N° d'ORDRE : 06/2013-M/INF

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE
« HOUARI BOUMEDIENE »
FACULTÉ D'ELECTRONIQUE ET D'INFORMATIQUE

MEMOIRE

Présenté pour l'obtention du diplôme de MAGISTER

EN : Informatique

Spécialité : Informatique Mobile

Par : MEDJADBA SANA

Sujet

Authentification de la diffusion dans les réseaux de capteurs sans fil

Soutenu publiquement le 27/11/2013, devant le Jury composé de :

M. M.BENCHAIBA	Maitre de Conférence/B, à l'USTHB	Président
M. N.BADACHE	Professeur, à l' USTHB	Directeur de mémoire
M. D.DJENNOURI	Directeur de Recherche, au CERIST	Examinateur
Mme C.BENZAID	Maitre de Conférence/A, à l'USTHB	Invitée

Remerciements

Tout d'abord, Je tiens à remercier Dieu le tout puissant de m'avoir aidé à réaliser ce travail.

J'exprime ma profonde reconnaissance et mes vifs remerciements à mes encadreurs, Professeur Nadjib BADACHE et Docteur Chafika BENZAID, pour leurs qualités d'encadrement exceptionnel, leurs lectures attentives et pour leurs critiques et suggestions qui ont été d'un grand apport pour la finalisation de ce travail.

Je tiens également à remercier Dr M. BENCHAIBA pour l'honneur qui m'a fait de présider le jury de soutenance.

Mes remerciements s'adressent à Dr D. DJENOURI qui a aimablement accepté d'examiner ce travail.

Je remercie tous mes enseignants du département informatique.

Je voudrais également remercier mes chers parents pour leur soutien, sacrifice et pour leur amour.

Enfin, que tous ceux et celles qui ont contribué de près ou de loin à l' accomplissement de ce travail trouvent l' expression de mes remerciements les plus chaleureux.

Table des matières

Liste des figures

Liste des tableaux

Liste des abréviations

AAI	Auxiliary Authentication Information
CA	Certificate Authority
CRL	Certificates Revocation List
DSA	Digital Signature Algorithm
DoS	Denial of Service
EAR	Eavesdrop And Register
ECC	Elliptic curve cryptography
ECDSA	Elliptic Curve Digital Signature Algorithm
GPS	Global Positioning System
IMBAS	Identity-based multi-user broadcast authentication
LEACH	Low Energy Adaptive Clustering Hierarchy
LEAP	Localized Encryption and Authentication Protocol
MAC	Message Authentication Code
MD5	Message Digest 5
RSA	Rivest Shamir Adleman
SAR	Sequential Assignment Routing
SHA	Secure Hash Algorithm
SMACS	Self organizing Medium Access Control for Sensor networks
SMP	Sensor Management Protocol
SNEP	Sensor Network Encryption Protocol
TADAP	Task Assignment and Data Advertisement Protocol
TCP	Transmission Control Protocol
µTESLA	Micro Timed Efficient Stream Loss-tolerant authentication
UDP- Like	User Datagram Protocol Like

Introduction générale

Les progrès réalisés ces dernières décennies dans les domaines de la microélectronique, de la micromécanique et des technologies de communication sans fil, ont permis de produire avec un coût raisonnable des composants de quelques millimètres cubes de volume. Ces derniers, appelés micro-capteurs, le déploiement de plusieurs d'entre eux, en vue de collecter et transmettre des données de plusieurs capteurs vers un ou plusieurs points de collecte, d'une manière autonome, forme un réseau de capteurs sans fil (WSN : Wireless Sensor Network).

La diminution des coûts matériels, ainsi que l'élargissement de la gamme des capteurs disponibles, a permis d'étendre le champ d'application des réseaux de capteurs. Le domaine militaire a été un moteur initial dans le développement de ces technologies pour l'analyse de terrains dangereux ou la surveillance de mouvements.**[HKY06][VHY09]** Les applications environnementales se sont ensuite multipliées, pour la détection de feux de forêts, la surveillance d'activité volcanique ou sismique **[LV06]**, ou encore le suivi du déplacement d'animaux. On utilise aussi les réseaux de capteurs pour des applications médicales **[WSV08]** comme la veille épidémiologique, ou dans un but commercial **[SWS04]**, pour l'optimisation des processus de stockage, ou encore dans l'agriculture de précision **[SWC07]**, et la construction de maisons intelligentes.**[CEH01]**

Dans beaucoup d'applications des réseaux de capteurs, les données peuvent être menacées par des événements extérieurs qui ne devraient pas arriver au cours du fonctionnement normal du réseau. En particulier, l'authentification, la confidentialité, l'intégrité et la disponibilité des données sont des fonctionnalités importantes que le réseau devrait pouvoir assurer. Garantir de telles caractéristiques est une tâche difficile à cause des spécificités des réseaux de capteurs, à savoir absence d'infrastructure, contrainte d'énergie, topologie dynamique, nombre important de capteurs, sécurité physique limitée, capacité réduite des nœuds. Les contraintes des réseaux de capteurs les rendent donc sujets à différents types de menaces et d'attaques telles que l'interception des données envoyées/reçues par le support sans fil et par la suite la possibilité de modifier et de rejouer les données. L'intrus peut également injecter, saturer ou endommager les équipements du réseau. Dans des applications critiques, de telles attaques peuvent être néfastes et peuvent engendrer des dégâts économiques et sécuritaires majeurs.

Nous nous intéressons dans notre étude à la notion de diffusion dans les réseaux de capteurs, qui est un paradigme de communication efficace et commune, dans laquelle une multitude d'utilisateurs se joignent le réseau et diffusent des messages afin d'obtenir des informations de leur intérêt. Malheureusement, en raison de la nature de la communication sans fil dans les réseaux de capteurs, des adversaires peuvent facilement intercepter le trafic,

usurper l'identité des autres utilisateurs, injecter de fausses données ou modifier le contenu des messages légitimes lors de la transmission multi-sauts. Ainsi, des mécanismes d'authentification doivent être mis en œuvre pour protéger les messages diffusés à partir de divers attaques malveillantes.

L'authentification de la diffusion devient alors un problème critique. Toutefois, en raison que la plupart des nœuds de capteurs sont des appareils à ressources restreintes, la cryptographie à clés publiques était considérée inappropriée auparavant. Quand le thème de l'authentification de diffusion est apparu, les chercheurs ont utilisé uniquement les primitives symétriques, tel que le protocole µTESLA**[LN04]**, qui est un protocole léger et auto-guérison, permettant à un récepteur hors ligne de récupérer les clés perdues immédiatement après qu'il se met en ligne. Cependant, µTESLA est fondé sur le principe de sauvegarde de paquets qui pourra causer des attaques de type Denis de Service et exige une forte synchronisation entre les nœuds de capteurs. Par conséquent, il était considéré inapproprié pour les réseaux de capteurs.

Des recherches récentes montrent que la signature à base de courbes elliptiques (ECC) de 160 bits, peut être vérifiée en environ une seconde. En parallèle, une clé de taille 163 bits à base de ECC offre plus de robustesse qu'une clé RSA de taille 1024 bits. **[CBL09]** Cette amélioration indique qu'on peut inclure des protocoles à base de cryptographie à clés publiques afin d'assurer l'authentification de la diffusion dans les WSNs. Cependant, la vérification de la signature dans ce type des systèmes est relativement lente, et provoque une grande consommation énergétique et un long délai de vérification. Pour cela, l'accélération de la vérification de signature est considérée comme un problème important, en particulier dans des environnements à ressources limitées.

Les auteurs Fan et Gong **[FG12]**, ont proposé une méthode d'accélération de signature à base de courbes elliptiques et à base du schéma ECDSA, par l'exploitation de la coopération entre les nœuds de capteurs. Contrairement aux systèmes de sécurité à base de clés publiques, la cryptographie à base d'identité ne nécessite pas la transmission des certificats, ce qui améliore le coût de calcul, et rend ces systèmes adéquats pour les réseaux de capteurs. Cependant, la vérification de signature à base d'identité est toujours lente et coûteuse en termes de ressources.

Dans notre étude, nous avons proposé une approche d'accélération de vérification de la signature vBNN-IBS**[CKD08]**, qui est une méthode basé sur l'identité ne nécessitant pas l'utilisation de certificat électronique. Nous nous basons dans notre approche sur l'aspect de coopération entre les nœuds, où un ensemble de capteurs doit divulguer les résultats de calculs intermédiaires à leurs voisins lors de la vérification de signature. Par conséquent, un grand nombre de nœuds peuvent accélérer leur procédure de vérification de signature en utilisant les informations divulguées.

Ce document est organisé en cinq chapitres suivis d'une conclusion générale. Un état de l'art est présenté dans les trois premiers chapitres et nos contributions sont détaillées dans les deux derniers.

Dans le premier chapitre, nous introduisons les réseaux de capteurs, leurs caractéristiques, domaines d'applications ainsi que leurs architectures.

Dans le deuxième chapitre, nous abordons l'aspect sécurité dans les réseaux de capteurs. Nous commençons d'abord par la présentation d'une taxonomie des attaques et les concepts gravitant autour de cette thématique, nous discutons par la suite les besoins de sécurité requis par ce type des réseaux, et enfin nous introduisons les différents concepts cryptographiques pour ce type de réseaux.

Dans le troisième chapitre, nous détaillons quelques méthodes symétriques et asymétriques d'authentification de diffusion dans les réseaux de capteurs, en montrant les avantages et les inconvénients de chacune d'elles.

Dans le quatrième et le cinquième chapitre, nous proposons notre méthode vBNN-IBS accélérée. Nous exposons les résultats d'implémentation et de tests de simulation de notre solution, ainsi qu'une comparaison par rapport à d'autres protocoles de diffusion.

Nous finalisons ce manuscrit par une conclusion générale et nous présentons les perspectives de recherche de ce domaine. Les travaux contenus dans ce document ont été publiés dans une conférence disponible dans la partie bibliographie (Voir **[BMB12]**).

Chapitre I : Présentation des réseaux de capteur

1.1. Introduction

Depuis quelques décennies, le besoin d'observer et de contrôler des environnements hostiles est devenu essentiel pour de nombreuses applications militaires et scientifiques. Les nœuds utilisés doivent être autonomes, d'une taille miniature et peuvent être déployés d'une manière dense et aléatoire dans le champ surveillé. Une classe spéciale des réseaux Ad Hoc appelée réseaux de capteurs sans fil vient au secours. Ceux-ci sont apparus grâce aux développements technologiques tels que la miniaturisation des composants électroniques, la diminution des coûts de fabrication et l'augmentation des performances et des capacités de stockage, d'énergie et de calcul.

Dans ce qui suit, nous étudierons ce type de réseau sans fil, ses principales caractéristiques, l'architecture d'un capteur ainsi que les éventuelles applications de ce type de réseaux. Enfin, nous présenterons l'ensemble des facteurs influençant l'architecture des réseaux de capteurs.

1.2. Définition

1.2.1. Capteur

Un capteur est un mini-composant, qui permet d'acquérir des données sur son environnement, les traiter et les communiquer. Son intégration est une tâche difficile à réaliser en tenant compte de certaines contraintes : l'espace mémoire, la consommation énergétique, etc. **[SS06]**

1.2.2. Un réseau de capteurs

Un réseau de capteurs sans fil est composé d'un grand nombre de nœuds capteurs dispersés dans le terrain d'intérêt appelé *champ de captage*. Les nœuds ont la possibilité de collecter périodiquement les données sur le phénomène surveillé et envoyer les rapports de captage à un nœud spécial appelé *puits* (sink). Ce dernier est considéré comme un point de collecte et peut transférer les données collectées via Internet ou satellite à un ordinateur central "gestionnaire de tâche " pour leur traitement (Figure I.1). Ce nœud est responsable, en plus de la collecte des rapports, de la diffusion des demandes sur les types de données requises aux capteurs via des messages de requête.

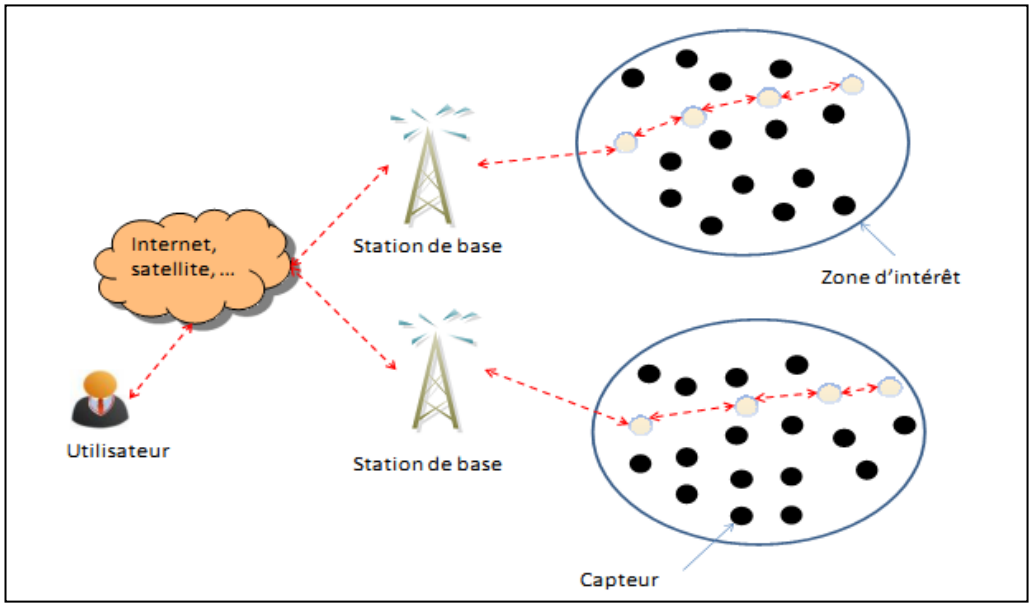


Figure I.1 : Architecture d'un réseau de capteurs sans-fil [AK04]

1.3. Architecture d'un capteur

Un nœud capteur contient quatre (04) unités de base : l'unité de captage, l'unité de traitement, l'unité de transmission et l'unité de contrôle d'énergie **[AK04].** Il peut contenir, suivant son domaine d'application, des modules supplémentaires tels qu'un générateur d'énergie pour les cellules solaires, un système de localisation (GPS) ou un système mobilisateur chargé de déplacer les nœuds capteurs en cas de nécessité.

La figure suivante (Figure I.2) représente l'architecture générale d'un nœud capteur.

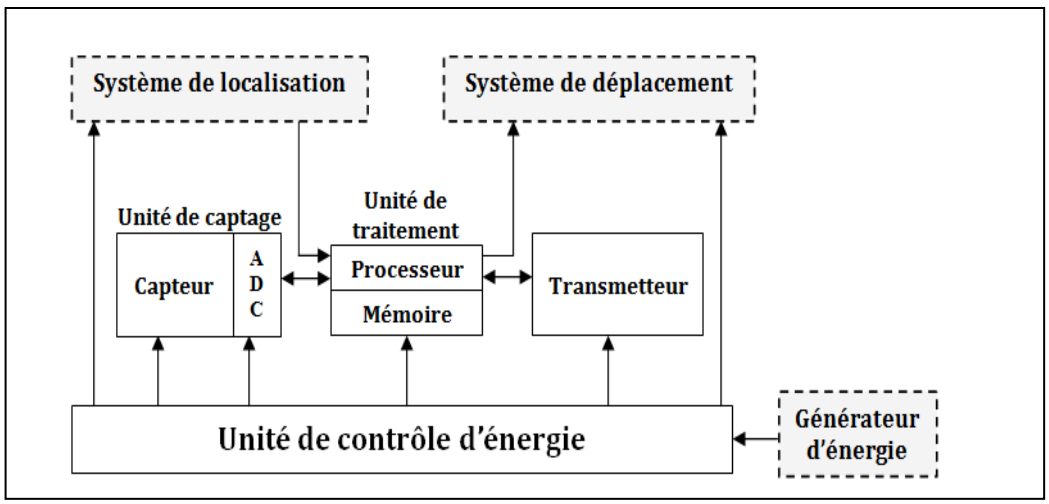


Figure I.2 : Architecture d'un capteur [AK04]

1.3.1. Unité de captage

Elle est composée de deux sous-unités : le capteur lui-même et un convertisseur Analogique/Numérique (ADC). Le capteur est responsable de fournir des signaux analogiques, basés sur le phénomène observé, au convertisseur Analogique/Numérique. Ce dernier transforme ces signaux en un signal numérique compréhensible par l'unité de traitement. **[AK04]**

1.3.2. Unité de traitement

L'unité de traitement est composée d'une mémoire (unité de stockage) et d'un processeur (unité de calcul) qui permet d'effectuer des calculs simples pour que ce nœud puisse collaborer avec les autres nœuds du réseau. De plus, elle possède deux interfaces : **[AK04]**

- La première, liée avec l'unité de captage, par laquelle elle reçoit les mesures détectées.
- La seconde, liée avec l'unité de transmission, par laquelle elle communique les données qu'elle a traitées.

1.3.3. Unité de transmission

Elle est responsable de toutes les émissions et réceptions de données qui représentent l'état actif du nœud. Par ailleurs, le nœud peut se mettre en veille ou écouter seulement le trafic. L'unité de transmission est l'unité qui consomme le plus d'énergie par rapport aux précédentes unités.**[AK04]**

1.3.4. Unité de contrôle d'énergie

L'énergie est la ressource la plus précieuse dans un réseau de capteurs, puisque elle influe directement sur la durée de vie des micro-capteurs et du réseau en entier.

L'unité de contrôle d'énergie constitue donc l'un des systèmes les plus importants. Elle est responsable de répartir l'énergie disponible aux autres modules et de réduire les dépenses en mettant en veille les composants inactifs. Cette unité peut aussi gérer des systèmes de rechargement d'énergie à partir de l'environnement observé telles que les cellules solaires, afin d'étendre la durée de vie totale du réseau. **[SS06]**

1.4. Domaines d'applications des réseaux de capteurs

Les capteurs sont devenus des éléments incontournables dans tous les systèmes où les informations issues de l'environnement extérieur sont nécessaires pour évaluer et agir. Les WSNs couvrent un large panel d'applications. Ceci grâce aux multiples domaines dans lesquels ils peuvent apporter une performance irréprochable. Les besoins de contrôler, prévoir, observer des phénomènes physiques (mesurer des conditions ambiantes tels que la température, l'humidité, le mouvement de véhicules, les incendies, etc.) ont permis aux

WSNs d'envahir de nombreuses applications militaires, industrielles, scientifiques, etc. Dans ce qui suit, des exemples d'applications potentielles sont classifiés en catégories :

1.4.1. Applications militaires

Comme dans le cas de plusieurs technologies, le domaine militaire a été un moteur initial pour le développement des WSNs qui permettent la détection et la collection d'informations sur la position de l'ennemi, la surveillance des zones hostiles et la détection d'agents chimiques et bactériologiques dans l'air.**[HKY06][VHY09]**

1.4.2. Applications à la sécurité

Les altérations dans la structure d'un bâtiment, suite à un séisme ou à un vieillissement, peuvent être détectées par des capteurs intégrés dans les murs ou dans le béton. **[LV06]**

1.4.3. Applications environnementales

Des thermo-capteurs dispersés à partir d'un avion sur une forêt peuvent signaler un éventuel début d'incendie, contrôler la qualité de l'air et recueillir des informations diverses sur l'état du milieu naturel. **[SWC07]**

Sur les sites industriels, les centrales nucléaires ou dans les pétroliers, des capteurs peuvent être déployés pour détecter des fuites de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc.).

Des WSNs peuvent détecter la présence humaine. Ainsi, la climatisation peut être déclenchée seulement aux endroits où il y a des personnes présentes. Une telle application permet de réduire la demande mondiale en énergie réduisant du même coup l'émission des gaz à effet de serre. Rien que pour les États-Unis, on estime cette économie à 55 milliards de dollars par an avec une diminution de 35 millions de tonnes des émissions de carbone dans l'air. **[TS08]**

1.4.4. Applications médicales

Les WSNs sont de plus en plus utilisés dans le domaine médical en se propageant dans de nombreuses applications : la mesure et l'analyse non intrusives de données physiologiques, la surveillance de la température, la fréquence cardiaque, l'oxygénation du sang et le pouls du patient. **[WSV08]**

1.5. Caractéristiques des réseaux de capteurs

L'intégration des WSNs avec le monde physique a rendu leur mode de fonctionnement différent de celui des réseaux informatiques traditionnels. Ils possèdent des caractéristiques particulières qui rendent le développement d'applications non-trivial.

Les WSNs sont apparentés aux réseaux ad-hoc. En effet, ces deux types de réseaux ont de nombreux points communs : **[AM08]**

- Réseaux sans infrastructure ;
- Architecture décentralisée ;
- Autonomie ;
- Utilisation des ondes radio pour communiquer ;
- Une topologie dynamique ;
- La communication se fait par diffusion.

Bien que de nombreux protocoles et algorithmes aient été proposés pour les réseaux ad-hoc traditionnels, ils ne sont pas bien adaptés aux caractéristiques et exigences des WSNs. Les points de différence entre les deux réseaux sont : **[AK04]**

- La densité des nœuds déployés est beaucoup plus importante dans les WSNs ;
- Les nœuds capteurs ont des capacités limitées en énergie et mémoire ;

Les nœuds capteurs, ayant un petit volume, sont limités dans la quantité d'énergie qu'ils peuvent stocker. En outre, ces nœuds sont sensibles à l'échec, suite à l'épuisement des batteries ou bien aux influences de l'environnement. Dans cette section, nous allons montrer les différentes caractéristiques liées aux WSNs telles que le déploiement, la couverture, la connectivité, l'énergie, la mobilité, etc.

1.5.1. Déploiement

Le déploiement des capteurs est la première phase dans le cycle de vie d'un réseau de capteurs. On peut envisager plusieurs formes de déploiements selon les besoins des applications. Les nœuds peuvent être déployés aléatoirement d'un avion ou d'une roquette par exemple, ou bien ils peuvent être placés un par un d'une manière déterministe par un humain ou un robot. Dans un grand nombre d'applications, le déploiement manuel est impossible. De plus, même lorsque l'application permet un déploiement déterministe, le déploiement aléatoire est adopté dans la majorité des scénarios à cause de raisons pratiques tels que le coût et le temps. Cependant, le déploiement aléatoire ne peut pas fournir une distribution uniforme sur la région d'intérêt, ce qui déclenche de nouveaux problèmes dans les WSNs. Les principaux problèmes engendrés sont la localisation, la couverture de la zone, la connexité et la sécurité.**[AM08]**

- **Localisation**

Comme les capteurs sont lâchés aléatoirement, chaque capteur doit définir lui-même sa position, afin de délivrer une information complète aux administrateurs. La localisation sert en particulier à identifier l'origine de l'information (d'où vient-elle?). La majorité des

applications exige la connaissance de la position physique des nœuds pour pouvoir localiser les événements. **[AM08]**

- **Couverture et connectivité**

Un nœud capteur permet de surveiller une zone appelée zone de couverture, cette zone est souvent considérée comme un disque de rayon *Rs*. Un nœud est capable de détecter n'importe quel événement qui se passe dans sa zone de couverture. D'un autre côté, la vision d'un capteur dépend du rayon de réception de son module de communication *Rc*. Un nœud ne peut pas communiquer avec un deuxième sauf si ce dernier se trouve dans sa zone de communication, c'est à dire si la distance Euclidienne entre les deux nœuds est plus petite ou égale à *Rc*.**[AM08]**

- **Sécurité**

En plus des problèmes de sécurité rencontrés dans les réseaux Ad Hoc, les WSNs 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 AdHoc tout en créant d'autres mécanismes de sécurité pour les WSNs. 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.

1.5.2. Énergie et durée de vie

La durée de vie est un élément essentiel pour tous les WSNs. Le but de n'importe quelle application est d'avoir des nœuds placés sur le terrain pour des mois ou des années. Le principal facteur limitant la durée de vie d'un réseau de capteurs est l'énergie.

Comme la seule source d'énergie d'un capteur est une batterie à durée de vie limitée, l'optimisation énergétique doit être prise en considération quelque soit le problème traité. En effet, un WSNs ne peut pas survivre si la perte de nœuds est très importante car ceci engendre des pertes de communications dues à une très grande distance entre les nœuds restants.

Donc, il est très important que les batteries durent le plus longtemps possible, étant donné que dans la plupart des applications, il est impossible de les changer.

1.5.3. Topologie dynamique et connectivité

La dynamicité du réseau découle des défaillances des nœuds ou des cassures des liens entre ceux-ci. La moindre défaillance énergétique d'un capteur peut donc changer significativement la topologie du réseau et imposer une réorganisation coûteuse de ce dernier.

La perturbation des communications (comme les obstacles, l'interférence, etc) peut induire des cassures des liens entre les nœuds voisins. Le redéploiement et l'ajout d'autres nœuds peuvent être envisagés pour pallier quelques défaillances. Dans tous les cas, le réseau de capteurs doit pouvoir se réorganiser rapidement avec un coût énergétique réduit.

1.5.4. Groupement « clustering »

Un réseau de capteurs est souvent constitué de plusieurs milliers de nœuds de capteurs. Pour réduire la complexité des algorithmes de routage, faciliter l'agrégation de données, simplifier la gestion du réseau comme l'affectation des adresses, et optimiser la consommation énergétique, les nœuds sont regroupés dans des clusters. Les nœuds qui sont regroupés ensemble dans un cluster seront capables de communiquer facilement les uns avec les autres. On trouve plusieurs stratégies de groupement parmi lesquelles, les nœuds sont organisés en une hiérarchie en fonction de leur puissance et de leur proximité. Un chef de cluster est élu pour effectuer plusieurs tâches, comme le filtrage, la fusion et l'agrégation, avec la possibilité d'être changé s'il tombe en panne ou s'il arrive à sa limite d'énergie. Toutes les communications de tous les nœuds seront effectuées par l'intermédiaire du chef du cluster auquel ils appartiennent. **[AM08]**

1.5.5. Communication multi-saut

Contrairement aux réseaux traditionnels, un réseau de capteurs est constitué d'un grand nombre de nœuds déployés dans une zone locale, ayant une courte portée (rayon de communication), un faible débit et aucune existence d'infrastructure. Un capteur peut communiquer directement avec ses voisins, c'est-à-dire ceux qui sont à sa portée de communication, et fait office de routeur pour les autres nœuds.

1.6. Facteurs influençant l'architecture des WSNs

La conception et la réalisation des WSNs sont influencées par plusieurs paramètres, parmi lesquels nous citons : la tolérance aux pannes, la sociabilité, le coût de production, l'environnement d'exploitation, la topologie du réseau, les contraintes matérielles et le support de transmission. Ces facteurs importants servent comme directives pour le développent des algorithmes et protocoles utilisés dans les WSNs, ils sont considérés également comme métriques de comparaison de performances entre les différents travaux dans le domaine. **[KB04]**

1.6.1. Tolérance aux fautes

Certains capteurs pourraient être défaillants, bloqués à cause de l'épuisement de leurs batteries, ou subissent des dommages physiques (écrasés par des animaux ou lorsqu'ils sont jetés par un avion). La défaillance de ces capteurs ne devrait pas avoir une influence sur le fonctionnement du réseau. La tolérance aux fautes est la capacité de soutenir les

fonctionnalités d'un réseau de capteurs sans causer d'interruption lorsqu'un capteur cesse de fonctionner.

1.6.2. Passage à l'échelle

La plupart des protocoles sont conçus pour des WSNs d'une grande taille. Cependant, ces protocoles sont dits efficaces si les performances des réseaux ne doivent pas chuter d'une manière drastique quand le nombre de capteurs augmente dans le réseau.

1.6.3. 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.

1.6.4. L'environnement

Les nœuds capteurs sont souvent déployés dans une région géographique distante et sans surveillance. Ils sont soumis à différentes conditions d'environnement. Ils peuvent fonctionner sous haute pression au fond de l'océan, dans un environnement dur tel que les champs de batailles ou même dans des milieux extrêmement froids.

1.6.5. Lien radio

Les capteurs possèdent le matériel nécessaire pour effectuer des communications par ondes radios. Toutefois, la diffusion de l'information via ces liens est peu aisée à cause de l'instabilité et du manque de fiabilité qu'ils présentent. En plus, l'utilisation d'un médium de communication partagé pour faire face aux interférences radio, réduit considérablement la capacité d'exploitation du canal.

1.6.6. Auto-configuration

Les WSNs sont généralement déployés aléatoirement dans des zones d'intérêt hostiles (capteurs largués par un avion). Par conséquent, aucune intervention humaine ne peut être requise pour assurer leur organisation. L'auto-configuration de ces réseaux s'avère nécessaire pour leur bon fonctionnement.

