UNIVERSITÉ DU BURUNDI
GRENIER DU SAVOIR DU BURUNDI
Dépôt institutionnel officiel
https://repository.ub.edu.bi

Etude des courbes elliptiques appliquées aux algorithmes de signature numérique (ECDSA)

Habimana, Roger

2016-01
UB, FSI
https://repository.ub.edu.bi/handle/123456789/211

Document téléchargé depuis le **Grenier du Savoir du Burundi** — https://repository.ub.edu.bi

UNIVERSITE DU BURUNDI

FACULTE DES SCIENCES DE L'INGENIEUR

DEPARTEMENT DES TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

OPTION DE GENIE INFORMATIQUE

Mémoire de Master en Génie Informatique

ETUDE DES COURBES ELLIPTIQUES APPLIQUEES AUX ALGORITHMES DE SIGNATURE NUMERIQUE (ECDSA)

PAR :

HABIMANA ROGER

En vue d'obtention d'un diplôme de Master en Génie Informatique

Sous la Direction de :

Dr NDAYISABA Longin

Membres de Jury

Dr. SAHINGUVU William : Président

Dr. MUKESHIMANA Michèle : Vice- Président

Dr. NDIKUMAGENGE Jérémie : Secrétaire

Dr. NDAYISABA Longin : Directeur

Année Académique 2015-2016

DEDICACES

A mes chers Parents ;

A mes chers Frères et Sœurs ;

A mes chers oncles et tantes;

A tous mes chers cousins et cousines;

A Tous ceux qui me sont chers ;

HABIMANA Roger

REMERCIEMENTS

Le présent travail est le fruit des efforts consentis par plusieurs personnes tant physiques que morales. Louange à Dieu qui m'a donné la vie, la force, le courage et l'espoir pour accomplir ce travail et surmonter l'ensemble des difficultés.

J'exprime ma gratitude envers mon Directeur, Dr Longin NDAYISABA, pour ses qualités professionnelles, son encadrement, sa rigueur, ses directives, ses remarques constructives et sa disponibilité.

Je tiens également à remercier tous mes enseignants depuis l'école primaire et particulièrement au personnel de l'Université du Burundi, spécialement ceux du département des Technologies de l'Information et de la Communication, option Génie Informatique, pour leur contribution à notre formation de Baccalauréat jusqu'au Mastère.

Je saisis cette opportunité pour exprimer mes sincères remerciements envers mes camarades de classe durant ce cycle, mes amis et toute personne qui, de près ou loin, m'ont aidé et encouragé tout au long de ce travail.

Il m'est aussi agréable d'exprimer ma gratitude envers tous les membres du jury qui ont accepté de juger mon travail.

Pour finir, toute personne qui, de près ou de loin, par sa parole, son écrit, ses conseils et ses critiques, a contribué dans la réalisation de ce travail nous disons <<Merci à tous>>.

LISTE DES SIGLES ET ABREVIATIONS

AAA	:	**A**uthentification, **A**utorisation, **A**udit
AES	:	**A**lgorithm **E**ncryption **S**tandard
CIA	:	**C**onfidentiality **I**ntegrity **A**vailability
DES	:	**D**ata **E**ncryption **S**tandard
DHP	:	**D**effie-**H**ellman **P**roblem
DSA	:	**D**igital **S**ignature **A**lgorithm
ECC	:	**E**lliptic **C**urve **C**ryptography
ECDDHP	:	**E**lliptic **C**urve **D**ecision **D**effie-**H**ellman **P**rotocol
ECDH	:	**E**lliptic **C**urve **D**effie-**H**ellman
ECDHP	:	**E**lliptic **C**urve **D**effie-**H**ellman **P**roblem
ECDLP	:	**E**lliptic **C**urve **D**iscrete **L**ogarithm **P**roblem
ECDSA	:	**E**lliptic **C**urve **D**igital **S**ignature **A**lgorithm
ECIES	:	**E**lliptic **C**urve **I**ntegrated **E**ncryption
EEA	:	**E**xtended **E**uclide **A**lgorithm
PGCD	:	**P**lus **G**rand **C**ommun **D**iviseur
RSA	:	Rivest, Shamir et Adleman
DH	:	Deffie-Hellman
DAD	:	Disclosure, Alteration, Disruption
PIN	:	Personal Identification Number

LISTE DES FIGURES

Figure 1 : La sécurité existe à plusieurs niveaux (portée de l'information) 7
Figure 2 : Stades de vie de l'information 7
Figure 3 : Transmission des données vue par Shannon –Weaver en 1949 8
Figure 4 : Transmission des données vue par Shannon –Weaver en 1949 8
Figure 5 :Types de menace active 10
Figure 6 : Confidentiality, availability and integrity of data and services (CIA) 11
Figure 7 : Pentagone de confiance 12
Figure 8 : Parkerian Hexad 13
Figure 9 : McCumber Cube 13
Figure 10 : Protocole de chiffrement 15
Figure 11 : Schémas d'un cryptosystème 17
Figure 12: Chiffrement symétrique 19
Figure 13: Chiffrement asymétrique 20
Figure 14 :Confidentialité dans un système symétrique 25
Figure 15 : Confidentialité d'un système asymétrique 25
Figure 16: Confidentialité d'un système hybride 26
Figure 17 : Vérification de l'intégrité par fonction de hachage 26
Figure 18 : Authentification dans un système symétrique 27
Figure 19 : Authentification dans un système asymétrique 27
Figure 20 : Authentification par MAC et système symétrique 28
Figure 21: Authentification par MAC et fonction de hachage 28
Figure 22 : Authentification par signature (technique asymétrique) 29
Figure 23:Une Scytale 30
Figure 24 : Exemple d'une courbe elliptique E 64
Figure 25 : Droite passant par P et Q 65
Figure 26 : Description géométrique de l'addition de deux points 65
Figure 27 : Calcul du doublement d'un point 66

Figure 28: Description géométrique de l'addition de deux points de la courbe elliptique $P - P = O$ 67

Figure 29 : Génération de la signature 70

Figure 30: Vérification de la signature..... 71

LISTE DES TABLEAUX

Tableau 1 : Exemple d'utilisation de l'algorithme d'Euclide étendu 54
Tableau 2 : Cout d'inversion [30] 57
Tableau 3 : Détermination des points de la courbe 63
Tableau 4 : Correspondance entre les notations de DSA et ECDSA [33] 72
Tableau 5 : Correspondance entre $\boldsymbol{Fp}*$et le groupe E ($\boldsymbol{ZP}$) [33]. 72
Tableau 6 : Comparaison des temps de calcul. [1] 73

RESUME

L'évolution croissante des nouvelles technologies de l'information et de la communication est caractérisée par l'apparition des nouvelles technologies de sécurité en informatique et télécommunication. La transmission des données impose l'utilisation des méthodes de sécurité, afin que celles-ci ne soient pas interceptées par un autre individu qui n'est pas le destinateur ; mais aussi de pouvoir identifier l'origine du message par le destinateur. Les différentes classes d'algorithmes cryptographiques seront généralement revues, avec des exemples illustratifs, tout en montrant l'application de l'arithmétique modulaire, branche de mathématique, en cryptologie. En effet, l'implémentation de la RSA est étudiée en se basant sur les nombre premiers, la congruence modulaire et le problème basé sur la factorisation tandis que le problème du logarithme discret est utilisé dans la définition du protocole d'échange de clés de DH ainsi que dans l'élaboration de la signature numérique DSA.

Comme le montre le sujet, il existe une autre façon d'implémenter l'algorithme de signature en utilisant les courbes elliptiques. Le travail est à l'étude générale de la cryptographie en partant de la connaissance au préalable d'un système d'information, descriptions des menaces et les différents modèles de sécurité .La génération de la signature sera faite en utilisant les points de la courbe elliptique générés en utilisant la fonction

Cette nouvelle technologie de sécurité est très avantageuse comparativement aux autres car elle consomme moins de capacité de la mémoire avec une même sécurité. Elle trouve son application dans les systèmes embarqués, puces électroniques, cryptomonaies.

TABLE DES MATIERES

DEDICACES i

REMERCIEMENTS ii

LISTE DES SIGLES ET ABREVIATIONS iii

LISTE DES FIGURES iv

LISTE DES TABLEAUX vi

RESUME vii

TABLE DES MATIERES viii

AVANT –PROPOS 1

INTRODUCTION 2

CHAPITRE I : PRESENTATION DU SUJET ET GÉNÉRALITÉS SUR LACRYPTOGRAPHIE 3

I.1Présentation du sujet 3

I.1.1 Introduction 3

I.1.2 Généralités 3

I.1.3 Objectif du projet 4

I.1.4 Contexte du travail 4

I.1.5 Etat des lieux 5

I.1.6 Problématique 5

I.1.7 Hypothèse 5

I.2. Introduction à la sécurité informatique 5

I.2. 1. Contexte et problématique 5

I.2.2. Techniques de sécurité informatiques 6

I.2.3 Description des menaces des systèmes informatiques 8

I.2.4Description des différents modèles de sécurité 10

I.3. Introduction à la cryptographie 14

I.3.1.Vocabulaire de base 15

I.3.2.Notations 17

I.3.3 Les principaux concepts cryptographiques 18
I.4 Signature 20
I.4.1. Définition 20
I.4.2 Fonctionnement de la signature numérique 21
I.4.3 Types de signature 22
I.4.4 Fonctions de hachage 23
I.5 Protocoles cryptographiques 24
I.5.1 Confidentialité 24
I.5.2 Intégrité 26
I.5.3 Authentification 27
I.6 Historique 29
I.6.1 Cryptographie classique 29
I.6.2 Cryptographie moderne 31
I.6.3Principe de fonctionnement 33
Conclusion 34
CHAPITRE II. COMPLÉMENTS MATHÉMATIQUES 35
II.1 Introduction à la théorie des ensembles 35
II.1.0 Introduction 35
II.1.1.Notations et définitions 35
II.2. Nombres premiers 38
II.2.1 Vocabulaire de base 39
II.2.2 Algorithme de génération d'un nombre premier 39
II.3 Problème de factorisation et applications en cryptographie 41
II.3.1. Protocole standard d'échange d'une information 42
II.4 Chiffrement de RSA 43
II.4.1 Introduction 43
II.4.2 Principe de fonctionnement 44
II.4.3 Génération des clés 44

II.4.4Le chiffrement 44
II.4.5 Déchiffrement 44
II.5 Problème du logarithme discret et application en cryptographie 45
II.6 Etude de l'algorithme de signature numérique 45
II.6.1 Historique 45
II.6.2 Principe de fonctionnement 46
II.6.3 Algorithme de génération de clés 46
II.6.4 Algorithme de signature 46
II.6.5Algorithme de vérification de la signature 47
II.6.6 Exemple d'illustration 47
Conclusion 49
CHAPITRE III : L'ARITHMETIQUE MODULAIRE ORIENTE A LA CRYPTOGRAPHIE BASEE SUR LES COURBES ELLIPTIQUES(ECC) 50
III.1.Introduction 50
III.2. Addition modulaire 51
III.2.1. Algorithme basé sur la retenue sortante d'Omura 51
III.3. Inversion modulaire 52
III.3.1.Inversion modulaire via l'algorithme d'Euclide étendu 52
III.3.2. Inversion modulaire via le théorème de Fermat. 55
III.4 Multiplication modulaire 57
III.4.1 Introduction 57
III.4.2. Algorithme de Taylor avec mémorisation 58
III.4.3. Algorithme de Blakley 59
Conclusion 60
CHAPITRE IV ETUDE DES COURBES ELLIPTIQUES APPLIQUÉES À L'ALGORITHME DE SIGNATURE NUMÉRIQUE(ECDSA) 61
IV.0. Introduction 61
IV.1 Généralité sur Courbes elliptiques appliquées à la cryptographie 61
IV.1.1 Groupe 62

IV.2 Groupe basé sur les courbes elliptiques. 62
IV.3 Algorithme de génération de points 63
IV.4 Opération sur les courbes elliptiques 64
IV.4.1 Addition 64
IV.4.2 Calcul du Doublement d'un point 66
IV.4.3 Calcul de la négation d'un point 67
IV.4.4 Multiplication scalaire d'un point 67
IV.5. Application des courbes elliptiques à l'algorithme de signature numérique 68
IV.5.1 Introduction 68
IV.5.2 Principe de fonctionnement ECDSA 69
IV.5.3 Génération des points de la courbe. 69
IV.5.4 Génération de clé de signature 69
IV.5.5 Génération de signature 70
IV.5.6 Vérification de signature 71
IV.5.7Comparaison entre DSA et ECDSA 72
CONCLUSION 74
CONCLUSION ET RECOMMANDATIONS 75
REFERENCES BIBLIOGRAPHIQUES 76
ANNEXE 79

AVANT –PROPOS

Depuis longtemps les systèmes d'informations non informatisés d'une entreprise, de l'administration ou tout autre groupement sont protégés pour qu'ils soient utilisés par une personne ou un objet doté des droits d'accès. Les données étaient stockées sur des supports manuscrits .En effet, la transmission d'une données de l'émetteur jusqu'au destinataire s'effectue par l'intermédiaire d'une personne. L'un des moyens utilisés fréquemment pour garantir le non répudiation était la signature de l'émetteur, mais ce dernier n'avait pas de l'outil efficace de confirmation de l'arrivée du message, de sécurisation du support et du porteur du message.

Avec l'avènement des systèmes informatisés, les données sont stockées sur des supports électroniques. Ces systèmes sont vulnérables car ils ont été conçus par les humains, des défaillances matérielles ou logicielles, des erreurs lors de la conception ainsi que des catastrophes artificielles ou naturelles peuvent y apparaitre. En effet, la sécurité doit s'effectuer au niveau matériel et logiciels, mais aussi environnementale où sont installés ces systèmes. Plusieurs attaques circulent sur le réseau. Pour se protéger contre ces attaques, des méthodes et techniques cryptographiques ont été mises en place afin de réduire les risques.

La question de sécurité informatique est une question pertinente et évolutive suivant l'évolution des nouvelles technologies de l'information et de la communication déjà intégrés dans la gestion des données de n'importe quel secteur des Etats. Avec l'interconnexion de différents réseaux informatiques (réseau téléphonique, réseau d'ordinateurs, réseau de capteurs,…), tout utilisateur a l'ambition de garantir la sécurité de ses données. La divulgation des données d'une entreprise ou autres institutions, peut engendrer le disfonctionnement du système ciblé ou même sa destruction totale. En effet, toute entreprise a ses propres stratégies de marketing, sa propre vision de l'avenir, or il y a des concurrents partout, il faut que toute information qui y sorte, doive avoir une signature de celui qu'il l'a envoyé afin de garantir la non-répudiation. La sécurité informatique doit garantir la confidentialité, l'authentification, la disponibilité et le non répudiation avant, pendant et après l'échange des données.

INTRODUCTION

La sécurité informatique est un processus évolutifs au fur et à mesure de la croissance du nombre des utilisateurs et des technologies variés. Tous les mécanismes, outils et concepts de la sécurité informatique sont basé sur la cryptologie, étymologiquement qui est la science du cache au sens discret et sciences du secret au sens concret [1]. La cryptologie est une technique très ancienne née avec l'apparition de l'écriture et fut justifiée par le besoin de protéger tout message écrit afin d'éviter que l'ennemi ne puisse, en se l'appropriant, exploiter les renseignements qu'il contenait [14], elle est restée un art réservé aux militaires et aux espions. C'est au cours de ces vingt-cinq dernières années que la cryptologie est devenue une science basée sur les Mathématiques et l'informatique. Le développement des moyens de communication á l'échelle planétaire constitue aujourd'hui le moteur de la recherche publique en cryptologie. Les applications principales de la cryptologie sont bien sûr le chiffrement des messages pour en garantir la confidentialité, mais aussi l'authentification á distance, la signature de documents numériques, et le contrôle de l'intégrité des messages transmis sur un réseau informatique [15].

