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

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

Mémoire de fin de cycle

En vue de l'obtention du diplôme de Master en Informatique
Option : Réseaux et Systèmes Distribués

Thème

Implimentation de la cryptographie à base d'identité dans les réseaux de capteurs sans fil

Réalisé par :

MOKRANI Yanis
TOULA Souad

Devant le jury composé de :

Président :	M^r *TARI* AbdelKamel	M.C.A, Université A. MIRA de Bejaia
Examinateur :	M^r *HAMOUMA* Moumen	M.A.A, Université A. MIRA de Bejaia
Examinateur :	M^r *MOHAMMEDI* Mohamed	Doctorant, à Université A. MIRA de Bejaia
Encadreur :	M^r *OMAR* Mawloud	M.C.B, Université A. MIRA de Bejaia
Co-encadreur :	M^{me} *OUADA* Farah Sarah	Doctorante, à Université A. MIRA de Bejaia

Année universitaire 2013/2014

Remerciements

Nos vifs remerciements vont d'emblée à Dieu tout puissant qui nous a doté d'une grande volonté et d'un savoir adéquat pour mener à bien ce travail.

Nos remerciements sont adressés également à Monsieur OMAR Mawloud pour son encadrement et pour l'encouragement et l'intérêt qu'il nous a apporté pour l'accomplissement de ce projet de fin de cycle et surtout pour sa grande aide et ses qualités humaines.

Nous tenons à remercier très chaleureusement notre Co-encadreur Madame OUADA Farah Sarah pour sa collaboration importante pendant la rédaction de ce mémoire, ses encouragements ainsi que leur conseille.

Nous poursuivons ces remerciements en saluant vivement les membres du jury pour l'honneur qu'ils nous ont fait en acceptant de juger ce travail.

Nous n'omettrons jamais d'exprimer toute notre gratitude à tous les membres du département d'informatique de l'université de Béjaia, que ce soit enseignants ou cadres administratifs, qui de près ou de loin n'ont épargné aucun effort pour que notre formation et nos travaux se termine dans de bonnes conditions.

Dédicaces

Nous dédions ce modeste travail :

A *nos chers parents.*

A *nos frères et soeurs.*

A *nos grands parents.*

A *tous nos amis.*

Yanis, Souad

Table des matières

Table des figures

Liste des tableaux

Liste des Acronymes

A	
ACK	Accusé de réception.
C	
CBID	La Cryptographie à Base d'Identité.
CC	Chef de Cluster .
CTR	CounTeR .
D	
DES	Data Encryption Standard.
DoS	Denial of Service.
G	
GPS	Global Position system.
H	
HMAC	Hachage de clés pour l'authentification de message.
I	
ID	Identifiant.
IBC	Identity-Based Encryption .
IBS	Identity-Based Signature .
IEEE	Institute of Electrical and Electronics Engineers.
M	
MAC	Message Authentication Code.
MPS	Most Probable Selfish.
MD5	Message Digest5.

P	
PKG	Private Key Generator.
R	
RCSF	Réseau de capteurs sans fil.
RSSI	Received Signal Strength Indication.
REQ	Requet.
RepREQ	Réponse Requet.
RepREQ1	Réponse Requet1.
RREP	Reponse Requet .
RREQ	Route Requet.
RSA	Rivest-Shamir-Adleman.
S	
SB	Station de Base.
W	
WSN	Wireless Sensor Network.

INTRODUCTION GÉNÉRALE

Au cours de ces dernières années, le développement technologique des réseaux de communication sans fil, a connu un essor important grâce aux avancées technologiques dans divers domaines liés à la micro-électronique. C'est ainsi que de nouvelles voies d'investigation ont été ouvertes avec l'émergence des réseaux de capteurs sans fil (RCSFs), qu'ils sont constitués d'un grand nombre de dispositifs physique appelés nœuds ou capteurs. Ils collaborent entre eux pour former un RCSF capable de superviser une région ou un phénomène d'intérêt, et de fournir des informations utiles par la combinaison des mesures prises par les différents capteurs et de les communiquer via des communications multi-sauts, jusqu'à atteindre les stations de base qui sont des points de collecte des données captées. Les stations de base à leur tour, communiquent ces données à l'utilisateur via Internet ou par satellite. Les réseaux de capteurs ont de nombreuses perspectives d'applications dans des domaines très variés : applications militaires, domotique, surveillance industrielle ou de phénomènes naturels, relevés de compteurs. Le faible coût de construction et la facilité de déploiement de tels réseaux ont contribué à leur popularité croissante.

L'élargissement du domaine d'application des réseaux de capteurs nécessite plus de sécurité pour assurer l'intégrité, l'authenticité et la confidentialité des données qui circulent sur le réseau. En effet, les réseaux de capteurs sont confrontés à de nombreux problèmes liés à leurs caractéristiques qui rendent les solutions de sécurité développées pour les réseaux filaires ou sans fil avec infrastructure inapplicable dans le contexte des réseaux de capteurs. Parmi les vulnérabilités qui touchent les réseaux de capteurs, on trouve l'absence d'infrastructure, une topologie souvent aléatoire, la vulnérabilité des nœuds eux-mêmes et du canal radio ainsi que les ressources limitées (énergie limitée, capacité de calcul et de stockage réduite ...). Par conséquent, de nombreux travaux de recherches ont surgi au cours des dernières années pour proposer des solutions de sécurité permettant de remédier à ces vulnérabilités et d'assurer les services de sécurité dans les réseaux de capteurs. Ces solutions se basent principalement sur des outils cryptographique tels que la chiffrement symétrique ou asymétrique. Malgré la diversité de ces solutions, une sécurité parfaite est loin d'être évidente à cause de la multiplicité des vulnérabilités. Dans ce cas, un RCSF reste toujours vulnérable, ce qui motive une recherche intensive d'une solution de sécurité minimisant l'impact des différentes attaques possibles.

La communication dans les RCSFs est basée sur de différents paradigmes ; un-à-un, plusieurs-à-un et un-à-plusieurs. Dans ces différents types de communications, les paquets traversent le réseau saut-par-saut jusqu'à ce qu'ils atteignent leurs destinations, et chaque nœud intermédiaire assure le relais de ces paquets. Le rôle du nœud relais est de transmettre le paquet reçu au prochain saut après l'avoir traité. Donc la communication entre deux nœuds dans un environnement ouvert est confrontée aux risques qu'il y ait d'autres nœuds qui cherchent à emprunter un ensemble de nœuds légitimes pour s'approprier leurs données. Dans ce cas, un attaquant pourra facilement se joindre au réseau et injecter de fausses données ou modifiée celles qucette i passe par son chemin.

Notre travail entre dans le cadre de l'étude du problème d'authentification de bout-bout dans les réseaux de capteurs sans fils, tout en prenant en considération toutes les caractéristiques du réseau dans le but de garantir de meilleures performances . Notre étude offre, principalement, une étude synthétique des travaux de recherche qu'ont été fait de proposer un nouveau protocole d'authentification, dans le but de résoudre le problème d'acheminement de données entre les nœuds du réseau.Dans ce travail, nous proposons un protocole basé sur la cryptographie à base d'identité pour offrir l'authentification de bout-en bout dans les réseaux de capteurs nommé IBC2HAP (IBC based Hop by Hop Authentication Protocol).

Le reste de ce mémoire est structuré comme suit. Le premier chapitre est consacré à l'état de l'art dans le cadre des protocoles de routage sécurisé et la cryptographie à base d'identité. Dans le second chapitre, nous présentons en détail notre proposition. Dans le chapitre trois, nous présentons la modélisation analytique que nous avons élaborée afin d'évaluer notre proposition en termes de robustesse contre les attaques suivie d'une simulation afin de valider le modele analytique.

Chapitre 1

Etat de l'art sur le routage sécurisé dans les RCSFs

1.1 Introduction

Le routage sécurisé est une méthode d'acheminement des informations vers une destination donnée dans un réseau de connexion en toute securité. Le développement des protocoles de routage spécifique aux réseaux de capteurs à attiré une grande part d'intention parmi les chercheurs dans le domaine. Mais les limites imposées par l'architecture matérielle des capteurs, en particulier celle de l'énergie et l'hostilité des environnements, figurent parmi les facteurs primordiaux à prendre en compte lors de la conception d'un protocole de routage .

Ce chapitre présente un état de l'art sur les protocoles de routage sécurisé dans les RCSFs. Il fournit une classification sur la conception de ces protocoles en décrivant le principe général de quelques uns, ainsi que la comparaison entre eux par rapport à un ensemble de critères qui ont un impact considérable sur les performances de routage sécurisé. Pour mieux cerner les enjeux du sujet, nous présenterons d'abord les réseaux capteurs sans fil qui est susceptible d'être déployée dans de divers domaines applicatifs , ainsi que nous décrivant, leurs caractéristiques, et sur la sécurité des réseaux de capteurs. Nous consacrons par la suite le reste du chapitre à définire la cryptographie à base d'identité.

1.2 Les réseaux de capteurs sans fil

1.2.1 Définition

Un réseau de capteurs se définit comme un ensemble de capteurs connectés entre eux, où chaque capteur étant muni d'un émetteur-récepteur. Les réseaux de capteurs sans fil ou "Wireless Sensor Network " sont considérés comme un type spécial des réseaux Ad hoc où l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle des hôtes et des routeurs [18]. Chaque nœud est capable de surveiller son environnement et de réagir en cas de besoin en envoyant l'information collectée à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil [3].

1.2.2 Les Caractéristiques des réseaux de capteurs

- ***Energie limitée :*** les RCSF visent la consommation d'énergie puisque l'alimentation de chaque nœud est assurée par une source d'énergie limitée et généralement irremplaçable à cause de l'environnement hostile où il est déployé. De ce fait, la durée de vie d'un RCSF dépend fortement de la conservation d'énergie au niveau de chaque nœud.

- ***Modèle de communication*** :les nœuds dans les RCSF communiquent selon un paradigme plusieurs-à-un. En effets, les nœuds capteurs collectent des informations à partir de leur environnement et les envoient toutes vers un seul nœud qui représente le centre de traitement.

- ***Densité de déploiement*** : elle est plus élevée dans les RCSF que dans les réseaux Ad Hoc. Le nombre de nœuds peut atteindre des millions de nœuds pour permettre une meilleure granularité de surveillance. De plus, si plusieurs nœuds capteurs se retrouvent dans une région, un nœud défaillant pourra être remplacé par un autre. Cependant, la densité de déploiement donne naissance à des challenges pour la communication entre les noeuds. En effet, elle provoque des collisions ou des endommagements des paquets transmis.

- ***Absence d'adressage fixe des nœuds :*** les nœuds dans les réseaux sans fil classiques sont identifiés par des adresses IP. Cependant, cette notion n'existe pas dans les RCSF. Ces derniers utilisent un adressage basé sur l'attribut du phénomène capté, on parle donc de l'adressage basé-attribut. En effet, les requêtes des utilisateurs ne sont pas généralement destinées à un seul nœud, mais plutôt, à un ensemble de nœuds identifiés par un attribut.

- ***Limitations de ressources physiques :*** : A cause de la miniaturisation des composants électroniques, les performances des nœuds capteurs sont limitées. Par conséquent, les nœuds capteurs

collaborent en traitant partiellement les mesures captées et envoient seulement les résultats à l'utilisateur. Une autre conséquence , ces limitations imposent des portées de transmission réduites contraignant les informations à être relayées de nœud en nœud avant d'atteindre le destinataire. C'est la raison pour laquelle les RCSF adoptent des communications multi-sauts.

– ***Sécurité :*** En plus des problèmes de sécurité rencontrés dans les réseaux Ad Hoc en général, les RCSF rencontrent d'autres handicaps dus à leurs challenges, à savoir l'autonomie et la miniaturisation des capteurs. Cela engendre l'inapplicabilité des mécanismes de défense utilisés dans les réseaux Ad Hoc tout en créant d'autres mécanismes de sécurité pour les RCSF. De plus, l'absence d'une sécurité physique dans l'environnement hostile où ils sont déployés expose les nœuds à un danger qui tend vers la falsification de l'information. En effet, les nœuds capteurs eux-mêmes sont des points de vulnérabilité du réseau car ils peuvent être modifiés, remplacés ou supprimés.

1.2.3 Quelques problématiques dans les RCSFs

Les recherches dans le domaine des réseaux de capteurs ont révélé plusieurs problématiques, parmi les problématiques cruciales, nous pouvons citer :

– ***La sécurité*** : comme les noeuds sont dispersés dans une zone publique, ils doivent être capables de maintenir privées les informations qu'ils recueillent. Par conséquent, pour les applications qui exigent un niveau de sécurité assez élevé telles que les applications militaires, des mécanismes d'authentification, de confidentialité, et d'intégrité doivent être mis en place au sein de leur communauté. La puissance de calcul limitée des capteurs ouvre de véritables défis pour concevoir des algorithmes de cryptographie et des politiques de confiance spécifiques à ces réseaux.

– ***Le routage*** : le problème de routage consiste à déterminer un acheminement optimal des paquets à travers le réseau au sens d'un certain critère de performance. D'où les protocoles de routages conçus doivent, en plus de leurs fonctions classiques, participer à la synthèse et l'agrégation des données retournées aux utilisateurs, tout en considérant d'autres facteurs tels que les limitations matérielles, la mobilité et la consommation d'énergie. En assurant un routage performant en termes de minimisation de la consommation de l'énergie, du choix des routes optimales pour l'acheminement de l'information d'un capteur à la station de base et vice versa, de réduction du délai de délivrance des paquets, etc.

– ***La couche Mac*** : la spécificité des réseaux de capteurs sans fil nécessite le développement de nouveaux protocoles MAC qui s'adaptent aux contraintes imposées par ces réseaux. Ceci

dans le but d'améliorer le débit, minimiser la consommation d'énergie, optimiser le partage du médium ainsi que minimiser le délai de délivrance des paquets.

- ***La Diffusion de l'information*** : les protocoles de diffusion conçus pour les réseaux de capteurs doivent tenir compte de leurs spécificités ainsi que de leurs contraintes intrinsèques imposées [7].

- ***La localistion*** : une problématique liée à la localisation est celle du placement des noeuds dans un réseau de capteurs, c'est à dire comment positionner les noeuds les uns par rapport aux autres. Plusieurs applications dans les réseaux de capteurs dépendent lourdement sur l'habilité des noeuds à connaître leur position ou à découvrir la topologie globale du réseau. Au moment où ce problème peut être contourné par l'utilisation d'un système GPS dans les réseaux installés en plein air, cette solution ne peut être adaptée à ceux déployés à l'intérieur des bâtiments[14], car il cumule des handicaps, et il est disponible seulement en extérieur.

- ***La syncronisation*** : la synchronisation du temps est un challenge important et coûteux dans les réseaux de capteurs à communication multi-saut. De nombreuses applications de réseaux de capteurs demandent une synchronisation des horloges locales des noeuds. Par exemple, les capteurs collaborent entre eux pour parvenir à une tâche de détection plus complexe [9].

- ***La couverture*** : la problématique de la couverture consiste à profiter de la redondance, issue du déploiement aléatoire des noeuds sur la zone surveillée, pour procéder à leur mise en veille alternée. En d'autres termes, il s'agit d'ordonnancer les noeuds dans des ensembles d'activation disjoints, tout en respectant les contraintes de couverture et de connectivité [9].

Toutes ces problématiques sont des domaines de recherche actifs. Dans la suite, nous nous intéresserons particulièrement à celle de la sécurité.

1.3 La sécurité dans les réseaux de capteurs sans fil

La communication entre les noeuds capteurs fait l'objet de constantes recherches destinées à améliorer son rendement. Mais, au-delà de cette problématique, d'autres enjeux commencent à apparaître telle que la garantie de sécurité de communication. En effet, les mécanismes de routage sont d'autant plus critiques dans les RCSF que chaque noeud participe à l'acheminement des paquets à travers le réseau, les noeuds eux-mêmes sont des points de vulnérabilité du réseau car une attaque peut compromettre un composant laissé sans surveillance. Plusieurs solutions de sécurité ont été proposées et malgré la diversité de ces solutions, le problème de sécurité reste toujours posé.

Cette partie traite les problèmes de la sécurité dans les RCSFs qui diffèrent des autres réseaux en ce qu'ils offrent des restrictions plus sévères en termes d'énergie, capacités de traitement et de communication. Nous commencerons donc à étudier les menaces contre les RCSFs. Par la suite, nous étudierons les services de base de la sécurité à respecter pour éviter ces menaces et décrirons les différents mécanismes permettant d'assurer ces services.

1.3.1 Les attaques des les RCSF

Dans cette section, nous décrivons une liste non exhaustive mais représentative des attaques les plus courantes et connues, qui menacent les RCSFs :

- ***Attaque de déni de sérvice (DoS)*** : cette attaque vise à saturer le réseau en entier. Pour ce faire, elle consiste à provoquer une saturation ou un état instable des noeuds victimes en leur envoyant des données ou des paquets de contrôle de constitution inhabituelle [8].

- ***Attaque d'un noeud égoïste (Selfisch)*** :dans les RCSFs, les nœud dépendent de chacun l'autre pour transmettre les données sur les sauts multiples en envoyant des paquets. Un noeud égoïste peut décider de ne pas transférer les paquets pour autres noeud pour sauver ses propres ressources mais quand même utilisent le réseau pour envoyer et recevoir ses données. Un tel comportement égoïste peut dégrader les performances du réseau significativement [5].

