Université Abou Bekr Belkaid جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de Fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: *Réseaux et Système Distribués (R.S.D)*

Thème

Contrôle d'accès basé sur les courbes elliptiques pour la sécurité dans les réseaux de capteurs

Réalisé par :

- Mlle. Bensafi Zineb

Présenté le 03 Juillet 2012 devant le jury composé de MM.

- *Mme. Labraoui Nabila* *(Présidente)*
- *Mr. Lehsaini Mohammed* *(Encadreur)*
- *Mr. Benmammar Badr* *(Examinateur)*
- *Mme Didi Fedoua* *(Examinateur)*

Année universitaire : 2011-2012

Table des Matières

Introduction Générale 6

Chapitre I : Présenttion générale des réseaux de capteurs 9

I.1 Introduction 8

I.2 Capteurs 9

II.2.1 Qu'est-ce qu'un Capteur ? 9

II.2.2 Caractéristiques d'un capteur 9

II.2.3 Architecture de base d'un capteur 10

II.2.4 Type de capteurs 11

II.2.5 Systèmes d'exploitation 12

I.3 Les réseaux de capteurs sans fil 13

I.3.1 Objectifs de base des RCSFs 13

I.3.2 Architecture de base d'un RCSF 14

I.3.3 Catégories de réseaux RCSFs 15

I.3.4 Facteurs et contraintes conceptuelles des RCSFs 17

I.3.5 Applications 20

I.4 Conclusion 21

Chapitre II : Sécurité dans les RCSF 23

II.1. Introduction 22

II.2. Buts de la Sécurité 22

II.3. Analyse de vulnérabilité 23

II.4. Contraintes influençant la sécurité dans un RCSF 24

II.5. Les attaques dans les RCSFs 25

II.5.1 Analyse du trafic 25

II.5.2 Brouillage radio 25

II.5.3 Flooding 25

II.5.4 Hello Flooding 26

II.5.5 Nœud compromis ou nœud malicieux 26

II.5.6 Attaque du trou noir (black hole attack) 27

II.5.7 Attaque du trou de ver (Wormhole Attack) 27

II.5.8 Attaque du trou de la base (sink hole attack) 28

II.5.9 Attaque sybile (Sybil attack) 29

II.5.10 Attaque sur les Pacemakers 29

II.6. Issues majeures de sécurité 30
II.6.1 Partitionnement des données 30
II.6.2 Génération 31
II.6.3 Localisation 32
II.6.4 Chien de garde (Watch Dog) 33
II.6.1 Cryptographie 34
a. Le chiffrement 34
b. La signature digitale 35
c. La fonction de hachage 35
d. Le code d'authentification de message MAC 36
II.7. Conclusion 36
Chapitre III : Cryptographie basée sur les courbes elliptiques 38
III.1 Introduction 37
III.2 Présentation des courbes elliptiques 38
III.2.1 Préliminaires 38
III.2.2 Présentation d'une courbe elliptique 39
III.2.1 Exemples de courbes elliptiques 40
III.2.2 Arithmétiques sur les courbes elliptiques 41
III.2.3 Les courbes elliptiques sur un corps fini Fp 44
III.3 Cryptographie basée sur les courbes elliptiques (ECC) 45
III.3.1 Description 45
III.3.2 Le logarithme discret [36] 45
III.3.3 Le logarithme discret elliptique 46
III.3.4 Génération et Échange de clé (Diffie-Hellman) [37] 46
III.3.5 Chiffrement 48
III.3.6 Déchiffrement 48
III.4 Comparaison entre RSA et ECC 48
III.5 Conclusion 49
Chapitre IV : Mécanisme efficace pour l'authentification dans les RCSF 51
IV.1 Introduction 50
IV.2 L'approche de sécurité proposée pour le contrôle d'accès 50
IV.2.1 Environnement 50
IV.2.2 Les étapes de la contribution 51
IV.2.3 Implémentation de l'algorithme 52

a. Fonction de Hachage f () 52
b. MAC (Message Authentification Code) 53
c. Cryptage de la clé avec les ECC 53
IV.3 Les choix techniques 56
IV.3.1 Outils logiciels 56
IV.3.1.1 Le système d'exploitation : TinyOs 56
IV.3.1.2 MIG (Message Interface Generator) 57
IV.3.1.3 Le langage NesC 57
IV.3.1.4 Le langage JAVA 58
IV.3.1.5 La base de données MySQL 59
IV.3.1.6 L'outil de développement : Netbeans 59
IV.3.2 Matériels 59
IV.3.3 Installation 60
IV.4 Fonctionnement de l'application 61
IV.4.1 La maquette 61
IV.4.2 Application côté Envoi : 62
IV.4.3 Application côté réception 64
IV.4.4 Base de données MySQL 65
IV.5 Conclusion 65
Conclusion Générale 66
Références 67

Liste des Figures

Figure I.1: Architecture de base d'un capteur 10
Figure I.2 Evolution des Capteurs 11
Figure I.3 Architecture d'un réseau de capteur 14
Figure I.4 Modèles de communication de données dans un RCSF 16
Figure II.1 Attaque de type HELLO Flooding 26
Figure II.2 Attaque du trou noir 27
Figure II.3 Attaque du trou de ver 28
Figure II.4 Attaque du trou de la base 28
Figure II.5 Routage par partitionnement de données 30
Figure II.6 Détection de nœud malicieux par génération de clés 31
Figure II.7 Localisation du signal avec capteur de beacon 32
Figure II.8 Mécanisme de chien de garde 33
Figure III.1 Représentation de la courbe elliptique $y^2=x^3 - x$ 40
Figure III.2 Représentation de la courbe elliptique $y^2 = x^3 - x + 1$ 41
Figure III.3 Représentation du cas d'addition 1 41
Figure III.4 Représentation du cas d'addition 2 42
Figure III.5 Représentation du cas d'addition 3 42
Figure III.6 Représentation du cas d'addition 4 43
Figure III.7 Processus de génération de clés 47
Figure IV.1 Environnement de travail 51
Figure IV.2 Schéma récapitulatif de la contribution 51
Figure IV.3 Trapèze représenté par les point P,-P,Q-,Q 53
Figure IV.4 Logo de TinyOs 56
Figure IV.5 capteur TelosB 59
Figure IV.6 Environnement générique de travail 60
Figure IV.7 Schéma générale de communication de la maquette 61
Figure IV.8 Interface principale de l'application de déchiffrement 66
Figure IV.9 Table Auth_Capteur 65

Liste des Tableaux

Tableau I-1 Caractéristiques des capteurs les plus utilisés 12
Tableau III-1 Tailles des clés dans RSA et ECC 49
Tableau IV-1 Les points de la courbe dans le corps F29 55

Liste des Abréviations

ADC Analog to Digital Convertor

AES Advanced Encryption Standard

DARPA Defense Advanced Research Projects Agency

DES Data Encryption Standard

DSN Distributed Sensor Network

ECC Elliptic Curve Cryptosystem

GPS Global Positioning System

HMAC Keyed-hash message authentication code

MAC Message authentification code

MD5 Message Digest 5

MIG Generating Packet Objects

NesC Network Embedded System C

PLD Problème du Logarithme Discret

QoS Quality of Service

RCSF Réseau de Capteurs sans fil

RC4 Rivest Cipher 4

RSA Rivest Shamir Adleman

SHA-1 Secure Hash Algorithm

SoS Simple operating system

TinyOs Tiny Operating System

WATS Wide Area Tracking System

WBAN Wireless Body Area Network

Wi-Fi Wireless-Fidelity

WSN Wireless Sensor Network

Introduction Générale

Les communications sans fil connaissent un succès sans cesse croissant depuis leurs créations au sein des communautés scientifiques et industrielles. Grâce aux nouvelles possibilités qu'elle offre, cette technologie a pu s'instaurer comme acteur incontournable dans les architectures réseaux actuelles. Le média hertzien offre en effet des propriétés uniques, qui peuvent être résumées en trois points : la facilité du déploiement, l'ubiquité de l'information et le coût réduit d'installation.

Au fil du temps, le paradigme sans fil n'a pas cessé de croître et il a vu naître diverses architectures dérivées, telles que : les réseaux cellulaires, les réseaux locaux sans fil et autres.

Durant cette dernière décennie, une architecture nouvelle a vu le jour: les réseaux de capteurs sans fil. Ce type de réseaux résulte d'une fusion de deux pôles de l'informatique moderne: les communications sans fil et les systèmes embarqués. Un réseau de capteurs sans fil (RCSF), ou "Wireless Sensor Network" (WSN), est composé d'un ensemble d'unités de traitements embarquées, communiquant via des liens sans fil. Le but général d'un RCSF est la collecte d'un ensemble de paramètres de l'environnement, telles que la température ou la pression de l'atmosphère, afin de les acheminer vers des postes de contrôle distants.

Les RCSFs bénéficient actuellement d'un engouement lié aux nouvelles possibilités qu'ils offrent. Moins chers et plus puissants, ils permettent avec leur capacité d'auto-organisation et leur utilisation à grande échelle, de réaliser des applications jusqu'ici impossible à mettre en place dans de nombreux domaines tels que l'environnement, la domotique, la médecine ou l'armée.

Ainsi, pour les applications militaires ou médicales, le besoin d'apporter une solution de sécurité fiable paraît important voire crucial. Or, les RCSFs sont limités dans leur capacité de calcul et de mémoire à utiliser des méthodes de sécurité traditionnelles comparé aux ordinateurs récents, et surtout avec leur durée de vie restreinte par une batterie non rechargeable dans la plupart des cas. Si dans les autres réseaux et particulièrement dans les réseaux filaires, la solution consiste à augmenter toujours plus la puissance de chiffrement, la problématique posée par la faiblesse de calcul et le besoin d'économiser l'énergie des capteurs amènent à se poser des questions

nouvelles sur les méthodes de sécurité à utiliser. En l'occurrence, ce sont les solutions apportant une sécurité maximale tout en préservant la durée de vie des capteurs, c'est-à-dire en utilisant moins de calcul pour consommer moins d'énergie.

Pour répondre en partie aux problèmes de sécurité, nous cherchons dans ce travail à définir des solutions pour l'authentification dans les RCSF peu coûteuses en énergie qui prennent en compte la relative faiblesse de défense d'un réseau autonome. Pour atteindre cette finalité, nous avons proposé un crypto-système basé sur les concepts des courbes elliptiques. Ce crypto-système permet de réaliser une cryptographie légère qui ne pénalise pas les capteurs en termes d'énergie.

Organisation du mémoire

Ce mémoire est organisé comme suit :

- Dans le chapitre I, nous présentons un préambule sur les réseaux de capteurs, leurs architectures, caractéristiques et leurs applications.
- Dans le chapitre II, nous effectuons une analyse de vulnérabilité, et parlons des buts de la sécurité, des attaques dans les RCSFs et des mesures pour y faire face.
- Ensuite, sont abordés dans le chapitre III les courbes elliptiques, et leurs apports dans le domaine de la sécurité.
- Dans le chapitre IV, nous exposons notre contribution qui porte sur l'authentification et l'intégrité dans les réseaux de capteurs en utilisant les courbes elliptiques.

Enfin, on conclue ce mémoire en résumant les résultats obtenus et on présente quelques perspectives.

Chapitre I
Présentation Générale des Réseaux de Capteurs: Concepts et Applications

I.1 Introduction

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 à un coût raisonnable des composants de quelques millimètres cubes de volume, capables de communiquer et de coopérer entre eux. De ce fait, un nouveau domaine de recherche s'est créé pour offrir des solutions économiquement intéressantes et facilement déployables à la surveillance à distance et au traitement des données dans les environnements hostiles: les réseaux de capteurs sans fil.

Les réseaux de capteurs sans fil sont constitués de nœuds déployés en grand nombre en vue de collecter et transmettre des données vers un ou plusieurs points de collecte appelés "station de base", d'une manière autonome. Ces réseaux ont un intérêt particulier pour les applications militaires, environnementales, domotiques, médicales, et bien sur les applications liées à la surveillance des infrastructures critiques. Ces applications ont souvent besoin d'un niveau de sécurité élevé. Or, de part de leurs caractéristiques (absence d'infrastructure, contrainte d'énergie, topologie dynamique, nombre important de capteurs, sécurité physique limitée, capacité réduite des nœuds,...), la sécurisation des réseaux de capteurs est à la source, aujourd'hui, de beaucoup de défis scientifiques et techniques.

I.2 Capteurs

I.2.1 Qu'est-ce qu'un Capteur ?

C'est un dispositif qui sert à détecter, sous forme de signal souvent électrique, un phénomène physique afin de le représenter format numérique et le communiquer à usager final. Les capteurs sont des petits appareils dotés d'une batterie, capables de communiquer entre eux et de détecter des évènements s'ils se trouvent à l'intérieur de leur rayon de perception.

Un capteur est un petit appareil doté de mécanismes lui permettant de relever des informations sur son environnement. La nature de ces informations varie très largement selon l'utilisation qui est faite du capteur : ce dernier peut tout aussi bien faire des relevés de température, d'humidité ou d'intensité lumineuse. Un capteur possède également le matériel nécessaire pour effectuer des communications sans fil par le biais des ondes radio. Afin que les nœuds capteurs travaillent d'une façon coopérative, les informations recueillies sont partagées entre eux par voie hertzienne. Le choix du lien radio plutôt que du lien filaire permet un déploiement facile et rapide dans un environnement pouvant être inaccessible pour l'être humain [1].

I.2.2 Caractéristiques d'un capteur

Un capteur est doté des caractéristiques suivantes [2] :

- Capable de calculer.
- Capable de communiquer.
- Capte toujours.
- Préposition / déploiement aléatoire.
- Limitation de la durée de vie des batteries.
- Densité (petit / grand nombre).
- La rapidité : c'est le temps de réaction d'un capteur entre la variation de la grandeur physique qu'il mesure et l'instant ou l'information prise en compte par la partie commande.
- L'étendue de la mesure : c'est la différence entre le plus petit signal détecte et le plus grand perceptible sans risque de destruction pour le capteur.
- La sensibilité : c'est la plus petite variation d'une grandeur physique que peut détecter un capteur.

I.2.3 Architecture de base d'un capteur

Un capteur est composé de quatre éléments principaux (Figure I.1) [1] :

- Un élément qui se charge de mesurer l'environnement extérieur (unité de capture),
- Une unité de calcul,
- Un élément émetteur / récepteur,
- Une alimentation.

Trois composants additionnels peuvent être implantés dans un capteur :

- Un système de recherche d'emplacement,
- Un générateur d'alimentation,
- Un mobilisateur (permettant de faire bouger le capteur).

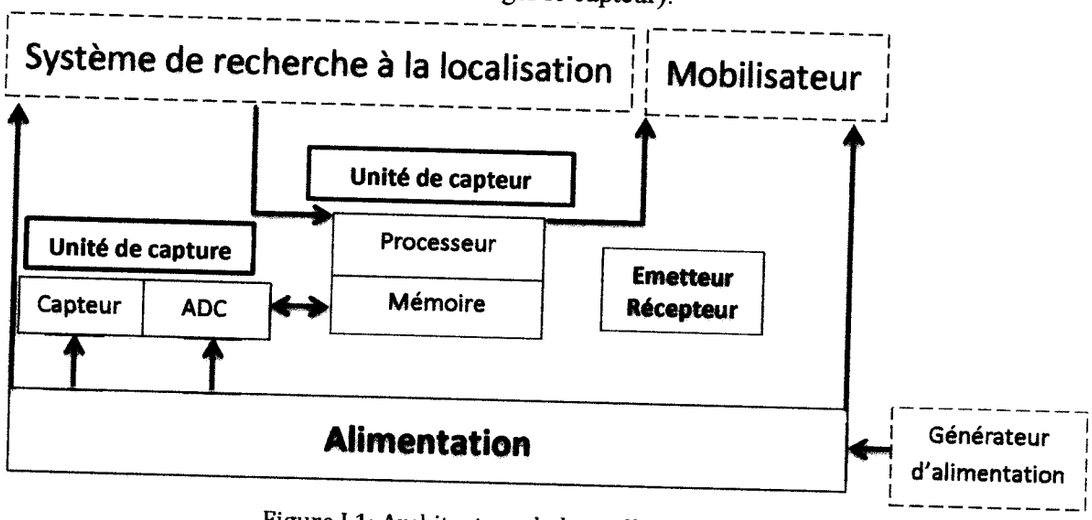

Figure I.1: Architecture de base d'un capteur

- ***Capteur :*** L'unité de capture ou élément capteur est composée de :

- Le capteur récupérant des données analogiques,
- Un convertisseur faisant passer les données analogiques du capteur a des données numériques (appelée ADC : analog to digital convertor) envoyées à une unité de calcul.

- ***Unité de calcul :*** Le composant regroupe :

- Un processeur,
- Mémoire réduite.

Cette unité permet de stocker les données, exécute les taches de perception qui lui sont assignées.

- ***Emetteur/Récepteur*** : composée d'une antenne et de la radio et permet à un nœud capteur de communiquer avec les autres nœuds capteurs.
- ***Alimentation*** : comme tout dispositif embarqué, ils disposent d'une alimentation autonome telle qu'une batterie.

I.2.4 Type de capteurs

Il existe actuellement un grand nombre de capteurs, avec des fonctionnalités diverses et variées. La plupart de ces capteurs dépendent de l'application pour lesquels ils ont été conçus (capteurs aquatiques, sous-terrain, etc...).
La figure I.2 illustre l'évolution des capteurs au cours des années 1996-2008.

Figure I.2 Evolution des Capteurs [3]

Les capteurs fabriqués par Crossbow au cours des dix dernières années (famille de capteurs Mica et Telos) sont les plus utilisés dans les expériences et les travaux de recherche. Ces capteurs sont capables de mesurer plusieurs métriques (température, humidité, luminosité) et la plupart d'entre eux s'articulent autour du Chipcon CC2420 qui est devenu le standard au niveau des modules de transmission utilisant le protocole de communication IEEE 802.15.4.

Tableau I -1 Caractéristiques des capteurs les plus utilisés [4]

Nom Capteur	MCU	RAM	Flash	Stockage	Radio	Dimension
Spec Node (2003)	AVR Risk 8 bit	KB	KB	KB	RF	2 x 2.5 mm
Mica (2002)	ATMega 128	KB	KB	KB	CC1000	58 x 32 x 7 mm
Mica2Dot (2002)	ATMega 128	KB	KB	KB	CC1000	25 x 6 mm
MicaZ (2004)	ATMega 128	KB	KB	KB	CC2420	58 x 32 x 7 mm
TelosA (2004)	TI MSP 430	KB	KB	KB	CC2420	
TelosB (2004)	TI MSP 430	KB	KB	KB	CC2420	65 x 31 x 6 mm
Tmote Sky (2004)	TI MSP 430	KB	KB	KB	CC2420	3.2 x 8 x 1.3 mm
BTnode3 (2004)	ATMega 128	KB	KB	KB	CC1000/	58 .15 x 33 mm
Imote (2003)	ARM7	KB	KB	KB	ZV4002/	
Imote2 (2007)	Intel PXA271	KB	KB	KB	CC2420	36 x 48 x 9 mm
Iris (2008)	ATmega 1281	KB	KB	KB	CC2420	58 x 32 x 7 mm

Le tableau ci-dessus reprend les principales caractéristiques des capteurs de la société Xbow(Crossbow), ainsi que les capteurs les plus utilisés dans le domaine de la recherche.

