République Algérienne Démocratique et Populaire

Ministère de L'Enseignement Supérieur et de la Recherche Scientifique Université 20 Août 1955 Skikda.

Faculté des Sciences Département d'Informatique

Mémoire de fin d'études en vue de l'obtention du diplôme De :

Master

OPTION : GÉNIE LOGICIEL AVANCÉ ET APPLICATIONS (GLAA).

Thème :

Mise en œuvre d'un algorithme de sécurisation des données dans un réseau de capteurs sans fil.

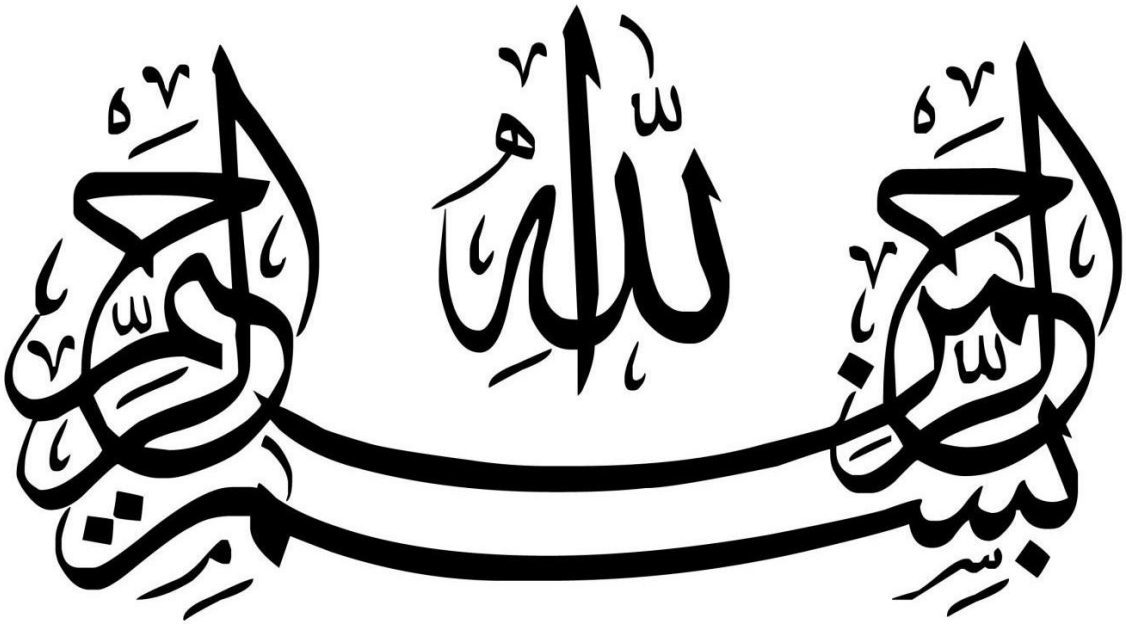
Réalisé par :

- *Belmeguenai Houssem Eddine.*
- *Mouhamed Ben Ali Alaa Eddine.*

Encadré par :

- ***Pr.*** *Redjimi Mouhamed.*

Session : Juin 2023


بسم الله الرحمن الرحيم

ℜ Remerciement ℘

Nos premiers remerciements et notre grande gratitude s'adressent au Pr ; Redjimi Mouhamed, notre directeur de projet. Ses conseils avisés et ses encouragements ont permis à ce travail d'aboutir avec succès. Ses compétences scientifiques ont été un grand soutien pour nous. La liberté qu'il nous a accordée et les responsabilités qu'il nous a confiées ont grandement contribué à notre formation et à notre autonomie de travail. Ses lectures attentives, ses critiques et ses suggestions ont été d'une précieuse aide pour la réalisation de ce travail de recherche.

Nous le remercions chaleureusement pour sa pédagogie, sa patience, sa disponibilité et son dévouement. Travailler sous sa direction a été pour nous un grand honneur et une immense satisfaction.

Nous tenons également à exprimer nos sincères remerciements aux membres du jury pour avoir accepté de juger ce modeste travail.

Nous voulons également exprimer notre profonde gratitude envers notre famille ; nos parents, grands-parents, notre frère et notre sœur qui nous ont aidé à surmonter tous les obstacles et nous ont soutenu durant toute cette période de travail.

Nous souhaitons remercier tout particulièrement une personne qui a été continuellement présente Pr ; Belmeguenai Aissa , qui nous a beaucoup épaulés par son aide, son soutien et ses encouragements.

Enfin, nous tenons à remercier tous ceux qui nous ont aidés de près ou de loin à la réalisation de ce travail.

Dédicaces

Je dédie ce projet :

A mon cher Pére ***Ibrahim***

Qui m'a toujours poussé et motivé dans mes études. Merci énormément papa pour ton soutien plus que précieux, merci pour ton grand cœur toutes vos qualités qui seraient trop longues à énumérer. Ma vie ne serait pas aussi magique sans ton présence et ton amour mon papounet d'amour

A ma très chère mère ***Saida***

La lumière de mes jours. Tu représente pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement. Tu as toujours été présente à mes cotés pour me soutenir et m'encourager. Je dédie ce travail en témoignage de mon profond amour.

A mes chers, mon frère ***Ahmed*** *et mes sœurs* ***Anfel*** *et* ***Asma*** *pour leurs encouragement spermanents et leur appui et leur soutien moral, les mots ne suffisent guère pour exprimé l'attachement, l'amour et l'affection que je porte pour vous.*

Enfin, a tous ceux qui ont contribué de prés ou de loin a la réalisation de ce travail.

Houssem Eddine .

Dédicaces

Je dédie ce projet :

A mon cher Pére ***Rachid***

Qui m'a toujours poussé et motivé dans mes études. Merci énormément papa pour ton soutien plus que précieux, merci pour ton grand cœur toutes vos qualités qui seraient trop longues à énumérer. Ma vie ne serait pas aussi magique sans ton présence et ton amour mon papounet d'amour

A ma très chère mère ***Dalila***

La lumière de mes jours. Tu représente pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement. Tu as toujours été présente à mes cotés pour me soutenir et m'encourager. Je dédie ce travail en témoignage de mon profond amour.

A mes chers, mon frère ***Soheyb*** *et mes sœurs pour leurs encouragement spermanents et leur appui et leur soutien moral, les mots ne suffisent guère pour exprimé l'attachement, l'amour et l'affection que je porte pour vous.*

Enfin, a tous ceux qui ont contribué de prés ou de loin a la réalisation de ce travail.

Ala Eddine.

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.

L'utilisation d'un algorithme de chiffrement est une méthode courante pour sécuriser les données dans ces réseaux. Dans ce travaille on a parler sur L'un de ces algorithmes : l'algorithme AES (Advanced Encryption Standard). L'intégration de l'algorithme AES avec le mode Galois/Counter (GCM) constitue une approche très efficace pour assurer une transmission sécurisée des données au sein des réseaux de capteurs sans fil. AES, un algorithme de chiffrement symétrique reconnu pour sa robustesse et son adoption généralisée, combiné à GCM, un mode de fonctionnement offrant des vérifications d'intégrité des données et une authentification, établit une base solide pour protéger les informations sensibles dans les RCSF.

En utilisant AES-GCM, les RCSF peuvent se défendre contre les accès non autorisés, la manipulation des données et les tentatives de contrefaçon. L'intégrité des données transmises est assurée grâce au chiffrement authentifié fourni par GCM, qui protège non seulement contre les modifications non autorisées, mais détecte également toute tentative de manipulation. Cette combinaison établit une base de confiance solide au sein du réseau et protège la confidentialité, l'intégrité et l'authenticité des données transmises.

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

Abstract :

Securing network communication is one of the most significant challenges in wireless sensor networks. Most security protocols are built around powerful encryption and authentication algorithms. To achieve security goals, key management is the first fundamental function since sensor nodes require a valid common key to operate cryptography mechanisms.

The use of an encryption algorithm is a common method to secure data in these networks. In our work we talked about One of these algorithms: AES algorithm, the integration of the AES algorithm with the Galois/Counter Mode (GCM) presents a highly effective approach to ensure secure data transmission within Wireless Sensor Network (WSN). AES, a symmetric encryption algorithm known for its robustness and widespread adoption, combined with GCM, a mode of operation offering data integrity and authentication, establishes a strong foundation for safeguarding sensitive information in WSN.

By employing AES-GCM, WSN can defend against unauthorized access, data tampering, and forgery attempts. The integrity of transmitted data is assured through the authenticated encryption provided by GCM, which not only protects against unauthorized modifications but also detects any tampering attempts. This combination establishes a strong trust foundation within the network and safeguards the confidentiality, integrity, and authenticity of the transmitted data.

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

ملخص :

تأمين الاتصالات الشبكية هو أحد أهم التحديات في شبكات الاستشعار اللاسلكية. تعتمد معظم بروتوكولات الأمان على خوارزميات قوية للتشفير والمصادقة. لتحقيق أهداف الأمان، إدارة المفاتيح هي الوظيفة الأساسية الأولى، حيث يحتاج أجهزة الاستشعار إلى مفتاح مشترك صالح لاستخدام آليات التشفير.

استخدام خوارزمية التشفير هو أسلوب شائع لتأمين البيانات في هذه الشبكات. في هذا العمل تحدثنا عن واحدة من هذه الخوارزميات وهي خوارزمية التشفير المتقدمة (أ,و,س). تكامل خوارزمية التشفير المتقدمة مع (ج، س،م) (العداد/جالوا) يمثل نهجًا فعالًا جدًا لضمان نقل البيانات بأمان في شبكات الاستشعار اللاسلكية. خوارزمية التشفير(أ،و،س) هي خوارزمية تشفير متناظرة تعتبر قوية ومستخدمة على نطاق واسع، وبالتزامن مع (ج، س،م) (العداد/ جالوا) ، الذي يوفر فحوصات للسلامة ومصادقة البيانات، يتم تأسيس أساس قوي لحماية المعلومات الحساسة في شبكات الاستشعار اللاسلكية.

خوارزمية التشفير المتقدمة (أ,و,س) ، مع طول المفتاح المتغير وحجم البلوكات، توفر مرونة لتلبية متطلبات الأمان المتنوعة. إن مقاومتها المثبتة للهجمات التشفيرية واستخدامها الواسع في تطبيقات مختلفة يجعلها اختيارًا موثوقًا لحماية سرية البيانات. عندما يتم دمجها مع وضع (ج، س،م) (العداد/جالوا)، الذي يوفر فحوصات للسلامة ومصادقة البيانات، يتم تعزيز الأمان العام لشبكات الاستشعار اللاسلكية بشكل كبير.

من خلال استخدامها، يمكن لشبكات الاستشعار اللاسلكية الدفاع ضد الوصول غير المصرح به وتزييف البيانات ومحاولات التزوير. يتم ضمان سلامة البيانات المرسلة من خلال التشفير المصادق المقدم، الذي لا يحمي فقط من التعديلات غير المصرح بها ولكن يكشف أي محاولة للتلاعب. هذه الجمعية تؤسس أساسًا قويًا للثقة داخل الشبكة وتحمي سرية وسلامة وأصالة البيانات المرسلة.

كلمات البحث الأساسية : شبكة المجسات الاسلكية, الامن , إدارة مفاتيح التشفير, المصداقية.

TABLE DES MATIERES :

Chapitre 3 : Attaques de Sécurité dans les RCSF.

Chapitre IV : Analyse et résultats.

LISTE DES FIGURES:

Liste Des Figures.

Liste Des Figures.

Liste Des Figures.

LISTE DES TABLEAUX:

LISTE DES ABRIVIATIONS:

MANET	Mobile Ad hoc NET Works.
ADCs	Analog Digital Converter.
GPS	Global Positioning System.
RCSF	Réseau de capteurs sans fil.
WSN	Wireless Sensor Network.
SMP	Sensor Management Protocol.
TADAP	Task Assignement and Data Advertisement Protocol.
UDP	User Datagram Protocol.
TCP	Transmission Control Protocol.
LEACH	Low-Energy Adaptive Clustering Hierarchy.
SAR	Sequential Assignement Routing.
SMACS	Self-organizing Medium Access Control for Sensor networks.
EAR	Eavesdrop And Register.
RSA	Rivest Shamir Adleman.

MD5	Message Digest 5.
SHA-1	Secure Hash Algorithm.
MAC	Message Authentication Code.
HMAC	keyed-Hash Message Authentication Code.
INSENS	Intrusion-tolerant routing for wireless sensor networks.
SAWN	Secure Aggregation for Wireless Networks.
AES	Advanced Encryption Standard.
GCM	Galois/Counter Mode.
LEAP	Lightweight Extensible Authentication Protocol.

Introduction Générale.

INTRODUCTION GÉNÉRALE.

Les réseaux capteurs sans fil (RCSF) sont devenus une technologie essentielle dans de nombreux domaines, tels que la surveillance environnementale, les systèmes de santé, la gestion de l'énergie, etc. Ces réseaux sont composés de nombreux capteurs sans fil interconnectés qui collectent des données et les transmettent à une station de base pour le traitement ultérieur. Cependant, en raison de leur nature sans fil et de leur déploiement dans des environnements non sécurisés, les RCSF sont vulnérables aux attaques et aux interceptions de données sensibles.

Dans ce contexte, l'une des préoccupations majeures est la sécurité des données transmises dans les RCSF. La protection de l'intégrité, de la confidentialité et de l'authenticité de ces données est essentielle pour garantir le bon fonctionnement des applications déployées et prévenir les fuites d'informations sensibles.

Afin de relever ce défi, un algorithme de sécurité avancé peut être développé pour sécuriser les données transmises dans les RCSF. Cet algorithme devrait être capable de fournir des mécanismes de chiffrement robustes, d'assurer l'authentification des nœuds et des données, de détecter les tentatives d'intrusion et de garantir la confidentialité des informations échangées.

L'algorithme de sécurité proposé peut être basé sur des techniques telles que le chiffrement symétrique ou asymétrique, les fonctions de hachage cryptographique, les certificats numériques, les signatures numériques, etc. En utilisant ces méthodes, les données transmises dans les RCSF peuvent être protégées de manière efficace contre les attaques potentielles.

En outre, il est également important de prendre en compte les contraintes spécifiques des RCSF, telles que les ressources limitées des nœuds capteurs (comme la puissance de calcul, la mémoire et la consommation d'énergie). L'algorithme de sécurité doit être conçu de manière à minimiser l'impact sur ces ressources tout en garantissant une protection adéquate des données.

Dans cette étude, nous nous proposons de concevoir et d'implémenter un algorithme de sécurité novateur pour les RCSF. Nous évaluerons les performances de cet algorithme en termes de confidentialité, d'intégrité et d'authentification des données transmises. En utilisant des mesures telles que la vitesse, la taille mémoire, nous évaluerons également l'efficacité de l'algorithme en termes de consommation de ressources.

En conclusion, la sécurité des données transmises dans les RCSF est d'une importance primordiale pour garantir leur bon fonctionnement et éviter les fuites d'informations sensibles. Un algorithme de sécurité efficace, basé sur des techniques cryptographiques avancées et prenant en compte les contraintes des RCSF, peut jouer un rôle crucial dans la protection des données. Cette étude vise à apporter une contribution significative dans ce domaine en proposant un nouvel algorithme de sécurité pour les RCSF et en évaluant ses performances.

Organisation du mémoire :

Le présent mémoire est organisé en quatre chapitres répartis dans deux parties principales : une partie état de l'art et une partie solution. La partie état de l'art présente des généralités sur les réseaux de capteurs, le routage et la sécurité. La partie solution expose notre proposition pour sécuriser l'opération de chifferement des données transmisent dans les réseaux de capteurs sans fil et les résultats obtenus après l'application de notre approche.

Dans le chapitre I : on présente les réseaux de capteurs sans-fil, en passant par les caractéristiques, les domaines d'application, l'architecture, la communication ainsi que les différentes problématiques des réseaux de capteurs sans-fil.

Dans le chapitre II : on aborde la sécurité et le concept de la confiance dans les réseaux de capteurs sans-fil.

Dans le chapitre III : on propose notre algorithme de chiffrement, pour sécuriser les données dans un réseau de capteurs sans-fil.

Dans le chapitre IV : nous simulons les résultats de notre algorithme proposé afin d'évaluer ces performances. Enfin, on termine par une conclusion générale et on présente quelques perspectives de travail pour le futur.

CHAPITRE I:

GENERALITE SUR LES RESAUX DU CAPTEUR SANS FIL.

I.1.Introduction :

Ces dernières années, Internet a suscité un intérêt croissant pour la recherche, l'éducation et les affaires. En conséquence, le nombre de personnes accédant à Internet pour le travail, les études ou les loisirs augmente considérablement, tout comme les services proposés sur ce réseau (messagerie électronique, e-commerce, e-learning, etc.). Cette diversité de services et d'utilisateurs s'explique en grande partie par le fait qu'Internet regroupe un grand nombre de réseaux différents. D'autre part, les progrès dans le domaine du sans fil ont contribué à la croissance d'Internet en facilitant l'accès des utilisateurs. Les développements dans le domaine de la communication sans fil et de l'informatique mobile sont de plus en plus populaires et les composants mobiles sont de plus en plus fréquents (PDA, ordinateurs portables, téléphones portables). Cela a permis une nouvelle classe de réseaux sans fil appelés réseaux de capteurs sans fil.

Dans ce chapitre, nous présenterons les réseaux de capteurs sans fil, leurs architectures de communication, leurs applications. Nous discuterons également les principaux facteurs et contraintes qui influencent la conception des réseaux de capteurs sans fil.

I.2. Les réseaux Ad hoc :

Les réseaux Ad hoc (en latin : ***« qui va vers ce vers quoi il doit aller »***, c'est-á-dire « formé dans un but précis », telle qu'une commission ***ad hoc***, formée pour régler un problème particulier sont des réseaux sans fil capables de s'organiser sans infrastructure définie préalablement.

Les réseaux ad hoc, dans leur configuration mobile, sont connus sous le nom de **MANET** (pour Mobile Ad hoc NET Works) [1]. est un ensemble d'unités mobiles équipées d'une interface de communication sans fil, formant un réseau temporaire sans recourir à aucune infrastructure fixe ou administration centralisée. Dans de tels environnements, les unités se comportent , comme des hôtes et/ou des routeurs.

Les nœuds des **MANET's** sont équipés d'émetteurs et de récepteurs sans-fil utilisant des antennes qui peuvent être omnidirectionnelles (broadcast), fortement directionnelles (point à point), ou une combinaison de ces deux types. Ils maintiennent d'une manière coopérative la connectivité du réseau, en fonction de leurs positions, la configuration de leurs émetteurs/récepteurs, la puissance de transmission et les interférences entre les canaux de communication. La modélisation de cette connectivité est détaillée dans la section suivante. Un réseau ad hoc peut être isolé, mais il peut aussi avoir des passerelles ou des interfaces qui le relient à un réseau fixe.

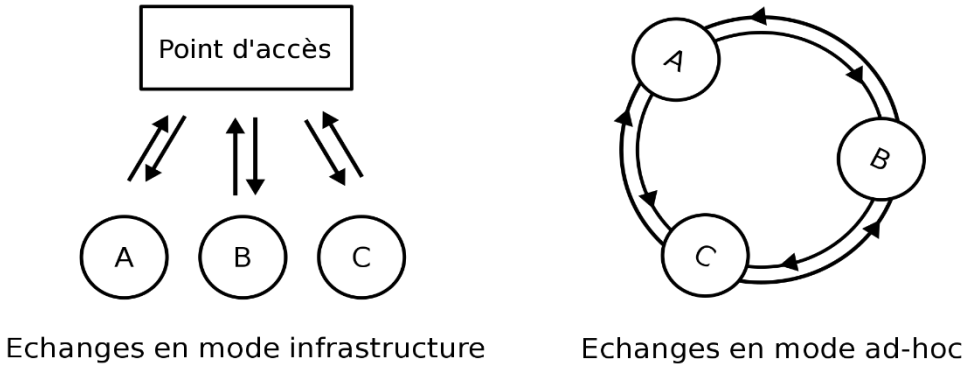


Figure I.1. *Les échanges dans les modes Infrastructure et Ad-hoc .*

I.2.1. Les problèmes de l'ad hoc :

Les principaux problèmes des réseaux ad hoc, et les problématiques à gérer sont :

- ✓ Absence d'infrastructure.
- ✓ Bande passante limitée.
- ✓ Perte de données.
- ✓ Perte de routes.
- ✓ Contraintes de consommation d'énergie.
- ✓ Sécurité limitée.
- ✓ Erreur de transmission.
- ✓ Interférences.
- ✓ Noeuds caches.

I.3. Capteurs sans-fil :

I.3.1. Définition :

Les capteurs sont des dispositifs miniaturisés possédants des ressources énergétiques limités et autonomes, capables de traiter des informations et de les transmettre via des ondes radio. Parmi ces phénomènes récoltés nous pouvons citer ceux ayant trait à la température, l'humidité, la pression, capture d'image, etc…

Les capteurs prélèvent une information sur le comportement de la partie opérative et la transforment en une information exploitable par la partie commande. Une information est une grandeur abstraite qui précise un événement parti culier parmi un ensemble d'événements possibles. Pour pouvoir être traitée, cette information sera portée par un support physique (énergie) on parlera alors de signal. Les signaux sont généralement de nature électrique ou pneumatique [2]. On peut caractériser les capteurs selon deux critères: - en fonction de la grandeur mesurée : on parle alors de capteur de position, de température, de vitesse, de force, de pression, etc.; - en fonction du caractère de l'information délivrée : on parle alors de capteurs logiques appelés aussi capteurs tout ou rien (TOR), de capteurs analogiques ou numériques.

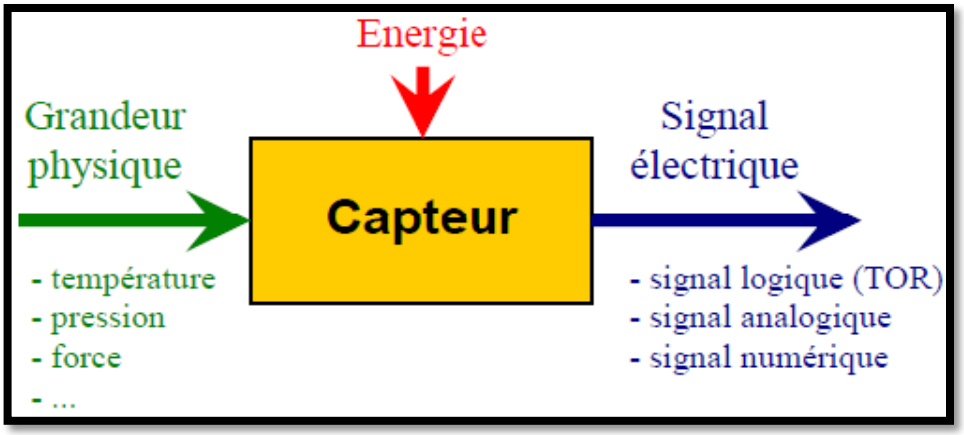


Figure I.2. *Schéma représentatif du mécanisme de traduction d'un capteur.*

On peut alors classer les capteurs en deux catégories, les capteurs à contact qui nécessitent un contact direct avec l'objet à détecter et les capteurs de proximité. Chaque catégorie peut être subdivisée en trois catégories de capteurs: **les capteurs mécaniques**, **électriques**, **pneumatiques**.

En fin nous dirons qu'un capteur permet l'analyse de l'environnement dans lequel il se trouve (car il effectue des relevé de mesures), cette analyse se fait quand le capteur transmet des informations aux autres capteurs se situant dans sa zone de couverture. L'information va alors circuler en se propageant de capteur en capteur traversant ainsi plusieurs zones du réseau permettant d'assurer la couverture de ce dernier.

I.3.2. Architecture d'un capteur :

Concernant l'architecture des capteurs on peut la scinder en deux parties, ces deux parties à s'avoir l'architecture matérielle et l'architecture logicielle son t complémentaires car elles composent le capteur dans son intégralité, à cet effet nous développerons les deux architectures. - Architecture matérielle Un capteur se compose de quatre unités de base qui se schématisent dans la figure qui suit :

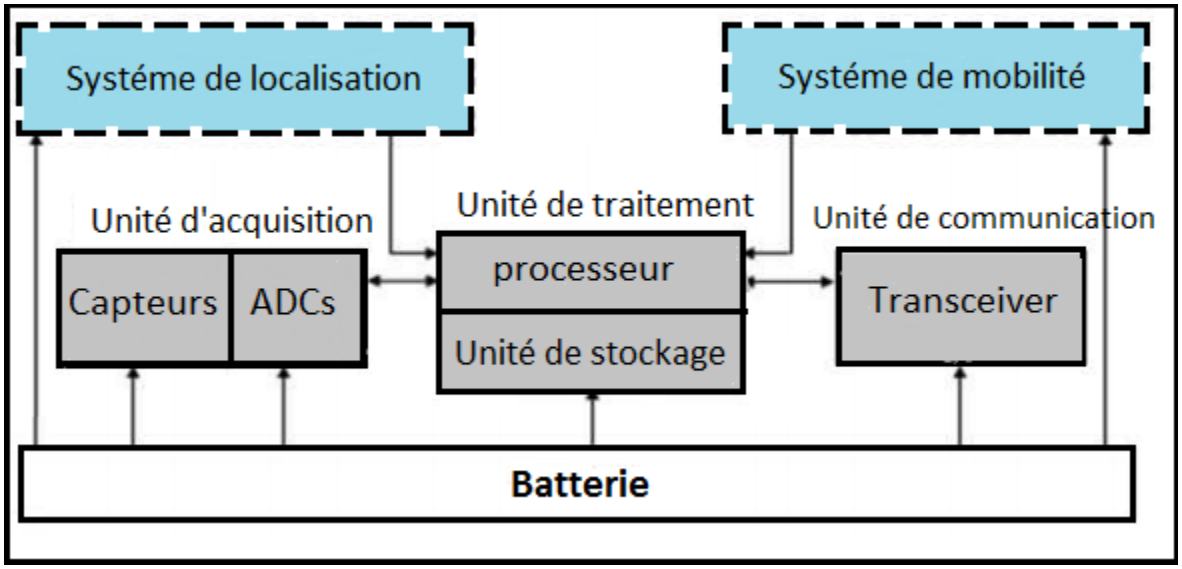


Figure I.3. *Architecture d'un nœud capteur.*

- ***L'unité d'acquisition :***

Appelée aussi unité de captage elle se compose généralement de deux sous unités à s'avoir les capteurs et les ADCs (Analog Digital Converter) qui sont des convertisseurs analogique-numérique. Les capteurs permettent une mesure sur des

paramètres environnementaux pour fournir des signaux analogiques obtenus après conversation de ces données récoltées.

Les ADCs vont convertir ces signaux analogiques en signaux numériques.

- ***L'unité de traitement :***

Se compose de deux interfaces une avec l'unité d'acquisition et l'autre avec l'unité de communication, son rôle est le contrôle du bon fonctionnement des autres unités un système d'exploitation nécessaire au fonctionnement du capteur peut y être embarqué sur certain modèle.

Cette unité permet l'exécution de procédures de communication qui permettent la collaboration d'un nœud avec les autres nœuds du réseau ; elle permet aussi l'analyse des données récoltées afin d'alléger le travail du nœud puits.

- ***L'unité de communication :***

Cette unité permet d'effectuer toutes les communications entre les différents nœuds sur un médium sans fil, car elle est dotée d'un émetteur/récepteur.

- ***Batterie :***

Elle alimente les unités que nous avons citées et elle n'est généralement ni rechargeable ni remplaçable.La contrainte majeure lors de la conception de protocoles pour les réseaux de capteurs réside dans la capacité d'énergie limitée de ces capteurs. Certains capteurs sont équipés de composants supplémentaires tels que des systèmes de localisation GPS (Global Positioning System) [3].

I.3.3. Exemples des types de capteurs :

Actuellement, il existe une grande variété de capteurs disponibles sur le marché, chacun présentant des caractéristiques et des fonctionnalités spécifiques. Ces capteurs sont généralement

conçus en fonction de l'application pour laquelle ils seront utilisés, ce qui conduit à une diversité considérable de capteurs disponibles.

La figure (I.4), illustre la diversité des domaines des capteurs :

Figure I.4. *Schéma représentant quelques types de capteurs.*

I.4. Les réseaux capteurs sans fil :

Grâce au progrès fait dans le domaine de la miniaturisation des systèmes de microélectro-mécanique (MEMS) et dans le marché des réseaux et des applications sans fil, s'est créée une nouvelle branche de réseaux mobile afin d'offrir des solutions économiquement intéressantes pour la surveillance à distance et le traitement des données dans des environnements complexes : les réseaux de capteurs sans fil (Wireless Sensor networks).

Les nouvelles technologies peuvent réduire la taille, le coût et la consommation d'énergie, et augmenter la précision et les performances des capteurs, des processeurs et des circuits spécifiques. Ainsi, un grand nombre de capteurs peuvent être conçus, intégrés et organisés en réseau. Les capteurs sans fil sont fabriqués à l'aide de la première génération de circuits commerciaux appelés {mote}[4], en raison du système de détection à distance .

Le développement des ces réseaux de capteurs constitue donc très certainement une prochaine étape dans l'évolution des technologies de l'information.

I.4.1. Définitions :

- ***Définition 1:*** Les réseaux de capteurs utilisent un grand nombre de dispositifs très petits, nommés « noeuds capteurs », pour former un réseau sans infrastructure établie. Dans ces réseaux, chaque noeud est capable de détecter son environnement et de traiter l'information au niveau local ou de l'envoyer à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil . [5]
- ***Définition 2 :*** Les réseaux de capteurs sont, par définition, des systèmes à milliers de noeuds ayant une zone de couverture extrémement réduite (de l'ordre de 3m), déployés d'une manière dense dans un environnement hétérogène. De plus, chaque noeud du réseau dispose d'une réserve énergétique (ex.pile) ayant une durée de vie limitée et dont le remplacement peut s'avérer impossible. [6]
- ***Définition 3 :*** Un RCSF est un acronyme pour réseau de capteurs sans fil ou WSN (Wireless Sensor Network), souvent apparenté au réseau **ad-hoc** de leur utilisation commune des ondes radio, ainsi que leurs architectures décentralisées ; les RCSF sont aussi considérés comme leurs successeurs. Les RCSF sont des systèmes embarqués, ils se composent d'un ensemble d 'unités de traitement embarqué appelé « motes » qui communiquent a travers des liens sans fil, le déploiement de plusieurs unités se fait dans le but de collecter ou de capter des données (de type sonores, vibrations, lumière…) et de les transmettre. Grace aux récentes avancées technologiques le développement de très petits capteurs à faible cout et consommant peu d'énergie a pu être réalisé. Afin de mieux cerner les réseaux de ces capteurs sans fil la figure qui suit représente leur mode de fonctionnement global.

