Université Abou Bekr Belkaid
Tlemcen Algérie
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: *Système d'Information et de Connaissances (S.I.C)*

Thème

Implémentation d'un mécanisme de sécurité basé sur les courbes elliptiques pour les réseaux de capteurs

Réalisé par :

- **RAMDANI Mohamed**

Présenté le 28 Septembre 2011 devant le jury composé de MM.

- *DIDI Fedoua* *(Président)*
- *LEHSAINI Mohamed* *(Encadreur)*
- *CHOUITI Sidi Mohamed* *(Examinateur)*
- *MIDOUNI Djallal* *(Examinateur)*
- *HALFAOUI Amel* *(Examinateur)*
- *EL YEBDRI Zineb* *(Examinateur)*
- *BENMANSOUR F.* *(Examinateur)*

Année universitaire: 2010-2011

Table des Matières

Table des figures

Liste des tableaux

Introduction générale

Les réseaux de capteurs sans fil (RdCs) bénéficient actuellement d'un engouement lié aux nouvelles possibilités qu'ils offrent, grâce notamment aux avancées technologiques dans l'élaboration des capteurs. 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 RdCs sont limités dans leur capacité à utiliser des méthodes de sécurité traditionnelle par leur faible puissance de calcul comparée 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 du capteur, c'est-à-dire en utilisant peu la puissance de calcul et l'énergie des capteurs, qui doivent permettre de répondre aux problèmes de sécurité dans ce type de réseaux. Pour répondre en partie aux problèmes de sécurité, nous cherchons dans ce travail à définir des solutions peu coûteuses en énergie pour les RdCs 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.

Ce mémoire s'articule autour de quatre chapitres:

La premier chapitre introduit les réseaux de capteurs sans fil en présentant leur évolution, leurs spécificités, et leurs domaines d'application. Le deuxième chapitre présente les attaques connues dans ce type de réseau et les solutions proposées pour

tenter d'endiguer ces menaces. Dans le troisième chapitre, nous présentons les courbes elliptiques et leurs avantages pour concevoir un crypto-système léger et efficace. Le quatrième chapitre est une contribution dans laquelle nous proposons un mécanisme de sécurité basé sur les courbes elliptiques pour les systèmes présentant des ressources limitées à l'instar des réseaux de capteurs. Nous concluons ce mémoire en rappelant les avantages et inconvénients des solutions que nous avons proposées, et nous y détaillons les perspectives possibles qui pourraient découler des travaux et propositions que nous auront abordées.

Chapitre I

Les réseaux de capteurs: concepts et applications

Chapitre I

Les réseaux de capteurs: concepts et applications

I.1 Introduction

Les réseaux de capteurs se composent généralement d'un grand nombre de petits dispositifs, qui communiquent entre eux via des liens radio pour le partage d'information et le traitement coopératif. Ces dispositifs sont déployés aléatoirement dans une zone d'intérêt pour superviser ou surveiller des phénomènes divers. Après le déploiement initial, les capteurs peuvent s'auto-organiser en une infrastructure réseau appropriée, souvent en mode multi-sauts. Les données collectées par ces capteurs sont acheminées directement ou via un routage multi-sauts à un nœud considéré comme "point de collecte", appelé station de base. Cette dernière peut être connectée à une machine puissante via internet ou par satellite. En outre, l'usager peut adresser ses requêtes aux capteurs en précisant l'information d'intérêt.

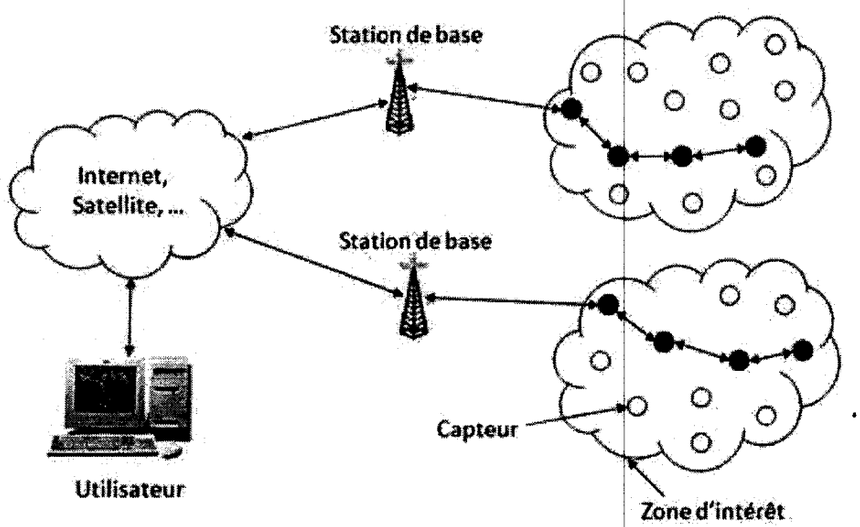


Figure I-1: Exemple de réseaux de capteurs

La figure I.1 illustre un exemple de réseaux de capteurs où les capteurs sont déployés de manière aléatoire dans une zone d'intérêt, et une station de base située à l'extrémité de cette zone, est chargée de récupérer les données collectées par les capteurs. Lorsqu'un capteur détecte un événement pertinent, un message d'alerte est envoyé à la station de base par le biais d'une communication multi-sauts. Les données collectées sont traitées et analysées par des machines puissantes.

Les réseaux de capteurs sont étroitement contraints en termes d'énergie, mémoire, capacité de traitement, et débit réalisable. Ils sont aussi contraints par une bande passante réduite et une latence élevée due à la nature du canal radio partagé. Le canal de communication radio est moins fiable qu'un médium filaire. Les capteurs peuvent aussi être mobiles, ce qui nécessite des algorithmes adaptatifs au changement de la topologie réseau. Néanmoins, le vrai défi critique dans ce type de réseaux est l'énergie car les capteurs sont dotés souvent de batteries non rechargeables. Ainsi, l'objectif principal dans ces réseaux est de minimiser la consommation d'énergie tout en assurant que le réseau effectue sa tâche dans des meilleures conditions. Par conséquent, une gestion de ressource rigoureuse en termes d'énergie sera exigée.

En outre, le déploiement d'un réseau de capteurs exige la fidélité d'acquisition c'est-à-dire que l'occurrence d'un événement pertinent doit être détectée par au moins un capteur et la fidélité de routage c'est-à-dire qu'il doit exister au moins un chemin entre le capteur qui a détecté l'événement et la station de base.

I.2 Les capteurs

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

Un capteur est un dispositif qui permet de transformer une mesure physique observée en une mesure généralement électrique qui sera à son tour traduite en une donnée binaire exploitable et compréhensible pour un système d'information et utile pour l'homme.

Parmi les différents types de mesures enregistrées par les capteurs, on peut citer entre autres : la température, l'humidité, la luminosité, l'accélération, la distance, les mouvements, la position, la pression, la présence d'un gaz ou d'une fumée, la vision (capture d'images), le son, etc…

La notion de capteur a évolué avec le temps. Là où les premiers capteurs n'étaient dédiés qu'à un unique type de mesure, les capteurs contemporains sont la combinaison de plusieurs dispositifs capables de mesurer différentes mesures physiques. On pourrait vulgairement dire qu'un nœud capteur, est un dispositif composé de plusieurs capteurs.

En outre, à ces possibilités de mesures multiples, les capteurs actuels ont vu se greffer des fonctionnalités qui leur permettent, en plus de l'enregistrement et de la détection d'événements mesurables, le traitement de ces données et leur communication

vers un autre dispositif. On parle alors de capteur intelligent, capable à la fois de mesurer des données et de les communiquer avec d'autres capteurs au sein d'un réseau.

I.2.2 Architecture

L'évolution de l'architecture des capteurs est un des facteurs qui a permis l'essor de solutions à base de réseaux de capteurs. En effet, les capteurs des générations précédentes avaient une architecture qui se limitait au capteur proprement dit (dispositif capable de mesurer une grandeur physique) et une unité d'alimentation (batterie, piles, etc. . .). Le détecteur de fumées est le bon exemple de ce type de capteur. Celui-ci n'est composé que d'un système de capture alimenté par une pile, capable de détecter la fumée et de déclencher une alarme.

Les capteurs actuels ont su évoluer pour ajouter les fonctionnalités de traitement de l'information et de la communication de cette information.

La figure I.2 illustre l'architecture générale d'un capteur dit intelligent.

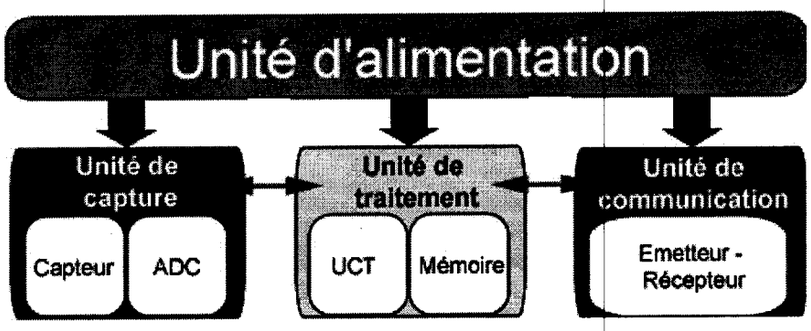


Figure I-0-2: Architecture d'un nœud capteur

Cette architecture s'articule autour de quatre éléments:

- l'unité de traitement: c'est l'unité principale du capteur. Elle est généralement composée d'un processeur couplé à une mémoire vive. Son rôle est de contrôler le bon fonctionnement des autres unités. Sur certains capteurs, elle peut embarquer un système d'exploitation pour faire fonctionner le capteur. Elle peut aussi être couplée à une unité de stockage, qui servira par exemple à y enregistrer les informations transmises par l'unité de capture.
- l'unité de capture: elle permet de mesurer des grandeurs physiques ou analogiques et les convertir en données numériques. Elle est composée du capteur lui-même et de

l'ADC[1]. Le capteur est chargé de récupérer les signaux analogiques qu'il les transmet à l'ADC qui a pour rôle de transformer les données analogiques en données numériques compréhensibles par l'unité de traitement.

- l'unité de communication: elle a pour fonction de transmettre et recevoir des données capturées. Elle est équipée d'un couple émetteur/récepteur appelé transceiver. La communication de l'information au sein du réseau se fait par le biais des ondes radios, la fibre optique, infrarouge, etc ...
- l'unité d'alimentation: c'est un élément primordial de l'architecture du capteur, c'est lui qui fournit l'énergie à toutes les autres unités. Elle correspond le plus souvent à une batterie ou une pile alimentant le capteur. La réalisation récente d'unité d'alimentation à base de panneaux solaires tente d'apporter une solution pour prolonger sa durée de vie.

Par ailleurs, l'architecture d'un nœud capteur peut être dotée d'autres unités. Citons, entre autres, la possibilité d'ajouter une unité de localisation, tel qu'un GPS, une unité de mobilité pour assurer la mobilité du capteur. Dans le cas de l'utilisation d'un GPS, il est intéressant de noter que leur utilisation dans les capteurs actuels a un coût non négligeable en termes de consommation énergétique, qui peut amener à une réduction drastique de la durée de vie d'un capteur.

I.2.3 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. . .). Il est intéressant de décrire les capteurs les plus utilisés et leur évolution au cours du temps.

En l'occurrence, la figure I.3 illustre l'évolution des capteurs au cours des ces vingt dernières années. Cette représentation met en avant l'importance des travaux de recherche de l'université de Berkeley dans l'essor des réseaux de capteurs, surtout sachant que l'entreprise Crossbow.

[1] ADC Analog to Digital Converter (convertisseur analogique numérique)

Figure I-0-3: Evolution des capteurs

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, etc. . .) 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.

Le tableau I.1 reprend les principales caractéristiques des capteurs de la société Xbow(Crossbow), ainsi que les capteurs les plus utilisés dans le domaine de la recherche.

Tableau 1: Caractéristiques des capteurs les plus utilisés

Nom capteur	MCU	RAM	Flash	Stockage	Radio	Dimension
Spec node (2003)	AVR RISC 8 bit	3 KB	0 KB	0 KB	RF	2 x 2.5mm
Mica (2001)	ATMega128	4KB	128KB	512KB	TR1000	
Mica2(2002)	ATMega128	4KB	128KB	512KB	CC1000	58 x 32 x 7mm
Mica2Dot (2002)	ATMega128	4KB	128KB	512KB	CC1000	25 x 6mm
MicaZ (2004)	ATMega128	4KB	128KB	512KB	CC2420	58 x 32 x 7mm
TelosA (2004)	TI MSP430	2KB	60KB	512KB	CC2420	
TelosB (2004)	TI MSP430	10KB	48KB	1MB	CC2420	65 x 31 x 6mm
Tmote Sky (2004)	TI MSP430	10KB	48KB	1MB	CC2420	3.2 x 8 x 1.3cm
BTnode3 (2004)	ATMega128	64KB	128KB	180KB	CC1000/	58.15 x 33mm
Imote (2003)	ARM7	64KB	512KB	0KB	ZV4002/	
Imote2 (2007)	Intel PXA271	256KB	32KB	0KB	CC2420	36 x 48 x 9mm
Iris (2008)	ATmega1281	8KB	128KB	512KB	CC2420	58 x 32 x 7mm

I.3 Systèmes d'exploitation

Les avancées technologiques récentes permettent de faire embarquer un système d'exploitation léger au sein des capteurs. Les fonctionnalités de ce système d'exploitation restent toutefois limitées.

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 multi-tâches. Ces systèmes d'exploitation permettent de développer un peu plus le caractère intelligent des capteurs. Parmi les systèmes d'exploitation actuels, TinyOs [1] fait office de référence dans le domaine scientifique et d'expérimentation des capteurs.

I.3.1. TinyOs

TinyOs est un système d'exploitation open-source pour les RdCs qui trouve sa genèse au sein de l'université de Berkeley (USA). 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 fonctionnalité pour le capteur. Par exemple le composant Timer permet l'utilisation de compteur de temps pour des applications du capteur.

Un composant se compose de deux fichiers:

- un fichier configuration qui détermine les interfaces proposées par le composant et les interfaces des autres composants nécessaires à ce composant. Ce fichier doit aussi expliciter les connexions qui existent entre ces différents composants.
- un fichier module qui contient l'implémentation des interfaces du composant.

Le langage de programmation de ces composants est le langage NesC, une variante du langage C pour les systèmes embarqués.

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 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.

I.3.2. Autres OS

Il existe d'autres systèmes d'exploitation qui sont conçus pour les réseaux de capteurs tels que:

- Contiki [2] : système d'exploitation open-source multitâche, développé pour les systèmes embarqués avec contraintes de mémoire.
- SOS [3] : 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 [4] : n'est pas un système d'exploitation à proprement dit, mais un noyau de système d'exploitation pour systèmes embarqués.
- Mantis OS [5] : 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 [6] : Système d'exploitation multitâches pour les systèmes embarqués avec une pile TCP/IP.

I.4 Applications

Selon les applications à mettre à place, les fonctionnalités désirées et les besoins pour un réseau de capteurs seront très variés.

