme$f(i,x)$, ils peuvent calculer une clé commune en utilisant l'identité de l'autre nœud. Ce schéma est évolutif. Cependant, tout le réseau est compromis si les nœuds sont compromis [81].

#### *3.1.3.5 Schémas de pré-distribution des clés basées sur une matrice*

Toutes les clés de liaison possibles dans un réseau de taille N peuvent être représentées comme une matrice de clés N × N. Il est possible de stocker une petite quantité d'informations sur chaque nœud capteur, de sorte que chaque paire de nœuds peut calculer le champ correspondant de la matrice et l'utiliser comme clé de liaison.

Le schéma de Blom [83] utilise une matrice publique G (α+ 1) × N et une matrice D privée N × (α + 1) qui est générée sur GF (q) et où N est la taille du réseau. La solution est appelée α-sécurisée, ce qui signifie que les clés sont sécurisées s'il n'existe pas un α nœuds compromis. La matrice G doit avoir (α + 1) des colonnes linéairement indépendantes (c.-à-d. Matrice Vandermonde) pour fournir une propriété α-sécurisé. La matrice clé est alors définie comme une matrice symétrique K = $(D.G)^T.G$. Le nœud capteur Si stocke des colonnes de taille α+1 à partir de la matrice G comme information publique, et une rangée de taille +1 à partir de la matrice $(D.G)^T$ comme information privée. Une paire de nœuds capteurs ($S_i$, $S_j$), échangent d'abord leurs informations publiques $column_i$ et $column_j$. La clé de lien est ensuite générée comme $K_{ij} = rang_i \times colonn_j$ et $K_{ji} = rang_j \times colonn_i$ respectivement. Le schéma nécessite une multiplication coûteuse de deux vecteurs de taille α+1 où les éléments sont aussi grands que la taille de clé cryptographique correspondante. Chaque nœud capteur diffuse un message et reçoit un message de chaque nœud dans sa couverture radio, où les messages portent un vecteur de taille α+1 [65].

#### *3.1.3.6 Schémas de pré-distribution des clés arborescentes*

Dans ces schémas, les nœuds sont organisés en arborescence dans laquelle chaque nœud communique avec son nœud parent. Ainsi, l'établissement de clé se fait entre les nœuds voisins le long de l'arbre d'agrégation. Le nouveau nœud reçoit deux tickets qui peuvent être vérifiés par deux nœuds existants choisis au hasard par l'administrateur réseau, avant de rejoindre le réseau. Après l'ajout d'un nouveau nœud dans le réseau, il génère une clé par paire pour son nœud parent. Pour transmettre, de manière, sécurisée la clé au parent, le nouveau nœud divise la clé en deux parties et

les envoie avec ses tickets aux nœuds sélectionnés par l'administrateur, qui authentifie le nouveau nœud et les documents clés avec les clés du parent du nouveau nœud. L'avantage majeur de cette approche est la réduction significative du coût de la mémoire**.** Ce schéma peut être divisé en deux types: arbre en étoile et arbre logique binaire.

#### *3.1.3.7 Schéma de pré-distribution de clés basées sur la conception combinatoire*

Un schéma de pré-distribution de clé basée sur la conception combinatoire utilise La probabilité de partager une clé entre les nœuds capteurs, cette probabilité est liée directement aux nombre de clés sélectionnées pour être affecter à chaque chaîne clé avant le déploiement des nœuds dans le réseau. Donc elle peut être augmentée en concevant les chaînes de clé. Ainsi, ce schéma organise les éléments d'un ensemble fini en sous-ensembles pour satisfaire certaines propriétés.

Le schéma de Camtepe et Yener [84] est un schéma combinatoire de pré-distribution de clé par paire. Il est basé sur la conception combinatoire c'est-à-dire sur des techniques de conception de blocs en théorie de conception combinatoire. Il utilise des techniques de conception de quadrangles symétriques et généralisées [65].

#### *3.1.3.8 Schémas de pré-distribution des clés hiérarchique*

L'utilité du réseau hiérarchique est de faciliter la collecte des données, la fusion et la propagation de requêtes dans les environnements hostiles. Dans les schémas de gestion de clés hiérarchiques, un arbre de clés est construit pour le réseau hiérarchique. Les nœuds de ce schéma de gestion de clés sont hiérarchisés à différents niveaux, où ils sont constitués de différents types de nœuds en fonction de leur capacité. Alors, les clés sont distribuées en fonction du nombre de sauts par l'architecture en cluster et au niveau hiérarchique correspond au nœud capteur.

LEAP (Localized Encryption and Authentication Protocol) [79], un protocole de gestion de clés conçu pour les réseaux de capteurs hiérarchiques afin de limiter l'impact du nœud compromis sur le voisinage immédiat. Pour réduire et minimiser l'implication de la station de base dans le processus de gestion de clés, LEAP prend en charge l'établissement de quatre types de clés pour chaque nœud capteur : une clé individuelle partagée avec la station de base, une clé par paire partagée avec un autre nœud capteur, une clé de cluster partagée avec plusieurs nœuds voisins et une clé de groupe partagée entre tous les nœuds du réseau, ce qui réduit en conséquence la consommation d'énergie et le trafic dans le réseau. LEAP se base sur l'exploitation du temps minimal Tmin (temps nécessaire pour qu'un adversaire puisse compromettre un nœud et récupérer la clé de ce dernier) pour permettre à deux nœuds voisins d'établir une clé symétrique, à partir de la clé pré-chargée sur chaque capteur avant le déploiement, ainsi que de supprimer cette dernière de la mémoire du nœud compromis en un temps T <Tmin. LEAP est très efficace pour conter plusieurs types d'attaques telles que l'attaque HELLO Flood et l'attaque de Sybil.

D'autres travaux [74] de gestion de clés sont récemment proposés dans les RCSF hiérarchiques, mais considèrent des réseaux hétérogènes où les nœuds ont des capacités différentes avec des rôles différents. La gestion de clés hiérarchique présente les points faibles suivants :

- Le chef de groupe (cluster head) constitue un point faible du réseau et peut être ciblé par l'adversaire.
- L'énergie du cluster head est consommée rapidement relativement aux autres nœuds**.**

## 3.2 Schéma de gestion de clés asymétriques

Les systèmes de gestion de clés publique ont été écartés et considérés comme étant irréalisables dans les RCSF pendant un certain nombre d'années à cause de la taille du code, la taille des données, les calculs mathématiques intenses et la consommation d'énergie. Cependant, des études récentes ont mis en œuvre avec succès la cryptographie à clé publique dans les réseaux de capteurs sans fil. Ces études se basent sur la bonne sélection d'algorithmes avec les paramètres associés, l'optimisation et les techniques de faible puissance de calcul et d'énergie.

La cryptographie asymétrique offre une meilleure résistance contre les attaques de compromission et permet une bonne évolutivité du système, mais exige des capacités supplémentaires en termes de logiciel et matériel des nœuds. Les solutions asymétriques les plus populaires et qui sont adaptées aux RCSF sont basées généralement sur les deux systèmes cryptographiques RSA et ECC tels que : TinyPK (Tiny Public Key) et Tiny ECC (Tiny Elliptic Curve Cryptosystem)

### 3.2.1 TinyPK (Tiny Public Key)

TinyPK [45] est un protocole basé sur l'utilisation des clés publiques qui permet de mettre en œuvre un système d'authentification et d'échange de clés entre des capteurs à contrainte de ressources tel que les RCSF. L'implémentation de TinyPK est basée sur le célèbre crypto système RSA et l'environnement de développement TinyOS.

TinyPK exploite l'algorithme RSA pour bénéficier de sa vitesse d'exécution des opérations de clé publique en utilisant e = 3 comme exposant public. Tant que des précautions appropriées sont prises (par exemple, utilisation appropriée d'un remplissage aléatoire lors du cryptage de texte), le système RSA à faible exposant est exempt d'attaques connues. Pour que TinyPK soit pratique pour les dispositifs de détection de faible puissance, TinyPK exige d'utiliser uniquement les opérations de clé publique (cryptage de données et vérification de signature) sur le réseau de capteurs. RSA a la propriété agréable que ses opérations publiques sont très rapides comparées à d'autres calculs de technologie de clé publique.

TinyPK implémente l'échange de clés Diffie-Hellman sur la plateforme MICA2. Le but de Diffie-Hellman est de fournir un secret partagé entre deux parties qui peut ensuite être utilisé pour créer une clé cryptographique. L'utilisation de Diffie-Hellman a pour but de généré un secret utilisable pour créer une nouvelle clé TinySec ou une clé de remplacement. Une telle clé permet à deux réseaux de capteurs disjoints de communiquer et permet également le déploiement et le remplacement des capteurs dans une zone de captage existante sans avoir besoin de recherché et pré chargé la clé TinySec utilisée dans cette zone.

### 3.2.2 TinyECC (Tiny Elliptic Curve Cryptosystem)

TinyECC [47] est implémenté sur le système d'exploitation open source conçu pour les réseaux de capteurs sans fil TinyOS. De plus, TinyECC inclut également des paramètres de courbe elliptique recommandés par SECG (Stands for Efficient Cryptography Group) L'objectif principal de TinyECC est de fournir un progiciel prêt à l'emploi et accessible au public pour les opérations PKC basées sur ECC, qui peut être configuré de manière flexible et intégré dans les applications de réseau de capteurs. TinyECC met à la disposition des développeurs différentes combinaisons d'optimisations en des temps d'exécution et des consommation de ressources différents, ce qui lui donne une grande flexibilité lors de l'intégration de TinyECC dans des applications de réseau de capteurs.

La version actuelle de TinyECC ne prend en charge que les schémas ECC bien étudiés tels que ECDSA (signatures numériques), ECDH (établissement de clés par paires) et ECIES (cryptage basé sur PKC). Le protocole d'échange de clés ECDH (EllipticCurve Diode-Hellman) est assez similaire au protocole Diffe-Hellman traditionnel, mais fonctionne dans un groupe de courbe elliptique E ($F_q$) au lieu de $Z^*_p$. La partie coûteuse en calcul de tous les schémas ECC, y compris ECDH, est la multiplication scalaire, une opération de la forme k.P où P est un point premier d'ordre n sur une courbe elliptique, et k est simplement un entier dans l'intervalle [1; n-1]. Il existe deux variantes majeures du protocole ECDH, à savoir l'ECDH statique et éphémère. Ce dernier produit une clé secrète unique dans chaque cycle du protocole, par conséquent, peut fournir un secret à terme, mais cela se fait au prix d'une multiplication scalaire supplémentaire. L'ECDH éphémère nécessite que chaque nœud exécute deux multiplications scalaires, un par un point de base fixe (pour générer une paire de clés éphémère) et l'autre par un point de base variable (pour obtenir la clé secrète partagée).

## 4 La gestion de clés distribuées dans les RCSF

A cause de l'absence d'une station de base dans les RCSF distribués, les nœuds capteurs utilisent directement des clés pré-distribuées ou utilisent des outils de chiffrement pour générer dynamiquement des clés par paire et par groupe. Le défi consiste à trouver un moyen efficace pour distribuer les clés et les outils de chiffrement au sien de nœuds capteurs avant leur déploiement. Les

solutions au problème de distribution de clés dans les RCSF distribués peuvent être classées en trois approches: (1) probabiliste, (2) déterministe, ou (3) hybride. Dans les solutions probabilistes, les chaînes de clés sont sélectionnées aléatoirement à partir d'un pool de clés et après sont distribuées aux nœuds capteurs. Dans les solutions déterministes, les processus déterministes sont utilisés pour concevoir le pool de clés et les chaînes de clés pour fournir une meilleure connectivité des clés. Enfin, les solutions hybrides utilisent des approches probabilistes et des solutions déterministes pour améliorer l'évolutivité et résistance.

# 5 La gestion des clés dynamiques

Dans les schémas de gestion de clés statique, toutes les clés sont pré-distribuées aux nœuds avant le déploiement et aucun processus de renouvellement de clés n'est appliqué. Par conséquent, la probabilité que les clés soient compromises est augmentée considérablement. Un schéma de gestion dynamique des clés est un processus de régénération des clés périodiquement, à la demande, à la détection d'une capture d'un nœud ou en fonction des besoins du réseau. La gestion dynamique de clé possède plusieurs avantages, parmi lesquels on peut citer les suivants :

- La durée de vie du réseau accrue, puisque toutes les clés capturées sont remplacées en temps opportun grâce à un processus connu sous le nom de rekeying ;
- Offre une plus grande probabilité de connectivité ;
- Fournit un meilleur support pour l'extension du réseau; lors de l'ajout de nouveaux nœuds ;
- Une taille de pool de clés plus petite, un minimum de stockage pour maintenir les clés dans les nœuds ;
- Moins de temps de communication pour générer des clés dynamiquement.

L'inconvénient majeur de ce type de gestion de clé est qu'il nécessite un nombre important de messages à échanger pour générer dynamiquement des clés, ce qui n'est pas toléré dans les RCSF. Le défi principal de la gestion dynamique de clé est de concevoir un mécanisme efficace pour sécuriser le processus de rekeying [36], [85], [86].

## 5.1 Classification des Schémas de gestion des clés dynamiques

L'ensemble [87] des schémas de gestion de clés dynamiques peuvent être classés comme distribués ou centralisés. D'une part, selon les différentes primitives cryptographiques sur lesquelles elles sont basées, la gestion distribuée de clés dynamique existante peut en outre être classifiée en trois catégories, à savoir, basée sur EBS, basée sur le partage de secret polynomial et basée sur le numéro de séquence déterministe. D'autre part, la gestion de clés dynamique centralisée peut également être classée selon la structure de réseau à savoir : réseau plat, réseau hiérarchique ou réseau hétérogène. La Figure 3 présente cette classification, ainsi qu'une liste des schémas pour chaque classe.

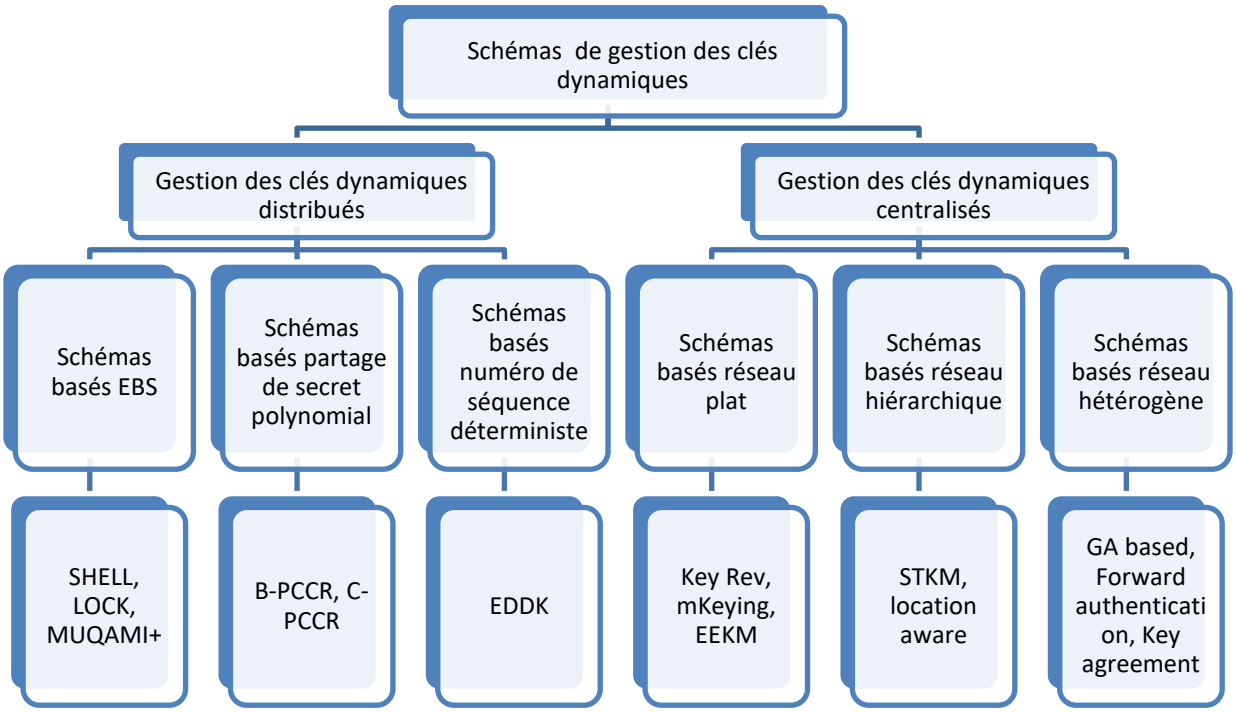


**Figure 3 : Classification des schémas de gestion des clés dynamiques.**

## 5.1 Quelques schémas de gestion des clés dynamiques

Le système de base d'exclusion (EBS) [88] est une formulation combinatoire du problème de gestion de clé par groupe qui permet de produire des résultats optimaux par rapport aux paramètres n, k et m, où n est la taille du groupe, k est le nombre de clés stockées par chaque membre, et m est le nombre de messages de rekeying. EBS utilise une technique pour déterminer les valeurs optimales de k et m en fonction n, et décrire le compromis entre k et m. Dans les schémas basés sur EBS, chaque nœud n est assigné à k clés d'un pool de taille P = k + m (1 <k; m <n, où n est le nombre de nœuds capteurs dans le réseau). Le processus de Rekeying est déclenché sur l'ensemble du réseau pour renouveler les clés du réseau si une capture d'un nœud est détectée. Les m clés, non connues par le nœud capturé dans le réseau, sont utilisés pour remplacer les clés des nœuds non compromis. Dans le processus de rekeying, les clés de remplacement sont générées, cryptées avec toutes les m clés inconnues aux nœuds capturés et distribuées seulement aux autres nœuds qui connaissent collectivement les m clés.

L'inconvénient majeur de ce système est le coût du processus de rekeying dans le cas où un petit nombre de nœuds dans le réseau est compromis (Rekeying est déclenché sur l'ensemble du

réseau), ainsi que des informations sur l'ensemble du réseau pourraient être découvertes par un adversaire après une attaque réussie.

Raazi et. Al, ont proposé MUQAMI + [89], qui est une version avancée de MUQAMI. MUQAMI + est un système de gestion de clés léger, évolutif et distribué localement pour les réseaux de capteurs en cluster. Dans MUQAMI +, un grand nombre de nœuds d'un cluster partagent des clés communes. MUQAMI + est efficace non seulement pour l'actualisation périodique des clés, mais aussi pour la révocation d'un nœud compromis et il dispose d'un mécanisme d'authentification de nœuds efficace. En outre, MUQAMI + permet de déplacer le rôle d'une tête de grappe d'un nœud à l'autre au fil du temps. Ce qui permet de réduire la communication, le calcul, le sur débit de stockage et la consommation d'énergie des nœuds capteurs dans le réseau. MUQAMI + est basé sur la matrice du système de base d'exclusion (EBS) et sur les chaînes de clés. La chaîne de clés est un mécanisme d'authentification basé sur les mots de passe à usage unique de Lamport.

