REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Abderrahmane Mira de Bejaïa
Faculté des Sciences Exactes
Département d'Informatique

Mémoire de fin de cycle
En vue de l'obtention du diplôme de
Master Recherche en Informatique

Option
Réseaux et Systèmes Distribués

Thème

Cryptographie partagée dans les réseaux de capteurs sans fil

Présenté par :

GUERMOUZ Lilia

LAIB Souad

Devant le jury composé de :

Président :	Mr, Khenous Lachemi	MAA	Département d'Informatique
Examinateur :	Mr, Saadi Mustapha	MAA	Département d'Informatique
Examinatrice :	Mme, Yessad Nawel	Doctorante	Département d'Informatique
Encadreur :	Mr, Omar Mawloud	MCB	Département d'Informatique
Co-encadreur :	Mme, Bouakkaz Feriel	Doctorante	Département d'Informatique

Promotion 2013/2014

Remerciements

En premier lieu, nous remercions le bon dieu pour le courage qu'il nous a donné afin de finaliser ce travail, nous tenons aussi à remercier notre promoteur ***Mr. OMAR Mawloud****, Maitre de Conférence de classe « B » à l'Université A/ Mira de Bejaia, pour nous avoir fait confiance et nous avoir donné l'opportunité d'effectuer un travail de recherche. Il nous a encouragé, nous a conseillé et nous a guidé tout au long de la préparation de ce mémoire, tout en nous laissant une grande liberté. On espère avoir été à la hauteur de ses attentes.*

Nous remercions également ***Mlle BOUAKKAZ Feriel****, Co-promotrice de notre mémoire et Doctorante à l'Université de A/ Mira de Bejaia, pour sa patience, sa gentillesse, sa qualité d'encadrement et les nombreuses discussions que nous avons pu avoir ensemble.*

Nous tenons aussi à remercier ***Mr. SAADI Mustapha,*** *Maitre-Assistant de classe « A » à l'université de Bejaia,* ***Mr. KHENOUS Lachemi****, Maitre-Assistant de classe « A » à l'Université de Bejaia pour avoir accepté d'examiner notre travail.*

Nous tenons à remercier aussi ***Mlle YESSAD Nawel*** *Doctorante à l'université A/ Mira de Bejaia pour ses encouragements et pour avoir accepté de faire partie de notre jury de soutenance master.*

Enfin, nous tenons à exprimer notre gratitude et notre reconnaissance à nos très chers parents pour leurs soutiens et leurs encouragements et cela depuis nos plus jeunes âges.

Dédicaces

Souad

Je dédie ce travail à mes parents

A mes Sœurs Yasmine et Sonia

A mes cousins

A mes cousines

A toute ma famille

A mes amis

A mon binôme, lilouche

A toutes personnes qui m'ont apportés de l'aide.

Lilia

Je dédie ce travail à mes chers parents

A mon frère Ryad et ma sœur Linda

A mes cousins et cousines

A mes deux meilleures amies

A toute ma famille

A mon binôme, Soussou

A toutes personnes qui m'ont apportés leurs soutiens.

Table des matières

Liste des figures

Liste des tableaux

Introduction générale

L’évolution rapide des communications sans fil a permis le développement de nouveaux réseaux dont l’organisation est fortement dynamique. Un réseau de capteur sans fil est un réseau ad hoc spécifique avec un grand nombre de nœuds qui sont des micro-capteurs capables de récolter et de transmettre des données environnementales d'une manière autonome. Tout ces nœuds sont déposés de manière hétérogène sur une zone ou sur des objets. La position de ces nœuds n'est pas obligatoirement prédéterminée. Ils peuvent être aléatoirement dispersés dans une zone géographique, appelée « champ de captage » correspondant au terrain d'intérêt pour le phénomène observé.

Les nœuds des réseaux de capteurs sont généralement équipés d’une mémoire et une énergie limitées ainsi qu’une puissance de calcul plus faible. L’utilisation des réseaux de capteurs sans fil est souvent corrélée avec l’absence d’infrastructure, ainsi leur fonctionnement exige l’utilisation de protocoles collaboratifs. Pour gérer au mieux ces réseaux, il faut ainsi trouver un compromis entre les contraintes inhérentes aux capteurs et les besoins exprimés par les applications.

Les réseaux de capteurs sans fil deviennent de plus en plus populaires et leurs utilisations augmentent de jour en jour dans tous les domaines, et ces réseaux sont utilisés dans divers domaines nécessitant l’acquisition d’information concernant l’environnement. L'évolution des supports de communication sans fil, la miniaturisation, le faible coût des micro-capteurs et l'élargissement de la gamme des sondes de captage (thermique, humidité, optique, vibrations, etc.) ont élargi le champ d'application des réseaux de capteurs. Ce qui permet de collecter et de traiter des informations complexes issues de l'environnement (météorologie, étude des courants, de l'acidification des océans, de la dispersion de polluants, etc.) tel que :

- Domaine militaire : détection et collecte d'informations sur la position de l'ennemi, surveillance des zones hostiles (contaminées), détection d'agents chimiques, bactériologiques, etc.
- Domaine environnemental : pour mesurer la température, humidité, la détection des feux de forêt, le contrôle de la qualité de l'eau, etc.
- Domaine de l'industrie : gestion des stocks.
- Domaine des transports : gestion du trafic.

La cryptographie est une des disciplines de la cryptologie s'attachant à protéger des messages (assurant confidentialité, authenticité intégrité) en s'aidant souvent de secrets ou clés. Elle se distingue de la stéganographie qui fait passer inaperçu un message dans un autre message alors que la cryptographie rend un message inintelligible à autre que qui-de-droit.

Dans un réseau de capteur sans fil, les capteurs sont souvent déployés dans des zones hostiles, ce qui exige une maitrise de la consommation d'énergie par les réseaux de capteurs et la maximisation de leur durée de vie qui restent l'une des problématiques les plus fondamentales. De plus, les capteurs se trouvant dans la même portée ne captent pas nécessairement la même donnée et ceci engendre un problème d'intégrité. La mise en place des techniques capables d'éviter ce type de problème devient donc une nécessité dans le domaine de la recherche.

Pour répondre à cette problématique et pour qu'un réseau de capteurs reste autonome pendant une longue durée (quelques mois ou quelques années) et ait par la suite une longévité maximale nous avons proposés un protocole qui assure l'intégrité des données au sein d'un réseau de capteur sans fil inspiré du protocole de Merkel & Hellman.

Pour mener à bien notre travail, nous l'avons organisé en quatre chapitres selon le plan suivant :

Nous présentons dans le premier chapitre des généralités sur les réseaux de capteurs sans fil, où nous allons présenter brièvement leur architecture, et leurs contraintes. Dans le deuxième chapitre, nous présentons un état de l'art sur la

cryptographie partagée dans les réseaux de capteurs sans fil. Dans le troisième chapitre, nous proposons une solution de cryptographie partagée qui assure l'intégrité (crédibilité) des données. Nous présentons brièvement la technique du Clustering ensuite nous détaillons notre proposition. Dans le quatrième chapitre, nous évaluons les performances de notre modèle à travers des simulations. Enfin, nous concluons ce mémoire avec une conclusion générale et perspectives.

Chapitre 1

Les réseaux de capteurs sans fil

1.1. Introduction

Les progrès récents dans la technologie des systèmes de la micro-électromécanique, et des technologies de communication sans fil, ont permis de produire avec un coût raisonnable des composants de quelques millimètres cubes de volume, de faible puissance, et qui peuvent communiquer entre eux, appelés micro-capteurs. Ces derniers intègrent : une unité de captage chargée de capter des grandeurs physiques (chaleur, humidité, vibrations) et de les transformer en grandeurs numériques, une unité de traitement informatique permettant d'agréger les données collectées, une unité de stockage de données, un module de transmission sans fil et une source d'alimentation.

De ce fait, les micro-capteurs sont de véritables systèmes embarqués. Le déploiement de plusieurs d'entre eux, en vue de collecter et transmettre des données environnementales vers un ou plusieurs points de collecte, d'une manière autonome, forme un réseau de capteurs sans fil. Ces dispositifs sont déployé pour supervisé surveiller des phénomènes divers.

1.2. Qu'est ce qu'un capteur ?

Un capteur est un petit dispositif électronique sans fil capable de mesurer une valeur physique environnementale (température, lumière, pression, etc.), et de la communiquer à un centre de contrôle via une station de base. Il est composé de quatre unités de base (cf. figure 1**)** [Leh09] :

- *L'unité d'acquisition :* est généralement composée de deux sous-unités : les capteurs et les convertisseurs analogique-numérique ADCs (Analog-to-Digital Converter). Les capteurs obtiennent des mesures numériques sur les paramètres environnementaux et les transforment en signaux analogiques. Les ADCs convertissent ces signaux analogiques en signaux numériques.
- *L'unité de traitement :* est composée d'une interface avec l'unité d'acquisition et une autre avec le module de transmission. Elle contrôle les procédures permettant au nœud de collaborer avec les autres nœuds pour réaliser les tâches d'acquisition et stocker les données collectées.
- *Le module de communication (Transceiver):* il est responsable de toutes les communications via un support de communication radio qui relie le nœud au réseau.
- *Batterie :* alimente les unités citées précédemment.

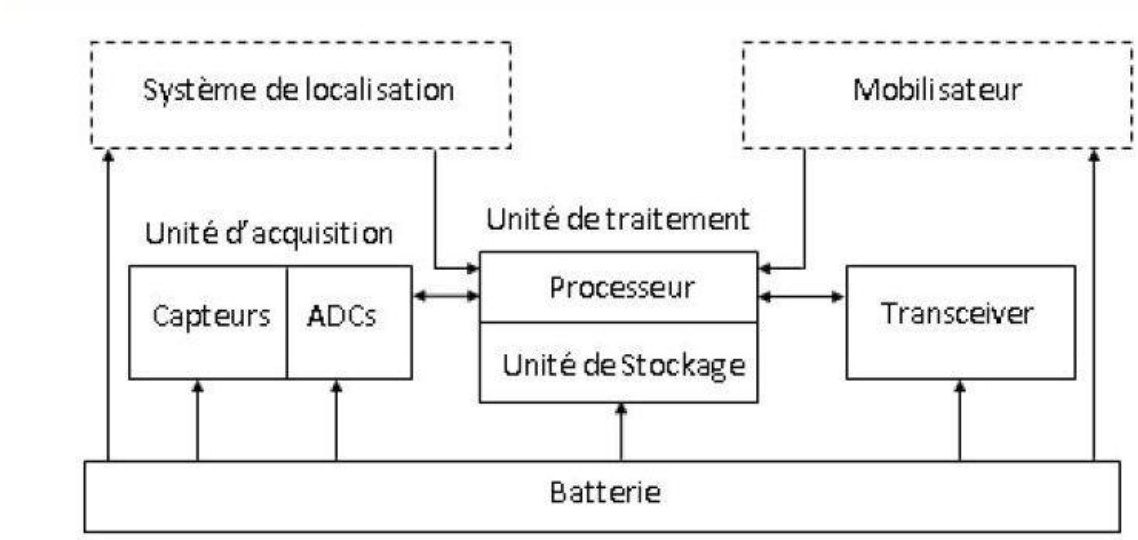


Figure 1. Les composants d'un nœud capteur

1.3. Les réseaux de capteurs sans fil

Les réseaux de capteurs sans fil sont considérés comme un type spécial des réseau ad hoc, avec un très grand nombre de nœuds où l'infrastructure fixe de communication et l'administration centralisée sont absentes. Les capteurs sont placés de manière plus ou moins aléatoire (par exemple par le largage depuis un hélicoptère) dans un

environnement pouvant être dangereux. Toute intervention humaine après le déploiement des capteurs est la plupart du temps exclue, le réseau doit donc s'autogérer. Le but d'un réseau de capteurs sans fil est de surveiller une zone géographique, par exemples un réseau détecteur de feu de foret, ou un réseau de surveillance de solidité d'un pont après un tremblement de terre [Mes08].

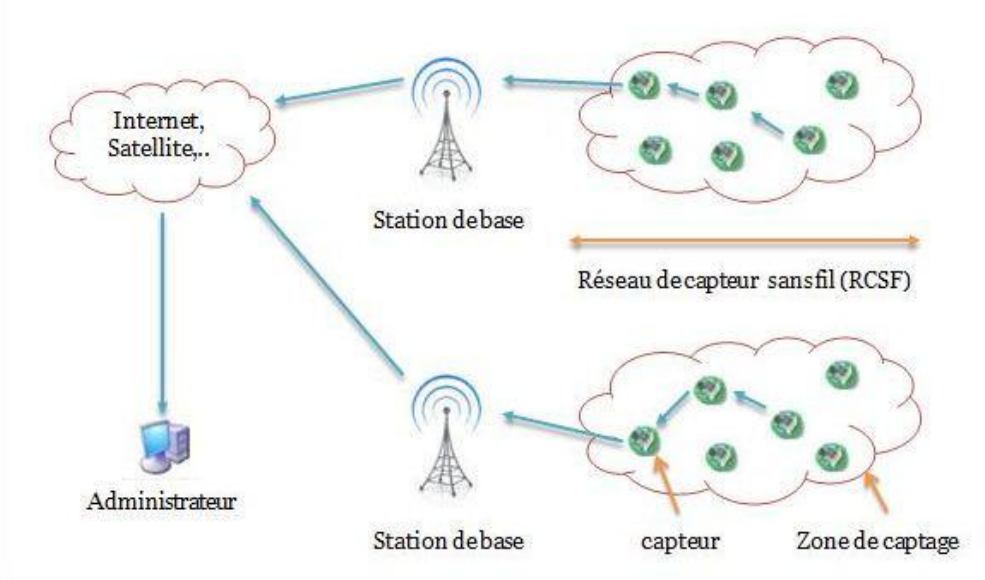


Figure 2. Réseaux de capteur sans fil

1.4. Domaines d'applications des réseaux de capteurs

La miniaturisation, l'adaptabilité, le faible coût et la communication sans fil permettent aux réseaux de capteurs d'envahir plusieurs domaines d'applications. Ils permettent aussi d'étendre le domaine des applications existantes. Parmi ces domaines où ces réseaux se révèlent très utiles et peuvent offrir de meilleures contributions, nous pouvons noter le domaine militaire, la santé, l'environnement, et les maisons intelligentes etc. [Leh09].