I.2.5 Systèmes d'exploitation

On fait distinguer plusieurs systèmes d'exploitation qui sont dédiés le plus souvent à un ou plusieurs types de capteurs spécifiques et sont sujets à des révisions fréquentes. Par ailleurs, ils ne sont pas comme les systèmes d'exploitation que l'on retrouve sur ordinateur qui supportent un système de fichier et les possibilités d'exécution multitâches. Ces systèmes d'exploitation permettent de développer un peu plus le caractère intelligent des capteurs. On cite quelques systèmes d'exploitation conçus pour les RCSF :

- TinyOs [5]: système d'exploitation open-source pour les RdCs qui trouve sa genèse au sein de l'université de Berkeley (USA).
- Contiki [6] : système d'exploitation open-source multitâche, développé pour les systèmes embarqués avec contraintes de mémoire.

- SOS [7]: Il est développé par l'université de Los Angeles (USA), écrit en langage C et qui reprend le système de programmation événementielle de TinyOs.
- FreeRTOS [8]: n'est pas un système d'exploitation à proprement dit, mais un noyau de système d'exploitation pour systèmes embarqués.
- Mantis OS [9]: système d'exploitation dédié aux réseaux de capteurs développé par l'université du Colorado (USA) et écrit en langage C. Contrairement à TinyOs qui se base sur un modèle de programmation événementielle, Mantis OS s'articule autour d'un modèle commandé par l'exécution de processus.
- Nut/OS [10]: Système d'exploitation multitâches pour les systèmes embarqués avec une pile TCP/IP.

I.3 Les réseaux de capteurs sans fil

Les réseaux de capteurs sans fil sont considérés comme un type spécial des réseaux ad-hoc ou l'infrastructure fixe de communication et l'administration centralisée sont absentes et les nœuds jouent, à la fois, le rôle des hôtes et des routeurs.

Les nœuds capteurs sont des capteurs intelligents "smart sensors", capables d'accomplir trois tâches complémentaires: le relevé d'une grandeur physique, le traitement éventuel de cette information et la communication avec d'autres capteurs. L'ensemble de ces capteurs, déployés pour une application, forme un réseau de capteurs. Le but de celui-ci est de surveiller une zone géographique, et parfois d'agir sur celle-ci et dans cas il s'agit de réseaux de capteurs-actionneurs [1]. On peut citer comme exemples un réseau détecteur de feu de forêt, ou un réseau de surveillance de solidité d'un pont après un tremblement de terre.

Un RCSF peut comporter un grand nombre de nœuds capteurs (des milliers). Ces derniers sont placés de manière plus ou moins aléatoire (par exemple par largage depuis un hélicoptère) dans des environnements pouvant être dangereux. En outre, toute intervention humaine après le déploiement des nœuds capteurs est la plupart du temps exclue et le réseau devrait donc s'auto-organiser.

I.3.1 Objectifs de base des RCSFs

Les objectifs de base des RCSF dépendent généralement des applications. Cependant, les tâches suivantes sont communes à plusieurs applications:

- Déterminer les valeurs de quelques paramètres suivant une situation donnée. Par exemple, dans un réseau environnemental, on peut chercher à connaitre la température, la pression atmosphérique, la luminosité, et l'humidité relative dans un nombre de sites.
- Détecter l'occurrence des évènements dont on est intéressé à surveiller. Dans les réseaux de contrôle de trafic, on peut vouloir détecter le mouvement de véhicules à travers une intersection et estimer la vitesse et la direction du véhicule.
- Classifier l'objet détecté.

I.3.2 Architecture de base d'un RCSF

L'architecture du réseau de capteurs est montrée dans la figure suivante (Figure I.3).

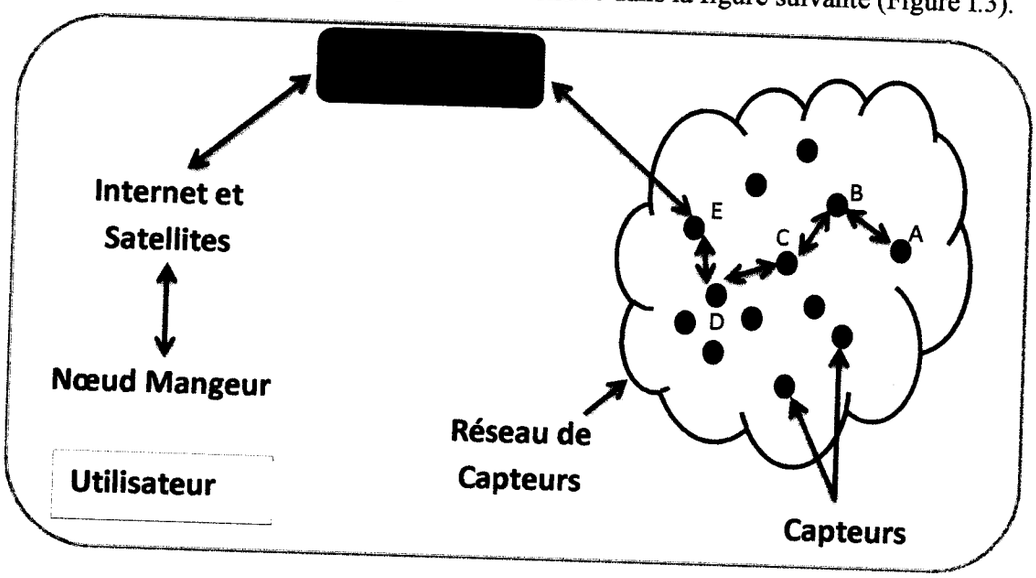

Figure I.3 Architecture d'un réseau de capteur

L'utilisateur accède à distance aux données capturées à travers un nœud appelé le nœud directeur de tâche "Task Manager Node". Ce dernier est relié à un nœud destinataire « puits ou Sink » via une connexion Internet ou par satellite. Il agit en tant que passerelle pour le réseau de capteurs, c'est-à-dire qu'il relie des réseaux de capteurs à d'autres réseaux. 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 requêtes. Il a également d'autre capacité de traitement de l'information pour une transformation ultérieure s'il y a lieu.

Les nœuds capteurs sont habituellement dispersés dans une zone d'intérêt. Ils rassemblent les données et les conduisent au destinataire. De cette manière, les utilisateurs peuvent rechercher l'information dans les nœuds destinataires pour surveiller et commander l'environnement à distance [11].

I.3.3 Catégories de réseaux RCSFs

Les réseaux de capteurs peuvent être classifiés selon plusieurs critères:

A. Classification suivant la taille du réseau

Suivant les caractéristiques des capteurs, la taille du réseau, des protocoles de communications, le Réseau de Capteur est structuré suivant quatre types principaux de plateformes [13] :

- **Plateforme de capteurs miniaturisés**

Plateforme dédiée aux capteurs de taille très réduite (quelques mm3) et de faible bande passante (<50Kbps). Un exemple très connu de ce genre de plateforme est Spec, développé par l'université de Berkeley. Avec une taille très petite (2mmx2.5mm), Spec est un des plus petits capteurs au monde.

- **Plateforme de capteurs généraux**

Plateforme développée pour capter et router des informations du monde ambiant. Quelques plateformes de cette famille ont été développés et la plus récente est basée sur MicaZ, un capteur de taille ~10cm3 avec les protocoles de communication IEEE 802.15.4. Aujourd'hui, MicaZ devient la référence dans les travaux de recherche dans le domaine des réseaux de capteurs.

- **Plateforme de capteurs à haute bande passante**

Ces plateformes ont pour but de transporter de gros volumes de données captées (la vidéo, le son, vibrations). Un exemple typique de cette famille est Imote dont la communication se base sur la norme Bluetooth 1.1.

- **Plateforme de passerelles**

Ces dispositifs servent à transporter les informations envoyées par le réseau de capteurs vers un réseau traditionnel (Ethernet, 802.11) dont Stargate est un exemple typique.

B. Suivant le modèle de transmission de données

On peut catégoriser aussi les capteurs par le modèle de transmission de données. Selon les interactions entre le réseau de capteurs et la station de base, nous nous intéressons à trois modèles principaux :

- **Modèle de mesure périodique**

Tous les capteurs envoient périodiquement leurs mesures à la station de base. Le type d'application visé concerne les applications de type "surveillance" où le but principal est d'avoir une information régulière de la zone surveillée.

- **Modèle de détection d'évènements**

Les capteurs envoient les mesures seulement lorsqu'il y a un évènement qui se produit. Ce type de modèle est recommandé pour les applications de surveillance

d'évènements critiques où le but principal est l'obtention d'une information sur l'évènement le plus rapidement possible.

- **Modèle de transmission suite à des requêtes**

Les capteurs mesurent des phénomènes et stockent ces mesures dans leur mémoire flash. Ils envoient ces mesures seulement lorsqu'ils reçoivent des requêtes de la base. Ce modèle peut également s'apparenter aux applications de type surveillance mais les capteurs utilisent une mémoire flash importante afin de stocker les mesures localement.

C. Selon modèles de communication dans les réseaux de capteurs

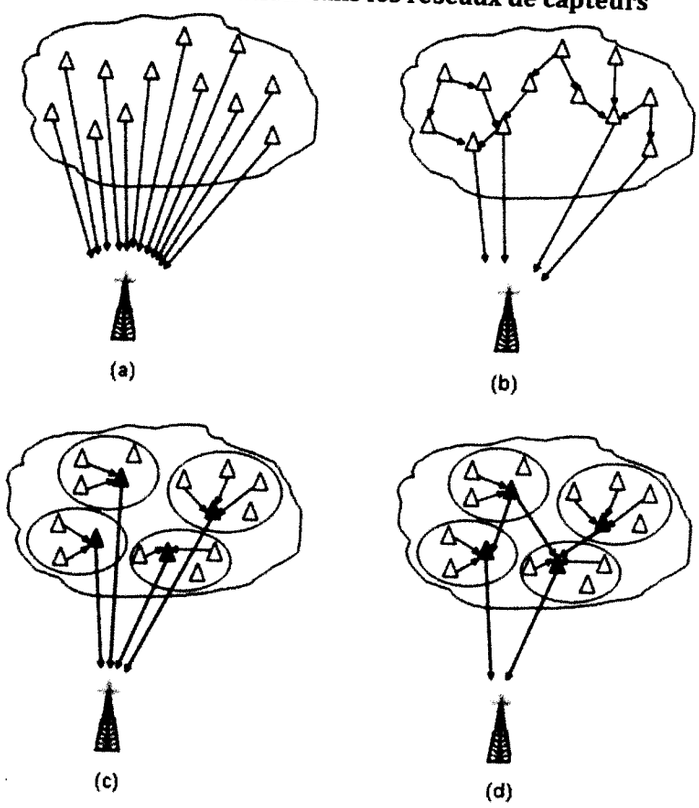

Figure I.4 Modèles de communication de données dans un RCSF [12]

Les architectures dans les réseaux de capteurs dépendent des applications et des techniques utilisées pour faire acheminer l'information des capteurs à la station de base. Une taxonomie des applications peut être dérivée et l'adaptabilité d'algorithmes à ce genre de scénario peut être évaluée [12].

Le processus d'acheminement de l'information des capteurs à la station de base peut prendre quatre formes. Dans les architectures à plat, les capteurs peuvent communiquer :

- Directement avec la station de base en utilisant une forte puissance (figure 1.4(a)),
- Via un mode multi-sauts avec des puissances très faibles (figure 1.4 (b)),
- Via un cluster-head (le nœud représentant le cluster), qui transmet directement les données à la station de base (figure 1.4 (c)),
- Via un mode multi-saut entre les cluster-heads (figure 1.4 (d)).

I.3.4 Facteurs et contraintes conceptuelles des RCSFs

La conception des réseaux de capteurs est influencée par de nombreux facteurs comme la tolérance aux pannes, les coûts de production, la consommation d'énergie, l'environnement ou la topologie du réseau. Ces facteurs représentent la base de la conception de protocoles ou d'algorithmes pour les réseaux de capteurs [14] :

- **Tolérance aux pannes**

Les nœuds peuvent être sujets à des pannes dues à leur fabrication (ce sont des produits de série bon marché, il peut donc y avoir des capteurs défectueux) ou plus fréquemment à un manque d'énergie.

Le réseau doit être capable de maintenir ses fonctionnalités sans interruption en cas de défaillance d'un de ses capteurs. Cette défaillance peut être causée par une perte d'énergie, dommage physique ou interférence de l'environnement. Le degré de tolérance dépend du degré de criticité de l'application et des données échangées.

Un premier défi sera donc d'identifier et de modéliser formellement les modes de défaillances des capteurs, puis de repenser aux techniques de tolérance aux fautes à mettre en œuvre sur le terrain. La fiabilité RK(t) d'un nœud de capteur est modelée en [15] par une distribution de Poisson pour capturer la probabilité de ne pas avoir un échec dans un intervalle de temps (0 ; t) :

$$RK(t) = \exp(\lambda K t)$$

Ou : λK est le taux de défaillance d'un nœud de capteur k et t c'est la période de temps.

- **Coût de fabrication**

Les nœuds sont des produits fabriqués en série du fait de leur grand nombre. Il faut que le coût de fabrication de ces nœuds soit tel que le coût global du réseau ne soit pas supérieur à celui d'un réseau classique afin de pouvoir justifier son intérêt.

- **L'échelle**

Une des caractéristique des RCSFs est qu'ils peuvent contenir des centaines voire des milliers de nœuds capteurs. Le réseau doit être capable de fonctionner avec ce

nombre de capteurs tout en permettant l'augmentation de ce nombre et la concentration (densité) des nœuds dans une région (pouvant dépasser 20 nœuds/m3).
Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales (implémentation d'une détection d'erreur, d'un contrôle de flux,..) et nécessite que le puits soit équipe de beaucoup de mémoire pour stocker les informations reçues.

- **L'environnement**

Les capteurs sont souvent déployés en masse dans des endroits tels que des champs de bataille au-delà des lignes ennemies, à l'intérieur de grandes machines, au fond d'un océan, dans des champs biologiquement ou chimiquement souillés,... Par conséquent, ils doivent pouvoir fonctionner sans surveillance dans des régions géographiques éloignées.

- **Topologie du réseau**

En raison de leur forte densité dans la zone à observer, il faut que les nœuds-capteurs soient capables d'adapter leur fonctionnement afin de maintenir la topologie souhaitée.

On distingue généralement trois phases dans la mise en place et l'évolution d'un réseau :

- **Déploiement**

Les nœuds sont soit répartis de manière prédéfinie soit de manière aléatoire (lancés en masse depuis un avion). Il faut alors que ceux-ci s'organisent de manière autonome.

- **Post-Déploiement - Exploitation**

Durant la phase d'exploitation, la topologie du réseau peut être soumise à des changements dus à des modifications de la position des nœuds ou bien à des pannes.

- **Redéploiement**

L'ajout de nouveaux capteurs dans un réseau existant implique aussi une remise à jour de la topologie.

- **Les contraintes matérielles**

La principale contrainte matérielle est la taille du capteur, et la consommation d'énergie par ses composants qui doit être moindre pour que le réseau survive le plus longtemps possible. Il devrait aussi s'adapter aux différents environnements (fortes chaleurs, eau,..), et qu'il soit autonome et très résistant vu qu'il est souvent déployé dans des environnements hostiles.

- **Les médias de transmission**

Dans un réseau de capteurs, les nœuds sont reliés par une architecture sans fil. Pour permettre des opérations sur ces réseaux dans le monde entier, le média de

transmission doit être normé. On utilise le plus souvent les communications ZigBee (qui consomme moins d'énergie), l'infrarouge et le bluetooth.

- **Consommation d'énergie**

L'économie d'énergie est une des problématiques majeures dans les réseaux de capteurs. En effet, la recharge des sources d'énergie est souvent trop coûteuse et parfois impossible. Il faut donc que les capteurs économisent au maximum l'énergie afin de pouvoir fonctionner.

Les réseaux de capteurs fonctionnant selon un mode de routage par saut, où chaque nœud du réseau joue un rôle important dans la transmission de données. Le mauvais fonctionnement d'un nœud implique un changement dans la topologie et impose une réorganisation du réseau.

- **Sécurité physique limitée**

Les RCSF sont plus touchés par le paramètre de sécurité que les réseaux filaires classiques. Cela se justifie par les contraintes et limitations physiques qui font que le contrôle des données transférées doit être minimisé.

- **La connectivité**

Un réseau de capteurs est dit connecté si et seulement si, il existe au moins une route entre chaque paire de nœuds. La connectivité dépend essentiellement de l'existence des routes. Elle est affectée par les changements de topologie dus à la mobilité, la défaillance des nœuds, les attaques, etc… Ce qui a pour conséquences : la perte de liens, l'isolement des nœuds, le partitionnement du réseau, la mise a jours des routes (le routage), etc...

Un graphe G est dit k-connecte, s'il y a au moins k chemins disjoints entre deux nœuds quelconques. La connectivité est une mesure de tolérance aux fautes ou de diversité de chemin dans le réseau. Le 1-connectivite est une condition fondamentale pour que le réseau soit opérationnel. En effet la connectivité d'un réseau s'exprime de la façon suivante [1]:

$$\mu(R) = \frac{N.\pi.R2}{A}$$

Ou :

- R est le rayon de transmission d'un nœud.
- A l'aire de calcul.
- N le nombre de nœuds situe dans l'aire A.

I.3.5 Applications

Plusieurs applications envisagées dans les réseaux de capteurs font toujours l'objet d'une recherche et d'un développement actifs universitaires ou industriels. On trouve des applications pour la détection et la surveillance des désastres, le contrôle de l'environnement et la cartographie de la biodiversité, le bâtiment intelligent, la surveillance et la maintenance préventive des machines, la médecine et la santé, la logistique et la télématique, etc. Nous décrivons ici brièvement quelques applications dans certains domaines qui nécessitent un niveau de sécurité [16].

a. Applications militaires

Les premières applications potentielles des réseaux de capteurs ont concerné le domaine militaire. L'idée était de déployer un réseau de capteurs invisible sur des champs de bataille ou des zones ennemies pour surveiller le mouvement des troupes. Historiquement, Le projet DSN (Distributed Sensor Network) [17] au DARPA (Defense Advanced Research Projects Agency) était l'un des premiers projets dans les années 80 ayant utilisés les réseaux de capteurs pour rassembler des données distribuées.