Dans EDDK (Energy-Efficient Distributed Deterministic Key Management for Wireless Sensor Networks) [90], les auteur proposent un schéma de gestion de clés déterministe et distribuée pour les RCSF, dans cette solution les clés par paires et les clés de cluster locales sont configurées via les informations de diffusion pendant la phase d'initialisation du réseau et aucun échange de messages ultérieur n'est nécessaire. Chaque nœud dans le réseau doit non seulement stocker les clés par paires et les clés de cluster locales, mais il doit également conserver ses propres clés publiques et privées. Par conséquent, la surcharge de communication est très faible. De plus, les clés par paires sont également totalement décentralisées. Par conséquent, la compromission de certains nœuds capteurs n'affectera aucune autre paire non-compromise. EDDK comporte trois phases: établissement de clés, transfert de données et maintenance des clés. La dernière phase inclut la mise à jour de clé, la révocation de clé compromise, la nouvelle jointure de nœud et la jointure de nœud mobile. Concernant l'établissement de clés pour les nouveaux nœuds et les nœuds mobiles, un mécanisme composite basé sur l'algorithme de signature numérique à courbe elliptique (ECDSA) est utilisé, dans lequel la consommation de ressources peut également être maintenue de manière très faible[87].

LOCK (localized combinatorial keying) [85] est un schéma de gestion de clés dynamiques basé sur EBS destiné aux réseaux de capteurs sans fil organisé en cluster. Ce dernier est structuré en trois niveaux d'hiérarchiques à savoir : une station de base (BS) en haut, suivie par des chefs de cluster (CL), ensuite des nœuds capteurs ordinaires. En LOCK, aucune information de pré-déploiement n'est supposée sur l'emplacement des nœuds. LOCK utilise deux couches de clés administratives EBS. La couche supérieure (niveau 1) est EBSb qui permet à la station de base de gérer les chefs de clusters en tant que groupe. La couche inférieure (niveau 0) implique un EBSCi pour chaque cluster Ci.

Lorsque [77] les nœuds sont déployés dans l'environnement, ils créent un ensemble de clés de sauvegarde. Ces ensembles de clés de sauvegarde sont partagés uniquement avec la station de base, et non avec les chefs de cluster locaux. Si un nœud est capturé, les autres nœuds sont reconfigurés localement afin que le nœud compromis ne puisse pas communiquer avec eux. Si un chef de cluster est compromis, la station de base lance un changement de clé au niveau des chefs de clusters.

# 6 La gestion des clés dans les réseaux de capteurs hétérogènes

Bien que le nombre de solutions de gestion des clés proposées pour les RCSF homogènes soient important, l'équilibre entre le niveau de sécurité et la consommation de ressources reste le problème majeur de ces solutions. Les réseaux de capteurs sans fil hétérogènes (HWSN) [32], [91]–[93]ont ouvert une nouvelle direction de recherche pour le problème de sécurité et offert plusieurs opportunités. En déployant des nœuds capteurs à haute capacité (HSN), les RCSF hétérogène surpassent les RCSF homogènes classiques. Les HSN sont équipés d'un processeur puissant, d'une haute capacité de stockage mémoire, une autonomie de batterie importante et peuvent communiquer sur de grandes distances. L'architecture de réseau hétérogène est divisée en deux niveaux où les tâches qui exigent des ressources élevées sont attribuées aux nœuds HSN et les tâches qui nécessitent des ressources restreintes sont déléguées à des nœuds capteurs simples (LSN). Les RCSF hétérogènes offrent des avantages beaucoup plus importants que les RCSF homogènes pour un ensemble varié d'applications de sécurité. Le schéma de gestion de clés peut également bénéficier de ce type de réseaux en exploitant les capacités des nœuds puissants HSN [94].

## 6.1 Etat de l'art sur les schémas de gestion des clés dans les RCSF hétérogènes

Dans cette section, un bref aperçu des travaux connexes sur les principaux protocoles de distribution de clés dans les HWSN est présenté. Seuls quelques travaux ont abordé le problème de la distribution des clés dans les HWSN et ce domaine de recherche est encore sous-exploré.

Dans [95], les auteurs ont proposé un schéma de gestion de clés asymétriques appelé AP qui comprend trois phases: phase de pré-distribution de clés, phase de découverte de clés partagées et phase de configuration de clés par paire. Premièrement, un grand groupe de clés P et les identifiants de clés correspondants sont générés dans la phase de pré-distribution de clé. Après cela, les L clés sont sélectionnées aléatoirement à partir du pool de clés (sans remplacement) pour être pré-chargées dans chaque L-nœud capteur (nœud de capacité réduite). Les L clés forment un anneau de clés dans chaque L-nœud capteur. Les clés M (M >> l) sont également sélectionnées aléatoirement (à partir du pool de clés sans remplacement) pour être pré-chargées dans chaque H -nœud (nœud de capacité importante) capteur. De plus, une clé spéciale KH (connue par la station de base, mais pas par les L-nœuds capteurs) est pré-chargée dans chaque H-nœud capteur.

Dans la phase de découverte de clés partagées, les nœuds capteurs tentent de trouver les clés partagées (le cas échéant) en diffusant un message en clair (non crypté) qui contient la liste de clés à leurs voisins (chemin distribué) ou en l'envoyant au chef du cluster correspondant, qui prend en charge la découverte de la clé partagée (approche centralisée). Malgré le bon niveau de sécurité produit par le protocole AP [95], il ne gère pas les contraintes de limitation des ressources des L-nœuds capteurs. En effet, les L-nœuds ne peuvent pas être pré-chargés avec un grand nombre de clés en raison de la limitation de la mémoire. Dans la phase de découverte de clés partagées, les auteurs supposent que chaque L-nœud capteur diffuse la liste des ID de clés sur son anneau de clés vers son voisin ou CH (cluster head) pour établir une clé par paire qui apporte plus de frais supplémentaires au réseau. De plus, le protocole AP introduit un débit de calcul élevé dans le cas où les L-nœuds capteurs ne partagent aucune clé pré-chargée avec des voisins, puisqu'ils doivent établir une clé par paire basée sur un système cryptographique inefficace en termes de ressources.

Un protocole d'établissement de clés authentique efficace en ressources appelé RAKE a été proposé dans [96]. Les auteurs ont introduit deux étapes d'établissement de clés partagées, verticales et horizontales, pour la conception du schéma RAKE. Trois niveaux de nœuds de réseau sont définis avant le déploiement: la station de base B, les têtes de groupe (c'est-à-dire les capteurs H) et les membres (c'est-à-dire les capteurs L). Des clés sages symétriques verticales sont générées et partagées respectivement entre les capteurs B et H, les capteurs B et L, ainsi que les capteurs H et L, qui peuvent être considérés comme un partage des clés entre les nœuds parents et enfants. Pour réduire l'espace mémoire requis pour le stockage des clés, chaque clé partagée entre deux nœuds est stockée uniquement dans l'un des nœuds (nœud enfant) et calculable efficacement par l'autre nœud (nœud parent). Par conséquent, moins d'espace mémoire est nécessaire pour le stockage des clés car le nombre de nœuds parents est beaucoup plus petit que celui des nœuds enfants. En effet, le protocole RAKE est efficace sur le plan des ressources, mais il ne gère pas les réseaux à grande échelle où le nombre de nœuds hétérogènes est beaucoup plus élevé, ce qui implique plus d'espace de stockage pour les clés sur les capteurs L. Pour surmonter cette limitation, les auteurs supposent dans ce cas qu'une partie des clés pourrait être stockée dans des capteurs H. Cependant, les capteurs L seront obligés de calculer les clés partagées qui introduisent plus de temps de calcul pour ces nœuds capteurs limités en ressources.

Un schéma d'échange de clés authentifiées pour les RCSF hétérogènes basé sur l'amélioration du Bloom Filter a été proposé dans [97]. L'idée principale est de proposer une version améliorée de l'algorithme IBKM [98], qui est adapté pour l'échange de clés authentifiés dans les réseaux de capteurs hétérogènes. Basé sur les résultats expérimentaux présentés. Le protocole proposé offre une bonne protection contre les attaques de capture de nœud et de force brute. En outre, il est évolutif et nécessite une faible communication. Cependant, le protocole présente certaines limites liées à l'espace de stockage que les nœuds capteurs prennent afin de sauvegarder le

vecteur de bit de filtre bloom. En outre, les auteurs supposent que la génération de clés privées (PKG) est basée sur des paramètres de systèmes ouverts qui peuvent réduire le niveau de sécurité.

Wenbin Yao et al ont proposé un schéma de gestion de clés de groupe efficace pour les réseaux de capteurs hétérogènes nommé WLKH [99][8]. Le protocole proposé est basé sur le schéma LKH ++ [100] permet de sécuriser la communication entre les nœuds capteurs, où une clé de groupe est partagée entre les nœuds capteurs pour crypter les informations acquises [99]. En effet, dans les schémas basés sur LKH ++, CH (Cluster Head) stocke les clés privées de chaque CM (Cluster Membre). Le schéma est non seulement faible aux attaques de compromissions du CH, mais le calcul des clés d'initialisation augmente également le temps de calcul sur CM. Pour réduire le calcul, le stockage et la surcharge du réseau introduite par le protocole LKH ++, les auteurs ont proposé de construire un arbre équilibré pour gérer la clé de groupe. L'organisation spéciale de l'arbre réduit les frais généraux de stockage. WLKH améliore également le processus d'initialisation d'arborescence de clés et de révocation de nœuds capteurs pour rendre WLKH plus approprié pour RCSF hétérogènes. Cependant, le protocole applique un cryptage de clé asymétrique pendant l'initialisation de l'arborescence de clés qui ne convient pas aux nœuds capteurs à ressources limitées. De plus, l'ajout ou la révocation d'un nœud impose une mise à jour de clé de tous les arbres et des clés de CM qui introduisent un surcoût de calcul supplémentaire.

Un schéma d'authentification léger a été proposé pour les réseaux de capteurs sans fil hétérogènes dans [101]. L'idée principale est d'utiliser des cartes à puce et des identités dynamiques pour prévenir les menaces contre la vie privée des utilisateurs et pour assurer une authentification mutuelle d'échange de clé. L'architecture de réseau hétérogène a été exploitée pour équilibrer efficacement la consommation d'énergie et prolonger la durée de vie du réseau. Les auteurs supposent que les têtes de cluster sont responsables de l'authentification mutuelle et de l'échange de clé, tandis que les nœuds capteurs génériques ne sont pas tenus de mener des opérations d'authentification lourdes. Après le déploiement, les têtes de cluster et les nœuds capteurs utilisent une clé secrète générée par la station de base pour établir la clé via le mécanisme d'établissement de clé sécurisé inconditionnellement de Das[102]. Cela permet à chaque tête de cluster d'établir des clés par paire qui sont partagées avec leurs nœuds capteurs et d'autres têtes de cluster pour protéger leur communication. De même manière, chaque nœud capteur peut établir des clés secrètes par paires avec son nœud voisin pour communiquer de manière sécurisée.

Un mécanisme d'établissement de clé inconditionnellement sécurisé pour RCSF hétérogène a été proposé dans [102]. L'idée principale est de garantir un haut niveau de résilience contre les attaques de capture de nœuds tout en offrant un bon compromis entre les coûts de communication, les coûts de calcul et la connectivité du réseau. Les auteurs utilisent le schéma de pré-distribution à

base de polynômes existants pour établir des clés par paires entre les têtes de groupe dans un réseau de capteurs. En outre, afin de faciliter l'établissement de clés par paires entre les nœuds capteurs réguliers dans un groupe, une version étendue du schéma de clés aléatoires par paires a été appliquée. Cependant, le protocole proposé présente certaines vulnérabilités en termes d'authentification (ne supporte pas l'authentification mutuelle) et de résistance à l'attaque de falsification et à l'attaque d'usurpation d'identité de l'utilisateur.

Un schéma de gestion dynamique des clés a été proposé dans [103] pour les réseaux de capteurs hétérogènes. Les auteurs ont proposé une nouvelle méthode de gestion des clés basée sur un schéma de sécurité qui charge une fonction de hachage dans la station de base, les têtes des clusters clusters et les nœuds capteurs. Ensuite, les têtes des clusters et les nœuds capteurs génèrent leurs propres chaînes de clés pour fournir une authentification directe en cas de changements de clé dus à des failles de sécurité. De plus, des clés par paires sont établies entre les têtes de grappe et les nœuds capteurs pour assurer la confidentialité de transmission pour chaque message, protéger l'intégrité des données et déterminer si les nœuds capteurs sont malveillants. La chaîne de clé est composée de clés continues et la clé à laquelle elle dépend. Cela permet au nœud capteur de confirmer la validité de chaque clé [103]. Une fonction de hachage est utilisée pour calculer la clé, compresser la longueur des données et éviter la collision de données. Pour réduire les besoins en mémoire des nœuds capteurs, quelques clés et une fonction de hachage doivent être stockées à la fois. Les résultats de la simulation démontrent l'efficacité du système proposé en diminuant le nombre de clés requises pour les nœuds capteurs et les têtes de grappes. Cependant, tout le système de sécurité est basé sur une chaîne de clés de groupe partagée, qui réduit considérablement la résilience du protocole aux attaques de sécurité.

[104] Dans cet article, les auteurs ont proposé un nouveau schéma de gestion de clés basé sur la cryptographie à courbe elliptique et la méthode de cryptage pour les RCSF hiérarchiques hétérogènes. Le schéma proposé en tant qu'infrastructure sécurisée présente une mobilité supérieure des nœuds capteurs et une évolutivité du réseau. De plus, les auteurs ont proposé une authentification périodique et un nouveau mécanisme d'enregistrement pour la prévention de la compromission du nœud capteur. En outre, le schéma proposé n'augmente pas le nombre de clés dans les nœuds capteurs et présente un surdébit de communication et de calcul raisonnable par rapport aux autres schémas.

Un mécanisme de sécurité efficace hybride pour des réseaux de capteurs hétérogènes a été proposé dans [105]. Les auteurs ont introduit un schéma probabiliste de gestion des clés déséquilibrées dans lequel le nombre de clés stockées par chaque nœud dans le réseau est proportionnel à ses ressources [105]. En effet, les nœuds avec plus de ressources intrinsèques sont

responsables d'une plus grande proportion des communications et de la surcharge mémoire associée à la sécurité. Un certain nombre de modèles de confiance pour l'établissement de clé ont été également proposé, ils sont associés à un protocole d'établissement de clé multimodale. Des évaluations approfondies des performances ont été menées par les auteurs pour prouver la taille réduite du code et la robustesse à la compromission du nœud.

Mizanur Rahman et al [106] ont proposé un nouveau protocole d'échange de clés (que l'on appellera PKA ci-après) qui utilise une cryptographie basée sur l'appariement sur une courbe elliptique. Dans ce protocole, deux nœuds quelconques qui ont besoin de communiquer peuvent calculer indépendamment la même clé secrète en utilisant des propriétés d'appariement et de chiffrement basées sur l'identité [106]. Le protocole proposé exploite la capacité de mémoire élevée des capteurs H pour mémoriser les identités (c'est-à-dire les clés publiques) de tous les capteurs L, tandis que les capteurs L stockent uniquement les informations sur leurs propres clés. Après le déploiement, une fonction d'appariement bilinéaire basée sur la courbe elliptique est appliquée pour calculer des cléspar paire en utilisant les clés pré-chargées. Les auteurs affirment que le protocole proposé peut réduire considérablement l'espace clé d'un nœud. En outre, il est robuste contre un certain nombre d'attaques, y compris l'attaque de trou ver, les attaques de mascarade, les attaques de rejoue et les attaques de manipulation de message [106].

CL-EKM (Un protocole de gestion de clé efficace sans certificat) a été proposé dans [107] pour sécuriser les communication dans les RCSF hétérogènes mobiles dynamiques. En CL-EKM assure la confidentialité des clés et prend en charge des mises à jour de clés lorsqu'un nœud quitte ou rejoint un cluster. En outre, le protocole réduit l'impact de l'attaque de capture de nœud en introduisant une révocation de clé efficace pour les nœuds compromis. Dans [108], les auteurs ont proposé un système de gestion des clés efficaces (EKM) pour les scénarios basés sur la communication multipartite. EKM est basé sur un algorithme de génération de clés polynomiale amélioré qui réduit la charge de calcul lors de la jonction de nœuds dans des clusters. En effet, pour calculer le polynôme, EKM applique XOR de valeurs sélectionnées au hasard au lieu de la multiplication séquentielle utilisée dans le travail connexe existant. Les résultats de la simulation présentés par les auteurs ont démontré l'efficacité du protocole EKM en termes d'utilisation de mémoire, de faible calcul et de surcharge de communication.

Suman et al [109] ont proposé une nouvelle technique qui utilise des clés aléatoires basée sur la mimétique. Les auteurs utilisent une combinaison de schémas de distribution de clés aléatoires avec des concepts mimétiques pour fournir une sécurité robuste aux RCSF. Les résultats obtenus prouvent que la méthode proposée est économe en énergie par rapport aux autres techniques cryptographiques largement utilisées comme l'ECC et le RSA, tout en luttant contre les attaques par

usurpation d'identité [109]. Dans [110], les auteurs ont proposé un algorithme de gestion de clés bio-inspiré basé sur une fonction pseudo-aléatoire appelée E-BIOSARP. L'idée principale est d'améliorer le protocole BIOSARP (BIOlogy-Inspired Self-organized Secure Autonomous Routing Protocol) [111] en impliquant des mesures de sécurité actives qui permettent la sécurité globale de la communication réseau. Deux clés maîtresses (k1 et k) générées avec une fonction pseudo-aléatoire sont injectées dans chaque nœud capteur. Le k1 est une clé principale pour le nouveau nœud, et k est une clé principale pour tous les nœuds [110]. La performance d'E-BIOSARP a été analysée en faisant varier le nombre de nœuds malveillants dans le réseau. Les résultats de la simulation ont démontré les améliorations de performances du protocole E-BIOSARP et confirment l'efficacité de la sécurité.

# 7 Conclusion

Les schémas de gestion des clés constituent l'épine dorsale de tous les protocoles de sécurité conçus pour les RCSF. L'objectif principal d'un système de gestion des clés est de fournir la première ligne de défense afin d'assurer une communication sécurisée tous en assurant la monodiffusion, la multidiffusion et la diffusion.