En regardant les besoins du marché, la cryptologie y trouve son intérêt et constitue un domaine scientifique en pleine activité. Elle intervient dans de multiples applications (communication sécurisée sur internet, cartes bancaires, monnaie électronique, authentification des documents électroniques, protection des droits d'auteur) et représente l'élément essentiel de la sécurisation du commerce électronique et du réseau Internet [16].

Au cours de ce projet, l'étude part de la généralité sur la cryptographie dans la sécurité informatique, tout en montrant l'interdépendance des différents cryptosystèmes avec les certains théorèmes mathématiques en insistant sur les courbes elliptiques appliquées aux algorithmes de signature numériques.

CHAPITRE I : PRESENTATION DU SUJET ET GÉNÉRALITÉS SUR LACRYPTOGRAPHIE

I.1Présentation du sujet

I.1.1 Introduction

L'évolution de la sécurité informatique suit l'évolution de la nouvelle technologie de l'information et de la communication. Pour pouvoir garantir la sécurité la des données transitant sur les réseaux au moment de leur échange, des différentes méthodes fondées sur des théories mathématiques ne cessent d'être développés. A part les cryptosystèmes classiques et modernes cités ci –haut, des recherches continuent afin de trouver des nouvelles méthodes puissantes et optimisant le temps, et l'espace. En effet, à part l'algorithme basé sur la factorisation, le problème du logarithme discret, il y a aussi des algorithmes utilisant les courbes elliptiques, algorithmes quantique, etc. Mon projet de recherche porte sur **<<étude des courbes elliptiques appliquées aux algorithmes de signature numérique (ECDSA)>>.**

I.1.2 Généralités

Les technologies de l'information et de la communication constituent le moteur incontournable de développement de développement. Aujourd'hui tant de services publiques ou privés sont entrés de s'investir dans les nouvelles technologies de l'information et de la communication. La sécurité des données transitant via le réseau demande une étude au préalable de l'environnement de stockage, définir le niveau d'accessibilité, le canal de communication, des outils de vérification de la l'arrivée du message,... La sécurité informatique implique le matériel utilisé, les logiciels, les utilisateurs et son environnement.

En effet, nous cherchons des moyens de sécurité parce que ces systèmes informatiques sont en face des menaces accidentelles ou intentionnelles. Différents cryptosystèmes ont été développés et sont intégrés dans nos matériels informatiques

sans le savoir afin de garder la confidentialité des données. Comme déjà j'ai déjà les différentes attaques, les différents modelés de sécurité, les cryptosystèmes existant, on voit que ces derniers sont fondées théories mathématiques tels que la théorie des nombres premiers, problème du logarithme discret, l'arithmétique modulaire, les courbes elliptiques, etc.

Ce projet vise l'étude des courbes elliptiques appliquées aux algorithmes de signature numériques. Ces derniers utilisent des clés de petite taille comparativement aux autres algorithmes ; et ont un même niveau de sécurité que sa variante DSA.

I.1.3 Objectif du projet

Globalement mon projet a pour l'objectif l'étude des courbes elliptiques appliquées aux algorithmes de signature numériques.

Spécifiquement, le projet a pour objectifs suivants

- Génération d'un nombre de 512 bits
- Génération de points de la courbe
- Stockage des points de la courbe
- Calcul de l'addition de deux points d'une courbe
- Calcul de double d'un point d'une courbe
- Multiplication d'un point d'une courbe par un scalaire
- Génération de clé de signature
- génération de la signature
- vérification de la signature

I.1.4 Contexte du travail

A la fin de ce travail, on obtiendra une application permettant de générer les points d'une courbe, de calculer les opérations d'addition de deux points, de double et de multiplication qui peut aider dans l'implémentation de signatures numériques basé

sur les courbes elliptiques. Le projet se limitera sur l'étude des elliptiques des courbes elliptiques appliquées aux algorithmes de signatures numériques.

I.1.5 Etat des lieux

La majorité des services publics ou privés de notre pays tend à dématérialiser l'ensemble de leur processus. Ils utilisent des moyens matériels et logiciels de sécurité standards qui ont besoin d'être amélioré. L'utilisation des appareils informatique de petite mémoire montre l'impact des courbes elliptiques en cryptographie.

I.1.6 Problématique

Recherche des algorithmes de cryptage plus puissant utilisant les clés de petite taille et facilitant les opérations de signature et de chiffrement.

I.1.7 Hypothèse

L'hypothèse à confirmer ou infirmer à la fin de notre travail est l'utilisation des courbes elliptiques lors de l'implémentation des algorithmes de signature numériques.

I.2. Introduction à la sécurité informatique

I.2. 1. Contexte et problématique

Actuellement, du fait de la migration du monde analogique vers numérique, les systèmes informatiques occupent une place prédominante dans les entreprises, dans les administrations et dans les activités quotidiennes de tous les jours des particuliers. Cette manifestation est activée, entre autres, par l'essor de l'internet qui attire du jour au jour beaucoup d'internautes par des nombreux avantages et la diversité des services rendus accessibles. En effet, ils peuvent bénéficier à moindre cout de moyens de communication rapide, partager des ressources de traitement et de stockage de grandes capacités(cloud computing),faciliter des échanges commerciaux et financiers(e-Commerce, e-Banking),fournir et utiliser de nombreux services en ligne(e-administration,e-Health,e-learning,etc),participer à des communautés virtuelles et à des réseaux sociaux, se divertir(e-gaming, e-Television, etc.) et plus généralement, partager et accéder à l'information. Par conséquent notre dépendance croissante aux systèmes informatiques dans divers aspect de la vie quotidienne et

leur omniprésence soulèvent inévitablement des questions relatives à leur sécurité et à la sécurité des données qui leur sont confiées.

La sécurité est une notion intégrant plusieurs acteurs et actions dont certains sont les suivants :

- assurer la disponibilité de service qui consiste à se protéger contre les pannes, le déni de service,…
- garder les données confidentielles et intègres
- protéger la vie privée des utilisateurs
- limiter les erreurs humaines
- protéger physiquement les individus(les incendies, surtension, dégradations manuelles,…)
- contrôler les accès et limiter les droits
- former les utilisateurs aux bons comportements
- garder les traces de ce qui a été fait
- définir un plan d'action en cas de problème
- et bien d'autres.

La partie suivante parle de techniques de sécurité d'un système informatique.

I.2.2. Techniques de sécurité informatiques

L'évolution des techniques de sécurité d'un système informatique tient compte de la portée de l'information. La sécurité doit être faite au niveau des données afin d'assurer la confidentialité de celle-ci, application en minimisant les erreurs au cours de leurs développement et surveillant des logiciels malveillants qui peuvent nuire d'autres logiciels d'application ou d'exploitation , des machines en définissant les modalités d' accès, des réseaux en définissant les utilisateurs et les règles de communication tout en surveillant les utilisateurs qui s'introduisent dans le réseau sans aucune autorisation.

La figure suivante nous montre les différents niveaux de sécurité en tenant compte de la portée de l'information [1].

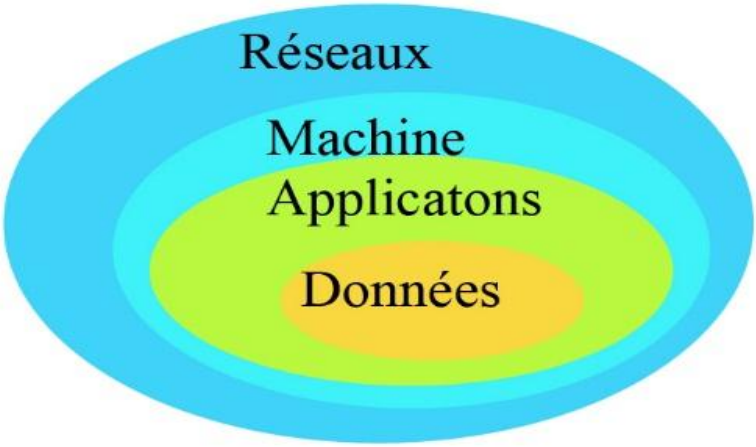


Figure 1 : La sécurité existe à plusieurs niveaux (portée de l'information)

Les techniques de sécurisation des données informatiques évoluent presque proportionnellement que les techniques de contournement de ces systèmes sécurisés. En effet, l'étude du contournement possible est simultanée à l'étude des protections. La tendance actuelle veut que les résultats découverts, tous domaines confondus, soient publiées .Dans le cadre de la sécurité informatique, cela permet de découvrir plus rapidement des faiblesses ou profit de certains techniques.

Toute information possède son origine et sa finalité. En effet, l'information doit avoir une source qui l'a traitée et l'a diffusée et sa destination où elle sera stockée au cours de sa utilisation et en fin elle est détruite.

La figure I.2 suivante montre les différents stades de vie de l'information

Figure 2 : Stades de vie de l'information

Selon Shannon, en 1948 puis en 1949 avec Weaver, le premier qui a défini les bases d'une transmission des données entre deux parties comme le montre la **figure I.3** suivante [1].

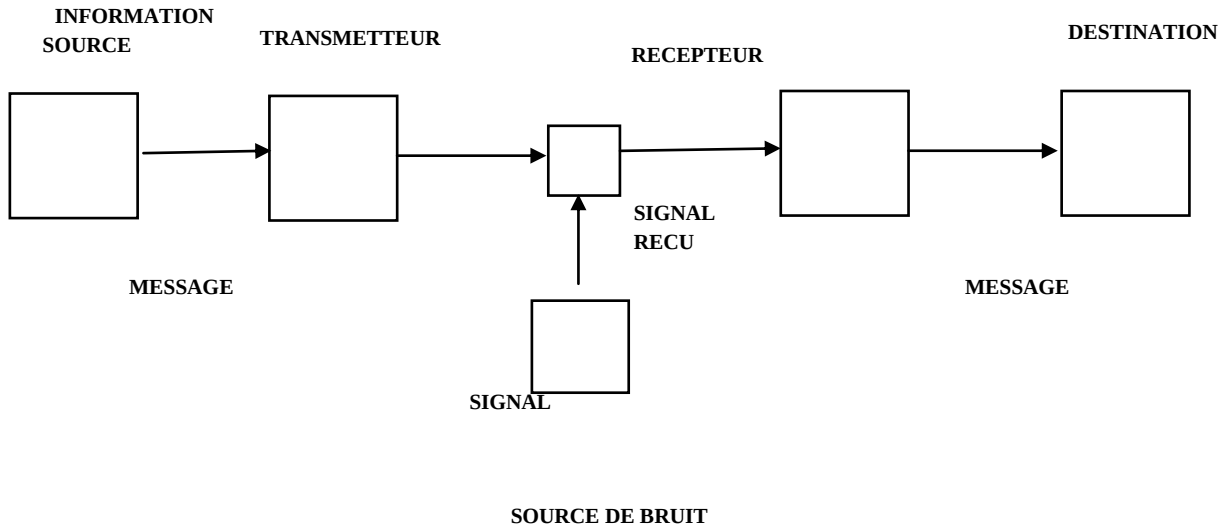


Figure 3 : Transmission des données vue par Shannon –Weaver en 1949

Au cours de cette étude, on a choisi un modèle simplifié utilisant 3 entités qui sont l'entité émettrice, l'entité réceptrice et l'ennemi comme le montre la figure 6 suivante.

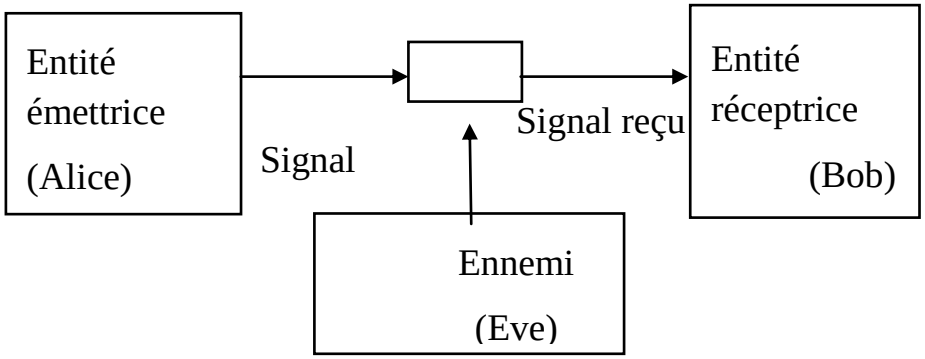


Figure 4 : Transmission des données vue par Shannon –Weaver en 1949

La partie suivante parle de la description des différents menaces de systèmes informatiques afin de savoir une technique a utilisé pour faire face à ces menaces.

I.2.3 Description des menaces des systèmes informatiques

Avec l'avènement croissant de l'interconnexion entre les systèmes informatiques de différentes organisations tant nationales qu'internationales; il y a des échanges de données provenant de différents utilisateurs qui ont des ambitions tout à fait différentes. Dans cette partie, l'étude ne s'attache pas aux utilisateurs de bonne foi

respectant les règles mises en place, mais s'attache à ceux qui lancent des menaces intentionnellement plutôt, sans toutefois oublier de failles du système causées par des défaillances matérielles ou logicielles. Les systèmes informatiques sont vulnérables, et subissent de menaces de différentes catégories. Parmi celles-ci, on trouve les menaces accidentelles et les menaces intentionnelles. Ces dernières regroupent les menaces passives et actives [1].

Les menaces accidentelles sont des menaces qu'on ne peut pas prévenir dues aux défaillances matérielles ou logicielles. Dans cette catégorie, il y a les bugs logiciels, les pannes matérielles, et autres défaillances "incontrôlables" tandis que les menaces intentionnelles quant à elles, apparaissent sous l'action d'un tiers désirant s'introduire dans la base de données et relever des informations qui lui sont utiles. Quand l'intrus s'introduit dans la base de données et dérobe les informations par audit, il est difficile de le détecter car il ne modifie pas les fichiers, il fait une surveillance afin de collecter des informations qui lui sont utile pour son auto développement intellectuelles ou économique. Ils ne provoquent pas aucune altération du système, on dit que c'est une attaque passive.

Dans le cas d'une attaque active, il est facile de détecter l'intrus, mais la détection peut être faite trop tard par rapport au moment d'intrusion. L'intrus aura déjà modifié volontairement les fichiers ou provoquer le disfonctionnement du système. Les menaces actives appartiennent principalement à quatre catégories comme le montre la **figure I.5** suivante [1]:

– Interruption = problème lié à la disponibilité des données

– Interception = problème lié à la confidentialité des données

– Modification = problème lié à l'intégrité des données

– Fabrication = problème lié à l'authenticité des données

En bref toute acte sur un système dont l'intention est de nuire l'une des propriétés de la sécurité est qualifié de malveillant et constitue, de ce fait, une attaque sur ce système. Ces attaques élémentaires peuvent intervenir à plusieurs niveaux d'abstraction du système. En effet, ces niveaux peuvent être logiciels,précisément sur leur état et structure, matériels surtout les pilotes de ceux-ci, canaux de communication par lesquels il interagit avec d'autres systèmes, son environnement, par la mise en œuvre d'attaque des canaux auxiliaires. La description de ces niveaux est précisée dans [17] où on ne montre comment est élaboré le modèle d'attaques.

La figure suivante montre les types de menaces actives.

