République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université Mouloud MAMMERI de Tizi-Ouzou

Faculté de Génie Electrique et d'Informatique

Département d'Informatique

Mémoire

En vue de l'obtention du diplôme de Master en informatique

Option :
Réseaux, mobilité et systèmes embarqués

Thème :

Optimisation des calculs cryptographiques dans un réseau de capteurs sans fil

Membres de jury:

Président : Mr M.DAOUI

Examinatrice : Mlle N.SEGHIRI

Encadreur : Mr M.RAMDANI

Réalisé par :

Mlle AIT SI SLIMANE Hayat.

Mlle OUARGLI Fella.

Promotion : 2017/2018

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail.

Ensuite nous tenons à remercier profondément notre promoteur Mr ***M.RAMDANI*** *pour sa disponibilité, ses précieux conseils et orientations tout au long de notre travail.*

Nos vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions.

Nos remerciements vont aussi à tous les enseignants du département d'Informatique

Qu'il nous soit enfin permis de remercier nos chers parents pour leur amour et soutien et à tous ceux qui d'une manière ou d'une autre, ont contribué à la réussite de ce travail

Fella, Haya

A cœur vaillant rien d'impossible
A conscience tranquille tout est accessible

Quand il y a la soif d'apprendre
Tout vient à point à qui sait attendre

Quand il y a le souci de réaliser un dessein
Tout devient facile pour arriver à nos fins

Malgré les obstacles qui s'opposent
En dépit des difficultés qui s'interposent

Les études sont avant tout
Notre unique et seul atout

Ils représentent la lumière de notre existence
L'étoile brillante de notre réjouissance

Comme un vol de gerfauts hors du charnier natal
Nous partons ivres d'un rêve héroïque et brutal

Espérant des lendemains épiques
Un avenir glorieux et magique

Souhaitant que le fruit de nos efforts fournis
Jour et nuit, nous mènera vers le bonheur fleuri

Aujourd'hui, ici rassemblés auprès des jurys,
Nous prions dieu que cette soutenance
Fera signe de persévérance
Et que nous serions enchantés
Par notre travail honoré

Je dédie ce modeste travail à

Ma très chère mami OUGUENOUNE Houria ,
Affable, honorable, aimable .

Tu représentes pour moi le symbole de la bonté par excellence, la source de tendresse et l'exemple du dévouement qui n'a pas cessé de m'encourager et de prier pour moi.

Ta prière et ta bénédiction m'ont été d'un grand secours pour mener à bien mes études. Aucune dédicace ne saurait être assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner depuis ma naissance, durant mon enfance et même à l'âge adulte.

Tu as fait plus qu'une mère puisse faire pour que ses enfants suivent le bon chemin dans leur vie et leurs études.

Je te dédie ce travail en témoignage de mon profond amour. Puisse Dieu, le tout puissant, te préserver et t'accorder santé, longue vie et bonheur.

A mon cher papa
Rien au monde ne vaut les efforts fournis jour et nuit pour mon éducation et mon bien être. Ce travail est le fruit de tes sacrifices que tu as consentis pour mon éducation et ma formation.

A mes sœurs Lydia et Nadjia, mes frères Mohamed et Ammar
Je vous souhaite un avenir plein de joie, de bonheur, de réussite et de sérénité

A mon binôme, Hayat
En témoignage de l'amitié qui nous unie et des souvenirs de tous les moments que nous avons passé ensemble, je vous dédie ce travail et je vous souhaite une vie pleine de santé et de bonheur.

Fella

Dédicaces

A toi mon exemple dans la vie, ma raison de vivre, Papa, que je dédie ce modeste travail. Aucune dédicace ne serait assez éloquente pour exprimer ce que tu mérites pour tous les sacrifices que tu n'as cessé de me donner.

Très chère maman, audacieuse, patiente, honorable et aimable, tu présentes pour moi le symbole de la force, du courage et de bonté par excellence. Aujourd'hui, je te dis merci pour tout ce que tu m'as apporté
Depuis ma naissance et je te dédie ce modeste travail
En témoignage de mon profond amour

Je dédie ce travail :
A toi ma sœur jumelle Ouiza, mon épaule, ma force, merci pour le soutien que tu m'apporte.
A mes chers frères, merci pour tout ce que vous faites pour moi. Ma réussite est la vôtre.
A mes belles sœurs et mes adorables petits neveux Eliane, Messi et Aya.

Je ne peux trouver les mots justes et sincères pour vous exprimer mon affection, en témoignage de l'amitié qui nous uni et des souvenirs de tous les moments qu'on a passés ensemble. A vous : Liza et Amel que je dédie ce modeste travail.
A tous mes amis " Malha, Lynda, Noria, Fatiha, Badro et Aziz "
A une personne particulière, qui m'a toujours apporté un soutien moral.
Grand merci Walid.

A ma camarade, mon amie, ma binôme Fella, en témoignage de l'amitié qui nous uni je dédie ce modeste travail à toi ainsi qu'à toute ta charmante famille.
Je dédie ce travail
A toutes les personnes qui m'ont soutenu et aidé de près ou de loin.

Hayat

Résumé

L'une des techniques utilisées pour ouvrir la voie aux applications réparties dans des milieux impraticables est l'utilisation de réseaux mobiles particuliers connus sous le nom de réseaux de capteurs sans fil (RCSF). Un RCSF est constitué d'un ensemble de nœuds, appelés capteurs, communicants par des liaisons sans fil. Les réseaux de capteurs sans fil connaissent un intérêt important, tant dans la recherche ou dans l'industrie. Leurs domaines d'application sont nombreux : applications médicales, domaine militaire, surveillance de phénomènes environnementaux…etc.

Parmi les principales contraintes des RCsF est leurs limites en ressources, notamment en calcul et en énergie. Souvent implantés dans des zones hostiles, pour des applications sensibles et critiques, les RcSF présentent un intérêt particulier mais requièrent un niveau de sécurité élevé. Pour cela, nous allons étudier à travers ce sujet les différents protocoles de sécurité utilisés dans les RcSF et en particulier les protocoles cryptographiques basés sur la cryptographie des courbes elliptiques ainsi que les différentes optimisations réalisées sur les calculs, et essayer de proposer, par la suite, une solution d'optimisation et de réduction de calculs dans les échanges entre les noeuds d'un réseau de capteurs sans fil.

Mots clés : cryptographie des courbes elliptiques, multiplication scalaire par un point, réseaux de capteurs sans fil, sécurité.

Abstract

One of the techniques used to pave the way for distributed applications in impractical media is the use of particular mobile networks known as Wireless Sensor Networks (RCSF). A RCSF consists of a set of nodes, called sensors, communicating via wireless links. Wireless sensor networks are of great interest, both in research and in industry. Their fields of application are numerous: medical applications, military field, monitoring of environmental phenomena ... etc.

Among the main constraints of the RCsF is their limits in resources, especially in calculation and energy. Often implemented in hostile areas, for sensitive and critical applications, the RcSFs are of particular interest but require a high level of security. For this purpose, we will study through this topic the various security protocols used in the RcSFs and in particular the cryptographic protocols based on the cryptography of the elliptic curves as well as the different optimizations carried out on the calculations, and try to propose, thereafter, a solution for optimizing and reducing calculations in the exchanges between the nodes of a wireless sensor network.

Key words: cryptography of elliptic curves, point scalar multiplication, wireless sensor networks, security.

نبذة مختصرة

تتمثل إحدى التقنيات المستخدمة في تمهيد الطريق للتطبيقات الموزعة في الوسائط غير العملية في استخدام شبكات متنقلة معينة تُعرف باسم شبكات الاستشعار اللاسلكية .(RCSF) يتكون RCSF من مجموعة من العقد ، تسمى أجهزة الاستشعار ، تتصل عبر الروابط اللاسلكية .تحظى شبكات المحسات اللاسلكية باهتمام كبير ، سواء في الأبحاث أو في المجال الصناعي .مجالات تطبيقهم عديدة :التطبيقات الطبية ، المجال العسكري ، مراقبة الظواهر البيئية ... الخ.

من بين المعوقات الرئيسية للصندوق هو القيود المفروضة على الموارد ، لا سيما في الحساب والطاقة .غالباً ما يتم تطبيق RcSFsفي المناطق المعادية ، للتطبيقات الحساسة والحرجة ، وهي ذات أهمية خاصة ولكنها تتطلب درجة عالية من الأمن .لهذا الغرض ، سوف ندرس من خلال هذا الموضوع بروتوكولات الأمان المختلفة المستخدمة في RcSF وبشكل خاص بروتوكولات التشفير المبنية على تشفير المنحنيات الناقصية بالإضافة إلى التحسينات المختلفة التي أجريت على الحسابات ، ومحاولة اقتراح بعد ذلك الحل الأمثل لتحسين وتقليل العمليات الحسابية في التبادلات بين عقد شبكة مستشعر لاسلكي.

الكلمات الدالة : تشفير المنحنيات الناقصية ، مضاعفة العددية النقطية ، شبكات الاستشعار اللاسلكية ، الأمن.

Table des matières

Liste des figures

Liste des tableaux

Liste des algorithmes

Introduction générale

Un réseau de capteur sans fil est une technologie sans fil qui découle de l'évolution de la télécommunication et de la microélectronique. C'est un nouveau concept qui jouera un rôle de plus en plus important et deviendra une partie omniprésente de notre vie dans les maisons, les véhicules, le trafic, la production alimentaire et soins de santé, suivi et contrôle de nos activités.

En quelques mots un RCSFs est composé de petits dispositifs appelés capteurs dotés de ressources limitées, autonomes, capables de détecter, traiter et communiquer. Cela implique qu'ils doivent utiliser efficacement leurs ressources, y compris l'utilisation de la mémoire, la puissance du processeur, et le plus important encore, l'énergie pour augmenter leur durée de vie et leur productivité.

L'énergie et la sécurité sont les principaux défis des RCSFs, cependant, aujourd'hui, les chercheurs dans ce domaine tentent de trouver des solutions pour minimiser et optimiser la consommation de la batterie tout en gardant les performances du réseau.

Vu les domaines d'applications des RCSFs, le besoin d'apporter une solution de sécurité fiable parait important voire crucial, mais 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.

La cryptographie est toujours considérée comme une des solutions dominantes pour assurer la confidentialité des informations, en utilisant des méthodes mathématiques qui nous permettent de rendre les messages illisibles. L'utilisation de la cryptographie implique souvent des calculs compliqués et intensifs, ce qui représente un challenge à relever dans les réseaux de capteurs. Il existe deux types de cryptographie : symétrique et asymétrique, la première offre une performance de calcul plus intéressante sans utiliser une clé extrêmement longue et comme on partage la même clé pour le chiffrement et le déchiffrement, la mise en place d'une distribution sûre des clés au sein d'un réseau contenant un grand nombre de nœuds devient un problème urgent et sérieux. Contrairement à ce type, la cryptographie asymétrique offre des protocoles sophistiqués pour la génération de clés et elle nous permet de signer des messages, cependant, l'application de cette solution nécessite des calculs plus complexes et l'utilisation de clés

beaucoup plus longues. Le cryptosystème le plus utilise est le RSA, qui pour avoir une sécurité assez robuste, il faut utiliser une clé comprise entre 1024 et 2048 bits. Un autre cryptosystème attire l'attention des chercheurs aujourd'hui est celui de ECC (Elliptic Curve Cryptography) qui offre le même niveau de sécurité que RSA avec une clé beaucoup plus courte et il existe des méthodes mathématiques qui nous permettent d'améliorer sa performance ce qui est intéressant dans les réseaux de capteurs et le domaine des systèmes embarqués.

Dans ce travail nous avons optimisé l'énergie dans un réseau de capteurs sans fil et cela en proposant une solution d'optimisation des calculs cryptographiques sur les courbes elliptiques en se basant sur une approche distribuée.

Notre mémoire s'articule autour de quatre chapitres :

Le chapitre 1 : Est une introduction aux réseaux de capteurs sans fil, il présente aussi le fonctionnement général de ce type de réseau, ses applications potentielles et ses principales caractéristiques.

Le chapitre 2 : Portera sur la définition et les opérations sur les courbes elliptiques et une taxonomie des protocoles de sécurité existants dans la littérature.

Le chapitre 3 : Nous présenterons les différentes méthodes et algorithmes d'optimisation et d'accélération des calculs cryptographiques elliptiques.

Le chapitre 4 : Est une présentation de notre solution et les outils matériels et logiciels pour l'implémentation de cette solution, ainsi que les résultats obtenus.

Enfin nous terminerons par une conclusion générale et des perspectives.

CHAPITRE I

GENERALITES SUR LES RESEAUX DE CAPTEURS SANS FIL

I.1 Introduction

Au cours de ces dernières années, le développement technologique des réseaux de communication sans fil a connu un essor important grâce aux avancées technologiques dans divers domaines, tels que la micro-électronique et la miniaturisation. C'est ainsi que de nouvelles voies d'investigation ont été ouvertes avec l'émergence des réseaux de capteurs sans fil qui sont composés de petits dispositifs électroniques. Ces éléments sont autonomes, équipés de capteurs et capables de communiquer entre eux sans fil.

I.2 Les réseaux de capteurs sans fil

I.2.1 Un capteur

Un capteur est un dispositif électronique de taille extrêmement réduite avec des ressources très limitées, autonomes, capable d'acquérir des données, de les traiter et de les transmettre, via les ondes radio, à une autre entité (capteurs, unité de traitements...) sur une distance limitée à quelques mètres.