1.4.1. 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. Plusieurs projets ont été lancés pour aider les unités militaires dans un champ de bataille et protéger les villes contre des attaques, telles que les menaces terroristes.

1.4.2. Applications environnementales

Le contrôle des paramètres environnementaux par les réseaux de capteurs peut donner naissance à plusieurs applications. Par exemple, le déploiement des thermo-capteurs dans une forêt peut aider à détecter un éventuel début de feu et par la suite faciliter la lutte contre les feux de forêts avant leur propagation. Dans les champs agricoles, les capteurs peuvent être semés avec les graines. Ainsi, les zones sèches seront facilement identifiées et l'irrigation sera donc plus efficace. De même leur déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.).

1.4.3. 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 (telles que : la tension artérielle, battements du cœur, etc.) à l'aide des capteurs ayant chacun une tâche bien particulière. Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient [JA96]. D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, etc.) chez les personnes dépendantes (handicapées ou âgées).

1.4.4. La domotique

Avec le développement technologique, les capteurs peuvent être embarqués dans des appareils, tels que les aspirateurs, les fours à micro-ondes, les réfrigérateurs, les magnétoscopes, etc. Ces capteurs embarqués peuvent interagir entre eux et avec un

réseau externe via Internet pour permettre à un utilisateur de contrôler les appareils domestiques localement ou à distance

Le déploiement des capteurs de mouvement et de température dans les futures maisons dites intelligentes permet d'automatiser plusieurs opérations domestiques telles que : la lumière s'éteint et la musique 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.

1.5. Contraintes et caractéristiques liées aux réseaux de capteurs

La réalisation des réseaux de capteurs dédiés aux applications citées précédemment, exigent des techniques et des protocoles qui prennent en compte les spécificités et les exigences de ces réseaux, vu que les techniques conçues pour les réseaux ad hoc traditionnels ne sont pas bien adaptées aux réseaux de capteurs. Pour illustrer ce point, les différences entre les réseaux de capteurs et les réseaux ad hoc sont décrits ci-dessous [Leh09]:

- Dans les réseaux de capteurs, les nœuds sont déployés en grand nombre.
- Les réseaux de capteurs sont non fiables, ou à tout moment, les capteurs peuvent être défaillants ou inhibés.
- La topologie des réseaux de capteurs change très fréquemment.
- Les capteurs sont limités en énergie, capacités de calcul, et mémoire.

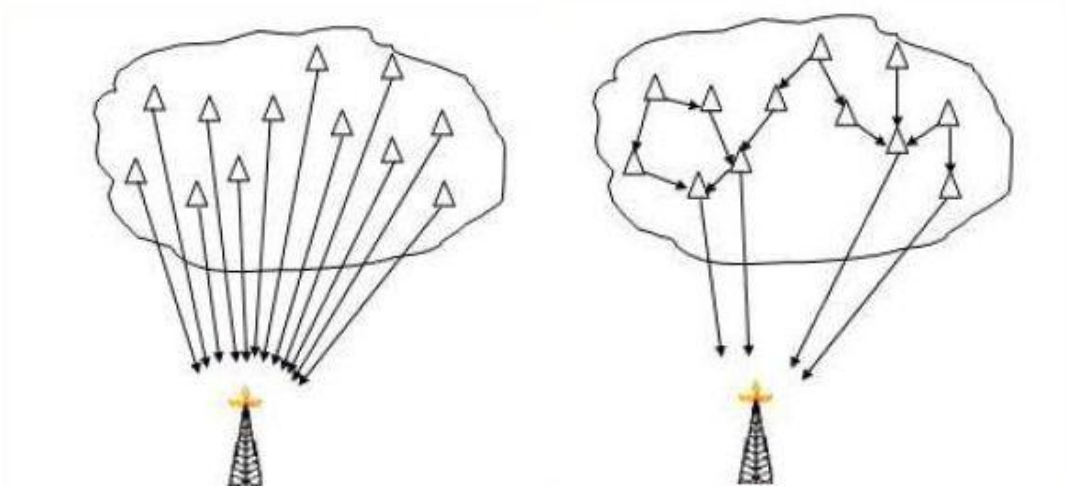
Le contexte de développement des réseaux de capteurs prend en compte leurs caractéristiques et leurs spécificités.

1.6. Architectures adoptée pour les réseaux de capteurs

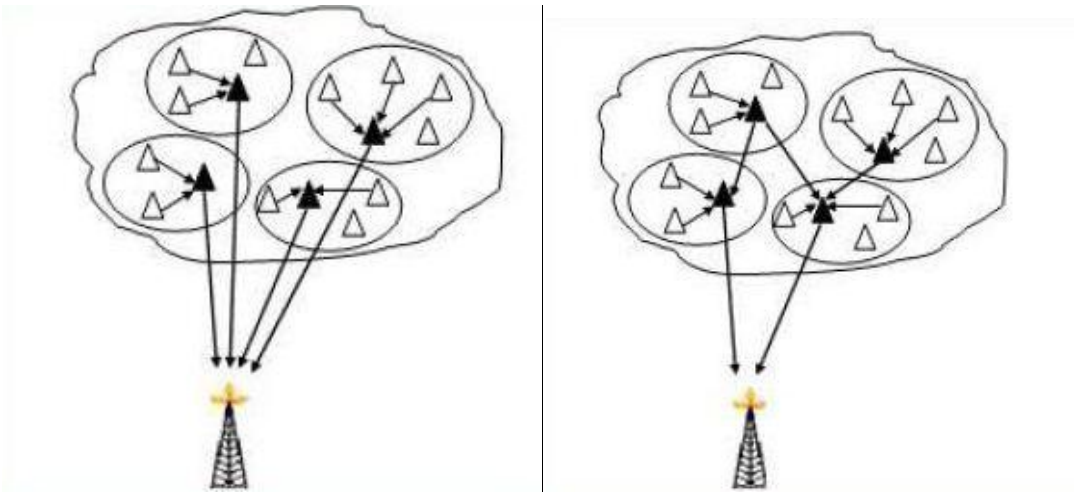
Puisque les réseaux de capteurs sans fil se caractérisent par l'absence d'une infrastructure déterminée au préalable, les nœuds la construisent tout en permettant l'interaction avec l'environnement où ils appartiennent et en répondant aux différentes requêtes venant des utilisateurs ou des réseaux externes. Par ailleurs, les capteurs comme tout autre composant de télécommunication adhèrent à une architecture

protocolaire spécifique. La réalisation de cette dernière requiert la mise en œuvre de techniques développées pour les réseaux ad hoc. Cependant, de nouveaux problèmes apparaissent engendrés entre autre par la sévérité des contraintes dues aux limitations de ressources physiques. C'est pourquoi, il est commode que la conception des protocoles de communication soit faite d'une manière optimale.

Le processus d'acheminement de l'information des capteurs à la station de base peut prendre quatre formes. Dans les architectures à plat, les capteurs peuvent communiquer directement avec la station de base en utilisant une forte puissance (cf. figure 3 (a)), via un mode multi-sauts avec des puissances très faibles (cf. figure 3 (b)), alors que dans les architectures hiérarchisées, le nœud représentant le cluster, appelé Cluster-Head, transmet directement les données à la station de base (cf. figure 3 (c)), ou via un mode multi-saut entre les Cluster-Heads (cf. figure 3 (d)) [Leh09].



(a) Communication directe des capteurs (b) Communication multi-saut des capteurs



(c). Communication directe des capteurs (d). Communication multi-saut des capteurs

Figure 3. Architecture de communication de données dans un réseau de capteur.

1.7. Conclusion

Les réseaux de capteurs sans fil possèdent des caractéristiques particulières qui les différencient des autres types de réseaux sans fil. Ces spécificités telles que la consommation d'énergie réduite, la scalabilité ou le routage incitent le besoin de concevoir de nouveaux protocoles d'accès au support, de routage, de sécurité, de transport ou d'application, qui s'adapteront aux caractéristiques des les réseaux de capteurs sans fil.

Chapitre 2

Etat de l'art sur la cryptographie partagée

2.1. Introduction

Ce chapitre présente les travaux réalisés dans le domaine lié à la cryptographie partagée ainsi que la cryptographie à seuil qui en est un cas particulier, afin d'offrir aux réseaux de capteurs sans fil une meilleure sécurité. Nous introduisons cependant quelques notions de bases qui permettent aux nœuds de partager une clé sans pouvoir déduire la clé entière. Le concept de la cryptographie à seuil a été introduit pour la première fois par Shamir [Sha79], qui a proposé une technique qui permet le partage d'une valeur secrète *S* à un ensemble de *n* serveurs. A partir d'au moins *k* serveurs nous pouvons reconstruire le secret *(k <= n)*, si le nombre de serveurs est inférieur à *k*, aucune information n'est obtenue sur le secret *S*. Nous concluons ce chapitre en comparant les travaux présentés par rapport aux critères suivants : charge de calcul, charge de stockage, charge de communication et scalabilité.

2.2. Les techniques de cryptographie

Le mot « cryptographie » est un mot d'origine grecque composé de deux parties : « Crypto » (*kruptos*) qui signifie caché et « graphie » (*graphein*) qui signifie écrire. D'une manière générale, la cryptographie est la science de la dissimulation de messages de sorte que seuls certains initiés disposant d'une donnée secrète soient en mesure de prendre connaissance de leur contenu. En effet, la cryptographie consiste en une paire d'opérations ($\mathcal{E}$, *D*). La première opération $\mathcal{E}$ est le chiffrement (appelée aussi *cryptage*). Elle permet de convertir un message initial *M*, dit message en clair, en un autre message *C*, dit message chiffré, incompréhensible à une tierce partie. La forme de *C* dépend d'un paramètre *K* appelé clé de chiffrement. La deuxième opération *D* est

le déchiffrement (appelée aussi *décryptage*). Le déchiffrement permet de reconstruire le message en clair à partir du message chiffré. Cette reconstruction requiert une deuxième clé K^{-1}, dépendante de la clé de chiffrement, dite clé de déchiffrement. La définition de la paire ($\mathcal{E}$, D) constitue un système cryptographique. Les systèmes cryptographiques les plus utilisés peuvent être classés en trois types : (1) systèmes cryptographiques à sens unique, (2) systèmes cryptographiques symétriques, et (3) systèmes cryptographiques asymétriques.

2.2.1. Cryptographie à sens unique

Dans ce type de cryptographie [ISG00], le message est chiffré de telle sorte qu'il est impossible de reconstruire le message original à partir du message chiffré. Cette technique de cryptographie a donné naissance à une catégorie de fonctions, appelées *fonctions de hachage*, qui sont largement utilisées pour le contrôle d'intégrité des données. Etant donné une fonction de hachage H et un message M. nous appelons le résultat de hachage $h = H(M)$ le condensé de M. Les fonctions de hachage sont caractérisées par :

- Il est impossible de retrouver le message M à partir de h.
- Il est difficile de trouver un message M' tel que : $H(M) = H(M')$.

Généralement, la taille de h est toujours constante et très petite par rapport à M. En effet, cette catégorie de fonctions est très utilisée dans les opérations cryptographiques, principalement dans le but de réduire la taille des données à traiter par la fonction de chiffrement. Une fonction de hachage reçoit une entrée de longueur variable et produit une sortie de longueur fixe. Ce type de fonctions assure que si l'information est modifiée, même d'un seul bit, une sortie totalement différente serait produite.

Il existe deux types de fonctions de hachage : les fonctions de hachage avec et sans clé. Les fonctions de hachage sans clé peuvent être calculées par n'importe quelle entité participante à la communication. La valeur calculée dans ce cas ne dépend que du message initial, alors que les fonctions de hachage avec clé sont en fonction du message initial et d'une clé de hachage. Seuls ceux qui possèdent cette clé peuvent calculer la valeur de hachage correspondante du message initial. Les fonctions de

hachage les plus répandues sont MD5 (*Message Digest*) [WFL04] et SHA (*Secure Hash Algorithm*) [NSA93].

