الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne Démocratique et Populaire

وزارة التعليم العالي والبحث العلمي
Ministère de L'enseignement Supérieur et de la Recherche Scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB DE BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الالكترونيك
Département D'électronique

Mémoire de Master

Filière : Télécommunications

Spécialité : Réseaux et Télécommunications

Présenté par : MEKNACI Fateh Nour-Essadet

Développement d'une Application de Signature Digitale Basée sur les Courbes Elliptiques

Proposé par :

Promoteur : DAHMANI Samir

Co- Promoteur : OUDJIDA Abdelkrim Kamel

Année Universitaire 2022-2023

Remerciements

Au Nom de la Paix.

Au Nom de la Science et de la Connaissance.

Au Nom de L'humanité.

Au Nom de L'amour et de la Fraternité.

J'offre Mes Salutations et Mes Remerciements à Ma Mère

Source de Lumière de Mon Chemin et de Ma Vie.

À Tous les Professeurs de L'université SAAD DAHLAB, BLIDA sans Exception. Qui se sont Occupés de mon Éducation et qui M'ont Encouragé à Poursuivre Mes Études.

Je Remercie Monsieur le Professeur DAHMANI Samir, pour M'avoir Accordé des Entretiens et Avoir Répondu à Mes Questions. Qui M'a Donné d'un Grand Soutien dans L'élaboration de Cette Mémoire.

Je Remercie Très Chaleureusement Mon Promoteur Professeur

OUDJIDA Abdelkrim Kamel

J'ai eu le Privilège de Travailler Parmi Votre Équipe et D'apprécier Vos Qualités et Vos Valeurs, Votre Sérieux et Votre Compétence.

Je Remercie Infiniment Mme NAIT ABDELSSELAM Fadila Pour Son Soutien et Son Précieux Conseils Tout Au Long de Mon Formation Au niveau du CDTA.

À Toute L'équipe de CDTA.

Sans Oublié Mon Frère Ainé.

Dédicace

Je Dédie Ce Modest Travail de Plusieurs Années de Réflexion

A Ma Très Chère Mère qui M'a Offert Depuis Toujours le Plus Belle Cadeau de L'univers.

A Toute Ma Famille Père Frère Sœur.

A Mon Cher Professeur OUDJIDA Abdelkrim Kamel

A Mme NAIT ABDELSSELAM Fadila

A Département D'électronique, Université Saad Dahlab BLIDA.

A L'Équipe : Binary Arithmetic for Numeric Applications (BANA) Au CDTA.

Abstract

The Radix 2^w reduction is a technique that speeds up modular multiplication operations in cryptography systems based on elliptic curves by eliminating a costly division operation. The electronic signature, or digital signature, is a method that ensures the authenticity and integrity of a digital document by guaranteeing that the signer cannot deny having signed it. The ECDSA digital signature protocol uses a key pair to sign and verify messages and is based on the mathematics of elliptic curves. Elliptic curves are an alternative to prime numbers in cryptography systems, offering equivalent security with smaller key sizes and faster computation. The problem of discrete algorithms is used in cryptography to ensure the security of encryption systems based on finites fields.

Keywords : Radix 2^w , ECDSA , Digital Signature Protocol , Elliptic Curves.

Résumé

La réduction de Radix 2^w est une technique qui accélère les opérations de multiplication modulaire dans les systèmes de cryptographie basés sur des courbes elliptiques en éliminant une opération de division coûteuse. La signature électronique, ou signature numérique, est une méthode qui garantit l'authenticité et l'intégrité d'un document numérique en garantissant que le signataire ne peut nier l'avoir signé. Le protocole de signature numérique ECDSA utilise une paire de clés pour signer et vérifier les messages et est basé sur les mathématiques des courbes elliptiques. Les courbes elliptiques sont une alternative aux nombres premiers dans les systèmes de cryptographie, offrant une sécurité équivalente avec des tailles de clé plus petites et un calcul plus rapide. Le problème des algorithmes discrets est utilisé en cryptographie pour assurer la sécurité des systèmes de cryptage basés sur des champs finis.

Mots-clés : Radix 2^w , ECDSA ,Le Protocole de Signature Numérique , Les Courbes Elliptiques.

ملخص

تخفيض راديكس هي تقنية تسرع عمليات الضرب المعيارية في أنظمة التشفير القائمة على المنحنى الإهليلجي من خلال القضاء على عملية التكلفة العالية للقسمة . التوقيع الإلكتروني ، أو التوقيع الرقمي ، هو طريقة تضمن صحة وسلامة المستند الرقمي من خلال ضمان عدم تمكن الموقع من إنكار توقيعه عليه. يستخدم بروتوكول التوقيع الرقمي اسدسا زوج مفاتيح للتوقيع والتحقق من الرسائل ويستند إلى الرياضيات من المنحنيات الإهليلجية. المنحنيات الإهليلجية هي بديل للأعداد الأولية في أنظمة التشفير ، مما يوفر أمانا مكافئا بأحجام مفاتيح أصغر وحساب أسرع. تستخدم مشكلة الخوارزميات المنفصلة في التشفير لضمان أمان أنظمة التشفير القائمة على الحقول المحدودة.

الكلمات المفتاحية : راديكس , اسدسا , بروتوكول التوقيع الرقمي , المنحنيات الاهليليجية .

Liste des Abréviations

ADD	**:**	**Addition**
AES	**:**	**Advanced Encryption Standard**
DA	**:**	**Doublement et Addition**
DBL	**:**	**Doubling**
DES	**:**	**Data Encryption Standard**
DPA	**:**	**Differential Power Analysis**
DPM	**:**	**Double Point Multiplication**
DLP	**:**	**Discrete Logarithm Problem**
DSP	**:**	**Digital Signal Processing**
ECC	**:**	**Elliptic Curve Cryptography**
ECDH	**:**	**Elliptic Curve Diffie-Hillman**
ECDSA	**:**	**Elliptic Curve Digital Signature Algorithm**
ECDLP	**:**	**Elliptic Curve Discrete Logarithm Problem**
ECDHP	**:**	**Elliptic Curve Diffie-Hillman Problem**
ECMSM	**:**	**Elliptic Curve Multi- Scalar Multiplication**
ECIES	**:**	**Elliptic Curve Integrated Encryption Scheme**
ECSM	**:**	**Elliptic Curve Scalar Multiplication**
FPGA	**:**	**Feld-Programmable Gate Array**
IAs	**:**	**Interleaved Algorithms**
NAF	**:**	**Non-Adjacent Form**
NIST	**:**	**National Institute of Standards and Technology**
MSAs	**:**	**Montgomery-Based SAs**
RSA	**:**	**Ronald Rivest, Adi Shamir et Leonard Adleman**
SAs	**:**	**Simultaneous Algorithms**
SCA	**:**	**Side-Channel Attacks**
SHA-256:		**Secure Hash Algorithm 256-bit**
SoC	**:**	**System on Chip**

SPA : Simple Power Analysis

PA : Point Addition

PD : Point Doubling

PM : Point Multiplication

TA : Timing Analysis

WSAs : Windowing SAs

Table des Matières

Liste des Figures

Liste des Tableaux

Introduction Générale

La Cryptographie est l'étude scientifique des techniques de communication sécurisée en présence d'adversaires. Elle concerne la conception et l'analyse des protocoles qui empêchent des tiers non autorisés de lire des messages privés[1]. Claude Shannon, considéré comme le père de la théorie de l'information, a établi les bases de la cryptographie moderne en proposant le concept de la sécurité parfaite ou secret[2]. Selon lui, la sécurité parfaite est atteinte lorsque la connaissance d'un message chiffré ne permet pas de déduire quoi que ce soit sur le message original. Shannon a également introduit le concept de la confusion et de la diffusion, qui sont deux propriétés fondamentales des algorithmes de chiffrement modernes. La confusion consiste à rendre le lien entre le texte en clair et le texte chiffré aussi complexe que possible, tandis que la diffusion consiste à répartir l'information à travers tout le texte chiffré pour rendre difficile la détection de la structure du texte en clair. la vision de la cryptographie de Shannon est centrée sur la sécurité des communications et la nécessité de développer des techniques de chiffrement robustes pour protéger les données confidentielles contre les adversaires.

La cryptographie à courbe elliptique est une branche de la cryptographie qui utilise les propriétés mathématiques des courbes elliptiques pour sécuriser les communications et protéger les données sensibles. Contrairement à d'autres méthodes de cryptographie qui utilisent des opérations sur des grands nombres premiers, la cryptographie à courbe elliptique exploite les propriétés algébriques des courbes elliptiques pour effectuer des calculs cryptographiques.

L'opération clé de la cryptographie à courbe elliptique est la multiplication scalaire, qui consiste à multiplier un point sur la courbe elliptique par un scalaire (un entier). Cette opération permet de générer des clés publiques et privées, de chiffrer et de déchiffrer des données, et de signer et de vérifier des messages.

La multiplication scalaire de la courbe elliptique k. P, où k est une constante non négative et P est un point de la courbe elliptique, nécessite deux opérations distinctes : l'addition (ADD) et le doublement (DBL). Pour réduire le nombre d'addition sans augmenter le nombre de doublement, un recodage de K avec moins de chiffres non nuls est nécessaire. Sur la base de l'arithmétique Radix-2^w, nous introduisons une méthode de fenêtrage à W bits de principe où les propriétés de vitesse, de mémoire et de sécurité sont décrites par des formules analytiques exactes comme preuve de supériorité. Contrairement aux algorithmes de fenêtrage existants, pour minimiser le nombre d'ajouts, la taille de la fenêtre (w) est guidée par un optimum dépendant de la longueur de bit (l) du scalaire k. Le nombre de précalculs requis est minimal en ce qui concerne la valeur de W. La

méthode proposée recode la chaîne binaire k et évalue la multiplication de gauche à droite et de droite à gauche, également. La méthode Radix-2^w est très facile à utiliser et hautement reconfigurable, permettant des compromis entre la mémoire et la vitesse et sécurité pour satisfaire différentes contraintes de cryptosystème. De plus, la méthode montre une grande résilience aux attaques par canaux secondaires basées sur la puissance, le temps d'execution et l'analyse statistique. Toutes les propriétés de Radix-2^w sont confrontées aux méthodes de fenêtrage standard grâce à une analyse approfondie des complexités [3] .en plus du recodage Radix-2^w, il existe d'autres techniques de recodage utilisées dans la cryptographie à courbe elliptique, telles que le recodage binaire et le recodage Non-Adjacent Form (NAF) et le recodage Montgomery Ladder. Ces techniques visent à optimiser les calculs de multiplication scalaire en réduisant le nombre d'opérations nécessaires.

la cryptographie à courbe elliptique et les opérations de multiplication scalaire, combinées avec des techniques de recodage telles que le recodage Radix-2^w, permettent de garantir la confidentialité, l'intégrité et l'authenticité des communications et des données dans les systèmes de sécurité modernes.

I. Généralités sur la Cryptographie

I.1 Préambule

La cryptographie est une discipline qui remonte à l'Antiquité, lorsque les humains ont commencé à chercher des moyens de protéger la confidentialité de leurs communications. Depuis lors, cette discipline a évolué pour devenir une science complexe qui étudie les techniques de chiffrement et de déchiffrement pour protéger les informations confidentielles [4].

I.2 Introduction à la Cryptographie

La cryptographie a pour objectif de rendre les messages incompréhensibles pour les personnes non autorisées à les lire [5], tout en les rendant accessibles à ceux qui en ont besoin. Elle repose sur l'utilisation de clés de chiffrement qui permettent de transformer les données originales en données chiffrées et inversement, Au fil du temps, les méthodes de cryptographie se sont améliorées et sont devenues plus sophistiquées. Les mathématiques ont été utilisées pour développer des techniques de cryptographie plus complexes, telles que le chiffrement à clé secrète et le chiffrement à clé publique, Le chiffrement à clé secrète utilise une clé secrète partagée entre les parties pour chiffrer et déchiffrer les messages. Le chiffrement à clé publique utilise une paire de clés, une publique et une privée, pour chiffrer et déchiffrer les messages. Les messages chiffrés à l'aide de la clé publique ne peuvent être déchiffrés qu'avec la clé privée correspondante [5].

I.3 La Cryptologie

La cryptologie est une discipline qui englobe deux domaines distincts : la cryptographie et la cryptanalyse. La cryptographie est la science de la conception de techniques de chiffrement pour protéger les informations confidentielles lors de leur transmission ou stockage. Elle peut être symétrique ou asymétrique. Dans le cas de la cryptographie symétrique, la même clé est utilisée pour chiffrer et déchiffrer les messages. Cela signifie que la clé doit être partagée entre les parties prenantes pour garantir la sécurité des communications. Dans le cas de la cryptographie asymétrique, deux clés sont utilisées : une clé publique et une clé privée. La clé publique est utilisée pour chiffrer les messages et peut être partagée sans compromettre la sécurité des données. La clé privée est utilisée pour déchiffrer les messages et doit être conservée secrète.

I.3.1 La Cryptographie

Les techniques de cryptographie ont été utilisées à des fins militaires et diplomatiques, ainsi que pour protéger les communications commerciales et privées. Les anciennes méthodes de

cryptographie comprenaient l'utilisation de codes secrets, tels que des substitutions de lettres ou des transpositions de mots, pour masquer le sens des messages. Elle est essentielle pour protéger les informations confidentielles et garantir la confidentialité, l'intégrité et l'authenticité des données [5].

I.3.2 La Cryptanalyse

Est la science de la recherche de faiblesses dans les techniques de chiffrement pour les exploiter et déchiffrer des messages protégés. Elle peut être utilisée à des fins légales, telles que la lutte contre la criminalité, ou illégales, telles que l'espionnage et le vol de données. Les cryptanalystes peuvent utiliser différentes méthodes pour briser les algorithmes de chiffrement, telles que l'analyse de fréquence, la cryptanalyse différentielle, la cryptanalyse linéaire et la cryptanalyse par force brute.