Les exemples suivants montrent au sein de ces différents types d'applications comment les RdCs peuvent être utiles et quelles réponses ils peuvent apporter à un problème donné.

I.4.1 Domotique

La domotique concerne tous ce qui gravite autour de l'automatisation de la maison. Les applications de type réseau de capteurs dans ce domaine tournent principalement autour du concept de la maison intelligente, à savoir une maison équipée de capteurs détectant diverses informations sur son état et agissant en conséquence. Ces applications peuvent par exemple servir à contrôler les appareils électroménagers ou détecter la présence d'un intrus dans la maison.

Le futur des réseaux de capteurs concernant la domotique s'articulera autour de l'utilisation des RFID, de l'utilisation de robots et de ce qui est appelé l'internet des objets définit par l'EPCGlobal[2].

En outre, le besoin en sécurité pour les réseaux de capteurs dans le domaine de la domotique reste limité pour la plupart des applications, mais nécessaire comme pour le cas d'applications détectant la présence d'intrus dans une maison ou pour éviter une prise de contrôle des appareils ménagers par un individu extérieur.

[2] EPCGlobal est une organisation ayant pour mission d'établir les standards pour l'utilisation des RFID

I.4.2. Environnementales

La plupart des capteurs actuels apporte une réelle solution pour la mesure des données environnementales. Déployer ces capteurs en grand nombre dans une zone d'intérêt apporte des précisions qui n'étaient pas possibles d'avoir auparavant à cause du coût important des capteurs et du besoin de relever à la source les informations, lié à l'absence de communication inter-capteurs.

Le déploiement des RdCs permet ainsi des applications dans le domaine environnemental comme dans le cas de réseau de capteurs déployés dans la forêt américaine [7] pour détecter les débuts d'incendie et prévenir rapidement les pompiers pour endiguer les feux de forêts.

D'autres applications utilisant les RdCs ont pour objectif de détecter des éboulements ou des tremblements de terre. L'utilisation de RdCs permet aussi de suivre l'évolution de l'activité volcanique et de prévenir les instances de sécurité si un risque trop grand lié à cette activité nécessite l'évacuation de la population locale. Actuellement, certains réseaux de capteurs sont aussi déployés pour surveiller l'évolution du réchauffement climatique comme dans le cadre du projet Hydro-Sensor-Flow [8] mené par les équipes Thema et FEMTO-ST de l'université de Franche-Comté. Ce programme consiste à surveiller l'évolution de la fonte du glacier Loven Est au Groenland pour évaluer l'impact du réchauffement climatique.

L'utilisation de ce type de réseau ne se limite pas à la mesure des grandeurs physiques environnementales, mais aussi il est utilisé dans le cadre de suivi des espèces en danger.

Le niveau de sécurité nécessaire dans des applications environnementales utilisant les réseaux de capteurs sans fil peut paraître un peu faible. Cependant, ce serait oublier les risques de sabotage qui peuvent être le fait de personnes agissant dans un but gratuit comme peut l'être la dégradation de bien publics.

I.4.3. Médicales

Le milieu médical trouve un grand intérêt dans l'utilisation de RdCs par exemple les applications qui permettent surveiller l'état d'un patient au sein de l'hôpital.

Le projet CodeBlue [9] a été un des premiers projets développé par l'université de Harvard (USA) utilisant les réseaux de capteurs sans fil et consistait à équiper des patients de capteurs relevant les informations telles que les pulsations cardiaques et le

niveau d'oxygénation. En cas de relevé anormal sur le patient, les capteurs transmettent l'information à un dispositif de type PDA porté par un membre du personnel soignant pour l'avertir afin qu'il intervienne sur le patient au plus vite.

Par ailleurs un grand nombre d'applications est actuellement développé dans le but de surveiller des patients à domicile (détection de chutes, suivi de l'état du patient, etc. . .). Toutes ces applications ont pour but de créer une surveillance active et permanente de la santé d'un patient.

Cependant, ces solutions se heurtent au problème de la sécurité des informations transitant sur le réseau. Il faut pouvoir s'assurer que les données ne puissent être falsifiées. Ainsi, une personne mal intentionnée pourrait envoyer de fausses informations (niveau d'insuline par exemple) sur une application non protégée, ce qui pourrait causer une mauvaise réaction du personnel médical et causer de graves troubles pour le patient.

I.4.4. Militaires

Les applications dans le domaine militaire utilisant les RdCs sont nombreuses et variées [10,11], mais sont principalement développées pour l'armée de terre. On pourra citer entre autres l'utilisation de RdCs denses dans un champ de bataille pour y détecter des ennemis potentiels et suivre leurs avancées sur le terrain. Les RdCs peuvent aussi servir de soutien aux troupes comme dans les cas où ils permettent de déterminer l'avancée des véhicules, des drones et des soldats sur le terrain, de les géo-localiser, de connaître leur état et de leur envoyer des informations ou des ordres.

Si la plupart des projets utilisant des RdCs sont inconnus au grand public, car ils sont liés au secret-défense, on peut tout de même citer les projets WATS [12] et JBREWS [13]. Le projet WATS (Wide Area Tracking System) consiste à utiliser un réseau de capteurs capable de détecter les rayons gamma et les neutrons afin de dépister des dispositifs nucléaires. Le projet JBREWS (Joint Biological Remote Early Warning System) quant à lui s'articule autour d'un réseau de capteurs capable de détecter et d'avertir les troupes sur l'utilisation d'armes biologiques.

L'utilisation des RdCs dans le domaine militaire peut ainsi avoir deux fonctions : l'une défensive, pour repérer et analyser un ennemi, une zone de guerre ou détecter un risque biologique ou nucléaire; l'autre offensive, pour identifier l'avancée et l'état de ses troupes ou téléguider des drones d'attaque.

L'évolution récente des RdCs par l'armée laisse présager un avenir où les actions militaires seront semi-automatisées afin de réduire les risques de pertes humaines.

Toutefois, il faut noter que l'utilisation des réseaux de capteurs à des fins militaires requiert une sécurité supérieure à tout autre domaine. En effet l'utilisation de réseau de capteurs peut avoir une incidence si des informations stratégiques sont récupérées par un ennemi. Au vue des enjeux liés à l'utilisation des réseaux de capteurs, il est donc avéré que la sécurité dans ce type de réseaux est d'une importance stratégique, voir vitale, puisque leur bon fonctionnement met en jeu des vies humaines.

I.4.5. Surveillance des infrastructures

Un autre domaine d'applications des RdCs concerne la surveillance des infrastructures. Ces infrastructures peuvent être par exemple des ponts ou des bâtiments.

Le but de ces applications consiste à surveiller l'état d'un pont [14] ou d'un bâtiment pour détecter des détériorations trop importantes, ou des oscillations dans le cas de ponts suspendus, qui pourraient engendrer de graves dégradations voir la destruction complète de la structure (comme dans le cas de l'effondrement du pont de Minneapolis dans le Minnesota (USA) le 2 Août 2007).

D'autres applications ont pour objectif de surveiller des installations comme les oléoducs et les gazoducs, pour vérifier par exemple la pression ou le débit circulant dans les tuyaux.

Si l'emploi de ces solutions est étroitement lié à la sécurité, le besoin de sécurisation des communications au sein de ces applications reste faible. Le principal risque encouru est la détérioration gratuite du réseau ou des capteurs.

I.5 Spécificités des RdCs

I.5.1 Topologie

La topologie que l'on retrouve de manière classique au sein des RdCs est représentée par la figure I.4: un ensemble de capteurs qui sont déposés de manière hétérogène sur une zone d'intérêt. Tous ces capteurs communiquent entre eux et chaque capteur peut communiquer directement avec les autres capteurs qui sont situés dans sa zone de couverture. Un capteur qui souhaite communiquer avec un capteur distant doit faire transiter son message à travers les autres capteurs du réseau pour l'atteindre. On parle dans ce cas de figure de communication multi-saut.

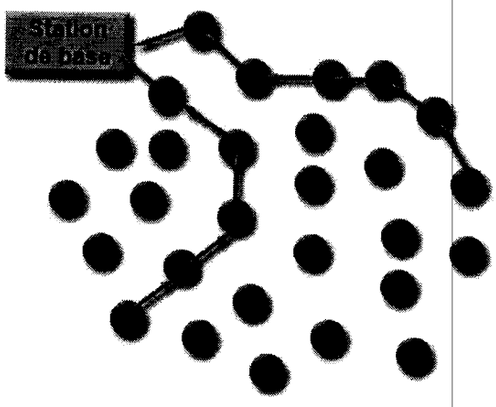


Figure I-0-4: Topologie d'un réseau de capteurs

Par ailleurs, les RdCs possèdent le plus souvent une, voire plusieurs stations de base. Ces stations de base ont pour mission de collecter les informations circulant dans le réseau, de les stocker ou de les transmettre directement ou périodiquement via une liaison Internet ou une liaison GSM à une autre entité. Les stations de base peuvent être un ordinateur portable ou un capteur de puissance plus grande que les autres capteurs classiques. Elles ont le plus souvent un rôle de contrôleur du réseau et .servent d'intermédiaire entre les utilisateurs du réseau et le réseau lui même.

Il existe d'autres topologies où les stations de base peuvent être mobiles, comme par exemple une station de base fixée à un avion ou un drone, afin de survoler la zone de couverture du réseau pour récupérer les informations des capteurs.

I.5.2 Medium

Le médium utilisé par les RdCs est l'onde radio. Trois grandes normes radios ont été utilisées pour des applications à bases de réseaux de capteurs. Ces trois normes sont:

- IEEE 802.11x/WiFi: WiFi est le protocole le plus utilisé pour toutes les applications sans fil. Il offre une large bande passante (11 à 320Mbits/s). Les premiers capteurs ont eu recours à ce protocole pour permettre la communication entre capteurs (seule solution effective à l'époque). Cependant, ce protocole n'apparait plus actuellement comme une solution viable pour les RdCs du fait d'un besoin énergétique trop important pour son utilisation. La durée de vie de capteurs alimentés par des piles ne dépasse que rarement quelques heures. C'est pourquoi les applications de capteurs à base de communication sans fil WiFi sont très peu répandues et en désuétude.

- IEEE 802.15.1/Bluetooth: Bluetooth avait pour objectif préalable de permettre des communications pour les WPANs[3], c'est-à-dire des communications sur de courtes distances avec un débit limité (1Mbits/s) et une durée de vie prolongée. Par exemple les capteurs BtNode sont conçus pour une communication de type Bluetooth. Pour autant, le protocole Bluetooth n'est pas le protocole le plus utilisé dans les RdCs, bien qu'il puisse répondre en partie aux problèmes de préservation de l'énergie, car il est gravement handicapé par la taille limitée du réseau qu'il peut former (8 nœuds : 1 maître et 7 esclaves). Ce faible nombre de nœuds est incompatible avec la volonté de former des RdCs denses.
- IEEE 802.15.4/Zigbee : Zigbee est basé sur le standard IEEE 802.15.4 qui définit sa couche PHY et MAC. Il est développé par la communauté industrielle Zigbee Alliance, conglomérat d'entreprises réunies dans le but commun de définir un protocole de communication avec un faible coût énergétique et un faible coût de production des composants nécessaires pour son utilisation. ZigBee permet de prolonger théoriquement la durée de vie d'un capteur sur plusieurs années. L'autre point fort de ce protocole est qu'il propose le déploiement de réseau dense à plus de 65000 nœuds avec une portée de l'ordre de 100 mètres pour un débit de 250 Kb/s. Ces spécificités en font aujourd'hui le principal protocole utilisé dans les RdCs.

Le tableau I.2 reprend les différentes caractéristiques de ces trois protocoles de communication sans fil.

Tableau 2: Comparaison des protocoles de communication pour les RdCs

Protocole	Bluetooth	WiFi	ZigBee
Norme IEEE	802.15.1	802.11x	802.15.4
Durée de vie moyenne sur pile	plusieurs jours	plusieurs heures	plusieurs années
Nombre de nœuds	8	2007	65 536
Débit théorique	1 Mb/s	320 Mb/s	250 Kb/s
Bande de fréquence	2.4 GHz	2.4 GHz ; 5 GHz	868/915MHz;2.4 GHz
Portée théorique	100 mètres	300 mètres	100 mètres

En outre le tableau I.2 illustre que la norme 802.15.4/Zigbee est plus intéressante que les autres technologies de communication utilisées par les réseaux de capteurs.

[3] Wireless Personal Area Network

D'une part elle offre la possibilité de faire communiquer un grand nombre de capteurs que dans WiFi ou Bluetooth, et d'autre part son autonomie est plus adaptée à des solutions à base de réseau de capteurs.

I.5.3 Routage des données

L'énergie est une ressource précieuse dans les réseaux de capteurs et les communications consomment plus d'énergie comparativement aux réceptions et traitements de données. De ce fait, les réseaux de capteurs sans fil requièrent des protocoles de routage efficaces visant à minimiser le nombre de communications [15]. Une des solutions employées par ces protocoles de routage est la clusterisation, qui consiste à diviser le réseau en plusieurs clusters. Dans chacun d'eux, un nœud bien particulier appelé clusterhead est élu et aura pour mission de récupérer les informations des nœuds du cluster et les transmettre aux autres clusters et inversement. Le choix du clusterhead est en fonction de certaines métriques telles que l'énergie restante, le nombre de voisins, etc... La figure I.4 représente un exemple de réseau clustérisé où les nœuds A, B et C ont été respectivement élus clusterheads des clusters 1, 2 et 3.

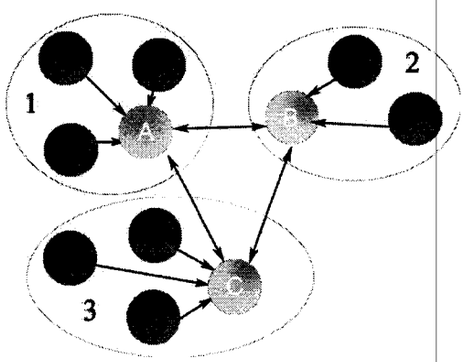


Figure I-0-5: Clustérisation d'un réseau de capteurs

En outre, d'autres problèmes de routage doivent aussi être pris en compte pour limiter le nombre de communications comme les problèmes d'implosion ou de chevauchement.

Le problème d'implosion est représenté par la figure I.6, où un nœud A va envoyer l'information a deux de ses voisins B et C, qui vont envoyer tous les deux l'information à leur nœud voisin D en absence d'un protocole efficace. Cette action aura pour effet une redondance de l'information, une consommation double de l'énergie du réseau, voir dans un cas critique, une collision lors de l'envoi simultané des deux capteurs.

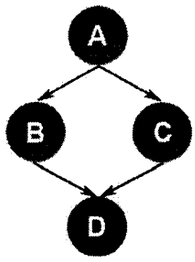


Figure I-0-6: Problème d'implosion (overlap)

Le problème du chevauchement est représenté par la figure I.7, où deux capteurs A et B qui surveillent leur zone respective 1 et 2, et partage une zone commune 3, détectent en même temps la même information en zone 3 et qui vont envoyer chacun d'eux la même information au nœud C, d'où une redondance de l'information, une surconsommation du réseau et un risque de collision.