- ***Brouillage (jamming)*** : c'est une attaque de type DoS qui vise les médias de communication utilisés dans les RCSF. L'attaquant peut émettre un signal d'une fréquence proche de celle utilisée dans le réseau afin de brouiller la communication. Cela empêche les nœud d'échanger les données et provoque l'indisponibilité des canaux de transmission sans fil dans les RCSF [10].

- ***Attaque de trou noir*** : l'attaque de trou noir consiste tout d'abord à insérer un noeud malicieux dans le réseau [5]. Ce noeud, par divers moyens, va modifier les tables de routage pour obliger le maximum de noeuds voisins à faire transiter leurs informations par lui. Ensuite, tel un trou noir dans l'espace, toutes les informations qui vont passer sur son chemin seront supprimés.

- ***Attaque Rejeu(Replay)*** :Dans ce cas, l'attaquant surveille les transmissions, intercepte les paquets de données et les rejoue [10].

1.3.2 Objectifs et services de base de la sécurité

Les objectifs de sécurité dans les RCSF ne sont pas différents de ceux dans les autres réseaux classiques. En effet, ils visent à assurer que l'information soit correcte, qu'elle n'ait pas été altérée et émane effectivement de la source légitime. La sécurité vise donc à assurer les services de base suivants :

- ***L'authentification*** : est un sérvice qui consiste à vérifier l'identité des nœuds. Pour assuer l'authentification dans les RCSF, il existe plusieurs mécanisme tels que : les MACs (Message Authentication Code) et la signature numérique.

- ***La non Répudiation*** : qssure que l'émetteur ne pourra pas nier l'envoie d'un message , et le récepteur ne pourra pas nier sa réception. La signature numérique met en œuvre ce service.

- ***L'itégrité*** : assure qu'un message transmis ne peut pas etre modifié pendant sa transmission par un nœud malicieux. Elle peut etre realiser par les mécanismes d'hachage.

- ***La disponibilité*** : Elle est assurée si seulement si la communication entre les nœuds est toujours possible, veut dire que l'envoie d'information ne doit pas être interrompu. A cause de l'absence d'une infrastructure un tel service est difficile à assurer dans les RCSFs.

- ***La confidentialité*** : assure que les données transmises ne sont divulgué que par le destinataire. Elle peut être assurée soit par le chiffrement symétrique ou asymétrique des données.

- ***Le contrôle d'accès*** : Consiste à empêcher l'accès au réseau à tout nœud étranger, il protège contre l'utilisation ou manipulation non autorisée des ressources.

1.4 Le routage sécurisé dans les RCSFs

Le routage sécurisé est une méthode d'acheminement des informations vers une destination donnée d'une manière sécurisée dans un réseau de connexion. Le développement des protocoles de routage spécifiques aux réseaux de capteurs a attiré une grande part d'intention parmi les chercheurs. Mais les limites imposées par l'architecture matérielle des capteurs, en particulier celle de l'énergie et l'hostilité des environnements, figurent parmi les facteurs primordiaux à prendre en compte lors de la conception d'un protocole de routage.De ce fait, les protocoles élaborés doivent assurer une consommation minimale d'énergie tout en maintenant le bon fonctionnement du réseau et sans

dégrader ses performances.

1.4.1 Classification des protocoles de routage dans les RCSFs

Dans les réseaux de capteurs sans fil, chaque nœud communique directement avec son voisin, et pour communiquer avec d'autres nœud du réseau, il est nécessaire de faire passer les données par des nœuds intermédiaires afin d'acheminer l'information au nœud destinataire. Pour cela, il est primordial que les nœud se situent les uns par rapport aux autres et soient capables de construire des liens sécurisés entre eux, et cela représente le rôle d'un protocole de routage sécurisé.

L'objectif d'un protocole de routage sécurisé est de trouver des chemins sécurisé qui mènent vers la destination et qui optimisent la métrique de consommation d'énergie. Plusieurs stratégies de routage sécurisé ont été proposées pour les réseaux de capteurs sans fil. Ces protocoles peuvent être classifiés suivant la structure du réseau en trois catégories : protocoles à plat (Flat based routing), protocoles hiérarchiques (Hierarchic based routing / Clusturing based routing) et protocoles basés sur la localisation géographique *Location based routing*.

- ***Routage à plat*** : dans ce type de protocoles, tous les nœud ont le même rôle, c.à.d, ils sont homogènes et ils communiquent entre eux sans aucun autre intermédiaire. seul, la station de base est chargé de la collecte des données issues des différents capteurs. Les topologies plates sont caractérisées par la simplicité des algorithmes utilisés pour le routage des données et la scalabilité du réseau, du fait que les nœuds ont besoin uniquement des informations sur leurs voisins directs afin de participer à la tâche de routage.

- ***Routage hiérarchique*** : dans cette architecture, le réseau est partitionné en sous ensembles appelés clusters, afin de faciliter la gestion du réseau ainsi que le routage de données. Dans ce type de protocoles, chaque cluster est constitué d'un nœud simple et d'un nœud leader (cfef de cluster). Seul le chef de cluster communique avec les autres capteurs ou avec la station de base. Tous les capteurs d'un cluster envoient les données à leur cluster-head. Ensuite, ce dernier s'en charge de les acheminer vers d'autres chef de cluster ou bien vers la station de base. Le choix du chef de cluster se fait soit à tour de rôle, soit selon le nombre de voisins en considérant comme chef de cluster le nœud ayant le maximum de voisins, soit selon le niveau d'énergie du nœud. Les nœuds à énergie élevée (chef de cluter) sont utilisés pour traiter, agréger et acheminer les informations vers la destination ; alors que les membres du cluster se chargent de la tâche de capture. Avec cette méthode on assure une minimisation efficace d'énergie [4].

- ***Routage basé sur la localisation*** : dans ce type de routage, les nœuds sont adressés à l'aide de leurs positions. Les informations sur la position de chaque nœud dans le réseau est nécessaire

pour estimer l'énergie consommée lors des déférentes émissions. Comme les nœuds capteurs sont déployés dans une région d'une manière aléatoire, les informations de localisation de ces dérniers peuvent être utilisées dans le routage des données d'une manière efficace en termes d'énergie dans le but de maximiser la durée de vie du réseau.

1.4.2 Critères d'évaluation des protocoles de routage sécurisé

Les problèmes liés à la gestion et la distribution de clés ont été largement étudiés dans le cadre des réseaux capteurs sans fil. Cependant ces problèmes sont liés aux contraintes imposées par la nature du réseau : limitation de ressources, d'énergie et de puissance de traitement,...ect. Les problèmes étudiés dans ce chace qui suit, touchent les points suivants :

- ***La capacité de stockage limitée*** : elle exige qu'un protocole de routage sécurisé soit conçu avec des besoins minimaux en mémoire. Pour cela, les protocoles doivent être optimisés en stockage, vu que les capteurs sont menis d'une mémoire de stockage limitée.

- ***La puissance de traitement limitée*** : elle n'implique que le protocole de routage sécurisé, présente des calculs peu coûteux pour minimiser la dépense d'énergie et réduire la latence.

- ***La communication sans fils*** : La portée de transmission est limitée pour raison de dépense d'énergie et de limitation de puissance en batterie. A cet effet, il nécessaire que le routage soit un modèle multi-sauts pour la diffusion des données dans un RCSF.

1.4.3 Quelque protocoles de routage sécurisés

1.4.3.1 Classification

Nous illutrons dans la figure 1.1 la classification des protocoles de routage sécurisé :

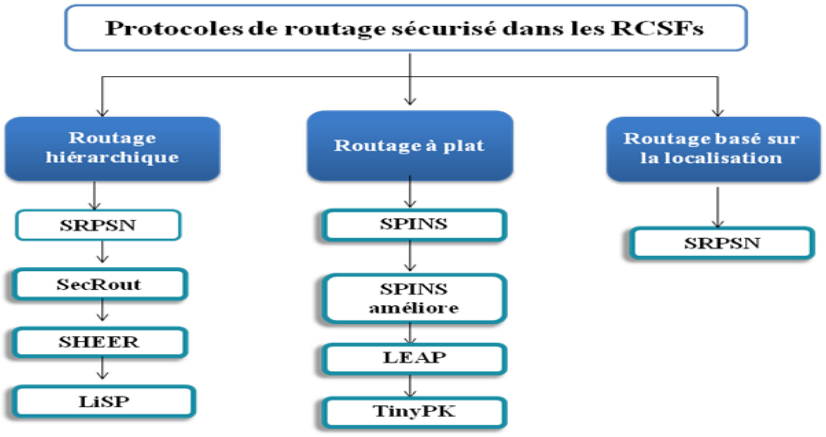

FIGURE 1.1 – Les protocoles de routage dans les réseaux de capteurs sans fil

1.4.3.2 Le protocole SRPSN *(Secured RoutingProtocol for Sensor Network)*

Tubaishat et al. ont proposé le protocole SRPSN (Secured Routing Protocol for Sensor Network)[15] est un protocole de routage hiérarchique pour les RCSFs et qui construit des routes sécurisées. Il se base sur le clustering dans lequel les chefs de cluster sont élus et ces derniers collectent des données à partir de ses membres et agrègent les données rassemblées par des procédures de fusion, puis ils les transmettent directement à la station de base. Quand un capteur devient Chef de cluster, ce dernier active son GPS afin de déterminer sa position exacte et de diffuser son identifiant (ID), ainsi que son niveau pour décider sur ses fils les mieux placés pour transmettre les paquets vers la station de base ou vers d'autres destinations.Le protocole SRPSN possède deux mécanismes de sécurité :

- Le mécanisme de découverte de routes sécurisées comporte trois phases :

– **Demande de routes sécurisées :** un nœud source émet une demande de route par l'envoie du message $RREQ$ à ses voisins.
– **Réponse de routes sécurisées :** dès la réception de la demande $RREQ$, le nœud récupère et vérifie le message, afin d'initier cette phase par la construction et par la diffusion d'une

réponse $RREP$.

– **Maintenance de routes sécurisées :** si un nœud source reçoit un message d'erreur après l'envoie d'un paquet, il commence l'entretien de la route en initiant le processus de découverte.

• Le mécanisme de transfert de données sécurisées comporte deux phases :

– **Communication-intra cluster :** lors de l'envoie de données au chef de cluster , le nœud capteur construit le paquet en utilisant l'ID du CC et la clé de cluster. Les nœuds recevant le paquet, ils vérifient l'ID contenue dans le paquet, s'il correspond à son ID, alors ce nœud vérifie l'authentification et l'intégrité du paquet par l'intermédiaire du MAC [25], Sinon, il abandonne le paquet.

– **Communication inter clusters :** le chef de cluster source vérifie dans sa table de routage s'il existe une route vers le noeud récepteur, autre temps il construit le paquet de données en utilisant son ID, sa clé, l'ID du noeud destinataire et deux champs réservés pour les noeuds intermédiaires (l'ID du nœud actuel et l'ID du nœud suivant), et un nombre aléatoire. Ce paquet est chiffré et envoyé avec un code MAC qui est généré par un algorithme de hachage [25]. Lors de la réception du paquet par le noeud intermédiaire, il vérifie l'ID du noeud destinataire. Si ce dernier correspond à l'ID qu'il détient, il met à jour l'ID de noeud suivant du paquet à diffuser, sinon, il ignore le paquet. Lorsque le noeud récepteur reçoit le paquet, il vérifie l'ID afin d'obtenir la clé du noeud source dans sa table. Ensuite, il vérifie l'authentification et l'intégrité du paquet par l'utilisation du MAC. Si les deux propriétés sont garanties, alors le récepteur accepte le paquet.

Discussion

Le protocole SRPSN est le premier modèle de routage hiérarchique conçu pour répondre au routage sécurisé dont la création des RCSFs. Il vérifie le critère de stockage dont il se base sur le clustering. Dans le cas où le nombre de noeud dans un cluster est grand, le coût de calcul des clés de groupes est coûteux, et devient une dépense d'énergie majeure pour les capteurs limités. Malgré le changement de taille du réseau, il suit un modèle de routage multi-sauts dont le Cef de cluster choisira l'un de ses fils pour l'acheminement des paquets à la station de base en vérifiant le critère de communication.

1.4.3.3 Le Protocole SecRout *(Secure Routing Protocol for Sensor Networks*

Yin et Madria [11] ont proposé un protocole de routage hiérarchique sécurisé nommé SecRout. Le réseau est organisé en clusters ayant chacun un chef. Le nœud collecteur est supposé connaître cette organisation du réseau, et doit maintenir localement une table contenant une clé secrète de chaque capteur. Cette clé est supposée pré-chargée dans chaque nœud. De plus, chaque cluster doit posséder une clé permettant de sécuriser les échanges intra-clusters. Cette clé doit être connue par tous ses membres. Le protocole SecRoute ne spécifie pas l'algorithme de construction de clusters, et suppose que les clusters ainsi que leurs clés sont établis par un autre protocole.

Le protocole SecRout possède deux mécanismes de sécurité : le mécanisme de découverte de routes et le mécanisme de transfert de données. Donc ce protocole a les mêmes mécanismes que le protocole SRPSN, dans lequel il ne fournit pas une découverte de voisins sécurisés et il ne spécifie pas la construction des clusters.

Discussion

Le protocole SecRout a les mêmes mécanismes que le protocole SRPSN décrit précédemment, d'où ce protocole minimise l'utilisation d'espace mémoire donc le critère de stockage est assuré, et lors de l'ajout de nouveaux capteurs, il suit un modèle de routage multi-sauts, par contre il est très couteux en terme de calculs.

1.4.3.4 Le Protocole SHEER *(Secure Hierarchical Energy-Efficient Routing)*

Ibriq et Mahgoub ont proposé un protocole de routage sécurisé SHEER (Secure Hierarchical Energy-Efficient Routing) [12]. C'est un protocole hiérarchique qui garantie une efficacité énergétique au sein du réseau. Il utilise HIKES (Hierarchical Key Establishment System) qu'il est un système hiérarchique permettant l'établissement et la distribution de clés, ainsi que l'authentification des nœuds capteurs. Il met en place un mécanisme de transmission probabiliste pour atteindre une bonne efficacité énergétique au sein du réseau. Chaque capteur est pré chargé par trois primitives de chiffrement :

1. Une clé séquestre partielle (PKET) comprenant 16 entrées, chaque entrée déverrouille 2^{12} clés, pour un total de 2^{20} ID utilisés pour l'identification des nœuds du réseau d'une manière unique.
2. Une valeur unique est utilisée pour chiffrer utilisé par la station de base pour l'authentification de la diffusion.

3. Un ensemble de clés de chiffrement :

La clé de nœud : est une clé spécifique générée à partir de la clé principale.
La clé de session : est une clé privée d'un nœuds capteur, elle est utilisée lors de la communication avec ses voisins.
La clé primaire : est une clé unique possédée par chaque capteur, et elle est utilisée lors de la communication avec la station de base.
La clé de cluster : est une clé utilisée entre le capteur et son CC.
La clé du groupe : est une clé partagée par tous les membres du même cluster pour une communication passive au sein du groupe.
La clé principale : est une clé utilisée pour générer des clés des capteurs.
La clé de CC : est une clé unique utilisée par le nœud capteur afin de communiquer avec le Sauvegarde du CC.

Le protocole proposé passe par quatre phases :

La phase d'initiation : après le déploiement des capteurs, la station de base envoie un appel sécurisé d'initiation pour les capteurs. Cet appel utilise le mécanisme de diffusion d'authentification HIKES.

La phase de découverte des voisins : dans cette phase, un capteur diffuse un message HELLO contenant son ID, un nonce, et un en-tête chiffré avec une clé spécifique au capteur, puis il attend un certain temps la réponse de ses voisins.

La phase de groupement : la sélection d'un CC se base sur la taille de cluster ou de sa densité. La taille de cluster est un paramètre spécifique à une application et elle est déterminée avant le déploiement des capteurs. Si l'application nécessite une densité p, la probabilité qu'un capteur devient un CC est 1/p. Pour qu'un nœud devient un CC de premier niveau, le capteur génère un nombre aléatoire P entre 0 et 1. Si $P < 1/p$, alors ce capteur est un CC de premier niveau. Et dans le cas d'un CC de second niveau, le CC du premier niveau répète le même processus, s'il entend la SB, il se confirme comme un CC de deuxième niveau. Sinon, il choisit au hasard un de ses voisins pour le remplacer et devenir un CC de deuxième niveau.

La phase de regroupement : lorsque le niveau d'énergie d'un CC atteint un certain seuil, le CC est remplacé par l'un de ses membres. en sélectionnant le voisin direct qui possède plus d'énergie, puis il informe la station de base du nouveau CC après la dernière mise à jour. La station de base met à jour sa base de données, puis elle envoie une authentification au nouveau CC, en lui fournissant des indices et des décalages des clés des membres du cluster. Le nouveau CC informe tous les membres de cluster.

La phase d'échange de messages de données : chaque capteur transmet ses données chiffrées avec la clé de cluster et un en-tête chiffré avec sa clé de session. Ces données sont transmises au CC, où il 'agrégé ces données, puis les envoie à la station de base.