I.4 Types de Cryptographie

I.4.1 La Cryptographie Symétrique

La cryptographie symétrique est une méthode de cryptage dans laquelle une seule clé est utilisée pour à la fois chiffrer et déchiffrer des messages. Cette clé est partagée entre les parties qui souhaitent communiquer de manière sécurisée[1].

Lorsque deux parties veulent échanger des messages de manière sécurisée, elles doivent d'abord se mettre d'accord sur une clé secrète à utiliser pour le chiffrement et le déchiffrement des messages. Cette clé doit être gardée confidentielle pour empêcher les tiers de lire les messages.

Les algorithmes de chiffrement symétrique sont généralement plus rapides et plus simples que les algorithmes de chiffrement asymétrique. Cependant, la sécurité de la cryptographie symétrique dépend de la sécurité de la clé. Si la clé est compromise, tous les messages chiffrés avec cette clé peuvent être facilement déchiffrés. Par conséquent, la gestion des clés est un aspect crucial de la cryptographie symétrique.

Les exemples courants d'algorithmes de chiffrement symétrique incluent le DES (Data Encryption Standard), l'AES (Advanced Encryptions Standard), et le Blow Fish [6]. La cryptographie symétrique est souvent utilisée pour protéger les données stockées sur des disques durs, des clés USB et d'autres périphériques de stockage. Elle est également utilisée pour sécuriser les communications en temps réel telles que les appels téléphoniques, les conversations vidéo et les chats en ligne [7].

I.4.2 La Cryptographie Asymétrique

La cryptographie asymétrique est une technique de cryptographie qui utilise des

algorithmes mathématiques pour créer une paire de clés cryptographiques, l'une publique et l'autre privée, qui sont mathématiquement liées. Il existe plusieurs types d'algorithmes de cryptographie asymétrique, notamment l'algorithme RSA et la cryptographie à courbe elliptique [8].

L'algorithme RSA est l'un des algorithmes de cryptographie asymétrique les plus utilisés. Il utilise des nombres premiers très grands pour générer les clés publiques et privées, ce qui rend le processus de cryptage et de décryptage très difficile à inverser. La sécurité de l'algorithme RSA repose sur la difficulté mathématique de factoriser des grands nombres premiers en temps raisonnable [2].

La cryptographie à courbe elliptique est une autre forme de cryptographie asymétrique qui utilise des courbes elliptiques pour générer les clés publiques et privées [8]. Les clés générées par les courbes elliptiques sont plus courtes que les clés générées par l'algorithme RSA, ce qui les rend plus adaptées aux applications mobiles et aux ressources limitées. La cryptographie à courbe elliptique est également considérée comme plus sécurisée que l'algorithme RSA pour des longueurs de clé équivalentes, 163-bits key in EC équivalent 1024-bits key in RSA.

En utilisant la cryptographie asymétrique à courbe elliptique ou RSA, deux parties peuvent communiquer de manière sécurisée sans jamais divulguer leur clé privée. La clé publique peut être distribuée à tout le monde sans compromettre la sécurité, tandis que la clé privée doit être gardée secrète. Les données peuvent être chiffrées en utilisant la clé publique et ne peuvent être déchiffrées qu'avec la clé privée correspondante.

I.5 Les Algorithmes de Cryptographie

Les algorithmes de cryptographie sont des procédures mathématiques utilisées pour protéger les informations confidentielles en les rendant inintelligibles à toute personne qui n'a pas la clé de déchiffrement approprié

I.5.1 Rivest, Shamir et Adelman (RSA)

RSA est un algorithme de chiffrement asymétrique inventé en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman. Il repose sur le principe de la factorisation des nombres premiers pour assurer la sécurité des communications.

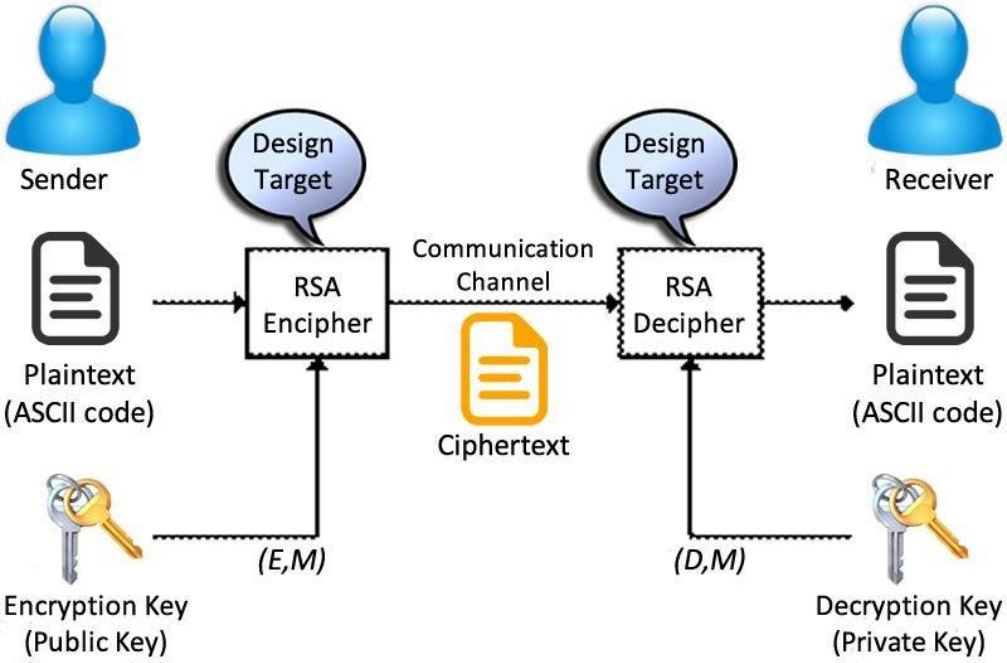


Figure 1:RSA Algorithme Structure [9]

Le chiffrement RSA est asymétrique : il utilise une paire de clés (des nombres entiers) composée d'une clé publique pour chiffrer et d'une clé privée pour déchiffrer des données confidentielles.

- Input: A message M to sign, the private key (p, q, d), with $p > q$, the pre-calculated values $d_p = d \bmod p - 1$, $d_q = d \bmod q - 1$, and $u = q^{-1} \bmod p$.
- Output: a valid signature S for the message M

1. Encode the message M in $m \in \mathbb{Z}_N$ (with PKCS#1)
2. Compute $s_p = m^{d_p} \bmod p$
3. Compute $s_q = m^{d_q} \bmod q$
4. Set $t = s_p - s_q$
5. If $t < 0$ then $t \leftarrow t + p$
6. Compute $S = s_q + ((t \cdot u) \bmod p) \cdot q$
7. Return S as a signature for the message M

Figure 2 : RSA Algorithme [10]

I.5.2 Data Encryption Standard (DES)

IBM a développé l'algorithme DES (Data Encryption Standard) dans les années 1970, et depuis, il est devenu la méthode préférée du Gouvernement Américain pour chiffrer les données sensibles et non classifiées [5]. Étant donné qu'il utilise un algorithme arithmétique de blocs, les données sont traitées en unités de taille fixe. Le DES utilise une clé de 56 bits, qui était considérée

comme sûre à l'origine, mais qui est aujourd'hui considérée comme assez faible et susceptible d'être attaquée par force brute.

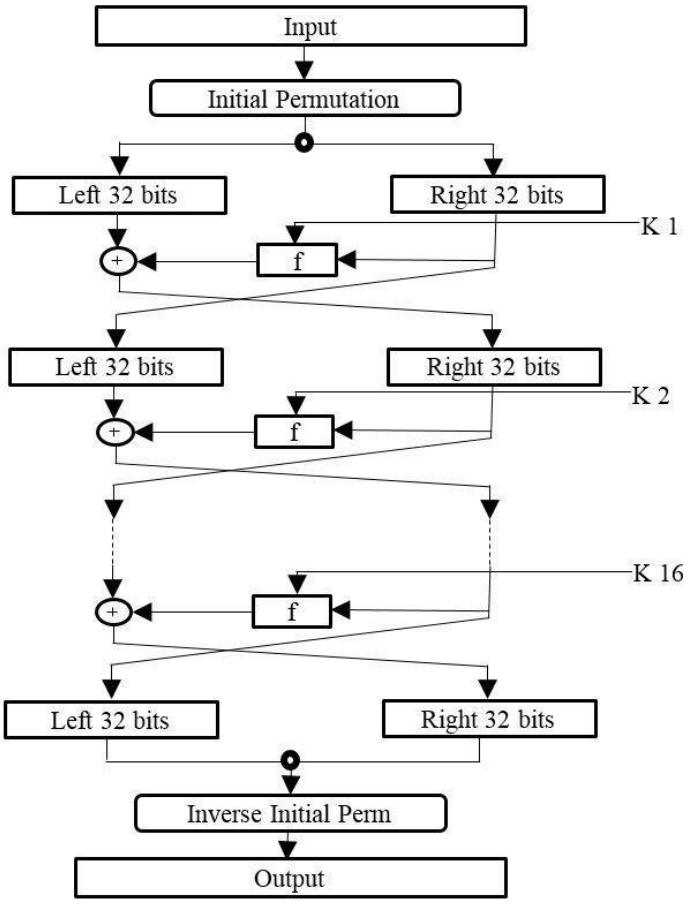


Figure 3 : DES Algorithme Structure [11]

Le processus de chiffrement DES implique une série d'opérations de substitution et de permutation, qui sont répétées 16 fois pour créer le texte de chiffrement final [5]. Le simple texte est divisé en blocs 64 bits, et chaque bloc est soumis à une permutation initiale (IP) qui réarrange les bits selon une table fixe. Le résultat est divisé en deux moitiés, et chaque moitié est soumise à une série d'opérations de substitution et de permutation qui dépendent d'une sous-clé générée à partir de la clé d'origine. Les deux moitiés sont ensuite échangées, et l'ensemble du processus est répété 16 fois en utilisant différentes sous-clés à chaque fois. Enfin, le texte de chiffrement résultant est soumis à une permutation finale (FP) qui renverse la permutation initiale [5].

Bien que le DES ait été largement utilisé pendant plusieurs décennies, sa taille de clé de 56 bits est maintenant considérée comme trop petite pour fournir une sécurité adéquate contre les attaques modernes. En réponse à cette faiblesse, le standard de cryptage avancé (AES) a été développé à la fin des années 1990 pour remplacer le DES. Cependant, le DES est encore parfois utilisé dans les systèmes anciens ou pour la compatibilité avec des logiciels plus anciens.

I.5.3 Advenced Encryption Standard AES

L'AES (Advanced Encryption Standard) est un algorithme de chiffrement symétrique développé par les cryptographes belges Joan Daemen et Vincent Rijmen. Il a été sélectionné par le National Institute of Standards and Technology (NIST) en 2001 comme norme de chiffrement [12] pour protéger les données sensibles et est devenu rapidement l'un des algorithmes de chiffrement les plus populaires et les plus fiables utilisés dans le monde entier.

L'AES utilise un algorithme de chiffrement par blocs, où les données sont divisées en blocs de taille fixe (128 bits) avant d'être chiffrées [12]. Le processus de chiffrement consiste en une série d'opérations complexes effectuées sur ces blocs de données, y compris des substitutions, des permutations et des opérations de mélange. Ces opérations sont répétées plusieurs fois (10, 12 ou 14 tours, selon la longueur de la clé utilisée) pour renforcer la sécurité [13].

L'AES offre une sécurité de niveau Militaire avec des clés de chiffrement de 128, 192 ou 256 bits, ce qui le rend beaucoup plus sûr que le DES qui n'utilisait qu'une clé de 56 bits. Les clés de chiffrement AES sont également beaucoup plus difficiles à casser par des attaques par force brute que celles du DES.

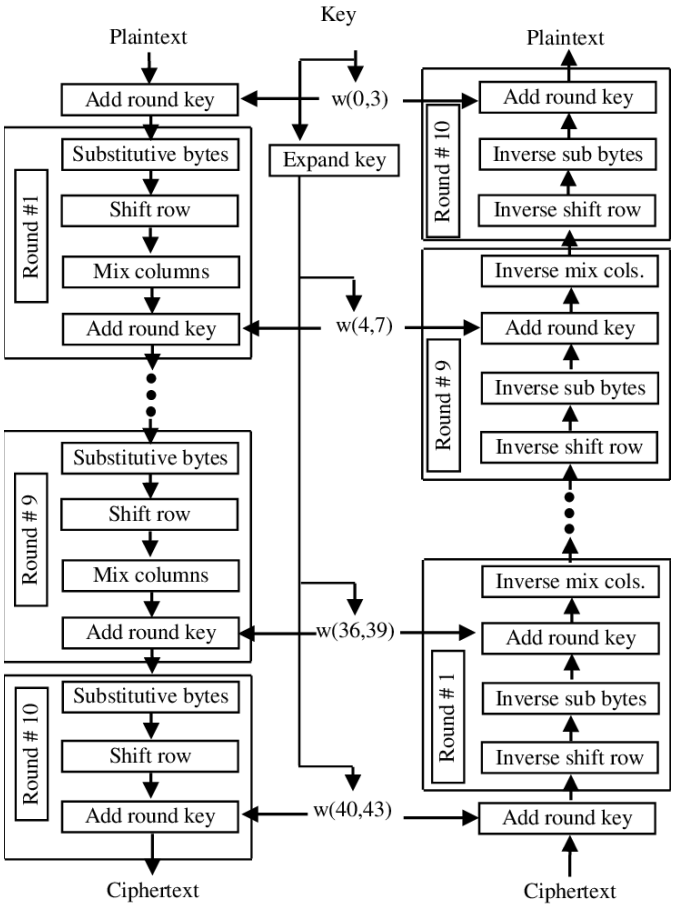


Figure 4 : Block-Diagram-for-AES-Encryption-and-Décryption [14]

I.6 Les Attaques

Les attaques en cryptographie peuvent être classées en deux grandes catégories : les attaques passives et les attaques actives.