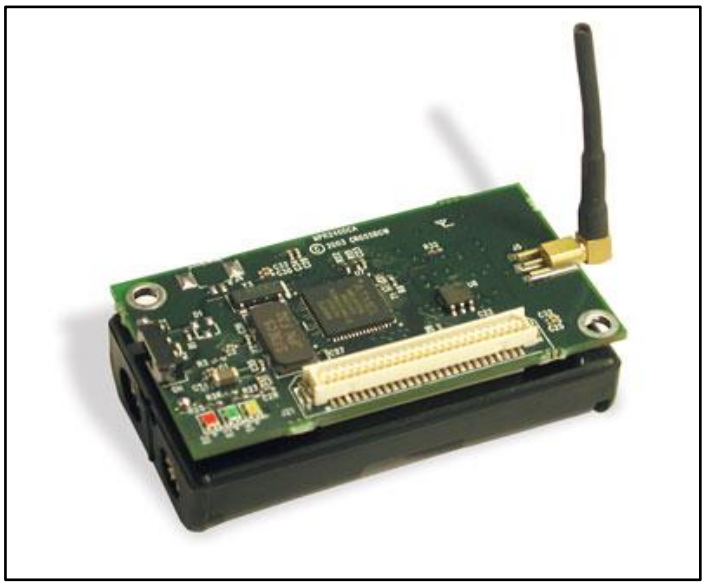

Figure I.1 : capteur MicaZ

I.2.2 Anatomie d'un capteur

I.2.2.1 Coté hardware

Un nœud senseur est composé de quatre (04) unités de base :

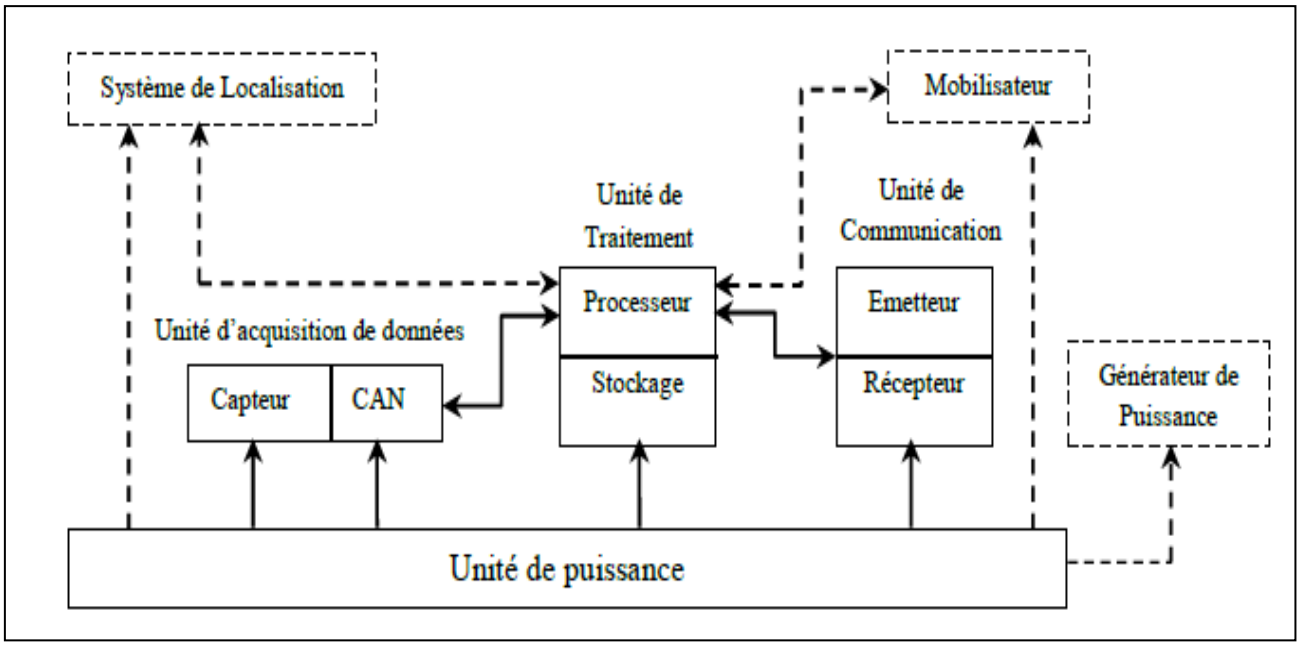

Figure I.2 : Architecture d'un capteur sans fil [30]

➢ **Une unité de détection (d'acquisition) :** composées de deux (02) sous unités :

a. Une sous unité de détection (capteur) : permet de prendre des mesures physiques (pression, température, pollution, son, radiation…)

b. Une sous unité ADC (Analog Digital Converter) : un convertisseur analogique numérique qui transforme le signal analogique en données numériques et les transmet à l'unité de traitement.

On retrouve donc des équipements de différents types de détecteur et d'autres entrées comme LED, interface, capteur…

➢ **L'unité de traitement :** composée de :

a- Processeur embarqué : Le rôle du processeur est d'ordonnancer les taches, traiter les données et contrôler les autres composants du senseur. Il opère autour de 16 Mhz de fréquence dans le cas d'un capteur de type MicaZ.

b- Unité de stockage : inclut la mémoire de travail et la mémoire réservée aux données.

➢ **L'unité de transmission/réception (communication) :** connecte le nœud au réseau (échange de donnée entres nœuds). Cette unité est la plus gourmande en énergie et sa consommation augmente lors de la transmission ou de réception.

Les technologies de communications:

- Optique (laser)
- Infrarouge
- Fréquence radio

- **L'unité d'énergie (batterie) :**

Permet d'alimenter le reste des composants, généralement on utilise une batterie de type AAA et fonctionne sur une plage théorique allant de 2,7 à 3,3 Volts. Elle est souvent inchangeable.

Les caractéristiques principales de quelques nœuds capteurs sans fil sont représentées dans le Tableau I.1 [28]

Nœud capteur	**MicaZ**	**TelosB**	**WSN430**
Processeur	Atmel AT-Mega 128L	TI MSP430	TI MSP430
Vitesse processeur	16 MHz	8 MHz	8 MHz
Taille RAM	4 Ko	10 Ko	10 Ko
Espace programme	128 Ko	48 Ko	48 Ko
Radio	TI CC2420 IEEE 802.15.4		TI CC 1100
Fréquence	2400-2483		315/433/868/915
Voltage	2.7 V	1.8 – 3.6 V	3.7 V

Tableau I.1 : Caractéristiques de quelques nœuds capteurs sans fil [28]

Un capteur peut contenir également, selon son domaine d'application, des modules supplémentaires tels qu'un système de localisation (GPS), ou bien un système générateur d'énergie (cellule solaire). On peut même trouver des micro-capteurs, un peu plus volumineux, dotés d'un système mobilisateur chargé de déplacer le micro-capteur en cas de nécessité.

I.2.2.2 Coté software

Le système d'exploitation et les applications embarquées sur le capteur. Le SE le plus utilisée est TinyOs sa conception est faite en NesC.

I.3 Spécificités des réseaux de capteurs sans fil

Un Réseau de Capteurs Sans Fil, Wireless Sensor Networks en anglais (WSN) est un système distribué de grande échelle mettant en communication un grand nombre de capteurs. Dans ces réseaux, chaque nœud est capable de surveiller son environnement et de réagir, en cas de besoins, en envoyant l'information collectée, à un ou plusieurs points de collecte, à l'aide d'une connexion sans fil.

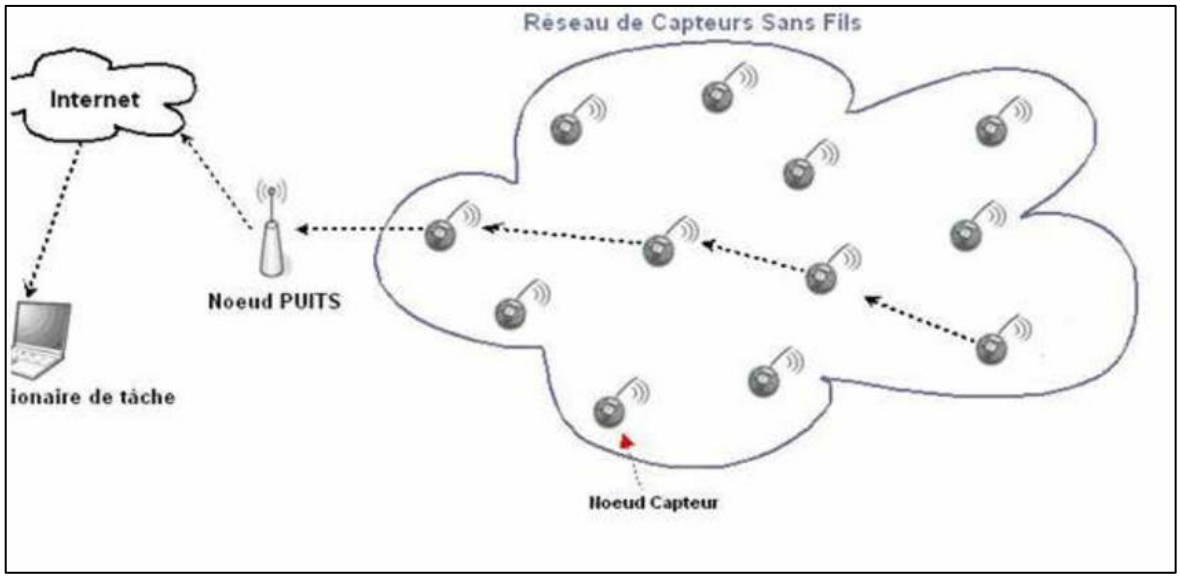

Figure I.3 : exemple de Schéma d'un réseau de capteurs sans fil

Les réseaux de capteurs se caractérisent par :

I.3.1 Energie limitée

Un capteur est limité en énergie. Dans la plupart des cas, le remplacement de la batterie est impossible. Ce qui veut dire que la durée de vie d'un capteur dépend grandement de la durée de vie de la batterie.

Les communications sont les actions les plus coûteuses en termes d'énergie. Les calculs le sont aussi, mais de moindre importance. Il est donc fortement nécessaire de limiter le nombre de communications entre capteurs et si possible le nombre de calculs.

I.3.2 Topologie

Un RCSF est composé d'un ensemble de nœuds capteurs. Ces nœuds capteurs sont organisés en champs « sensor fields ». Chacun de ces nœuds a la capacité de collecter des données et de les transférer aux nœuds passerelles (dit "sink" en anglais ou puits) par

l'intermédiaire d'une architecture multi-sauts. Le puits transmet ensuite ces données par Internet ou par satellite à l'ordinateur central « Gestionnaire de taches » pour analyser ces donner et prendre des décisions.

I.3.3 Routage

Pour limiter le nombre de communications, coûteuses en énergie, les réseaux de capteurs sans fil utilisent des protocoles de routage efficaces. Une solution souvent utilisée est la clusterisation, qui divise le réseau en plusieurs clusters. Dans chacun de ces clusters, un nœud maître (cluster-Head) est élu et a pour mission de récupérer les informations des nœuds du cluster dont il a la charge de les transmettre aux autres clusters et inversement. Le choix du nœud maître sera fait en désignant par exemple le nœud avec l'énergie la plus importante, pour augmenter la vie du réseau.

I.3.4 Tolérance aux pannes

Certains nœuds peuvent générer des erreurs ou ne plus fonctionner à cause d'un manque d'énergie, un problème physique ou une interférence. Ces problèmes ne doivent pas affecter le reste du réseau. Dans ce cas de figure, le réseau doit être capable de détecter ce type d'erreur est d'y remédier, en cherchant par exemple à modifier ses tables de routage pour trouver un autre chemin permettant de transmettre l'information et de maintenir le réseau toujours opérationnel. De la même manière, les capteurs doivent pouvoir détecter des capteurs défaillants qui envoient des informations erronées du fait de leur état.

Donc, la tolérance de fautes est la capacité de maintenir les fonctionnalités du réseau sans interruptions dues à une erreur intervenue sur un ou plusieurs capteurs.

I.3.5 Mise à l'échelle

Le nombre de nœuds déployés pour un projet peut atteindre un million. Un nombre aussi important de nœuds engendre beaucoup de transmissions inter nodales et nécessite que le puits "sink " soit équipé de beaucoup de mémoire pour stocker les informations reçues ; Pour cela les protocoles des réseaux de capteurs sans fil doivent être capables de fonctionner et de s'adapter selon le nombre de nœuds.

I.3.6 Faible 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 en calculs qui ne permet pas

d'utiliser des algorithmes complexes dans les réseaux de capteurs sans fil, et particulièrement dans la cryptographie poussée. La faiblesse de la 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, sa réactivité va sensiblement se détériorer.

I.4 Cas d'utilisation des réseaux de capteurs sans fil

Le succès de cette technologie est dû à l'ensemble de ses propriétés, notamment les couts réduits et la facilité de déploiement, la communication sans fil, la possibilité de déploiement des capteurs au sein de l'environnement surveillé parfois hostile à la présence humaine. Ces propriétés ont encouragé l'utilisation des RCSF dans diverses applications de monitoring qui étaient jusqu'à-là couteuses, complexes, voire irréalisables [29]. Parmi ces applications, nous citons :

- ***Environnementale*** : le réchauffement de la planète, les changements climatiques, l'augmentation de la pollution, la sécheresse…
- **Découverte des catastrophes naturelles :** feux de forêts, tempêtes, inondations, volcans, séismes …etc.
- **Surveillance médicale :** surveiller la progression d'une maladie ou la reconstruction d'un muscle…etc.
- **Militaire :** pour la surveillance des frontières, la surveillance des mouvements des ennemis lors d'une guerre.

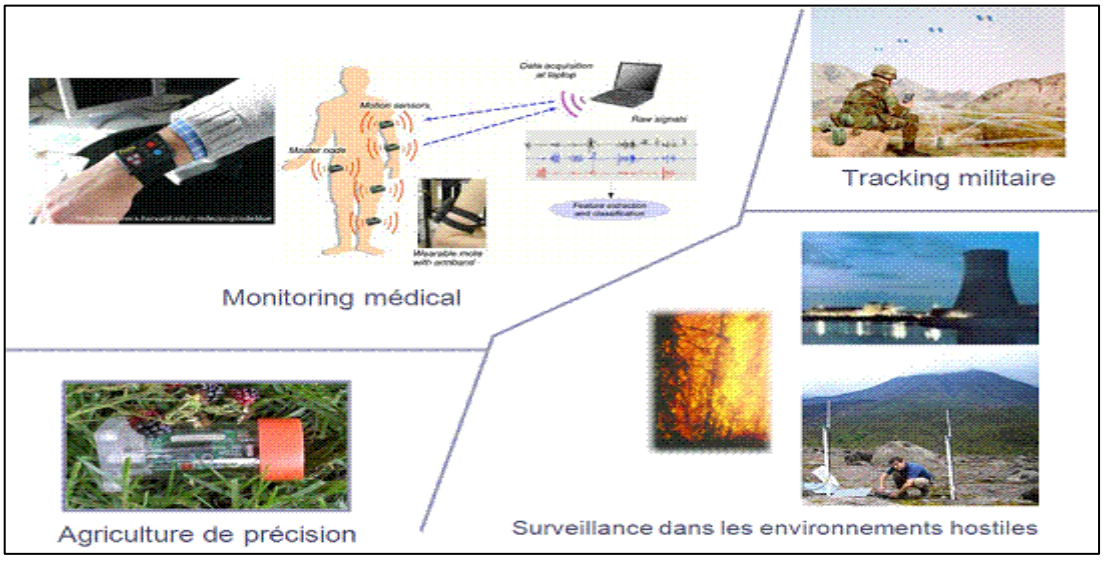

Figure I.4 : Domaines d'application des WSN

En général, on peut placer des capteurs qui détectent les personnes, les véhicules, la voix et les mouvements dans un réseau avec une caméra miniature pour capturer des images.

Cependant, vu la sensibilité de ces applications potentielles, généralement étroitement liées au monde physique, au contrôle des systèmes et à l'humain, un déploiement à grande échelle de RCSF dépend de la fiabilité qu'offre cette technologie. En particulier, la sécurité émerge comme une question difficile dans les RCSF en raison de limitation des ressources (faible puissance de calculs, Energie, mémoire, bande passante, etc.) dans ce type de réseaux.

Pour contrer les attaques qui menacent les réseaux de capteurs sans fil, plusieurs équipes de recherche tentent de trouver des solutions appropriées. Ces solutions doivent bien sûr prendre en compte les spécificités des réseaux de capteurs sans fil. Il faut donc trouver des solutions simples qui permettent de sécuriser le réseau tout en consommant le moins d'énergie possible et adapter ces solutions à une puissance de calcul faible.

Dans l'éventail de ces solutions, on trouve la cryptographie des courbes elliptiques (ECC) basée sur la cryptographie asymétrique ou à clé publique.

I.5 Conclusion

Dans cette partie, nous avons présenté brièvement les caractéristiques des réseaux de capteurs et leurs domaines d'application qui exigent la sécurité des données. Cependant, Les capteurs sans fil ne détiennent qu'une mémoire et une puissance de calcul limitée, et les calculs cryptographiques restent toujours très lourds. Dans le chapitre qui suit nous allons étudier la cryptographie des courbes elliptiques ainsi que les différents protocoles de sécurité dédiés aux RCSFs.