1.7. Architecture protocolaire

Dans le but d'un établissement efficace d'un WSN, une architecture en couches (Figure I.3) est adoptée afin d'améliorer la robustesse du réseau. Une pile protocolaire de cinq couches est donc utilisée par les nœuds du réseau. Citons la couche application, transport, réseau, liaison de données et la couche physique. **[AG07][ASS02]**

De plus, cette pile possède trois plans (niveaux) de gestion : le plan de gestion des tâches qui permet de bien affecter les tâches aux nœuds capteurs, le plan de gestion de mobilité qui

permet de garder une image sur la localisation des nœuds pendant la phase de routage, et le plan de gestion de l'énergie qui permet de conserver le maximum d'énergie. **[BK07]**

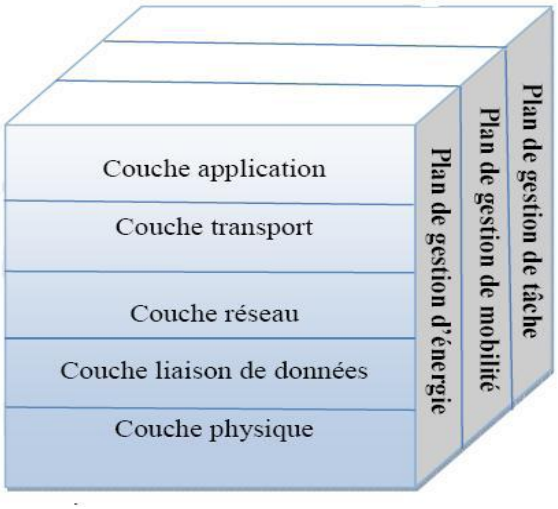


Figure I.3 : La pile protocolaire dans les réseaux de capteur.

1.7.1. Couche application

Elle assure l'interface avec les applications. Il s'agit donc de la couche la plus proche des utilisateurs, gérée directement par les logiciels. Parmi les protocoles d'application, nous citons : SMP *(Sensor Management Protocol)***[SRB05]** et TADAP *(Task Assignment and Data Advertisement Protocol)*.**[SAC04]**

1.7.2. Couche transport

Elle assure le bon acheminement des données et la qualité de la transmission. Dans les WSNs, la fiabilité de transmission n'est pas majeure. Ainsi, les erreurs et les pertes sont tolérées. Par conséquent, un protocole de transport proche du protocole UDP et appelé UDP-Like*(User Datagram Protocol Like)* **[WSH05]**est utilisé.

Cependant, comme le protocole de transport universel est TCP *(Transmission Control Protocol*), les WSNs doivent donc posséder, lors d'une communication avec un réseau externe, une interface TCP-*spliting* **[KSF02]** pour vérifier la compatibilité entre ces deux réseaux communicants.

1.7.3. Couche réseau

Elle établit les routes entre les nœuds capteurs et le nœud puits et sélectionne le meilleur chemin en termes d'énergie, délai de transmission, débit, etc.

Les protocoles de routage conçus pour les WSNs (*Directed Diffusion* **[KB08]** *SAR* (Sequential Assignment Routing) **[AH07],** *LEACH* (Low-Energy Adaptive Clustering

Hierarchy)**[DJB08])** sont différents de ceux conçus pour les réseaux Ad-Hoc puisque les WSNs sont différents selon plusieurs critères comme :

- l'absence d'adressage fixe des nœuds tout en utilisant un adressage basé-attribut ;
- l'établissement des communications multi-sauts ;
- l'établissement des routes liant plusieurs sources en une seule destination pour agréger des données similaires, etc.

1.7.4. Couche liaison de données

Elle est responsable de l'accès au media physique et la détection et la correction d'erreurs intervenues sur la couche physique. De plus, elle établit une communication saut-par-saut entre les nœuds. C'est-à-dire, elle détermine les liens de communication entre eux dans une distance d'un seul saut. Parmi les protocoles de liaison de données, nous citons: SMACS *(Self-organizing Medium Access Control for Sensor networks)* **[LWS04]** et EAR *(Eavesdrop And Register).***[CHC09]**

1.7.5. Couche physique

Elle permet de moduler les données et les acheminer dans le media physique tout en choisissant les bonnes fréquences.

1.8. Conclusion

Les réseaux de capteurs sans fil connaissent un grand essor grâce à la multitude d'applications qu'ils offrent ainsi que leurs caractéristiques inhérentes telles que leur déploiement aléatoire et de faible coût, leur grande mobilité et aussi, grâce aux récents développements concernant la miniaturisation des composants électroniques.

Les mécanismes de sécurité conçus pour les réseaux Ad Hoc ne sont pas applicables aux WSNs. En effet, ceux-ci possèdent plusieurs vulnérabilités à cause de leurs principales caractéristiques qui font que l'application des mesures classiques de sécurité soit restreinte. Le chapitre suivant est consacré à l'étude de la sécurité dans les WSNs.

Chapitre II : La sécurité dans les réseaux de capteur

2.1. Introduction

Les WSNs connaissent actuellement une grande extension et une large utilisation dans différents types d'applications, exigeant une grande sécurité. Nous avons abordé dans le chapitre précédent les principales caractéristiques d'un WSN, entre autres une densité importante, une capacité limitée de calcul et de stockage, une communication sans-fil et une ressource énergétique limitée. Ces contraintes font que l'application des mesures classiques de sécurité soit restreinte. Les points abordés, dans ce chapitre, traitent l'aspect sécurité dans les WSNs, nous commencerons donc à étudier les menaces contre ce type de réseau, par la suite, nous étudierons les services de base de gestion de clés, et nous décrirons les différentes bases mathématiques sur lesquelles se base l'art de la cryptographie récente.

2.2. Les menaces contre les WSNs

Les WSNs peuvent faire l'objet d'un grand nombre d'attaques, chacune avec ses propres objectifs. Par exemple, certaines attaques visent à affecter l'intégrité des messages qui transitent dans le réseau, tandis que d'autres visent à réduire la disponibilité du réseau ou de ses composants. Les attaques se produisent souvent par l'insertion d'éléments intrus dans le réseau (nœud malveillant, qui s'introduit illégitimement). Il existe aussi des attaques contre l'environnement extérieur au réseau, lesquelles provoquent des altérations ou des interférences sur les signaux transmis. Une bonne classification des attaques est présentée dans **[WS02]**. Nous abordons dans la suite les attaques les plus connues dans les WSNs

2.2.1. Déni de service (*Denial of Service*)

Les dénis de service (*DoS*) sont définis comme un mauvais fonctionnement des capteurs d'une manière intentionnée ou par action malveillante. Le déni de service peut ne pas résulter d'une attaque, mais d'un simple événement empêchant le fonctionnement normal d'un de ses services. La plus simple attaque de ce type consiste à empêcher le fonctionnement normal du capteur victime en lui envoyant énormément de messages sans importance, et en interdisant l'accès aux autres utilisateurs. Les contraintes physiques des réseaux de capteurs, et la nature de leur environnement de déploiement, les rendent vulnérables aux attaques *DoS* par rapport à tout autre type de réseau.

Chaque couche du réseau de capteurs WSN a ses propres dénis de service.

- Au niveau de la couche physique, le déni de service se présente comme une attaque "flooding" ou "Tampering".

- Au niveau de la couche liaison, le déni de service peut se présenter comme une collision, ou un "Jamming".
- Au niveau de la couche réseau, le déni de service peut se présenter comme une attaque, "Homing", ou "Black hole".
- Au niveau de la couche de transport, le déni de service est une attaque "Flooding" malveillante ou une "Désynchronisation".

La couche	L'attaque	La défense
Physique	Jamming	Détection et mise en sommeil
	Falsification des capteurs	Route autour des régions de Jamming
Liaison/MAC	Interrogation	Authentification
	Déni de sommeil	Détection de mise en sommeil
Réseau	Modification du contenu des messages de contrôle	Authentification
		Formation des grappes(clusters) sûrs
	Hello flooding	Le routage géographique
	Homing	Encryptage des entêtes
Transport	Synchronisation flooding	Les cookies de synchronisation
	Attaque de désynchronisation	Authentification de paquets
Application	Écrasement de capteurs	L'agrégation de donnée
	DoS basé sur le chemin	Authentification des paquets
	Attaque de déluge	Authentification

Tableau II.1 Les dénis de service par couche et leur défense [RM08]

- **Interférence (*jamming*)**

  L'attaquant peut émettre un signal d'une fréquence proche de celle utilisée dans le réseau afin de brouiller la communication. Cela empêche les nœuds d'échanger les données et provoque l'indisponibilité des canaux de transmission sans fil dans les WSNs.

- **Violation physique (Tampering)**

  La taille des capteurs, plus la manière dont ils sont déployés les rendent très sensibles aux attaques physiques. L'attaquant peut extraire les clés cryptographiques à partir du nœud capturé, altérer ses circuits, modifier les codes de programme ou même de le remplacer par un capteur malveillant.

- **Espionnage des informations de routage**

  L'attaque la plus directe contre un protocole de routage est de cibler les informations de routage dans le réseau. Un attaquant peut usurper, modifier ou rediffuser les informations de routage afin de perturber le trafic dans le réseau.

- **Retransmission sélective (selective forwarding)**

  L'intrus néglige son rôle de routeur et ne transmet pas certains messages qui sont choisis selon certains critères ou même aléatoirement.

- **Attaque du trou noir (Sinkhole ou blackhole)**

  Un nœud falsifie les informations de routage pour forcer le passage des données par lui-même. Sa seule mission est ensuite de ne rien transférer, créant ainsi une sorte de puits ou trou noir dans le réseau. L'intrus se place sur un endroit stratégique de routage dans le réseau et supprime tous les messages qu'il devrait retransmettre, causant la suspension du service de routage du réseau dans les routes qui passent par le nœud intrus.

- **L'attaque d'identités multiples (Sybil attack)**

  L'attaque Sybil est définie comme un «dispositif malveillant illégitime prenant sur des identités multiples».**[WLS06]** Dans certains algorithmes, la fiabilité du routage est implémentée par l'instauration d'une redondance de chemins. Un attaquant peut altérer ce genre de systèmes en "endossant" plusieurs identités, ce qui permet de créer plusieurs routes passant par le nœud malicieux, qui ne sont en réalité qu'un seul chemin. **[YC08]**

- **L'attaque de trou de ver (Wormhole)**

  L'intrus capture un message et, en utilisant un canal de faible latence, le retransmet vers un lieu distant dans le réseau. Le canal ainsi créé fait transiter un message à un endroit du réseau auquel il ne devrait normalement pas arriver. Cette attaque a une influence notable sur le routage dans le réseau.

- **Inondation par des paquets Hello (Hello Floods)**

  Comme il est déjà mentionné, la topologie des WSNs n'est pas déterminée au préalable. Pour cela, les nœuds capteurs utilisent des paquets « Hello » pour découvrir leurs nœuds voisins et ainsi établir une topologie du réseau. Les paquets « Hello » peuvent être exploités par un attaquant pour inonder le réseau et empêcher d'autres paquets d'être échangés. De plus, si l'attaquant possède une forte puissance, il pourra envoyer des paquets «Hello» à des nœuds distants dans le réseau afin qu'ils croient que cet attaquant fait partie de leurs voisins. Par conséquent, ces nœuds peuvent choisir des routes qui contiennent ce voisin imaginaire, provoquant ainsi un envoi important des paquets à cet attaquant.

- **Espionnage des connaissances (Acknowledgment Spoofing)**

  Plusieurs protocoles de routage utilisés dans les réseaux de capteurs sans fil nécessitent l'utilisation des paquets d'acquittement (acknowledgment packets).Ces paquets sont normalement envoyés par le nœud destinataire au nœud expéditeur pour confirmer la réception d'un message donné. Un nœud malicieux peut envoyer des paquets d'acquittement à ses voisins expéditeurs afin de leur fournir de fausses informations. Par exemple, faire croire à l'expéditeur que le destinataire est encore fonctionnel tandis qu'il est hors service est une fausse information qui peut causer une perte de paquets et une perte d'énergie de la part de l'expéditeur. **[IGE00][KW03]**

- **Le référencement (Homing)**

  Dans la plupart des réseaux de capteurs, certains capteurs ont des responsabilités spéciales, par exemple être leaders des communications de groupe, d'autres sont des gestionnaires de clés de cryptage. Ces capteurs attirent la curiosité, car ils fournissent des services critiques au réseau de capteurs. Les protocoles de localisation exposent le réseau aux attaques de "homing", car un adversaire passif observe le trafic afin de connaître la présence et la localisation des ressources critiques, une fois ces capteurs spéciaux localisés, ils sont attaqués par des capteurs collaborateurs ou des adversaires mobiles.**[SB11]**

- **Rejeu, Délai et Altération de Données**

  L'intrus répète, retarde ou altère le contenu des messages en transit. Les messages peuvent contenir des données de perception prélevées et des données de configuration ou de routage. Ces types d'attaques visent entre autres à créer des boucles, attirer vers l'attaquant ou éloigner le trafic, augmenter ou diminuer le nombre de routes, générer de fausses erreurs, partitionner le réseau, et augmenter la latence de distribution des données. **[LN12]**

2.2.2. Attaque contre la vie privée

Les réseaux de capteurs sont capables de collecter automatiquement les données grâce à leurs déploiement efficace et stratégique, par ailleurs, ces réseaux sont vulnérables à l'abus potentiel de ces sources de données. Garantir la confidentialité des données sensibles dans un WSN est un défi particulièrement difficile.

La préservation de la vie privée du WSN est encore plus difficile puisque ces réseaux rendent un gros volume de données facilement disponibles grâce à des mécanismes d'accès à distance. En effet, l'attaquant ne doit pas être physiquement présent pour surveiller ou collecter les informations. En outre, l'accès à distance permet à un adversaire unique de surveiller plusieurs sites simultanément. Voici quelques attaques courantes contre la vie privée des capteurs : **[WLS06]**

- **Écoute du réseau (eavesdropping)**

  Elle permet à l'attaquant d'écouter facilement les transmissions pour récupérer le contenu des messages circulant dans le réseau.

- **Analyse de trafic**

  Grâce à une analyse efficace du trafic, un attaquant peut identifier certains nœuds de capteurs avec des rôles et des activités spéciales dans un WSN. Par exemple, une augmentation soudaine de la communication de messages entre certains nœuds signifie que ces nœuds ont des activités spécifiques et des événements à surveiller.

2.3. Objectifs et services de base de la sécurité dans les WSN

Les réseaux de capteurs partagent certaines caractéristiques des réseaux mobiles ad hoc mais aussi possèdent des propriétés spécifiques. Donc, les objectifs de sécurité englobent ceux des réseaux traditionnels et les objectifs issus des contraintes intrinsèques aux WSNs. Parmi les principaux objectifs de sécurité, nous citons :

2.3.1. Authentification de l'origine de données

C'est la propriété qui permet de vérifier que la source de données est bien l'identité prétendue. En effet, la communication entre deux nœuds dans un environnement ouvert est confrontée aux risques qu'il y ait d'autres nœuds qui cherchent à emprunter une identité des nœuds légitimes pour s'approprier leurs données. Dans ce cas, un attaquant pourra facilement se joindre au réseau et injecter des messages erronés s'il réussit à s'emparer de cette identité. Plus simplement, l'authentification est un mécanisme qui permet de séparer les amis des ennemis. **[LN07]**

Par exemple, l'utilisation de Code d'Authentification de Message(MAC), permet d'assurer à la fois l'authentification de l'origine et l'intégrité du message.

2.3.2. Intégrité de données

Ce service permet de vérifier que les données ne subissent aucune altération ou destruction volontaire ou accidentelle, et conservent un format permettant leur utilisation lors de leurs traitements, de leurs conservations ou de leurs transmissions.

2.3.3. La confidentialité

Une fois les parties authentifiées, la confidentialité reste un point important, étant donné la communication sans fil des WSNs. Elle consiste à préserver le secret des messages échangés et ne pas les révéler aux adversaires. La confidentialité peut être assurée par l'usage de la cryptographie à clé symétrique ou asymétrique.

2.3.4. La disponibilité

Elle signifie que le réseau est disponible pour assurer ses services et autoriser les parties communicantes lorsque ceci est nécessaire. Cette propriété reste difficile à assurer dans les WSN étant donné les contraintes qui pèsent sur ces réseaux, à savoir : topologie dynamique, ressources limitées des nœuds de transit, communications sans fil pouvant être facilement brouillées ou perturbées.

2.3.5. La fraîcheur

Ce service permet de garantir que les données échangées sont actuelles et ne sont pas une réinjection de précédents échanges interceptés par un attaquant.

2.4. Mécanismes de sécurité dans les WSNs

2.4.1. Outils cryptographiques

2.4.1.1. Le chiffrement

Le chiffrement est une transformation cryptographique qui assure la confidentialité en transformant le message clair en un message inintelligible (dit message chiffré), afin de cacher sa signification originale aux tierces entités non autorisées à l'utiliser ou le lire.
On distingue deux classes de primitives : symétrique et asymétrique. **[CB08**]

- **Le chiffrement symétrique (à clé secrète)**

Une même clé est utilisée entre les communicants pour chiffrer et déchiffrer les données en utilisant un algorithme de chiffrement symétrique. (voir Figure II.1).
Le principal inconvénient de ce type d'algorithmes est que la clé doit rester secrète pour toute personne autre que l'émetteur et le récepteur; elle ne doit notamment pas être captée par un espion lorsque les deux entités la communiquent entre eux lors de la création de celle-ci (il faut assurer la confidentialité même de la clé qui est censée assurer la confidentialité du message).

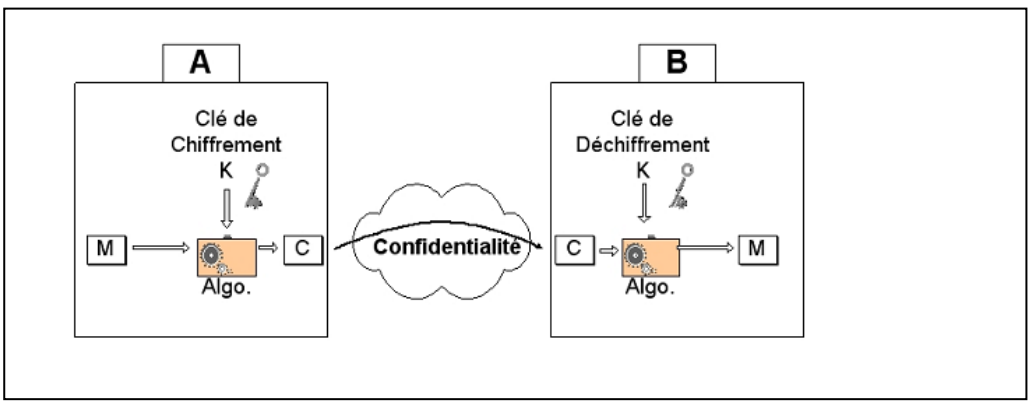


Figure II.1 : Confidentialité d'un système symétrique

- **Le chiffrement asymétrique [CB08]**

Dans un système asymétrique, comme illustré dans la Figure II.2, deux clés différentes sont générées par le récepteur, une clé publique qui est diffusée à tout le monde et une clé privée maintenue secrète chez le récepteur servant pour le déchiffrement de ces données lorsque ce dernier les reçoit.

La particularité de cette paire de clé est que la clé privée ne peut être calculée à partir de la clé publique correspondante et tout message chiffré avec la clé publique ne peut être déchiffré qu'avec la clé privée correspondante. D'où la confidentialité des messages chiffrés avec la clé publique d'un récepteur.

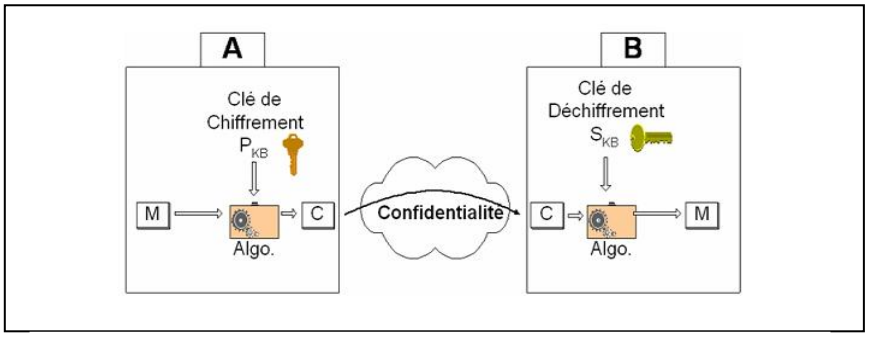


Figure II.2 : confidentialité d'un système asymétrique

2.4.1.2. Fonction de hachage

C'est le mécanisme qui assure l'intégrité de données. Cette fonction calcule une courte empreinte de taille fixe à partir d'une donnée de taille arbitraire (étape 1 dans la Figure II.3). Les propriétés de cette fonction de hachage sont :

- Impossibilité de retrouver le texte à partir de l'empreinte (fonction à sens unique)
- Impossibilité de trouver deux textes ayant la même empreinte.

Cette empreinte est recalculée par le récepteur (étape 2 de la Figure II.3) afin qu'il la compare à celle calculée par l'émetteur. Si elles sont différentes (3), alors les données ont été altérées pendant leur transmission. Les fonctions de hachage les plus courantes sont: MD5 (Message Digest 5), SHA-1 (Secure Hash Algorithm) **[CB08]**

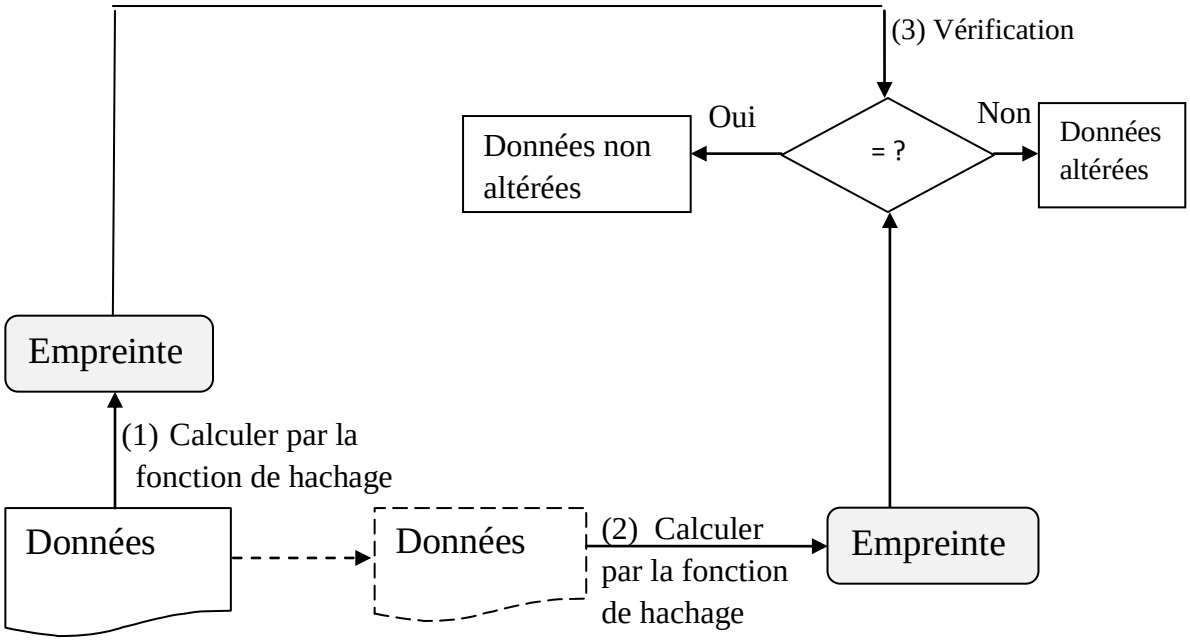


Figure II.3: La Fonction de hachage [BOK09]

2.4.1.3. La signature digitale

C'est un mécanisme cryptographique qui permet d'assurer la non répudiation de l'origine. Ce mécanisme repose sur un système cryptographique asymétrique.

La signature est calculée en utilisant la clé privée de l'émetteur et elle est vérifiée en utilisant la clé publique de l'émetteur.

Comme montré dans la Figure II.4, l'émetteur (A) signe les données à transmettre avec sa clé privée (A) en produisant une signature digitale (étape 1 de la Figure II.4). Cette dernière est par la suite envoyée avec les données (2). Si elle peut être déchiffrée avec la clé publique (A) par le récepteur (B) et si son résultat est identique aux données reçues alors la signature est valide (4), c'est-à-dire, les données proviennent bien de leur émetteur légitime qui ne pourra pas nier l'émission de ces données dans le futur.

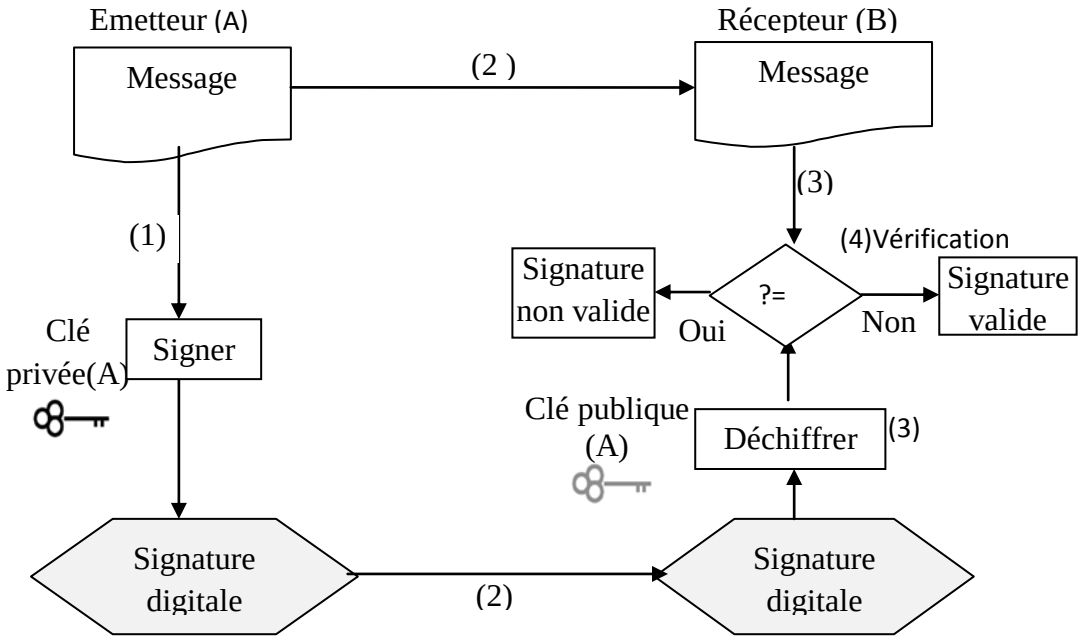


Figure II.4: La signature digitale. [BOK09]

2.4.1.4. Le code d'authentification de message MAC

Le code d'authentification de message MAC (Message Authentication Code) est un mécanisme cryptographique qui permet de vérifier l'authenticité de l'origine des données et leur intégrité en même temps.

Pour garantir d'authenticité de l'origine, l'émetteur et le récepteur doivent partager une clé symétrique qui sera utilisée pour calculer le code MAC par l'émetteur (étape 1 de la Figure II.5). Ce code est par la suite envoyé avec les données (2).

Le récepteur calcule à son tour le code MAC avec cette même clé et le compare au code qu'il a reçu (3). S'ils sont bien identiques (4), alors la source est authentique et les données n'ont pas été altérées**.** Sinon, soit le message ou l'origine n'est pas authentique. **[CB08**]

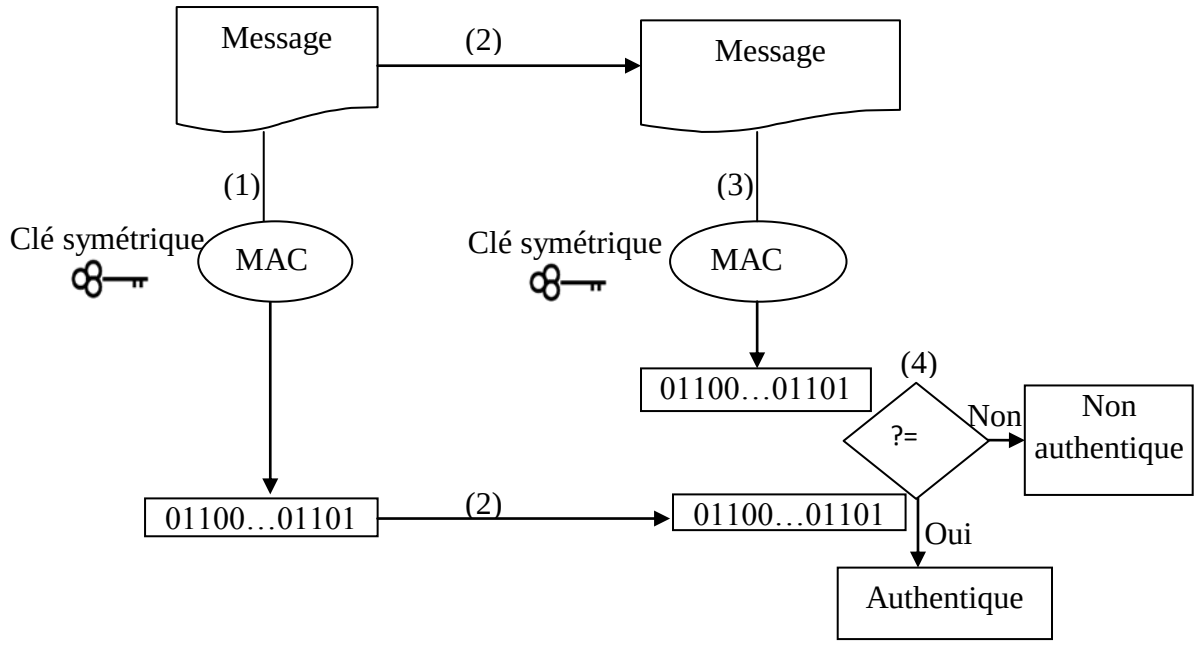


Figure II.5:authentification par MAC et système symétrique [BOK09]

2.4.2. La gestion de clés dans les WSNs

L'établissement des clés cryptographiques lors du déploiement d'un réseau de capteurs sans fil est l'une des premières nécessités pour se prémunir contre les attaques. Depuis quelques années, les chercheurs ont proposé plusieurs variétés de protocoles pour ce problème.