2.2.2. La cryptographie symétrique

Dans ce type de systèmes [ISG00], la même clé *K* est utilisée à la fois pour le chiffrement et le déchiffrement (cf. figure 4). Si nous supposons que $M' = \mathcal{E}(M, K)$ est le chiffrement du message *M* en utilisant la clé *K*, et $D(M', K)$ pour le déchiffrement du message *M*' en utilisant la même clé *K*, alors la caractéristique fondamentale de cette technique de cryptographie est : $D(\mathcal{E}(M, K), K) = M$.

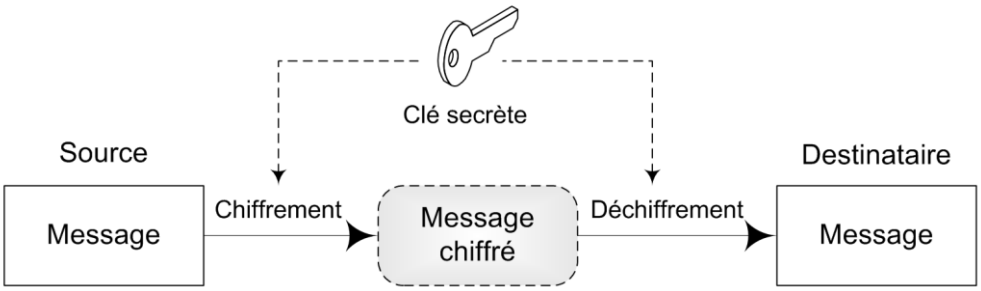


Figure 4. Chiffrement symétrique

Chaque communicant dans le système désirant transmettre des données confidentielles doit partager avec son partenaire une clé secrète, et cette dernière sera utilisée par l'expéditeur pour chiffrer les données avant de les envoyer et par le destinataire pour les déchiffrer une fois reçues. Les systèmes cryptographiques symétriques les plus répandus sont : DES (*Data Encryption Standard*) [NBS77], AES (*Advanced Encryption Standard*) [NIST01], et IDEA (*International Data Encryption Algorithm*) [LM91].

2.2.3. La cryptographie asymétrique

Le concept de la cryptographie asymétrique a été inventé par *Diffie* et *Hellman* [DH76]. Cette technique utilise une paire de clés complémentaires : une *clé publique* qui chiffre les données, et une *clé privée* pour les déchiffrer (cf. figure 5). La clé publique doit être diffusée à tous les correspondants dans le système, par contre la clé privée doit rester secrète au niveau de son propriétaire. Toute entité en possession d'une copie de la clé publique peut chiffrer des informations que seul le propriétaire de

la clé privée pourra les déchiffrer [ISG00]. Les systèmes cryptographiques à clés asymétriques les plus répandus sont RSA (*Rivest Shamir Adelman*) [RSA78] et Elgamal [Elg98]. La signature numérique est l'un des services réalisés grâce à la cryptographie asymétrique. Elle fournit les services d'authentification, d'intégrité des données, et la non-répudiation. Sur le plan conceptuel, la façon la plus simple de signer un message consiste à chiffrer celui-ci avec la clé privée de l'expéditeur. Seul le possesseur de cette clé est capable de générer la signature. Cependant dans la pratique, cette méthode est peu utilisée du fait de sa lenteur. La méthode réellement utilisée consiste à calculer une *empreinte* du message à signer et de ne chiffrer que celle-ci. Pour calculer l'empreinte, nous utilisons les fonctions de hachage, là où le condensé généré est considéré comme résumé du message, et ainsi nous pouvons le considérer en tant qu'empreinte du message.

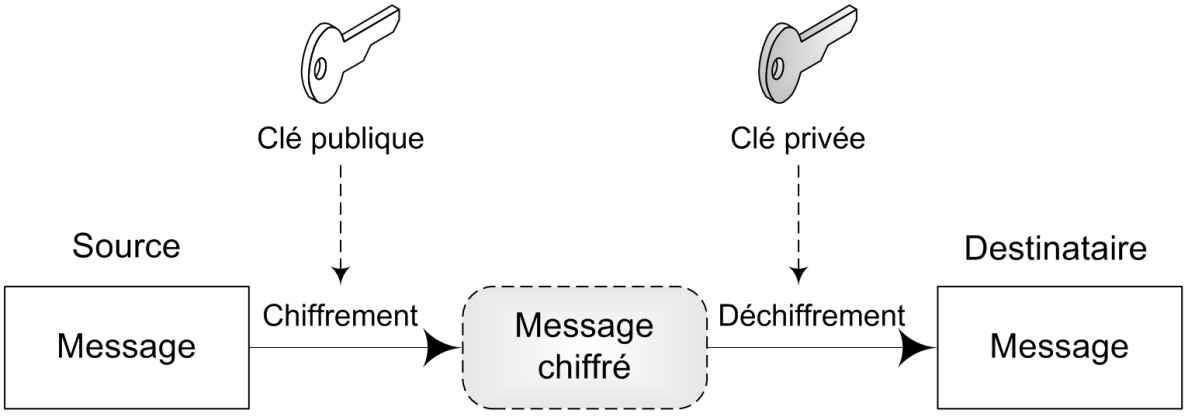


Figure 5. Chiffrement asymétrique

2.2.4. La cryptographie à base d'identité

Le cryptage à base d'identité (IBE - *Identity-Based Encryption*) [Sha85] est une technique de cryptographie asymétrique, dans laquelle la clé publique de chaque utilisateur est liée à son identité. Cette identité sera traitée par une tierce partie centralisée, appelée PKG (*Private Key Generator*), qui va la combiner avec sa propre clé privée en générant celle qui est propre à l'utilisateur. Ainsi, les utilisateurs peuvent envoyer des messages chiffrés (ou signés) sans avoir besoin de solliciter le PKG en utilisant directement les identités des correspondants comme des clés publiques.

2.2.5. La cryptographie à seuil

Le partage de secret repose sur le concept de détention d'une portion d'une information secrète par plusieurs personnes, comme un coffre-fort bancaire dont l'ouverture est commandée par l'introduction simultanée de plusieurs clés. Le partage de secret est traditionnellement utilisé en informatique pour scinder des clés de déchiffrement en plusieurs parties de sorte que chacune d'elles possède une portion de la clé. Le concept de la cryptographie à seuil est inventé par *Shamir* [Sha79]. Il a proposé un mécanisme basé sur l'interpolation polynomiale. Il permet le calcul et le partage d'une valeur secrète *S* à un ensemble de *n* serveurs, sans que chacun d'eux connaisse la valeur. A partir d'au moins *k* serveurs nous pouvons reconstruire le secret. Si le nombre de serveurs est inférieur à *k*, aucune information n'est obtenue sur le secret *S*. Cette technique de cryptographie a été combinée avec le système cryptographique asymétrique RSA pour avoir un système qui permet de partager le pouvoir de signature à un ensemble de serveurs [ZH99].

2.3. Critères d'évaluation des solutions existantes

Pour évaluer les protocoles cryptographiques, nous nous intéressons au coût en termes de calcul, d'énergie et de surcoût de communication.

- **Charge de calcul :**

Les capteurs sans fil actuels souffrent d'un manque de puissance de calcul, qui est fortement préjudiciable pour le temps de réponse du réseau, et leur utilisation dans des applications avec un nombre de nœuds élevé nécessite l'utilisation de capteurs puissants. Pour cela, il est recommandé d'utiliser des algorithmes moins complexes dans les réseaux de capteurs sans fil. Ce paramètre a une répercussion directe sur l'énergie.

- **Charge de stockage :**

La capacité de stockage et de traitement des nœuds capteurs est relativement faible. Pour cela, le système doit limiter le nombre et la taille des clés à stocker.

- **Charge de communication :**

Dans un réseau de capteurs sans fil, les communications sont les actions les plus coûteuses en termes d'énergie. Il est donc fortement nécessaire de limiter la charge de communication entre les capteurs en limitant le nombre de messages échangé entre les nœuds.

- **Scalabilité :**

Un réseau est dit 'scalable' si le réseau peut facilement être augmenté sans perdre ses propriété essentielle ou sans réduire ses performances.

2.4. Les travaux antérieurs

2.4.1. Solution de Jing-feng et al.

Dans ce travail, Jing-feng et al. [JDH08] proposent un système de surveillance sécurisé basé sur l'identité fondée sur la signature numérique à seuil dans les réseaux de capteurs sans fil. Ce système considère un réseau de capteurs sans fil où les nœuds sont décomposés en sous-régions *I* constitués de *n* nœuds dont l'ensemble des identités ***i*** internes est $Ui = \{M_{i1}, M_{i2}, ..., M_{in}\}$ qui sont à leurs tours connectés à un générateur de clés privées (*PKG*). Ce dernier calcule la fonction de hachage correspondante pour chaque sous-région Q_I et génère la clé privée partagée pour chaque nœud dans la même région, ensuite il génère des clés publiques Q_{BS} et une clé privée S_{BS} et les transmet à la station de base. Si un message anormale est détecté et capté par *t* nœuds, tel que *t* est le seuil, ces derniers vont utiliser la signature numérique à seuil pour le traiter. De ce fait, chacun d'eux génère sa propre signature partielle en utilisant le système de signature basé sur l'identité de Chang et al. [CLW06] puis l'un des capteurs sera désigné pour combiner toutes les signatures et construire une signature finale afin de l'envoyer à la station de base pour la vérifier.

Phase de configuration des paramètres du système

Le PKG génère et publie les paramètres du système $params = \{G_1, G_2, n_1, \hat{e}, P, P_{PUB}, E, D, H_1, H_2, H_3\}$, tel que G_1 est un groupe additif cyclique d'ordre *q* généré par *P*, G_2 un groupe multiplicatif d'ordre *q*, $\hat{e}$ est un système sécurisé de cryptage

symétrique, P_{PUB} est la clé publique et P est le générateur de G_1 et G_2. H_1: $\{0,1\}^* \rightarrow G_1$, H_2: $G_2 \rightarrow \{0,1\}^{n1}$, H_3: $\{0,1\}^* \rightarrow Z_q^*$ trois fonctions de hachage, l'entité hors ligne *PKG* sélectionne aléatoirement une clé publique du système $s \in Z_q^*$ et la sauvegarde en toute sécurité et calcule le système à clé publique $P_{PUB} = sP$.

Phase de la génération et distribution de la clé

Avant le déploiement du réseau de capteurs sans fil, pour une région *A*, le PKG génère la clé privée $S_A = {}_sQ_A$ pour chaque nœud appartenant à la même sous-région et calcule sa fonction de hachage $Q_A = H_1(A)$, puis partage la clé privée dans la sous-région en sélectionnant aléatoirement des nombres $F_1, ..., F_{t-1} \in G_1^*$, construit le polynôme $F(x) = S_A + xF_1 + ... + x^{t-1}F_{t-1}$ et calcule $S_i = F(i)$, sachant que S_0 est la clé privée, puis calcule $y_0 = \hat{e}(S_A, P)$ et $y_j = \hat{e}(F_j, P)$ sachant que $j = 1,2,...., t$ et $i = 1,2,...., n$, $(1 \leq t \leq n \leq q)$, ensuite il distribue S_i , y_0 et y_j à M_{Ai}, enfin le PKG détruit la clé privée. PKG génère les clés publiques $Q_{BS} = H_1(BS)$ et la clé privée $S_{BS} = sQ_{BS}$ pour la station de base (cf. figure 6).

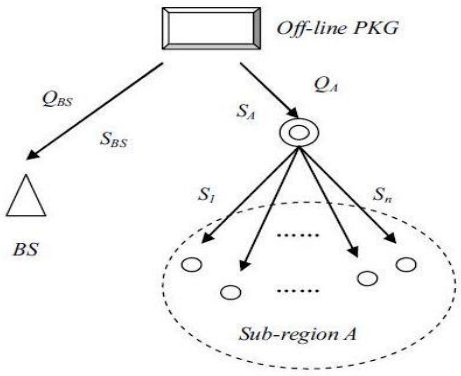


Figure 6. Génération et distribution des clés

Phase de signature numérique à seuil d'un message alarmant

Lorsque un message anormal se produit dans une sous-région *A*, il est détecté par ***t*** nœuds. La coopération des signatures numérique à seuil de ces nœuds est nécessaire pour le traiter et l'un des nœuds (*C*) choisit selon des critères, combine toutes les signatures partielles valides afin de construire une seule signature. Pour cela chaque nœud M_{Ai} $(1 \leq i \leq t)$ choisit aléatoirement $x_i \in Z_q^*$, calcule $R_{1i} = X_i P$ et $R_{2i} = x_i P_{pub}$. Puis,

il envoie (R_{1i}, R_{2i}) à C qui à son tour calcule $R_1 = \sum_{i=1}^{t} R_{1i}$, $R_2 = \sum_{i=1}^{t} R_{2i}$, $\tau = \hat{e}(R_2, Q_{BS})$, $k = H_2(\tau)$ et $h = H_3(m, R_1, k)$. Ensuite, il envoie h à M_{Ai}. Chacun de ces derniers va produire une signature partielle W_i en utilisant le système de signature basé sur l'identité de Chang et a. [CLW06], puis l'envoie à C. Une fois toutes les signatures partielles W_i, $W_i = x_i P_{pub} + h\eta_i S_i$, $\eta_i = \prod_{j=1, j\neq i}^{t} -j(i-j)^{-1} \bmod q$ sont reçues par C, il vérifie leur validité. Si toutes les signatures sont valides, il calcule la somme des signatures et il envoie une signature à seuil $\sigma = (c, R_1, W)$ à la station de base. Sinon C refusera W_i et demande une nouvelle signature partielle valide (cf. figure 7).