Les attaques passives : sont des attaques où l'attaquant ne modifie pas les messages, mais écoute simplement les communications entre les parties pour essayer de récupérer de l'information. Parmi les techniques utilisées dans ce type d'attaque, on peut citer l'analyse de trafic, la cryptanalyse, le calcul de la fréquence des lettres ou des symboles, la recherche d'erreurs dans les messages, ou encore la recherche de motifs.

Les attaques actives : sont des attaques où l'attaquant modifie les messages échangés entre les parties ou insère de nouveaux messages. Les attaques actives peuvent avoir pour but de voler des informations, de corrompre ou de modifier des données, de bloquer des transmissions, ou encore d'usurper l'identité de l'une des parties. Les techniques utilisées dans ce type d'attaque incluent l'injection de paquets, la redirection de flux de données, la répétition de messages, ou encore la création de faux messages [6].

Il est important de noter que ces deux types d'attaques peuvent être menées à différents niveaux de la communication : au niveau des couches basses du réseau (comme les attaques par déni de service), au niveau des protocoles de communication (comme les attaques man-in-the-middle), ou encore au niveau des algorithmes de cryptographie eux-mêmes (comme les attaques par force brute ou les attaques par clé apparente).

Il est donc crucial de prendre en compte les différentes menaces et d'adapter les systèmes de sécurité en conséquence pour assurer une protection efficace contre les attaques par force brute que celles du DES.

I.6.1 Les Attaques Passives

Les attaques passives sont des tentatives d'interception des messages chiffrés sans altérer leur contenu. Il existe plusieurs types d'attaques passives, notamment :

I.6.1.1 L'Analyse Simple de la Consommation de Puissance (SPA)

L'attaque par Analyse Simple de la Consommation de Puissance (SPA) est une méthode d'attaque contre des dispositifs cryptographiques qui consiste à mesurer la consommation de puissance du dispositif pendant qu'il effectue une opération cryptographique. Cette attaque exploite les variations minuscules de consommation d'énergie électrique qui se produisent pendant que le dispositif effectue une opération cryptographique, ce qui peut révéler des informations sur la clé secrète utilisée dans le processus.

La SPA peut être utilisée pour attaquer des dispositifs cryptographiques tels que des cartes à puce, des processeurs embarqués et des systèmes de chiffrement matériel. Cette attaque est considérée comme une attaque passive car elle n'implique pas d'altération du dispositif ou de l'environnement de l'attaque.

Pour se protéger contre l'attaque SPA, des techniques telles que la dissimulation de la consommation de puissance, l'utilisation de circuits de protection et la conception de dispositifs à faible consommation d'énergie peuvent être mises en œuvre. Cependant, des attaques plus sophistiquées, telles que les attaques par Analyse Différentielle de la Consommation de Puissance (DPA), ont été développées pour surmonter ces défenses.

I.6.1.2 L'Analyse Différentielle de la Consommation de Puissance (DPA)

L'Analyse Différentielle de la Consommation de Puissance (DPA) est une attaque qui exploite les variations minuscules de consommation de puissance d'un dispositif cryptographique pendant son fonctionnement pour récupérer des informations sensibles telles que les clés de chiffrement.

L'attaque DPA est basée sur le fait que la consommation de puissance du dispositif cryptographique varie en fonction des opérations effectuées par celui-ci. En mesurant la consommation de puissance pendant l'exécution de ces opérations, il est possible d'extraire des informations sur les données manipulées, telles que les bits de clé de chiffrement.

L'attaque DPA est considérée comme une attaque puissante contre les dispositifs cryptographiques, car elle ne nécessite pas d'accès direct au dispositif ou à son environnement. Au lieu de cela, un attaquant peut simplement écouter la consommation de puissance du dispositif à distance, par exemple en utilisant une sonde de mesure de courant placée entre le dispositif et son alimentation.

Pour se protéger contre les attaques DPA, les concepteurs de dispositifs cryptographiques peuvent utiliser des techniques telles que le masquage de la clé, qui consistent à ajouter du bruit aléatoire à la consommation de puissance pour rendre l'analyse DPA plus difficile. D'autres techniques comprennent l'utilisation de matériel de sécurité spécifique, tels que des FPGA (Field-Programmable Gate Arrays) et des SoC (System-on-Chip) sécurisés, qui intègrent des mesures de sécurité matérielles pour réduire la sensibilité de la consommation de puissance aux opérations cryptographiques.

I.6.1.3 L' Analyse du Temps de Calcul (TA)

L'Analyse du Temps de Calcul (TA), ou Timing Attack en Anglais, est une technique

d'attaque qui vise à exploiter les différences de temps de calcul entre les différentes opérations d'un algorithme cryptographique pour en déduire des informations sensibles.

En effet, les algorithmes cryptographiques sont souvent implémentés sur des microprocesseurs ou des microcontrôleurs qui peuvent avoir des temps de calcul variables en fonction des opérations effectuées. Par exemple, le temps de calcul nécessaire pour effectuer une opération de chiffrement peut varier en fonction de la valeur des données en entrée ou de la clé utilisée. En mesurant ces temps de calcul, il est possible de déduire des informations sur la clé secrète utilisée pour le chiffrement.

Les attaques par Analyse du Temps de Calcul peuvent être réalisées à distance, en observant les temps de réponse d'un système lorsqu'il effectue des opérations de chiffrement, ou localement, en mesurant les temps d'exécution sur une machine locale. Ces attaques peuvent être particulièrement dangereuses car elles peuvent être effectuées sans avoir accès à la clé secrète elle-même.

Il existe plusieurs contre-mesures pour se prémunir contre les attaques par Analyse du Temps de Calcul, notamment en utilisant des méthodes de calcul constantes qui prennent toujours le même temps quelle que soit la valeur des données en entrée, ou en introduisant du bruit dans les temps de calcul pour empêcher les attaquants de mesurer avec précision les temps d'exécution.

I.6.2 Les Attaques Actives

Les attaques actives sont des tentatives d'altération de messages chiffrés. Il existe plusieurs types d'attaques actives, notamment :

1- L'attaque par Rejeu, qui consiste à répéter un message chiffré pour déterminer la clé de chiffrement utilisée.

2- L'attaque par Modification, qui consiste à modifier un message chiffré pour déterminer la clé de cryptage considéré.

3- L'attaque par Injection, qui consiste à injecter des données malveillantes dans un message chiffré pour déterminer la clé secret.

I.7 Sécurité et Longueur des Clés

La sécurité des algorithmes de chiffrement tels que RSA et les algorithmes basés sur les courbes elliptiques dépend de la longueur de leurs clés [8].

Pour RSA, les clés typiques utilisées dans les applications modernes sont de 2048 bits ou plus. Cela fournit un niveau de sécurité suffisant pour la plupart des utilisations, car la factorisation d'un nombre entier de 2048 bits est considérée comme actuellement inattaquable. Cependant, si la

clé est trop longue, elle peut ralentir le processus de chiffrement et de déchiffrement. Par exemple, une clé RSA de 3072 bits est environ deux fois plus lente qu'une clé de 2048 bits.

Pour les algorithmes basés sur les courbes elliptiques, une longueur de clé typique est de 256 bits ou plus. Les courbes elliptiques offrent une sécurité équivalente à celle de RSA, mais avec des clés plus courtes, ce qui les rend plus rapides et plus efficaces en termes d'utilisation des ressources. Cependant, les courbes elliptiques sont plus récentes que RSA et sont donc moins testées dans le temps.

I.8 Discussion

La cryptologie est une science qui englobe la cryptographie et la cryptanalyse. La cryptographie implique la création de techniques de chiffrement, tandis que la cryptanalyse consiste à trouver des faiblesses dans ces techniques pour les exploiter. Les algorithmes de cryptographie sont des méthodes mathématiques utilisées pour protéger les informations en les rendant illisibles à toute personne sans la clé appropriée. Les attaques à la cryptographie peuvent être passives ou actives, et certaines des méthodes les plus populaires sont l'Analyse de la consommation de puissance (SPA), l'Analyse différentielle de la consommation de puissance (DPA) et l'Analyse du temps de calcul (TA). La sécurité des algorithmes de chiffrement dépend de la longueur des clés utilisées, mais il est également important de trouver un équilibre entre la sécurité et la performance.

II. Cryptographie avec les Courbes Elliptiques

II.1 Préambule

La cryptographie avec les courbes elliptiques est une technique de chiffrement qui utilise les propriétés mathématiques des courbes elliptiques pour protéger les données. Cette technique est de plus en plus utilisée dans les applications de sécurité, en particulier pour la création de protocoles cryptographiques sûrs.

II.2 Le Problème du Logarithme Discret des Courbes Elliptiques ECDLP

Le Problème du Logarithme discret des courbes elliptiques (ECDLP, acronyme en anglais "Elliptic Curve Discrete Logarithm Problem") est un problème mathématique fondamental dans le domaine de la cryptographie basée sur les courbes elliptiques. Il est similaire au Problème du Logarithme discret (DLP) qui se pose dans les groupes cycliques, mais il s'applique spécifiquement aux courbes elliptiques. L'ECDLP consiste à trouver l'exposant inconnu d'un point donné sur une courbe elliptique, en utilisant uniquement des opérations algébriques sur cette courbe.

Formellement, soit une courbe elliptique E définie sur un corps fini, et soit P un point appartenant à cette courbe. Le Problème du Logarithme discret des courbes elliptiques consiste à trouver un entier n tel que nP = Q, où Q est un autre point connu de la courbe. Autrement dit, il s'agit de déterminer l'exposant n qui permet de multiplier le point P pour obtenir le point Q.

L'ECDLP est un problème difficile à résoudre, et sa complexité augmente exponentiellement avec la taille du corps fini et la taille de la courbe elliptique utilisée. Cela en fait une base solide pour les algorithmes de cryptographie à clé publique basés sur les courbes elliptiques, tels que l'algorithme de Diffie-Hellman ECDH (Elliptic Curve Diffie-Hellman)et l'algorithme de signature ECDSA (Elliptic Curve Digital Signature Algorithm).

La sécurité de ces algorithmes repose sur l'hypothèse que résoudre le Problème du Logarithme discret des courbes elliptiques est difficile, ce qui signifie qu'il n'existe pas d'algorithme efficace pour le résoudre en temps raisonnable, même avec des ressources de calcul puissant.

II.3 Les Courbes Elliptiques

La courbe elliptique E est une courbe algébrique qui peut être représentée par l’équation Weierstrass [15]:

$$E : y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

Nous supposons que la courbe est définie dans un corps K et les paramètres $a_1, a_2, a_3, a_4, a_6 \in K$.

Les courbes elliptiques ont des propriétés intéressantes en mathématiques, en particulier en théorie des nombres. Elles sont également utilisées en cryptographie pour leur capacité à fournir des algorithmes de chiffrement efficace et sécurisé.

En utilisant des courbes elliptiques, il est possible de générer des clés de chiffrement qui sont plus courtes que les clés RSA ou DSA, tout en offrant le même niveau de sécurité [16].

Il existe plusieurs types des courbes elliptiques, notamment :

1/Courbes elliptiques ordinaires : ce sont des courbes elliptiques définies sur un corps fini, qui ont un groupe des points régulier et prévisible.

2/Courbes supersingulières : ces courbes ont un groupe des points singulier et imprévisible, ce qui les rend utiles pour la cryptographie.

3/Courbes de Montgomery : ces courbes sont définies par une équation différente de celle des courbes elliptiques ordinaires et sont utilisées dans certains protocoles cryptographiques.

4/Courbes tordues : ce sont des courbes qui ont la même structure que les courbes elliptiques ordinaires, mais dont l'équation est modifiée par une transformation affine.

5/Courbes de Koblitz : ces courbes sont un type particulier des courbes tordues, utilisées pour la cryptographie à clé publique.

Chacun de ces types des courbes elliptiques présente des propriétés et des caractéristiques particulières qui les rendent adaptées à différents types d'applications en cryptographie.

II.4 L'arithmétique des Courbes Elliptiques

L'arithmétique des courbes elliptiques est la base mathématique de la cryptographie à courbe elliptique. Elle repose sur la définition des opérations d'addition et de multiplication sur les points de la courbe. Ces opérations sont utilisées pour générer des clés, chiffrer et déchiffrer des messages, et pour effectuer des opérations de signature [15].