Après cette étude détaillée concernant les schémas de gestion des clés, nous concluons qu'il est difficile pour un système de gestion de clés de respecter toutes les contraintes et limitations exigées par les RCSF. Précisément, il est très difficile de concevoir un schéma de gestion de clé optimal adapté à toutes les topologies et architectures des réseaux de capteurs et leurs applications. D'une manière générale, la gestion des clés est un problème qui n'est pas encore résolu dans les RCSF, et le domaine de recherche reste toujours ouvert pour proposer des schémas équilibrés en termes de niveau de sécurité, ressources et domaines d'application. Pour cela, nous constatons qu'une architecture hétérogène du réseau de capteurs offre un avantage important qui permet d'alléger la conception d'un schéma de gestion des clés même avec les contraintes sévères exigées par les RCSF. L'hétérogénéité des capteurs permet de déléguer les tâches lourdes aux nœuds puissants (HSN) qui possèdent des ressources importantes par apport aux nœuds ordinaires (LSN), ainsi, qu'une gestion de clés dynamiques permet d'augmenter considérablement le niveau de sécurité des schémas de gestion des clés. Dans la deuxième partie consacrée à notre contribution, nous allons proposer un schéma de gestion des clés dynamiques basé sur une architecture hétérogène du réseau de capteurs sans fil.

# Partie 2 :

# Contribution

# CHAPITRE 5 :

# UN MECANISME EFFICACE DEDIE À LA GESTION DES CLES ET L'AUTHENTIFICATION

# 1 Introduction

Ce chapitre est consacré à la présentation de notre nouveau mécanisme de sécurité, nommé EDAK ou « Efficient Dynamic Authentication and Key management mechanism » [112], dédié pour la gestion et la distribution efficace des clés ainsi que la garantie d'authentification des nœuds légitimes dans le réseau. La première partie de ce chapitre est consacrée à la présentation du modèle de communication (protocole de routage et accès au média de transmission) ainsi que l'architecture réseau sur laquelle se base notre mécanisme de sécurité. Ensuite, nous présentons le principe de fonctionnement et le point clé de notre mécanisme de sécurité ainsi que les différents algorithmes de gestion des clés et d'authentification. L'idée principale est de fournir un seul protocole léger pour l'authentification et l'établissement de clés tout en optimisant le niveau de sécurité.

# 2 Préliminaires

Dans cette section, nous décrivons les hypothèses sur l'architecture du réseau sur laquelle se base notre proposition, ainsi que quelques paramètres essentiels nécessaires pour une meilleure compréhension de notre mécanisme de sécurité.

## 2.1 Modèle de réseau hétérogène

Nous supposons une architecture réseau hétérogène où trois types de nœuds capteurs sont déployés: un nombre restreint de nœuds capteurs avec un niveau élevé de ressources appelés HSN (High capacity Sensor Node), un grand nombre de nœuds capteurs simples avec un faible niveau de ressources appelés LSN (Low capacity Sensor Node) tels que les nœud capteurs MicaZ [21] et une station de base avec des ressources illimitées (bande passante, énergie, mémoire et traitement) disponibles pour traiter toute information si nécessaire.

### 2.1.1 Formation du cluster

Dans notre travail, nous supposons une architecture de réseau hiérarchique basée sur le principe du clustering. Dans cette architecture, les nœuds capteurs sont déployés de manière aléatoire et uniforme. Pour organiser le réseau, les capteurs sont divisés en clusters où chaque cluster est géré par une tête de cluster (CH : Cluster Head). En outre, chaque groupe est divisé en plusieurs sous-groupes (SC :Sub Cluster) avec un nœud HSN, en tant que chef de sous groupe (SCH) comme le montre la Figure 4. Le rôle du CH et du SCH est périodiquement attribué aux nœuds HSN appartenant au cluster. D'un autre côté, les LSN servent comme membres de cluster et de sous-cluster dont le rôle est la collecte des informations de surveillance. Nous supposons que chaque HSN possède une grande capacité d'écoute et peut communiquer directement avec ses LSN voisins dans le sous-cluster, ainsi que tous les HSN de son cluster. Les nœuds CH peuvent écouter et communiquer directement avec tous leurs membres HSN et tous les CH du réseau. De plus, nous

supposons que la BS peut écouter et communiquer avec tous les HSN du réseau. De l'autre côté, les LSN ont une petite portée d'écoute et ne peuvent intercepter que la transmission de leur voisin proche.

Nous supposons que la station de base est responsable de la formation des clusters du réseau, dessous-clusters ainsi que la sélection du CH et du SCH en se basant sur le même algorithme proposé dans [17]. Cette supposition est basée sur le fait que la réorganisation centralisée du réseau offre un meilleur choix pour la gestion efficace des ressources et des performances de sécurité.

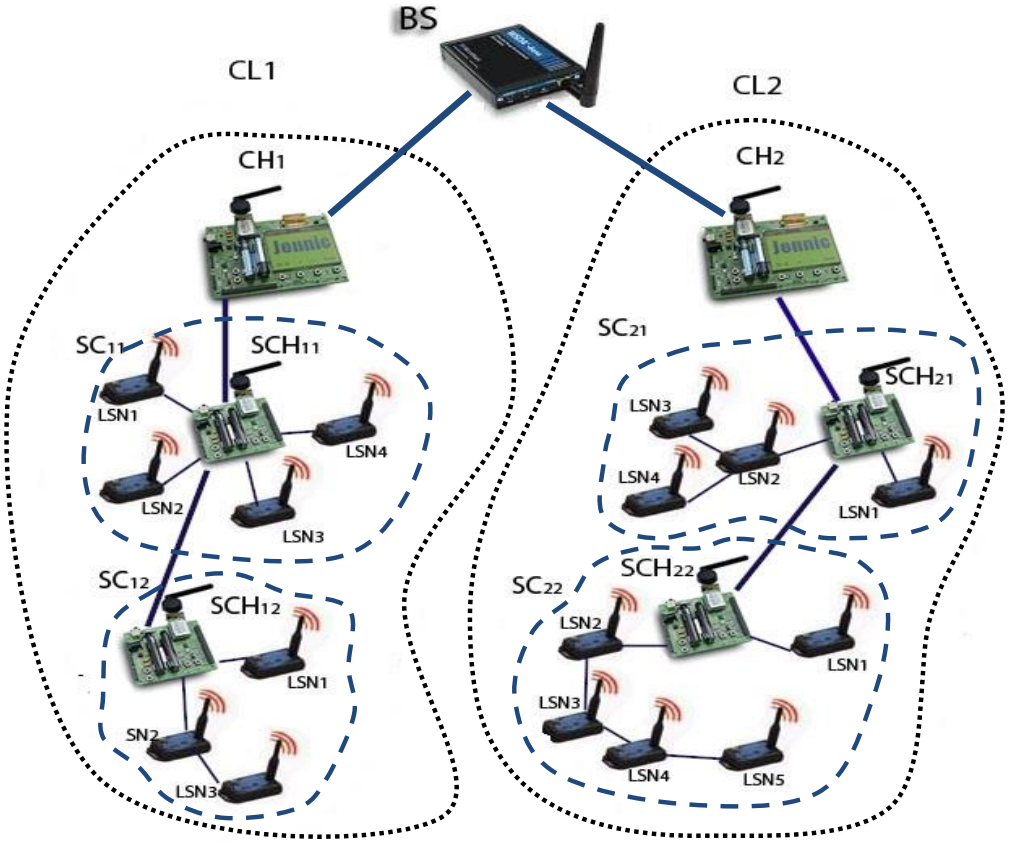


**Figure 4 : Modèle de réseau.**

### 2.1.2 Protocole de routage

Notre mécanisme de sécurité est construit autour d'un protocole de routage multi-sauts [113] où tous les LSN transmettent leurs données collectées, via leurs voisins LSN, à leur SCH. Chaque SCH agrège les données reçues et envoie le résultat directement au CH correspondant ou via les SCH voisins (si le CH est trop loin). Après, les CH envoient les données collectées directement à la station de base ou via les CH voisines. Le protocole de routage est divisé en deux phases: la phase d'initialisation où le réseau est réorganisé périodiquement et les CHs, SCH sont réélus et la phase de transmission où les données collectées sont transmises à la BS.

### 2.1.3 Protocole Mac

Au niveau de la couche MAC, un protocole basé sur le principe du duty-cycling est supposé gérer les périodes de réveil et de sommeil des nœuds capteurs. Par conséquent, nous adoptons le même mécanisme proposé dans le protocole SMAC [114] où les nœuds capteurs forment un groupe de nœuds voisins (SC) en fonction de leur plan de réveil et de sommeil. Dans notre schéma, tous les membres du Sub-cluster partagent le même plan de réveil et de sommeil.

## 2.2 Format des paquets de données

Nous adoptons le même format des paquets de données proposé dans [115] qui divise le paquet en trois parties: l'en-tête (2 octets), les données (29 octets) et la partie remorque ou Trailer (1 octet). Dans notre proposition, nous supposons qu'un octet de la partie des données est réservé au code d'authentification $Code_{Auth}$ (que nous détaillerons ultérieurement), où chaque nœud ajoute à ses paquets transmis le $Code_{Auth}$ calculé. La position du $Code_{Auth}$ change à chaque paquet transmis sur la base d'un algorithme d'allocation dynamique qui sera détaillé dans la sous-section 4.5.

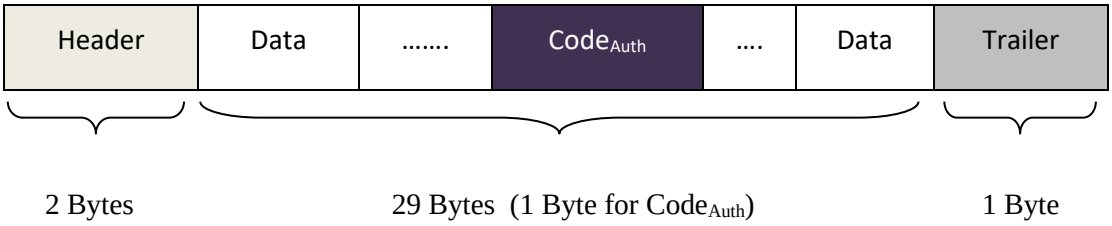


**Figure 5 : Format de Paquet de Donnée**

## 2.3 Notations

Cette sous-section fournit une liste de notations qui seront utilisées pour la définition de notre mécanisme de sécurité. Ces notations, qui seront expliquées en détail dans la section suivante, comprennent:

| Notation | Definition |
|---|---|
| BS | Station de base |
| HSN | Nœud capteur de haute capacité |
| LSN | Nœud capteur de faible capacité |
| CL | Cluster |
| SC | Sous-cluster |
| SCH | Tête de sous-cluster |

| | |
|---|---|
| DK | Clé dynamique |
| DMK | Clé de matrice dynamique |
| IK | Clé initiale du réseau |
| VIK | Vecteur de clé initiale |
| PK | Clé partielle |
| LDK | La clé linéaire par paire dynamique |
| ODK | Clé dynamique orthogonale de groupe |
| DDK | Clé dynamique diagonale de groupe |
| $Code_{Auth}$ | Code d'identification |
| $LCode_{Auth}$ | Le code d'authentification linéaire |
| $DCode_{Auth}$ | Le code d'authentification diagonal |
| $OCode_{Auth}$ | Le code d'authentification orthogonal |

**Tableau 1 : Liste des notations.**

# 3 Le mécanisme de sécurité proposé (EDAK)

Dans cette section, nous présentons le mécanisme d'authentification dynamique et de gestion des clés proposé, que nous intitulons EDAK (Efficient Dynamic Authentication and Key management mechanism) [112]. L'idée principale est de mettre en place un mécanisme léger d'authentification et de distribution des clés, tout en optimisant le niveau de sécurité. Nous proposons un algorithme d'établissement de clés efficace pour créer des clés de paires entre les nœuds LSN, des clés de groupe entre les SCH dans le CL et une clé de groupe pour les CH et la station de base (BS). Pour optimiser le niveau de sécurité et empêcher la capture des clés, un processus de génération de clés dynamiques est introduit, dont l'objectif et de créer une nouvelle clé pour chaque message transmis sans nécessiter d'échange d'informations supplémentaires. Un mécanisme d'authentification est également proposé pour identifier les nœuds capteurs légitimes. Pour générer les clés de chiffrement, l'algorithme EDAK est basé sur une clé de matrice dynamique DMK qui est générée en utilisant des informations préexistantes. Chaque nœud capteur dans le réseau crée et maintient un DMK pour générer des clés de cryptage / décryptage pour crypter ses données transmises et décrypter les données reçues de ses voisins.

## 3.1 La matrice de clé dynamique DMK

La clé *DMK* doit être régénérée avant chaque phase de transmission de données. La clé DMK est également mise à jour après chaque paquet de données transmis ou intercepté dans le groupe d'écoute des nœuds. Pour une meilleure explication, nous considérons $DMK_t$ la clé matricielle dynamique générée par le nœud *t*. La première cellule matricielle $DMK_{t,\,0}$ représente l'ID du nœud

capteur (avec une longueur de 16 bits) où *t* est l'indice de ligne qui correspond à l'ID du nœud. Le $DMK_{t,\ 1}$ (avec une longueur de mémoire de 29 octets) représente les dernières données transmises par le nœud capteur correspondant *t*. Les cellules $DMK_{t,\ j}$ (j = 2, .., 33) représentent le $PK_{t,\ j}$ (Partial Keys) qui sera utilisé pour générer la clé dynamique $DK_t$ utilisée par le nœud émetteur pour chiffrer les paquets de données. Le $PK_{t,\ j}$ est calculé en soustrayant les nouvelles données transmises du dernier. En effet, nous proposons d'utiliser la différence entre les données transmises pour réduire l'espace mémoire nécessaire à la matrice *DMK*. La plupart des RCSF sont déployés pour surveiller l'environnement (détection incendie, surveillance météo ...), où les variations des données transmises sont faibles. Nous supposons que chaque $PK_{t,\ j}$ prend une longueur de mémoire de 4 bits.

$DMK_{i,\ 0}$ (i = 0 .. n, i ≠ t, où n est le nombre de nœuds dans la zone d'écoute du nœud *t* et *i* représente l'indice de ligne dans *DMK* qui correspond au nœud à partir duquel le paquet sera intercepté dans le groupe d'écoute ) représente tous les IDs des nœuds capteurs dans la zone d'écoute du nœud *t* (la zone de couverture radio du nœud *t*). Le $DMK_{i,\ 1}$ (i = 0 .. n, i ≠ t) représente les dernières données interceptées dans le rang d'écoute. Les cellules $DMK_{i,\ j}$ (i = 0 .. n; j = 2 .. 33) représentent le $PK_{i,\ j}$ (Partial Keys) qui sera utilisé pour générer les clés dynamiques $DK_i$ pour déchiffrer les paquets reçus à partir des nœuds dans le groupe d'écoute.

$$DMK_t = \begin{pmatrix} ID_{node\ 0} & DATA_{interc} & PK_{0,2} & .. & PK_{0,33} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ ID_{node\ t} & DATA_{trans} & PK_{t,2} & .. & PK_{t,33} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ ID_{node\ n} & DATA_{interc} & PK_{n,2} & .. & PK_{n,33} \end{pmatrix}$$

**Figure 6 : La matrice de clés dynamiques DMK pour le nœud t.**

A chaque nouveau paquet transmis, le nœud capteur *t* calcule $PK_{t,\ (Lmod34)\ +2}$ et met à jour la cellule $DMK_{t,\ 1}$ (remplace les dernières données transmises par les nouvelles) L correspond au numéro de séquence du paquet. D'un autre côté, si le nœud t reçoit ou intercepte un paquet transmis du nœud y dans son groupe d'écoute, il met à jour le $DMK_{y,\ 1}$ (où y correspond à la ligne $DMK_t$ attribuée au nœud y) et calcule le $PK_{y,\ (\ L\ mod\ 34)\ +\ 2}$. La Figure 7 présente un exemple de mise à jour de la matrice DMK. Nous supposons la même topologie présentée dans le sous-cluster $SC_{12}$ (Figure 4), où trois nœuds LNS sont déployés.

$$DMK_{Lsn1} = \begin{pmatrix} 00000001 & 00..0000 & 0000 & \dots\ 0000 \\ 00000010 & 00..0000 & 0000 & \dots\ 0000 \\ 00000011 & 00..0000 & 0000 & \dots\ 0000 \end{pmatrix}$$

$$DMK_{Lsn\,3} = \begin{pmatrix} 00000001 & 00..0000 & 0000 & \dots\ 0000 \\ 00000010 & 00..0000 & 0000 & \dots\ 0000 \\ 00000011 & 00..0000 & 0000 & \dots\ 0000 \end{pmatrix}$$

$$DMK_{Lsn2} = \begin{pmatrix} 00000001 & 00..0000 & 0000 & \dots\ 0000 \\ 00000010 & \mathbf{00..0101} & \mathbf{0101} & \dots\ 0000 \\ 00000011 & 00..0000 & 0000 & \dots\ 0000 \end{pmatrix}$$

(a). Au début de la communication, le nœud 2 envoie les données mesurées (0101) à son nœud voisin dans le chemin de routage.

$$DMK_{Lsn\,3} = \begin{pmatrix} 00000001 & 00..0000 & 0000 & \dots\ 0000 \\ 00000010 & \mathbf{00..0101} & \mathbf{0101} & \dots\ 0000 \\ 00000011 & 00..0000 & 0000 & \dots\ 0000 \end{pmatrix}$$

$$DMK_{Lsn1} = \begin{pmatrix} 00000001 & 00..0000 & 0000 & \dots\ 0000 \\ 00000010 & \mathbf{00..0101} & \mathbf{0101} & \dots\ 0000 \\ 00000011 & 00..0000 & 0000 & \dots\ 0000 \end{pmatrix}$$

(b) Tous les nœuds de la zone d'écoute mettent à jour leurs DMK en fonction des données reçues.

$$DMK_{Lsn2} = \begin{pmatrix} 00000001 & 00..0111 & 0000 & 0000 \dots\ 0000 \\ 00000010 & \mathbf{00..1000} & \mathbf{0101} & \mathbf{0011} \dots\ 0000 \\ 00000011 & 00..1010 & 0000 & 0000 \dots\ 0000 \end{pmatrix}$$

(c). Le nœud 2 a envoie de nouvelles données captées (1000).

$$DMK_{Lsn1} = \begin{pmatrix} 00000001 & 00..0111 & 0000 & 0000 \dots\ 0000 \\ 00000010 & \mathbf{00..1000} & \mathbf{0101} & \mathbf{0011} \dots\ 0000 \\ 00000011 & 00..1010 & 0000 & 0000 \dots\ 0000 \end{pmatrix}$$

$$DMK_{Lsn3} = \begin{pmatrix} 00000001 & 00..0111 & 0000 & 0000 \dots 0000 \\ 00000010 & \mathbf{00..1000} & \mathbf{0101} & \mathbf{0011} \dots 0000 \\ 00000011 & 00..1010 & 0000 & 0000 \dots 0000 \end{pmatrix}$$

(d). Tous les nœuds de la zone d'écoute mettent à jour leurs DMK en fonction des nouvelles données reçues.