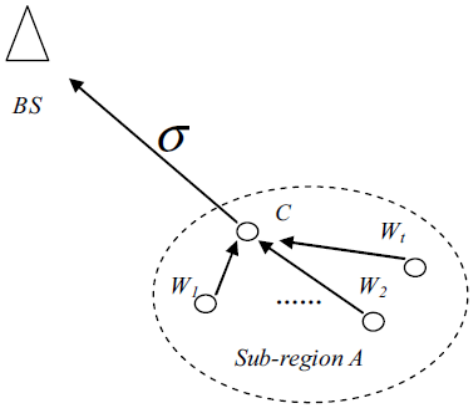


Figure 7. Processus de la signature numérique à seuil

Phase de vérification de la signature digitale

Une fois la station de base reçoit $\boldsymbol{\sigma}$, elle calcule $\tau = \hat{e}(R_1, S_{BS})$ et $k = H_2(\tau)$ et récupère le message $m = D_k(c)$, calcule $h = H_3(m, R_1, k)$, et accepte $\boldsymbol{\sigma}$ si et seulement si $\hat{e}(P, W) = \hat{e}(P_{pub}, R_i + {}_hQ_A)$.

Le coût de communication est déterminé par la longueur du message chiffrée $|m| + 2|G_1|$, et le coût de calcul de chaque nœuds est faible ce qui représente l'avantage de ce protocole.

2.4.2. Solution de Sliti et al.

Dans [SHB08], la technique développée par les auteurs est appelé k-sécurité pour les réseaux de capteurs sans fil hétérogènes, qui repose sur une authentification basée sur la cryptographie à courbe elliptique (ECC) et la signature à seuil avec une

vérification intermédiaire des messages transmis afin de contrer les messages falsifiés conduisant à de mauvaises interprétations et de mauvaises décisions.

Les auteurs ont adapté la solution pour le suivi des cibles militaires. Il s'agit d'un réseau de capteurs hétérogènes, composé d'une couche centrale ainsi qu'une couche de détection. L'un des principaux concepts est la *k*-couverture qui permet la collecte d'information sur une cible potentielle détectée par les *k* capteurs, qui est étroitement liée au concept appelé *k* – sécurité pour les réseaux de capteurs sans fil introduit par les auteurs, et cela peut être résumé par les propriétés suivantes :

- Chaque capteur s_i possède une clé privée notée k_i.
- Une unique clé publique π et un algorithme peuvent être utilisés pour vérifier si les *k* signatures du même message généré par des capteurs distincts sont valides ou non.
- Un événement détecté par la couche de détection est considéré comme valable à la couche centrale, si et seulement si, *k* messages d'alerte sont reçus et vérifiés avec succès.

C'est un nouveau concept avec un niveau de sécurité maximal qui peut atteindre *k*, en raison du nombre de capteurs qui seraient capables de détecter la cible hostile.

a. Authentification

Les auteurs décrivent la structure d'authentification proposée en 3 phases. La première est l'enregistrement des nœuds tel que, une autorité de certification génère une clé publique π associée à *n* clés privées k_1 ... k_n en utilisant un système à seuil elliptique. et répartit la confiance dans tout le groupe de nœuds capteurs en signant individuellement un message *m* avec un seuil de *k* nœuds afin de générer une signature valide globale.

Après avoir reçu une demande d'enregistrement à partir d'un nœud s_i ($1 \leq i \leq n$), une autorité de certification inscrit l'empreinte digitale du nœud $\varphi(s_i)$ et génère un certificat correspondant désigné comme $Cert_{si}$ (cf. figure 8).

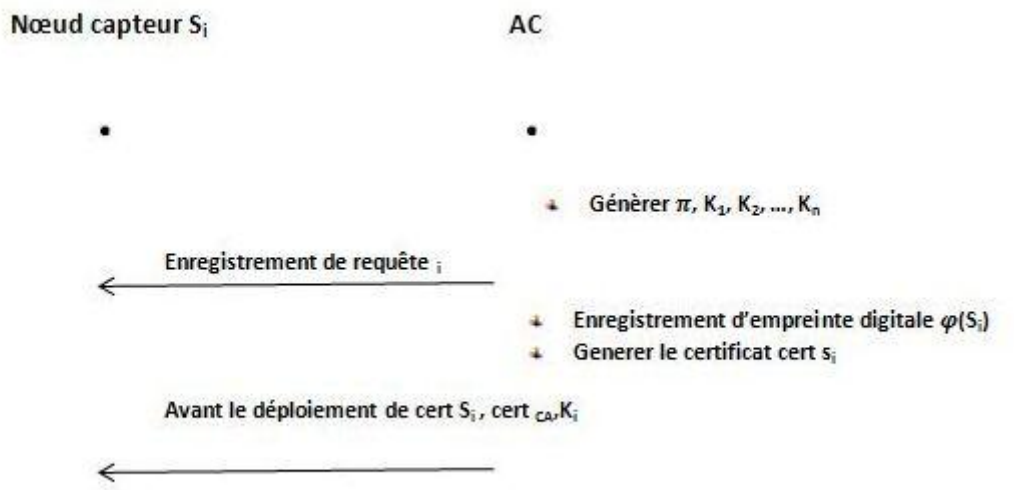


Figure 8. Enregistrement des nœuds

Puis le message *m* sera transmis au nœud voisin s_k le plus proche qui va décomposer le message en données et en signatures intermédiaires générées par d'autres nœuds. s_k va alors sur-signer le message *m* et l'envoie au nœud voisin le plus proche s_i, et cela si les signatures intermédiaires sont valables. Dans le cas échéant il sera rejeté.

Lors de la réception des *k* signatures, le capteur central procède à la vérification de la validité des signatures combinées reçues à l'aide de l'unique clé publique π. Ainsi que la validité des *k* certificats en vérifiant la validité de la signature de l'autorité de certification et l'empreinte digitale φ (s_i) pour chaque certificat de nœud $Cert_{si}$ (cf. figure 9).

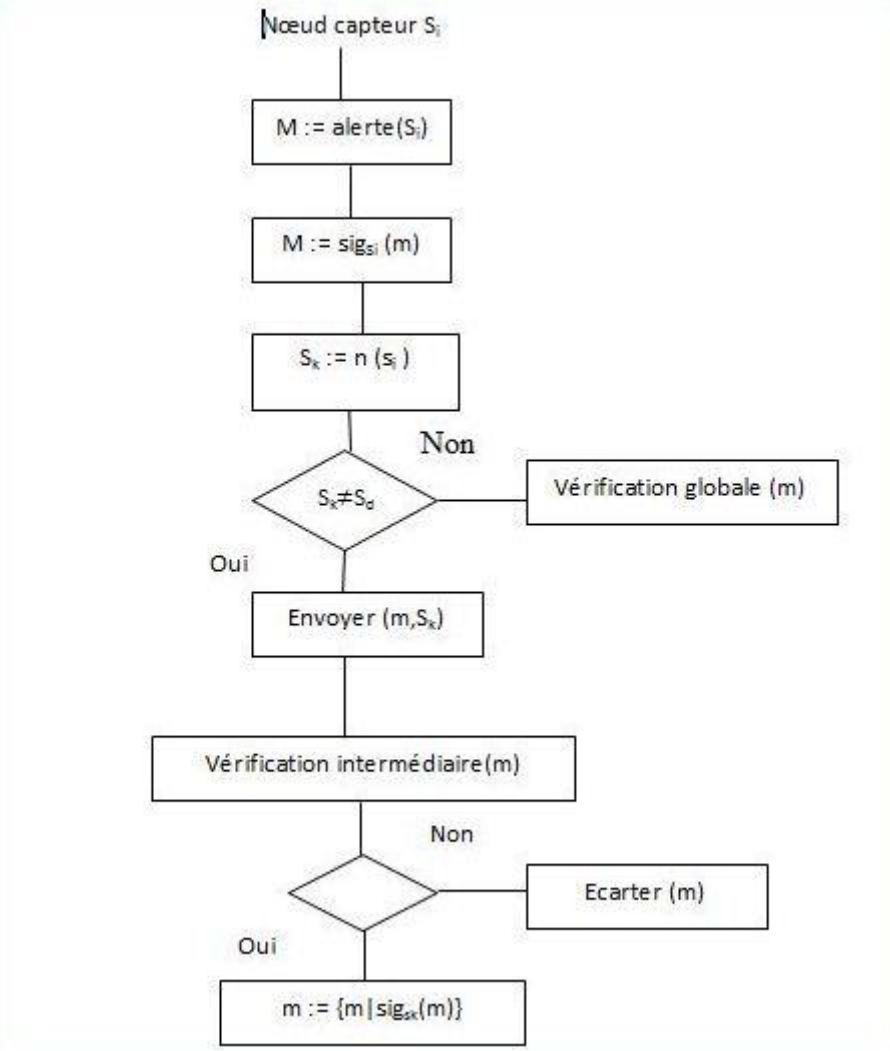


Figure 9. Organigramme de signature entre nœuds.

b. Le système de signature à seuil elliptique

Pour mettre en œuvre l'architecture décrite dans la section qui précède, un schéma de signature à seuil *k* de *n* est développé (*n* est le nombre de capteurs et *k* le seuil).

Phase de génération de la clé

La phase de génération de la clé est effectuée par l'autorité de certification en générant aléatoirement deux grands nombres premiers p et q tel que $p = 2p'+1$, $q = 2q'+1$ ou p' et q' sont eux-mêmes des nombres premiers. Sachant que t = p.q et m = p'.q', en générant l'exposant public e $\in$ P tel que e > n (n est le nombre de clés privées). On obtient une clé publique π = (t, e), puis enfin calculer la clé privée k_i du nœud s_i.

Phase de génération et de vérification de la signature intermédiaire

Un événement *V* liée à la détection d'une cible hostile doit être convertie en un point *P* = (*v1, v2*) dans E_m (*a, b*). La conversion d'un message en un point ECC est effectuée en fonction de l'approche proposée dans [SEC00]. Ensuite, un nœud s_i peut générer sa signature individuelle (σ_i^x, σ_i^y) à l'aide de sa clé privée selon une équation appropriée.

Vérification de la signature globale

Lorsqu'un capteur central recueille k signatures générées par un sous-ensemble de nœuds, il procède à la vérification de la validité de la signature globale à l'aide des expressions adéquates.

La tolérance envers les faux nœuds dans les réseaux de capteurs sans fil

En utilisant la technique d'authentification proposée, les capteurs centraux peuvent identifier les nœuds représentés. La fonctionnalité de vérification intermédiaire peut être utilisée pour localiser l'origine des messages signés de façon inappropriée. Ce mécanisme peut être utilisé pour construire un plan de tolérance aux pannes pour les réseaux de capteurs sans fil, un état de tolérance, où le capteur peut transmettre les paquets sans émettre d'alerte des messages. Ces derniers peuvent être initiés lors de la détection d'une activité suspecte exercée par ce nœud.

L'utilisation de ECC ainsi que la signature à seuil permettent de minimiser les ressources mémoires et ainsi que celles de traitement, et contrairement au système traditionnel qui utilise une clé publique pour chaque clé privée [SEC00], le système développé par les auteurs utilise une seule clé publique pour vérifier plusieurs signatures ainsi qu'une vérification intermédiaire des signatures tout au long du chemin vers le capteur central, ce qui permet un gain des ressources énergétiques et de réduire le cout de communication.

2.4.3. Solution de Koschuch et al.

Dans [KHKLW10], Koschuch et al. ont combiné la cryptographie à seuil au protocole des calculs multipartis pour distribuer le secret et l'appliquer aux réseaux de capteurs sans fil, puis avec l'algorithme de RSA le secret est signé par les capteurs et envoyé au Cluster Head pour calculer la signature complète.

Koschuch et al. ont modifié le protocole décrit en [GRR98] en appliquant la cryptographie à seuil pour désigner le nombre de nœuds qui doivent coopérer afin de créer le secret. Au début du protocole de multiplication, chaque nœud P_i détient les deux valeurs $f_a(i)$ et $f_b(i)$ des deux polynômes f_a et f_b de degré t et $a = f_a(0)$, $b = f_b(0)$. A la fin du protocole, chaque nœud possède la valeur de la fonction $H(i)$ du polynôme H de degré t qui est une partie du produit ab tel que $ab = H(0)$.

Les étapes principales du protocole [GRR98] sont présentées ci-dessous :

- ***Première étape***

Avant le déploiement, chaque nœud P_i ($1 \leq i \leq 2t+1$), calcule $f_a(i) * f_b(i)$ et partage cette valeur avec les autres nœuds en utilisant un polynôme $h_i(x)$ de degré t, puis envoie au joueur P_j la valeur $h_i(j)$ ($1 \leq j \leq n$).

- ***Deuxième étape***

Chacun des capteurs calcule sa partie $H(j)$ de ab en combinant les valeurs $hi(j)$ pour $i = 1, 2, \ldots, 2t+1$.