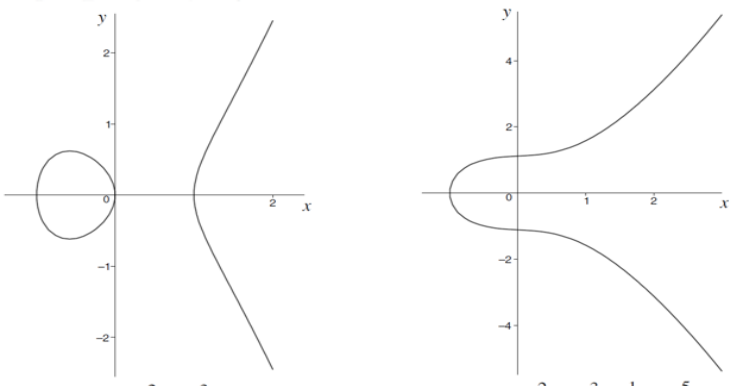


Figure 5: Courbes Elliptiques Sur R [8]

II.5 Les Algorithmes de Calcul de la Multiplication Scalaire

Les algorithmes de calcul de la multiplication scalaire sont des algorithmes utilisés pour effectuer des opérations de chiffrement et de signature sur les courbes elliptiques. Il existe plusieurs algorithmes de multiplication scalaire, notamment : échelle de Montgomery , Forme Non-Adjacente [15]:

II.5.1 Doublement et Addition (DA)

L'algorithme de doublement et d'addition (DA) est l'un des algorithmes les plus simples pour effectuer des opérations de multiplication scalaire sur les courbes elliptiques. Il repose sur la répétition de l'opération d'addition et de doublement sur un point de la courbe [8].

i) Calcul Algébrique :

•Ajout de deux points avec $x1 \neq x2$:

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$$

$$\lambda = (y_2 - y_1)/(x_2 - x_1)$$

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

• Doublement d'un point avec $x1 \neq 0$:

$$(x_1, y_1) + (x_2, y_2) = (x_3, y_3)$$

$$\lambda = (3x_1^2 + x_1)/(2y_1)$$

$$x_3 = \lambda^2 - 2x_1$$

$$y_3 = \lambda(x_1 - x_3) - y_1$$

ii) Calcul Géométrique :

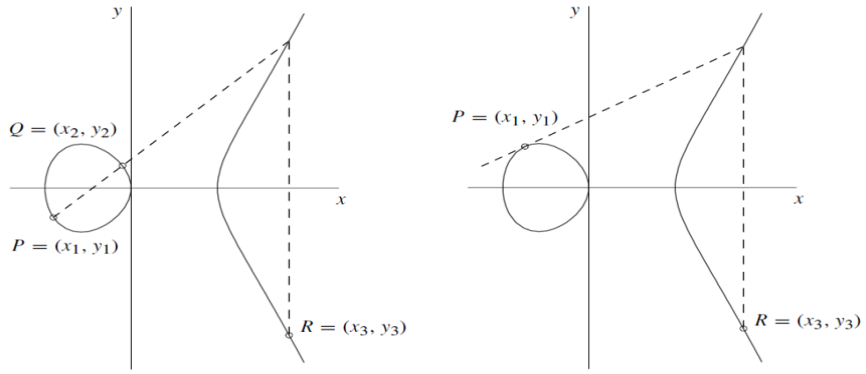


Figure 6 : Addition et Doublement Géométrique des Points de Courbe Elliptique [8]

II.5.2 Échelle de Montgomery

Cette technique a été décrite pour la première fois par Montgomery pour un type particulier de courbe en grande caractéristique et a été généralisée à d'autres courbes et à la même caractéristique [15].

Quelle que soit la forme de la courbe, nous utilisons une version modifiée de l'algorithme adaptée à la multiplication scalaire pour calculer [n] P.

Algorithm 1. Left-to-Right Binary algorithm for Montgomery's ladder

INPUT: A point P on E_m and a positive integer $n = (n_t \dots n_0)_2$
OUTPUT: The point $[n]P$

$P_1 \leftarrow P$ and $P_2 \leftarrow [2]P$
for $i = t - 1$ down to 0 do
  if $n_i = 0$ then
    $P_2 \leftarrow P_2 + P_1$ (P); $P_1 \leftarrow 2P_1$
  else
    $P_1 \leftarrow P_2 + P_1$ (P); $P_2 \leftarrow 2P_2$
  end if
end for
return P_1

Figure 7 : Montgomery Ladder Algorithm [17]

II.5.3 Forme Non-Adjacente (NAF)

Bien que la forme normale de Zeckendorf (NAF) garantisse que chaque entier est représenté de manière unique, son principal avantage est que le poids de Hamming de chaque valeur sera aussi faible que possible. Normalement, la moitié de tous les bits dans une représentation binaire d'une valeur seront non nuls, mais avec NAF, ce pourcentage est réduit à seulement un tiers de tous les chiffres. Cela se traduit par des implémentations de réseau d'addition/soustraction efficaces (comme la multiplication par une constante) dans la DSP.

Comme l'encodage Booth, il a été développé par G.W. Reitweisner pour accélérer les premiers algorithmes de multiplication car il est évident qu'au plus la moitié des chiffres sont non nuls. la représentation NAF peut être mise en œuvre de manière à ne nécessiter qu'un maximum de m + 1 bits pour un nombre qui serait généralement représenté en binaire avec m bits, car chaque chiffre non nul doit être contigu à deux 0.

Les caractéristiques de NAF le rendent bénéfique dans divers algorithmes, en particulier plusieurs en cryptographie, en réduisant la quantité de multiplication requise pour effectuer une exponentiation. La quantité de multiplications dans la technique d'exponentiation par carré dépend de la quantité des bits non nuls. Une valeur de chiffre de 1 indique une multiplication par la base et une valeur de chiffre de -1 implique son réciproque si l'exposant est fourni sous forme de NAF.

Algorithm : Computing the NAF of a positive integer

INPUT: A positive integer k.
OUTPUT: NAF(k).
1. $i \leftarrow 0$.
2. While $k \geq 1$ do
   2.1 If k is odd then: $k_i \leftarrow 2 - (k \bmod 4)$, $k \leftarrow k - k_i$;
   2.2 Else: $k_i \leftarrow 0$.
   2.3 $k \leftarrow k/2$, $i \leftarrow i + 1$.
3. Return$(k_{i-1}, k_{i-2}, \ldots, k_1, k_0)$.

Figure 8 : Non-Adjacent Form , (NAF) Algorithme [8]

II.6 Les Protocoles Cryptographiques sur les Courbes Elliptiques

Les protocoles cryptographiques sur les courbes elliptiques sont des protocoles de chiffrement et de signature qui utilisent les propriétés mathématiques des courbes elliptiques pour protéger les données. Il existe plusieurs protocoles cryptographiques sur les courbes elliptiques, notamment :

II.6.1 Le Protocole d'Échange de Clé

Le protocole d'échange de clé sur les courbes elliptiques (ECDH) est un protocole de chiffrement à clé publique qui permet à deux parties de s'échanger une clé de session secrète. Ce protocole est basé sur le fait que le calcul de la multiplication scalaire sur.

Une courbe elliptique est difficile à inverser, même pour un adversaire qui connaît la clé publique.

Le protocole ECDH se déroule en plusieurs étapes :

1/Les deux parties conviennent d'une courbe elliptique et d'un point de départ, appelé point de base.

2/Chacune des parties génère une clé privée aléatoire et calcule sa clé publique en multipliant le point de base par sa clé privée.

3/Les parties s'échangent leur clé publique.

4/Chaque partie calcule la multiplication scalaire entre sa clé privée et la clé publique de l'autre partie, ce qui donne la même clé de session secrète.

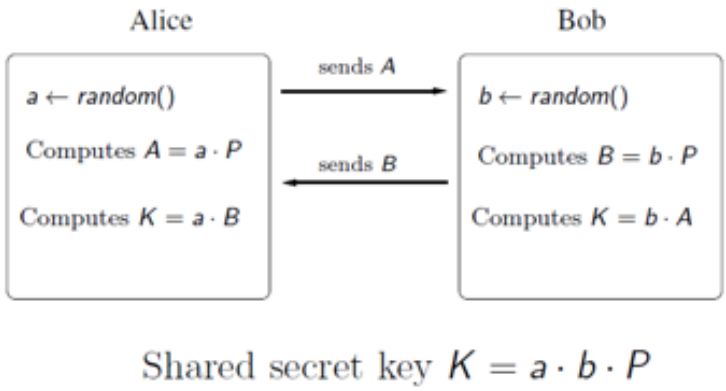


Figure 9 : Protocol ECDH

La sécurité du protocole ECDH repose sur la difficulté de calculer la clé privée à partir de la clé publique, ce qui est supposé être impossible avec les technologies actuelles.

II.6.2 Le Protocole de Signature ECDSA

ECDSA signifie Algorithme de signature numérique à courbe elliptique , Le protocole ECDSA est un protocole de signature numérique qui permet à une partie de signer un message de manière non répudiable. Ce protocole est basé sur le calcul de la multiplication scalaire sur une courbe elliptique et utilise des fonctions de hachage cryptographiques pour garantir l'intégrité du message signé.

Le protocole ECDSA se déroule en plusieurs étapes [18] :

Ètape 1/ La partie qui souhaite signer le message génère une paire des clés publique/privée et calcule le point de base multiplié par sa clé privée.

Ètape 2/ La partie qui souhaite signer le message calcule la valeur de hachage du message et la transforme en un entier.

Ètape 3/ La partie qui souhaite signer le message génère une valeur aléatoire appelée k et calcule le point de base multiplié par k.

Ètape 4/ La partie qui souhaite signer le message calcule la signature en utilisant le point

calculé à l'étape précédente et le nombre entier obtenu à l'étape 2.

Ètape 5/ La partie qui souhaite signer le message envoie la signature et le message signé à la partie qui doit vérifier la signature.

Ètape 6/ La partie qui doit vérifier la signature calcule la valeur de hachage du message et la transforme en un entier.

Ètape 7/ La partie qui doit vérifier la signature calcule la signature en utilisant la clé publique de la partie qui a signé le message.

Ètape 8/ La partie qui doit vérifier la signature compare la signature calculée à l'étape précédente avec la signature envoyée par la partie qui a signé le message. Si les deux signatures sont identiques, la partie qui a signé le message est authentifiée.

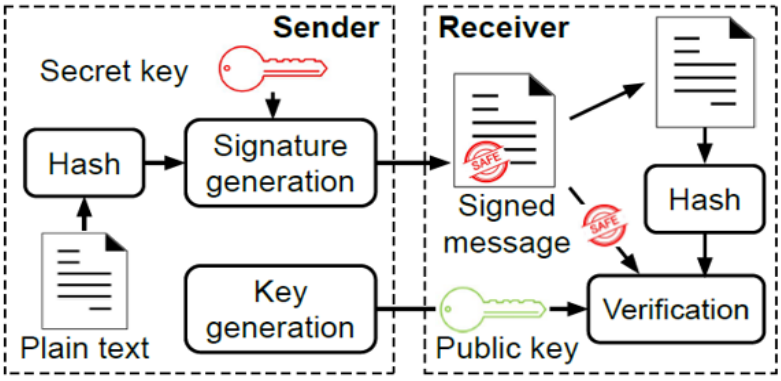


Figure 10 : Digital Signature Algorithm Protocol [19]

La sécurité du protocole ECDSA repose sur la difficulté de calculer la clé privée à partir de la clé publique, ce qui est supposé être impossible avec les technologies actuelles. Grâce à cette difficulté, ECDSA est considéré comme un standard sécurisé pour le déploiement des systèmes de signature numérique. Leur utilisation est aujourd'hui si variée qu'elles sont appliquées dans presque tous les domaines informatiques. Par exemple, l'infrastructure de certificat de sécurité SSL & TLS d'Internet fait un usage intensif de l'ECDSA. La Bitcoin pionnier de la technologie blockchain, utilise également ECDSA pour atteindre le haut niveau de sécurité qui le caractérise.

II.7 Discussion

Les courbes elliptiques sont une méthode de cryptographie relativement nouvelle et en constante évolution. Elles présentent de nombreux avantages par rapport aux autres méthodes de cryptographie, notamment en ce qui concerne la taille des clés et la sécurité. Les algorithmes basés sur les courbes elliptiques sont également plus rapides que les algorithmes classiques, ce qui les rend idéaux pour une utilisation dans des environnements à ressources limitées.

Cependant, il y a encore des préoccupations concernant la sécurité des courbes elliptiques,

notamment en ce qui concerne la façon dont elles sont choisies et les attaques potentielles qui peuvent être utilisées contre elles. Il est important de continuer à rechercher des courbes elliptiques plus sûres et de nouvelles méthodes d'attaque pour garantir que ces méthodes restent fiables.

En fin, les courbes elliptiques sont une méthode de cryptographie passionnante et prometteuse, qui a le potentiel de révolutionner la sécurité dans les années prochaines.

III. L'arithmétique Radix-2^w Pour la Multiplication Scalaire Dans la Cryptographie à Courbe Elliptique (ECC)

III.1 Préambule

Multiplication scalaire de la courbe elliptique k.P, où k est une constante non négative et P est un point sur la courbe elliptique, requiert deux opérations distinctes : addition (ADD) et doublement (DBL). Réduire le nombre ADDs sans augmenter le nombre de nombre de DBLs, un recodage de k avec moins de chiffres non zéro est nécessaires [20]. Sur la base de l'arithmétique Radix-2^w, nous introduisons une méthode de fenêtres w-bit où les propriétés de la vitesse, la mémoire et la sécurité sont décrites par l'analyse exacte Les formules comme preuve de supériorité. Contrairement aux fenêtres existantes algorithmes, pour minimiser le nombre d'ADDs à la taille de la fenêtre (w) est guidé par un optimum en fonction de la longueur de bit (l) du Le scalaire K. Le nombre de précompte requis est minimal Quant à la valeur de w. La méthode proposée reprend le série binaire k et évalue la multiplication De droite à gauche et de gauche à droite, de même. La méthode Radix-2^w est très facile à utiliser et hautement reconfigurable, permettant la mémoire de vitesse et les compromis de sécurité de la vitesse pour satisfaire les différentes contraintes du système cryptographique. En autre, la méthode montre une haute résistance aux attaques des canaux latéraux basées (SCA) sur la puissance, le timing Attack (TA) et Analyse statistique (SA). Toutes les propriétés de Radix-2^w sont confrontées à méthode des fenêtres standard à travers une analyse approfondie de Les complexités. Une comparaison globale est faite à l'aide des champs finis recommandés par NIST GF(2^1).[3]

III.2 L'arithmétique Radix-2^w

L'arithmétique Radix-2^w est une méthode qui divise les chiffres en groupes de w bits, où w est une puissance de 2. Cela permet de réduire le nombre d'opérations de base, telles que les multiplications et les additions, nécessaires pour effectuer des calculs sur des nombres à grand nombre de bits[3].