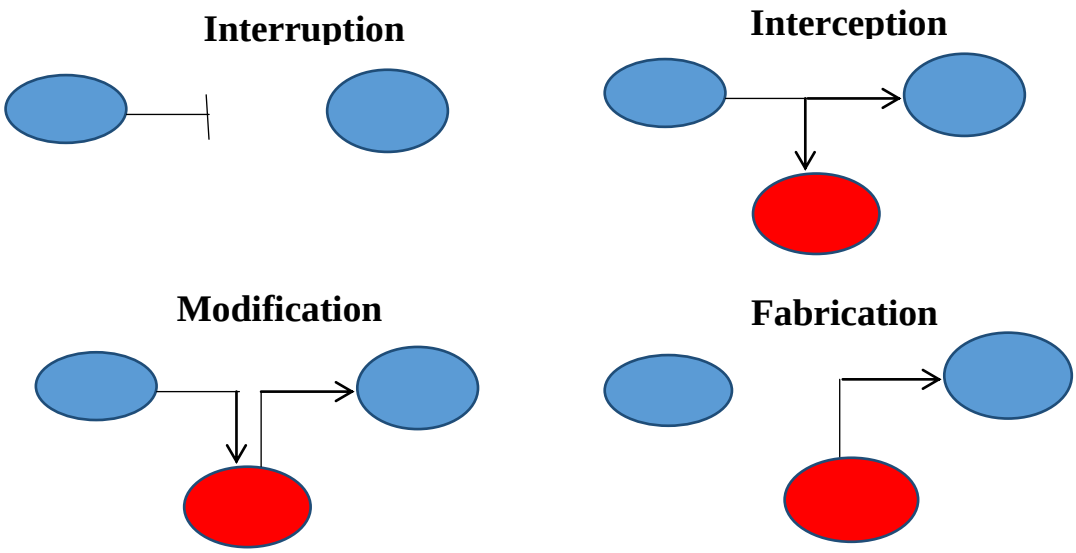


Figure 5 :Types de menace active

Les auteurs de ces attaques sont notamment les hackers (agissant souvent par défi personnel), les concurrents industriels (vol d'informations concernant la stratégie de l'entreprise ou la conception de projets), les espions, la presse ou encore les agences nationales.

La partie suivante montre des différents modèles de sécurité afin de faire face aux menaces citées ci-haute.

I.2.4Description des différents modèles de sécurité

Cette partie fait la description détaillée des différents modèles de sécurité en se basant aux différentes modifications qu'a subies le concept de sécurité. Parmi ces modèles il y en a la CIA mis en place en 1987, le protocole AAA, Parkerian Hexad, McCumber Cube (1991),la sécurité en parallèle et la sécurité en série [1].

- Le triangleCIA, crée en 1987, est le modèle inspirateur, présentant les grands axes de la sécurité. La plupart des autres modèles utilisent cette représentation en tant que modèle base.

La figure suivante nous montre le triangle CIA centré sur l'intégrité, la confidentialité et la disponibilité de données et services.

Figure 6 : Confidentiality, availability and integrity of data and services (CIA)

Le triangle opposé existe également. Il porte le nom de DAD, pour Disclosure, Alteration, Disruption.

On peut définir les différents termes employés comme suit :

- Confidentialité : l'information reste entre les entités communicantes
- Intégrité : l'information n'a pas été modifiée entre sa création et son traitement
- Disponibilité : l'information est toujours accessible et ne peut être bloquée/perdue

❖ Le protocole AAA est venu pour compléter le modèle CIA afin de vérifier l'identité des parties communicantes. Le contrôle d'accès se fait en 4 étapes :

1. Identification : Qui êtes-vous ?
2. Authentification : Prouvez-le !
3. Autorisation : Avez-vous les droits requis ?
4. Accounting/Audit : Qu'avez-vous fait ?

On parle du protocole AAA (les deux premières étapes sont fusionnées).Dans certaines situations, on scindera la dernière étape. On parlera d'Accounting lorsque le fait de comptabiliser des faits sera demandé, et d'Audit lorsque des résultats plus globaux devront être étudiés.

Notons également que l'authentification, visant à prouver l'identité d'un individu peut se faire de plusieurs manières :

- Ce que vous savez (mot de passe, code PIN, etc.)
- Ce que vous avez (carte magnétique, lecteur de carte, etc.)
- Ce que vous êtes (empreintes digitales, réseau rétinien, etc.)

L'authentification forte résultera de la combinaison de 2 de ces facteurs.

Le modèle exemplaire est le pentagone de confiance, initié par Piscitello en 2006, il précise la notion d'accès à un système. Indépendamment des notions définies dans le triangle CIA, le modèle de Piscitello précise la confiance que peut/doit avoir l'utilisateur en présence d'un système informatisé [1].

La figure suivante nous montre le pentagone de confiance

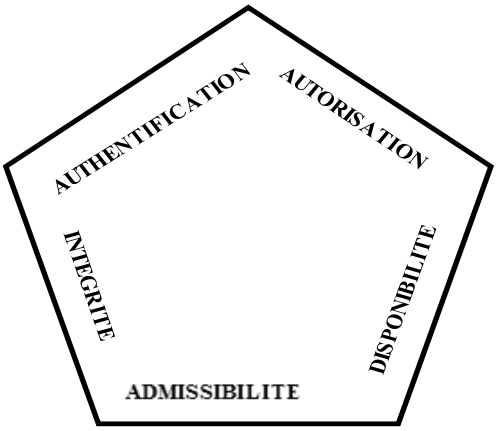


Figure 7 : Pentagone de confiance

L'authentification, la disponibilité, l'autorisation, l'intégrité et l'admissibilité nous garantissent un plus niveau de sécurité de la machine cible, donc plus de confiance.

- Le parkerian Hexad est un ensemble d'éléments de sécurité d'information proposé par Donn B.Parker en 1998.Il a ajouté 3 autres attributs de la sécurité appart les trois qui était classique. Les attributs de parkerian Hexad sont les suivants : confidentialité, possession ou le contrôle, intégrité, authenticité, disponibilité, utilité comme le montre la figure 8

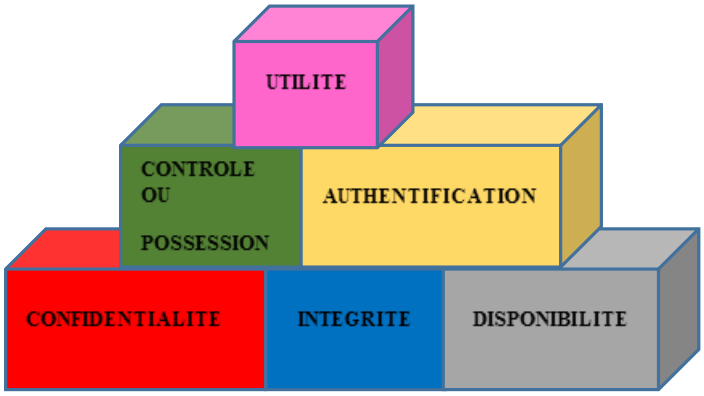


Figure 8 : Parkerian Hexad

McCumber Cube (1991) a les trois piliers de la sécurité (CIA), mais deux autres dimensions apparaissent :

– L'état des données : le stockage, la transmission, l'exécution

– Les méthodes : les principes et règles à adopter pour atteindre le niveau de sécurité souhaité [1].

La figure 9 ci-dessous montre McCumber Cube

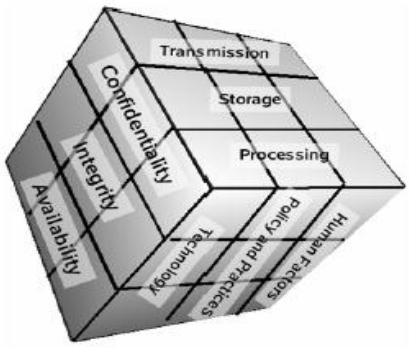


Figure 9 : McCumber Cube

- La sécurité en parallèle combine plusieurs mécanismes de sécurité protégeant un système possèdent le même rôle. Dans ce cas, le niveau de protection du système est équivalent à celui du mécanisme le moins sûr.

En tant qu'exemple, citons un ordinateur portable que l'on peut déverrouiller par mot de passe, dongle ou empreinte digitale.

La sécurité en série utilise plusieurs mécanismes de sécurité pour protéger un système et ont des rôles différents. On parlera de « défense en profondeur ».

Citons par exemple le réseau d'une entreprise où :

– Le réseau est sécurisé par un FW hardware,

– Les liaisons entre machines sont protégées,

– Les machines individuelles sont munies d'un FW software,

– Les accès aux machines se font par empreinte biométrique,

– Le logiciel à utiliser est accessible par mot de passe,

– etc.

I.3. Introduction à la cryptographie

La cryptographie est constituée de deux disciplines complémentaires qui sont la cryptographie et la cryptanalyse. La cryptographie est l'art de cacher l'information. Elle désigne l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre incompréhensibles. Le fait de coder un message de telle façon à le rendre secret s'appelle chiffrement, la cryptographie répond aux besoins suivants :

- **Confidentialité :** garantir que les données échangées ne sont dévoilées qu'aux personnes voulues.
- **Intégrité :** garantir que les données échangées ne sont pas altérées intentionnellement ou non.
- **Authenticité :** prouver l'identité d'une personne ou l'origine d'un document.
- **Signature :** permettre à une personne de signer informatiquement un document sans qu'elle puisse le renier par la suite.

La cryptographie utilise des concepts issus de nombreux domaines (Informatique, Mathématiques, Electronique). Toutefois, les techniques évoluent et trouvent aujourd'hui régulièrement racine dans d'autres branches (Biologie, Physique, etc.).Pour mieux comprendre d'une façon générale la cryptographie, revenons sur la définition de certains termes couramment utilisés en cryptologie comme le montre la Figure 10.

I.3.1.Vocabulaire de base

La compréhension des principes cryptographiques exige la compréhension de certains mots clés au préalable. La figure10 suivante nous montre différentes modifications qu'a subies une information de la source à la destination, à condition qu'une information envoyée soit identique à celle reçue.

La figure suivante nous montre les différentes étapes de chiffrement avant l'envoi du message et de déchiffrement après par le récepteur du message

L'émetteur doit avoir une clé de chiffrement et le récepteur une clé de déchiffrement. On va définir une clé déchiffrement, chiffrement, texte clair, texte chiffré, cryptanalyse, cryptographie.

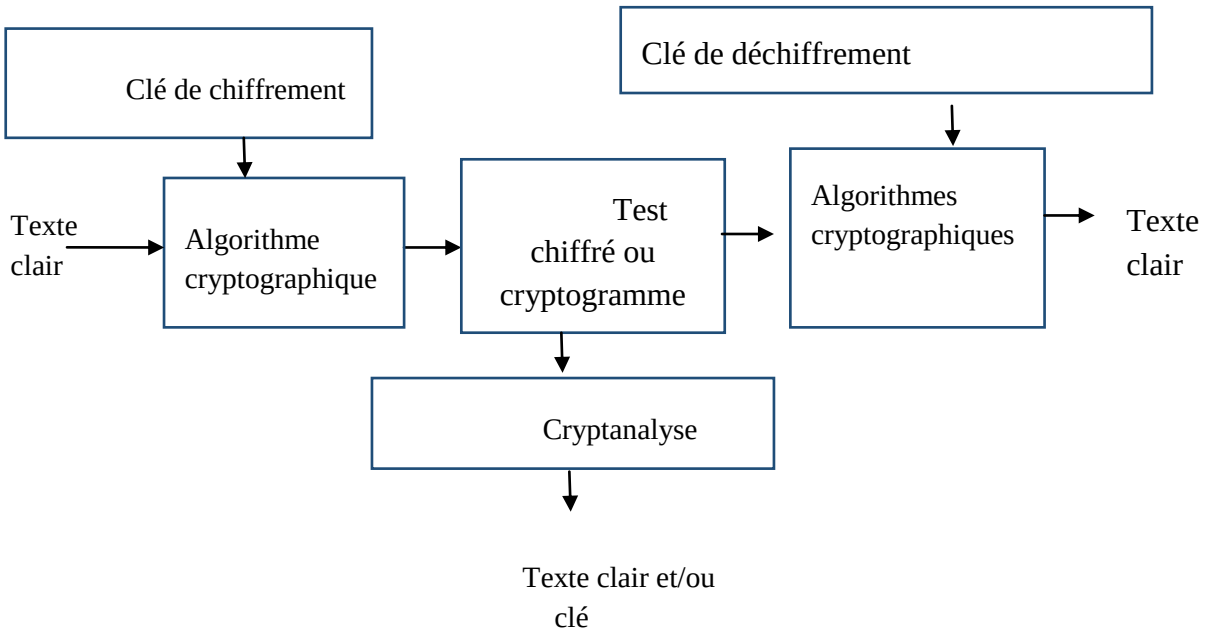


Figure 10 : Protocole de chiffrement

Cryptologie est une science mathématique comportant deux branches qui sont la cryptographie et la cryptanalyse.

Cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné.

Chiffrement consiste à transformer une donnée (texte, message, ...) afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et

celui qui en est le destinataire. La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de **déchiffrement.**

Texte chiffré appelé également **cryptogramme**, le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair.

Clef est un paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement.

Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations, contrairement aux algorithmes asymétriques, elle diffère pour les deux opérations.

Cryptanalyse : Opposée à la cryptographie, elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés.

En effet, On distingue habituellement quatre méthodes de cryptanalyse :

- Une attaque sur texte chiffré seulement consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés ;
- Une attaque sur texte clair connu consiste à retrouver la clé de déchirement à partir d'un ou plusieurs textes chiffrés, connaissant le texte en clair correspondant;
- Une attaque sur texte clair choisi consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de les générer à partir de textes en clair ;
- Une attaque sur texte chiffré choisi consiste à retrouver la clé de déchiffrement à partir d'un ou plusieurs textes chiffrés, l'attaquant ayant la possibilité de les générer à partir de textes en clair

Cryptosystèmes : Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné. Les cryptosystèmes à clé secrète est sur l'échange au préalable d'une clé secrète commune de taille variable ; elle peut être sous la forme d'un chiffre, d'une lettre, d'un livre… Le cryptage et le décryptage s'effectue grâce à cette clé.

Aujourd'hui de nombreux cryptosystèmes à clé secrète sont utilisés et sont basés sur la permutation et des transformations de blocks et n'utilise pas de l'arithmétique modulaire.

L'échange de cette clé secrète était l'inconvénient de ces cryptosystèmes à clé secrète. D'où est apparu en 1970, inventé par W.Diffie et M.E.Hellman, un moyen fiable d'échange de clé de façon publique [8].

C'est le début de la cryptographie à clé publique. Elle permet l'échange des données d'une façon sécurisée. Cette absence d'échange au préalable de la clé qui dégage une nette différence entre la cryptographie à clé publique et la cryptographie à clé secrète.

Voici un schéma I.11 montrant un cryptosystème

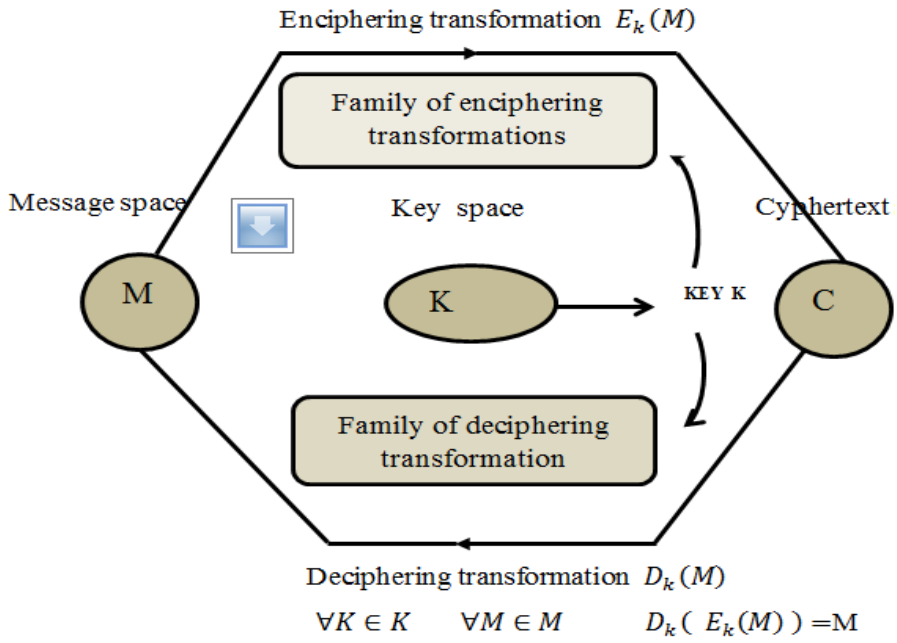


Figure 11 : Schémas d'un cryptosystème

L'algorithme est en réalité formé de trois algorithmes :

– un algorithme de génération de clés K,

– un algorithme de chiffrement M, et

– un algorithme déchiffrement C.