La complexité de la première étape de [GRR98] nécessite $O(N^2 k \log n)$ opérations binaires par capteur, où k est le nombre de bits du facteur premier q et n est le nombre de capteurs. Par contre dans [Lor07], la complexité est réduite à $O(N^2 k)$. La deuxième étape du protocole [GRR98] nécessite $O(Nk^2)$ opérations binaires pour chaque capteur, par contre en [Lor09] la complexité est de $O(N^2 k)$. Ce résultat n'est valable que dans le cas ou $n < k$. Cela n'est pas généralement vrai car $k \geq 1024$ et cela

nécessite des tours de communication dans la première étape. L'évaluation des deux protocoles se fait par rapport au nombre de cycles.

2.4.4. Solution de Singh et al.

HGKMTC (*Hierarchical Group Key Management using Threshold Cryptography in Wireless Sensor Networks*) est une technique de gestion des clés utilisant la cryptographie à seuil dans les RCSF. Les chercheurs qui ont proposés cette méthode, ont évalué ses performances en la comparant avec la méthode EEKM (*Energy-Efficient Key-Management*) [PKHLS08].

HGKMTC considère un réseau de capteurs hiérarchiques, où les nœuds de transfert *FN* sont connectés à la station de base *BS* et les nœuds de détection sont coordonnés par *FN*. La technique suppose que les nœuds sont pré chargés avec des clés secrètes initiales. Après le déploiement des nœuds dans le réseau, les nœuds de transfert forment des clusters, Chaque groupe de nœuds capteurs sera connecté à un nœud de transmission *FN* qui a son tour transmet une demande à la station de base *BS* afin d'obtenir la clé secrète du groupe. Une fois l'authentification réussie, *BS* transmet la clé du groupe au *FN* correspondant à l'aide du système de partage du secret à seuil (une version améliorée de la technique de Shamir), puis *FN* transmet la clé à chaque membre de son groupe. La clé de groupe obtenue est divisée en deux clés pour le chiffrement et le déchiffrement de données.

Nous détaillons à présent les deux phases principales du protocole de partage du secret à seuil.

Phase de distribution du secret partagé

Initialement, la station de base (supposée sûr) sélectionne un secret S_e qui doit être divisé en *p-1* blocs de taille *r* (*p* est un nombre premier supérieur ou égale au nombre de nœuds participants) comme suit : $Se_1Se_2 \ldots Se_{p-1} \in \{0,1\}^r$, $Se_0 = 0^r$ est la première partie du secret partagé. Puis l'algorithme sélectionne aléatoirement $(K\text{-}1)_{p-1}$ parties de *r* bites $r_0^0, \ldots, r_{p-1}^0, r_0^1, \ldots, r_{p-1}^1, \ldots, r_0^{k-2}, \ldots, r_{p-2}^{k-2}$ qui sont utilisés par la BS pour partitionner le secret en utilisant une formule mathématique avec un seuil *K :*

$$a_{(i,j)} = \left\{ \bigoplus_{x=0}^{k-2} r^{x}_{x.i+j} \right\} \oplus Se_{j-i}$$ Où $\oplus$ représente un XOR, $0 \leq i \leq$ n-1 et $0 \leq j \leq p$-2.

A la fin de cette phase la station de base concatène toutes les partitions calculées $a_i = a_{(i,0)} \,||\, \ldots \,||a_{(i,p-2)}$, et envoi le secret généré aux nœuds de transmission *FN* correspondants.

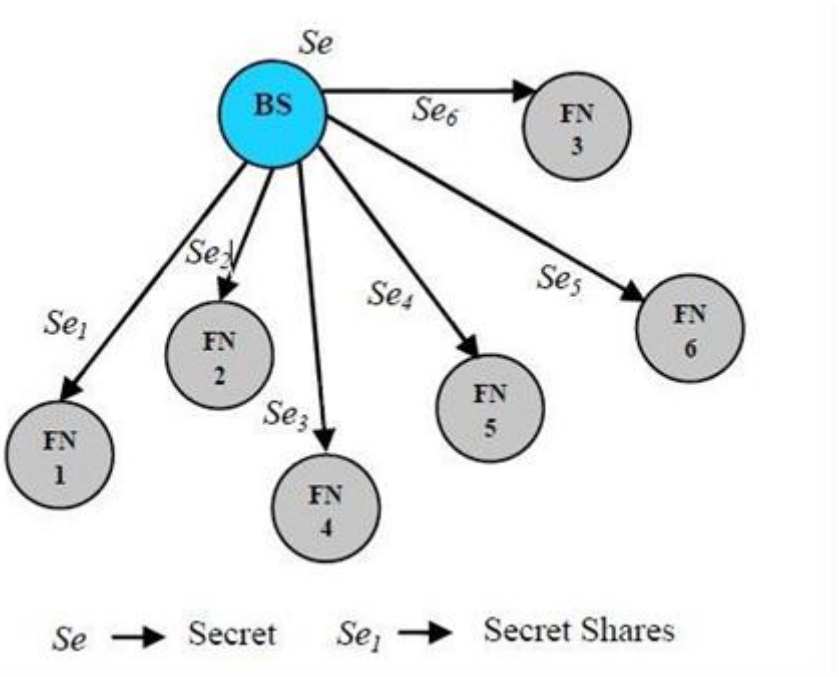


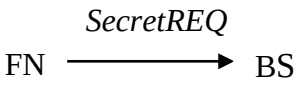
Figure 10. Système de distribution du secret partagé

Phase de recouvrement du secret à seuil

Une fois que chaque part de secret est divisé en séquences de r bits, un vecteur de dimension D_V de *k (P-1)* est produit. Ce vecteur symbolise les pièces partitionnées du secret partagé. Après cette étape, une matrice binaire *R* est générée par *k(P-1)*k(P-1)* en utilisant une procédure spécifique appelé *GEN* (Générer). Par le biais de la matrice *R*, les secrets partagés Se_1, Se_2, ... Se_n sont retrouvés. Enfin, les secrets partagés Se_1, Se_2 ... Se_n sont joints et le secret *Se* est obtenu.

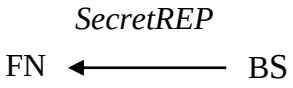
Canal sécurisé pour la transmission de la clé

Le système suppose que chaque nœud de détection S_i ($i = 1,2, .. n$), *BS* et *FN* sont pré chargés d'une clé secrète initiale K_{ini} et des certificats publics $Cert_{pu,}$ et privés $Cert_{pr}$ produits par une autorité de certification (*CA*). *FN* demande à la *BS* de transmettre le secret par une requête qui comprend l'identifiant du nœud de détection FN_{ID}, le nonce généré par le nœud de détection FN_N, le certificat publique de la station de base *E* ($Cert_{pu}(BS)$), et un code d'authentification du message $MAC_{Kini}(S_{ID})$, FN_{ID}, FN_N.



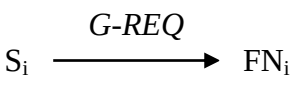
SecretREQ*:* FN_{ID}, $E_{(Certpu(BS)}(FN_N)$, $MAC\ K_{ini}(S_{ID} || FN_{ID} || FN_N)$.

Dès la réception de la requête *SecretREQ*, la *BS* décrypte le nonce utilisant son $Cert_{pr}$ et vérifie le *MAC* à l'aide de sa clé secrète initiale. Si la vérification réussie, alors *BS* transmet à la *FN* le secret correspondant par un message de réponse secret.



SecretREP: S_{ID}, $E_{(Certpu(FN)}(Se_i)$, $MAC\ K_{ini}(S_{ID} || FN_{ID} || FN_{BS})$.

FN décrypte le secret en utilisant son $Cert_{pr}$ et vérifie le *MAC*. Une fois la vérification réussi, il accède à la valeur secrète et récupère la clé du groupe à l'aide du système de secret à seuil. Les nœuds de détection S_i transmettent une requête *G-REQ* à FN_i demandant ainsi la clé du groupe.



G-REQ : S_{ID}, $E_{(Certpu(FN))}(S_N)$, $MAC\ K_{ini}(S_{ID} || FN_{ID} || S_N)$.

Après la réception d'un message *G-REQ*, FN_i effectue le déchiffrement et vérifie l'authentification en utilisant la clé secrète initiale, une fois l'authentification réussie, FN_i décide de transmettre la clé du groupe ($G_{K(i)}$) en utilisant le message *G-REP*.

G -REP : FN_{ID}, $E_{(Certpu(S\,id))}(G_{K(i)})$, $MAC\ K_{ini}(S_{ID} \| FN_{ID} \| FN_N)$.

Le nœud (S_i) vérifie le MAC puis accédé à la clé de groupe par décryptage de la valeur avec $Cert_{pr}(S_{id})$. Chaque nœud utilise cette clé ($G_{K(i)}$) pour une transmission ultérieure de données dans le réseau. ($G_{K(i)}$) obtenu du groupe *i* est divisé en deux clés, une clé privée ($Pri\ G_{K(i)}$) et une clé publique ($Pub\ G_{K(i)}$) pour plus de cryptage et processus de décryptage au lieu de $Cert_{pu}$ et $Cert_{pr}$.

Les résultats de simulation du protocole HGKMTC [SS13] par rapport à EEKM [PKHLS08] présentés dans [SS13], ont montré que le taux de livraison des paquets du premier protocole est plus élevé que le deuxième. Tandis que le nombre moyen des paquets perdus, l'énergie ainsi que le retard de transmission sont manifestement inférieur à ceux de la technique EEKM [PKHLS08] et donc meilleurs.

2.5. Etude comparative

Nous présentons une comparaison des protocoles cryptographiques décrits précédemment en se basant sur les métriques d'évaluation citées au début du chapitre.

2.5.1. Charge de calculs

En comparant les travaux étudiés selon cette métrique, nous relevons de manière générale qu'ils sont couteux en termes de charge de calculs. Nous plaçons en première position le travail de Sliti et al. [SHB08] ainsi que le travail de Jing-feng et al. [JDH08] dont la charge est approximativement la même, car les Clusters Head sont chargés de combiner et vérifier les signatures partielles des nœuds capteurs et construire une seule signature valide à l'aide de plusieurs formules mathématiques complexes. Par la suite nous plaçons le protocole de Singh et al. [SS13] et celui de

Koschuch et al. [KHKLW10], pour le premier les Clusters Head doivent récupérer les partitions de la clé afin de les transmettre aux capteurs de leurs groupes. Et pour le deuxième protocole les capteurs combinent l'ensemble des signatures avec l'interpolation de Lagrange et calculent leurs signatures complètes.

2.5.2. Charge de stockage

Du fait que dans le travail de Singh et al. [SS13] les nœuds de transmission (Clusters Head) enregistrent toutes les partitions du secret partagé, ainsi que des requêtes reçues de la part de la station de base et les nœuds capteurs, nous concluons que la charge de stockage de ce protocole est élevée. De même pour le travail de Sliti et al. [SHB08] chaque capteur doit enregistrer le certificat public, le certificat privé et la clé transmise par l'autorité de certification, et le Cluster Head enregistre toutes les signatures numériques reçues de la part des *k* capteurs (*k* est le seuil). Pour le protocole de Jing-feng al. [JDH08] la charge de stockage est moyenne, car les capteurs enregistrent les signatures qu'ils génèrent et d'autres valeurs numériques transmise du PKG, et le cluster Head doit enregistrer toutes les signatures partielles des nœuds du Cluster et les valeurs R_{1i}, R_{2i} transmises de la part des capteurs. Pour Koschuch et al. [KHKLW10] la charge de stockage est également moyenne malgré que chaque capteur enregistre son polynôme h_i qu'il génère et (n-1) polynômes $h_i(j)$, ou n est le nombre de capteurs du réseau et $j \neq i$, car la taille de stockage d'un polynôme est inferieur par rapport à la taille d'une requête ou un certificat.

2.5.3. Charge de communication

Nous allons étudier la charge de communication au niveau d'un Cluster Head et puis au niveau d'un capteur :

- Au niveau du Cluster Head :

Le travail de **Singh et al. [SS13]** dépasse de loin les autres protocoles car les paquets transmis ont une taille de 4096 bits (512octets) et donc pour chaque évènement, le Cluster Head (CH) effectue (2N+2) communications dont une émission et une réception avec la station de base et 2*N communications avec ses capteurs, (N

est le nombre de capteurs dans le Cluster) et le travail de Jing-feng al. [JDH08] est moyen car (CH) effectue N*500*(2 réceptions + 2 émission) avec les capteurs, et nous trouvons que la charge de communication des travaux de Koschuch et al. [KHKLW10] et Sliti et al. [SHB08] sont faibles car au niveau du Cluster Head, le nombre de communications est faible, le premier exécute deux communications de taille 500 bits et le deuxième une seule réception de la même taille.