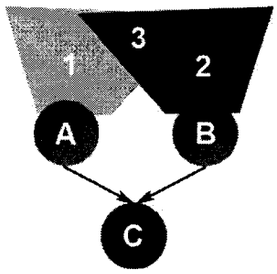


Figure I-0-7: Problème de chevauchement (Overhearing)

Ces deux problèmes montrent le besoin dans un protocole de routage efficace d'une négociation préalable entre nœuds avant l'envoi d'information.

I.5.4 Tolérance aux fautes

Dans les RdCs, un ou plusieurs capteurs peuvent ne pas fonctionner correctement, car les capteurs sont des entités sensibles aux altérations d'états comme des phénomènes climatiques (humidité, chaleur, électromagnétisme) ou du fait d'une batterie faible. Dans ces cas de figure, le réseau doit être capable de détecter ce type d'erreurs et d'y remédier, en cherchant par exemple à modifier ses tables de routage pour trouver un autre chemin afin de transmettre l'information et permettre que le réseau soit toujours opérationnel. De la même manière, on doit pouvoir trouver le moyen de détecter les capteurs qui envoient des informations erronées du fait de leur état défaillant.

I.5.5. Mise a l'échelle

Le nombre de capteurs utilisés dans les RdCs peut varier de quelques entités à plusieurs dizaines de milliers. C'est d'ailleurs la principale utilité des RdCs qui doivent

pouvoir s'auto-organiser à une grande échelle et être efficace quelque soit leur nombre. Pour cela les protocoles conçus pour les RdCs doivent être capables de fonctionner et de s'adapter selon le nombre de nœuds.

I.5.6 Energie

Les capteurs sont équipés de batteries, comme par exemple des piles LR6 dans le cas des MicaZ [16] ou TelosB. L'énergie de ces batteries est limitée (plusieurs jours à quelques années). De plus, les RdCs sont généralement déployés dans des zones hostiles. Il devient alors inenvisageable de vouloir changer les batteries des capteurs. Si le nombre des capteurs dépasse la centaine d'entités, il est encore plus difficile d'intervenir pour trouver le capteur défaillant et changer sa batterie. La consommation de l'énergie des RdCs doit être la plus faible possible. Dans ce but, les capteurs actuels ont des périodes de veille durant leur inactivité pour préserver leur batterie. En plus, les communications consomment plus d'énergie comparativement aux autres opérations. Donc, il est fortement nécessaire de limiter le nombre de communications redondantes entre capteurs.

I.5.7 Puissance de calcul

Malgré les progrès récents dans la fabrication de capteurs de plus en plus puissants, les capteurs actuels souffrent d'un manque de puissance de calcul (par exemple seulement 16 Mhz de puissance et 128Koctets de mémoire programmable pour un capteur MicaZ). Cette faible puissance ne permet pas d'utiliser des algorithmes complexes, et particulièrement des algorithmes cryptographiques gourmands en ressources CPU. Or, la faiblesse de puissance de calcul est aussi préjudiciable pour le temps de réponse du réseau. Si l'on demande à un capteur d'effectuer de nombreux calculs, la latence va sensiblement augmenter.

I.5.8 Sécurité

Comme évoqué précédemment les RdCs nécessitent dans de nombreuses applications des solutions qui assurent la sécurité des informations circulant dans le réseau. Cette sécurité doit répondre à plusieurs pré-requis:

- Confidentialité des données: le réseau doit s'assurer que les données transmises soient confidentielles et ne puissent être lues par des dispositifs ou personnes autres que ceux ayant droit de le faire. Les données doivent être cachées ou cryptées de telle manière que personne ne puissent y accéder. La confidentialité des données est

prépondérante dans des applications de types médicales où les informations du patient ne doivent pas être divulguées. Il en est de même pour des applications militaires où ces informations peuvent avoir une conséquence stratégique sur des actions en cours.

- Intégrité des données : Les données circulant dans le réseau ne doivent pas pouvoir être altérées au cours de la communication. Il faut donc s'assurer que personne ne puisse capturer et modifier les données du réseau. De la même manière, il faut vérifier que les données n'ont pas subi d'altération due à un disfonctionnement du matériel, qui est un risque important sur des capteurs sensibles aux altérations d'états.
- Fraîcheur des données : Cela signifie qu'il faut s'assurer que la donnée transmise est récente et corresponde à un état présent. Sans mécanisme de sécurité vérifiant que les données transmises sont récentes, un attaquant pourrait capturer des informations circulant dans le réseau à une date T, puis les retransmettre à une date T+1 pour tromper le réseau et faire circuler de fausses informations. On peut prendre pour exemple un RdCs pour les feux de forêts, qui détecterait une première fois un incendie réel. L'attaquant enregistrerait les informations envoyées lors de cet événement. Il pourrait alors plus tard renvoyer de nouveau ces mêmes données pour déclencher une fausse alerte.
- Disponibilité du réseau : Le réseau doit pouvoir être disponible à tout instant, c'est-à-dire que l'envoi d'information ne doit pas être interrompu, de même que la circulation de l'information ne doit pas être stoppée. Dans le cas d'un réseau de capteurs réactifs, il faut qu'un capteur qui détecte un événement puisse transmettre à tout instant cette information vers la station de base pour l'en informer.
- Auto-organisation : Les capteurs du réseau doivent être capables, après avoir été déployés, de s'auto-organiser et surtout de se sécuriser eux-mêmes, sans autres interventions extérieures. Ce besoin d'auto-organisation se retrouve dans l'établissement automatique de la distribution des clés de cryptage entre les nœuds du réseau et la gestion de ses clés ou bien encore dans le développement des relations de confiance entre capteurs du réseau (principalement dans l'utilisation de sécurité utilisant les principes des réseaux de confiance). Pour cela les capteurs doivent avoir été munis au préalable des outils qui leur permettent de telles fonctionnalités.

- Localisation sécurisée : Le besoin de se localiser et de connaître la position des autres nœuds peut être primordial dans de nombreux cas pour déjouer d'éventuelles attaques jouant sur les distances, attaques détaillées dans le chapitre suivant.
- Temps synchronisé: De nombreuses solutions de sécurité nécessitent des capteurs synchronisés pour qu'elles soient effectives. Il faut ainsi s'assurer que les capteurs du réseau ou des sous-réseaux du réseau ont une horloge commune afin par exemple d'éviter des attaques de type rejeu de paquets.
- Authentification : L'authentification des capteurs est nécessaire pour s'assurer que l'identité déclarée par un capteur est bien celle du capteur déclarant. En l'absence d'un mécanisme permettant d'authentifier clairement un nœud du réseau, de nombreuses attaques peuvent se mettre en place.

Chapitre II

Mécanismes de sécurité pour les réseaux de capteurs

Chapitre II

Mécanismes de sécurité pour les réseaux de capteurs

II.1 Introduction

Les réseaux de capteurs sont exposés à de nombreuses menaces. Si certaines de ces menaces peuvent se retrouver dans les réseaux ad-hoc, d'autres sont spécifiques à ce type de réseau et profitent des caractéristiques contraignantes de ces réseaux telles que l'énergie limitée, faible puissance de calcul, utilisation des ondes radio, etc...

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, l'attaquant ne cherchant ici qu'à écouter et récupérer les informations. 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.2 Taxinomie des attaques dans les réseaux de capteurs

Dans cette section, nous décrivons une liste non exhaustive mais représentative des attaques les plus courantes et connues, actives ou passives, que nous pouvons trouver dans les réseaux de capteurs sans fil.

II.2.1 Ecoute passive

Cette attaque consiste à écouter le réseau et à intercepter les informations circulant sur le médium. Elle est facilement réalisable si les messages circulant sur le réseau ne sont pas cryptés. Par ailleurs l'écoute passive est difficile à détecter car elle ne cherche pas à modifier l'activité du réseau.

II.2.2 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.

Le problème du chasseur de panda [17] est le parfait exemple des risques encourus par le réseau qui ferait face à une analyse du trafic. Dans ce cas d'étude, les

différents paquets envoyant des informations sur la position du panda, détectés par les capteurs du réseau, permettent à un attaquant de connaître la zone où se trouve le panda en analysant la zone de trafic de ces paquets.

L'analyse du 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.2.3 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 RdCs 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.2.4 Attaque sur la couche de lien

La couche de lien dans les RdCs a pour principale fonction de gérer l'accès au médium et contrôler les erreurs. L'attaque sur cette couche [18] est un autre type d'attaque de type déni de service, qui consiste à provoquer des collisions lors de l'envoi de message afin, d'une part d'empêcher la transmission d'un paquet, et d'autre part obliger le capteur émetteur à retransmettre le paquet. Pour cela, un attaquant peut envoyer continuellement des informations sur le réseau où il peut plus stratégiquement violer les protocoles de communication pour créer des collisions lors des envois de paquets d'acquittement. Ainsi, si le protocole de communication de la couche de lien spécifie à un capteur de renvoyer le paquet jusqu'à ce qu'il soit acquitté, le capteur va épuiser sa batterie en renvoyant continuellement le même paquet.

II.2.5 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 [18].

II.2.6 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 [19], 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 $N_1(X)$ (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 $N_1(X)$ 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.

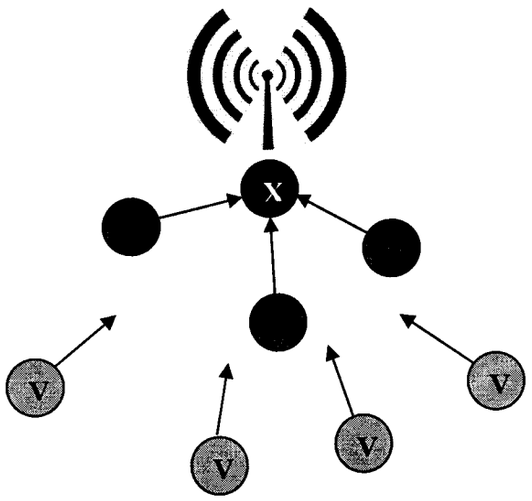


Figure II-1 : Attaque de type Hello flooding

II.2.7 Injection de messages

L'attaquant cherche par divers moyens (utilisation de nœuds malicieux, envoi de paquets sur la même fréquence radio, etc. . .) à injecter des messages dans le réseau. Cette injection de messages peut avoir pour effet de perturber le réseau, le saturer ou le tromper en envoyant de fausses informations.

II.2.8 Réplication des données

Si les paquets envoyés dans le réseau peuvent être lus et enregistrés par un attaquant. Ce dernier peut renvoyer ces mêmes paquets à une date ultérieure pour tromper le réseau. Pour illustrer cette attaque, on peut prendre pour exemple un réseau de capteurs qui a pour mission de détecter un incendie: si un premier incendie est détecté et qu'un capteur envoie un paquet d'alerte pour en informer la base, l'attaquant pourra enregistrer ce paquet, même s'il est chiffré et qu'il ne peut le déchiffrer, puis l'émettre à une autre date postérieure et faire croire à un nouvel incendie.

Cette attaque est réalisable si le paquet ne contient pas d'informations concernant la date de l'envoi ou si cette date est accessible et facilement modifiable par un attaquant.

II.2.9 Destruction ou vol

Les attaques actives les plus élémentaires dans les RdCs sont le vol et la destruction. Ces attaques sont facilitées par le fait que les capteurs sont le plus souvent déployés dans des zones qui ne peuvent être surveillées ou hostiles. De ce fait, une personne physique peut subtiliser un ou plusieurs capteurs, voire les détruire. Dans ce cas, le réseau doit être capable de s'adapter à la situation, sous peine de ne plus fonctionner, ou d'être coupé en plusieurs sous-réseaux incapables de communiquer entre eux car les nœuds qui faisaient le pont entre sous-réseaux sont détruits ou subtilisés. En plus, un nœud volé, peut divulguer certaines informations à un attaquant.

II.2.10 Nœud compromis

La plupart des RdCs sont déployés dans des zones larges et hostiles qui ne permettent pas une surveillance humaine de l'ensemble des capteurs. Il est alors tout à fait possible pour un attaquant de compromettre un capteur. 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. 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 nœud.

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. Cependant la plupart des capteurs utilisés aujourd'hui sont très vulnérables aux attaques physiques, comme démontré par [20] qui a prouvé qu'un capteur de type Mica2 peut être compromis dans un temps inférieur à une minute.

II.2.11 Attaque du trou noir

L'attaque du trou noir consiste tout d'abord à insérer un nœud malicieux par divers moyens dans le réseau [19]. Ce nœud va modifier les tables de routage pour obliger le maximum de nœuds voisins à faire transiter leurs informations par lui. Ensuite, toutes les informations qui passent par ce nœud ne seront jamais retransmises.

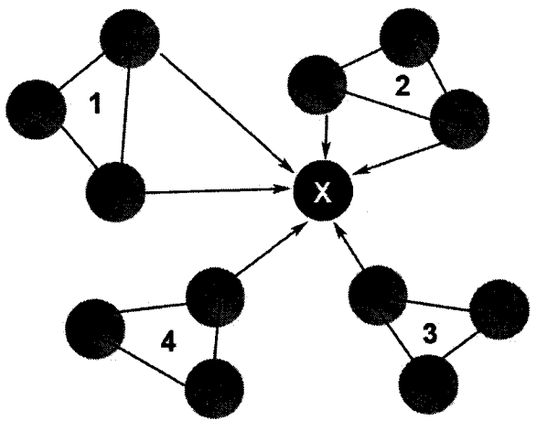


Figure II-2: Attaque du trou noir

La figure II.2 représente une attaque de type trou noir où un nœud malicieux X qui a modifié la table de routage des clusters 1, 2, 3 et 4 en faisant passer l'information par lui pour communiquer entre clusters. Dans ce cas de figure, le nœud X ne retransmettra aucune information, ce qui empêche toute communication entre les différents clusters qui représentent des sous-réseaux.

II.2.12 Attaque du trou gris

L'attaque du trou gris est une variante plus subtile de l'attaque du trou noir [19]. Tout comme le trou noir, il s'agit d'un nœud malicieux qui va être inséré dans le réseau et qui va modifier les tables de routage pour faire transiter un maximum d'informations par lui. Contrairement au trou noir, le trou gris relaye certaines informations. Par exemple, le trou gris peut relayer toutes les informations concernant

le routage, mais ne le fera pas pour des informations critiques. Ce type d'attaque est ainsi plus difficile à détecter que l'attaque du trou noir, car le capteur malicieux tant qu'il se comporte de manière normale ne peut être facilement détecté.

II.2.13 Attaque du trou de ver

L'attaque du trou de ver nécessite l'insertion dans le réseau de capteurs d'au moins deux nœuds malicieux [21]. Ces deux nœuds sont reliés entre eux par une connexion puissante qui peut être filaire ou radio (connexion WiFi à 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. Dans le cas d'une attaque du trou de ver, les deux nœuds malicieux permettent d'atteindre un lieu éloigné en un seul saut. 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.