CHAPITRE II

LA CRYPTOGRAPHIE SUR LES COURBES ELLIPTIQUES

II.1 Introduction

La cryptographie est toujours considérée comme une des solutions dominantes pour assurer la confidentialité des informations. La nature de cette solution permet de rendre un message illisible en utilisant un ensemble de méthodes mathématiques, ce qui implique souvent des calculs compliqués et intensifs, qui posent de problèmes sérieux pour les systèmes avec contrainte de ressources.

Il existe deux manières de crypter un message : symétrique ou asymétrique. La cryptographie symétrique se base sur le partage d'une même clé K de chiffrement entre deux entités pour chiffrer ou déchiffrer les données en utilisant un algorithme de chiffrement symétrique, dans notre étude on s'intéresse à la cryptographie asymétrique qui utilise deux clés; une clé publique, connues par tous, sert à chiffrer la donnée afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire, et l'autre privé sert à déchiffrer et n'est connue que par son détenteur.

Un des crypto-systèmes asymétrique qui a récemment attiré l'attention des chercheurs, est la cryptographie sur les courbes elliptiques, elliptic curve cryptography (ECC) en Anglais, proposée indépendamment par Koblitz [11] et Miller [15] dans les années 80. Elle comprend un ensemble de techniques qui nous permettent de sécuriser des données en consommant moins de ressources. Elle attire récemment de plus en plus l'attention des chercheurs du monde entier, notamment pour les systèmes embarqués dans lequel les dispositifs électroniques ne possèdent qu'une puissance de calcul très limitée.

L'avantage le plus important d'ECC par rapport aux autres algorithmes de cryptographie asymétrique, par exemple RSA [12], est que l'on peut avoir un bon niveau de sécurité en utilisant une clé beaucoup plus courte. Le Tableau suivant présente une comparaison en termes de taille de clés des crypto systèmes ECC et RSA pour le même niveau de sécurité.

Clés RSA	Clés ECC
1024	160
2048	224
3072	256
7680	384
5360	521

Tableau II.1 : Comparaison entre ECC et RSA (paramètres de NIST)

Cette qualité rend les ECCs plus attractifs pour les dispositifs aux ressources limitées telles que la capacité de calcul, de transmission/réception et de stockage mémoire qui peuvent impacter fortement sur la consommation énergétique dans le cas des capteurs sans fil.

Pour expliquer le fonctionnement d'ECC, dans ce chapitre nous étudions d'abord les concepts mathématiques fondamentaux de la courbe elliptique, ainsi que l'opération la plus importante sur les courbes, la multiplication scalaire. Ensuite nous présentons quelques protocoles de sécurité basés sur les courbes elliptiques.

II.2 Les courbes elliptiques pour la cryptographie

Nous nous inspirons ici de la présentation faite par Hankerson et al. Dans [5]. Une courbe elliptique E est définie sur un corps fini F notée E(F), par son équation à la forme de Weierstrass [3] :

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6. \quad \textbf{(II.1)}$$

Où a_1, a_2, a_3, a_4 et $a_6 \in F$.

Dans cette étude, nous allons travailler avec les corps premiers F_p, l'équation de Weierstrass pour une courbe elliptique définie sur un corps fini premier notée E (F_p) peut être représentée par :

$$E : y^2 = x^3 + ax + b. \quad \textbf{(II.2)}$$

Où a et b ϵ F_p.

II.2.1 Les opérations sur les courbes elliptiques

II.2.1.1 L'addition et le doublement

Dans [2] et [7], l'addition et le doublement s'effectuent entre points d'une courbe elliptique et le résultat est un autre point de la courbe elliptique.
Le calcul de l'addition de point est montré dans les formules II.3 et II.4. Supposons que

$P = (x_1, y_1) \epsilon E$, $Q = (x_2, y_2) \epsilon E$ et $P \neq \pm Q$, alors $P + Q = (x_3, y_3)$ où :

$$x_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)^2 - x_1 - x_2 \qquad et \qquad y_3 = \left(\frac{y_2 - y_1}{x_2 - x_1}\right)(x_1 - x_3) - y_1 \quad \textbf{(II.3)}$$

Si $P = (x_1, y_1) \in E$ et $P \neq -P$, alors $2P = (x_3, y_3)$ où

$$x_3 = \left(\frac{3x_1^2 + a}{2y^1}\right)^2 - 2x_1 \quad et \quad y_3 = \left(\frac{3x_1^2 + a}{2y_1}\right)(x_1 - x_3) - y_1 \qquad \textbf{(II.4)}$$

Une représentation géométrique est donnée dans la figure II.1. Pour additionner les points P et Q, nous traçons une droite qui passe par ces 2 points, le résultat de l'addition est le point symétrique par rapport à l'axe abscisse du 3ème point d'intersection avec la courbe. Pour doubler le point P, c'est-à-dire 2P, il suffit de trouver la tangente à la courbe au point P, et le résultat du doublement est le point symétrique par rapport à l'axe abscisse du 2ème point d'intersection avec la courbe.

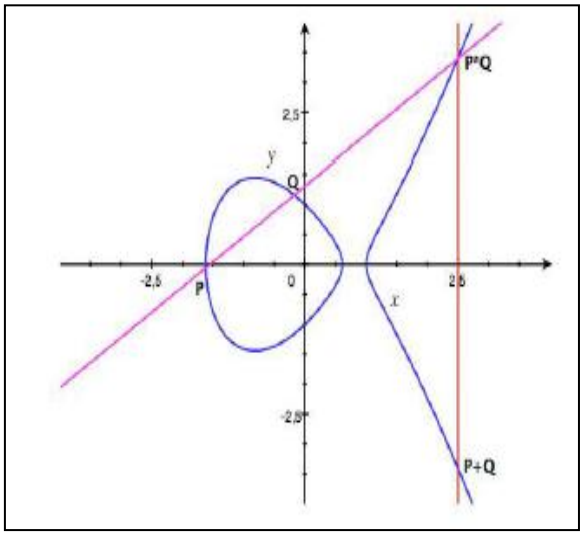

Figure II.3 : addition de points

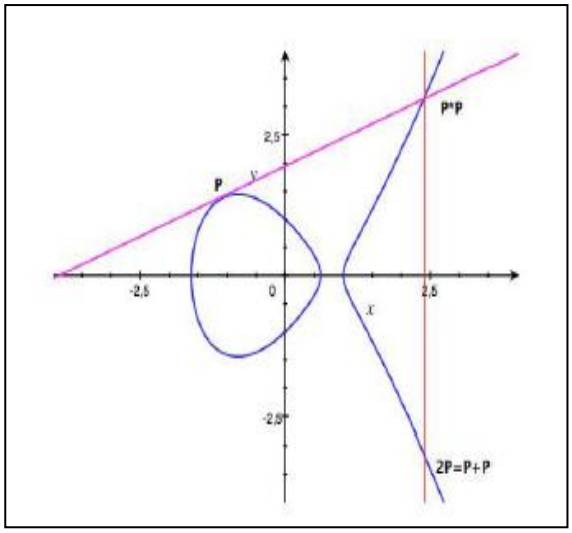

Figure II.4: doublement de points

II.2.1.2 Multiplication scalaire

En se basant sur l'addition de point, nous pouvons encore effectuer la multiplication de point, notée Q = kP sur une courbe elliptique E où k ∈ Z+ et (P, Q) ∈ E. Cette opération est appelée aussi la multiplication scalaire, qui est considérée comme l'opération la plus coûteuse et importante sur les courbes elliptiques. La multiplication de point est une opération cruciale dans les protocoles cryptographiques basés sur les courbes elliptiques. Elle est considérée comme une suite d'addition de points consécutifs :

$$Q = k*P = \underbrace{P + P + P + \ldots + P}_{\textbf{K fois P}}$$

II.2.2 Algorithme de produit scalaire : double-and-add (DA)

Supposons que P est un point sur une courbe elliptique qui est définie sur un corps premier, notée $E(F_p)$, pour calculer kP où k est un nombre entier positif de longueur *l* bits, nous représentons k en binaire $k = \sum_{i=0}^{l-1} \boldsymbol{k_i 2^i}$, et ensuite nous parcourons k du bit de poids faible au bit de poids fort (ou inversement). Un doublement est effectuée pour chaque bit k_i de *k*, suivi d'une addition pour chaque bit non nul ($k_i \neq 0$).(voir algorithme II.1).

Algorithme II.1 : Algorithme doublement-et-addition pour calculer Q = kP bit faible /bit fort

Données : k = $\sum_{i=0}^{l-1} k_i 2^i$
Résultat : Q = [k]P
Q ← ∞;
pour i de 0 à l-1 **faire**
// début du balayage bit par bit de bit faible / vers bit fort
si ki = 1 **alors**
Q ← Q + P //on effectue une addition
Fin
P ← 2P //on effectue un doublement
fin
retourner Q

Avec cette méthode, la densité moyenne des bits non-nuls (bits à 1) sur l'ensemble de tous les scalaires k de longueur *l* bits est approximativement *l*/2. Ainsi, l'algorithme 1 effectue en moyenne *l* doublements et *l*/2 additions.

II.3 Logarithme discret

Le principe de la cryptographie à clé publique repose sur un couple de clés, l'une publique, l'autre privée. Retrouver la clé privée à partir de la clé publique doit revenir à résoudre le problème de Logarithme discret considéré comme difficile. Si P est un point de la courbe elliptique E (F_p) d'ordre n, et Q un autre point de la courbe, la difficulté de trouver l'entier k Є [0, n-1] tel que Q=kP est appelé : le problème du logarithme discret. La solution la plus naïve pour résoudre ce problème est de calculer exhaustivement 1P, 2P, 3P … jusqu'à ce que nous trouvions Q, mais le calcul peut devenir extrêmement long si la valeur de k est suffisamment grande. Il est donc énormément difficile de retrouver la valeur de k à partir de Q et P. Il n'y a pas de preuve mathématique qui peut démontrer que l'ECDLP est insoluble, mais la résolution d'un tel problème est toujours considérée comme infaisable en prenant compte de l'état actuel des technologies informatiques [5].

- **Génération de clés avec le Logarithme Discret Elliptique :** Une paire de clés est associée à un ensemble de paramètres publiques (p, E, P, n) où p est un nombre premier, E une courbe elliptique, P le point générateur et n son ordre, c'est à dire que nP = ∞.

Une clé privée d est sélectionnée au hasard dans l'intervalle [1, n-1] et la clé publique correspondante est Q = dP.

Le problème du logarithme discret consiste à déterminer d à partir des paramètres publics (p, E, P, n).

Algorithme II.2 : Génération de clés avec le Logarithme Discret Elliptique

Entrées : p, E, P, n // les paramètres publiques générés

Sorties : Clé publique (Q) et clé privée d générées

1. Sélectionner au hasard d dans l'intervalle [1, n-1]
2. Calculer Q = dP
3. Retourner (Q, d)

Le problème du logarithme discret sur une courbe elliptique bien choisie est plus rapide que celui du logarithme discret dans les corps finis. L'algorithme connu comme étant le plus efficace pour résoudre un tel problème est à temps exponentiel, contrairement au système RSA pour lequel il existe des algorithmes a temps sous-exponentiel. Un algorithme est sous-exponentiel si le logarithme du temps d'exécution croît asymptotiquement moins vite que tout polynôme donné [28].

II.4 les protocoles de sécurité basés sur les courbes elliptiques

II.4.1 Déffie- Hellman-échange de clés

Dans [21], WHITFIELD DIFFIE et MARTIN E. HELLMAN, ont proposé un protocole d'échange de clés (voir figure II.3) Alice et Bob veulent avoir une clé en commun pour s'échanger des données en toute sécurité. Supposons que leur seul moyen de communication soit public. Un des moyens de sécuriser leurs données est qu'ils établissent une clé privée entre eux.

A et B veulent partager un secret entre eux, et chacun détient une clé privée appelé respectivement x_A et x_B, p est un grand nombre premier, et α est la racine primitive modulo p (le générateur). Le déroulement de l'algorithme est montré dans la figure II.3 :

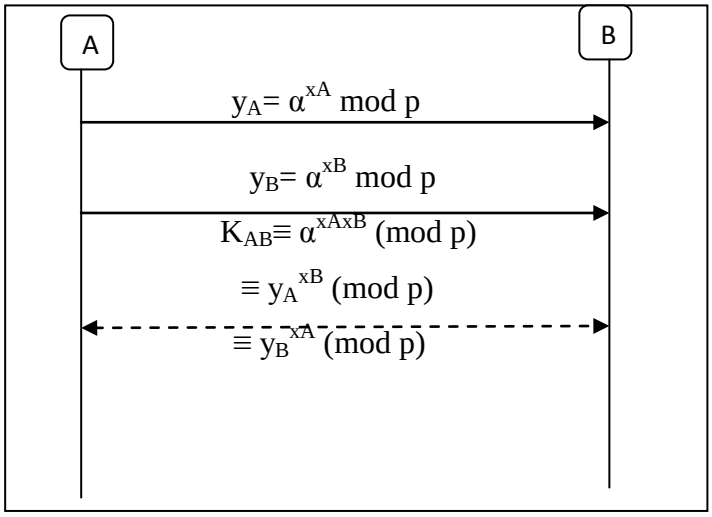

Figure II.3: échange de clés Diffie-Hellman

Cet algorithme peut être appliqué sur les courbes elliptiques :

1. Alice et Bob choisissent une courbe elliptique E définie sur un corps fini Fp tel que le logarithme discret soit difficile à résoudre. Ils choisissent aussi un point $P \in E(Fq)$ tel que le sous-groupe généré par P ait un ordre de grande taille. (En général, la courbe E et le point P sont choisis de manière à ce que l'ordre soit un grand nombre premier.)

2. Alice choisit un nombre entier secret x_A, calcule $P_{\mathbf{A}} = x_A P$ et envoie $P_{\mathbf{A}}$ à Bob.

3. Bob choisit un nombre entier secret x_B, calcule $P_{\mathbf{B}} = x_B P$ et envoie $P_{\mathbf{B}}$ à Alice.

4. Alice calcule $K_{AB} = x_A x_B P$.

5. Bob calcule $K_{AB} = x_B x_A P$.

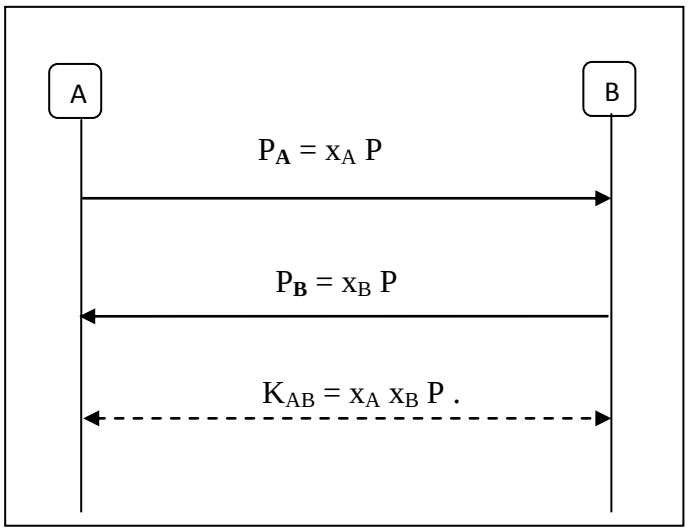

Figure II.4 : échange de clés Diffie-Hellman ECC.

Grâce à la difficulté de résoudre le problème du logarithme discret, Alice et Bob peuvent s'envoyer P_A et P_B sans risque, car il est extrêmement difficile de calculer les valeurs x_A et x_B. Une fois le secret partagé (K_{AB}) établi entre Alice et Bob, ils peuvent sécuriser la communication entre eux avec un algorithme de cryptographie symétrique qui est moins lourd à gérer.