I.4.2. Comparaison entre les RCSF et les réseaux Ad-hoc :

Le tableau (I.1) présente une comparaison entre les réseaux de capteurs et les réseaux ad hoc [7] :

Réseaux de capteurs sans fil	***Réseaux Ad-hoc***
Objectif bien ciblé	Objectif général en communication
Noeuds en collaboration	Chaque noeud a son propre objectif
Flot de données « many to one »	Flot «any to any»
Très grand nombre de noeuds	Nombre limité de noeuds avec notion de ID
Energie comme facteur déterminant	Débit majeur
Communication broadcast	Communication point à point

Tableau I.1. *Comparaison entre capteurs et A-hoc.*

I.4.3. Historique d'évolution des réseaux de capteurs sans fil :

La technologie des réseaux de capteurs sans fil est devenue l'une des merveilleuses technologies dans le 21ème siècle ; les réseaux de capteurs ont montré leur impact sur notre vie quotidienne, CHONG, et AL [8] ont parlé de trois générations des noeuds de capteurs. le tableau ci-dessus montre les différentes générations de noeuds de capteurs :

Génération	***Période***	***Taille***	***poids***	***Batterie***
1ère	1800 - 1900	Taille d'une grande boite de chaussure	Quelques Kg	Grosse
2ème	2000 - 2003	Taille d'une boite de cartes	Quelques grammes	AA
3ème	A partir de 2010	Taille de particule de poussière	Négligeable	Solaire

Tableau I.2. *Génération de noeuds de capteurs.*

Les capteurs traditionnels mesurant une grandeur physique, sont présents depuis des décennies dans des domaines comme l'industrie, l'aéronautique ou l'automobile. Ils sont en général reliés à la base de traitement filaire mais la nouveauté des nouveaux réseaux de capteurs est qu'ils ont la possibilité de communiquer par ondes radio (Wifi ou Zig Bee) avec d'autres capteurs proches.

La miniaturisation du matériel et la multiplication des moyens de connexions associées à l'augmentation des capacités de calcul et de mémoire en informatique ont permis aux réseaux de capteurs d'exister, et cela à une échelle très large au point d'accomplir les tâches les plus complexes pour l'humain. Ainsi, on peut les retrouver désormais dans l'armement, le nucléaire, le sauvetage, la sauvegarde de l'environnement, la médecine, etc.

La recherche continue pour perfectionner le fonctionnement des futurs réseaux de capteurs. De nombreux travaux sont effectués pour résoudre les trois difficultés majeures auxquelles sont confrontés les capteurs : l'énergie, la puissance d'émission, la capacité de stockage et de calcul.

I.4.4. Les réseaux de capteurs multimédia :

Les réseaux de capteurs multimédia sont des réseaux sans fil composés de nœuds capables de gérer des données multimédias non scalaires telles que des images, des vidéos et des sons. Comparés aux réseaux de capteurs sans fil standards, ces réseaux présentent des défis particuliers en ce qui concerne la détection et le routage des données vers les collecteurs et les utilisateurs. Cela est dû à la volumétrie importante des données traitées, ce qui complique leur traitement, leur stockage et leur transmission.

I.5. Domaines d'applications des RCSF :

Grace aux évolutions de la technologie touchant les domaines : électronique, informatique, industrielle, instrumentation, réseaux et télécommunication, le champ d'applications des réseaux de capteurs sans fils est de plus en plus en élargissement. Parmi les applications des RCSF nous trouvons :

I.5.1. Applications militaires :

Le faible coût, le déploiement rapide, l'auto-organisation et la tolérance aux pannes sont des caractéristiques qui ont rendu les réseaux de capteurs efficaces pour les applications militaires.

Comme la plupart des technologies, les applications militaires étaient les premières à intégrer les RCSFs. Ils sont déployés dans un secteur stratégique ou difficile d'accès, pour y surveiller tous les mouvements (alliés ou ennemis), pour analyser le champ de bataille avant d'envoyer du renfort, la détection des attaques biologiques ou chimiques,…etc

Figure I.5.*Robot militaire commandé sans fil.*

I.5.2. Applications médicales :

Dans le domaine de la médecine, les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers,…etc.). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques (la tension artérielle, battements du cœur,…etc.) à l'aide des capteurs ayant chacun une tâche bien particulière. C'est ainsi que parmi les nouvelles venues on peut citer la télésurveillance des signes vitaux et des niveaux d'activité à domicile des personnes âgées ou

handicapées ; le déploiement de micro-capteurs sur le corps humain ou éventuellement implantés permet une surveillance permanente des constantes vitales d'un patient.

I.5.3. Applications environnementales:

Les réseaux de capteurs peuvent être utilisés pour surveiller les changements environnementaux. Ils servent à déterminer les valeurs de certains paramètres à un endroit donné, comme par exemple : la température, la pression atmosphérique, etc. En dispersant des noeuds capteurs dans la nature, on peut détecter des événements tels que des feux de forêts, des tempêtes ou des inondations. Ceci permet une intervention beaucoup plus rapide et efficace des secours. Dans le domaine de l'agriculture, les capteurs peuvent être utilisés pour réagir convenablement aux changements climatiques, par exemple en déclenchant le processus d'arrosage lors de la détection de zones sèches dans un champ agricole.

Figure I.6. *Cas d'application des RCSF relative aux feux de forêt.*

I.5.4. Applications à la surveillance :

L'application des réseaux de capteurs dans le domaine de la sécurité permet de réduire considérablement les dépenses financières engagées pour assurer la sécurité des lieux et des personnes. Par conséquent, l'intégration de capteurs dans de grandes structures telles que des ponts ou des bâtiments aidera à détecter les fissures et les changements dans les structures après des tremblements de terre ou une dégradation structurelle. Le déploiement d'un réseau de capteurs de mouvement peut constituer un système d'alarme qui servira à détecter les intrusions dans une zone de surveillance.

I.5.5. La domotique :

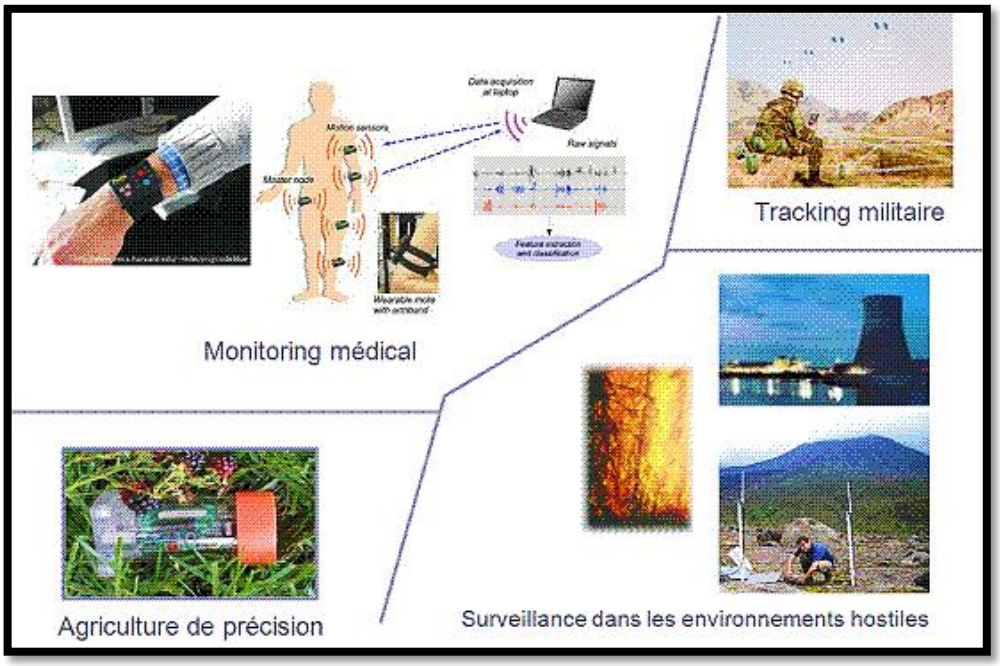
Le déploiement de capteurs de mouvement et de température dans les maisons dites intelligentes du futur permettra l'automatisation de plusieurs opérations domestiques, telles : la lumière qui s'éteint et la musique qui se met en état d'arrêt quand la chambre est vide, la climatisation et le chauffage s'ajustent selon les points multiples de mesure, le déclenchement d'une alarme par le capteur anti-intrusion quand un intrus veut accéder à la maison.

I.5.6. Applications commerciales:

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

I.5.7. Applications dans le domaine sportif :

L'évolution des réseaux de capteurs est utilisée de plus en plus dans le domaine sportif, à savoir les systèmes de surveillance, les systèmes de calcul de trajectoires (comme dans le tennis), systèmes de détection d'erreurs d'arbitrage (comme dans le football indiquent si le balle a franchi la ligne de but).



FigureI.7. *Applications des RCSF.*

I.6. Architecture des RCSF :

Un RCSF est composé d'un ensemble de noeuds capteurs, Les noeuds de ce type des réseaux consistent en un grand nombre de micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome. La position de ces noeuds n'est pas obligatoirement prédéterminée. Ils sont dispersés aléatoirement à travers une zone géographique, appelée champ de captage, qui définit le terrain d'intérêt pour le phénomène capté. Les données captées sont acheminées grâce à un routage multi-saut à un noeud considéré comme « un point de collecte », appelé noeud puits (ou sink). Ce dernier peut être connecté à l'utilisateur du réseau via Internet ou un satellite. Ainsi, l'usager peut adresser des requêtes aux autres noeuds du réseau, précisant le type de données requises et récolter les données environnementales captées par le biais du noeud puits [9] .

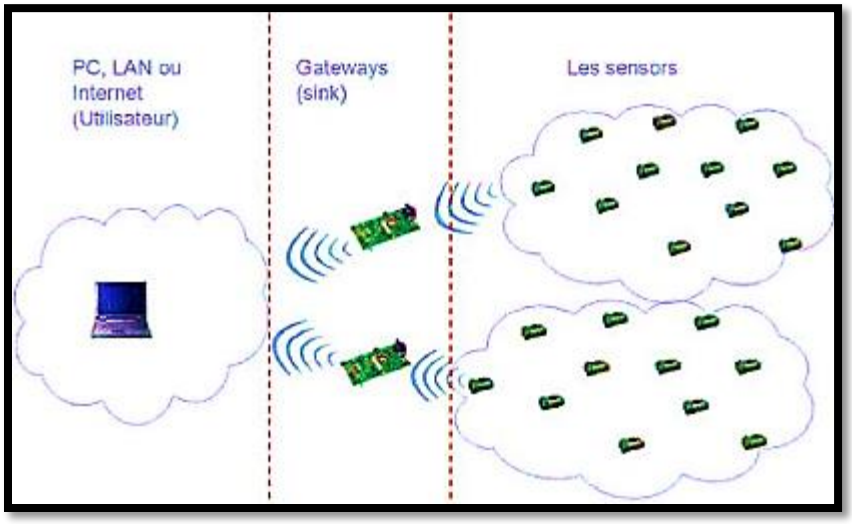


***Figure I.8**. Architecture d'un RCSF.*

I.6.1. Collecter les informations:

Il y a deux méthodes pour collecter les informations d'un réseau de capteurs :

- ***Collecter les informations à la demande :*** Lorsque l'on souhaite avoir l'état de la zone de couverture à un moment T, le puits émet des brodcasts vers toute la zone pour que les capteurs remontent leur dernier relevé vers le puits. Les informations sont alors acheminées par le biais d'une communication multi-sauts [10].

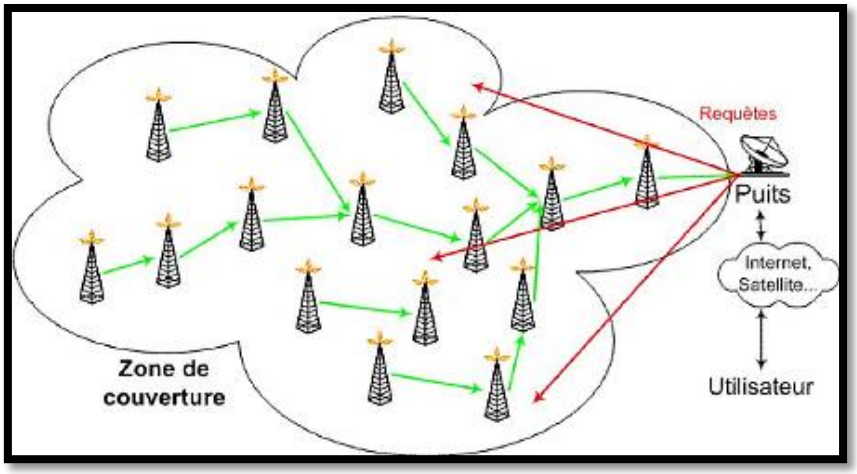


***Figure I.9.** Collecter les informations à la demande.*

- ***Collecter les informations Suite à un événement :*** Un événement se produit en un point de la zone de couverture (changement brusque de température, mouvement...), les capteurs situés à proximité remontent alors les informations relevées et les acheminent jusqu'au puits [10].

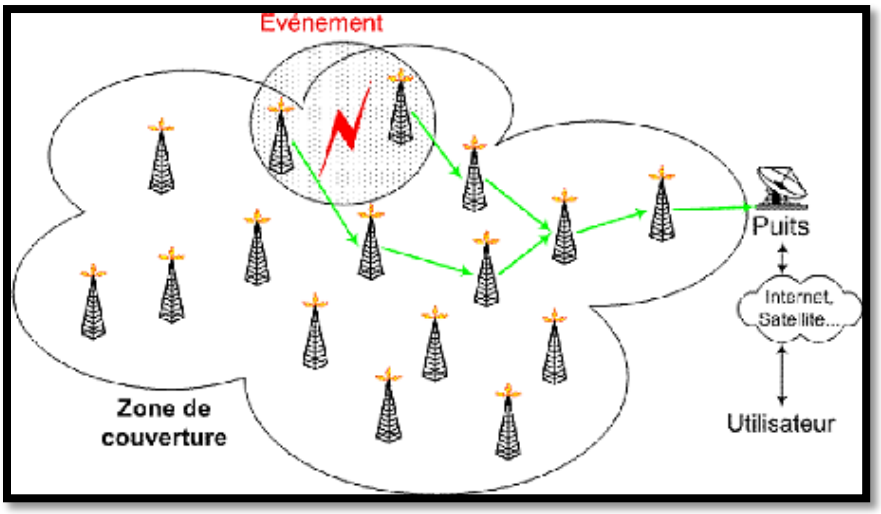


Figure I.10. *Collecter les informations Suite à un événement.*

I.6.2. Les types d'architecture des RCSF :

Il existe deux types d'architectures pour les réseaux de capteurs sans fil:

1. l'architecture **plate**
2. l'architecture **hiérarchique.**

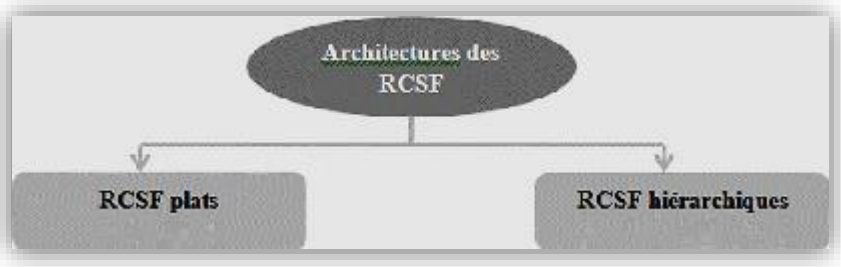


Figure I.11. *Les types d'architecture des RCSF.*

Architecture plate :

Un réseau de capteurs sans-fil plat est un réseau homogène, où tous les nœuds sont identiques en termes de batterie et de complexité du matériel, excepté le nœud puits qui joue le rôle d'une passerelle et qui est responsable de la transmission de l'information collectée à l'utilisateur final. Selon le service et le type de capteurs, une densité de capteurs élevée (plusieurs nœuds capteurs/m2) ainsi qu'une communication multi sauts peut être nécessaire pour l'architecture plate. En présence d'un très grand nombre de nœuds capteurs, le passage à l'échelle devient critique. Le routage et le contrôle d'accès au médium (MAC) doivent gérer et organiser les nœuds d'une manière très efficace en termes d'énergie [11].

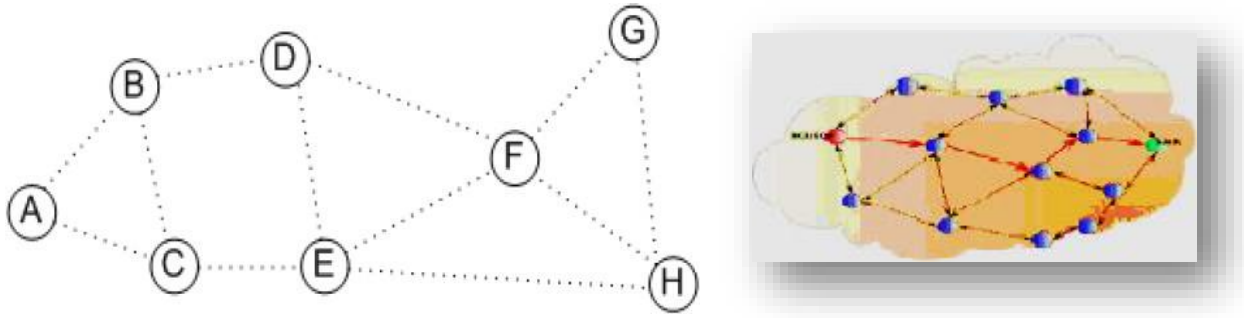


Figure I.12. *Exemple de topologie plate.*

Architecture hiérarchique :

Une architecture hiérarchique a été proposée pour réduire la complexité de la plupart des nœuds capteurs et leur déploiement, en introduisant un ensemble de nœuds capteurs plus puissants. Ceci permet de décharger la majorité des nœuds simples à faible coût de plusieurs fonctions du réseau.

L'architecture hiérarchique est composée de plusieurs couches : une couche de capteurs, une couche de transmission et une couche de point d'accès. Cette architecture sans-fil est influencée par un certain nombre de facteurs et contraintes tels que la tolérance aux fautes, le redimensionnement, les coûts de production, l'environnement, la topologie du réseau, les contraintes matérielles, les médias de transmission et la consommation d'énergie [11].

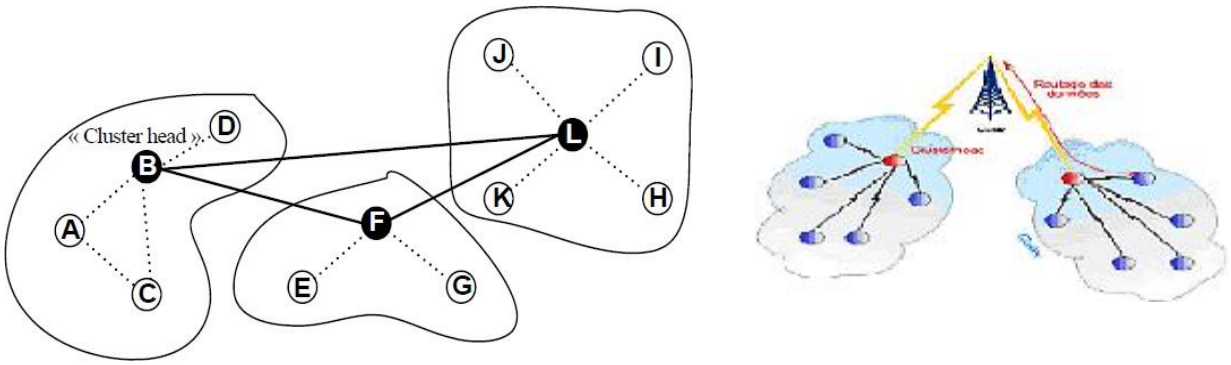


Figure I.13. *Exemple de topologie hiérarchique.*

I.7. Communication dans les RCSF :

I.7.1. Modèle en couches :

Le rôle de ce modèle consiste à standardiser la communication entre les composants du réseau afin que différents constructeurs puissent mettre au point des produits (logiciels ou matériels) compatibles. Contrairement aux réseaux traditionnels, les réseaux de capteurs utilisent une pile protocolaire de communication composée de cinq couches (une couche application, une couche transport, une couche réseau, une couche liaison de données et une couche physique), qui ont les mêmes fonctions que celles du modèle OSI ainsi que de trois niveaux ou plans intégrés dans la pile protocolaire pour la gestion de la puissance d'énergie, la gestion de la mobilité ainsi que la gestion des tâches (interrogation du réseau de capteurs).

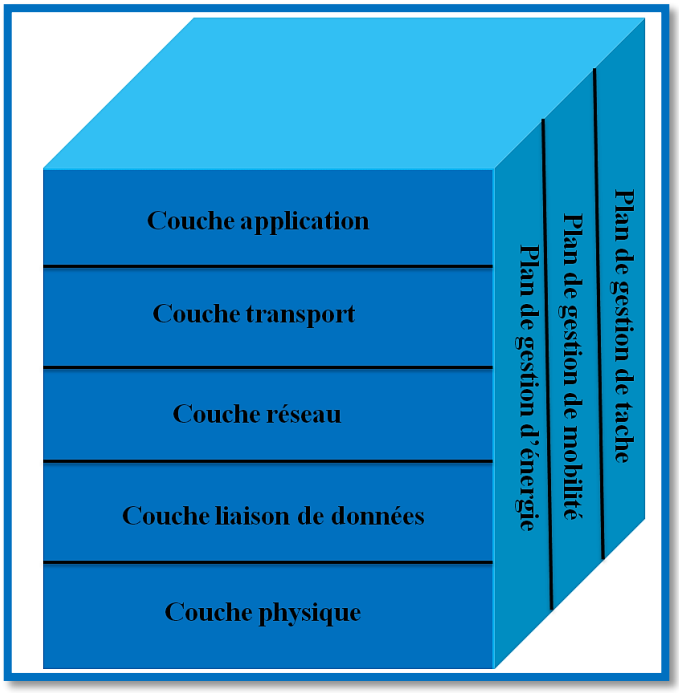


Figure I.14. *Pile protocolaire dans les réseaux de capteurs.*

Le but d'un système en couches est de séparer le problème en différentes parties (les couches) selon leur niveau d'abstraction. Chaque couche du modèle communique avec une couche adjacente (celle du dessus ou celle du dessous). Chaque couche utilise ainsi les services des couches inférieures et en fournit à celle de niveau supérieur [12]**.**

I.7.2. Rôles des couches :

Dans ce qui suit on examinera les différentes couches qui constituent la pile protocolaire et on donnera le rôle et les fonctions de chacune d'entre elles **:**

- ***Couche application :*** Elle assure l'interface avec les applications. Il s'agit donc de la couche la plus proche des utilisateurs, gérée directement par les logiciels. Parmi les protocoles d'application, nous citons : SMP (Sensor Management Protocol) et TADAP (Task Assignement and Data Advertisement Protocol).

- ***Couche transport :*** Elle vérifie le bon acheminement des données et la qualité de la transmission. Dans les RCSF, la fiabilité de transmission n'est pas majeure. Ainsi, les erreurs et les pertes sont tolérées. Par conséquent, un protocole de transport proche du protocole **UDP** et appelé **UDP-Like** (*User Datagram Protocol Like*) est utilisé. Cependant, comme le protocole de transport universel est **TCP** (*Transmission Control Protocol*), les RCSF doivent donc posséder, lors d'une communication avec un réseau externe, une interface TCP-spliting pour vérifier la compatibilité entre ces deux réseaux communicants.

- ***Couche réseau :*** Elle s'occupe du routage de données fournies par la couche transport. Elle établit les routes entre les nœuds capteurs et le nœud puits et sélectionne le meilleur chemin en termes d'énergie, délai de transmission, débit, etc. Les protocoles de routage conçus pour les RCSF sont différents de ceux conçus pour les réseaux Ad Hoc puisque les RCSF sont différents selon plusieurs critères comme :

  ▪ l'absence d'adressage fixe des nœuds tout en utilisant un adressage basé-attribut.

  ▪ l'établissement des communications multi-sauts.

  ▪ l'établissement des routes liant plusieurs sources en une seule destination pour agréger des données similaires, etc.

  ▪ Parmi ces protocoles, nous citons : **LEACH** (*Low-Energy Adaptive Clustering Hierarchy*) et **SAR** (*Sequential Assignement Routing*).

- ***Couche liaison de données :*** Elle est responsable de l'accès au media physique et la détection et la correction d'erreurs intervenues sur la couche physique. De plus, elle établit une communication saut-par-saut entre les nœuds. C'est-à-dire, elle détermine les liens de communication entre eux dans une distance d'un seul saut.Parmi les protocoles de liaison de données, nous citons: **SMACS** (*Self-organizing Medium Access Control for Sensor networks*) et **EAR** (*Eavesdrop And Registe*r).

- ***Couche physique* :** La couche physique est responsable du support acheminant les données communiquées entre les nœuds. Ainsi, il existe trois types de médias pouvant être utilisés pour les réseaux de capteurs : optique (Laser), les infrarouges et les radiofréquences. Le mode de communication par radio fréquence est le plus facile à employer et il reste le mode préféré par la plupart des projets de recherche menés sur les réseaux de capteurs, car les paquets échangés dans ces réseaux sont de petite taille et ils sont transmis à un faible débit. La possibilité de réutilisation de fréquence est également considérable en raison de la petite distance entre les nœuds. Ainsi, il est possible de résumer les tâches accomplies au niveau de la couche physique en quatre points :

  - La sélection des fréquences.
  - La génération des ondes porteuses.
  - La détection du signal.
  - La modulation.

I.7.3. Plans de gestion :

a. ***Plan de gestion de l'énergie :*** Les fonctions intégrées à ce niveau consistent à gérer l'énergie consommée par les capteurs, dès lors, un capteur peut par exemple éteindre son interface de réception dès qu'il reçoit un message d'un noeud voisin afin d'éviter la réception des messages dupliqués. De plus, quand un noeud possède un niveau d'énergie faible, il peut diffuser un message aux autres capteurs pour ne pas

participer aux tâches de routage, et conserver l'énergie restante aux fonctionnalités de captage**.**

b. ***Plan de gestion de la mobilité :*** Ce niveau détecte et enregistre tout les mouvements des noeuds capteurs, d'une manière à leur permettre de garder continuellement une route vers l'utilisateur final, et maintenir une image récente sur les noeuds voisins, cette image est nécessaire pour pouvoir équilibrer l'exécution des tâches et la consommation d'énergie.

c. ***Plan de gestion des taches :*** Lors d'une opération de capture dans une région donnée, les noeuds composant le réseau ne doivent pas obligatoirement travailler avec le même rythme. Cela dépend essentiellement de la nature du capteur, son niveau d'énergie et la région dans laquelle il a été déployé. Pour cela, le niveau de gestion des taches assure l'équilibrage et la distribution des taches sur les différents noeuds du réseau afin d'assurer un travail coopératif et efficace en matière de consommation d'énergie, et par conséquent, prolonger la durée de vie du réseau**.**

I.8. Caractéristique des RCSF :

Par ailleurs les caractéristiques essentielles des RCSF s'établissent comme suit :

- ***La consommation réduite d'énergie :*** Les nœuds capteurs utilisent des batteries de taille minuscule comme ressources en énergie, ce qui limite leur durée de vie. La spécificité des applications des RCSF (militaires, sismiques et autres) fait que la recharge ou le remplacement de ces batteries est une tâche difficile ou presque impossible, ce qui nous mène à déduire que la durée de vie d'un nœud est essentiellement dépendante de la durée de vie de la batterie. Ainsi, la méthode de gestion de consommation d'énergie constitue une contrainte majeure dans ce type de réseau.
- ***L'auto-configuration des nœuds capteurs :*** Dans un RCSF, les nœuds sont déployés soit d'une manière aléatoire (missile, avion…), soit placés nœud par nœud

par un humain ou un robot, et ceci à l'intérieur ou autour du phénomène observé (champ de guerre, surface volcanique, patient malade…). Ainsi, un nœud capteur doit avoir des capacités d'une part, pour s'auto-configurer dans le réseau, et d'autre part pour collaborer avec les autres nœuds dans le but de reconfigurer dynamiquement le réseau en cas de changement de topologie du réseau . Dans un RCSF, chaque nœud X possède une unité émettrice/réceptrice qui lui permet de communiquer avec les nœuds qui lui sont proches; En échangeant des informations avec ces derniers, le nœud X pourra alors découvrir ses nœuds voisins et ainsi connaître la méthode de routage qu'il va adopter selon les besoins de l'application. L'auto-configuration apparaît comme une caractéristique nécessaire dans le cas des RCSF étant donné que d'une part, leur déploiement s'effectue d'une manière aléatoire dans la majorité des applications, et d'autre part le nombre des nœuds capteurs est très grand.

- ***La scalabilité :*** Contrairement aux réseaux sans fil traditionnels (personnel, local ou étendu), un RCSF peut contenir un très grand nombre de nœuds capteurs (des centaines, des milliers…) . Un réseau de capteur est scalable parce qu'il a la faculté d'accepter un très grand nombre de nœuds qui collaborent ensemble afin d'atteindre un objectif commun.

- ***La tolérance aux pannes :*** Dans le cas de dysfonctionnement d'un nœud (manque d'énergie, interférences avec l'environnement d'observation…) ou aussi en cas d'ajout de nouveaux nœuds capteurs dans le réseau, ce nœud doit continuer à fonctionner normalement sans interruption. Ceci explique le fait qu'un RCSF n'adopte pas de topologie fixe mais plutôt dynamique.

- ***Une densité importante des nœuds :*** Les RCSF sont caractérisés par leur forte densité Cette densité peut atteindre, selon le type d'application, 20 nœuds/m3.

- ***La capacité de communication :*** Elle peut prendre deux aspects : Le multisaut ou à un seul saut . Parce que le multisaut est moins énergivore, il reste le type de

communication le plus sollicité par les applications de RCSF qui requièrent une faible consommation d'énergie.

- ***Les types de communication :*** Il existe différents types de communication utilisée dans les RCSF:

  - ***Unicast*** : ce type de communication est utilisé pour échanger des informations entre deux nœuds sur le réseau.
  - ***Broadcast :*** la station de base ou « Sink » transmet des informations vers tous les nœuds du réseau. Ces informations peuvent être des requêtes de données bien précises (ex : la température dans la région A), des mises à jour de programmes ou des paquets de contrôle…
  - ***Local Gossip*** : ce type de communication est utilisé par des nœuds situés dans une région bien déterminée qui collaborent ensemble afin d'avoir une meilleure estimation de l'évènement observé et d'éviter l'émission du même message vers le nœud « Sink » ce qui contribue à consommer moins d'énergie.
  - ***Convergecast :*** il est utilisé dans les communications entre un groupe de nœuds et un nœud bien spécifique (qui peut être le « Sink »). L'avantage de ce type de communication est la diminution de contrôle d'entête des paquets (« control overhead ») ce qui économise l'énergie au niveau du nœud récepteur .
  - ***Multicast :*** il permet une communication entre un nœud et un groupe de nœuds. Ce type de communication est utilisé dans les protocoles qui incluent le « clustering » dans lesquels, le « Clusterhead » s'intéresse à communiquer avec un groupe de nœuds.