- Au niveau d'un seul capteur :

Nous trouvons que la charge de communication du travail de Sliti et al. [SHB08] est élevée jusqu'à un nombre maximum de 241 nœuds, car un capteur effectue une émission et une réception de taille de 500bits pour chacune des communications et reçoit deux certificats de taille de 15ko pour chacun ainsi qu'une clé de taille de 170*bits*. Dans le travail de Koschuch et al. [KHKLW10], un capteur effectue une seule émission et une seule réception avec le Cluster Head et 2* (N-1) communications avec les autres capteurs, une fois le nombre de capteur dépasse 242, le travail de Koschuch et al. [KHKLW10], devient plus élevé que celui de Sliti et al. [SHB08]. Dans le travail de Jing-feng al. [JDH08] et Singh et al. [SS13] la charge de communication est faible car dans le premier, un capteur effectue deux émissions et deux réceptions de taille 500 *bits* chacune et dans le deuxième une seule émission et une seule réception de taille 4096 *bits* chacune.

2.5.4. La scalabilité

Évaluer la scalabilité d'un réseau de capteur sans fil dépend fortement du nombre de nœuds ajoutés au réseau, pour cela nous allons comparer la scalabilité des protocoles étudiés en se basant sur la taille du trafic entre les nœuds capteurs; nous remarquons que la scalabilité des travaux de Singh et al. [SS13] et Koschuch et al. [KHKLW10] est faible car la taille des communications entre les nœuds du réseau est élevée quand le réseau atteint 1000 nœuds. Le travail de Jing-feng al. [JDH08] et celui de Sliti et al. [SHB08] sont de scalabilité moyenne, puisqu'ils assurent une augmentation régulière de la taille du trafic par rapport aux nombre de nœuds et cela jusqu'à 1000 nœuds.

2.6. Comparaison

L'étude du comportement des protocoles traités ci-dessus et leurs comparaisons entre eux selon des métriques précises nous ont permis de dresser ce tableau comparatif.

	Jing-Feng et al.	**Sliti et al.**	**Koschuch et al.**	**Singh et al.**
Charge de calcul	Elevée	Elevée	Moyenne	Moyenne
Charge de stockage	Moyenne	Elevée	Moyenne	Elevée
Charge de communication	Moyenne	Faible	Moyenne	Elevée
Scalabilité	Moyenne	Moyenne	Faible	Faible

Tableau 1. Tableau comparatif des solutions étudiées

2.7. Conclusion

Dans ce chapitre, nous avons donné une vue d'ensemble sur les critères concernant la cryptographie partagée dans les réseaux de capteurs sans fil : la charge de calcul, la charge de stockage, la charge de communication, et la scalabilité. Nous avons aussi résumé les solutions existantes. Puis nous les avons comparé entre elles par rapport aux critères choisis.

Chapitre 3

Un protocole de cryptographie partagée

3.1. Introduction

Etablir une communication sécurisée pour assurer l'intégrité des données échangées est indispensable pour la majorité des applications dans les réseaux de capteurs sans fil. Le problème est comment conserver l'état d'une donnée lors de son traitement, de sa conservation ou de sa transmission afin de ne subir aucune altération ou destruction volontaire ou accidentelle. La gestion des clés est la brique de base pour assurer cette protection dans ce type de réseau. Cependant, fournir une gestion efficace des clés est difficile en raison de la nature ad hoc, la connectivité intermittente et la limitation des ressources des réseaux de capteurs. Dans ce chapitre, nous proposons un protocole cryptographique inspiré du protocole de Merkel et Hellman en utilisant la cryptographie partagée qui permet la collaboration de tous les nœuds du réseau afin que chacun chiffre une partie du message et l'envoie au nœud principal qui va par la suite restituer le message initial représentant l'évènement détecté.

3.2. Architecture du réseau

Les réseaux de capteurs sans fil sont constitués d'un ensemble de nœuds communiquant via des interfaces sans fil sans infrastructure fixe. De tels réseaux, à large échelle, offrent des perspectives d'applications très intéressantes cependant ils nécessitent des algorithmes économes en énergie de façon à ne pas dépasser leurs capacités matérielles.

L'approche de clustering consiste à partitionner le réseau en un certain nombre de clusters, homogènes (en termes de caractéristiques) selon une métrique spécifique ou une combinaison de métriques pour former une topologie virtuelle. Les clusters sont généralement identifiés par un nœud particulier appelé Cluster-Head (CH). Ce dernier permet de coordonner entre les membres de son cluster et d'agréger les données

collectées et de les transmettre à la station de base. Le cluster-Head est sélectionné soit d'une manière déterministe (chef de cluster prédéfini) ou d'une manière aléatoire (Cluster Head élu parmi les nœuds du réseau selon une métrique bien particulière ou une combinaison de métriques). Il existe plusieurs méthodes de clustering, la plus répondu s'exécute comme suit :

- Chaque nœud devra connaître son voisinage par le biais des messages Hello.
- Chaque nœud prend la décision selon sa connaissance locale de la topologie pour être Cluster-Head ou non.
- Le nœud choisi comme Cluster-Head diffuse son statut dans son voisinage et invite ses voisins qui ne sont pas encore affiliés à d'autres clusters de le rejoindre.
- Chaque Cluster est formé de telle sorte que sa taille soit égale à un nombre *n* bien défini de nœuds.

3.3. Hypothèses et notations

En premier lieu nous avons posé quelques hypothèses afin de tester notre protocole, cela nous facilite l'analyse de son comportement, par la suite nous prévoyons d'étendre nos tests sur d'autres environnements :

- Les capteurs sont statiques.
- Le réseau est structuré sous forme d'un ensemble de clusters de tailles *n*, tel que *n* représente la taille en bits de la donnée relative au phénomène observé par le réseau.
- Les canaux de communications sont considérés fiables.

Nous utilise les notations suivantes dans notre proposition :

- $\boldsymbol{s_i}$ **:** dénote l'identifiant (unique) du capteur i.
- $\boldsymbol{\{ai, w, n\}_k}$ **:** le chiffrement de la clé privée (*ai*, *w*, *n*) en utilisant la clé symétrique *k.*
- ***Na*** **:** nonce généré par le Cluster-Head dans le but de mettre à jour un nœud.

- $S_i = a_i^{w+bi+Na} \bmod n$ **:** chiffrement du i-ème bit du message par le i-ème nœud de capteur.
- $S_i' = a_i^{w+Na} \bmod n$ **:** valeur calculée par le Cluster-Head pendant la phase de recouvrement du message.

3.4. Notre proposition

Avant le déploiement du réseau, chaque nœud est préconfiguré par un ensemble de clés symétriques avec le reste des nœuds du réseau. Le but principal de l'usage de cette clé est de sécuriser la distribution des parts de la clé aux nœuds de chaque cluster. Après le déploiement, chaque capteur doit découvrir ses voisins afin d'identifier les clés symétriques dont il a besoin et procède par la suite à la suppression des clés des nœuds qui ne sont pas dans sa portée afin de gagner de l'espace mémoire. Notre proposition est basée sur la cryptographie partagée et se déroule en différentes phases que nous décrivons ci-après.

3.4.1. Phase de génération de la clé

Le Cluster Head génère une suite super-croissante $A = \{a_1, a_2, a_3, ..., a_i\}$ et deux nombres *w* et *n* tel que *n* est supérieur à la somme de tous les a_i, et *w* n'a pas de facteur commun avec aucun nombre de la suite.

3.4.2. Phase de partage de la clé

Pour distribuer la clé privée (*ai*, *w*, *n*) à tous les nœuds dans le cluster, le Cluster Head la chiffre à l'aide de la clé symétrique qu'il partage avec ses capteurs afin que ces derniers cryptent partiellement leurs message de façon individuelle.

3.4.3. Phase de collecte des parts du message

Le Cluster Head génère un nonce de façon aléatoire avec une taille maximal de 8 bits et le distribue aux différents capteurs de son cluster. Une fois que les capteurs ont détecté un événement et intercepté la valeur *M* dont nous avons supposé que sa taille est égale au nombre de nœuds dans le cluster, chaque capteur chiffre individuellement

le bit du message qui coïncide avec son identifiant tel que $S_i = a_i^{\,w+bi+Na} \bmod n$, ou b_i est le i-eme bit du message M puis chacun envoie son bit chiffré au Cluster Head.

3.4.4. Phase de recouvrement du message

Cette phase se déroule au niveau du Cluster Head. A la réception des bits chiffrés en provenance des capteurs de son cluster, le Cluster Head entame la phase de recouvrement afin de reconstituer le message. Pour ce faire, le Cluster Head calcul $S_i' = a_i^{w+Na} \bmod n$ et compare le résultat avec la valeur de S_i reçue. Deux cas sont possibles :

- Si $S_i = S_i'$ cela signifie que le bit $b_i = 0$.
- Si $S_i \neq S_i'$ cela signifie que le bit $b_i = 1$.

Enfin, le cluster Head concatène les bits b_i selon l'ordre croissant des identifiants afin de reconstituer le message qui sera transmis à la station de base.

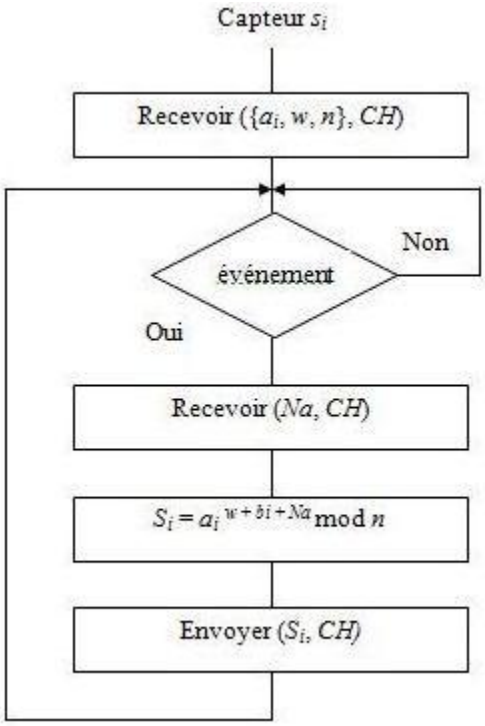


Figure 11. Organigramme de transmission des données captées (exécutée par le capteur s_i)

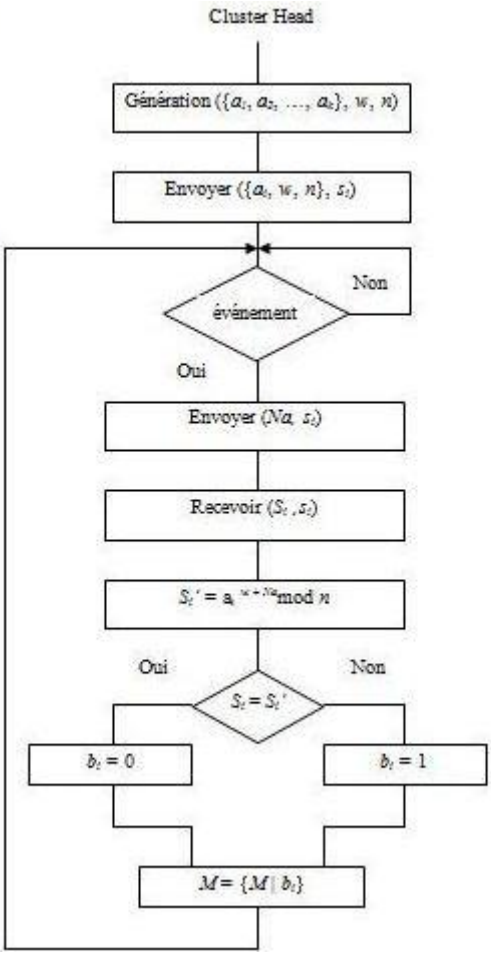


Figure 12. Organigramme de reconstruction de la donnée captée (exécutée par le Cluster-Head)

3.4.5. Exemple d'illustration du protocole

Le Cluster-Head génère une suite super-croissante A = {2, 3, 6, 13, 27, 52}, w = 31, et n = 105. Ensuite, il distribue à chacun des capteurs une partie de la clé privée.

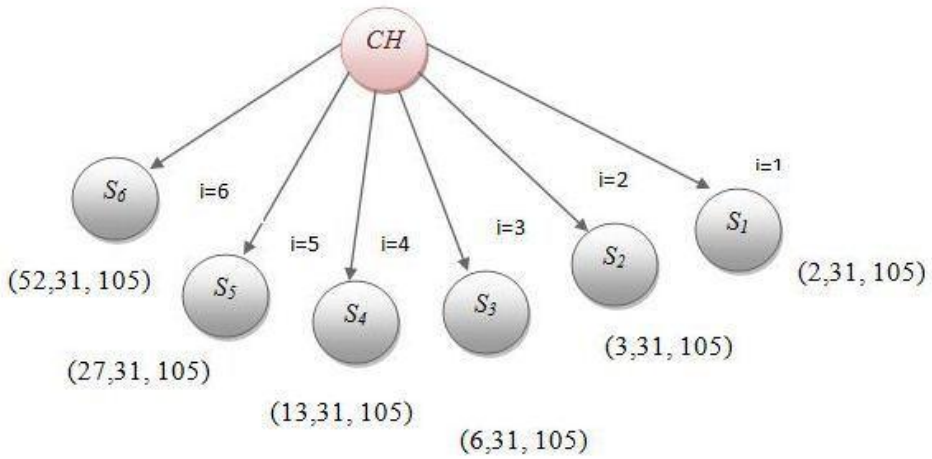


Figure 13. Distribution des clés privées.