Voici la formule pour un scalaire positive k d'une longueur L bits exprimé en Radix-2^w**:**

$$K = \sum_{i=0}^{\lceil (l+1)/w \rceil - 1} \left(-2^{w-1}k_{w\times i+w-1} + 2^{w-2}k_{w\times i+w-2} + \ldots + 2^2 k_{w\times i+2} + 2^1 k_{w\times i+1} + 2^0 k_{w\times i} + k_{w\times i-1}\right) \times 2^{w\times i}$$
$$= \sum_{i=0}^{\lceil (l+1)/w \rceil - 1} Q_i \times 2^{w\times i}$$
$$= \sum_{i=0}^{\lceil (l+1)/w \rceil - 1} (-1)^{k_{w\times i+w-1}} \times m_i \times 2^{w\times i+n_i}$$

$$|Q_i| = 2^{n_i} \times m_i \qquad n_i \in \{0, 1, 2, \ldots, w-1\} \qquad m_i \in Os(2^w) \cup \{0, 1\} \qquad Os(2^w) = \{3, 5, 7, \ldots, 2^{w-1} - 1\}$$

$$Rs(2^w) = \left\{\left(k_l, m_{\lceil (l+1)/w \rceil - 1}, n_{\lceil (l+1)/w \rceil - 1}\right), \left(k_{l-w}, m_{\lceil (l+1)/w \rceil - 2}, n_{\lceil (l+1)/w \rceil - 2}\right), \ldots, (k_{2\times w-1}, m_1, n_1), (k_{w-1}, m_0, n_0)\right\}$$

Figure 11 : Radix 2^w Representation[3]

Et m_i,n_i Calculer avec cet Algorithme :

Algorithm . Computation of m_i and n_i
Input: Q_i
Output: m_i and n_i
1: Compute Q_i according to (1)
2: $n_i = 0$
3: **While** Q_i is even **do**
3.1: $Q_i = Q_i / 2$
3.2: $n_i = n_i + 1$
4: $m_i = Q_i$
5: **Return** (m_i, n_i)

Figure 12: Computation of m_i , n_i [3]

pour meilleur performance on stocke les m_i , n_i dans un tableau :

on fixe w=4

Table 1 : m_i , n_i [3]

Q_i					m_i	n_i
k_{4*j+3}	k_{4*j+2}	k_{4*j+1}	k_{4*j}	k_{4*j-1}		
0	0	0	0	0	0	0
0	0	0	0	1	1	0
0	0	0	1	0	1	0
0	0	0	1	1	1	1
0	0	1	0	0	1	1
0	0	1	0	1	3	0
0	0	1	1	0	3	0
0	0	1	1	1	1	2
0	1	0	0	0	1	2
0	1	0	0	1	5	0
0	1	0	1	0	5	0
0	1	0	1	1	3	1
0	1	1	0	0	3	1
0	1	1	0	1	7	0
0	1	1	1	0	7	0
0	1	1	1	1	1	3
1	0	0	0	0	1	3
1	0	0	0	1	7	0
1	0	0	1	0	7	0
1	0	0	1	1	3	1
1	0	1	0	0	3	1
1	0	1	0	1	5	0
1	0	1	1	0	5	0
1	0	1	1	1	1	2
1	1	0	0	0	1	2
1	1	0	0	1	3	0
1	1	0	1	0	3	0
1	1	0	1	1	1	1
1	1	1	0	0	1	1
1	1	1	0	1	1	0
1	1	1	1	0	1	0
1	1	1	1	1	0	0

In Radix-2^4, $n_i \in \{0,1,2,3\}$ and $m_i \in \{0,1,3,5,7\}$.

Pour illustrer on donne un exemple , en prenant un scaler k=5892973 en decimal qui donne (10110011110101101101101)en binaire implique l=23 , pour w=4 ça donne 6 slices de :

$$Rs(2^4)=\left\{\begin{matrix}(0,3,1),(1,3,1),(1,1,0),\\(1,5,0),(0,7,0),(1,3,0)\end{matrix}\right\}$$

Figure 13 : Les Tranches de Radix 2^4 [3]

Équivalent:

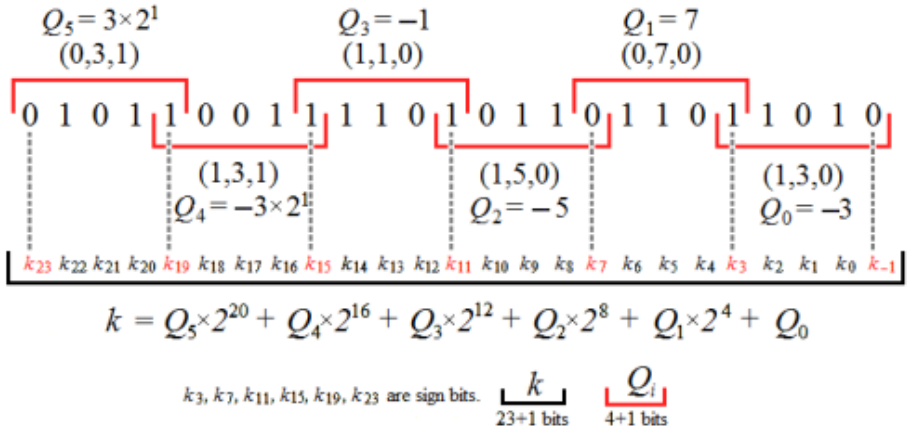


Figure 14 : Décomposition de $(5892973)_{10}$ en Radix 2^4 [3]

Implique : $k=3*2^{21}-3*2^{17}-1*2^{12}-5*2^8+7*2^4-3*2^0$

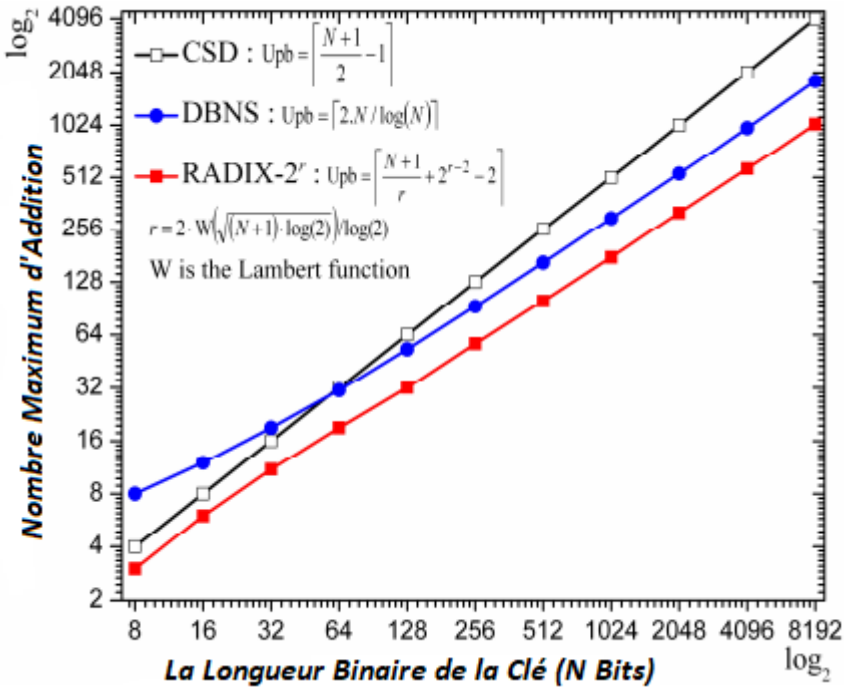
La multiplication des deux nombres de n bits peut être effectuée en utilisant des multiplications des chiffres plus petits, et l'addition des deux nombres de n bits peut être effectuée en utilisant des additions de chiffres plus petits. Avec l'arithmétique Radix-2^w, les opérations sont effectuées sur des chiffres de w bits, ce qui réduit le nombre total d'opérations nécessaires.

III.3 Les Avantages du Recodage Radix-2^w

Il a beaucoup de privilège de sa part comme :

III.3.1 Augmentation de la Vitesse et Réduction de la Consommation de Mémoire

L'arithmétique Radix-2^w offre une meilleure vitesse et une consommation de mémoire réduite par rapport aux méthodes d'arithmétique conventionnelles. Cela est dû au fait que les opérations sur des chiffres plus petits sont plus rapides et nécessitent moins de mémoire [20].



Comparaison du nombre d'additions pour une constante de N bits [20]

III.3.2 Augmentation de la Sécurité

L'arithmétique Radix-2^w peut également offrir une meilleure sécurité dans la multiplication scalaire. Cela est dû au fait que la méthode réduit la probabilité d'attaques par analyse de consommation de puissance (SPA) et par analyse différentielle de puissance (DPA).

Tableau 1 : Comparaison de la Complexité En Fonction de La Longueur de Bit l de La Clé k Et de La Fenêtre W[3]

ECSM Algorithms	↦	Complexity					
		Maximum number of ADDs	Average number of ADDs	Number of DBLs	Stored points	Additional memory (bits)	Average number of combinations to hack the key k
NAF [11]	No	$\lceil (l+1)/2 \rceil - 1$	$\lceil (l+1)/3 \rceil - 8/9$	$l-1$	0	0	$2^{\lceil (l+1)/3 \rceil - 1/9}$
ML [36]	Yes	$l-1$	$l-1$	$l-1$	0	0	2^{l-1}
w-NAF [25]	No	NA	$(l+5/4)/(w+3/2)+2^{w-1}-1$	$\lceil l-w+11/4 \rceil$	$2^{w-1}-1$	$O(l)$	NA
w-MOF [7]	Yes	$\lceil (l+1)/w \rceil + 2^{w-2}$	$\approx (l+1)/(w+1)+2^{w-2}-1$	$l-1$	2^{w-2}	$\lceil w-1+\log_2(w-2) \rceil \times (l/w)$	NA
SBR m-Ary [14]	No	$\lceil (l+1)/w \rceil + 2^{w-1}-1$	$\approx \lceil (l+1)/w \rceil + 2^{w-1}-1$	$l-1$	$2^{w-1}-1$	NA	NA
DBC [17]	No	NA	$\approx 0.19 \times l$	NA	2	$O(l^2)$	NA
Radix-2^w Algorithm 2	Yes	$\lceil (l+1)/w \rceil + 2^{w-2} - 1$ (4)	$(1-2^{-w}) \times \lceil (l+1)/w \rceil + 2^{w-2} - 1$ (6)	$l-1$	$2^{w-2}-1$ (7)	$2^w \times \lceil (w-1) + \lceil \log_2(w-1) \rceil \rceil$ (8)	$An(m_i)^{(1-2^{-w}) \times (l+N\Sigma)/w}$ (11)
Radix-2^{w_s} Algorithm 3	Yes	$\lceil (l+1)/w_s \rceil + 2^{w-2} - 1$ (12)	$-1+2^{w-2}+\sum_{w_s=w_s}^{w}(1-2^{-w_s}) \times \left\lceil (l+1)/\sum_{j=w_s}^{w} j \right\rceil$ (13)				$\prod_{w_s=w_s}^{w} An(m_i)^{(1-2^{-w_s}) \times N\Sigma}$ (14) $2^{-1+(w-1) \times \sum_{w_s=w_s}^{w}(1-2^{-w_s}) \times N\Sigma}$ (15)

III.4 L'algorithme de Radix-2^w pour l'ECSM

L'arithmétique Radix-2^w est utilisée pour la multiplication scalaire dans l'algorithme Elliptic Curve Scalar Multiplication (ECSM). L'ECSM est un algorithme qui permet de calculer un point P sur une courbe elliptique en multipliant un point de départ G par un scalaire n.

Algorithm : Proposed Radix-2^w Method

Input: $P \in E(\mathbb{F})$ and scalar k of bit-length l
Output: $R = k.P \in E(\mathbb{F})$
1: Compute w according to (5)
2: Determine w_{min} among $(\lfloor w \rfloor, \lceil w \rceil)$ according to (4)
3: Compute and store $Ps(2^{w_{min}})$ set
4: Concatenate a zero to k_0 according to (1)
5: Concatenate Nz zeros to k_{l-1} according to (9)
6: $R = P_\infty$ // P_∞ is the point at infinity
7: **For** $i = (l + Nz)/w_{min} - 1$ **down to** 0 **do**
7.1: $Q_i[1] = \overline{Q_i[0]}$; $Q_i[0] = Q_i\left[k_{w_{min} \times i + w_{min} - 1}\right]$
7.2: Use $Q_i[0]$ to load (m_i, n_i) from the indexation table
7.3: **For** $j = w_{min} - 1$ **down to** 0 **do**
7.3.1: $R = R + R$ // DBL
7.3.2: **If** $m_i \neq 0$ **then**
7.3.2.1: **If** $j = n_i$ **then**
7.3.2.1.1: $R = R + (-1)^{k_{w_{min} \times i + w_{min} - 1}} \times (m_i \times P)$ // ADD
8: **Return** R

RADIX-2^w CONFIGURATION PARAMETERS FOR NIST-RECOMMENDED GF(2^l)

Bit-length (l)	163	233	283	409	571
w_{min} (5)	5 bits		6 bits		
Nz (9)	2 bits		5 bits		

Figure 15 : Méthode Radix 2^w Proposée[3]

III.4.1 Analyse de la Vitesse à l'aide des Opérations Arithmétiques

L'arithmétique radix-2^w est connue pour sa capacité à accélérer les calculs de multiplication scalaire dans la cryptographie à courbe elliptique (ECC). Cependant, il est important de quantifier la vitesse d'un algorithme radix-2^w et de la comparer à d'autres méthodes de multiplication scalaire pour évaluer ses performances[3].