Les applications militaires sont les premières et certainement les plus représentatives des applications trouvées actuellement dans le domaine des réseaux de capteurs sans fil. Les chercheurs du laboratoire national Lawrence Livermore ont mis en place le réseau WATS (Wide Area Tracking System) [18]. Ce réseau est composé de détecteurs des rayons gamma et des neutrons pour détecter et dépister les dispositifs nucléaires. Il est capable d'effectuer la surveillance constante d'une zone d'intérêt. Ces chercheurs ont mis en place ensuite un autre réseau appelé JBREWS (Joint Biological Remote Early Warning System) [19] pour avertir les troupes dans le champ de bataille des attaques biologiques possibles. Ces applications nécessitent d'être sécurisées, pour éviter toute attaque possible des ennemis.

b. Applications à la sécurité

L'application des réseaux de capteurs dans le domaine de la sécurité peut diminuer considérablement les dépenses financières consacrées à la sécurisation des lieux et des êtres humains. Ainsi, l'intégration des capteurs dans de grandes structures telles que les ponts ou les bâtiments aidera à détecter les fissures et les altérations dans la structure suite à un séisme ou au vieillissement de la structure. Le déploiement d'un réseau de capteurs de mouvement peut constituer un système d'alarme qui servira à détecter les intrusions dans une zone de surveillance.

c. Applications environnementales

Le contrôle des paramètres environnementaux par les réseaux de capteurs peut donner naissance à plusieurs applications. Par exemple, le déploiement des thermo-capteurs dans une forêt peut aider à détecter un éventuel début de feu et par suite faciliter la lutte contre les feux de forêt avant leur propagation. Le déploiement des capteurs chimiques dans les milieux urbains peut aider à détecter la pollution et analyser la qualité d'air. De même leur déploiement dans les sites industriels empêche les risques industriels tels que la fuite de produits toxiques (gaz, produits chimiques, éléments radioactifs, pétrole, etc)

d. Applications médicales

Dans le domaine de la médecine, les réseaux de capteurs peuvent être utilisés pour assurer une surveillance permanente des organes vitaux de l'être humain grâce à des micro capteurs qui pourront être avalés ou implantés sous la peau (surveillance de la glycémie, détection de cancers, ..). Ils peuvent aussi faciliter le diagnostic de quelques maladies en effectuant des mesures physiologiques telles que : la tension artérielle, battements du cœur, ... à l'aide des capteurs ayant chacun une tâche bien particulière.

Les données physiologiques collectées par les capteurs peuvent être stockées pendant une longue durée pour le suivi d'un patient [20]. D'autre part, ces réseaux peuvent détecter des comportements anormaux (chute d'un lit, choc, cri, ...) chez les personnes dépendantes (handicapées ou âgées). Dans ces applications, l'intégration de la sécurité est indispensable, car si un attaquant arrive à altérer les données physiologiques d'un patient, il pourra mettre en danger sa vie.

I.4 Conclusion

Les réseaux de capteurs restent une nouvelle technologie peu accessible au grand publique. Elle est principalement répandue dans les laboratoires de recherches.
Des progrès sont encore à réaliser dans ce domaine. Néanmoins ils correspondent à une certaine vision du futur et permettront des améliorations dans d'innombrables domaines de la vie quotidienne.

Chapitre II
Sécurité dans les Réseaux de Capteurs

II.1. Introduction

La sécurité est un domaine très important pour les RCSFs, particulièrement pour des applications sensibles du domaine militaire, médicale, et autres. La sécurité intervient pour certaines fonctions sensibles telles que l'expédition des paquets, le cheminement et la gestion d'un réseau, fonctions effectuées par certains ou tous les nœuds disponibles dans les RCSF.

Les données circulant doivent être garanties correctes et valides. Il est primordial de pouvoir s'assurer que l'information n'a pas été altérée et émane effectivement de la source considérée.

II.2. Buts de la Sécurité

II.2.1 Disponibilité

La disponibilité donne une assurance sur la réactivité et le temps de réponse d'un système pour transmettre une information d'une source à la bonne destination. Cela signifie aussi que les services du réseau sont disponibles aux parties autorisées si nécessaire et assure les services de réseau en dépit des attaques de déni de service (DoS) pouvant affecter n'importe quelle couche du réseau.

II.2.2 Authentification

Un adversaire n'est pas simplement limité à modifier le paquet de données mais il peut également injecter des paquets supplémentaires. Ainsi le récepteur doit s'assurer que les données utilisées proviennent de la source correcte.

D'autre part, en construisant le réseau de capteurs, l'authentification est nécessaire pour beaucoup de tâches (transmissions des mesures prélevées vers la station de base, synchronisation,...).

II.2.3 Contrôle d'accès

Un service très important consiste à empêcher un accès au réseau à tout élément étranger au système. Le contrôle d'accès donne aux participants légitimes un moyen de détecter les messages provenant de sources externes au réseau.

II.2.4 Intégrité des données

C'est un service qui garantit que les données n'ont pas été altérées pendant la transmission. On peut distinguer les altérations accidentelles liées par exemple, à une mauvaise couverture des ondes, et les altérations volontaires d'un attaquant. Cela Concerne aussi la protection contre l'injection ou la modification des paquets.

II.2.5 Confidentialité

La confidentialité est la garantie que l'information d'un nœud n'est rendue accessible ou révélée qu'à son destinataire. Dans notre cadre, il est important qu'aucun capteur étranger au système ne puisse être mis à proximité dans l'intention de surveiller les informations échangées.

II.2.6 Fraicheur

Ce critère concerne la fraicheur de données et la fraicheur des clés. Puisque tous les réseaux de capteurs fournissent quelques formes de mesures variables dans le temps, nous devons assurer que chaque message est frais. La fraicheur de données implique que les données soient récentes, et elle assure qu'aucun adversaire n'a rejoué les anciens messages.

II.2.7 Non répudiation

Mécanisme destiné à prévenir que la source ou la destination désavoue ses actions ou nie qu'un échange ait eu lieu.

II.3. Analyse de vulnérabilité

Quelques faiblesses sont inhérentes à la nature des RCSF et d'autres à la technologie retenue pour leur mise en œuvre et leur déploiement. Nous distinguons deux catégories de précarité: la vulnérabilité physique et la vulnérabilité technologique.

II.3.1 Vulnérabilité physique

La vulnérabilité physique est le fait qu'un capteur est fréquemment installé dans un lieu peu sur, c.-à-d. dont l'accès n'est nullement restreint. Nous pouvons citer les lieux publics, les environnements naturels (forets, régions montagneuses) ainsi que les bâtiments, maisons intelligentes et musées ("smart environment").

II.3.2 Vulnérabilité technologique

La vulnérabilité est liée à la technologie sans-fil sous-jacente, quiconque possédant un récepteur adéquat peut potentiellement écouter ou perturber les messages échanges. Les mécanismes de routage sont d'autant plus critiques dans les RCSF que chaque nœud participe à l'acheminement des paquets à travers le réseau.

II.4. Contraintes influençant la sécurité dans un RCSF

Des contraintes parfois strictes et intrinsèques aux RCSFs imposent de penser à une sécurité mieux adaptée que son équivalent traditionnel des réseaux filaires.

II.4.1 Puissance d'énergie basse

L'énergie des capteurs est limitée, et généralement irremplaçable. Les réseaux ad-hoc visent à réaliser une haute qualité de service (QoS) tels que minimiser le temps d'attente et la réservation de débit, alors que les protocoles des réseaux de capteur doivent se concentrer principalement sur la conservation d'énergie.

La puissance supplémentaire consommée par les nœuds due à la sécurité est liée :

- Au calcul requis pour les fonctions de sécurité, tels que le chiffrement, déchiffrement, signature des données, vérification de la signature.
- A l'énergie requise pour la transmission des données de sécurité (par exemple, vecteurs d'initialisation requis pour le chiffrement/déchiffrement).
- A l'énergie requise pour le stockage des paramètres de sécurité, tel que le stockage des clés de chiffrement/déchiffrement.

II.4.2 La communication sans fils multi-sauts

La communication multi sauts est indispensable pour la diffusion des données dans un RCSF. Cela introduit de nombreuses failles de sécurité : attaque de la construction et maintenance des routes, et attaque des données utiles par injection, la modification ou la suppression de paquets.

II.4.3 Espace mémoire et capacité de calcul limités

Dans la majorité des RCSFs les nœuds n'ont pas la capacité de mémoriser des clés de taille importante ou d'exécuter des protocoles cryptographique complexes.

II.5. Les attaques dans les RCSFs

Une variété d'attaques contre les RCSF est rapportée dans la littérature spécialisée. Pour faire face à ces attaques, diverses contre-mesures ont été proposées.
Nous présentons dans la suite les principaux types d'attaques et dans la section 5 les issues majeurs de sécurité dans les RCSF.

Dans les cas d'attaques que l'on retrouve dans les RdCs, un attaquant peut chercher à récupérer les informations du réseau en écoutant le médium, si le réseau n'encrypte pas ses données. Dans ce cas de figure, on parlera d'attaque passive. Dans le cas où l'attaquant cherche à modifier ou supprimer des informations, ou bien encore à empêcher le réseau de fonctionner correctement, on parlera d'attaque active.

II.5.1 Analyse du trafic

L'analyse du trafic est une attaque qui met en jeu des mécanismes d'écoute passive et de surveillance du réseau. L'attaquant en analysant uniquement les chemins empruntés par les paquets sur le réseau pourra récupérer des informations précieuses sur les vulnérabilités de ce réseau.

Analyser le trafic peut permettre à un attaquant de connaître la position des nœuds d'agrégation de données ou des bases du réseau en repérant les lieux où le plus grand nombre de paquets transitent.

II.5.2 Brouillage radio

Le médium de transmission des informations est un point vulnérable dans un réseau. En l'occurrence, il est quasiment impossible de restreindre l'accès à un médium utilisant des ondes radio. Un attaquant peut donc envoyer des ondes sur la même fréquence que le RCSF pour brouiller les ondes radio. Les nœuds du réseau n'ont alors plus accès au médium et ne peuvent plus communiquer du fait de ce brouillage radio. Or un réseau sans accès au médium est un réseau hors service.

II.5.3 Flooding

Dans une attaque de type flooding, un attaquant utilise un ou plusieurs nœuds malicieux ou un dispositif particulier avec dans certains cas une puissance d'émission forte, pour envoyer régulièrement des messages sur le réseau afin de le saturer. On est en présence d'une attaque active qui est de même type que les attaques de type déni de service dans les réseaux classiques.

II.5.4 Hello Flooding

Les protocoles de découverte dans les réseaux ad-hoc utilisent des messages de type "HELLO" pour découvrir ses nœuds voisins et pour s'insérer dans un réseau.
Dans une attaque dite de HELLO Flooding, un attaquant utilise ce mécanisme pour consommer l'énergie des capteurs et empêcher leurs messages d'être routés.
Dans [21], on trouve un exemple, représenté par la figure II.1, d'un nœud malicieux X avec une connexion radio puissante qui lui permet d'envoyer à un grand nombre de nœuds des messages de type "HELLO", de manière continue. Les nœuds voisins N1(X) (ensemble des nœuds voisins de X) vont alors considérer le nœud malicieux comme un voisin, même s'ils sont situés à des distances qui ne permettent pas de l'atteindre. Lorsqu'ils chercheront à envoyer des données, les nœuds de N1(X) vont essayer de passer par le nœud X qu'ils le considèrent comme leur voisin, mais leurs messages ne pourront jamais l'atteindre. Comme X est inaccessible, ils vont utiliser leur antenne radio au maximum de sa puissance, consommant alors plus d'énergie, et leurs messages ne seront jamais transmis car jamais reçus par le nœud X.

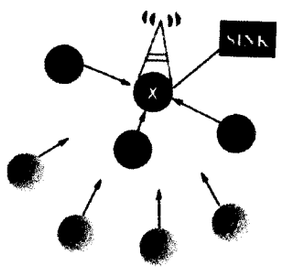

Figure II.1 Attaque de type HELLO Flooding [21]

II.5.5 Nœud compromis ou nœud malicieux

Comme expliqué précédemment, la plupart des réseaux de capteurs sont déployés dans des zones larges ou exsangues qui ne permettent pas une surveillance humaine de l'intégrité de l'ensemble des capteurs. Il est alors tout à fait possible pour un attaquant de s'en prendre physiquement à un capteur pour le compromettre. Cette attaque physique peut permettre à un attaquant d'extraire par exemple les clés cryptographiques contenues dans le capteur, modifier ces circuits électroniques ou modifier le programme qu'il contient pour le remplacer par un autre, afin que le capteur devienne ce que l'on appelle un nœud compromis ou nœud malicieux (malicious node).

Ce nœud malicieux contrôlé par l'attaquant va lui permettre de s'intégrer au réseau, de récupérer des informations ou de lancer d'autres attaques à partir de ce noud ou

plusieurs de ces nœuds comme décrit dans les attaques suivantes. Des travaux de recherche récents ont pour objectif de créer des capteurs résistants aux attaques physiques avec des mécanismes tels que la suppression des clés cryptographiques lors de la détection d'une atteinte physique du capteur.

II.5.6 Attaque du trou noir (black hole attack)

L'attaque du trou noir ou black hole attack consiste tout d'abord à insérer un nœud malicieux dans le réseau [21]. Ce nœud, par divers moyens, va modifier les tables de routage pour obliger le maximum de nœuds voisins à faire transiter leurs informations par lui. Ensuite tel un trou noir dans l'espace, toutes les informations qui vont passer en son sein ne seront jamais retransmises.

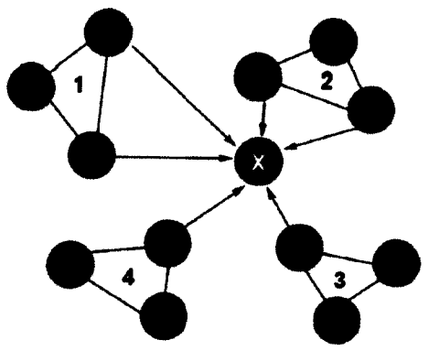

Figure II.2 Attaque du trou noir

La figure II.2 représente un trou noir mis en place par un nœud malicieux X qui a modifié le routage pour que les clusters 1, 2, 3 et 4 fassent passer l'information par lui pour communiquer entre clusters. Dans ce cas de figure, le trou noir X ne retransmettra aucune information, empêchant toute communication entre les différents clusters.

II.5.7 Attaque du trou de ver (Wormhole Attack)

L'attaque du trou de ver nécessite l'insertion dans le réseau de capteurs d'au moins deux nœuds malicieux [22]. Ces deux nœuds sont reliés entre eux par une connexion puissante qui peut être filaire ou radio (connexion Wifi par exemple de plus grande portée). Le but de cette attaque est de tromper les nœuds voisins sur les distances les séparant. Généralement le protocole de routage cherche le chemin le plus court en nombre de sauts (appelé généralement nombre de hop). Dans le cas d'une attaque du trou de ver, les deux nœuds malicieux permettent d'atteindre un lieu éloigné en un saut unique. Cette possibilité va tromper les autres nœuds sur les distances réelles qui

séparent les deux nœuds, mais va surtout avoir pour conséquence que les nœuds voisins vont principalement passer par ces nœuds malicieux pour faire circuler leurs informations. Ainsi les nœuds malicieux qui forment le trou de ver vont se trouver dans une position privilégiée qui va leur permettre d'avoir une priorité sur l'information circulant à travers leurs nœuds proches.

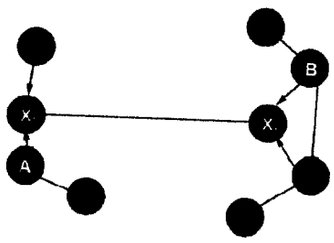

Figure II.3 Attaque du trou de ver

Cette attaque est représentée par la figure II.3 où deux nœuds malicieux X1 et X2, reliés par une connexion puissante, forment un trou de ver.

Une variante de l'attaque du trou de ver consiste à répliquer dans une zone du réseau des paquets d'un capteur, enregistrés dans une zone éloignée. Le but est de tromper les nœuds par rapport à leurs voisins respectifs.

En reprenant l'exemple de la figure II.3, un paquet du nœud A va être enregistré par le nœud X1 puis répliqué par le nœud X2 dans une zone plus éloignée du nœud A. Le nœud B va alors considérer le nœud A comme un de ces nœuds voisins dans sa table de routage. Celle-ci sera alors tronquée.

II.5.8 Attaque du trou de la base (sink hole attack)

Dans cette attaque un nœud malicieux va s'attaquer directement à l'information circulant par et vers la base, qui est le dispositif qui recueille le plus d'informations de l'intégralité du réseau [21]. Pour cela, le nœud malicieux va proposer aux nœuds du réseau le chemin le plus rapide pour atteindre la base, en utilisant par exemple une connexion plus puissante, comme représenté dans la figure II.4.

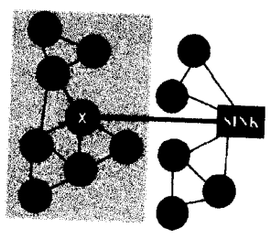

Figure II.4 Attaque du trou de la base

Ainsi l'ensemble de ces nœuds va s'adresser en particulier à ce nœud malicieux pour transmettre l'information à la base. Toute information qui transite de ces nœuds vers la base par ce nœud sera récupérée par l'attaquant.

II.5.9 Attaque sybile (Sybil attack)

Une attaque sybile ou "Sybil attack" [23] consiste à ce qu'un capteur malicieux se fasse passer pour plusieurs capteurs en utilisant l'identité d'autres capteurs légitimes du réseau.

L'attaque sybile va alors pouvoir tenter de mettre en péril les mécanismes comme l'agrégation des données, la sécurité, le routage, l'allocation de ressource ou la détection d'intrus. Un nœud malicieux qui peut se faire passer pour plusieurs nœuds peut gagner un avantage important pour une élection de cluster head. Fort d'un nombre de votes plus important, il pourra tromper ses nœuds voisins pour par exemple inciter le cluster à l'élire comme clusterhead. Si le nœud malicieux obtient cette distinction, ses décisions au sein du cluster auront une incidence plus forte (refus de routage des informations en dehors du cluster, envoi d'information tronquée sur les clusters voisins, etc...).

Par ailleurs pour la détection d'intrus dans le réseau, avec des mécanismes comme le chien de garde, une attaque sybile peut aussi lui permettre de devenir le chien de garde. L'attaquant pourra alors dire détecter des nœuds légitimes comme des nœuds malicieux et vice-versa.

II.5.10 Attaque sur les Pacemakers

L'attaque sur les pacemakers[1] est un cas particulier, puisqu'il n'y a pas de réseau à proprement parlé dans ce dispositif. Cependant c'est un parfait exemple d'attaque spécifique au type de capteur et qui montre la dangerosité d'une absence de sécurité pour des capteurs de type médical.