- ***Une collaboration entre les nœuds :*** Les contraintes strictes de consommation d'énergie mènent les nœuds capteurs à détecter et traiter les données d'une manière coopérative afin d'éviter le traitement redondant d'une même donnée observée, source de la perte d'énergie.
- ***La bande passante (ou capacité du canal***) : c'est une caractéristique beaucoup plus importante dans les réseaux cellulaires (GSM) et les réseaux locaux sans fils

(WLAN), que dans les RCSF ; le débit étant en effet un objectif secondaire pour les RCSF.

I.9. Les limites des RCSF :

La localisation dans les réseaux de capteurs comporte certaines limites et ceci causés par l'absence de dispositif d'auto positionnement. Les contraintes de conception étant nombreuses nous citerons les principales telles que reprisent par les grandes théories :

- Assurer une continuité du réseau sans disfonctionnement. .
- Une mémoire capable de stoker toutes les informations reçues. .
- Maitrise des couts de production. .
- Surmonter les défaillances liées à l'environnement. .
- Un déploiement topologique grâce à une maintenance assidue.
- Cernement du facteur matériel telles que la taille des capteurs le gain d'énergie, l'adaptation aux milieux et sa résistance.
- Mise en place d'une norme de transmission telle que l'infrarouge, le Bluetooth et les communications radio ZigBee.
- Une adéquation entre l'utilisation et la consommation de l'énergie capable d'assurer une longévité du réseau.

I.10. Systèmes d'exploitation et technologies utilisées dans les RCSF :

Plusieurs systèmes d'exploitation ont été développés pour répondre aux contraintes particulières des réseaux de capteurs, le système plus connu est : ***TinyOS***.

a. **TinyOS :** est un système d'exploitation open source conçu pour les capteurs sans fils et développé par l'Université de Berkeley. Il est basé sur une architecture à base de modules : pilotes pour les capteurs, les protocoles réseau et les services distribués. Les composants sont programmés en *NesC*, un langage de programmation dérivé du C adapté aux faibles ressources physiques des capteurs. Un certain nombre de plateformes sont directement programmables comme par exemple : les Tmote ou les MicaZ (ces deux modèles sont

compatibles avec ZigBee). TOSSIM est un simulateur de capteurs pour les programmes TinyOS, tout programme en NesC peut être compilé de manière à être exécuté dans TOSSIM, ce qui permet de simuler le comportement d'un ou plusieurs capteurs ainsi de les programmer. [13]

b. Bluetooth (***IEEE 802.15.4***) **:** Bluetooth est une spécification de l'industrie des télécommunications. Elle utilise une technique radio courte distance destinée à simplifier les connexions entre les appareils électroniques. Malheureusement, un grand défaut de cette technologie est sa trop grande consommation d'énergie.

c. ZigBee : Beaucoup moins connue que Bluetooth, c'est une norme de transmission de données sans fil permettant la communication de machine à machine. Sa très faible consommation électrique et ses coûts de production très bas en font une candidate idéale pour la domotique ou le matériel de type capteur, télécommande ou équipement de contrôle dans le secteur industriel. Les débits autorisés sont relativement faibles, entre 20 et 250 Kbits/s, mais c'est véritablement sa très faible consommation électrique qui en fait son atout principal. ZigBee fonctionne sur la bande de fréquences des 2,4 GHz et sur 16 canaux, sa portée était au début d'une dizaine de mètres, elle est désormais de 100 mètres.

I.11. Conclusion :

Ce chapitre propose une présentation complète des réseaux de capteurs sans-fil, en mettant en évidence leur diversité et leur mode de fonctionnement, ainsi que leurs caractéristiques et spécificités. Des informations détaillées sur les capteurs et leurs domaines d'application ont également été fournies tout au long du chapitre.

CHAPITRE II :

LA SÉCURITÉ DANS LES RÉSEAUX DE CAPTEURS SANS FIL.

II.1. Introduction :

Les réseaux de capteurs sans fil (RCSF) sont devenus omniprésents dans de nombreux domaines, tels que la surveillance environnementale, les systèmes de santé, les infrastructures intelligentes et l'Internet des objets (IoT). Ces réseaux se composent de nombreux capteurs sans fil autonomes, déployés dans un environnement spécifique, qui collaborent pour collecter, traiter et transmettre des données vers une station de base ou un nœud central.

Alors que les RCSF offrent d'énormes possibilités en termes de collecte de données en temps réel et de surveillance de l'environnement, ils sont également confrontés à des défis de sécurité considérables. En effet, ces réseaux sont généralement déployés dans des environnements ouverts et potentiellement hostiles, ce qui les expose à diverses menaces et attaques malveillantes.

Ce chapitre se concentre sur l'importance de la sécurité dans les RCSF et examine en détail les différentes dimensions de la sécurité auxquelles ces réseaux sont confrontés. Nous aborderons les vulnérabilités spécifiques des RCSF, les attaques courantes auxquelles ils sont exposés et les contre-mesures utilisées pour prévenir ou atténuer ces attaques. Nous discuterons également des protocoles de sécurité et des mécanismes de chiffrement utilisés pour protéger les données et assurer l'intégrité et la confidentialité des communications.

II.2. Conditions De Sécurité :

Les réseaux de capteurs sans fil (RCSF) sont des réseaux particuliers qui présentent certaines similitudes avec les réseaux informatiques classiques, mais qui ont également des caractéristiques uniques qui nécessitent des exigences spécifiques en matière de sécurité. Ainsi, pour assurer une sécurité adéquate, un protocole de sécurité pour un RCSF doit répondre à une ou plusieurs des conditions de sécurité suivantes [14] :

II.2.1. Confidentialité Des Données :

La préservation de la confidentialité des données représente une préoccupation majeure en matière de sécurité de réseau. Pour protéger le transfert de données, l'approche généralement adoptée consiste à utiliser un algorithme de chiffrement pour coder les données à l'aide d'une clé secrète connue uniquement par l'émetteur et le récepteur.

II.2.2. Intégrité des données :

Il est possible qu'un nœud malveillant (appelé "adversaire") altère les données transmises dans le réseau de capteurs sans fil. Par exemple, cet intrus pourrait ajouter ou modifier certains éléments d'un paquet de données avant de l'envoyer au récepteur prévu. Dans certains cas, même sans la présence d'un nœud malveillant, les données peuvent être perdues ou altérées en raison des conditions difficiles de transmission. Pour cette raison, la préservation de l'intégrité des données est essentielle pour garantir que les données reçues n'ont pas été altérées pendant leur transfert.

II.2.3. Fraîcheur Des Données :

Assurer la confidentialité et l'intégrité des données ne suffit pas à garantir la sécurité du réseau de capteurs sans fil. Il est également essentiel de s'assurer que chaque message est récent et que les anciens messages ne peuvent pas être rejoués. Cette condition est particulièrement importante lorsque des clés de partage sont utilisées dans la conception, car les clés partagées doivent être modifiées régulièrement.

Cependant, le processus de propagation de nouvelles clés peut prendre du temps, ce qui crée une vulnérabilité potentielle pour les attaques de rejouer. Pour résoudre ce problème, un compteur de temps peut être ajouté dans le paquet pour garantir la fraîcheur des données.

II.2.4. Auto-Organisation :

Les réseaux de capteurs sans fil sont souvent des réseaux ad hoc, où chaque nœud capteur doit être autonome et suffisamment flexible pour s'auto-organiser, car il n'y a pas d'infrastructure fixe pour gérer le réseau. Cette auto-organisation représente un défi majeur pour la sécurité du réseau de capteurs sans fil [14].

II.2.5. La Localisation :

La capacité d'un réseau de capteurs à localiser automatiquement chaque capteur dans le réseau est souvent essentielle à son utilité. Par exemple, un réseau de capteurs conçu pour détecter des anomalies aura besoin d'informations précises sur l'emplacement de chaque capteur afin d'indiquer l'endroit exact d'un défaut.

II.2.6. Authentification :

Un attaquant peut aller au-delà de la simple modification de données et injecter des paquets entiers. Cela signifie que le destinataire doit s'assurer que les données qu'il utilise pour toute décision proviennent de la source appropriée

En outre, l'authentification est nécessaire pour de nombreuses tâches administratives telles que la reprogrammation ou le contrôle du réseau de capteurs. Il est donc clair que l'authentification des données est importante pour de nombreuses applications dans ces réseaux. En général, cela permet au destinataire de vérifier que les données sont effectivement envoyées par l'expéditeur prétendu.

Dans le cas d'une communication bidirectionnelle, l'authentification des données peut être réalisée à l'aide d'un mécanisme purement symétrique, où l'expéditeur et le destinataire partagent une clé secrète pour calculer le code d'authentification de message (IMPER) pour toutes les données transmises. [14] [15].

II.3. Les obstacles de sécurité aux réseaux de capteurs :

II.3.1. Des ressources limitées :

La contrainte énergétique est l'un des principaux défis dans la conception d'un réseau de capteurs sans fil. Chaque nœud doit gérer sa réserve d'énergie pour prolonger la durée de vie de l'ensemble du réseau. Étant donné que les capteurs sont généralement co-localisés, l'information

transmise peut être redondante. Ainsi, l'agrégation de données est une technique efficace pour économiser de l'énergie.

Cependant, cela nécessite une détection précise des fausses données ou des modifications de données défectueuses lors des opérations d'agrégation effectuées par les nœuds intermédiaires.

II.3.2. Communication non fiable :

Certainement, la communication est un autre obstacle pour la sécurité des capteurs :

- ***Transfert non fiable***

  Les paquets peuvent être endommagés en raison des erreurs de transmission ou supprimés dans les nœuds fortement encombrés. D'une manière primordiale, le protocole doit disposer d'une gestion d'erreur appropriée sinon il serait possible de perdre des paquets critiques de sécurité tels que les paquets contenant les clés cryptographiques.

- ***Collisions***

  Même si le canal est fiable, la communication ne peut pas toujours l'être. Ceci est dû à la nature d'émission des paquets dans les réseaux de capteurs sans fil (broadcast).

  Si les paquets se rencontrent lors du transfert, les collisions se produisent et le transfert lui même échouera.

II.3.3. Couplage étroit avec l'environnement :

Les applications des réseaux de capteurs sans fil (RCSF) exigent souvent un déploiement étroit des nœuds à l'intérieur ou à proximité des phénomènes à surveiller, ce qui augmente la probabilité de compromissions intentionnelles ou accidentelles des nœuds. Compte tenu du faible coût des nSuds, ils ne peuvent pas offrir une protection physique inviolable. Les attaquants bien équipés peuvent extraire des informations cryptographiques des nœuds capteurs, ce qui représente un risque important pour les missions de RCSF généralement sans surveillance.

Par conséquent, il est crucial de gérer les clés cryptographiques et les informations sensibles de manière à augmenter la résistance à la capture des nœuds.

La figure (II.1), résume les problèmes de sécurité émergeant des caractéristiques d'un RCSF et les solutions à entreprendre :

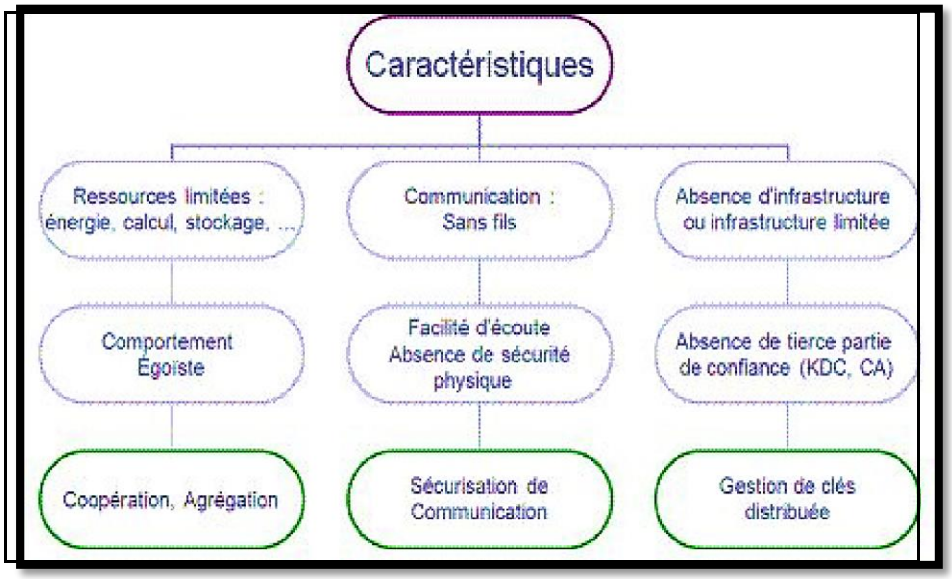


Figure II.1. *Sécurité dans les RCSF : propriétés, challenges et solutions* [16].

II.4. Blocs fonctionnels de la sécurité dans les RCSF :

Comme illustré à la figure **(**II.2) , on distingue quatre blocks fonctionnels des solutions de sécurité dans les RCSF : la gestion de clés, la sécurité du routage, la sécurité de l'agrégation de données, et la sécurité de l'accès au canal.

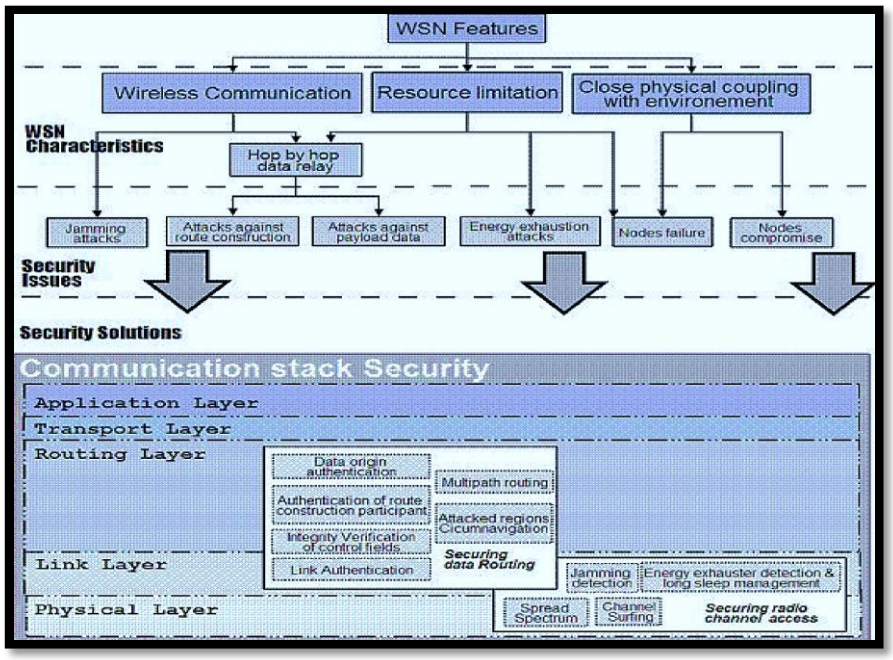


Figure II.2. *Taxonomie des challenges et solutions de sécurité dans les RCSF* [16].

II.5. Mécanismes de sécurité :

II.5.1. Définition de la cryptographie :

Le mot « cryptographie » est composé des mots grecques: ***« crypto »*** signifie caché, ***«graphy»*** signifie écrire. C'est donc l'art de l'écriture secrète [17].

La cryptographie permet de protéger des informations en les transformant de manière à ce qu'elles soient illisibles pour des personnes non autorisées, appelées des "adversaires". Le but de la cryptographie est de permettre la communication sécurisée et confidentielle entre deux parties même si un tiers non autorisé est en mesure d'intercepter ou de modifier les messages échangés.

La cryptographie utilise des algorithmes mathématiques pour chiffrer les données en utilisant des clés de cryptage. Le processus de chiffrement transforme les données en un format illisible, appelé texte chiffré, tandis que le processus de déchiffrement utilise une clé de déchiffrement pour retransformer les données en format lisible. La cryptographie est réalisée selon certains outils. Avant de les aborder, il est commode de définir la notion de clé qui sera utilisée tout au long de cette partie.

- ***Une clé :*** Dans la cryptographie moderne, l'habilité de maintenir un message crypté secret, repose non pas sur les algorithmes, mais sur une information secrète dite clé qui est un paramètre utilisé en entrée d'une opération cryptographique et qui doit être utilisée avec les algorithmes pour produire le message crypté. [17] [18].

II.5.2. Les outils cryptographiques :

II.5.2.1. Le chiffrement : Le chiffrement est une méthode de cryptographie qui assure la confidentialité en utilisant des clés. Selon l'utilisation de ces clés, il existe deux types de techniques de chiffrement : symétrique et asymétrique.

- **Le chiffrement symétrique :** Une même clé est utilisée entre deux nœuds communicants pour chiffrer et déchiffrer les données en utilisant un algorithme de chiffrement symétrique. Les algorithmes de chiffrement symétriques sont décomposés en deux catégories :
- **Le chiffrement en chaîne :** est fait bit à bit sans attendre la réception entière des données. L'algorithme le plus connu est : RC4 (Rivest Cipher 4) [19].
- **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 a taille envisagée. Les algorithmes les plus utilisés sont: DES (Data Encryption Standard), AES (Advanced Encryption Standard) [20].

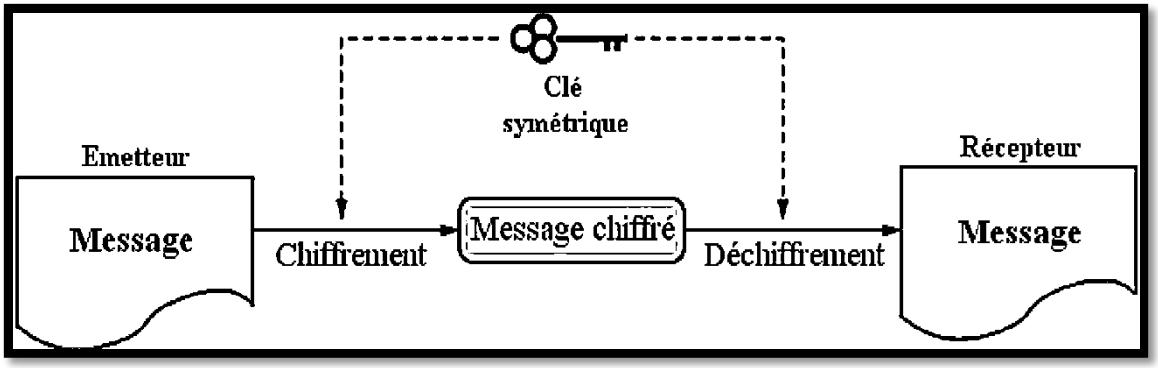


Figure II.3. *Le chiffrement symétrique* [21].

Les algorithmes de chiffrement symétrique tels que RC4 sont connus pour leur rapidité et leur efficacité. Cependant, la construction et la distribution des clés peuvent causer une grande dépense d'énergie. En effet, dans un système symétrique, chaque nœud doit avoir une clé partagée avec tous les autres nœuds du réseau. Ainsi, si le réseau compte n nœuds, il faudra gérer n*(n-1)/2 clés [17].

Malgré cela, les algorithmes de chiffrement symétrique sont souvent les plus adaptés aux applications des RCSF car ils ne nécessitent pas d'opérations mathématiques complexes pour crypter ou décrypter les données. Par conséquent, ils n'exigent pas une grande consommation d'énergie pendant les phases de chiffrement et de déchiffrement.

- **Le chiffrement asymétrique :** Dans le chiffrement asymétrique, le récepteur crée deux clés différentes : une clé publique qui est distribuée à tous les nœuds pour chiffrer les données qu'ils souhaitent envoyer au récepteur, et une clé privée qui reste secrète et est utilisée par le récepteur pour déchiffrer les données qu'il reçoit. La sécurité du chiffrement asymétrique repose sur l'impossibilité de déduire la clé privée à partir de la clé publique. RSA (Rivest Shamir Adleman) est l'algorithme de chiffrement asymétrique le plus connu [20].

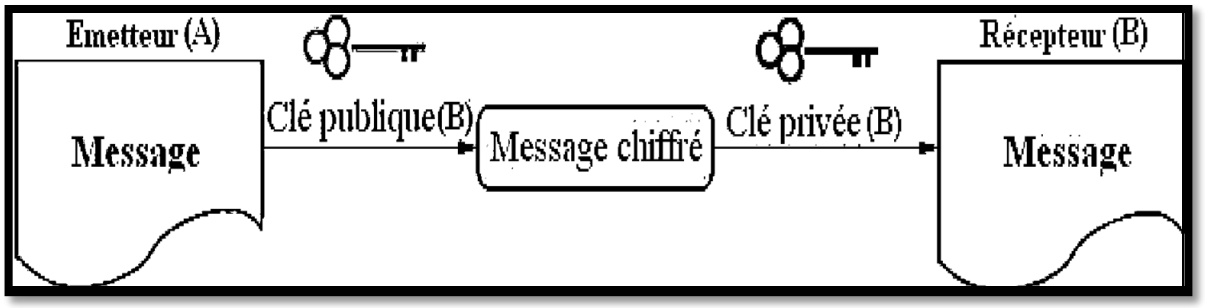


Figure II.4. *Le chiffrement asymétrique* [21].

Les algorithmes de chiffrement asymétrique peuvent être confrontés à des problèmes mathématiques complexes lors du déchiffrement des données, ce qui peut entraîner une surcharge de traitement et une consommation d'énergie plus importante pour les nœuds capteurs. De plus, le stockage des clés publiques de tous les autres nœuds du réseau peut entraîner une occupation importante de la mémoire de chaque nœud [18].

Cependant, la distribution des clés est simplifiée car chaque nœud ne nécessite qu'une paire de clés. Néanmoins, pour un réseau de nœuds, cela signifie la gestion de 2n clés [17].

Malgré ses avantages, le chiffrement asymétrique n'est pas adapté aux RCSF en raison de sa lenteur d'exécution et de son coût élevé en termes de ressources.

L'utilisation du chiffrement symétrique dans les RCSF est également problématique en raison de la difficulté à établir des clés entre les nœuds de manière efficace.

II.5.2.2. La signature digitale :

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

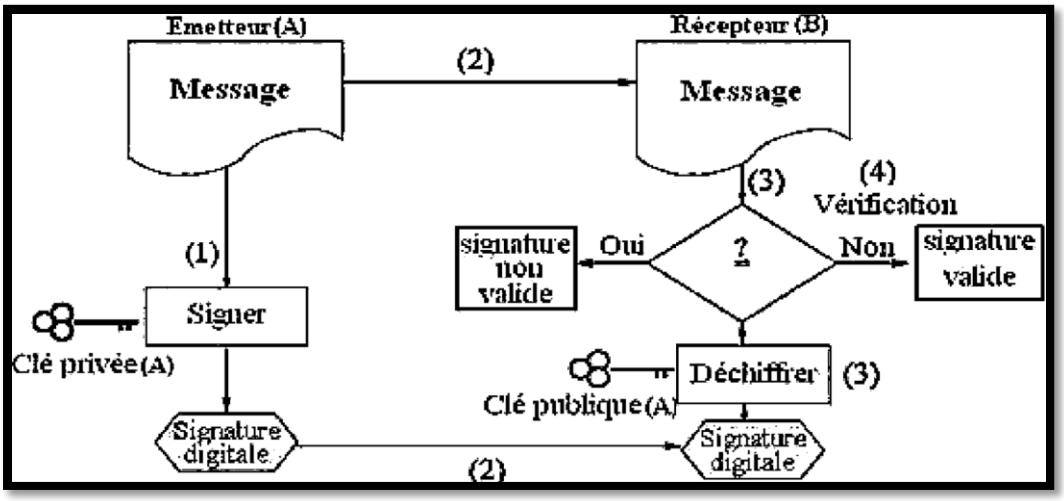


Figure II.5. *La signature digitale* [21].

II.5.2.3. La fonction de hachage :

C'est le mécanisme qui assure l'intégrité de données. Cette fonction calcule une courte empreinte de taille fixe à partir d'une donnée de taille arbitraire (1). Etant donnée une fonction de hachage f, et un message à transmettre m. La fonction f doit remplir ces conditions [17] :

- 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. Les fonctions de hachage les plus courantes sont: **MD5** (Message Digest 5), **SHA-1** (Secure Hash Algorithm) [22].

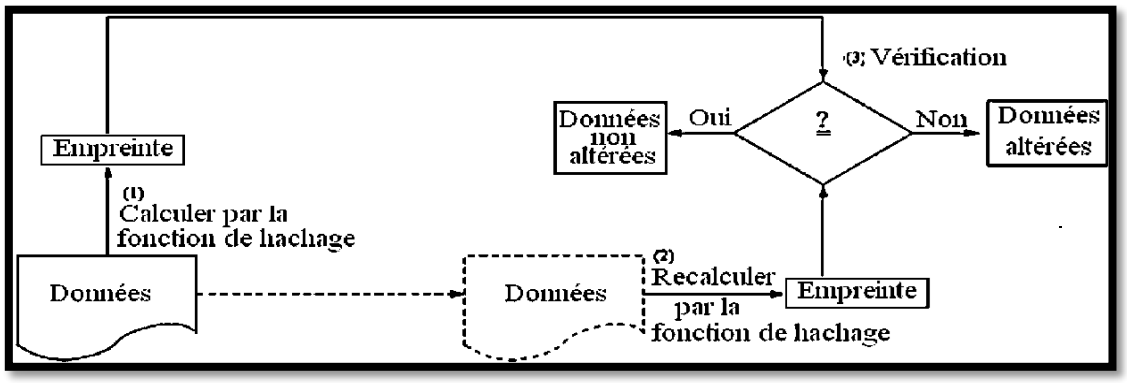


Figure II.6. *La fonction de hachage*[21].

II.5.2.4. Le code d'authentification de message MAC :

Le code d'authentification de message **MAC** (Message Authentication Code) 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 [23].

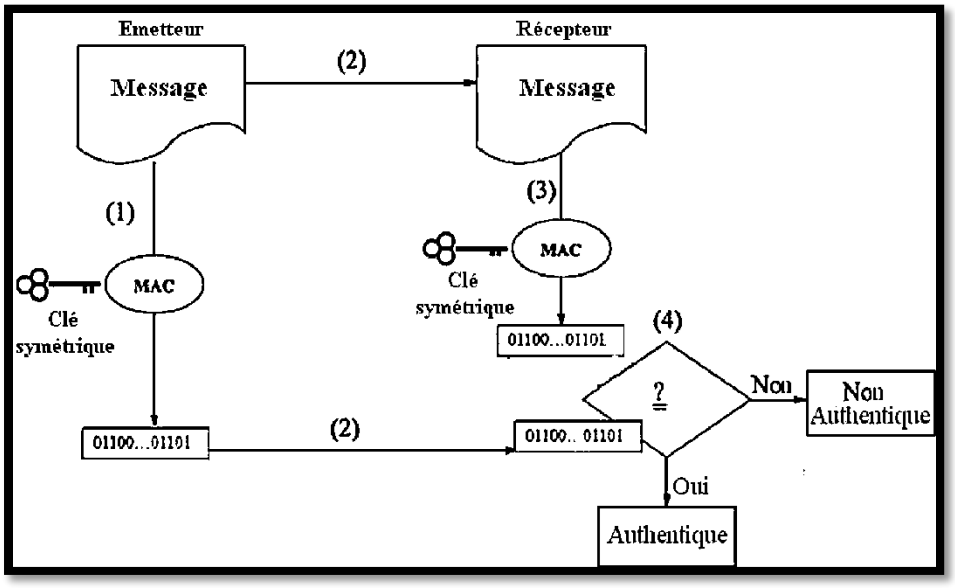


Figure II.7. *Le code d'authentification de message MAC* [21].

II.6. La gestion des clés dans les RCSF:

La gestion de clés est un service crucial pour la sécurité de tout système de communication. Elle offre des mécanismes efficaces, sûrs et stables pour la gestion des clés utilisées dans les opérations cryptographiques. Cependant, dans le contexte des RCSF, la conception d'un système de gestion de clés est particulièrement difficile en raison des contraintes spécifiques de ces réseaux. En outre, le choix d'une solution cryptographique adaptée pour les RCSF représente un autre défi important.

II.6.1. La fonction de gestion des clés dans les RCSF :

II.6.1.1. Définition :

La gestion des clés est un élément crucial de tout système de sécurité cryptographique. Pour que le système fonctionne de manière sécurisée, chaque utilisateur doit disposer d'un ensemble de clés secrètes (dans un système à clés secrètes) ou d'une paire de clés publiques/privées (dans un système à clés publiques). Cela implique de générer les clés de manière sûre et de les distribuer aux utilisateurs de manière sécurisée, ou de fournir à l'utilisateur le moyen de les générer.

En outre, il est essentiel de pouvoir stocker et gérer ces clés publiques et privées de manière sûre. Dans les systèmes à clés publiques, la gestion des clés comprend également la capacité à vérifier et à gérer les clés publiques des autres utilisateurs, qui sont souvent présentées sous forme de certificats numériques signés. En résumé, la gestion des clés est un processus complexe qui nécessite une attention particulière pour assurer la sécurité de l'ensemble du système cryptographique.

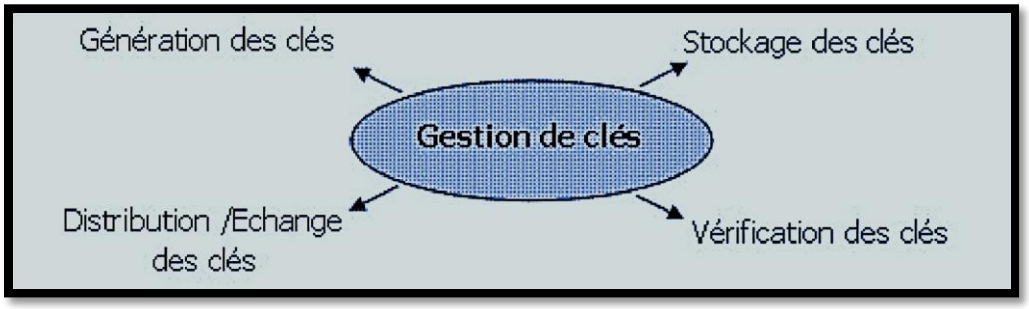


Figure II.8. *Fonctions de la gestion des clés .*

II.6.1.2. Pourquoi la gestion des clés dans les RCSF ? :

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

- Sécuriser le routage
- Sécuriser l'agrégation
- Coopération (authentification), etc.