Discussion

SHEER est un protocole de routage hiérarchique utilisant le système HIKES pour la distribution et l'authentification de clés et en fournissant une efficacité en termes d'énergie, dont il minimise cette dernière, et il vérifie le critère de calcul. Ce protocole n'assure pas le critère de stockage puisque chaque capteur est pré-chargé avec les trois primitives de chiffrement et comme la capacité de stockage est limitée dans un capteur enregistré, si le nombre de nœuds dans un cluster est grand, alors ce dernier ne pourra pas les enregistrer. SHEER suit un modèle de routage multi-sauts car le CC choisira l'un de ses fils pour transmettre les paquets à la station de base.

1.4.3.5 Le protocole SPINS *(Sensor Protocols for Information via Negotiation)*

L'une des premières solutions de sécurité pour les RCSFs est appelée SPINS (Sensor Protocols for Information via Negotiation) [19]. Il se base sur un modèle de négociation et sur deux protocoles de sécurité : SNEP (Secure Network Encryption Protocol) et μTESLA (the micro version of the Timed Efficient Stream Loss-Tolerant Authentication Protocol) [20]. Le protocole SNEP utilise deux mécanismes de sécurité. Le premier consiste à chiffrer les données pour assurer leur confidentialité et le second de calculer un code MAC (Message Authentication Code) afin d'assurer l'authentification et l'intégrité des données entre les entités. A chaque premier échange de données entre deux nœuds avec le protocole SNEP, le nœud émetteur précède le message d'une chaine de bits aléatoires, aussi appelé vecteur initial. . Cette technique empêche un intrus d'entrer en écoute dans le réseau et en possédant le message chiffré précédemment, on déduit que le même message a été envoyé car le nouveau message chiffré sera totalement différent du message chiffré antérieur par l'utilisation d'une chaine de bits aléatoire et du chiffrement par bloc. Les deux nœuds partagent un compteur qui permet l'utilisation des chiffrements par bloc et du non utilisation du vecteur initial.

Un attaquant ne peut déchiffrer l'information sauf si le même message chiffré est vu plusieurs fois. L'utilisation d'un vecteur aléatoire et d'un compteur empêchent l'écoute et l'interception du message, puisque le message est envoyé en clair suivi soit, d'une chaine de bits, ou d'un compteur incrémenté qui est différent à chaque bloc échangé. L'utilisation de ce compteur permet d'éviter des attaques de rejeu par paquet dont chaque message est numéroté et cela permettra de garantir la

fraicheur des données.

Le protocole μTESLA [20] permet l'authentification par diffusion. Cette version est adoptée aux RCSFs. Il utilise une authentification symétrique liée à une méthode asymétrique où les clés symétriques sont divulguées au cours du temps. Pour permettre cette authentification, il est nécessaire que la station de base et les différents nœuds soient vaguement synchronisés.

La station de base a pour rôle d'ajouter au paquet à envoyer le code MAC calculé à partir d'une clé secrète. Un nœud recevant ce paquet peut vérifier que la clé de déchiffrement du code MAC n'a pas encore été divulguée et ce grâce à son horloge de synchronisation. Si la clé n'est pas encore divulguée dans ce cas, il peut en déduire que seul la station de base qui a une connaissance sur la clé MAC et qu'aucun attaquant n'a pu altérer le message pendant sa transition. Pour cela, il peut stocker le paquet dans son cache en attendant la prochaine divulgation de la clé. Quand la clé sera divulguée il déchiffrera le message et vérifiera son authenticité. Chaque clé K est une clé issue d'une chaine de clés générée par une fonction à sens unique F, de telle manière que $K_i = F(K_{i+1})$ Cette clé de chiffrement utilisé pour le code MAC est générée dans un intervalle régulier de telle sorte que si un capteur ne reçoit pas tous les paquets de clés, il est capable de retrouver les anciennes à partir de la dernière clé reçue.Si un nœud possède la clé initiale K_0 et la cléK_2, mais ne reçoit pas la clé K_1, dans ce cas il vérifiera que la clé K_2 est bien celle envoyée par la SB, d'où $K_0 = F(F(K_2))$ et d'autre part il peut retrouver K_1 car $K_1 = F(K_2)$.

Discussion

SPINS ne vérifie pas les critères de stockage et de calcul car il utilise le protocole μTESLA [20], donc les nœuds sont attendus à stocker tous les paquets qui contiennent la clé de communication. L'application successive de la fonction à sens unique comme MD5 (Digest message) ou HMAC sont coûteuses en termes de calcul.Ce protocole suit un modèle de routage point à point et ce en vérifiant le critère de communication.

1.4.3.6 Le protocole SPINS Amélioré *(Sensor Protocols for Information via Negotiation)*

Jiang et al.[2] ont amélioré le protocole SPINS, cette amélioration est basée sur l'authentification de l'identité de Schnorr au lieu du protocole d'authentification uTESLA qui permet l'authentification par diffusion. Ce protocole authentifie l'identité d'un nœud du réseau avant que la station de base établie la clé de session temporaire qui représente une clé de communication. Seul les nœuds authentiés qu'ils peuvent répondre afin d'éviter le problème de divulgation de la clé principale partagée. Pour cela ils ont introduit le schéma d'authentification de l'identité pour améliorer le protocole

SPINS existant. En effet, vant la diffusion des nœuds de capteurs, la station de base signe l'identité de chaque capteur puis, elle envoie cette identité et le message signé à chaque capteur. D'où chaque capteur obtient et stocke le certificat (C_i) délivré par la station de base. Apres l'authentification de l'ensemble des nœuds du réseau par la station de base, cette dernière génère et renvoie la clé de session à chaque un de ces nœuds.

– LE MÉCANISME D'AUTHENTIFICATION SE DÉROULE SUR 9 ÉTAPES :

1. Le nœud A choisit un nombre aléatoire k_1, et il calcule y_1 le message signer, puis il envoie (y_1, C_A, B) à la station de base.
2. Lorsque la station de base reçoit le message depuis le nœud A, elle vérifie la signature du message. Dans le cas ou elle est vérifiée, la station de base sélectionne un nombre aléatoirer_1 et le communique au nœud A. Dans le cas contraire le protocole se termine.
3. A la réception du nombre aléatoire r_1, le nœud récepteur A calcule la valeur v_1 en utilisant r_1, puis il l'envoie à la station de base.
4. A la réception de v_1. La station de base vérifie la signature y_1 avec r_1 et v_1 qui sont reçues à l'étape précédente. Dans le cas où la vérification est assurée, alors A est un nœud authentifié, sinon le protocole se termine.
5. Lorsque le nœud A initie la demande de communication, la station de base envoie une notification au nœud B qui représente la demande de A.
6. A la réception de la notification, B choisit un nombre aléatoire k_2 puis il le signe en obtenant y_2, ensuite il communique le message suivant (y_2, CB) à la station de base.
7. La station de base exécute le même processus comme l'étape (2) en assurant l'authentification et la signature.
8. La station de base authentifie le nœud B, de la même manière que l'étape 3.
9. Une fois que A et B ont été authentifiés, la station de base génère et envoie la clé de session à A et B respectivement, et elle assure la vérification des nombres aléatoires envoyés par A et B, afin de garantir la fraîcheur du protocole.

Discussion

Le protocole SPINS amélioré possède les mêmes mécanismes que le protocole SPINS décrit précédemment, d'où ce protocole utilise l'authentification de l'identité de Schnorr au lieu du protocole d'authentification uTESLA, et cela pour minimiser l'espace de stockage et la puissance de calcul au niveau des capteurs, donc les deux critères sont assurés, mais le nombre de communications est augmenté. L'authentification entre deux capteurs se fait sur plusieurs échanges de messages, et cela mène a une saturation au niveau de communication.

1.4.3.7 Le protocole LEAP *(Localized Encryption and Authentication Protocol)*

Zhu et al. [17] ont proposé un protocole de gestion de clés pour les réseaux de capteurs appelé LEAP (Localized Encryption and Authentication Protocol). Il génère et met en place quatre types de clés : une clé individuelle, une clé de groupe, une clé de cluster, et une paire de clés partagée.Le mécanisme de gestion de clés fourni par ce protocole support le traitement interne (inter network processing) tout en limitant l'impact de sécurité d'un nœud compromis sur son voisinage immédiat dans le réseau. LEAP crée et met en place quatre types de clés pour chaque nœud :

- **La clé individuelle :** c'est une clé unique partagée entre chaque capteur et la station de base pour assurer la communication entre eux.
- **La clé du groupe :** c'est une clé globalement partagée par tous les capteurs dans le réseau et elle est utilisée par la station de base pour chiffrer les messages et les envoyer aux membres du groupe.
- **La clé globale :** est une clé partagée par un nœud avec tous ses voisins et elle est utilisée pour chiffrer les émissions locales.
- **La clé par paire :** chaque nœud partage une clé principale avec chacun de ses voisins immédiats.

LEAP est basé sur une clé initiale transitoire K_i qui est généré par la station de base. Cette clé est chargée dans chacun des nœuds du réseau pour dériver une clé principale. Ce protocole suppose que pour compromettre un nœud, l'adversaire nécessite un temps minimal T_{min} : c'est le temps de copier le contenu de la mémoire du nœud compromis. Il exploite ce temps pour permettre à deux nœuds voisins d'établir d'une manière sécurisée une clé de session symétrique à partir de la clé initiale K_i. Après un temps T_{min}, la clé K_i est supprimée de la mémoire du nœuds. Ce protocole passe par quatre phases :

- **Chargement de la clé initiale :** la station de base génère une clé initiale K_i et cette dernière sera chargée au niveau de chaque nœud . Chaque nœud u dérive une clé principale $K_u = f\ (K_i(u))$, f étant une fonction pseudo-aléatoire.
- **Découverte de voisins :** après le déploiement, le nœud u découvre ses voisins en diffusant un message HELLO qui contient son ID. Aussi, il initie un temps qui sera déclenché après le temps T_{min}. Le nœuds u attend un acquittement ACK de chacun de ses voisins v qui contient l'identificateur de v. L'ACK est authentifié en utilisant la clé principale K_v, qui est dérivée comme suit : $K_v = f(K_i(v))$.
- **Etablissement de la clé par-paire :** le nœud u calcule sa clé par paire K_{uv} avec le nœuds v, comme suit : $K_{uv} = f(K_v(u))$.Le nœud v peut de même calculer K_{uv}de la même manière, K_{uv} sert comme clé entre u et v.
- **suppression des clés :** Lorsque le temps expire aprèsT_{min}, le nœud u efface la clé initiale K_i

et toutes les clés principales K_v de ses voisins. Il est à noter que le nœuds u ne supprime pas sa clé principale K_u.

A la fin de ces quatre étapes, le nœuds 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 K_i. Un adversaire peut écouter clandestinement tout le trafic dans cette phase, mais sans la clé K_i il ne peut pas injecter des informations incorrectes ou déchiffrer les messages. Un nœud malveillant compromettant un nœuds après T_{min} , et il obtient seulement les clés du nœuds compromis. Quand un nœud compromis est détecté, ses voisins suppriment simplement les clés qui ont été partagée avec lui. Cependant, le message HELLO est non-authentifié, et donc un adversaire peut exploiter ceci pour lancer des attaques en injectant un grand nombre de ce type de message. Le protocole LEAP emploie μTESLA afin d'assurer l'authentification d'émission avec la station de base. Mais pour le trafic et la communication entre les nœuds, ces derniers s'authentifient avec leurs propre clé de cluster.

Discussion

Le protocole LEAP est une solution de sécurité pour RCSFs. Il n'assure pas le critère de stockage puisqu'il utilise le protocole μTESLA. A cet effet il induit les nœuds à stocker tous les paquets de clés. LEAP vérifie le critère de calcul car il n'utilise pas des calculs coûteux. LEAP suit un modèle de routage multi-sauts puisque un chef de cluster choisit l'un de ses fils afin d'acheminer les paquets.

1.4.3.8 Le protocole LiSP *(Lightweight Security Protocol)*

Le protocole de routage sécurisé LiSP (Lightweight Security Protocol) [24] a été suggéré par Park et Shin [16]. Il décompose le réseau en multi groupes de capteurs où chaque sous groupe est contrôlé par un nœud et un serveur de clés. Chaque serveur de clés contrôle la sécurité et la communication au sein de son groupe. LiSP met en œuvre le même mécanisme de génération de clés dont il est proposé par Setia et al. [23]. LiSP utilise des fonctions de hachage et de algorithmes de renouvellement périodique des clés, Il définit deux types de clés une clé temporelle TK (temporal key) qu'elle est utilisé afin de chiffrer et déchiffrer les messages, et une clé principale spécifique au capteur utilisé par le serveur de clés pour chiffrer et déchiffrer la clé TK. La clé TK est partagée par tous les capteurs dans un même groupe et elle est rafraîchie périodiquement afin d'assurer la confidentialité des données. Dans le mécanisme d'authentification de LiSP, le serveur de clés vérifie l'ajout d'un nouveau nœud capteur en utilisant une approche du tierce de confiance comme dans [21]. Il dispose également de deux composants de sécurité : un système de détection d'intrusion et un protocole de gestion de clés TK.

Discussion

LiSP vérifie le critère de stockage puisque les capteurs n'ont pas besoin de stocker les clés et les informations globales de routage. Il utilise des algorithmes cryptographiques de hachage qui sont très coûteuses en termes de calculs. LiSP suit un modèle de routage point à point et cela garantie le critère de communication.

1.4.3.9 Le protocole TinyPK *(Securing sensor networks with public key technology)*

TinyPK est un protocole de routage sécurisé proposé par Watro et al. [13]. Il assure l'authentification à l'aide d'une paire de clé publique et privée. Les opérations cryptographiques se basent sur l'algorithme de chiffrement RSA [22]. Ces opérations sont exécutées par une entité externe du réseau. Le protocole nécessite une autorité de certification possédant une paire de clé publique et privée qu'elle est accordé avec l'ensemble des nœuds du réseau.Toute partie externe qui souhaite communiquer avec les capteurs doivent avoir une clé publique que cette dernière doit être signée par l'autorité de certification afin de lui établir identifiant unique et une clé privée correspondante à la clé publique. Chaque capteur est également pré charge par d'une clé publique correspondante à l'autorité de certification. Les nœuds capteurs peuvent se communiquer les uns avec les autres par l'intermédiaire de la partie externe mais ils doivent établir leurs identités à une partie externe qui s'en charge de toutes les opérations cryptographiques.

Discussion

Le protocole TinyPK vérifie le critère du stockage puisque les capteurs n'ont pas besoin de stockes les clés et les informations globales du routage. Il nécessite une autorité de certification qu'elle est très coûteuse en termes de calcul, et cela il ne vérifie pas le critère de calcul. Le protocole proposé suit un modèle de routage point à point, donc il vérifie le critère de communication.

1.4.4 Synthèse des solutions existantes :

1.4.4.1 Comparaison

Le tableeau I.1 illustre la comparaison entre les protocoles présentés précédemment.

Protocole	La capacité de stockage limitée	La puissance de traitement limitée	Communication
SRPSN	Vérifie	N'est pas vérifié	multi-saut
SecRout	Vérifie	N'est pas vérifié	multi-saut
SPINS	N'est pas vérifié	Vérifié	multi-saut
SPINSA	N'est pas vérifié	Vérifié	multi-saut
SHEER	N'est pas vérifié	Vérifié	multi-saut
LEAP	N'est pas vérifié	Vérifié	multi-saut
LiSP	Vérifie	N'est pas vérifié	point à point
TinyPK	Vérifie	N'est pas vérifie	Point à point

TABLE 1.1 – Comparaison entre les différents protocoles de routage sécurisé dans les RCSFs

1.5 La cryptographie à base d'identité

Dans le cadre de la cryptographie à base d'identité, une grande série de calcul largement utilisée pour construire les différents schémas de chiffrements et de signatures. Dans cette partie, nous présentons le schéma de chiffrement, de signature et de leurs fonctionnement.

1.5.1 Définition

Shamir [26] est le premier à proposer le concept de la cryptographie à base d'identité. Ce concept permet à l'utilisateur de communiquer en toute sécurité et de vérifier les signatures des uns avec les autres sans échange de clés. Cependant, à la place d'un certificat numérique, on utilise l'identité d'un utilisateur afin d'assurer le chiffrement ou la vérification de signature. Par conséquent, la cryptographie à base d'identité réduit la complexité du système. Le cout de la création et la gestion des clés publiques dans le cadre d'authentification connue (PKI). Le chiffrement et la signature numérique sont basés sur les concepts suivants :

– Concepts de bases de chiffrement à base d'identité (cf. la figure 1.2) :

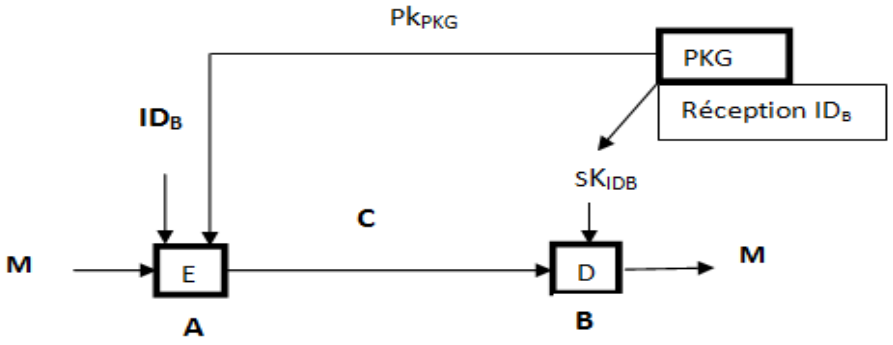

FIGURE 1.2 – Schéma de chiffrement à base d'identité

Dans le cadre du schéma *Identity-Based Encryption (IBE)*, l'entité émettrice peut utiliser l'identité d'une entité réceptrice afin de chiffrer un message, qui obtient par la suite un message chiffré C, puis ce dernier sera envoyé. L'entité réceptrice récupère la clé privée associée à son identité à partir du centre de génération de clés afin de déchiffrer le message C. Le fonctionnement est décrit comme suit :

Mise en place : Le centre PKG crée une paire de clés privée, publique, notées respectivement sK_{PKG} , pK_{PKG} . La clé publique est diffusée à toute personne , qui demeure pour une longue période comme un système de paramètres constants.