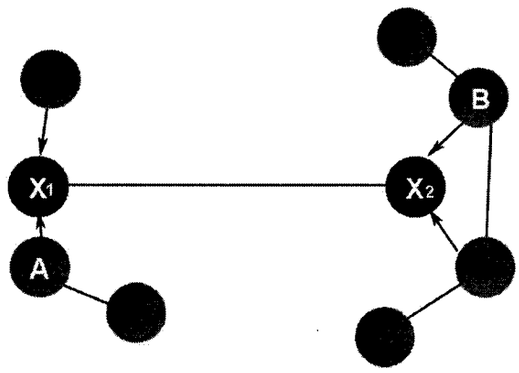


Figure II-0-3: Attaque du trou de ver

La figure II.3 illustre une attaque du trou de ver où deux nœuds malicieux X_1 et X_2, reliés par une connexion puissante, et forment un trou de ver. Les nœuds A et B vont alors privilégier la route la moins longue formée par le trou de ver, envoyer respectivement leurs paquets de données aux nœuds X_1 et X_2 et donc ces informations pourront être récupérée par l'attaquant.

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 1.9, un paquet du nœud A va être enregistré par le nœud X_1 puis répliqué par le nœud X_2 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.

II.2.14 Attaque du trou de la station de base

Dans cette attaque, un nœud malicieux s'attaque directement à l'information circulant par et vers la station de base [19]. Pour cela, le nœud malicieux propose aux nœuds du réseau le chemin le plus court pour atteindre la station de base, en utilisant par exemple une connexion plus puissante, comme représenté dans la figure II.4. Ainsi l'ensemble de ces nœuds s'adresse en particulier à ce nœud malicieux pour transmettre l'information à la station de base. Toute information qui transite par ce nœud, sera récupérée par l'attaquant.

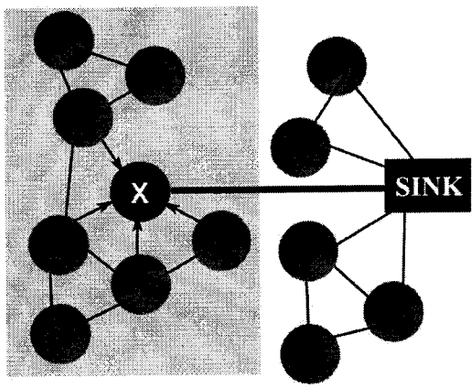


Figure II-0-4: Attaque du trou de la station de base

Pour générer une attaque encore plus puissante, un attaquant peut utiliser des attaques de type trou de ver associées à une attaque de type trou de la station de base. Le but est d'utiliser ces trous de ver pour couvrir tous les nœuds du réseau, comme représenté dans la figure II.5, où les nœuds malicieux X_1, X_2 et X_3 sont reliés par des connexions puissantes et forment des trous de ver. X_3 est lui relié à la station de base par une connexion puissante pour réaliser une attaque du trou de la station de base. Dans ce cas, l'attaquant est capable de récupérer de toutes les informations qui circulent dans le réseau de capteurs via les trous de ver et le trou de la base.

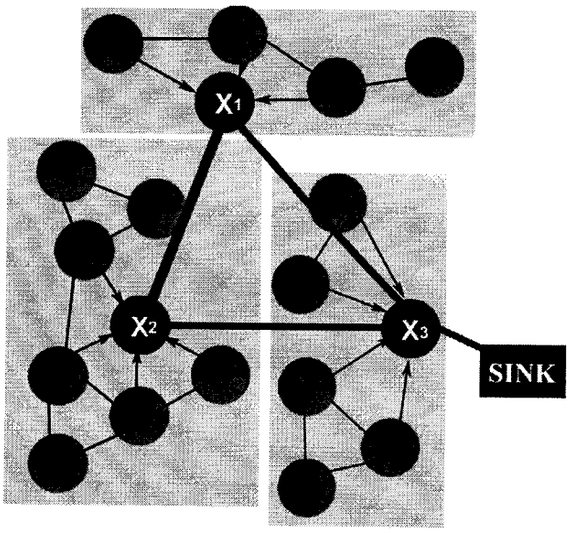


Figure II-0-5: Attaque du trou de ver pour réaliser une attaque du trou de sink

II.2.15 Attaque Sybille

Une attaque Sybille ou "Sybil attack" [22] 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 Sybille 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 clusterhead. D'où, 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 Sybil peut aussi lui permettre de devenir le chien de garde.

II.2.16 Insertion de boucles infinies

Un attaquant peut modifier le routage du réseau avec un ou plusieurs nœuds malicieux, dans le but d'envoyer des messages qui seront routés en boucles infinies.

Comme ce message va être envoyé par le réseau de manière infinie, les capteurs épuiseront leurs batteries.

II.2.17 Altération des messages

Un nœud malicieux va récupérer un message et l'altérer, en lui ajoutant des fausses informations (sur le destinataire, l'émetteur, l'information en elle-même), en le modifiant ou bien en détruisant des paquets pour rendre incompréhensible le message.

II.2.18 Ralentissement

Un attaquant peut programmer des nœuds malicieux qui seront comme des agents dormants et qui n'auront que pour but de ralentir l'information (par exemple avec une attaque de type trou gris). Le nœud malicieux quand il recevra une information importante qui nécessite d'en informer très rapidement le réseau va volontairement attendre un certain temps avant de relayer cette information. L'information ralentie permettra à l'attaquant d'en tirer un avantage.

II.2.19 Privation de mise en veille

Cette attaque active a pour but de priver un capteur de se mettre en veille par différents moyens. Le capteur s'il ne peut plus se mettre en veille va consommer très rapidement sa batterie, jusqu'à se retrouver non opérationnel.

II.2.20 Attaque spécifique au type de capteur

Ce type d'attaque dépend du capteur en lui même. Un attaquant modifie de manière physique le comportement du capteur. Il peut par exemple allumer une flamme devant un capteur thermique ou bien allumer une lampe devant un capteur de luminosité. Le but est de tromper le capteur, et ainsi d'envoyer ou d'enregistrer de fausses informations sur le réseau, ou bien tout simplement de faire réagir assez longtemps un nœud ou le réseau pour qu'ils consomment leur énergie, comme dans le cas d'une attaque de type privation de mise en veille.

II.2.21 Attaque sur les Pacemakers

L'attaque sur les pacemakers 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.

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. Ce paramétrage à distance évite de devoir réopérer le patient après l'implantation du pacemaker, pour le paramétrer de nouveau.

Dans [23], les auteurs ont montré qu'il était 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 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 porteuses de pacemakers qui pourra entraîner la mort sans intervention rapide sur le patient.

II.3 Mécanismes de sécurité déployés

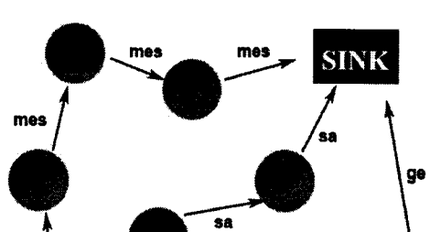
Les attaques présentées précédemment montrent l'étendue des possibilités d'attaque d'un réseau de capteurs et les différents points de sécurité (intégrité du réseau, authentification, confidentialité des données, etc. . .).

Il s'avère impossible de préconiser une seule solution pour contrer l'ensemble de ces attaques. Néanmoins la littérature récente cherche à trouver diverses solutions basées sur de nombreux mécanismes de sécurité pour répondre à ces attaques, tout en pensant à la problématique de la consommation d'énergie inhérente aux RdCs.

Nous présentons dans ce qui suit quelques unes des solutions de sécurité les plus répandues, ainsi que les protocoles de sécurité proposés par la communauté scientifique pour sécuriser les RdCs.

II.3.1 Partitionnement des données

Dans [24], les auteurs ont proposé une solution pour empêcher la récupération d'information dans les RdCs en partitionnant les données qui devront être envoyées par un capteur en plusieurs paquets de taille fixe. Chaque paquet sera ensuite envoyé sur des chemins différents, c'est à dire que ces paquets ne passeront pas par la même route. Ces paquets seront finalement reçus par la station de base, qui pourra ensuite les rassembler pour pouvoir reproduire l'information. Ce mécanisme oblige un attaquant à récupérer l'ensemble des paquets s'il veut pouvoir lire l'information. Il doit aussi être capable d'écouter l'ensemble du réseau, pour récupérer les différents paquets qui circulent sur des chemins différents. La figure II.6 illustre le principe de fonctionnement de cette solution, où un capteur A divise un message en trois paquets qui vont suivre respectivement trois chemins différents.



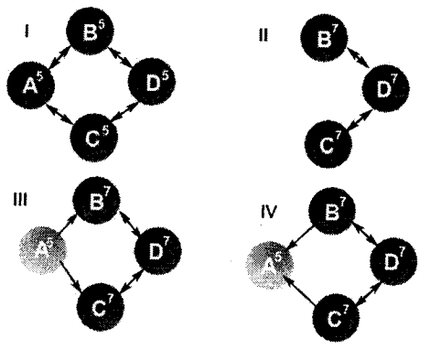


Figure 0-7: Détection de nœud malicieux par génération de clé

La figure II.7 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 leur clés, ils se sont aperçus qu'elles ne correspondaient pas.

Si cette solution permet effectivement de détecter les nœuds malicieux, 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.3.3 Localisation

Un mécanisme utilisé pour détecter particulièrement 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, 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. La figure II.8 montre un exemple de vote entre quatre capteurs de type balise A, B, C et D qui ont quadrillé leur zone d'écoute respective et qui ont chacun voté pour chaque zone de quadrillage. Suite à ce vote, ils peuvent estimer la position du capteur qu'il recherche et qui doit se trouver dans la zone avec le maximum de votes c'est à dire dans cet exemple la zone avec trois votes.

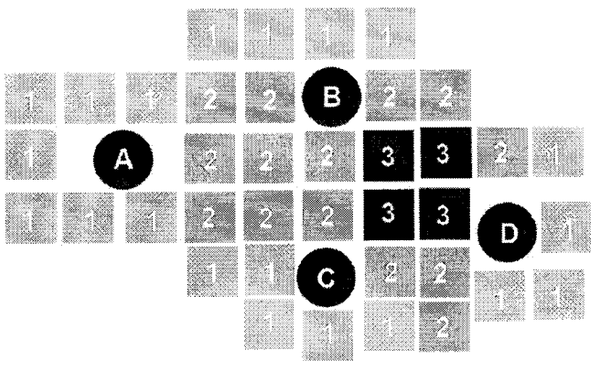


Figure 0-8: Localisation du signal avec capteur de beacon

Dans le cas d'une attaque dite du trou de ver, les deux nœuds malicieux qui tentent une attaque vont être géo-localisés par les nœuds balises qui seront donc capables de déterminer que la distance entre ces deux nœuds est plus grande que la distance normale pour une communication en un seul saut, et ainsi détecter l'attaque.

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.3.4 Chien de garde

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.

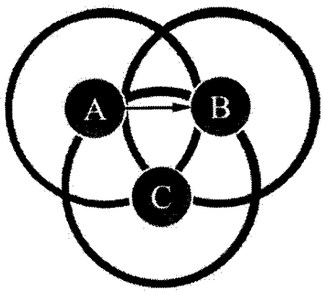


Figure 0-9: Mécanisme de chien de garde (Watchdog)

La figure II.9 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.3.5 Cryptographie

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

Dans le cadre des réseaux filaires et des réseaux sans fil traditionnels disposant d'une capacité de calcul et de mémoire conséquente, 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 (DES, RSA, etc. . .).

Les travaux de recherche actuels s'attachent à trouver des solutions dites de cryptographie légères [28]. Ces solutions consistent à adapter les algorithmes de cryptographies classiques pour les RdCs. Ces différentes solutions de cryptographie sont de deux types: symétrique et asymétrique.

Cryptographie à clé symétrique

Le principe de la cryptographie symétrique se base sur le partage d'une même clé K de chiffrement entre deux entités, communément nommé Alice et Bob.

Si Alice souhaite envoyer un message à Bob, il chiffrera son message M avec la clé K, puis transmettra son message crypté M_K à Bob. Ce message ne peut être déchiffré sans la clé K, ce qui garantit sa confidentialité. Quand Bob reçoit le message M_K, il déchiffre le message M avec la clé K de chiffrement.

Les algorithmes de cryptographie symétrique se décomposent en deux sous ensembles, les algorithmes de chiffrement par flux et les algorithmes de chiffrement par blocs.

Chiffrement par flux:

Dans cette classe, la technique utilisée consiste à chiffrer le message à transmettre en effectuant un XOR avec la clé de chiffrement.

Soit M le message à chiffrer, K la clé de chiffrement, et $\oplus$ l'opération booléenne XOR, le chiffrement correspond à

$$M \oplus K = M_k$$

où M_k est le message chiffré.

Le déchiffrement se fera alors par:

$$M_k \oplus K = M \oplus K \oplus K = M$$

Cette technique de chiffrement par flux est celle utilisée entre autre par l'algorithme de chiffrement RC4.

Chiffrement par blocs:

Le chiffrement par blocs consiste à découper un message M en blocs de n bits. Ces blocs seront ensuite chiffrés par une fonction F et une clé k extraite d'une clé maître K.

Soient M le message à chiffrer et K la clé de chiffrement dont sont extraites les clés k et F la fonction de chiffrement. M sera découpé en r blocs de n bits. Pour chaque bloc b_x de M, le chiffrement se fera de la manière suivante:

$$C_1 = F(k_1, b_x)$$

F est ensuite itérée avec une nouvelle clé extraite de la clé maître K pour garantir la sécurité de l'algorithme de chiffrement, ainsi:

$$C_2 = F(k_2;C_1)$$
$$C_y = F(k_y;C_{y-1})$$

Le déchiffrement se fait avec une fonction G, inverse de la fonction F et les différentes clé k partagées extraites de la clé commune K, de la manière suivante:

$$C_{y-1} = G(k_y,C_y) = G(k_y, F(k_y,C_{y-1}))$$
$$b_x = G(k_1,C_1)$$

La cryptographie symétrique par blocs est la technique de cryptographie symétrique la plus répandue, utilisée entre autre par les algorithmes de cryptographies comme DES, AES ou Blow-fish.

L'algorithme de cryptographie DES se base sur un chiffrement avec des clés de 56 bits et était jusqu'en 2001 recommandé pour tout chiffrement, mais aujourd'hui il est considéré comme non sûr car il existe des attaques qui permettent casser la clé de chiffrement dans un bref temps. Il en résulte qu'il n'est pas utilisé dans les solutions de cryptographies pour les RdCs.

L'algorithme de chiffrement par blocs recommandé depuis 2001 est l'algorithme AES avec une clé 128 bits. Cet algorithme a été proposé comme algorithme de chiffrement dans le standard IEEE 802.15.4 pour sécuriser au niveau de la couche MAC les données transitant sur des RdCs. Le temps d'exécution de cet algorithme de chiffrement donne d'assez bons résultats (quelques microsecondes) et limite le surcoût énergétique du chiffrement de données. Néanmoins ces bons résultats sont obtenus par le fait que le chiffrement est effectué au niveau hardware par le module

transceiver. Cependant, dans le cas d'un chiffrement software cela nécessite plusieurs millisecondes [29].