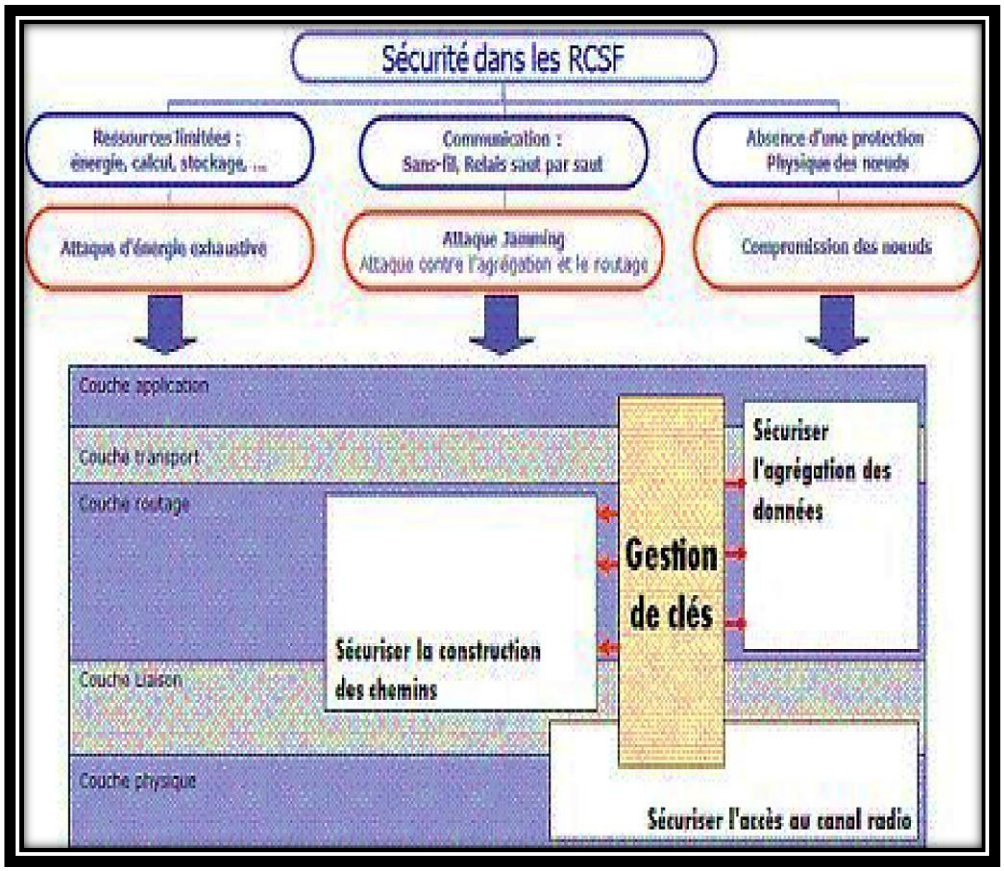


Figure II.9. *Positionnement de la gestion de clés dans un RCSF sécurisé.*

II.6.1.3. Contraintes de conception :

La figure (II.10), présente un récapitulatif des contraintes qui découlent des propriétés des RCSF et qui doivent être prises en compte lors de la conception d'une solution de gestion de clés pour les RCSF.

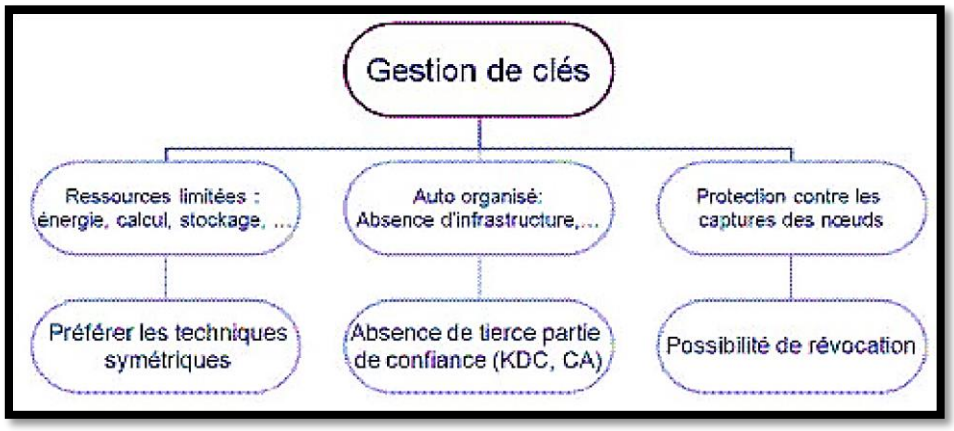


Figure II.10. *Contraintes de conception de solutions de gestion de clés.*

II.6.1.4. Systèmes asymétriques ou symétriques ? :

Dans les systèmes à clés publiques, la procédure d'échange de clés est simplifiée. Chaque partie impliquée dans la communication publie sa clé publique. Les clés publiques sont généralement distribuées à l'aide de certificats numériques, que le destinataire utilise pour authentifier la clé publique reçue. Toutes les communications avec cette partie sont ensuite chiffrées à l'aide de cette clé. L'avantage principal des algorithmes à clés publiques est la facilité de gestion des clés et leur fiabilité.

Cependant, cette approche présente certains inconvénients. Elle entraîne une consommation d'énergie accrue en raison des calculs liés aux algorithmes à clés publiques, ainsi qu'une consommation d'énergie due à la transmission des certificats. De plus, les clés publiques sont généralement plus grandes que les clés symétriques, ce qui nécessite un espace de stockage plus important.

L'utilisation de mécanismes de clés symétriques pour établir la confiance permet de réduire considérablement la consommation d'énergie des nœuds capteurs et l'espace de stockage nécessaire pour ces clés. Cependant, l'échange de clés dans les systèmes à clés symétriques est plus complexe. Habituellement, une seule clé symétrique est utilisée entre deux parties pour une session unique ou une période limitée.

Bien que la cryptographie à clé publique présente certains avantages par rapport à la cryptographie à clé symétrique, et malgré les recherches visant à l'appliquer aux RCSF, la cryptographie à clé symétrique possède ses propres qualités qui la rendent toujours préférée dans le contexte des RCSF. C'est pourquoi la plupart des schémas de gestion de clés proposés pour les RCSF sont basés sur la cryptographie symétrique.

Le principal défi de la cryptographie symétrique est de trouver une méthode qui facilite l'établissement des clés entre les nœuds. La solution couramment utilisée est l'utilisation d'une méthode de pré-distribution, où les clés sont chargées dans les nœuds capteurs avant leur déploiement.

La figure (II.11), illustre une taxonomie des solutions de gestion de clés basée sur la pré-distribution. Dans cette taxonomie, les protocoles sont classés selon la façon avec laquelle les nœuds voisins partagent des clés communes (probabiliste ou déterministe), et selon la topologie du réseau (hiérarchique ou plate).

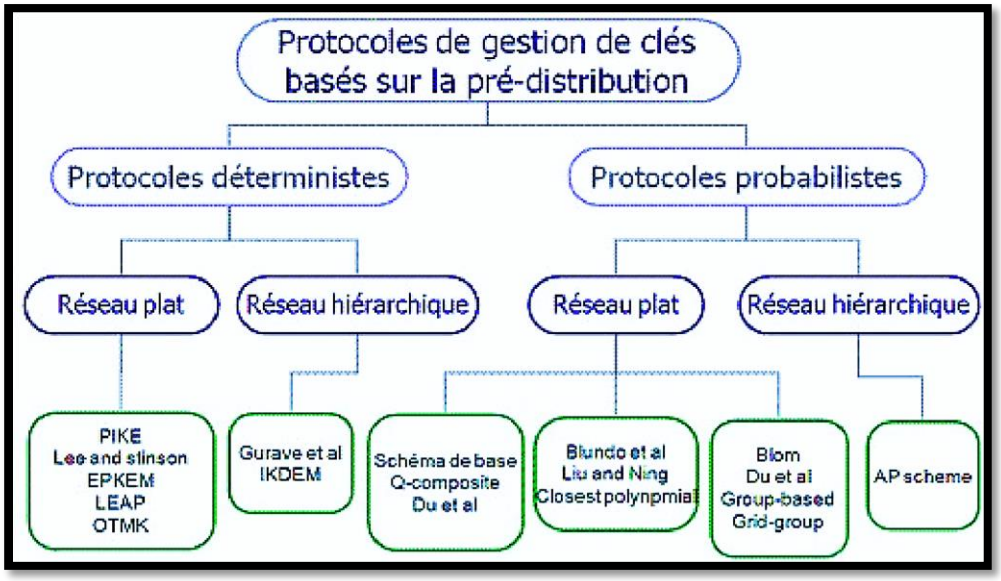


Figure II.11. *Taxonomie de pré-distribution de clés pour les RCSF.*

II.6.2. Schéma aléatoire de pré-distribution des clés de L.ESCHENAUER et D.GLIGOR :

Eschenauer et Gligor ont développé un schéma de gestion de clés pour les réseaux de capteurs sans fil, basé sur la probabilité de partage de clés entre les nœuds d'un graphe aléatoire. Ce schéma comprend des techniques de pré-distribution de clés, de découverte de clés partagées, d'établissement de chemins de clés et de révocation de clés.

L'idée principale de ce schéma est de distribuer de manière aléatoire un ensemble fini de clés à chaque nœud du réseau avant son déploiement. Ainsi, deux nœuds peuvent échanger des messages sécurisés s'ils possèdent une clé commune [37].

II.6.2.1. Phase de pré-distribution des clés :

Un grand ensemble S de clés est générée (217-220 Clés). Pour chaque nœud, m clés sont choisies au hasard de l'ensemble S (S= {(kid1, key1), (kid2, key2),....}). Ces m clés sont stockées dans la mémoire du nœud et forment le trousseau de clés du nœud. Le nombre de clés |S| de l'ensemble est choisi de telle manière que deux sous-ensembles aléatoires de S de taille m auront une certaine probabilité p d'avoir au moins une clé en commun, par exemple pour une probabilité p=0.5 on a besoin d'un sous ensemble de taille m=75 clés de l'ensemble S de taille |S|=10,000 clés.

II.6.2.2. Phase de découverte des clés partagées :

Les nœuds découvrent leurs voisins et plus particulièrement ceux avec qui ils sont en mesure de communiquer de façon sécurisée car ils possèdent une clé identique dans leur trousseau de clés respectif. Le protocole peut être de diffuser la liste des identités kidi des clés possédées. La clé partagée devient la clé de session du lien entre les deux nœuds. La figure (II.12), illustre cette phase :

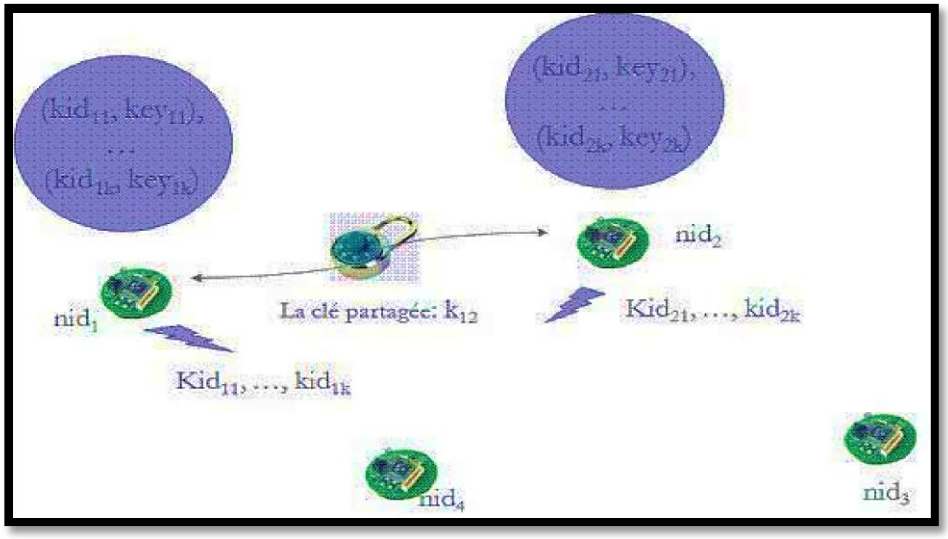


Figure II.12. *Découverte des clés partagées.*

II.6.2.3. Phase d'établissement de chemin des clés :

Après la phase de découverte des clés partagées, le réseau devient un graphe connecté formé de quelques liens sécurisés. Les nœuds peuvent alors utiliser les liens existants pour mettre en place des clés partagées avec leurs voisins qui ne partageaient pas de clé en commun avec eux. La figure (II.13), illustre cette phase :

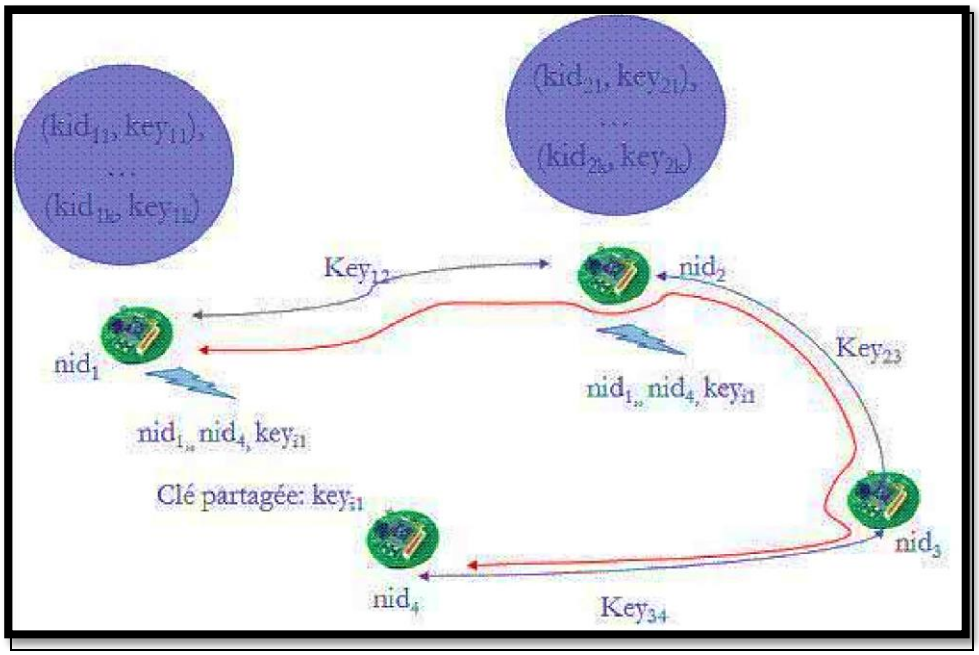


Figure II.13. *Etablissement de chemins sécurisés.*

II.6.2.4. La révocation des clés :

La révocation d'un nœud compromis se fait par l'élimination de leur trousseau des clés. Pour cela, un nœud contrôleur (qui a une grande connectivité et peut être mobile) annonce un message simple de révocation contenant une liste signée de k identificateurs des clés (kidi) pour que ces clés soient retirées des trousseaux de clés des autres nœuds.

La liste des identités est signée par une clé de signature Ke générée par le nœud contrôleur et envoyée en unicast à chaque nœud i en la chiffrant avec la clé Kci (la clé Kci est partagée entre le contrôleur et le ième nœud pendant la phase de pré- distribution de clés). Quelques liens seront disparus à cause de la suppression de clés du nœud compromis ce qui nécessite une reconfiguration

de ces liens (par la découverte de clés partagées ou l'établissement de chemin de clé). La figure (II.14) illustre cette phase :

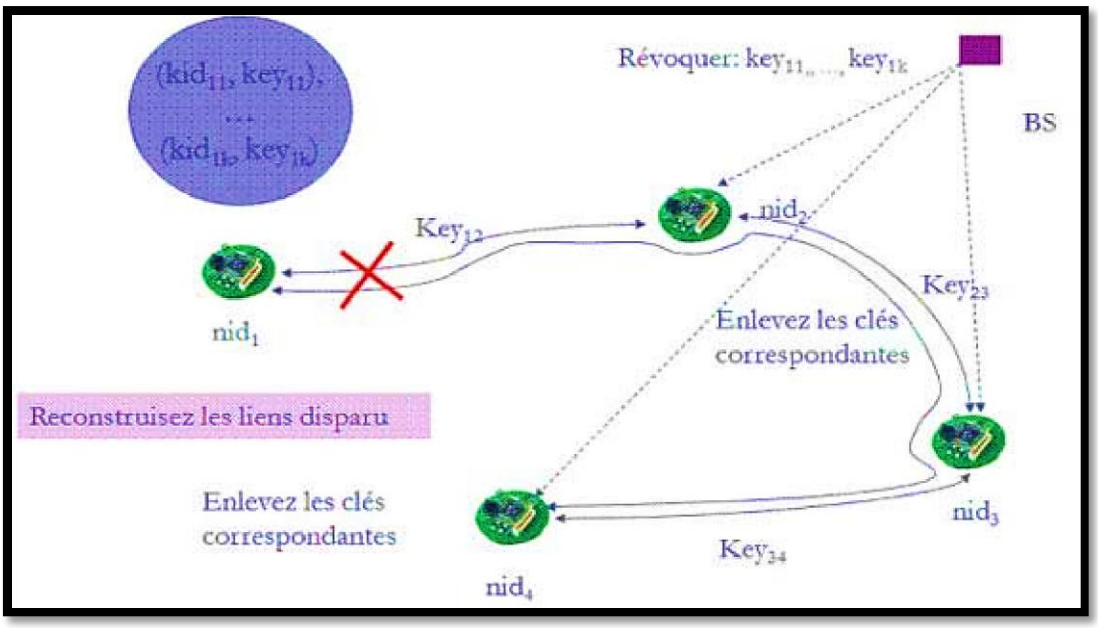


Figure II.14. *Révocation des clés.*

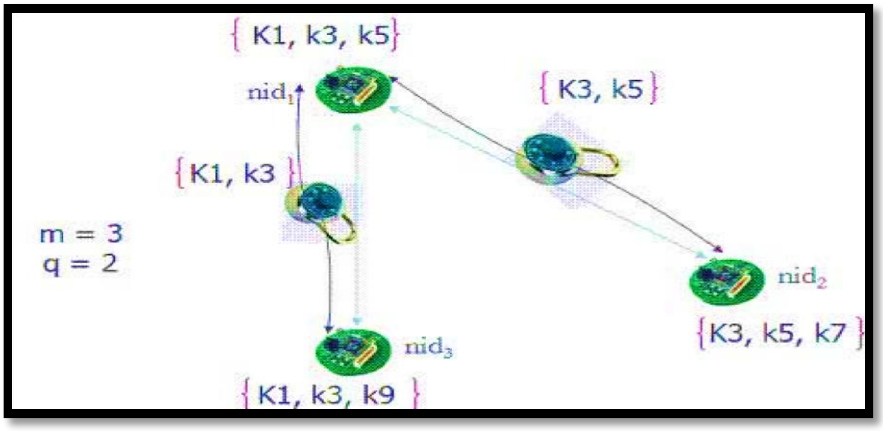
II.6.2.5. Schéma q-composite de H.CHAN , A.PERRIG et D.SONG :

Ce schéma est identique à celui de Eschenaur et Gligor [37], sauf qu'au lieu d'exiger le partage d'une clé commune pour sécuriser un lien, une paire de nœud doit partager q clés avec q>1 pour établir un lien sécurisé. La nouvelle clé utilisée pour la communication entre ces deux nœuds est le hash de toutes les clés partagées, par exemple pour deux nœuds quelconque qui partage q' clés ($q' \geq q$) la clé utilisée pour la communication est K = hash (k1 || k2 || ... || kq'). Plus le nombre de clé partagées augmente plus la résilience contre la capture du nœud augmente [38].

Autrement, lorsque le nombre, exigé, de clés partagées augmente, il devient plus difficile à un attaquant avec un ensemble donné de clés de casser un lien.

Cependant, pour préserver une probabilité donnée p que deux nœuds partageant des clés suffisantes pour établir un lien sécurisé, il est nécessaire de réduire la taille de l'ensemble de clés S. Ceci permet à un attaquant de gagner un plus grand échantillon de S en cassant peu de nœuds.

La figure (II.15), illustre un exemple de partage de clés avec q=2.



FigureII.15. *Schéma q-composite.*

II.6.3. LEAP :

LEAP [39], est un protocole de gestion de clés pour les réseaux de capteurs sans fil qui est conçu pour être déterministe et sécurisé. Le système de gestion de clés LEAP prend en charge le traitement en réseau interne « in-network processing » tout en réduisant l'impact de sécurité d'un nœud compromis sur ses nœuds voisins dans le réseau. LEAP permet l'établissement de quatre types de clés pour chaque nœud du réseau, notamment *la clé individuelle, la clé par paire, la clé de groupe et la clé globale.*

II.6.3.1. Hypothèse de fonctionnement :

LEAP utilise une clé initiale transitoire KIN pour chaque nœud. Les auteurs de LEAP supposent qu'un adversaire nécessite un temps minimal (Tmin) pour compromettre un nœud, qui inclut le temps de brancher un câble série et le temps de copier le contenu de la mémoire.

LEAP tire parti de cette fenêtre de confiance pour permettre à deux nœuds voisins d'établir une clé symétrique de session à partir de la clé KIN de manière sécurisée. Une fois le temps (Tmin) écoulé, la clé KIN est supprimée de la mémoire du nœud pour éviter toute vulnérabilité ultérieure.

II.6.3.2. Chargement de la clé initiale :

Le contrôleur (SB) crée une clé initiale KIN et la charge dans chaque nœud. Chaque nœud u dérive une clé principale (Master Key) Ku en appliquant la fonction pseudo-aléatoire fk sur KIN, ce qui donne Ku = fKIN(u).

II.6.3.3. Découverte des voisins :

Immédiatement après son déploiement, le nœud u essaye de découvrir ses voisins en diffusant un message HELLO qui contient son id. Aussi, il initie un timer qui sera déclenché après le temps Tmin. Le nœud u attend un ACK de chacun de ses voisins v qui contient l'identificateur de v. l'ACK est authentifié en utilisant la clé principale Kv, qui est dérivée comme suit : Kv=fKIN(v). Comme le nœud u a la clé KIN, il pourra aussi dériver Kv, ainsi il pourra vérifier l'authenticité du ACK reçus :

u ==> *, u

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

II.6.3.4. Etablissement de la clé par-paire :

Le nœud u calcule sa clé par paire Kuv avec v, comme suit : Kuv=fKv(u).

Le nœud v peut de même calculer Kuv de la même manière. Kuv sert comme clé entre u et v.

II.6.3.5. Effacement des clés :

Lorsque le timer expire après Tmin, le nœud u efface KIN et toutes les clés principales Kv de ses voisins. Il est à noter que le nœud u n'efface pas sa clé principale Ku.

II.6.3.6. Sécurité de LEAP :

A la fin de ces quatre étapes, le nœud u aura établi une clé par paire partagée avec chacun de ses voisins. Cette clé sera utilisée pour sécuriser les données échangées entre eux. de plus, aucun nœud dans le réseau ne possède la clé KIN. Un adversaire peut écouter clandestinement tout le trafic dans cette phase, mais sans la clé KIN il ne peut injecter des informations incorrectes ou déchiffrer les messages. Un adversaire compromettant un nœud après Tmin, obtient seulement les clés du nœud compromis. Quand un nœud compromis est détecté, ses voisins suppriment simplement les clés qui ont été partagée avec lui.

II.7. Sécurité du routage dans les RCSF :

La couche de routage joue un rôle crucial dans la transmission de données à travers le réseau de capteurs sans-fil. Elle est constituée de deux blocs fonctionnels distincts, à savoir la construction de routes et le relais de données. Le premier bloc est responsable de la création d'un réseau de chemins reliant les nœuds du réseau à leur destination respective.

Le deuxième bloc utilise ensuite ce réseau pour transmettre les données collectées aux destinataires finaux. Il est important de noter que cette couche peut être la cible d'attaques de la part d'adversaires cherchant à compromettre la sécurité du réseau. Par conséquent, il est crucial de protéger ces deux blocs fonctionnels pour assurer le bon fonctionnement du réseau.

II.7.1. Attaques sur les protocoles de routage dans les RCSF :

Vus les contraintes des RCSF, la plus part des protocoles de routage sont assez simple, et par conséquent assez vulnérables aux attaques. Un nœud malicieux peut opérer sur deux niveaux :

- Les données échangées entre les nœuds
- La topologie du réseau créée par le protocole

Ces attaques peuvent être classées en deux catégories : **actives et passives.**

II.7.1.1. Attaques actives :

- ***Attaque de "jamming " :***

La technique de ***"jamming"***, est une attaque qui exploite la vulnérabilité des médias sans fil à l'interférence. En émettant des signaux à une fréquence spécifique, un nœud peut causer un déni de service. Ce type d'attaque présente un danger potentiellement élevé, car elle peut être réalisée par une personne non authentifiée et extérieure au réseau.

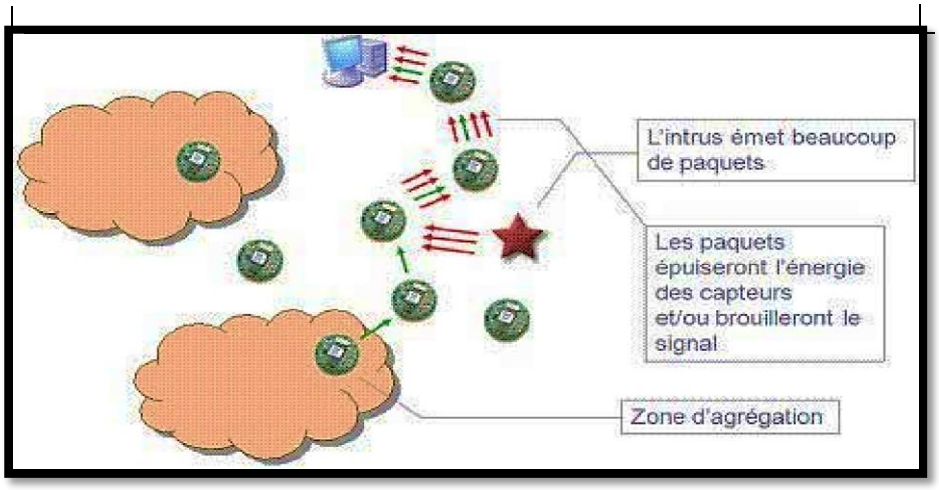


Figure II.16. *Attaque de jamming.*

- ***Attaque Sink hole :***

Lors d'une attaque de type ***"sinkhole"***, un nœud malveillant cherche à attirer à lui un maximum de chemins afin d'exercer un contrôle sur la majorité des données circulant dans le réseau. Pour y parvenir, l'attaquant doit donner l'apparence d'être extrêmement attrayant pour les autres nœuds en présentant des routes optimales.

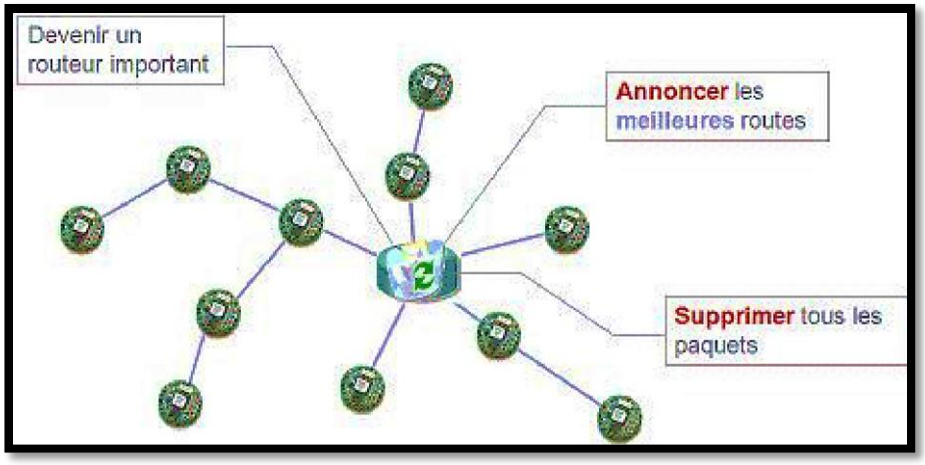


Figure II.17. *Attaque sinkhole.*

- ***Attaque Wormhole :***

L'attaque de type ***"wormhole"***, implique qu'un attaquant intercepte des paquets de données à un point du réseau, les encapsule, puis les transfère vers un autre attaquant pour les réintroduire dans le réseau. L'encapsulation peut être réalisée de deux manières différentes:

- **Multi-sauts:** L'encapsulation multi-sauts est une technique qui permet de dissimuler les nœuds intermédiaires entre les deux attaquants. Ainsi, les chemins qui passent par le nœud malveillant semblent plus courts. Cela facilite la création de sinkholes, notamment dans les protocoles qui utilisent le nombre de sauts comme critère pour choisir les chemins.
- **Communication directe:** Les routes qui passent par les attaquants sont plus rapides car ils se situent à seulement un saut de distance. Par conséquent, cette technique peut être utilisée pour compromettre les protocoles qui se basent sur la latence des routes ou qui sélectionnent la première route découverte.

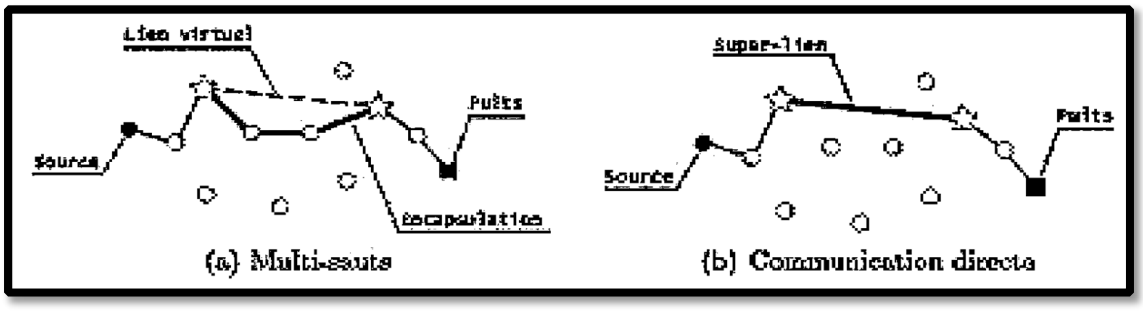


Figure II.18. *Attaque Wormhole .*

- ***Routing table poisoning :***

Des améliorations ont été apportées pour optimiser la connaissance des chemins dans un réseau. Lorsqu'un nœud reçoit une information de routage, il met à jour sa propre table de routage locale en conséquence. Cependant, un nœud malveillant a la possibilité de diffuser un grand nombre de fausses informations de routage dans le but de saturer les tables de routage des autres nœuds. Étant donné que ces tables ont une capacité limitée, cela peut provoquer un débordement et une saturation avec de fausses informations, perturbant ainsi le bon fonctionnement du réseau.

- ***Attaque Sybil :***

Il existe des algorithmes de routage qui mettent en place une redondance de chemins pour assurer la fiabilité du routage. Cependant, un attaquant peut compromettre ces systèmes en créant plusieurs identités pour lui-même, ce qui lui permet de faire passer plusieurs routes par son propre nœud malveillant, qui ne sont en réalité qu'un seul chemin.

- ***Attaque Hello flooding :***

L'attaque ***"hello flooding"***, est une nouvelle méthode d'attaque exploitant la faible portée des capteurs et la présence d'attaquants équipés d'ordinateurs portables. Cette attaque se fonde sur le fait que la plupart des liens entre l'attaquant et les capteurs sont unidirectionnels. L'attaquant peut alors diffuser une information de route optimale à tous les nœuds du réseau en émettant avec un signal puissant, ce qui incite les nœuds à mettre à jour leurs tables de routage locales en conséquence. Toutefois, lorsque les nœuds tentent d'utiliser cette route pour communiquer, ils sont bloqués car le prochain saut, qui est l'attaquant, est hors de portée.