I.3.2.Notations

En cryptographie, la propriété de base est que **M = D(E(M))** où **M** représente le texte clair, **C** représente texte chiffré, **K** représente la clé (dans le cas d'un algorithme à clé symétrique), $\mathbf{E_k}$ et $\mathbf{D_k}$ représentent respectivement l'algorithme symétrique et l'algorithme asymétriques, **E(x)**est la fonction de chiffrement et

D(x)est la fonction de déchiffrement. Ainsi, avec un algorithme à clef symétrique, **M = D(C)** si **C = E(M)**

Selon le principe de Kirchhoff, la sécurité du chiffre ne doit pas dépendre de ce qui ne peut pas être facilement changé. Donc le secret réside au niveau de la clé laquelle nous permet de retrouver le texte clair à partir du texte chiffré ; mais l'algorithme reste accessible

Les algorithmes sont subdivisés en deux catégories qui sont les algorithmes secrets et les algorithmes publiés. Les algorithmes secrets sont utilisés par un plus petit nombre d'utilisateurs. Et comme souvent dans ce cas, moins il y a de monde l'utilisant, moins il y a d'intérêts à le casser. Ces algorithmes sont rarement distribués par-delà des frontières, afin de garder un nombre d'utilisateurs restreint. Sa cryptanalyse est base sur le secret de la clé, elle exige un mécanisme de récupération pour pouvoir l'entièreté de l'algorithme.

Les algorithmes publiés donnent le droit à n'importe qui d'être explorés. Ainsi, les failles (laissées intentionnellement ou non par les concepteurs) peuvent être plus facilement découvertes. La sécurité en est donc améliorée.

- Comme la publication est autorisée, il n'est pas nécessaire de chercher à protéger le code contre le reverse-engineering.
- Cette publication permet d'étendre les travaux sur l'algorithme au niveau mondial. Toute une série d'implémentations logicielles peuvent donc être réalisées.
- Tout le monde utilise la même version publique ce qui permet une standardisation générale. En conséquence, on préférera les algorithmes publiés, souvent plus sûrs pour les raisons explicitées ci-dessus.

I.3.3 Les principaux concepts cryptographiques

Etant donné que les cryptosystème sont subdivisés en deux catégories, dans cette section on va les généraliser en se basant sur leurs caractéristiques.

Les cryptosystèmes à clé symétrique sont caractérisés par les clés qui sont identiques **($K_E = K_D = K$)** entre les deux interlocuteurs mais la clé doit rester secrète. Les algorithmes les plus répandus sont le DES [6], triple DES [6], AES [7]...

La génération des clés est choisie aléatoirement dans l'espace des clés, ces algorithmes sont basés sur des opérations de transposition et de substitution des bits

du texte clair en fonction de la clé, la taille des clés est souvent de l'ordre de 128 bits. Le DES en utilise 56, mais l'AES peut aller Jusqu'à 256

L'avantage principal de ce mode de chiffrement est sa rapidité. Le principal désavantage réside dans la distribution des clés : pour une meilleure sécurité, on préfèrera l'échange manuel. Malheureusement, pour de grands systèmes, le nombre de clés peut devenir conséquent. C'est pourquoi on utilisera souvent des échanges sécurisés pour transmettre les clés. En effet, pour un système à N utilisateurs, il y aura N. (N − 1)/2 paires de clés. La figure suivante nous montre le chiffrement symétrique

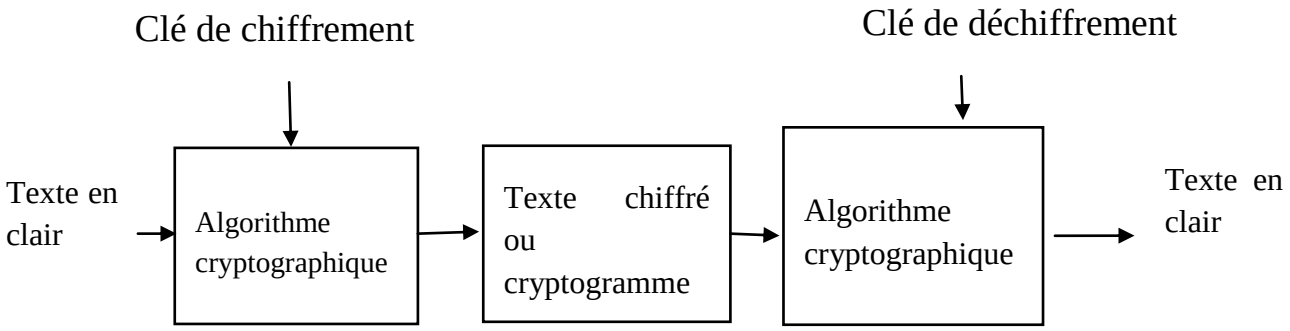


Figure 12: Chiffrement symétrique

Le cryptosystème à clé publique est caractérisé par deux clés qui sont la clé publique PK (symbolisée par la clé verticale) et une clé privée secrète SK (symbolisée par la clé horizontale).La propriété qui lui est propre ce que la connaissance de PK ne permet pas de déduire SK, DSK (EPK(M)) = M. L'algorithme de cryptographie asymétrique le plus connu est le RSA. Le principe de ce genre d'algorithme est qu'il s'agit d'une fonction unidirectionnelle. Une telle fonction à la particularité d'être facile à calculer dans un sens, mais difficile même impossible dans le sens inverse. La seule manière de pouvoir réaliser le calcul inverse est de connaître une trappe. Une trappe pourrait par exemple être une faille dans le générateur de clés. Cette faille peut être soit intentionnelle de la part du concepteur ou accidentelle.

Ces algorithmes se basent sur des concepts mathématiques tels que l'exponentiation de grands nombres premiers (RSA), le problème des logarithmes discrets (ElGamal), ou encore le problème du sac à dos (Merkle-Hellman)[11].La taille des clés s'étend de 512 bits à 2048 bits en standard.

Du point de vue de la taille des clés, il y a une différence au niveau de la sécurisation. Plus la clé est de grande taille, plus est plus sécuritaire. Dans le cas du RSA, une clé de 512 bits n'est plus sûre au sens "militaire" du terme, mais est toujours utilisable de particulier à particulier.

Du point de vie de la performance entre l'algorithme à clé symétrique et celui à clé asymétrique est que le chiffrement par voie asymétrique est environ 1000 fois plus lent que le chiffrement symétrique.

Cependant, à l'inverse du chiffrement symétrique où le nombre de clés est le problème majeur, ici, seules n paires sont nécessaires. En effet, chaque utilisateur possède une paire (SK, PK) et tous les transferts de message ont lieu avec ces clés.

La distribution des clés est grandement facilitée car l'échange de clés secrètes n'est plus nécessaire. Chaque utilisateur conserve sa clé secrète sans jamais la divulguer. Seule la clé publique devra être distribuée.

La figure I.13 ci-dessus montre le fonctionnement d'un algorithme à clé publique.

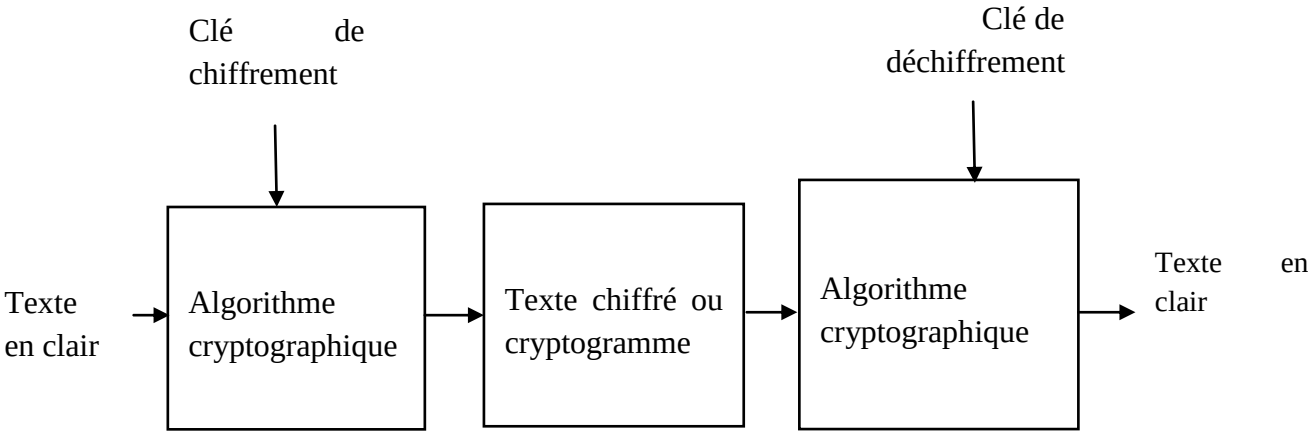


Figure 13: Chiffrement asymétrique

I.4 Signature

I.4.1. Définition

Cette partie nous montre en compréhension la définition de la signature manuscrite et signature numérique, principe de fonctionnement de cette dernière et en fin la fonction de hachage.

Signature, selon Larousse, est toute marque distinctive et personnelle manuscrite, permettant d'individualiser, sans doute possible, son auteur et traduisant la volonté non équivoque de consentir à un acte.

La signature numérique est un mécanisme qui permet d'authentifier un message. Donc une information codée permettant d'authentifier l'émetteur d'un message électronique.

La signature d'un document utilise à la fois la cryptographie asymétrique et lesfonctions de hachage. C'est en effet par l'association de ces deux techniques que nous pouvons obtenir les cinq caractéristiques d'une signature (authentique, infalsifiable, non réutilisable, inaltérable, irrévocable).

Comparativement à la signature manuscrite qui peut être facilement imitée, la signature numérique est pratiquement infalsifiable. De plus, elle atteste du contenu des informations, ainsi que de l'identification du signataire.

I.4.2 Fonctionnement de la signature numérique

Le mécanisme de chiffrement permet d'assurer la fonction de confidentialité. La signature électronique est un autre mécanisme qui permet d'assurer les fonctions d'authentification et d'intégrité. Elle est utilisée en particulier dans la messagerie électronique. Pour générer une signature électronique, il faut dans un premier temps utiliser une fonction de hachage sur le texte, dont le résultat est une suite de bits de taille fixe, bien inférieure à la taille du texte initial. Cette suite de bits est aussi appelée condensé ou empreinte, car la fonction de hachage est telle que si un bit du texte d'origine est modifié, le résultat de la fonction sera, avec de très fortes probabilités, différent. MD5 (*Message Digest*) et SHA (*Secure Hash Algorithm*) sont parmi les plus connues.

On supposé qu'A (par exemple Alice) souhaite envoyer un document signé à B (Par exemple Bob).

1. Tout d'abord, A va générer l'empreinte du document au moyen d'une fonction de hachage.

2. Puis, elle crypte cette empreinte avec sa clé privée.

3. Elle obtient ainsi la signature de son document. Elle envoie donc ces deux éléments à B.

4. Pour vérifier la validité du document, B doit tout d'abord déchiffrer la signature en utilisant la clé publique de A. Si cela ne fonctionne pas, c'est que le document n'a pas été envoyé par A.

5. Ensuite, B génère l'empreinte du document qu'il a reçu, en utilisant la même fonction de hachage qu'A (On supposera qu'ils suivent un protocole établi au préalable).

6. Puis, il compare l'empreinte générée et celle issue de la signature.

7. Si les deux empreintes sont identiques, la signature est validée. Nous sommes donc sûr que :

- C'est A qui a envoyé le document,
- Le document n'a pas été modifié depuis qu'A l'a signé.

8. Dans le cas contraire, cela peut signifier que :

- Le document a été modifié depuis sa signature par A,
- Ce n'est pas ce document qu'A a signé

I.4.3 Types de signature

Une **signature** est une marque permettant d'identifier l'auteur d'un document, d'une œuvre ou la cause d'un phénomène. Il existe une signature numérique et électronique. La différence entre les deux réside au niveau de la technologie utilisé. Mais avec la signature électronique, l'intégrité des données n'est pas assurée. Il existe 4 clés à tenir en compter pour garantir la signature qui sont l'authenticité, identité, l'intégrité et l'authentification.

Les types de signature qu'on peut trouver sont :

- **Signature attachée :** C'est le type de signature le plus courant, la signature étant apposée en entête ou à la suite des données à signer (la position dépend de la structure du document, de son type et des accords d'échanges éventuels).
- **Signature détachée :** Pour la signature détachée, l'élément de signature sera enregistré dans un fichier indépendant des données à signer.
- **Signature enveloppante :** Pour effectuer une signature enveloppante, il faut ajouter les données à signer dans un élément Object qui sera inclus à la signature. Cet élément doit posséder une référence propre.

Ces différents types de signature peuvent être cascadés et utilisés simultanément.

On parle alors de multi-signature où l'on distingue :

- **La cosignature :** elle consiste à signer le document (ou l'élément) par plusieurs personnes. Les signatures se trouveront toutes au même niveau. Cela

se fait très simplement en effectuant une signature enveloppée du document ou de l'élément déjà signé. Cette signature viendra s'ajouter à la précédente. On obtient ainsi une cosignataire.

- **La sur-signature ou contre-signature :** elle consiste à signer une signature existante. Cela revient à réaliser une signature de l'élément représentant la signature à contresigner et d'inclure le bloc de la signature ainsi générée dans la signature à contresigner. Afin de pouvoir contresigner une signature, il est nécessaire d'ajouter une identification sur la valeur de la signature qui servira de référence pour la contre-signature.

I.4.4 Fonctions de hachage

Une fonction de hachage est une fonction qui transforme un long message en un résumé court, de taille fixe. L'image d'un message par une fonction de hachage s'appelle le condensé du message, l'empreinte du message, le résumé du message ou encore le message haché. Une fonction de hachage doit posséder deux qualités indispensables :

- Résistance à la détermination d'un pré image, ce qui signifie qu'il doit être impossible en pratique, à partir d'un résumé m, de trouver un message M ayant origine de ce résumé, c'est-à-dire tel que m=h(M).
- Résistance aux collisions, ce qui signifie qu'il est impossible en pratique de construire deux messages M1 et M2 ayant le même résumé : h(M1)=h(M2).

Il est conseillé d’utiliser une fonction de hachage de grande taille. Une taille de 128 bits est maintenant considérée comme trop courte et il est conseillé de prendre au moins 160 bits. Il existe de nombreuses fonctions de hachage.

- SHA : est la fonction de hachage utilisée par SHS, la norme du gouvernement Américain pour le hachage, SHA-1 est une amélioration de SHA qui fournit en sortie un bloc de160 bits.
- MD5 : donne une empreinte sur 128 bits.
- SHA-224, SHA-256, SHA-384, SHA-512 : Ce sont des améliorations de

  SHA-1. Les trois dernières fonctions fournissent une résistance aux attaques brutales comparables à la résistance des diverses versions de AES (SHA-256 à mettre en rapport avec AES-128, SHA-384 à mettre en rapport avec AES-192, SHA-512 à mettre en rapport avec AES-256). Elles ont été retenues par le projet NESSIE.