Tableau 2 : L'exigences de Mémoire[3]

Bit-length (l)		163	233	283	409	571
Stored precomputations (points)	NAF [11]	0				
	w-NAF^{+} [25]	15		31		
	w-MOF^{-} [7]	8		16		
	Radix-2^{w} (7)	7		15		
Additional Memory (bits)	NAF [11]	0				
	w-NAF* [25]	NA				
	w-MOF^{--} [7]	183	261	330	477	667
	Radix-2^{w} (8)	192		512		

Tableau 3 : Nombre des Moyens d'Ajouts[3]

Bit-length (l)	163	233	283	409	571
NAF* [11]	53.78	77.11	93.78	135.78	189.78
w-NAF^{+} [25]	40.27	51.04	58.73	85.70	107.30
w-MOF^{-} [7]	34.33[a]	46.00[a]	55.57[a]	73.57[a]	96.71[a]
Radix-2^{w} (6)	38.97	52.53	62.21	82.92	109.50
Saving (%) / w-NAF	3.22	–2.91	–5.92	3.24	–2.05

Tableau 4 : Nombre des Moyens d'Ajouts[3]

Bit-length (l)	163	233	283	409	571
NAF* [11]	81	116	141	204	285
w-NAF [25]	NA				
w-MOF^{+} [7]	41	55	64	85	112
Radix-2^{w} (4)	40	54	63	84	111
Saving (%) / NAF	50.62	53.45	55.32	58.82	61.05

III.4.2 Analyse du Potentiel de Résilience à L'attaque SPA

L'attaque par analyse de la consommation de puissance (SPA) est une méthode courante pour attaquer les systèmes cryptographiques. Cette attaque consiste à mesurer la consommation de puissance d'un dispositif pendant son fonctionnement afin de récupérer les informations clés utilisées dans l'algorithme de cryptage.

Une analyse du potentiel de résilience à l'attaque SPA est donc importante pour déterminer si l'utilisation de l'arithmétique Radix-2^{w} dans la multiplication scalaire de l'ECC est sûre contre ce type d'attaque. Monsieur OUDJIDA[3] a montré que l'utilisation de l'arithmétique radix-2^{w} peut offrir une résistance supérieure à l'attaque SPA par rapport à d'autres méthodes de multiplication scalaire telles que Montgomry Ladder, NAF.

Cependant, il est important de noter que la résistance à l'attaque SPA dépend également de la mise en œuvre spécifique de l'algorithme, de l'environnement de l'attaque et d'autres facteurs.

Par conséquent, il est toujours recommandé d'utiliser des contre-mesures supplémentaires pour renforcer la sécurité de l'ECC contre les attaques SPA.

III.5 Améliration de la Sécurisé - Recodage par Fenétrage Aléatoire

Une méthode proposée pour améliorer la sécurité du recodage est le recodage par fenêtrage aléatoire. Cette méthode consiste à partitionner la séquence binaire du scalaire en plusieurs fenêtres, et pour chaque fenêtre, un nombre aléatoire est choisi pour décaler la position de la fenêtre dans la séquence binaire. Le recodage par fenêtrage aléatoire est plus résistant à l'attaque SPA que le recodage standard, car l'attaque SPA nécessite la connaissance précise de la position des bits dans la séquence binaire[3].

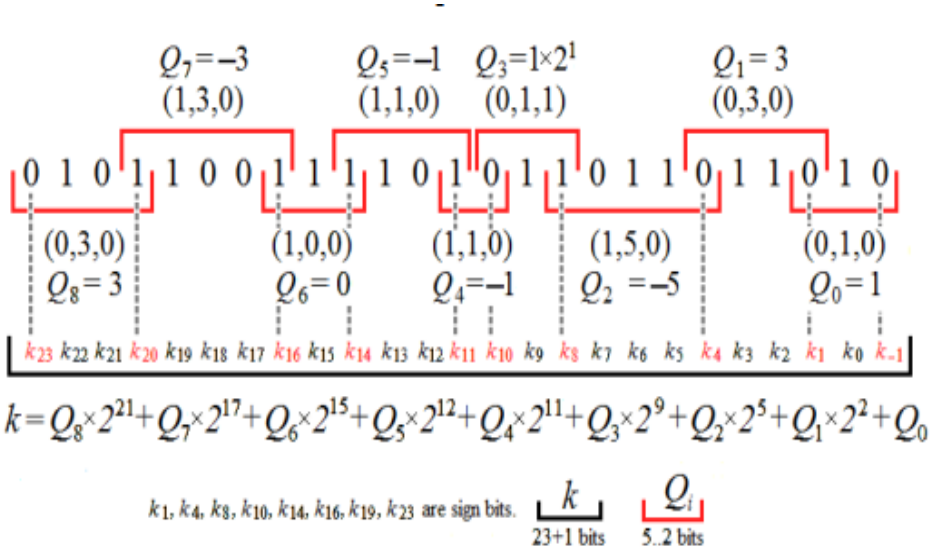


Figure 16 : Decomposition de (5892973).$_{10}$ en Radix $2^{4.1}$ [3]

III.6 Implémentation Logicielle

Le développement logicielle de l'arithmétique radix-2^w pour la multiplication scalaire dans la cryptographie à courbe elliptique (ECC) est une étape importante pour évaluer les performances de l'algorithme. Les performances dépendent des opérations arithmétiques implémentées, du choix des structures de données et des optimisations de code.

Plusieurs langages de programmation peuvent être utilisés pour implémenter l'arithmétique radix-2^w, tels que C, C++, Java, Python, etc. La sélection du langage de programmation dépend des besoins de l'application, de la complexité de l'implémentation, des performances requises et des ressources disponibles.

On choisit danc ce développement le langage Python , car est un langage de programmation de haut niveau, interprété et orienté objet, qui a été créé par Guido van Rossum et publié pour la première fois en 1991. Depuis lors, Python est devenu un langage de programmation très populaire et largement utilisé dans divers domaines de l'informatique. Python est considéré comme facile à apprendre et à lire, en grande partie en raison de sa syntaxe claire et concise. Python est souvent très utilisé en sécurité informatique , science des données pour l'analyse des données, la visualisation des données, la machine learning et la modélisation statistique. De nombreux packages et bibliothèques Python sont disponibles pour ces tâches, tels que cryptography , hashlib , NumPy, Pandas, Matplotlib, Scikit-learn, TensorFlow, Keras, etc.

Pour intégrer le recodage de Radix 2^w dans la cryptographie a courbe elliptique pour la signature numérique en Python , nous sommes besoin un ordinateur avec c'est caractéristiques :

Tableau 5 : Les Caractéristiques de L'ordinateur

Unité	Caractéristiques
Processeur	Intel(R) Core (TM) i3-7020U CPU @ 2.30Ghz 2.30Ghz
Mémoir RAM	12.0 Go
Disque DUR	256 Go SSD
Systéme d'exploitation	WIN 10-64 Xbits / UBUNTU 20.04 – 64 Xbits

Pour le dévellopment nous pouvons suivre les étapes suivantes :

Ètape 1/Choisir une bibliothèque de courbes elliptiques qui prend en charge le recodage de Radix 2^w, comme **PyCrypto** , **Charm Crypto** ou **ECpy**.

i) PyCrypto est une bibliothèque de chiffrement en Python qui prend en charge les courbes elliptiques et les fonctions de hachage cryptographique, mais elle n'est plus activement développée depuis 2013.

ii) **Charm Crypto** est une bibliothèque de cryptographie en Python qui prend en charge les courbes elliptiques et les fonctions de hachage cryptographique. Elle est toujours en développement actif et est compatible avec Python 2 et 3.

iii) **ellipticcurve** est une bibliothèque de cryptographie en Python qui prend en charge les courbes elliptiques, les fonctions de hachage cryptographique et le recodage de Montgomery. Elle

est compatible avec Python 3.

danc ce travail en utilisant **ellipticcurve**.

Ètape 2/Générer une paire des clés (publique et privée) pour une courbe elliptique choisie.en spécifiant les paramètres de la courbe et en utilisant une fonction de génération des clés.

```
Private key: 0xdd700a558a957c001a90668548fc98f68d7bb85cc65fe44de91126c51bb6e0c
Public point (Uncompressed):
0x0439164ecba3241b30ede33dd6ac1bd716b73803e4d9425e24719f1dfe782f7a7bbc795295bc678121dcfd239df
c07fadf4bc2c646ac05f9ad665be100c728b223
```

Figure 17 : Génération des Clés

Ètape 3/Générer un message à signer.

Utilisez n'importe quelle méthode pour générer un message à signer, comme la saisie manuelle ou la lecture d'un fichier.

```
## Sign some data
data = b"this is some data to sign"
```

Figure 18 : Message a Signer

Ètape 4/Calculer le hash du message en utilisant une fonction de hachage (La bibliothèque hashlib) telle que SHA-256.

```
5eb9fea36f5267baf61bb8f78e06fc7fb4f2b02e461052efddabb9c65e46c59e
```

Figure 19 : La Valeur Hachée du Message

Ètape 5/Utiliser le recodage de Radix 2^w pour convertir la clé publique en une forme accélérée.

Ètape 6/ Utilisez la clé privée et l'algorithme de signature choisi pour calculer la signature du message.

```
Signature:
0x3045022030db9ba2d1db3d976905aee255c9b312f3da5b1496c1710091e5e61b774b7239022100891399c1e7522
8c1d48b4604d6def337a46db76f33e076a4ec192d9b0bac1773
```

Figure 20 : Calcul de Signature

Ètape 7/Pour Vérifier la signature on Utilisez la clé publique, la signature calculée et le message d'origine. Pour améliorer les performances, utilisez le recodage de Radix 2^w pour convertir la clé publique en une forme accélérée avant la vérification de la signature.Si la vérification de la signature est réussie, le message a été signé avec succès.

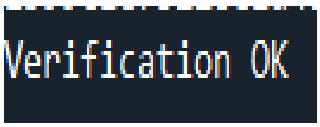


Figure 21 : Le Message a été Signé Avec Succès

La figure 22 représente l'interface de l'application :

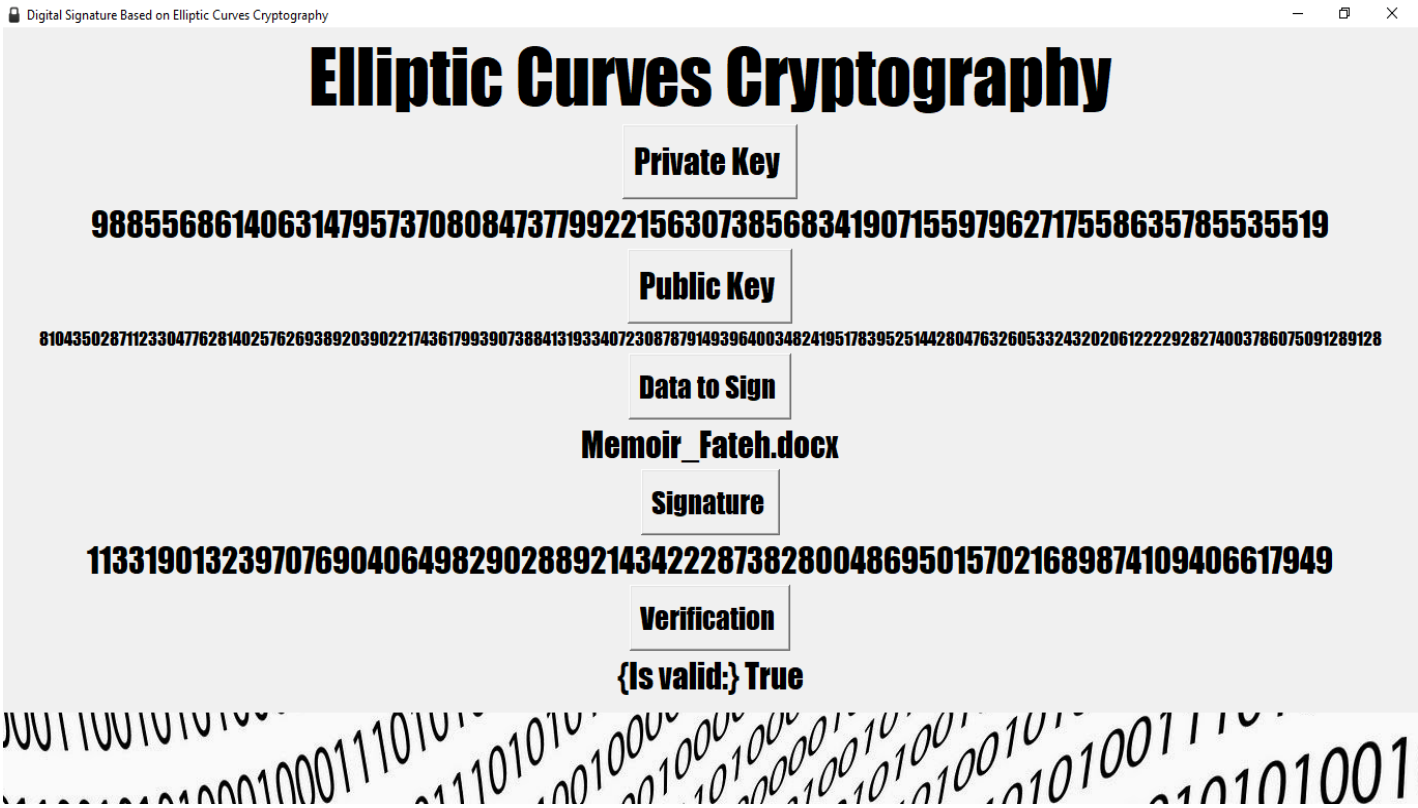


Figure 22 : L'interface de L'application

III.7 Discussion

L'arithmétique Radix-2^w pour la multiplication scalaire dans la cryptographie à courbe elliptique (ECC) est une technique efficace pour améliorer la sécurité et les performances des algorithmes de cryptographie ECC. Les avantages de l'utilisation de l'arithmétique radix-2^w comprennent une augmentation de la vitesse et de la sécurité des opérations arithmétiques, ainsi qu'une réduction de la consommation de mémoire.