Des algorithmes de chiffrement symétriques ont été testés sur les capteurs dont les résultats sont visibles dans [30]. Ces résultats permettent de dire qu'aujourd'hui les solutions de chiffrement à clés symétriques sont exploitables au sein des RdCs. Cependant, il est inenvisageable dans un réseau de capteurs dense, que chaque capteur, au vue de la taille réduite de sa mémoire, embarque une clé pour chaque nœud du réseau. Pour un réseau de taille N, il faudrait alors que le capteur ait en mémoire (N-1) clés dans sa mémoire pour pouvoir lui envoyer un message sécurisé. En prenant l'exemple d'un réseau de capteurs dense de taille 1000 avec une sécurisation par clé de 128 bits, il lui faudrait alors 999 clés, soit 999*128 = 127872 bits en mémoire, soit supérieur au 128Ko de la mémoire d'un capteur de type MicaZ.

En outre, le problème heurté dans les crypto-systèmes symétriques réside dans la distribution des clés. Chacune des solutions suivantes répond à un besoin différent selon le type d'application pour laquelle un réseau de capteur est déployé (récupération d'information uniquement par la station de base, échange de données entre capteurs collaboratifs, etc. . .). Les quatre principales solutions de distribution des clés sont les suivantes :

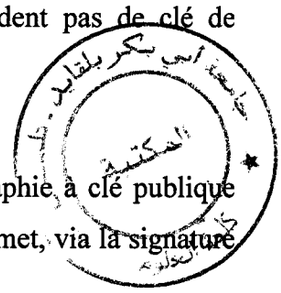
- **Clé globale**: Une clé est partagée par l'ensemble des nœuds du réseau. Pour envoyer un message, l'information est encryptée avec cette clé. Une fois le message reçu, il peut être décrypté avec cette même clé ou une autre déduite de celle-ci. Cette solution est une des solutions de cryptographie la moins coûteuse en énergie, puisque l'information n'est encryptée qu'une fois par l'émetteur et décryptée une seule fois par le récepteur. De plus, cette solution permet de résoudre en partie le problème de l'écoute passive, car l'information ne circule plus en clair. Cependant, si un attaquant arrive à trouver la clé, il est capable d'écouter l'ensemble du réseau.

- **Clé partagée par paire de nœuds**: Chaque nœud possède une clé différente pour communiquer avec un nœud voisin qui partage cette clé. Ainsi, si un nœud possède N voisins, il aura N clés à stocker pour pouvoir communiquer avec ses voisins. De ce fait, un nœud qui cherche à envoyer un message, doit l'encrypter avec la clé du voisin destinataire. Ce dernier devra décrypter l'information pour la ré-encrypter avec la clé correspondante au destinataire suivant. Cette solution, permet d'augmenter

considérablement la sécurité du réseau, car une clé découverte ne permet de communiquer qu'avec deux nœuds, et limite la possibilité de nuisance d'un attaquant. Cependant cette technique est très coûteuse en énergie et surtout en temps de calcul, car chaque paire de nœuds qui transmet l'information doit effectuer le travail d'encryptage et de décryptage. Ce qui aura pour effet de réduire la durée de vie du réseau mais aussi sa latence.

- **Clé partagée par groupes de nœuds**: Dans ce cas, chaque groupe partage une clé en commun qui lui permet de communiquer à l'intérieur du groupe. Les nœuds maîtres (clusterheads), communiquent entre eux avec, soit une clé commune à tous les clusterheads, soit une clé commune par paire de clusterheads. Cette solution est une solution hybride des deux premières techniques d'encryptage. Elle permet de limiter le nombre d'encryptages dans les communications. Cependant elle a pour défaut de reporter l'essentiel du travail d'encryptage sur les clusterheads. Pour rester efficace, il faut donc s'assurer de changer régulièrement de nœud maître dans un groupe pour ne pas consommer toute l'énergie du clusterhead.

- **Clé individuelle**: Dans cette solution, chaque nœud possède une clé pour crypter son information. Cette clé n'est connue que par la station de base. Un message envoyé par ce nœud circulera dans le réseau dans un état encrypté jusqu'à atteindre la station de base. Cette solution est une des meilleures techniques pour minimiser la consommation énergétique dans le réseau. Cependant, elle ne permet pas de sécuriser les informations transmises entre nœuds car ceux-ci ne possèdent pas de clé de chiffrement pour communiquer de manière sécurisée entre eux.



Cryptographie à clé publique

La cryptographie à clé asymétrique, appelée aussi cryptographie à clé publique est réputée plus sûre que la cryptographie symétrique car elle permet, via la signature numérique, d'authentifier l'auteur du message.

La cryptographie à clé publique se base sur l'utilisation de deux clés, une clé privée K_s gardée sécrète par son propriétaire et une clé publique K_p diffusée par son propriétaire pour permettre aux nœuds souhaitant lui communiquer des informations de chiffrer leur message avec la clé publique K_p.

La cryptographie à clé publique est fondée sur le principe de fonction à sens unique. Ainsi, un message crypté avec la clé publique K_p par Bob ne peut être

déchiffré avec cette même clé. Elle n'est déchiffrable que par le possesseur de la clé privée K_s à savoir Alice.

Formellement le chiffrement et le déchiffrement de données d'un message entre deux nœuds Alice et Bob correspondent au mécanisme suivant: soit K_p la clé publique et K_s la clé privée d'Alice, F la fonction de chiffrement et G la fonction de déchiffrement. Alice diffuse sa clé publique dans le réseau. Soit M le message que souhaite transmettre Bob à Alice, alors

$M_k = F(K_p,M)$ où M_k est le message chiffré,

$M \neq G(K_p, M_k)$

Bob envoie le message à Alice qui va ensuite pouvoir déchiffrer avec sa clé privée:

$M = G(K_s,M_k)$

Par ailleurs Bob peut demander à Alice de prouver son identité avec le mécanisme de la signature numérique. Pour cela Alice cryptera un message avec sa clé privée, Bob déchiffrera alors le message d'Alice avec la clé publique d'Alice. Comme seul Alice possède la clé privée correspondant à sa clé publique, si le message est déchiffrable avec sa clé publique, Bob est normalement assuré que le message provient effectivement d'Alice.

Les mécanismes de cryptographie publique sont potentiellement de bonnes solutions pour sécuriser les réseaux de capteurs en fournissant d'une part la confidentialité des données et d'autre part le mécanisme d'authentification qui leur fait défaut. Cependant le besoin de puissance CPU pour l'exécution d'algorithmes de chiffrement fait défaut.

Le tableau 1.3 montre les temps d'exécution nécessaire à l'exécution du protocole sécurisé d'authentification SSL/TLS utilisant l'algorithme de cryptographie RSA sur différents type de capteurs les plus répandus [31].

Tableau 3: Temps d'exécution sur capteurs du protocole d'authentification SSL/TLS

Type de capteur	RSA -1024
MICA2DOT	22.00 s
MICAZ	12.00 s
TelosB	5.70 s

A partir de ce tableau, on remarque que les temps d'exécution de quelques fonctions cryptographiques publiques dans les réseaux de capteurs posent de gros problèmes de latence et de consommation énergétique. Dans cette optique, des travaux de recherche essaient d'optimiser les algorithmes de chiffrement à clé publique comme présenté dans [32] où les auteurs proposent une version optimisée de RSA, appelée WM-RSA pour capteurs de type MicaZ. Bien que les résultats soient meilleurs que ceux de la version originale de RSA, les temps d'exécution restent toujours de l'ordre de la seconde. Un temps qui n'est pas envisageable dans des réseaux de capteurs qui nécessitent une intervention rapide. De plus la taille des clés nécessaires avec l'algorithme RSA à savoir 1024 bits peut poser des problèmes de stockage. Dans cette direction, des travaux récents tentent d'apporter une réponse avec l'utilisation de cryptographie à clé publique basée sur les courbes elliptiques (ECC). La cryptographie utilisant les courbes elliptiques nécessitent des tailles de clés inférieures à RSA. Le niveau de sécurité d'une clé de 160 bits utilisée avec des algorithmes de chiffrement à courbes elliptiques correspond à un chiffrement avec une clé de 2046 bits avec l'algorithme de chiffrement RSA. Dans [33], les auteurs ont proposé la librairie cryptographique TinyECC pour le système d'exploitation TinyOS, qui offre la possibilité de chiffrer et d'authentifier des données avec des algorithmes à base de courbes elliptiques (ECDSA, ECIES, ECDH). Le temps d'exécution de cette solution dépasse la minute à quelques millisecondes selon le type d'architecture utilisé. Cette solution ne peut donc pas être envisagée sur tout type de capteur. De plus la librairie TinyECC a été écrite pour la version 1.x de TinyOS et n'est plus compatible avec la version 2.x de TinyOS.

Dans le chapitre qui suit nous présentons la cryptographie basée sur les courbes elliptiques.

II.4. Protocoles de sécurité

Plusieurs protocoles ont été proposés pour sécuriser les RdCs. Certains d'entre eux visent seulement à détecter les attaques de type trou noir ou trou de ver. Toutefois, les protocoles de sécurité SPINS et TinySEC sont eux considérés comme des solutions de sécurité générales et apportent des solutions pour la confidentialité des données et leur authentification. Nous décrivons dans cette section les mécanismes utilisés par ces deux protocoles pour sécuriser les réseaux de capteurs.

II.4.1 SPINS (Security Protocols for Sensor Networks)

SPINS [34] est un protocole basé sur deux blocs de sécurité que sont SNEP et µTESLA [35].

SNEP

SNEP utilise deux mécanismes de sécurité, le premier consiste à chiffrer les données pour assurer leur confidentialité et le second de calculer un code MAC7 pour assurer l'authentification et l'intégrité des données entre deux entités.

Dans SNEP durant un premier échange de données entre deux nœuds, le nœud émetteur précède le message d'une chaîne de bits aléatoires, appelée vecteur initial, avant de l'encrypter avec une fonction de chiffrement de type DES-CBC. Puis le message chiffré sera ajouté au bloc suivant et ainsi de suite. Cette technique empêche un attaquant qui écoute le réseau et qui a en sa possession le même message chiffré précédemment, de pouvoir en déduire que le même message a été envoyé.

Les deux nœuds partagent ensuite un compteur qui leur permet d'utiliser des chiffrements par bloc en mode compteur (CTR) et de ne plus utiliser de vecteur initial. A chaque bloc échangé, le compteur est incrémenté. Or un attaquant ne peut décrypter l'information que s'il peut voir le même message plusieurs fois encryptés. L'utilisation d'un vecteur initial aléatoire et d'un compteur empêche cette possibilité, puisque un même message sera suivi en clair soit d'une chaîne de bits, soit d'un compteur incrémenté qui une fois chiffré sera à chaque fois différent. L'utilisation de ce compteur permet d'éviter les attaques par rejeu de paquet car chaque message est numéroté, et donc, garantit la fraicheur des données.

II.4.2 µTESLA

µTESLA permet l'utilisation de broadcast authentifié. C'est une version adaptée aux réseaux de capteurs sans fil du protocole TESLA.

µTESLA utilise une authentification symétrique liée à une méthode asymétrique où les clés symétriques sont divulguées au cours du temps. Pour permettre cette authentification, il est nécessaire que la station de base et les différents nœuds soient vaguement synchronisés. La station de base a pour rôle d'ajouter au paquet à envoyer un code MAC calculé à partir d'une clé qui reste secrète à cet instant. Un nœud qui reçoit ce paquet peut vérifier que la clé pour déchiffrer le code MAC n'a pas encore été divulguée grâce à son horloge de synchronisation. Si elle n'a pas encore été

divulguée, il peut en déduire que seule la station de base connaît la clé MAC et qu'aucun attaquant n'a pu altérer le message pendant son transit. Il peut alors stocker le paquet dans son buffer en attendant la prochaine divulgation de la clé. Quand la clé sera divulguée il pourra décrypter le message et vérifier son authenticité. Chaque clé K est une clé issue d'une chaîne de clé générée par une fonction à sens unique F, de telle manière que:

$$K_i = F(K_{i+1})$$

Cette clé de chiffrement pour code MAC est générée à des intervalles de temps réguliers de telle manière que si un capteur ne reçoit pas tous les paquets et donc toutes les clés, il sera capable de retrouver les anciennes clés en fonction de la dernière reçue. Ainsi, si un nœud du réseau possède la clé initiale K_0 et la clé K_2, mais n'a pas reçu la clé K_1, d'une part il peut vérifier que la clé K_2 est bien celle envoyée par la station de base car

$$K_0 = F(F(K_2))$$

et d'autre part il peut retrouver K_1 car

$$K_1 = F(K_2)$$

Si SPINS a été un des premiers protocoles à proposer une solution garantissant la confidentialité et l'authenticité des données. Son approche n'est pas sans faille parce qu'il utilise un algorithme de chiffrement DES, qui n'est plus réputé sûr. En outre, μTesla nécessite un envoi de données permanent aux capteurs qui a un coût non négligeable pour la durée de vie du réseau. En plus, il est à noter que ce protocole a été défini mais jamais implémenté, et donc nous ne pouvons pas savoir son efficacité en termes de consommation d'énergie et latence sur des cas réels.

II.4.3 TinySec

TinySEC est une librairie de sécurité intégrée dans le système d'exploitation TinyOS 1.x. L'objectif de cette librairie est de pouvoir détecter les paquets non autorisés lorsqu'ils sont injectés pour la première fois dans le réseau et éviter leur propagation dans le réseau qui amènerait par les communications engendrées, à une perte d'énergie. Pour cette raison, TinySEC met en place des mécanismes d'authentification basés sur le code MAC, de chiffrement des informations et une protection contre les redondances d'informations.

Pour permettre une plus grande liberté d'actions, TinySEC supporte deux options de sécurité différentes :

- TinySEC-Auth: la sécurité concerne seulement l'authentification des données. Les données ne sont pas chiffrées, contrairement au code MAC qui est calculé à partir de l'entête du paquet pour assurer l'authenticité de l'expéditeur.
- TinySEC-AE: la sécurité porte à la fois sur l'authentification et l'encryptage des données. Les données sont chiffrées et envoyées avec un code MAC généré à partir des données chiffrées et de l'entête du paquet.

Pour l'authentification et l'encryptage des données TinySEC utilise un chiffrement par blocs de type CBC-MAC. De la même manière que pour SNEP, TinySEC utilise un vecteur initial pour le chiffrement du premier bloc et des chaînes de bits aléatoires ajoutés au message pour empêcher un attaquant d'analyser le trafic par comparaison des paquets. Cependant TinySEC n'utilise pas de compteur pour chaque chiffrement, ce qui empêche de garantir la fraîcheur des données et laisse possible les attaques de type rejeu de paquets. Il est aussi à noter que TinySEC n'est pas adapté à la deuxième version de TinyOs (TinyOS 2.x).