- Whirlpool : Cette fonction a une sortie de 512 bits. Whirlpool a été retenue par le projet NESSIE. Ce dernier avait pour mission d'évaluer l'AES américain et permettre le développement de nouvelles primitives cryptographiques. Il s'agit des fonctions de chiffrement symétrique et asymétrique, fonctions de hachage, fonctions de signature électronique à destination des industriels européens.

Les cryptosystèmes sont basés sur des problèmes mathématiques difficiles à résoudre tel que la factorisation et le problème de logarithme discret. Le principe d'un protocole cryptographique est qu'il doit être simple de chiffrer un message, mais qu'il doit être très difficile voire impossible à déchiffrer sans posséder la clé du système.

Il s'agit de la troisième grande famille d'algorithmes utilisés en cryptographie. Le principe est qu'un message clair de longueur quelconque doit être transformé en un message de longueur fixe inférieure à celle de départ. Le message réduit portera le nom de "Haché" ou de "Condensé". L'intérêt est d'utiliser ce condensé comme empreinte digitale du message original afin que ce dernier soit identifié de manière univoque. Deux caractéristiques (théoriques) importantes sont les suivantes :

1. Ce sont des fonctions unidirectionnelles : à partir de H(M) il est impossible de retrouver M.

2. Ce sont des fonctions sans collisions : à partir de H(M) et M il est impossible de trouver $M' \neq M$ tel que H (M') = H(M).

I.5 Protocoles cryptographiques

Dès que plusieurs entités sont impliquées dans un échange de messages sécurisés, des règles doivent déterminer l'ensemble des opérations cryptographiques à réaliser, leur séquence, afin de sécuriser la communication.

C'est ce que l'on appelle les protocoles cryptographiques.

Lorsque l'on parle de "sécuriser un échange", on souhaite prêter attention aux 3 services suivants : la confidentialité, l'intégrité et l'authentification [1].

Signalons la distinction entre "services" (confidentialité, intégrité, etc.) et "mécanismes" (les moyens utilisés : chiffrement, signature, hachage, etc.).

I.5.1 Confidentialité

Elle est amenée par le chiffrement du message. Dans le cas de systèmes à clés symétriques, la même clé est utilisée pour $E_K(M)$ et $D_K(C)$. Ce type de chiffrement nécessite un échange sûr préalable de la clé K entre les entités A et B. La figure 14

ci-dessous montre les processus de chiffrement symétrique du message afin de créer la confidentialité.

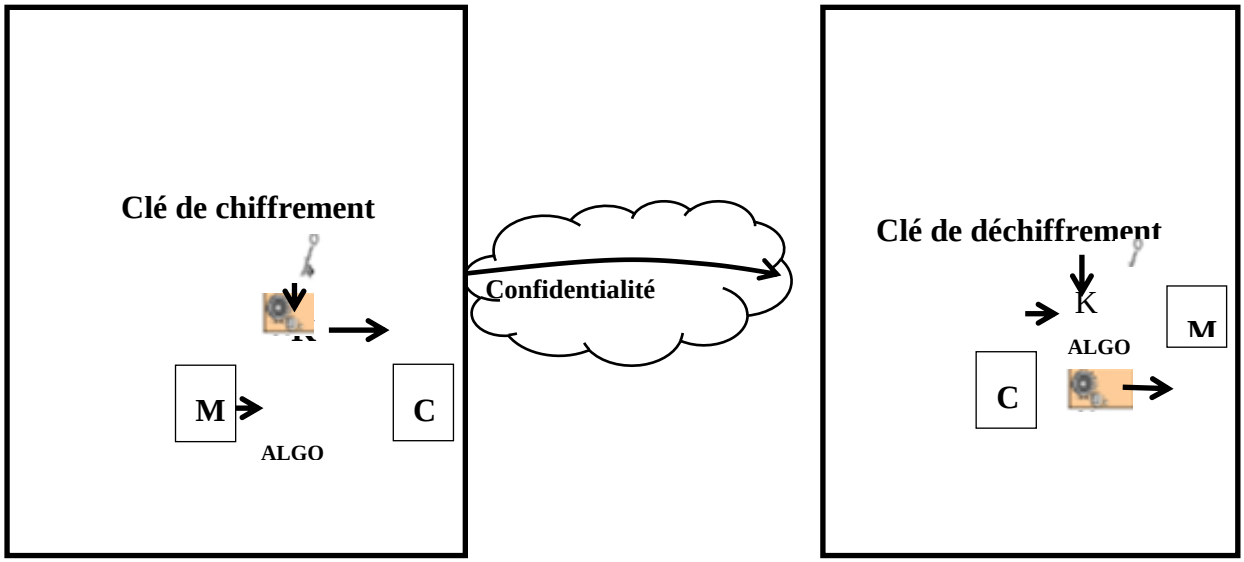


Figure 14 :Confidentialité dans un système symétrique

Comme se dit précédemment, à l'aide d'un cryptosystème asymétrique, cet échange préalable n'est pas nécessaire. Chaque entité possède sa propre paire de clés. On aura donc la paire PKA, SKA pour l'entité A et la paire PKB, SKB pour l'entité B.

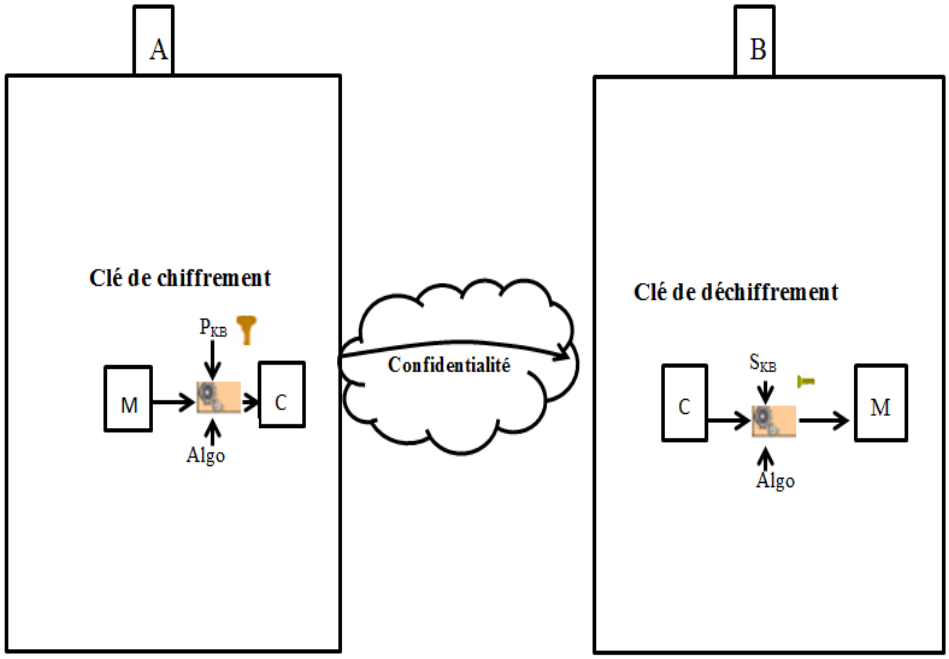


Figure 15 : Confidentialité d'un système asymétrique

Appart ces deux systèmes, il existe également un système appelé "hybride" (figure 2.7), reposant comme son nom l'indique sur les deux systèmes précédents. Par l'intermédiaire du système à clé publique, on sécurise l'échange de la clé K. Ensuite, les deux parties ayant acquis de manière sécurisée cette clé de chiffrement K3, on utilisera le système à clé symétrique pour chiffrer le message.

Comme le montre la figure 16 suivante

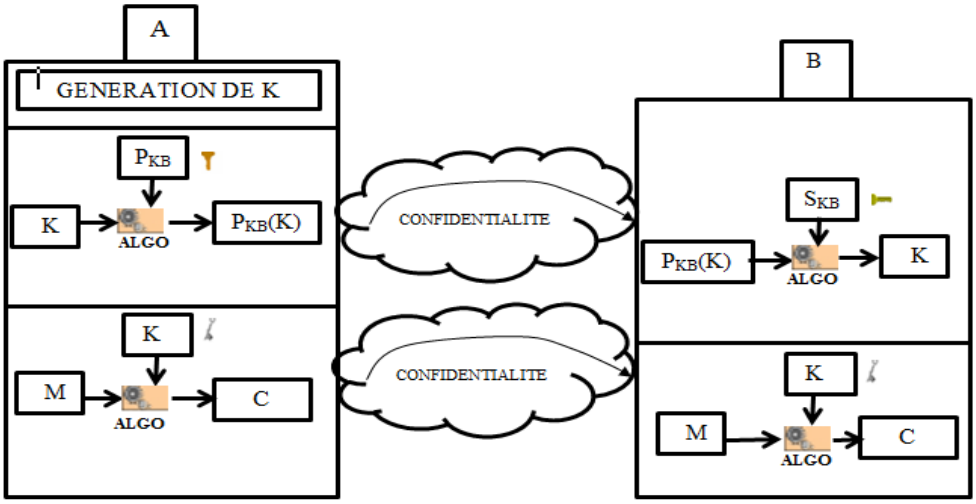


Figure 16: Confidentialité d'un système hybride

I.5.2 Intégrité

Il faut, ici, vérifier si le message n'a pas subi de modification durant la communication. C'est ici qu'interviennent les fonctions de hachage. Voici la figure I.17 montrant la vérification de l'intégrité par la fonction de hachage

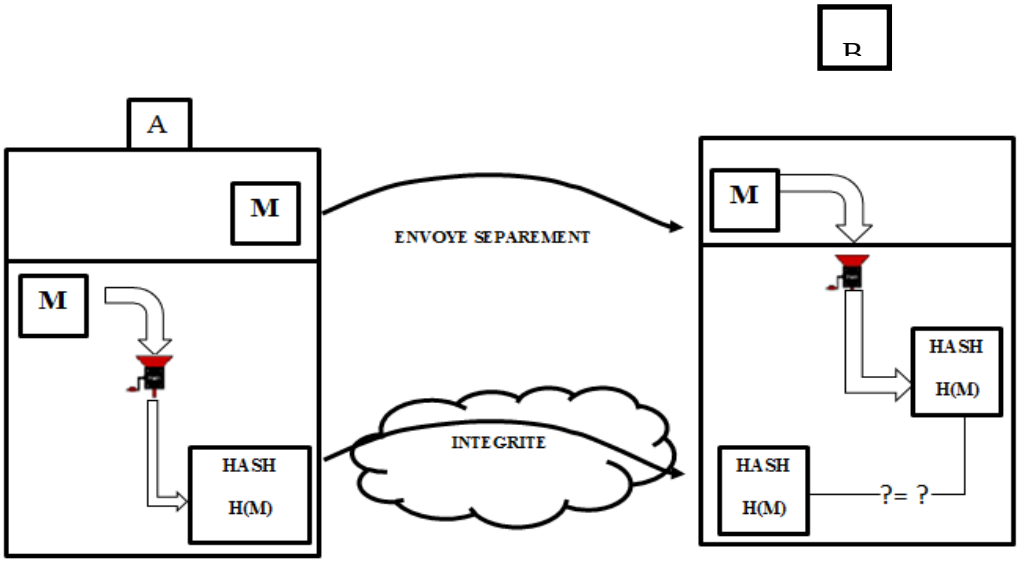


Figure 17 : Vérification de l'intégrité par fonction de hachage

I.5.3 Authentification

Elle a lieu à plusieurs niveaux.

– Au niveau des parties communicantes, dans le cas d'un système symétrique (figure 18) ou asymétrique (figure 19). A la première figure, Ra est une nonce (par. ex. nombre aléatoire), propre à l'utilisateur A. Les lettres A et B représentent des identificateurs personnes

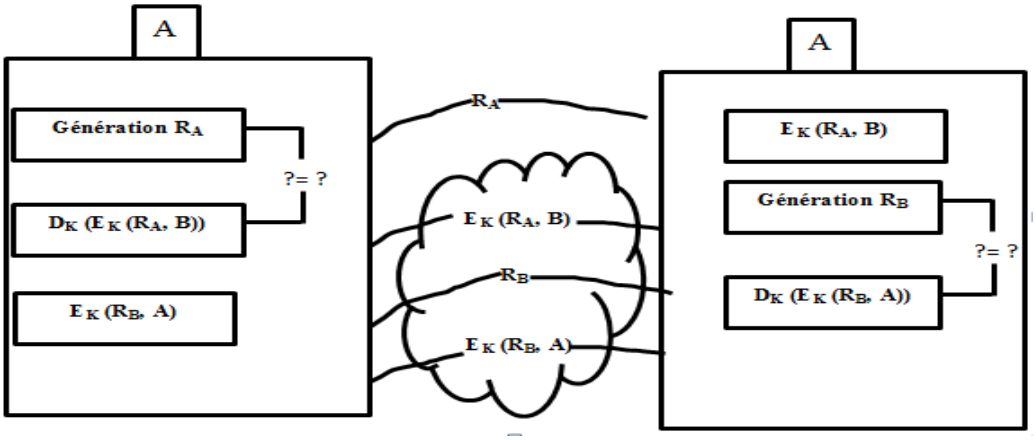


Figure 18 : Authentification dans un système symétrique

A la s figure 19, la clé de chiffrement utilisée est bien la clé privée. Comme le propriétaire de cette clé est le seul à la connaitre, cela prouve qu'il est bien la personne ayant chiffré le message. Attention, dans cet exemple, seule l'authentification est souhaitée. Le message envoyé pourra être lu par toute personne possédant la clé publique, c'est-à-dire, n'importe qui. La confidentialité est ici nulle.

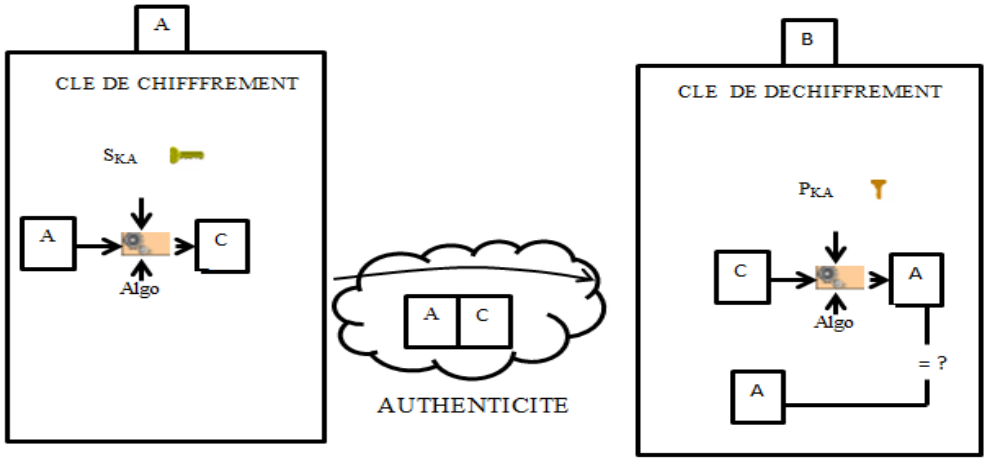


Figure 19 : Authentification dans un système asymétrique

– Au niveau du message :par l'utilisation d'un MAC généré à l'aide d'un cryptosystème à clé symétrique où le MAC est constitué des derniers digits de C (figure 20), ou généré à l'aide d'une fonction de hachage (figure 21), la clé secrète K utilisée étant partagée par les deux entités A et B. Dans les deux cas, l'authentification repose sur l'utilisation de la clé K.