Dans la littérature, on trouve plusieurs schémas pour accomplir la tâche de gestion de clé dans un réseau de capteurs. Les approches disponibles actuellement tombent dans une des deux classes suivantes : approche utilisant la cryptographie symétrique ou la cryptographie asymétrique

2.4.2.1. Approches utilisant la cryptographie symétrique

- **Clé partagée par le réseau (Master Key based pre-distribution)**

Étant donné que les capteurs souffrent toujours de contraintes de ressources, le chiffrement à clé publique n'était pas encore pratique. Les techniques d'établissement des clés doivent être extensibles à des réseaux formés par des centaines et des milliers de capteurs sans fil. De plus, le protocole de communication des réseaux de capteurs sans fil est différent de celui des réseaux traditionnels. Parfois un nœud de capteurs a besoin d'établir une clé avec son nœud voisin ou avec un autre nœud du réseau. La solution la plus simple pour l'établissement des clés est une clé pour tout le réseau.

Malheureusement, le compromis d'un seul capteur du réseau est suffisant pour dévoiler la

clé secrète du réseau et ainsi le déchiffrement de tout le trafic. L'idée d'une clé partagée par tout le réseau pourra servir à établir des clés de sessions entre les nœuds voisins avant qu'elle soit effacée de la mémoire, mais ces clés de sessions ne permettront plus d'ajouter de nouveaux capteurs au réseau, ce qui cause un problème d'extensibilité de ce dernier.

- **Clé partagée par paire de nœuds (Paire-wise key pre-distribution)**

Une autre approche possible serait de préconfigurer le réseau avec une clé symétrique partagée uniquement entre chaque paire de nœuds. Par contre, dans un réseau de *n* capteurs, chaque nœud doit pouvoir sauvegarder dans sa mémoire $n-1$ clés, d'où le besoin d'établir $n \times (n-1) \div 2$clés pour tout le réseau, ce qui n'est pas pratique. Cette approche souffre aussi d'un problème d'extensibilité.

- **Participation de la station de base (Base station participation)**

Le problème principal d'employer le schéma "Paire-wise key pre distribution" est que chaque nœud dans le réseau doit stocker $n-1$ clés, ceci peut être supprimé si on emploie une station de base confiante pour envoyer les clés de session pour la communication entre deux nœuds quelconques. Dans ce cas, chaque nœud peut partager une clé avec n'importe quel autre nœud du réseau. Par contre, cette solution n'est pas appropriée aux WSNs, car elle ne permet pas le passage à l'échelle à cause du nombre de messages requis, entre la station de base et les nœuds capteurs, afin d'installer des clés symétriques entre deux nœuds communicants.

Dans la pratique on trouve le protocole SPINS *(Security Protocols for Sensor Networks)* **[AP02]**, qui assume un arbre comme topologie du réseau. La racine de l'arbre est une station de base, les nœuds de capteur forment le reste de l'arbre. SPINS comprend deux modules : le module SNEP (Secure Network Encryption Protocol) pour la confidentialité et l'authentification nœud-à-nœud et le module *µTESLA (Micro-Timed Efficient, Streaming, Loss-tolerant Authentication Protocol)* qui assure l'authentification des paquets diffusés par la station de base dans un réseau de capteurs. Une partie détaillée sur ce protocole est prévue dans le prochain chapitre.

D'autres schémas existe pour la gestion de clés, basés sur différentes approche, telles que : L'approche probabiliste (*Random Key Predestribution)***[EG02]** ou déterministe LEAP *(Localized Encryption and Authentication Protocol)* **[SS03]** .

2.4.2.2. La cryptographie à clé public dans les WSNs

La gestion de clé dans les schémas asymétriques, consiste à donner à chaque utilisateur deux clés associées, l'une secrète et l'autre rendue publique. En effet, afin de chiffrer un message à l'intention d'un utilisateur, l'idée consiste à utiliser la clé publique du destinataire alors que le déchiffrement nécessite la connaissance de la clé privée.

Ce concept naturel permet de communiquer de manière confidentielle sans avoir à partager la moindre information secrète initialement. D'autre part, les mécanismes permettant la réalisation d'une telle asymétrie se fondent sur l'utilisation d'opérations mathématiques que l'on ne sait pas inverser efficacement d'un point de vue algorithmique.

Pour l'authentification, les utilisateurs utilisent la notion de signature digitale décrite précédemment. En effet, chaque utilisateur chiffre un petit message (généralement le condensé du message utile) avec sa clé privée, dans ce cas, les autres utilisateurs peuvent utiliser la clé publique du signataire pour vérifier qu'il s'agit vraiment de la personne en question. Cependant, la clé publique utilisée pour la vérification de la signature doit au préalable être authentifiée avant son utilisation. En effet, les utilisateurs doivent généralement obtenir un certificat électronique au près d'une autorité digne de confiance (*Trusted Authority* ou *Certification Authority*), qui certifie la propriété d'une clé publique à une personne quelconque.

Dans les réseaux traditionnels comme Internet, la cryptographie asymétrique dite aussi à clé publique a montré son efficacité, et sa robustesse dans plusieurs services et protocoles de sécurité (e.g. SSL et IPsec) en fournissant des mécanismes plus sûrs et fiables pour la confidentialité, l'intégrité, l'authentification et la distribution des clés. Comparé avec la cryptographie symétrique. La cryptographie à clé publique exige un espace mémoire assez grand et de haute capacité de calcul, ce qui la rend inappropriée pour les WSNs.

Cependant, Piotrowski et al[**KP06]** ont montré qu'il est possible d'appliquer la solution à clé publique aux réseaux de capteurs en choisissant les bons algorithmes et les paramètres appropriés. Afin de quantifier le coût d'énergie des algorithmes de cryptographie à clé publique, les études **[SW05]**[**NG04]** montrent que la cryptographie à base de courbes elliptiques (ECC, *Elliptic Curve Cryptography*) **[BSS05]**a un avantage significatif par rapport au RSA, car elle réduit le temps de calcul ainsi que la quantité de données transmises et stockées. Dans **[GPW04]**, les auteurs ont démontré que la cryptographie à base de ECC peut atteindre le même niveau de sécurité en utilisant une clé plus courte que celle de RSA(*Rivest Shamir Adleman)* (une clé de 160 bits dans ECC est équivalente à une clé de 1024 bits dans RSA).

Dans la suite de ce rapport, nous introduisons la cryptographie à base de courbes elliptiques :

2.4.3. La cryptographie à base de courbes elliptiques

La cryptographie à base de courbes elliptiques (ECC) est une approche cryptographique à clé publique, basée sur la structure algébrique des courbes elliptiques sur des corps finis.

Son utilisation dans la cryptographie a été suggérée par Victor S. Miller et Neal Koblitz en 1985 (respectivement dans **[VM85]** et **[NK87]).**

2.4.3.1. ECC une base mathématique

Une courbe elliptique *E* sur le corps fini F_p d'ordre *p*, avec *p* premier est définie par une équation de Weierstrass du type :**[WST07]**

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 pour\ a_i \in F_p (1)$$

La courbe elliptique $E(F_p)$ est l'ensemble des points $P = (x, y) \in (F_p)^2$ vérifiant l'équation (1) plus un point à l'infini notéP_∞.

$$E(F_p) = \{(x, y) \in F_p \times F_p\ tels\ que : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{P_\infty\}$$

Pour toute courbe elliptique $E(F_p)$, il existe un changement de variables tel que $E(F_p)$ admettant une équation de Weierstrass réduite $y^2 = x^3 + ax + b$.
La figure suivante (Figure II.6) montre l'addition de deux points et le doublement d'un point sur les courbes elliptiques.

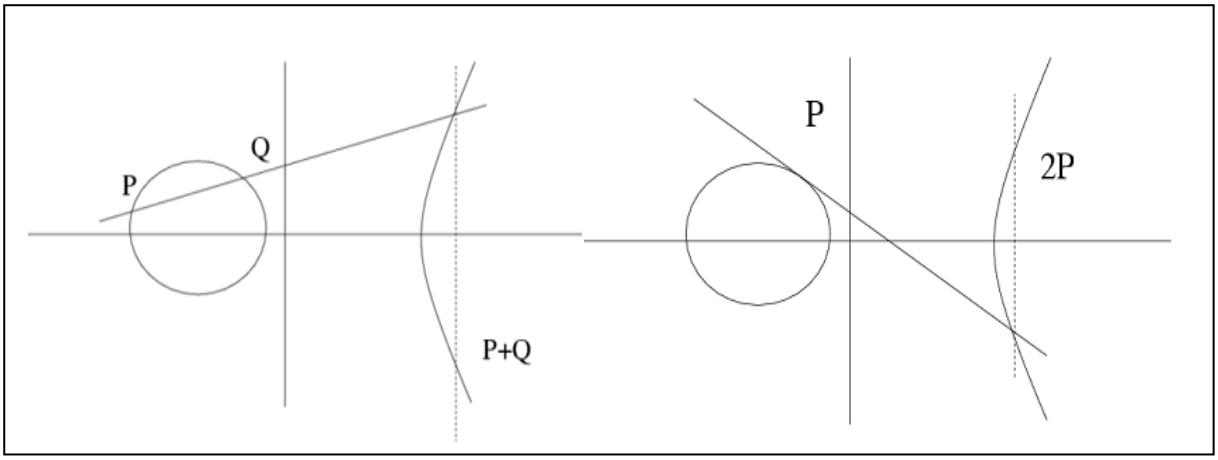


Figure II.6 Addition de deux points et doublement d'un point [SG05]

2.4.3.2. Loi de groupe

L'opération du groupe notée +, permet d'additionner deux points de la courbe et qui donne un troisième point qui appartient au groupe. La loi de groupe peut être interprétée géométriquement grâce à la méthode dite corde et tangente présentée dans la figure II.6. L'ensemble des points d'une courbe elliptique E muni de la loi + vérifie les propriétés suivantes :

- la loi + est interne :$\forall P, Q \in E, P + Q \in E$;
- existence d'un élément neutre P_∞: $\forall P \in E, P + P_\infty = P$;
- la commutativité: $\forall P, Q \in E, P + Q = Q + P$;

- l'associativité : $\forall P, Q, R \in E, (P + Q) + R = P + (Q + R)$;
- la symétrie : $\forall P \in E$, il existe un point noté $-P$ tel que $P + (-P) = P_\infty$;

2.4.3.3. Problème du logarithme discret

Soient G un groupe (noté additivement) et P un élément de G d'ordre fini. Soit $H =< P >$le sous-groupe engendré par P, alors

$$\forall Q \in H, \exists n \in \mathbb{N}: Q = nP;$$

n est appelé le logarithme discret de Q par rapport à P.

Le problème du logarithme discret dans un groupe consiste donc à retrouver l'entier n à partir de la donnée publique (H, P, Q). La sécurité des protocoles basés sur les courbes elliptiques repose sur la résolution de ce problème. **[NM07]**

Exemple : Échange de clés de Diffie-Hellman

Supposons que deux capteurs A et B veuillent partager un secret commun mais ces derniers ne peuvent pas communiquer par un canal sécurisé. L'échange de clés de Diffie Hellman se déroule de la manière suivante : **[NM07]**

1. A et B choisissent publiquement un groupe G et un point P ∈ G,
2. A choisit secrètement un entier$k_A$et calcule $(k_A * P)$,
3. B choisit secrètement un entier $k_B$et calcule $(k_B * P)$,
4. A et B échangent publiquement les données $(k_A * P$)et $(k_B * P)$,
5. A peut calculer $[k_A * (k_B) * P]$,
6. B peut calculer$[k_B * (k_A) * P]$,
7. A et B possèdent tous deux la quantité $(k_B * k_A * P)$, qui sera leur secret commun.

Un attaquant éventuel n'aura quant à lui en sa possession que les données $G, P, k_A * P, K_B * P$ pour retrouver $k_A * K_B * P$.C'est ce que l'on appelle le problème de Diffie Hellman calculatoire (PDHC). Il est évident que le PDHC est réductible au Problème de Logarithme Discret (PLD), dans le sens où savoir résoudre le PLD permet la résolution immédiate du PDHC. La réduction réciproque a été prouvée dans plusieurs cas et on considère en général ces deux problèmes comme équivalents. De ce point de vue, la sécurité de l'échange de Diffie-Hellman repose bien sur la difficulté du PLD.

2.4.3.4. Multiplication d'un point de la courbe par un scalaire

Nous pouvons définir, pour tout entier $k \in \mathbb{N}$, le morphisme de multiplication scalaire par k :

$$[k]: E \rightarrow E$$
$$P \rightarrow [k]P = \underbrace{P + P + \cdots + P}_{\text{k fois}}$$

Le temps nécessaire pour réaliser la multiplication d'un point sur la courbe par un scalaire est trop long. Il existe de nombreux algorithmes permettant d'effectuer une multiplication de

point par un scalaire. Ils découlent en général directement de la façon dont on représente le scalaire.

L'algorithme qui suit est un algorithme classique basé sur la représentation binaire du scalaire. C'est principalement une série de doublements entrecoupée d'additions (dépendantes du nombre de 1 dans la représentation du scalaire).

Algorithme 3.1 Doublement et addition

Données : $P \in E(K), k = (k_{l-1} \ldots k_o)_2 \in \mathbb{N}$. Avec l le nombre de bits dans P.

Résultat : $[k]P \in E(K)$.

Début

$Q \leftarrow P$
  Pour $i = l - 2$ à 0 faire
    $Q \leftarrow [2]Q$
      Si $k_i = 1$ alors
      $Q \leftarrow Q + P$
      Fin
    Fin
Fin
Retourner Q

2.4.4. La signature digitale basée sur les courbes elliptiques (ECDSA)

L'algorithme de signature numérique basé sur les courbes elliptiques (ECDSA) est un algorithme analogue à celui de la signature numérique (DSA). Il s'agit de l'algorithme standardisé des schémas de signatures basés sur les courbes elliptiques. **[SG05]**

Nous allons présenter les deux algorithmes essentiels pour l'utilisation de ce système. Le premier algorithme est un algorithme de génération de signature pour un message donné (Figure II.7) et le second est celui qui va servir de vérificateur de cette signature (Figure II.8):

Soit G un groupe cyclique de la courbe elliptique $E(F_p), P$ un générateur de ce groupe d'ordre n premier. Soit $H : \{0,1\} \rightarrow \mathbb{Z}_n^*$ une fonction de hachage à sens unique.

Un émetteur A choisit aléatoirement un entier $d \in [1, n-1]$ et publie sa clé publique $Q = d$P. Le paramètre d est gardé privé pour A.

L'émetteur A utilise sa clé privée d pour générer la signature (r, s) d'un message $m \in \{0,1\}^*$: **[SG05]**

1. Tire aléatoirement un entier k, $1 \leq k \leq n-1$.
2. Calcule $kP = (x_1, y_1)$ et convertit x_1 en un entier x_1.
3. Calcule $r = x_1 \, mod \, n$. Si $r = 0$ alors revenir à l'étape 1.
4. Calcule $k^{-1} mod \, n$.

5. Calcule $H(m)$ et convertit la chaîne de bits résultante en un entier e.
6. Calcule $s = k^{-1}(e + dr)\ mod\ n$. Si $s = 0$ alors retourner à l'étape 1.
7. La signature du message m est donc le couple (r, s).

Pour la vérification de la signature ECDSA d'un message m, le récepteur du message suit les étapes suivantes **[SG05]**:

1. Vérifie que r et s sont deux entiers dans l'intervalle $[1, n-1]$.
2. Calcule H (m) et convertit la chaîne de bits résultante en un entier e.
3. Calcule $w = s^{-1} mod\ n$.
4. Calcule $u_1 = ew\ mod\ n$ et $u_2 = rw\ mod\ n$.
5. Calcule $X = u_1 P + u_2 Q$.
6. Si $X = O$, alors rejeter la signature. Sinon, convertir la coordonnées x_1 de X en un entier et calculer $v = x_1\ mod\ n$.
7. Accepter la signature si et seulement si : $v = r$.

2.4.5. Le couplage bilinéaire

Les couplages sont une notion mathématique qui apparaît dans les années 90 en cryptographie dans un but crypte-analytique. La notion de couplage consiste à construire un lien entre deux groupes bien définis permettant ainsi la conception d'un nouveau schéma cryptographique, dont la sécurité est basée sur la réduction d'un problème dans un premier groupe en un problème généralement plus facile dans un autre groupe.

Les premiers couplages introduits en cryptographie sont les couplages de Weil et de Tate, qui utilisent des groupes sur des courbes elliptiques, tandis que le schéma de chiffrement le plus populaire basé sur le couplage est le « schéma cryptographique basé sur l'identité », proposé par Boneh et Franklin en 2001 **[BF01]**.

2.4.6. Généralités sur les couplages

Soient G_1, G_2et G_3 trois groupes abéliens de même ordre r. Les groupes G_1 et G_2 sont additifs, G_3 est un groupe multiplicatif. Un couplage est une application bilinéaire et non dégénérée notée e du produit cartésien de G_1 et G_2à valeur dans G_3:

$$e : G_1 \times G_2 \rightarrow G_3$$

Un couplage symétrique est un couplage pour lequel$G_1 = G_2$.

Le couplage a les propriétés suivantes **[POS08]**:

1- Bilinéarité :

$$\forall\ P \in G_1, \forall\ Q \in G_2\ et\ \forall\ a, b \in \mathbb{Z}^*, on\ a :$$

$$e\ (aP, bQ) = e(P, bQ)^a = e(aP, Q)^b = e\ (P, Q)^{ab}.$$

2- **Non dégénéré**: $e(P, P) \neq 1$.
3- **Calculable :** Il existe des algorithmes efficaces pour calculer $e(P, Q)$ pour chaque $P, Q \in G_1$.

2.4.7. Cryptographie à base de couplage (Pairing-based cryptography PBC)

Les couplages ont permis la création et la réalisation de protocoles cryptographiques originaux. En 2000, A. Joux **[AJ00]** propose un schéma d'échange de clé tripartite à la Diffie-Hellman utilisant la propriété de bilinéarité des couplages.

Une utilisation intéressante des couplages est la cryptographie à base d'identité. En 2001 dans **[BF01]**, D. Boneh et M. Franklin proposent une solution basée sur les couplages au challenge proposé par Shamir en 1984. Ce challenge consistait à créer un protocole cryptographique basé sur le principe où la clé publique d'un utilisateur est son identité. La clé privée de l'utilisateur lui est fournie par une autorité de confiance. Cette dernière construit les clés privées à l'aide des clés publiques des utilisateurs et les leur transmet par un canal sécurisé. Ce premier schéma cryptographique basé sur l'identité fut réalisable principalement grâce à la propriété de bilinéarité des couplages.

Nous commençons d'abord par décrire le protocole d'échange de clé tripartite de Diffie-Hellman bilinéaire.

2.4.8. Problème de Diffie-Hellman bilinéaire BDHP

De nouvelles applications ont vu le jour, tel que la cryptographie à base d'identité, dont la sécurité est basée sur des problèmes de Diffie-Hellman bilinéaires. La définition de la méthode de Diffie-Hellman bilinéaire présentées ci-après a été formellement présentée dans **[BF01]**.

Nous considérons G_1 un groupe additif d'ordre principal q, et P un générateur de ce groupe. Nous considérons également un groupe multiplicatif G_2 et un couplage bilinéaire $e: G_1 \times G_1 \rightarrow G_2$. Ayant (P, aP, bP)pour $a, b, \in Z_q^*$, le problème de Diffie-Hellman bilinéaire consiste à calculer : $W = e(P, P)^{ab} \in G_2$. **[AS05]**

2.5. La cryptographie à base d'identité

Quand on utilise la cryptographie à clé publique, on doit être certain que cette clé appartient bien à l'interlocuteur et n'est une clé d'un attaquant.

La plupart des méthodes cryptographiques à clés publics utilisent des services d'autorité de certification (CA) qui est une autorité de confiance et qui est responsable de produire des certificats pour des clés publiques. Un tel certificat contenait une information sur l'identité du propriétaire de la clé publique ainsi qu'une signature du CA.

En 1984 Shamir a introduit la notion de la cryptographie basée sur l'identité, qui a atténué le problème de certificat, mais ce n'est qu'à l'année 2001 que Boneh et Franklin ont proposé le premier protocole praticable basé sur l'identité, ce protocole utilise la propriété de bilinéarité des couplages **[BF01].**

- **Description du protocole**

En 2001, Boneh et Franklin ont proposé le premier schéma cryptographique basé sur l'identité. Leur méthode utilise un couplage bilinéaire $e(G_1, G_T)$ pour lequel le BDHP est incassable (où G_1 un groupe additif et G_T un groupe multiplicatif). Ils utilisent aussi deux fonctions de hachage $H_1 : \{0,1\}^* \rightarrow G_1$et $H_2 : G_T \rightarrow \{0,1\}^l$, où l représente la longueur du message en bits. **[AME05]**

La clé de l'autorité de confiance(CA) est $T = tP$. Chaque partie du réseau pourra avoir une copie de la clé publique T.Quand une entité A veut avoir sa clé secrèted_A, elle sera calculée par l'autorité de confiance $d_A = tH_1(ID_A)$ et envoyée d'une manière sécurisée.

Pour chiffrer un message $m \in \{0,1\}^l$et l'envoyer à l'entité A, l'entité B calcule la clé publique$Q_A = H_1(ID_A)$, sélectionne une variable aléatoire $r \in [1, n-1]$, et calcule $R = rP$ et $c = m \oplus H_2(e(Q_A, T)^r)$. Elle transmet par la suite (R, c) à l'entité A. Pour déchiffrer le message, cette dernière utilise sa clé privée d_A afin de calculer $m = c \oplus H_2(e(d_A, R))$. La propriété de bilinéarité des couplages a pour conséquence:

$$e(d_A, R) = e(tQ_A, rP) = e(Q_A, tP)^r = e(Q_A, T)^r .$$

Un attaquant désire calculer m à partir de(R, c), il doit calculer $e(Q_A, T)^r$en ayant (P, Q_A, T, R), qui est une instance de BDHP. **[SKS09]**

2.6. Application de la cryptographie basée sur l'identité pour les WSNs

La procédure de communication entre les nœuds de capteurs consomme beaucoup d'énergie, ceci signifie que l'échange de clés dans les réseaux de capteurs doit également maintenir le nombre de messages échangés le plus réduit. L'arrivée du schéma cryptographique PBC a permis d'accomplir ces contraintes. Cette méthode fournit des moyens non-interactifs et distribués des clés entre deux nœuds du réseau, même s'ils ont été déployés à différents moments.**[ODL07]**

Considérons que chaque nœud possède un identifiant unique, et un secret unique non partagé avec une autre entité. Deux parties A et B, chacune ne connaissant que l'ID de l'autre et sans communication, sont alors en mesure de tirer un secret mutuel inconnu à toute autre partie, qui sert pour obtenir une clé de chiffrement afin de sécuriser leurs communications.

L'autorité de confiance dans le cas des WSNs est la station de base qui assigne des identités et des clés secrètes à chaque nœud. Pour démarrer un système cryptographique basé sur l'identité, la station de base a besoin d'abord de produire et de distribuer des clés privées et des paramètres publics. De manière générale, cette procédure peut être accomplie comme suit : Premièrement, la station de base génère une clé secrète maîtresse *s*, puis calcule la clé privée de chaque nœud. Pour ce faire, elle lie les identités de chaque nœud à un point sur la courbe elliptique, via une fonction de hachage et de mappage; ainsi pour un nœud$X$, la clé publique$P_X = H(ID_X)$. La station de base calcule par la suite la clé privée du nœud $\boldsymbol{S_X} = [\boldsymbol{s}]\boldsymbol{P_X}$, et pré-charge chaque nœud X avec les informations suivantes:

- L'ID du nœudID_X,
- La clé privée du nœudS_X.

Chaque nœud est également équipé de la fonction H de sorte qu'il peut prendre n'importe quelle identité (ID_y, par exemple) en entrée et calcule la clé publique correspondante (P_Y). Il est a noté que, à part la station de base, seul le nœud X connaît la clé privéeS_X.

Nous supposons que deux nœuds A et B dont chacun connait l'ID de l'autre, ayant les clés privées $S_A = [s]P_A$ et$S_B = [s]P_B$, respectivement. En conséquence, par bilinéarité, nous avons :

$$e(S_A, P_B) = e([s]P_A\ , P_B) = e(P_A, P_B)^S = e(P_A, [s]P_B) = e(P_A, S_B)$$

$$= e(S_B, P_A)$$

Notons que A connaitS_A et pourra calculer $P_B = H(ID_B)$ et B connaitS_B et pourra calculer $P_A = H(ID_A)$. Par la suite, les deux nœuds peuvent calculer la clé secrète : $k_{A,B} = e(S_A, P_B) = e(S_B, P_A)$.Ce calcul n'exige aucune communication autre que l'identité de l'autre nœud doit être connue.

Cependant, la méthode de cryptographie à base d'identité a des exigences fortes telles que l'existence d'une entité de confiance, qui est chargée de délivrer les clés privées des utilisateurs. Les WSNs possèdent intrinsèquement une telle entité, qui est la station de base. Une autre exigence de cette approche est que les clés doivent être livrées via des canaux confidentiels et authentiques pour les utilisateurs. Cependant, dans la plupart des applications WSNs, les clés privées peuvent être distribuées hors ligne (ie. elles peuvent être générées et pré-chargées dans les nœuds avant le déploiement).

Nous observons que, en raison de la nature non interactive de la communication, les nœuds peuvent s'entendre sur les clés, même s'ils ne sont pas en ligne simultanément. Ceci est particulièrement utile dans les WSNs, où les nœuds pourraient suivre les habitudes de

sommeil, et deviennent souvent temporairement indisponibles en raison des obstacles physiques ou des dysfonctionnements.

Enfin, les auteurs ont supposé que les nœuds connaissent au préalable les identifiants les uns des autres, une hypothèse raisonnable puisque dans les réseaux de capteurs, les nœuds ont déjà besoin de connaître les identités de leurs voisins pour échanger des informations ordinaires. **[ODL07]**

2.7. Conclusion

Dans ce chapitre, nous avons procédé à l'étude de l'aspect sécurité dans les réseaux de capteurs. Nous avons posé les briques de base et fédéré quelques concepts généraux de sécurité nécessaires à la compréhension de nos problématiques dans la suite de ce manuscrit.

Nous avons présenté par la suite le concept des courbes elliptiques (ECC) et leurs apports dans la cryptographie. Nous avons constaté que les ECC représentent un moyen efficace de sécurité et ils sont adaptables aux dispositifs présentant des ressources limitées en termes de calcul, mémoire et énergie, tels que les capteurs.

Dans les réseaux de capteurs, la diffusion de données est un mode de communication efficace, où les utilisateurs peuvent diffuser des messages afin obtenir les informations désirées. Nous nous sommes intéressés dans notre étude à l'authentification de la diffusion dans les réseaux de capteurs, en réduisant au maximum le coût de calcul et de communication et en minimisant la consommation de l'énergie.

Dans le chapitre suivant, nous présentons quelques méthodes d'authentification de la diffusion proposées dans le domaine des réseaux de capteurs, ainsi que les avantages et les inconvénients de chacune d'elles.

Chapitre III : Authentification de la diffusion dans les WSNs

3.1. Introduction

Dans les réseaux de capteurs, la diffusion est un paradigme de communication efficace et commune, dans laquelle une multitude d'utilisateurs se joignent au réseau et diffusent des messages (requêtes ou commandes) pour obtenir les informations désirées.

Malheureusement, en raison de la nature des communications sans fil dans les réseaux de capteurs, les attaquants peuvent facilement intercepter le trafic, usurper l'identité des autres utilisateurs, injecter de fausses données ou modifier le contenu des messages légitimes lors de la communication multi-sauts. Ainsi, des mécanismes d'authentification doivent être mises en œuvre dans les réseaux de capteurs pour protéger les messages diffusés à partir de divers attaques malveillantes. Selon les primitives cryptographiques utilisées, plusieurs solutions ont été proposées dans la littérature pour traiter l'authentification de la diffusion dans les réseaux de capteurs.

Dans ce qui suit, nous nous intéressons à étudier quelques méthodes symétriques et asymétriques d'authentification de diffusion dans les réseaux de capteurs.

3.2. Authentification de la diffusion à base de cryptographie symétrique

3.2.1. µTESLA Broadcast Authentication Protocol (Timed Efficient Stream Loss-tolerant Authentication)

Proposer par Perrig**[AP02],** le protocole µTESLA est basé sur une chaîne de clés d'authentification liés entre eux par une fonction pseudo-aléatoire **[GGM86]**. Chaque clé dans la chaîne est l'image de la clé suivante dans la fonction pseudo-aléatoire.

µTESLA réalise l'authentification de diffusion par la divulgation tardive des clés d'authentification dans la chaine de clés. L'efficacité de µTESLA est basée sur le fait que seulement une fonction pseudo-aléatoire et des opérations de chiffrement à base de clé secrète sont nécessaires pour l'authentification d'un message de diffusion.