Il est facilement réalisable de modifier à distance les réglages du pacemaker d'un patient avec un matériel à moindre frais. Les pacemakers traditionnels ne sont équipés d'aucune protection (processus authentification ou chiffrement de l'information) concernant la connexion entre le pacemaker et le dispositif de réglage. Ils en ont déduit qu'il était possible grâce à cette faille de provoquer un arrêt cardiaque sur les personnes

[1] Un pacemaker est un dispositif qui permet d'aider un patient à réguler les battements de son cœur. Ce dispositif peut envoyer via une connexion sans fil l'état des pulsations et être paramétrable à distance.

porteuses de pacemakers qui pourra entraîner la mort sans intervention rapide sur le patient.

II.6. Issues majeures de sécurité

Les attaques présentées précédemment montrent l'étendue des possibilités d'attaque d'un réseau de capteurs et les différents points de sécurité (intégrité du réseau, authentification, confidentialité des données, etc...).

Il s'avère impossible de trouver une seule solution pour contrer l'ensemble de ces attaques. Néanmoins la recherche récente cherche à trouver diverses solutions basées sur de nombreux mécanismes de sécurité pour répondre à ces attaques. La partie suivante décrit et critique quelques-unes des solutions de sécurité les plus répandues :

II.6.1 Partitionnement des données

[24] propose une solution pour empêcher la récupération d'information dans les réseaux de capteurs sans fil par le partitionnement des données. Comme son nom l'indique, le but est de découper l'information en plusieurs paquets. Chaque paquet sera ensuite envoyé sur des chemins différents, c'est à dire qu'elles ne passeront pas par la même route et donc les mêmes nœuds. Ces paquets seront finalement reçus par la base, qui pourra ensuite les rassembler pour pouvoir reproduire l'information. Ce mécanisme oblige un attaquant à écouter tout le réseau pour pouvoir récupérer l'ensemble des paquets qui circulent sur les différents chemins s'il veut pouvoir lire l'information. La figure II.5 illustre le principe de fonctionnement de cette solution, où un capteur A divise un message en 3 paquets qui vont suivre respectivement 3 chemins différents.

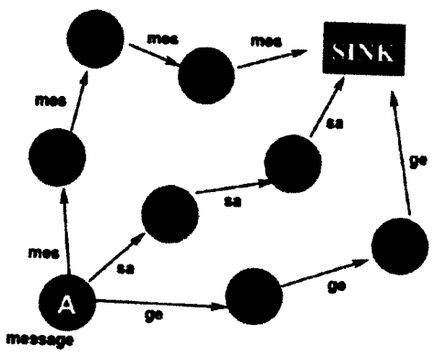

Figure II.5 Routage par partitionnement de données

Le problème de cette solution réside dans le fait qu'elle augmente considérablement la consommation énergétique du réseau. Ainsi, si un message parcourt 3 chemins différents dans le réseau, il y aura plus de capteurs à impliquer dans cette opération.

D'autre part, cette solution devient obsolète si l'un attaquant arrive à se positionner au niveau de la station de base de récupérer l'intégralité des messages.

II.6.2 Génération

Une solution proposée par [25] consiste à utiliser une clé de génération. A chaque période dite génération, la base envoie une nouvelle clé à l'ensemble du réseau. Cette clé sert de certificat à chacun des nœuds, pour prouver son appartenance au réseau. Si un nœud non identifié tente de rentrer dans le réseau de capteurs sans fil et qu'il ne possède pas cette clé de génération, il ne pourra être accepté en son sein. Un autre intérêt de cette technique est qu'elle permet de limiter les attaques de substitution d'un capteur et de sa reprogrammation pour être réinjecté dans le réseau.

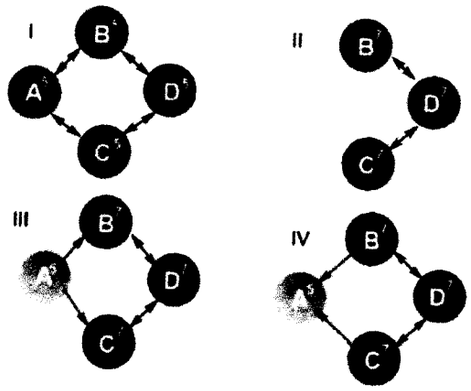

Figure II.6 Détection de nœud malicieux par génération de clés

La figure II.6 présente une solution de protection basée sur la génération de clés, où quatre capteurs A, B, C, D font partie d'un réseau de capteurs qui communiquent par clés symétriques par paire de nœuds.

A l'étape I, les capteurs ont pour clé de génération K_5. A l'étape II, le nœud A est subtilisé par un attaquant, et pendant son absence sur le réseau, la station de base transmet une nouvelle clé de génération K_7. A l'étape III, le capteur A reprogrammé est réinséré dans le réseau et fait une demande d'insertion dans le réseau. A l'étape IV, les nœuds voisins B et C refusent la demande de A, car en comparant leurs clés, ils se sont aperçus qu'elles ne correspondaient pas.

Cette solution détecte effectivement les nœuds malicieux, mais elle n'est pas non plus sans faille. D'une part, elle nécessite que la distribution des clés de génération sur le réseau soit sécurisée et qu'un attaquant ne soit pas capable de les récupérer.

D'autre part, cette solution suppose que tous les capteurs du réseau soient déployés une fois pour toute et ne laisse pas la possibilité de découverte de nouveaux capteurs, donc elle ne peut pas être adaptable aux RdCs mobiles.

En plus, les capteurs sont des éléments sujets à des pannes qui peuvent les empêcher d'être toujours opérationnels. Dans le cas où un tel disfonctionnement interviendrait sur un capteur l'empêchant pendant le laps de temps de la distribution des clés de recevoir la nouvelle clé, celui-ci serait exclu du réseau malgré qu'il soit un capteur sain.

II.6.3 Localisation

Utilisé particulièrement pour détecter les attaques de type trou de ver, consiste à utiliser une technique de localisation géographique, comme proposé par [26]. Pour cette solution, le réseau de capteurs doit être équipé de capteurs dits balises, qui sont des capteurs qui connaissent leur position géographique, par exemple au moyen d'un GPS. Avec la localisation, si un capteur demande d'être inséré dans le réseau, les capteurs balises qui vont recevoir cette demande vont pouvoir estimer sa localisation par rapport à son domaine d'écoute.

Les capteurs balises vont ensuite quadriller leur zone d'écoute respective (figure IV.7), et chaque nœud qui a reçu la demande d'insertion dans le réseau votera pour une zone du quadrillage qu'il est capable d'entendre. La zone qui obtiendra le plus grand nombre de voix sera considérée comme la zone où est censé se trouver le nouveau capteur.

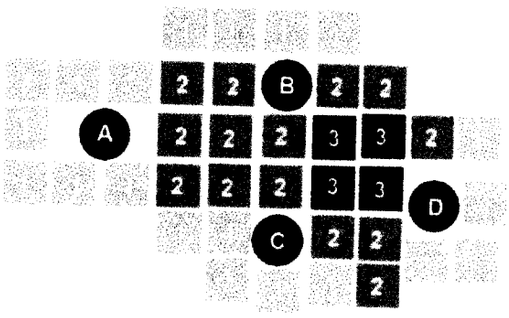

Figure II.7 Localisation du signal avec capteur de beacon

Les solutions de localisation présentent quelques défauts. Ainsi, les capteurs doivent connaitre leur position, cela suppose d'avoir enregistré lors du déploiement leur position géographique, chose impossible pour des capteurs déployés de manière aléatoire. Dans le cas d'utilisation de capteur équipé de GPS, c'est l'aspect financier qui entre en compte. D'autre part, il n'est pas impossible qu'un attaquant réussisse à faire passer certains de

ces capteurs malicieux pour des capteurs balises, ou bien encore de compromettre directement les nœuds balises pour que ses attaques aient un impact plus important.

II.6.4 Chien de garde (Watch Dog)

Le mécanisme du chien de garde [27] consiste à déterminer au sein du réseau de capteurs, des capteurs spécifiques chargés de vérifier le bon transit de l'information.

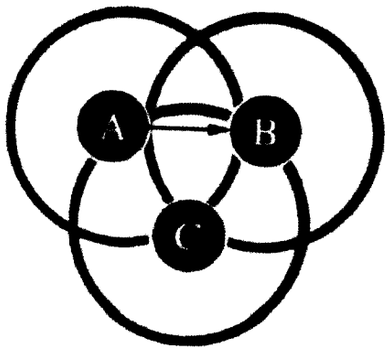

Figure II.8 Mécanisme de chien de garde

La figure II.8 présente le fonctionnement du mécanisme du chien de garde. Un capteur A envoie une information à un capteur B. Le capteur C désigné comme chien de garde écoute la communication, il peut ainsi vérifier la nature de l'information envoyée et vérifier qu'elle correspond aux informations censées circuler dans le réseau. D'autre part si le capteur B n'est qu'un capteur chargé de retransmettre l'information à un autre capteur, pour par exemple avertir la station de base d'un événement détecté, le capteur C vérifiera alors que le capteur B effectue bien cette tâche. Dans le cas contraire, il pourra par exemple déterminer que le capteur B est un capteur malicieux, et détecter une attaque de type trou noir ou trou gris.

Si cette technique apporte une réelle solution pour la détection des capteurs malicieux, elle nécessite une consommation énergétique supplémentaire pour le capteur qui joue le rôle de chien de garde car ce dernier doit écouter chaque communication.

D'autre part, il se pose aussi le problème suivant : le capteur chien de garde désigné peut être en réalité un capteur malicieux, qui ne détectera pas les nœuds malicieux, et pourra jeter le discrédit sur un capteur normal.

II.6.5 Cryptographie

La cryptographie est le fait de cacher un message, de l'écrire de manière illisible, incompréhensible, pour éviter qu'un éventuel indiscret puisse le lire [28]. Elle est réalisée selon certains outils, on va citer quelque uns [29]:

a. Le chiffrement

Le chiffrement est le système cryptographique assurant la confidentialité. Pour cela, il utilise des clés. Selon cette utilisation, on distingue deux classes de primitives : symétrique ou asymétrique.

- ***Le chiffrement symétrique***

Une même clé est utilisée entre deux nœuds communicants pour chiffrer et déchiffrer les données en utilisant un algorithme de chiffrement symétrique. Les algorithmes de chiffrement symétriques sont décomposés en deux catégories :

- Le chiffrement en chaine est fait bit à bit sans attendre la réception entière des données. L'algorithme le plus connu est : RC4.
- Le chiffrement par bloc consiste à fractionner les donnes en blocs de taille fixe (64, 128 bits). Chaque bloc sera ensuite chiffré une fois qu'il atteint la taille envisagée. Les algorithmes les plus utilisés sont: DES, AES.

Bien que l'on connaisse des algorithmes de chiffrement symétriques rapides et efficaces (comme RC4), ils peuvent exposer un grand contraste dans l'énergie dissipée pendant la construction et la distribution de clés. En effet, la distribution de clés est difficile car dans un système symétrique, chaque nœud a besoin d'une clé partagée avec chaque autre nœud du réseau. Donc on aura à gérer n*(n-1)/2 clés si on considère que le nombre de nœuds dans le réseau est égal à n [30].

- ***Le chiffrement asymétrique***

Deux clés différentes son générées par le récepteur : une clé publique diffusée à tous les nœuds servant au chiffrement de données qu'ils vont émettre au récepteur, et, une clé privée maintenue secrète chez le récepteur servant pour le déchiffrement de ces données lorsque ce dernier les reçoit. Le point fondamental sur lequel repose la sécurité du chiffrement asymétrique est l'impossibilité de déduire la clé privée à partir de la clé publique. L'algorithme de chiffrement asymétrique le plus connu est : RSA

Il existe d'autres algorithmes asymétriques, dont l'ECC (Elliptic Curve Crypto-systems, Encryptage par Courbe Elliptique). Ce crypto-système est basé sur une courbe paramétrique qui passe par un certain nombre de points de coordonnées entières.

Dans ces algorithmes, différents problèmes mathématiques se présentent à cause des calculs utilisés pour déchiffrer les données reçues. La complexité de telles opérations est très importante parce que les nœuds capteurs exigent une capacité de traitement plus élevée et une dissipation d'énergie plus haute.

La distribution de clés est moins pénible car leur échange est fortement simplifié. En effet, avec un système asymétrique, chaque nœud a besoin d'une paire de clés. Si on considère que le nombre de nœuds dans le réseau est égal à n, il faudra donc gérer 2*n clés [30].

b. La signature digitale

La signature digitale est un système cryptographique assurant la non-répudiation de la source. Elle repose sur les clés asymétriques. L'émetteur (A) signe les données à transmettre avec sa clé privée (A) en produisant une signature digitale. Ce dernier est par la suite envoyé avec les données. 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, 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.

c. La fonction de hachage

C'est le mécanisme qui assure l'intégrité de donnes. Cette fonction calcule une courte empreinte de taille fixe à partir d'une donnée de taille arbitraire. Etant donnée une fonction de hachage f, et un message à transmettre m. La fonction f doit remplir ces conditions [30]:

- Il est facile de calculer f(m), c'est-à-dire, de calculer l'empreinte à partir du contenu du message.
- Il est difficile de calculer m tel que f(m) = f, c'est-à-dire, de trouver le contenu du message à partir de l'empreinte. C'est pourquoi la fonction f est dire «à sens unique ».
- Il est difficile de trouver un autre message m2 tel que f(m) = f(m2), c'est-à-dire, il est difficile de trouver deux messages aléatoires qui donnent la même empreinte et cela mène à la résistance aux collisions. Cette empreinte est recalculée par le récepteur afin qu'il la compare à celle calculée par l'émetteur. Si elles sont différentes, alors les données ont été altérées pendant leur transmission. Les fonctions de hachage les plus courantes sont : MD5, SHA-1 [31].

d. Le code d'authentification de message MAC

Le code d'authentification de message MAC (Message Authentification Code) fait partie des fonctions de hachage à clé symétrique assurant l'intégrité de données comme toute autre fonction de hachage , en plus, l'authenticité de la source de données. Cette clé est utilisée pour calculer le code MAC par l'émetteur. Ce code est par la suite envoyé avec les données.

Le récepteur calcule à son tour le code MAC avec cette même clé et le compare au code qu'il a reçu. S'ils sont bien identiques, alors la source est authentique et les données n'ont pas été altérées [32].

II.7. Conclusion

Les solutions proposées pour protéger les réseaux de capteurs sans fil permettent de limiter le nombre des attaques possibles sur ce type de réseau. Malheureusement chacune des solutions n'est capable de contrer qu'une partie des attaques et parfois leur utilisation a un coût énergétique non supportable par les réseaux de capteurs.

Dans notre cas, on propose de développer une application qui vérifie l'authenticité d'un capteur en utilisant les courbes elliptiques ECC, qui assurent un niveau de sécurité assez élevé avec des clés de petites tailles comparativement à RSA.

Chapitre III
Cryptographie basée sur les courbes elliptiques

III.1 Introduction

La cryptographie est la technique la plus utilisée dans la plupart des mécanismes de sécurisation actuelle. Elle permet d'empêcher l'écoute des données transitant dans un réseau sans fil et de garantir la confidentialité des données.

Les solutions de cryptographie sont réputées comme des solutions sûres qui répondent à l'ensemble des problèmes liés à la sécurité des données. Cependant, les spécificités des réseaux de capteurs, à savoir une faible puissance de calcul et une mémoire limitée auxquelles se rajoute la contrainte de préservation de l'énergie, sont des handicaps considérables à l'utilisation des crypto-systèmes courant réputés sûrs.

Pour certaines applications, la cryptographie à clé publique n'est pas pertinente et pour d'autres, elle est impossible à mettre en œuvre, car plus coûteuse en ressources que la cryptographie à clé secrète. Dans ce contexte, les chercheurs essaient de réduire au maximum la taille des clés et le temps d'exécution de la cryptographie à clé publique, afin de la rendre adaptable pour les systèmes à ressources limitées, à l'instar les RCSF.

La cryptographie basée sur les courbes elliptiques est le principal système concurrent au système RSA. Il permet d'assurer le même niveau de sécurité que RSA tout en utilisant une clé de taille extrêmement inférieure que celle utilisée par RSA. Cette caractéristique permet aux courbes elliptiques d'être commode pour les systèmes qui ont des ressources limitées (mémoire, CPU, ...) tels que les réseaux de capteurs.

Dans ce chapitre, nous présentons les courbes elliptiques et leurs apports dans le domaine de la sécurité.

III.2 Présentation des courbes elliptiques

Tout d'abord avant de présenter les courbes elliptiques, nous allons rapidement rappeler les définitions des objets mathématiques dont nous aurons besoin pour les définir.

III.2.1Préliminaires

- **Groupe**

Un groupe est un ensemble non vide muni d'une loi de composition interne (G, *) tels que :

- * est associative,
- * admet un neutre e_G,
- tout élément de G admet un symétrique pour *.

Si * est commutative, on dit que (G, *) est commutatif.

- **Anneau**

Un anneau est un ensemble muni de deux lois de composition interne (A,+, *) tels que:

- (A, +) est un groupe commutatif de neutre noté 0_A.

- La loi * est une loi de composition interne sur A associative et distributive à gauche et à droite par rapport à :

$$\forall x, y, z \in A, x*(y + z) = x*y + x*z \text{ et } (x + y)*z = x*z + y*z$$

- La loi * admet un neutre différent de 0A, noté 1A.

Si la loi * est commutative, l'anneau est dit commutatif.

- **Sous-Anneau**

Soit (A,+, *) un anneau. Une partie non vide A1 de A est un sous-anneau de A lorsque :

- $1_A \in A_1$,

- les lois + et * induisent des lois de composition interne, et, muni de ces lois, (A1,+, *) est un anneau.

Remarques

Contrairement aux sous-groupes, on ne peut pas se passer de la condition $1_A \in A_1$, qui ne découle pas des autres conditions.

En outre, on pourra montrer qu'une partie A1 de A est un sous-anneau si et seulement si :

- $(A_1, +)$ est un sous-groupe de (A, +),
- $1_A \in A_1$,
- * induit une loi de composition interne sur A_1.

- **Corps**

Un corps est un anneau commutatif dans lequel tout élément non nul est inversible. Si (K,+, *) est un corps, un sous-corps de K est un sous-anneau K1 de K tel que pour tout élément non nul x de K1, on a $x^{-1} \in K_1$; (K1,+, *) est alors un corps.