II.5 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.

Chapitre III

Cryptographie basée sur les courbes elliptiques

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. Le chiffrement des données 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.

Les tailles de clés ne sont pas un problème pour les ordinateurs modernes, ni même pour les Smartphones puissants, même si pour ces derniers, toute économie de consommation d'énergie est un avantage. En revanche, de nouveaux besoins cryptographiques apparaissent constamment dans des systèmes atypiques, où les ressources sont souvent contraintes comme le cas des RdCs dispersés dans un terrain d'intérêt.

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 cas, 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 qui présentent certaines contraintes en termes d'énergie et de puissance de calcul.

Le principal système concurrent au système RSA qui est réputé sûr est fondé sur des objets mathématiques, les courbes elliptiques. La cryptographie basée sur les courbes elliptiques permet d'assurer le même niveau de sécurité que RSA tout en utilisant une clé de taille extrêmement inférieure que celle utilisée par RSA.

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

III.2 Préliminaires

Dans cette section, nous présentons quelques notions mathématiques qui s'avèrent utiles pour comprendre le mécanisme de sécurité basé sur les courbes elliptiques.

III.2.1 Définition d'un 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.

III.2.2 Définition d'un 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 + :

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

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

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

III.2.3 Définition d'un sous-anneau

Soit (A,+, *) un anneau. Une partie non vide A_1 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, (A_1,+, *) 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 A_1 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.

III.2.4 Définition d'un 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 K_1 de K tel que pour tout élément non nul x de K_1, on a $x^{-1} \in K_1$; (K_1,+, *) est alors un corps.

III.2.5 Equation de Weierstrass

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

$$E: y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \qquad avec\ a_i \in K$$

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

$$\begin{cases} a_1y = 3x^2 + 2a_2x + a_4 \\ 2y + a_1x + a_4 = 0 \end{cases}$$

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

$$f(x,y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6$$

III.3 Les courbes elliptiques

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.

$$E = \{ (x,y) \in \overline{K}^2 \mid y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6 \} \cup \{O\}$$

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 $b_2 = a_1^2 + 4a_2$ b_2, nous obtenons:

$$E: y^2 = x^3 - 27c_4x - 54c_6$$

$$avec\ \ b_4 = 2a_4 + a_1a_3$$

$$b_6 = a_3^2 + 4a_6$$

$$c_4 = b_2^2 - 24b_4$$

$$c_6 = -b_2^3 + 36b_2b_4 - 216b_6$$

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.3.1 Exemples de courbes elliptiques

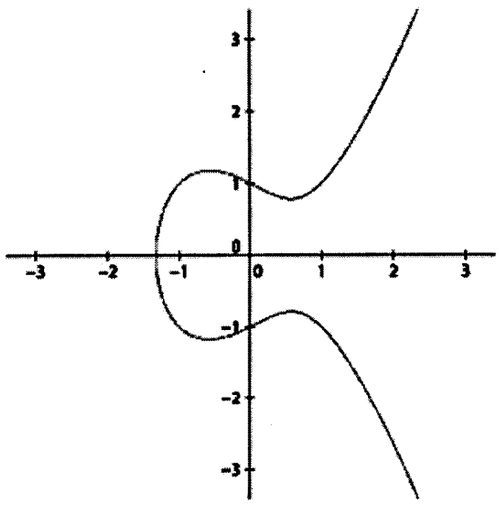


Figure 0-1: Représentation de la courbe d'équation $y^2 = x^3 + x - 1$

Représentation de la courbe elliptique d'équation $y^2 = x^3 + 2x -$

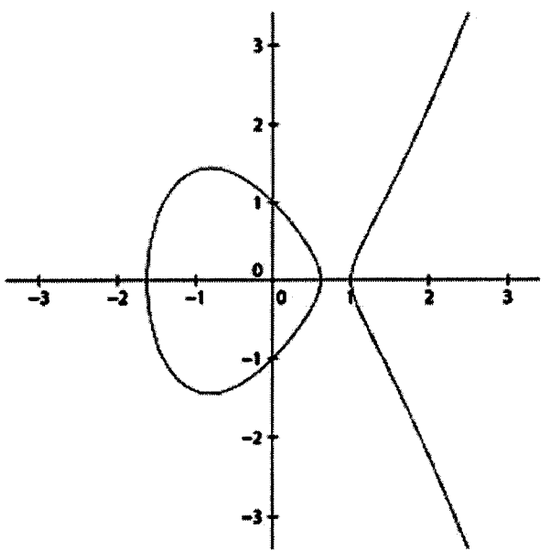


Figure 0-2: Représentation de la courbe elliptique d'équation $y^2 = x^3 + 2x - 1$

III.3.2 Addition et doublement des points de la courbe

Pour faire l'addition des points sur la courbe elliptique en utilisant la méthode des tangentes et des sécantes.

Soient P et Q deux points de la courbe elliptique, différents du point à l'infini O. L'addition de ces deux points est basée sur la droite tracée (PQ) qui les joint. D'où, on fait relever quatre cas de figures:

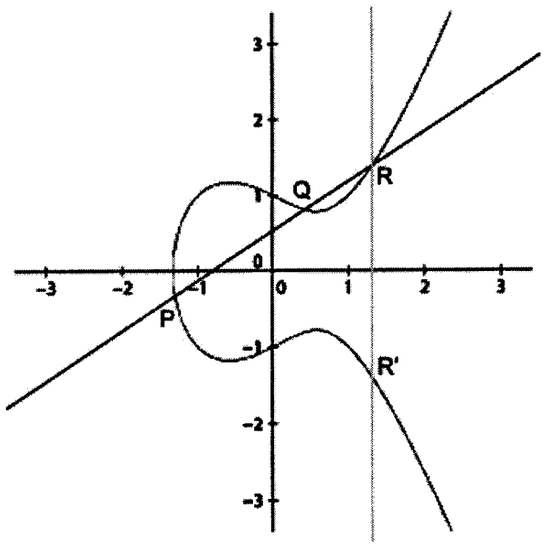


Figure 0-3: Représentation du 1er cas

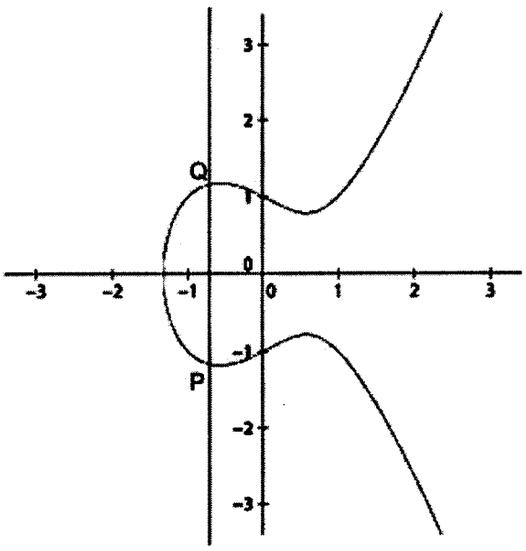


Figure 0-4: Représentation du 2^{ième} cas

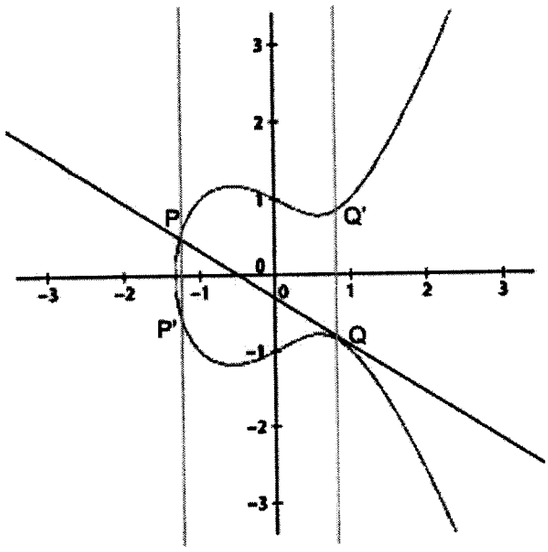


Figure 0-5: Représentation du 3ième cas

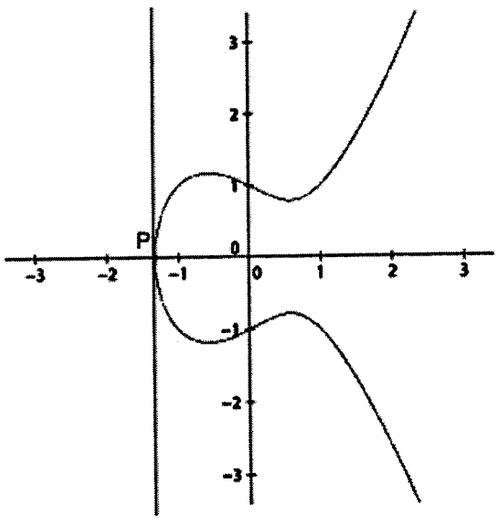


Figure 0-6: Représentation du 4ième cas

- La droite (PQ) coupe la courbe en un troisième point R. Dans ce cas, la somme des deux points sera le point R' qui est l'image symétrique du point R par rapport à l'axe des abscisses comme montre la figure III.3.
- La droite (PQ) est verticale: donc c'est une droite qui est 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.

- 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. Si la tangente est en Q alors R = Q, sinon 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 (voir 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 par rapport à l'axe des abscisses comme montre la figure III.6.

Nous présentons d'une manière explicite la somme de deux points d'une courbe elliptique.

Soient E : $y^2 = x^3 + ax + b$ une courbe elliptique et $P_1(x_1,y_1)$ et $P_2(x_2,y_2)$ deux points de la courbe (E), avec $P_1, P_2 \neq O$. On a $P_1 + P_2 = P_3(x_3,y_3)$:

1- Si $x_1 \neq x_2$, alors

$$x_3 = m^2 - (x_1 + x_2), \quad y_3 = m(x_1 - x_3) - y_1, \quad \text{où } m = \frac{(y_2 - y_1)}{(x_2 - x_1)}$$

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

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

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.4 Cryptographie basée sur les courbes elliptiques (ECC)

Les courbes elliptiques ont une propriété essentielle à leur utilisation en cryptographie, l'addition géométrique des points de la courbe.

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 $2*\text{Log}_2$ 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 $P + \infty = P$. 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..

III.4.1 Construction de ECC

On peut concevoir un système cryptographique à clé publique appelé cryptographie à base des courbes elliptiques (ECC), où le problème de la factorisation d'entiers utilisé dans RSA est remplacé par le problème du logarithme discret elliptique. Les paramètres publics du système sont la courbe elliptique et un point de départ P. Un individu qui souhaite utiliser ce système se crée une clé secrète en choisissant un grand entier n aléatoire, puis il calcule le point nP, sa clé publique, qu'il peut transmettre aux autres pour recevoir des messages chiffrés. Plusieurs algorithmes cryptographiques ont été élaborés à partir de la difficulté du logarithme discret elliptique, et notamment l'algorithme de Diffie-Hellman, qui permet à deux personnes qui ne sont pas rencontrées avant de se fabriquer un secret commun en communiquant uniquement sur un canal ouvert.

III.4.2 Apport de 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.

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.

III.5 Les courbes elliptiques sur les corps finis F_p

III.5.1 Définition d'une courbe elliptique sur un corps

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 $4a^3 + 27b^2$ (mod 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. En outre, pour utiliser E(F_P) dans la cryptographie, l'équation de la courbe elliptique doit être satisfaite pour des valeurs arbitraire a, b et p.

En cryptographie, on s'intéresse surtout aux courbes elliptiques sur des corps finis. En particulier, il est crucial de savoir calculer la clôture algébrique #E(F_p) pour E une courbe elliptique définie sur F_p. Le théorème de Hasse permet de déterminer le nombre de points qui appartiennent à la courbe elliptique E et dont les coordonnées sont engendrées par le corps fini F_p. Dans notre contribution, nous avons déterminer ce nombre par une méthode algorithmique.

Théorème de Hasse :

Soit E une courbe elliptique définie sur un corps fini F_p, alors

$$\left| p+1 - \neq E(F_p) \right| \leq 2\sqrt{p}$$

III.5.2 Nécessité d'un corps fini

Avec des nombres rationnels, le problème du logarithme discret elliptique peut être résolu. En effet, avec un point P de départ dont les coordonnées sont formées de x chiffres, la multiplication par n de ce point conduit à un point dont les coordonnées sont formées de $(x*n^2)$ chiffres décimaux. En Outre, lorsque n est grand, la manipulation de telles coordonnées devient très délicate et la taille de nP trahit le nombre n que l'on souhaite dissimuler. D'où, il apparaît la nécessité d'un corps fini pour résoudre le problème du logarithme discret. Ce corps fini est constitué d'un nombre fini d'éléments qui représente le nombre de symboles différents (alphabet, chiffres, caractères spéciaux) dans un message. L'ensemble des opérations (additions et multiplications) effectuées sur les points d'une courbe elliptique sont basées sur les opérations *modulo p,* p étant un nombre premier qui définit le corps fini dans lequel on travaille.

III.6 Conclusion

Dans ce chapitre, nous avons présenté les courbes elliptiques et nous avons tiré profit de leurs avantages pour concevoir un crypto-système. Puis, nous avons défini ces courbes elliptiques dans un corps fini pour que le crypto-système conçu soit efficace et son logarithme discret soit difficile à résoudre pour décourager tout attaquant.

L'approche utilisée pour la génération des clés et l'algorithme de chiffrement et déchiffrement font l'objet du chapitre suivant.

Chapitre IV

Implémentation d'un crypto-système basé sur les courbes elliptiques

IV.1 Introduction

Les courbes elliptiques offrent le même niveau de sécurité que d'autres systèmes cryptographiques à l'instar de RSA tout en utilisant des clés de taille nettement inférieure. Cette caractéristique permet aux courbes elliptiques d'être commode pour les systèmes qui des ressources limitées (mémoire, CPU, ...) telles que les cartes à puce et les réseaux de capteurs.

Dans ce chapitre, nous proposons un crypto-système basé sur les courbes elliptiques. Ce crypto-système s'adapte aux réseaux de capteurs et il permet une cryptographie légère tout en offrant plus de performances que d'autres crypto-systèmes (RSA, AES, ...).

IV.2 Implémentation d'un crypto-système basé sur les courbes elliptiques

Nous présentons dans cette section la démarche à suivre pour implémenter un crypto-système en utilisant les courbes elliptiques. Ce crypto-système est appelé ECC.

ECC utilise un groupe fini composé de points (x, y) se trouvant sur une courbe elliptique dont la procédure de chiffrement et de déchiffrement est basée sur l'addition et la multiplication (addition successive du même point) des points qui font partis à la courbe et dont les coordonnées sont inférieures à la caractéristique du corps.

La réalisation de ce crypto-système s'effectue en trois étapes:

- génération des clés,
- chiffrement,
- déchiffrement

Nous présentons par la suite un procédé de génération de clés basé sur l'approche de Diffie-Hellman et un mécanisme de chiffrement basé sur l'algorithme d'El-Gamal et les ECC.