**Figure 7 : Exemple de mise à jour du DMK.**

## 3.2 Phases d'établissement des clés EDAK

Le processus d'établissement des clés EDAK passe par deux phases: la phase de pré-déploiement et la phase de génération des clés.

### 3.2.1 Phase de pré-déploiement

Les nœuds capteurs sont pré-chargés avec une clé de réseau unique symétrique *IK* avant d'être déployés dans la zone cible. Cette clé *IK* n'est utilisée qu'une seule fois au début de chaque phase d'émission (après la formation des clusters, réélection CH et SCH). Pour un niveau de sécurité efficace, nous supposons que la longueur de la clé *IK* est fixée à 128 bits. Celle-ci représente la longueur de clé recommandée par la plupart des systèmes cryptographiques symétriques. Pour simplifier l'implémentation de la clé *IK*, un vecteur *VIK* est utilisé. La longueur de *VIK* est de 32 cellules où chacune d'elles représente 4 bits de la clé *IK*.

### 3.2.2 Phase de génération de la clé

La phase de génération de clé est basée sur un algorithme léger qui s'exécute à chaque transmission et réception des paquets pour assurer un niveau de sécurité élevé. En effet, chaque clé générée est utilisée une seule fois et une nouvelle clé sera régénérée (en se basant sur la mise à jour *DMK*) pour la prochaine communication. Avant de mettre à jour sa matrice *DMK* (calculer le $DMK_{t,(Lmod\ 34)\ +2}$ correspondante et mettre à jour $DMK_{t,\ 1}$) le nœud émetteur génère la clé de cryptage dynamique *DK* pour crypter le paquet à transmettre. Le nœud récepteur utilise sa matrice DMK pour générer le même DK appliqué par le nœud émetteur pour déchiffrer le paquet reçu. La Figure 8 explique le processus d'établissement de clé:

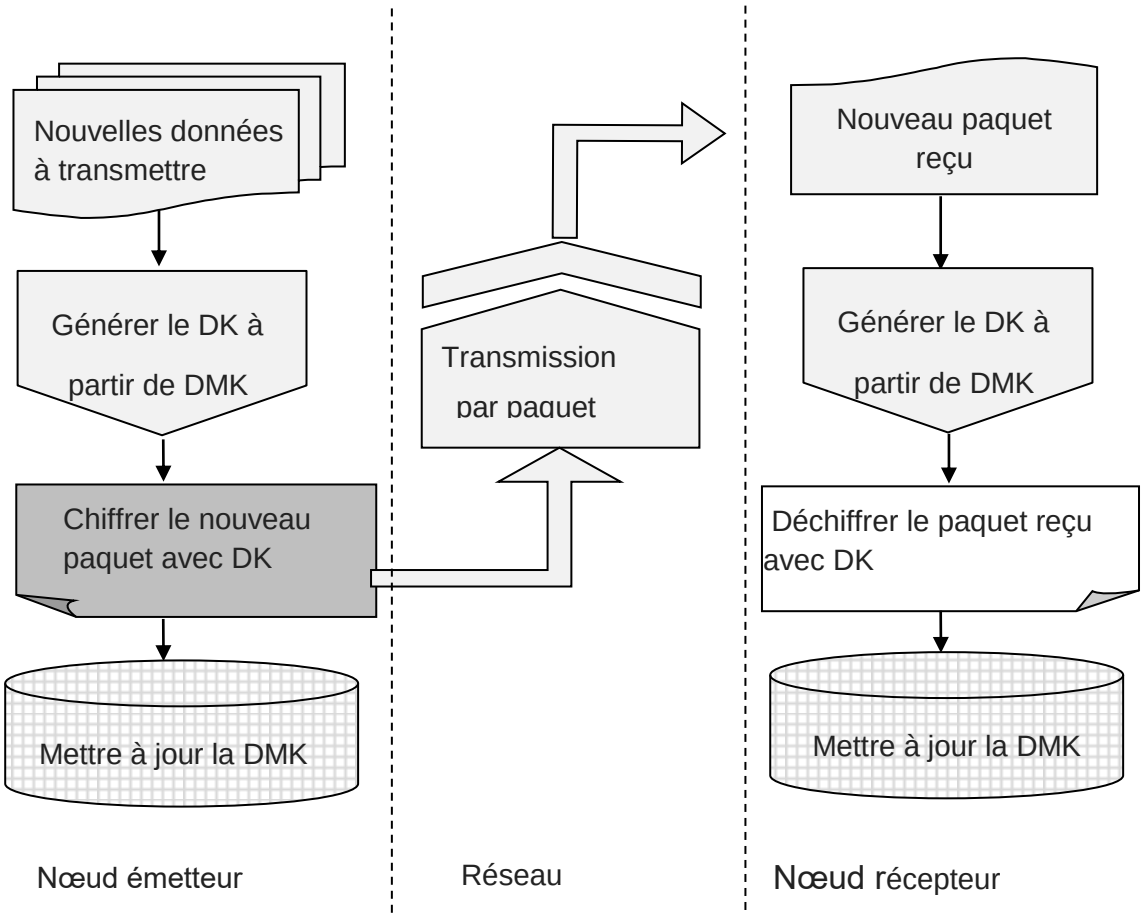


**Figure 8 : Processus d'établissement de clé.**

Le processus d'établissement de clé est divisé en deux niveaux: inter cluster et intra cluster.

### *3.2.2.1 Le niveau Inter cluster*

Deux types de clés dynamiques sont générés dans le cluster: les clés de paires linéaires et les clés de groupe orthogonales. La clé linéaire est établie pour sécuriser la communication entre les nœuds LSNs ou entre les nœuds LSNs et les nœuds HSNs. La clé de groupe orthogonale est appliquée entre les nœuds HSNs dans le réseau pour crypter et décrypter les données échangées. Nous décrivons trois cas dans le processus d'établissement de la clé inter-cluster:

***Cas 1: Etablissement de clés de paires linéaires entre nœuds LSN***

Le processus de génération de clé linéaire de pair *LDK* est basé sur la concaténation de tous les $DMK_{i,j}$du nœud correspondant. Par exemple, pour générer le $LDK_{encrypt}$, utilisé pour crypter les données transmises, le $node_x$ concatène tous les $DMK_{x,j}$ (où j correspond à j = 2 ... (L mod 34) de son DMK. Dans l'autre cas, générer la clé $LDK_{decrypt}$ utilisé pour décrypter, le $node_x$ concatène tout les

$DMK_{y,\,j}$ de son DMK (où y correspond à la ligne DMK, du $node_x$, attribué au $node_y$) .La Formule 4 explique comment $LDK_{encrypt}$ et $LDK_{decrypt}$ sont générés:

$$LDK_{encrypt} = DMK_{x,2} \,||\, DMK_{x,3} \,||\ldots\ldots||\, DMK_{x,33}$$

$$LDK_{decrypt} = DMK_{y,2} \,||\, DMK_{y,3} \,||\ldots\ldots||\, DMK_{y,33}$$

**Équation 4 : Génération de $LDK_{encrypt}$ et $LDK_{decrypt}$.**

Au début de la phase de communication, la longueur de concaténation $DMK_{i,\,j}$ peut ne pas atteindre 128 bits. Par conséquent, nous proposons que le résultat de la concaténation est concaténé avec la clé pré chargée *IK*. La formule suivante explique la génération de la clé *LDK* dans ce cas:

$$LDK = (DMK_{i,2} \,||\, DMK_{i,3} \,||\, \ldots\ldots \,||\, DMK_{i,m}) \,||\, (VIK_0 \,||\, VIK_1 \,||\, \ldots\ldots \,||\, VIK_{31-(m-2)})$$

**Équation 5 : Génération de la clé LDK inferieur à 128 bits.**

Où m (m <32) représente l'indice de la dernière cellule calculée dans la ligne correspondante du DMK.

La figure 6 présente un exemple de génération de la clé LDK. Nous supposons la même matrice DMK présentée dans la Figure 7.

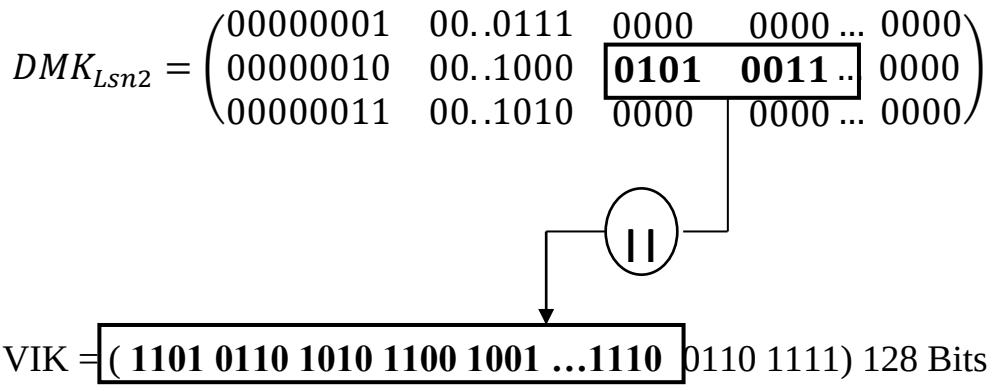


**Figure 9 : Exemple de génération de la clé LDK.**

***Cas 2: Établissement de clés de paires linéaires entre LSN et HSN***

La génération de la clé linéaire par paire entre LSN et HSN est basée sur la même méthode décrite précédemment dans le cas 1. La clé LDK générée est utilisée pour sécuriser la communication dans le sous-cluster où les LSN transmettent leurs données à leurs SCH.

***Cas 3: HSN à l'établissement de clé de groupe orthogonal HSN***

La clé de groupe dynamique orthogonale (ODK) est utilisée pour sécuriser la communication entre HSN (entre SCHs ou SCH et CH) à l'intérieur du cluster. En effet, les nœuds HSN sont le sujet de plusieurs attaques malveillantes du fait de leurs rôles importants (agrégation et transmission des données, gestion et contrôle des clusters et des sous-clusters ...); il est donc crucial d'appliquer un mécanisme de sécurité de haut niveau. L'algorithme de génération de clés ODK est plus efficace que celui dédié pour le LDK, et est basé sur la concaténation de la $DMK_{i,\ j}$ XOR $VIK_{i \bmod 32}$ ($i = 0 .. nbr_{Nbr\_HSN}$) où j est l'indice de colonne de la dernière cellule modifiée de l'DMK ($j> = 2$) et nbrNbr_HSN représente le nombre de nœuds HSN dans le cluster. La formule suivante explique comment la clé ODK est calculée dans le cluster:

$$ODK = (DMK_{0,\ j} \oplus VIK_{0 \bmod 32}) \,||\, (DMK_{1,\ j} \oplus VIK_{1 \bmod 32}) \,||\, .. \,||\, (DMK_{Nbr\_HSN,\ j} \oplus VIK_{Nbr\_HSN \bmod 32})$$

**Équation 6 : Génération de la clé ODK.**

En effet, si le nombre de nœuds HSN dans le cluster dépasse 32 (le nombre de lignes de la matrice DMK dépasse 32), la longueur de la clé ODK dépassera 128 bits. Par conséquent, une fonction de hachage sera appliquée pour ajuster la longueur de la clé ODK. Dans notre proposition, l'algorithme MD5 est adopté comme fonction de hachage. En outre, si le nombre de nœuds HSN est inférieur à 32, la clé ODK est concaténée avec l'IK pré chargé pour atteindre 128 bits. On suppose que la matrice DMK doit être triée par identifiant de nœud avant le processus de génération de clé. En outre, toutes les lignes DMK correspondant aux nœuds LSN doivent être exclues. Pour réduire la charge de calcul, le tri de la matrice DMK ne peut être effectué que sur la colonne concernée par la génération de clé (colonne qui correspond à la dernière cellule modifiée). La Figure 10 présente un exemple de génération de clé ODK.

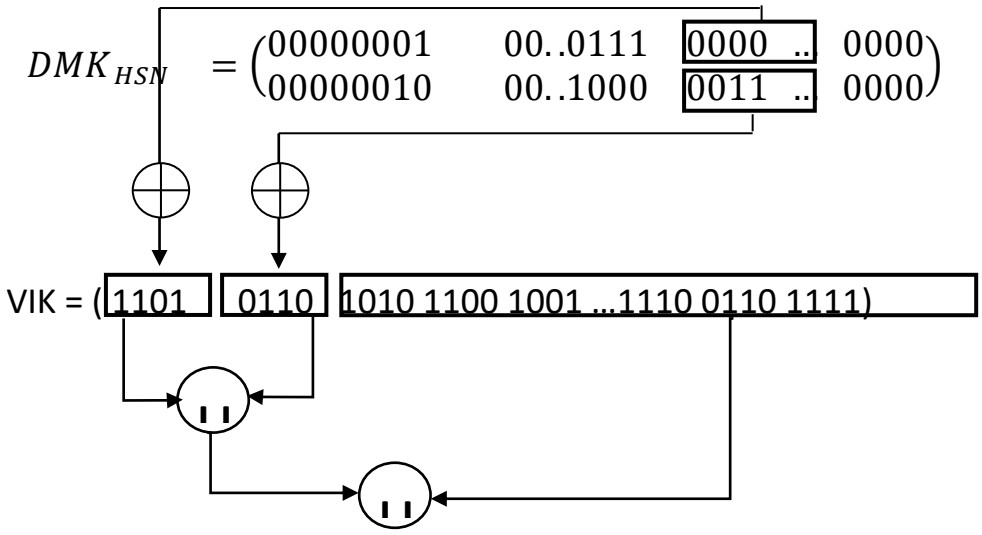


**Figure 10 : Exemple de génération de la clé ODK.**

### *3.2.2.2 Le niveau intra cluster*

Le niveau intra-cluster consiste en la génération d'une clé de groupe qui représente la clé de groupe dynamique diagonale DDK. Cette clé est utilisée pour sécuriser les communications entre les têtes de clusters (CH) et entre les CH et la BS.

Semblablement à la clé dynamique orthogonale ODK, la clé dynamique diagonale DDK introduit plus de niveau de sécurité dans le processus de génération de clé. Cependant, au lieu d'utiliser une seule dimension de la matrice DMK, le DDK est plus complexe et dynamique et utilise les deux dimensions de la matrice DMK. En effet, la génération de la clé DDK est basée sur la concaténation des cellules p $DMK_{p,p}$ en diagonale, où p est calculé (selon l'Équation 7) pour générer la matrice carrée de $DMK_{i, j}$.

$$P = \begin{cases} \text{Min } (nbr_{CH},m) \quad \text{if} \quad \text{Max}(nbr_{CH},m) \text{ Mod Min } (nbr_{CH},m) = 0 \\ \text{Min } (nbr_{CH},m) - 1 \quad \text{if} \quad \text{Max}(nbr_{CH},m) \text{ Mod Min } (nbr_{CH},m) = 1 \end{cases}$$

**Équation 7 : Calcul de p de la matrice carrée DMKp,p**

Pour calculer $DMK_{p,p}$, on exclut les deux premières colonnes de la matrice DMK qui représentent l'identifiant des nœuds et les dernières données transmises / interceptées dans le réseau intra-cluster.

Après avoir calculé le $DMK_{p,\ p}$ la formule suivante est utilisée pour calculer la clé de groupe dynamique diagonale DDK:

$$DDK = (DMK_{0,2} \oplus VIK_{0 \bmod 32}) \,||\, (DMK_{1,3} \oplus VIK_{1 \bmod 32}) \,||\, .. \,||\, (DMK_{p-2,\ p} \oplus VIK_{nbrCH \bmod 32})$$

**Équation 8 : Calcul de la clé DDK.**

Comme le processus de génération ODK, une fonction de hachage est appliquée pour ajuster la longueur DDK à 128 bits.

### 3.2.3 Révocation et ajout de nœud EDAK

Dans notre algorithme de distribution de clés proposé, l'ajout et la révocation de nœud capteurs est simple et efficace. Avant la révocation, le nœud capteur informe tous ses nœuds voisins dans le sous-cluster en envoyant un message de révocation. Après avoir reçu ce message de contrôle, chaque LSN du SC supprime la ligne DMK correspondante du nœud révoqué. En outre, tous les SCH dans le cluster et la BS mettent à jour leur DMK pour révoquer le nœud correspondant. D'autre part, l'ajout de nouveaux nœuds au réseau est autorisé à chaque phase de réorganisation du réseau, où les matrices DMK sont réinitialisées. Par conséquent, les nouveaux nœuds ajoutés doivent être pré chargés seulement avec la clé IK.

### 3.2.4 Algorithme d'authentification EDAK

Le deuxième objectif principal de notre travail est d'assurer l'authentification des nœuds capteurs légitimes dans le réseau. Par conséquent, le protocole EDAK introduit un nouvel algorithme d'authentification léger pour vérifier l'identité des nœuds capteurs tout en établissant la clé dynamique DMK. En effet, à chaque nouveau paquet transmis, le nœud capteur génère un code d'authentification $Code_{Auth}$ qui sera ajouté au paquet. Le $Code_{Auth}$ est composé de deux parties: $DATA_{dif}$ et $XORD_{dif}$. Le $DATA_{dif}$ représente le résultat de la soustraction entre les dernières données reçues et la nouvelle. Le $XORD_{dif}$ représente le résultat XOR des cellules DMK correspondantes. La Figure 11 décrit la structure de $Code_{Auth}$.

| $DATA_{dif}$(4 bits) | $XORD_{dif}$(4bits) |
|---|---|

**Figure 11 : Structure du code d'authentification.**

Après avoir reçu le paquet, le nœud récepteur peut authentifier l'émetteur de paquet en vérifiant le code d'authentification reçu en le comparant au $code_{Auth}$ calculé localement basé sur la matrice DMK. La figure suivante présente le processus d'authentification.