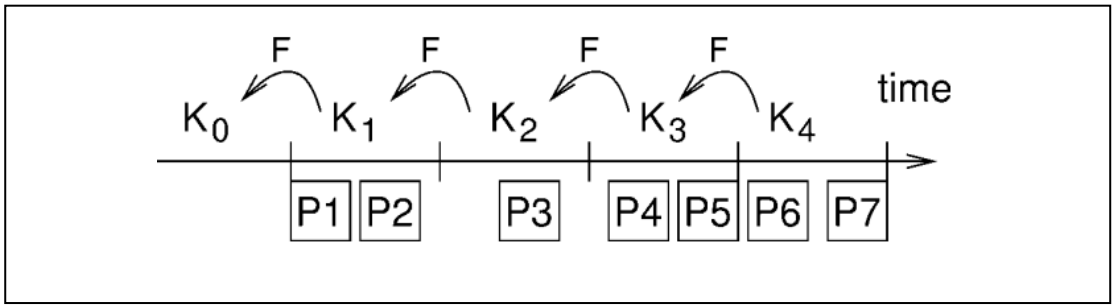


Figure III.1: Gestion de la chaîne de clés à sens unique µTESLA[AP02]

- **Procédure de l'émetteur**

Comme illustrer dans la Figure III.1, l'émetteur génère d'abord une séquence de clés secrètes de longueur n (une chaine de clés à sens unique). Pour générer cette chaine, l'émetteur choisit aléatoirement une valeur K_n. Il génère les valeurs restantes de la chaine, en appliquant successivement une fonction de hachage à sens unique F de la façon suivante : $F(K_n) = K_{n-1}, F(K_{n-1}) = K_{n-2} \dots, F(K_i) = K_{i-1} \dots, F(K_1) = K_0$. Puisque F est à sens unique, ayant K_{j+1}, n'importe qui peut calculer les valeurs K_1, jusqu'à K_j, mais personne ne peut calculer K_{j+1} si on connait une des clés précédentes $(K_1, .., K_j)$. **[AP02]**

- **Diffusion des paquets authentiques**

La source de données divise le temps en N intervalles de temps de taille identique, l'émetteur associe à chaque intervalle de temps i une clé de la chaîne à sens unique crée précédemment. Dans l'intervalle de temps i, l'émetteur utilise la clé de l'intervalle courant K_i afin de calculer le code d'authentification (MAC). Après δ intervalles (i.e. l'intervalle $i + \delta$), l'émetteur doit révéler la clé K_i. Le temps de révélation de clé δ est de l'ordre de quelques intervalles de temps, il doit être plus grand qu'un temps d'aller-retour entre l'émetteur et les récepteurs.

- **Initialisation d'un nouveau récepteur**

Dans une chaîne de clés à sens unique, les clés sont auto-authentifiées. Le récepteur peut facilement et efficacement authentifier les clés suivantes de la chaîne de clés, en utilisant une des clés précédente. Par exemple, si un récepteur a une valeur K_i authentique de la chaîne de clés, il peut facilement authentifier la clé K_{i+1}, en vérifiant $K_i = F(K_{i+1})$.

Pour initialiser µTESLA, chaque récepteur a besoin d'avoir une clé authentique de la chaîne de clés. D'autre part, les récepteurs doivent synchroniser leurs horloges avec celle de la source des données et recevoir d'autres informations nécessaires pour le procédé µTESLA. Pour cela, les nœuds capteurs opèrent le mécanisme décrit dans la figure III.2.

Le récepteur M (*Mote*) envoie à l'émetteur S(Sink) une valeur N_M choisit aléatoirement dans un message (m_1). De son côté, l'émetteur S envoie un message (m_2) contenant la valeur courante du temps T_S, une clé authentique de la chaîne à sens unique K_i (initialement c'est la clé K_0), qui correspond à l'intervalle i précédent, les valeurs « *key disclosure scheduler* » qui correspondent aux temps T_i de début de l'intervalle i (initialement c'est T_0), la durée T_{int} de chaque intervalle de temps et le délai δ de révélation de clé. La valeur du MAC est générée à partir de la clé secrète partagée par le nœud et la station de base (K_{MS}) pour authentifier les données, la valeur aléatoire N_M permet au nœud de vérifier que le message reçu correspond à son message qu'il a envoyé précédemment (m_1). La figure III.2, illustre le mécanisme d'initialisation des nœuds récepteurs.

$M \rightarrow S$: N_M

$S \rightarrow M$: $T_S \mid K_i \mid T_i \mid T_{\text{int}} \mid \delta$

$\text{MAC}(K_{MS}, N_M \mid T_S \mid K_i \mid T_i \mid T_{\text{int}} \mid \delta)$.

Figure.III.2 structure du message envoyé (µTESLA)

- **Vérification de l'authentification des paquets de diffusion**

Lorsqu'un récepteur reçoit le paquet avec son MAC, il doit s'assurer que le paquet n'a pas été intercepté par un adversaire. Le récepteur a besoin d'être sûr que l'émetteur n'a pas encore divulgué cette clé qui a été utilisée pour calculer le MAC du paquet entrant. Donc, le récepteur calcule le plus grand intervalle x dans lequel pourrait être l'émetteur : $x = (t_j - T_0)/T_{int}$ où

- t_j : est le temps enregistré par le récepteur lors de réception du paquet.
- T_0 : le temps initial.
- T_{int} : la durée de chaque intervalle.

Le récepteur vérifie par la suite, si $x < i + \delta$, sinon, il rejette le paquet. Si la condition de sécurité est vérifiée alors l'expéditeur met ce paquet reçu dans un *buffer* en attendant que la clé K_i soit révélée. Il met le paquet dans un *buffer* pour δ intervalles.

Après δ intervalles, la nouvelle clé K_i est révélée dans un paquet envoyé pendant l'intervalle $i + \delta$. Il authentifie cette clé en vérifiant à partir de la dernière clé reçue K_v, si $K_v = F^{i-v}(K_i)$, si c'est le cas, alors le récepteur peut authentifier tous les messages reçus dans l'intervalle de temps de v à i. Il remplace la valeur enregistrée K_v par la nouvelle valeur K_i.

- **Diffusion des données authentifiées à partir des nœuds de capteur**

De nouveaux défis apparaissent lorsqu'un nœud diffuse des données authentiques. Puisque le nœud de capteur a une mémoire limitée, il ne peut pas stocker les clés de la chaîne utilisée. Par ailleurs, recalculant chaque clé à partir de la clé initiale K_n est coûteux. En outre, le nœud ne peut pas partager une clé avec chaque récepteur (mémoire limitée). D'autre part, l'initialisation authentique de nœud (*Bootstrapping of new receivers*), impliquerait un coût très important (authentification point-à-point). Enfin, la diffusion de clés communiquées à tous les récepteurs est coûteuse pour le nœud et consomme l'énergie de la batterie **[AP02]**. Deux approches ont été proposées pour faire face contre ce problème :

- Dans la première solution, le nœud diffuse les données à travers la station de base. Il utilise le protocole SNEP **[AP02]** pour envoyer les données de manière authentifiée à la station de base, cette dernière se charge par la suite de diffuser les données.

- Dans la seconde solution, le nœud de capteur diffuse les données. Mais, la station de base conserve la chaîne de clés à sens unique et envoie les clés vers le nœud de diffusion au besoin. Afin de conserver l'énergie pour le nœud qui diffuse les données, la station de base peut également diffuser les clés divulguées, et/ou effectuer des procédures d'initialisation pour les nouveaux récepteurs.

- **Discussion de la méthode µTESLA**

µTESLA présente des avantages significatifs tels que le passage à l'échelle et la tolérance aux pertes de paquets et de clés. En effet si le paquet contenant la clé K_i révélé est perdu, on peut toujours authentifier les paquets dont leurs MAC sont calculés avec la clé K_i et cela à partir de n'importe quelle autre clé K_{i+u} reçue ultérieurement, car on peut déduire K_i de K_{i+u}en appliquant u fois la fonction F à sens unique :$F^u(K_{i+u}) = K_i$.

Par ailleurs, le protocole µTESLA a les inconvénients suivants :

- L'obstacle majeur de l'utilisation de µTESLA dans les réseaux de capteurs à grande échelle, réside dans la difficulté de répartir les clés de la chaîne de clés dans un réseau à grande échelle. Cela conduit à un volume élevé de paquets dans les WSNs à grande densité, pour initialiser un groupe avec un nombre important de nouveaux récepteurs. La raison essentielle de cette difficulté est la différence entre la répartition unicast des clés et l'authentification de la diffusion des messages. Autrement dit, la technique est développée pour l'authentification de la diffusion, mais elle repose sur une technique unicast pour distribuer les paramètres initiaux. **[EG09]**
- Chaque récepteur doit être synchronisé avec la source individuellement.
- Un attaquant peut inonder l'ensemble du réseau avec des messages qui doivent être sauvegardés jusqu'au prochain intervalle. Et puisque la transmission sans fil et l'énergie sont des contraintes cruciales dans les WSNs, le flooding pourra causer une attaque de type Denis de Service.

3.2.2. µTESLA avec chaine de clé initiale prédéterminée

Afin d'améliorer et de surmonter les limitations du schéma µTESLA classique. Les auteurs D. Liu et P. Ning dans **[LN04]** ont proposé l'approche PKCC (*Predetermined Key Chain Commitment*) dans laquelle les nœuds de capteurs sont pré-chargés par les paramètres d'initialisation du protocole µTESLA. Ceci a comme objectif d'éliminer la phase de distribution unicast des paramètres µTESLA. Cette méthode étant simple permet de réduire le coût de communication nécessaire pour l'initialisation de nouveaux récepteurs.

- **Discussion du protocole**

Ce schéma permet effectivement d'éliminer le problème de distribution unicast des paramètres µTESLA (initialisation de nouveaux récepteurs) rencontré dans le schéma original. En particulier lorsque le réseau WSN est à large échelle. Cependant, la chaîne prédéterminée va certainement couvrir une période fixe de la durée de vie du réseau. Dans ce cas, couvrir une plus grande période nécessite : soit d'utiliser une plus longue chaîne de clé, soit d'utiliser de larges intervalles de temps. Dans les deux cas, un problème se présente : pour la première où on doit utiliser une longue chaîne de clés, la station de base devrait réserver un espace mémoire important pour le stockage de ces clés. Alors que pour la seconde proposition, le délai attendu avant la divulgation d'une clé d'authentification devient grand, ce qui va surcharger le buffer des nœuds avec un grand nombre de messages, d'autre part, une fois que la clé en question soit divulguée, un coût de calcul très important sera nécessaire pour authentifier les messages présents dans le buffer. En plus, ayant un délai d'authentification important, rend le système vulnérable aux attaques Dos (*Buffer Flooding*).

D'autres approches ont étaient proposé dans **[LN04]**, pour l'amélioration du schéma µTESLA, on peut citer l'approche µTESLA à deux niveaux, µTESLA multi-niveaux et le µTESLA avec arbre de Meckel. Leur objectif principal, est le passage à échelle en termes de nombre d'émetteur. Par ailleurs, l'approche µTESLA est une méthode intéressante et peut être appliquée à certains domaines d'application des WSNs (principalement celles qui ne nécessite pas une authentification immédiate des messages diffusés). Cependant, elle présente plusieurs inconvénients tels que :

- Vulnérable aux attaques de DoS.
- Difficulté quant au passage à grande échelle en termes de nombre d'émetteurs. En effet, malgré les variantes proposées (µTESLA avec arbre Merkel **[LN04]**), les schémas tendent à se compliquer en termes d'implémentation.

Cependant, le domaine de cryptographie à clé publique devient de plus en plus praticable dans les réseaux de capteurs. De nombreuses recherches ont été élaborées dans ce domaine. Dans la suite de ce rapport, nous étudierons quelques méthodes d'authentification de la diffusion à base de clé publique, afin de surmonter les manques des méthodes cryptographiques symétriques.

3.3. Méthodes cryptographiques à clé publique pour l'authentification de la diffusion

3.3.1. Authentification basée sur le certificat

Dans cette méthode **[RZL07]**, chaque utilisateur du WSN est équipé d'une paire de clés publique et privée (PK/SK), et signe chaque message qu'il diffuse avec sa clé secrète en

utilisant un schéma de signature numérique tel que RSA ou DSA. Pour prouver la propriété de l'utilisateur sur sa clé publique, la station de base est également équipée d'une paire clé publique et privée et sert en tant qu'autorité de certification (CA). La station de base génère un certificat de clé publique qui est composé de :

$$Cert_{UID} = U_{ID}, \mathbf{PK}_{UID}, ExpT, \mathbf{SIG}_{SKSink} \{ h(U_{ID}||ExpT || PK_{UID})\}$$

Où U_{ID} indique l'identité de l'utilisateur, PK_{UID} indique sa clé publique, ExpT désigne le temps d'expiration du certificat et SIG_{SKSink} {h (U_{ID} || ExpT || PK_{UID})} est une signature signée sur h(U_{ID}||ExpT||PK_{UID}) avec SK_{Sink}. Ainsi, un message de diffusion est maintenant de la forme suivante :

$$< M, tt, SIG_{SKUID} \{ h(U_{ID}||tt||M)\} , Cert_{UID} >$$

Où M représente le message diffusé, tt désigne l'heure actuelle et U_{ID} l'identificateur de l'utilisateur.

A la réception d'un message signé, les nœuds capteurs, sont amenés à vérifier l'authenticité du message en suivant deux étapes : La première consiste à vérifier l'authenticité du certificat, en utilisant la clé publique de la station de base, pré-chargée avant le déploiement. Ensuite, ils opèrent la vérification de la signature du message en utilisant la clé publique authentique de l'utilisateur qui est contenue dans le certificat

- **Discussion**

Ce schéma simple souffre de plusieurs problèmes. Le plus important, est qui est impossible de maintenir la révocation des utilisateurs dans ce schéma. Afin de permettre la révocation des utilisateurs et par conséquent la révocation de certificats, les nœuds de capteurs ont à recevoir et à stocker une liste de révocation de certificats (CRL). En effet, la CRL a besoin d'un espace de stockage linéaire pour le nombre total de certificats révoqués à chaque nœud capteur.

Toutefois, cela est pratiquement impossible en raison de limitation de stockage dans les nœuds de capteurs. Par exemple, supposant que la clé publique est de longueur de 20 octets, la liste CRL ne contenant que 1000 certificats révoqués est au moins de la taille de 19,5 Ko, même dans le format le plus simple (c'est à dire, contenant uniquement la clé publique).

En parallèle, le recours à la station de base pour la vérification de la liste CRL est évidemment impossible, car cela pourrait introduire des coûts de communication très élevés. En outre, afin d'authentifier un message, il faut toujours deux opérations de vérification de signature, c'est parce que le certificat devrait toujours être authentifié en premier lieu.

3.3.2. Authentification basée sur l'arbre de Merkle

Ayant observé le problème de la liste des certificats révoqués du schéma précédent, un autre schéma **[RZL07]** été proposé qui assure l'authentification de la diffusion en se basant sur l'arbre de Merkle. Cette méthode diminue largement la taille de l'information stockée.

La station de base rassemble toutes les clés publiques des utilisateurs du réseau actuel et construit un arbre de hachage. Cet arbre est construit de N feuilles, où chaque feuille correspond à un utilisateur courant du réseau. Dans ce cas, chaque nœud feuille contient une liaison entre l'identificateur ID de l'utilisateur correspondant et la clé publique correspondante $h(U_{ID}, PK_{UID})$.

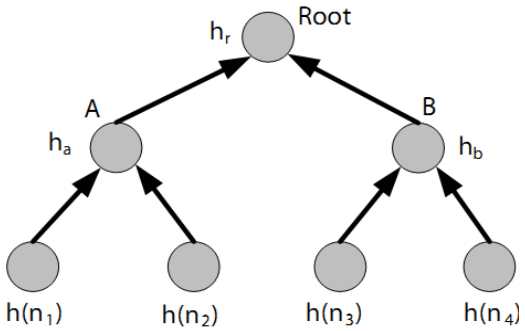


Figure III.3 : Arbre de Hachage de Merkle[RZL07]

Supposant que $w = 4$ (nombre de nœuds feuilles n_i qui doivent être authentifiés). Les valeurs des quatre nœuds feuilles sont les valeurs $h(n_i)$, $i = 1$ à 4, calculées avec une fonction de hachage h à sens unique (Ex. SHA-1). La valeur de chaque nœud interne est dérivée de ses nœuds enfants. Par exemple, la valeur du nœud A est $h_a = h(h(n_1)|h(n_2))$. Les valeurs des niveaux de l'arborescence sont calculées de façon récursive à partir des nœuds feuilles jusqu'au nœud racine. La valeur du nœud racine est $h_r = h(h_a|h_b)$, qui est utilisée pour authentifier un sous-ensemble de nœud n_1 jusqu'à n_4 en utilisant une petite valeur d'informations d'authentification auxiliaires AAI. Si un utilisateur, qui est supposé avoir la valeur de la racine authentique h_r, demande à n_3 d'être authentifié. Avec la valeur n_3, la source envoie les valeurs AAI: <h_a, $h(n_4)$> à l'utilisateur. Ce dernier peut ensuite vérifier l'authenticité de n_3 en calculant trois valeurs $h(n_3), h_b = h\big(h(n_3)\big|h(n_4)\big)$ et $hr = h(h_a|h_b)$, il vérifie par la suite si la valeur calculée de h_r est égale à la valeur de la racine authentique enregistrée par l'utilisateur. n_3 sera accepté, seulement si cette vérification est positive. **[RZL07]**

La station de base précharge/diffuse chaque nœud capteur avec la valeur h_r avant le déploiement du réseau ou pendant le temps de fonctionnement du réseau. Cependant, si h_r est diffusé pendant le fonctionnement du réseau, la valeur h_r devrait être signée par la station de base pour prouver son authenticité, évidemment, dans ce cas, le nœud capteur doit être

préchargé avec la clé publique de la station de base. En parallèle, chaque utilisateur doit obtenir son AAI correspondant à la localisation de son nœud feuille dans l'arbre de hachage de Merkle. Soit T l'ensemble de tous les nœuds le long du chemin d'un nœud feuille à la racine (la racine n'est pas comprise). Alors A est défini comme l'ensemble des nœuds correspondant aux voisins des nœuds dans T, et AAI correspond aux valeurs associées aux nœuds dans A.

Le message envoyé par l'utilisateur U_{ID} est de la forme suivante :

$$< M, tt, SIG_{SKUID} \mathbf{\{ h(U_{ID}||tt||M)\}} ,U_{ID}, PK_{UID} , AAI_{UID} >$$

Chaque nœud vérifie un tel message en deux étapes. D'abord, il vérifie la clé publique PK_{UID} en utilisant AAI_{UID} attachée dans le message et h_r enregistrés dans le nœud. L'opération de vérification est une chaîne d'opérations de hachage avec la valeur finale égale à h_r comme nous l'avons démontré dans la section précédente. Une valeur différente autre que la valeur h_r suggère le refus de la clé publique correspondante. Ensuite, le nœud de capteur vérifie la valeur de $SIG_{SKUID}\{h(U_{ID} \| tt \| M)\}$ en utilisant la clé PK_{UID}.

A la révocation ou l'ajout d'un nouvel utilisateur, le nœud puits met à jour l'arbre de Merkle et obtient un nouveau h_r. Cette nouvelle valeur de h_r est alors signée par la station de base, en utilisant SK_{Sink}. Elle est diffusée aux différents nœuds de capteurs immédiatement. Par ailleurs, chaque utilisateur recevra depuis la station de base la valeur du AAI_{UID} à jour.

- **Discussion du protocole**

Dans ce schéma, une clé publique d'un utilisateur retirée du réseau ne vérifie jamais l'authentification. Par conséquent, les certificats ne sont plus nécessaires et peuvent être éliminés. En outre, le problème de révocation d'utilisateur (révocation de certificat) est maintenant réduit au problème de mis à jour de la seule valeurh_r. Cependant, le schéma est lourd quand N devient grand. C'est parce que la taille de la valeur AAI augmente exponentiellement avec N.

Cette méthode a été améliorée en augmentant la taille de la valeur stockée par les nœuds de capteurs, et diminuant ainsi la valeur du AAI envoyée. Malgré que cette amélioration diminue le coût de communication, elle augmente en parallèle le coût de stockage. Par conséquent, ce schéma est encore inefficace lorsque N est grand.

3.3.3. Authentification basée sur l'identité

Contrairement aux schémas d'authentification précédents, les schémas basés sur l'identité ne nécessitent pas la transmission des certificats numériques, ce qui réduit considérablement la taille des paquets, et le coût de calcul nécessaire pour la vérification des certificats. D'autre

part, les clés publiques sont directement déduites à partir des identificateurs des nœuds du réseau. Ce qui favorise l'utilisation des schémas à base d'identité dans les WSNs.

Dans ce qui suit, nous présentons un protocole d'authentification de diffusion dans les WSN basés sur l'identité des nœuds ; à savoir le protocole IMBAS *(Identity-based multi-user broadcast authentication).*

3.3.4. Le protocole IMBAS

Pour la sécurisation des messages diffusés par la station de base ou par un nœud utilisateur du réseau, le protocole IMBAS **[CKD08]** emploie différentes primitives cryptographiques. En effet, les nœuds ordinaires emploient le schéma vBNN-IBS (voir prochainement dans la section 4.2) pour l'authentification des messages diffusés par des nœuds utilisateurs. Alors que la station de base utilise la signature de Schnorr **[Sch91]** pour la génération de clés privées pour chaque utilisateur, et que cette signature est plus efficace que celle utilisée dans le vBNN-IBS, la station de base utilisera la signature Schnorr **[Sch91]** pour signer ses messages diffusés. La diffusion d'un message ayant la structure <ID_{sink}, tt, M> par la station de base va donc consister à :

- Découper le message M en deux parties, M_1 et M_2, avec ($|M_1| \leq 10\ octets$) et M_2 inclut la donnée ID_{sink} et l'instant tt.
- Choisir aléatoirement un entier $y \in Z_p$ et calcule $Y = [y]P$.
- Coder-et-hacher le point Y en entier i.
- Ajouter des redondances au message M_1 suivant un certain standard comme le standard IEEE P1363a **[Iee00]** pour aboutir à une valeur f_1.
- Calculer ensuite $f_2 = H_1(M_2)$ et calculer $c = i + f_1 + f_2[p]$ jusqu'à ce que $c \neq 0$ sinon retour à l'étape 1 (découpage du message).
- Calculer $d = y - c\ s_{PKG}\ [p]$ pour enfin avoir la signature du message (c, d). Le message ainsi diffusé par la station de base est $< M_2, c, d >$.

D'une autre part, lorsqu'un nœud reçoit le message, il vérifie si tt est valide dans M_2 et si c'est le cas il effectue les opérations suivantes :

- Rejette le message si la proposition ($c \in [1, p-1]$ et $d \in [1, p-1]$) n'est pas vérifiée.
- Calcule $Q = [d]P + [c]P_0$.
- Rejette le message si $Q = O$.
- Code et hache le point Q en un entier.
- Calcule $f_2 = H1(M_2)$ et $f_1 = c - i - f_2\ [p]$.
- Rejetter le message si f_1 est incorrecte, sinon accepter le message et reconstruit $\{ID_{sink}, tt, M\} = \{M_1 || M_2\}$

- *Révocation de noeuds* : Pour la révocation d'un utilisateur, la station de base diffuse un message publiant l'identificateur de l'utilisateur révoqué. Les nœuds capteurs à l'écoute

dans ce cas, établissent une liste de révocation locale. Ensuite, si un nœud reçoit un message provenant d'un nœud révoqué (son ID est dans la liste de révocation locale du récepteur), le message sera automatiquement ignoré **[CKD08].**

Pour réduire les chances qu'un nœud utilisateur ne soit compromis, suite essentiellement à une capture physique du nœud par un adversaire, IMBAS intègre un système de protection des clés privées basé sur l'utilisation de mot de passe. En effet, chaque utilisateur choisit un mot de passe PW_{ID}, et calcule $R' = H_1(PW_{ID})^{-1}$ R et $s' = H_1(PW_{ID})^{-1}s$. (R', s') est ensuite stocké dans dispositif d'exploitation du réseaux de capteurs, au lieu de (R,s). Si l'utilisateur veux utiliser sa paire de clé (R, s), il doit utiliser tout abord le mot de passe correct PW_{ID}, pour restituer le couple (R, s) à partir de (R', s') [Xue07].

- **Discussion du protocole IMBAS**

D'une part, IMBAS emploie le schéma vBNN-IBS (voir prochainement dans 4.2) pour l'authentification des messages diffusés par les utilisateurs du réseau. De plus, la signature vBNN-IBS ne peut être forgée **[Xue07**]. D'autre part, les messages diffusés par la station de base sont authentifiés moyennant l'utilisation de la signature sécurisée **[Nyb93, Nac00]** de Schnorr **[Sch91]**. Dans ce cas, il est impossible pour un attaquant d'injecter des bogues ou de modifier des messages diffusés. De plus, les messages diffusés contiennent l'instant tt (de synchronisation) et qui permet d'éviter les attaques de type rejeu de paquet. On peut ainsi citer les points suivants :

- Le protocole IMBAS permet une résilience contre la compromission d'un nœud. En effet, la station de base tient l'ensemble du réseau au courant de toute éventuelle révocation, en diffusant les identificateurs des nœuds révoqués. D'autre part, il emploie un mode de protection de clé privée basé sur l'utilisation de mot de passe. Dans ce cas, un nœud capturé par un adversaire, ne va contenir que le chiffré de sa clé privée. L'attaquant sera donc en face d'un problème similaire à l'ECDLP.

- Le protocole IMBAS permet une authentification immédiate des messages reçus. Dans ce cas, les paquets forgés seront automatiquement ignorés au lieu de les mettre dans le buffer, ou les faire envoyer vers un prochain saut (communication multi-sauts). D'autre part, lorsqu'un attaquant mène une attaque Flood-DoS sur un réseau non sécurisé, les nœuds physiquement proches, seront considérablement touchés par un DoS sur batterie, due essentiellement au coût nécessaire pour la vérification de signature des paquets inondés par l'attaquant. Cependant IMBAS, permet de réduire le risque d'une telle attaque, en limitant le nombre de signatures échouées.

- IMBAS permet de prendre en charge un réseau à large échelle. Autrement dit, le passage à grande l'échelle est faisable, essentiellement lorsqu'on ajoute des nœuds supplémentaires

(scalabilité) ou lorsqu'on remplace des nœuds défaillants (flexibilité). En effet, pour un identificateur d'utilisateur sur 2 octets, le réseau peut prendre en charge 65535 utilisateurs.

3.3.5. Accélération de la signature de l'authentification de diffusion dans les WSNs

Les chercheurs **[FG12]** ont constaté que la vérification de la signature dans les schémas d'authentification de diffusion basés sur la clé publique est beaucoup plus lente comparée à la vérification de l'authentification de message utilisé dans les solutions basées sur la clé symétrique. Par conséquent, un grand nombre de paquets pourrait attendre dans une file de messages d'un nœud capteur pour les vérifications de signature lorsque de nombreux utilisateurs diffusent des messages. Afin d'améliorer la qualité du service de l'authentification de la diffusion dans les réseaux de capteurs, les auteurs**[FG12]** ont procédé à l'accélération de la vérification de la signature à base de courbe elliptique ECDSA.

- **Description de l'algorithme**

Ce schéma se base sur la signature digitale basée sur la courbe elliptique (ECDSA) vue dans le chapitre précédent. L'idée de base de ce schéma est que certains nœuds divulguent leurs résultats intermédiaires de calcul à leurs voisins lors de la vérification de la signature. Ainsi, de nombreux nœuds peuvent utiliser les résultats de calcul intermédiaires reçus afin d'accélérer leurs vérifications de signature.

En effet, les auteurs ont observé que tous les nœuds de capteurs exécutent indépendamment la même procédure de vérification de signature. Par conséquent, certains nœuds de capteurs consomment leur énergie à libérer une partie des résultats intermédiaires, pour que la vérification de signature de leurs voisins puisse être accélérée de manière significative. Malgré que dans cette méthode le coût de communication est supérieur par rapport à la méthode ECDSA traditionnelle à cause de la diffusion des informations intermédiaires, les auteurs ont constaté que la méthode ECDSA accélérée offre une réduction énergétique d'une manière significative.

- **Discussion du schéma**

Cette méthode exploite pleinement la coopération entre les nœuds de capteurs, et permet une importante diminution dans la consommation d'énergie pour l'ensemble du réseau. Une analyse de performance quantitative montre que ce système peut économiser environ 17,7% à 34,5% de consommation d'énergie et s'exécute 50% plus rapide que la méthode de vérification de signature traditionnelle ECDSA. Cependant, cette méthode nécessite la transmission des certificats numériques avec les messages diffusés, où les nœuds capteurs sont sensés vérifier d'abord l'authenticité du certificat avant celle du message utile. La

maintenance de la liste de révocation des certificats quant à elle, pose d'autres contraintes sur la mémoire des nœuds capteurs qui est limitée.

3.4. Conclusion

La motivation principale pour utiliser la cryptographie basée sur la clé publique dans les WSNs est due aux avancés dans la technologie industrielle des nœuds de capteurs sans fil, aussi bien que dans la mise en œuvre efficace des algorithmes cryptographiques à clés publiques dans les réseaux de capteurs.

Par conséquent, l'utilisation de la cryptographie à clé publique pour garantir l'authentification de diffusion dans les réseaux de capteurs fournit des solutions simples, une résilience de sécurité solide, un meilleur passage à l'échelle, et une authentification immédiate des messages, par rapport aux solutions basées sur la cryptographie symétrique. Nous présentons dans le prochain chapitre la conception de notre méthode, qui est basée sur la cryptographie à clé publique à base d'identité en utilisant la coopération entre les nœuds.