IV.2.1 Génération des clés

La génération des clés avec ECC nécessite de déterminer la courbe elliptique sur un corps fini $E(F_P)$ selon l'équation suivante:

$$y^2 \bmod p = x^3 + ax + b \bmod p$$

Soit une courbe elliptique E dans le corps fini E(F_P), et soit P le point générateur de E(F_P). La courbe E permet alors de générer n sous-groupes cycliques correspondant à des points de la courbe:

$$P = \infty, P, 2P, ..., (n-1)P$$

Exemple :

Soient le corps fini F_7 et l'équation suivante:

$$y^2 \pmod 7 = x^3 + x + 4 \pmod 7$$

La courbe correspondante à cette équation contient 10 points selon le critère de Hasse :

$$\left| q + 1 - \neq E(F_q) \right| \leq 2\sqrt{q}$$

Ces points vérifient bien l'équation de la courbe. La détermination du nombre de points qui vérifient cette équation et dont les coordonnées sont inférieures à 7 permet de déterminer la clé privée et la clé publique.

Tableau 4: Les points de la courbe dans le corps F_7

N° point	1	2	3	4	5	6	7	8	9	10
Coordonnées	(0,2)	(2,0)	(4,3)	(5,1)	(6,3)	(0,5)	(4,4)	(5,6)	(6,4)	∞

Comment se fait le calcul de ces deux clés ?

Clé privée:

La clé privée est un entier k secret choisi aléatoirement entre 1 et (n-1). Dans l'exemple précédent k prend sa valeur entre 1 et 9.

Clé publique:

La clé publique Q correspond à la multiplication d'un point aléatoire choisi P de la courbe elliptique par la clé privée k soit :

$$Q = kP$$

Pour généraliser cette démarche, nous proposons un mécanisme de sécurité basé sur l'algorithme de Diffie-Hellman et les courbes elliptiques. 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 :

$$K_A = n_A Q_B$$
$$K_B = n_B Q_A$$

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 Q_A, Q_B, 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 IV.1 résume la démarche de génération de clés par le procédé de Diffie-Hellman.

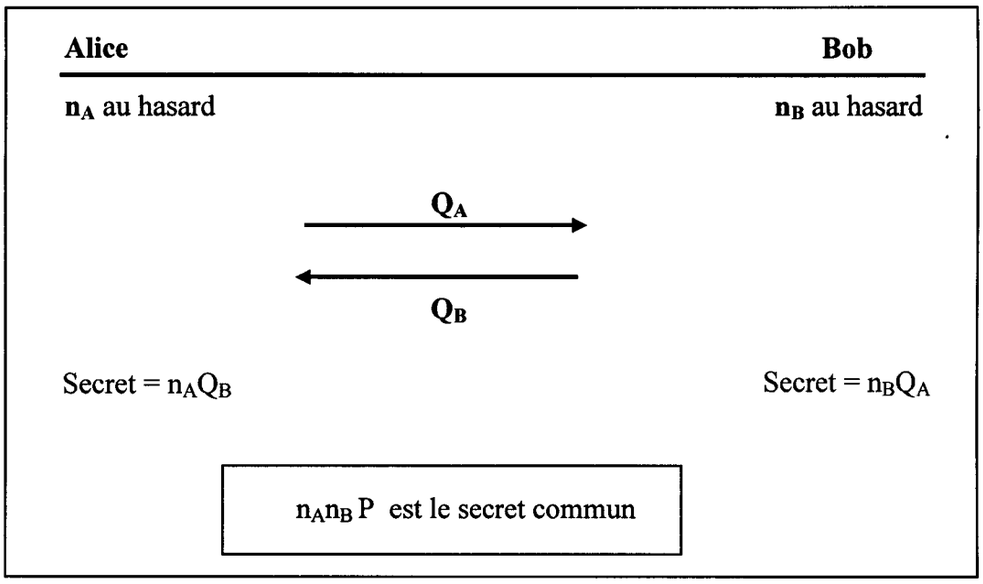


Figure 0-1: Processus de génération de clés

IV.2.2 Chiffrement

Imaginons le scénario où Alice souhaite communiquer un message à Bob. La procédure de chiffrement se fait par la conversion de ce message m en plusieurs points m_i et qui correspond chacun d'eux à un point M_i dans le corps fini F_P. Ensuite, le chiffrement s'effectue en additionnant ce point m_i à un entier aléatoire n_A 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 E_2 par multiplication de l'entier n_A 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

IV.2.3 Déchiffrement

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

Le point M correspond à :

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

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

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

Soit :

$$m_i = E_1 - n_B E_2$$

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

IV.3 Contexte d'exécution

Pour expliciter notre contribution, nous proposons de la dérouler sur un exemple.

Données

Soit la courbe elliptique E d'équation:

$$y^2 \pmod{29} = x^3 + 11x + 17 \pmod{29}$$

Le tableau IV.1 illustre l'ensemble des points de la courbe elliptique E dans le corps fini F_{29}.

Tableau 5: Points de la courbe dans un corps fini F_{29}

N° du point	**1 (a)**	**2 (b)**	**3 (c)**	**4 (d)**	**5 (e)**	**6 (f)**	**7 (g)**	**8 (h)**	**9 (i)**
Coordonnées	(9,2)	(5,20)	(28,11)	(25,24)	(20,1)	(6,3)	(27,4)	(17,10)	(4,3)
N° du point	**10 (j)**	**11 (k)**	**12 (l)**	**13 (m)**	**14 (n)**	**15 (o)**	**16 (p)**	**17 (q)**	**18 (r)**
Coordonnées	(23,24)	(19,3)	(10,24)	(1,0)	(10,5)	(19,26)	(23,5)	(4,26)	(17,19)
N° du point	**19 (s)**	**20 (t)**	**21 (u)**	**22 (v)**	**23 (w)**	**24 (x)**	**25 (y)**	**26 (z)**	
Coordonnées	(27,25)	(6,26)	(20,28)	(25,5)	(28,18)	(5,9)	(9,27)	∞	

Dans cet exemple, les points sont codés en fonction de leurs coordonnées selon le schéma algorithmique suivant :

- Soit le point A de coordonnées (x,y) tels que $A \in E$ et $x, y \in F_{29}$
  - Convertir x et y en binaire
  - Concaténer x et y $(xy)_2$
  - $(xy)_2$ représente le code du point A

On considère que le point 22 a pour coordonnées (25,5) et sera codé comme suit:

(25 , 5)

↓ ↓

11001 00101

↓

1100100101

Le tableau IV.3 suivant récapitule le codage des points de la courbe elliptique E.

Tableau 6: Codage des points de la courbe

Lettre	N° du point	Code
a	1	0100100010
b	2	0010110100
c	3	1110001011
d	4	1100111000
e	5	1010000001
f	6	0011000011
g	7	1101100100
h	8	1000101010
i	9	0010000011
j	10	1011111000
k	11	1001101101
l	12	0101011000
m	13	0000100000
n	14	0101000101
o	15	1001111010
p	16	1011100101
q	17	0010011010
r	18	1000110011
s	19	1101111001
t	20	0011011010
u	21	1010011100
v	22	1101100101
w	23	1110010010
x	24	0010101001
y	25	0100111011
z	26	1110111101

Chiffrement

Avant de présenter le procédé de chiffrement, nous rappelons comment se fait le calcul de l'addition de deux points et en particulier l'ajout d'un point à lui-même.

Soit le point P qui a pour coordonnées (x_P, y_P), celles du point (P+P) noté 2P s'expriment ainsi:

$$x_{2P} = s^2 - 2x_P$$
$$y_{2P} = -y_P + s(x_P - x_{2P})$$

avec

$$s = \frac{(3x_P^2 + a)}{2y_P}$$

Addition de points dans un corps Fp

Soient les point P (x_P , y_P) et Q (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,

$$x_{P+Q} = x_R = s^2 - x_P - x_Q \pmod{p}$$
$$y_{P+Q} = y_R = s(x_P - x_R) - y_P \pmod{p}$$

Doublement d'un point dans un corps Fp

Soit le point P (x_P,y_P), on cherche à déterminer R = 2P = (x_{2P} , y_{2P}). Soit $s = \frac{(3x_P^2 + a)}{2y_P}$

alors

$$x_{2P} = x_R = s^2 - 2x_P \pmod{p}$$
$$y_{2P} = y_R = -y_P + s(x_P - x_R) \pmod{p}$$

Soustraction d'un point dans un corps Fp

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

$$x_R = x_P$$
$$y_R = -y_P \pmod{p}$$

La construction requérant les quatre opérations, on a donc besoin d'un corps fini, c'est-à-dire d'un ensemble d'éléments sur lesquels s'appliquent ces opérations. Puisque l'alphabet est formé de 26 lettres alors nous choisissons le premier plus grand nombre premier qui est supérieur à 26 et qui permet d'engendrer tout l'alphabet. Soit 29 ce nombre. Donc, tous les calculs s'effectuent modulo 29.

Dans notre contribution, nous supposons que le chiffrement peut concerner plusieurs caractères à la fois et dans ce cas, nous réalisons l'addition des points représentant ces caractères avant de l'ajouter au point $n_A Q_B$.

Pour bien éclaircir notre contribution, nous traitons le cas particulier là où le chiffrement se fait caractère par caractère.

On suppose qu'Alice veut envoyer un message m= "bonsoir" à Bob, en se référent au tableau IV.3, m se transforment en points d'ordre suivant:
2, 15, 14, 19, 15, 9 et 18 dont les coordonnées sur la courbe sont respectivement (5,20), (19,26), (10,5), (27,25), (19,26), (4,3) et (17,19).

Maintenant Alice doit calculer E_1 et E_2 connaissant la clé publique de Bob qui est:

$$Q_B = n_B P$$

On suppose que le point générateur P est le point numéro 10 qui a pour coordonnées (23,24), et que la clé secrète de Bob $n_B = 3$. Donc:

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

Supposons qu'Alice a choisi *(n_A=2)*. Donc, il calcule E_1 pour le premier point de coordonnées (5,20) de la manière suivante:

$$\begin{aligned} E_1 &= M + n_A Q_B \\ &= (5,20) + (17,10) \\ &= (23,24) \end{aligned}$$

Puis Alice calcule E_2:

$$\begin{aligned} E_2 &= n_A P \\ &= (23,24) + (23,24) = (6,26) \end{aligned}$$

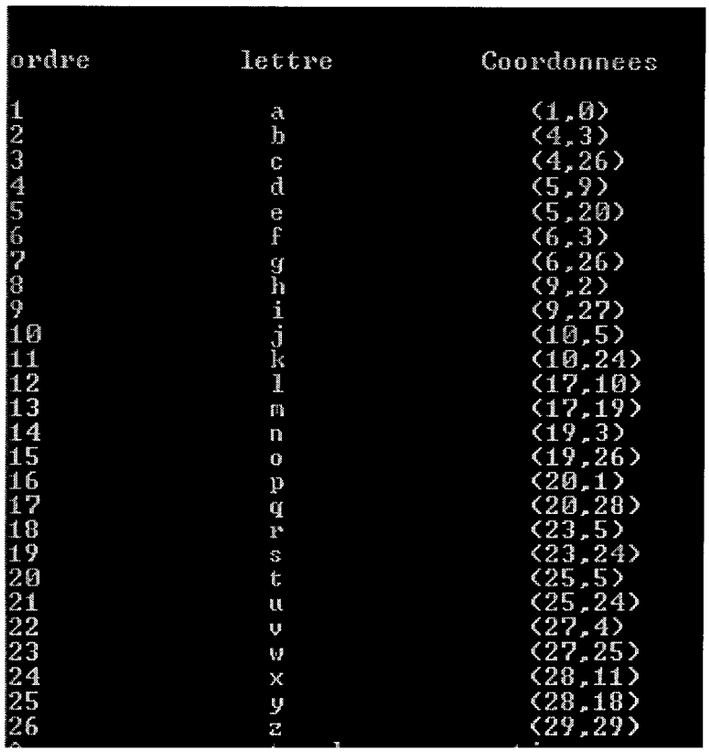
Alice envoie donc à Bob le couple (E_1, E_2) dans cet ordre.

IV.4 Application

Nous avons utilisé la courbe elliptique dont l'équation est:

$$y^2 \pmod{29} = x^3 + 11x + 17 \pmod{29}$$

Au début, on génère l'ensemble des points de la courbes qui appartiennent au corps $E(F_{29})$. La figure IV.2 illustre l'ensemble de ces points:



ordre	lettre	Coordonnees
1	a	(1,0)
2	b	(4,3)
3	c	(4,26)
4	d	(5,9)
5	e	(5,20)
6	f	(6,3)
7	g	(6,26)
8	h	(9,2)
9	i	(9,27)
10	j	(10,5)
11	k	(10,24)
12	l	(17,10)
13	m	(17,19)
14	n	(19,3)
15	o	(19,26)
16	p	(20,1)
17	q	(20,28)
18	r	(23,5)
19	s	(23,24)
20	t	(25,5)
21	u	(25,24)
22	v	(27,4)
23	w	(27,25)
24	x	(28,11)
25	y	(28,18)
26	z	(29,29)

Figure 0-2: Points de la courbe

Après la détermination des points de la courbe qui appartiennent au corps fini F29, on code ces points en utilisant seulement cinq bits puisque on représente un ensemble de points dont le cardinal est égal à 26 (lettres de l'alphabet). La figure IV-3 montre l'ordre des points avec leurs coordonnées et leurs codes.

num	Lettre	Coordonnees	Codage
1	a	(1, 0)	0 0 0 0 1 0 0 0 0 0
2	b	(4, 3)	0 0 1 0 0 0 0 0 1 1
3	c	(4, 26)	0 0 1 0 0 1 1 0 1 0
4	d	(5, 9)	0 0 1 0 1 0 1 0 0 1
5	e	(5, 20)	0 0 1 0 1 1 0 1 0 0
6	f	(6, 3)	0 0 1 1 0 0 0 0 1 1
7	g	(6, 26)	0 0 1 1 0 1 1 0 1 0
8	h	(9, 2)	0 1 0 0 1 0 0 0 1 0
9	i	(9, 27)	0 1 0 0 1 1 1 0 1 1
10	j	(10, 5)	0 1 0 1 0 0 0 1 0 1
11	k	(10, 24)	0 1 0 1 0 1 1 0 0 0
12	l	(17, 10)	1 0 0 0 1 0 1 0 1 0
13	m	(17, 19)	1 0 0 0 1 1 0 0 1 1
14	n	(19, 3)	1 0 0 1 1 0 0 0 1 1
15	o	(19, 26)	1 0 0 1 1 1 1 0 1 0
16	p	(20, 1)	1 0 1 0 0 0 0 0 0 1
17	q	(20, 28)	1 0 1 0 0 1 1 1 0 0
18	r	(23, 5)	1 0 1 1 1 0 0 1 0 1
19	s	(23, 24)	1 0 1 1 1 1 1 0 0 0
20	t	(25, 5)	1 1 0 0 1 0 0 1 0 1
21	u	(25, 24)	1 1 0 0 1 1 1 0 0 0
22	v	(27, 4)	1 1 0 1 1 0 0 1 0 0
23	w	(27, 25)	1 1 0 1 1 1 1 0 0 1
24	x	(28, 11)	1 1 1 0 0 0 1 0 1 1
25	y	(28, 18)	1 1 1 0 0 1 0 0 1 0
26	z	(29, 29)	1 1 1 0 1 1 1 1 0 1

Figure 0-3: Codage des points de la courbe

Puis Alice et Bob créent chacun d'eux sa propre clé privée représentée respectivement par n_A et n_B. Ensuite, ils calculent respectivement les points ($Q_A = n_A P$) et ($Q_B = n_B P$) qu'ils les échangent entre eux. Ces points seront utilisés pour chiffrer un message comme montre la figure IV-4.