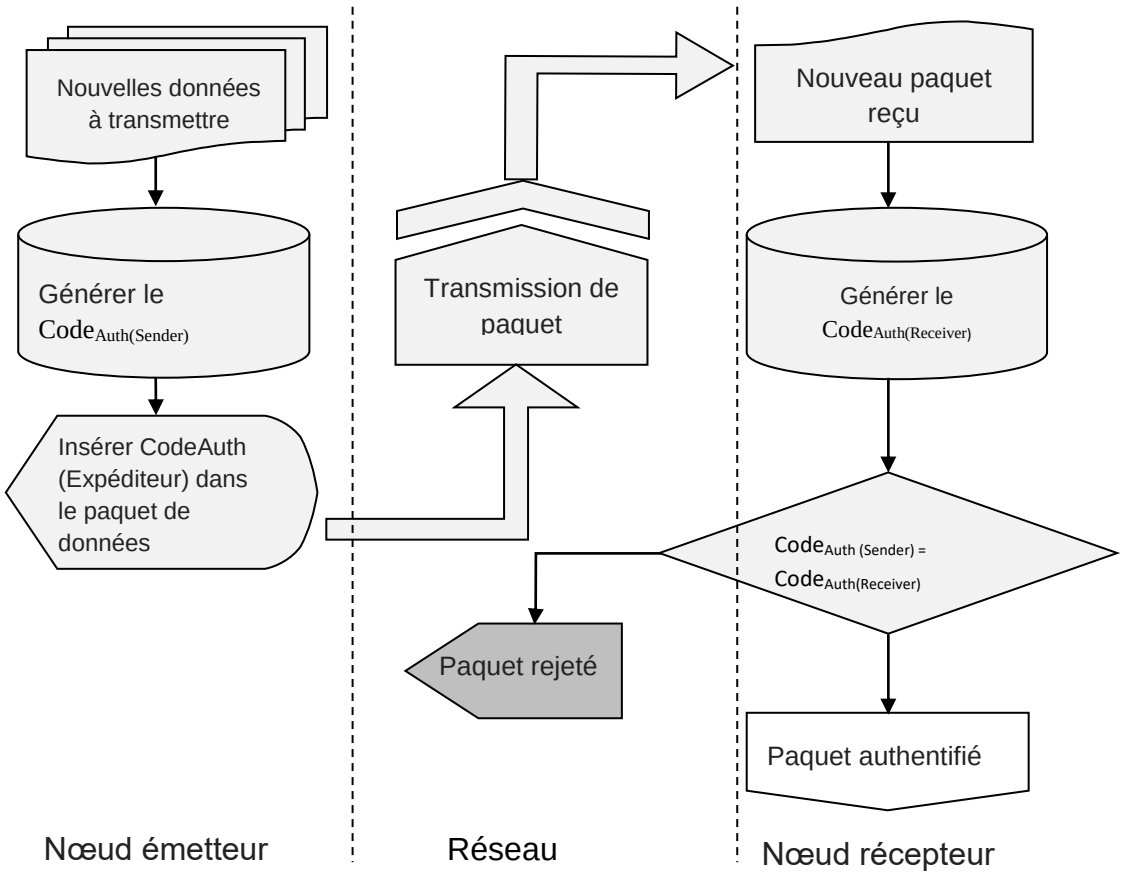


**Figure 12: Le processus d'authentification EDAK.**

L'algorithme d'authentification proposé est simple et ne nécessite pas d'échange de paquets de contrôle supplémentaires. En outre, il est plus approprié pour les nœuds capteurs limités en ressources car il n'implique pas une surcharge de calcul élevée. Pour optimiser le niveau de sécurité de l'algorithme d'authentification proposé, nous introduisons trois catégories de code d'authentification($Code_{Auth}$) basées sur le type de nœuds émetteur et récepteur, à savoir: $LCode_{Auth}$, $OCode_{Auth}$ et $DCode_{Auth}$.

#### *3.2.4.1 Le code d'authentification linéaire ($LCode_{Auth}$)*

Le code d'authentification linéaire $LCode_{Auth}$ est principalement utilisé pour authentifier les paquets échangés entre LSN et entre LSN et HSN.

La méthode de calcul est simple et est basée sur la concaténation de $DATA_{dif}$ et le résultat XOR des cellules de ligne DMK correspondantes. La formule suivante explique comment le $LCode_{Auth}$ est calculé sur nœud i :

$$LCode_{Auth} = DATA_{dif} \,||\, (DMK_{i,2} \oplus DMK_{i,3} \oplus \ldots\ldots \oplus DMK_{i,33})$$

**Équation 9 : Calcul du $LCode_{Auth}$ pour le nœud i.**

#### *3.2.4.2 Le code d'authentification orthogonal ($OCode_{Auth}$)*

Le code d'authentification orthogonal est utilisé pour authentifier la communication entre les nœuds HSN en utilisant le même principe introduit par les clés ODK. L'$OCode_{Auth}$ est calculé par la formule suivante:

$$OCode_{Auth} = DATA_{dif} \,||\, (DMK_{0,j} \oplus DMK_{1,j} \oplus \ldots\ldots \oplus DMK_{Nbr\_HSN,j})$$

**Équation 10 : Calcul du $OCode_{Auth}$ pour le nœud i.**

Où Nbr_HSN représente le nombre de nœuds HSN dans le cluster.

#### *3.2.4.3 Le code d'authentification diagonal ($DCode_{Auth}$)*

La communication entre la station de base et les nœuds HSN est peut être ciblée par plusieurs attaques et nécessite un niveau de sécurité élevé. Nous proposons donc une nouvelle approche d'authentification efficace en termes de sécurité basée sur la méthode de calcul de clé DDK. Après avoir généré la matrice $DMK_{p,p}$ (expliquée dans la sous-section 4.2.2), nous calculons le $DCode_{Auth}$ en utilisant la formule suivante:

$$DCode_{Auth} = DATA_{dif} \,||\, (DMK_{0,2} \oplus DMK_{1,3} \oplus \ldots\ldots \oplus DMK_{p-2,p})$$

**Équation 11 : Calcul du $DCode_{Auth}$ pour le nœud i.**

#### *3.2.4.4 Algorithme d'allocation dynamique*

Pour optimiser le niveau de sécurité du processus d'authentification et empêcher les nœuds d'intrus de détecter et de capturer le $Code_{Auth}$ inséré dans le paquet de données, nous supposons que la position des huit bits alloués à $Code_{Auth}$ n'est pas fixe. Par conséquent, nous introduisons un

algorithme d'allocation dynamique qui change la position du $Code_{Auth}$ à chaque nouvelle donnée transmise sur la base du numéro de séquence du paquet. La formule suivante explique comment est calculée la position de $Code_{Auth}$:

$$Code_{Auth}\ Position = Packet_{Nsequence} \bmod (D_{Length})$$

**Équation 12 : Calcul de la position de $Code_{Auth}$.**

Où: $Packet_{Nsequence}$ est le numéro de séquence du nouveau paquet transmis, $D_{Length}$ est la longueur de la partie de données dans le paquet (29 octets).

# 4 Conclusion

Dans ce chapitre, un mécanisme de distribution dynamique de clés et d'authentification a été proposé pour les réseaux de capteurs hétérogènes. Le protocole proposé répond non seulement à l'exigence d'authentification et de gestion des clés pour un réseau de capteurs hétérogènes, mais optimise également la consommation mémoire, réduit la complexité de calcul et la surcharge de communication, ce qui améliore l'efficacité énergétique. L'algorithme de distribution de clés est basé sur des informations préexistantes pour générer des clés dynamiques et ne nécessite aucun canal sécurisé et de phases de partage de clés, pour confirmer l'efficacité de notre protocole nous devons analyser leur résistance face aux attaques, aussi une évaluation selon quelques métriques de performance doit être faite.

# CHAPITRE 6 :

# EVALUATION DES PERFORMANCES

# 1 Introduction

Dans ce chapitre, nous analysons notre mécanisme de sécurité par rapport à quelques attaques de sécurité les plus dévastatrices dont l'objectif est de nuire aux schémas de gestion des clés et d'authentification. Nous supposons qu'un nœud intrus peut essayer de rejoindre le réseau et de participer au processus de distribution des clés ou essayer de capturer la clé secrète des nœuds capteurs. Plusieurs types d'attaques peuvent être exécutées par le nœud intrus où les attaques principales sont: l'usurpation d'identité, la force brute, l'injection de nœud, l'attaque Sybil, les retransmissions de messages et la capture de nœuds. Ensuite nous présentons une étude des performances de notre protocole, dans laquelle ce dernier est évalué en fonction de consommation en mémoire, et surtout de communication et de temps de calcul. Nous préférons expérimenter le mécanisme EDAK proposé sur un environnement d'essai réel pour de meilleurs résultats d'évaluation. Les résultats obtenus sont comparés avec quelques protocoles proposés dans la littérature et présentés dans la section 6.1 du chapitre 4.

# 2 Analyse de sécurité

Dans cette section, nous analysons le comportement du protocole EDAK proposé contre les attaques malveillantes suivantes :

## 2.1 Attaque de spoofing

Le nœud attaquant peut écouter le trafic réseau pour capturer d'importantes informations de partage de clés et obtenir ensuite une clé secrète par paire. Dans le protocole EDAK, il n'est pas nécessaire d'échanger des informations pour générer des clés de paires partagées. Comme expliqué précédemment, le processus de génération de clé est basé sur des informations de nœud local qui le rendent robuste contre les attaques de spoofing.

## 2.2 Attaques par force brute

L'attaque par force brute est utilisée par le nœud intrus pour capturer les clés des nœuds. L'attaquant essaie, dans ce cas, de deviner la bonne clé secrète en testant de nombreuses clés potentielles. Dans le protocole EDAK, il est difficile, voire impossible, pour le nœud attaquant d'obtenir les clés du nœud car elles sont dynamiques et changent à chaque paquet transmis.

## 2.3 Injection de nœud

Un nœud malveillant peut être injecté dans le réseau pour provoquer par exemple un trou noir ou des attaques de trou de ver. Pour effectuer ce genre d'attaques; le nœud injecté doit d'abord rejoindre le réseau puis être authentifié en tant que nœud légitime. En plus de sécuriser l'algorithme de distribution des clés, le protocole EDAK offre un système d'authentification efficace qui peut authentifier la légitimité de l'identité des nœuds et détecter les nœuds injectés. En effet, pour être

authentifié, le nœud injecté doit d'abord reconstituer la matrice DMK puis obtenir la ligne ou colonne correspondante dans la matrice DMK et la dernière différence de données calculée qui n'est pas réalisable dans ce type d'attaque.

## 2.4 Attaque de Sybil

Plusieurs fausses identités sont utilisées dans cette attaque pour effectuer des opérations illicites. Comme l'attaque par injection de nœud, l'attaque Sybil doit d'abord contourner le mécanisme d'authentification proposé. De plus, le nœud attaquant doit reconstituer plusieurs matrices DMK et calculer différents codes d'authentification vu qu'il utilise de nombreuses identités.

## 2.5 Attaque de retransmission

Un ancien message peut être retransmis par un nœud intrus pour perturber le routage du réseau ou pour modifier les résultats d'agrégation des données. Cependant, puisque le code d'authentification utilisé pour valider la provenance du paquet est basé sur la dernière différence de données, le protocole EDAK peut détecter et rejeter tous les messages retransmis.

## 2.6 Capture de nœud et résilience

La capture de nœuds est l'attaque la plus percutante sur les schémas de gestion de clés et il est difficile de la contrer. En effet, le nœud attaquant peut capturer et compromettre un ou plusieurs nœuds capteurs et accéder ensuite à leurs clés stockées. Dans les réseaux RCSF hétérogènes, les nœuds HSNs sont censés être des composants inviolables car ils sont équipés de ressources puissantes [96], [116]. Par conséquent, la plupart des attaques de capture ciblent les nœuds LSN qui sont plus vulnérables. Le protocole EDAK offre une résilience tolérable contre l'attaque par capture de nœud sur les LSN. Etant donné que les nœuds LSN ne communiquent qu'avec un petit nombre de nœuds (nœuds voisins les plus proches dans la plage de communication) et génèrent ainsi un nombre limité de clés paires, la capture des nœuds LSN affecte seulement une petite partie du sous-cluster qui représente une petite partie du réseau. Cependant, dans l'environnement réel, le processus de capture de nœud peut provoquer l'isolation des nœuds ciblés pour une petite période de communication actuelle (écoute ou transmission), ce qui conduit à manquer certains paquets transmis entre les nœuds. Par conséquent, les nœuds capturés ne peuvent générer des clés correctes puisque leurs matrices DMK sont inexactes. Dans ce cas, le protocole EDAK offre une bonne résilience contre les attaques de capture de nœuds.

Dans ce qui suit, nous présentons une évaluation théorique de la résilience du schéma de gestion de clés proposé par rapport à l'attaque par compromission de nœud. Nous voulons découvrir l'effet des nœuds LSN compromis sur le reste du réseau. En effet, pour capturer avec succès la clé de cryptage utilisée par un nœud LSN donné, l'attaquant doit compromettre un nœud LSN appartenant au même SC et se trouver dans la zone de communication voisine du nœud LSN ciblé. Par exemple,

soit P (A) la probabilité qu'un nœud capturé donné appartienne au même SC du nœud LSN ciblé. La probabilité P (A) peut être calculée par la formule suivante:

$$P(A) = \frac{1}{\mathrm{N\_sc}}$$

**Équation 13 : La probabilité de capturer un nœud appartienne au SC.**

Noter que N_sc représente le nombre de SC dans le réseau. Soit P (B) la probabilité que le nœud compromis soit dans la zone de communication voisine du nœud LSN ciblé. P (B) est fourni par la formule suivante:

$$\mathrm{P(B)} = \frac{\mathrm{N\_neighboring}}{\mathrm{N\_sc\_node}}$$

**Équation 14 : La probabilité que le nœud compromis soit un voisin du nœud LSN ciblé.**

Où N_neighboring est le nombre de LSN voisins du nœud cible et N_sc_node représente le nombre de LSN dans le sous-cluster du nœud ciblé.

Ainsi, la probabilité qu'un nœud compromis puisse capturer la clé de cryptage d'un nœud LSN donné est égale à:

$$\mathrm{PA(B)} = \frac{(\mathrm{P(A \cap B)})}{(\mathrm{P(A)})}$$

**Équation 15 : La probabilité de capturer une clé par un nœud compromis.**

Comme nous pouvons le constater, plus le nombre de LSN voisins du nœud cible est petit, plus est réduite la probabilité de capturer un nœud ciblé. De plus, à mesure que le nombre de SC augmente dans le réseau, le nœud attaquant aura plus de difficultés à capturer les nœuds LSN.

# 3 Évaluation des performances

Dans cette section, nous évaluons les performances du mécanisme EDAK proposé en fonction de trois métriques importantes: l'exigence d'espace mémoire, la complexité de calcul et le

surcoût de communication. Bien que l'efficacité énergétique ne soit pas explicitement évaluée, sa consommation est reflétée par la surcharge de calcul et de communication.

## 3.1 Environnement d'expérimentation

Dans notre étude de performance, nous préférons expérimenter le mécanisme EDAK proposé sur un environnement d'essai réel pour de meilleurs résultats d'évaluation. Nous implémentons donc notre proposition sur deux types de plates-formes de capteurs: la première est dédiée aux nœuds LSN et est équipée d'un CPU de faible calcul (ATmega328P 16MHz), d'un espace mémoire limité (2KB de SRAM et 32Ko de mémoire flash) alimentation (2 Joules). La seconde plate-forme de test est utilisée pour implémenter les nœuds hétérogènes (HSN) et comprend un puissant processeur de calcul (Atmel SAM3X8E ARM Cortex-M3 84 MHz), un grand espace mémoire (96 Ko de SRAM et 512 Ko de mémoire flash) et un haut niveau d'approvisionnement en énergie (200 Joules).

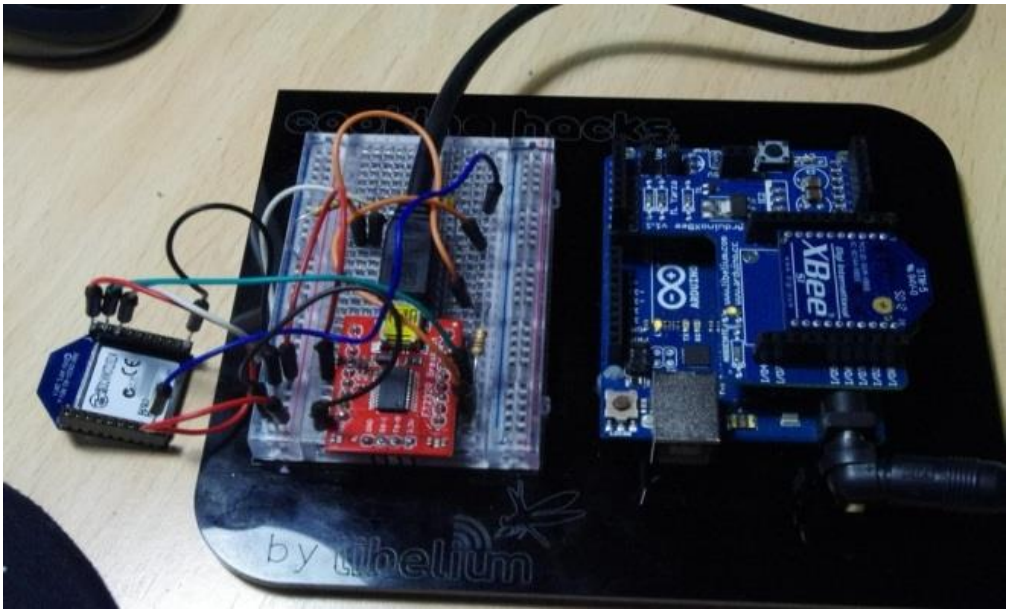


**Figure 13 : Plateforme d'expérimentation.**

## 3.2 Evaluation des Résultats

Trois métriques d'évaluation ont été utilisées pour analyser les performances du protocole EDAK. En effet, le stockage mémoire, la complexité de calcul et le surcoût de communication sont les principales mesures utilisées pour évaluer efficacement les schémas de sécurité dans les RCSF. Par conséquent, nous avons évalué notre protocole proposé en fonction de ces métriques centrales et comparé les résultats obtenus avec les travaux connexes.

### 3.2.1 Stockage de la mémoire

La première étape d'évaluation de l'espace mémoire consiste à mesurer la consommation initiale en mémoire du protocole EDAK. Le tableau suivant présente les exigences de RAM et de mémoire Flash sur les nœuds LSN et HSN :

| SRAM | Flash |
|---|---|
| 0,178 KB | 5,439 KB |

**Tableau 2 : Conditions initiales de stockage de la mémoire**

Sur la base des résultats obtenus, nous démontrons que le protocole EDAK est efficace en mémoire et nécessite un espace mémoire initial raisonnable. En effet, dans les nœuds LSN, l'algorithme EDAK consomme environ 17% (5 439 Ko sur 32 Ko de mémoire disponible) de mémoire flash pour le code d'implémentation et 8,9% (183 octets sur 2 Ko) de mémoire RAM pour les données initiales (mémoire avec données initialisées, par exemple clé IK et mémoire initialisée à zéro par exemple matrice DMK). De l'autre côté, dans les nœuds HSN, la consommation mémoire est modeste car les HSN sont équipés de mémoire de stockage importante (96 Ko de SRAM et 512 Ko de mémoire flash) et équivalent à 0,185% de SRAM et 1,062% de mémoire flash.