Cependant, l'implémentation de l'arithmétique radix-2^w peut être complexe et nécessite une expertise en programmation, en mathématiques et en cryptographie. De plus, l'utilisation de

l'arithmétique radix-2^w peut entraîner une augmentation de la consommation d'énergie dans certains cas, ce qui doit être pris en compte lors de la conception des systèmes de cryptographie basse consommation.

Enfin, l'utilisation de l'arithmétique radix-2^w peut augmenter la sécurité contre certaines attaques, mais cela ne garantit pas une sécurité totale. Il est important de prendre en compte d'autres facteurs, tels que la longueur de la clé, l'utilisation de protocoles de sécurité appropriés et la gestion de clés efficace pour garantir la sécurité globale du système de cryptographie.

IV. L'arithmétique Radix-2^w Appliquée au Protocole de la Signature Électronique (ECDSA)

IV.1 Préambule

L'arithmétique à courbe elliptique (ECC) est largement utilisée dans les protocoles de sécurité moderne. Les signatures électroniques sont l'un des éléments clés de ces protocoles et sont utilisées pour garantir l'intégrité et l'authenticité des données transmises. Le protocole de signature électronique ECDSA (Elliptic Curve Digital Signature Algorithm) est largement utilisé dans les systèmes de sécurité modernes basés sur l'ECC. Cependant, la multiplication scalaire est une opération coûteuse qui nécessite une grande quantité de ressources. Cela peut affecter la vitesse et l'efficacité des systèmes basés sur l'ECC. L'utilisation de l'arithmétique radix 2^w peut améliorer et réduisant la consommation d'énergie et en augmentant la vitesse de calcul.

IV.2 Le Protocole de Signature Électronique

La signature électronique est un protocole de sécurité qui permet à une partie d'authentifier et de garantir l'intégrité des données transmises. Le protocole ECDSA est un algorithme de signature électronique basé sur l'ECC qui est largement utilisé dans les systèmes de sécurité modernes. Le protocole ECDSA utilise la multiplication scalaire pour générer une paire des clés publique et privée [21].

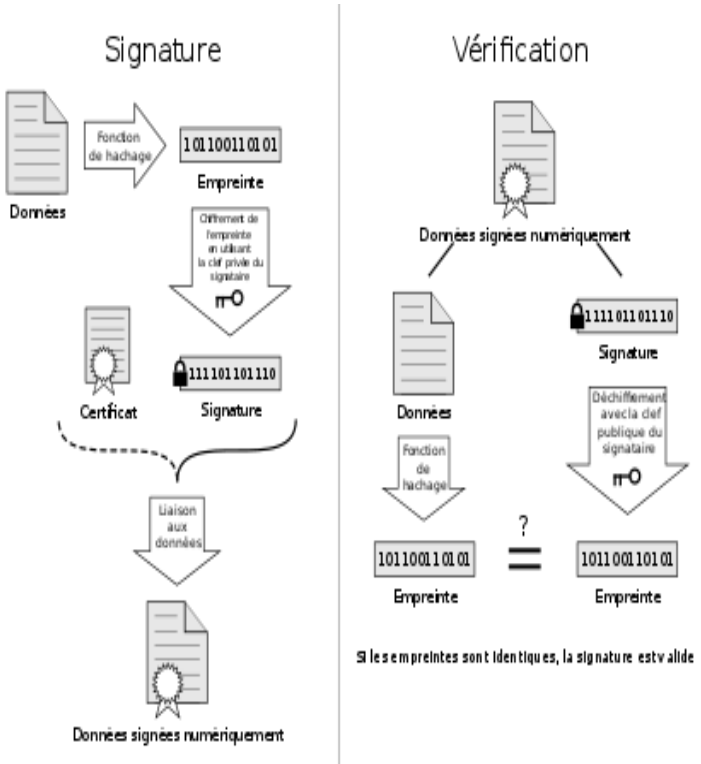


Figure 23 : Structure d'une Signature Numérique [22]

IV.2.1 Définition de la Signature Électronique

La Signature électronique, ou signature digitale, est une technique de sécurité utilisée pour garantir l'authenticité et l'intégrité d'un document numérique. Elle permet de prouver que le document a été signé par une personne spécifique et assure que le contenu du document n'a pas été modifié depuis sa signature.

La Signature électronique est basée sur un système de clé publique et privée. Le signataire utilise sa clé privée pour signer le document, ce qui crée une empreinte numérique unique du document. Cette empreinte numérique est ensuite cryptée avec la clé publique du signataire pour créer la signature électronique. Lorsque le document est reçu, la signature électronique est vérifiée en utilisant la clé publique du signataire pour déchiffrer l'empreinte numérique. Si l'empreinte numérique déchiffrée correspond à celle du document reçu, cela prouve que le document n'a pas été modifié depuis la signature et que le signataire est bien la personne qui a signé le document.

La Signature électronique est largement utilisée dans les domaines tels que la finance, le droit et l'administration pour garantir l'authenticité des documents numériques et la non-répudiation.

IV.2.2 Définition de la Signature Digitale

La Signature digitale est un mécanisme de sécurité utilisé dans les communications électroniques pour garantir l'authenticité, l'intégrité et la non-répudiation des données. Elle est principalement utilisée pour vérifier l'identité d'un expéditeur et s'assurer que le contenu d'un message n'a pas été altéré pendant la transmission.

La Signature digitale repose sur des techniques de cryptographie asymétrique, où une paire de clés est générée : une clé privée et une clé publique. L'expéditeur utilise sa clé privée pour générer une signature numérique en appliquant une fonction de hachage sur le contenu du message. Cette signature est ensuite attachée au message.

Le Destinataire peut vérifier l'authenticité de la signature en utilisant la clé publique correspondante à la clé privée de l'expéditeur. La fonction de hachage est à nouveau appliquée sur le message reçu, et la signature numérique est déchiffrée à l'aide de la clé publique. Si les valeurs correspondent, cela prouve que le message n'a pas été modifié et qu'il provient bien de l'expéditeur légitime.

La Signature digitale offre également la non-répudiation, ce qui signifie qu'une fois qu'une signature a été apposée sur un message, l'expéditeur ne peut pas nier l'avoir envoyé. Cela est possible car la clé privée utilisée pour générer la signature est censée être connue uniquement par

l'expéditeur.

IV.3 L'arithmétique Radix 2^w Appliquée à l'ECMSM

Multiplication par deux points (DPM) de la courbe elliptique u.P+v.Q, où u, v sont des entiers non négatifs et P, Q sont des points sur la courbe elliptique, est une opération critique dans la vérification de la signature numérique. Son schéma de calcul détermine l'ensemble des performances du système en terme de vitesse et de la sécurité. Dans cet mémoire, nous présentons une gamme d'algorithmes très simples pour DPM. Ils utilisent un modèle itératif uniforme basé sur une arithmétique à temps constant. Cela a l'avantage de contrecarrer les attaques de canaux secondaires (SCA) qui utilisent des mesures de synchronisation ou de consommation d'énergie pour pirater les clés secrètes u et v. Les algorithmes proposés utilisent une méthode de fenêtre de bits w qui recode simultanément les deux chaînes binaires u et v et évalue la double multiplication au volant de gauche à droite. Ce processus de recodage/évaluation en un seul passage a l'avantage d'accélérer le DPM, d'améliorer la résilience contre SCA et de réduire la consommation de mémoire. Les nouveaux algorithmes sont des méthodes basées sur des principes évalués avec des formules analytiques précises en termes de vitesse, de sécurité et de mémoire. Simplicité et flexibilité sont les principales caractéristiques des algorithmes proposés, permettant des compromis faciles entre vitesse - sécurité et vitesse-mémoire pour répondre à différentes contraintes. Les nouveaux algorithmes sont comparés aux méthodes des pointes grâce à un examen complet de la complexité à l'aide des courbes GF(2^l) recommandées par le NIST ainsi que des courbes GF(p) tordues d'Edwards et Montgomery.

IV.4 La Méthode Radix 2^w Proposée pour l'ECMSM

La Figure 24 montre un résumé des trois catégories principales des techniques de DPM. La technique la plus simple pour mettre en œuvre DPM est de cumuler les deux PM, u.P et v.Q, après les avoir calculés séquentiellement. L'inconvénient de ceci est qu'il appelle PDs, où l est la longueur de bit de u et v . De nombreuses stratégies différentes ont été proposées dans la littérature pour réduire le nombre de PD à l. Les algorithmes interliés et simultanés (IA et SA) sont les deux catégories dans lesquelles ils sont divisés . Bien qu'ils traitent les deux PM en parallèle, les IAs effectuent des recodings u et v individuellement alors que les SA les combinent . Les IAs et les SA exigent un prélèvement d'un nombre, tandis que dans les SA ils sont combinés . Un prélèvement d'un certain nombre de points est nécessaire pour les IAs et les SA, et ces points sont ensuite conservés dans la mémoire pour être utilisés lors de l'étape d'évaluation DPM. La catégorie des SA est divisée en deux sous-catégories: les SA à base de Montgomery (MSA) et Windowing SAs

(WSAs). Le recodage de fenêtre W-bit augmente la vitesse de DPM, mais augmente le nombre de points précalculés lorsqu'il est utilisé dans les IA et les WSA [23].

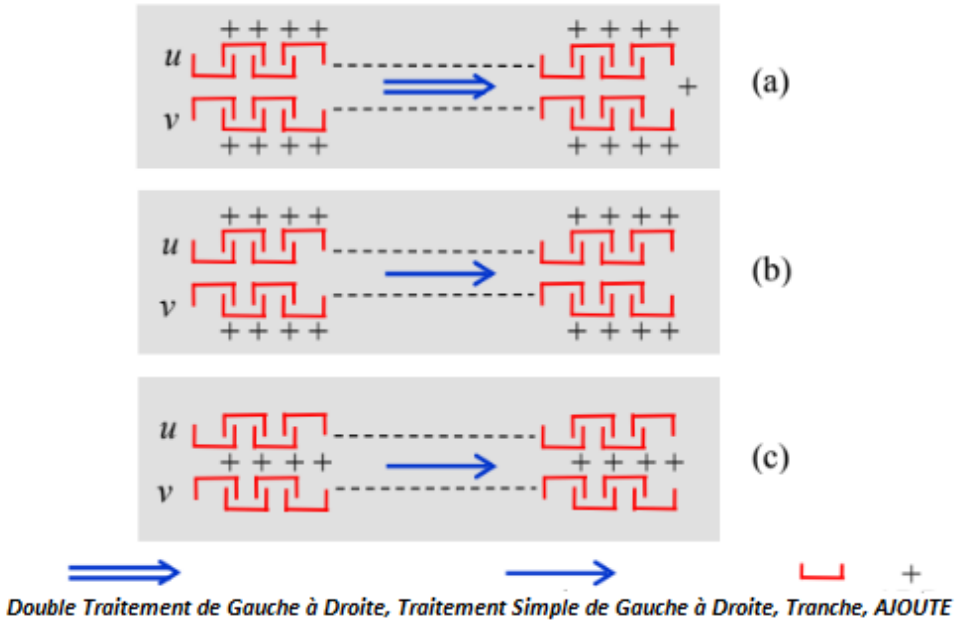


Figure 24 : Les Différents Méthodes de DPM , (a) Naïf (b) Entrelacés (c) Simultané [23]

Parmi les avantages du Radix 2^w pour ECMSM est [23] :

1. Simplicité et élégance : sur la base d'une formule simple et flexible, divers DPM sont facilement conçus et analysés.

2. Sécurité : le modèle uniforme constant-temps et indépendant de la clé est utilisé pour le calcul du DPM. Cela constitue une couverture contre les attaques d'analyse du temps et du pouvoir.

3. Vitesse, mémoire et sécurité : processus fusionnés de gauche à droite de recodage et d'évaluation de DPM.

4. Vitesse et mémoire : le nombre de PA est réduit à une limite quasi-optimale. Les mémoires de vitesse sont possibles.

5. Démonstration de la supériorité : algorithmes DPM fondés sur des principes où toutes les propriétés de la vitesse, de la mémoire et de la sécurité sont exprimées dans des formules analytiques précises pour la comparaison.

6. Indépendance des champs finis : algorithmes DPM génériques qui gèrent n'importe quel champ fini, sans aucune restriction.

IV.4.1 Le Calcul Indépendant de la Multiplication Multi-Scalaire à Temps Variable

L'algorithme radix 2^w pour l'ECMSM à temps variable est utilisé pour calculer les multiplications multi-scalaires à temps pas fixe. Cette méthode permet de réduire la dépendance entre les différentes opérations , ce qui permet de les exécuter en parallèle.

IV.4.1.1 Algorithme Radix 2^w pour l'ECMSM à Temps Variable

L'algorithme radix 2^w pour l'ECMSM à temps variable peut être divisé en deux étapes principales. La première étape consiste à pré-calculer les points de base, tandis que la seconde étape consiste à calculer la multiplication scalaire.

Figure 25 : Radix Naïf 2^2 , ECMSM [23]

Dans la première étape, les points de base sont pré-calculés en utilisant la méthode radix 2^w.

Algorithm : Calculation of m_i and n_i

Input: bits of slice d_i // w+1 bits
Output: m_i and n_i
1: Compute d_i according to (1)
2: $n_i = 0$; $d_i = |d_i|$ // | | is the absolute value
3: **While** d_i is even **do**
3.1: $d_i = d_i / 2$
3.2: $n_i = n_i + 1$
4: $m_i = d_i$
5: **Return** (m_i, n_i)

Figure 26 : Algorithme Radix 2^w pour Calculer m_i ,n_i pour DPM [23]

Cette étape permet de réduire le nombre de multiplications nécessaires pour calculer la multiplication scalaire. Dans la seconde étape, la multiplication scalaire est calculée en utilisant les points de base pré-calculés. Cette étape peut également être accélérée en utilisant la méthode radix 2^w.