Extraction de la clé privée : Le récepteur B s'authentifie auprès du serveur PKG afin d'obtenir une clé privée sK_{ID_B} associée à son identité ID_B .

Chiffrement : L'émetteur A chiffre le message en claire M en obtenant un message chiffré C en utilisant l'identité du récepteur B ID_B ainsi la clé publique pK_{PKG} générée par le centre PKG.

Déchiffrement : Après avoir reçu le message chiffré de l'entité émettrice A, le récepteur B déchiffre le message reçu en utilisant sa clé privée sK_{ID_B} et comme résultat il récupère le message en claire M.

– Concepts de base de la signature à base d'identité (cf. la figure 1.3) :

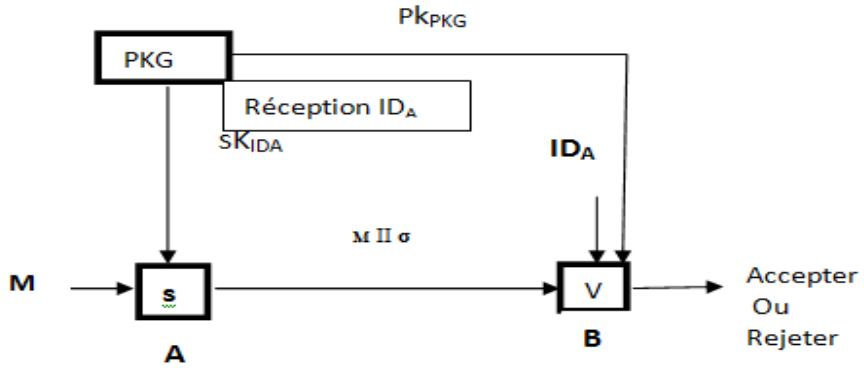

FIGURE 1.3 – Schéma de chiffrement à base d'identité

Dans le schéma *Identity-Based Signature (IBS)*, le signataire A obtient une clé de signature à partir du serveur PKG qui est une clé privée associée à son identificateur. Le nœud A signe le message en utilisant sa clé privée sK_{ID_A} , et le vérificateur B utilise la clé publique associée àl'entité A afin de vérifier sa signature. Le processus s'exécute comme suit : Les deux étapes de mise en place et l'extraction de la clé privée sont les mêmes que dans le schéma précédent. Il y a aussi deux autres étapes

La signature du message : l'émetteur A crée une signature σ du message M en utilisant sa clé privée sK_{ID_A}.

Vérification de la signature : le vérificateur B, après avoir reçu la signatur σ et le message M, il utilise l'identité de l'émetteur ID_A et la clé publique du serveur PKG pour vérifier si la signature σ est authentique à celle du message M.

1.6 Conclusion

Ce chapitre nous a permis de voir les différents protocoles de routage sécurisé dans les RCSFs, et nous avons dressé un état de l'art des principales approches proposées. Nous avons donné une vue d'ensemble sur les objectifs concernant le routage sécurisé en les classifiant selon plusieurs critères : la capacité de stockage limitée des capteurs, la puissance de traitement limitée, la scalabilité et le modèle de communication. En autre pour assurer la fiabilité du système, nouveaux mécanisme et mesure de sécurité doivent être mis en place afin d'éliminer les vulnérabilités dans RCSFs qui sont sujets à de nombreuses menaces et attaques.

Nous nous intéressons par la suite à la conception d'un nouveau protocole de routage sécurisé qui assure l'authentification de bout en bout entre les nœuds communiquant en prenant en considération les contraintes des RCSFs tel qu' il est économe en terme de consommation d'énergie et qui a pour but d'augmenter la duré de vie du réseau.

Chapitre 2

IBC - based Hop by Hop Authentication Protocol for Wireless Sensor Networks

2.1 Introduction

Un réseau de capteurs sans fil a beaucoup de contraintes importantes parmi elles la consommation d'énergie, la puissance de calcul et les ressources limitées. De ce fait, l'épuisement énergétique d'un certain nombre de nœuds entraîne un changement sur la topologie globale du réseau. L'efficacité en consommation d'énergie représente une métrique de performance significative qui influence directement sur la durée de vie du réseau. Dans ce chapitre nous proposons un protocole de routage sécurisé pour les réseaux de capteurs sans fils nommé ***IBC2HAP*** *(*IBC based Hop by Hop Auth entication Protocol*) basé sur la cryptographie à base d'identité (CBID). L'objectif principal est d'assurer l'authentification de bout-en-bout afin d'isoler les nœuds malicieux qui peuvent injecter ou supprimer les données qui circulent sur le réseau. L'injection de données augmente inutilement la consommation énergétique des capteurs, en revanche la suppression de données engendre la retransmission ce qui induit aussi sur la consommation d'énergie.*

2.2 Signature numérique avec CBID à base RSA

Les schémas basés sur la cryptographie d'identité parviennent à résoudre les problèmes liés à l'authentification et en particulièrement à la gestion et à la révocation des clés publiques. Il nécessite la présence d'un centre PKG (*Private Key Generator*) qui fournit à chaque nœud la clé privée correspondante à sa clé publique, dont lequel le nœud est capable d'effectuer des signatures ou des

déchiffrements sans solliciter le PKG.

Pour la création de la paire de clés PKG, le centre choisit deux nombres premiers p et q et calcule n et $\phi(n)$ tels que $n = p \times q$ et $\phi(n) = (p-1) \times (q-1)$. Ensuite, il choisit $e < \phi(n)$ de telle sorte que $pgcd(e, \phi(n)) = 1$, et enfin il calcule d tel que $e \times d \; mod \; \phi(n) = 1$. La clé publique est (e, n) et la clé privée correspondance est (d, n). Et pour générer la clé privée associe à l'identité d'un nœud d_{IDi} , le centre PKG calcule $d_{IDi} = ID_i^d \; mod \; n$. Donc chaque nœud a une paire de clés $(ID_i, d_{IDi}))$. Le centre PKG génère une fonction de hachage H, tel que $H(t, M) = t \times M \; mod \; 100$.

Exemple

- $p = 19$, $q = 23$, $n = 19 \times 23 = 437$ et $\phi(n) = (19 - 1) \times (23 - 1) = 396$.
- $e = 17 \Rightarrow 17d \; mod \; 396 = 1 \Rightarrow d = 233$.
- La clé publique PKG $(e, n) = (17, 437)$.
- La clé privée PKG $(d, n) = (233, 437)$.
- Pour une clé publique d'un nœud $ID_i = 15$.
- sa clé privée $d_{ID}i = 15^{233} \; mod \; 437 = 166$.

Pour calculer la signature numérique d'un message M, on choisit un nombre aléatoire r et on calcule $t = r^e \; mod \; n$. Ensuite, on signe le message en utilisant la clé privée, on obtient un message signé $S = d_{IDi} \times r^{H(t,M)} \; mod \; n$. Après la réception du message signé, on vérifie la validité de la signature (M ,t,S) et cela par l'évaluation de légalité $S^e \; mod \; n = ID_i \times t^{H(t,M)} \; mod \; n$.

Exemple

- $M = 75$, $r = 27$, et $t = 27^{17} \; mod \; 437 = 278$.
- $H(278, 75) = 278 \times 75 \; mod \; 100 = 50$.
- $S = 166 \times 27^{5O} \; mod \; 437 = 79$.
- $(M, t, S) = (75, 278, 79)$.
- Vérification : $79^{17} \; mod \; 437 = 15 \times 278^{50} \; mod \; 437 = 431$.

2.3 Motivations

La motivation principale de notre protocole est le fait que la station de base (**SB**) possède une grande capacité de stockage et de calcul, d'où l'idée d'exploiter cette propriété permet de concevoir notre protocole. Au niveau de la station de base est installé le centre PKG en qui tous les capteurs font confiance, ce qui permet de gérer les clés publiques sans avoir recours à l'usage des certificats qui sont très couteux en termes de calcul et de stockage. Afin de minimiser la consommation énergétique et cela par la diminution des communications, le réseau est organisé en clusters. Le clus-

tering consiste à diviser le réseau en groupe virtuel, chacun de ces derniers comporte un ensemble de capteurs physiquement adjacents supervisés par un chef de cluster (**CC**). Ce dernier responsable de communication entre les capteurs et la station de base (SB).

La plupart des protocoles de routage sécurisé conçus pour les réseaux de capteurs sans fil utilisent des solutions cryptographiques, puisqu'elles sont réputées comme des solutions sûres et qui répondent à l'ensemble de problèmes liés à la protection de données. Ces différentes solutions se basent soit sur la cryptographie symétrique ou asymétrique. Les problèmes liés à la cryptographie symétrique sont la distribution et le stockage de clés de chiffrement, par contre la cryptographie asymétrique est réputée par sa robustesse comparant à la cryptographie symétrique, et cela le fait d'utiliser les certificats comme un mécanisme d'authentification, qui est un document électronique qui garantit l'authenticité d'une manière unique et sure un capteur, mais la gestion de certificats est un problème majeur. Pour remédier à ce problème nous présentons un mécanisme d'authentification basé sur la cryptographie à base d'identité. Ce qui va nous permettre d'écarter la solution de gestion des certificats. Ceci est dû à la clé publique de chaque capteur est fortement liée à son identité. Avec cette façon, notre protocole gagne en termes de performance où les contraintes de stockage et de calcul dont la vérification des certificats ne sont pas considérées.

Dans ce qui suit, nous présentons notre protocole don son objectif principal est l'authentification de bout-en-bout des capteurs intermédiaires relayant les données jusqu' à la station de base.

2.4 Hypothèses et contexte

Nous considérons un réseau de capteurs composé de n nœuds statiques et homogènes et leurs positions sont précises et uniques. Ainsi que tous les capteurs collectent des informations et les communiquent à une unique station de base dont sont énergie est illimitée.

2.5 IBC2HAP : IBC-based Hop by Hop Authentication Protocol

Dans cette section, nous présentons notre protocole **IBC2HAP** qui est une solution centralisée pour sécuriser le processus de transfert de données vers la station de base, et cela en assurant le service d'authentification de bout-en-bout dans un réseau exposé aux attaques. Il est appliqué sur une architecture hiérarchique, et qu'il exploite les mécanismes de cryptographie à base d'identité afin d'économiser l'énergie, minimiser l'espace de stockage et le calcul aux niveaux des capteurs. Lorsqu'un capteur u collecte des informations puis les communique à la station de base, il les transmet d'abord au chef de cluster (CC) qui est à son tour les transmet à la SB à travers des nœuds

intermédiaires $v1$,$v2$,...,vn. Avant qu'un capteur u reçoit ou transmet l'information à l'un de ses voisins directs v, il doit s'authentifie d'abord auprès de v, pour s'assurer de son identité déclarée précédemment, et cela pour empêcher le nœud u de nier l'envoi ou la réception de l'information.

Le principe de ce protocole se résume en deux phases principales : la phase d'installation et la phase de transmission.

La phase d'installation

L'objectif principal de cette phase consiste à la distribution de clés par le centre PKG , et cela ce fait avant le déploiement du réseau en effectuant une pré-distribution de clés. Puis partitionner le réseau en groupe virtuel appelé clusters.

La phase de transmission

Cette phase est exécutée à chaque fois qu'un capteur veut communiquer des informations, son objectif est de passer l'information à la station de base en toute confiance et cela en évitant la perte et/ou l'injection de données par des nœuds malicieux, c.-à-d lorsque un capteur veut envoyer des données à la SB, il passe par deux étapes : (1) le capteur envoie directement sur un saut l'information capturée à son chef de cluster, puis (2) ce dernier envoie et communique cette information à la SB à travers des nœuds intermédiaires tout en assurant l'authentification de bout-en-bout.

▷ LES VARIABLES QUI CARACTÉRISENT UN CAPTEUR N_i SONT :

- N_i **:** est l'identité ou la clé publique d'un nœud dans le réseau.
- S_{N_i} **:** est la clé prive d'un nœud dans le réseau.
- $suivant_i$ **:** c'est l'identité d'un nœud à qui il va transmettre le message pour atteindre la station de base suivant sa table de routage.
- $precedent_i$: est l'identité de nœud où il a reçu le message pour l'acheminer à la SB.

▷ LES FONCTIONS UTILISÉES PAR UN CAPTEUR N_i SONT :

- **réception** (M, CC_i) : le chef de cluster CC_i reçoit l'information (M) depuis l'un de ses

membres.

- **envoyer** $(REQ, suivant_i)$: le nœud N_i envoie à son voisin direct à l'identité $suivant_i$ un message de type REQ.

- **réception** (REQ, N_i) : le nœud N_i reçoit le message de type REQ depuis le nœud $prcdent_i$.

- **enregistrer** $(precedent_i, M)$: le nœud N_i enregistre l'information (M) et l'identité du nœud précédent.

- **récupérer** (M) : le nœud N_i extraire l'information (M) du message REQ.

- **envoyer** $(repREQ, prcdent_i)$: le nœud N_i envoie à son voisin direct $prcdent_i$ un message de type $repREQ$.

- **réception** $(repREQ, N_i)$: le nœud N_i reçoit le message de type $repREQ$ depuis le nœud $suivant_i$.

- **récupérer** (S) : le nœud N_i extraire le message signé (S) du message $repREQ$ ou $repREQ1$.

- **non-reçu** $(repREQ, N_i)$: le nœud N_i ne reçoit pas le message de type $repREQ$ depuis le nœud $suivant_i$.

- **enregistrer** $(suivant_i, repREQ)$: le nœud N_i enregistre le message $repREQ$, l'identité du nœud suivant.

- **ignorer** $(repREQ)$: le nœud N_i ignore le message de type $repREQ$ qu'il a reçu.

- **envoyer** $(repREQ1, suivant_i)$: le nœud N_i envoie à son voisin direct $suivant_i$ un message de type $repREQ1$.

- **reception** $(repREQ1, N_i)$: le nœud N_i reçoit le message de type $repREQ1$ depuis le nœud $prcdent_i$.

- **non-reçu** $(repREQ1, N_i)$: le nœud N_i ne reçoit pas le message de type $repREQ1$ depuis le nœud $precedent_i$.

- **ignorer** $(repREQ1)$: le nœud N_i ignore le message de type $repREQ1$ qu'il a reçu.

- **supprimer** $(precedent_i, M)$: le nœud N_i supprime l'identité du nœud précédent et l'infor-

mation (M).

- **enregistrer** $(precedent_i, repREQ1)$: le nœud N_i enregistre le message $repREQ1$, l'identité du nœud $precedent_i$.

- **envoyer** $(ACK, precedent_i)$: le nœud N_i envoie à son voisin direct $prcdent_i$ un message de type ACK.

- **reception** (ACK, N_i) : le nœud N_i reçoit le message de type $repREQ1$ depuis le nœud $suivant_i$.

- **envoyer** $(repACK, suivant_i)$: le nœud N_i envoie à son voisin direct $suivant_i$ un message de type $repACK$.

- **non-reçu** $(repACK, N_i)$: le nœud N_i ne reçoit pas le message de type $repACK$ depuis le nœud $precedent_i$.

- **supprimer** $(precedent_i, repREQ1)$: le nœud N_i supprime le message $repREQ1$ et l'identité du nœud $prcdent_i$.

- **supprimer** $(precedent_i, M)$: le nœud N_i supprime l'identité du nœud $precedent_i$ et l'information (M).

- **supprimer** $(suivant_i, repREQ)$: le nœud N_i supprime le message $repREQ$ et l'identité du nœud $suivant_i$ et l'information (M).

- **ignorer** (ACK) : le nœud N_i ignore le message de type ACK qu'il a reçu.

2.6 Description détaillée du protocole

2.6.1 Phase d'installation

Cette phase est divisée en deux phases, la première consiste à la pré-distribution de clés par le centre PKG à tous les capteurs et d'établir la clé privée S_{ID} à l'ensemble du réseau par le centre PKG, puis la deuxième phases consiste à la formation des clusters.

2.6.1.1 Phase de pré-distribution de clés

Chaque capteur possède une paire de clés publique, privée, la seule méthode pratique pour la distribution de clés au niveau d'un RCSF dans le cas ou la topologie est inconnue, et cela avant le déploiement qui devra compter sur la pré-distribution de clés. Le centre PKG ou SB possède une paire $(pK_{PKG}$, $SK_PKG)$. Il attribue à chaque nœud du réseau sa clé P_{PKG}, un identifiant unique qui représente la clé publique (P_{Ni}) du capteur, et une clé privée (S_{Ni}) correspond à la clé publique P_{Ni}.(cf.figure 2.1)

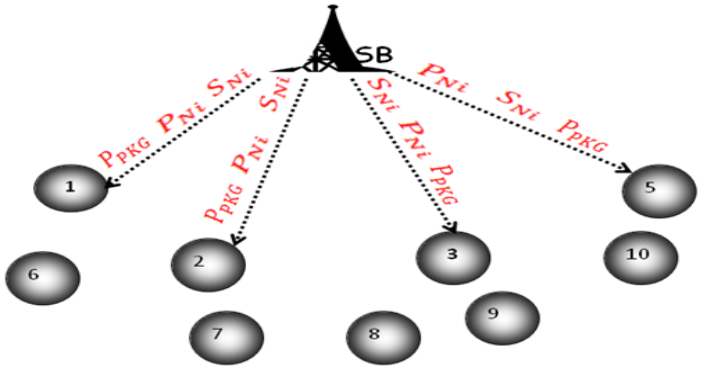

FIGURE 2.1 – Pré-distribution de clés.