L'étape suivante de l'évaluation de la consommation de mémoire consiste à analyser la demande d'espace de stockage nécessaire pour mémoriser les clés générées par notre protocole proposé par rapport aux protocoles de distribution de clés connus présentés précédemment dans les travaux connexes. Par conséquent, nous avons mesuré la consommation de la mémoire de stockage des clés (mémoire SRAM) avant et après le déploiement du réseau pour analyser les coûts de stockage des clés pré chargées et générées respectivement.

La Figure 14 et la Figure 15 présentent la taille des clés pré chargées avant le déploiement du réseau dans les nœuds HSN et LSN respectivement.

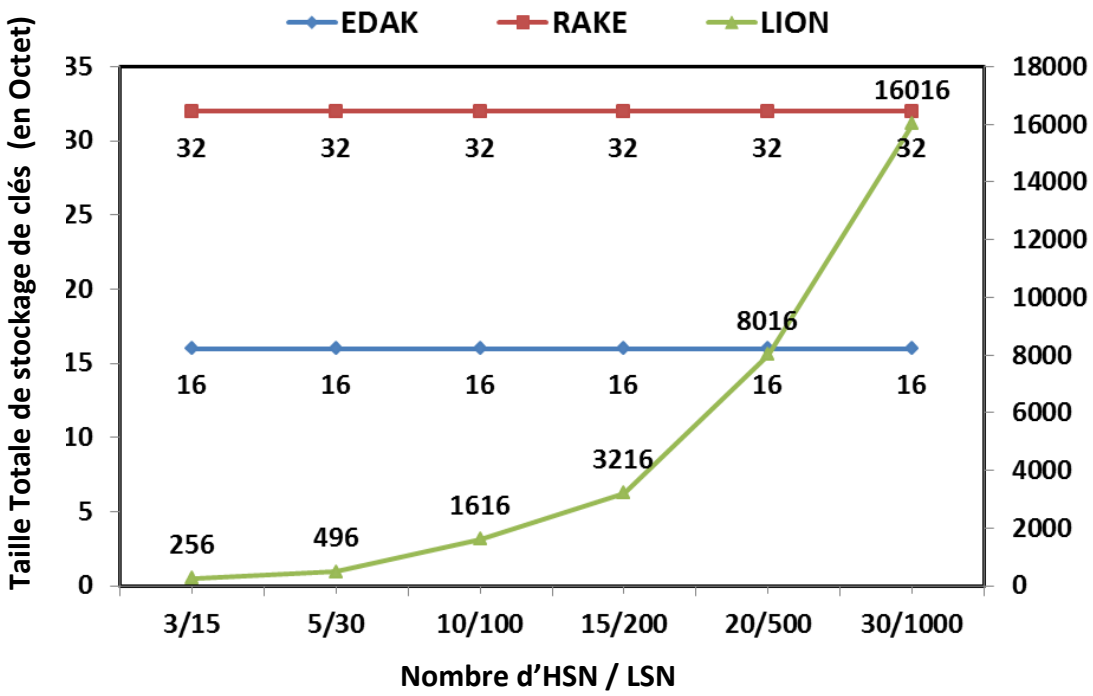


**Figure 14 : Espace de stockage total des clés dans les nœuds HSN avant le déploiement.**

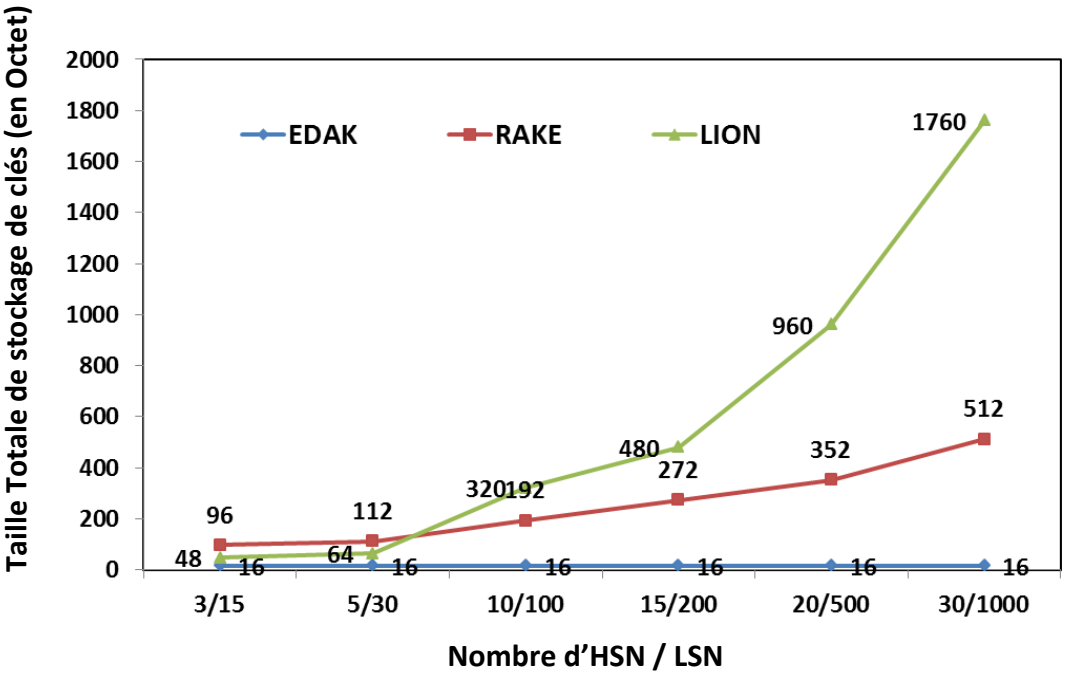


**Figure 15 : Espace de stockage total des clés dans les nœuds LSN avant le déploiement.**

Comparé aux protocoles RAKE et LION, EDAK est efficace en termes de consommation mémoire et nécessite moins d'espace mémoire pour le stockage initial des clés. En effet, seuls 16 octets (128 bits) sont nécessaires pour stocker une clé maîtresse (IK), sur les nœuds LSN et HSN, et ceci indépendamment de la taille du réseau. De l'autre côté, dans le protocole RAKE, deux clés principales (32 octets) doivent être stockées dans les nœuds HSN et LSN. En outre, pour chaque nœud HSN, les nœuds LSN doivent stocker une clé unique de paire qui augmente la surcharge mémoire sur les nœuds LSN. LION est le protocole le plus coûteux en mémoire et implique le stockage de n + 1 (n est le nombre de nœuds LSN dans le réseau) clés de paire sur les nœuds HSN. En outre, le nombre de clés stockées dans les nœuds LSN augmente considérablement avec la taille du réseau.

Après le déploiement du réseau, les besoins de stockage en mémoire augmenteront, étant donné que des clés de paires symétriques seront générées pour sécuriser la communication entre les nœuds de réseau. Dans le protocole EDAK, de nouvelles clés de cryptage / décryptage temporaires sont générées dynamiquement à chaque paquet transmis / reçu. Par conséquent, il n'est pas nécessaire de stocker des clés supplémentaires. Cependant, l'augmentation du stockage mémoire est due à la mise à jour de la matrice DMK avec des informations de clé partielle PK (comme expliqué dans la section 3.1).

Dans notre expérimentation, nous supposons que chaque nœud LSN maintient au maximum une matrice DMK [5] [32] qui représente les informations PK de ses cinq nœuds voisins les plus proches. Comme les LSN ont une faible portée de communication, nous supposons qu'ils utilisent une communication à un seul saut avec au plus cinq nœuds voisins (qui représentent une moyenne de nœuds voisins dans le cas d'un réseau à haute densité) et une communication multi-sauts pour atteindre les autres nœuds correspondants.

Contrairement aux nœuds LSN, les nœuds HSN doivent maintenir une matrice DMK plus grande puisqu'ils ont une plus grande portée de communication. En effet, chaque SCH maintient au maximum la matrice DMK [k] [32] où k = q + h + 1 (q est le nombre de nœuds LSN dans le sous-cluster, h est le nombre de SCH dans le cluster et 1désignela ligne réservée pour le CH dans la DMK). D'autre part, les têtes de grappe conservent au maximum la matrice DMK [p] [32] où p = r + h + s (r est le nombre de nœuds LSN dans le cluster et s le nombre de têtes de cluster dans le réseau). Figure 16 présente l'espace de stockage total du DMK dans les nœuds HSN et LSN après le déploiement.

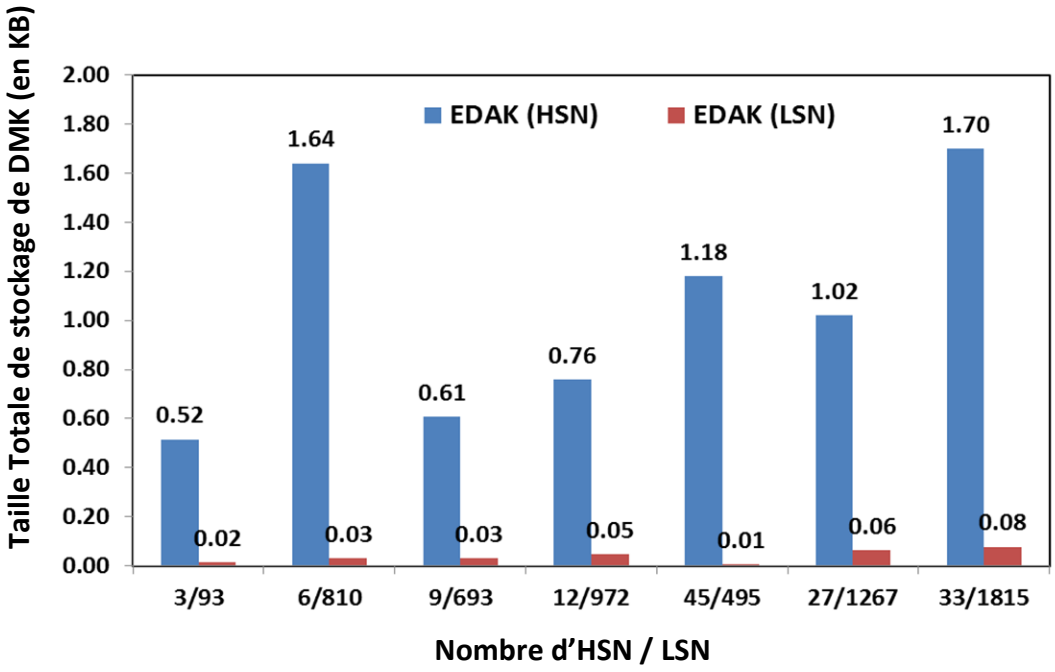


**Figure 16 : Espace de stockage total des clés dans les nœuds HSN et LSN après le déploiement.**

Comme le montre la Figure 16, il est clairement démontré que l'espace mémoire requis pour la matrice DMK respecte la limitation des nœuds LSN restreints aux ressources et il est largement tolérable dans les nœuds HSN. En effet, sur la base des différentes tailles de réseau utilisées dans notre expérimentation, la mémoire moyenne requise est inférieure à 0,1 Ko, ce qui représente 5% de la mémoire SRAM totale (étant donné que les nœuds LSN disposent d'une mémoire SRAM de 2 Ko). Dans les nœuds HSN, la mémoire réservée pour la DMK prend moins de place (environ 1,8%) car elle dispose d'un espace mémoire plus important (96 Ko de SRAM). En outre, la consommation de mémoire dans les nœuds LSN diminuera de manière significative à mesure que le nombre de nœuds HSN augmente dans le réseau. Ceci est justifié par l'augmentation du nombre de sous-clusters qui réduit le nombre des LSN les voisins et ainsi l'espace de stockage de la DMK.

Pour comparer les performances mémoire du protocole EDAK avec les protocoles RAKE et LION, nous avons mesuré l'espace mémoire total requis après le déploiement du réseau. La Figure 17 présente les résultats obtenus.

Dans les quatre premières tailles de réseau expérimentées, où il y a un petit nombre de nœuds HSN, nous pouvons observer que la consommation totale de mémoire est inférieure au protocole LION et se rapproche du protocole RAKE. Cependant, la consommation mémoire s'améliore à mesure que le nombre de nœuds HSN augmente dans le réseau et le protocole EDAK demande beaucoup moins d'espace mémoire comparé aux protocoles LION et RAKE.

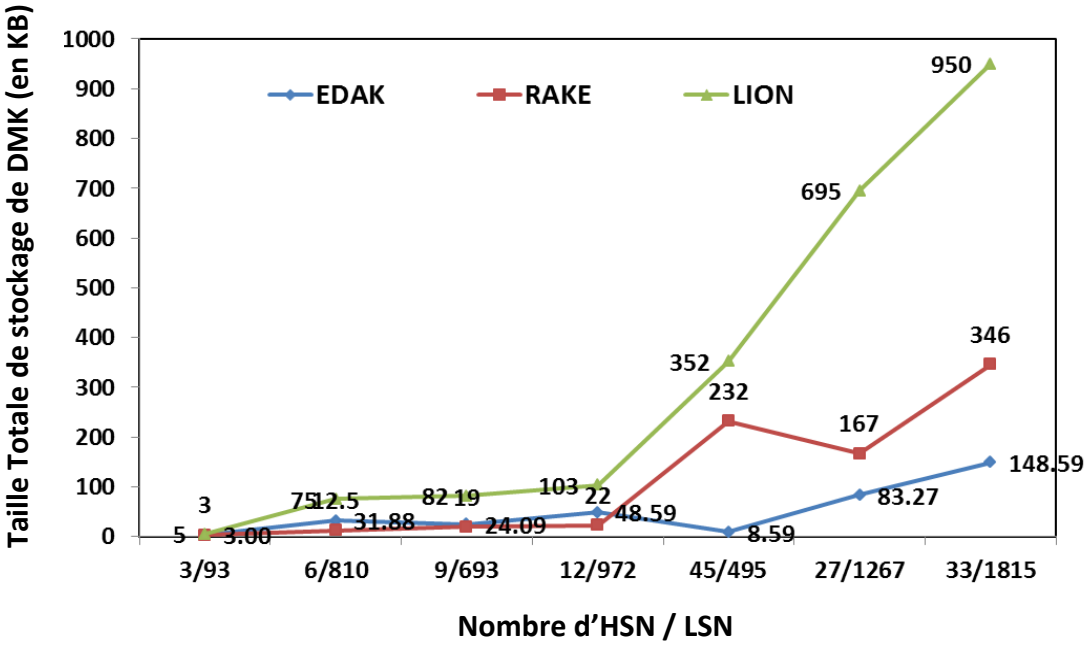


**Figure 17 : Total de la consommation mémoire après le déploiement.**

### 3.2.2 Complexité de calcul

La surcharge de calcul est une métrique importante pour évaluer la complexité des protocoles de distribution de clé dédiés pour les RCSF. Dans notre protocole proposé, la fonction XOR, les fonctions de différence et de concaténation sont utilisées pour générer le code d'authentification et les clés de paires symétriques ce qui souligne la simplicité d'EDAK. L'algorithme AES est également appliqué pour crypter et décrypter les paquets échangés entre les nœuds capteurs. Dans une situation particulière, les nœuds HSN peuvent exécuter la fonction de hachage MD5 pour adapter la longueur de clé de paire (128 bits). Le temps CPU requis du protocole EDAK a été mesuré et détaillé dans le Tableau 3.

| LSN | | HSN | |
|---|---|---|---|
| **Algorithme** | **Temps CPU ≈** | **Algorithme** | **Temps CPU ≈** |
| **Code Auth** | 0,032 ms | **Code Auth** | 0.005 ms |
| **Génération de clé** | 0,042 ms | **Génération de clé** | 0.009 ms |
| **AES Enc/Dec** | 0,54/0,704 ms | **AES Enc/Dec** | 0.096/0.163 ms |
| **hash clé** | -- | **hash clé** | 0.048 ms |

**Tableau 3: Temps de calcul du processeur**

Comme présenté dans le tableau 3, l'EDAK est un protocole de sécurité léger en termes de calcul, et introduit une surcharge tolérable. En effet, les algorithmes d'authentification, de génération de clé et de distribution s'exécutent pendant environ 0,77ms et 0,22ms sur les LSN et les HSN respectivement.

La complexité de calcul d'EDAK a été également évaluée et comparée aux protocoles RAKE, LION, PKA et WLKH en se basant sur le nombre d'opérations de cryptage et de décryptage symétrique, la fonction de hachage appliquée et les calculs asymétriques (ECC, ...). Le tableau 4 présente les résultats obtenus.

| Protocole | Symmetrique Encr/Decr | Hash | Asymmetrique Encr/Decr |
|---|---|---|---|
| **EDAK** | 2 | [0,2] | 0 |
| **RAKE** | 6 | [1,4] | 0 |
| **LION** | [2,4h] | 0 | 0 |
| **PKA** | [0,2] | 4 | 2 (ECC/ECDH) |
| **WLKH** | 2 | [1,t+1] | [0,2] |

**Tableau 4 : Comparaison de la complexité de calcul.**

En effet, le protocole EDAK est basé sur le cryptosystème symétrique léger (AES) et sur la fonction de hachage unidirectionnelle (MD5). Dans le tableau, [x, y] représentent les valeurs minimales et maximales des opérations. Comme illustré, EDAK exécute deux opérations symétriques (cryptage et décryptage) et au maximum deux fonctions de hachage. Par rapport à EDAK, les protocoles RAKE et LION sont également basés sur un système de cryptographie symétrique (AES); De plus, RAKE utilise la fonction de hachage SHA1. Cependant, le protocole RAKE applique trois fois plus d'opérations de cryptage et de décryptage symétriques et exécute au moins une fonction de hachage. Le protocole LION peut introduire un temps de calcul similaire à EDAK, bien que la complexité de calcul augmente avec l'augmentation du nombre de sauts (4h) entre les nœuds qui ne partagent pas les clés de paires [105]. Les protocoles PKA et WLKH exécutent des calculs asymétriques en plus des opérations de chiffrement / déchiffrement symétriques, ce qui signifie plus de temps de calcul. En outre, une utilisation extensive des fonctions de hachage est appliquée dans WLKH pour générer des clés d'arbre de nœuds où le nombre d'opérations de hachage exécutées dépend de la quantité des nœuds d'arbre correspondants (t) [99].