Son fonctionnement d'une façon schématique est montré à la figure suivante où on a deux interlocuteurs A comme Alice et B comme Bob. Puisqu' il fonctionne dans un système symétrique du point sécuritaire. En effet, Alice et Bob partagent la même clé de chiffrement et de déchiffrement.

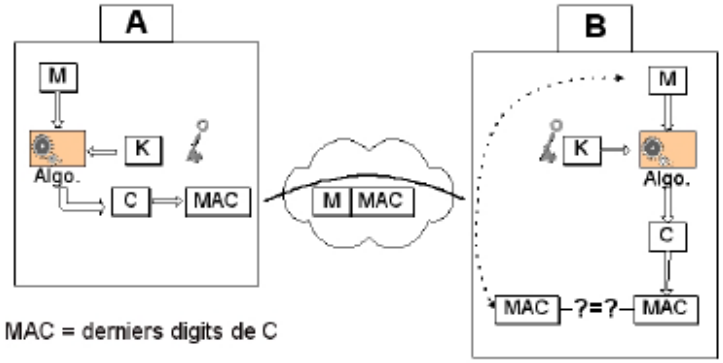


Figure 20 : Authentification par MAC et système symétrique

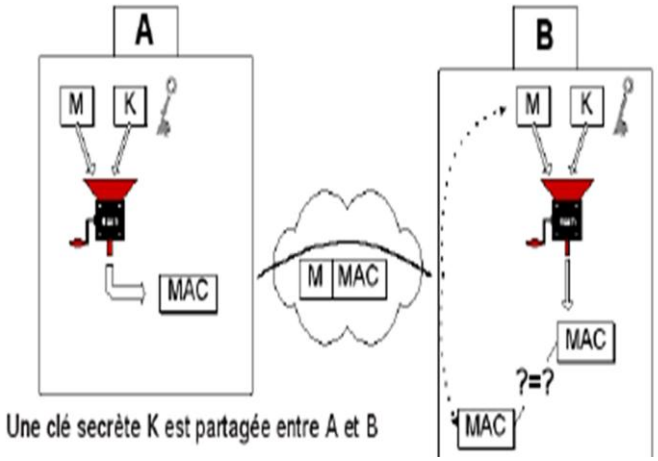


Figure 21: Authentification par MAC et fonction de hachage

Parmi les propriétés remarquables de ces signatures, on peut dire qu'elles doivent être authentiques, infalsifiables, non-réutilisables, non-répudiables, et inaltérables. Dans la figure XXII, on fait une abstraction de la confidentialité. C'est l'authentification qui importe.

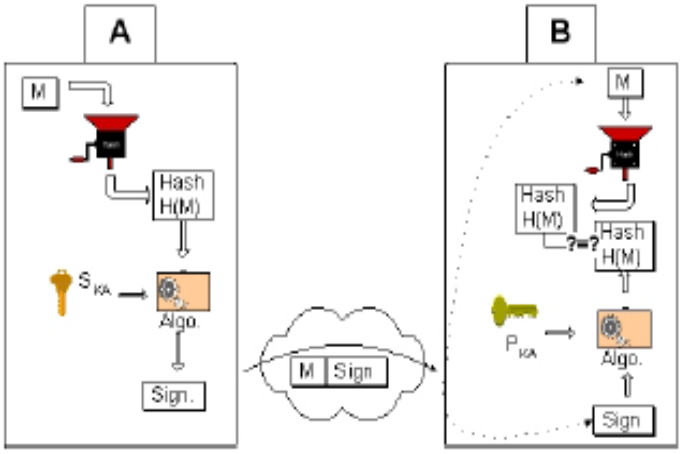


Figure 22 : Authentification par signature (technique asymétrique)

Dans les parties précédentes j'ai pu montrer la sécurité des systèmes informatiques, différentes menaces, la place de la cryptographie dans la sécurité et maintenant .Dans la partie suivante, je vais montrer son évolution, de la cryptographie antique jusqu'à nos jours (cryptographie moderne). Cette historique sera partagée en deux catégorie respectivement la cryptographie classique et la cryptographie moderne.

I.6 Historique

I.6.1 Cryptographie classique

Depuis l'antiquité, la cryptographie était réservée à un petit groupe de personnes qui avaient les capacités d'imaginer et de développer une telle idée. Il n'y avait pas là aucun manuel ou aucune aide quelconque, donc il ne s'agissait pas de la vraie cryptographie.

Ainsi, la sécurité était faible car il y avait beaucoup de risque que les messages codés soient découvert et décoder par d'autres personnes En effet, du fait que la majorité de la population était de la population était illettrée, la plus grande partie des gens n'auraient même pas pu imaginer ce concept et même s'ils y avaient songé, les risques restaient minimes. Par exemple en Egypte, seuls les scribes et les personnes importantes ayant reçues une certaine éducation savaient lire et écrire les hiéroglyphes. Voici une suite de période marquante des origines de cryptographie : [1]

- **1900 av. J.-C.** Un scribe égyptien utilise des hiéroglyphes qui ne sont pas standards racontant la vie de son maître. Le but n'était pas de rendre le texte

  Incompréhensible mais plutôt de lui donner un caractère plus solennel.

- **1600 av. J.-C.** Le premier document chiffré connu remonte à l'Antiquité.

Il s'agit d'une tablette d'argile, retrouvée en Irak. Un potier y avait gravé sa recette secrète en supprimant des consonnes et en modifiant l'orthographe des mots.

- **600 av. J.-C.** Un roi de Babylone écrit sur le crâne rasé de ses esclaves, attend que leurs cheveux aient repoussé, et les envoie à ses généraux. Il suit ensuite de raser à nouveau le messager pour lire le texte.
- **600 av. J.-C.** La Mésopotamie, grande civilisation de l'antiquité, avait atteint un niveau cryptologique étonnamment moderne. On a retrouvé en Iran des fragments de tablettes où des nombres correspondaient à des mots.
- **500 av. J.-C.** Des scribes hébreux emploient l'ATBASH, un simple algorithme de déchiffrement par substitution utilisant l'alphabet renversé, afin de transcrire le livre de Jeremiah. (Par exemple bonjour devient ruojnob).
- **487 av. J.-C.** Des grecs utilisent une scytale, aussi appelé bâton de Plutarque (Historien et moraliste de la Grèce Antique). Il s'agit d'un bâton autour duquel on enroulait une longue et mince bande de cuir sur laquelle on écrivait notre message secret (souvent codé). Une fois la bande déroulée, il était difficile de retrouver le message. Seule le destinataire, connaissant le diamètre du bâton de celui ayant servi à écrire le message, pouvait le déchiffrer.

Figure 23: Une Scytale

La principale faiblesse de ce système est qu'un bâton de diamètre approximativement égal suffit à déchirer le texte. La sécurité reste donc sur le secret autour du procédé de chiffrement du message. Les premiers systèmes cryptographiques : Ancien Âge

Dans le reste du monde la situation restait à peu près semblable jusqu'à environs

- **50 av. J.-C.** L'homme prend enfin l'initiative de développer les techniques de protections des informations confidentielles de façon plus efficace.
- **150 av. J.-C.** Polybe, historien grec, imagine le procédé de chiffrement très innovant pour son temps. Cette méthode utilise un système de transmission basé sur un carré de 25. Cependant l'alphabet français contient 26 lettres donc

il faut supprimer une lettre, en général il s'agit du W. Les cryptologies modernes ont vu dans cette méthode plusieurs caractéristiques très intéressantes dont la conversion de lettres en chiffres, la représentation de chaque lettre par deux éléments séparés.

- **50 av. J.-C.** Jules César utilise une substitution dans l'alphabet pour les communications gouvernementales. En effet, il décalait la lettre qu'il souhaitait coder de 3 lettres vers la droite (en revenant au début de l'alphabet si besoin).
- **200** Le papyrus de Leyde est le plus ancien manuscrit connu concernant l'alchimie. Il utilise un algorithme de chiffrement pour cacher les parties importantes de certaines recettes.
- **1499 Jean Trithème (1462-1516)**, ancien abbé, est considéré comme un des pères de la cryptographie. En effet, il est l'auteur d'un des premiers systèmes poly alphabétiques et il a créé une technique de stéganographie (fait de cacher un message au sein d'un autre message) où les lettres sont remplacées par des mots choisis de manière à former, par leur réunion, une prière par exemple.
- **1560 Blaise de Vigenère (1523 - 1596)** est l'auteur de l'un des premiers systèmes de substitution poly-alphabétique, il utilise donc une clé. Cette méthode restera dominante pendant trois siècles. Sa particularité est qu'il n'utilise non pas un alphabet, mais 26 alphabets décalés pour chiffrer un message. On peut résumer ces décalages avec un carré de Vigenère. Ce chiffre utilise une clé qui définit le décalage pour chaque lettre du message.

I.6.2 Cryptographie moderne

L’avènement croissant des nouvelles technologies de l’information et de la communication exige des nouvelles méthodes de sécurité afin de garantir la fiabilité de l’information transitant d’un nœud à une autre. La cryptographie a longtemps resté une science secrète réservée aux gouvernements, services secrets ou les armées. Dans cette partie on montrera les différentes phases de son développement.

En 1918 Gilbert Vernam met au point l'algorithme One Time Pad (traduisez masque jetable) aussi appelé chiffre de Vernam. En effet, il fonctionne sur le même principe que le chiffrement de Vigenère, avec quelques règles supplémentaires : Une clé ne doit être utilisée qu'une seule fois, elle doit être de la même taille que le message et elle doit être générée aléatoirement. Ce système est donc reconnu comme étant l'algorithme de chiffrement le plus sécuritaire.

En 1923 Le Dr Arthur Scherbius, hollandais résidant en Allemagne, met au point une machine nommée Enigma qui sert à encoder des messages .Différentes version d'Enigma sont utilisées dans les communications radios allemandes ainsi que pour la communication télégraphique pendant la guerre. Même les bulletins météos sont codés avec Enigma. Elle continuera à être utilisée dans l'armée encore jusqu'en 1939 (date à laquelle le code de cette machine a été cassé). Après la seconde guerre mondiale la situation a considérablement changé.

En1976, IBM publie un algorithme basé sur Lucifer (l'une des premières méthodes de chiffrement moderne destiné à un usage civil). Il devient le DES (Data Encryption Standard). C'est un chiffrement qui transforme des blocs de 64 bits avec une clé secrète de 56 bits au moyen de permutations et de substitutions. Le DES est considéré comme étant raisonnablement sécuritaire. En 1978 le RSA est inventé par Ronald L. Rivest, Adi Shamir et Leonard M. Adleman. Il est un des plus populaires des systèmes à clés publiques. En 1992 Le MD5 (Message Digest 5) est développé par Ronald L. Rivest. Il s'agit d'une fonction de hachage. Très utilisée sur l'Internet mais n'est pas considéré comme étant un algorithme sûr.

2000 L'AES (évolution de l'algorithme Rijndael) devient le standard du chiffrement avancé pour les organisations du gouvernement des Etats-Unis.

2005 La cryptographie quantique, qui repose sur la physique quantique, serait considérée comme sûre à presque 100%. Ce sera peut-être la méthode du futur car elle est toujours en cours d'expérimentation.

Les cryptosystème sont basés sur des problèmes mathématiques difficiles à résoudre tel que la factorisation et le problème de logarithme discret. Le principe d'un protocole cryptographique est qu'il doit être simple de chiffrer un message, mais qu'il doit être très difficile voire impossible à déchiffrer sans posséder la clé du système. Dans la suite de ce travail, on va revoir certains complément mathématiques qui nous permettrons de comprendre le fonctionnement de certains algorithmes. En effet, je vais généraliser les notions relatives à la primalité des nombre, la factorisation, la

congruence modulaire, problème du logarithme discret, et enfin les courbes elliptiques appliqués à la signature numériques.

I.6.3Principe de fonctionnement

Notre travail débute de l'étude de l'algorithme de signature numérique (DSA) afin de comprendre ses principes de fonctionnement. Mais les détails de l'algorithme seront traités après avoir décrit quelques compléments mathématiques.

Le DSA est similaire à un autre type de signature développée par Schnorr. Il a aussi des points communs avec la signature El Gamal. Le processus se fait en trois étapes :

- Génération des clés.
- Signature du document.
- Vérification du document signé.

Une signature numérique DSA est calculée en utilisant un ensemble de paramètres de domaine, une clé privée **x**, un numéro secret **k** par message, les données doivent être signés, et une fonction de hachage. Une signature numérique est vérifiée en utilisant les mêmes paramètres de domaine, une clé **y** publique qui est mathématiquement liée aux **x** clés privée utilisée pour générer la signature numérique, les données à vérifier, et la même fonction de hachage qui a été utilisée lors de la génération de signature [18].

Conclusion

La compréhension de ce thème a été rendu possible grâce à une analyse sur sous différents points qui m'a permis de le débuter. Pour le faire, ce chapitre a fourni les informations relatives à la sécurité informatique surtout la description des différents attaques des systèmes informatiques, de la description des techniques de sécurités, description des modèles de sécurité ; l'introduction de la cryptographie comme outil de sécurité tout en montrant les différents cryptosystème et leurs historiques. Ensuite il a défini le contexte de travail en faisant sortir l'état des lieux, problématique et hypothèse.

CHAPITRE II. COMPLÉMENTS MATHÉMATIQUES

II.1 Introduction à la théorie des ensembles

II.1.0 Introduction

Dans cette partie je vais revenir sur certains concepts de la théorie des ensembles utiles dans la compréhension de la manière dont les algorithmes informatiques en général, et surtout cryptographique en particulier, ont été conçus.

Etant donné que le système informatique est un ensemble formé de deux éléments indissociables qui sont les matériels et le logiciel, je vais essayer de définir les concepts tels que l'ensemble, l'élément, l'ensemble des parties d'un ensemble, le couple, le produit cartésien, relation, relation d'ordre, relation d'équivalence, une partition, une application, graphe d'une application, sous-espace vectoriel, système générateur, le groupe, sous-groupe, puissance d'un élément de groupe, ordre d'un élément de groupe, ordre d'un groupe, l'anneau, le corps commutatif .

II.1.1.Notations et définitions

Un ensemble est collection d'objets soumises aux certaines contraintes [2].Il est noté par une lettre majuscule.

Exemple : Ensemble A

Un élément d'un ensemble est défini comme un objet de cette collection. Il est noté par une lettre minuscule ou chiffres ou d'autres caractères

Exemple : a, b, c, d, 1, 12

L'ensemble qui ne contient aucun élément est appelé l'ensemble vide et il est noté ∅; ou parfois {}. On remarquera bien que E = {;} n'est pas l'ensemble vide puisqu'il contient un élément.

Il existe des relations définies entre ensemble et son élément comme la relation d'appartenance symbolisé par∈, mais aussi des relations entre les ensembles eux-mêmes comme la relation d'inclusion symbolisée par ⊂, union symbolisée par ⋃, la différence symbolisée par / , intersection symbolisée par ⋂, complément symbolisée par ∁, Si p désigne une propriété portant sur l'ensemble A, on écrira en extension ∀x ∈ A p(x). [2]

Quand on veut désigner un ensemble forme d'ensemble vérifiant la propriété p(x), on écrira {x | p(x)} [3].

L'ensemble des parties d'un ensemble A est l'ensemble dont les éléments sont les parties de A, il est noté $\mathcal{P}(A)$.