- **Equation de Weirstrass**

Soit K un corps, on appelle équation de Weierstrass sur K une équation du type E :

$$y^2 + a_1 x y + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6 \text{ avec } a_i \in K_i$$

Une courbe donnée par une telle équation est dite lisse si le système suivant n'admet pas de solution:

$$\begin{cases} a1\ y = 3\ x2 + 2\ a2\ x + a4 \\ 2\ y + a1\ x + a4 = 0 \end{cases}$$

Autrement dit si les dérivées partielles en x et en y de des équations ci-dessus ne s'annulent pas en même temps.

$$f(x,y) = y^2 + a_1 x y + a_3 y - x^3 + a_2 x^2 + a_4 x + a_6$$

III.2.2 Présentation d'une courbe elliptique [33]

En mathématiques, une courbe elliptique est un cas particulier de courbe algébrique, munie entre autres propriétés d'une addition géométrique sur ses points.

Les courbes elliptiques ont de nombreuses applications dans différents domaines : en mécanique classique dans la description du mouvement des toupies, en théorie des nombres dans la preuve du dernier théorème de Fermat, en cryptologie dans le problème de la factorisation des entiers ou pour fabriquer des codes performants.

Contrairement à ce que son nom pourrait laisser croire, l'ellipse n'est pas une courbe elliptique. Le nom des courbes elliptiques vient historiquement de leur association avec les intégrales elliptiques, elles-mêmes appelées ainsi car elles servent en particulier à calculer la longueur d'arcs d'ellipses.

Une courbe elliptique E définie sur K est une courbe lisse donnée par une équation de Weierstrass définie sur K à laquelle on ajoute un point "à l'infini", noté O.

$$E = \{ (x,y) \in K2 \mid y^2 + a_1 x y + a_3 y = x^3 + a_2 x^2 + a_4 x + a_6 \} \cup \{O\}$$

Si la caractéristique de K n'est pas 2 ni 3, alors en faisant les deux changements de variables successifs :

$y \rightarrow 1/2(y - a_1x - a_3)$ et,

$(x, y) \rightarrow ((x - 3b_2)/36, y/216)$ *dans E,*

où $b_2 = a^2_1 + 4\,a_2 b_2$,

Nous obtenons:

$$E : y^2 = x^3 - 27\,c_4 x - 54\,c_6$$

Avec:

$b_4 = 2a_4 + a_1 a_3$

$b_6 = a_3^2 + 4a_6$

$c_4 = b_2^2 - 24 b_4$

$c_6 = -b_2^3 + 36\,b_2 b_4 - 216\,b_6$

De même, si la caractéristique de K n'est pas 2 ni 3, nous pouvons toujours travailler avec des courbes elliptiques de la forme:

$$E : y^2 = x^3 + Ax + B$$

Dans ce cas la courbe est lisse si:

$$4A^3 + 27B^2 \neq 0$$

III.2.3 Exemples de courbes elliptiques

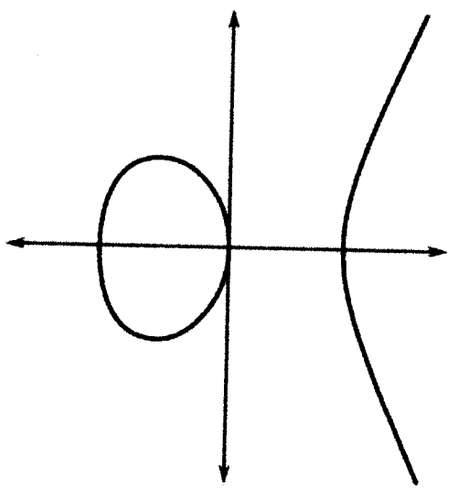

Figure III.1 Représentation de la courbe elliptique $y^2 = x^3 - x$

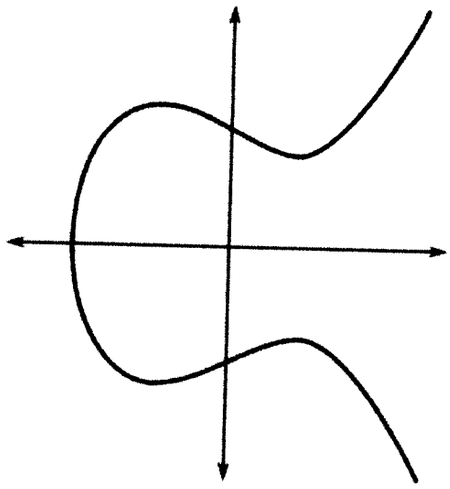

Figure III.2 : Représentation de la courbe elliptique $y^2 = x^3 - x + 1$

III.2.4Arithmétiques sur les courbes elliptiques

Pour faire l'addition des points sur la courbe elliptique, on utilise la méthode des tangentes et des sécantes, soient P et Q deux points de la courbe elliptique, différents du point à l'infini O. La droite tracée (PQ) donne quatre cas possibles :

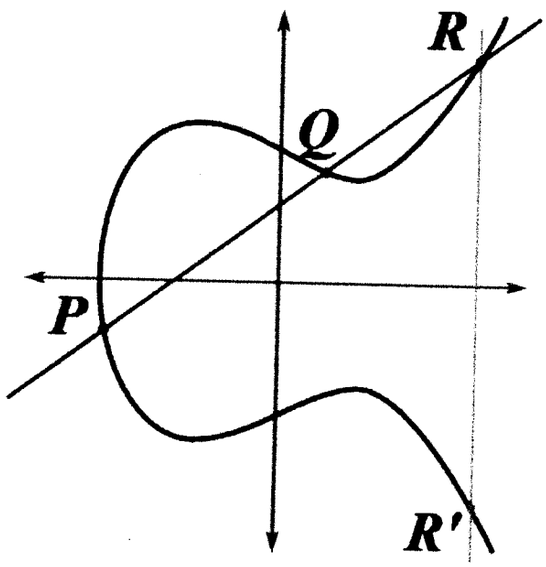

Figure III.3 Représentation du cas d'addition 1

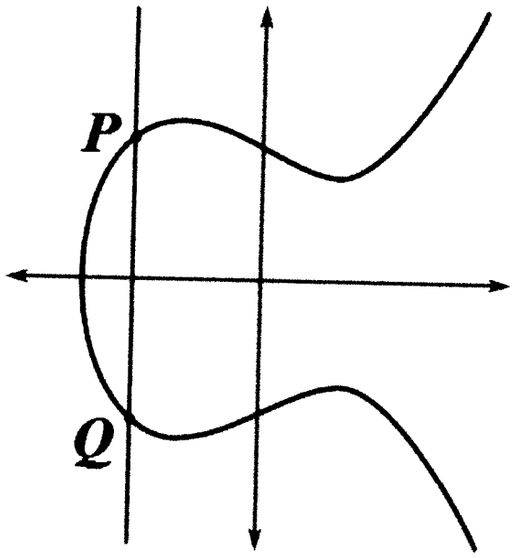

Figure III.4 Représentation du cas d'addition 2

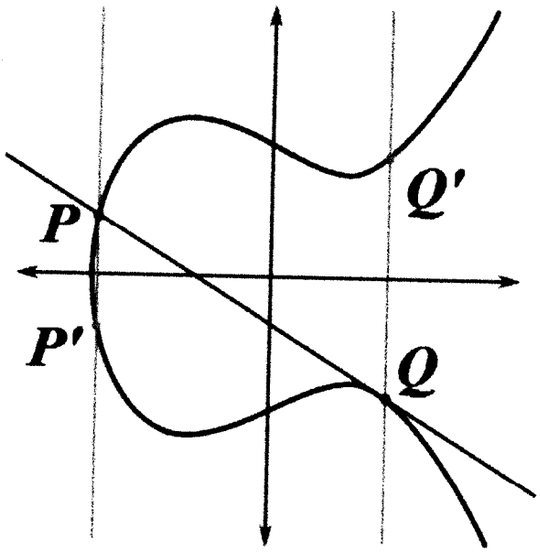

Figure III.5 Représentation du cas d'addition 3

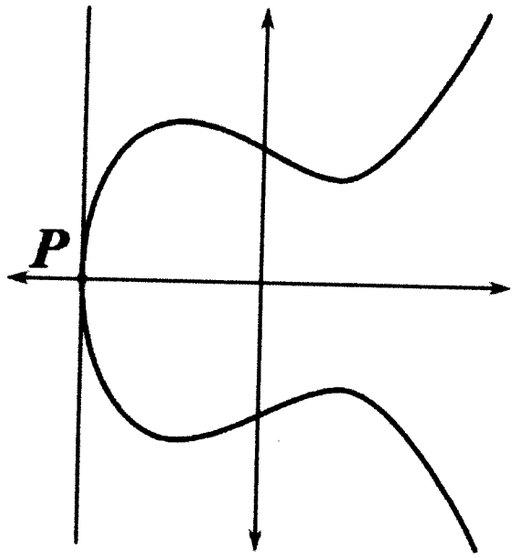

Figure III.6 Représentation du cas d'addition 4

- la droite (PQ) coupe la courbe en un troisième point R : alors la somme des deux points sera le point R' qui est l'image symétrique du point R par rapport à l'axe des abscisses. (représenté dans la figure III.3) $P + Q = R'$
- la droite (PQ) est verticale : c'est à dire parallèle à l'axe des ordonnées. On considère que cette droite coupe la courbe en un troisième point R qui est le point à l'infini, et qu'il est son propre symétrique par rapport à l'axe des ordonnées. (représenté dans la figure III.4) La somme des deux points est donc le point à l'infini $P + Q = 0$
- La droite (PQ) ne coupe la courbe en aucun point : elle est tangente à la courbe en l'un des deux points. On considère que le troisième point d'intersection R est le point de tangence, $R = Q$ ou $R = P$. Si la tangente est en Q alors $R = Q$, si elle est en P alors $R = P$ qui correspond au premier cas, $P + Q = R'$ où R' est l'image symétrique du point R par rapport à l'axe des abscisses. (représenté dans la figure III.5)
- On additionne un point avec lui-même : pour doubler un point P +P ou 2P, on utilise la tangente au point $P = Q$. Elle recoupe la courbe en un troisième point (qui peut être le point P et dans ce cas $2P$ = O) et $2P$ est le symétrique de ce point (représenté dans la figure III.6)

- ***Calcul sur les courbes elliptiques [34]***

Soient $E : y2 = x3 + a\,x + b$ une courbe elliptique, pour les calculs, il suffit alors de calculer les équations de cette droite et l'intersection avec la courbe en fonction des coordonnées des points P1 = (x1; y1) et P2 = (x2; y2) pour obtenir les coordonnées P3(x3; y3) de leur somme. Cela donne les formules suivantes :

1- Si $x_1 \neq x_2$, alors

$$x_3 = m_2 - (x_1+x_2), \quad y_3 = m\,(x_1-x_3)-y_1, \quad où\ m = \frac{(y2-y1)}{(x2-x1)}$$

2- Si $x_1 = x_2$ mais $y_1 \neq y_2$, alors $P_3 = O$.

3- Si $P_1 = P_2$ et $y_1 \neq 0$, alors

$$x_3 = m_2 - 2x_1, \quad y_3 = m\,(x_1-x_3)-y_1 \quad où\ m = \frac{3x_1^2+a}{2\,y1}$$

4- Si $P_1 = P_2$ et $y_1 = 0$, alors $P_3 = O$.

De plus, on a P + O = P pour tout P sur E.

III.2.5Les courbes elliptiques sur un corps fini Fp

- **Définition d'une courbe elliptique sur un corps fini Fp**

Soit K = F_P un corps fini à p éléments et E(F_P) une courbe elliptique définie sur ce corps, tous les traitements sur E(F_P) sont réduits par modulo (p).

Une courbe elliptique contenant des entiers variables non négatifs a, b, x et y peut être définie comme suit : $y^2 \bmod p = x^3 + ax + b \pmod p$

Si le discriminant $4a^3 + 27b^2 \pmod p$ est non nul.

Les courbes elliptiques E(F_P) définies sur un corps fini sont constituées par un nombre fini de points ce qui les rend utilisables pour la cryptographie.

Pour utiliser E(Fp) dans le cadre de la cryptographie, l'équation $y^2 \pmod p = x^3+ax+b \pmod p$ doit être satisfaite pour des valeurs arbitraire a, b et p.

- **Calcul sur les courbes elliptiques sur un corps fini Fp :**

  - **Addition des points**

On détermine les point P et Q de la manière suivante : P = (x_P, y_P) et P = (x_Q, y_Q).

L'addition des points P et Q correspond au Point R = P + Q = (x_{P+Q}; y_{P+Q}).

Soit s = (y_P - y_Q)/(x_P - x_Q) alors,

$$x_{P+Q} = x_R = s2 - xP - x_Q \pmod p$$

$$y_{P+Q} = y_R = s(x_P - x_R) - y_P \pmod p$$

- **Doublement d'un point**

Soit le point $P = (x_P ; y_P)$, on cherche à déterminer $R = 2P = (x_{2P} ; y_{2P})$.

Soit $s = (3x^2_P + a) = (2y_P) \pmod p$, alors

$$x^2_P = x_R = s^2 - 2x_p \pmod p$$

$$y2_P = y_R = -y_P + s(x_P - x_R) \pmod p$$

- **Soustraction d'un point**

Soit R = -P le point recherché correspond à la soustraction du point P :

$$x_R = x_P$$

$$y_R = -y_P \pmod p$$

III.3 Cryptographie basée sur les courbes elliptiques (ECC)

III.3.1 Description

La cryptographie asymétrique basée sur les courbes elliptiques est née en 1985, indépendamment par V. Miller [34] et N. Koblitz [35].

ECC est un crypto-système asymétrique standardisé par le consortium IEEE. La sécurité de ce crypto-système est basée sur la difficulté de résolution du problème du logarithme discret.

ECC utilise un groupe fini composé de points (x, y) se trouvant sur une courbe elliptique dont la procédure de cryptage et de décryptage est basée sur l'addition et la multiplication (addition successive du même point) des points mentionnés précédemment.

L'utilisation de ce crypto-système s'effectue en trois étapes :

- Génération des clés.
- Chiffrement.
- Déchiffrement.

Pour montrer l'apport des ECC, nous les appliquons a l'algorithme Elgamal, mais avant nous allons présenter le problème du logarithme discret (PLD) et le PLD elliptique :

III.3.2 Le logarithme discret [36]

On travaille dans le sous-groupe multiplicatif d'un corps ni F_q, noté F_q^* . On choisit un générateur α de F_q^* (une racine primitive). Chaque élément de F_q^* est alors défini par une unique puissance de α. La fonction à sens unique est la fonction classique d'exponentiation dans le sous-groupe multiplicatif F_q^* :

$$y = \alpha^x \text{ avec } 1 \leq x \leq q - 1$$

et, bien entendu, x est le logarithme de y en base α:

$$x = log_{\alpha} y \; avec \;\; 1 \leq y \leq q\text{-}1$$

III.3.3Le logarithme discret elliptique

Etant donné un point P, on peut définir le point P+P, noté 2P. Pour cela, on effectue la même construction que précédemment. Cependant, ici, P=Q, aussi la droite qui passe par les deux points à additionner est remplacée par la tangente en P à la courbe elliptique. D'une manière similaire, on pourra calculer 3P, 4P, ...Pour tout n, nP est un point de la courbe que l'on peut calculer en faisant (n-1) fois la construction géométrique.

Lorsque n est grand, on peut accélérer le processus de calcul par utilisation de l'algorithme d'exponentiation binaire dont le principe est le suivant: au lieu d'ajouter à chaque fois P, on additionne ensemble des multiples de P que l'on vient de calculer. Le plus facile est lorsque n est une puissance de 2. Par exemple, si l'on veut calculer 8P, on calcule 2P, que l'on ajoute à lui-même pour obtenir 4P et, après la même opération pour obtenir 8P. Ainsi, pour calculer nP, on montre que l'on a besoin d'effectuer au plus 2*Log2 n fois additions. Si n est un entier de 300 bits (un nombre de 91 chiffres en décimal), on obtient nP après au plus 600 fois la construction.

Pour que cette addition soit toujours définie, en particulier lorsque l'on veut additionner deux points symétriques, on prend en plus un point "à l'infini" noté ∞.

Celui-ci se comporte comme un zéro vis-à-vis de l'addition P + ∞ = P. Enfin, deux points symétriques sont opposés, au sens où leur somme est définie comme valant ∞.

Dans le sens inverse, étant donné un point P sur une courbe elliptique et le point nP, on peut retrouver n en essayant toutes les valeurs possibles. Cependant, on ne connaît pas d'analogue de l'algorithme d'exponentiation binaire : si n est grand, on n'y arrivera pas, même avec tous les ordinateurs du monde: pour un entier n de 300 bits, il y a 10^{91} possibilités. Le problème de trouver n est appelé le logarithme discret elliptique [2].

III.3.4Génération et Échange de clés (Diffie-Hellman) [37]

Dans ce mécanisme, nous supposons qu'Alice et Bob qui ne sont jamais rencontrés, souhaitent se fabriquer un secret commun en utilisant un algorithme de chiffrement à clé publique. N'ayant à leur disposition qu'un canal non sécurisé par exemple un courrier électronique. Ils vont utiliser l'algorithme de Diffie-Hellman et les courbes elliptiques pour atteindre cet objectif. Dans cet algorithme, la fonction à sens unique n'est plus la

factorisation comme dans le crypto système RSA, mais c'est le logarithme discret elliptique.

On se donne une courbe elliptique représentée par les coefficients a et b avec un point de départ P appelé point générateur. Ces données sont publiques (a, b et P). Alice et Bob procèdent comme suit pour générer des clés:

- Alice choisit aléatoirement un entier assez grand n_A et calcule le point $Q_A = n_A P$,
- Alice envoie Q_A à Bob par une voie non sécurisée tel que le courrier électronique,
- Bob choisit au hasard un entier assez grand n_B et calcule le point $Q_B = n_B P$,
- Bob envoie Q_B à Alice par une voie non sécurisée,

Après ces échanges, Alice et Bob calculent respectivement les points K_A et K_B comme suit :

$$K_A = n_A Q_B$$

$$K_B = n_B Q_B$$

En fait, K_A et K_B représentent le même point et il s'agit de:

$$K = n_A n_B P$$

Ce point K est leur nouveau secret commun qui peut leur servir comme clé. Un espion éventuel peut voir passer sur Internet QA, QB, P et les coefficients a et b qui représentent la courbe elliptique. Donc, s'il sait résoudre le logarithme discret elliptique, il en déduit les entiers n_A et n_B qu'Alice et Bob ont gardé secrets, et calcule par la suite K. La figure III.7 résume la démarche de génération de clés par le procédé de Diffie-Hellman.