Chapitre IV : Conception

4.1. Introduction

Bien que l'utilisation de la cryptographie à base de clés publiques présente de meilleures performances par rapport à la cryptographie symétrique dans les réseaux de capteurs, la vérification de signature dans ces systèmes est relativement lente, ce qui entraîne une grande consommation énergétique et un long délai de vérification. Pour cela, l'accélération de vérification de signature est considérée comme un problème important, en particulier dans des environnements à ressources limitées. Les auteurs Fan et Gong **[FG12]**ont proposé une méthode d'accélération à base de courbe elliptique ECDSA, par l'exploitation de la coopération entre les nœuds de capteurs. Cependant, leur méthode est basée sur le concept de certificat.

Contrairement aux systèmes de sécurité à base de clés publiques, la cryptographie à base d'identité ne nécessite pas la transmission des certificats, ce qui améliore le coût de calcul, et rend ces systèmes adéquats pour les réseaux de capteurs. Cependant, la vérification de signature à base d'identité est toujours lente et coûteuse en termes de ressources.
Nous nous somme inspirés de la méthode ECDSA accélérée **[FG12],** afin de proposer notre méthode basée sur l'accélération de la vérification de la signature vBNN-IBS en utilisant la coopération entre les nœuds.

4.2. La méthode vBNN-IBS

vBNN-IBS**[CKD08]** est une méthode de signature à base d'identité, où la clé publique de l'utilisateur est dérivée directement de son identité publique, et la clé privée est calculée par un tiers de confiance, appelé PKG (Private Key Generator). L'identité de l'utilisateur représente sa clé publique, et la clé privée représente le certificat de cette clé publique. vBNN-IBS est définit comme suit**:**

Setup:

L'autorité de confiance dans les réseaux de capteurs est la station de base, qui assigne les identités et les clés secrètes à chaque nœud du réseau. Dans un système cryptographique à base d'identité, la station de base doit d'abord produire et distribuer les paramètres publics, elle doit assurer les étapes suivantes :

- Choisir une courbe elliptique E/F_q; un point P de la courbe d'ordre p.

- Générer aléatoirement une clé secrète du système x dans $\mathbb{Z}_p$, et calculer la clé publique du système $P_0 = xP$.
- Choisir deux fonctions de hachage $H_1: \{0,1\} \times G_1^* \rightarrow \mathbb{Z}_p$ et $H_2: \{0,1\}^* \rightarrow \mathbb{Z}_p$.
- Publier les paramètres publics du système $(E/\mathbb{F}_q; P; p; P_0; H_1; H_2)$ et maintenir x privé.

Génération de clés privées

Nous supposons que chaque utilisateur A du réseau, possède un identifiant unique $ID_A \in \{0,1\}^*$. La station de base génère une clé privée Pri_A en utilisant la signature Schnorr **[RZL07]** comme suit:

- Choisir une valeur aléatoire $r \in \mathbb{Z}_p$ et calculer $R = rP$.
- Signer l'identité de l'utilisateur A en utilisant la clé secrète x du système: $s = r + cx$, avec $c = H_1(ID_A \| R)$.

La clé privée de l'utilisateur A est le couple (R, s). Elle est envoyée à travers un canal sécurisé.

Génération de signature

L'utilisateur A signe un message m avec sa clé privée $Pri_A = (R, s)$ comme suit :

- Choisi r une valeur aléatoire $y \in \mathbb{Z}_p$ et calculer $Y = yP$.
- Calculer les deux valeurs h et z, tel que $\boldsymbol{h = H_2(ID_A, m, R, Y)}$ et $\boldsymbol{z = y + hs}$.

La signature du message m par l'utilisateur A est (R, h, z).

Vérification de signature

À la réception du message m avec sa signature (R, h, z) depuis un utilisateur ayant l'identité ID_A, le destinataire effectue les opérations suivantes :

1. Calculer $c = H_1(ID_A \| R)$
2. Calculer les valeurs zP et $h(R + cP_0)$.
3. Calculer $\boldsymbol{H_2(ID_A, m, R, zP - h(R + cP_0))}$ et vérifier le résultat avec le h reçu afin d'accepter la signature.

4.3. Authentification et diffusion des messages

Afin de garantir la fraicheur des messages envoyés et éviter les attaques de rejeu des données, dans lesquelles une transmission est malicieusement répétée par une tierce partie interceptant les paquets sur le canal. Nous supposons dans notre méthode que si un utilisateur ayant l'identité ID veut diffuser un message M, il diffuse le paquet : $< \boldsymbol{M, tt, ID, Sig}\ \{\boldsymbol{M, tt, ID}\} >$ avec ;

- $\boldsymbol{M}$ **:** le message à signer.
- $\boldsymbol{tt}$ **:** le temps courant.

- $\boldsymbol{ID}$**:** l'identité de l'utilisateur.
- $\boldsymbol{Sig}\ \{\boldsymbol{M}, \boldsymbol{tt}, \boldsymbol{ID}\}$ **:** la signature du message en utilisant la clé privée de l'utilisateur.

À la réception du message, le nœud de capteur vérifie s'il ne s'agit pas d'une attaque de rejeu, en vérifiant si le temps d'envoi *tt* ne dépasse pas un certain seuil. Par la suite, il exécute la procédure de vérification de signature.

4.4. Amélioration de la méthode vBNN-IBS

Dans cette section, nous décrivons d'abord le problème de l'authentification de diffusion dans les réseaux de capteurs, ensuite nous montrons comment utiliser la méthode vBNN-IBS pour accélérer la vérification de la signature à travers la coopération des nœuds.

4.4.1. Problématique

Les réseaux de capteurs sont généralement déployés dans des environnements hostiles, pour cela l'authentification de diffusion devient un mécanisme de sécurité nécessaire pour assurer la fiabilité des applications réseaux.

Après qu'un utilisateur s'identifie auprès du réseau, il diffuse un message à l'ensemble des nœuds dans son voisinage. Ceci mène à une procédure d'authentification simultanée entre les nœuds de capteurs afin d'accorder l'accès seulement aux utilisateurs légitimes et garantir la fiabilité du réseau.

Comme illustré dans la Figure IV.1, le nœud émetteur signe son message et le diffuse aux nœuds de capteurs situés dans son voisinage (par exemple, les nœuds A, B et C). Par la suite, les nœuds A, B et C vérifient respectivement la signature en utilisant l'identité de l'émetteur. Si la signature est correcte, ils rediffusent le message (au sein de leur zone de communication). Quand un nœud (E par exemple), reçoit le paquet de diffusion pour la première fois, il exécute la même procédure de vérification de signature afin de déterminer si le paquet reçu doit être retransmis à ces nœuds voisins.

Cette procédure d'authentification de diffusion continue jusqu'à ce que tous les nœuds accessibles reçoivent le message. Si la vérification n'est pas garantie lors de la diffusion, le nœud de capteur supprime le message et envoie un rapport à la station de base.

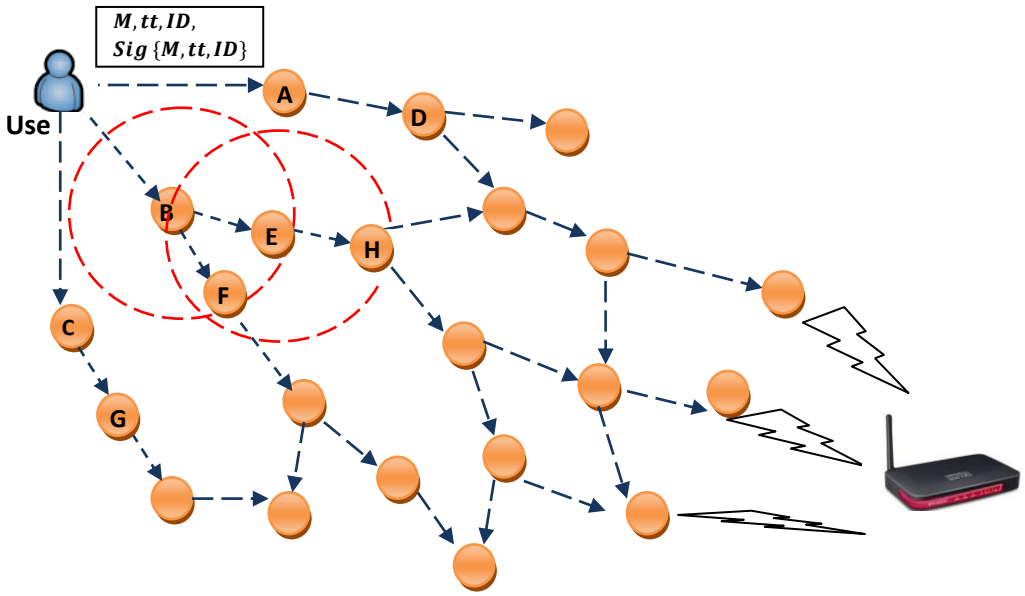


Figure IV.1 : Diffusion de paquet dans les réseaux de capteurs.

4.4.2. Accélération de vérification de la signature

Dans la procédure d'authentification de la diffusion vBNN-IBS présentée dans la Figure IV.1, tous les nœuds de capteur exécutent la même procédure de vérification de signature. En effet, chaque nœud doit calculer $\boldsymbol{H_2(ID_A, m, R, zP - h(R + cP_0))}$, afin de vérifier si la signature est valide. Pour cela, chaque nœud ayant reçu le message, doit d'abord calculer trois multiplications scalaires sur la courbe elliptique$\boldsymbol{l_1P = zP, l_2R = hR,}$ et $\boldsymbol{l_3P_0 = hcP_0}$, qui consomment énormément l'énergie du réseau.

Pour de meilleures performances, nous nous sommes inspirés de la méthode d'accélération présentée dans **[FG12]**. L'idée principale de l'accélération provient du fait que tous les nœuds exécutent indépendamment la même procédure de vérification de signature lors de l'authentification de la diffusion. Pour cela, certains nœuds du réseau consomment leur énergie afin de délivrer certaines informations intermédiaires. Par conséquent, la vérification de signature de leurs voisins peut être accélérée significativement et la consommation de l'énergie dans l'ensemble du réseau pourra être diminuée. L'idée de base est illustrée dans la Figure IV.2.

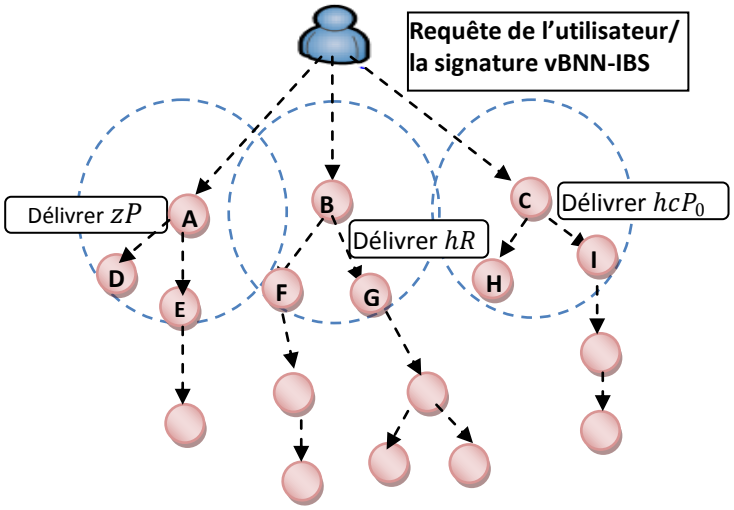


Figure IV.2 : Accélération de vérification de signature vBNN-IBS

Dans la Figure IV.2, l'utilisateur diffuse le paquet$< \boldsymbol{M, tt, ID, Sig}\{\boldsymbol{M, tt, ID}\} >$, où (R, h, z) est la signature vBNN-IBS de (M, tt, ID). Lors de la réception du paquet diffusé par les nœuds A, B et C, ces derniers commencent la procédure de vérification de signature traditionnelle, donc ils calculent les trois multiplications scalaires sur la courbe elliptique$\boldsymbol{zP, hR}$ *et* $\boldsymbol{hcP_0}$**,** afin d'accepter la signature et vérifier si le $\boldsymbol{h}$ reçu est égal à la valeur $\boldsymbol{H_2(ID_A, m, R, zP - h(R + cP_0))}$calculée.

Les nœuds A, B et C de leurs côté, décident de délivrer avec le paquet reçu, les trois informations intermédiaires calculées durant la vérification de signature. Ils délivrent respectivement $\boldsymbol{l_1P = zP}$**,** $\boldsymbol{l_2R = hR}$et$\boldsymbol{l_3P_0 = hcP_0}$.

Par conséquent, les voisins du nœud A (les nœuds D et E), reçoivent le paquet diffusé et l'information intermédiaire $\boldsymbol{zP}$. Ils peuvent vérifier la signature par seulement deux multiplications scalaires sur la courbe elliptique $\boldsymbol{hR}$ et $\boldsymbol{hcP_0}$ et deux additions $\boldsymbol{zP - hR - hcP_0}$. De leurs côtés, les voisins du nœud B (les nœuds F et G) peuvent également effectuer une vérification de signature accélérée de la même façon. Par conséquent, si un nœud du réseau publie son résultat du calcul intermédiaire, la vérification de signature de tous ces voisins pourra être accélérer significativement, en effectuant seulement deux multiplications scalaires, et deux additions sur la courbe elliptique (au lieu de trois), ce qui peut améliorer les performances du réseau de 33% par rapport à la vérification de la signature traditionnelle vBNN-IBS.

Pour une meilleure optimisation, nous avons proposez dans notre méthode, que si un nœud désire délivrer ses résultats intermédiaires, il envoie deux informations à ses nœuds voisins parmi les trois, alors il peut envoyer $\boldsymbol{(l_1P - l_2R) = (zp - hR)}ou\boldsymbol{(l_1P - l_3P_0) = (zp - hcP_0)}$ ou $\boldsymbol{(l_2R + l_3P_0) = (hR + hcP_0)}$. Par conséquent, le récepteur calcule seulement une

multiplication scalaire sur la courbe elliptique et une addition, ce qui améliore la procédure de vérification de signature traditionnelle vBNN-IBS d'environ 66%. Malheureusement, dans notre schéma, les nœuds de capteurs ne peuvent pas délivrer les trois informations intermédiaires (i.e.$\boldsymbol{l_1P}$,$\boldsymbol{l_2R}$, et $\boldsymbol{l_3P_0}$) afin d'accélérer la vérification de la signature par seulement deux additions. La raison et qu'un attaquant peut capturer le nœud émetteur ayant l'identité ID, et pourra lancer l'attaque suivante:

L'attaquant :

1. Choisit aléatoirement les valeurs$m', R', l_1^{'}, l_2^{'} et\ l_3^{'}$.
2. Calcule $l_1^{'}P - l_2^{'}R^{'} - l_3^{'}P_0$.
3. Calcule$h' = H_2(ID, m', R',\ l_1^{'}P - l_2^{'}R^{'} - l_3^{'}P_0)$.
4. Délivre la signature$(R', h',\ l_1^{'}P)$pluslestroisvaleurs $l_1^{'}P, l_2^{'}R^{'}, l_3^{'}P_0$.

La victime :

1. Calcule$c^{'} = H_1(ID \parallel R')$.
2. À partir des trois informations délivrées plus la signature, la victime vérifie bien que $h' = H_2(ID, m', R',\ l_1^{'}P - l_2^{'}R^{'} - l_3^{'}P_0)$.

Par conséquent, si les nœuds de capteurs utilisent les trois informations intermédiaires délivrées à partir de leurs voisins afin d'accélérer la vérification de signature, un message altéré pourra être accepté.

Afin d'éviter l'attaque précédente, nous permettons dans notre solution aux nœuds de capteurs d'utiliser au maximum deux informations intermédiaires parmi les trois $\{l_1P, l_2R, l_3P_0\}$ pour vérifier la signature. Nous supposons, dans la suite de notre rapport que si un nœud divulgue son résultat intermédiaire, il envoie l_2R etl_3P_0. Par conséquent, il envoie le message suivant:

$$\{\boldsymbol{M, tt, ID, Sig\{M, tt, ID\}, (l_2R + l_3P_0)}\}\ (\mathbf{1})$$

Où $Sig\{M, tt, ID\}$ désigne la signature vBNN-IBS du paquet $\{M, tt, ID\}$ par le nœud ID.

4.4.3. Performances de notre méthode

Soient MUL et ADD désignant respectivement une multiplication et une addition de deux points sur la courbe elliptique. Dans notre schéma, un nœud de capteur peut recevoir un paquet de données de la forme $\{M, tt, ID, (R, h, z)\}$ou $\{M, tt, ID, (R, h, z), l_2R + l_3P_0\}$. Lorsqu'un paquet est reçu, le nœud effectue la procédure de vérification de signature, il calcule d'abord la valeurl_1P, puis il attend un délai très court α, afin de vérifier s'il a reçu l'information intermédiaire depuis ses voisins. Si c'est le cas, le nœud peut compléter la vérification de signature avec au total 1MUL + 1ADD. Dans le cas contraire, après l'expiration de la période du temps α, le nœud réalise la procédure de vérification complète avec 3MUL + 2ADD. En d'autres termes, si un paquet de la forme $\{M, tt, ID, (R, h, z), l_2R +$

$l_3P_0\}$ est reçu, le nœud de capteur calcule d'abordl_1P, puis effectue une vérification de signature rapide en utilisant 1MUL + 1ADD.

Si la signature est vérifiée dans les deux cas précédents, le nœud poursuit la diffusion du paquet à ses voisins, sinon, il transmettra un rapport signé à la station de base, qui à son tour, et après avoir reçu suffisamment de rapports à partir des nœuds du réseau, effectue des mécanismes de sécurité appropriés afin d'identifier les nœuds compromis dans le réseau. Bien que le schéma précédent soit simple et efficace, il est toujours vulnérable à l'attaque suivante :

L'attaquant :

1. Choisit aléatoirement m', z', R', Y', et mettre $M' = (m', tt, ID_A)$.
2. Calcule$h' = H_2(ID_A, M', R', Y'), l_1'P = z'P \; et \; Q = l_1'P - Y'$.
3. Utilise (R', h', z') comme signature du messageM', et diffuse le paquet$\{M', (R', h', z'), Q\}$à ses voisins.

La victime de son côté :

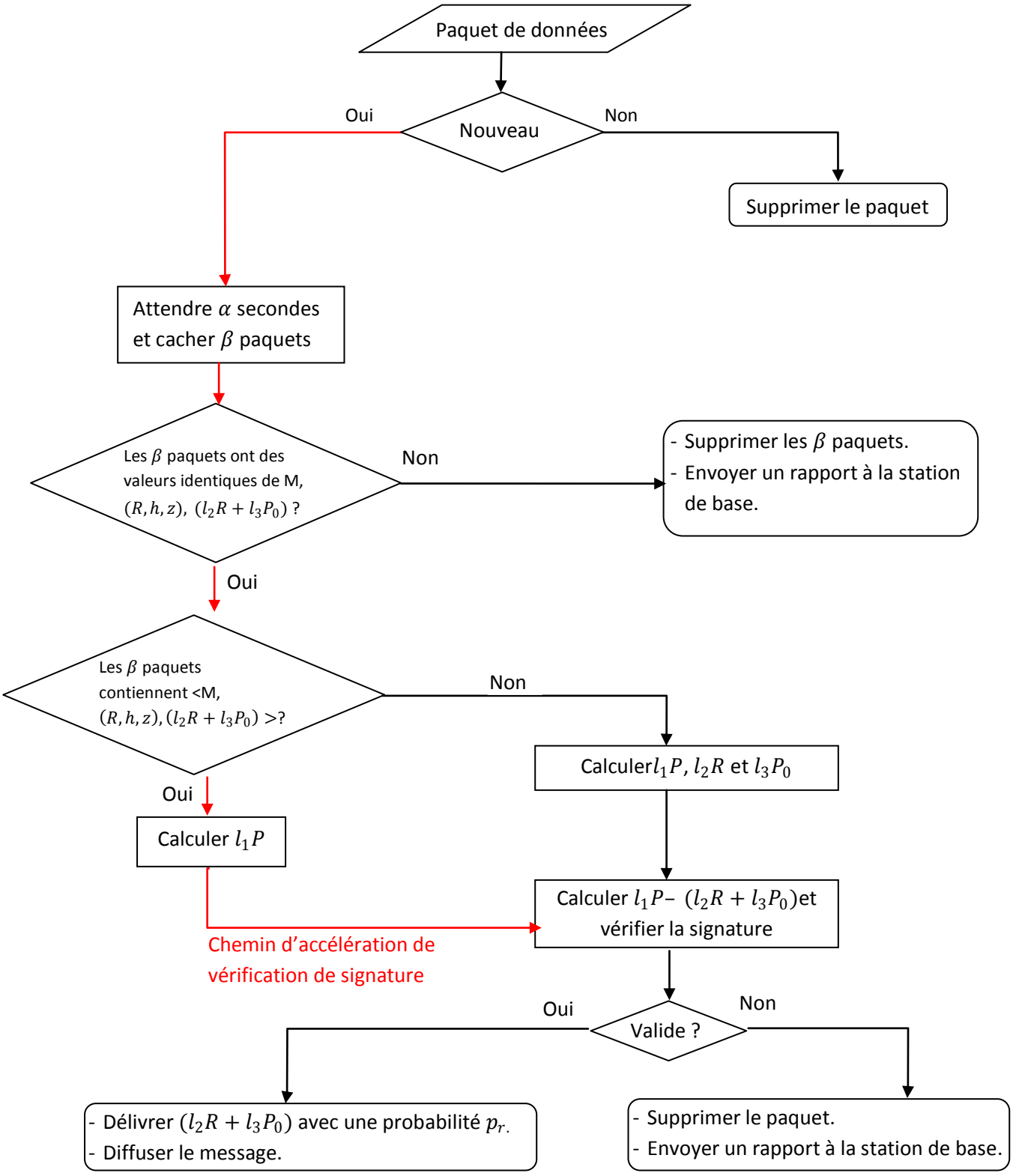
1. Calcule$c' = H_1(ID_A \parallel R')$ et$l_1'P = z'P$.
2. À partir de l'information intermédiaire reçue et la signature, la victime calcule $H_2(ID_A, M', R', z'P - Q)$et le compare avec h' reçu. par conséquent, le message M' est accepté.

L'attaque précédente est valide car l'attaquant est persuadé que tous les voisins de l'émetteur calculent d'abord la valeurl_1P, puis utilisent l'information intermédiaire reçue, afin d'accélérer la vérification de signature. Par conséquent, l'attaquant choisit aléatoirementz', R', Y' et un message altérém', il diffuse par la suite le paquet $\{M', (R', h', z'), Q\}$ qui pourra réussir la procédure de vérification avec succès. Notons que si les nœuds récepteurs utilisent la vérification de signature traditionnelle vBNN-IBS, le paquet altéré$\{M', (R', h', z'), Q\}$ne pourra pas être vérifié correctement, à moins que la valeur délivrer Q est égale à$(h'R' + h'c'P_0)$ avec une probabilité négligeable.

4.4.4. Amélioration de l'accélération de vérification de signature

Afin de faire face à l'attaque précédente, nous avons amélioré notre méthode en utilisant la redondance des paquets de diffusion dans le réseau de capteurs. Donc, dans le schéma amélioré, chaque nœud doit attendre d'abord un délai de α secondes et enregistre β paquets de données (par exemple, (R, h, z) ou $\{(R, h, z), l_2R + l_3P_0\}$) reçus depuis ses voisins, où α et β sont choisis de telle sorte que le nœud peut recevoir au moins un paquet de données à partir d'un voisin honnête. Le nœud vérifie par la suite si les β paquets de données enregistrés ont des valeurs identiques de (R, h, z) et de $l_2R + l_3P_0$. Si le nœud constate que les paquets de données reçus ont des valeurs différentes de R, h, z ou$l_2R + l_3P_0$, il envoie immédiatement, un rapport d'erreur à la station de base. D'autre part, si tous les β paquets sont identiques, le nœud de capteur vérifie s'il a reçu l'information utile $l_2R + l_3P_0$, afin de l'utiliser pour accélérer la vérification de signature, si c'est le cas, il calcule seulement la valeurl_1P. Par

conséquent, il réalise la vérification de signature avec 1MUL + 1ADD. Sinon, le nœud capteur effectuera la vérification traditionnelle de signature vBNN-IBS avec 3MUL + 2ADD.



FigureIV.3 : la méthode vBNN-IBS accélérée.

4.4.5. Sélection des paramètres α et β

Nous supposons dans notre schéma qu'un nœud de capteur A procède en moyenne λ voisins, et la moitié de ces nœuds diffusent leurs paquets au nœud A à un certain degré de communication. Nous supposons également que, parmi les $\lambda/2$ voisins de A,vnœuds peuvent être compromis par des attaquants et chacun de ces nœuds peut envoyer au maximum w paquets altérés au nœud A.

Pour que l'attaque soit valide, tous les nœuds compromis doivent s'entendre pour envoyer des paquets altérés *identiques*, Sinon, le nœud A supprime tous les paquets de données cachés et envoie un rapport à la station de base.

Afin de rendre notre système résilient contre les attaques collusoires, le paramètre β doit satisfaire la condition suivante:

$$\boldsymbol{\lambda/2 \geq \beta \geq v.w + 1} \ (\mathbf{2})$$

La condition précédente garantit que le nœud A peut recevoir au moins un paquet de diffusion à partir d'un voisin honnête. Par conséquent, le nœud A n'accepte pas des messages d'attaques car tous les paquets reçus ne sont pas identiques.

Après que nous avons déterminé le paramètre β sur lequel notre méthode se base, le délai α est choisi de façon que les β paquets soient reçus correctement par le nœud A. Le paramètre α dépend de la vitesse de transmission des paquets, et le délai d'attente consommé par la transmission radio. Deux délais d'attente doivent être pris en considération, à savoir: le délai d'attente d'initialisation et le délai d'attente de congestion. Prenons en considération tous ces facteurs, le délai α doit satisfaire la condition suivante :

$$\boldsymbol{\alpha \geq (Taille_{MAX}/Vitesse_{MAX} + Délai\_Init_{MAX} + Délai\_Cong_{MAX}) * \beta} (\mathbf{3})$$

Où,$Taille_{MAX}$, $Vitesse_{MAX}$, $Délai\_Init_{MAX}$, et $Délai\_Cong_{MAX}$ sont respectivement, la taille maximale du paquet, la vitesse maximale de transmission, le délai d'attente initial maximal, et le délai de congestion maximal.

4.4.6. Sélection de la probabilité de libération des informations intermédiaires p_r

Dans la méthode proposée, la probabilité qu'un nœud de capteur délivre son résultat du calcul intermédiaire p_r, est un paramètre prédéfinis caractérisant un compromis entre la rapidité de vérification de signature et la consommation énergétique dans le réseau de capteurs. En général, si une grande valeur de $p_r$est utilisée, plusieurs nœuds de capteurs vont consommer de l'énergie supplémentaire afin de diffuser leurs résultats de calculs intermédiaires. Par conséquent, la vérification de signature d'un grand nombre de nœuds du réseau sera accélérée, et inversement.

Le choix de la probabilité de libération p_r est étroitement lié à la topologie et le déploiement du réseau. Une fois que les nœuds du réseau soient déployés, l'administrateur du réseau doit analyser la topologie du réseau afin d'estimer le nombre moyen de nœuds de capteurs pouvant exécuter la vérification de signature pendant la transmission d'un paquet de diffusion. L'administrateur détermine alors une probabilité p_r de libération qui peut atteindre un compromis optimal entre la consommation énergétique dans l'ensemble du réseau et l'efficacité de la vérification de signature. Plus précisément, nous supposons qu'il y a en moyenne N nœuds de capteur qui effectuent la vérification de signature à chaque tour de communication, la probabilité que T nœuds de capteurs délivrent leurs résultats de calculs intermédiaires est :

$$p_T = \binom{N}{T} p_r^T (1 - p_r)^{(N-T)}$$

Nous supposons que les trois valeurs $E_s, E_r,$ et E_{MUL} désignent respectivement la consommation énergétique de l'envoi, la réception, et le calcul d'une multiplication scalaire sur la courbe elliptique. Nous avons supposez auparavant que chaque nœud possède en moyenne λ voisins. Par conséquent, nous pouvons estimer la conservation/consommation supplémentaire de l'énergie grâce à l'utilisation de notre technique de vérification de signature accélérée comme suit:

1. T nœuds de capteur diffusent localement leurs résultats de calcul intermédiaire, avec une consommation énergétique de $T * E_s$.
2. Environ $\lambda T/2$ nœuds de capteurs reçoivent des résultats de calcul intermédiaire, avec une consommation énergétique de $\lambda T/2 * E_r$.
3. Environ $\lambda T/2$ nœuds de capteurs accélèrent leur vérification de signature à partir des résultats intermédiaires reçus, avec une minimisation de consommation d'énergie de $\lambda T/2 * 2E_{MUL} = \lambda T * E_{MUL}$

Par conséquent, la conservation/consommation supplémentaire de l'énergie est :

$$\sum_{T=1}^{N} \mathrm{p_T} \left(\mathrm{T} * \mathrm{E_s} + \frac{\lambda \mathrm{T}}{2} * \mathrm{Er} - \lambda \mathrm{T} * \mathrm{E_{MUL}} \right) (4)$$

4.5. Conclusion

Nous avons présenté dans ce chapitre, une nouvelle technique d'accélération de vérification de signature dans les réseaux de capteurs. Notre proposition est basée sur la cryptographie à base d'identité et exploite pleinement la coopération entre les nœuds de capteurs. Une étude quantitative de notre méthode **[BMB12]** a montré une importante conservation dans la consommation énergétique dans un réseau de forme de grille de 4×4, une réduction énergétique d'environ 38.62% et une exécution 66% plus rapide que la

vérification de signature traditionnelle vBNN-IBS. Une comparaison avec d'autre méthode de diffusion a montré que l'énergie est aussi conservée de 23.25% et $25,14\%$ et 31.23% par rapport aux méthodes d'authentification de la division IMBAS **[CKD08]** et EIBAS **[SLP12]** et ECDSA accélérée **[FG12].** Les résultats obtenus ont été présenté dans une conférence disponible dans la partie bibliographie.(Voir **[BMB12]**).