II.4.2 El gamal

Dans [14], ElGamal propose un protocole de chiffrement et de signature numérique basé sur le protocole d'échange de clé Diffie-Hellman [21].

II.4.2.1 Chiffrement et déchiffrement

Alice veut envoyer un message secret à Bob. Tout d'abord, Bob fabrique une clé publique de la manière suivante. Il choisit une courbe elliptique E définie sur un corps fini F_p de telle manière que le problème du logarithme discret soit plus difficile à résoudre sur $E(F_p)$ que sur F_p. Il choisit aussi un point P sur E tel que l'ordre de P soit un grand nombre premier. Il choisit un nombre entier secret x_B et calcule $P_B = x_B P$. La courbe E, le corps fini F_p et les points P et P_B sont la clé publique de Bob. La clé secrète de Bob est x_B. Pour envoyer le message, Alice fait comme suit (voir figure II.5).

1. Elle télécharge la clé publique de Bob.

2. Elle convertit son message en clair M en un point P_M sur la courbe,

3. Elle choisit un nombre entier secret $k \in \{1, p-1\}$, et calcule $M1 = kP$.

4. Elle calcule $M_2 = P_M + k.P_B$, Le texte chiffré est une paire de point $Pc = (M_1, M_2)$

5. Elle envoie (M_1, M_2) à Bob.

Bob déchiffre le message en calculant :

$P_M = M_2 - x_B.M_1 = P_M + k.P_B - x_B.k.P = P_M + (k.P_B - k.P_B)$

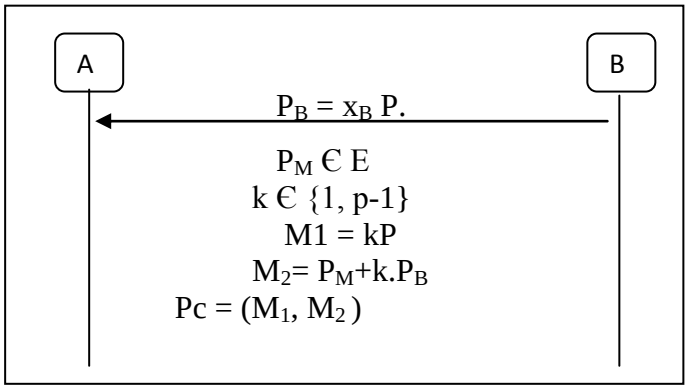

Figure II.5 : Protocole de chiffrement d'Elgamal

II.4.2.2 Signature numérique

L'objectif de la signature numérique est de garantir l'intégrité d'un document et d'authentifier l'auteur. En vérifiant la signature, le récepteur du document peut identifier le signataire du document et détecter si le document a été modifié pendant la transmission.
Pour signer un document, le signataire doit trouver une solution qui lui permette de signer le document un utilisant sa clé privée, et les autres personnes peuvent vérifier la signature avec la clé publique du signataire.

Un nouveau schéma de signature est décrit dans ElGamal [14], le fichier public contient les mêmes clés publiques pour le cryptage de messages ainsi que la vérification des signatures.
Soit m un document à signer, où $m \in [0, p-1]$. Le fichier public est toujours constitué de la clé publique $y \equiv \alpha^x \pmod p$ pour chaque utilisateur. Pour signer un document, un utilisateur A devrait pouvoir utiliser la clé secrète x pour trouver une signature pour m dans une telle façon que tous les utilisateurs peuvent vérifier l'authenticité de la signature en utilisant la clé publique y, avec (α et p), et personne ne peut forger une signature sans connaître le secret x.
La signature de m est le couple (r, s) que $\alpha^m \equiv y^r r^s \pmod p$ (II.5)

Avant de lancer la procédure de signature, le signataire trouve un nombre entier
$k \in [0, p-1]$ tel que PGCD $(k, p-1) = 1$. Ensuite il calcule

$r \equiv \alpha^k \pmod p$

et la formule (II.5) peut être transformée à

$\alpha^m \equiv \alpha^{xr}\alpha^{ks} \pmod p$

A partir de laquelle nous pouvons calculer s en utilisant l'équation

$m \equiv xr + ks \pmod{p-1}$

Une fois que le document est arrivé chez destinataire, il suffit que le récepteur calcule les deux côtés de l'équation (II.5) pour voir s'ils sont égaux.

Cet algorithme doit être légèrement modifié avant de pouvoir être utilisé avec les courbes elliptiques [23]. Afin de sécuriser la communication entre eux, utilisateurs A et B choisissent la même courbe $E(F_p)$ dont le point de générateur est P. Avant d'envoyer le message M à B, il faut qu'il soit signé par A. Supposons que la clé privée de A soit x_A, et avec laquelle on peut calculer facilement sa clé publique $P_A = x_A P$.
La procédure de signature est constituée de 5 étapes.(voir figure II.6) :

1. Choisir d'abord un nombre entier aléatoire k Є [0, p − 1].

2. Calculer R = kP = (x_R, y_R) et $r \equiv x_R \pmod p$. Si r = 0, répéter l'étape précédente.

3. Calculer e = h(M) où h est une fonction de hachage.

4. Calculer $s \equiv k^{-1}(e + rx_A) \pmod p$. Si s = 0, recommencer depuis l'étape 1.

5. Envoyer (R, s) à utilisateur B.

Après la réception de (R, s), afin de vérifier la signature du document, B calcule $V_1 = sR$ et $V_2 = h(M)P + rP_A$. La signature est valide si $V_1 = V_2$

$$\begin{cases} V1 = k^{-1}(e + rxA)kP = P(h(M) + rxA) \\ V2 = h(M)P + rxAP = P(h(M) + rxA) \end{cases} (3.39)$$

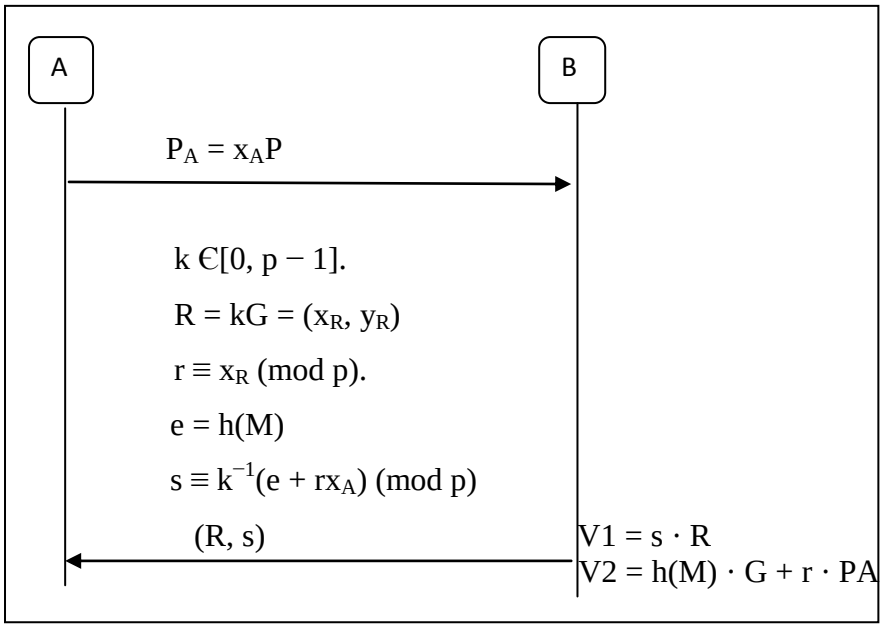

Figure II.6 : Protocole de signature numérique d'Elgamal

II.4.3 ELLIPTIC CURVE INTEGRATED ENCRYPTION SCHEME (ECIES)

Le protocole d'Elgamal est rarement utilisé directement avec les courbes elliptiques. Avant de chiffrer un message, il faut d'abord le convertir à un point sur la courbe elliptique utilisée. Il y a différentes techniques qui existent, mais la conversion nécessite plus de calcul.

Généralement on utilise les courbes elliptiques pour établir une clé partagée entre les deux parties d'une conversation [21], ensuite nous pouvons utiliser un algorithme de cryptographie symétrique pour sécuriser la communication entre elles.

Le protocole ECIES est en effet une variante d'Elgamal standardisée. Supposons qu'Alice désire envoyer un message M à Bob d'une manière sécurisée, ils doivent d'abord disposer de toutes les informations suivantes :

– KDF (Key Derivation Function) : Une fonction de dérivation de clé qui permet de générer plusieurs clés à partir d'une valeur secrète de référence.

– MAC (Message Authentication Code) : Code transmis avec les données dans le but d'assurer l'intégrité de ces dernières.

– SYM : Algorithme de chiffrement symétrique.

– $E(F_p)$: La courbe elliptique utilisée avec le point de générateur P dont ord(P) = n.

– K_B : La clé publique de Bob $K_B = k_B.P$ où $k_B \in [1, n-1]$ est sa clé privée.

Pour chiffrer le message M, Alice doit effectuer des opérations suivantes :

1. Choisir un nombre entier $k \in [1, n-1]$ et calculer R = kP.

2. Calculer $Z = k.K_B$.

3. Générer les clés (k_1, k_2) = KDF (abscisse(Z), R).

4. Chiffrer le message C = SYM (k_1, M).

5. Générer le code MAC t = MAC (k_2, C).

6. Envoyer (R, C, t) à Bob.

Pour déchiffrer le message (R, C, t), Bob doit effectuer des calculs ci-dessous :

1. Rejeter le message si R n'appartient pas à $E(F_p)$.

2. Calculer $Z = k_B.R = k_B.k.P = k.K_B$.

3. Générer les clés (k_1, k_2) = KDF (abscisse(Z), R).

4. Générer le code MAC $t' = MAC (k_2, C)$.

5. Rejeter le message si $t' \neq t$.

6. Déchiffrer le message $M = SYM^{-1} (k_1, C)$.

Dans ce protocole, la valeur critique est k avec laquelle Bob peut calculer $Z = k.K_B$, et générer le couple (k_1, k_2) qui est utilisé pour déchiffrer et authentifier le message. Grâce aux propriétés du problème de logarithme discret, Alice peut envoyer R = k.P sans problème. Une représentation graphique du protocole est illustrée dans la figure II., et une description plus détaillée est donnée dans [25] avec les paramètres recommandés dans [24].

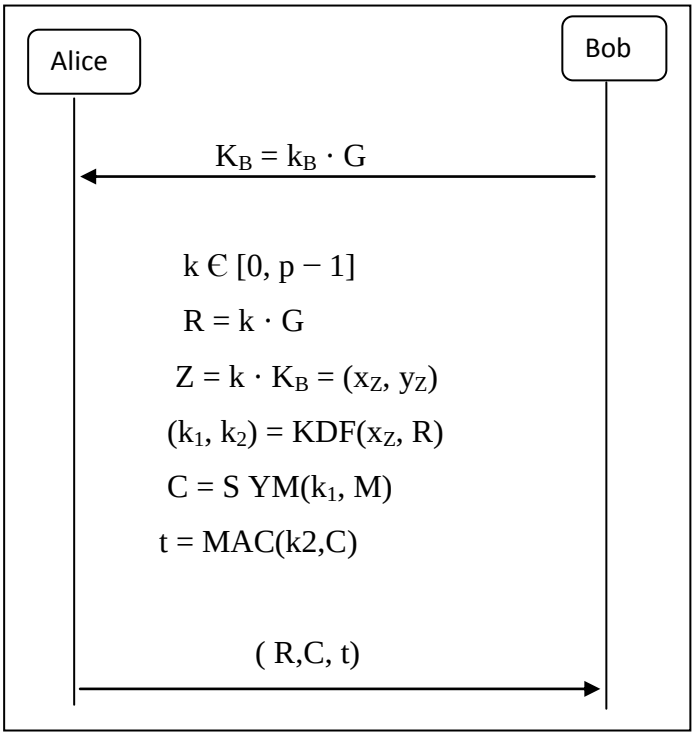

Figure II.7 : Protocole de chiffrement ECIES

II.4.4 Elliptic Curve Digital Signature Algorithm (ECDSA)

Le protocole ECDSA est proposé par Johnson et al. dans [26] est une variante de DSA qui utilise les techniques de cryptographie sur les courbes elliptiques. Le protocole DSA, signifie Digital Signature Algorithm en Anglais, c'est un algorithme de signature numérique standardisé par le NIST aux États-Unis [27].

Le protocole est basé sur l'idée du protocole de signature d'ElGamal. Nous supposons qu'Alice et Bob utilisent la même courbe elliptique $E(F_p)$ pour sécuriser la communication entre eux. Nous supposons que la clé publique d'Alice est $K_A = k_A.P$ où k_A est sa clé privée et P est le point de générateur de l'ordre n.

Pour signer un message M, Alice doit suivre les opérations suivantes :

1. Choisir un nombre aléatoire k Є [1, n − 1].

2. Calculer R = kP.

3. Calculer r ≡ abscisse(R) (mod n). Si r = 0, retourner à l'étape 1.

4. Calculer $s \equiv k^{-1}(H(M) + k_A r) \pmod n$ où H est une fonction de hachage. Si s = 0, retourner à l'étape 1.

5. Envoyer (r, s) à Bob.

Après avoir reçu le message signé, Bob vérifie la signature du message :

1. Vérifier si $K_A \neq \infty$ (point à l'infini) et KA Є $E(F_p)$.

2. Vérifier si $n.K_A = \infty$ car $n.K_A = n.k_A.P$ et ord(P) = n.

3. Vérifier si (r, s) Є [1, n − 1].

4. Calculer $R = (H(M)s^{-1} \bmod n)P + (rs^{-1} \bmod n)K_A$ (voir la formule II.6).

5. Vérifier si r ≡ abscisse(R) (mod n).

$$\begin{aligned} R &= (H(M)s^{-1})P + (rs^{-1})K_A \pmod n \\ &= (H(M)s^{-1})P + (rs^{-1})k_AP \pmod n \\ &= s^{-1}P(H(M) + rk_A) \pmod n \qquad \text{(II.6)} \\ &= k\,(H(M) + k_Ar)^{-1}P(H(M) + rk_A) \pmod n \\ &= kP \end{aligned}$$

Le déroulement du protocole est montré dans la figure II.8.

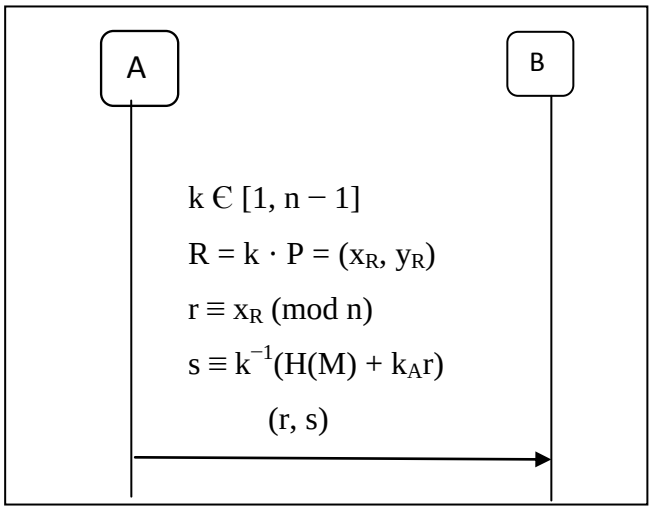

Figure II.8 : Protocole de signature numérique ECDSA

II.5 Conclusion

Dans ce chapitre nous avons présenté les concepts de base des courbes elliptiques, puis l'opération la plus gourmande en termes d'énergie et de temps de calcul et elle est impliquée dans le chiffrement et enfin quelques protocoles cryptographiques utilisant les courbes elliptiques.