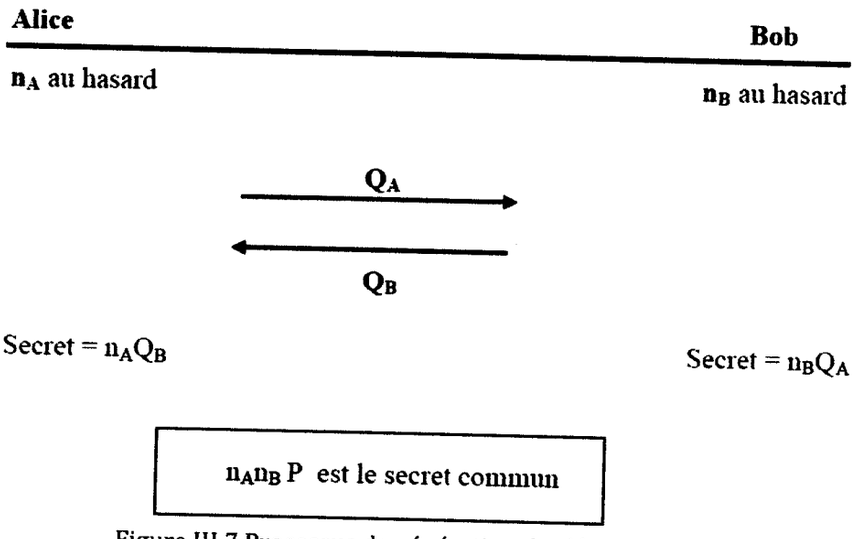

Figure III.7 Processus de génération de clés

III.3.5Chiffrement

Alice souhaite communiquer un message à Bob. La procédure de chiffrement se fait par la conversion de ce message m en plusieurs parties mi et qui correspond chacun d'eux un point M_i de la courbe E dans un corps fini F_P. Ensuite, le chiffrement s'effectue en additionnant ce point M_i à un entier aléatoire nA multiplié par la clé publique de Bob Q_B qui correspond elle-même à un point de la courbe. On obtient alors un point E_{1i} qui correspond à :

$$E_{1i} = M_i + n_A * Q_B$$

Alice doit ensuite calculer le point E2 par multiplication de l'entier nA choisi précédemment avec le point P initialement utilisé lors de la création de sa clé publique:

$$E_2 = n_A P$$

Les points E_{1i} et E_2 sont ensuite transmis à Bob dans cet ordre précis. Ces deux points doivent être précédés d'un ensemble de paramètres T qui doit être échangé avant entre Alice et Bob. Cet ensemble T contient les paramètres :

p : la taille du corps F.

a, b : coefficients de la courbe.

P : point générateur.

III.3.6Déchiffrement

Pour déchiffrer le message m, Bob doit retrouver à l'aide de des points E_{1i} et E_2 reçus, le point mi de la courbe E.

Le point M correspond à :

$$M_i = E_{1i} - n_A * Q_B$$

La première étape consiste ainsi à retrouver $n_A Q_B$, à l'aide de la clé privée de Bob nB. Ainsi :

$$n_A Q_B = n_A (n_B P) = n_B (n_a P) = n_B E_2$$

Soit :

$$M_i = E_{1i} - n_B E2$$

Bob peut alors facilement calculer le point mi connaissant la courbe E et P le point générateur transmis initialement par Alice, et retrouver le message m en entier.

III.4 Comparaison entre RSA et ECC

L'avantage des courbes elliptiques par rapport au système RSA est que le meilleur algorithme connu pour résoudre le problème du logarithme discret elliptique est de

complexité exponentielle. On rend le travail d'un attaquant deux fois plus dur lorsqu'on ajoute seulement deux bits à la taille de la clé, et avec 40 bits, la tâche est un million de fois plus dure. En conséquence, tout en assurant le même niveau de sécurité, les clés et les temps de calculs sont bien plus petits pour les courbes elliptiques que pour le RSA [28].

Le tableau III.1 compare la taille des clés utilisées en cryptographie dans RSA et ECC.

La rapidité d'un système cryptographique basé sur les ECC est dictée par la vitesse à laquelle on effectue l'addition géométrique. En effet, une opération cryptographique se résume le plus souvent à une ou deux exponentiations binaires du point de base P par un entier n dont la taille est similaire à celle de la clé.

Ce constat permet de favoriser l'utilisation des courbes elliptiques pour les systèmes possédant des ressources limitées en termes de mémoire et CPU tels que les capteurs. Dans ce cas, nous pourrons parler de cryptographie légère.

Tableau III-1 Tailles des clés dans RSA et ECC [38]

Taille de la clé (bits)	
RSA	**ECC**
1024	160
2048	224
3072	256

III.5 Conclusion

Dans ce chapitre, nous avons donné un aperçu général sur les courbes elliptiques toute en donnant leurs définition sur un corps fini ainsi que leurs propriétés d'addition et de doublement de points, ensuite nous avons vu comment on peut utiliser ces propriétés dans le domaine de la cryptographie pour concevoir un crypto-système efficace pour les réseaux de capteurs, tout en étant performant par la difficulté de la résolution de son logarithme discret.

En outre, on a opté à accélérer les calculs en utilisant un des algorithmes qui permettent d'optimiser la multiplication d'un point par un scalaire. Ceci dans le but est de minimiser le temps de réponse.

Dans le chapitre suivant on va présenter notre contribution qui porte sur le contrôle d'accès dans un réseau de capteurs basée sur les courbes elliptiques.

Chapitre IV
Mécanisme efficace pour l'authentification dans les RCSFs

IV.1 Introduction

Etablir une communication sécurisée est très important pour la majorité des applications dans les réseaux de capteurs. Cette sécurité à plusieurs aspects, nous en avons cité quelques uns précédemment, dans notre contexte on s'intéresse surtout à l'authentification des capteurs qui consiste de s'assurer que l'identité déclarée par un capteur est bien celle du capteur déclarant.

Dans ce chapitre on va expliquer la contribution proposée en présentant la démarche à suivre et les outils nécessaires pour sa réalisation et son implémentation sur les réseaux de capteurs.

IV.2 L'approche de sécurité proposée pour l'authentification

Avant de présenter notre contribution, on décrit tout d'abord l'environnement dans lequel sera appliquée ainsi que les outils matériels et logiciels pour y parvenir.

IV.2.1 Environnement

Dans le domaine de la télémédecine, on utilise des capteurs pour mesurer à distance des grandeurs physiologiques (tension artérielle, battements du cœur…) pour des personnes dépendantes (vieux, handicapés, …). La sécurité dans ce type d'applications est très importante, car l'erreur est fatale et aura un impact sur la santé du patient. Ce qui nécessite qu'avant de consulter la grandeur physiologique remontée, on doit s'assurer que l'information provient bien d'un capteur légitime.

Dans cette optique, on propose un mécanisme de sécurité pour vérifier l'authentification dans ce type d'applications.

L'architecture de ce mécanisme est présentée par la figure (IV.1)

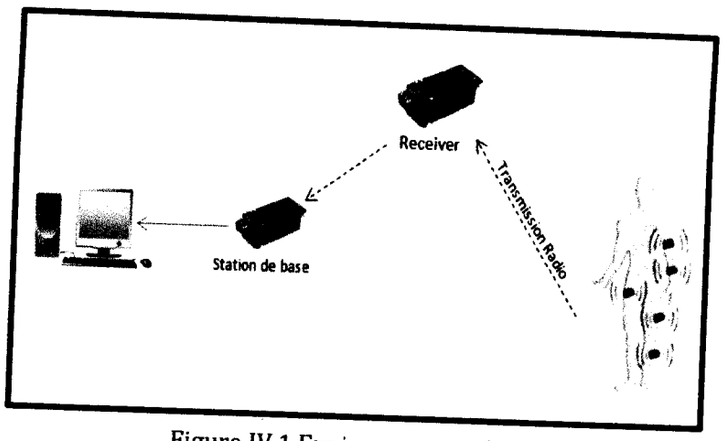

Figure IV.1 Environnement de travail

IV.2.2 Les étapes de la contribution

Avant de présenter les étapes de la contribution, on décrit brièvement l'architecture de l'application qu'on va développer. La figure IV.2 résume cette architecture.

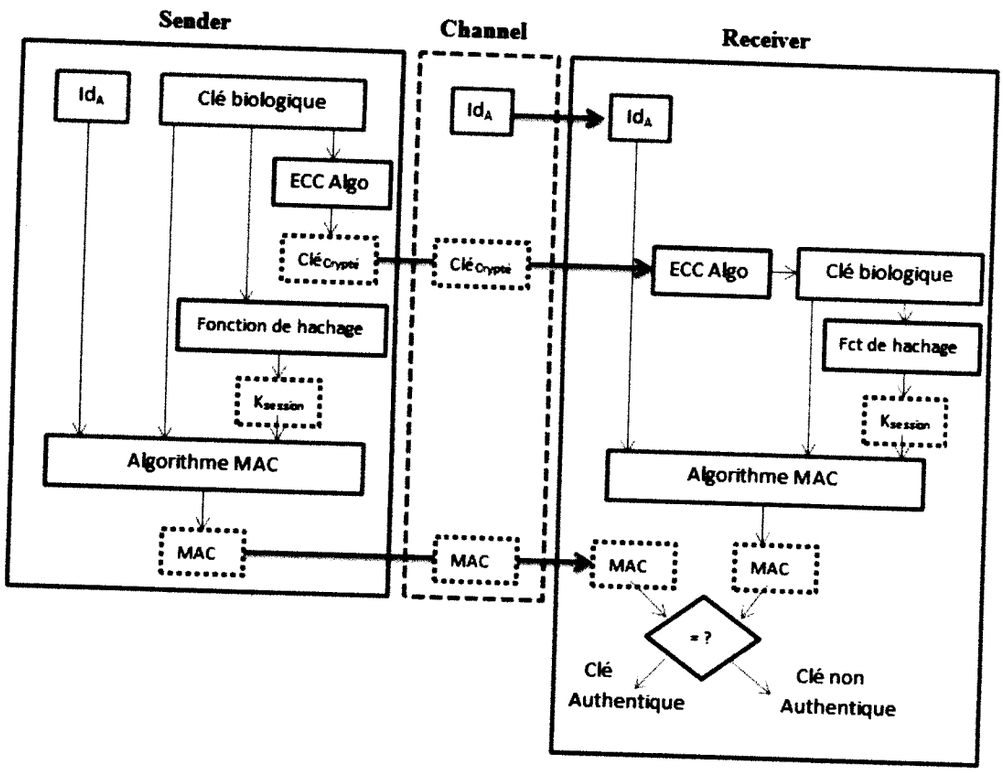

Figure IV.2 Schéma récapitulatif de la contribution

La démarche qu'on a suivie pour établir notre mécanisme de sécurité s'exécute selon le schéma algorithmique suivant :

- Générer une clé biologique (Empreintes digitale ou rétinienne) (Key_B).
- Calculer la clé de session ($K_{session}$) à partir de la clé biologique Key_B en utilisant une fonction de Hachage f ( ) :

$$K_{session} = f(Key_B)$$

- Crypter la (Key_B) avec la clé publique de la station de base :

$$E_{pub}(Key_B)$$

- Calculer le MAC :

$$MAC(K_{session}, Id_A, Key_B)$$

- Transmettre le résultat obtenu à la station de base :

$$Id_A, E_{pub}(Key_B), MAC(K_{session}, Id_A, Key_B)$$

Le but de cet algorithme est de générer une clé unique (*Key_B*) pour chaque capteur (*A*), la crypter et l'envoyer à la station de base avec l'identifiant du capteur concernée (*Id*). La station de base vérifie dans sa base de données si cette clé correspond bien au capteur (*A*).

En plus de la clé (*Key_B*), on envoie aussi à la station de base le résultat du MAC de l'identifiant, la clé Key_B et une clé de session, pour renforcer encore plus notre système.

IV.2.3 Implémentation de l'algorithme

Dans cette partie, on décrit en détail l'approche utilisée :

a. Fonction de Hachage f ()

Pour obtenir la clé de session, on calcule le haché de la Key_B avec la fonction f(), qui est connu par l'émetteur et le récepteur, de la manière suivante :

- Convertir la clé Key_B en binaire, on obtiendra la suite ($a_0 \ldots a_n$) de 1 et de 0,
- Ajouter un 1 au début de la suite binaire (1, $a_0 \ldots a_n$),
- Appliquer un ou exclusif entre les a_i et a_{i+1},

Exemple

- $Key_B = 21 \rightarrow$ En binaire : 10101 $\rightarrow$ Ajouter le 1 : 110101

	1	1	0	1	0	1
	↓	↓	↓	↓	↓	
Résultat :	0	1	1	1	1	

b. MAC (Message Authentification Code)

Pour vérifier l'intégrité des données envoyées à la station de base, on a réalisé une fonction MAC (Id_A, Key_B, $K_{session}$).Cette fonction est réalisé selon le schéma algorithmique suivant :

- Convertir les trois paramètres en points d'une courbe E sur un corps fini F, par les ECC. ($Id_A \rightarrow P_1$, $Key_B \rightarrow P_2$, $K_{session} \rightarrow P_3$)
- Additionner les deux points P_2 et P_3.
- Donc on aura (P1, P2+P3) → (P, Q).
- On calcule les symétriques de P et Q par apport aux axes des coordonnées.
- Si on joint les quatre points (P, (-P), Q, (-Q)) on aura un trapèze (figure IV.3)
- On calcule la surface de ce trapèze $S = ½ * (b + B) * h$

La fonction MAC retourne un entier, qui représente la surface du trapèze formé par le quadruplet de points: [P, (-P), Q, (-Q)].

Exemple

MAC (5,1,3) =

- 5 → P1 (6,3), 1→ P2 (4,3), 3→ P3(5,9)
- P = (6,3) , Q = P2 + P3 = (17,4)
- -P (6,-3), -Q(17,-4)
- Calculer la surface S du trapèze :

$$S = (b+B)*h*1/2$$

- b : petite base = $2 * y_P = 2 * 3 = 6$
- B : grande base = $2 * y_q = 2 * 4 = 8$
- H : hauteur = $x_Q - x_p = 17 - 6 = 11$

$S = (8-6) * 11 * ½ = 11$

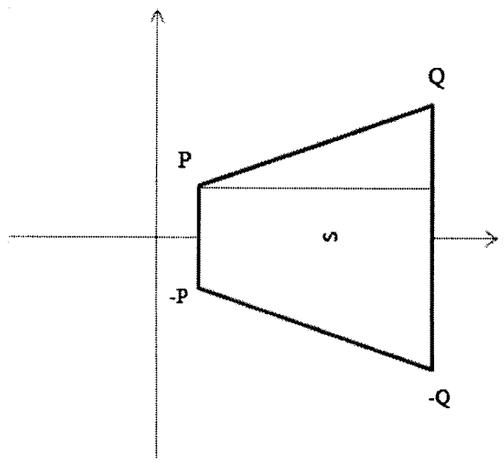

Figure IV.3 Trapèze représenté par les point P-,P,Q-,Q

c. Cryptage de la clé avec les ECC

Pour implémenter la partie de chiffrement plusieurs solutions sont possibles. On a choisi d'implémenter notre algorithme avec un système basé sur les courbes elliptique (ECC), en raison de sa résistance et son efficacité pour les RCSF (gain en énergie et temps de calcul). La réalisation de ce crypto-système s'effectue en trois étapes: Génération des clés, Chiffrement et Déchiffrement.

Le mécanisme de génération de clés est basé sur l'approche de Diffie-Hellman et le mécanisme de chiffrement basé sur l'algorithme d'El-Gamal et les ECC.

✓ **Génération de clés**

- On commence par le choix d'une courbe elliptique sur un corps fini E(Fp) basé sur l'équation : $y^2 \bmod p = x^3 + ax + b \ (\bmod p)$
- On génère les points de la courbe, et on associe à chaque point un chiffre m
- On choisit le point générateur P
- Le Sender et la station de base créent leurs propres clés privées représentées respectivement par n_A et n_B.
- Ensuite, ils calculent respectivement les points ($Q_A = n_A * P$) et ($Q_B = n_B * P$) qui représentent leurs clés public. Puis ils les échangent entre eux.

✓ **Chiffrement**

- On cherche le point B qui correspond à Key_B dans la courbe C,
- On calcule E_1 en additionnant ce point M à la clé privée n_A multiplié par la clé publique de la station de base Q_B qui correspond elle-même à un point de la courbe : $E_1 = M + n_A * Q_B$
- Le Sender doit ensuite calculer le point E_2 par multiplication de l'entier n_A avec le point P choisi précédemment : $E_2 = n_A * P$

Le résultat de cette opération (message chiffré) c'est les points E_1 et E_2. Ces deux points doivent être précédés d'un ensemble de paramètres T qui doit être échangé avant entre receiver et la station de base. Cet ensemble T contient les paramètres :

p : la taille du corps F

a, b : coefficients de la courbe

P : point générateur

✓ **Déchiffrement**

Pour déchiffrer le message, la station de base doit retrouver à l'aide des points E_1 et E_2 le point M, avec la formule suivante :

$M = E_1 - n_A * Q_B$

$= E_1 - n_A * (n_B * P)$

$= E_1 - (n_A * n_B) * P$

$= E_1 - (n_B * n_A) * P$

$= E_1 - n_B * (n_A * P)$

Donc $M = E_1 - n_B * E_2$

Exemple :

Soit une courbe elliptique $_{EP,}$ définie sur le corps F29 par l'équation suivante :

$$y^2 \bmod 29 = x^3 + 11x + 17 \ (mod\ 29)$$

La courbe relative à l'équation précédente contient 26 points, illustré dans le tableau IV.1 suivant :

Tableau 2-1 Les points de la courbe dans le corps F29

(9,2)	(5,20)	(28,11)	(25,24)	(20,1)	(6,3)	(27,4)	(17,10)	(4,3)
(23,24)	(19,3)	(10,24)	(1,0)	(10,5)	(19,26)	(23,5)	(4,26)	(17,19)
(27,25)	(6,26)	(5,20)	(25,5)	(28,18)	(5,9)	(9,27)		

- **Chiffrement**

Pour chiffrer la Key_B, le sender doit calculer les deux points E_1 et E_2 connaissant la clé publique de la station de base qui est $Q_B = n_B * P$.

On suppose que le point générateur P est le point numéro 10 (23,24), et que la clé secrète du sender est 2 et celle de la station de base est 3. Donc :

$$Q_B = 3P = 2P + P$$
$$= 2(23,24) + (23,24)$$
$$= (6,26) + (23,24) = (25,24)$$

Supposant que $Key_B = 21$, qui est représenté par le point M de coordonnées (5,20)

$E1 = M + n_A * Q_B = (5,20) + 2*(25,24) = (5,20) + (17,10) = (23,24)$

$E2 = n_A * P = (23,24) + (23,24) = (6,26)$

Le sender envoie alors les deux points E1, E2.

- **Déchiffrement**

La station de base peut facilement calculer le point M, et retrouver la Key_B :

$$M = E_1 - n_B * E_2$$
$$= (23,24) - 3*(6,26) = (5,20)$$

La station de base consulte la table des points, elle trouvera l'entier correspondant au point (5,20) → $Key_B = 21$.

d. Données envoyées par le sender :

On envoie le résultat de cet algorithme à la station de base, qui contiendra :

- L'identifiant du capteur A (Id_A).
- Les deux points E1 et E2, qui représente la clé (Key_B).
- Et le résultat du MAC.