Dans le prochain chapitre, nous commençons par l'évaluation des performances de notre système à travers une analyse théorique. Par la suite nous présentons les résultats de l'implémentation et de l'évaluation des performances à travers des scénarios de simulation et des expérimentations sur les nœuds de capteurs.

Chapitre V : Résultats et performances

5.1. Introduction

Dans une étude empirique **[BMB12]**, nous n'avons considéré que les opérations vitales i.e. (émission et réception d'un message, multiplication de points elliptiques). Autrement dit, nous avons négligé d'autres opérations telles que : l'addition des points elliptiques, le calcul des condensés, le passage en mode Standby, Idle,…etc. Pour cela, afin d'évaluer les performances de notre méthode, une étude empirique ainsi qu'une simulation sur le schéma vBNN-IBS accéléré s'avère nécessaire. Dans une première partie nous donnons les résultats de notre étude empirique **[BNB12]** pour étudier l'impact de l'approche accélérée sur ce schéma basé identité. Dans une seconde partie nous présentons les résultats élaborés dans un projet de Master, où les auteurs **[LOU13]** ont réalisé une simulation partielle, qui consiste à simuler chaque opération vitale et prélever le niveau d'énergie consommé par celle-ci, afin de comparer les résultats obtenus avec les résultats empiriques dans le même contexte. Ensuite, les auteurs ont réalisé une simulation complète en prenant en considération toutes les opérations et comportements d'un nœud pour ainsi pouvoir juger les quatre schémas, à savoir : vBNN-IBS Classique, vBNN-IBS Accéléré, ECDSA Classique et ECDSA Accéléré. Dans la première section, les nœuds capteurs passent en mode Power Sleep après avoir réalisé leur tâche (càd quand le nœud est éteint et n'est pas capable de détecter des signaux radio, donc aucune communication n'est possible). Dans une seconde section, ils restent éveillés (en mode Standby).

5.2. Etude empirique vBNN-IBS

Lors d'une diffusion authentifiée, l'ensemble du réseau (tous les nœuds capteurs) exécute le même procédé de signature pour vérifier l'authenticité des paquets diffusés. En effet, d'après la figure V.1, quand l'utilisateur diffuse un message signé (contenant par exemple une requête), tous les nœuds à sa portée vont recevoir le message (dans notre cas de figure, il s'agit du nœud 1).

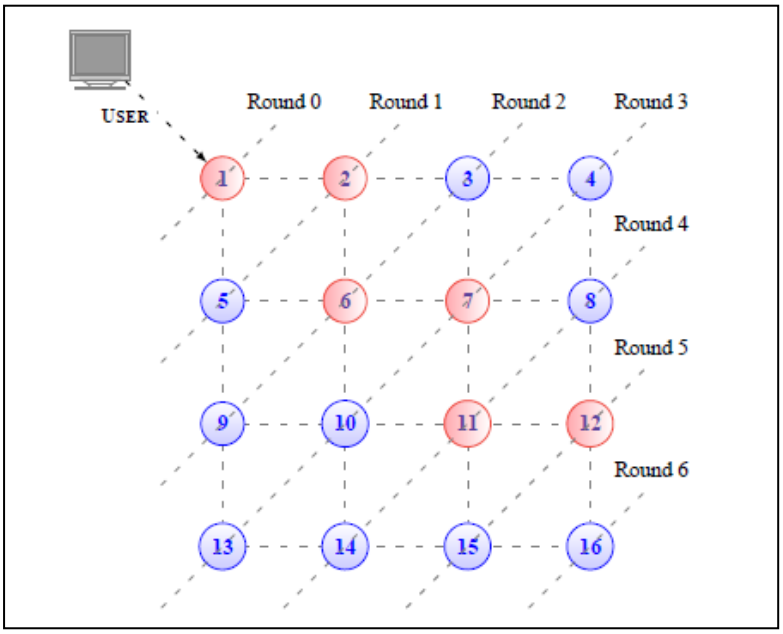


Figure V.1 : Architecture du réseau WSN grille 4x4

Dans ce cas, une fois le nœud (1) ait reçu le message, il commence à vérifier la signature du message en employant le procédé de signature vBNN-IBS, et si le paquet est authentique, il rediffuse à son tour le message pour que les autres voisins non encore atteint par le message, puissent eux aussi l'authentifier et exploiter son contenu (exécuter la requête). Dans notre cas, il s'agit des nœuds (2 et 5). Le procédé de diffusion continu jusqu'à atteindre tous les nœuds du réseau. Si un des nœuds trouve que le message n'est pas authentique, il le rejette.

L'idée proposée par les deux auteurs X. Fang et G. Gang **[FG12],** consiste à sacrifier certains nœuds capteur (par exemple les nœuds : 1, 2, 6, 7, 11 et 12 dans la figure V.1) à consommer en plus leur énergie pour diffuser à leurs voisins, des résultats intermédiaires issues du procédé de vérification. Autrement dit, les nœuds voisins, vont exploiter ces résultats au lieu de les calculer, et donc ils vont accélérer le procédé de vérification de la signature et ainsi économiser de l'énergie. Dans ce cas, pour que l'impact de l'idée soit rentable, les résultats intermédiaires sont issus d'un calcul coûteux comme l'opération de multiplication de points elliptiques. Les auteurs X. Fang et G. Gang **[FG12]** ont réalisé une étude empirique sur l'impact de l'approche accélérée sur le procédé de signature ECDSA, et ont montré que l'approche en question permet d'épargner 34,50% de l'énergie nécessaire pour une diffusion authentifiée. Cependant, les auteurs n'ont pas pris en considération les certificats électroniques.

D'autre part, nous avons réalisé une étude empirique sur le schéma vBNN-IBS pour étudier l'impact de l'approche accélérée sur ce schéma basé identité. Nous présenterons dans ce qui suit, les principaux points de l'étude empirique faite sur le procédé vBNN-IBS.

Suivant la figure V.1, lorsque l'utilisateur diffuse un paquet signé avec le procédé vBNN-IBS $\{M, tt, ID, Sig\{M, tt, ID\}\}$, où $Sig\{M, tt, ID\} = (R, h, z)$ représente la signature digitale vBNN-IBS sur l'entité (M, tt, ID), les nœuds A, B et C vont recevoir le paquet, et lancent la vérification de l'authenticité du paquet. Une fois terminé, les trois nœuds décident respectivement de révéler leurs résultats intermédiaires $l_1P = [z]P$, $l_2R = [h]R$ et $l_3P_0 = [hc]P_0$.

Le procédé de signature vBNN-IBS comporte lors de la vérification d'une signature trois opérations de type multiplication de points elliptiques; à savoir : $[z]P$, $[h]R$ et $[hc]P_0$, avec P le générateur de la courbe elliptique, R étant la première composante de la signature vBNN-IBS et P_0 la clé publique du système. Dans ce cas, les nœuds D et E, qui sont les voisins de A, vont accélérer leur procédé de vérification de signature en opérant une vérification avec deux multiplications de points elliptiques au lieu de trois, plus deux additions de points elliptiques $([z].G - [h].R - [h.c].P_0)$. De cette manière, la vérification va être améliorée de 33% par rapport au schéma classique.

Pour des raisons d'efficacité et de sécurité, nous avons supposé que, tout nœud désirant révéler ses résultats intermédiaires, il révèle deux points parmi trois ; à savoir : $l_2R = [h]R$ et $l_3P_0 = [hc]P_0$. D'autre part, nous avons proposé que l'envoi du résultat intermédiaire soit en une seule partie pour plus de sécurité. Autrement dit, au lieu d'envoyer les deux résultats intermédiaires indépendamment, il est préférable d'envoyer leur somme, qui va prendre 50% de l'espace pris par deux résultats puisque le récepteur aura tôt ou tard à réaliser cette opération d'addition et donc il n'aura plus à la réaliser. Dans ce cas, le résultat intermédiaire sera le point $l_2R + l_3P_0$. Ainsi le paquet à envoyer aura la forme suivante :

$$\{M, tt, ID, Sig\{M, tt, ID\}, (l_2R + l_3P_0)\} \quad (1)$$

Ainsi, pour l'étude empirique, nous avons considéré les points suivants :

- La topologie de la figure V.1.
- La diffusion se propage en ROUND (un niveau en diagonal).
- L'environnement étant idéal, autrement dit, pas d'attaquants, pas d'interférences...
- La taille du paquet (1) du procédé vBNN-IBS accéléré est de 140 octets. La taille maximale autorisée étant de 128 octets.

Dans ce cas, le nœud capteur désirant révéler ses résultats intermédiaires aura à envoyer deux paquets ; à savoir : le paquet original $\{M, tt, ID, Sig\{M, tt, ID\}\}$ et le paquet accéléré

$\{M, tt, ID, (l_2R + l_3P_0)\}$. Dans l'étude empirique ce paquet est considéré de taille maximale de 128 octets.

- Les nœuds capteurs sont de type Mica2.
- D'après le *Datasheet* du mica2, l'intensité du courant électrique pour le mode active est de $8mA$, à la réception $10mA$, pour la transmission $27mA$ et un débit de transmission de $12.4kbps$.
- Une multiplication de points elliptiques dure 810 $\boldsymbol{ms}$.
- Une fois un nœud capteur d'un certain ROUND, fini de vérifier la signature et de forwarder si nécessaire le message diffusé, il passe en mode Power Sleep.

Nous avons ainsi obtenu les résultats suivant:

- Pour la transmission, un nœud de type mica2, consomme $E_{Send} = 3.0 \times 27 \times \frac{8}{12.4} = 52.258\ J$ et $E_{Receive} = 3.0 \times 10 \times \frac{8}{12.4} = 19.35 J$ respectivement pour l'envoi et la réception d'un seul octet. Dans ce cas, pour l'envoi du paquet de $128\ octets$, il consomme $E_{Send} = 6.689\ mJ$ et $E_{Receive} = 2.477\ mJ$.

- L'opération de multiplication de point étant la plus coûteuse dans le procédé de vérification. En effet, elle consomme $E_{mul} = 3.0 \times 8.0 \times 0.81 = 19.44\ mJ$. Dans ce cas, on peut dire que pour une vérification, le nœud va consommer $E_{Ver} = 58.32\ mJ$.

- Les nœuds (1, 2, 6, 7, 11 et 12 représentés en rouge dans la figure V.1) vont localement diffuser leurs résultats intermédiaires à leurs voisins. Dans ce cas, le réseau va consommer en plus 40.134mJ d'énergie supplémentaire pour l'envoi des résultats intermédiaires.

- D'autre part, les nœuds (2, 3, 5, 6, 7, 8, 10, 11, 12, 15 et 16) vont recevoir le paquet accéléré. Le réseau va donc consommer en plus 27.24mJ pour la réception de ces paquets. Autrement dit, une diffusion authentifiée avec le schéma accéléré, consomme en plus 67.38mJ.

- En même temps, les nœuds (2, 3, 5, 6, 7, 8, 10, 11, 12, 15, et 16) vont accélérer leur procédé de vérification de 66% en épargnant $11 \times 2 \times 19.44\ mJ = 427.68\ mJ$. Dans ce cas, le réseau va épargner $427.68 - 67.38 = 360.299\ mJ$.

- Ce gain d'énergie représente **34.33%** de l'énergie totale nécessaire pour la diffusion authentifiée vBNN-IBS.

En résumé, l'approche de l'accélération de la vérification de la signature vBNN-IBS permet d'accélérer la vérification de la signature (vérification **66%** plus rapide pour certains

nœuds), de réduire le coût de calcul, et par conséquence épargner **34.33%** de l'énergie nécessaire pour une diffusion authentifiée dans un réseau WSN grille 4x4.

5.3. Implémentation de l'application vBNN-IBS accélérée

Une application pour notre schéma de signature vBNN-IBS accéléré était implémentée, de manière à ce qu'elle réalise une diffusion authentifiée *One-to-Many***[LOU 13]**. En effet, nous avons deux types de nœuds, les nœuds capteurs ordinaires et les nœuds utilisateurs. Ces derniers sont équipés avec un autre type de matériel qui leur permet d'exploiter le réseau. Le nœud utilisateur ne fait que générer un message m, incrémenter le temporisateur tt, signer le message avec le procédé de signature vBNN-IBS puis diffuser le message dans le réseau; alors que les nœuds capteurs ordinaires réalisent les opérations décrites dans l'algorithme suivant :

```
Début
A la Réception d'un paquet.
Vérification de la fraîcheur du paquet et Buffering.
Si (le paquet est nouveau frais: vérification du champ tt)
      Si (le paquet reçu est le premier de la période tt)
      Mettre le paquet en question dans le buffer 1.
            Déclencher le compte à rebours.
      Sinon Si (le paquet est celui utilisé pour l'accélération)
      Mettre le paquet en question dans le buffer 2.
            Sinon
            Mettre le paquet en question dans le buffer 1.
            Fin

      Fin
Sinon
      Rejeter le paquet (ignore).
Fin

A l'épuisement du compte à rebours (partie 1)
Comparaison de tous les paquets du buffer 1 et du buffer 2
Si (les paquets du buffer 1 sont identiques)
   Si (Le nœud a reçu des paquets pour l'accélération)
      Si (les paquets du buffer 2 sont identiques)
         Ok, tous les paquets sont identiques.
              Sinon
                  Rejeter tous les paquets.
              Fin
       Fin
Sinon
                  Rejeter tous les paquets.

Fin
```

Nous supposons que le délai d'attente **α,** après lequel les nœuds de capteurs commencerons la procédure de vérification est prédéfini. En effet, cette valeur ne doit pas être assez grande pour qu'elle ne retarde pas la diffusion dans le réseau, et donc doit permettre une authentification immédiate. D'autre part, la valeur de **α**, ne doit pas être trop petite, pour qu'un nœud puisse récolter un maximum de paquets. Pendant ce temps, les nœuds capteurs vont sauvegarder dans leur buffer, le maximum de paquets possible (au minimum β paquets).

```
A l'épuisement du compte à rebours (partie 2)
Extraction du contenu d'un paquet de chaque buffer
Si(le nœud n'a pas reçu au moins un paquet pour l'accélération
   (Buffer 2 vide))
     Vérification classique de la signature vBNN-IBS.
Etiq E_1 :Si(la signature n'est pas authentique)
                          Rejeter le message.
                          Aller à E_2
                    Sinon Si(le nœud est sensé révéler les
                    résultats intermédiaires)
                   Envoi du message d'origine.
                              Envoi du message pour
                              l'accélération.
                         Sinon Si(le nœud n'est sensé envoyer
                                    que le paquet d'origine)
                                    Envoi du message d'origine.
                              Sinon
                                    Aller à E_2.
                                Fin
                           Fin
     Fin
Sinon
     Vérification accélérée de la signature vBNN-IBS.
     Allez à E_1.
Fin
Etiq E_2 :Nœud en mode Power-Sleep (Standby) ou en mode Idle.
Fin
```

Lorsque les nœuds capteurs finissent de réaliser leur travail (vérification de l'authenticité de la signature et *Forwarding* des messages aux voisins si nécessaire) ; ils passent en mode Power Sleep. Cependant, pour garantir plus de disponibilité (le réseau est toujours prêt : les nœuds capteurs doivent rester éveillés, et donc en mode Idle). Une étude des protocoles

(vBNN-IBS et ECDSA) a été faite dans les deux cas, les nœuds passent en mode power-sleep et le cas où ils restent éveillés.

5.3.1. Environnement de simulation

Afin d'implémenter l'application vBNN-IBS, des ressources logicielles, ainsi que matérielles ont été utilisées. Nous présentons ainsi :

- **Système d'exploitation TinyOS**

Travaillant sur les réseaux de capteurs, l'environnement de développement le plus répondu de nos jours étant le TinyOS. Il s'agit d'un système d'exploitation intégré, orienté événements et conçu pour les réseaux de capteurs. Il est implémenté entièrement en NesC et respecte une architecture basée sur une association de composants, permettant de réduire la taille nécessaire à sa mise en place. Occupant un espace mémoire faible (512octet), il s'adapte aux capteurs pourvus de ressources mémoire très limitées.

- **Langage de programmation NesC**

Le NesC (***N**etwork **e**mbedded **s**ystems C*) est un langage de programmation dérivé du langage C, basé composants, orienté événements, et utilisé principalement pour le développement d'application TinyOS.
L'application vBNN-IBS accélérée (ci-dessus) est implémentée en langage NesC dans l'environnement TinyOS. Il s'agit en fait, d'une programmation orientée composants et de nature dirigée événement.

5.3.2. La bibliothèque TinyECC

La bibliothèque TinyECC est un package, contenant des composants qui permettent de réaliser des opérations cryptographiques basées sur les courbes elliptiques (ECC-based PKC). D'autre part, le package en question, comporte en plus quelques simples applications cryptographiques pré à l'usage telles que le procédé de signature ECDSA, le protocole d'échange de clés basé sur le scénario de Diffie-Hellman ECDH, et le schéma de chiffrement ECIES.

Cette bibliothèque (version 2.0) complètement écrite en NesC, est particulièrement compatible avec la version 2 de l'environnement TinyOS (TinyOS-2.x) et a été testée sur plusieurs plateformes (mica2/micaZ, TelosB/TmoteSky, BSNV3, et Imote2). Elle comporte plusieurs méthodes d'optimisations, pour permettre une exécution plus rapide de certaines opérations cryptographiques. Parmi ces techniques, on cite :

- **Système à coordonnées projectives :** L'utilisation des coordonnées projectives dite *Jacobian representation*, étant plus efficace que l'utilisation des coordonnées Affine. En effet, ceci permet d'accélérer les opérations d'addition, de doublement et de multiplication de points elliptiques.

- **La méthode Sliding Window :** Cette méthode est principalement utilisée pour accélérer considérablement l'opération de multiplication de points elliptiques. Elle nécessite le pré-calcul de certaines valeurs (un tableau de points elliptiques). Autrement dit, sacrifier de l'espace mémoire.
- **Shamir's Trick :** C'est une méthode qui permet d'accélérer le calcul des opérations de la forme $[u].P + [v].G$, avec u et v deux entiers de **168 bits**, P et G deux points elliptiques. Elle emploie l'algorithme 3.48 de **[HMV04].**

D'autre part, pour les paramètres cryptographiques (de la courbe elliptique), la bibliothèque TinyECC 2.0 utilise les paramètres de courbe elliptique : SECP128-bits, SECP160-bits et SECP192-bits recommandées par le SECG (*Standards for Efficient Cryptography Group).*

5.3.3. Composants utilisés pour l'implémentation

Une application écrite en NesC est essentiellement constituée d'un ensemble de composants reliés entre eux. Pour le développement des applications vBNN-IBS et ECDSA ainsi que leur version accélérée, le package TinyECC 2.0 était utilisé, qui comporte un ensemble de composants NesC permettant de réaliser des opérations cryptographiques basées sur l'utilisation des courbes elliptiques. Nous donnons ainsi une présentation des composants suivants :

- **Le composant ECC**

Le composant ECC permet via l'interface **`ECC.nc`** de faire appel à un ensemble de commandes implémentées par le module **`ECCM.nc`**, nous citons ainsi :

- Initialisation des paramètres de la courbe elliptique et le calcul des valeurs Sliding Window pour le générateur G de la courbe elliptique.
- Réalisation de l'addition de deux points elliptiques (méthode : Affine et projective).
- Multiplication de points elliptiques par un scalaire.
- Le pré-calcul des valeurs Sliding Window pour un point elliptiques données.
- Récupération des paramètres elliptiques.

D'autre part, le composant comporte un fichier en-tête **`ECC.h`** qui définit les structures des : points elliptiques, paramètres de la courbes elliptique,… ; et permet de configurer certains paramètres tels que la taille de la fenêtre Sliding Window (3 points, 15 points, ou plus).

- **Le composant NN**

Le composant NN permet via l'interface **`NN.nc`** de faire appel à un ensemble de commandes implémentées par le module **`NNM.nc`** pour la réalisation des opérations arithmétiques modulaires et non modulaires.

- **Le composant SHA1**

Le composant SHA1 permet via l'interface **`SHA1.nc`** de faire appel à un ensemble de commandes implémentées par le module **`SHA1M.nc`** selon le RFC3174, pour l'utilisation de la fonction de Hachage SHA1.

- **Le composant secpX**

Le composant secpX (secp128r1, secp128r2, secp160k1, secp160r1, secp160r2, secp192r1 ou secp192r2) permet via l'interface **`CurveParam.nc`** de choisir les paramètres de la courbe elliptique prédéfinis par le SECG. L'utilisateur n'aura qu'à spécifier le nom de la courbe elliptique à utiliser au niveau du fichier Makefile utilisé lors de la compilation de l'application. Pour notre cas, les auteurs **[LOU13]** ont utilisé la courbe définie par le secp160r1, les paramètres sont sur 168bits (160bits et 1 octets pour éviter les dépassements de capacité). Autrement dit, les entiers de $\mathbf{Z_p}$ sont représentés par des tableaux d'octets (tableaux de 21 octets).

- **Le composant ECDSA**

Le composant ECDSA permet via l'interface **`ECDSA.nc`** d'utiliser les fonctions de signature digitales ECDSA et de vérification de la signature ECDSA implémentées par le module **`ECDSAM.nc`**.

A partir de là, les auteurs **[LOU13]** se sont inspirés de la structure interne du composant ECDSA, pour développer le composant vBNN-IBS, en utilisant les interfaces des différents composants (ECC, NN, SHA1, …) et qui nous permet ainsi de réaliser principalement les opérations suivantes :

- Initialiser les paramètres de la courbe elliptique.
- Calcul des valeurs de la Sliding Window pour les deux points : générateur de la courbe elliptique *G*, et la clé publique du système *P*.
- Génération d'un entier aléatoire de $\mathbf{Z_p}$.
- Fonction de hachage basée sur la fonction SHA1.
- Signature digitale vBNN-IBS.
- Vérification de la signature vBNN-IBS.
- Vérification de la signature vBNN-IBS accélérée.

Ainsi que d'autres composants TinyOS :

- **Le composant de communication**

Pour la communication (Activation de la Radio, Envoi et réception de paquets), les composants : ActiveMessageC, AMSenderC et AMReceiveC ont été utilisés. La structure de la charge utile envoyée dans un paquet contenant la signature vBNN-IBS $\{M, tt, ID, (R, h, z)\}$ est de la forme suivante :

```
#define message_size 10
typedef nx_struct PaquetMsg
        {
                nx_uint8_t node_id[2];
                nx_uint8_t util[message_size];
                nx_uint8_t tt[2];
                nx_uint8_t h[21];
                nx_uint8_t z[21];
                nx_uint8_t Rx[21];
                nx_uint8_t Ry[21];

        } PaquetMsg ;
```

Figure V.2 : Structure du paquet vBNN-IBS

Tel que :

- Le champ **nœud_id** désigne l'identificateur de l'utilisateur sur 2 octets.
- Le champ **util** désigne le message utile de l'utilisateur sur 10 octets.
- Le champ **tt** désigne un numéro de séquence pour garantir la fraîcheur des données, il est sur 2 octets.
- Le champ **h** désigne la première composante de la signature vBNN-IBS, il est sur 21 octets (168 bits).
- Le champ **z** désigne la seconde composante de la signature vBNN-IBS, il est sur 21 octets (168 bits).
- Le champ **Rx** désigne la composante selon l'axe des abscisses, du point R ; troisième composante de la signature vBNN-IBS, il est sur 21 octets (168 bits).
- Le champ **Ry** désigne la composante selon l'axe des ordonnées, du point R.

D'un autre côté, la structure du message utilisé pour l'envoi des résultats intermédiaires $\{M, tt, ID, Q = (-h)R + (-h.c)P_0\}$ est de la forme :

```
typedef nx_struct AccMsg
        {

                nx_uint8_t node_id[2];
                nx_uint8_t util[message_size];
                nx_uint8_t tt[2];
                nx_uint8_t Qx[21];
                nx_uint8_t Qy[21];

        } AccMsg ;
```

Figure V.3 : Structure du paquet vBNN-IBS intermédiaire

- Le champ **Qx** désigne la composante selon l'axe des abscisses du point Q, elle est sur 21 octets (168 bits).

- Le champ **Qy** désigne la composante selon l'axe des ordonnées du point Q.

- **Le composant TimerC**

Ce composant nous permet d'utiliser des compteurs (TIMER) pour déclencher un événement lors de l'épuisement d'un compte à rebours.

5.4. Implémentation de l'application ECDSA Accélérée

Afin d'implémenter l'application ECDSA accélérée, les auteurs **[LOU13]** ont exploité le composant ECDSA pour développer une application NesC, réalisant le même scénario de diffusion que celui de l'application vBNN-IBS. D'un autre côté, le format de la charge utile qui transporte le certificat numérique de l'utilisateur dans la méthode ECDSA accélérée est de la forme suivante:

```
typedef nx_struct CertMsg
        {

                nx_uint8_t node_id[2];
                nx_uint8_t PK_IDx[21];
                nx_uint8_t PK_IDy[21];
                nx_uint8_t Expt[2];
                nx_uint8_t CA_SIGNATURE[21];
                nx_uint8_t param[2];

        } CertMsg ;
```

Figure V.4 : Structure du certificat numérique dans la méthode ECDSA accélérée

- Le champ **nœud_id** désigne l'identificateur de l'utilisateur sur 2 octets.
- Le champ **PKx** désigne la composante selon l'axe des abscisses de la clé publique de l'utilisateur, elle est sur 21 octets (168 bits).
- Le champ **PKy** désigne la composante selon l'axe des ordonnées de la clé publique de l'utilisateur.
- Le champ **Expt** désigne la date d'expiration du certificat sur 2 octets.
- Le champ **CA_SIGNATURE** désigne la signature digitale ECDSA de l'autorité de certification PKG. Elle est sur 21 octets (168 bits).
- Le champ **param** désigne autres paramètres nécessaires sur 2 octets (Ex : Identificateur du certificat, algorithme de signature de l'autorité de certification, ...).

La structure des messages contenant la signature ECDSA (r,s) étant la suivante :

```
typedef nx_struct PaquetMsg
        {
                nx_uint8_t util[message_size];
                nx_uint8_t tt[2];
                nx_uint8_t node_id[2];
                nx_uint8_t r[21];
                nx_uint8_t s[21];

        } PaquetMsg ;
```

Figure V.5 : Structure du message et sa signature dans la méthode ECDSA accélérée

Afin de vérifier la signature, le nœud doit calculer deux multiplications scalaires sur la courbe elliptique, à savoir : u_1P et u_2Q tel que P désigne le générateur du système, et Q la clé publique de l'émetteur. Par conséquent, afin d'accélérer la vérification de la méthode ECDSA, les auteurs **[FG12]** supposent que certains nœuds divulguent leurs résultats intermédiaires, en envoyant l'information $I = u_1P$. Le message d'accélération de la vérification est de la forme :

```
typedef nx_struct AccMsg
        {
                nx_uint8_t util[message_size];
                nx_uint8_t tt[2];
                nx_uint8_t node_id[2];

                nx_uint8_t Ix[21];
                nx_uint8_t Iy[21];

        } AccMsg ;
```

Figure V.6 : Structure du message intermédiaire dans la méthode ECDSA accélérée

5.5. Résultats de simulation

Dans cette partie, nous élaborons les résultats de simulation présentés dans **[LOU13]**, en utilisant le simulateur Avrora (version **1.7.115**), et en calculant le gain d'énergie apporté par l'approche accélérée aux deux schémas vBNN-IBS et ECDSA, afin de le comparer avec celui obtenu dans l'étude empirique (Etudes partielles)**[BMB12]**. L'utilisateur diffuse un message après l'avoir signé avec le procédé vBNN-IBS.

5.5.1. Etude par simulation partielle

D'après le scénario suivi lors de l'étude empirique ; l'utilisateur diffuse un message après l'avoir signé avec le procédé vBNN-IBS. Chaque nœud recevant ce message va immédiatement vérifier la signature et rediffuser le message à son tour si la signature est authentique, autrement il rejette le message. Le procédé continue jusqu'à atteindre tous les nœuds du réseau. L'application vBNN-IBS est accélérée comme suit :

- La diffusion s'effectue dans un réseau de seize (16) capteurs organisés en grille comme illustré dans la Figure V.1. La diffusion s'effectue par ROUND, telle que chaque nœud d'un ROUND, une fois qu'il vérifie la signature digitale vBNN-IBS, il passe en mode Power Sleep (Mode veille).
- Pour l'approche accélérée, nous avons supposé suivant le scénario respecté dans l'étude empirique, que les nœuds colorés en rouge dans la Figure V.1, sont sensés diffuser leur résultats intermédiaires à leurs voisins du prochain ROUND (car ceux du ROUND précédant sont passés en mode veille). Nous avons ainsi via une variable booléenne (*Selected*) différencié les nœuds entre eux quant à la diffusion des résultats intermédiaires. Si un nœud est *Selected,* alors il doit divulguer ses résultats intermédiaires.