### 3.2.3 Le surcoût de communication

Le surcoût de communication se produit principalement pendant la phase d'initialisation et de génération de clé.

Tableau 5 présente la quantité de données supplémentaires transmises (en octets) dans le protocole EDAK et d'autres travaux connexes.

| **Protocole** | **Transmission de données supplémentaire (Octets)** |
|---|---|
| **EDAK** | 1 |
| **RAKE** | 56 |
| **LION** | 2k+28 |
| **PKA** | 100 |
| **WLKH** | $(1+\log_2 n) \times 32$ |

**Tableau 5 : Transmission de données supplémentaires.**

Les résultats obtenus montrent clairement que le protocole EDAK introduit un faible surcoût de communication car un seul octet est intégré dans le paquet de données à des fins d'authentification. En effet, le processus de génération de clé est basé sur des informations locales existantes (matrice DMK) et n'a pas besoin d'échanger d'autres données et clés, ce qui réduit considérablement le temps de communication et optimise le niveau de sécurité. En revanche, le protocole RAKE doit envoyer deux messages supplémentaires (56 octets) pour enregistrer et partager une nouvelle paire de clés entre les nœuds (dans le cas d'un nouveau lien de communication). Les protocoles PKA et LION nécessitent plus d'échange de données supplémentaires qu'EDAK et RAKE, en particulier dans les réseaux de grande taille. En outre, dans le protocole LION, le débit de communication dépend directement du nombre de clés pré chargées (k) dans les LSN. En WLKH, les clés de nœuds sont calculées par une clé privée partagée par les membres du cluster ce qui implique l'envoi de $n\log_2 n$ messages pour partager une clé de paire dans le cluster (où n est la taille du cluster). En outre, un message d'enregistrement est transmis pendant la phase d'initialisation du protocole pour rejoindre le cluster.

# 4 Conclusion

L'analyse d'EDAK part apport aux attaques : l'usurpation d'identité, la force brute, l'injection de nœud, l'attaque de Sybil, les retransmissions de messages et la capture de nœuds, prouve sa résistance contre ces derniers ce qui permet d'amélioré significativement le niveau de sécurité de notre protocole. L'évaluation des performances de notre mécanisme EDAK en fonction de trois métriques importantes (l'exigence d'espace mémoire, la complexité de calcul et le surcoût de communication) nous permettons de démontrer que :

- L'espace mémoire requis pour la matrice DMK respecte la limitation des nœuds LSN restreints aux ressources et il est largement tolérable dans les nœuds HSN.

- Le protocole EDAK introduit un faible surcoût de communication. En effet, le processus de génération de clé est basé sur des informations locales existantes (matrice DMK) et n'a pas besoin d'échanger d'autres données et clés, ce qui réduit considérablement le temps de communication et optimise le niveau de sécurité.

- EDAK est un protocole de sécurité léger en termes de calcul, et introduit une surcharge tolérable. la fonction XOR, les fonctions de différence et de concaténation sont utilisées pour générer le code d'authentification et les clés de paires symétriques ce qui souligne la simplicité d'EDAK.

Les résultats expérimentaux obtenus en termes d'amélioration des performances confirment l'efficacité de notre protocole parapport aux travaux proposés dans la littérature. Le mécanisme de sécurité proposé est également plus flexible et évolutif pour son application à grandes échelles.

# CONCLUSION GENERALE

- **Rappel de la problématique**

Grâce à leurs divers avantages, les RCSF ont connu un succès sans cesse croissant au sein des communautés scientifiques et industrielles. Ce succès émergent des réseaux de capteurs pourrait être entravé par leurs problèmes de sécurité inhérents. Afin de poursuivre cette réussite, plusieurs travaux de recherche sont consacrés pour fournir des mécanismes de sécurité efficaces et légers.

La plupart des protocoles de sécurité sont construits autour d'algorithmes de cryptage et d'authentification puissants. Pour atteindre les objectifs de sécurité, la gestion des clés est la première fonction fondamentale puisque les nœuds capteurs ont besoin d'une clé commune valide pour exploiter les mécanismes cryptographiques. Le problème de distribution des clés a été largement abordé dans les RCSF homogènes et divers mécanismes ont été proposés. Malgré la variété des solutions efficaces proposées dans ces catégories, l'équilibre entre le niveau de sécurité et la consommation de ressources reste le problème majeur dans les RCSF homogènes. Les réseaux de capteurs sans fil hétérogènes (HWSN) ont ouvert une nouvelle direction de recherche pour le problème de sécurité et ont offert plusieurs opportunités.

En effet, la gestion des clés reste inexplorée dans les RCSF hétérogènes et seules quelques recherches ont abordé le problème. La plupart de ces recherches sont basées sur des schémas symétriques de pré-distribution qui souffrent de problèmes tels que la distribution des clés probabilistes entre les HSN et les LSN, la non-extensibilité après le déploiement, le manque de stockage mémoire sur les LSN limités en ressources et les frais généraux de communication.

- **Contributions**

Un protocole appelé EDAK a été proposé dans cette thèse, dont l'objectif est d'établir un schéma efficace d'authentification et de gestion des clés dynamiques pour les réseaux de capteurs sans fil hétérogènes. L'objectif principal de notre protocole est de résoudre les principaux problèmes de sécurité introduits par les schémas de distribution de clés et d'authentification afin d'optimiser le niveau de sécurité. En outre, les processus d'authentification et de gestion des clés dynamiques exploitent les avantages des RCSF hétérogènes, ce qui nous a permis de réduire considérablement les surcoûts de communication et de préserver la consommation d'énergie. Pour récapituler, les contributions apportées dans la thèse sont les suivantes:

Notre première contribution consiste en la proposition d'un schéma de gestion de clés dynamique. L'idée principale est de mettre en place un mécanisme léger d'établissement et de distribution des clés tout en optimisant le niveau de sécurité. Nous proposons un algorithme

d'établissement de clés efficace pour créer des clés par paires entre des LSNs, des clés de groupe entre des SCHs dans le CL et une clé de groupe pour CHs et la station de base (BS). Pour optimiser le niveau de sécurité et empêcher la capture des clés, un processus de génération de clés dynamique est introduit, et qui crée une nouvelle clé pour chaque message transmis sans nécessiter d'échange d'informations supplémentaires. Pour générer les clés de chiffrement, l'algorithme EDAK est basé sur une matrice de clés dynamique DMK qui est générée en utilisant des informations préexistantes. Chaque nœud capteur dans le réseau crée et maintient sa propre matrice DMK pour générer des clés de cryptage / décryptage afin de crypter ses données transmises et décrypter les données reçues de ses voisins.

La deuxième contribution dans notre travail est d'établir un mécanisme d'authentification des nœuds capteurs légitimes dans le réseau. Par conséquent, le protocole EDAK introduit un nouvel algorithme d'authentification léger pour vérifier l'identité des nœuds capteurs tout en se basant sur la matrice des clés dynamique DMK. En effet, à chaque nouveau paquet transmis, le nœud capteur génère un code d'authentification $Code_{Auth}$ qui sera ajouté au paquet. Le $Code_{Auth}$ est composé de deux parties: $DATA_{dif}$ et $XORD_{dif}$. Le $DATA_{dif}$ représente le résultat de la soustraction entre les dernières données reçues et la nouvelle. Le $XORD_{dif}$ représente le résultat XOR des cellules DMK correspondantes. Après avoir reçu le paquet, le nœud récepteur peut authentifier l'émetteur de paquets en vérifiant le code d'authentification reçu en le comparant au $Code_{Auth}$ calculé localement basé sur la matrice DMK.

L'algorithme d'authentification proposé est simple et ne nécessite pas d'échange de paquets de contrôle supplémentaires. En outre, il est plus approprié pour les ressources des nœuds capteurs limités car il n'implique pas de surcharge de calcul élevée. Pour optimiser le niveau de sécurité de l'algorithme d'authentification proposé, nous introduisons trois catégories $Code_{Auth}$ basées sur le type de nœuds émetteur et récepteur: $LCode_{Auth}$, $OCode_{Auth}$ et $DCode_{Auth}$.

Nous avons analysé le comportement du protocole EDAK proposé contre quelques attaques malveillantes. Nous supposons qu'un nœud d'intrus peut essayer de rejoindre le réseau et de participer au processus de distribution de clés ou d'attraper la clé secrète des nœuds capteurs. Plusieurs types d'attaques peuvent être exécutées par le nœud intrus où les attaques principales sont: l'usurpation d'identité, la force brute, l'injection de nœud, l'attaque Sybil, les repliais de messages et la capture de nœuds.

Nous avons évalué les performances du mécanisme EDAK proposé en fonction de trois métriques importantes: l'exigence d'espace mémoire, la complexité de calcul et le surcoût de communication. Bien que l'efficacité énergétique ne soit pas explicitement évaluée, sa consommation est reflétée par le calcul et la communication. Pour de meilleurs résultats d'évaluation, les performances de notre schéma proposé sont expérimentées sur un environnement de test réel. Puisque notre proposition est destinée aux RCSF hétérogènes, nous avons implémenté notre mécanisme sur deux types de plates-formes de capteurs: la première est dédiée aux nœuds

LSN et équipée d'un CPU de faible calcul (ATmega328P 16MHz), d'un espace mémoire limité (2KB de SRAM et 32Ko de mémoire Flash) et une batterie de 2 Joules. La seconde plate-forme de test est utilisée pour implémenter les nœuds hétérogènes (HSN) et comprend un puissant processeur de calcul (Atmel SAM3X8E ARM Cortex-M3 84 MHz), un grand espace mémoire (96 Ko de SRAM et 512 Ko de mémoire flash) et une haute capacité en énergie (200 Joules).

- **Perspectives**

Les bons résultats d'expérimentation ne signifient pas que nos deux protocoles sont optimaux. En effet, il existe plusieurs points à améliorer en perspective. Un algorithme de tri global doit être mis en place pour avoir un ordre global pour les lignes de toutes les matrices DMK. Notre solution est plus efficace lorsqu'un nombre important de nœuds HSN est déployé dans le réseau, nous devrons améliorer notre proposition pour qu'elle soit efficace pour toute configuration possible du réseau. Nous essayerons de concevoir un protocole de sécurité complet qui intègre à la fois les politiques de sécurité et les systèmes de détection d'intrusions.

Nous espérons implémenter notre protocole sur des capteurs *(tels que : Tmote Sky, MICA, Imote ou BTnode)*, ce qui permettra d'évaluer les performances de nos contributions sur d'autres plateformes du marché.

# BIBLIOGRAPHIE


[1] A. S. Uluagac, C. P. Lee, R. A. Beyah, et J. A. Copeland, « Designing Secure Protocols for Wireless Sensor Networks », in *Wireless Algorithms, Systems, and Applications*, vol. 5258, Y. Li, D. T. Huynh, S. K. Das, et D.-Z. Du, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, p. 503-514.

[2] D. G. Padmavathi et M. D. Shanmugapriya, « A Survey of Attacks, Security Mechanisms and Challenges in Wireless Sensor Networks », vol. 4, n° 1, p. 9, 2009.

[3] Y. FAYE, « Algorithmes d'authentification et de cryptographie efficaces pour les réseaux de capteurs sans fil. » Thèse de doctorat. Université de Franche-Comté, 2014.

[4] J. Sen, « A survey on wireless sensor network security », *Int. J. Commun. Netw. Inf. Secur.*, vol. 1, n° 2, p. 55-78, août 2009.

[5] J. P. Walters, Z. Liang, W. Shi, et V. Chaudhary, « Wireless Sensor Network Security: A Survey », in *Security in Distributed, Grid, and Pervasive Computing*, 2006 Auerbach Publications, CRC Press, p. 50.

[6] S. Athmani, « Protocole de sécurité Pour les Réseaux de capteurs Sans Fil. » Mémoire de Magistère. Université de Batna 2., 2010.

[7] D. M. K. Jain, « Wireless sensor networks: Security issues and challenges », *Int. J. Comput. Inf. Technol.*, vol. 2, n° 1, p. 62-67, 2011.

[8] J. Ibriq, I. Mahgoub, et M. Ilyas, « Secure Routing in Wireless Sensor Networks », in *Handbook of Information and Communication Security*, P. Stavroulakis et M. Stamp, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, p. 553-578.

[9] H. Kupwade Patil et T. M. Chen, « Wireless sensor network security », in *Computer and Information Security Handbook*, Second Edition., 2013, p. 301-322.

[10] C. Claude et F. Aurélien, « Protéger les réseaux de capteurs sans fil », présenté à SSTIC08, 2008, p. 1-11.

[11] M. Matin, Éd., *Wireless sensor networks-technology and protocols*, Janeza Trdine 9, 51000 Rijeka, Croatia. Croatia: InTech, 2012.

[12] C. YADAV, K. RAKSHA, et S. HEGDE, « Security Techniques in Wireless Sensor Networks: A Survey. », *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 4, n° 4, p. 289-295., 2015.

[13] S. Rupinder, S. Jatinder, et S. Ravinder, « Security challenges in wireless sensor networks », *Int. J. Comput. Sci. Inf. Technol. Secur. IJCSITS*, vol. Vol.6, n° No3, p. 1-6, 2016.

[14] P. MOHANTY, S. PANIGRAHI, N. SARMA, et S. S. SATAPATHY, « SECURITY ISSUES IN WIRELESS SENSOR NETWORK DATA GATHERING PROTOCOLS: A SURVEY », p. 14, 2005.

[15] A. G. A. Alquraishee et J. Kar, « A survey on security mechanisms and attacks in wireless sensor networks », *Contemp. Eng. Sci.*, vol. 7, p. 135-147, 2014.

[16] Y.-X. Li, L. Qin, et Q. Liang, « Research on Wireless Sensor Network Security », présenté à Computational Intelligence and Security (CIS), 2010 International Conference on. IEEE, 2010, p. 493-496.

[17] W. Wang, S. Zhang, G. Duan, et H. Song, « Security in Wireless Sensor Networks », in *Wireless Network Security*, Higher Education Press, Beijing and Springer-Verlag Berlin Heidelberg, 2013, p. 129-177.

[18] E. Çayirci et C. Rong, *Security in Wireless Ad Hoc and Sensor Networks*. Chichester, UK: John Wiley & Sons, Ltd, 2009.

[19] A. R. Dhakne et P. N. Chatur, « Detailed Survey on Attacks in Wireless Sensor Network », in *Proceedings of the International Conference on Data Engineering and Communication Technology*, vol. 469, S. C. Satapathy, V. Bhateja, et A. Joshi, Éd. Singapore: Springer Singapore, 2017, p. 319-331.

[20] S.-E. BENBRAHIM, « Défense contre l'attaque d'analyse de trafic dans les réseaux de capteurs sans fil (WSN) », Thèse de doctorat. École Polytechnique de Montréal., 2011.

[21] D. E. BOUBICHE, « Une approche Inter-Couches (cross-layer) pour la Sécurité dans les RCSF. », Thèse de doctorat. Université de Batna 2., 2013.

[22] S. C.-H. Huang, D. MacCallum, et D. Du, Éd., *Network security*. New York: Scott C.-H. Huang. Springer, 2007.

[23] Y. Wang, G. Attebury, et B. Ramamurthy, « A survey of security issues in wireless sensor networks », *IEEE Commun. Surv. Tutor.*, vol. 8, n° 2, p. 2-23, 2006.

[24] V. Rathod et M. Mehta, « Security in Wireless Sensor Network: A survey », *Ganpat Univ. J. Eng. Technol.*, vol. 1, n° 1, p. 35-44, 2011.

[25] S. Athmani, D. E. Boubiche, et A. Bilami, « Hierarchical energy efficient intrusion detection system for black hole attacks in WSNs », présenté à Computer and Information Technology (WCCIT), 2013 World Congress on. IEEE, 2013, p. 1-5.

[26] D. R. Raymond et S. F. Midkiff, « Denial-of-Service in Wireless Sensor Networks: Attacks and Defenses », *IEEE Pervasive Comput.*, vol. 7, n° 1, p. 74-81, janv. 2008.

[27] Z. Li et G. Gong, « Survey on Security in Wireless Sensor Networks », *J. Korea Inst. Inf. Secur. Cryptol.*, vol. 18, n° 6B, p. 233-248, 2008.

[28] P. Khatawkar, K. Gaikwad, V. Solanke, G. Kulkarni, R. Shelk, et S. Gujar, « Wireless sensor network security threats », 2013, p. 131-135.

[29] A. Dubey, V. Jain, et A. Kumar, « A Survey in Energy Drain Attacks and Their Countermeasures in Wireless Sensor Networks », *Int. J. Eng. Res.*, vol. 3, n° 2, p. 6, 2014.

[30] S. Ozdemir et Y. Xiao, « Secure data aggregation in wireless sensor networks: A comprehensive overview », *Comput. Netw.*, vol. 53, n° 12, p. 2022-2037, août 2009.

[31] D. E. Boubiche, S. Boubiche, H. Toral-Cruz, A.-S. K. Pathan, A. Bilami, et S. Athmani, « SDAW: secure data aggregation watermarking-based scheme in homogeneous WSNs », *Telecommun. Syst.*, vol. 62, n° 2, p. 277-288, juin 2016.

[32] A. Boukerche, Éd., *Algorithms and protocols for wireless sensor networks*. Hoboken, N.J: Wiley, 2009.

[33] G. Sharma, S. Bala, et A. K. Verma, « Security Frameworks for Wireless Sensor Networks-Review », *Procedia Technol.*, vol. 6, p. 978-987, 2012.

[34] K.-A. Shim, « A Survey of Public-Key Cryptographic Primitives in Wireless Sensor Networks », *IEEE Commun. Surv. Tutor.*, vol. 18, n° 1, p. 577-601, 2016.

[35] M. Panda, « Security in Wireless Sensor Networks using Cryptographic Techniques », *Am. J. Eng. Res.*, p. 7, 2014.

[36] M. M. Lamine, « Sécurité dans les Réseaux de Capteurs Sans-Fil », Memoire de Magistere en Informatique, Ecole Doctorale d'Informatique de bejaia, 2008.

[37] « Les algorithmes de chiffrement ». [En ligne]. Disponible sur: http://dspace.univ-tlemcen.dz/bitstream/112/1046/9/Chapitre3.pdf. [Consulté le: 27-mai-2018].

[38] S. JULIA, « techniques de cryptographie », 2004-2003. [En ligne]. Disponible sur: http://deptinfo.unice.fr/twiki/pub/Linfo/PlanningDesSoutenances20032004/blanc-degeorges.pdf. [Consulté le: 27-mai-2018].