IV.3 Les choix techniques

Dans cette section, on va présenter les choix techniques logiciels et matériels qu'on a choisis pour parvenir à créer ce mécanisme d'authentification pour les RCSF :

IV.3.1 Outils logiciels

IV.3.1.1 Le système d'exploitation : TinyOs [16]

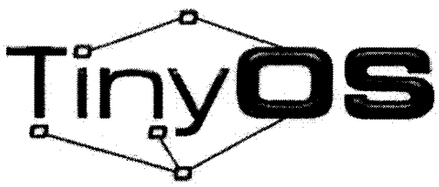

Figure IV.4 Logo de TinyOs

TinyOs est un système d'exploitation open source développé et suivi par l'université de Berkeley.

Ce système d'exploitation a été développé avec pour objectif principal de réduire au maximum la taille d'allocation mémoire nécessaire à son installation et à son fonctionnement. Pour cela, TinyOs a une architecture qui s'articule autour de composants où chaque composant fournit des interfaces ou utilise des interfaces des autres composants. L'ensemble des composants permettent l'ajout de fonctionnalités pour le capteur.

Un composant contient un fichier de configuration et un fichier module pour l'implémentation des interfaces du composant.

Un programme sous TinyOs ne doit comporter que les composants nécessaires à son exécution, ce qui a pour effet une réduction de la taille du programme à insérer dans l'unité de traitement du capteur.

Un autre objectif de TinyOs est de prolonger la durée de vie du capteur. Dans cette optique, la programmation sous TinyOs est une programmation événementielle, c'est-à-dire que l'exécution des différentes instructions s'effectue en fonction de l'occurrence des événements. Ce type de programmation est généralement adapté aux capteurs, puisqu'il n'y a de traitements que lors d'apparitions d'événements, ce qui permet au capteur de rester dans un état de veille le reste du temps afin de préserver son énergie.

Fort de ces deux objectifs atteints, TinyOs est devenu aujourd'hui le système d'exploitation le plus complet pour les réseaux de capteurs. Cependant son utilisation se

limite à certains capteurs (Mica2, MicaZ, TelosB, TelosA, etc...), qui sont essentiellement les capteurs développés par la société Xbow.

IV.3.1.2 MIG (Message Interface Generator)

TinyOs fournit des outils pour produire automatiquement des objets message des descriptions de paquet. Ainsi, au lieu d'analyser les formats de paquet manuellement, l'outil MIG établit une interface Java à la structure de message. En effet, pour une séquence d'octets donnée, le générateur de code MIG devra analyser automatiquement chacun des champs des paquets et fournit un ensemble d'accesseurs et de mutateurs standard pour visualiser les paquets reçus ou produire des nouveaux paquets.

IV.3.1.3 Le langage NesC [39]

Le langage NesC (Network Embedded System C) est une extension du langage C. Il a été conçu pour la réalisation de systèmes embarqués distribués. Il offre une réactivité importante vis-à-vis de l'environnement, une gestion de la concurrence et un support de communication. NesC s'appuie sur des composants ce qui permet de décomposer une application en modules réutilisables.

NesC utilise trois abstractions de programmation: interface, module et configuration :

a. Interface : contient les signatures des commandes et des évènements que le module devra implémenter. Comme TinyOs n'autorise pas les pointeurs de fonctions, NesC propose une alternative : Les interfaces paramétrées. interface SendMessage [uint8 id] permet de créer 256 interfaces de type SendMessage, il suffit alors d'en sélectionner une à l'aide de son identifiant.

b. Module : permet d'implémenter les composants. Il est composé de deux blocs : le premier contient les déclarations, le second l'implémentation. Dans le premier bloc, les mots clés « use » et « provide » permettent de savoir si le module fait appel à une fonction de l'interface « use » ou redéfini son code « provide ». « use nomInterface » signifie que le module doit fournir l'implémentation des évènements de cette interface, et « provide nomInterface » signifie que le module doit fournir les commandes de cette interface. Le second bloc commence par le mot clé « implementation » et se compose des variables privées et de l'implémentation des différentes commandes et évènements. Un module est de la forme suivante :

```
Module nomModule{
Provides {
        //liste des interfaces fournies
        Interface nomInterface }
Uses {
        //liste des interfaces requises
        Interface nomInterface  }
Implémentation {
        //déclaration des variables
        //implémentation des fonctions décrites par les interfaces fournies
} }
```

c. **Configuration** : définit le ou les composants qui seront utilisés par l'application. Elle permet de décrire les composants composites. Elle est constituée de modules et/ou d'interfaces et de la déclaration des liaisons entre composants. Elle est de la forme suivante :

```
Configuration nomConfig{}

Implémentation {
        //liste des modules et configurations utilisées
        Composants Main, Module1,.....ModuleN, Config1.....ConfigM
        //description des liaisons
        Interfaces requises->Interfaces fournies}
```

IV.3.1.4 *Le langage JAVA*

JAVA[40] a été développé par Sun dans une filiation avec le langage C et surtout le langage objet C++ mais dans une optique de plus grande portabilité d'une machine à une autre et d'une plus grande fiabilité.

On a choisis ce langage car il a de nombreuses caractéristiques telles que :

- ✓ Simplicité : plus simple que le C ou le C++ car on lui a retiré les caractéristiques peu utilisées ou difficile à utiliser (Pointeurs, surcharge de méthodes, libération de mémoire …)
- ✓ Orienté Objet : l'ensemble des instructions de base a été réduit pour augmenter sa fiabilité mais donne accès à de nombreuses classes permettant de définir et de manipuler des objets.
- ✓ Sûr : L'interpréteur de java vérifie que le byte code n'a pas été altéré depuis sa compilation. Sun travaille sur un système qui permettra à terme la transmission codée des informations sur le réseau.
- ✓ Dynamique : les classes de Java peuvent être modifiées sans modification du programme qui les utilise.

✓ Portable : les types de données sont indépendants de la plateforme.

IV.3.1.5 La base de données MySQL

MySQL [41] est un SGBD open-source, fiable, très simple d'utilisation, compatible avec Windows, Mac et Linux et possède une documentation complète.

L'accès à la base de données se fait via l'API JDBC (Java DataBase Connectivity). Cette API permet à une application Java d'accéder à une base de données si celle-ci possède des drivers JDBC (actuellement des drivers existent pour tous les SGBD).

Pour se connecter à un SGBD avec cette API, il suffit de préciser le driver du SGBD qui sera utilisé.

IV.3.1.6 L'outil de développement : Netbeans

NetBeans [42] est un environnement de développement intégré (EDI), open-source. Il offre plusieurs fonctionnalités :

- Supporte différents langages en outre Java, Python , C, C++, JavaScript, XML, Ruby, PHP et HTML.
- Comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).
- Disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation comme TinyOs (requérant une machine virtuelle Java).

Un environnement Java Development Kit JDK est requis pour les développements en Java. L'IDE Netbeans s'enrichit à l'aide de plugins.

IV.3.2 Matériels

Dans le cadre de notre travail, on a utilisé des capteurs de la famille Telos. Il s'agit des capteurs TelosB (Figure IV.5). Ce type de capteurs est fabriqué par Crossbow et développé par l'université de Berkeley.

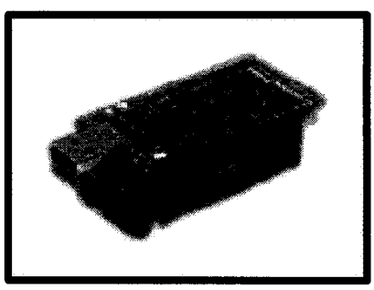

Figure IV.5 capteur TelosB

Ces capteurs ont les caractéristiques suivantes [43]:

1. Radio réseau sans fil IEEE 802.15.4/ZigBee
2. Connecteur USB pour programmation, lien au point de collecte
3. Capteur de température, lumière, humidité
4. Microcontrôleur TI MS430 avec 10 ko de RAM
5. Mémoire Flash de 1 Mo
6. 2 piles AA
7. Système d'exploitation TinyOs-1.x et TinyOs-2.x

IV.3.3 Installation

La première étape de ce travail consiste à installer les logiciels cités précédemment : le système d'exploitation TinyOs, le langage NesC qui lui est associé, Netbeans, JDK pour Java, et Easyphp pour la base de données.

Une fois l'installation logicielle terminée, il a fallu installer le matériel : une station de base reliée à l'ordinateur via un câble USB, deux capteurs TelosB (Figure IV.6) dont un joue le rôle de Sender et l'autre de Receiver :

- ✓ Le Sender permet l'acquisition des grandeurs environnementales et les envoie au Receiver.
- ✓ Le Receiver communique avec la station de base via une liaison sans fil.
- ✓ La station de base communique avec l'ordinateur via le câble USB.

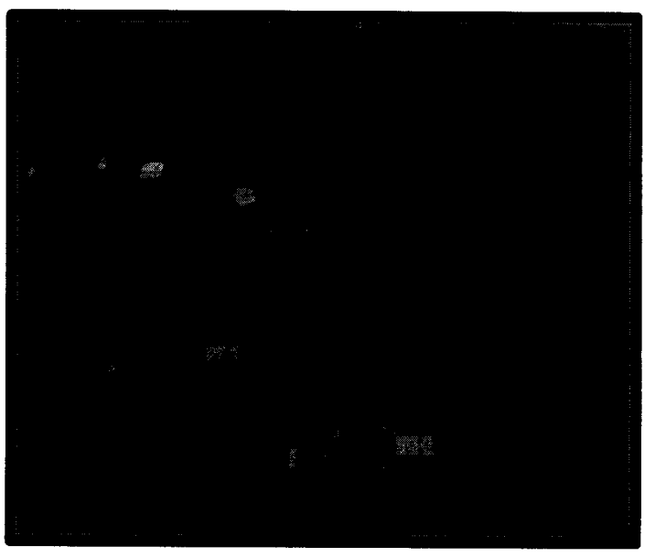

Figure IV.6 Environnement générique de travail

IV.4 Fonctionnement de l'application

Dans cette section, on décrit l'architecture de l'application, qui est composé de trois parties : chiffrement côté Sender, déchiffrement et authentification dans la station de base, et la base de données.

IV.4.1 La maquette

Les communications se font suivant la figure IV.7. Le sender envoie périodiquement les paquets au receiver, et ce dernier les transmet à la station de base via la liaison radio. La station de base transmet alors ces données à l'ordinateur qui traite ces informations via la liaison USB.

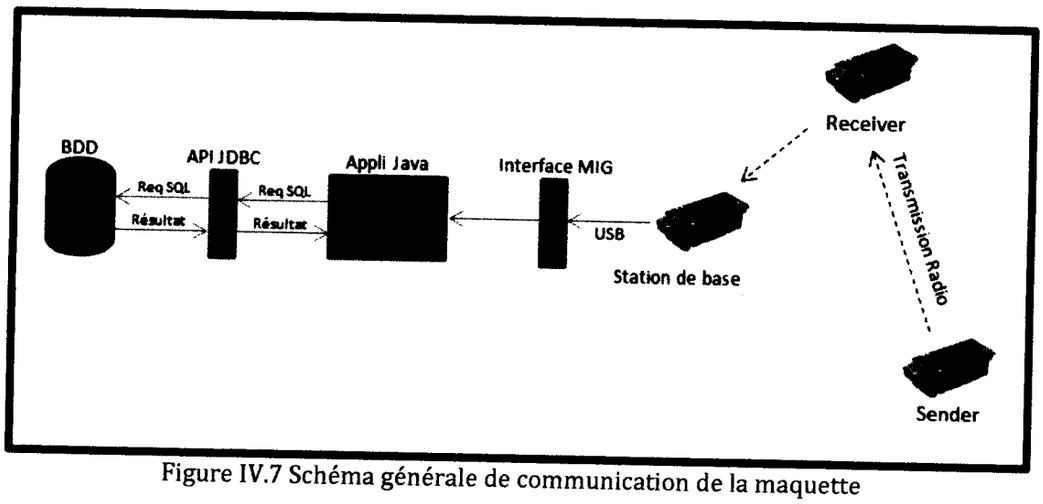

Figure IV.7 Schéma générale de communication de la maquette

Le processus d'authentification se déroule comme suit :

a. Le Sender envoie à intervalle régulier un message contenant son identifiant et sa clé biologique crypté par l'algorithme qu'on a cité précédemment au receiver.
b. Le receiver s'occupe de les transmettre à la station de base. Dans cette architecture, le receiver jour le rôle de relai.
c. La station de base transmet ces messages à l'application java via l'interface MIG qui permet d'établir une interface Java à la structure de message.
d. L'application reçoit donc les paquets chiffrés, les déchiffre pour retrouver la clé biologique,
    - Il calcule le haché de cette clé pour obtenir la clé de session
    - Il calcule le MAC (clé biologique, clé de session, l'identifiant du capteur), pour vérifier l'intégrité des données envoyées.

e. ensuite il consulte sa base de données pour vérifier si la clé biologique correspond à l'identifiant reçu.

f. Pour communiquer avec la base de données, l'application java utilise l'API JDBC. Cette API java permet la liaison entre une base de données et une application java, l'application envoie une requête à la base de données via l'API, la base de données retourne la réponse (vrai ou faux) pour vérifier si le capteur est légitime ou non.

IV.4.2 Application côté Envoi :

L'application de chiffrement nécessaire à la maquette s'articule autour de ces composants :

- **L'interface Encrypt** : Contient toutes les fonctions (commandes) nécessaires pour le chiffrement de la clé (ECC, fonction de hachage, MAC).

```
interface Encrypt {
command Point_Msg multi_pt (int, Point_Msg);
command Point_Msg add_pt (Point_Msg, Point_Msg);
command int fct_hachage (int a);
command int MAC (int a, int b); }
```

- **Le module SenderC** :

C'est en quelque sorte la partie principale de notre application NesC coté Sender. C'est un composant dans lequel on met l'implémentation de l'interface qu'on doit fournir (Encrypt), et les implémentations des événements. Il utilise aussi des interfaces :

- ✓ Boot : chargée de la gestion de démarrage.
- ✓ TimerMilli : permet le déclenchement d'évènements périodiques.
- ✓ Leds : permet de suivre le déroulement du programme en allumant les leds du capteur à des moments précis.
- ✓ Packet : Interface qui gère les paquets.
- ✓ AMsend : interface fournissant la commande Send pour envoyer des paquets.
- ✓ SplitControl : interrupteur de l'antenne radio.

```
module SenderC {
provides { interface Encypt ; }
 uses interface Boot;
 uses interface Timer<TMilli>;
 uses interface Leds;
 uses interface Read<uint16_t>;
 uses interface Packet;
 uses interface AMSend;
 uses interface SplitControl as RadioControl; }
implementation {
event void Boot.booted() { /* IMPLÉMENTATION */ }
event void RadioControl.startDone { /* IMPLÉMENTATION */ }
event void RadioControl.stopDone {}
event void Timer.fired() { /* IMPLÉMENTATION */ }
event void Read.readDone(error_t err, uint16_t val) { /* IMPLÉMENTATION */ }
event void AMSend.sendDone(message_t* msg, error_t err) { /* IMPLÉMENTATION */

command Point_Msg multi_pt (int, Point_Msg) {/* IMPLÉMENTATION */ }
command Point_Msg add_pt (Point_Msg, Point_Msg) { /* IMPLÉMENTATION */ }
command int fct_hachage (int a) { /* IMPLÉMENTATION */ }
command int MAC (int a, int b) { /* IMPLÉMENTATION */ }
 } }
```

- **La configuration SenderAppC** : C'est dans le code de configuration que l'on doit relier les utilisateurs et les fournisseurs des interfaces.

```
Configuration SenderAppC {}
implementation {
 components SenderC as App;
 components MainC, LedsC ;
 components new TimerMilliC () as Timer;
 components ActiveMessageC ; //CalculeP;
 components new SensirionSht11C () as Sense;
 App.Boot -> MainC;
 App.Leds -> LedsC;
 App.Timer -> Timer;
 App.Read -> Sense.Temperature;
 App.Packet -> ActiveMessageC;
 App.AMSend -> ActiveMessageC.AMSend
[AM_TEMPERATURE_MSG];
 App.RadioControl -> ActiveMessageC;
}
```

- **Le fichier Message.h** : contient la définition de la structure du message que l'on va envoyer.

```
typedef nx_struct Message_Envoye {
        Point E1;       //Point E1
        Point E2;       //Point E2
        Courbe c;       //Courbe c
        nx_uint16_t MAC;        //Résultat MAC
        nx_uint16_t id;         //identifiant du capteur
} Message_Envoye;
```

IV.4.3 Application côté réception

Sur le PC, on a un programme Java qui reçoit et affiche les paquets reçus, mais pour que Java puisse interpréter les informations dans un paquet, il faut qu'il connaisse la structure du contenu, c'est-à-dire la structure que l'on a défini dans le fichier « Message.h ». TinyOs fournit l'outil MIG, qui est capable d'analyser un fichier « .h » et convertir toutes les structures « nx_struct » en classes Java disposant des méthodes nécessaires qui nous permettent d'accéder aux valeurs des attributs.

L'application java (Figure IV. 8) s'occupe du déchiffrement du paquet reçu qui contiendra : l'identifiant du capteur, deux points qui représentent la clé Key_B et le résultat de la fonction MAC. A partir de ces paramètres il doit retrouver la Key_B du capteur A pour vérifier son authenticité.

Pour assurer l'intégrité de la clé, on calcule le MAC, et on le compare avec le résultat reçu. Si le résultat est le même donc il s'agit bien de la clé du capteur A. Ensuite, La station de base consulte sa base de données et vérifie si l'identifiant Id_A reçu correspond bien à la clé retrouvé, si c'est le cas donc le capteur A est légitime et on l'autorise à envoyer d'autres paquets, sinon on interrompe la connexion.