Des solutions ont été proposées pour accélérer et optimiser les calculs de la multiplication scalaire afin de prolonger la durée de vie des capteurs. Le chapitre suivant sera consacré à l'étude de quelques méthodes et algorithmes d'optimisation de produit scalaire connus dans la littérature.

CHAPITRE III

LES PROTOCOLES D'OPTIMISATION DE LA MULTIPLICATION SCALAIRE

III.1 Introduction

Un des avantages de ECC est la rapidité des calculs car nous utilisons des clés plus courtes. ECC est souvent un choix adapté pour les matériels qui ne disposent que d'une mémoire et d'une puissance de calcul limités (carte à puce, microcontrôleur etc.). Comme nous avons vu dans les sections précédentes que l'opération la plus complexe sur les courbes elliptiques est la multiplication scalaire, elle est aussi fréquemment utilisée dans les protocoles cryptographiques. La performance de la multiplication scalaire a une forte influence sur la performance de l'ensemble du cryptosystème. Dans cette section nous allons voir les différentes techniques mathématiques et algorithmiques qui nous permettent d'accélérer le calcul de multiplication scalaire. Une optimisation possible consiste à trouver une représentation qui contient le plus petit nombre de bits non nuls afin de réduire le nombre d'opérations d'additions : c'est l'objectif des solutions qu'on va décrire dans la suite.

III.2 Optimisation de performance de la multiplication scalaire

III.2.1 La méthode NAF

Cette forme de codage d'un entier vient de la remarque suivante : $\forall k \in N$ tel que $k = 2^i - 1$, on a l'expression en binaire :

$$(k)_2 = \underbrace{111...1}_{\text{i fois}}$$

On peut alors écrire :

$$NAF(k) = \underbrace{100...00 - 1}_{\text{i+1 chiffres}} \qquad \textbf{(III.1)}$$

La représentation NAF est donc une transformation telle que : pour $k \in N$, $k = (k_{l-1}, ..., k_1, k_0)_2$ avec $ki \in \{0, 1\}$, $\mathbf{k} = \sum_{i=0}^{l-1} \mathbf{k'}_i \mathbf{2}^i$ où $k'i \in \{-1, 0, 1\}$.

Cette représentation utilise le chiffre -1 en plus, en revanche, elle diminue le nombre de chiffres non nuls dans la représentation d'un entier. Elle élimine en effet tous les 1 consécutifs, et les remplace comme nous venons de le voir dans l'équation (I.5). Ceci permet de réduire d'autant le nombre d'additions de points de courbe elliptique, et donc le

temps de calcul. La proportion moyenne de chiffres non nuls passe de $\frac{1}{2}$ à $\frac{1}{3}$.. Hankerson et al dans [6] décrivent un algorithme permettant de calculer cette représentation. Si la longueur de NAF(k) = *l*, alors

$\frac{2^l}{3} < k < \frac{2^{l+1}}{3}$. NAF(k) peut être calculé efficacement avec l'algorithme III.1 :

Algorithme III.1 : Calcul de la forme NAF d'un scalaire k

Données : le scalaire k (entier)
Résultat : NAF(k)
i ← 0;
tant que k ≥ 1 **faire**
  si k est impair **alors**
    ki ← 2 − (k mod 4);
    k ← k − ki;
  sinon
    ki ← 0;
  fin
  k ← $\frac{k}{2}$;
  i ← i + 1;
fin
retourner (k_{i-1}, k_{i-2},...,k_1,k_0)

Le Double-and-add est alors modifié comme montré dans l'algorithme III.2, toujours d'après Hankerson et al. Dans [6], afin de prendre en compte les coefficients valant −1 de la représentation du scalaire. La complexité de cet algorithme, pour un scalaire de longueur *l* est de *l* doublements et en moyenne de *1*/3 additions de points, pour un scalaire tiré au sort.

Pour le calcul de la multiplication scalaire kP on utilise l'algorithme suivant :

Algorithme III.2 : Méthode NAF pour calculer la multiplication

Données : NAF(k), P ∈ E(Fp)
Résultat : Q = [k]P
Q ← ∞ ;
pour i ← *l-1* **à** 0 **faire**
  // début du balayage chiffre par chiffre du chiffre fort vers chiffre faible
  Q ←2Q // On effectue un doublement
  si (ki = 1) **alors**
    Q ← Q + P ; // On effectue une addition
  si (ki = −1) **alors**
    Q ← Q – P ; // On effectue une soustraction
Fin
Retourner (Q)

III.2.2 La w-NAF

Cette représentation est une extension de la représentation NAF. En effet, plutôt que de limiter le codage de chaque chiffre à l'ensemble {−1, 0, 1}, on utilise l'ensemble : $\{-2w-1+1, ..., -5, -3, -1, 0, 1, 3, 5, ..., 2^{w-1}-1\}$. Cette astuce augmente significativement le nombre de chiffres nuls. Il n'en reste qu'une proportion de 1 / (w + 1) en moyenne. L'algorithme de cette fonction se base sur le même principe que l'algorithme 4, le Double-and-add est quant à lui modifié comme montré dans l'algorithme 6 (toujours d'après Hankerson et al. dans [6]). Le calcul de la forme NAFw(k) est montré dans l'algorithme III.3.

Algorithme III.3 : Calcul de la forme $NAF_w(k)$ d'un nombre entier positif

Données : k,w
Résultat : $NAF_w(k)$
i ← 0;
tant que k ≥ 1 **faire**
  si k est impair **alors**
    $k_i \leftarrow k \bmod 2^w$;
    $k \leftarrow k - k_i$;
  sinon
    $k_i \leftarrow 0$;
  fin
  $k \leftarrow \frac{k}{2}$;
  i ← i + 1; +
fin
retourner $(k_{i-1}, k_{i-2}, ..., k_1, k_0)$

Le calcul de multiplication scalaire en utilisant la méthode de la fenêtre est donné dans l'algorithme III.4.

Algorithme III.4 : Méthode $NAF_w(k)$ pour le calcul de multiplication scalaire

Données : $NAF_w(k)$, P ∈ E(Fp)
Résultat : k.P
Calculer $P_i = i.P$ où $i \in \{1, 3, 5, ...2^{w-1}-1\}$;
Q ← ∞;
pour i de *l* − 1 à 0 **faire**
  Q ← 2Q;
  si ki ≠ 0 **alors**
    si $k_i > 0$ **alors**
      $Q \leftarrow Q + P_{ki}$;
    sinon
      $Q \leftarrow Q - P_{-ki}$;
    fin
fin
fin
retourner Q

Nous pouvons constater qu'avant de lancer la boucle pour parcourir les k_i, il faut d'abord calculer et stocker en mémoire P_i = i.P où i ∈ {1, 3, 5, . . . $2^{w-1} - 1$}. Donc cette méthode est applicable si et seulement si le système possède assez de mémoire pour stocker temporairement l'ensemble de P_i pré-calculés.

III.2.3 Système de coordonnées

Les formules d'addition et de doublement font intervenir des inversions dans F_p, une opération compliquée et coûteuse sur les corps finis. Nous devons alors utiliser d'autres systèmes de coordonnées pour éviter ces inversions. Parmi ces systèmes, nous considérerons les systèmes de coordonnées projectives et jacobiennes.

Avec le système de coordonnées projectives, un point est représenté par 3 coordonnées (X, Y, Z) qui correspondent au point affine ($\frac{X}{z^c}$, $\frac{Y}{z^d}$) et équivalent à un point projectif ($\frac{X}{z^c}$, $\frac{Y}{z^d}$,1). Dans un système de coordonnées projectives standards, c = 1 et d = 1, le système de coordonnées jacobiennes dans lequel c = 2 et d = 3, Brown et al. dans [2] le présentent sous la forme suivante :

P = (x, y) en (X : Y : Z), avec $\begin{cases} X=\frac{X}{Z^2} \\ Y=\frac{Y}{z^3} \end{cases}$

Si on remplace x par $\frac{X}{Z^2}$ et y par Y par $\frac{Y}{z^3}$ dans l'équation courte de Weierstrass et dans les formules de doublement en coordonnées affines (I.3, I.4) données plus haut, on obtient la forme projective de l'équation de Weierstrass : $Y^2 = X^3 + aXZ^4 + bZ^6$

Pour P = (X1, Y1, Z1) ∈ E et P ≠ −P, on a P = ($\frac{X_1}{{z_1}^2}$, $\frac{Y_1}{{z_1}^3}$, 1), et en posant 2P = (X'_3, Y'_3, 1), on a :

$$X'_3 = \left(\frac{3\frac{{X_1}^2}{{Z_1}^4}+a}{2\frac{Y_1}{z_1^3}}\right) - 2\frac{X_1}{{Z_1}^2} = \frac{(3{X_1}^2+a{Z_1}^4)^2-8X_1{Y_1}^2}{4{Y_1}^2{Z_1}^2} \qquad \textbf{(III.2)}$$

$$Y'_3 = \left(\frac{3\frac{{X_1}^2}{{Z_1}^4}+a}{2\frac{Y_1}{z_1^3}}\right)\left(\frac{X_1}{{Z_1}^2} - X'_3\right) - \frac{Y_1}{z_1^3} = \frac{(3{X_1}^2+a{Z_1}^4)(4X_1{Y_1}^2-4X'_3{Y_1}^2{Z_1}^2)-8{Y_1}^4}{8y_1^3z_1^3} \qquad \textbf{(III.3)}$$

Enfin pour éliminer les dénominateurs, nous calculons le point (X_3, Y_3, Z_3) dont $X_3 = X'_3.Z_3^2$, $Y_3=Y'_3.Z_3^3$ et $Z_3 = 2Y_1Z_1$. Le point (X_3, Y_3, Z_3) est équivalent à $(X'_3, Y'_3, 1)$, car dans un systèmes de coordonnées projectives, (X_1,Y_1,Z_1) est équivalent à , (X_2,Y_2,Z_2), si $X_1=\lambda^c X_2$, $X_1 = \lambda^d Y_2$ et $Z_1 = \lambda Z_2$. Les nouvelles coordonnées sont données dans la formule (III.4) :

$$\left.\begin{aligned} X_3 &= (3X_1^2 + aZ_1^4)^2 - 8X_1Y_1^2 \\ Y_3 &= (3X_1^2 + aZ_1^4)(4X_1Y_1^2 - X_3) - 8Y_1^4 \\ Z_3 &= 2Y_1Z_1 \end{aligned}\right\} \quad \textbf{(III.4)}$$

Pour optimiser le nombre de multiplications modulaires et d'élévations au carré, on peut utiliser des variables pour stocker des résultats intermédiaires. Ceci conduit aux formules données dans la table III.1. Une démarche similaire conduit à des formules pour l'addition. Dans le cas où l'un des points est en coordonnées jacobiennes et l'autre point en coordonnées affines (avec $Z_2 = 1$ dans ce cas), on obtient une addition en coordonnées mixtes. Les formules correspondantes sont fournies dans la table III.2.

L'intérêt des systèmes de coordonnées projectives réside dans le fait de supprimer l'opération d'inversion modulaire dans les formules d'addition et de doublement de points.

$$2(X_1 : Y_1 : Z_1) = (X_3 : Y_3 : Z_3) \text{ avec}$$

$$\left.\begin{aligned} &A = 4X_1 \cdot Y_1^2,\ B = 8Y_1^2,\ C = 3(X_1 - Z_1^2) \cdot (X_1 + Z_1^2),\ D = -2A + C^2, \\ &X_3 = D,\ Y_3 = C \cdot (A - D),\ Z_3 = 2Y_1 \cdot Z_1. \end{aligned}\right.$$

TABLE III.1: Formules pour le doublement en coordonnées jacobiennes, courbe $E(F_p)$ d'après les auteurs dans [2].

$$(X_1 : Y_1 : Z_1) + (X_2 : Y_2 : 1) = (X_3 : Y_3 : Z_3)$$

$$\left.\begin{aligned} &A = X_2 \cdot Z_1^2,\ B = Y_2 \cdot Z_1^3,\ C = A - X_1,\ D = B - Y_1. \\ &X_3 = D^2 - (C^3 + 2X_1 \cdot C^2),\ Y_3 = D \cdot (X_1 \cdot C^2 - X_3) - Y_1 \cdot C^3,\ Z_3 = Z_1 \cdot C. \end{aligned}\right.$$

TABLE III.2 : Formules pour l'addition en coordonnées mixtes (jacobiennes et affines), courbe $E(F_p)$, d'après les auteurs dans [2].

Remarque :

Le point à l'infini est (1, 1, 0), et la négation d'un point (X, Y, Z) est (X, −Y, Z).

III.2.3.1 Co-Z ADDITION

Une optimisation supplémentaire de l'addition jacobienne a été proposée par Meloni [39]. Il considère deux points d'entrée partageant la même coordonnée Z. Soit P = (X_1, Y_1, Z) et Q = (X_2, Y_2, Z), l'addition co-Z de P et Q (avec P ≠Q) est définie par P + Q = (X_3, Y_3, Z_3) où:

$$X_3 = D - (B + C),\ Y_3 = (Y_2 - Y_1)(B - X_3) - E \text{ et } Z_3 = Z(X_2 - X_1) \qquad \textbf{(III.5)}$$

Avec $A = (X_2 - X_1)^2$, $B = X_1A$, $C = X_2A$, $D = (Y_2 - Y_1)^2$ et $E = Y_1(C - B)$.

Cette addition co-Z est très efficace, de plus, elle permet aussi de mettre à jour les coordonnées de P gratuitement de sorte qu'elle partage la même coordonnée Z que P + Q. En effet, un examen attentif de l'expression de B et E révèle que $B = X_1 (X_2\text{-}X_1)^2 = x_1 Z_3^2$ et $E = Y_1 (X_2 - X_1)^3 = y_1 Z_3^3$ où $(x_1, y_1) = (X_1 / Z^2, Y_1 / Z^3)$ désigne les coordonnées affines de P ; nous avons donc : $P \equiv (E : B : Z_3)$. Une telle mise à jour libre permet l'utilisation ultérieure de l'addition co-Z entre P + Q et P. On a également montré dans [38, 4] que le conjugué P - Q partageant la même coordonnée Z que P + Q peut être obtenu avec un petit coût supplémentaire. En effet, P - Q = (X'_3, Y'_3, Z_3) où :

$$X'_3 = F - (B + C) \text{ et } Y'_3 = (Y_1 + Y_2)(X'_3 - B) - E, \qquad \textbf{(III.6)}$$

Avec $F = (Y_1 + Y_2)^2$ (et A, B, C, D, et E sont défini comme dans (III.5)).

- **Attaque side-channel analysis :** Les algorithmes binaires sont simples et efficaces, mais ils ne sont pas sûrs dans un contexte où le scalaire est secret et où la mise en œuvre est soumise à une side-channel analysis (par exemple carte à puce effectuant une signature ECDSA). Side-channel analysis (SCA) exploite la fuite de l'information physique produite par un appareil lors d'un calcul cryptographique tel que sa puissance de consommation et ses rayonnements électromagnétiques [8, 17, 9]. L'implémentation de multiplication scalaire est vulnérable à deux types principaux de SCA : simple power analysis (SPA) et differential power analysis (DPA). Ce dernier utilise des corrélations entre la fuite et les données traitées, et il peut généralement être efficacement vaincu par l'utilisation de techniques de randomisation [35, 36, 37]. Un SPA peut récupérer le scalaire secret à partir d'une trace de fuite unique d'un calcul d'algorithme binaire (même en présence de la randomisation des données). La raison d'un tel défaut est ce point ajouté

et les dédoublements de points ont des flux de fonctionnement différents et induisent donc différents modèles de fuite. Le trace de fuite (leakage trace) est donc composée de plusieurs points modèles de dédoublement entrelacés par ajout de point modèles uniquement pour les itérations en boucle où le bit scalaire est égal à 1 (ou -1 dans une représentation signée). Par conséquent, une seule trace de fuite de la multiplication scalaire peut révéler tout le scalaire secret (ou des informations significatives à ce sujet dans une déclaration signée).