Soit un évènement capté M = 53 = $(110101)_2$, le Cluster-Head génère un nonce *Na*=50 et le diffuse aux capteurs. Chaque capteur chiffre le bit du message qui correspond à son identifiant et l'envoie au Cluster-Head comme suit :

- $S_1 = 2^{31+1+50} \bmod 105 = 79$.
- $S_2 = 3^{31+1+50} \bmod 105 = 39$.
- $S_3 = 6^{31+0+50} \bmod 105 = 6$.
- $S_4 = 13^{31+1+50} \bmod 105 = 64$.
- $S_5 = 27^{31+0+50} \bmod 105 = 27$.
- $S_6 = 52^{31+1+50} \bmod 105 = 4$.

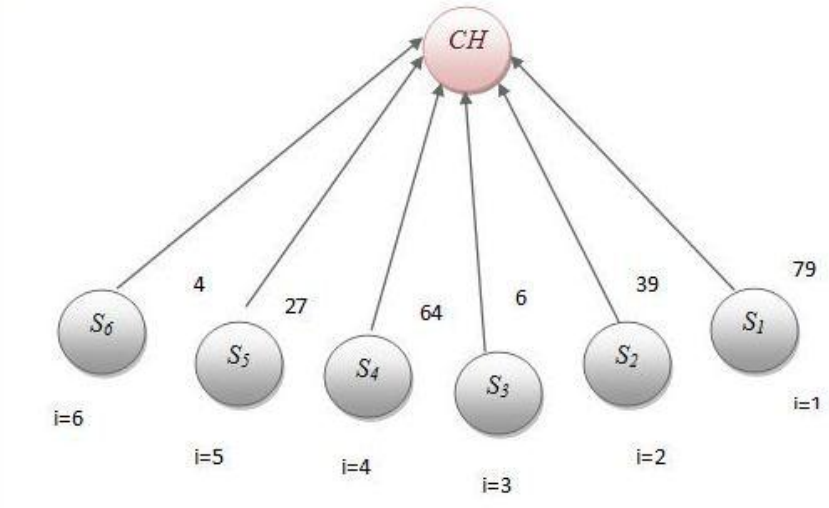


Figure 14. Distribution du message chiffré

Le Cluster Head calcule les $S_i^{'}$ comme suit :

- $S_1' = 2^{31+50} \bmod 105 = 92$, $S_1 \neq S_1 \Rightarrow b_1 = 1$.
- $S_2' = 3^{31+50} \bmod 105 = 48$, $S_2 \neq S_2' \Rightarrow b_2 = 1$.
- $S_3' = 6^{31+50} \bmod 105 = 6$, $S_3 = S_3' \Rightarrow b_3 = 0$.
- $S_4' = 13^{31+50} \bmod 105 = 13$, $S_4 \neq S_4' \Rightarrow b_4 = 1$.
- $S_5' = 27^{31+50} \bmod 105 = 27$, $S_5 = S_5' \Rightarrow b_5 = 0$.
- $S_6' = 52^{31+50} \bmod 105 = 97$, $S_6 \neq S_6' \Rightarrow b_6 = 1$.

La concaténation des bits selon l'ordre des identifiants des capteurs permet de reconstituer le message M' = M = $(110101)_2$ = 53 qui correspond au message initial.

3.5. Conclusion

Dans ce chapitre, nous avons proposé un protocole de cryptographie partagée dans les réseaux de capteurs sans fil. En premier, nous avons décrie les hypothèses et l'architecture du réseau, et enfin notre proposition. Dans le chapitre suivant, nous présentons les résultats de simulations.

Chapitre 4

Evaluation de performances

4.1. Introduction

Dans ce chapitre, nous nous évaluons les performances des protocoles étudiés précédemment en comparaison avec notre protocole en termes de consommation d'énergie et de charge de communication.

4.2. Environnement et paramètres de simulations

Les simulations ont été faites sous l'environnement *Matlab*. Nous avons opté pour une durée de simulation de 900*s*. Le simulateur estime si un lien radio existe entre deux nœuds quelconques en fonction de la distance qui les sépare. Le nombre de nœuds dans le réseau est 100 à 200, chaque nœud possède une portée de signal de 10*m* à 50*m*, et sur une surface rectangulaire de 1000m^2. Les nœuds ont les mêmes caractéristiques matérielles et la même puissance de traitement, et sont configurés par des interfaces de communication sans fil avec un débit de 22 *Mbps*, une énergie empirique de $2*10^{(-6)}$ *joule*, énergie électrique de $1*10^{(-6)}$ *joule* et une énergie initiale de 1 *joule*. La taille de chaque cluster est fixé à 9 nœuds ce qui correspond à une taille de donnée de 8 bits.

Les critères évalués sont : l'énergie moyenne consommée et la charge de communication. L'énergie consommée est le taux d'énergie perdue, la transmission des données qui se révèle extrêmement consommatrice par rapport aux tâches d'un capteur. Cette caractéristique conjuguée a pour objectif de maximiser la durée de vie du réseau. Nous avons mesuré la charge de communication selon le nombre de paquets transmis et la taille du trafic existant. Les impacts étudiés sont respectivement : le nombre de nœuds dans le réseau et la portée du signal.

4.3. Résultats obtenus

Dans cette section, nous nous sommes intéressés à comparer les performances des quatre protocoles avec notre proposition. Nous avons varié le nombre de nœuds dans le réseau ainsi que la portée en observant la consommation moyenne d'énergie et la charge de communication du réseau. Les résultats des simulations sont illustrés dans les figures suivantes :

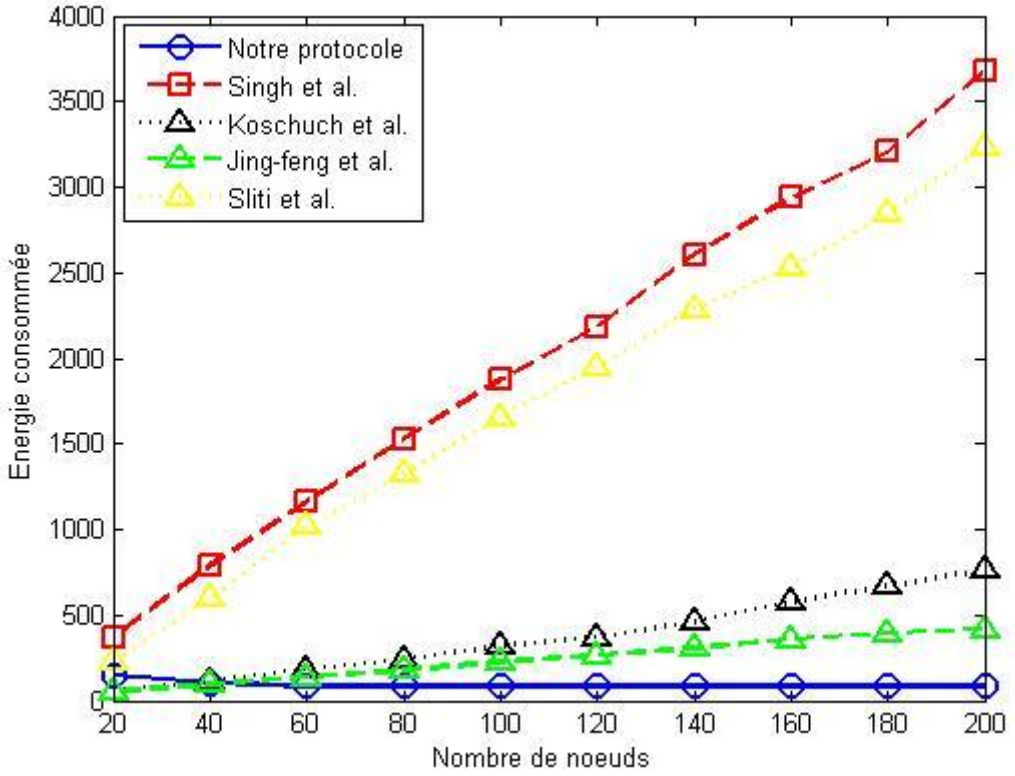


Figure 15. Energie moyenne consommée par rapport au nombre de nœuds

La figure 15 compare l'énergie moyenne consommée en fonction de la taille du réseau. Nous remarquons que l'énergie augmente quand le nombre de nœuds augmente, alors que pour notre protocole, l'énergie reste constante. Quand le nombre de nœuds dépasse 40, notre protocole donne de bons résultats. L'information étant codée sur 8 bits, nécessite 8 capteurs dans le cluster. Ainsi le protocole proposé ne consomme pas plus d'énergie même si le nombre de nœuds dans le réseau augmente, contrairement aux autres protocoles.

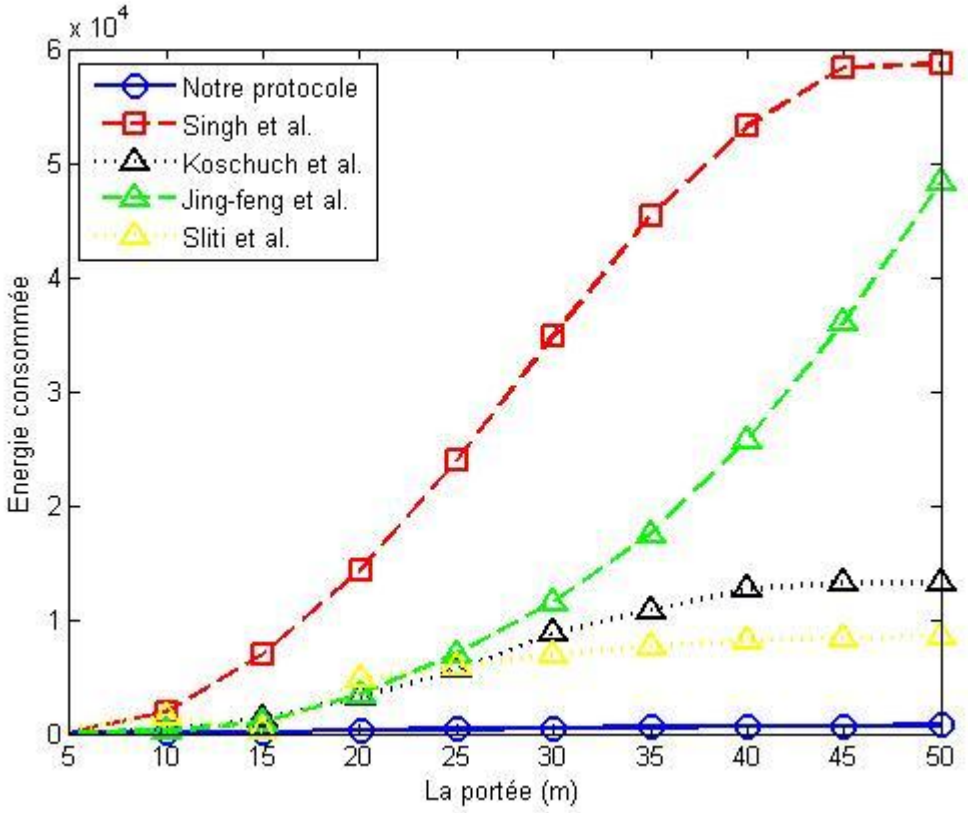


Figure 16. Energie moyenne consommée par rapport à la portée

La figure 16 illustre une comparaison de notre protocole avec les autres protocoles en termes de consommation d'énergie en fonction de la portée de signal. Nous constatons que l'énergie consommée est faible pour notre protocole alors qu'avec les protocoles de Singh et al. [SS13] et Jing-feng et al. [JDH08], celle-ci augmente exponentiellement respectivement pour une portée de $10m$ et$15m$. Dans les protocoles de Koschuch et al. [KHKLW10] et Sliti et al. [SHB08], l'énergie augmente de façon moindre. Ceci est dû au fait qu'une grande portée détecte plus de nœuds dans le cluster, ce qui augmente la consommation d'énergie.