D'autre part, nous avons utilisé une autre variable booléenne (*Acc*) différenciant les nœuds entre eux quant à l'accélération de la vérification de la signature. En effet, seulement certains nœuds vont accélérer la vérification (i.e les nœuds 2, 5, 3, 6, 7, 8, 10, 11, 12, 15 et 16 sur la Figure V.1).

Dans ce cas, lorsqu'un nœud, dont la valeur du *Acc = 0,* reçoit le paquet du message d'origine, il réalise immédiatement la vérification de la signature. Autrement dit, il n'attend pas le délai α pour la réception de plusieurs paquets (le temporisateur est à Nuls). Les nœuds avec (Acc = 1) attendent la réception d'un paquet pour l'accélération. Si par erreur ce paquet n'est pas reçu, le nœud effectue une vérification traditionnelle après l'épuisement du délai α.

5.5.1.1. Simulation des opérations vitales

Nous entendons par simulation partielle : simuler chaque opération vitale et de prélever le niveau d'énergie consommé par celle-ci. De cette manière, les résultats obtenus seront proches de ceux obtenus dans l'étude empirique. **[BMB12]**

5.5.1.2. Les opérations d'émission/réception et multiplication

Pour mesurer l'énergie consommée lors d'une émission/réception d'un paquet, une petite application *TinyOS* a été programmée, pour l'envoi et la réception de paquets 802.15.4. Ce programme écrit en *NesC,* et compilé pour la plateforme *micaZ,* sera ensuite exploité par le simulateur *Avrora*.

En utilisant la version 1.7.115 de *Avrora,* les paquets transmis ont une taille supplémentaire (Header & Footer) de **19 octets**. Dans notre cas, un paquet contenant une charge utile de **56 octets**, aura une taille de **75 octets**. Des résultats de simulation présentés dans **[LOU13]**, nous montrent que l'émission d'un **octet** consomme **53.737µJ**. Autrement dit,

pour notre application vBNN-IBS accélérée qui envoie des paquets avec une charge utile de **98** (respectivement **56**) **octets**, nous considérons la consommation énergétique de **6.287** (respectivement **4.030**) **milli Joule**, pour les deux charges utiles précédentes. D'un autre côté, on remarque par simulation que l'énergie consommée par l'émission est trois (03) fois celle de la réception=**17.912µJ**.**[LOU13]**

Par simulation, les auteurs montrent que l'énergie moyenne consommée par une multiplication de point elliptique est de **51.795mJ**. **[LOU13]**

5.5.1.3. Simulation partielle et évaluation des performances du vBNN-IBS vs vBNN-IBS Accéléré

Après avoir prélevé les principales valeurs énergétiques des différentes opérations par la simulation, nous présentons dans ce qui suit, une analyse de consommation énergétique du schéma vBNN-IBS, et nous évaluons l'impact de l'approche accélérée sur ce schéma.

- **Simulation partielle vBNN-IBS Classique**

Dans le schéma vBNN-IBS classique, chaque nœud une fois qu'il reçoit un paquet de données, contenant une signature, il opère une vérification complète de la signature (i.e. 3 multiplications de points elliptiques). Si la signature est vérifiée avec succès, le nœud en question retransmet le paquet à ses voisins du prochain ROUND. Le procédé continue jusqu'à atteindre tous les nœuds du réseau.

Lorsque les nœuds : 1, 2, 3, 5, 6, 7, 9, 10, 11 et 12, aient vérifié la signature vBNN-IBS, ils rediffusent le paquet de données à leurs voisins. Nous aurons ainsi :

- 10 paquets de données envoyés par les nœuds (98 octets de charge utile pour chaque paquet).
- 20 paquets de données reçus par les nœuds (98 octets de charge utile pour chaque paquet).

En utilisant les valeurs moyennes des consommations énergétiques des différentes opérations vitales, obtenues lors de leur simulation (**53.737µJ** pour l'envoi d'un octet, **17.912µJ** pour la réception d'un octet et **51.795mJ** pour une multiplication de point elliptique), nous obtenons les résultats suivants :

Lors de la diffusion, le réseau consomme :

- 10 * 117 * 53.737 µJ = **62.872mJ** pour l'envoi des paquets de données.
- 20 * 117 * 17.912 µJ = **41.914mJ** pour la réception des paquets de données.

Autrement dit, pour une diffusion authentifiée avec le schéma vBNN-IBS, le réseau consomme (62.872 + 41.914) = **104.786mJ** d'énergie pour la communication. D'autre part, le procédé de vérification consomme 51.795*3 = **155.385mJ** d'énergie par nœud.

En résumé, une diffusion authentifiée par le schéma vBNN-IBS classique dans un réseau WSN grille 4x4, consomme (155.385*16 + 104.786) = **2590.946mJ**.

- **Simulation partielle vBNN-IBS Accéléré**

Dans le schéma accéléré, un certain nombre de nœuds (1, 2, 6, 7, 11 et 12 : représentés en rouge) doivent envoyer un paquet supplémentaire de **56 octets** de charge utile, contenant les valeurs intermédiaires, nécessaires pour accélérer le procédé de vérification vBNN-IBS. Dans ce cas, le réseau va consommer en plus ((56 + 19) * 53.737µJ * 6) = **24.182mJ** pour l'émission des paquets supplémentaires. D'autre part, les voisins de ces nœuds (du prochain ROUND) vont recevoir le paquet en question. En effet, les nœuds 2, 3, 5, 6, 7, 8, 10, 11, 12, 15 et 16 vont recevoir le paquet et vont consommer chacun ((56 + 19) * 17.912µJ) = **1.343mJ** en plus. Autrement dit, pour la réception de tous ces paquets lors d'une diffusion authentifiée, le réseau consomme en plus (1.343*11) = **14.773mJ**.

En résumé, par rapport au schéma classique, notre schéma accéléré consomme en plus **38.955mJ**, pendant la diffusion. Cependant, les nœuds ayant reçu le second paquet (les valeurs intermédiaires), vont accélérer le processus de vérification en opérant une seule multiplication de point. Par conséquence, l'énergie épargnée dans le réseau en question, sera donc de 11*2*51.795 = **1139.534mJ**.

De manière générale, le gain d'énergie dans tout le réseau, et pour une diffusion est de **1100.579mJ**. Par conséquent, notre schéma accéléré permet d'épargner le pourcentage suivant d'énergie :

$$E_{\%} = (1100.579*100)/(2590.946) = \mathbf{42.477\%.}$$

Nous remarquons que l'impact de l'approche accélérée sur le schéma vBNN-IBS est d'épargner **42.477%** d'énergie pendant une diffusion. Nous résumons ainsi les résultats dans le tableau suivant (*Nous avons supposé dans l'étude empirique que le paquet de données et le paquet de l'accélération sont tout les deux de taille 128 octets*):

		Energie d'Emission milli Joule	Energie de réception milli Joule	Energie de vérification milli Joule	Energie consommée (Total en milli Joule)	Gain d'énergie
Empirique	**vBNN-IBS classique**	**66.890**	**49.536**	**933.12**	**1049.546**	**34.33 %**
	vBNN-IBS Accéléré	**107.024**	**76.781**	**505.44**	**689.245**	
Simulation	**vBNN-IBS classique**	**62.872**	**41.914**	**2486.16**	**2590.946**	**42.47 %**
	vBNN-IBS Accéléré	**87.054**	**56.687**	**1346.67**	**1490.411**	

Tableau V.1 : Comparaison entre les résultats empiriques et de simulation pour le schéma vBNN-IBS

Nous remarquons que les résultats (gain d'énergie) obtenus dans les deux études empirique et simulation se rapprochent de très peu, et ceci est dû principalement à l'énergie consommée par l'opération de multiplication de point, considérée par l'étude empirique à **19.44mJ**, étant donné un temps d'exécution de **0.81 seconde**. Cependant, les auteurs **[LOU13]** ont démontré par simulation, que le temps d'exécution d'une telle opération est en moyenne égal à **1958ms** et ne peut pas aller au-dessous de **1500ms**. Donc, si nous considérons un tel temps d'exécution pour l'étude empirique, nous prendrons en considération une énergie de **45.6mJ** par opération. Nous obtenons ainsi le pourcentage suivant :

	Energie d'Emission milli Joule	Energie de réception milli Joule	Energie de vérification milli Joule	Energie consommée (Total en milli Joule)	Gain d'énergie
vBNN-IBS classique (Empirique)	**66.890**	**49.536**	**2188.8**	**2305.226**	**40.596%**
vBNN-IBS Accéléré (Empirique)	**107.024**	**76.781**	**1185.6**	**1369.405**	

Tableau V.2 : Les valeurs empiriques pour le schéma vBNN-IBS

Nous remarquons cette fois-ci, que les résultats (gain d'énergie) obtenus dans les deux études empirique et simulation se rapprochent et la différence n'étant pas considérable. Cette dernière est due principalement à la taille des paquets choisis dans l'étude empirique. En effet, nous avons supposé que les paquets de données et ceux contenant les résultats intermédiaires

sont de taille maximale de 128 octets, alors que réellement, les deux paquets n'atteignent pas cette taille.

5.5.1.4. Simulation partielle et évaluation des performances de ECDSA vs ECDSA Accéléré

Pour réaliser une diffusion authentifiée avec le procédé de signature ECDSA, l'utilisateur doit impérativement faire accompagner le message avec son certificat (certificat signé par l'autorité de certification : SINK). Cependant, la taille du message $<U_{ID}, tt, M, Sig_{ECDSA}, Cert_{ID}>$, dépasse la taille autorisée pour la charge utile. Dans ce cas, l'utilisateur doit envoyer deux messages ; un message de 56 octets contenant les données suivi d'un message de 88 octets contenant son certificat.

A la réception des deux messages, les nœuds sont amenés à vérifier l'authenticité du certificat pour valider et utiliser la clé publique de l'utilisateur pour ainsi vérifier la signature ECDSA du message.

- **Simulation partielle ECDSA**

Lorsque les nœuds : 1, 2, 3, 5, 6, 7, 9, 10, 11 et 12, aient vérifié la signature ECDSA, ils diffusent un paquet de données suivi d'un paquet de certification à leurs voisins. Nous aurons ainsi :

- 10 paquets de données envoyés par les nœuds (56 octets chaque paquet).
- 10 paquets de certification envoyés par les nœuds (88 octets chaque paquet).
- 20 paquets de données reçus par les nœuds (56 octets chaque paquet).
- 20 paquets de certification reçus par les nœuds (88 octets chaque paquet).

En utilisant les valeurs moyennes des consommations énergétiques des différentes opérations vitales, obtenus lors de leur simulation (**53.737µJ** pour l'envoi d'un octet, **17.912µJ** pour la réception d'un octet et **51.795mJ** pour une multiplication de point elliptique), nous avons les résultats suivants:

Lors de la diffusion, le réseau consomme :

- 10 * 75 * 53.737 **µJ** = **40.302mJ** pour l'envoi des paquets de données.
- 10 * 107 * 53.737 **µJ** = **57.498mJ** pour l'envoi des paquets de certification.
- 20 * 75 * 17.912 **µJ** = **26.868mJ** pour la réception des paquets de données.
- 20 * 107 * 17.912 **µJ** = **38.33mJ** pour la réception des paquets de certification.

Donc, un total de **97.8mJ** pour l'envoi des paquets et **65.198mJ** pour leur réception. Autrement dit, la diffusion consomme **162.998mJ** pour la communication.

Les résultats de simulation montrent que dans le schéma classique, le procédé de vérification de la signature ECDSA du message consomme au total **187.343mJ**.

En résumé, une diffusion authentifiée par le schéma ECDSA classique dans un réseau WSN grille 4x4, consomme (187.343*16 + **162.998**) = **3160.486mJ**.

- **Simulation partielle ECDSA Accéléré**

Dans ce mode, certains nœuds appelés (Selected Nodes) vont devoir envoyer, en plus des deux paquets précédents, un autre paquet contenant une valeur intermédiaire. Les nœuds en question sont le 1, 2, 6, 7, 11 et le 12. On aura ainsi :

- 6 paquets supplémentaires envoyés par les nœuds (98 octets de charge utile pour chaque paquet).
- 11 réceptions (98 octets de charge utile pour chaque paquet).

En utilisant les valeurs moyennes des consommations énergétiques des opérations vitales, nous obtenons les résultats suivants :

- Nous avons, 6* 117* 53.737µJ **= 37.723mJ** pour l'envoi des paquets supplémentaires.
- Nous avons, 11*117*17.912µJ = **23.05mJ** pour la réception des paquets supplémentaires.

Donc, un total de 60.773mJ d'énergie consommée par le réseau.

Cependant, les nœuds 2, 5, 3, 6, 7, 10, 8, 11, 12, 15 et 16 qui vont recevoir le paquet supplémentaire vont réaliser une accélération de la vérification de la signature ECDSA du message en opérant seulement une seule multiplication de point elliptique au lieu de deux. Par simulation, une telle opération consomme 51.795mJ d'énergie. Dans ce cas, 11*51.795 = 569.745mJ seront économisées lors de la vérification. Autrement dit, une diffusion ECDSA accélérée, permet au réseau WSN grille 4x4 d'épargner 569.745 – 60.773 = 508.972mJ d'énergie. Nous obtenons ainsi le pourcentage d'épargne d'énergie suivant :

$$E_{\%} = (508.972*100)/(3160.486) = 16.10\%.$$

5.5.1.5. Comparaison vBNN-IBS vs ECDSA dans le cas d'une simulation partielle

L'approche accélérée permet d'épargner **42.47%** de l'énergie du réseau quand le schéma vBNN-IBS est appliqué. En revanche, les résultats de simulation montrent que l'approche d'accélération permet d'épargner seulement **16%** de l'énergie du réseau quand le schéma ECDSA est appliqué **[LOU13].** Autrement dit, la consommation d'énergie dans le schéma

vBNN-IBS Accéléré est réduite à **26%** par rapport au schéma ECDSA Accéléré. Cela dit que :

- L'impact de l'approche accélérée sur le schéma vBNN-IBS, est meilleur que sur le schéma ECDSA. Autrement dit, il est plus intéressant d'appliquer l'approche de l'accélération sur le schéma vBNN-IBS que sur le schéma ECDSA. Ceci est dû principalement au fait que dans le schéma ECDSA, l'accélération est appliquée seulement sur la signature du message et non pas sur la signature du certificat.
- D'autre part, l'accélération appliquée sur ECDSA fait passer le procédé de vérification d'une fonction à deux multiplications de points elliptiques à une fonction à une seule multiplication de points elliptiques. Alors que, dans le schéma vBNN-IBS, le procédé de vérification passe d'une fonction à trois multiplications de points elliptiques à une fonction à une seule multiplication de points elliptiques.
- D'autre part, une diffusion authentifiée avec le schéma ECDSA, consomme plus d'énergie qu'une diffusion avec vBNN-IBS, à cause de la vérification supplémentaire du certificat par l'ensemble du réseau. Nous résumons ainsi les résultats dans le tableau suivant :

Schémas	Energie d'Emission milli Joule	Energie de réception milli Joule	Energie de vérification milli Joule	Gain d'énergie
ECDSA classique (Simulation)	**97.80**	**65.198**	**2997.488**	**16%**
ECDSA Accéléré (Simulation)	**135.523**	**88.248**	**2427.743**	
vBNN-IBS classique (Simulation)	**62.872**	**41.914**	**2486.16**	**42 %**
vBNN-IBS Accéléré (Simulation)	**87.054**	**56.687**	**1346.67**	

Tableau V.3 : Consommation énergétique par les protocoles vBNN-IBS, ECDSA et leurs versions accélérées dans le cas d'une simulation partielle.

5.5.2. Etude complète (Section 1)

Dans cette partie, les auteurs **[LOU13]** ont réalisé une simulation complète sur l'impact de l'approche accélérée. Cependant, nous entendons par Section1, le scénario dans lequel les nœuds capteurs passent en mode Power Sleep lorsqu'ils terminent leur tâche. Notons que dans la simulation complète, toutes les opérations sont prises en considération sans exception. Il s'agit de la simulation de tout le programme.

5.5.2.1. Simulation complète vBNN-IBS vs vBNN-IBS accéléré et ECDSA vs ECDSA accéléré

Le même procédé de simulation (10 simulations) a été appliqué sur le schéma vBNN-IBS et ECDSA dans leurs versions classique et accélérée afin de prélever les valeurs énergétiques pour chaque nœud du réseau. La somme totale de l'énergie consommée par le réseau WSN grille de 4x4 est présentée dans la figure suivante :

	Energie de diffusion dans le réseau Joule	**Gain d'énergie**
vBNN-IBS classique	**42.79**	**36.26%**
vBNN-IBS Accéléré	**27.26**	
ECDSA classique	**43.46**	**16.02%**
ECDSA Accéléré	**36.49**	

Tableau V.4 : Consommation énergétique par les protocoles vBNN-IBS, ECDSA et leurs versions accélérées dans le cas de simulation complète (section1)

A partir du tableau V.4, nous constatons que le schéma accéléré permet de préserver **36.26%** de l'énergie totale nécessaire pour réaliser une diffusion authentifiée vBNN-IBS dans un réseau WSN grille 4x4. D'autre part, nous remarquons que le gain trouvé par la simulation complète (**36.26%**) est inférieur à celui trouvé lors de l'étude partielle (empirique + simulation). En effet, en plus de la différence vue lors de la simulation partielle (les opérations non prises en considération), un nœud en état (Idle + Standby) consomme de l'énergie en plus. En effet, un nœud en mode Idle consomme de l'énergie pour rester à l'écoute du support de communication tant que sa Radio est activée, et en mode Standby, le nœud a besoin de l'énergie pour alimenter l'ensemble de ses composants, pour ensuite passer en mode Power Sleep après la vérification de son authenticité et le forwarding du message si nécessaire. L'énergie consommée dans ces deux états augmente avec le temps. Autrement dit, plus la diffusion authentifiée (sa propagation dans le réseau) met du temps, et plus les nœuds consomment de l'énergie. En effet, plus un nœud est situé loin de l'utilisateur qui diffuse et plus l'intervalle de temps pendant lequel le nœud capteur est en mode (Standby + Idle) est grand, et donc consomme de l'énergie.

En d'autre partie, le schéma ECDSA accéléré permet de préserver seulement **16.02%** de l'énergie totale nécessaire pour réaliser une diffusion authentifiée dans un réseau WSN grille 4x4. D'autre part, nous remarquons que le gain trouvé par la simulation complète (**16.02%**) est égal à celui trouvé lors de la simulation partielle (**16.10%**). Autrement dit, l'énergie

consommée par les deux modes Standby et Idle dans l'approche accélérée n'a pas une grande différence par rapport à celle consommée dans le schéma ECDSA classique.

5.5.2.2. Durée de vie du réseau

Si nous supposons l'utilisation de nœuds capteurs alimentés par des batteries de type AA, à 2450mAh. Notre réseau de 16 nœuds en grille 4x4, va initialement avoir une énergie égale à $16*E_0 = 16*$ ((2450*3*3600)/1000) = 16*30780 = **423360Joule**.

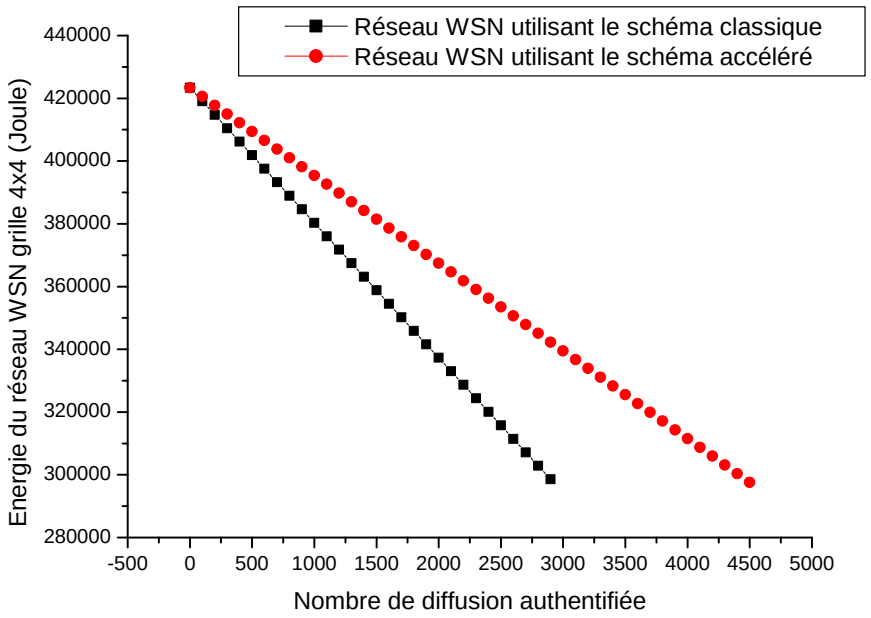
Les résultats de simulation présentés dans **[LOU13]**sont donnés dans le tableau suivant :

Scénario	vBNN-IBS Classique	vBNN-IBS Accéléré
Temps d'exécution	50298 **ms**	25175 **ms**

Tableau V.5 : duré de vie du réseau

En utilisant les valeurs prélevées lors de la simulation complète **[LOU13]** et l'énergie consommée par un nœud lorsqu'il n'effectue aucune opération avec et sans la Radio activée. Nous pouvons calculer la durée de vie du réseau comme suit :

- Dans le schéma vBNN-IBS classique, l'ensemble du réseau consomme **42.79 Joule** pendant la diffusion qui dure 50 secondes.
- Dans la version accélérée, l'ensemble du réseau consomme **27.26 Joule** pendant la diffusion qui dure 25 secondes.
- Les résultats de simulation montrent qu'un nœud en mode Idle consomme **4.43mJ/seconde.**



¸Figure V.7 : la consommation de l'énergie du réseau WSN grille de 4x4

Si nous réalisons une diffusion chaque minute, alors on aura :

- Dans le schéma classique, le réseau consomme **42.79Joule** pendant **50 secondes** et **0.0443 Joule** pendant **10 secondes**. Donc un total de **42.83 Joule** par **minute**.
- Dans le schéma accéléré, le réseau consomme **27.26 Joule** pendant **25 secondes** et **0.155 Joule** pendant **35 secondes**. Donc un total de **27.41 Joule** par **minute**.

D'autre part, les nœuds capteurs de type Mica2 ou MicaZ ne peuvent fonctionner correctement quand leur source d'énergie chute au-dessous de **2.1 Volt**. Autrement dit, lorsque la source d'énergie atteigne les **18522Joule**, le nœud ne peut fonctionner correctement. De ce fait, le réseau s'épuise lorsqu'il atteint les **296352Joule**.

Avec de telles valeurs énergétiques consommées, nous pouvons dire que :

- Le schéma vBNN-IBS classique permet de réaliser **2965 diffusions**.
- Le schéma vBNN-IBS accéléré permet de réaliser **4633 diffusions**.

Puisque nous avons supposé que notre application permet une diffusion chaque minute, alors nous aurons les durées de vie suivantes :

- Le schéma vBNN-IBS classique permet au réseau de durer **2965 minutes,** autrement dit, le réseau dure **49 heures**.

- Le schéma vBNN- IBS accéléré permet au réseau de durer **4633 minutes,** autrement dit, le réseau dure plus de**77heures**.

D'après la Figure VI.8 , nous remarquons que l'énergie du réseau en utilisant la diffusion par schéma vBNN-IBS classique, chute plus rapidement que celle d'un réseau à diffusion accélérée. Ceci est dû au débit d'énergie consommée par diffusion qui différencie les deux schémas. En effet, la diffusion avec le schéma classique consomme **42.83 Joule**, et donc **4283 Joule** pour **100 diffusions authentifiées**. Pour ce nombre de diffusion, le schéma accéléré consomme quant à lui **2741 Joule**. De cette manière, le schéma accéléré va permettre un plus grand nombre de vérifications, et une plus grande durée de vie pour le réseau. En effet, le schéma accéléré permet au réseau de fonctionner plus de **27 heures en plus** par rapport au schéma classique et donc de réaliser **1668 diffusions authentifiées** en plus.

5.5.3. Etude complète (Section 2)

Dans cette partie, les auteurs **[LOU13]** ont réalisé une simulation complète sur l'impact de l'approche accélérée. Cependant, nous entendons par Section 2, le scénario dans lequel les nœuds capteurs restent éveillés en mode Idle lorsqu'ils terminent leur tâche.

5.5.3.1. Simulation complète vBNN-IBS vs vBNN-IBS accéléré et ECDSA vs ECDSA accéléré

La Figure V.6 montre l'énergie consommée de la simulation complète de chaque schéma des applications vBNN-IBS et ECDSA et leurs versions accélérées. Dans ce cas, nous considérons que les nœuds restent toujours éveillés (en mode Idle).

	Energie de diffusion dans le réseau Joule	**Gain d'énergie**
vBNN-IBS classique	**59.407**	**48.12%**
vBNN-IBS Accéléré	**30.818**	
ECDSA classique	**60.32**	**21.36%**
ECDSA Accéléré	**47.433**	

Tableau V.6 : consommation énergétique par les protocoles vBNN-IBS, ECDSA et leurs versions accélérées dans le cas de simulation complète (section 2)

Le schéma vBNN-IBS accéléré permet donc de préserver **48.12%** de l'énergie. Le gain d'énergie est donc meilleur que celui trouvé dans le scénario de la section 1. En effet, l'impact du mode Idle est important (**3.99Joule/minutes**), autrement dit, plus la diffusion authentifiée

prend du temps plus il y a de l'énergie consommée par les nœuds lorsqu'ils sont dans l'état Idle.

Comme première proposition, nous nous permettons de dire que l'approche accélérée a un meilleur impact lorsqu'elle est appliquée dans une application qui exige que les nœuds capteurs restent éveillés en permanence. D'autre part, ce scénario permet une plus grande disponibilité, puisque chaque utilisateur peut à n'importe quel moment bénéficier du service réseau. Autrement, il aura à attendre que les nœuds capteurs s'activent (active leur Radio), pour qu'il puisse diffuser.

5.6. Performances de notre schéma contre l'injection des données erronées

Les performances de notre méthode d'accélération de vérification de signature sont étroitement liées au déploiement du réseau de capteurs et de la distribution des attaquants dans le réseau. Pour étudier l'impact des attaques sur notre système, nous procédons dans une première étape à une analyse dans un réseau de forme de grille de taille 4×4, illustré auparavant dans la Figure V.1.

5.6.1. Cas de base

Dans le réseau de capteurs ci-dessus, chaque nœud peut communiquer directement avec ses voisins à un saut. L'utilisateur envoie son paquet de diffusion signé au nœud d'identité 1 pendant le tour de communication 0. Après six tours de communication, le paquet de diffusion sera reçu et vérifié par tous les nœuds de capteur. Cependant, pour accélérer la vérification de signature, nous supposons dans notre schéma, qu'un seul nœud de capteur publiera ces résultats intermédiaires $(l_2R + l_3P_0)$pendant chaque tour de communication (seuls les nœuds en bleu 1, 2, 6, 7, 11 et 12 diffuseront l'information intermédiaire). Nous présentons dans le tableau suivant les nœuds de capteur qui reçoivent et utilisent les résultats des calculs intermédiaires lors de la procédure de diffusion.

Tour de communication	L'émetteur	Les récepteurs
0	Nœud 1	Nœud 2,5
1	Nœud 2	Nœud 3,6
2	Nœud 6	Nœud 7,10
3	Nœud 7	Nœud 8,11
4	Nœud 11	Nœud 12,15
5	Nœud 12	Nœud 16

Tableau V.7 : Procédure de diffusion dans un réseau de grille 4×4

Comme montre le réseau précédent, certains nœuds de capteurs (par exemple, le nœud 6) ont quatre voisins à un saut, seulement deux d'entre eux (les nœuds 7 et 10) recevront les résultats de calcul intermédiaires, car les deux autres (les nœuds 2 et 5) ont effectué la vérification de signature pendant le tour de communication précédent (tour 1) et passent en mode veille. Par conséquent, il ya au total 11 nœuds de capteurs qui reçoivent les résultats de calcul intermédiaires (les nœuds 2, 3, 5, 6, 7, 8, 10, 11, 12, 15 et 16) dont la vérification de signature pour ces nœuds sera accélérée de 66% dû à l'utilisation de l'information intermédiaire.

5.6.2. Performances contre les attaques des adversaires indépendants

Nous analysons la sécurité de notre schéma dans un réseau de forme de grille de taille 4×4 précédent (Figure V.1), avec l'existence des adversaires totalement indépendants. Pour cela, nous supposons que le réseau comporte deux adversaires indépendants (i.e 12.5% des nœuds du réseau seront compromis et deviennent des nœuds malicieux), et chacun de ces nœuds diffusera un message erroné à ces voisins. Afin de maximiser l'influence des adversaires, nous supposons que les nœuds 3 et 9 sont des nœuds malicieux. Nous supposons aussi dans notre schéma que les nœuds 2, 3, 4, 5, 9, et 13 qui peuvent recevoir uniquement un paquet de données depuis leurs voisins pendant un certain tour de communication, vont exécuter une vérification de signature traditionnelle en effectuant trois multiplications scalaires et deux additions afin d'éviter les attaques. Les autres nœuds du réseau (i.e. nœuds 6, 7, 8, 10, 11, 12, 15, et 16) reçoivent deux paquets de données depuis leurs voisins, à partir desquelles, ils décident s'ils peuvent exécuter une vérification de signature accélérée ou traditionnelle. Comme dans le cas de base, nous supposons que les six nœuds en bleus (i.e. nœuds 1, 2, 6, 7, 11, 12) vont diffuser localement leurs résultats intermédiaires ($l_2R + l_3P_0$)à leurs voisins.