Afin de résister à SPA, il faut rendre la multiplication scalaire régulière, c'est-à-dire effectuer un flux d'opération constant quelle que soit la valeur scalaire. Une première possibilité est de faire de l'addition et du doublement des motifs indiscernables. Cela peut être réalisé en utilisant des formules unifiées pour l'addition ponctuelle et le doublement ponctuel [1] ou par la moyenne de side-channel atomicity dont le principe est de construire des algorithmes d'addition et de dédoublement de points à partir du même modèle atomique d'opérations [22]. Une autre possibilité est de rendre l'algorithme de multiplication scalaire lui-même régulier, indépendamment des flux d'opérations dans chaque opération ponctuelle. A savoir, on conçoit une multiplication scalaire avec un flux constant d'opérations ponctuelles. Cette approche a d'abord été suivie par Coron dans [35] qui a proposé d'effectuer une addition fictive dans la boucle de l'algorithme binaire à chaque fois le bit scalaire est égal à 0. L'algorithme double-and-add-always obtenu effectue un doublement de point et une addition de points à chaque itération de boucle et les bits scalaires ne sont plus distinguables de leakage trace. Utiliser les coordonnées jacobiennes pour R_0 (en supposant a = -3) et les coordonnées affines pour R_1, l'algorithme double-et-add-always de gauche à droite effectue une multiplication scalaire au prix de 12M + 7S + 19A par itération de boucle.

III.2.3.2 L'Echelle de Montgomery

D'autres algorithmes binaires normaux existent dans la littérature et présentent des caractéristiques attrayantes, tel que Montgomery ladder [13] et l'algorithme double-and-add proposé par Joye dans [33]. Ces algorithmes sont basés sur les invariants de boucle des registres de points R_0 et R_1. Dans l'échelle de Montgomery, la relation $R_1 - R_0 = P$ est satisfaite à la fin de chaque itération de boucle (voir [13] pour plus de détails)

Algorithme III.5: Montgomery ladder

Données: P 2 E(Fq), k = $(k_{n-1},\ldots\ldots, k_1, k_0)_2 \in N$

Résultats: Q = [k]P

$R_0 \longleftarrow$ □; $R_1 \longleftarrow$ P

pour i = n - 1 **à** 0 **faire**

b $\longleftarrow k_i$; $R_{1-b} \longleftarrow R_{1-b} + R_b$

$R_b \longleftarrow 2R_b$

fait

retourner R_0

L'échelle de Montgomery a été initialement proposée comme un algorithme de multiplication scalaire pour une sorte de courbes elliptiques avec une arithmétique ponctuelle très efficace : les courbes dites de Montgomery [13].

La proposition de Montgomery atteint également une accélération supplémentaire en calculant uniquement les coordonnées (X, Z) des points intermédiaires. Ceci est rendu possible car l'échelle de Montgomery implique une soi-disant différentielle addition qui calcule la somme de deux points dont la différence est connue. Cette approche était ensuite généralisée à des courbes elliptiques dans la forme de Weierstrass [1, 20,34]. En particulier, il a été montré dans [34] qu'une itération en boucle de l'échelle de Montgomery en utilisant (X ; Z)-coordonnées seulement peut être effectuée en 11M + 4S + $2M_a$ + 18A où M indique le coût de la multiplication par le paramètre de courbe a (qui est quelques ajouts si a est petit, par exemple a = -3).

L'échelle de Montgomery fournit également des multiplications scalaires régulières efficaces basé sur des formules d'addition co-Z. Dans [38], Venelli et Dassande ont proposé plusieurs variantes de l'échelle de Montgomery basée sur l'arithmétique co-Z. La variante la plus efficace est basée sur les ajouts de points co-Z (X, Y) -only et il atteint un coût de calcul de 9M + 5S (plus plusieurs additions) par bit du scalaire. Indépendamment de ce travail, Goundar et al ont appliqué l'arithmétique co-Z à l'échelle de Montgomery dans [4]. Les algorithmes résultants impliquent 11M + 5S + 23A par bit scalaire, qui peut être réduit à 9M + 7S + 27A en utilisant des astuces standard.

Récemment, Hutter et al ont proposés des algorithmes d'échelle de Montgomery avec co-Z projectif (X; Z) arithmétique dans [10]. Leur variante la plus rapide implique 10M + 5S + 13A par bit scalaire.

III.2.3.3 Algorithmes binaires réguliers de l'arithmétique jacobienne (X ; Y) -only Co Z

Deux algorithmes binaires réguliers à partir de l'arithmétique jacobienne (X, Y) -only Co-Z. Les algorithmes III.9 et III.10 donnent les versions (X, Y) -only de l'addition co-Z avec update et le co-Z addition de conjugué. L'algorithme dit XYCZ-ADD prend les coordonnées (X, Y) de deux co-Z points P et Q et calcule les coordonnées (X, Y) de P + Q et les coordonnées de mise à jour (X, Y) de P (c'est-à-dire tel que P et P + Q sont co-Z). D'autre part, l'algorithme XYCZ-ADDC calcule les (X ; Y) -coordonnées de P + Q et de son co-Z conjugué P - Q. Ils montrent que l'addition avec update peut être calculée au coût de 4M + 2S + 7A. Pour l'addition du conjugué, un compromis temps-mémoire est possible. Une première implémentation implique 5M + 3S + 11A, tandis que le second a un coût de 5M + 3S + 16A.

Algorithme III.6: (X, Y)-only co-Z addition with update – XYCZ-ADD

Données : (X_1, Y_1) et (X_2, Y_2) s.t. P = (X1 : Y1 : Z)
et Q = (X2: Y2: Z) pour même $Z \in F_p$, $P; Q \in E(F_p)$
Résultats: (X_3, Y_3) et (X_1', Y_1') s.t. P = $(X_1' : Y_1' : Z3)$
Et P + Q = $(X_3: Y_3: Z_3)$ pour même $Z_3 \in F_p$
$A \leftarrow (X_2 - X_1)^2$
$B \leftarrow X_1A$
$C \leftarrow X_2A$
$D \leftarrow (Y_2 - Y_1)^2$
$E \leftarrow Y_1(C - B)$
$X_3 \leftarrow D - (B + C)$
$Y_3 \leftarrow (Y_2 - Y_1)(B - X_3) - E$
$X_1' \leftarrow B$
$Y_1' \leftarrow E$
Retourner $((X_3, Y_3), (X_1', Y_1'))$

Algorithme III.7: (X, Y)-only co-Z conjugate addition – XYCZ-ADDC

Données : (X_1, Y_1) and (X_2, Y_2) s.t. $P = (X_1 : Y_1 : Z)$
et $Q = (X_2 : Y_2 : Z)$ pour même $Z \in F_p$, $P; Q \in E(F_p)$
résultats : (X_3, Y_3) and $(X_3' ; Y_3')$ s.t. $P + Q = (X_3 : Y_3 : Z_3)$ et $P - Q = (X_3' : Y_3' : Z_3)$
pour même $Z_3 \in Fp$
$A \leftarrow (X_2 - X_1)^2$
$B \leftarrow X_1 A$
$C \leftarrow X_2 A$
$D \leftarrow (Y_2 - Y_1)^2$; $F \leftarrow (Y_1 + Y_2)^2$
$E \leftarrow Y_1(C - B)$
$X_3 \leftarrow D - (B + C)$
$Y_3 \leftarrow (Y_2 - Y_1)(B - X_3) - E$
$X_3' \leftarrow F - (B + C)$
$Y_3' \leftarrow (Y_1 + Y_2)(X_3' - B) - E$
Retourner $((X_3, Y_3), (X_3', Y_3'))$

Dans [4] l'algorithme co-Z double and add est proposé :

Algorithme III.8: (X, Y)-only co-Z doubling-addition with update – XYCZ-DA

Données: (X_1, Y_1) et (X_2, Y_2) s.t. $P = (X_1 : Y_1 : Z)$ et $Q = (X_2 : Y_2 : Z)$ pour même $Z \in F_p$, $P; Q \in E(F_p)$

Résultats: (X_4, Y_4) et (X_4', Y_4') s.t. $2P + Q = (X_4 : Y_4 : Z_4)$ et $Q = (X_4'; Y_4' : Z4)$ pour même $Z_4 \in Fp$

$((X_3, Y_3), (X_1', Y_1')) \leftarrow$ XYCZ-ADD $((X_1, Y_1), (X_2, Y_2))$

$((X_4, Y_4), (X_4', Y_4')) \leftarrow$ XYCZ-ADDC $((X_3, Y_3), (X_1', Y_1'))$

Retourner $((X_4, Y_4) ; (X_4', Y_4'))$

L'échelle de Montgomery (X,Y)-only co-Z pour le calcul de la multiplication est décrite dans l'algorithme suivant :

Algorithme III.9 : Montgomery ladder with (X; Y)-only co-Z addition

Données: P Є E(Fq), k = $(k_{n-1};\ldots; k_1; k_0)_2$ Є N avec $k_{n-1} = 1$
Résultats: Q = [k]P
$(R_1;R_0)$ ← XYCZ-IDBL(P)
Pour i = n - 2 **à** 1 **fait**
b← ki
$(R_{1-b}; R_b)$ XYCZ-ADDC $(R_b;R_{1-b})$
$(R_b; R_{1-b})$ XYCZ-ADD $(R_{1-b};R_b)$
faire
b ← k_0
$(R_{1-b}; R_b)$ XYCZ-ADDC $(R_b;R_{1-b})$
λ ← FinalInvZ (R0;R1;P; b)
$(R_b; R_{1-b})$ ← XYCZ-ADD $(R_{1-b};R_b)$
Retourner $(X_0\lambda^2; Y_0\lambda^3)$

L'algorithme (III.9) implique (n - 2) calculs de XYCZ-ADDC, (n - 2) calculs XYCZ-ADD, le calcul initial XYCZ-IDBL, le calcul FinalInvZ et 3M + 1S pour obtenir les coordonnées affines du résultat**.**

Algorithme III.10: Montgomery ladder with (X; Y)-only co-Z double addition

Données: P Є E(Fq), k = $(k_{n-1};\ldots; k_1; k_0)_2$ Є N avec $k_{n-1} = 1$
Résultats: Q = [k]P
$(R_1;R_0)$ ← XYCZ-IDBL(P)
b ← k_{n-2}
$(R_{1-b}; R_b)$ ← XYCZ-ADDC $(R_b;R_{1-b})$
Pour i = n - 2 **à** 1 **faire**
b← ki ; d ← k_{i-1} ; s ← d⊕b
$(R_{1-b}; R_b)$ ← XYCZ-DA $(R_{1-b}; R_b)$
R_d ← $(-1)^s R_d$
fait
b ← k_0
λ ← FinalInvZ (R0;R1;P; b)
$(R_b; R_{1-b})$ ← XYCZ-ADD $(R_{1-b};R_b)$
Retourner $(X_0\lambda^2; Y_0\lambda^3)$

L'algorithme (III.10) implique (n - 2) calculs de XYCZ-DA, (n - 2) inversion conditionnelle des points, 1 calcul de XYCZ-ADD, 1 calcul de XYCZ-ADDC, le calcul initial XYCZ-IDBL, le FinalInvZ et 3M + 1S pour obtenir les coordonnées affines du résultat.

II.2.4 Calcul parallèle

Les processeurs modernes sont constitués de plusieurs cœurs. En effet, il y a en général deux ou quatre cœurs physiques dans les processeurs généralistes que l'on trouve dans nos ordinateurs de bureaux ou portables, tels que ceux produits par Intel ou AMD, mais aussi dans la plupart des plates-formes mobiles (téléphones ou tablettes). Ce nombre de cœurs va probablement croître dans le futur. Tirer avantage de cette multiplication des cœurs est un défi à relever qui se formule de la façon suivante : comment améliorer les performances et/ou la résistance aux attaques en parallélisant les implantations logicielles ? Dans le domaine de la cryptographie, il est assez naturel d'effectuer plusieurs tâches en parallèle, par exemple, le chiffrement de plusieurs messages. En revanche, le défi à relever est plus difficile en ce qui concerne les opérations internes aux protocoles, la multiplication scalaire de points de courbe elliptique en particulier. Il semble intéressant de paralléliser ces calculs si cela permet de réduire la latence de ces opérations.

Par exemple dans un réseau de capteurs sans fil, l'opération de la multiplication scalaire peut être réalisée en collaboration de plusieurs nœuds voisins où chacun effectue une tâche d'une manière indépendante et simultanée.

III.3 Comparaison de performances

Le tableau suivant présente les couts d'exécution des algorithmes vu précédemment.

Opérations / Méthodes	Addition	Doublement	Addition-doublement
Coordonnées affines	2M+1S+3R+1I	2M+2S+4R+1I	/
Coordonnées jacobiennes	12M+4S+7A	4M+6S+8A	/
Coordonnées mixées Jacobiennnes-affines	8M+3S+7A	/	11M+7S+27A
(X,Y)-only co-Z	4M+2S+7A With update	/	8M+6S+26A
	5M+3S+11A Conjugate		

Tableau III.3: coûts d'exécution d'algorithmes

III.4 Conclusion

Dans ce chapitre nous avons présenté les algorithmes qui permettent d'accélérer la multiplication d'un scalaire par un point car cette opération est la plus gourmande en termes de temps de calcul et elle est impliquée dans le chiffrement.

Dans le chapitre qui suit nous allons présenter une solution d'optimisation de calculs dans les échanges entre les nœuds d'un réseau de capteurs sans fil

CHAPITRE IV

IMPLEMENTATION ET CONTRIBUTION

IV.1 Introduction

Dans le chapitre précédent, nous avons présenté la cryptographie sur les courbes elliptiques, ainsi que l'ensemble de techniques qui nous permettent d'accélérer le calcul des multiplications scalaires, qui est considérée comme l'opération la plus importante et coûteuse sur les courbes elliptiques.

La partie implémentation de notre projet consiste à présenter notre contribution pour réduire le temps de calcul et optimiser l'énergie des capteurs en se basant principalement sur une approche distribuée ; pour enfin faire une évaluation de notre solution et établir une comparaison de performance en prenant en considération le temps de calcul de chaque algorithme et l'énergie consommée.

IV.2 Présentation de la solution :

Partons du principe que l'algorithme standard de calcul de produit scalaire (Algorithme II.1 : Algorithme doublement-et-addition right to left) permet le parallélisme, notre solution consiste à distribuer les calculs de doublements et additions de points entre les nœuds d'un même cluster et cela en désignant un nœud qu'on appellera nœud calculateur à l'aide d'un algorithme de sélection de nœud qui a le plus d'énergie.