2.6.1.2 Phase de formation des clusters

Dans cette étape nous utilisons le protocole CLENER (*CLuster-based approach for ENERgy-efficiency*)[1], qui est un protocole de formation de clusters pour l'efficacité énergétique, il comporte deux phases :

La première phase : consiste à l'élection des chefs de clusters périodiquement, puis partitionner les nœuds capteurs en groupes. Les chefs de cluster sont utilisés pour des tâches plus complexes, telles que : la gestion de groupe, la collecte et l'agrégation de données, et enfin l'envoi de données collectées à la station de base. Il est important d'utiliser plusieurs paramètres afin de sélectionner un CC et cela pour le but de fournir un modèle d'équilibrage de charge à haute efficacité énergétique.

Pour la sélection des CCs, la station de base diffuse un message d'initialisation, qui permet aux nœuds de calculer la distance qui les sépare de la SB, cette distance est calculée au moyen de la force du signal reçu (RSSI) [6]. Suite à cela, les nœuds sont capables d'ajuster la puissance d'émission en fonction de la distance, ce qui réduit la consommation d'énergie. Avec ces paramètres le capteur décide de devenir un CC pendant une période et ça en diffusant *un message-CC* qui contient la

valeur de l'énergie restante. Puis chaque CC se met en attente des messages rejoindre envoyer par des nœuds membres.

La deuxième phase : consiste à divisé le réseau en clusters d'où chaque nœud doit appartenir à un clusters. Pour former ces groupes, chaque nœud sélectionne le meilleur CC par rapport à son énergie résiduelle et à la distance qui les séparent. Ensuite, les nœuds calculent une valeur probabiliste de chaque candidat chef de cluster, en choisissant le CC qui a la valeur de probabilité la plus élevée et cela en envoyant un message de jointure au CC (cf.figure 2.2).

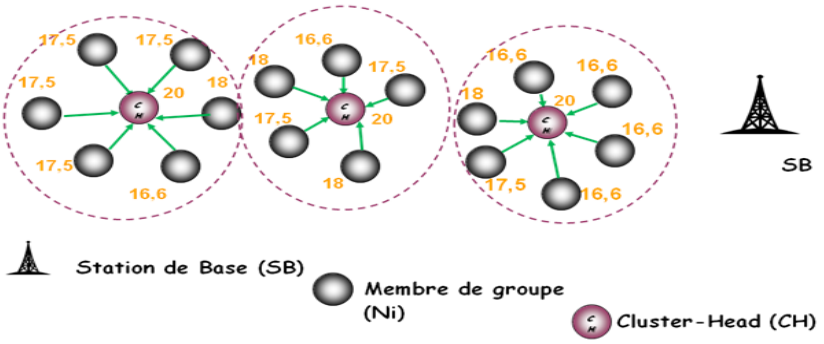

FIGURE 2.2 – Clustering d'un réseau de capteurs sans fil.

2.6.2 Phase de transmission

Cette phase est exécutée à chaque fois qu'un capteur veut communiquer des informations à la SB. Elle comporte deux étapes, la première étape est la communication intra clusters qui est une communication entre le chef de clusters CC et ces nœuds membres, qui se fait en un seul saut . Dans notre cas la communication est directe en un seul saut, les paquets de données sont envoyés directement au CC en toute confiance. on suppose que les nœuds membres sont capables d'atteindre le CC en utilisant une transmission assez puissante. Et la communication inter cluster représente une communication entre le CC et la station de base à travers des nœuds intermédiaires. Ces nœuds peuvent être des CCs ou bien des nœuds membres des clusters.

2.6.2.1 Communication intra cluster

Lorsque un nœud N_i collecte des informations, il les transmet directement à son chef de clusters.comme illustre la figure suivante :

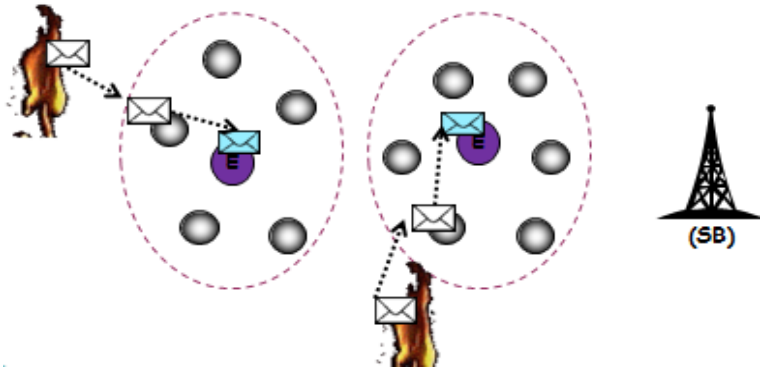

FIGURE 2.3 – Communication intra cluster

2.6.2.2 Communication inter cluster

Solution1 : sans accusé de réception (détective)

Lorsqu'un chef de cluster CC_i reçoit l'information (M) dont le but de la transmettre à la station de base, il construit un paquet REQ, puis il consulte sa table de routage pour déterminer le voisin N_j le mieux placé pour acheminer le paquet. Ensuite, il envoie ce paquet à son voisin et il se met en attente d'une réponse $repREQ$ pendant une période t, S'il ne reçoit pas le message $repREQ$, il se met à la recherche d'un autre voisin N_k dans sa table de routage et de répéter le même processus. A l'arrivée du paquet REQ à un nœud N_j depuis le nœud voisin N_i, il enregistre l'identité N_i et l'information (M), puis il construit un paquet $repREQ$ signé avec sa clé privée S_{Nj} afin de le renvoyer au nœud N_i, puis il se met en attente pendent une période t pour recevoir le paquet $repREQ1$. S'il ne le reçoit pas après cette durée, le nœud N_j supprime le paquet REQ. Dans le cas où le paquet $repREQ$ est reçu par le nœud N_i , il vérifie la signature du paquet en utilisant la clé publique P_{Nj} du nœud N_j, si la vérification est positive alors le nœud N_i enregistre le paquet reçu $repREQ$ et il construit un paquet $repREQ1$ signé avec sa clé privée S_{Ni}, puis le communiquer au nœud N_j.dans le cas ou la vérification de la signature n'est pas valide, le nœud cherche un autre voisin N_k dans sa table de routage et répéte le processus d'envoi, en communiquant le paquet REQ. A l'arrivée du paquet $repREQ1$ au nœud voisin N_j depuis le capteur N_i,N_j vérifie la signature du paquet en utilisant la clé publique P_{Ni} du nœud N_i, dans le cas où elle est vérifiée, il mémorise le paquet reçu $repREQ1$, ensuite il répète le processus en effectuant une recherche d'un nœud suivant et cela par la construction du paquet REQ. Si la signature n'est pas vérifiée alors il ignore le paquet REQ reçu au paravent.

Algorithm 1: Communication inter cluster sans un accusé de réception

DEBUT

(1) :**répéter**

(2) :**pour** *chaque nœd* CC_i *recevant* (M, CC_i) **faire**

(3) : $REQ = (N_i, M)$;

(4) : consulter sa table de routage pour choisir N_j ;

(5) : $suivant_i = N_j$;

(6) : **envoyer** $(REQ, suivant_i)$ vers le nœud $suivant_i$

(7) : **tant que** *(non-reçu* $(repREQ1, precedent_i)$ *et* $t \neq 0$*)* **faire**

(8) : $t = t - 1$;

fin Tant que

(9) : **si** *(*$t = 0$*)* **alors**

(10) : consulter la table de routage pour voir N_k tq $N_k \neq suivant_i$;

(11) : $suivant_i = N_k$;

(12) : **return** 6 ;

fin si

fin pour

(13) :**pour** *chaque nœud* N_i *recevant* (REQ, N_i) *depuis* N_j **faire**

(14) :$prcdent_i = N_j$;

(15) : **récupérer** (M) ;

(16) : **enregistrer** $(precedent_i, M)$;

(17) : req = (N_i, M) ;

(18) : $S = S_{N_i} \in r^{H(t,req)} \ mod \ n$;

(19) : $repREQ = S$;

(20) :**envoyer** $(repREQ, precedent_i)$ vers le nœud $precedent_i$;

fin pour

(21) :**pour** *chaque nœud* N_i *recevant* $(repREQ, N_i)$ *depuis* $suivant_j$ **faire**

(22) : **récupérer** (S) ;

(23) :$S_1 = S_e \ mod \ n$;

(24) :$S_2 = N_j \times t^{H(t,M)} \ mod \ n$;

(25) :**si** *(*$S_1 = S_2$*)* **alors**

(26) : **enregistrer** $(suivant_i, repREQ)$;

(27) : $S = S_{N_i} \times r^{H(t,M)} \ mod \ n$;

(28) : $repREQ1 = S$;

(29) : **envoyer** $(repREQ1, suivant_i)$ vers le nœud $suivant_i$;

(30) : **sinon**

(31) : consulter sa table de routage pour choisir N_k tq $N_k \neq N_j$;

(32) : $suivant_i = N_k$;

(33) : **return** 6 ;

finsi

fin pour

(34) : **pour** *(chaque nœud* N_i *recevant* $(repREQ1, N_i)$ *depuis* $precedent_i$*)* **faire**

(35) :**récupérer** (S) ;

(36) : $S_1 = S^e \ mod \ n$;

(37) : $S_2 = N_j \times t^{H(t,M)} \ mod \ n$;

(38) :**si** *(*$S_1 = S_2$*)* **alors**

(39) :**enregistrer** ($precedent_i$,$repREQ1$) ;

(40) : **return** 3 ;

(41) : **sinon**

(42) : **ignorer** $(repREQ1)$;

(43) : **supprimer** $(precedent_i, M)$;

fin si

fin pour

jusqu'à $(repREQ1, N_i)$ *arrive à la SB*;

FIN

,

Afin de mieux illustrer le fonctionnement de l'algorithme, nous présentons ci-dessous un exemple illustratif :

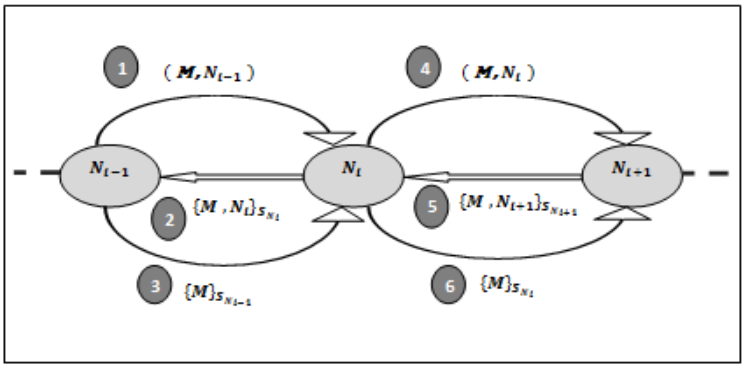

FIGURE 2.4 – Protocole d'authentification.

Solution2 : avec accusé de réception (préventive)

A la réception d'un paquet par la SB, cette dernière envoie un accusé de réception au nœud émetteur dont le but de supprimer la trace des paquets pour diminuer le coût de stockage et éviter la saturation des nœuds, et ce dernier répète le processus en accusant la réception des paquets reçus précédemment. Lorsque le paquet $repREQ1$ arrive à la SB envoyé depuis le nœœud voisin N_i, elle construit un paquet ACK signé avec sa clé privée S_{PKG}, puis elle le communique au nœud N_i et elle se met en attente pendent une période t la réception du paquet $repACK$, Si elle ne le reçoit pas alors elle retransmit le paquet ACK. A l'arrivée du paquet ACK au nœud récepteur N_j depuis le nœud voisin N_i, il vérifie la signature du paquet en utilisant la clé publique N_i, si elle est vérifiée alors il construit un paquet $repACK$ signé avec sa clé privée S_{N_j} et l'envoie au nœud N_i, puis il supprime le paquet enregistré $repREQ$ et l'identité N_i. Sinon il ignore le paquet ACK. A l'arrivée du paquet $repACK$ au nœud récepteur N_i depuis le nœud voisin N_j, il vérifie la signature du paquet en utilisant la clé publique N_j, si elle est vérifiée alors il supprime le paquet enregistre $repREQ1$, l'identité N_i et l'information (M). Dans le cas ou la signature n'est pas vérifiée alors, il ignore le paquet $repACK$.

L'utilité des messages ACK, $repACK$ est de supprimer les traces de communication dont le but de garantir l'espace de stockage des nœuds.

Algorithm 2: Communication inter cluster avec un accusé de réception

DEBUT

(44) :**répéter**

(45) : **si** *SB reçoit le paquet* $repREQ1$ *depuis* N_j **alors**

(46) : $S = S_{N_i} \in r^{H(t,M)} \ mod\, n$;

(47) : $ACK = S$;

(48) : consulter sa table de routage pour choisir N_j dont elle a reçu $repREQ1$;

(49) : $precedent_i = N_j$;

(50) : **envoyer** $(ACK, precedent_i)$ vers le nœud $precedent_i$;

(51) :**tant que** *non-reçu* $(repACK, precedent_i)$ *et* $t \neq 0$) **faire**

(52) : $t = t - 1$;

(53) : fin tant que

(54) : **si** $t = 0$ **alors**

(55) : **return** 50 ;

(56) : **finsi**

(57) :**pour** *chaque nœud* N_i *recevant* (ACK, N_i) *depuis* $suivant_i$ **faire**

(58) : **récupérer** (S) ;

(59) : $S = S_{N_i} \in r^{H(t,req)} \ mod\, n$;

(60) : $S = S_{N_j} \in r^{H(t,M)} \ mod\, n$;

(61) : **si** $S_1 = S_2$ **alors**

(62) : $S = S_{N_i} \times r^{H(t,M)} \ mod\, n$; (63) : $repACK = S$;

(64) :**envoyer** $(repACK, suivant_i)$ vers le nœud $suivant_i$;

(65) : $ACK = S$;

(66) : $N_k = precedent_i$;

(67) : **supprimer** $(suivant_i, repREQ)$;

(68) : **return** 50 ;

(69) :**sinon**

(70) : **ignorer** (ACK) ;

fin si

fin pour

pour *chaque nœud* N_i *recevant* $(repACK, N_i)$ *depuis* $suivant_i$ **faire**

(71) : **récupérer** (S) ;

(72) : $S_1 = S_{N_i} \in r^{H(t,req)} \ mod\, n$;

(73) : $S_2 = S_{N_j} \in r^{H(t,M)} \ mod\, n$;

(74) : **si** $S_1 = S_2$ **alors**

(75) : $S = S_{N_i} \times r^{H(t,M)} \ mod\, n$;

(76) : **supprimer** $(precedent_i, repREQ1)$;

(77) : **supprimer** $(precedent_i, M)$;

(78) : **sinon**

(79) : **ignorer** $(repACK)$;

fin si

fin pour

jusqu'à (ACK, CC_i) *arrive au* CC_i;

FIN

2.7 Conclusion

Dans ce chapitre, nous avons présenté un protocole d'authentification de bout-bout que nous avons appelé *IBC - based Hop by Hop Authentication Protocol for Wireless Sensor Networks*. Le but principal de ce protocole est d'assurer le service d'authentification, puisque, si ce service n'est pas assuré, cela mettra en péril le réseau. En effet, on ne pourra pas assurer une confidentialité ou une intégrité de messages échangés si dès le départ, on n'est pas sûr de communiquer avec le bon nœud. Pour atteindre ce but, nous avons choisi le mécanisme de sécurité le plus adéquat " *la cryptographie à*

base d'identité" qui prend en charge les spécificités des RCSF notamment celles liées aux ressources limitées en termes d'énergie, puissance de calcul et de mémoire. Le prochain chapitre sera consacré à la modélisation analytique et résultat de notre protocole.

Chapitre 3

Modélisation analytique et résultats

3.1 Introduction

Ce chapitre est consacré à l'analyse et à l'évaluation des performances de notre protocole. Ce chapitre est divisé en trois parties, dans la première partie on s'intéresse à l'évaluation du coût de communication et le coût de stockage des solutions proposées. Dans la seconde, nous présentons une modélisation analytique à base de chaînes de Markov, à travers laquelle nous menons une étude comparative entre les deux solutions proposées. Nous validons enfin le modèle analytique à travers des simulation.

3.2 Analyse du coût de communication et du coût de stockage

Dans cette section, nous évaluons le coût de communication et le coût de stockage pour les deux solutions.

Soit :
S : le nombre total des nœuds dans le réseau.
L : le nombre de sauts de CC_1 vers la station de base.
α **:** la taille de l'information M tel que $\alpha = \log_2(M)$ bits.
Db : le débit de transmission.

La taille de l'identité d'un nœud est $\log_2(S)$ bits et la taille d'un paquet signé est $\log_2(n)$ bits.