II.7.1.2. Attaques passives :

- ***Selective Forwarding :***

Tous les protocoles de routage reposent sur l'hypothèse que les nœuds agissent de manière "honnête" en relayant les paquets qui passent par eux de manière normale. Cependant, un attaquant peut enfreindre cette règle en supprimant totalement ou partiellement ces paquets. De plus, si l'attaquant a préalablement utilisé une attaque de type sinkhole, il occupe une position de routeur important dans le réseau. Par conséquent, en abandonnant son rôle de routeur, les performances du système seront considérablement dégradées.

- ***Eavesdropping :***

Comme le média sans fil est un média ouvert, un nœud peut entendre toutes les communications de ses voisins. Cela peut divulguer d'importantes informations, comme la localisation d'un nœud important. La combinaison avec une attaque sinkhole aggrave d'avantage l'impact de cette attaque.

II.7.2. Types des solutions :

Nous distinguons trois niveaux de solutions aux attaques sur le routage de données dans les RCSF :

- **La prévention contre les attaques actives:** Dans le but de prévenir les attaques actives, des mécanismes cryptographiques sont couramment utilisés pour protéger la signalisation qui est nécessaire à la construction de routes. Les mécanismes d'authentification et de contrôle d'intégrité sont particulièrement employés pour empêcher un nœud malveillant d'injecter, de modifier ou de supprimer des informations qui seront utilisées pour la découverte, la construction ou la maintenance d'une route. Ces mécanismes permettent de vérifier que la source de la signalisation est légitime et que les informations n'ont pas été altérées en transit.
- **La détection de comportements suspects:** vise à identifier les signes pouvant indiquer la présence d'une attaque passive. Les comportements non coopératifs, tels que le refus de relayer des paquets, le manque de coopération ou la non-conformité aux protocoles établis, sont généralement considérés comme des signaux d'alerte. La détection de ces comportements peut être réalisée à l'aide d'algorithmes de surveillance ou de mesures de performance pour identifier les nœuds qui ne respectent pas les règles établies et pour prévenir les attaques sur les réseaux de communication.
  - **La tolérance :** dans cette catégorie,on vise à introduire des mécanismes qui permettent de tolérer les défaillances de nœuds dans le contexte d'attaques ou de pannes. Parmi ces mécanismes, on trouve le routage multi-chemin qui permet de sélectionner plusieurs chemins pour acheminer les données plutôt qu'un seul chemin. Cette approche permet de garantir que les données seront toujours transmises même si l'un des chemins est compromis ou indisponible. Les mécanismes de tolérance de défaillance sont essentiels pour assurer une communication fiable et robuste dans les réseaux de communication.

La figure (II.19), résume les catégories de solutions à préconiser pour faire face à différents types d'attaques :

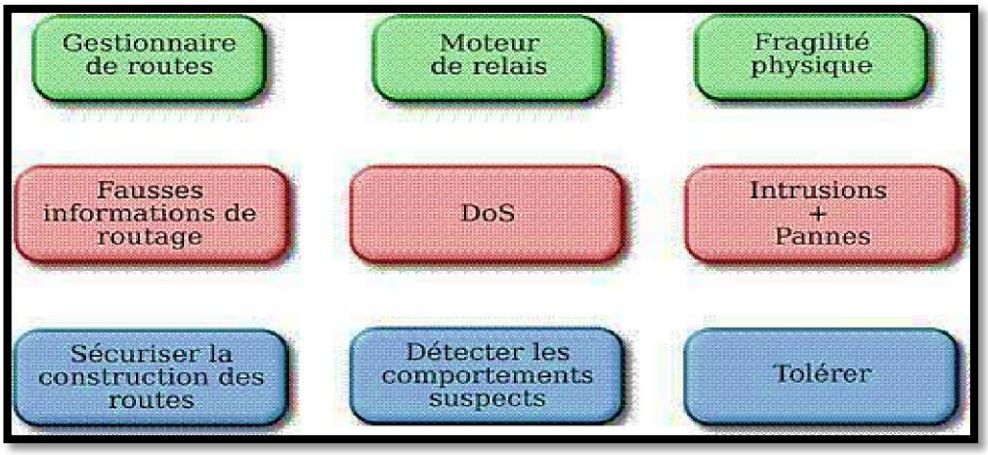


Figure II.19. *Catégories de solutions contre les attaques sur le routage .*

II.7.3. INSENS (Intrusion-tolerant routing for wireless sensor networks) :

L'idée du protocole est de permettre à la SB de tracer une cartographie correcte du réseau qui permettra d'établir les tables de routage pour chaque capteur. Ces tables seront transmises par la suite aux nœuds concernés de façon sécurisée [39]. Le protocole vise deux objectifs :

- **Opérer correctement en présence d'intrus:** L'un des objectifs clés d'INSENS est de garantir le fonctionnement correct en présence d'intrus. Pour y parvenir, INSENS utilise un protocole tolérant aux intrusions qui permet d'assurer le routage même en présence d'intrus. Le protocole crée plusieurs chemins indépendants pour chaque couple de communicants. Ces chemins indépendants sont conçus de manière à partager un nombre limité de nœuds et de liens, idéalement seulement la source et la destination. Cette caractéristique permet de limiter l'impact des intrus sur les communications, car ils ne peuvent altérer que les données transitant par un seul chemin. En résumé, le protocole INSENS est conçu pour garantir une communication fiable et sécurisée en présence d'intrus.

- **Scalabilité et économie d'énergie :** INSENS vise à offrir une solution à la fois scalable et économe en énergie pour le calcul des chemins indépendants et l'établissement des tables de routage. Cependant, cette tâche est assez lourde et requiert des ressources importantes. Pour y faire face, INSENS utilise une approche centralisée. Les calculs sont effectués dans une station de base (SB) et les résultats sont transmis de manière sécurisée à chaque nœud. Le protocole suit un processus en plusieurs phases pour atteindre ses objectifs. En somme, INSENS est conçu pour fournir une solution de routage efficace et économe en énergie, en utilisant une approche centralisée pour gérer la complexité des calculs nécessaires.

II.7.3.1. Initiation authentifiée de la construction de l'arbre :

La SB doit d'abord dresser un arbre couvrant tout le réseau, dont elle est la racine. Cet arbre permettra d'acheminer les messages de contrôle entre les capteurs et la SB. Afin d'éviter le spoofing de SB, INSENS emploie un mécanisme de broadcast authentifié. Pour ce faire, la SB génère une chaîne de hachage à sens unique $(ni)0\leq i\leq k$ comme suit : $ni+1=h(ni)$, $0<i<k$ où n0 est choisi aléatoirement, et nk est connu par tous les nœuds, et h est une fonction de hashage à sens unique.

Périodiquement, la SB génère une requête pour re-construire les tables de routage des capteurs. Le format du message est le suivant :

type|ows|size|path|MACRx

Le champ ows (One-Way Sequence number) désigne la valeur courante de la chaîne de hachage (i.e. pour la requête i, ows=ni). Chaque nœud maintient localement la dernière valeur reçue de ows, désignée par owsfresh. Lorsqu'un nœud x reçoit une requête, il vérifie sa fraîcheur et son authenticité par la relation suivant :

Trouver j, tel que j>0 et ows=hj(owsfresh)

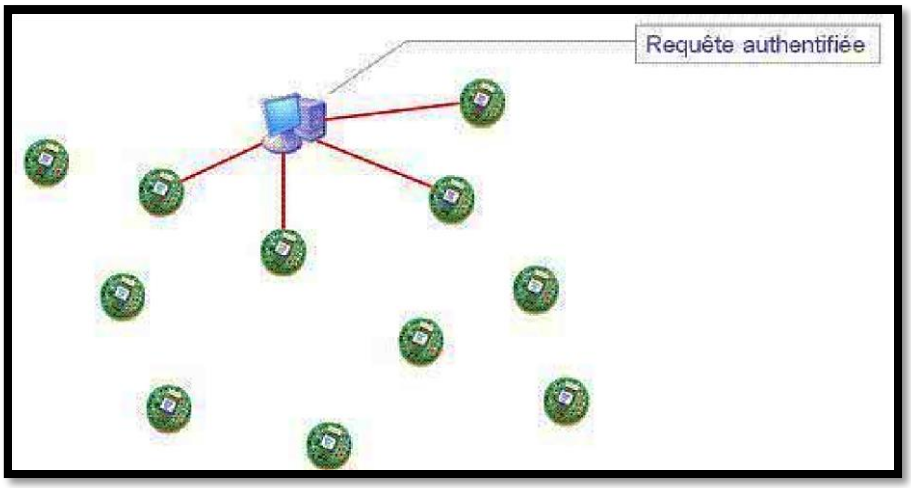


Figure II.20. *Requête authentifiée de construction de l'arbre* [16].

II.7.3.2. Construction de l'arbre par relayage de la requête :

Un nœud x qui reçoit le message de requête précédent, ajoute son identificateur au champ path et calcule le champ MACRx :

MACRx = MAC (kx, size|path|ows|type)

où kx représente la clé sectrète partagée entre le nœud x et la SB. Le nœud doit aussi choisir son "upstream" vers la SB, qui représente le premier voisin qui a émis une requête,valide, et sauvegarde son MACR, qui sera désigné par MACRupstream.

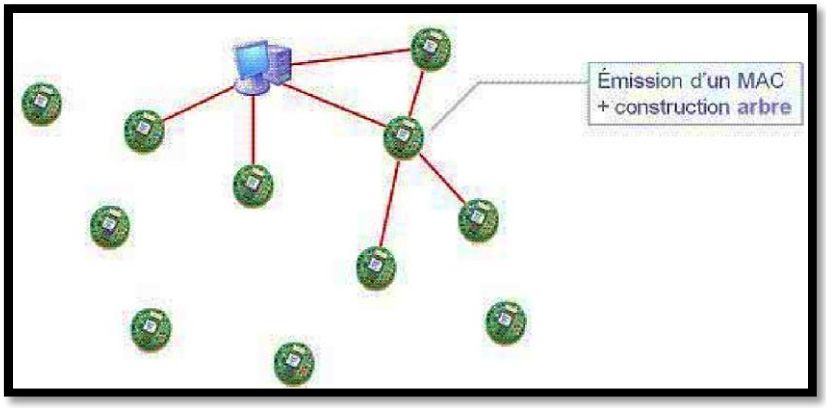


Figure II.21. *Construction de l'arbre* [16].

II.7.3.3. Route feedback :

Après l'émission de la requête, le nœud attend un certain temps pour envoyer un feedback à la SB. Cette période lui permet de récolter les informations sur son voisinage, qui vont permettre à la SB d'établir une vue globale du réseau d'une manière sécurisée. Un message feedback contient les informations suivantes:

type|ows|path_info|nbr_info|MACRupstream|MACFx

Le champ "path_info" contient la liste des nœuds entre le nœud x et la SB, l'identification de x et son MACR:

IDx|size|path|MACRx

Le champ "nbr_info" contient la liste des identificateurs des voisins ainsi que leurs MACR :

Size|IDa|MACRa|IDb|MACRb| . . .

Le champ MACFx est calculé comme suit :

MACFx=MAC (kx, path_info|nbr_info|ows|type)

Pour renforcer la sécurité, le routage du message s'effectue grâce au champ MACRupstream. Ainsi, lorsqu'un nœud reçoit un message feedback, il compare la valeur du MACRupstream reçue avec sa valeur liée au ows reçu, et relaie éventuellement le message.

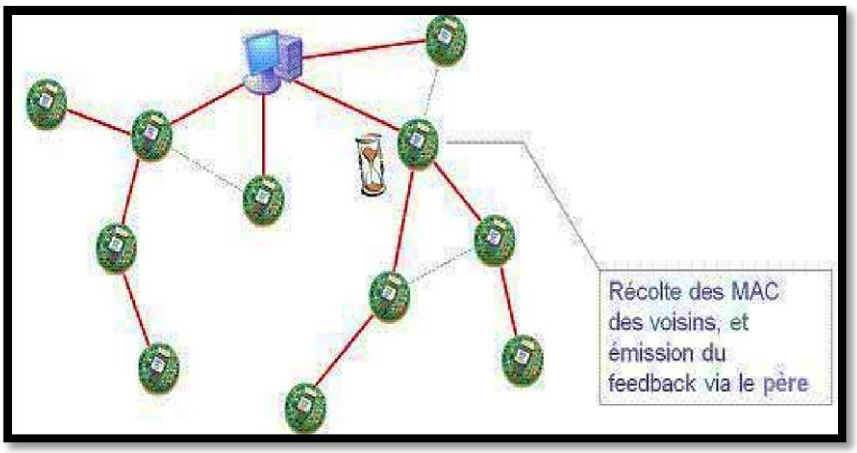


Figure II.22. *Route feedbac* [16].

II.7.3.4. Construction des tables de routage :

Après avoir émis une requête, la SB attend une période pour traiter les messages de feedback et dresser le graphe du réseau. Pour vérifier les informations du feedback d'un nœud x, la SB recalculera le MACFx et comparera les valeurs des MACR de chaque voisin de i avec la valeur MACRi reçue dans le feedback du nœud i. Les informations falsifiées seront rejetées, et un graphe correct pourra être établi.

Ensuite, la SB pourra rechercher les chemins indépendants en vue de construire les tables de relais pour chaque nœud. Les chemins indépendants seront choisis pour minimiser le nombre de nœuds en commun. L'algorithme utilisé trouve d'abord le chemin le plus court entre chaque couple de communicants, puis essaye de trouver un autre chemin dans le sous-graphe ne contenant pas les nœuds du premier chemin, leurs voisins et les voisins des voisins. Si cela est impossible, le processus est réitéré en ajoutant l'ensemble des voisins des voisins, puis l'ensemble des voisins. Pour chaque capteur, la SB calcule une table de relais contenant une entrée pour chaque chemin

passant par le capteur. Le calcul des chemins indépendants et la construction des tables de routage sont effectués par la SB, ce qui permet une économie d'énergie pour les nœuds du réseau.la table est encapsulée dans le message suivant:

type|ows|size|routingTable|MAC

où le MAC est calculé comme suit :

MAC=MAC(kx,type|ows|size|table)

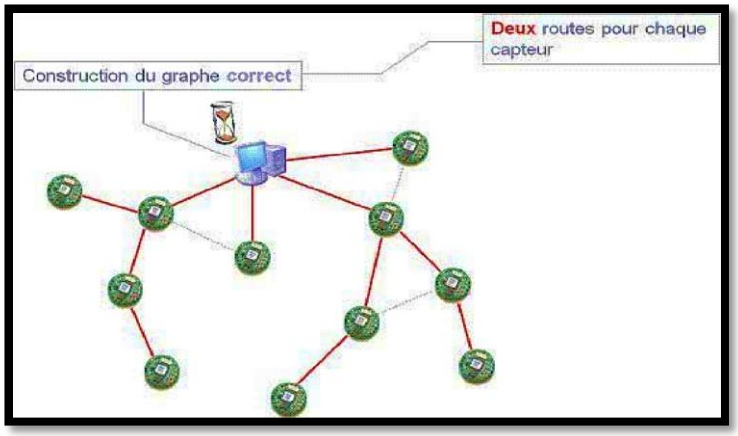


Figure II.23. *Construction et distribution des tables de routage* [16].

II.7.4. SecRoute :

Le protocole SecRoute est un protocole de routage hiérarchique sécurisé qui organise le réseau en clusters avec un chef pour chaque cluster. La table de clés secrètes de chaque capteur est maintenue localement par le nœud collecteur, qui est censé connaître l'organisation du réseau en clusters. Chaque capteur doit avoir une clé pré-chargée. De plus, pour sécuriser les échanges intra-cluster, chaque cluster doit posséder une clé qui doit être connue par tous les nœuds du groupe ainsi que par le chef du cluster. Le protocole SecRoute ne spécifie pas la méthode de construction de clusters et suppose que cette tâche est accomplie par un autre protocole tel que LEAP.

II.7.4.1. Propriétés du SecRoute :

Le protocole possède les propriétés suivantes :

- Les paquets de routage ne sont pas volumineux, car ils ne contiennent qu'une information partielle sur le chemin parcouru.:
- Le protocole utilise une architecture à deux niveaux, dans laquelle les chefs agrègent les données des membres puis les transmettent au nœud collecteur.
- Le protocole emploie seulement des méthodes de chiffrement symétrique.
- Pour des raisons de sécurité, le protocole remplace les unicasts par des broadcasts locaux ciblés. En effet, en évitant les unicasts, un message émis est reçu par tous les voisins. Donc, cela permet de vérifier, lors du relais, l'intégrité du message émis par le prochain saut.
- Chaque capteur stocke une table de routage ayant le format suivant :

Source	*Pre*	*Next*
ID_{Soure}	ID_{pre_2}, ID_{pre_1}	D_{next_1}, D_{next_2}
⋮	⋮	⋮

Figure II.24. *Format de la table de routage dans SecRoute .*

La table est organisée suivant l'adresse des sources. Le champ Pre (respectivement Next) indique les deux prochains sauts vers la SB (respectivement source) sur le chemin entre la source et la SB.

II.7.4.2. Découverte des chemins :

Le nœud source initie une découverte des chemins en émettant vers ses voisins directs un paquet de requête RREQ contenant les informations suivantes:

IDsource|IDsink|IDRREQ|Nsource MAC(Ksource, IDsource|IDsink|IDRREQ|Nsource)

Avec :

- **Dsource, IDsink, IDRREQ:** les identificateurs de la source, du collecteur et de la requête.
- **Nsource :** un nonce généré par la source.

Lorsqu'un nœud reçoit une requête, elle n'est acceptée qu'avec l'unicité de son identificateur IDREQQ. Il met à jour par la suite sa table de routage en utilisant l'information des deux sauts précédents vers la source. Avant de relayer la requête, le nœud remplace les valeurs de IDpre et IDthis par, respectivement, IDthis et son identificateur [18] :

IDthis|IDpre|IDsource|IDsink|IDRREQ|NsourceMAC(Ksource,IDsource|IDsink|IDRREQ| Nsource)

II.7.4.3. Relais de la réponse :

Lorsque le collecteur reçoit la première requête, il vérifie le MAC construit par la source en utilisant la clé relative à son identificateur IDsource, sauvegardée dans la table locale. Si le MAC est correct, le collecteur met à jour sa table de routage en utilisant les champs IDpre et IDthis. Il génère ensuite une réponse RREP ayant le format suivant :

IDpre|IDthis|IDnext|IDsink|IDRREQ|NsinkAC(Ksource,IDsource|IDsink |IDRREQ| Nsink)

La requête est émise en broadast local, ciblé à l'aide du champs IDnext. Lorsque le capteur voisin ayant l'identificateur IDnext reçoit cette réponse, il met à jour sa table de routage en conséquence, puis remplace les champs IDpre et IDthis par IDthis et son identificateur [16].

Il doit aussi modifier le champ IDnext par l'identificateur connus lors de la phase de découverte de chemins. Si le nœud ne reçoit pas la réponse émise par IDnext après un certain temps, il ignore toutes les requêtes émises pendant la prochaine phase de découverte. Le nœud de relais doit aussi vérifier que la répon se émise par le prochain saut est valide, en s'assurant qu'elle est bien destinée au nœud à deux sauts contenu dans le chemin vers la source (i.e. le champ IDpre2 de la table de routage).

Lorsque la source reçoit la première réponse, elle vérifie le MAC généré par le collecteur et met à jour sa table de routage en ajoutant IDthis et IDpre comme prochains sauts vers le collecteur [24] [25].

II.7.4.4. Relais des données :

Le relais des données est effectué en deux étapes. Les nœuds membres émettent leurs données vers le chef du groupe, qui va par la suite envoyer le résumé vers le collecteur. L'établissement du chemin entre le chef et le collecteur s'effectue grâce au procédé décrit précédemment, i.e.: le chef représente la source de son groupe. Pour les communications intra-groupe, le protocole utilise la clé de cluster CK établie lors de sa création. Chaque donnée D d'un capteur est émise vers le clusterhead ID, encapsulée comme suit [21] :

IDthis|IDnext|IDsource|IDsink|QID|{D}Ksource MA (Ksource,IDsource|IDsink|QID|{D} Ksource)

Un nœud intermédiaire avec le même identificateur que IDnext relaie le paquet en remplaçant IDthis et IDnext. Si un nœud de relais ne reçoit pas le paquet émis par le prochain saut, une maintenance de la route doit être effectuée. Pour cela, il doit émettre vers la source un message d'erreur permettant d'enclencher à nouveau la procédure de découverte de routes. De plus, le prochain saut est ajouté à la liste noire (black list).

Cette liste contient les identificateurs dont le nœud doit ignorer leurs paquets de réponse. Lorsque le paquet atteint le collecteur, il vérifie le MAC en utilisant la clé de la source et peut donc décrypter la donnée et l'utiliser.

II.8. Sécurité de l'agrégation dans les RCSF :

II.8.1. Attaques sur l'agrégation de données dans les RCSF :

L'agrégation de données est un processus important dans les RCSF car il permet de réduire les transmissions redondantes et d'économiser de l'énergie. Pour ce faire, un nœud intermédiaire doit collecter les données transmises par ses voisins et les agréger en utilisant une fonction spécifique, telle que la moyenne, le maximum, le minimum, etc. Cependant, cette approche peut être vulnérable aux attaques de nœuds malveillants qui peuvent injecter de fausses données dans le réseau ou falsifier le résultat de l'opération d'agrégation. Si un nœud malicieux réussit à s'introduire dans le réseau, il peut falsifier l'information captée dans toute une zone, compromettant ainsi l'intégrité des données agrégées [16].

La figure (II.25), montre le risque qui peut être encouru par une mauvaise :

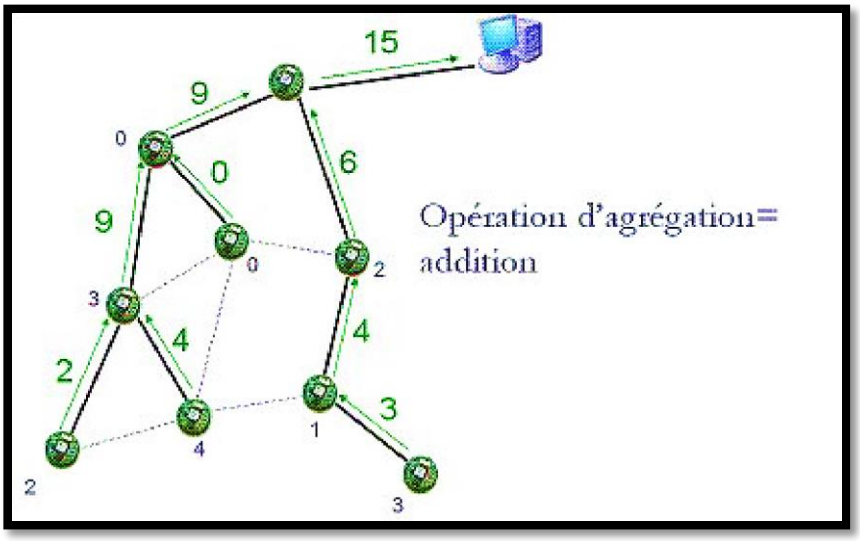


Figure II.25. *Fonctionnement correcte de l'agrégation* [16].

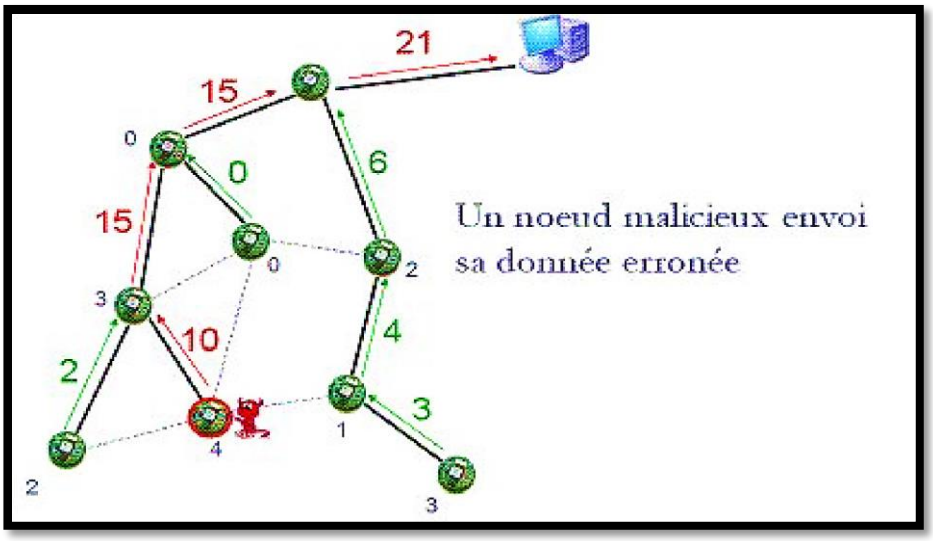


Figure II.26. *Un malicieux injecte une fausse donnée* [16].

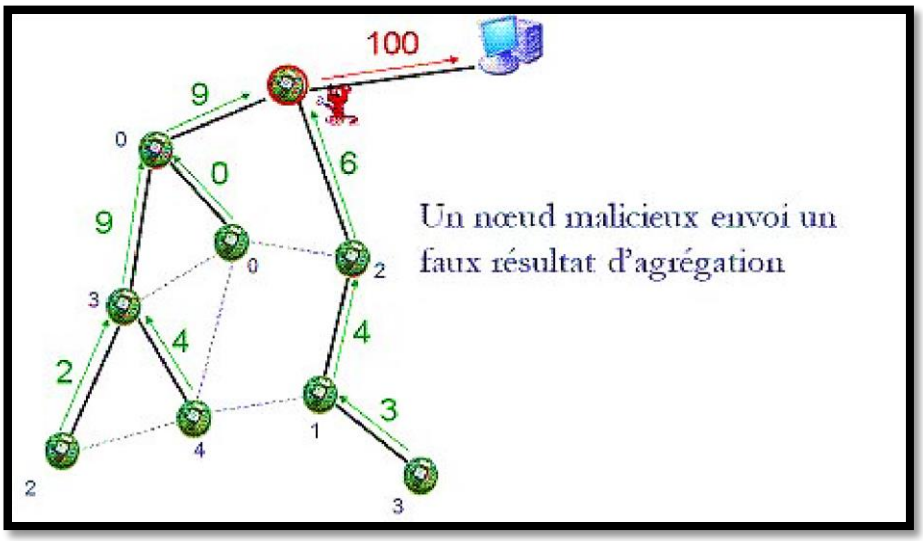


Figure II.27. *Un malicieux falsifie le résultat d'une agrégation* [16].

L'un des principaux défis dans les réseaux de capteurs sans fil est de permettre aux nœuds de relais d'agréger les données transmises par les nœuds voisins afin de calculer l'information utile, tout en évitant les risques de falsification, de suppression ou de modification des données.

Cela nécessite une conception soigneuse des mécanismes de sécurité pour garantir l'intégrité et la confidentialité des données agrégées, ainsi que la protection contre les attaques malveillantes.

Il existe deux grandes catégories de solutions selon le mécanisme cryptographique utilisé :

- **Solutions basées sur le cryptage de bout en bout** : pour sécuriser l'information tout en permettant aux nœuds intermédiaires d'effectuer l'agrégation des données. Ces solutions utilisent des mécanismes cryptographiques pour assurer la confidentialité et l'intégrité des données pendant leur transmission. Cependant, dans ce type de solution, la vérification de l'information se fait généralement uniquement au niveau du collecteur, ce qui peut conduire à une propagation de fausses informations dans le réseau.
- **Solutions basées sur le cryptage de proche en proche :** Les solutions basées sur le cryptage de proche en proche permettent de vérifier l'authenticité de l'information à chaque niveau de l'arbre couvrant le RCSF. Cette approche garantit que les données transmises sont fiables et peuvent être rejetées à n'importe quel niveau de l'arbre en cas de détection de fausses données [26].

La figure (II.28), résume cette classification :

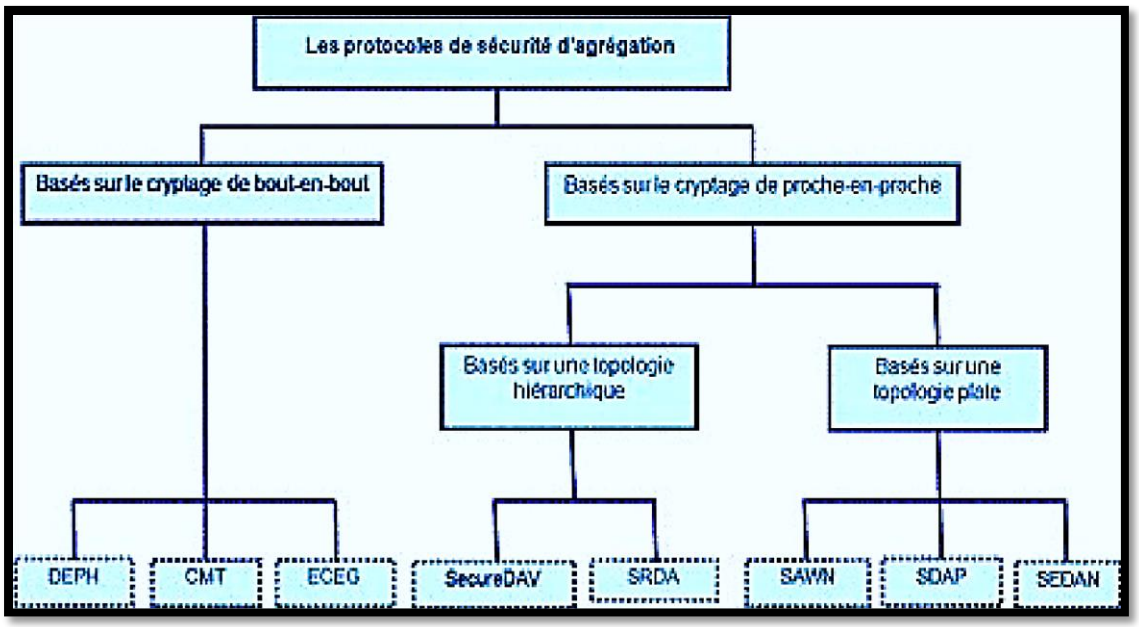


Figure II.28. *Classification des solutions d'agrégation sécurisée* [21].

II.8.2. SAWN (Secure Aggregation for Wireless Networks) :

Le protocole SAWN repose sur l'hypothèse que deux nœuds adjacents ne peuvent pas être compromis simultanément. Pour assurer la fiabilité de l'agrégation des données, chaque nœud effectue une vérification à deux sauts : il vérifie que l'agrégation des données de ses petits-fils, réalisée par son fils, est correcte. La vérification de l'agrégation est effectuée de manière différée dans le temps, en utilisant le protocole μTESLA pour authentifier les clés utilisées dans l'authentification des données et de leur agrégation.