[39] R. B. Philippe, « Principaux algorithmes de cryptage ». Department of Computer Science SEPRO Robotique France, 11-juill-2002.

[40] D. Boyle et T. Newe, « Securing Wireless Sensor Networks: Security Architectures », *J. Netw.*, vol. 3, n° 1, p. 13, 2008.

[41] N. Gura, A. Patel, A. Wander, H. Eberle, et S. C. Shantz, « Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs », in *Cryptographic Hardware and Embedded Systems - CHES 2004*, vol. 3156, M. Joye et J.-J. Quisquater, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, p. 119-132.

[42] R. L. RIVEST, A. SHAMIR, et L. ADLEMAN, « A method for obtaining digital signatures and public-key cryptosystems », *Commun. ACM*, vol. 21, n° 2, p. 120-126, 1978.

[43] D. Westhoff, B. Lamparter, C. Paar, et A. Weimerskirch, « On digital signatures in ad hoc networks », *Eur. Trans. Telecommun.*, vol. 16, n° 5, p. 411-425, sept. 2005.

[44] S. M. G, R. J. D'Souza, et G. Varaprasad, « Digital Signature-Based Secure Node Disjoint Multipath Routing Protocol for Wireless Sensor Networks », *IEEE Sens. J.*, vol. 12, n° 10, p. 2941-2949, oct. 2012.

[45] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, et P. Kruus, « TinyPK: securing sensor networks with public key technology », présenté à Proceedings of the 2nd ACM workshop on Security of ad hoc and sensor networks. ACM, 2004, p. 59.

[46] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, et K. Pister, « System Architecture Directions for Networked Sensors », *ACM SIGOPS Oper. Syst. Rev.*, vol. 34, n° 5, p. 93-104, 2000.

[47] A. Liu et P. Ning, « TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks », présenté à Proceedings of the 7th international conference on Information processing in sensor networks. IEEE Computer Society, 2008, p. 245-256.

[48] M. H. Ahmed, S. W. Alam, N. Qureshi, et I. Baig, « Security for WSN based on elliptic curve cryptography », présenté à Computer Networks and Information Technology (ICCNIT), 2011 International Conference on. IEEE, 2011, p. 75-79.

[49] P. G. Shah, X. Huang, et D. Sharma, « Analytical Study of Implementation Issues of Elliptical Curve Cryptography for Wireless Sensor networks », présenté à Advanced Information Networking and Applications Workshops (WAINA), 2010 IEEE 24th International Conference on. IEEE, 2010, p. 589-592.

[50] Qing Chang, Y. Zhang, et Lin-lin Qin, « A node authentication protocol based on ECC in WSN », présenté à Computer Design and Applications (ICCDA), 2010 International Conference on. IEEE, 2010, p. V2-606-V2-609.

[51] G. D. Murphy, E. M. Popovici, et W. P. Marnane, « Area-Efficient Processor for Public-Key Cryptography in Wireless Sensor Networks », présenté à Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on. IEEE, 2008, p. 667-672.

[52] I. Mansour, « Contribution à la sécurité des communications des réseaux de capteurs sans fil », Thèse de doctorat, Université Blaise Pascal-Clermont-Ferrand II, 2013.

[53] C.-H. Ling, C.-C. Lee, C.-C. Yang, et M.-S. Hwang, « A Secure and Efficient One-time Password Authentication Scheme for WSN », *Int. J. Netw. Secur.*, vol. 19, n° 2, p. 177-181, mars 2017.

[54] S. Athmani, D. E. BOUBICHE, et A. . BILAMI, « Mécanisme de contrôle d'Autentification pour la Sécurité dans les RCSFs », présenté à International Conférence On Next Génération Networks & Services NGNS'10, MARRAKESH, 2010, p. 5.

[55] A. Perrig, R. Canetti, J. D. Tygar, et D. Song, « The TESLA Broadcast Authentication Protocol », *Rsa Cryptobytes*, vol. 5, p. 1-11, 2005.

[56] M. Luk, A. Perrig, et B. Whillock, « Seven cardinal properties of sensor network broadcast authentication », présenté à Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks. ACM, 2006, p. 147-156.

[57] K. Ren, W. Lou, K. Zeng, et P. Moran, « On Broadcast Authentication in Wireless Sensor Networks », *IEEE Trans. Wirel. Commun.*, vol. 6, n° 11, p. 4136-4144, nov. 2007.

[58] A. PERRIG, R. SZEWCZYK, J. D. TYGAR, V. WEN, et D. E. CULLER, « SPINS: Security Protocols for Sensor Networks », *Wirel. Netw.*, vol. 8, n° 5, p. 14, 2008.

[59] C. Karlof, N. Sastry, et D. Wagner, « TinySec: a link layer security architecture for wireless sensor networks », présenté à Proceedings of the 2nd international conference on Embedded networked sensor systems. ACM, 2004, p. 162-175.

[60] D. Jinwala, D. Patel, et K. Dasgupta, « FlexiSec: A Configurable Link Layer Security Architecture for Wireless Sensor Networks », *J. Inf. Assur. Secur.*, vol. 4, p. 582-603, 2009.

[61] M. Luk, G. Mezzour, A. Perrig, et V. Gligor, « MiniSec: A Secure Sensor Network Communication Architecture∗ », présenté à Proceedings of the 6th international conference on Information processing in sensor networks. ACM, 2007, p. 479-488.

[62] Y. CHALLAL, « Réseaux de Capteurs Sans Fils », *Univ. Technol. Compiegne Fr.*, p. 109, 2008.

[63] N. Mitta, R. Gouri, H. Lamari, et B. Moalige, « Crypto-Security Contribution in WSNs », *J. Emerg. Technol. Web Intell.*, vol. 6, n° 1, févr. 2014.

[64] Y. Xiao, V. K. Rayi, B. Sun, X. Du, F. Hu, et M. Galloway, « A survey of key management schemes in wireless sensor networks », *Comput. Commun.*, vol. 30, n° 11-12, p. 2314-2341, sept. 2007.

[65] S. A. CAMTEPE et B. YENER, « Key Distribution Mechanisms for Wireless Sensor Networks: a Survey », Rensselaer Polytechnic Institute, Troy, New York, Technical Report, 2005.

[66] F. Hu, J. Ziobro, J. Tillett, et N. K. Sharma, « Secure Wireless Sensor Networks: Problems and Solutions », *Rochester Inst. Technol. Rochester N. Y. USA*, vol. 1, n° 4, p. 11, 2004.

[67] Y. Maleh et A. Ezzati, « Etude et développement d'un protocole symétrique pour sécuriser les communications des RCSF », Thèse de doctorat, Faculté des Sciences et Technique de Settat, 2015.

[68] M. Momani, « Trust Models in Wireless Sensor Networks: A Survey », in *Recent Trends in Network Security and Applications*, vol. 89, N. Meghanathan, S. Boumerdassi, N. Chaki, et D. Nagamalai, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, p. 37-46.

[69] J. Lopez, R. Roman, I. Agudo, et C. Fernandez-Gago, « Trust management systems for wireless sensor networks: Best practices », *Comput. Commun.*, vol. 33, n° 9, p. 1086-1093, juin 2010.

[70] Z. Chen, L. Tian, et C. Lin, « Trust Model of Wireless Sensor Networks and Its Application in Data Fusion », *Sensors*, vol. 17, n° 4, p. 703, mars 2017.

[71] M. Momani et S. Challa, « Survey of Trust Models in Different Network Domains », *Int. J. Ad Hoc Sens. Ubiquitous Comput.*, vol. 1, n° 3, p. 1-19, sept. 2010.

[72] V. U. Rani et K. S. Sundaram, « Review of Trust Models in Wireless Sensor Networks », *Int J Comput Inf Syst Control Eng*, vol. 8, n° 2, p. 7, 2014.

[73] I. Butun, S. D. Morgera, et R. Sankar, « A Survey of Intrusion Detection Systems in Wireless Sensor Networks », *IEEE Commun. Surv. Tutor.*, vol. 16, n° 1, p. 266-282, 2014.

[74] K. Ioannis et T. Dimitriou, « Towards Intrusion Detection in Wireless Sensor Networks », présenté à Prociding of the 13th European Wireless Conference, 2007, p. 1-10.

[75] Z. S. Bojkovic, B. M. Bakmaz, et M. R. Bakmaz, « Security Issues in Wireless Sensor Networks », *Int. J. Commun.*, vol. 2, n° 1, p. 10, 2008.

[76] A. Abduvaliyev, A.-S. K. Pathan, J. Zhou, R. Roman, et W.-C. Wong, « On the Vital Areas of Intrusion Detection Systems in Wireless Sensor Networks », *IEEE Commun. Surv. Tutor.*, vol. 15, n° 3, p. 1223-1237, 2013.

[77] A. A. Strikos, « A full approach for Intrusion Detection in Wireless Sensor Networks », School of Information and Communication Technology Stockholm, Sweden 16453, 2007.

[78] B. Lai, S. Kim, et I. Verbauwhede, « Scalable Session Key Construction Protocol for Wireless Sensor Networks », *IEEE Workshop Large Scale Realt. Embed. Syst. LARTES*, p. 6, 2002.

[79] S. Zhu, S. Setia, et S. Jajodia, « LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks ∗ », *ACM Trans. Sens. Netw. TOSN 2006 P 500-528*, vol. 2, n° 4, p. 14, 2006.

[80] L. Eschenauer et V. D. Gligor, « A Key-Management Scheme for Distributed Sensor Networks∗ », *Proc. 9th ACM Conf. Comput. Commun. Secur. ACM*, p. 41-47, 2002.

[81] J. Zhang et V. Varadharajan, « Wireless sensor network key management survey and taxonomy », *J. Netw. Comput. Appl.*, vol. 33, n° 2, p. 63-75, mars 2010.

[82] D. Liu et P. Ning, « Establishing Pairwise Keys in Distributed Sensor Networks », *Distrib. Sens. Netw. ACM Trans. Inf. Syst. Secur. TISSEC*, vol. 8, n° 1, p. 41-77, 2005.

[83] R. Blom, « An Optimal Class of Symmetric Key Generation Systems », in *Advances in Cryptology*, vol. 209, T. Beth, N. Cot, et I. Ingemarsson, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, p. 335-338.

[84] S. A. Camtepe et B. Yener, « Combinatorial Design of Key Distribution Mechanisms for Wireless Sensor Networks », *Eur. Symp. Res. Comput. Secur. Springer Berl. Heidelb.*, p. 293-308., 2004.

[85] M. Eltoweissy, M. Moharrum, et R. Mukkamala, « Dynamic key management in sensor networks », *IEEE Commun. Mag.*, vol. 44, n° 4, p. 122-130, avr. 2006.

[86] V. T. KESAVAN, « Scalable and Secure Dynamic Key Management Framework for Static and Mobile Wireless Sensor Networks », Thèse de doctorat, KALASALINGAM ACADEMY OF RESEARCH AND EDUCATION, TAMIL NADU, INDIA, 2015.

[87] X. He, M. Niedermeier, et H. de Meer, « Dynamic key management in wireless sensor networks: A survey », *J. Netw. Comput. Appl.*, vol. 36, n° 2, p. 611-622, mars 2013.

[88] M. Eltoweissy, M. H. Heydari, L. Morales, et I. H. Sudborough, « Combinatorial Optimization of Group Key Management », *J. Netw. Syst. Manag.*, vol. 12, n° 1, p. 33-50, mars 2004.

[89] M. K.-R. R. Syed, H. Lee, S. Lee, et Y.-K. Lee, « MUQAMI+: a scalable and locally distributed key management scheme for clustered sensor networks », *Ann. Telecommun. - Ann. Télécommunications*, vol. 65, n° 1-2, p. 101-116, févr. 2010.

[90] X. Zhang, J. He, et Q. Wei, « EDDK: Energy-Efficient Distributed Deterministic Key Management for Wireless Sensor Networks », *EURASIP J. Wirel. Commun. Netw.*, vol. 2011, p. 1-11, 2011.

[91] G. Smaragdakis, I. Matta, et A. Bestavros, « SEP: A Stable Election Protocol for clustered heterogeneous wireless sensor networks », Boston University Computer Science Department, Boston, MA 02215, USA, BUCS-TR-2004-022, 2004.

[92] A. Pozzebon, C. Bove, I. Cappelli, F. Alquini, D. Bertoni, et G. Sarti, « Heterogeneous Wireless Sensor Network for Real Time Remote Monitoring of Sand Dynamics on Coastal Dunes », *IOP Conf. Ser. Earth Environ. Sci.*, vol. 44, p. 042030, oct. 2016.

[93] V. Mhatre et C. Rosenberg, « Homogeneous vs Heterogeneous Clustered Sensor Networks: A Comparative Study », présenté à Communications, 2004 IEEE International Conference on. IEEE, . p., 2004, p. 3646-3651.

[94] D. KIM, D. KIM, et S. AN, « Communication Pattern Based Key Establishment Scheme in Heterogeneous Wireless Sensor Networks », *KSII Trans. Internet Inf. Syst.*, vol. 10, n° 3, mars 2016.

[95] X. Du, Y. Xiao, M. Guizani, et H.-H. Chen, « An effective key management scheme for heterogeneous sensor networks », *Ad Hoc Netw.*, vol. 5, n° 1, p. 24-34, janv. 2007.

[96] Q. Shi, N. Zhang, M. Merabti, et K. Kifayat, « Resource-efficient authentic key establishment in heterogeneous wireless sensor networks », *J. Parallel Distrib. Comput.*, vol. 73, n° 2, p. 235-249, févr. 2013.

[97] J. Wang, H. Wang, X. A. Wang, et Y. Cao, « An Authentication Key Agreement Scheme for Heterogeneous Sensor Network Based on Improved Counting Bloom Filter », présenté à P2P, Parallel, Grid, Cloud and Internet Computing (3PGCIC), 2015 10th International Conference on. IEEE, 2015, p. 815-820.

[98] Z. Qin, X. Zhang, K. Feng, Q. Zhang, et J. Huang, « An Efficient Identity-Based Key Management Scheme for Wireless Sensor Networks Using the Bloom Filter », *Sensors*, vol. 14, n° 10, p. 17937-17951, sept. 2014.

[99] W. Yao, S. Han, et X. Li, « LKH++ Based Group Key Management Scheme for Wireless Sensor Network », *Wirel. Pers. Commun.*, vol. 83, n° 4, p. 3057-3073, août 2015.

[100] R. D. Pietro, L. V. Mancini, et S. Jajodia, « Efficient and Secure Keys Management for Wireless Mobile Communications∗ », présenté à Proceedings of the second ACM international workshop on Principles of mobile computing. ACM, 2002, p. 66-73.

[101] C.-C. Chang, W.-Y. Hsueh, et T.-F. Cheng, « A Dynamic User Authentication and Key Agreement Scheme for Heterogeneous Wireless Sensor Networks », *Wirel. Pers. Commun.*, vol. 89, n° 2, p. 447-465, juill. 2016.

[102] A. K. Das, « An unconditionally secure key management scheme for large-scale heterogeneous wireless sensor networks », présenté à Communication Systems and Networks and Workshops, 2009. COMSNETS 2009. First International. IEEE, 2009, p. 1-10.

[103] J.-Y. Huang, I.-E. Liao, et H.-W. Tang, « A Forward Authentication Key Management Scheme for Heterogeneous Sensor Networks », *EURASIP J. Wirel. Commun. Netw.*, vol. 2011, p. 1-10, 2011.

[104] M. R. Alagheband et M. R. Aref, « A Secure Key Management Framework for Heterogeneous Wireless Sensor Networks », in *Communications and Multimedia Security*, vol. 7025, B. De Decker, J. Lapon, V. Naessens, et A. Uhl, Éd. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, p. 18-31.

[105] P. Traynor, R. Kumar, H. Choi, G. Cao, S. Zhu, et T. La Porta, « Efficient Hybrid Security Mechanisms for Heterogeneous Sensor Networks », *IEEE Trans. Mob. Comput.*, vol. 6, n° 6, p. 663-677, juin 2007.

[106] S. M. Mizanur Rahman et K. El-Khatib, « Private key agreement and secure communication for heterogeneous sensor networks », *J. Parallel Distrib. Comput.*, vol. 70, n° 8, p. 858-870, août 2010.

[107] Seung-Hyun Seo, Jongho Won, S. Sultana, et E. Bertino, « Effective Key Management in Dynamic Wireless Sensor Networks », *IEEE Trans. Inf. Forensics Secur.*, vol. 10, n° 2, p. 371-383, févr. 2015.

[108] Z. Mahmood, H. Ning, et A. Ghafoor, « A Polynomial Subset-Based Efficient Multi-Party Key Management System for Lightweight Device Networks », *Sensors*, vol. 17, n° 4, p. 670, mars 2017.

[109] E. S. Kumar, « Random Keying Technique for Security in Wireless Sensor Networks Based on Memetics », *Int. J. Comput. Sci. Theory Appl.*, vol. 1, n° 2, p. 25-31, 2014.

[110] K. Saleem *et al.*, « Cost-Effective Encryption-Based Autonomous Routing Protocol for Efficient and Secure Wireless Sensor Networks », *Sensors*, vol. 16, n° 4, p. 460, mars 2016.

[111] K. Saleem, N. Fisal, M. S. Abdullah, et S. H. S. Ariffin, « Biological inspired secure autonomous routing mechanism for wireless sensor networks », *Int. J. Intell. Inf. Database Syst.*, vol. 5, n° 4, p. 313-337, 2011.

[112] S. Athmani, A. Bilami, et D. E. Boubiche, « EDAK: An Efficient Dynamic Authentication and Key Management Mechanism for heterogeneous WSNs », *Future Gener. Comput. Syst.*, nov. 2017.

[113] D. E. BOUBICHE et A. . BILAMI, « HEEP (Hybrid Energy Efficiency Protocol) based on chain clustering », *Int. J. Sens. Netw.*, vol. 10, n° 1/2, p. 25-35, 2011.

[114] W. YE, J. HEIDEMANN, et D. ESTRIN, « An Energy-Efficient MAC Protocol for Wireless Sensor Networks », *Wirel. Sens. Netw.*, p. 11, 2008.

[115] Y. Sankarasubramaniam, I. F. Akyildiz, et S. W. McLaughlin, « Energy efficiency based packet size optimization in wireless sensor networks », présenté à Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on. IEEE, 2003, p. 1-8.

[116] B. MAALA, Y. CHALLAL, et H. BETTAHAR, « Node capture attack impact on key management schemes for heterogeneous wireless sensor networks », présenté à Information Infrastructure Symposium, 2009. GIIS'09. Global. IEEE, 2009, p. 1-7.