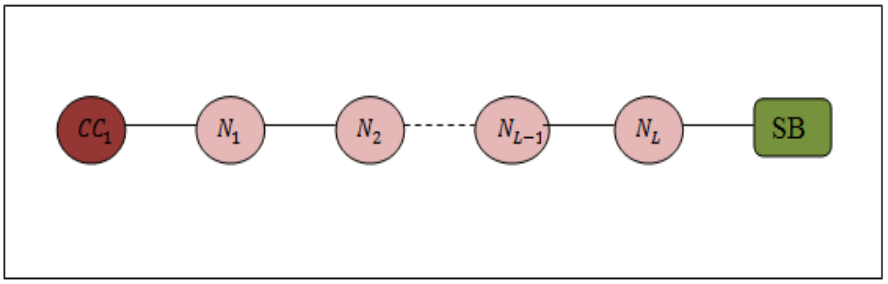

FIGURE 3.1 – La route entre la station de base et le chef de cluster CC_1

3.2.1 La première solution : sans accusé de réception (détective)

Dans cette partie, nous évaluons le cout de communication et de stockage sans prendre en compte que la station de base envoie un accusé de réception à la destination et pour cela :

- Le coût d'une communication d'un paquet entre deux capteurs noté O_1, tel que $O_1 = \alpha + \log_2(S) + 2\log_2(n)$ bits.
- La communication totale entre CC_1 et la station de base est noté O_{T_1}, tel que $O_{T_1} = L \times O_1$ bits.
- Le coût de stockage d'un nœud est $S_1 = \alpha + \log_2(S) + 2\log_2(n)$ bits.
- La durée de la communication d'un paquet est $T_1 = O_1/Db$ secondes.

- Le coût de communication d'un paquet M entre CC_i et la station de base par rapport au temps est :

- Si $0 \leq t \leq L \times T_1$ alors $O_{r_1}(t) = O_1/T1 \times t$ bits.
- Sinon $O_{T_1}(t) = O_1/T_1 \times L \times T_1 \Rightarrow O_{T_1} = L \times O_1$ bits.

- Le coût de stockage des paquets entre CC_i et la station de base par rapport au temps :

- Si $0 \leq t \leq L \times T_1$ alors $S_{T_1}(t) = S_1/T1 \times t$ bits.
- Sinon $S_{T_1}(t) = S_1/T_1 \times L \times T_1 \Rightarrow S_{T_1} = L \times S_1$ bits.

- Le coût de communication de deux paquets M_1 et M_2 entre de chef cluster CC_i et la station de base par rapport au temps tel que le deuxième paquet arrive à l'instant λ_1 :

- Si $0 \leq t \leq \lambda_1$ alors $O_{T_1}(t) = O_1/T_1 \times t$ bits.
- Si $\lambda_1 < t \leq T_1$ alors $O_{T_1}(t) = (O_1/T_1 \times t) + (O_1/T_1 \times (t - \lambda_1))$ bits.
- Si $L \times T_1 < t \leq t\ L \times T_1) + \lambda_1$ alors $O_{T_1}(t) = O_1/T_1 \times (t - \lambda_1)$ bits.

- Sinon $O_{T_1}(t) = O_{T_1}((L \times T1) + \lambda_1) = O_1/T_1 \times ((L \times T_1) + \lambda_1)$ bits.

– Le coût de stockage de deux paquets M_1 et M_2 entre le de cluster CC_i et la station de base par rapport au temps tel que le deuxième paquet arrive à l'instant λ_1 :

- Si $0 \leq t \leq \lambda_1$ alors $S_{T_1}(t) = S_1/T_1 \times t$ bits.
- Si $\lambda_1 < t \leq T_1$ alors $S_{T_1}(t) = (S_1/T_1 \times t) + (S_1/T_1 \times (t - \lambda_1))$ bits.
- Si $L \times T_1 < t \leq t\ L \times T_1) + \lambda_1$ alors $S_{T_1}(t) = S_1/T_1 \times (t - \lambda_1)$ bits.
- Sinon $S_{T_1}(t) = S_{T_1}((L \times T1) + \lambda_1) = S_1/T_1 \times ((L \times T_1) + \lambda_1)$ bits.

La figure 3.2 illustre respectivement, le cout de stockage et de communication du service d'authentification détective en fonction du temps.

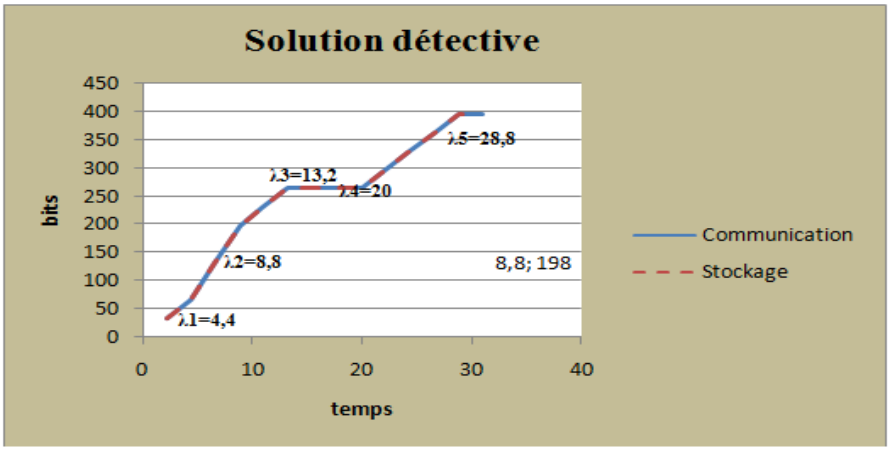

FIGURE 3.2 – Le coût de communication et de stockage d'un paquet suivant la solution détective

– λ_1 **:** temps d'envoie du deuxième message par rapport au premier message.
– λ_2 **:** temps d'arrivée du premier message à la station de base.
– λ_3 : temps d'arrivée du deuxième message à la station de base.
– λ_4 **:** temps d'envoie du troisième message par rapport au premier message.
– λ_5 **:** temps d'arrivée du troisième message à la station de base.

D'après les résultats, nous constatons que le cout de stockage dépend du cout de communication. D'où le stockage augmente avec le nombre de communications vue que les nœuds stockent les trois informations (le paquet $repREQ1$, l'identité du nœud suivant, le message M) échangées. En effet, dans cette solution, en minimisant le nombre de transmissions possibles pour économiser la quantité d'énergie consommée qui est une contrainte importante dans les RCSFs, et de ce fait l'épuisement énergétique d'un certain nombre de nœuds entraine un changement significatif sur la topologie globale du réseau, et peut nécessiter un routage de paquets différent et une réorganisation totale du réseau. Cependant, à un certain temps, il y aura des nœuds qui peuvent être saturés à cause de la limitation en termes de stockage. Pour cette raison, cette solution est applicable dans les cas ou les informations échangées ne demandent pas assez d'espace mémoire ou les informations échangées

sont de tailles très petites.

3.2.2 La deuxième solution : avec accusé de réception (préventive)

Dans cette partie, nous évaluons le coût de communication et de stockage tel que la station de base envoie un accusé de réception à la destination et pour cela :

- Le coût d'une communication d'un paquet entre deux capteurs noté O_1, tel que $O_1 = \alpha + \log_2(S) + 2\log_2(n)$ bits.
- Le coût d'une communication d'un paquet d'accusé de réception entre deux capteurs est noté O_2, tel que $O_2 = 2\log_2(n)$ bits.
- La communication totale entre CC_1 et la station de base est $O_{T_2} = L \times (O_1 + O_2)$ bits.
- La durée d'une communication d'un paquet est $T_1 = O_1/Db$ secondes.
- La durée d'une communication d'un accusé de réception ACK est $T_2 = O_2/Db$ secondes.
- Le coût de communication d'un paquet par rapport au temps :

- Si $0 \leq t \leq L \times T_1$ alors $O_{r_2}(t) = O_1/T1 \times t$ bits.
- Si $L \times T1 < t \leq L \times (T1 + T2)$ alors $O_{T_2}(t) = O_2/T_2 \times t$ bits.
- Sinon $O_{T_2}(t) = O_{T_2}(L \times (T_1 + T_2)) = O_2/T_2 \times [(L \times T_1) + (L \times T_2)]$ bits.

- Le coût de stockage des paquets par rapport au temps :

- Si $0 \leq t \leq L \times T_1$ alors $S_{T_2}(t) = S_{T_1}(t) = S_1/T_1 \times t$ bits.
- Si $L \times T1 < t \leq L \times (T_1 + T_2)$ alors $S_{T_2}(t) = -(S_1/T_1)$ bits.
- Sinon $S_{T_2}(t) = S_{T_2}(L \times (T_1 + T_2))$ bits.
  Le coût de communication de deux paquets M_1 et M_2 entre CC_i et la station de base par rapport au temps tel que le deuxième paquet arrive à l'instant λ_1 :

- Si $0 \leq t \leq \lambda_1$ alors $O_{T_2}(t) = O_1/T_1 \times t$ bits.
- Si $\lambda_1 < t \leq L \times T_1$ alors $O_{T_2}(t) = (O_1/T_1 \times t) + (O_1/T_1 \times (t - \lambda_1))$ bits.
- Si $L \times T_1 < t \leq (L \times T1) + \lambda_1$ alors $O_{T_2}(t) = (O_2/T_2 \times t) + (O_1/T_1 \times (t - \lambda_1))$ bits.
- Si $(L \times T_1) + \lambda_1 < t \leq L \times (T_1 + T_2)$ alors $O_{T_2}(t) = (O_2/T_2 \times t) + (O_2/T_2 \times (t - \lambda_1))$ bits.
- Si $L \times (T_1 + T_2) < t \leq L \times (T_1 + T_2) + \lambda_1$ alors $O_{T_2}(t) = O_2/T_2 \times (t - \lambda_1)$ bits.
- Sinon $O_{T_2}(t) = O_{T_2}(L \times (T_1 + T_2) + \lambda_1)$ bits.

  Le coût de stockage de deux paquets M_1 et M_2 entre CC_i et la station de base par rapport au temps tel que le deuxième paquet arrive à l'instant λ_1 :

- Si $0 \leq t \leq \lambda_1$ alors $S_{T_2}(t) = S_1/T_1 \times t$ bits.

- Si $\lambda_1 < t \leq L \times T_1$ alors $S_{T_2}(t) = (S_1/T_1 \times t) + (S_1/T_1 \times (t - \lambda_1))$ bits.
- Si $L \times T_1 < t \leq (L \times T1) + \lambda_1$ alors $S_{T_2}(t) = (S_1/T_1 \times (t - \lambda_1)) - (S_1/T_1)$ bits.
- Si $(L \times T_1) + \lambda_1 < t \leq L \times (T_1 + T_2)$ alors $S_{T_2}(t) = -2 \times (S_1/T_1)$ bits.
- Si $L \times (T_1 + T_2) < t \leq L \times (T_1 + T_2) + \lambda_1$ alors $S_{T_2}(t) = -(S_1/T_1)$ bits.
- Sinon $S_{T_2}(t) = S_{T_2}(L \times (T1 + T2) + \lambda_1)$ bits.

La figure 3.3 illustre respectivement, le cout de stockage et de communication du service d'authentification préventive en fonction du temps.

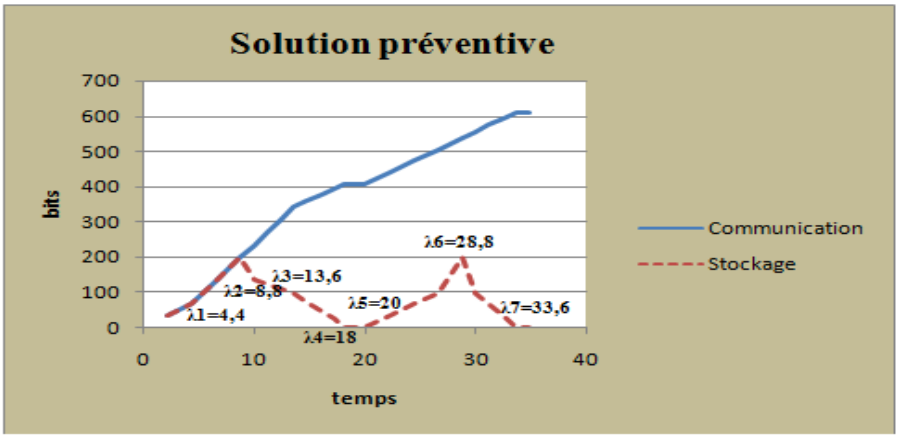

FIGURE 3.3 – Le coût de communication et de stockage d'un paquet suivant la solution préventive

- λ_1 **:** temps d'envoie du deuxième message par rapport au premier message.
- λ_2 **:** temps d'arrivée du premier message à la station de base.
- λ_3 **:** temps d'arrivée de l'accusé de réception du premier message au CC_1.
- λ_4 **:** temps d'arrivée de l'accusé de réception du deuxième message au CC_1.
- λ_5 **:** temps d'envoie du troisième message par rapport au premier message.
- λ_6 **:** temps d'arrivée du troisième message à la station de base.
- λ_7 **:** temps d'arrivée de l'accusé de réception du troisième message au CC_1.

D'après les résultats, nous pouvons constater que le cout de stockage est très faible par rapport au cout de communication, puisque les nœuds suppriment les informations enregistrés dès qu'ils reçoivent un l'accusé de réception afin d'éliminer toute information inutile et décharger la mémoire des nœuds capteurs. Cependant l'overhead de communication augmente en raison du nombre de paquets envoyés à chaque communication.

3.3 Modélisation analytique

Actuellement, il existe beaucoup de formalismes de modélisation. Le choix de formalismes est déterminé d'une part par la nature du système à modéliser et d'autre part par les résultats attendus. Nous présentons une modélisation analytique à base de chaînes de Markov à travers laquelle nous menons une étude des deux solutions.

3.3.1 Chaîne de Markov stochastique

La figure 3.4 illustre le graphe de transition de la première solution et la figure 3.5 illustre le graphe de transition de la deuxième solution qui montrent comment transférer un paquet d'un capteur à un autre selon certaines probabilités.

3.3.2 Modèle de la solution détective

Le modèle de la première solution comporte un ensemble de M_1 d'états tel que $M_1 = 6(N-1)+3$ et N est le nombre de nœuds de la route entre le CC_i et la station de base.

- **L'ensemble des états E_1 sont :** $E_1 = \{EM1_1, EM1_2, EM1_3, ..., EM1_N, EM2_1, EM2_2, EM2_3, ..., EM2_N, EM2_{SB}, EM3_1, EM3_2, EM3_3, ..., EM3_N, AttDoS_1, AttDoS_2, AttDoS_3, ..., AttDoS_N, AttDrp_1, AttDrp_2, AttDrp_3, ..., AttDrp_N, AttDrp_{SB}, AttSlf_1, AttSlf_2, AttSlf_3, ..., AttSlf_N, AttSlf_{SB}, Succs, Echec\}$.

– L'état " Succès" représente le succès du service d'authentification.
– L'état " Echec" représente l'échec du service d'authentification.
– L'état "$EM1_i$" représente l'envoie du paquet REQ du capteur i.
– L'état "$EM2_i$" représente l'envoie du paquet $repREQ$ du capteur i.
– L'état "$EM3_i$" représente l'envoie du paquet $repREQ1$ du capteur i.
– L'état "$AttDoS_i$" représente une attaque DoS au niveau du capteur i.
– L'état "$AttDrp_i$" représente une attaque Dropping au niveau du capteur i.
– L'état "$AttSlf_i$" représente une attaque Selfish au niveau du capteur i.

- **Les probabilités de transition :**

– P^i_{DoS} représente la probabilité d'une attaque DoS au niveau du capteur i.
– $1 - P^i_{DoS}$ représente la probabilité de l'envoie du paquet $repREQ1$ du capteur i.
– P^i_{Drp} représente la probabilité d'une attaque Dropping au niveau du capteur i.

- $1 - P^i_{Drp}$ représente la probabilité de l'envoie du paquet REQ du capteur i.
- P^i_{Slf} représente la probabilité d'une attaque Selfish au niveau du capteur i.
- $1 - P^i_{Slf}$ représente la probabilité de l'envoie du paquet $repREQ$ du capteur i.