Dans ce qui suit nous supposons que les nœuds ont le moyen de vérifier l'authenticité des clés partagées entre les nœuds et la SB, lorsque cette dernière révèle les clés pour la vérification [16][24].

Chaque nœud feuille transmet sa lecture à son père. Les messages incluent la lecture des données du nœud, son id, ainsi qu'un MAC calculé grâce à la clé KAi. Cette dernière est partagée entre le nœud A et la station de base, mais n'est pas encore connue par les autres capteurs. Le nœud père stocke le message ainsi que son MAC jusqu'à la révélation de clé KAi par la station de base. A cet instant, il vérifiera le MAC et envoi une alarme en cas de différence. L'agrégation des lectures est exécutée dans chaque étape intermédiaire. Les nœuds attendent pendant un temps indiqué pour recevoir des messages de leurs fils et retransmettent ensuite les messages et les MACs qu'ils reçoivent directement de leurs fils immédiats [19].

Les nœuds agrègent les données qu'ils reçoivent de leurs petits-fils (via leurs fils) et transmettent le MAC de la valeur d'agrégation. Après l'arrivée de tous les messages à la station de base, cette dernière révèle les clés temporaires des nœuds. Une fois que la clé (KAi) est révélée, les nœuds passent à la clé temporaire suivante (KAi+1).

Considérons l'arbre d'agrégation illustré par la figure (II.29), L'exemple illustre le ième tour où l'en utilise les clés Kxi pour authentifier les messages transmis.

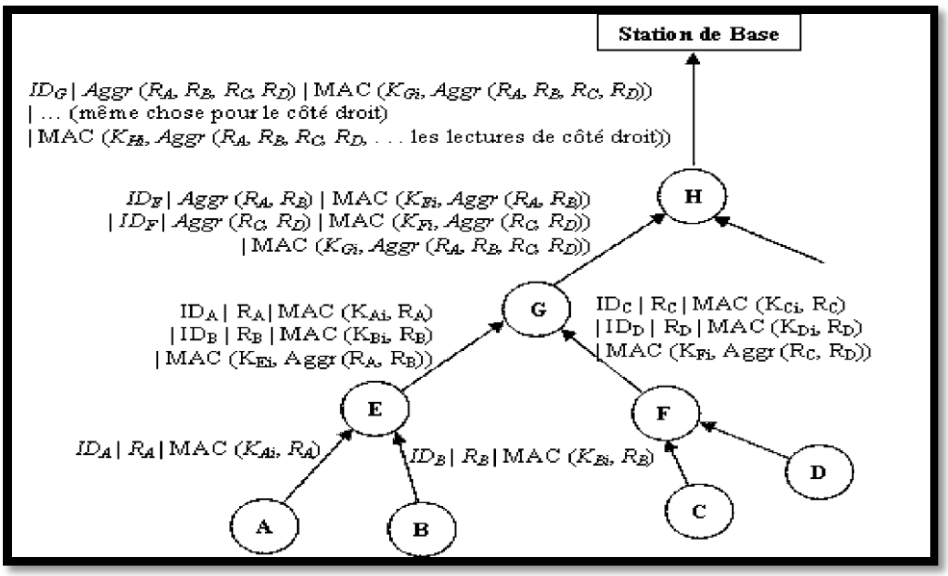


Figure II.29. *Exemple d'arbre d'agrégation sécurisé avec SAWN* [23].

- ***Transmission de données :***

Les nœuds A, B, C et D envoient des données à la station de base via l'arbre d'agrégation construit avec un protocole de routage [17].

- Les nœuds feuilles envoient des données à leur père. Les messages incluent des MACs calculés avec la clé d'authentification courante :

A==>E : RA | IDA | MAC(KAi,R A)

Chaque clé est utilisée pour authentifier un seul message ce qui empêchera l'attaque de rejeu.

- Les nœuds intermédiaires reçoivent les messages de leurs fils. Le nœud père ne peut pas encore vérifier le MAC car la clé du fils ne sera révélée que pendant la phase de vérification. Pour le moment, le père stocke le message et le MAC. Le nœud intermédiaire attend les paquets des fils, et envoie ensuite un message à son père

contenant les lectures des fils, leurs MACs, ainsi que le MAC calculé sur la valeur d'agrégation:

E==>G : RA | IDA | MAC(KAi,RA) | RB | IDB | MAC(KBi,RB)|MAC(KEi, Aggr(RA,RB))

Il n'y a aucun besoin de transmettre la valeur d'agrégation calculée, puisque G peut calculer Aggr (RA,RB) depuis les valeurs RA et RB. Ce n'est pas aussi nécessaire de transmettre l'IDE à G, parce que G connaît la topologie du réseau donc peut déterminer le nœud qui envoie le message

- Le nœud G reçoit les messages des nœuds E et F. Pour chacun d'eux, G calcule les valeurs de l'agrégation des lectures de ses petits-fils c'est-à-dire (A, B, C et D). Il transmet alors les valeurs agrégées de ses petits-fils, l'ID de ses fils et leurs valeurs de MAC. G calcule aussi et transmet le MAC de la valeur d'agrégation suivante :

Aggr(RA,RB,RC,RD)=Aggr(Aggr(RA,RB),Aggr(RC,RD)).

Puisque la fonction de l'agrégation est connue à tous les nœuds, le MAC calculé par E authentifiera la valeur calculée par G. Les lectures des capteurs et les valeurs de MAC reçues à partir de E et F sont stockées pour vérification postérieure [21].

G==>H : IDE | Aggr(RA,RB) | MAC(KEi, Aggr(RA,RB)) | IDF | Aggr(RC,RD) | MAC(KFi,Aggr(RC,RD)) | MAC(KGi,Aggr(RA,RB,RC,RD))

De la même façon, le nœud H reçoit des messages de G et d'une autre branche, et transmet à son tour le message agrégé à la station de base. Noter que la longueur du message n'augmente pas si le réseau était plus profond.

- La station de base reçoit le message de H. Elle peut calculer la valeur de l'agrégation finale, Aggr (RA, RB, RC, RD, ...) en utilisant Aggr (RA,RB,RC,RD) et les autres valeurs de ses nœuds fils.

- ***Validation des données :***

Le protocole SAWN a pour objectif d'authentifier toutes les lectures ayant contribué à l'agrégation de données, sans avoir à recevoir toutes ces lectures. Pour cela, la station de base transmet un unique message contenant la clé courante Kxi qu'elle partage avec chaque nœud x du réseau pour valider les données (lectures des nœuds et valeurs agrégées). Ce message est authentifié par un MAC en utilisant une clé préalablement authentifiée via μTESLA.

Si un nœud détecte une erreur dans l'étape de validation des données, il émet une alarme. Cette alarme est générée par un parent lorsqu'il constate une incohérence entre le MAC d'agrégation de son fils et les données de ses petits-fils, ou lorsque les MAC des données elles-mêmes sont erronés [16] [19].

II.8.3. Protocoles basés sur le cryptage de bout en bout :

Les protocoles de cette catégorie utilisent une clé partagée entre chaque nœud et le nœud collecteur afin d'assurer l'intégrité des données transmises dans le réseau. En raison du cryptage des contenus de données, les nœuds utilisent une forme de cryptographie spécifique appelée "Privacy Homomorphism (PH)" qui leur permet d'effectuer des opérations d'agrégation.

Un algorithme est considéré comme Privacy Homomorphism (PH) s'il permet, sans avoir à décrypter les valeurs x et y, de calculer E(x ¤ y) à partir des valeurs cryptées E(x) et E(y). Cela signifie que l'algorithme satisfait la propriété suivante :

EK1(x1) ¤ EK2(x2)=EK1+K2(x1 ¤ x2), où Ki sont les clés et xi sont les données

- Le point unique de vérification dans ce type de protocoles est le nœud collecteur.
- Ce dernier ayant toutes les clés utilisées pour crypter les données dans le réseau.
- Castelluccia, Myldetun et Tsudik ont proposé le protocole CMT basé sur l'hypothèse que chaque nœud utilise une clé symétrique partagée entre ce nœud et le nœud collecteur. L'idée de ce protocole est que chaque nœud fait l'addition modulaire entre sa clé stockée et sa sonnée. Pendant la phase d'acheminement des données, l'agrégation se fait sur ces données qui sont déjà cryptées. L'algorithme suivant montre les différentes étapes de ce protocole [16][22]:

L'algorithme de CMT

Paramètre :

Sélection d'un grand nombre entier M.

Cryptage :

Le message $m \in [0, M-1]$.

Aléatoirement générer une clé $k \in [0, M-1]$.

$C = (m + k) mod M$.

Décryptage :

$m = (c - k) mod M$.

Agrégation :

$c_{12} = (c_1 + c_2) mod M$.

Figure II.30. *Algorithme CMT* [23].

La taille du paquet dans ce protocole dépend de la taille de M, et une seule addition modulaire suffit pour l'agrégation et le cryptage. Ainsi, ce protocole ne consomme pas beaucoup d'énergie.

- ***Protocole Elliptic Curve ElGamel :***

Ce protocole sert de l'algorithme de chiffrement asymétrique ElGamel basé sur les courbes elliptiques (ECEG), qui est une méthode qui nécessite moins d'énergie que les systèmes asymétriques conventionnels tels que RSA [22].

L'algorithme de ECEG

Paramètre :

Une clé privé x.

Une clé publique(G, H), G et H des points dans **ECEG**, $H = xG$.

Cryptage :

$C = [c_1, c_2] = [kG, kH + mG]$ = un point dans **ECEG**.

Décryptage :

$mG = (kH + mG) - x(kG)$.

Agrégation :

$C_{12} = C_1 + C_2 = [(c_{11} + c_{21}), (c_{12} + c_{22})]$.

Figure II.31. *Algorithme ECEG* [23].

II.9. Conclusion :

Dans ce chapitre, nous avons étudié la question de la sécurité dans les RCSF ainsi que les limitations qui empêchent l'application des méthodes de sécurité classiques. Nous avons également examiné les protocoles et solutions de gestion de clés proposés pour les RCSF, soulignant que les méthodes de pré-distribution de clés sont les plus appropriées pour leur faible coût. La faible adaptabilité de la cryptographie asymétrique a conduit à un intérêt accru pour la cryptographie symétrique et les méthodes de pré-distribution de clés.

Cependant, il est possible que cela change dans l'avenir avec l'émergence de méthodes de cryptographie asymétrique peu coûteuses, telles que la ECC, qui pourraient être prometteuses pour sécuriser les RCSF et méritent davantage d'études approfondies.

CHAPITRE III :

APPROCHE DE SÉCURITÉ PROPOSÉE.

III.1. Introduction :

Les réseaux de capteurs sans fil (RCSF) ont connu une croissance rapide au cours des dernières années. l'utilisation des capteurs dans des systèmes critiques tels que les centrales nucléaires, les avions et les hôpitaux nécessite des mécanismes efficaces pour garantir l'authenticité, la confidentialité et l'intégrité des données détectées et transmises. Cependant, la sécurité est un problème difficile à résoudre dans les (RCSF), car les capteurs sont généralement déployés dans des environnements hostiles.

En outre, la mémoire et la puissance de traitement limitées, ainsi que la courte portée de communication des nœuds de capteurs, posent plusieurs problèmes lors de la mise en œuvre des schémas cryptographiques traditionnels dans les environnements sans fil.

Les RCSF nécessitent donc des schémas de cryptage efficaces en termes d'espace de stockage, de consommation d'énergie et de vitesse de fonctionnement. L'enjeu principal de la conception de systèmes cryptographiques pour les RCSF est de maintenir le compromis entre la sécurité, la performance et le coût. C'est pourquoi la recherche actuelle se concentre sur la conception d'algorithmes de chiffrement sécurisés et légers. Malgré tous les efforts des chercheurs, bon nombre de ces ciphers légers ont des performances relativement médiocres par rapport aux schémas cryptographiques classiques. Pour remédier à ces lacunes, nous proposons un algorithme de chiffrement qui s'appelle AES (Advanced Encryption Standard) qui utilise le chiffrement à clé symétrique, ce qui signifie que la même clé est utilisée à la fois pour le chiffrement et le déchiffrement des données. Cette clé est partagée entre l'expéditeur et le destinataire des données chiffrées, et les deux parties doivent avoir accès à la clé pour chiffrer et déchiffrer les données.

Dans ce chapitre nous avons présenter ce mécanisme de sécurité dédié aux (RCSF).

III.2. Approche De Sécurité Proposée :

III.2.1. Définition:

Le Standard de Chiffrement Avancé (AES) est un algorithme de chiffrement de bloc symétrique Développé par Joan Daemen et Vincent Rijmen. utilisé pour chiffrer et déchiffrer des données électroniques. Il a été adopté par le gouvernement américain en 2002 comme norme pour le chiffrement sécurisé des données.

AES opère sur des blocs de données de texte en clair de longueur fixe et utilise une clé secrète pour effectuer le chiffrement et le déchiffrement. La longueur de la clé peut être de 128, 192 ou 256 bits, avec des clés plus longues offrant une sécurité plus forte. L'algorithme se compose d'une série de tours, chaque tour utilisant une transformation différente pour brouiller les données.

AES est considéré comme l'un des algorithmes de chiffrement les plus sûrs et est largement utilisé dans diverses applications, telles que le transfert de fichiers sécurisé, le chiffrement des emails et les réseaux privés virtuels (VPN). Son utilisation généralisée en a fait une norme pour le chiffrement sécurisé des données dans de nombreuses industries et applications [27].

III.2.2. Principe de fonctionnement du protocole de sécurité proposée :

AES/Rijndael est un chiffrement de bloc itéré, ce qui signifie que le bloc d'entrée initial et la clé de chiffrement subissent plusieurs rounds de transformation avant de produire la sortie. Chaque résultat intermédiaire de chiffrement est appelé un état. Le processus de chiffrement utilise un ensemble de clés spécialement dérivées appelées clés de round. Ces clés sont appliquées, ainsi que d'autres opérations, sur un tableau de données qui contient exactement un bloc de données à chiffrer. Ce tableau que nous appelons le tableau d'état. Voici les étapes pour chiffrer un bloc de 128 bits [28] :

- ✓ Dérivez l'ensemble des clés de round à partir de la clé de chiffrement.
- ✓ Initialisez le tableau d'état avec les données de bloc (texte en clair).
- ✓ Ajoutez la clé de round initiale au tableau d'état de départ.

- ✓ Effectuez neuf rounds de manipulation d'état.
- ✓ Effectuez le dixième et dernier round de manipulation d'état.
- ✓ Copiez le tableau d'état final en tant que données chiffrées (texte chiffré).

La raison pour laquelle les rounds ont été énumérés comme "neuf suivis d'un dernier dixième round" est parce que le dixième round implique une manipulation légèrement différente des autres. Le bloc à chiffrer n'est qu'une séquence de 128 bits. AES travaille avec des quantités de bytes, donc nous convertissons d'abord les 128 bits en 16 bytes. Les opérations dans AES sont effectuées sur un tableau de bytes bidimensionnel de quatre rangées et quatre colonnes.

Une itération des étapes ci-dessus est appelée un round. Chaque round du processus de chiffrement nécessite une série d'étapes pour modifier le tableau d'état. Ces étapes impliquent quatre types d'opérations appelées :

- **Sub Bytes.**
- **Shift Rows.**
- **Mix Columns.**
- **Add Round Key.**

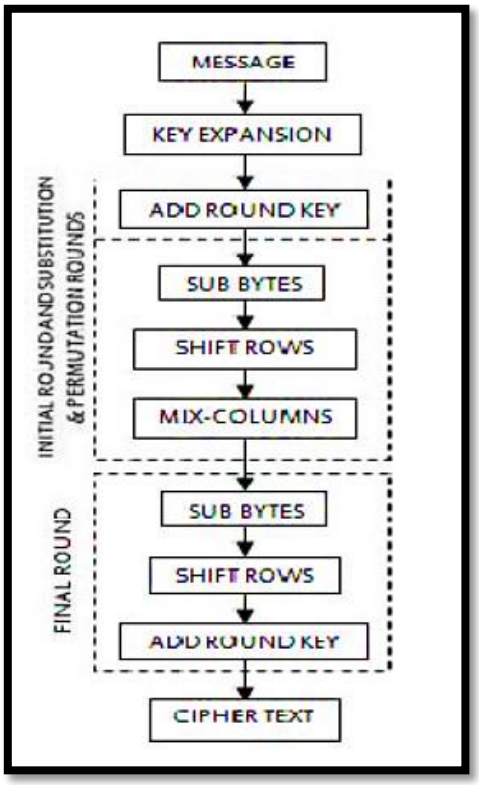


Figure III.1. *AES Algorithm* [28].

La seule exception est que dans le dernier round, l'étape Mix Column n'est pas effectuée pour rendre l'algorithme réversible pendant le déchiffrement. Le nombre de rounds de l'algorithme dépend de la taille de la clé comme indiqué dans le tableau ci-dessous :

Number of rounds(Nr)	128 -bit of Data	192-bit of Data	256 -bit Data
128-bit Key	10	12	14
192-bit Key	12	12	14
256-bit Key	14	14	14

Figure III.2. *Combinaisons Clé-Bloc-Ronde* [28].

III.2.3. Architecture de l'algorithme AES de Rijndael :

L'ordonnancement des étapes est illustré dans la figure (III.3) :

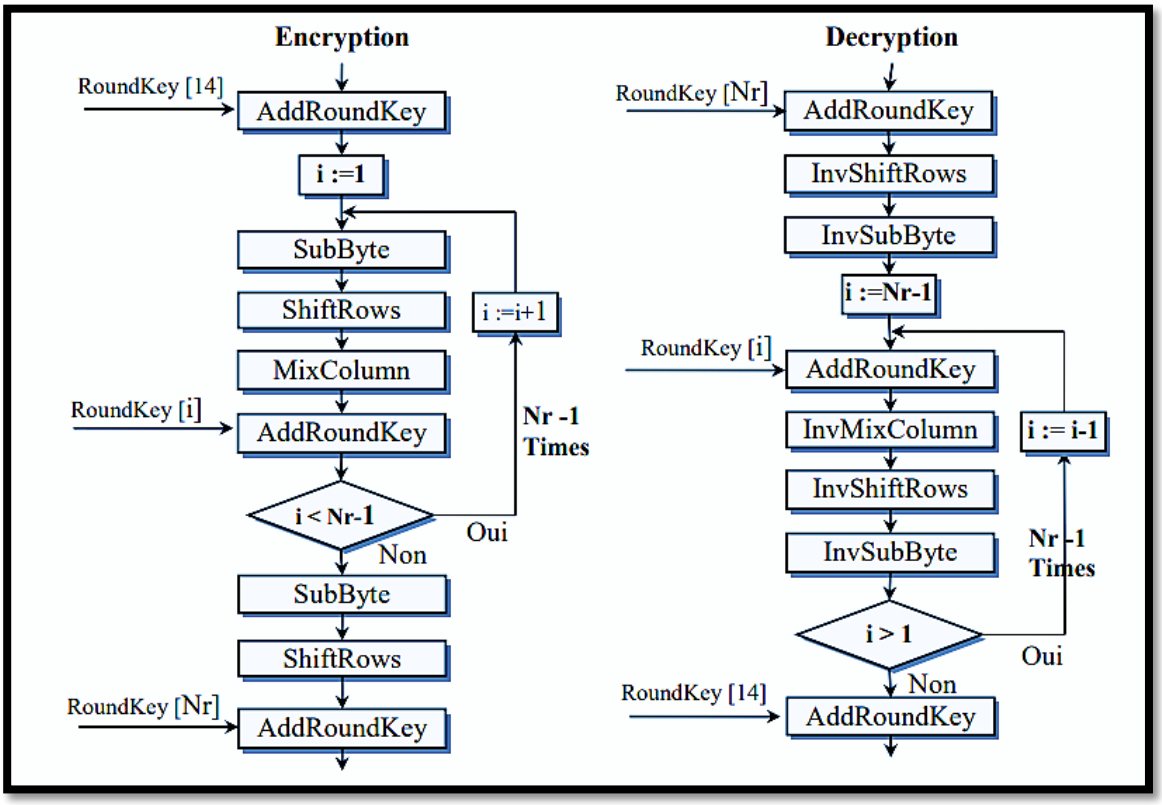


Figure III.3. *L'organigramme général des différentes étapes* [29].

III.2.3.1. Propriété des transformations :

Le Rijndael a été conçu de manière à ce que les étapes SubBytes et ShiftRows puissent être échangées entre elles sans affecter le résultat du chiffrement. Cette remarque s'applique également lors du déchiffrement, ce qui signifie que l'ordre dans lequel ces étapes sont appliquées peut être permuté sans altérer le résultat final de la décryption [29].

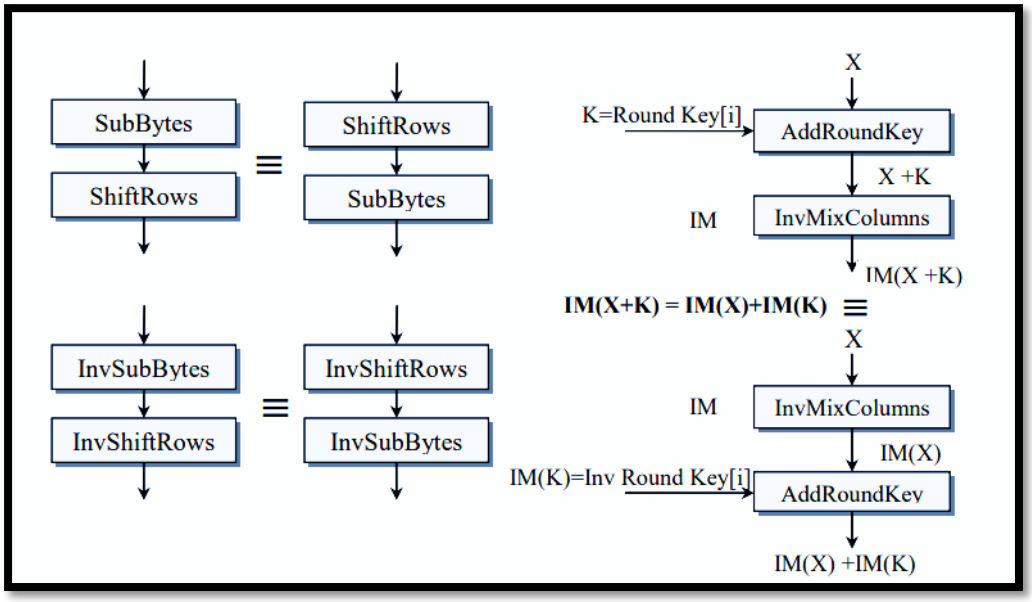


Figure III.4. *Particularité des transformations* [29].

III.2.3.2. Description de l'architecture (AES-128, 192, 256) :

Le chiffrement et le déchiffrement de Rijndael se fait par Nr tours (10,12 ou 14) pour un bloc (message) de taille 128 bits et une clé de tour à une longueur (128,192 ou 256) bits, à chaque ronde, quatre transformations sont appliquées [29]:

1. Substitution d'octets dans le tableau de message.

2. Décalage des rangées dans le tableau de message.

3. Mélange des colonnes dans le tableau de message (sauf à la dernière ronde) .

4. Addition d'une "clef de ronde" qui varie à chaque ronde.

Le schéma d'un seul tour est illustré dans la figure (III.5) *:*

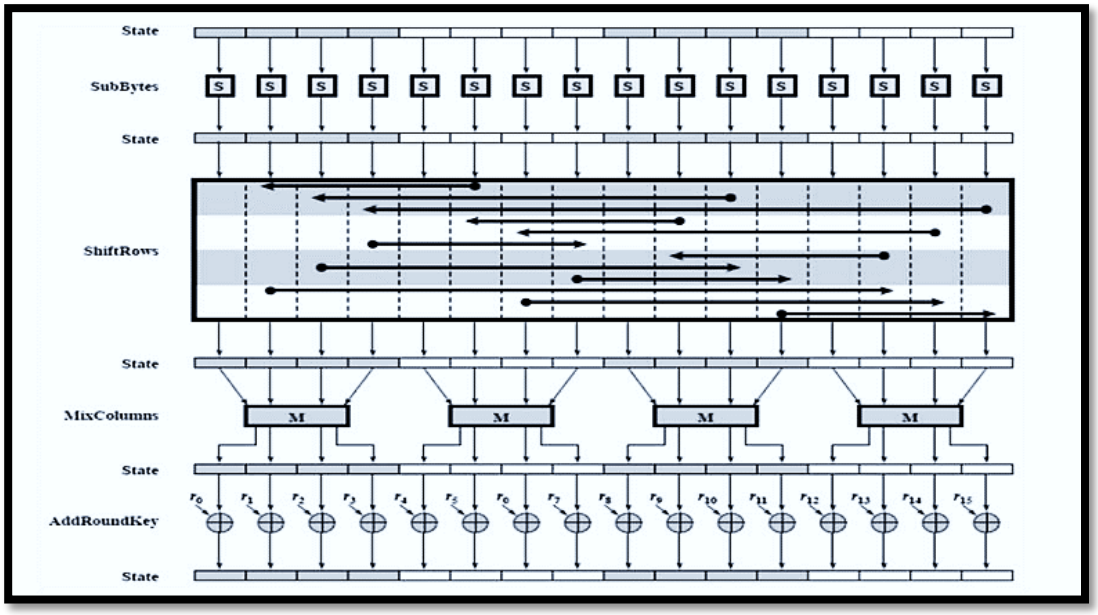


Figure III.5.*Schéma des étapes d'un seul tour.*

Afin de simplifier les opérations de transformation et d'augmenter la vitesse de chiffrement et de déchiffrement, le message à chiffrer ainsi que la clé de chiffrement sont stockés sous forme de tableaux. Un exemple de tableau est illustré dans la figure (III.6) pour un bloc de 128 bits. Le nombre de colonnes dans le tableau dépend à la fois de la taille du bloc de données à chiffrer et de la longueur de la clé de chiffrement. Pour les clés de 192 et 256 bits, le nombre de colonnes dans le tableau sera différent.

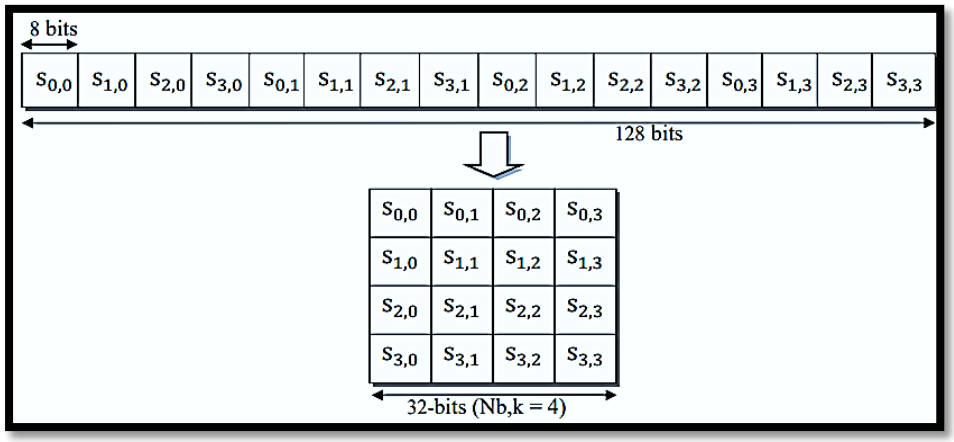


Figure III.6.*Transformation d'un bloc à une table.*

III.2.3.3. Chiffrement (Encryption) [30] *:*

a. La substitution (S-Box /SubBytes) :

La transformation de ***SubBytes*** (Figure III.7) est une substitution non linéaire d'un bloc de 8-bits (byte) qui fonctionne indépendamment sur chaque byte de bloc en utilisant une table de substitution (boîte de substitution). Cette boîte de substitution (figure III.8), qui est inversible, une seule boîte est suffisante pour toute la phase de chiffrement.

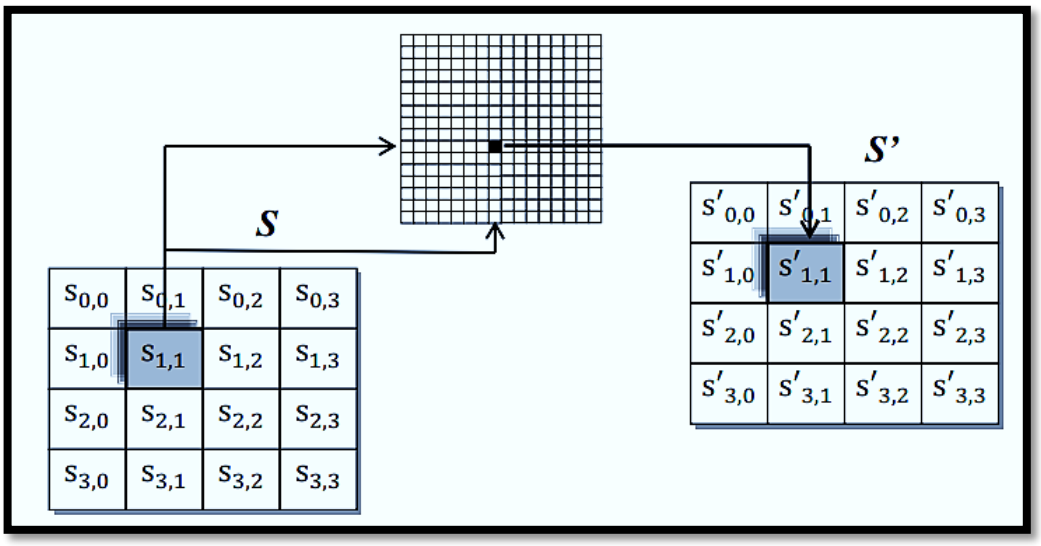
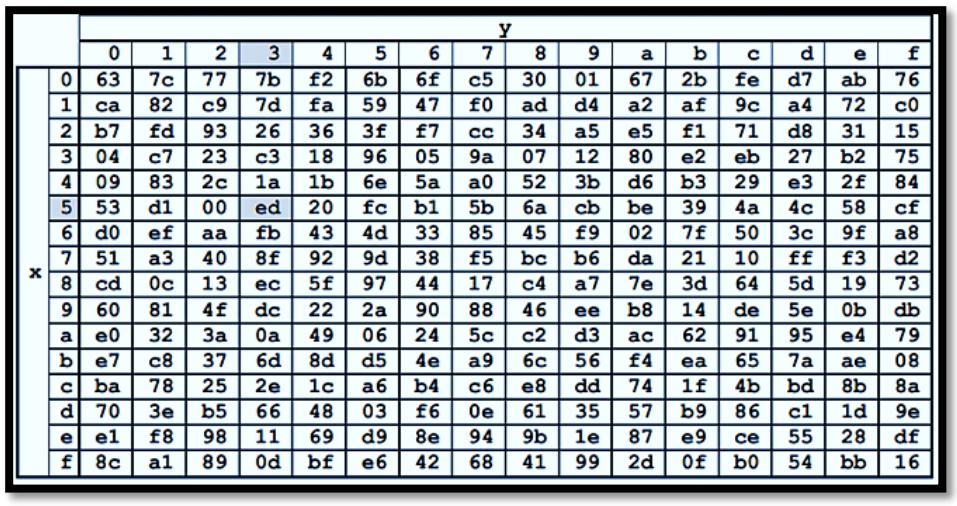


Figure III.7. *la transformation SubBytes.*

La table de substitution utilisé dans le ***SubBytes*** est représenté en Hexadécimal, voir la (figure III.8), Par exemple pour $s_{1,1}$ = {53}, donc leur substitution est déterminée par l'intersection de la ligne d'indice '5' et la colonne d'indice '3', le résultat de la substitution obtenu est $s'_{1,1}$={ed}.



x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

Figure III.8. *La table de substitution utilisé dans le SubBytes.*

b. Le décalage de rangées (ShiftRows) :

La transformation ***ShiftRows*** (figure III.9) consiste en un décalage cyclique vers la gauche des octets des trois dernières lignes de la table (message) La première ligne (r = 0) n'est pas décalée. Ainsi, les octets de la deuxième ligne sont décalés d'une position vers la gauche, les octets de la troisième ligne sont décalés de deux positions vers la gauche et les octets de la quatrième ligne sont décalés de trois positions vers la gauche [28].