Couple est formé de deux objets distincts ou égaux. Le couple formé de deux objets x et y est noté (x, y).Pour tout ensemble fini A, si n désigne le nombre d'éléments de A, le nombre d'élément de $\mathcal{P}(A)$ est 2^n

Le produit cartésien de deux ensembles A et B est l'ensemble des couples (x, y) où xest un élément de A et y un élément de B. On note A ×B le produit cartésien de A et B.[3]

Exemple: {m, n} × {i, j} = {(m, i), (m, j), (n, i), (n, j)}.

Une relation est la description de liens entre certains éléments de l'ensemble. Une relation $\mathcal{R}$ sur un ensemble A peut être réflexive, transitive, symétrique, antisymétrique.

Une relation est dite **relation d'ordre** lorsqu'elle est simultanément réflexive, transitive et antisymétrique.

Une relation est dite **relation d'équivalence**, symbolisé par ∼, lorsqu'elle est simultanément réflexive, symétrique et transitive.

Une partition d'un ensemble E est un ensemble Q de parties de E vérifiant les trois propriétés suivantes :

(i) L'ensemble vide n'est pas un élément de Q.

(ii) Deux éléments distincts de Q sont disjoints.

(IV) Tout élément de E appartient à un élément de Q.

Une application f est un moyen de faire correspondre à chaque élément x d'un ensemble E (son ensemble de départ) un élément noté f(x) (et appelé l'image de x) d'un ensemble F (son ensemble d'arrivée).

On appelle **graphe d'une application** f d'un ensemble E vers un ensemble F l'ensemble {(x, f(x)) | x 2 E}.

R^n représente l'ensemble des n-uplets de réels.

Combinaison linéaire : soit n ≥0 fixé, soit (e_1, . . .,e_k) un système d'éléments de R^n, et soit f un élément de R^n. On dit que f est une **combinaison linéaire** de (e_1, . . . , e_k) lorsqu'il existe des scalaires α_1, . . . , α_k ∈R tels que f = $\alpha_1 e_1$+ · · · + $\alpha_k e_k$[4].

Sous-espace vectoriel : Soit n ≥ 0 fixé. On dit qu'un sous-ensemble E de R^n est **un sous-espace vectoriel** de R^n lorsque les trois conditions suivantes sont vérifiées :

(i) E n'est pas vide.

(ii) Pour tous u, v de E, la somme u + v est aussi dans E.

(IV) Pour tout u de E et tout scalaire α, le produit α u est aussi dans E. [4]

Système générateur : Soit n $\geq$ 0 fixé, $(g_1 \ldots \ldots \ldots g_k)$ un système de vecteurs de R^net E un sous-espace de R^n,on dit que g est un **générateur** de E (ou qu'il engendre E) lorsque E = Rg_1+ · · · + Rg_k.

Groupe : Soit G un ensemble muni d'une opération $\star$, on dit que G est un **groupe** lorsque les trois conditions suivantes sont réalisées:

(i) $\star$ est associative.

(ii) $\star$ possède un ´élément neutre.

(IV) Tout élément de G possède un symétrique pour $\star$

Sous-groupe : Soit G un groupe. Un sous-ensemble H de G est un sous-groupe de G si et seulement si les deux conditions suivantes sont vérifiées:

(1) H n'est pas vide.

(2) Pour tous a, b de H, le produit ab^{-1} est aussi dans H.

Puissance d'un élément de groupe : Soit b un élément d'un groupe et n un entier relatif. On appelle puissance n$^{\text{-ème}}$ de e l'élément e^n défini comme valant $\underbrace{bb \ldots \ldots \ldots bb}_{n\ fois}$ si n$\geq$ 1, comme valant l'inverse de b^{-1} si n≤ -1 et comme valant l'élément neutre si n = 0.L'ensemble des puissances de b est noté <b>.

Ordre d'un élément de groupe : Soit b un élément d'un groupe, dont le neutre est noté e. Si pour tout n $\geq$1, $b^n \neq e$ on dit que b est d'ordre infini. Sinon on appelle ordre de b le plus petit entier n $\geq$ 1 tel que $a^n = e$

Ordre d'un groupe : Soit G un groupe, et a un élément de G. L'ensemble <a> est un sous-groupe de G.

Anneaux : Soit A un ensemble muni de deux opérations, notées + et ×. On dit que A est un anneau lorsque :

(i) Pour l'addition, A est un groupe commutatif.

(ii) La multiplication est associative.

(IV) La multiplication possède un élément neutre.

(iv) Pour tous a, b, c de A, (a + b) c = ac + bc et c (a + b) = ca + cb.

Un anneau est commutatif lorsque sa multiplication est commutative. Il est intègre lorsque il possède au moins deux éléments et pour tout a, b non nul, ab≠0.

Exemple : ensemble des entiers relatifs Z

Corps commutatif : On dit qu'un anneau K est un corps commutatif lorsque :

(i) La multiplication est commutative.

(ii) K possède au moins deux éléments.

(IV) Tout élément non nul de K possède un inverse pour la multiplication. [4]

Exemples : ensemble des fractions Q, ensemble des réels R, ensemble des nombres complexes

Les différentes notations de la théorie des ensembles citées ci-hautes apparaitrons dans chapitre suivant lors de la généralisation des notions mathématiques sur lesquelles sont fondés les algorithmes cryptographique .Dans la partie suivante je vais montrer la structure des nombres premiers qui sont à la base des différents cryptosystèmes, par après je vais montrer qu'il est possible de générer un nombre premier de longueur de 64 bits, 128 bits, 256 bits, 512bits, 1024 bits. Dans la génération de ces nombres j'ai utilisé un langage de programmation java avec netbeans comme environnement de développement. Le chapitre sera clôturé après avoir décrit le problème de la factorisation des nombres en utilisant les théorèmes déjà connus de l'arithmétique, l'arithmétique modulaire, problèmes du logarithme discret tout en restant sur l'essentiel utile pour la cryptologie.

II.2. Nombres premiers

Comme même le nom l'indique, un nombre premier est un nombre divisible par un et lui-même. Les nombres premiers ont depuis toujours fasciné les mathématiciens. Parce que bien qu'ils soient définis par une propriété simple, un nombre premier est un entier naturel défini par le fait d'avoir exactement deux diviseurs distincts, 1 et lui-même, il existe une infinité de nombres de ce type, et leur répartition, qui ne semble être régie par aucune règle, paraît très irrégulière. Ces nombres sont particulièrement importants en arithmétique, la branche des mathématiques qui traite des nombres entiers. Mais ils font également l'objet d'une actualité brûlante dans les nouvelles technologies, en particulier dans la cryptographie, pour le codage des informations. [5]

II.2.1 Vocabulaire de base

Tout nombre premier est un entier mais tout entier n'est pas premier. Il existe des entiers composés admettant d'autres diviseurs à part un et lui-même.

Soit N un entier strictement supérieur à 1, si N est composé alors il existe deux entiers p et q strictement supérieurs à 1 tel que N = p × q. Si p et q sont strictement supérieurs à √N alors on aurait p ×q >N. On en déduit donc que tout nombre composé admet au moins un diviseur autre que 1 inférieur ou égal à √N. [6]

On dit qu'un entier a (∈Z) est un multiple d'un entier b (∈Z), ou que b est un

diviseur de a lorsqu'il existe un entier k (∈ Z) tel que a = kb[4].

II.2.2 Algorithme de génération d'un nombre premier

On peut générer un nombre premier soit en précisant un nombre fixe et par après on détermine les nombres premiers inférieures à ce nombre ; soit en générant aléatoirement un nombre premier en créant un objet de type SecureRandom dans une fonction qui retourne un BIginteger .Dans ce travail j'ai fait un programme qui génère 5 nombre premier de 512, le choix du nombre du bit est aléatoire.

Pour le cas d'un nombre fixe, voici l'algorithme que j'ai utilisé :

Entrée ; N, un entier fixe qui est le nombre de bit

Sortie : A, l'ensemble des nombres de pi éléments premiers inférieures à N

Début :

Soit B, l'ensemble des entiers ni, avec ni $\in [2, N-1]$, i le numéro de l'entier.

Soit D, l'ensemble des diviseurs dj de ni, dj $\in [1, di]$

Soit c=0 le compteur des diviseurs

Pour i = 0 à ni-1 faire

I) Répéter : pour j =0 à dj-1 faire

1) m =dj modulo ni

Si m=0, di est le diviseur de ni

Incrémenter c de 1
Incrémenter j de 1
Fin si

retour à I)

Si c = 2 alors ni est premier

pi←ni

Retourner pi

Fin si

Sinon

ni n'est pas premier

Fin sinon

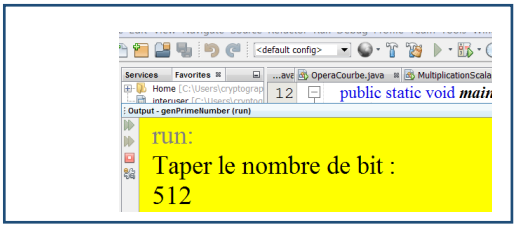
II) incrémenter i

III) Retourner pi

FIN

Les 2 figures suivantes montrent respectivement l'interface où l'utilisateur est demandé de saisir le nombre de bits que constituera le nombre, l'autre montre après le saisi et enfin du fait que le résultat est de grande taille, je fais la copie du résultat.

Interface pour entrer le nombre de bit



Interface après l'entrée de bit

Run:

Taper le nombre de bit :

512

1:849241999352573170093962680330218277639079617824150674573725593077739358127322245746062419537343980848593300263285920307175272546943531737481189027353994 1

2:829167656435863544823902865042407465712800133787423357769854276014343010775660746677963695510714350574839137921402511280821643008016610402438035550791652 7

3:980474653336982436142747786411657869281909276945850487848548780666899593104607495264986592825973007564095754944577810552321758352004294202682024851001127 1

4:109433400674882474565944983370018171607459667585670448320669097277421726219868452788376939891020513277196564057301902754239982365997185902856508436276983 57

5:100928261952551666854871573988752658185730297180553582150740073132797581110171312456677644960038202705271279380870220384899412820295110179492857157894443 11

BUILD SUCCESSFUL (total time: 6 minutes 17 seconds)

II.3 Problème de factorisation et applications en cryptographie

La factorisation est un problème algorithmique dont la solution est difficile à trouver, il est facile de trouver deux grands nombres premiers p et q et de calculer leur produit $N = p \times q$. Par conséquent, la difficulté réside dans la manière dont vous utilisez pour retrouver p et q suffisamment grands, connaissant N. Ce problème est appelé la factorisation.

Il est utilisé en cryptographie pour concevoir les algorithmes de sécurité comme le cryptosystème RSA, Rabin, etc. Les nombres p et q doivent être des nombres premiers comme on a déjà décrit les propriétés de ces derniers.

Comme déjà classé les algorithmes en deux catégories c'est - à - dire les algorithmes à clé secrète et les algorithmes à clé publique, j'aimerais donner, après avoir montré le principe d'échange des algorithmes à clé publiques, un exemple illustratif de l'application du problème de factorisation à l'algorithme RSA. Dans ce sous points, je vais d'abord montre le protocole standard d'échange d'une information dans un

système asymétrique, l'application de ce protocole dans l'échange de clé de Deffie Hellman avec un exemple à l'appui

II.3.1. Protocole standard d'échange d'une information

La cryptographie à clé publique est centrée sur l'échange sécurisée d'une information secrète via les communication publiques sans toutefois échanger le moindre information comparativement à la cryptographie à clé secrète. Il est nécessaire de comprendre son simple protocole d'échange via cet algorithme standard. Ici, pour introduire protocole, supposons qu'Alice veut envoyer de l'argent à Bob. Comment Bob parviendra –t-il à récupérer de l'argent.

Entrée : Alice possède de l'argent

Sortie : Bob possède l'argent d'Alice

Début : Alice met son argent dans un coffre

Alice ferme le coffre avec son cadenas A

Alice envoi le coffre à Bob

Bob rajoute son propre cadenas sur le coffre

Bob envoie le coffre à Alice

Alice enlève son cadenas A du coffre

Alice envoie le coffre à Bob

Bob enlève son cadenas du coffre

Bob récupère l'argent

Fin

En regardant ce simple protocole, tout transport qu'a subit le coffre, celui-ci est sécurisé par le cadenas. A partir de simple protocole, on comprend le fonctionnement des protocoles à clé publiques. Le premier protocole fut imposé par Diffie et Hellman en 1976 [8] utilisant l'exponentiation modulaire comme fonction de trappe. Il parait facile de calculer g^amod p, mais difficile de retrouver a à partir de g^amod p.

Voici comment le protocole d'échange de Diffie Hellman

Au niveau de l'entrée, on a deux éléments publics qui sont un nombre premier p et un générateur g de $Z_p^\star$; et la clé secrète et commune mais privé à la sortie.

Le protocole est formulé de la façon suivante.

Entrée : public : p et un générateur g de $Z_p^\star$

Sortie : privé : une clé K secrète et commune

Début :

Alice choisit un entier a Bob choisit un entier b

Alice envoi A= g^amod p à Bob Bob envoi B= g^bmod p à Alice

Alice calcule K_a = B^amod p. Bob calcule K_b= A^bmod p.

Fin

Il faut vérifier que K produite est commune pour Alice et Bob. En effet, on a

K= K_a = B^amod p=g^{ab} mod p=K_b= A^bmod p.

Un intrus trouve la clé secrète K, il faut qu'il soit en mesure de calculer g^{ab} à partir g,g^a, g^bet p

Exemple illustratif.

Soient p=11,g=7,a=4,b=10, K,K_a ,K_b

Alice calcule : Bob calcule :

1) A=7^4mod11=3 B= 7^{10}mod11=1
2) $K_a = 1^4$mod11=1 K_b= 3^{10}mod11=1

Pour vérifier que les valeurs de K_a et K_b sont vraies, il faut calculer

K=g^{ab} mod p= 7^{4*10} mod 11

=7^{40} mod 11

=1

D'où la clé K est commune pour Alice et Bob.

II.4 Chiffrement de RSA

II.4.1 Introduction

L'algorithme RSA, premier algorithme de chiffrement asymétrique a été conçu par Ron Rivest, Adi Shamir et Leonard Adleman, en 1977 au Massachusetts Institute of Technology. Il trouve son application dans différents systèmes informatiques tels que les systèmes d'exploitations, cartes à puces bancaires [25] et bien sûr le réseau internet pour assurer la confidentialité du courrier électronique et authentifier les

utilisateurs .Il est basé sur la théorie des nombres premiers et il n'existe aucun algorithme de décomposition d'un nombre en facteurs premiers d'où réside sa robustesse. Son principe se trouve dans la partie suivante.

II.4.2 Principe de fonctionnement

Le RSA est un cryptosystème à clé publique, que ça soit l'algorithme de calcul, la clé de codage, tous les deux sont accessibles. L'importance d'accéder à la clé du destinataire est de permettre aux émetteurs de crypter le message dont le décryptage est effectué par le destinataire grâce à sa clé secrète.

II.4.3 Génération des clés

Le RSA fonctionne à partir de deux nombres premiers p et q. Ces nombres doivent être de grande taille en termes du nombre de bit, ils constituent la base de cryptage. En effet, il faut calculer n qui est le produit de p et q, et puis on calcule l'indicatrice d'Euler $\varphi(n)$= $\varphi(p)$. $\varphi(q)$=$(p-1)(q-1)$.

Donc, à partir de$\varphi(n)$, on détermine e< $\varphi(n)$ et doit être premier avec$\varphi(n)$, on calcule son inverse modulo$\varphi(n)$, que est noté d, avec d=e^{-1}modulo $\varphi(n)$.[26]

La clé publique est le couple (n, d) et la clé privée est le couple(n, d).

II.4.4Le chiffrement

Il s'effectue e la façon suivante :

- Avant d'être chiffré, le message original doit être décomposé en une série d'entiers M de valeurs comprises entre 0 et n-1.
- Pour chaque entier M il suffit de le mettre à la puissance e, d'où
  C≡ M^emodulo n.
- Le message chiffré est constitué de la succession des entiers C.