Graphe de transition

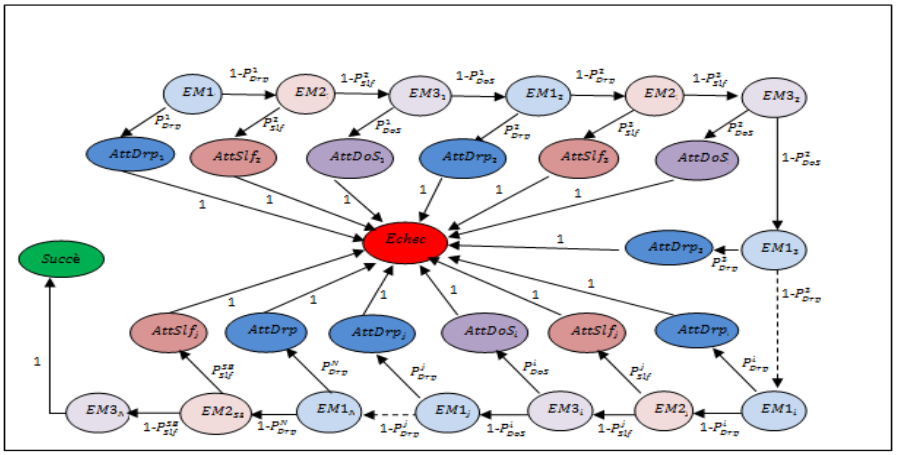

FIGURE 3.4 – Graphe de transition de la chaîne de Markov de la solution détective

3.3.2.1 Métriques de performance

Afin d'évaluer cette solution, nous avons défini les métriques suivants :

1. **Probabilité de succès du service d'authentification de bout en bout :** elle représente la probabilité de succès du service d'authentification à un nœud j qui est définie comme suit :

$$P^{j}_{succ1} = \prod_{i=1}^{j}[(1 - P^{i}_{Drop}) \times (1 - P^{i}_{Slf}) \times (1 - P^{i}_{DoS})]$$

2. **Probabilité d'échec du service d'authentification avec une attaque Dropping :** la probabilité d'échec à un nœud j signifie que tous les nœuds précédents ont assurés le service d'authentification, mais le nœud j a subit une attaque de type Dropping, cette probabilité est définie comme suit :

$$P^{j}_{AttDrp1} = P^{j-1}_{succ1} \times P^{j}_{Drp}$$

3. **Probabilité d'échec du service d'authentification avec une attaque Selfish :** la probabilité d'échec à un nœud j signifie que tous les nœuds précédents ont assuré le service d'authentification, mais le nœud j a subit une attaque de type Selfish, cette probabilité est définie comme suit :

$$P^{j}_{AttSlf1} = P^{j-1}_{succ1} \times P^{j}_{slf}$$

4. **Probabilité d'échec du service d'authentification avec une attaque DoS :** la probabilité d'échec à un nœud j signifie que tous les nœuds précédents ont assuré le service d'authentification, mais le nœud j a subit une attaque de type Dos, cette probabilité est définie comme suit :

$$P^{j}_{AttDoS1} = P^{j-1}_{succ1} \times P^{j}_{DoS}$$

5. **Probabilité d'échec du service d'authentification avec une attaque quelconque :** la probabilité d'échec à un nœud j signifie que tous les nœuds précédents ont assuré le service d'authentification, mais le nœud j a subit une attaque de type DoS, Selfish où bien Dropping. Cette probabilité est définie comme suit :

$$P^{j}_{Att1} = P^{j-1}_{succ1} \times (P^{j}_{DoS} + P^{j}_{DoS} + P^{j}_{DoS})$$

3.3.3 Modèle de la solution préventive

Le modèle de notre deuxième solution comporte un ensemble de M_2 états tel que $M_2 = M_1 + 8(N-1) + 5$ et N est le nombre de nœuds de la route entre le CC_i et la station de base.

- **L'ensemble des états E_2 sont :** $E_2 = E_1 \cup ER1_1, ER1_2, ER1_3, ..., ER1_N, ER1_{SB}, ER2_1, ER2_2, ER2_3, ..$

– L'état "$ER1_i$ " représente la bon envoie du paquet ACK du capteur i.
– L'état "$ER2_i$ " représente la bon envoie du paquet repACK du capteur i.

- **Les probabilités de transition :**

– P^{i}_{DoS} représente la probabilité d'une attaque DoS au niveau du capteur i.
– $1 - P^{i}_{DoS}$ représente la probabilité du bon envoie du paquet $repREQ1$ du capteur i.
– P^{i}_{Drp} représente la probabilité d'une attaque Dropping au niveau du capteur i.
– $1 - P^{i}_{Drp}$ représente la probabilité du bon envoie du paquet REQ du capteur i.
– P^{i}_{Slf} représente la probabilité d'une attaque Selfish au niveau du capteur i.
– $1 - P^{i}_{Slf}$ représente la probabilité du bon envoie du paquet $repREQ$ du capteur i.

- **Graphe de transition**

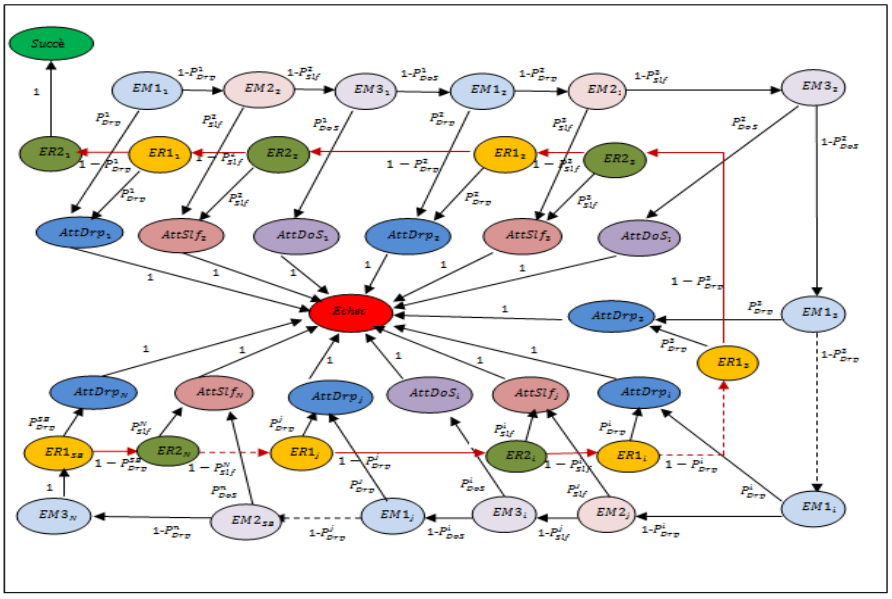

FIGURE 3.5 – Graphe de transition de la chaîne de Markov de la solution détective

3.3.3.1 Métriques de performance

Afin d'évaluer cette solution, nous avons défini les métriques suivant :

1. **Probabilité de succès du service d'authentification de bout en bout :**elle représente la probabilité de succès du service d'authentification à un nœud j qui est définie comme suit :
   $P^j_{succ2} = \prod_{i=1}^{SB}[(1 - P^i_{Drp})(1 - P^i_{Slf})(1 - P^i_{DoS})] \times \prod_{i=SB}^{j+1}[(1 - P^i_{Drp})] \times (1 - P^i_{Slf})$
   D'où

$$P^j_{succ2} = P^{SB}_{succ1} \times \prod_{i=SB}^{j+1}[(1 - P^i_{Drp})] \times (1 - P^i_{Slf})$$

2. Probabilité d'échec du service d'authentification avec une attaque Dropping : la probabilité d'échec à un nœud j signifie que tous les nœuds précédents ont assuré le service d'authentification, mais le nœud j a subit une attaque de type Dropping, cette probabilité est définie comme suit :

$$P^j_{AttDrp2} = (P^{j-1}_{succ1} \times P^j_{Drp}) + (P^{j-1}_{succ2} \times P^j_{Drp})$$

3. Probabilité d'échec du service d'authentification avec une attaque Selfish : la probabilité d'échec à un nœud j signifie que tous les nœuds précédents ont assuré le service d'authentification, mais le nœud j a subit une attaque de type Selfish. Cette probabilité est définie comme suit :

$$P^j_{AttSlf2} = (P^{j-1}_{succ1} \times P^j_{Slf}) + (P^{j-1}_{succ2} \times P^j_{Slf})$$

4. Probabilité d'échec du service d'authentification avec une attaque DoS : la probabilité d'échec à un nœud j signifie que tous les nœuds précédents ont assuré le service d'authentification, mais le nœud j a subit une attaque de type Dos. Cette probabilité est définie comme suit :

$$P^{j}_{AttDoS2} = P^{j}_{AttDoS1} = (P^{j-1}_{succ1} \times P^{j}_{DoS})$$

5. Probabilité d'échec du service d'authentification avec une attaque quelconque : la probabilité d'échec à un nœud j signifie que tous les nœuds précédents ont assuré le service d'authentification, mais le nœud j a subit une attaque de type DoS. Cette probabilité est définie comme suit :

$$P^{j}_{Att2} = P^{j-1}_{succ1} \times (P^{j}_{DoS} + P^{j}_{DoS} + P^{j}_{DoS}) + P^{j-1}_{succ2} \times (P^{j}_{Slf} + P^{j}_{Drp})$$

3.3.4 Résultats obtenus

3.3.4.1 Probabilité d'échec en fonction du nombre de nœuds dans le réseau

Supposons que le nombre de nœuds dans le réseau S est variable et le nombre de nœuds malicieux N_m est fixé. La probabilité de présence d'une attaque est $Ps_{Att} = N_m/S$.

- **La première solution (détective)**
  La probabilité d'échec du service d'authentification avec une attaque quelconque est définie comme suit :

$$Ps^{j}_{Att1} = \prod_{i=1}^{j-1}[(1 - Ps_{Att})] \times Ps_{Att}$$

  La figure 3.6 illustre la probabilité d'échec du service d'authentification en fonction du nombre de nœuds dans le réseau, en fixant le nombre de nœud malicieux (N_m) tele que :

– Le nombre de nœuds dans le réseau S varie de 50 à 500.
– Le nombre de nœuds malicieux dans le réseau $N_m = 10$.

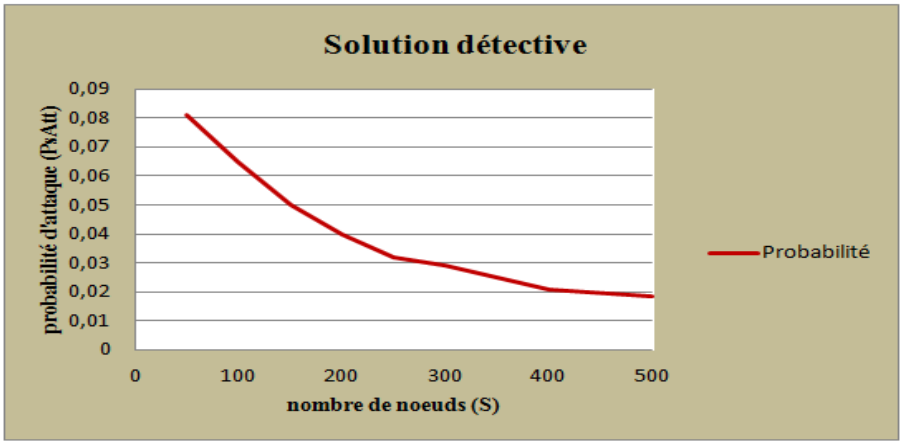

FIGURE 3.6 – Probabilité d'échec du service d'authentification pour la solution détective

- **La deuxième solution (préventive)**
  La probabilité d'échec du service d'authentification avec une attaque quelconque est définie comme suit :

$$Ps^{j}_{Att2} = Ps^{j}_{Att1} + \prod_{i=SB}^{j+1}[(1 - Ps_{Att}] \times Ps_{Att}$$

  La figure 3.7 illustre la probabilité d'échec du service d'authentification en fonction du nombre de nœuds dans le réseau en fixants le nombre de nœuds malicieux (N_m), tel que :

– Le nombre de nœuds dans le réseau S varie de 50 à 500.
– Le nombre de nœuds malicieux dans le réseau $N_m = 10$.

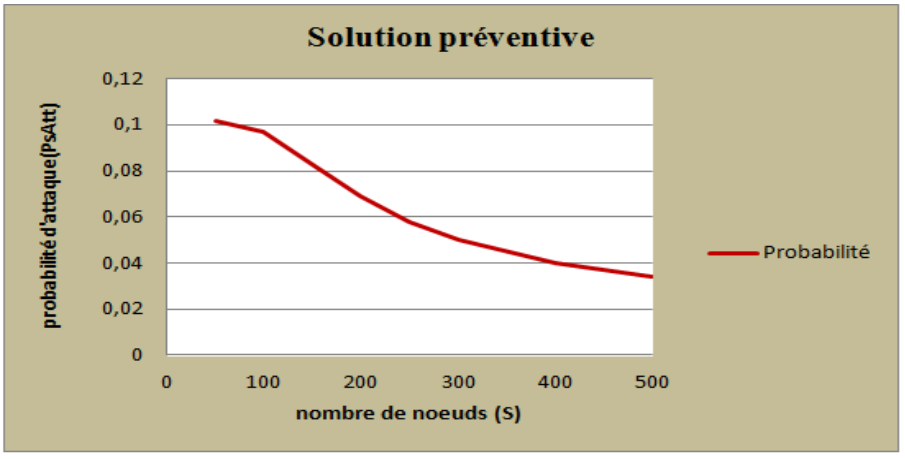

FIGURE 3.7 – Probabilité d'échec du service d'authentification en fonction du nombre de nœuds dans le réseau pour la solution préventive

3.3.4.2 Probabilité d'échec en fonction du nombre de nœuds malicieux

Supposons que le nombre de nœuds dans le réseau est fixé et le nombre de nœuds malicieux N_m est variable et la probabilité des trois attaques est définie comme suit : $Pn_Att = N_m/S$.

- **La première solution (détective)**

  La figure 3.8 illustre la probabilité d'échec du service d'authentification en fonction du nombre de nœuds malicieux dans le réseau (N_m), en fixants le nombre de nœuds dans le réseau.
  Tel que :

$$Pn^j_{Att1} = \prod i = 1^{j-1}[(1 - PnAtt)] \times Pn_{Att}$$

– Le nombre de nœuds dans le réseau $S = 500$.
– Le nombre de nœuds malicieux dans le réseau N_m varie de 10 à 200.

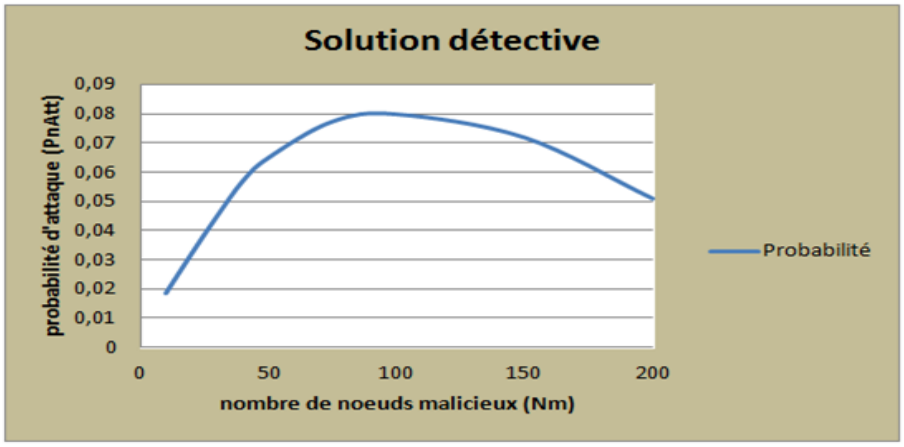

FIGURE 3.8 – Probabilité d'échec de service d'authentification en fonction du nombre de nœuds malicieux dans le réseau pour la solution détective

- **La deuxième solution (préventive)**
  La probabilité d'échec du service d'authentification avec une attaque quelconque est définie comme suit :

$$Pn^j_{Att2} = Pn^j_{Att1} + \prod_{i=SB}^{j+1}[(1 - Pn_{Att}] \times Pn_{Att}$$

  La figure p3.4 illustre la probabilité d'échec de service d'authentification en fonction du nombre de nœuds malicieux dans le réseau (N_m), en fixant le nombre de nœuds dans le réseau. Tel que :

– Le ombre de nœuds dans le réseau $S = 500$.
– Le nombre de nœuds malicieux dans le réseau N_m varie de 10 à 200.

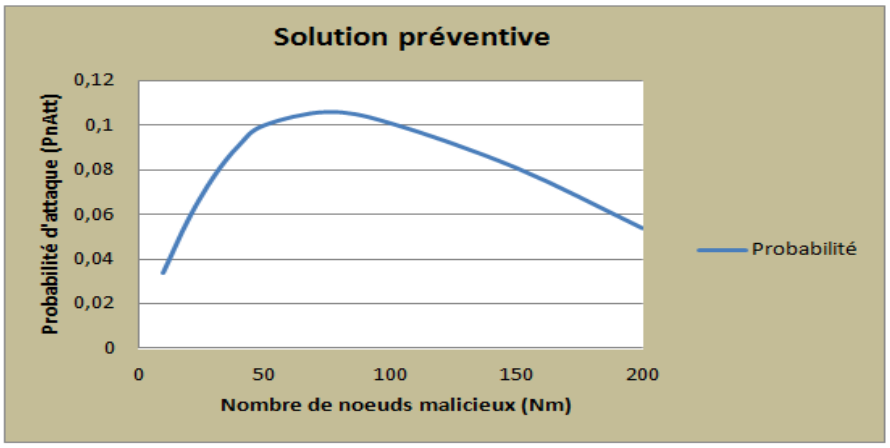

FIGURE 3.9 – Probabilité d'échec de service d'authentification en fonction du nombre de nœuds malicieux dans le réseau pour la solution préventive

3.3.4.3 Probabilité d'échec en fonction du nombre de sauts

Supposons que le nombre de nœuds dans le réseau S est fixe et le nombre de nœuds malicieux N_m est fixe. La probabilité des trois attaques et la mêmes donc cette probabilité est définie comme suit :
$Pn_{Att} = Nm/S$

- **La première solution (détective)**
  La probabilité d'échec du service d'authentification avec une attaque quelconque est définie comme suit :

$$Pn^{j}_{Att1} = \prod i = 1^{j-1}[(1 - Pn_{Att})] \times Pn_{Att}$$

  La figure 3.10 illustre la probabilité d'échec de service d'authentification en fonction du nombre de sauts dans le réseau, en fixons S et N_m. Tel que :

– Le nombre de nœuds dans le réseau $S = 100$.
– Le nombre de nœuds malicieux dans le réseau $N_m = 30$.