Ce nœud aura le rôle de calculer les doublements d'un même point P et stocker les résultats dans un tableau de taille n puis envoyer ce dernier aux autres nœuds du cluster qui vont utiliser ces résultats pour calculer les additions. Les algorithmes suivants permettent d'illustrer les taches de chaque nœud :

- Le nœud calculateur des doublements exécutera l'algorithme suivant :

Algorithme IV.1 : doublement

```
Données : P ;
Résultats: D [];
Début
    D [0] = P;
 Pour i=1 à d-1 faire
    D[i] = 2*D[i-1];
 fait
fin
```

- Les autres nœuds exécuteront l'algorithme suivant :

Algorithme IV.2: addition

Données: P, D[ ];

Résultats: Q= KP;

```
Début
  pour i=0 à d-1 faire
      si (di=1) alors
          Q= Q+ D[i];
      Fin si
  fin
fin
```

Le schéma ci-dessous représente l'implémentation de notre solution, nous avons déployé la distribution des calculs entres les nœuds A, B et le nœud calculateur en utilisant le Protocol ECDH pour l'échange de clé C entres A et B.

Le calcul de m (message secret) dans le nœud A, E (E = C + m , C clé partagée) dans le nœud B et l'étape 4 ne sont pas considérés et concernent seulement le protocole ECIES. Leur coût en termes de calcul et d'énergie sont négligeables.

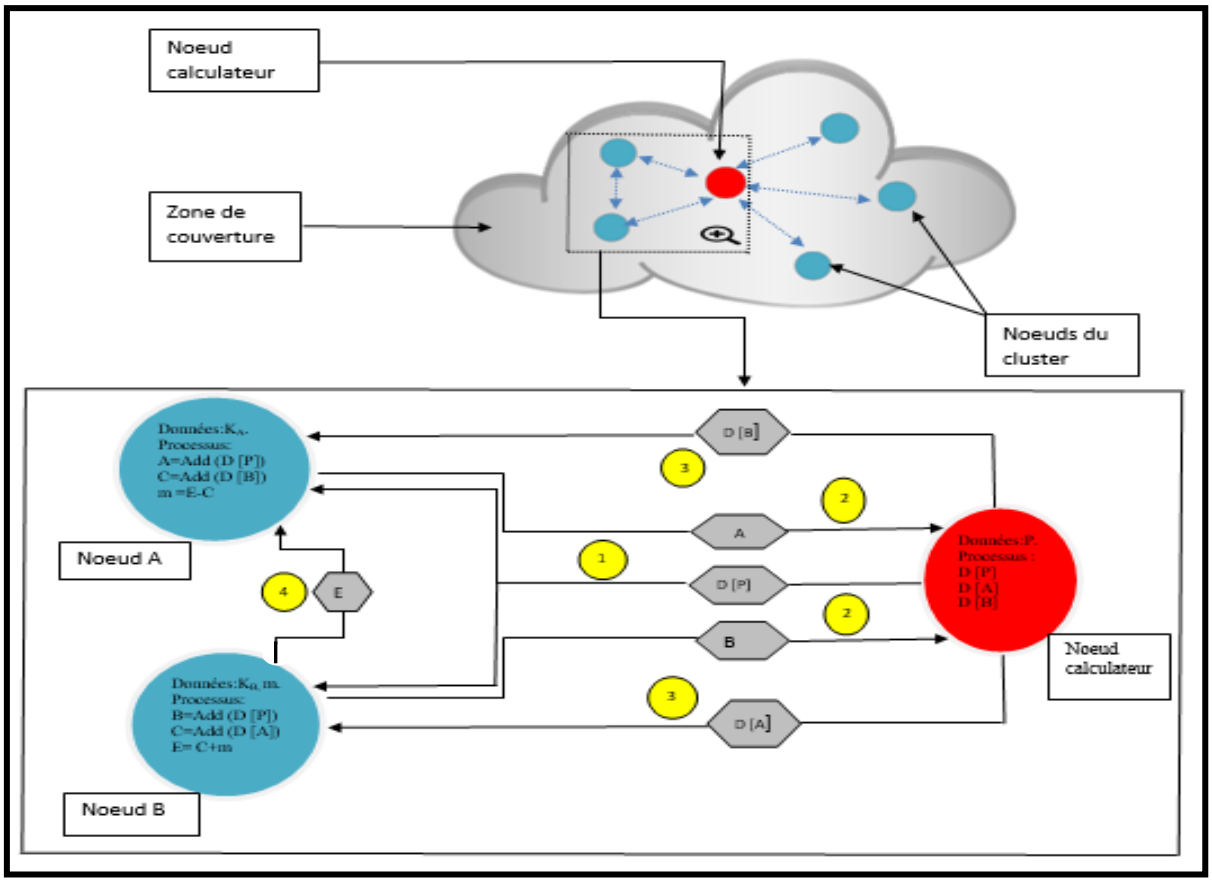

Figure IV.1: processus d'exécution de notre solution

IV.3 Environnement et choix du matériel

IV.3.1 Arduino Uno

Nous avons choisi de travailler avec la carte Arduino UNO R3, ses caractéristiques sont proches de celles des capteurs. La carte Arduino Uno est basée sur un ATMega328 cadencé à 16 MHz. Elle peut se programmer avec le logiciel Arduino. Le contrôleur ATMega328 contient un bootloader qui permet de modifier le programme sans passer par un programmateur [40]. Le logiciel est téléchargeable gratuitement sur [41].

La figure suivante représente les composants de la carte arduino :

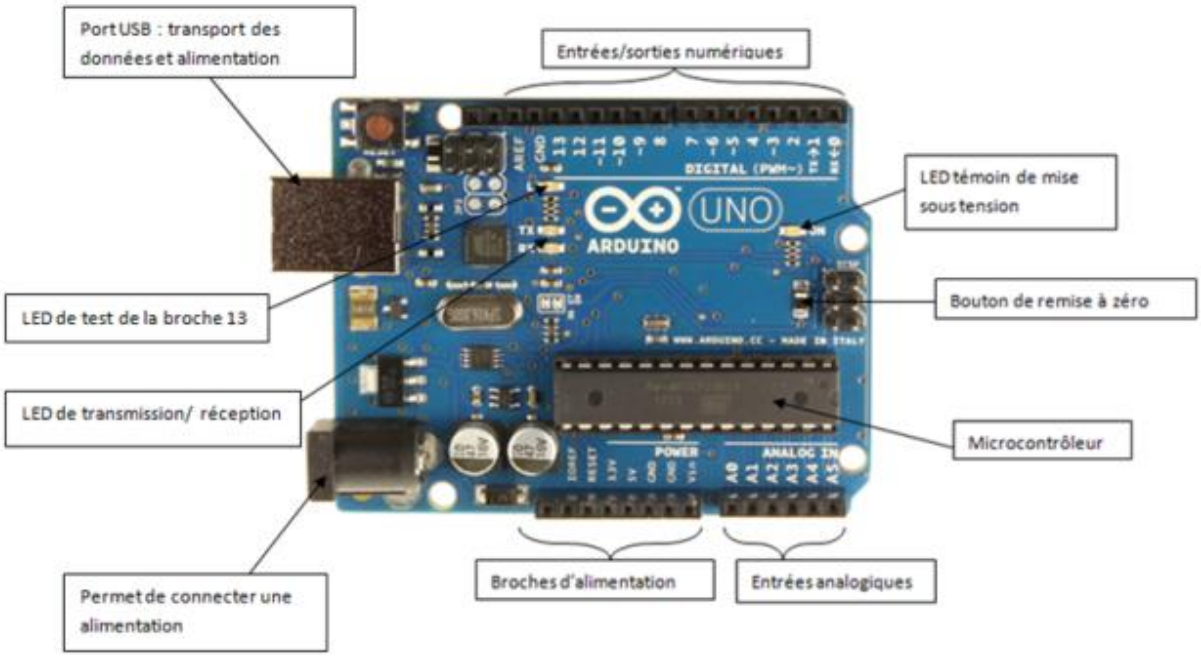

Figure IV.2: architecture d'une carte Arduino Uno

- **Caractéristiques principales:** Arduino est caractérisée par :

-Version: Rev. 3

-Alimentation :

- via port USB
- 7 à 12 V sur connecteur alim

- microprocessor: ATMega328

-mémoire flash: 32 kB

-mémoire SRAM: 2 kB

-mémoire EEPROM: 1 kB

-14 broches d'E/S dont 6 PWM

-6 entrées analogiques 10 bits

-Intensité par E/S: 40 mA

-Cadencement: 16 MHz

-Bus série, I2C et SPI

-Gestion des interruptions

-Fiche USB B

-Dimensions: 74 x 53 x 15 mm.

Après l'installation du l'IDE Arduino, nous avons passé à l'installation d'une bibliothèque qui nous permet d'implémenter les algorithmes et les fonctions dont on aura besoin pour effectuer notre travail.

IV.3.2 La bibliothèque micro-ecc (µECC)

Il existe des librairies fournissant des opérations PKC basées sur ECC qui peuvent être configurées de manière flexible et intégrées dans des applications de réseau de capteurs. Parmi ces bibliothèques on trouve :

- **TinyECC** utilisé dans l'environnement TinyOS et qui prend en charge les capteurs MICA2 / MICAz, TelosB / Tmote Sky, BSNV3 et Imote2 et les paramètres de domaine de courbe elliptique de 128 bits, 160 bits et 192 bits recommandés par SECG.

Pour d'autres environnements comme l'arduino on trouve des librairies comme :

- **nano-ecc** qui est une très petite implémentation ECDH et ECDSA pour les microcontrôleurs 8 bits. Elle est basée sur la micro-ecc qu'on utilisera dans notre travail.
- **Micro-ecc** est une implémentation d'ECDH et ECDSA, petite et rapide pour les processeurs 8 bits, 32 bits et 64 bits. La version statique de micro-ecc (c'est-à-dire où la courbe a été sélectionnée au moment de la compilation) se trouve dans la branche "statique"[42]

➢ **Caractéristiques :**

- Résistant aux attaques latérales connues.
- Écrit en C, avec un assemblage en ligne GCC en option pour les plates-formes AVR, ARM et Thumb.
- Prend en charge les architectures 8, 32 et 64 bits.
- Petite taille de code.
- Aucune allocation de mémoire dynamique.
- Prise en charge de 5 courbes standard : secp160r1, secp192r1, secp224r1, secp256r1 et secp256k1.
- Licence BSD à 2 clauses.

IV.3.3 ZigBee

ZigBee est un LP-WPAN (Low Power – Wireless Personal Area Network) : c'est un réseau sans fil à bas débit et à courte portée qui utilise les ondes hertziennes pour transporter des messages entre deux ou plusieurs entités réseaux. Il est caractérisé par une portée comprise entre quelques mètres et quelques centaines de mètres et un débit faible (maximum 250 kbits/s). La différence entre ZigBee et la plupart des autres réseaux locaux et personnels sans fil (WiFi, Bluetooth) se situe au niveau de l'utilisation du médium hertzien : ZigBee est optimisé pour une faible utilisation du médium partagé par tous, par exemple 0,1 % du temps [14]. Typiquement, un module ZigBee occupera le médium pendant quelques millisecondes en émission, attendra éventuellement une réponse ou un acquittement, puis se mettra en veille pendant une longue période avant l'émission suivante, qui aura lieu à un instant prédéterminé. ZigBee est conçu pour interconnecter des unités embarquées autonomes comme des capteurs/actionneurs, à des unités de contrôle ou de commande. De telles entités embarquées peuvent dès lors être alimentées pendant plusieurs mois par des piles classiques [31].

- **Caractéristiques :**

Voici en résumé les valeurs typiques caractérisant IEEE 802.15.4 et ZigBee :

– Débit : 20 kbits/s 868 MHz.

– Puissance d'émission typique : entre 0 et 3 dBm.

– Portée radio : quelques centaines de mètres en espace libre.

– Consommation du composant d'émission / réception (hors traitement CPU) :

- 3 μA en hibernation (hibernate mode),
- 40 μA en somnolence (doze mode),
- 1 mA au repos (idle mode),
- 30 mA en émission,
- 40 mA en réception.

– Taille de la pile protocolaire (code + mémoire) :

- inférieure à 20 ko pour une entité réduite (RFD),
- entre 40 à 60 ko pour une pile complète (FFD).

– Nombre d'entités connectables au réseau :

- 28 dans une étoile,
- 216 dans un PAN maillé,

• 264 adresses MAC disponibles.

– Accès au médium : pur CSMA/CA (sans RTS/CTS) ou organisé (Mode balisé avec slots dédiés).

– Détection / correction d'erreurs : CRC 16 bits dans la trame MAC.

IV.4 Expérimentation

Nous avons procédé comme suit:

1. Récupérer le temps d'exécution des algorithmes cités en haut (dans le système de coordonnées jacobiennes) grâce à la fonction **micros ()**.

2. Calculer l'énergie consommée par le capteur en utilisant la formule E = V * I * T ou V = 5v, I = 15.15 mA, T variable (le temps d'exécution).

3. Calculer l'énergie de réception des résultats de l'algorithme IV.1.

4. Comparer les résultats avec l'algorithme standard de produit scalaire (II.1) et deux autres algorithmes optimisés existants qui sont l'algorithme III.10 (Montgomery ladder with (X ;Y)only co-Z addition) et l'algorithme III.12 (Montgomery ladder with (X ;Y)only co-Z double addition) .

Le tableau ci-dessous, représente les résultats (temps d'exécution) de différentes opérations avec des clés de 192 bits et 224 bits :

Operation	192	224
J-Doublement	5.572	7.432
J-Addition	10.292	13.628
XYCZ-IDBL	9.712	12.952
XYCZ-ADDC	4.976	6.596
XYCZ-ADD	3.720	4.916
FinalInvZ	20.128	26.216
XYCZ-DA	11.252	14.884

Tableau IV.1: Temps d'exécution des opérations (en ms et mJ)

Selon [32], l'énergie consommée pour recevoir un point Q (X: Y: Z) en coordonnées jacobiennes est 1,064 mJ. Recevoir un tableau sur n points nécessite 4n octets avec 3n

octets comme taille de point et n octets en tant que données de tableau (n est la taille de la clé privée k). Pour n = 192, la taille du tableau est de 768 octets nécessitant environ 7 paquets de données. Le coût total pour recevoir le tableau de données est Rx = 50,176 mJ.

Le tableau ci-après indique le temps et l'énergie consommés pour calculer la multiplication scalaire d'un point (kP) en coordonnées jacobiennes.