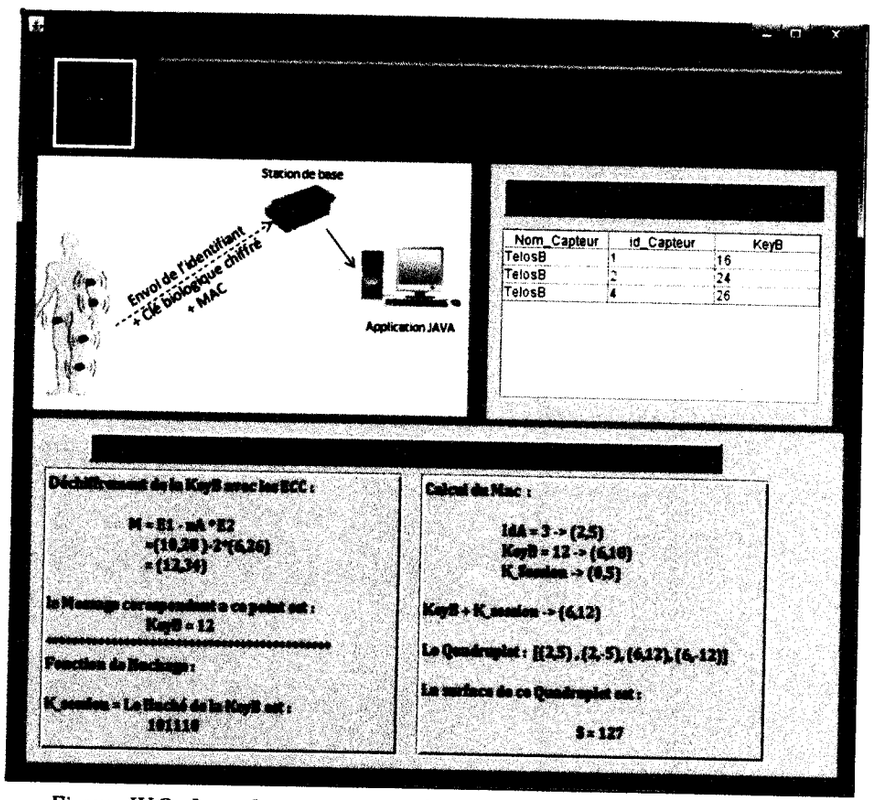

Figure IV.8 : Interface principale de l'application de déchiffrement

IV.4.4 Base de données MySQL

Dans notre application, on a eu besoin d'une base de données MySQL, pour vérifier si la clé biologique provient bien du capteur dont on a reçu l'identifiant. Donc on a créé une base de données nommée « Capteurs », qui contient une seule table « Auth_Capteur » (Figure IV.9), qui a les attributs suivants : (Identifiant du capteur, Nom du capteur, Clé biologique)

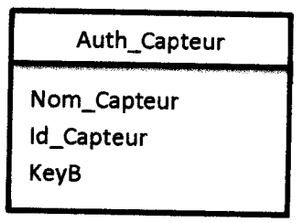

Figure IV.9 Table Auth_Capteur

La figure IV.8 résumé les étapes qu'on a utilisées pour réaliser notre mécanisme de sécurité.

IV.5 Conclusion

Dans ce chapitre, on a proposé un mécanisme d'authentification efficace pour les réseaux de capteurs. Ce mécanisme est basé sur les courbes elliptiques. Les ECC permettent de garantir un niveau de sécurité très élevé tout en utilisant des clés de tailles moyennes. Ce qui les rend adaptables aux systèmes à ressources limitées à l'instar des réseaux de capteurs.

En outre, pour renforcer la sécurité dans ce type de réseaux, on a implémenté une fonction MAC pour assurer l'intégrité des données échangées.

Conclusion Générale

Les RCSFs constituent des sujets de recherche innovants pour diverses disciplines des sciences et techniques de l'information et de la communication mais avec toutefois des contraintes spécifiques s'érigeant en défis certains à relever. Parmi les problèmes posés à l'heure actuelle dans ce type de réseaux, la sécurité en est un véritable problème et auquel une solution adéquate doit être apportée.

Dans ce mémoire, on a traité ce problème, et plus précisément l'aspect d'authentification (contrôle d'accès), on a pu voir que la principale problématique de l'application de méthodes de sécurité dans les RCSF réside dans le souci de préserver l'énergie des capteurs. En l'occurrence, il apparaît que la plupart des solutions actuelles ne permettent pas de garantir une consommation énergétique faible, ou dans le cas contraire ne propose pas un protocole de sécurité fiable qui puisse garantir la sécurité des informations circulant.

Dans cette optique, on a proposé un mécanisme de sécurité basé sur les courbes elliptiques. Ce mécanisme permet d'une part de garantir une faible consommation d'énergie puisque il fait recours à la cryptographie légère et d'autre part il rend difficile la résolution du logarithme discret elliptique.

Pour implémenter ce mécanisme de sécurité, on a utilisé l'approche de Diffie-Hellman pour la génération de clés, l'algorithme d'El-Gamal pour le chiffrement et le déchiffrement et on a ajouté au message chiffré un MAC pour renforcer encore plus la sécurité. Dans ce mécanisme le codage des messages est représenté par les points d'une courbe elliptique et l'ensemble des opérations sur les points de la courbe s'effectue dans un corps fini. Enfin, on a expérimenté le mécanisme proposé sur trois capteurs TelosB.

En perspectives on propose d'optimiser cette solution (en terme de temps de calcul) et de l'essayer sur d'autres types de capteurs, comme les WBAN (wireless body area networks) pour contrer un certain nombre d'attaques.

Références

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. I. Cayirci. "A survey on sensor networks ". IEEE Communications Magazine, Vol. 40, No. 8, pp. 102-116, August 2002.

[2] Messai Mohamed Lamine. "Sécurité dans les Réseaux de Capteurs Sans-Fil ". Mémoire Magistère, Université Abderrahmane Mira de BEJAIA, 2007.

[3] The Sensor Museum. In http://www.btnode.ethz.ch/Projects/SensorNetworkMuseum/.

[4] David Martins, "Sécurité dans les réseaux de capteurs sans fil stéganographie et réseaux de confiance", Thèse de doctorat, Université de FRANCHE-COMTÉ, 2010.

[5] Tinyos. http:/www.tinyos.net/, 2010.

[6] C.C Han, R. Kumar, R. Shea, E. Kohler, and M. Srivastava. "A dynamic operating system for sensor nodes. In proceedings of the 3rd international conference on Mobile systems, applications, and services, pp.163-176, New York, USA, 2005.

[7] Contiki . HTTP://www.contiki-os.org/

[8] FreeRTOS. http://www.freertos.org/

[9] Matis os. http:/www.mantisos.org/

[10] Nut/OS. http ://www.ethernut.de/en/software/index.html.

[11] Equipe de Get 2005 Capt Ad-hoc. "Sensor networks: State of the art". Technical Report, Telecom Paris, ENST Br, INT, INRIA, Mars 2006.

[12] Lehsaini Mohamed. " Diffusion et couverture basées sur le clustering dans les réseaux de capteurs : application à la domotique ", Thèse de doctorat, Université A.B Tlemcen et Université de Franche-Comté U.F.R Sciences et Techniques École Doctorale SPIM, 2009.

[13] Hung-Cuong LE. "Optimisation d'accès au médium et stockage de données distribuées dans les réseaux de capteurs". Thèse de Doctorat, Université de FRANCHE-COMTÉ.

[14] Kacimi Rahim. " Techniques de conservation d'énergie pour les réseaux de capteurs sans fil ", Thèse de Doctorat, Institut National Polytechnique de Toulouse, Septembre 2009.

[15] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci. "Wireless sensor networks: a survey". Computer Networks 38, Elsevier Science, pp. 393–422, 2002.

[16] Yacine challal "Réseaux de Capteurs sans fils", Cours, Système intelligents pour le transport, Université de Technologie de Compiègne, France, 17 novembre 2008

[17] C.Y. Chong and S.P. Kumar. Sensor Networks: Evolution, Opportunities, and Challenges. In Proceedings of the IEEE, vol.91, no.8, pp. 1247-1256, 2003.

[18] T.B. Gosnell, J.M. Hall, C.L. Ham, D.A. Knapp, Z.M. Koenig, S.J. Luke, B.A. Pohl, A. Schach von Wittenau, and J.K. Wolford. Gamma-Ray Identification of Nuclear Weapon Materials. Technical Report DE97053424. Lawrence Livermore National Lab., Livermore, CA (USA). February 1997.

[19] M. J. Brown. "Users Guide Developed for the JBREWS Project". Technical Report LA-UR- 99- 4676. Los Alamos National Laboratory of California University. 1999.

[20] P. Johnson and D.C Andrews. "Remote continuous monitoring in the home". Journal of Telemedicine and Telecare, vol.2, no.2, pp.107-113, June 1996.

[21] Chris Karlof and David Wagner. Secure routing in wireless sensor networks: attacks and countermeasures. Ad Hoc Networks, 293-315, 2003.

[22] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Wormhole attacks in wireless networks. IEEE Journal on Selected Areas in Communications, 24(2):370_380, 2006.

[23] J. Newsome, E. Shi, D. Song, and A. Perrig. The Sybil attack in sensor networks: analysis & defenses. In Information Processing in Sensor Networks, 2004. IPSN 2004. Third International Symposium on, pages 259_268, 2004.

[24] Michel Abdalla Thomas Claveirole, Marcelo Dias De Amorim and Yannis Viniotis. Résistance contre les attaques par capture dans les réseaux de capteurs. In JDIR, 2007.

[25] Bekara Chakib and Maryline Laurent-Maknavicius. A new resilient key management protocol for wireless sensor networks. In Damien Sauveron, Constantinos Markantonakis, Angelos Bilas, and Jean-Jacques Quisquater, editors, WISTP, volume 4462 of Lecture Notes in Computer Science, pages 14_26. Springer, 2007.

[26] D. Liu, P. Ning, and W. Du. "Detecting malicious beacon nodes for secure location discovery in wireless sensor networks ". In ICDCS, pp.609-619, 2005.

[27] Lopez J. Roman R., J. Zhou. "Applying intrusion detection systems to wireless sensor networks ". In proceedings of the 3rd IEEE Int. Conference on Consumer Communications and Networking Conference, (CCNC'06), pp.640-644, 2006.

[28] Thomas IZARD, "Opérateurs arithmétiques parallèles pour la cryptographie asymétrique", Thèse de Doctorat, Université de Montpelier II, Décembre 2011.

[29] Samir Athmani. "Protocole de Sécurité pour les Réseaux de Capteurs Sans-Fil ". Mémoire Magistère, Université Hadj Lakhdar Batna, 2010.

[30] Hatem Bettahar, Yacine Challal, « introduction à la sécurité informatique », Support de cours, Systèmes intelligents pour le transport, Université de Technologie de Compiègne, France, 15 Octobre 2008.

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

[32] Noureddine LASLA « La gestion de clés dans les réseaux de capteurs sans-fil » mémoire de magistère, Ecole supérieure d'informatique (E.S.I) Oued-smar, Alger.

[33] S. Duquesne, « Cryptographie sur les courbes elliptiques», Support de cours, Ecole jeunes cherceurs, 5 avril 2005.

[34]N. Koblitz, "Elliptic curve cryptosystems", In Mathematics of Computation, volume 48, pages 203–209, 1987.

[35] V. Miller, "Use of elliptic curves in cryptography", Advances in Cryptology—CRYPTO'85, Vol. 218 of Lecture Notes in Computer Science, pages 417–426. Springer-Verlag, 1986.

[36] Thomas IZARD, "Optimisation de la multiplication scalaire sur les courbes elliptiques pour de petits scalaires ". Mémoire Master, Université Montpellier 2, juin 2008.

[37] W. Diffie and M. Hellman. New directions in cryptography. IEEE Transactions on Information Theory, IT-22:644{654, 1976.

[38] N. Gura, A. Patel, A. Wander, H. Eberle, and S.C. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In Cryptographic hardware and embedded systemsCHES 2004: 6th international workshop, Cambridge, MA, USA, August 11-13, 2004: proceedings, volume 6, page 119. Springer-Verlag New York Inc, 2004.

[39] NesC: A Programming Language for Networked Systems. http://nescc.sourceforge. net /.

[40] Java. http://www.java.com/fr/.

[41] MySql. http://www.mysql.fr/.

[42] NetBeans. http://fr.netbeans.org/.

[43] Jun SHI, "Implémentation de mécanismes de sécurité efficaces pour les réseaux de capteurs ", Mémoire de master, Université de Franche-Comté, Juin 2010.

Annexe

1. Installation de TinyOs sous Ubuntu

Etape 1 : Ajouter TinyOs repository a /etc/apt/source.list :

```
deb http://tinyos.stanford.edu/tinyos/dists/ubuntu feisty main
```

Etape 2 : Mise à jour du repository cache

```
sudo apt-get update
```

Etape 3 : Installer les packets

```
sudo apt-get install tinyos tinyos-avr tinyos-msp430 nesc
tinyos-tools
```

Etape 4 : Désinstaller le driver (car on utilise le capteur TelosB).

```
sudo apt-get remove brltty
```

Etape 5 : Installer Java sous Ubuntu : https://jdk-distros.dev.java.net/ubuntu.html

Etape 6 : Configurer l'environnement :

```
export TOSROOT=/opt/tinyos-2.x
export TOSDIR=$TOSROOT/tos
export CLASSPATH=$TOSROOT/support/sdk/java/tinyos.jar:.
export MAKERULES=$TOSROOT/support/make/Makerules
export PATH=/opt/msp430/bin:$PATH
```

En plus de ces variables d'environnement, on exécute les deux instructions suivantes :

1. Changer le ownership dans /opt/tinyos-2.x files:

```
chown -R <your uid> /opt/tinyos-2.x
```

2. Changer les permissions dans :

```
serial (/dev/ttyS<N>), usb (/dev/tts/usb<N>,/dev/ttyUSB<N>)
```

Etape 7 : Installer Graph Drawing Tools

```
apt-get install graphviz
```

Etape 8 : Compilation et installation

```
$ tos-check-env
$ configure
```

Si on a besoin de tcl et otcl arguments, on les fixe :

```
$ configure -with-tcl=/yourpath/tclxxx/ -with-otcl=/yourpath/otcl-
xxx/

 $ make
 $ make install
```

Etape 9 : Vérifier MAKERULES environment

```
$ printenv MAKERULES
```

Voilà ce qui doit être affiché :

```
/opt/tinyos-2.x/support/make/Makerules
```

Si rien ne s'affiche, modifier MAKERULES :

```
$ export MAKERULES=/opt/tinyos-2.x/support/make/Makerules
```

2. Installation des capteurs TelosB sous Linux

Etape 1 mise à jour de repository standard de TinyOs

```
sudo echo deb http://tinyos.stanford.edu/tinyos/dists/ubuntu
karmic main >> /etc/apt/sources.list
sudo apt-get update
```

Etape 2 Installer le compilateur nesc

```
sudo apt-get install nesc
```

Etape 3 Installer les drivers

- Debian MSP430:

```
sudo apt-get install msp430- bintuils-tinyos msp430-gcc-tinyos
msp430-libc-tinyos
```

- Debian AVR:

```
sudo apt-get install avr-bintuils-tinyos msp430-gcc-tinyos
msp430-libc-tinyos
```

Etape 4 checkout l'arbre source de TinyOS 2.x

```
cd ~
mkdir -p local/src
cd local/src
cvs -z3 -
d:pserver:anonymous@tinyos.cvs.sourceforge.net:/cvsroot/tinyos
co -P tinyos-2.x
```

Etape 5 Compilation TinyOS

```
cd tinyos-2.x/tools
./Bootstrap
./configure --prefix=$HOME/local
```

```
make all
make install
```

Etape 6 Configuration des variables environnement

Ajouter dans le ficher .bashrc ou .bash_profile les lignes suivantes :

```
export PATH=$HOME/local/bin:$PATH
export TOSROOT=$HOME/local/src/tinyos-2.x
export TOSDIR=$TOSROOT/tos
export MAKERULES=$TOSROOT/support/make/Makerules
export CLASSPATH=$TOSROOT/support/sdk/java/tinyos.jar:.
export PYTHONPATH=.:$TOSROOT/support/sdk/python:$PYTHONPATH
export PATH=$HOME/local/src/tinyos-2.x/support/sdk/c:$PATH
```

Etape 7 Installer JDK 5

Nous recommandons installer JDK la version 5.

3. Faire marcher les programmes sur le capteur

- Se placer dans le répertoire ou se trouve l'application
- Accéder au sender
- Compiler le programme :

  ```
  make telosb
  ```

- Récupérer l'identifiant du capteur avec la commande :

  ```
  Motelist
  ```

- Installation :

  ```
  Make telosb install /dev/tty/USB0
  ```

  `/dev/tty/USB0` : le chemin du capteur, 0 est l'identifiant du capteur qu'on a récupéré avec motelist.
- Refaire la même chose pour le Receiver
- Lancer l'application java :

  ```
  java net.tinyos.tools.MsgReader Message -comm serial@/dev/ttyUSB0:telosb
  ```

- Si le programme n'affiche rien, faite redémarrer les capteurs.

Contrôle d'accès basé sur les courbes elliptiques pour la sécurité dans les réseaux de capteurs

Nous nous intéressons aux solutions de sécurité visant à protéger la communication des composants sans fil dans un réseau de capteurs. Malheureusement, ces dispositifs sont limités par leur puissance de calcul et par leur durée de vie. Ce qui ne permet pas d'utiliser les mêmes solutions que les réseaux traditionnels comme la cryptographie à clé publique.

Dans ce mémoire, nous avons proposé un mécanisme de sécurité léger pour l'authentification dans les réseaux de capteurs en utilisant les courbes elliptiques. Dans ce mécanisme, on utilise les ECC pour crypter la clé du capteur et une fonction MAC pour assurer l'intégrité des données de l'authentification et l'authenticité de leurs sources.

Mots clés : Authentification, Code d'authentification de message, Courbes elliptiques, Cryptographie, Réseaux de capteurs sans fil, Sécurité.

Access control based on elliptic curve for security in sensor networks

We are interested in security solutions to protect wireless communication components in sensor networks. Unfortunately, sensors are limited by their low computing power and their lifetime. So we cannot use the same solutions as traditional networks, as public key cryptography.

In this memory, we proposed a lightweight security mechanism for authentication in sensor networks using elliptic curves. In this mechanism, we used the ECC system to encrypt the sensor key and a MAC function to ensure the integrity of authentication data and the authenticity of the source.

Keywords: Authentication, Message authentication code, Elliptic curves, Cryptography, Wireless sensor network, Security.

مراقبة الولوج على أساس المنحنيات الاهليلجية لتأمين شبكات الاستشعار

نهتم بالحلول التي تضمن حماية الاتصالات اللاسلكية لمكونات شبكات الاستشعار. و لكن تتميز هذه الأجهزة بقدرة حسابية محدودة و مدة حياة قصيرة.و هذا لا يسمح لها بإستعمال نفس الحلول الّتي تستخدم في الشبكات التقليدية مثل التشفير بشفرات عمومية .

في هذه المذكرة إقترحنا آلية لحلّ مشكلة إثبات المصدر في شبكات الاستشعار بإستخدام المنحنيات الاهليلجية . في هذه الآلية يتم إستخدام نظام المنحنيات الاهليلجية لتشفير مفتاح جهاز الإستشعار ونظام رمز مصادقة الرسالة للتحقق من أصالة هذا المفتاح وصحة مصدره.

الكلمات الرئيسية :الرئيسية : إثبات المصدر , رمز مصادقة الرسالة ,المنحنيات الاهليلجية ,التشفير, شبكات الاستشعار ,حماية المعطيات .