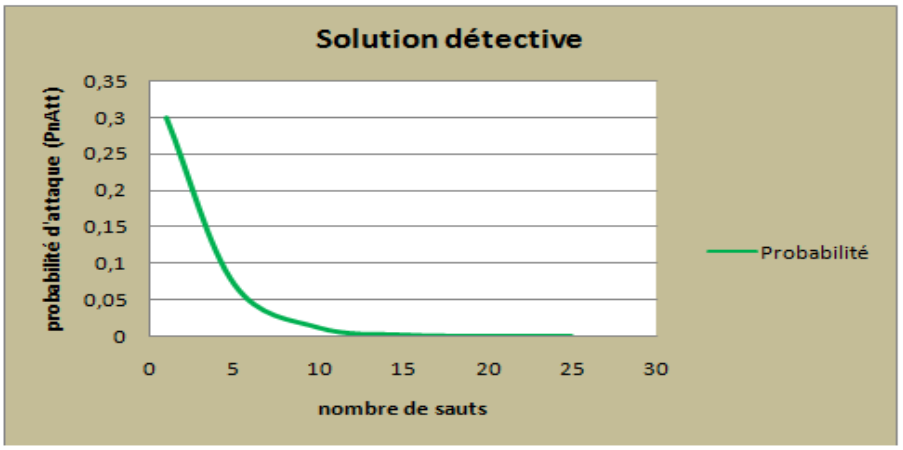

FIGURE 3.10 – Probabilité d'échec de service d'authentification en fonction du nombre de sauts pour la solution détective

- **La deuxième solution (préventive)** La probabilité d'échec du service d'authentification avec une attaque quelconque est définie comme suit :

$$PnAtt_2^j = PnAtt1^j + \prod_{i=SB}^{j+1}[(1 - Pn_{Att}] \times PnAtt$$

La figure 3.11 illustre la probabilité d'échec de service d'authentification en fonction du nombre de sauts dans le réseau, en fixants S et N_m. Tel que :

– Le nombre de nœuds dans le réseau $S = 100$.
– Le nombre de nœuds malicieux dans le réseau $N_m = 30$.

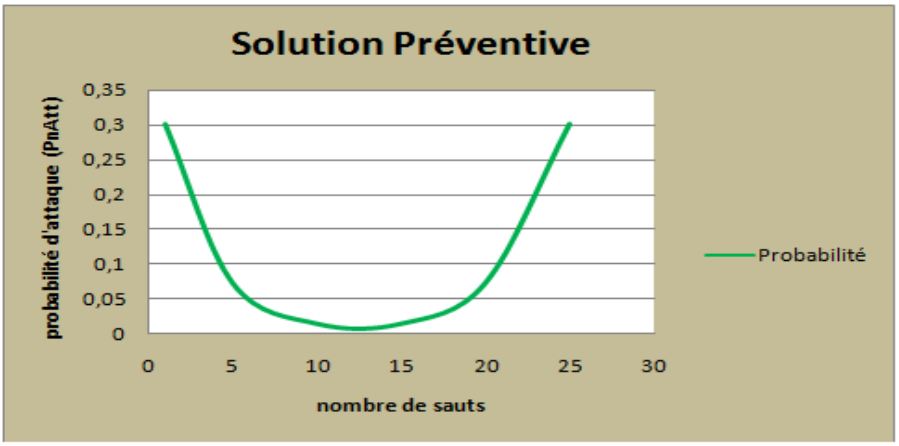

FIGURE 3.11 – Probabilité d'échec de service d'authentification en fonction du nombre de sauts pour la solution préventive

3.4 simulation

Nous utilisons la simulation afin de validé les résultats analytiques de notre protocole proposée,comme outil de simulation, nous avons développé le simulateur sous C++. Le déploiement des nœuds est uniforme et aléatoire. Les les résultats obtenus sont la moyenne de 100 itérations simulées. Dans un premier temps nous avons examiné le comportement du protocole en fonction du nombre de capteurs dans le réseau qui varie de 50 à 500 et en fixant le nombre de capteurs malicieux à 10. Les nœuds sont déployés sur une zone carrée de dimension 1000x1000 m^2. Dans un deuxième temps nous avons évalué les performances du protocole en fonction de nombre de nœuds malicieux qui varie de 10 à 500 et fixant $n = 500$.

3.4.1 Résultats

3.4.1.1 En fonction de nombre de nœuds dans le réseau

La figure 3.12 représente les résultats de la probabilité d'échec avec la variation de nombre de nœuds S de 50 à 500 déployés aléatoirement tel que le nombre de nœuds malicieux est $ma = 10$.

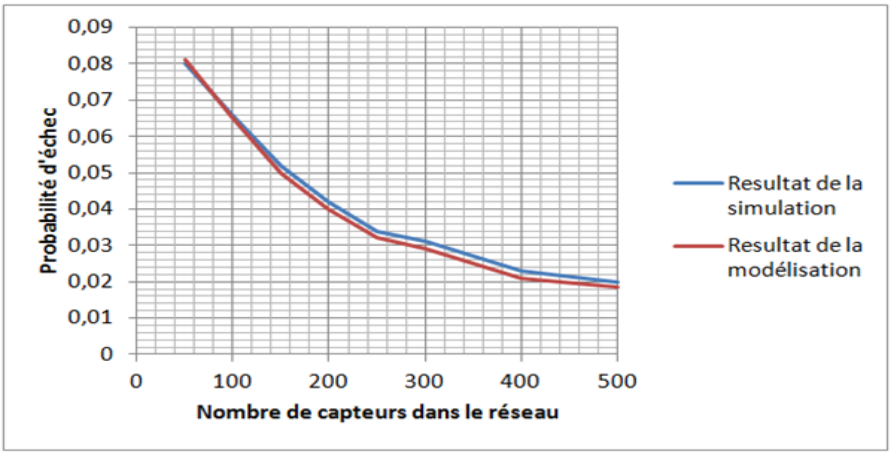

FIGURE 3.12 – Le taux d'échec en fonction de nombre de nœuds dans le réseau

3.4.1.2 En fonction de nombre de nœuds malicieux dans le réseau

La figure 3.13 représente les résultats de la variation de nombre de nœuds malicieux ma de 10 à 500 telle que le nombre de nœuds dans le réseau $S = 500$.

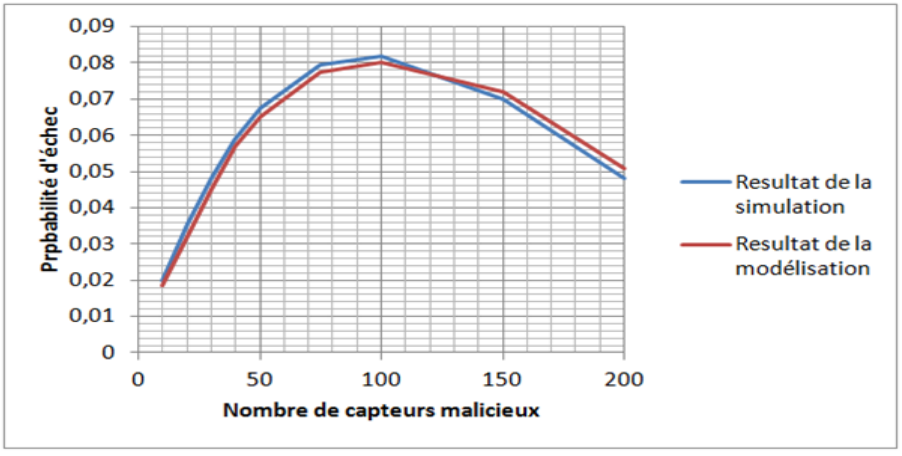

FIGURE 3.13 – Le taux d'échec en fonction de nombre de nœuds malicieux dans le réseau

Conclusion

Ce chapitre a été axé sur la validation et l'analyse de performances du protocole proposé. Nous avons développé un modèle analytique à base de chaîne de Markov et évalué les performances de notre protocole en termes de l'échec de service d'authentification en fonction de nombre de capteurs dans le réseau et en fonction de nombre de capteurs malicieux dans le réseau.

CONCLUSION GÉNÉRALE ET PERSPECTIVES

Un RCSF est une collection de capteurs déployés dans une zone géographique pour collecter des informations. Ce genre de réseau est utilisable par plusieurs applications à cause de sa facilité, sa rapidité et son coût réduit de déploiement, mais il est vulnérable à des attaques possibles à cause de ses caractéristiques inhérentes. Pour remédier à ce problème de sécurité, plusieurs solutions ont été proposées. La plupart des protocoles de routage sécurisés proposées pour les RCSFs se basent sur la cryptographie asymétrique et la méthode de pré-distribution de clés afin d'achever l'établissement des clés entre les entités communicantes dans le réseau. Nous avons étudié un ensemble de ces protocoles de routage sécurisés qui permettent d'offrir les services de sécurité de base en les classifiant selon la topologie du réseau. Après l'étude de ces solutions, nous avons remarqué que ces protocoles sont très couteux en termes de stockage, de calcul et de communication puisque les nœuds doivent sauvegarder un ensemble de clés ainsi que les certificats. D'où le défi est de trouver un compromis entre la solution sécurisée et les contraintes caractérisant les RCSFs. Dans le premier chapitre, une introduction a été apportée aux réseaux de capteurs sans fil, d'où nous avons introduit certaines applications potentielles qui démontrent leur utilité ainsi que les différentes problématiques révélées dans ce type des réseaux. Ensuite, nous avons présenté un état de l'art sur certains protocoles de routage sécurisés. Le deuxième chapitre est consacré pour notre protocole nommé IBC2HAP (IBC based Hop by Hop Authentication Protocol) basé sur la cryptographie à base d'identité (CBID) pour objectif principal d'assurer l'authentification de bout-en-bout et qui permet de prendre en considération les contraintes caractérisant les RCSFs. Le protocole IBC2HAP apporte deux solutions, une solution détective applicable dans les cas ou les informations échangées ne demandent pas assez d'espace mémoire ou l'échange de données sont de tailles très petites. La deuxième solution, est préventive qui reste applicable dans les domaines où les communications sont réduites en ce qui concerne le troisième chapitre, il est consacré à la modélisation et à l'évaluation de performance des deux solutions selon un modèle analytique suivie par une simulation afin de validé les résultats.

En guise de perspective, nous envisageons d'intégrer notre protocole IBC2HAP à un mécanisme de routage sécurisé et d'approfondir l'analyse de notre protocole par la vérification du modèle à travers des simulations, afin de représenter les résultats pour d'éventuelles comparaisons et amélio-

rations et mesurer les faiblesses de notre solution. En fin nous souhaitons de mettre en pratique notre protocole IBC2HAP dans une application réelle des réseaux capteurs.

Bibliographie

[1] Claudio Silva, Rodrigo Costa, Adonias Pires, Denis Rosário, Eduardo Cerqueira, Kássio Machado , Augusto Neto, and Jó Ueyama. *A Cluster-based Approach to provide Energy-Efficient in WSN*. IJCSNS International Journal of Computer Science and Network Security, VOL.13 No.1, page 55-62, January 2013.

[2] Yi Jiang, Jiang -Pingli, An-Ping xiong1, Zhong-Jinglin. *A Research of SPINS based on identity authentication for wireless sensor networks*. IEEE, pages 325-329, 2012.

[3] Y.Yaser. *Routage pour la gestion de l'énergie dans les réseaux de capteurs sans fil*. PhD Thesis, Universite de Haute Alsace Faculté des Sciences et Techniques, 2011.

[4] A.Ayache and D.Mouloudj. *Routage avec conservation d'énergie dans les réseaux de capteurs sans fil*. 2010.

[5] D.Martins. *Sécurité dans les réseaux de capteurs sans fil Stéganographie et réseaux de confiance*. Université de Franche-Comté, 2010.

[6] J.Xu, W.Liu, F.Lang, Y.Zhang, and C.Wang. *Distance measurement model based on RSSI in WSN*, Jornal of Wireless Sensor Network 606-611, 2010.

[7] M.Lehsaini. *Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique*. PhD thesis, Université A.B Tlemcen, 2009.

[8] Y. Challal. *Réseaux de Capteurs Sans Fils. Systemes intelligent pour le transport*, 2008.

[9] A.Makhoul. *Réseaux de capteurs : localisation, couverture et fusion de données*. PhD thesis, Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC), 2008.

[10] Imad Bou Akl. *Etude des protocoles et infrastructures de sécurité dans les réseaux*, Université Paul Sabatier-I.R.I, 2006.

[11] J. Ibriq, I. Mahgoub. *A secure hierarchical routing protocol for wireless sensor networks*, Proc. 10th IEEE Int. Conference on Communication Systems (ICCS'06), Singapore, pages 1-6, October 2006.

[12] J. Yin, S. Madria : Secrout. *A secure routing protocol for sensor networks*, Proc. IEEE 20th Int. Conference on Advanced Information Networking and Applications (AINA'06), Vienna, Austria, pages 18-20, April 2006.

[13] R. Watro, D. Kong, S. Cuti, C. Gardiner, C. Lynn, P. Kruus. *TinyPK Securing sensor networks with public key technology*, Proc. 2nd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'04), Washington, DC, USA, pages 59-64, October 2004.

[14] L.Khelladi and N.Badache. *Les réseaux de capteurs : état de l'art*, Février 2004.

[15] M. Tubaishat, J. Yin, B. Panja, S. Madria. *A secure hierarchical model for sensor network*, ACM SIGMOD Rec. 2004.

[16] Park, K. Shin. *LiSP : A lightweight security protocol for wireless sensor networks*, Trans. Embed. Comput. Syst, 2004.

[17] S. Zhu, S. Setia, S. Jajodia. *LEAP : Efficient security mechanisms for large-scale distributed sensor networks*, Proc. 10th ACM Conference on Computer and Communication Security, Washington, DC, USA, pp. 62-72, October 2003 .

[18] I.F.Akyildiz, W. Su, Y. Sankarasubramaniam and K. Cayirci. *A survey on sensor networks*. IEEE Communication Magazine, 2002.

[19] A. Perrig, R. Canetti, J. Tygar, D. Song. *The TESLA broadcast authentication protocol, RSA Cryptobytes*, 2002.

[20] S. Lindsey, C.S. Raghavendra *PEGASIS : power efficient gathering in sensor information systems*, Proc. IEEE Aerospace Conference, Big Sky, Montana, USA, pages1125-1130 March 2002.

[21] L. Zhou, F. Schneider, R. Van Renesse. *COCA : A secure distributed online certification authority*, ACM Trans. Comput. Syst. Pages 329-368, 2002.

[22] D. Boneh, H. Shacham.*Fast variants of RSA, RSA Cryptobytes* , pages 1-9, 2002.

[23] S. Setia, S. Koussih, S. Jajodia, E. Harder : Kronos. *A scalable group re-keying approach*, Proc. IEEE Symposium on Security and Privacy, Oakland, CA, USA, May 2000.

[24] V. Park, S. Corson. *Temporally-ordered routing algorithm*, October 1999.

[25] H. Krawczyk, M. Bellare, R. *Canetti : HMAC : keyed-hashing for message authentication*, February 1997.

[26] A. Shamir, *Identity-based Cryptosystems and Signature Schemes, Proceedings of CRYPTO'84*, LNCS 196, pages 47-53, Springer-Verlag, 1984.

Résumé

Grâce aux progrès récents des technologies micro-électroniques et des communications sans fil, un nouveau type de réseaux a vu le jour : les réseaux capteurs. Par ailleurs, l'un des problèmes majeurs de ce type de réseaux, est la garantie de sécurité de communication. Et pour cela, plusieurs travaux ont été proposés afin de résoudre ce genre de problème. Dans ce travaille, nous proposons un protocole d'authentification, nommé IBC2HAP *(IBC based Hop by Hop Authentication Protocol*). L'idée fût d'adapter le mécanisme de la cryptographie à base d'identité, celle qui parviennent à résoudre les problèmes liés à l'authentification, à la gestion et à la révocation des clés publiques, néanmoins ils nécessitent la présence d'un centre de génération de clés (Private Key Generator) qui fournit à chaque utilisateur la clé privée correspondante à sa clé publique. Dont l'objectif c'est d'assurer l'authentification de bout-en- bout les données des capteurs intermédiaires relayant les données jusqu' à la station de base, tout en pérennant en considération les spécificités des RCSFs.

Mots clés **:** Réseaux de capteurs sans-fills(RCSFs), Sécurité, Cryptographie à base d'identité, Authentification , Protocole.

Abstract

Thanks to recent advances in micro-electronic technology and wireless communications, a new type of networks has emerged : sensor networks. Moreover, one of the major problems of this type of network is a safety communication. And for this, several studies have been proposed to solve this kind of problem. In this work, we propose an authentication protocol, called IBC2HAP (IBC based Hop by Hop Authentication Protocol). The idea was to adapt the mechanism of identity-based cryptography, which are able to solve problems related to authentication, management and revocation of public keys, they still require the presence of a Private Key Generator which provides each user with the private key corresponding to the public key. The objective of which is to assure the authentication of data intermediate sensors relaying data until the base station, while only taking into account the specificities of WSN.

Keywords **:** Wireless senssor networks, Security, Identity-Based Cryptography, Authentication, Protocol.