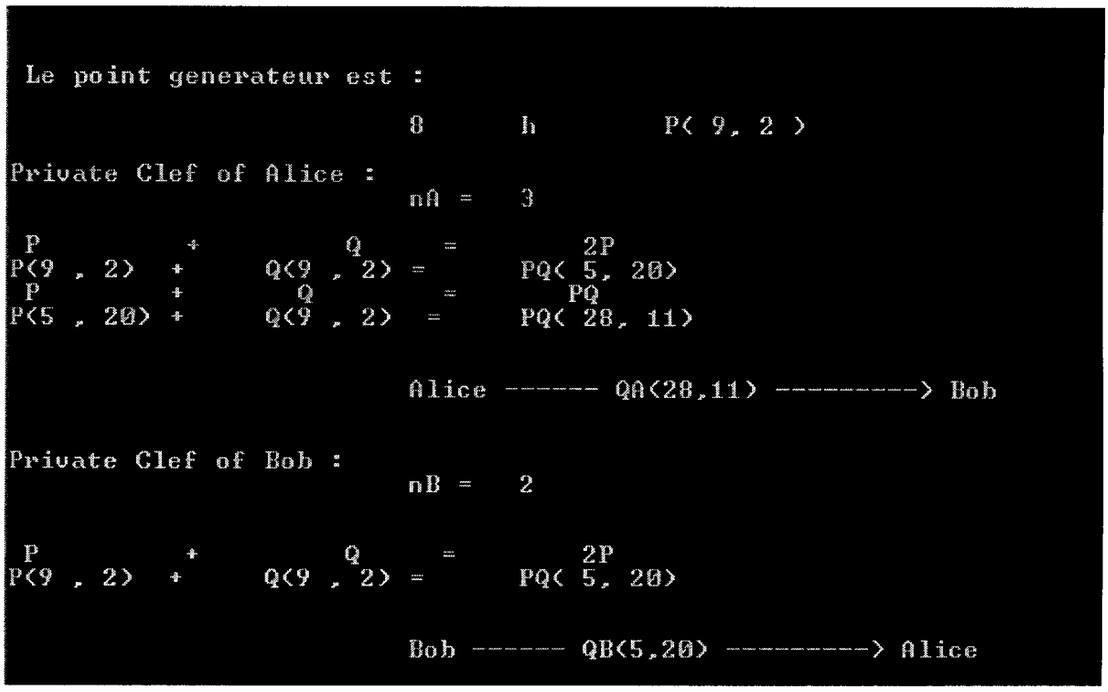


Figure 0-4: Génération de clés

La figure IV-5 montre comment se fait le chiffrement d'un message ? Après l'échange des points Q_A et Q_B entre Alice et Bob. Les deux peuvent facilement chiffrer un message qui sera représenté par un point de la courbe. Par exemple, si Alice veut communiquer un message à Bob, il calculera le produit $n_A * Q_B$ qu'il l'ajoute au message qu'il veut envoyer. Le résultat de cette opération (message chiffré) représente un point de la courbe.

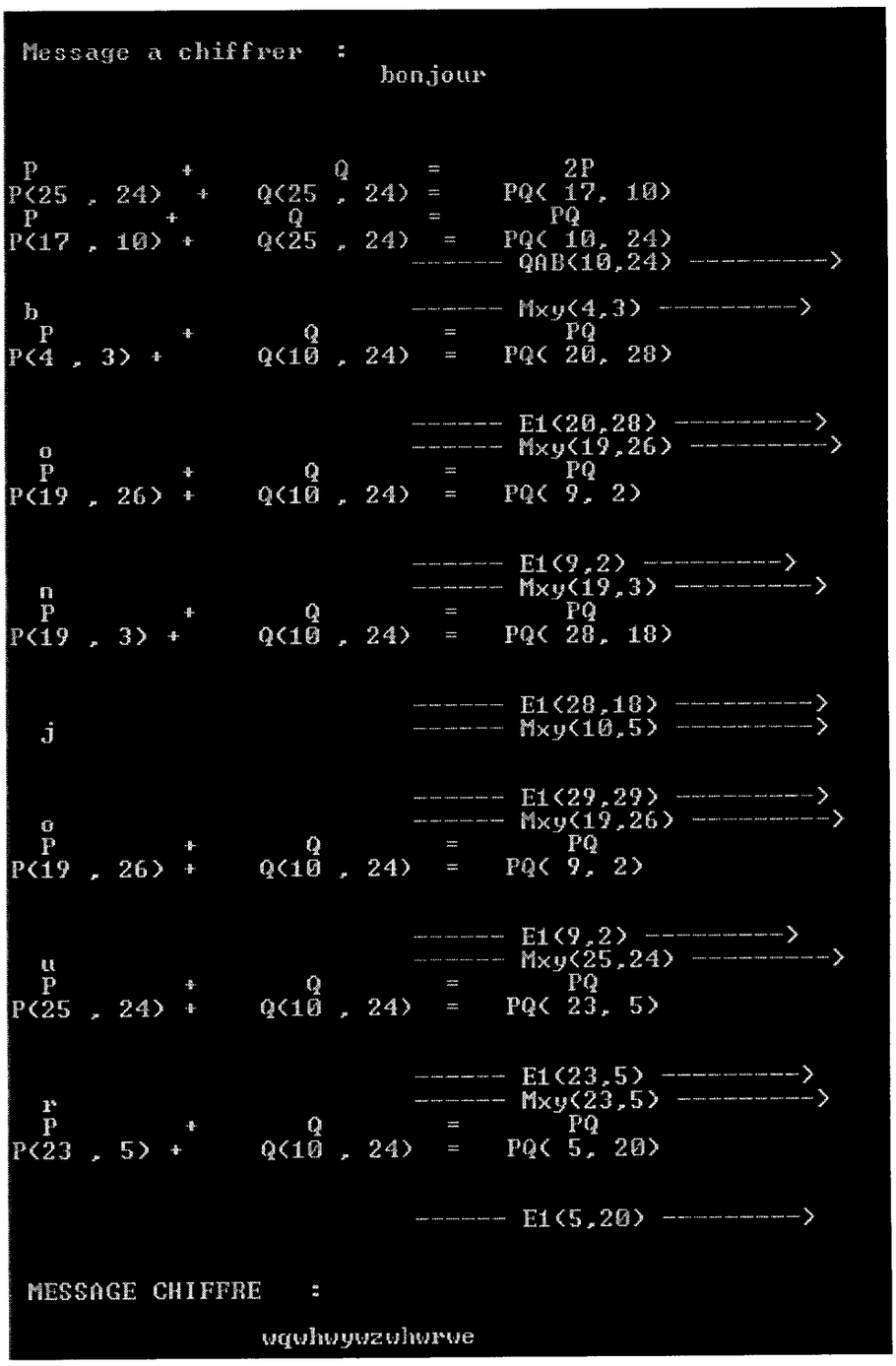


Figure 0-5: Chiffrement d'un message

IV.5 Conclusion

Le crypto-système que nous avons proposé dans ce chapitre, garantit un niveau de sécurité très élevé tout en utilisant une clé de taille moyenne. Avec une clé de taille 300 bits , il est très difficile voire impossible de résoudre le logarithme discret elliptique. Le crypto-système proposé utilise une cryptographie légère, ce qui le rend adaptable aux systèmes à ressources limitées à l'instar des réseaux de capteurs.

Conclusion

Conclusion

Dans ce mémoire, nous avons traité le problème de la sécurité dans les réseaux de capteurs sans fil.

Nous avons pu voir que la principale problématique de l'application de méthodes de sécurité dans les RdCs réside dans le souci de préserver l'énergie des capteurs. En l'occurrence, il apparaît que la plupart des solutions actuelles ne permet 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, nous avons 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, nous avons utilisé l'approche de Diffie-Hellman pour la génération de clés et l'algorithme d'El-GAMAL pour le chiffrement et le déchiffrement. En outre, dans ce mécanisme la 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.

En perspectives, nous proposons de mettre en place cette solution de sécurité sur un réseau de capteurs réel.

Bibliographie

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

[2] A. Dunkels, B. Gronvall, and T. Voigt."Contiki - a lightweight and flexible operating system for tiny networked sensors". In proceedings of the 29th IEEE International Conference on Local Computer Networks, pp.455-462, Washington, USA, 2004.

[3] 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.

[4] FreeRTOS. In http ://www.freertos.org/

[5] S. Bhatti, J. Carlson, H. Dai, J. Deng, J. Rose, A. Sheth, B. Shucker, C. Gruènwald, A. Torgerson, and R. Han. "Mantis os : an embedded multithreaded operating system for wireless micro sensor platforms". Mobile Network Applications, 10(4):563-579, 2005.

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

[7] J. Zhang, W. Li, Z. Yin, S. Liu, and X. Guo. "Forest fire detection system based on wireless sensor network". In proceedings of the 4th IEEE Conference on Industrial Electronics and Applications (ICIEA '09), pp. 520-523, 2009.

[8] Projet Hydro Sensor Flow. http://www.hydro-sensor-flow.com/

[9] D. Malan, T. Fulford-Jones, M. Welsh, and S. Moulton. "Codeblue : An ad hoc sensor network infrastructure for emergency medical care". In International Workshop on Wearable and Implantable Body Sensor Networks, april 2004.

[10] P. Volgyesi, G. Balogh, A. Nadas, C.B. Nash, and A. Ledeczi. "Shooter localization and weapon classification with soldier-wearable networked sensors". In Proceedings of the 5th international conference on Mobile systems, applications and services (MobiSys '07) , pp.113-126, New York, USA, 2007.

[11] E.I. Gaura, J. Brusey, J. Kemp, and C. Douglas Thake. "Increasing safety of bomb disposal missions : a body sensor network approach". Trans. Sys. Man Cyber Part C, 39(6) :pp.621-636, 2009.

[12] Jam C. L. Knapp D. A. Koenig Z. M. Luke S. J. Pohl B. A. Schach von Wittenau A. Wolford J. K. Gosnell T. B., Hall J. M. "Gamma-ray identification of nuclear weapon materials". Technical report, Lawrence Livermore National Lab., USA, 2007.

[13] M.J. Brown. "Users guide developed for the jbrews project ". Technical Report LA-UR-. 99-4676. Los Alamos National Laboratory of California University, 1999.

[14] S. Kim, S. Pakzad, D. E. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. "Wireless sensor networks for structural health monitoring". In Andrew T. Campbell, Philippe Bonnet, and John S. Heidemann (ACM), editors, SenSys, pp. 427-428. 2006.

[15] J. N. Al-Karaki and A. E. Kamal. Routing techniques in wireless sensor networks : a survey. In IEEE Wireless Comm., vol. 11, pp.6-28, 2004.

[16] Crossbow technology inc. mpr/mib user's manual. http://www.xbow.com/Support/Support_pdf_files/MPRMIB/Series_Users_Manual.pdf, 2010.

[17] C. Ozturk and Y. Zhang. "Source-location privacy in energy-constrained sensor network routing". In ACM SASN, pp.88-93, 2004.

[18] A.D. Wood and J.A. Stankovic. "Denial of services in sensor networks". IEEE Computer, October 2002.

[19] C. Karlof and D. Wagner. "Secure routing in wireless sensor networks : attacks and countermeasures". Ad Hoc Networks, vol.1, no 2, pp.293-315, 2003.

[20] C. Hartung, J. Balasalle, and R. Han. "Node compromise in sensor networks : The need for secure systems". Technical Report CU-CS-988- 04, Department of Computer Science, University of Colorado at Boulder, 2004.

[21] Y.C. Hu, A. Perrig, and D. B. Johnson. "Wormhole attacks in wireless networks". IEEE Journal on Selected Areas in Communications, vol. 24 no.2, pp.370-380, 2006.

[22] 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, pp.259-268, 2004.

[23] D. Halperin, T. S. Heydt-Benjamin, B. Ransford, S. S. Clark, B. Defend, W. Morgan, K. Fu, T. Kohno, and W. H. Maisel. "Pacemakers and implantable cardiac defibrillators : Software radio attacks and zero-power defenses". In Proceedings of the IEEE Symposium on Security and Privacy, pp.129-142, Washington, USA, 2008.

[24] J. Deng, R. Han, and S. Mishra. "Countermeasures against traffic analysis attacks in wireless sensor networks ". In Proceedings of the First International Conference on Security and Privacy for Emerging Areas in Communications Networks, pp.113-126, Washington, USA, 2005.

[25] C. Bekara and M. L. 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, pp.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] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel. "A survey of lightweight-cryptography implementations ". IEEE Des. Test, 24(6) :522-533, 2007.

[29] A. Dunkels, B. Gronvall, and T. Voigt. "Contiki - a lightweight and flexible operating system for tiny networked sensors". In Proceedings of the 29th IEEE Int. Conference on Local Computer Networks, pp.455-462,Washington, USA, 2004.

[30] Y. W. Law, J. Doumen, and P. Hartel. "Survey and benchmark of block ciphers for wireless sensor networks". ACM Trans. Sen. Netw., vol. 2 no.1, pp.65-93, 2006.

[31] K. Piotrowski, P. Langendoerfer, and S. Peter. "How public key cryptography influences wireless sensor node lifetime". In Proceedings of the 4th ACM workshop on Security of ad hoc and sensor networks, pp.169-176, New York, USA, 2006.

[32] H. Wang and Q. "Li. Efficient implementation of public key cryptosystems on mote sensors". In International Conference on Information and Communication Security, LNCS 4307, pp.519-528, 2006.

[33] A. Liu and P. Ning. "Tinyecc : A con_gurable library for elliptic curve ryptography in wireless sensor networks". In Proceedings of the 7th international conference on Information processing in sensor networks, pp.245-256, Washington, USA, 2008.

[34] A. Perrig, R. Szewczyk, J. D. Tygar, V. Wen, and D. E. Culler. Spins: Security protocols for sensor networks. Wireless Networks, vol. 8, no.5, pp.521-534, 2002.

[35] D. Liu and P. Ning. "Multilevel μtesla : Broadcast authentication for distributed sensor networks". Trans. on Embedded Computing Sys., vol. 3 no.4, pp.800-836, 2004.