Spécifiquement, la transformation de ***ShiftRows*** procède comme suit :

$$s'_{r,c} = s_{r,(c-r) \bmod Nb}, \qquad avec \ (0 \le r \le 3 \ et \ 0 \le c < Nb)$$

Où ***Nb*** = 4 pour AES-128

La figure (III.9), illustre la transformation ***ShiftRows*** :

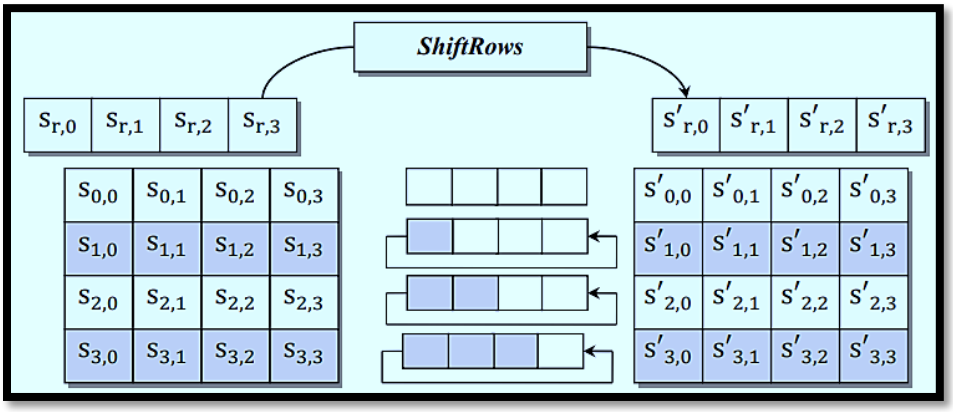


Figure III.9. *la transformation ShiftRows* [28].

c. Mélange des colonnes (MixColumns) :

Une différence sur 1 byte d'entrée se propage sur les 4 bytes de sortie. On a donc encore une étape de diffusion. La matrice utilisée est définie par Rijndael. Elle contiendra toujours ces valeurs Figure (III.10) :

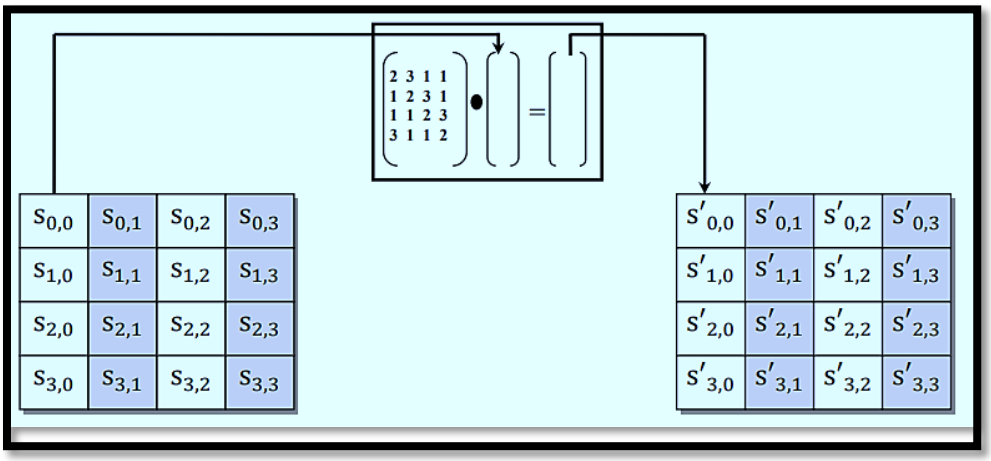


Figure III.10. *la transformation MixColumns.*

En raison de cette multiplication, les quatre bytes dans une colonne sont remplacés par ce qui suit :

$$\begin{pmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{pmatrix} = \begin{pmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{pmatrix} \bullet \begin{pmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{pmatrix} \quad \text{Pour } 0 \leq c \leq Nb \quad \text{.....(I.2)}$$

$$S'_{0,c} = (\{02\}\ S_{0,c}) \oplus (\{03\} \bullet S_{1,c}) \oplus S_{2,c} \oplus S_{3,c}$$

$$S'_{1,c} = S_{0,c} \oplus (\{02\} \bullet S_{1,c}) \oplus (\{03\} \bullet S_{2,c}) \oplus S_{3,c}$$

$$S'_{2,c} = S_{0,c} \oplus S_{1,c} \oplus (\{02\} \bullet S_{2,c}) \oplus (\{03\} \bullet S_{2,c}) \quad \text{.....(I.3)}$$

$$S'_{3,c} = (\{03\} \bullet S_{0,c}) \oplus S_{1,c} \oplus S_{2,c} \oplus (\{02\} \bullet S_{3,c})$$

Figure III.11. *Remplacement des quatre bytes dans une colonne.*

- ***Remarque :*** la multiplication polynomiale (●) est définie sur le champ fini GF(2^8).

d. Addition d'une clé de ronde (AddRoundKey) :

Dans la transformation ***AddRoundKey***, une clé de ronde (pour chaque ronde il y'a une clé différente) est ajoutée au message par une opération Ou-Exclusif (XOR) au niveau du bit.

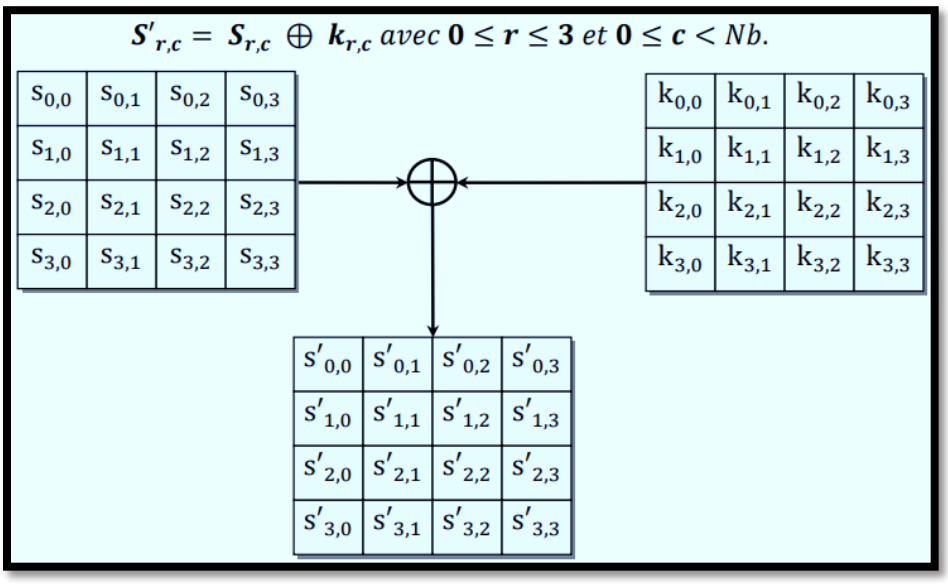


Figure III.12. *l'étape AddRoundKey.*

e. Génération (extension) des clés :

La clé étendue est un tableau à une dimension de mots de 32 bits noté **W[Nb*(Nr+l)]** dont les **Nk** premiers mots sont la clé principale Key0. Les autres mots sont obtenus récursivement à partir des mots précédents. La fonction d'expansion de la clé dépend de la valeur de **Nk.** Il y a en fait deux fonctions d'expansion : une lorsque **Nk < 6** et une autre lorsque **Nk ≥ 6**. Chaque ronde consiste une clé qui est composé de 4 mots de 32 bits cette clé est générée comme montre la figure (III.13) :

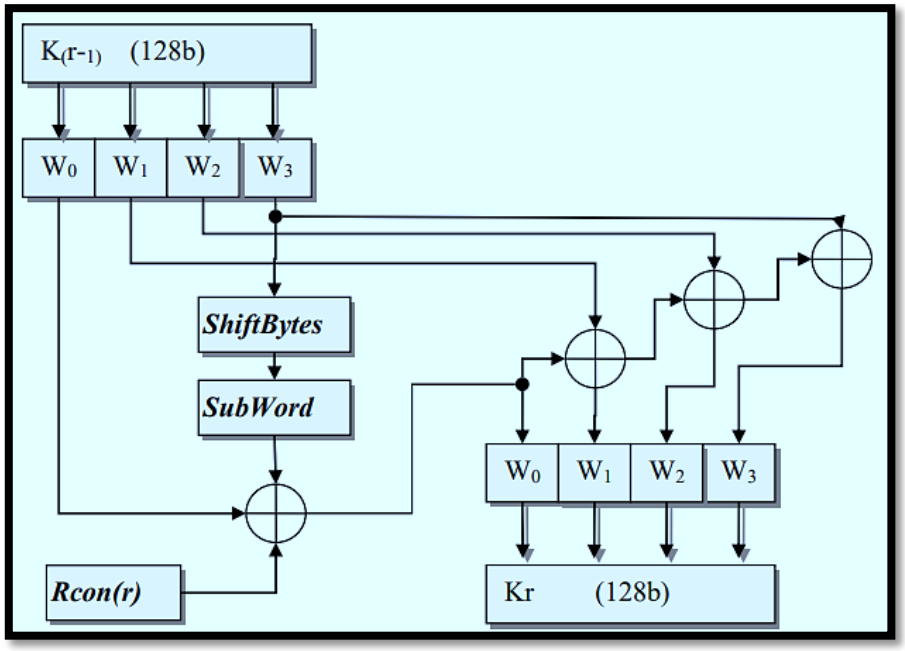


Figure III.13. *Le schéma général d'extension d'une clé.*

- ***SubWord :*** c'est une substitution de 32 bits des données qui est composée de quatre *SubBytes* en parallèle.

- ***ShiftBytes :*** c'est une transformation de décalage à gauche des bytes (8-bits) des données, etson fonctionnement est comme suite :

ShiftBytes(a,b,c,d) = (b,c,d,a), avec **a**, **d**, **c** et **d** de 8-bits.

Rcon(r) : c'est un vecteur de Nr cases chaque case est coder en hexadécimale la valeur de la case r est ajoute à la sortie de SubBytes avec l'opération xor.

R	1	2	3	4	5	6	7	8	9	10
Rcon[r]	0x01	0x02	0x04	0x08	0x10	0x20	0x40	0x80	0x1b	0x36

Figure III.14. *La table Rcon utilisée dans l'extension des clés.*

La règle de construction de cette table est :

Rcon(1) = 1 et Rcon(r) = 2* Rcon(r-1) **;** **(**pour 1 < **r** ≤ 10)

III.2.3.4. Déchiffrement (Decryption) [30] *:*

Rijndael a été conçu pour fonctionner de la même manière au chiffrement qu'au déchiffrement en remplaçant chaque opération élémentaire par son inverse, l'ordre du déchiffrement est l'inverse du chiffrement, les clés additionnées sont les mêmes du chiffrement, mais son ordre est décroissant [31].

Les différentes transformations utilisées dans le déchiffrement sont : ***InvShiftRows, InvSubBytes, InvMixColumns,*** et ***AddRoundKey.***

a. La substitution inverse (InvSubBytes):

C'est la même opération que ***SubBytes***, seulement la table de substitution qui change.

La table de substitution utilisé dans le ***InvSubBytes*** est représenté en Hexadécimal. Par exemple pour $a_{1,1} = \{ed\}$, donc leur substitution est déterminée par l'intersection de la ligne d'indice '5' et la colonne d'indice '3', le résultat de la substitution obtenu est $a'_{1,1} = \{53\}$.

		y															
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
	1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
	2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
	3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
	4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
	5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
	6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
	7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
	8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
	9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
	a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
	b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
	c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
	d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
	e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
	f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure III.15. *La table de substitution utilisé dans le InvSubBytes.*

b. La décalage inverse de rangées (InvShiftRows) :

Dans la transformation de ***InvShiftRows***, les bytes dans les trois dernières lignes de tableau de donnée sont cycliquement décalés à droite. La première ligne, r = 0, n'est pas décalée. Spécifiquement, la transformation de ***InvShiftRows*** procède comme suit :

$$S'_{r,c} = S_{r,(c-r+Nb) \bmod Nb} \quad , \qquad avec \ \ (0 \le r \le 3 \ \ et \ 0 \le c < Nb)$$

Où ***Nb*** = 4 pour AES-128

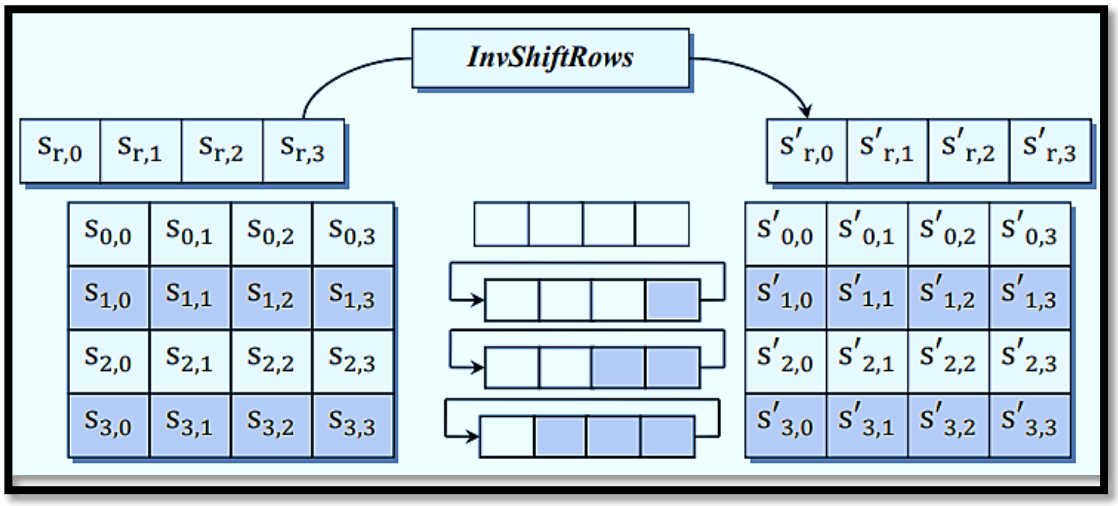


Figure III.16. *la transformation InvShiftRows.*

c. Mélange inverse de colonnes (InvMixColumns) :

Cette opération est différente de ***MixColumns***, dans la matrice de multiplication, cette matrice est illustrée dans les équations suivantes :

$$\begin{pmatrix} S'_{0,c} \\ S'_{1,c} \\ S'_{2,c} \\ S'_{3,c} \end{pmatrix} = \begin{pmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{pmatrix} \bullet \begin{pmatrix} S_{0,c} \\ S_{1,c} \\ S_{2,c} \\ S_{3,c} \end{pmatrix} \text{ Pour } 0 \leq c \leq Nb$$

$$S'_{0,c} = (\{0e\} \bullet S_{0,c}) \oplus (\{0b\} \bullet S_{1,c}) \oplus (\{0d\} \bullet S_{2,c}) \oplus (\{09\} \bullet S_{3,c})$$

$$S'_{1,c} = (\{09\} \bullet S_{0,c}) \oplus (\{0e\} \bullet S_{1,c}) \oplus (\{0b\} \bullet S_{2,c}) \oplus (\{0d\} \bullet S_{3,c}) \dots$$

$$S'_{2,c} = (\{0d\} \bullet S_{0,c}) \oplus (\{09\} \bullet S_{1,c}) \oplus (\{0e\} \bullet S_{2,c}) \oplus (\{0b\} \bullet S_{3,c})$$

$$S'_{3,c} = (\{0b\} \bullet S_{0,c}) \oplus (\{0d\} \bullet S_{1,c}) \oplus (\{09\} \bullet S_{2,c}) \oplus (\{0e\} \bullet S_{3,c})$$

Figure III.17. *les équations de la matrice de multiplication.*

- ***Remarque :***

- L'opération **AddRoundKey,** avec son extension de ses clés sont identiques que dans chiffrement.
- Dans le déchiffrement les clés sont ajoutées de manière décroissante (de clé 10 vers la clé 0), c'est le contraire du chiffrement.

III.2.4. Optimisation de AES :

III.2.4.1. GCM (Galois/Counter Mode) :

III.2.4.2. Définition :

Le GCM (Galois/Counter Mode) est un mode de chiffrement qui combine le chiffrement de compteur (CTR) avec l'authentification des données pour fournir à la fois la confidentialité et l'intégrité des données. Il est souvent utilisé avec l'algorithme de chiffrement symétrique AES (Advanced Encryption Standard) [32].

III.2.4.3. Principe de fonctionement (GCM) :

Le GCM utilise un compteur qui est incrémenté pour chaque bloc de données à chiffrer. Ce compteur, combiné à un vecteur d'initialisation (IV), est entré dans le chiffrement par bloc (comme AES) pour générer un flux de clés. Ce flux de clés est ensuite combiné avec les blocs de texte en clair en utilisant une opération de XOR (OU exclusif) pour produire les blocs de texte chiffré correspondants [33].

En parallèle, le GCM effectue également une opération d'authentification des données pour garantir l'intégrité et l'authenticité des données. Il utilise une multiplication de champ de Galois, basée sur l'arithmétique polynomiale dans un champ de Galois, pour générer un code d'authentification (tag) qui est ajouté au texte chiffré.

Le GCM nécessite un vecteur d'initialisation unique pour chaque opération de chiffrement et une clé secrète partagée entre l'expéditeur et le destinataire. Il prend également en charge l'inclusion de données supplémentaires non chiffrées, appelées "Données Authentifiées Supplémentaires" (AAD), qui sont incluses dans le calcul du code d'authentification.

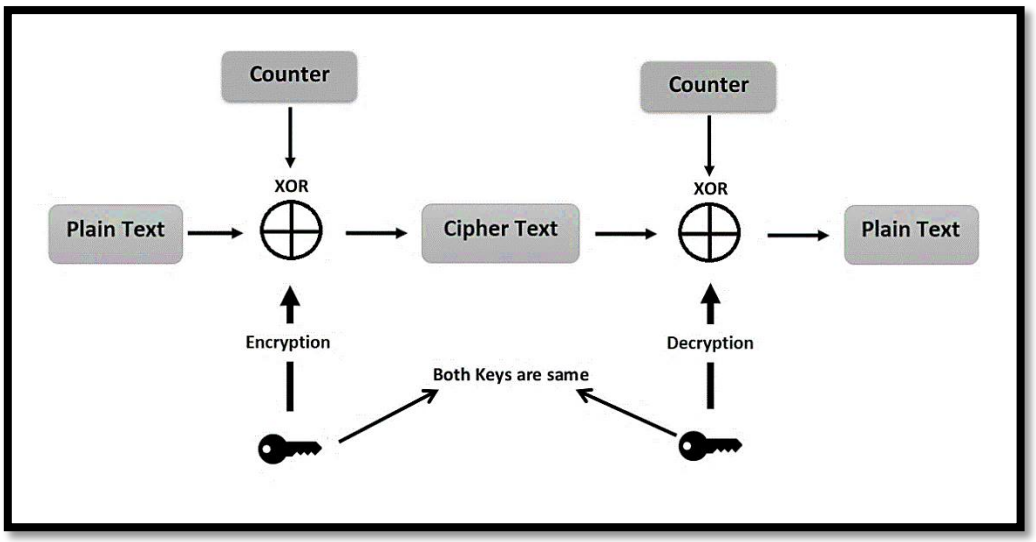


Figure III.18. *Java AES GCM Encryption and Decryption.*

- **Notre algorithme proposé fonctionne comme suit :**

Chiffrement (Encryption Process):

```
Function Encrypt (data, key) {

  // Generate a random 12-byte IV.
  // This is a unique value that is used to encrypt the data.
  // It is important to generate a new IV for each encryption operation.
  var iv = new SecureRandom().nextBytes(12);

  // Create a GCMParameterSpec object with the given tag length and IV.
  // The tag length is the size of the authentication tag that is generated by the cipher.
  // The IV is the value that was generated in the previous step.
  var gcmSpec = new GCMParameterSpec(128, iv);

  // Initialize the cipher in encryption mode with the given key and GCMParameterSpec.
  // The cipher is a cryptographic algorithm that is used to encrypt and decrypt data.
  // The key is the value that is used to encrypt the data.
  // The GCMParameterSpec object specifies the parameters that are used by the cipher.
  var cipher = Cipher.getInstance("AES/GCM/NoPadding");
  cipher.init(Cipher.ENCRYPT_MODE, key, gcmSpec);

  // Encrypt the data and return the encrypted data and IV as a byte array.
  // The cipher is used to encrypt the data.
  // The IV is used to authenticate the encrypted data.
  // The encrypted data and IV are returned as a byte array.
  var encrypted = cipher.doFinal(data);
  var result = new byte[iv.length + encrypted.length];
  System.arraycopy(iv, 0, result, 0, iv.length);
  System.arraycopy(encrypted, 0, result, iv.length, encrypted.length);
  return result;
}
```

L'algorithme que nous avons fourni est une implémentation du chiffrement des données à l'aide du mode AES-GCM (Advanced Encryption Standard-Galois/Counter Mode). voici une description détaillée de chaque étape du processus de chiffrement :

1. Génération d'un IV aléatoire :

- Un IV (vecteur d'initialisation) est une valeur unique et aléatoire utilisée dans les algorithmes de chiffrement.

- Dans cette implémentation, un IV de 12 octets est généré à l'aide d'un générateur SecureRandom. La classe SecureRandom fournit un générateur de nombres aléatoires cryptographiquement sécurisé.

- L'IV est crucial pour la sécurité du processus de chiffrement, et un nouvel IV doit être généré pour chaque opération de chiffrement.

2. Création d'un objet GCMParameterSpec :

- L'objet ***GCMParameterSpec*** est utilisé pour configurer le chiffrement AES-GCM.

- Il nécessite deux paramètres : la longueur de l'étiquette (tag) et l'IV.

- La longueur de l'étiquette spécifie la taille de l'étiquette d'authentification générée par le chiffreur. Dans ce cas, elle est définie sur 128 bits (16 octets).

- L'IV est la valeur générée aléatoirement à partir de l'étape précédente.

3. Initialisation du chiffreur :

- Le chiffreur est initialisé en mode chiffrement avec la clé et le ***GCMParameterSpec*** donnés.

- La clé est une valeur secrète utilisée pour chiffrer et déchiffrer les données. Elle doit être générée de manière sécurisée et partagée entre les parties qui chiffrent et déchiffrent les données.

- L'instance du chiffreur est créée en utilisant la transformation ***"AES/GCM/NoPadding",*** qui spécifie l'algorithme de chiffrement AES-GCM sans aucun remplissage supplémentaire.

4. Chiffrement des données :

- La méthode ***`doFinal`*** du chiffreur est appelée avec le paramètre ***`data`,*** qui représente le texte en clair à chiffrer.

- Le chiffreur effectue le chiffrement en utilisant la clé, l'IV et les données fournies.

- Les données chiffrées résultantes sont renvoyées sous forme d'un tableau d'octets.

5. Combinaison de l'IV et des données chiffrées :

- L'IV et les données chiffrées sont concaténés dans un seul tableau d'octets pour le résultat final.

- Un nouveau tableau d'octets ***`result`*** est créé avec une taille égale à la somme des longueurs de l'IV et des données chiffrées.

- La méthode ***`System.arraycopy`*** est utilisée pour copier l'IV et les données chiffrées dans le tableau ***`result`.***

6. Renvoi du résultat :

- Le tableau d'octets ***`result`,*** qui contient l'IV suivi des données chiffrées, est renvoyé en tant que sortie de la fonction de chiffrement.

Déchiffrement (Decryption Process) :

```
Function Decrypt (encrypted, key) {

  // Get the IV from the encrypted data.
  // The IV is a 12-byte value that is used to decrypt the data.
  // It is important to use the same IV that was used to encrypt the data.
  var iv = encrypted.slice(0, IV_LENGTH_BYTES);

  // Create a GCMParameterSpec object with the given tag length and IV.
  // The tag length is the size of the authentication tag that was generated by the cipher.
  // The IV is the value that was extracted in the previous step.
  var gcmSpec = new GCMParameterSpec(TAG_LENGTH_BITS, iv);

  // Initialize the cipher in decryption mode with the given key and GCMParameterSpec.
  // The cipher is a cryptographic algorithm that is used to encrypt and decrypt data.
  // The key is the value that is used to decrypt the data.
  // The GCMParameterSpec object specifies the parameters that are used by the cipher.
  var cipher = Cipher.getInstance("AES/GCM/NoPadding");
  cipher.init(Cipher.DECRYPT_MODE, key, gcmSpec);

  // Decrypt the data and return the decrypted data.
  // The cipher is used to decrypt the data.
  // The decrypted data is returned.
  var decrypted = cipher.doFinal(encrypted.slice(IV_LENGTH_BYTES));
  return decrypted;
}
```

1. Obtention de l'IV à partir des données chiffrées :

- L'IV (vecteur d'initialisation) est extrait des données chiffrées.

- Dans le processus de chiffrement, l'IV était préfixé aux données chiffrées.

- En découpant les données chiffrées depuis le début jusqu'à la longueur de l'IV, nous pouvons extraire l'IV.

2. Création d'un objet GCMParameterSpec :

- L'objet ***GCMParameterSpec*** est utilisé pour configurer le chiffreur AES-GCM lors du déchiffrement.

- Il nécessite deux paramètres : la longueur de l'étiquette (tag) et l'IV.

- La longueur de l'étiquette spécifie la taille de l'étiquette d'authentification générée lors du chiffrement. Elle doit correspondre à la valeur utilisée lors du chiffrement.

- L'IV est la valeur extraite à partir de l'étape précédente.

3. Initialisation du chiffreur :

- Le chiffreur est initialisé en mode déchiffrement avec la clé et le ***GCMParameterSpec*** donnés.

- La clé est la valeur secrète utilisée pour chiffrer et déchiffrer les données. Elle doit correspondre à la clé utilisée lors du chiffrement.

- L'instance du chiffreur est créée en utilisant la transformation ***"AES/GCM/NoPadding",*** spécifiant l'algorithme de chiffrement AES-GCM sans aucun remplissage supplémentaire.

4. Déchiffrement des données :

- La méthode ***`doFinal`*** du chiffreur est appelée avec les données chiffrées moins l'IV.

- Le chiffreur effectue le déchiffrement en utilisant la clé, l'IV et les données chiffrées fournies.

- Les données déchiffrées résultantes sont renvoyées.

5. Renvoi des données déchiffrées :

- Les données déchiffrées sont renvoyées en tant que sortie de la fonction de déchiffrement.

III.2.4.4. La combinaison GCM / AES :

La combinaison de GCM (Galois/Counter Mode) et de l'algorithme AES (Advanced Encryption Standard) apporte plusieurs aspects positifs à la cryptographie et à l'authentification des données [32]:

- ✓ **Confidentialité :** AES est un algorithme de chiffrement largement reconnu et sécurisé. Lorsqu'il est combiné avec GCM, il garantit la confidentialité des données en les chiffrant à l'aide d'AES. Cela signifie que seules les parties autorisées disposant de la clé correcte peuvent décrypter et accéder aux informations d'origine.

- ✓ **Intégrité des données :** GCM offre un mécanisme solide de vérification de l'intégrité des données. Il utilise la fonction de hachage universelle GHASH pour générer des étiquettes d'authentification pour les données chiffrées. Ces étiquettes sont ensuite utilisées pour vérifier que les données n'ont pas été modifiées ou altérées pendant la transmission ou le stockage.

- ✓ **Efficacité :** GCM est conçu pour être très efficace, en particulier dans les environnements informatiques modernes. Il permet le traitement parallèle des données, ce qui peut exploiter simultanément plusieurs processeurs ou cœurs. Ce parallélisme permet un chiffrement et une authentification plus rapides, ce qui le rend adapté aux applications où les performances sont cruciales.

- ✓ **Chiffrement authentifié :** GCM combine le chiffrement et l'authentification dans un seul mode de fonctionnement. Cela signifie que l'intégrité des données est vérifiée en même temps qu'elles sont chiffrées. Avec GCM, vous pouvez garantir à la fois la confidentialité et l'authenticité des données avec une seule opération, ce qui simplifie le processus de chiffrement.

- ✓ **Applicabilité étendue :** GCM est largement pris en charge dans divers protocoles et applications, ce qui le rend hautement interopérable. Il est couramment utilisé dans les protocoles de sécurité réseau tels que IPsec et TLS, ainsi que dans les normes de sécurité Wi-Fi telles que WPA2 et WPA3. Cette large prise en charge et adoption facilitent l'intégration de GCM avec les systèmes existants et garantissent la compatibilité entre différentes plateformes.

- ✓ **Sécurité basée sur les nonces :** GCM nécessite l'utilisation de nonces, ce qui garantit que le même texte en clair chiffré avec la même clé produira des textes chiffrés différents. Cela protège contre les attaques visant à exploiter les schémas ou les répétitions dans le processus de chiffrement. En utilisant des nonces uniques pour chaque opération de chiffrement, GCM renforce la sécurité de AES.

GCM (Galois/Counter Mode) peut améliorer la vitesse d'exécution de l'algorithme AES (Advanced Encryption Standard) et présente généralement une faible utilisation de la mémoire. Voici comment GCM contribue à ces aspects :

- **Vitesse d'exécution :** GCM est conçu pour être hautement efficace et peut accélérer le processus de chiffrement et d'authentification lorsqu'il est utilisé avec AES. En combinant le chiffrement et l'authentification en une seule opération sur les données, GCM réduit le temps de traitement global par rapport aux modes de chiffrement et d'authentification séparés. De plus, GCM peut tirer parti des capacités de traitement parallèle, permettant à plusieurs processeurs ou cœurs de travailler sur le chiffrement de différentes parties des données simultanément. Cette parallélisation améliore la vitesse d'exécution de GCM-AES.