Toutefois, en raison d'existence des adversaires indépendants, les nœuds 7 et 10 recevront deux paquets de données différents à partir de leurs voisins. Pour cela, ces deux nœuds vont exécuter la procédure de vérification de signature traditionnelle sans prendre en considération l'information intermédiaire reçue depuis le nœud 6. Par conséquent, la vérification de signature sera accélérée seulement pour les nœuds 6, 8, 11, 12, 15, et 16.

Ainsi, pour un réseau de taille 4×4. Les attaques des nœuds indépendants n'ont pas d'effet sur la sécurité du système et tous les paquets de données erronées des adversaires indépendants sont écartés par des nœuds légitimes.

5.6.3. Performance de notre schéma contre des attaques des adversaires collusoires

Nous analysons les performances de notre schéma dans le cas d'un réseau de la forme de grille de taille 4×4 précédent (Figure V.1), avec l'existence des attaques des adversaires collusoires. Nous supposons que les deux adversaires collusoires sont déployés dans le réseau

de façon à diffuser des paquets de données falsifiés *identiques* à leurs voisins. Afin de maximiser l'influence des adversaires collusoires, nous choisissons le nœud 2 et le nœud 5 comme deux adversaires collusoires dans le réseau précédent.

D'un autre côté, nous utilisons les mêmes hypothèses que celles du cas des adversaires indépendants pour les autres nœuds. Toutefois, en raison d'existence des adversaires collusoires (les nœuds2 et 5), le nœud 6 sera trompé et recevra deux paquets falsifiés identiques à partir des deux nœuds 2 et 5. Bien que le nœud 6 continue la diffusion de l'ensemble des paquets falsifiés, les nœuds 7 et 10 vont recevoir deux paquets de données différents de leurs voisins et donc vérifient la signature avec la méthode traditionnelle en négligeant l'information intermédiaire reçue depuis le nœud 6. Pour cela, la vérification de signature sera accélérée uniquement pour les nœuds 8, 11, 12, 15, et 16.

Par conséquent, les paquets de données falsifiés des deux adversaires collusoires ne peuvent influencer les réseaux avec succès. En outre, si le nœud 6 écoute le canal pour le prochain tour de communication après la diffusion du paquet falsifié, il pourra déterminer l'attaque. Plus précisément, après que les nœuds 7 et 10 vérifient la signature avec succès pendant le troisième tour de communication, ils vont diffuser un paquet correct à leurs voisins. Le nœud 6 va constater que tous les paquets de données reçus de ses voisins (les nœuds 2, 5, 7, 10) sont différents et donc une attaque s'est produite.

Par conséquent, les attaques des deux adversaires collusoires ont un effet très limité sur la sécurité du système et elles peuvent également être détectées après un certain tour de communication.

5.6.4. Performance de notre schéma contre le rejeu des données

Le rejeu des données est un problème assez crucial dans les protocoles de sécurité dans les réseaux de capteurs, dans lesquelles une transmission est malicieusement répétée par une tierce partie interceptant les paquets sur le canal de transmission. Nous avons supposez dans notre méthode que si un utilisateur ayant l'identité ID veut diffuser un message M, il diffuse le paquet : $< \boldsymbol{M}, \boldsymbol{tt}, \boldsymbol{ID}, \boldsymbol{Sig}\ \{\boldsymbol{M}, \boldsymbol{tt}, \boldsymbol{ID}\} >$où $\boldsymbol{tt}$ représente le temps d'envoi.

À la réception de ce message, les nœuds de capteurs vérifient s'il ne s'agit pas d'une attaque, en vérifiant la validité du temps d'envoi $\boldsymbol{tt}$ s'il ne dépasse pas un certain seuil, sinon le paquet sera rejeté. Par conséquent, l'attaque de rejeu de données ne pourra pas se produire dans notre méthode.

5.6.5. Attaques de compromission des utilisateurs du réseau

Notre méthode fournit une solution robuste afin de résister contre les attaques de compromission des utilisateurs du réseau. En effet, quand un utilisateur s'est compromis, la station de base diffuse un message de révocation à l'ensemble du réseau.

En outre, pour résister à ce type d'attaques, on suppose dans notre méthode qu'un utilisateur protège sa paire de clé privée avec un mot de passe. Il choisit d'abord un mot de passe PW, puis il calcule $R^{'} = H_1\,(PW)^{-1}R$ et$s^{'} = H_1\,(PW)^{-1}s$. Les valeurs $(R^{'}, s')$seront stockées auprès de l'utilisateur au lieu de la clé(R, s). Si l'utilisateur veut utiliser sa clé privée, il doit utiliser le mot de passe PW pour la calculer. Ceci est possible seulement si PW est correct. Même si un attaquant pourrait capturer le dispositif de l'utilisateur, il ne pourra connaître que la paire de clé privée chiffrée$(R^{'}, s')$. Si l'attaquant ne connait pas le mot de passe PW, il ne pourra pas calculer la clé privée(R, s) à partir de $(R^{'}, s')$ qui consiste en un problème ECDLP. D'un autre côté, seulement les paramètres publics du système sont stockés dans les nœuds de capteurs, donc un attaquant ne pourra pas calculer la clé privée à partir de la clé publique du système, ce qui revient à résoudre le problème ECDLP.

Par conséquent, notre méthode est robuste contre les attaques de compromission des nœuds de capteurs et des utilisateurs du réseau.

5.6.6. Attaques Sybils

Les systèmes cryptographiques à base d'identité exigent l'existence d'une autorité de confiance (PKG) qui est chargée de délivrer les clés privées aux utilisateurs à travers des canaux confidentiels et authentiques. Dans les réseaux de capteurs, il peut y avoir cette autorité centrale qui gère le réseau et qui détient une liste de toutes les identités des nœuds déployés. Ainsi, cette autorité centrale qui est la station de base dans les cas des réseaux de capteurs, peut détecter une attaque Sybil en interrogeant le réseau et en comparant les résultats des requêtes avec le déploiement connu auparavant. D'un autre côté, un nœud du réseau pourra vérifier la liste des identités des nœuds "connus comme bons" pour valider d'autres nœuds du réseau.

La liste des identités connues au déploiement doit être protégée d'une éventuelle modification malicieuse. En effet, si l'adversaire est capable d'ajouter de nouvelles identités à cette liste, il pourra rajouter des nœuds Sybils au réseau. De plus, l'entité qui gère le réseau de capteurs doit être capable de rajouter de façon sécurisée de nouveaux nœuds au réseau.

5.7. Conclusion

L'étude (complète) réalisées montre que l'approche de l'accélération permet d'épargner (Par la simulation 36.26%) de l'énergie totale nécessaire pour réaliser une diffusion

authentifiée dans un réseau de capteurs WSN grille 4x4 où les nœuds sont autorisés à passer en mode Power Sleep après avoir terminé leur tâche. Autrement dit, les nœuds passent en mode économie d'énergie, ce qui rend le service réseau indisponible pendant un certain moment. D'autre part, l'approche accélérée permet d'épargner seulement (Par la simulation 16.02%) de l'énergie totale lorsque le procédé ECDSA est appliqué. Ceci favorise le schéma vBNN-IBS, sur le procédé ECDSA.

Dans la section 2, lorsque nous avons considéré un réseau de capteurs où les nœuds doivent rester en mode Idle, et donc plus de disponibilité, l'étude (complète) réalisée montre que l'approche de l'accélération permet d'épargner (Par la simulation **48.12%)** de l'énergie totale nécessaire pour réaliser une diffusion authentifiée dans un réseau de capteurs WSN grille 4x4. Par ailleurs, l'approche accélérée permet d'épargner seulement (Par la simulation **21.36%)** de l'énergie totale lorsque le procédé ECDSA est appliqué. Ceci favorise encore une fois le schéma vBNN-IBS, sur le procédé ECDSA.

Dans ce cas, il est plus intéressant d'appliquer l'approche de l'accélération sur le schéma vBNN-IBS que sur le schéma ECDSA. Autrement dit, le schéma vBNN-IBS Accéléré est plus adapté à l'authentification de la diffusion dans un réseau de capteurs. Cependant le protocole vBNN-IBS accéléré reste vulnérable à certaines attaques de dénis de service :

Lorsque le nombre de signatures vBNN-IBS échouées, atteint un certain seuil, le protocole IMBAS considère l'utilisateur en question comme utilisateur malveillant, ce qui incite le protocole à révoquer l'utilisateur. Ceci peut être exploité par un attaquant en faisant échouer toutes les vérifications des signatures produites par un utilisateur, pour ainsi le révoquer et le priver du service réseau.

Conclusion générale

Les réseaux de capteurs vont sans doute dans les années à venir constituer un développement technologique majeur apportant des solutions aux différents problèmes dans plusieurs domaines d'applications liés à la sécurité, la santé, l'agronomie, la domotique, etc.

Dans notre étude, nous nous sommes intéressés à la sécurité dans les WSNs, plus particulièrement à l'authentification de la diffusion dans ce type de réseaux, qui est un mécanisme de sécurité essentiel, permettant à un ensemble des utilisateurs à diffuser des messages dans le réseau d'une manière authentique.

L'étude réalisée dans cet aspect de recherche, est basée sur l'accélération de vérification de la signature vBNN-IBS. Notre proposition repose sur la cryptographie à base d'identité et exploite pleinement la coopération entre les nœuds de capteurs.

Les résultats présentés dans notre rapport et en collaboration avec le travail de **[LOU13]**, montrent que l'approche de l'accélération permet d'épargner (par simulation en prenant en considérations toutes les contraintes) **36.26%** de l'énergie totale nécessaire pour réaliser une diffusion authentifiée dans un réseau de capteurs WSN grille 4x4, où les nœuds sont autorisés à passer en mode Power Sleep après avoir terminé leur tâche. Autrement dit, les nœuds passent en mode économie d'énergie, ce qui rend le service réseau indisponible pendant un certain moment. D'autre part, l'approche accélérée permet d'épargner seulement **16.02%** de l'énergie totale lorsque le procédé ECDSA est appliqué. Ceci favorise le schéma vBNN-IBS, sur le procédé ECDSA.

D'un autre côté, lorsque les nœuds doivent rester en mode Idle, on a plus de disponibilité du réseau, l'étude (complète) réalisée montre que l'approche de l'accélération permet d'épargner **48.12%**de l'énergie totale nécessaire pour réaliser une diffusion authentifiée dans un réseau de capteurs WSN grille 4x4. Par ailleurs, l'approche accélérée permet d'épargner seulement **21.36%**de l'énergie totale lorsque le procédé ECDSA est appliqué. Ceci favorise encore une fois le schéma vBNN-IBS, sur le procédé ECDSA. D'autre part, nous pouvons dire que l'étude par simulation est fiable et donc on peut l'utiliser pour l'évaluation de n'importe quel autre protocole.

Par conséquent, il est plus intéressant d'appliquer l'approche de l'accélération sur le schéma vBNN-IBS que sur le schéma ECDSA. Autrement dit, le schéma vBNN-IBS Accéléré est plus adapté à l'authentification de la diffusion dans un réseau de capteurs.

Dans ce cas, il est plus intéressant d'appliquer l'approche de l'accélération sur le schéma vBNN-IBS que sur le schéma ECDSA. Autrement dit, le schéma vBNN-IBS Accéléré est plus adapté à l'authentification de la diffusion dans un réseau de capteurs.

Cependant le protocole vBNN-IBS accéléré reste vulnérable à certaines attaques de dénis de service :

Un attaquant peut modifier un seul paquet puis le rejouer pour inciter les nœuds récepteurs à rejeter tous les paquets. Dans ce cas, il est préférable d'implémenter une stratégie permettant aux nœuds de sélectionner un ensemble de paquets considérés légitime (les paquets majoritairement identiques) parmi plusieurs, pour ne pas tout rejeter.

Lorsque le nombre de signature vBNN-IBS échouées, atteint un certain seuil, le protocole IMBAS considère l'utilisateur en question comme utilisateur malveillant, ce qui incite le protocole à révoquer l'utilisateur. Ceci peut être exploité par un attaquant en faisant échouer toutes les vérifications des signatures produites par un utilisateur, pour ainsi le révoquer et le priver du service réseau.

Bibliographie

[AG07] **Antoine Gallais,** «*Ordonnancement d'activité dans les réseaux de capteurs : l'exemple de la couverture de surface*», Université des Sciences et Technologies de Lille, France, document pour l'obtention du titre de Docteur en Sciences mathématiques, 26 Juin 2007.

[AK04] **Akyildiz, I. F. and Kasimoglu, I. H.,** «*Wireless sensor and actor networks: Research challenges*», Ad Hoc Networks (Elsevier), Volume 2, Issue 4, October 2004, Pages 351–367.

[AM08] **Abdallah Makhoul** « Réseaux de capteurs : localisation, couverture et fusion de données.» Thèse de doctorat, Laboratoire d'Informatique de l'Université de Franche-Comté (LIFC), 14 novembre 2008.

[AME05] **Alfred Menezes**, «An Introduction to Pairing-Based Cryptography.» Department of Combinatorics and Optimization, University of Waterloo, Ontario, Canada N2L 3G1, 2005.

[AP02] **Adrian Perring, Robert Szewczyk J.D. Tygar, Victor Wen and David E. Culler**, «SPINS: Security Protocols for Sensor Network.» *Wireless Networks*, Volume 8 Issue 5, Pages 521 - 534, Kluwer Academic Publishers Hingham, MA, USA, September 2002.

[AP08] **A. Liu and P. Ning, "**TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks", in Proceedings of the 7th International Conference on Information Processing in Sensor Networks (IPSN 2008), SPOTS Track, pages 245-256, April 2008.

[AS05] **Abdullatif Shikfa, «**Bilinear Pairings over Elliptic Curves » , Ecole doctorale STIC de Nice Sophia-Antipolis, mémoire de Master, Juin 2005.

[BF01] **D. Boneh and M. Franklin.** «Identity-based encryption from the weil pairing.» SIAM J. of Computing, Vol. 32, No. 3, pp. 586-615, 2003. An extended abstract of this paper appears in the Proceedings of Crypto 2001, vol. 2139 of Lecture Notes in Computer Science, pp. 213–229, Springer-Verlag, 2001.

[BK07] **Bouabdellah Kechar**, « *Problématique de la consommation de l'énergie dans les réseaux de capteurs sans fil* »,Thèse de Doctorat, Université d'Oran, Octobre 2007.

[BMB12] **Chafika Benzaid, Sana Medjadba, Ameer Al-Nemrat et Nadjib Badache «**Accelerated Verification of an ID-based Signature Scheme for Broadcast Authentication in Wireless Sensor Networks.» In Proceedings of IEEE 15th International Conference on Computational Science and Engineering

(CSE'2012), p633-639, Decembre 2012.

[BOK09] **A.Bachir, A.Ouadjaout, L.Khelladi, M.Bagaa, N.Lasla, Y.Challal**, «Information Security in Wireless Sensor Networks», ISBN: 978-981-283-348-8, Encyclopedia On Ad Hoc And Ubiquitous Computing, Octobre 2009.

[BSS05] **Ian F. Blake, Gadiel Seroussi, Nigel P. Smart**, "Advances in Elliptic Curve Cryptography", **Editeur :** Cambridge University Press, 298 page, (25 avril 2005) .

[CB08] **Yacine Challal, Hatem Brttahar** « Introduction à la Sécurité informatique», Université de Technologie de Compiègne, France, 15 Octobre 2008.

[CBL09] **Tony Cheneau, Aymen Boudguiga, Maryline Laurent-Maknavicius** «Amélioration des performances des adresses CGA et du protocole SEND: étude comparée de RSA et d'ECC/ECDSA» Institut TELECOM, TELECOM SudParis, France, pp. 139-156, SAR-SSI 2009.

[CEH01] **Alberto Cerpa, Jeremy Elson, Michael Hamilton, Jerry Zhao, Deborah Estrin et Lewis Girod.** «Habitat monitoring: application driver for wireless communications technology.» In Workshop on Data communication in Latin America and the Caribbean, SIGCOMM LA '01, pages 20–41, New York, USA, 2001.

[CHC09] **Sung-Chul Jung and Hyoung-Kee Choi**. «An energy-aware routing protocol considering link-layer security in wireless sensor networks.» Volume:01, Page(s): 358 – 361, Advanced Communication Technology, ICACT 2009. 11th International Conference, Février 2009.

[CKD08] **X.Cao, W.Kou , L.Dang, B.Zhao**« IMBAS: Identity-based multi-user broadcast authentication in wireless sensor networks": Computer Communications Volume 31, Issue 4, Pages 659–667, 5 Mars 2008.

[DJB08] **Djallel Eddine Boubiche,** «Protocole de routage pour les réseaux de capteurs sans fil», Mémoire de magistère, Université de l'Hadj Lakhdar, Batna, Algérie, 2008.

[DKB05] **Djamel Djenouri, Lyes Khelladi, Nadjib Badache,** « A survey of security issues in mobile ad hoc and sensor networks», IEEE Communications Surveys and Tutorials,Journal, Page(s): 2-29, 2005.

[EG02] **L. Eschenauer and V. Gligor.** «A Key Management Scheme for Distributed Sensor Networks». In 9th ACM conference on Computer and Communications Security, November 2002

[EG09] **L. Ertaul, M. Ganta** , « Security in Wireless Sensor Networks –A Study » The 2009 International Conference on Wireless Networks ICWN'09, July, Las Vegas.

[FG12] **X.Fan and G.Gong** X. Fan and G. Gong. Accelerating signature-based broadcast authentication for wireless sensor networks. *Ad Hoc Netw.*, 10:723–736, 2012.

[GGM86] **GOLDREICH,O.,GOLDWASSE,S.,ANDMICALI,S.** "How to construct random functions."J.ACM 33,4(Octobre 1986)

[GPW04] **N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz. «**Comparing elliptic curve cryptography and RSA on 8-bit cpus.» Cryptographic hardware and embedded systems CHES 2004, roceedings, volume 6, page 119. Springer-Verlag New York Inc, 2004

[HKY06] **Tian He , Sudha Krishnamurthy , Ting Yan, Liqian Luo, Lin Gu , Radu Stoleru , Gang Zhou , Qing Cao , Pascal Vicaire , John A. Stankovic , Tarek F. Abdelzaher , Jonathan Hui , Bruce Krogh ,: «Vigilnet:** An integrated sensor network system for energy efficient surveillance.» ACM Transactions on Sensor Networks 2(1), 1–38 (2006)11.

[HMV] **D. Hankerson, A. Menezens, and S. Vanstone.** *Guide to Elliptic Curve Cryptography*. Springer, 2004.

[HPJ03] **Y. Hu, A. Perrig and D. B. Johnson,** «Packet leashes: a defense against wormhole attacks in wireless networks». In INFOCOM 2003. IEEE, Vol. 3, pp. 1976–1986, 2003.

[Iee00] **IEEE P1363a Standard**, Standard specifications for public key cryptography, 2000. Available from: <http://grouper.ieee.org/groups/1363/index.html/>.

[IGE00] **C. Intanagonwiwat, R. Govindan and D. Estrin**, «Directed diffusion: A scalable and robust communication paradigm for sensor networks» Proc. of ACM MobiCom, pp. 56-67, 2000.

[KB04] **Lyes Khelladi, Nadjib Badache**« *Les réseaux de capteurs: état de l'art* », Rapport de recherche, Algérie, Février 2004.

[KB08] **Lyes Khelladi, Nadjib Badache,** « Improving Directed Diffusion With Power-Aware Topology Control For Adaptation to High Density», LOCALGOS'08 workshop, in conjunction with The 4th IEEE/ACM International Conference on Distributed Computing In Sensor SystemS (DCOSS 2008), Algeria, 2008.

[**KP06]** **Krzysztof Piotrowski et al**, "How Public Key Cryptography Influences Wireless Sensor Node Lifetime", SASN '06 Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks, Pages 169 – 176, October 2006.

[KSF02] **Swastik Kopparty , Srikanth V . Krishnamurthy, Michalis Faloutsos, Satish K. Tripathi**« Split TCP for Mobile Ad Hoc Networks» Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE, 138 - 142 vol.1, 17-21 Nov. 2002

[KW03] **C. Karlof and D. Wagner,** «Secure routing in wireless sensor networks: attacks and counter measures», In Proc. 1st IEEE Int'l. Wksp. Sensor Network Protocols and Applications (SNPA‟03), pp. 293-315, 2003.

[LN12] **LABRAOUI Nabila** «La sécurité dans les réseaux sans fil AD HOC » thèse de Doctorat, université Abou Bekr Belkaid, Telemcen, 2012.

[LN04] **DONGGANG LIU and PENG NING** :«Multi-Level µTESLA: Broadcast Authentication for Distributed Sensor Networks» ACM Transactions on Embedded Computing Systems (TECS),Volume 3 Issue 4, Pages 800 – 836, Novembre 2004.

[LN07] **LASLA Noureddine**, « La gestion de clés dans les réseaux de capteurs sans-fil », mémoire de Magister, Institut National de formation en Informatique (I.N.I), promotion 2006-2007

[LOU13] **Karim Lounis.** "Diffusion Authentifiée dans les Réseaux de Capteurs Sans fil". Mémoire de Master, Spécialité Réseaux et Systèmes Distribués, Département Informatique, Faculté d'Électronique et d'Informatique, USTHB, Juin 2013.

[LV06] **Akos Léczi, A. N. et P. Volgyesi,** « Deploying a wireless sensor network on an active volcano », IEEE Internet Computing, 10(2), p. 18-25, 2006.

[LWS04] **Ruizhong Lin, Zhi Wang, and Youxian Sun.«** Energy efficient medium access control protocols for wireless sensor networks and its state-of-art.» Industrial Electronics, 2004 IEEE International Symposium, Page(s):669 - 674 vol. 1, 4-7 May 2004.

[MWS04] **D. J. Malan, M. Welsh, and M. D. Smith**, "A Public-Key Infrastructure for Key Distribution in TinyOS based on Elliptic Curve Cryptography", Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004 Page(s): 71 – 80, 4-7 Oct. 2004.

[NG04] **N. Gura et al**, «Comparing Elliptic Curve Cryptography and RSA on 8-bit CPUs» CHES, volume 3156 of Lecture Notes in Computer Science, page 119-132. Springer, (2004)

[NK87] **Neal Koblitz,** "Elliptic curve cryptosystems", Math. Comp. **48** (1987), 203-209**, 1987.**

[NM07] **Nicolas Méloni**:«Arithmétique pour la Cryptographie basée sur les Courbes Elliptiques» Thèse de doctorat, Université Montpellier II, Sciences et Technique du Languedoc, 2007

[ODL07] **Leonardo B. Oliveira, Ricardo Dahab, Julio L'opez,** « Identity-Based Encryption for Sensor Networks », 24 septembre 2007.

[POS08] **Piotr Szczechowiak, Leonardo B . Oliveira, Michael Scott, Martin Collier, and Ricardo Dahab ,**« NanoECC: Testing the Limits of Elliptic Curve Cryptography in Sensor Networks.» EWSN'08 Proceedings of the 5th European conference on Wireless sensor networks, Pages 305-320, 2008.

[PSW01] **Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, JD Tygar,** «SPINS : Security Protocols for Sensor Networks.»Proceedings of the 7th annual international conference on Mobile computing and networking ACM, Pages 521 – 534, Volume 8 Issue 5, September 2002.

[RM08] **D.R. Raymond and S.F. Midki. «**Denial-of-service in wireless sensor networks: Attacks and defenses.» Pervasive Computing, IEEE, 7(1):74-81, jan.-march 2008.

[RZL07] **K. Ren, K. Zeng, W. Lou, and P. Moran.«**On broadcast authentication in wireless sensor networks.» IEEE Trans. Wirel. Commun. , 6(11):4136–4144, 2007.

[SAC04] **Weilian Su, Ozgiir B. Akan, and Erdal Cayirci**:« Communication protocols for sensor networks », Wireless sensor networks, Pages 21 – 50, 2004

[SB11] **Salah-Eddine Benbrahim** «Défense contre l'attaque d'analyse de trafic dans les réseaux de capteurs sans fil (WSN)». Mémoire présenté en vue de l'obtention du diplôme de maitrise ès sciences appliquées. Montréal 2011.

[ASS02] **Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci** «*A Survey on Sensor Networks* », Communications Magazine, IEEE (Volume:40 , Issue: 8), Page(s):102 - 114 , 07 novembre 2002.

[Sch91] **C.P. Schnorr, "**Efficient signature generation for smart card, J. Cryptol." 4 (3) (1991) 161–174.

[SG05] **Samuel GRAU**« Courbes Elliptiques Implémentation de la Signature Électronique», Mémoire pour l'obtention de Master, Faculté des Sciences et des Techniques, Université de Rouen, 2005.

[SKS09] **Piotr Szczechowiak, Anton Kargl, Michael Scott**: «On the Application of Pairing Based Cryptography Wireless Sensor Networks» WiSec '09 Proceedings of the second ACM conference on Wireless network security Pages 1-12, janvier 2009.

[SLP12] **K.-A.Shim, Y-R.Lee**et **C-M.Park** «$\mathcal{E}IBAS$: An efficient identity-based broadcast authentication scheme in wireless sensor networks», Ad Hoc Netw. (2012),http://dx.doi.org/10.1016/j.adhoc.2012.04.015

[SRB05] **Fabricio A. Silva, Linnyer B. Ruiz, Thais R. M. Braga, José Marcos S. Nogueira and Antonio A. F. Loureiro** «Defining aWireless Sensor Network Management Protocol» LANOMS 2005 - 4th Latin American Network Operations and Management Symposium, 2005.

[SS03] **Sencun Zhu, Sanjeev Setia, Sushil Jajodia** «LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks». A conference version was appeared in CCS'03, pages = 62--72, 2003.

[SS06] **Séverine Sentilles**, « Architecture logicielle pour capteurs sans-fil en réseau », Mémoire pour l'obtention de diplôme de Master Technologies de l'Internet, Université de Pau et des Pays de l'Adour, 2006.

[SW05] **A. S. Wander et al**, «Energy Analysis of Public-Key Cryptography for Wireless Sensor Networks», Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on, Page(s): 324 – 328, Mars 2005.

[SWC07] **Selavo L., Wood A., Cao Q., Srinivasan A., Liu H., Sookoor T., Stankovic J.: «**Luster:Wireless sensor network for environmental research.» SenSys '07 Proceedings of the 5th international conference on Embedded networked sensor systems, Pages 103-116, (2007).

[SWS04] **Xingfa Shen, Zhi Wanget Youxian Sun.**« Wireless sensor networks for industrial applications. »In Intelligent Control and Automation, 2004. WCICA

2004. Fifth World Congress on, volume 4, pages 3636 – 3640 Vol.4, Juin 2004.

[VHY09] **Vicaire P., He T., Yan T., Cao Q., Gu G.Z.L., Luo L., Stoleru R., Stankovic J., Abdelzaher T.: «**Achieving long term surveillance in vigilnet.» ACM Transactions on Sensor Networks (TOSN), Volume 5, Issue 1,2009, 1–39, Février 2009.

[TS08] www.techno-science.net , 2008.

[VM85] **Victor S. Miller,** "Use of elliptic curves in cryptography", Advances in Cryptology, CRYPTO'85, pp 417-426, 218, 1985.

[WLS06] **John Paul Walters, Zhengqiang Liang, Weisong Shi, and Vipin Chaudhary** « Wireless Sensor Network Security: A Survey » Distributed, Grid, and Pervasive Computing, Yang Xiao (Eds), pages 0-849, 2006.

[WS02] **Wood A D and Stankovic J A**,«Denial of service in sensor networks,» IEEE Computer, volume :35, 2002, pp. 54-62.

[WSH05] **Chonggang Wang, K. Sohraby, Yueming Hu, Bo Li, and Weiwen Tang**; «Issues of transport control protocols for wireless sensor networks.» In Communications, Circuits and Systems, 2005 International Conference, 422 - 426 Vol. 1, May 2005.

[WSL06] **Haodong Wang, Bo Sheng and Qun Li,** «Elliptic curve cryptography-based access control in sensor networks», International Journal of Security and Networks, Volume 1 Issue ¾, Pages 127-137, Décembre 2006.

[WST07] **Haodong Wang, Bo Sheng, Chiu C. Tan and Qun Li**, M-ECC:«an Elliptic Curve Cryptography Suite on Sensor Motes», College of William & Mary Department of Computer Science, WM-CS-2007-11.

[WSV08] **Wood A., Stankovic J., Virone G., Selavo L., He Z., Cao Q., Doan T., Wu Y., Fang L., Stoleru R.:** «Context-aware wireless sensor networks for assisted living and residential monitoring.» IEEE Network 22(4), 26–33 (2008).

[YC08] **Yacine Challal** « Réseaux de Capteurs Sans Fils', Université de Technologie de Compiègne, Ressource pédagogique, France, 17 Novembre 2008.