IV.4.1.2 Algorithme Radix 2^w pour l'ECMSM à Temps Constant

L'algorithme radix 2^w pour l'ECMSM à temps constant utilise une technique de recodage

par fenêtrage pas par pas. Cette technique permet de réduire les temps d'accès à la mémoire et de minimiser les risques d'attaques par canal auxiliaire.

Figure 27 : Radix 2^2 Entrelacés [23]

et L'algorithme Radix 2^w Entrelacés pour l'ECMSM démontré comme :

Algorithm : Interleaved Radix-2^w DPM Method

Input: P and $Q \in E(\mathbb{F})$ and scalars u and v of bit-length l
Window bit-length w and number of zeros Nz

Output: $R = u.P+v.Q \in E(\mathbb{F})$

1: Compute and store $\{3P, 5P, 7P, \ldots, (2^{w-1}-1)P\}$

2: Compute and store $\{3Q, 5Q, 7Q, \ldots, (2^{w-1}-1)Q\}$

3: Concatenate a zero to u_0 and v_0 according to (1)

4: Concatenate Nz zeros to u_{l-1} and v_{l-1} according to (5)

5: $R = P_\infty$ // P_∞ is the point at infinity

6: **For** $i = (l+Nz)/w-1$ **down to** 0 **do**

6.1: Use $|du_i|$ to load (mu_i, nu_i) from the LUT(2^w)

6.2: Use $|dv_i|$ to load (mv_i, nv_i) from the LUT(2^w)

6.3: **If** $mu_i = 0$ **then** $Au = P_\infty$ **else** $Au = mu_i \cdot P$

6.4: **If** $mv_i = 0$ **then** $Av = P_\infty$ **else** $Av = mv_i \cdot Q$

6.5: **For** $j = w-1$ **down to** 0 **do**

6.5.1: $R = R+R$ // DBL

6.5.2: **If** $j = nu_i$ **then** $R = R+(-1)^{\cdots} \cdot Au$ // ADD

6.5.3: **If** $j = nv_i$ **then** $R = R+(-1)^{\cdots} \cdot Av$ // ADD

7: **Return** R

Figure 28 : Algorithme Radix 2^2 Entrelacé [23]

IV.4.2 Le Calcul Simultané de la Multiplication Multi-Scalaire

Outre le calcul indépendant de la multiplication multi-scalaire à temps variable, il existe une autre méthode pour accélérer le processus de calcul de la multiplication multi-scalaire. Cette méthode consiste à calculer simultanément plusieurs multiplications scalaires en utilisant une technique appelée « multi-exponentiation ». Cette technique combine les multiplicateurs pour effectuer une seule multiplication qui calcule simultanément plusieurs résultats.

Algorithm : Simultaneous Radix-2^2 Method

Input: P and $Q \in E(\mathbb{F})$ and scalars u and v of bit-length l

Output: $R = u.P+v.Q \in E(\mathbb{F})$

// Points of line 1 must be computed in the indicated order

1: $P+Q$; $P+2Q$; $2P+Q$; $P-Q$; $P-2Q$; $2P-Q$

2: Concatenate a zero to u_0 and v_0 according to (1)

3: Concatenate Nz zeros to u_{l-1} and v_{l-1} according to (5)

4: $P = P_\infty$ // P_∞ is the point at infinity

5: **For** $i = (l+Nz)/2-1$ **down to** 0 **do**

5.1: **Case** $(|du_i|, |dv_i|)$

5.1.1: (0,0) : $M = P_\infty$

5.1.2: (1,0) (2,0) : $M = P$

5.1.3: (0,1) (0,2) : $M = Q$

5.1.4: (1,1) (2,2) : $M = P + (-1)^{v_{2i+1} - u_{2i+1}} .Q$

5.1.5: (1,2) : $M = P + (-1)^{v_{2i+1} - u_{2i+1}} .2.Q$

5.1.6: (2,1) : $M = 2.P + (-1)^{v_{2i+1} - u_{2i+1}} .Q$

5.2: $R = R + R$ // DBL

5.3: **Case** $(|du_i|, |dv_i|)$

5.3.1: (0,0) (0,1) : $R = R + R$; $R = R + (-1)^{v_{2i+1}} .M$

5.3.2: (0,2) : $R = R + (-1)^{v_{2i+1}} .M$; $R = R + R$

5.3.3: (1,0) (1,1) (1,2) (2,1) : $R = R + R$;
$R = R + (-1)^{u_{2i+1}} .M$

5.3.4: (2,0) (2,2) : $R = R + (-1)^{u_{2i+1}} .M$; $R = R + R$

6: **Return** R

Figure 29 : La Method Radix 2^2 pour le Calcul Simultané de la Multiplication Multi-Scalaire [23]

La technique de multi-exponentiation est basée sur le fait que les multiplicateurs sont généralement des nombres relativement petits et que la plupart des bits de ces nombres sont nuls. La méthode consiste à diviser chaque multiplicateur en plusieurs parties et à regrouper ces parties en plusieurs ensembles. Chaque ensemble est ensuite utilisé pour calculer une multiplication scalaire indépendante. Les résultats de ces multiplications sont ensuite combinés pour obtenir le résultat final de la multiplication multi-scalaire.

L'arithmétique radix 2^w peut être utilisée pour accélérer le calcul simultané de la multiplication multi-scalaire. En utilisant la méthode radix 2^w simultanément, il est possible de diviser chaque multiplicateur en plusieurs parties de taille égale, ce qui facilite le calcul simultané de plusieurs multiplications scalaires. Cette méthode peut également être utilisée en conjonction avec la méthode de calcul indépendant de la multiplication multi-scalaire à temps variable pour

améliorer encore la vitesse de calcul.

IV.5 Amélioration Supplémentaire

L'utilisation de la cryptographie à courbes elliptiques pour la sécurité des systèmes repose sur la possibilité de calculer efficacement les multiplications scalaires sur la courbe elliptique. Les techniques de DPM sont utilisées pour accélérer les multiplications scalaires, en particulier lorsqu'il est nécessaire de calculer simultanément plusieurs multiplications scalaires. L'algorithme radix 2^w pour l'ECMSM à temps variable est une méthode efficace pour calculer les multiplications multi-scalaires à temps pas fixe. Cette méthode réduit la dépendance entre les différentes opérations, ce qui permet de les exécuter en parallèle. L'algorithme radix 2^w pour l'ECMSM à temps constant utilise une technique de recodage par fenêtrage pas par pas pour réduire les temps d'accès à la mémoire et minimiser les risques d'attaques par canal auxiliaire. En outre, la technique de multi-exponentiation peut être utilisée pour calculer simultanément plusieurs multiplications scalaires en combinant les multiplicateurs pour effectuer une seule multiplication qui calcule simultanément plusieurs résultats. L'utilisation de l'arithmétique radix 2^w peut également accélérer le calcul simultané de la multiplication multi-scalaire en divisant chaque multiplicateur en plusieurs parties de taille égale, ce qui facilite le calcul simultané de plusieurs multiplications scalaires. Ces méthodes permettent d'améliorer considérablement la vitesse de calcul des multiplications scalaires sur les courbes elliptiques, ce qui est essentiel pour la sécurité des systèmes modernes.

IV.6 Discussion

La multiplication multi-scalaire est une opération courante en cryptographie, notamment en cryptographie à clé publique basée sur les courbes elliptiques. L'algorithme radix 2^w est particulièrement utile pour la multiplication multi-scalaire à temps variable, car il permet de réduire le nombre des multiplications nécessaires pour calculer la multiplication scalaire. De plus, l'utilisation de l'arithmétique radix 2^w permet de diviser chaque multiplicateur en plusieurs parties de taille égale, ce qui facilite le calcul simultané de plusieurs multiplications scalaires. Cela améliore considérablement la vitesse de calcul et rend l'algorithme plus efficace.

En ce qui concerne la technique de multi-exponentiation, elle est utile pour accélérer le calcul simultané de plusieurs multiplications scalaires. Cependant, elle peut nécessiter plus de mémoire que l'algorithme radix 2^w , car elle doit stocker les résultats intermédiaires de chaque multiplication scalaire. De plus, la technique de multi-exponentiation peut être plus complexe à mettre en œuvre que l'algorithme radix 2^w.

En fin de compte, le choix de la technique à utiliser dépendra des besoins spécifiques de chaque application. Il peut être nécessaire de prendre en compte des facteurs tels que la sécurité, la vitesse de calcul et la disponibilité de la mémoire. Dans tous les cas, il est important de comprendre les avantages et les inconvénients de chaque technique pour pouvoir prendre une décision éclairée.

Conclusion Générale

Les Courbes Elliptiques ont joué un rôle essentiel dans le domaine de la cryptographie moderne. Leur structure mathématique unique et leurs propriétés intrinsèques en font un outil puissant pour la sécurisation des communications et des transactions numériques. L'une des applications les plus connues des courbes elliptiques en cryptographie est l'Elliptic Curve Digital Signature Algorithm (ECDSA). L'ECDSA est un algorithme de signature numérique basé sur les courbes elliptiques. Il permet de garantir l'intégrité, l'authenticité et la non-répudiation des données échangées sur des réseaux non sécurisés.

L'ECDSA présente plusieurs avantages par rapport à d'autres algorithmes de signature numérique. Tout d'abord, les clés utilisées dans l'ECDSA sont beaucoup plus courtes que celles des algorithmes basés sur la factorisation, tels que le RSA. Cela réduit les besoins en termes de puissance de calcul et de capacité de stockage.

L'ECDSA offre une meilleure résistance aux attaques cryptographiques, notamment les attaques par force brute et les attaques basées sur la factorisation. Les courbes elliptiques ont une structure mathématique complexe qui rend la résolution du problème du logarithme discret plus difficile que dans d'autres groupes, tels que les groupes basés sur la factorisation.

Le Recodage Radix 2^w, il s'agit d'une technique utilisée pour optimiser les opérations arithmétiques sur les courbes elliptiques. Le recodage consiste à représenter les entiers utilisés dans les calculs sur les courbes elliptiques sous une forme spéciale qui permet des calculs plus rapides et plus efficaces.

Les avantages du recodage Radix 2^w sont multiples. Il permet de réduire le nombre d'opérations nécessaires pour effectuer des calculs sur les courbes elliptiques, ce qui se traduit par des gains de performance significatifs. De plus, le recodage Radix 2^w peut aider à prévenir certaines attaques basées sur des fuites d'informations, telles que les attaques par canal auxiliaire.

En résumé, les courbes elliptiques sont devenues un élément essentiel de la cryptographie moderne. Leur utilisation dans des algorithmes tels que l'ECDSA permet de garantir la sécurité des communications et des transactions numériques. De plus, le recodage Radix 2^w offre des avantages significatifs en termes de performances et de résistance aux attaques.

References

[1] Claude Shannon. Communication Theory of Secrecy Systems.1949

[2] Niels Ferguson, Bruce Schneier, and Tadayoshi Kohno. Cryptography Engineering: Design Principles and Practical Applications

[3] Abdelkrim Kamel Oudjida & Ahmed Liacha. Radix-2^w Arithmetic for Scalar Multiplication in Elliptic Curve Cryptography, CDTA.2015

[4] Stinson Douglas R. Cryptography: Theory and Practice

[5] Jonathan Katz and Yehuda Lindell. Introduction to Modern Cryptography

[6] Stallings William. Cryptography and Network Security: Principles and Practice.

[7] C.Bruce Schneier. Applied Cryptography: Protocols, Algorithms and Source Code.1995

[8] Darrel Hankerson, Alfred J. Menezes, and Scott Vanstone. Guide to Elliptic Curve Cryptography.

[9] Hüseyin Bodur,Resul Kara. Application Secure SMS Encryption Using RSA Encryption Algorithm on Android Message.ResearchGate.2015

[10] Poupard, Pierre-Alain Fouque and Gwenaëlle Martinetand Guillaume.Attacking unbalanced RSA-CRT using SPA. ReserchGate.2003

[11] Deris Muhammad Faheem Mushtaq and Sapiee Jamel and Abdulkadir Hassan Disina and Mustafa Mat.A Survey on the Cryptographic Encryption Algorithms.ReserchGate.2017

[12] National Institute of Standards and Technology. "Announcing approval of the advanced encryption standard (AES)." Federal Information Processing Standards Publications (FIPS PUBS).2001

[13] Daemen Joan and Vincent Rijmen. AES proposal: Rijndael.1999

[14] Wadi, Ahmed Ghanim Wadday and Salim M. Study of WiMAX Based Communication Channel Effects on the Ciphered Image Using MAES Algorithm. ReserchGate.2018

[15] Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren .Handbook of elliptic and hyperelliptic curve cryptography.2006

[16] Steven D Galbraith. Mathematics Department, University of Auckland, New Zealand. CRYPTREC EX-3003-2020

[17] Rachidi, David Pointcheval and abderrahmane Nitaj and Tajjeeddine. Progress in Cryptology AFRICACRYPT.2016

[18] NIST.FIPS.186-4.

[19] Site Web MDPI ,[Consulté Le 10 juillet 2023], https://www.mdpi.com/2410-387X/6/2/25

[20] Abdelkrim K.Oudjida and Nicolas Chaillet Member IEEE. Radix-2^r Arithmetic for Multiplication by a Constant.CDTA.2015

[21] Bruce Schneier. Applied Cryptography

[22] Site Web Wikipédia. Signature numérique [Consulté Le 10 juillet 2023], https://fr.wikipedia.org/wiki/Signature_num%C3%A9rique

[23] Abdelkrim Kamel Oudjida, Abdelhakim Khouas and Ahmed Liacha and Fadila Nait-Abdesselam, Simple and Efficient Algorithms for Double Point Multiplication in Elliptic Curve Cryptography.CDTA