Opération	**Coût**	**192**	**224**
J-Doublement	Temps	1068.82	1664.76
	Energie	80.96	126.165
J-Addition	Temps	988.03	1526.34
	Energie	74.843	115.62
Rx	Temps	358	409
	Energie	50.1	57.2

Tableau IV.2: Temps et énergie pour kP (en ms et mJ)

L'algorithme standard pour calculer kP (Algorithme II.1) effectue en moyenne n doublements et n/2 additions par contre la solution que nous avons proposée effectue 0 doublement et n/2 additions (tableau IV.3)

Opération	**Notre solution**	**Algorithme II.1**
J-Doublement	0	n
J-Addition	n/2	

Tableau IV.3: Nombre d'opérations de doublement et d'addition de point

Le tableau suivant représente une comparaison en temps et en énergie entre les algorithmes cités en haut et notre solution:

	Coût	**192**	**224**
Notre solution	Temps	1346.03	1935.34
	Energie	124.943	172.82
Algorithm double and add right to left	Temps	2056.85	3191.1
	Energie	155.803	241.785
Montgomery ladder with (X ;Y)only co-Z addition	Temps	1690.77	2606.34
	Energie	128.07	197.40
Montgomery ladder with (X ;Y)only co-Z double addition	Temps	3354.92	2176.41
	Energie	164.86	254.13

Tableau IV.4: temps et énergie pour kP de différents algorithmes (en ms et mj)

Ces représentations graphiques illustrent les résultats du tableau IV.4 :

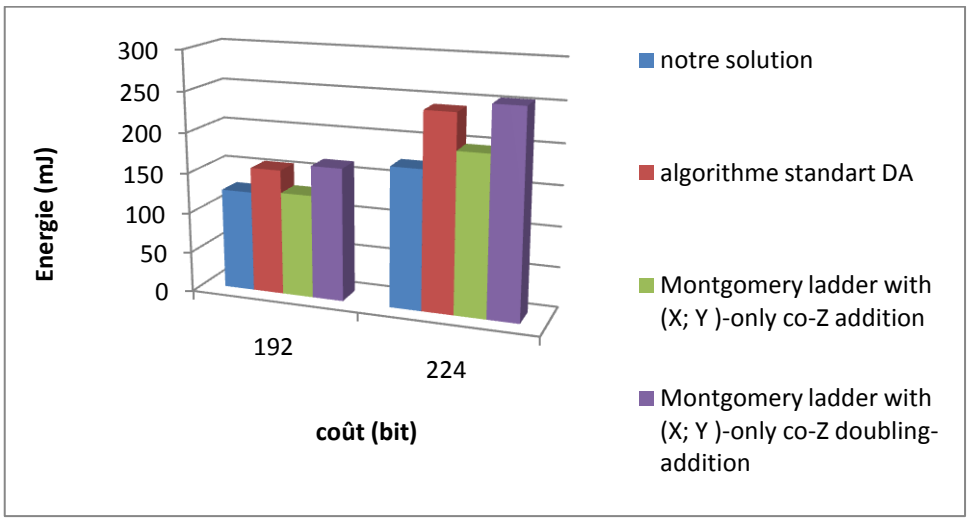

Figure IV.3 : comparaison de notre solution et les autres algorithmes en termes d'énergie.

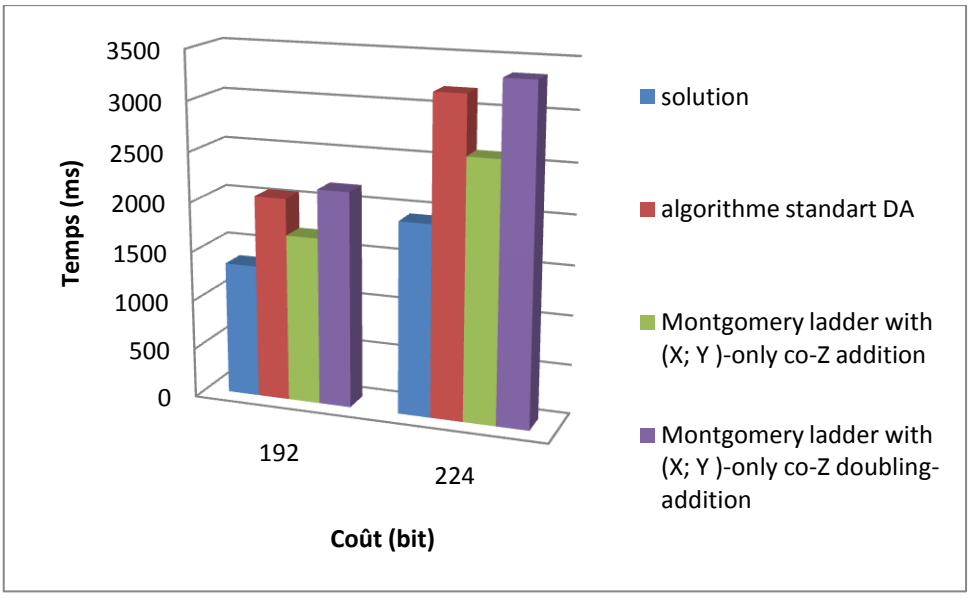

Figure IV.4 : comparaison de notre solution et les autres algorithmes en termes de temps d'exécution.

Les résultats obtenus montrent que le temps de calcul et l'énergie consommée sont réduits à 35% et à 19.8% respectivement dans notre solution par rapport à l'algorithme standard DA, 20.38% et 2.44% par rapport à l'algorithme Montgomery ladder with (X ;Y) only co-Z addition et 59.87% et 24.21% par rapport à l'algorithme Montgomery ladder with (X ;Y)only co-Z double addition.

IV. 5 Conclusion

Dans ce chapitre, nous avons présenté et testé notre solution et d'après les résultats obtenus nous pouvons déduire que notre contribution a bien atteint l'objectif initial qui s'agit de réduire le temps de calculs et d'optimiser l'énergie.

Conclusion générale et perspectives

Dans notre travail, nous nous sommes intéressées à la problématique de la conservation d'énergie dans un réseau de capteurs. Dans ce dernier, les nœuds capteurs sont alimentés par des batteries à faible capacité, généralement irremplaçables car les nœuds capteurs sont déployés dans des zones difficilement accessibles.

Afin de prolonger la durée de vie du réseau de capteurs par la minimisation de la consommation d'énergie, plusieurs solutions ont été proposées dont la majorité essayent d'éviter les différentes causes de perte d'énergie. Généralement, ces solutions ne sont pas suffisamment optimales ce qui laisse l'énergie dans les réseaux de capteurs un problème de recherche ouvert.

Notre contribution dans ce mémoire consiste à proposer une solution d'optimisation des calculs cryptographiques sur les courbes elliptiques en se basant sur une approche distribuée, son idée clé est de partager les calculs de la multiplication scalaire, qui est considérée comme l'opération la plus gourmande en termes de calcul et d'énergie, sur les nœuds d'un même cluster et cela en désignant un nœud calculateur qui aura le rôle de calculer les doublements et envoyer les résultats aux autres nœuds pour calculer les additions .

Les résultats de la simulation montrent que notre solution est efficace en termes de temps de calculs et de consommation d'énergie ce qui prolonge la durée de vie du capteur et celle du réseau en général.

Par ailleurs, dans notre implémentation actuelle, nous avons déjà appliqué une des méthodes mathématiques permettant d'améliorer la performance de la multiplication scalaire, qui est les coordonnées jacobiennes.

Cependant, nous n'avons pas encore découvert tous les aspects des courbes elliptiques, il reste encore d'autres techniques à voir et tester, par exemple la courbe Montgomery, la méthode de la fenêtre etc. Notre technique de distribution est basée sur le partage des calculs de doublements et additions. Il sera plus intéressant de combiner l'approche distribuée avec d'autres techniques, et de tester leur performance.

BIBLIOGRAPHIE

[01] Eric Brier and Marc Joye. Weierstraß elliptic curves and side-channel attacks. In David Naccache and Pascal Paillier, editors, Public Key Cryptography – PKC 2002, volume 2274 of Lecture Notes in Computer Science, pages 335–345. Springer, 2002.

[02] Michael Brown,DarrelHankerson, Julio López, andAlfred Menezes. Software Implementation of the NIST Elliptic Curves Over Prime Fields. In Topics in Cryptology - CT-RSA, The Cryptographer's Track at RSA Conference, San Francisco, CA, USA, April 8-12, Proceedings, pages 250–265, 2001.

[03] Ross Anderson, Haowen Chan, and Adrian Perrig. Key infection : Smart trust for smartdust. In Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on, pages 206–215. IEEE, 2004.

[04] Raveen R. Goundar, Marc Joye, and Atsuko Miyaji. Co-Z Addition Formulæ and Binary Ladders on Elliptic Curves - (Extended Abstract). In Stefan Mangard and Franc¸ois-Xavier Standaert, editors, Cryptographic Hardware and Embedded Systems, 12th InternationalWorkshop – CHES 2010, volume 6225 of Lecture Notes in Computer Science, pages 65–79. Springer, 2010.

[05] Darrel Hankerson, Alfred Menezes, and Scott Vanstone. Guide to Elliptic Curve Cryptography. Springer-Verlag New York Inc, 2004.

[06] Darrel Hankerson, Julio López Hernandez, and Alfred Menezes. Software Implementation of Elliptic Curve Cryptography over Binary Fields. In Cryptographic Hardware and Embedded Systems - CHES, volume 1965 of LNCS, pages 1–24. Springer, 2000.

[07] Joseph H. Silverman Jeffrey Hoffstein, Jill Pipher. An Introduction to Mathematical Cryptography. Springer Science+Business Media, LLC. Springer, 2008.add and double

[08] P. Kocher, J. Jaffe, and B. Jun. Differential Power Analysis. In Michael J. Wiener, editor, Advances in Cryptology, 19th Annual International Cryptology Conference – CRYPTO '99, volume 1666 of Lecture Notes in Computer Science, pages 388–397. Springer, 1999.

[09] D. Agrawal, B. Archambeault, J.R. Rao, and P. Rohatgi. The EM Side-Channel(s). In Burton S. Kaliski Jr., C¸ etin Kaya Koc¸, and Christof Paar, editors, Cryptographic Hardware and Embedded Systems, 4th International Workshop – CHES 2002, volume 2523 of Lecture Notes in Computer Science, pages 29–45. Springer, 2002.

[10] Michael Hutter, Marc Joye, and Yannick Sierra. Memory-Constrained Implementations of Elliptic Curve Cryptography in Co-Z Coordinate Representation. To appear in AFRICACRYPT 2011, 2011.

[11] Neal Koblitz. Elliptic curve cryptosystems. Mathematics of computation, 48(177) :203–209, 1987.

[12] Ronald L Rivest, Adi Shamir, and Len Adleman. A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2) :120–126, 1978.

[13] Peter L. Montgomery. Speeding the Pollard and Elliptic Curve Methods of Factorization. Mathematics of Computation, 48(177):243–264, 1987.

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

[15] Victor S Miller. Use of elliptic curves in cryptography. In Advances in Cryptology—CRYPTO'85 Proceedings, pages 417–426. Springer, 1986.

[16] Claude Castelluccia et Aurélien Francillon, "Protéger les réseaux de capteurs sans fil", 2008, dans:http://actes.sstic.org/SSTIC08/Proteger _Reseaux_Capteurs _Sans_Fi I/SSTIC08-article-Castelluccia _Francillon-Proteger _Reseaux_Capteurs _Sans _Fil.pdf.

[17] Jean-Jacques Quisquater and David Samyde. Eddy Current for Magnetic Analysis with Active Sensor. Presented at e-Smart 2002, 2002.

[18] Olivier Leran. Parallélisation d'un algorithme de chiffrement utilisant les courbes elliptiques. Master's thesis, Université de Franche-Comté, Décembre 2012.

[19] P. Szczechowiak, L. B. Oliveira, M. Scott, M. Collier, et R.Dahab, "NanoECC: Testing the Limits ofElliptic Curve Cryptography in Sensor Networks", European conference on Wireless Sensor Networks, LNCS, vol. 4913. pp. 305-320, Feb.2008.

[20] Tetsuya Izu and Tsuyoshi Takagi. A Fast Parallel Elliptic Curve Multiplication Resistant against Side Channel Attacks. In David Naccache and Pascal Paillier, editors, Public Key Cryptography –PKC 2002, volume 2274 of Lecture Notes in Computer Science, pages 280–296. Springer, 2002.

[21] Whitfield Diffie and Martin Hellman. New directions in cryptography. Information Theory, IEEE Transactions on, 22(6) :644–654, 1976.

[22] Benoît Chevallier-Mames, Mathieu Ciet, and Marc Joye. Low-Cost Solutions for Preventing Simple Side-Channel Analysis: Side-Channel Atomicity. IEEE Transactions on Computers, 53(6):760–768, 2004.

[23] Kefa Rabah. Elliptic curve elgamal encryption and signature schemes. Information Technology Journal, 4(3) :299–306, 2005.

[24] Certicom Research. Sec 2 : Recommended elliptic curve domain parameters, September 2000.

[25] Daniel R. L. Brown. Sec 1 : Elliptic curve cryptography, May 2009.

[26] Don Johnson, Alfred Menezes, and Scott Vanstone. The elliptic curve digital signature algorithm (ecdsa). International Journal of Information Security, 1(1) :36–63,2001.

[27] William M Daley and Raymond G Kammer. Digital signature standard (dss). Technical report, DTIC Document, 2000.

[28] Youssou Faye. Algorithmes d'authentification et de cryptographie efficaces pour les réseaux de capteurs sans fil. Autre [cs.OH]. Université de Franche-Comté, 2014. Français.

[29] Yacine Challal. Sécurité des réseaux de capteurs sans fil. Février 2015.

[30] ARAAR Chaouki, Compression d'images dans les réseaux de capteurs sans fil, Université Badji Mokhtar – ANNABA, 2015

[31] ZigBee thierry val,eric campo et adrien van den bossche, technologie ZugBee/802.15.2 protocoles,topologie et domaines d'application edition T.I

[32] Mohamed Ramdani, Mohamed Benmohammed, Nadjia Benblidia-Distributed solution of scalar multiplication on elliptic curves over Fp for resource-constrained networks. 2018 ICFNDS, June 26-27, Amman, Jordan, ACM ISBN 978-1-4503-6428-7.

[33] Marc Joye. Highly Regular Right-to-Left Algorithms for Scalar Multiplication. In Pascal Paillier1 and Ingrid Verbauwhede, editors, Cryptographic Hardware and Embedded Systems, 9th International Workshop – CHES 2007, volume 4727 of Lecture Notes in Computer Science, pages 135–147. Springer, 2007.

[34] Tetsuya Izu, Bodo M¨ oller, and Tsuyoshi Takagi. Improved elliptic curve multiplication methods resistant against side channel attacks. In Alfred Menezes and Palash Sarkar, editors, Progress in Cryptology, Third International Conference on Cryptology in India – INDOCRYPT 2002, volume 2551 of Lecture Notes in Computer Science, pages 296–313. Springer, 2002.

[35] Jean-S´ebastien Coron. Resistance against Differential Power Analysis for Elliptic Curve Cryptosystems. In C¸ etin Kaya Koc¸ and Christof Paar, editors, Cryptographic Hardware and Embedded Systems, First International Workshop – CHES'99, volume 1717 of Lecture Notes in Computer Science, pages 292– 302. Springer, 1999.

[36] Marc Joye and Christophe Tymen. Protections against Differential Analysis for Elliptic Curve Cryptography. In C¸ etin Kaya Koc¸, David Naccache, and Christof Paar, editors, Cryptographic Hardware and Embedded Systems, 3rd International Workshop - CHES 2001, volume 2162 of Lecture Notes in Computer Science, pages 377–390. Springer, 2001.

[37] Mathieu Ciet and Marc Joye. (Virtually) Free Randomization Techniques for Elliptic Curve Cryptography. In Sihan Qing, Dieter Gollmann, and Jianying Zhou, editors, Information and Communications Security, 5th International Conference – ICICS 2003, volume 2836 of Lecture Notes in Computer Science, pages 348–359. Springer, 2003.

[38] Alexandre Venelli and Franc¸ois Dassance. Faster Side-Channel Resistant Elliptic Curve Scalar Multiplication. In David Kohel and Robert Rolland, editors, Arithmetic, Geometry, Cryptography and Coding Theory 2009, volume 521 of Contemporary Mathematics, pages 29–40. American Mathematical Society, 2010.

[39] Nicolas Meloni. New Point Addition Formulae for ECC Applications. In Claude Carlet and Berk Sunar, editors, Arithmetic of Finite Fields, First International Workshop – WAIFI 2007, volume 4547 of Lecture Notes in Computer Science, pages 189–201. Springer, 2007.

[40] https://store.arduino.cc/arduino-uno-rev3

[41]https://www.arduino.cc/en/Main/Software

[42]Micro ECC https://github.com/kmakay