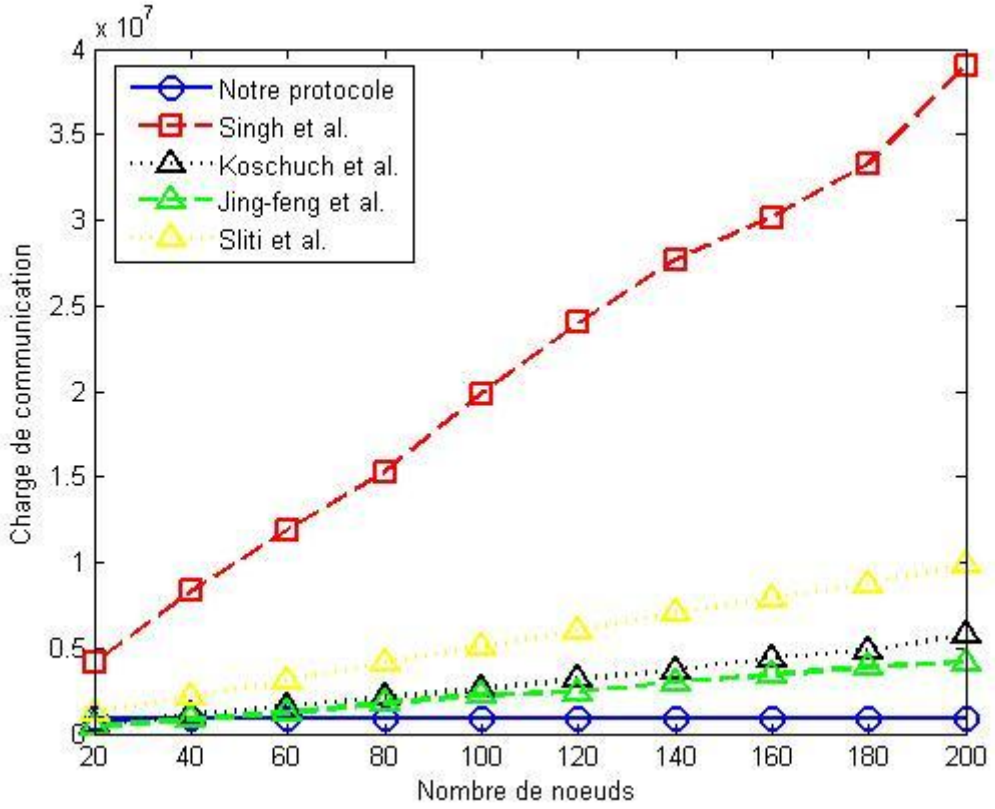


Figure 17. Charge de communication par rapport au nombre de nœuds

En termes de charge de communication, la figure 17 montre une différence importante entre le protocole Singh et al. [SS13] et les autres. Ce résultat est crédible dans le sens où la taille des paquets (*512 octets*) qui transite dans le cluster introduit une forte charge de communication dans le réseau. Dans le cas de notre protocole, la charge de communication est constante à cause du nombre de nœuds fixe dans le cluster.

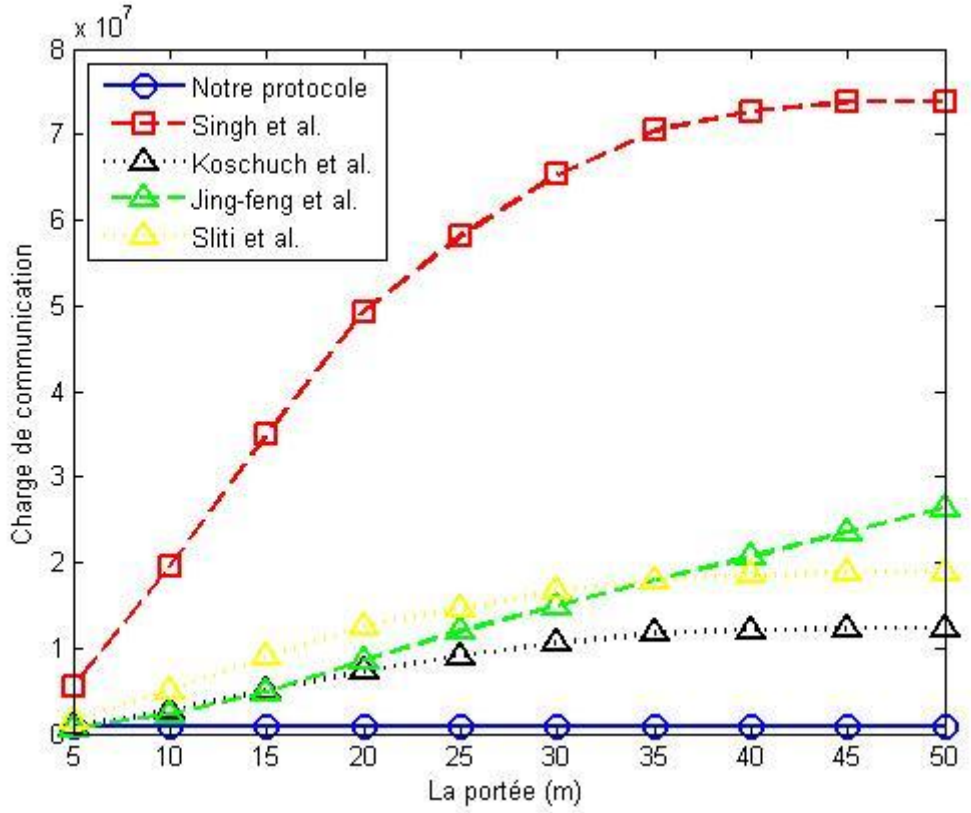


Figure 18. Charge de communication par rapport à la portée

Dans la figure 18, nous présentons l'évolution de la charge de communication en fonction de la portée de transmission pour les cinq protocoles. On constate une légère hausse de la charge de communication pour les protocoles Sliti et al. [SHB08], Jing-feng et al. [JDH08] et Koschuch et al. [KHKLW10] et une hausse significative pour le protocole de Singh et al. [SS13], tandis que pour notre protocole elle est constante.

4.4. Conclusion

Les résultats de simulations ont montré que le nombre de nœuds dans le réseau et la portée ont un impact significatif sur les performances des protocoles. Les résultats obtenus démontre l'efficacité de notre protocole en prenant en compte les l'énergie consommé et la charge de communication des réseaux de capteurs sans fil.

Conclusion générale

Dans ce mémoire, nous nous sommes intéressés aux réseaux de capteurs sans fil, qui sont un cas particulier des réseaux ad hoc. Ce type de réseaux est caractérisé par l'absence d'infrastructure pour la gestion des échanges, et de ce fait, ils ont besoin d'être auto configurables. Les nœuds communiquent en multi-sauts et les entités interagissent essentiellement avec la nature ou l'environnement ou entre elles. Les réseaux de capteurs sans fil sont caractérisés par une forte densité, de ce fait, le nombre de nœuds est important et les capteurs peuvent être densément déployés [You12].

Nous avons présenté des généralités sur les réseaux de capteurs sans fil, où nous avons exposé leur architecture, leurs applications et leurs contraintes. Ensuite, nous avons présenté un état de l'art sur la cryptographie partagée dans les réseaux de capteurs sans fil, où nous avons développé une étude des différents protocoles et des comparaisons par rapport aux critères suivants : charge de calculs, stockage, communication et scalabilité. Ensuite, nous avons proposé un nouveau protocole de cryptographie partagée qui garantit l'intégrité (crédibilité) des données. Le protocole consiste à associer à chaque nœud dans un cluster une part de clé avec laquelle il chiffre un seul bit du message de l'événement détecté. Le nœud central (Cluster-Head) récupère les bits chiffrés des nœuds et reconstitue la donnée. Enfin, nous avons exposé la partie pratique du travail en simulant de notre proposition sous Matlab. Nous avons effectué une comparative avec tous les protocoles étudiés, pour évaluer leurs performances en termes de consommation d'énergie, charge de communication et dans lesquelles notre solution a fourni des résultats très intéressants.

En guise de perspective, nous envisageons d'élargir l'étude sur notre protocole en analysant sa robustesse face aux attaques.

Bibliographie

[CLW06] X. Cheng, J.Liu, X.Wang, *An Identity-based Signature and its Threshold version*, In Proceedings of IEEE AINA, 2006.

[DCL04] S. Duan, Z. Cao, R. Lu. *Robust ID-based threshold signcryption scheme from pairings.* In Proceedings of the 3rd international conference on Information security, 2004.

[DH76] W. Diffie, M. Hellman. *New directions in cryptography*. Institute of Electrical and Electronics Engineers, Transactions on Information Theory, 1976.

[Elg98] T. Elgamal. *A public key cryptosystem and a signature scheme based on discrete logarithms*. Springer Verlag, 1998.

[GRR98]R.Gennaro, M. O. Rabin, and T.Rabin, (1998). *Simplified VSS and fast-track multiparty computations with applications to threshold cryptography*.In Proceedings of the ACM Symposium on Principles of Distributed Computing, 1998.

[ISG00] RFC 2828. *Internet Security Glossary*, 2000.

[JA96] P. Johnson, D.C Andrews, *Remote continuous monitoring in the home*. Journal of Telemedicine and Telecare, 1996.

[JDH08] L.Jing-feng, W.Da-wei, K.Hong-zhao.*Secure Monitoring Scheme Based on Identity-based Threshold Signcryption for Wireless Sensor Networks.* In Proceedings of IEEE WiCom, 2008.

[KHKLW10] M.Koschuch, M.Hudler, M. Krüger, P. Lory, J.Wenzel. *Applicability of multiparty computation schemes for wireless sensor networks*. In Proceedings of IEEE DCNET, 2010.

[Leh09] M. Lehsaini, *Diffusion et couverture basées sur le Clustering dans les réseaux de capteurs : application à la domotique*, Thèse de Doctorat Université A.B Tlemcen Université de Franche-Comté, 2009.

[LM91] X. Lai, J. Massey. *Markov ciphers and differential cryptanalysis*. Cryptology - EUROCRYPT'91, Springer-Verlag, 1991.

[Lor07] P. Lory. *Reducing the complexity in the distributed multiplication protocol of two polynomially shared values*. In Proceedings of IEEE AINAW, 2007.

[Lor09] P. Lory. *Secure distributed multiplication of two polynomially shared values: Enhancing the efficiency of the protocol*. In Proceedings of IEEE SECURWARE '09, 2009.

[Mes08] M. L. Messai, *Sécurité dans les Réseaux de Capteurs Sans-Fil*, Université Abderrahmane Mira de Bejaia, 2008.

[NBS77] National Bureau of Standards. *Data encryption standard (DES)*. Federal Information Processing Standards Publication, National Technical Information Service, Springfield, 1977.

[NIST01] National Institute of Standards and Technology. *Specification for the advancedencryption standard (AES)*. FIPS 197, 2001.

[NSA93] National Security Agency. *Secure hash algorithm*. Federal Information Processing Standard du National Institute of Standards and Technology. 1993.

[PKHLS08] K. Paek, J. Kim, C. Hwang, S. Lee and U. Song, *Group-Based Key Management Protocol For Energy Efficiency In Long- Lived And Large-Scale Distributed Sensor Networks*. Computing and Informatics, 2008.

[PL05] C.Peng, X.Li, *An identity-based threshold signcryption scheme with semantic security*. In Proceedings of the international conference on Computational Intelligence and Security, 2005.

[RSA78] R. Rivest, A. Shamir, L. Adleman. *A Method for obtaining digital signatures and public-key Cryptosystems*. Communication of ACM, 1978.

[SEC00] Standards for Efficient Cryptography, "*SEC 1: Elliptic Curve Cryptography"*, 2000.

[Sha79] A. Shamir. *How to share a secret*.Communication of the ACM, 1979.

[Sha85] A. Shamir. *Identity-based cryptosystems and signature schemes*. Lecture Notes in Computer Science, Berlin, Springer-Verlag, 1985.

[SHB08] M. Sliti, M. Hamdi, N.Boudriga, *an Elliptic Threshold Signature Framework for k-Security in Wireless Sensor Networks*, In Proceedings of IEEE ICECS, 2008.

[SS13] K. Singh L. Sharma, *Hierarchical Group Key Management using Threshold Cryptography in Wireless Sensor Networks*, International Journal of Computer Applications, 2013.

[WFL04] X. Wang, D. Feng, X. Lai, H. Yu. *Collisions for hash functions MD4, MD5. Institute of Software*, Chinese Academy of Sciences, 2004.

[You12] Y. Younes, *Minimisation d'énergie dans un réseau de capteurs*, Mémoire de magister université de Tizi Ouzou, 2012.

[ZH99] L. Zhou, Z. Haas. *Securing ad hoc networks*. IEEE Networks, 1999.

.

Résumé

Les réseaux de capteurs sans fil se composent d'un grand nombre de nœuds qui communiquent entre eux pour le partage d'information et le traitement coopératif. Ces dispositifs sont déployés aléatoirement dans une zone d'intérêt pour superviser ou surveiller des phénomènes divers. Cependant, une attaque sur les capteurs ou l'intervention humaine sur la zone peut nuire à l'intégrité des données captées. Dans ce mémoire, une technique assurant l'intégrité des données captées en utilisant la cryptographie partagée dans les réseaux de capteurs sans fil est proposée. La technique considère un réseau de capteurs, où les nœuds sont coordonnés par des Clusters Head qui à leurs tours, sont connectés à la station de base. Les capteurs chiffrent collectivement le message et l'envoient au Cluster-Head qui enfin concatène et récupère les bits du message. La technique proposée est simulée en utilisant l'environnement de développement Matlab et les résultats des simulations montrent les performances de notre protocole.

***Mots clés** : Réseaux de capteurs sans fil, Intégrité, Crédibilité, Cryptographie partagée.*

Abstract

Wireless sensor networks consist of a large number of nodes, which communicate among them in order to share information and treatments. Sensors are randomly deployed in a zone of interest to observe diverse phenomena. However, an attack against sensors or the human intervention in the zone can deny the integrity of data. In this report, a technique assuring the data integrity using shared cryptography in wireless sensor networks is proposed. The technique considers network of sensors spread randomly, where sensors are coordinated by the Clusters Head and in a turn, they are connected to the base station. Sensors encrypt collectively the data and send it to the Cluster-Head and the latter finally concatenates and gets back the message. The proposed technique is simulated using the Matlab development environment in which the results show the pertinence of our protocol.

***Keywords:** Wireless sensor networks, Integrity, Credibility, Shared cryptography.*