- **Utilisation de mémoire réduite :** GCM fonctionne de manière similaire à un flux, ce qui signifie que les données peuvent être traitées par petits morceaux sans nécessiter le chargement de l'intégralité du message en mémoire en une seule fois. Cette approche basée sur un flux réduit l'empreinte mémoire et permet un traitement efficace des données en temps réel ou à la volée. GCM nécessite généralement une quantité modeste de mémoire pour stocker la clé, le nonce et les données intermédiaires nécessaires au chiffrement et à l'authentification.

  L'utilisation de mémoire de GCM est généralement considérée comme raisonnable et bien optimisée.

  Dans l'ensemble, la combinaison de GCM et de l'algorithme AES offre une solution puissante et efficace pour le chiffrement et l'authentification sécurisés des données. Elle offre la confidentialité, l'intégrité des données, l'efficacité et une applicabilité étendue, ce qui en fait un choix populaire pour divers scénarios de communication et de stockage sécurisés.

III.3. Conclusion :

Dans ce chapitre en a présenter la mise en œuvre de l'algorithme AES (Advanced Encryption Standard) en combinaison avec GCM (Galois/Counter Mode) qui offre une solution puissante et efficace pour le chiffrement et l'authentification sécurisés des données. AES fournit un algorithme de chiffrement fiable et sécurisé, tandis que GCM ajoute le composant crucial de l'authentification, garantissant l'intégrité des données. Cette combinaison permet à la fois la confidentialité et l'authenticité des données chiffrées.

En résumé, la mise en œuvre d'AES avec GCM offre une solution robuste et efficace pour le chiffrement et l'authentification sécurisés des données, garantissant la confidentialité, l'intégrité des données et de hautes performances dans diverses applications.

CHAPITRE IV :

analyse et Résultat.

IV.1. Introduction :

Le chapitre final de notre mémoire met en évidence la phase cruciale de la réalisation et de l'analyse, où nous dévoilons notre application développée ainsi que les mesures exhaustives démontrant la sécurité et l'efficacité de l'algorithme AES (Advanced Encryption Standard) que nous avons proposé. Dans cette section, nous présentons en détail notre application, mettant en lumière sa conception et son fonctionnement. De plus, nous exposons les résultats obtenus à travers différentes évaluations et mesures de performance, démontrant ainsi l'efficacité et la robustesse de notre approche. En parallèle, nous nous penchons également sur les mesures de sécurité mises en place, révélant comment notre algorithme AES assure une protection optimale des données sensibles. Cette étape cruciale de réalisation et d'analyse représente donc une étape charnière de notre recherche, apportant des preuves tangibles et des résultats convaincants quant à la viabilité et à l'efficacité de notre proposition.

IV.2. Présentation de l'application développée :

Avant de présenter notre application dévloppée , nous présentons d'abord le matériel et le langage de programation utilisé.

IV.2.1. Environnement de développement :

Dans cette partie nous allons citer l'environnement logiciel (Software) et matériel (Hardware) utilisés.

IV.2.1.1. Environnement logiciel :

Nous avons utilisé l'environnement Eclipse (Java) pour l'implémentation de notre approche .

1. Eclipse :

Un environnement Eclipse, ou IDE Eclipse (Integrated Development Environment), est un outil logiciel utilisé par les développeurs pour écrire, tester et déboguer différents types d'applications logicielles. Il offre un ensemble complet de fonctionnalités et d'outils qui aident les développeurs à créer des logiciels de manière efficace.

L'IDE Eclipse est une plateforme open source qui prend en charge plusieurs langages de programmation tels que Java, C/C++, Python, etc., ce qui la rend polyvalente et largement utilisée dans différents domaines. Il propose un framework hautement personnalisable et extensible, permettant aux développeurs d'adapter leur environnement de développement en fonction de leurs besoins spécifique [34].

2. Java :

Java est un langage de programmation de haut niveau et orienté objet connu pour son indépendance de plateforme et son principe de "write once, run anywhere" ("écrivez une fois, exécutez partout"). Il est utilisé pour le développement de différentes applications et offre un ensemble riche de bibliothèques et de frameworks. Les programmes Java sont compilés en bytecode et exécutés par la machine virtuelle Java (JVM). Il prend en charge des fonctionnalités telles que la gestion automatique de la mémoire, le multithreading et bénéficie d'une large communauté de développeurs [35].

Nous avons choisi d'utiliser Java pour notre projet de cryptage de données dans les réseaux de capteurs sans fil en raison de sa portabilité, de sa sécurité, de ses bibliothèques spécialisées, de ses performances et de son support, qui nous permettent de développer et de mettre en œuvre efficacement et de manière fiable l'algorithme AES. De plus, Java offre des architectures de cryptographie telles que l'Architecture de Cryptographie (JCA) et l'Extension de Cryptographie Java (JCE), ce qui facilite le processus de programmation et nous permet de nous concentrer davantage sur l'idée du projet.

3. MATLAB :

MATLAB est un langage de programmation utilisé pour effectuer des calculs mathématiques, des manipulations de matrices, des simulations et des visualisations. Il est largement utilisé dans le domaine du calcul numérique et de l'analyse technique [36].

Nous avons utilisé le Matlab pour effectuer des opérations des analyses et des comparaisons.

IV.2.1.2. Environnement matériel :

L'application a été développée sur un PC (Laptop-0EVMP5B1) ayant les caractéristiques suivantes :

- **Processeur :** 12th Gen Intel(R) Core(TM) CPU i5-12405H 2.00GHz.
- **Mémoire :** installée (RAM) : 8,00 Go.
- **Disque Dure :** HDD 500 Go.
- **Carte graphique :** Intel r HD graphics5500.
- **Système d'exploitation :** Windows 11 Version 22H2 Professionnel 64 bit.

IV.2.2. L'outil d'optimisation :

GCM est un mode de chiffrement qui combine le chiffrement par blocs et l'authentification des données pour assurer la confidentialité et l'intégrité des informations. Il est couramment utilisé en cryptographie symétrique avec l'algorithme AES pour garantir la sécurité des données sensibles lors de leur stockage et de leur transmission sur des réseaux. GCM est reconnu pour sa haute sécurité, sa performance élevée et sa résistance aux attaques par canaux auxiliaires.

IV.2.3. Interface principal de l'application :

Notre application est simple et compréhensible dés sa première utilisation, elle Permet d'illustrer le processus de la sécurité des données (chiffrement / déchiffrement) entre deux capteurs sans fil selon notre projet.

Au démarrage de l'application, le système affiche l'interface présentée dans la figure (IV.1).

Figure IV.1. *l'interface principal de l'application.*

IV.2.3.1. L' interface AES GCM Algorithm :

L'interface « AES GCM Algorithm » présentée dans la figure (IV.2) et la figure (IV.3).

- **Avant le chiffrement :**

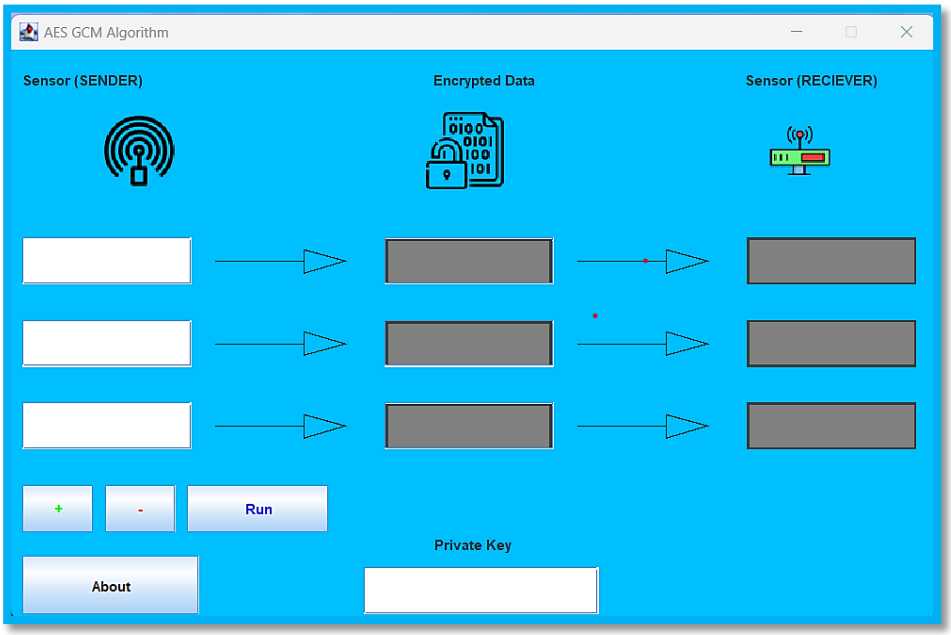


Figure IV.2. *L'interface AES GCM Algorithm avant le chiffrement.*

- **Après le chiffrement :**

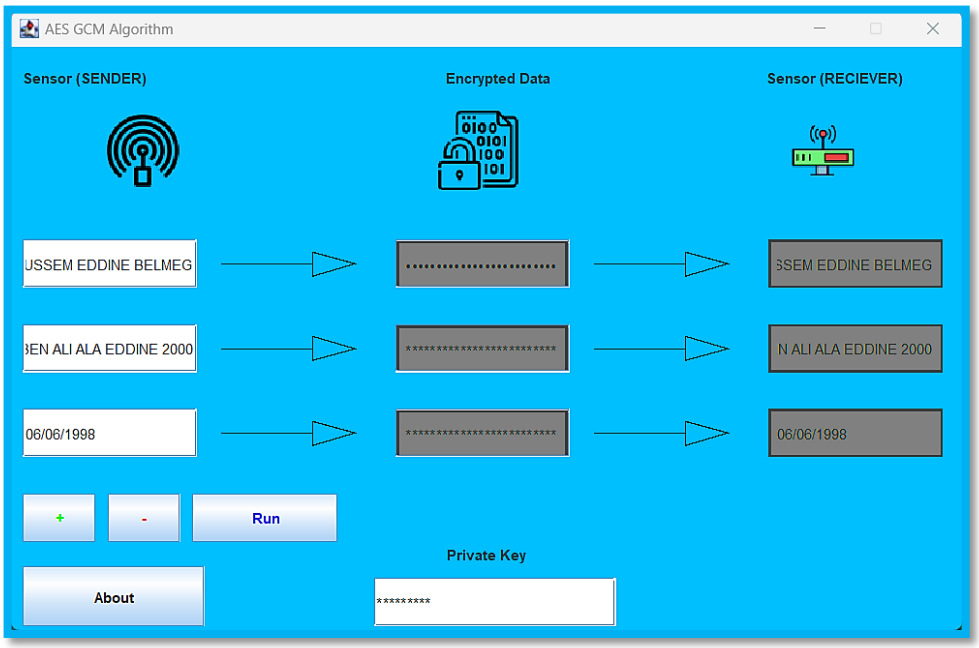


Figure IV.3. *L'interface AES GCM Algorithm après le chiffrement.*

IV.2.3.2. Méthode de travail de l'interface :

Pour expliquer la méthode de travail de l'interface « AES GCM Algorithm » , nous l'avons divisée en trois zones (zone d'outil, zone d'affichage, zone d'information).

- **Zone d'outil :**

  La Zone d'outil contient quatre bouttons :

Nom de boutton	Rôle
+	Pour ajouter plus des nœuds.
-	Pour diminuer le nombre des nœuds
Run	Pour de mettre en œuvre le processus de cryptage.
About	Pour revenir à l'interface principale.

Tableau IV.1. La zone d'outil**.**

- **Zone d'affichage :**

  La zone d'affichage est composée de deux espaces « Encrypted data » et « sensor (RECEIVER) ».

Nom de l'espace	Rôle
Encrypted data	Pour afficher le donné cryptée.
Sensor (RECEIVER)	Pour affiche le donnée après décryptée.

Tableau IV.2. *La zone d'affichage.*

- **Zone d'information :**

Cette zone contient deux espaces « sensor (SENDER)» et « Private key ».

Nom de l'espace	Rôle
Sensor (SENDER)	Pour entrez les données que nous voulons Cryptées.
Private key	Pour entre la clé de cryptage

Tableau IV.3. *La zone d'information.*

- **Remarque :**

Les données cryptées et les données décryptée et la clé de cryptage peuvent être montrées en touchant les espaces de la zone d'affichage.

IV.3. Résultat et Analyse :

Dans cette section, les performances de l'algorithme proposé seront discutées à travers les résultats obtenus. De plus, certains types de tests seront utilisés pour montrer la supériorité de la méthode de cryptage proposée. Les quantités à mesurer sont : l'analyse espace de clé, vitesse, taille

mémoire, méthode de cryptage. nous avons aussi comparé notre algorithme avec d'autres algorithmes AES.

IV.3.1. Analyse de l'espace clé :

Pour un bon système de chiffrement, la taille de l'espace clé doit être suffisamment grande pour rendre impossible le succès des attaques par force brute impossible, **l'espace clé doit être supérieur à 2^{100} ,** Pour notre projet, nous avons utilisé un algorithme AES 256 bits .

Dans le cas spécifique de l'AES-256 GCM, l'espace de clé correspond à toutes les combinaisons possibles de 256 bits. Un bit peut prendre l'une des deux valeurs (0 ou 1), ce qui signifie qu'il y a 2^256 (environ 1,16 * 10^77) combinaisons différentes de bits pour former une clé.

Pour donner une idée de la taille de l'espace de clé, 2^256 est un nombre extrêmement grand. En fait, il est estimé qu'il y a plus de combinaisons possibles de clés AES-256 que le nombre approximé d'atomes dans l'univers observable. Cela rend l'AES-256 extrêmement résistant aux attaques par force brute, où un attaquant essaierait toutes les clés possibles jusqu'à trouver la bonne.

IV.3.2. Analyse de vitesse :

Pour évaluer les performances de notre algorithme, il existe un facteur qui influence la mesure de la vitesse de chiffrement : le temps. Nous avons utilisé MATLAB comme environnement expérimental. Le temps d'exécution du chiffrement est illustré dans la figure (IV.4), qui démontre les excellentes performances de vitesse de notre algorithme AES proposé. Par conséquent, il peut être utilisé efficacement dans un réseau de capteurs sans fil.

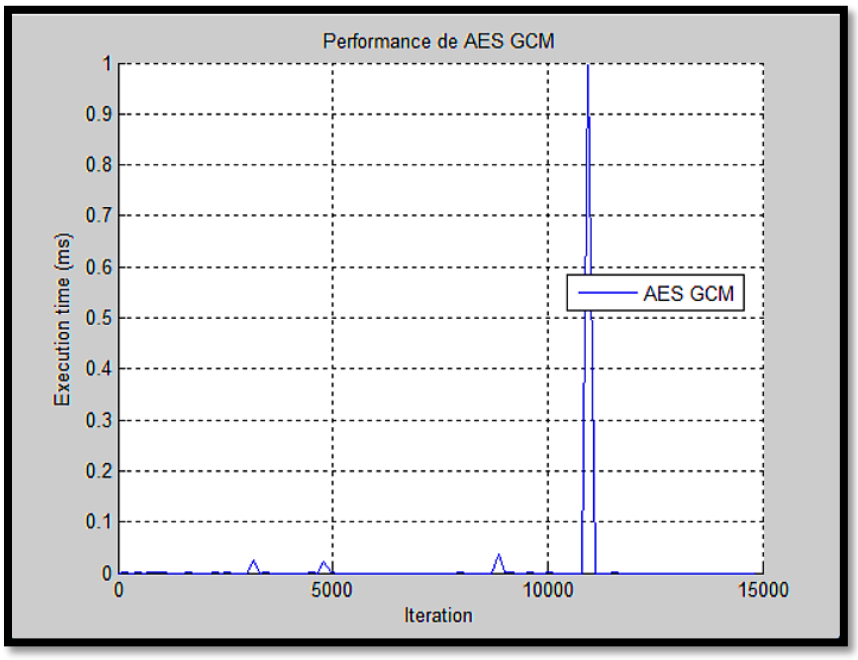


Figure IV.4. *Analyse de vitesse.*

En outre, la figure (IV.5) compare le temps d'exécution entre notre algorithme et un autre algorithme AES, démontrant que notre algorithme est plus performant et nécessite moins de temps d'exécution. Cela peut être attribué à l'optimisation de notre algorithme grâce à l'utilisation de GCM.

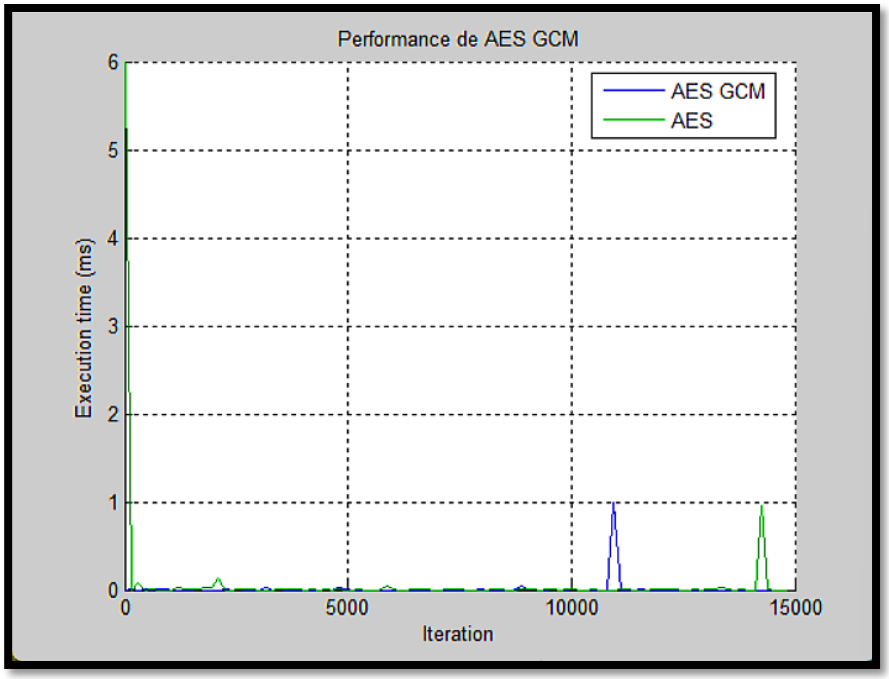


Figure IV.5. *Comparaison du temps d'exécution de notre algorithme avec un autre algorithme AES.*

IV.3.3. Analyse de taille de mémoire :

Dans la figure (IV.5), nous comparons la taille mémoire utilisée par notre algorithme avec celle d'autres algorithmes AES. Les résultats montrent que notre algorithme utilise le moins d'espace mémoire parmi tous les algorithmes répertoriés dans la figure. Cela est dû à l'utilisation de la méthode GCM pour optimiser notre algorithme. Ainsi, la faible empreinte mémoire de notre algorithme le rend adapté à une utilisation dans un capteur sans fil.

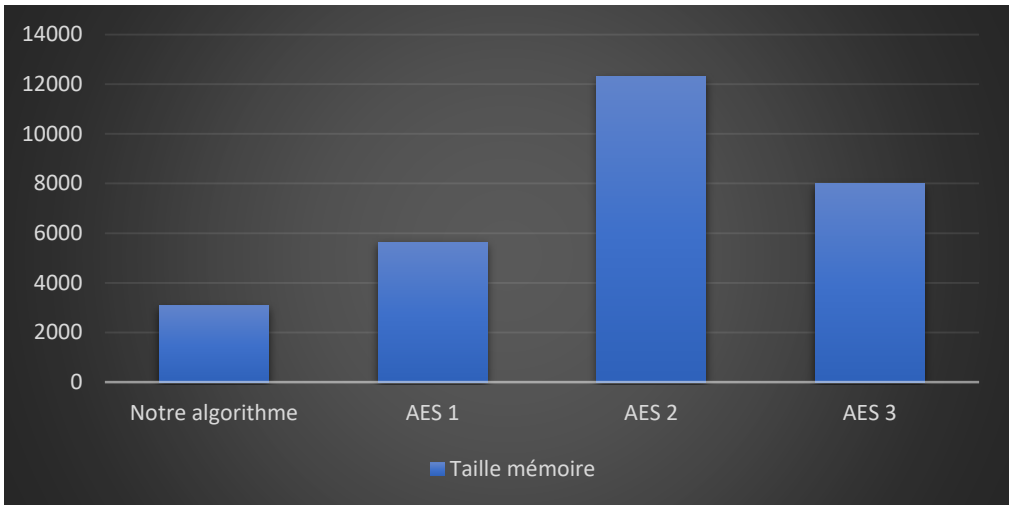


Figure IV.6. *Analyse de taille de mémoire.*

IV.3.4. Analyse de méthode de cryptage :

Il est important de noter que la sécurité dans les RCSF est un domaine complexe et nécessite une approche globale. Outre le chiffrement des données avec AES, d'autres aspects tels que l'authentification des nœuds, la gestion des clés et la protection contre les attaques spécifiques aux RCSF doivent également être pris en compte pour assurer un niveau de sécurité adéquat. Nous présentons dans le tableau (IV.4) un exemple de la méthode de cryptage utilisée dans notre algorithme.

Sensor(SENDER)	Private key	Encrypted data	Sensor(RECEIVER)
Houssem Eddine Belmeguenai	AppMaster23	**********	Houssem Eddine Belmeguenai
aefrrbfumfveoioyfn	AppMaster23	**********	aefrrbfumfveoioyfn
06061998	AppMaster23	**********	06061998

Tableau IV.4. *Exemple de la méthode de cryptage de notre algorithme.*

IV.3.5. Résultat :

Selon les analyses précédentes et les résultats positifs obtenus, nous avons démontré que notre algorithme AES offre une sécurité élevée contre les attaques par force brute, ainsi qu'une performance de vitesse rapide avec des temps d'exécution courts. De plus, il utilise un espace mémoire réduit.

En conclusion, notre algorithme AES est adapté à une utilisation dans les réseaux de capteurs sans fil.

IV.4. Conclusion :

Nous avons présenté dans ce chapitre notre application réalisée, et les différents outils nécessaires utilisés dans son développement. Nous avons aussi terminé par un ensemble d'analyses et tests pour les résultats expérimentaux qui montrent la résistance de notre algorithme de chiffrement.

Conclusion Générale.

Conclusion Générale.

Les réseaux de capteurs sans fil (RCSF) jouent un rôle essentiel dans divers domaines, mais leur déploiement dans des environnements non sécurisés les expose à des risques potentiels, tels que les attaques et les interceptions de données sensibles. La sécurité des données transmises dans les RCSF est donc une préoccupation majeure pour garantir le bon fonctionnement des applications déployées et prévenir les fuites d'informations confidentielles.

Dans ce travail de master, nous nous sommes attaqués à ce défi en concevant et en implémentant un algorithme de sécurité novateur spécifiquement conçu pour les RCSF. Notre approche s'appuie sur des techniques cryptographiques avancées telles que le chiffrement symétrique, les fonctions de hachage cryptographique et l'authentification des nœuds et des données. Nous avons également pris en compte les contraintes spécifiques des RCSF, telles que les ressources limitées des nœuds capteurs, afin de minimiser l'impact sur ces ressources tout en garantissant une protection adéquate des données.

Les résultats de notre évaluation ont démontré l'efficacité de notre algorithme en termes de confidentialité, d'intégrité et d'authentification des données transmises dans les RCSF. Nous avons également pris en compte des mesures de performance telles que la vitesse et la consommation de ressources, montrant ainsi que notre approche parvient à sécuriser les données tout en minimisant l'impact sur les ressources limitées des nœuds capteurs.

En conclusion, notre étude apporte une contribution significative à la sécurité des RCSF en proposant un nouvel algorithme de sécurité spécifiquement adapté à ces réseaux. Notre approche offre une protection robuste des données sensibles tout en prenant en compte les contraintes des RCSF. Cependant, il reste des pistes de recherche à explorer, telles que l'optimisation de la consommation d'énergie ou l'adaptation de notre algorithme à des scénarios spécifiques, Détection et réponse aux attaques avancées, Intégration de l'intelligence artificielle, Ces perspectives de travail offrent de nouvelles opportunités pour améliorer davantage la sécurité des RCSF et garantir leur utilisation sûre et fiable dans les domaines de la surveillance environnementale, des systèmes de santé, de la gestion de l'énergie et au-delà.

BIBLIOGRAPHIE :

[1]: A. Beghriche et A. Bilami , Un modèle de confiance pour l'authentification dans un réseau sans-fil Ad hoc, Journées Ecole Doctorale & Réseaux de Recherche en Sciences et Technologies de l'Information JED'08, Université Annaba, Juin 2008

[2]: Philippe Berger, Les capteurs, Support de cours, 2010.

[3]: Yaser Youcef, Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil. Computer Science. Université de Haute Alsace - Mulhouse, 2010.

[4]: Akyildiz, I. F.Su, W.Sankarasubramaniam, Y. and Cayirci, E. « Wireless sensor networks » a survey. Computer Networks 38: 393-422, 2002.

[5]: Louise Lamont, Louise, Centre de recherches sur les communications Canada CRC. Juin 2006. http://www.cihr-ircs.gc.ca/f/documents/cihr_aanual_report_2004

[6]: Benahmed Khélifa «Approche théorie des graphes pour la surveillance d'un Réseau de Capteurs sans fil» Magister en Informatique.

[7]: G.Chelius, E.Fleury et T.Mignon. **«**Broadcast, énergie et réseaux de capteurs sans fil ». CITI, Insa de Lyon-ARES, INRIA, 2004.

[8]: Chee-Yee Chong, Srikanta P. Kumar, «Sensor Networks: Evolution, Opportunities, and Challenges», 2010.

[9]: M.Assia et M.Zohra «Optimisation de la durée de vie des noeuds d'un réseau de capteurs sans fil» Ingénieur en Informatique, université de Béchar.

[10]: **https://igm.univ-mlv.fr/~dr/XPOSE2006/Bunel/Presentation.html** Présentation des réseaux de capteurs sans fil.

[**11]:** Yasser Romdhane « Evaluations des performances des protocoles S-MAC et directed diffusion dans les réseaux des capteurs» rapport de projet de f'in d'étude, école supérieure des communications de Tunis, 2007.

[12]: Yacine Challal « réseau de capteur sans fil », support de cours.

[13]: R. Kazi Chandrima, «A Survey on Sensor Network», International Journal of Computer and Information Technology (IJCIT), Vol. 1, July 2010.

[**14]:** J. P. Walters, Z. Liang, W. Shi, and V. Chaudhary, "Wireless Sensor Network Security: A Survey ", Department of Computer Science Wayne State University.

[15]: H. Krawczyk and M. Bellare and R. Canetti, "HMAC : Keyed-Hashing for Message Authentication", RFC 2104, 1997, February.

[16]: Yacine Challal, « Réseaux de Capteurs Sans Fils », Cours, Systèmes Intelligents pour le Transport, Université de Technologie de Compiègne, France, 17 Novembre 2008.

[17]: Hatem Bettahar, Yacine Challal, « Introduction à la sécurité informatique », Supports de cours, Systèmes Intelligents pour le Transport, Université de Technologie de Compiègne, France, 15 Octobre 2008.

[18]: Mohamed-El-Amine Chetibi, Souad Djerroud, « Sécurisation des échanges sur les réseaux Wifi », Thèse d'ingéniorat, Institut National de formation en informatique INI, Algérie, Juin 2008.

[19]: Emmanuel. Bresson, «Cryptographie: chiffrement par flot», Séminaire de la cryptographie, Page(s): 22-34, Laboratoire de cryptographie, Université de Paris XII, 2001/2002.

[20]: D.Baker, H.X.Mel, «La cryptographie décryptée», Livre, Nombre de Pages: 413, Edition Campus Press, 2001.

[21]: A.Bachir, A. Ouadjaout, L. Khelladi, M. Bagaa, N. Lasla, Y.Challal, « Information Security in Wireless Sensor Networks», Handbook/Encyclopedia on Ad Hoc and Ubiquitous Computing, edited by: Agrawal Dharma P., and Xie Bin., World Scientific, 2009.

[22]: Ghislaine Labouret, « Introduction à la cryptographie », Supports de cours, Cabinet Hervé Schauer Consultants-HSC, 09 Février 2001.

[23]: Noureddine Lasla « La gestion de clés dans les réseaux de capteurs sans-fil » mémoire de magister, Institut National de formation en Informatique (I.N.I) Oued-Smar, Alger.

[24]: Djallal Boubiche « protocole de routage pour les réseaux de capteur sans fil » mémoire de magister, université de batna, juin 2008.

[25]: www.securiteinfo.com, « Le Grand Livre de SecuriteInfo.com », 19 Février 2004

[26]: H. Krawczyk and M. Bellare and R. Canetti, "HMAC : Keyed-Hashing for Message Authentication", RFC 2104, 1997, February.

[27]: Shammi Didla ,Aaron Ault ault, Bagchi ,Center for Wireless Systems and Applications (CWSA) Purdue University, West Lafayette, IN 47906, USA.

[28]: Madhumita Panda SUIIT ,Sambalpur University Odisha,India. IEEE Sponsored 9th International Conference on Intelligent Systems and Control (ISCO)2015.9 (Data Security in Wireless Sensor Networks via AES Algorithm).

[29]: Renaud Dumont. Introduction à la Cryptographie et à la Sécurité informatique, Faculté des Sciences Appliquées, Université de Liège, 2006-2007. 46,47, 66-71.

[30]: National Institute of Standards and Technology, Federal Information Processing Standard 197, The Advanced Encryption Standard (AES), P5-34, http://csrc.nist.gov/publications/fips/ fips197/fips-197.pdf, November 26, 2001. Télécharger le 7 Février 2013.

[31]: Martin, B. (2004). *Codage, cryptologie et applications*. PPUR presses polytechniques.

[32]: Dworkin, M., "NIST Special Publication 800-38D ,Recommendation for Block Cipher Modes of Operation:Galois/Counter Mode (GCM) and GMAC." , U.S. National Institute of Standards and Technology http://csrc.nist.gov/publications/nistpubs/800-38D/SP-800-38D.pdf.

[33]: Bo Yang, Sambit Mishra, Ramesh Karri ECE Department Polytechnic University, Brooklyn, NY https://eprint.iacr.org/2005/146.pdf.

[34]: Eclipse Foundation. (s.d.). Eclipse IDE. Dans Eclipse Documentation. à partir de https://www.eclipse.org/documentation/.

[35]: La documentation officielle de Java d'Oracle : https://docs.oracle.com/javase/ .

[36]: Site officiel de MATLAB : Le site officiel de MathWorks, la société qui développe MATLAB: www.mathworks.com.

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

[38]: H. Chan, A. Perrig, and D. Song, ”Random key predistribution schemes for sensor networks”, In IEEE Symposium on Security and Privacy, Berkeley, California, May 11-14 2003, pp. 197–213.

[39] : X. Du et Al, ”An improved key distribution mechanism for large-scale hierarchical wireless sensor networks”, Security Issues in Sensor and Ad Hoc Networks, January 2007, Pages 35-48.

[40] : Deng, J., Han, R., & Mishra, S. (2006). INSENS: Intrusion-tolerant routing for wireless sensor networks. *Computer Communications*, *29*(2), 216-230.