II.4.5 Déchiffrement

Conformément à la manière dont il a été chiffré, le message reçu doit être composé d'une succession d'entiers *C* de valeurs comprises entre 0 et $n-1$. [27]

Pour chaque entier *C* il faut calculer $M\equiv C^d$modulo *n*. Le message original peut alors être reconstitué à partir de la série d'entiers *M*.

II.5 Problème du logarithme discret et application en cryptographie

Le problème du logarithme discret est un autre problème algorithmique difficile à résoudre. Il est constitué de la façon suivante.

Etant donné y∈ G, trouver x ∈N tel que g^x=y si un tel x existe avec g le générateur du groupe.

Dans le cas des courbes elliptiques, le problème du logarithme discret de E par rapport à la base P est : étant donné $Q \in E$, de trouver $x \in N$ tel que $Q = x \cdot P$ si un tel x existe [28].

Dans la partie suivante, on va montrer le cas pratique d'un algorithme de signature numérique (DSA),lequel nous servira dans la compréhension de sa variante basé sur les courbes elliptiques (ECDSA) après avoir étudier les courbes elliptiques appliquée à ce dernier. Comme la théorie peut paraitre incompréhensible dans l'interprétation de l'algorithme, j'ai pu donner un exemple illustratif malgré qu'il ne respecte pas la règle standard au niveau du choix des paramètres participants dans la génération de la signature.

II.6 Etude de l'algorithme de signature numérique

Cette partie est nous donne les principes de fonctionnement de l'algorithme de signature numérique (DSA) basé sur le logarithme discret ce qui va nous permettre comment sa variante ECDSA utilisant les courbes elliptiques

II.6.1 Historique

Le système DSA a été proposé par l'organisme américain NIST (et conçu par la

NSA) en 1991, approuvé en 1994, et mis à jour en 2000.En 1991, l'agence gouvernementale américaine NIST (National Institute of Standards and Technology) a proposé un système de signature digitale destiné à être utilisé dans les transactions commerciales et gouvernementales. Une signature digitale est un message court ajouté à un document électronique, qui certifie la provenance et l'intégrité du document et ne doit pas pouvoir être copiée. Un tel protocole s'appuie nécessairement sur un système cryptographique à clef publique, puisque la signature doit pouvoir être identifiée par quiconque le souhaité. Le système choisi par le NIST est basé sur le logarithme discret dans le groupe multiplicatif d'un corps premier ni. C'est une variante de protocoles proposés auparavant [ElGamal 1985, Schnorr 1990].De ce fait sa sécurité repose sur la difficulté du calcul du logarithme discret

II.6.2 Principe de fonctionnement

Le DSA est similaire à un autre type de signature développée par Schnorr. Il a aussi des points communs avec la signature El Gamal. Le processus se fait en trois étapes :

- Génération des clés.
- Signature du document.
- Vérification du document signé.

Une signature numérique DSA est calculée en utilisant un ensemble de paramètres de domaine, une clé privée **x**, un numéro secret **k** par message, les données doivent être signés, et une fonction de hachage. Une signature numérique est vérifiée en utilisant les mêmes paramètres de domaine, une clé **y** publique qui est mathématiquement liée aux **x** clés privée utilisée pour générer la signature numérique, les données à vérifier, et la même fonction de hachage qui a été utilisée lors de la génération de signature [22].

II.6.3 Algorithme de génération de clés

La sécurité de DSA repose sur la difficulté du problème du logarithme discret dans un groupe fini.

- Choisir un nombre premier p de longueur L tel que 512 <L<1024, et L est divisible par 64.
- Choisir un nombre premier q de 160 bits, de telle façon que p-1 = qz, avec z un entier.
- Choisir m, avec 1 < h < p-1 de manière à ce que g = h^z mod p> 1 avec h c'est le générateur d'un sous-groupe d'ordre q.
- Générer aléatoirement un x, avec 0 < x < q.
- Calculer y = h^xmod p.

Alors :

La clé publique est : (p, q, g, y)

La clé privée est : x

II.6.4 Algorithme de signature

- Choisir un nombre aléatoire k, 1 < k < q.
- Calculer r = pk mod p mod q.
- Calculer s = k^{-1}(H(m)+r *x) mod p, où H(m) est le résultat d'un hachage cryptographique.

La signature de message m est (r, s).

II.6.5Algorithme de vérification de la signature

- Si s, r$\notin]0, q[$, dans ce cas la signature n'est pas correcte. Au cas contraire, passer aux étapes suivantes.
- Calculer w = s^{-1}mod q
- Calculer u1 = H(m) * w mod q.
- Calculer u2= r * w mod q.
- Calculer v = ($g^{u1} * y^{u2}$ mod p) mod q

Si **v** n'est pas égal **r**, le message ou la signature peut avoir été modifié, il peut y avoir une erreur dans le processus de génération de la signature, ou bien un intrus a peut-être tenté de forger la signature. La signature doit être considérée comme non valide.

II.6.6 Exemple d'illustration

Soit le message à signer m=15.

Génération des clés

- Choisir p = 41 et q = 5, avec q est le nombre de facteurs premiers de p-1
- h choisi égale à 20< p - 1 donc g≡ 20^8 mod 41≡37> 1.
- Clé secrète choisi x = 4< q.
- y ≡ g^xmod p ≡ 37^4mod 41≡10 mod 41.

Alors :

La clé publique est : (41, 5, 37,10).

La clé privée est :4

La signature

Expéditeur signer le message :

- Choisi k = 3, (1 < k < q = 5).
- r ≡ (g^k mod p) mod q≡ (37^3 mod 41) mod 5≡3mod 5
- Supposons H = H(m) = 15 et on calcule :

S≡ k^{-1}(H + x*r) mod q.

S≡ 3^{-1}(15 + 2*3) mod 5:

S≡ (9 * 24) mod 11 ≡ 216 mod 5≡ 4 mod 5.

La signature de message m est (3 ,4).

La vérification

Récepteur prouver signatures numériques dans le calcul :

- W≡ s^{-1}mod q ≡ 7^{-1} mod 11 ≡ 8 mod 11.
- u_1 ≡(H*w) mod q≡ (10*8) mod 11≡ 3 mod 11.
- u_2 ≡(r*w) mod q≡(2* 8) mod 11≡ 5 mod 11.
- v ≡ ((g^{u_1} *g^{u_2}) mod p) mod q≡((33 * 25) mod 23)mod 11 ≡ (864 mod23)mod 11 ≡ 13 mod 11 ≡ 2 mod 11.

Puisque v = r = 3, la signature numériques est validée.

Conclusion

De ce chapitre, j'ai pu montrer la relation, d'une façon générale, des mathématiques et de la cryptographie .En effet, pour faire étudier la sécurité de tout système, il faut une compréhension au préalable des éléments qui constituent ce système, d'où intervient la théorie des ensembles. Le corps des nombres premiers joue un rôle de grande importance dans différents cryptosystème asymétrique. Les problèmes de factorisation et du logarithme discret sont utilisés respectivement par RSA et DSA. Le chapitre suivant nous donne une vision globale de l'arithmétique modulaire orienté à l'ECC.

CHAPITRE III : L'ARITHMETIQUE MODULAIRE ORIENTE A LA CRYPTOGRAPHIE BASEE SUR LES COURBES ELLIPTIQUES(ECC)

III.1.Introduction

Comme les mathématiques sont la mère de la science, il existe une relation intime entre les mathématiques et l'informatique. En intégrant ce chapitre dans mon travail, je n'ai pas voulu prolonger dans la dite arithmétique modulaire en développement tout l'arithmétique modulaire, mais surtout donner une aperçu sur les différentes opérations utilisées par la cryptographie basée sur les courbes elliptiques donc nécessaire au protocole ECC :l'addition, l'inversion et la multiplication modulaire.

Les protocoles d'ECC sont des protocoles cryptographiques utilisant des additions, des inversions et des multiplications sur un corps fini. Dans le cadre de corps premiers Z/p Z, ces opérations deviennent des opérations modulaires modulo le nombre premier p : additions modulaires, multiplications modulaires et inversions modulaires. L'addition modulaire est bien moins couteuse que la multiplication modulaire. L'inversion peut se calculer par une exponentiation modulaire.[30].

De nombreux autres protocoles, de Diffie- Hellman [8] à RSA [31], en passant par ElGamal [20] effectuent une exponentiation soit sur un corps premier Z/p Z soit sur un anneau Z/nZ .L'exponentiation modulaire peut se décomposer en une suite de multiplications modulai et de mises au carré modulaires. La multiplication modulaire est une des opérations les plus utilisées de la cryptographie publique. Toute amélioration de sa complexité a de répercussions positives sur la complexité de l'exponentiation ou de l'inversion modulaire et par conséquent sur les protocoles cryptographiques déjà présentés dans le chapitre précédent. La multiplication modulaire étant l'opération centrale de l'arithmétique modulaire pour la cryptographie à clé publique, elle est fortement étudiée [32] et sa complexité largement améliorée

Quant à l'addition modulaire, elle peut être étudiée en utilisant l'algorithme de J.omura, de Takagi en représentation redondante, j'ai choisi celle de J.omura, du fait que je l'ai bien interprété et compris. L'inversion modulaire est illustrée en utilisant l'algorithme d'Euclide étendue après avoir étudié l'algorithme d'Euclide. L'inversion modulaire est étudiée via le théorème de Fermat en utilisant l'indicateur d'Euler, l'exponentiation modulaire binaire, exponentiation modulaire en base 2^k.

En fin la multiplication modulaire peut être faite en utilisant l'algorithme de Taylor avec mémorisation, Blakley, Montgomery, approximation du quotient par Barrett, l'utilisation d'une écriture redondante par Takagi.

III.2. Addition modulaire

Comme la multiplication modulaire, l'addition modulaire peut être décomposée en une addition suivie d'une réduction. Il existe plusieurs classes de moduli mais qui n'ont pas été développées en totalité qui peuvent améliorer la complexité de l'addition modulaire. La réduction dans ce cas précis correspond à une seule soustraction. La complexité de l'addition modulaire est relativement faible. Cette complexité peut tout de même être améliorée. J.omura [33] modifie les deux opérations à effectuer, addition et soustraction. Ainsi, il évite d'avoir à faire une comparaison et fait uniquement des additions.
La méthode proposée par N. Takagi [34] utilise un système redondant de représentation des nombres. C'est à ce jour une des méthodes les plus efficaces.

III.2.1. Algorithme basé sur la retenue sortante d'Omura

En 1990, J. Omura optimise l'addition modulaire [23]. Il propose l'algorithme qui permet d'accélérer les calculs en transformant une comparaison par la simple analyse de la retenue sortante.

Entrée :a, b, p avec $0 \leq a,b<p$ et $c=2^n-p$

Sortie :s avec $s=a+ b \bmod p$

Début :

$S \leftarrow a+b$

$t \leftarrow s+c$

If($t>2^n$) alors $s \leftarrow t$ modulo 2^n

Fin

Nous vérifions l'exactitude de l'algorithme

- Si $a + b < p$ alors $a + b + c < p + c = 2n$ et l'algorithme retourne $s = a + b$ qui est bien inférieur àp.
- Si $a + b \geq p$ alors $a + b + c \geq p + c = 2n$ et l'algorithme retourne $a + b + c \bmod 2n = a + b + c - 2n = a + b - p$ qui est bien inférieur à p puisque $a + b < 2p$ et supérieur à 0 puisque $a + b \geq p$. Le test de la troisième ligne ne coûte rien : il correspond é l'observation de la retenue sortante du

calcul de t. La réduction modulo la puissance de 2 n'est pas comptée non plus dans le cout d'algorithme.

Au final, cet algorithme effectue deux additions de n bits.

c_{algo} = 2Addn

III.3. Inversion modulaire

L'inversion modulaire d'un entier x modulo un nombre premier p est noté x −1 mod p. Elle correspond à trouver l'entier y tel que x y = 1 mod p. Deux méthodes sont possibles. La première est une extension de l'algorithme proposé par Euclide pour calculer le PGCD, Plus Grand Commun Diviseur, de deux nombres. La seconde est basée sur le petit théorème de Fermat qui transforme l'inversion en une exponentiation [20].

III.3.1.Inversion modulaire via l'algorithme d'Euclide étendu

L'algorithme 9 dû à Euclide (4ème et 3ème siècles avant JC) calcule le PGCD de deux entiers a et b [28].

En utilisant l'algorithme de calcul du plus grand commun diviseur, on a :

Entrée :a, b

Sortie :d, le plus grand diviseur commun à a et b

Début

u←a

v←b

tant que u>0 faire

q← $\left\lfloor \frac{u}{v} \right\rfloor$

t←v

v←u-qv

u←t

Fin tant que

d←u

Fin

Cet algorithme retourne bien le plus grand commun diviseur de a et b. Le PGCD vérifie deux propriétés :

a) Premièrement, le PGCD est multiple de tous les diviseurs communs à a et b. Or si un entier divise a et b alors il divise u et v à chaque tour de la boucle : un diviseur de u et v divise aussi u − qv. Nous obtenons d qui est bien divisible par tous les diviseurs communs à a et b.
b) Deuxièmement, le PGCD divise a et b. Nous vérifions que d divise bien a et b en inversant le raisonnement de la première étape. A la dernière étape, nous savons que d divise u (puis que u = d) et d divise aussi v. Or si d divise u − qv et v alors d divise u. Ainsi nous pouvons remonter jusqu'au fait que d divise a et b.

S'inspirant de ce cette algorithme, Aryabhata, un indien du 5`eme siècle, crée l'algorithme d'Euclide étendu. Cet algorithme reçoit deux entiers a et b et calcule des coefficients u et v tels qu'ils satisfassent l'identité de Bézout.

a*u + b*v = d avec d = PGCD (a, b)
Entrée : a, b
Sortie: u, v, d tel que au+bv=d où d=PGCD(a, b)**(III.1)**
Début :

$$(u_1, u_2, u_3) \leftarrow (1,0,a)$$
$$(v_1, v_2, v_3) \leftarrow (1,0,b)$$

Tant que $v_3 \neq 0$ faire

$$q \leftarrow \left\lfloor \frac{u_3}{v_3} \right\rfloor$$
$$(t_1, t_2, t_3) \leftarrow (v_1, v_2, v_3)$$
$$(v_1, v_2, v_3) \leftarrow (u_1, u_2, u_3) - q(v_1, v_2, v_3)$$
$$(u_1, u_2, u_3) \leftarrow (t_1, t_2, t_3)$$

Fin tant que

$$(u, v, d) \leftarrow (u_1, u_2, u_3)$$

Fin

Nous avons deux invariants de boucle : $u_1a + u_2b = u_3$ et $v_1a + v_2b = v_3$. Les valeurs de u_3 et de v_3 suivent exactement les mêmes calculs que l'algorithme d'Euclide initial qui calcule le PGCD. Nous obtenons u3 = d. Grâce à l'invariant de boucle, nous vérifions : $u_1a + u_2b = d$ où d = PGCD (a, b).Cette algorithme est utilisé pour calculer l'inverse dans un corps premier. Dans le cas de l'inversion modulaire, nous posons a=p et b=x pour obtenir $v=x^{-1} \pmod p$

Nous avons alors up+vx=1 car PGCD(x, p).Cette équation donne v≡1(mod p).

Exemple : Calculer l'inverse modulo x=29 modulo 37 .Cela est illustré dans le Tableau 1

Tableau 1 : Exemple d'utilisation de l'algorithme d'Euclide étendu

(u_1, u_2, u_3)	(1,0,37)	(0,1,29)	$(1,-1,8)$	$(-3,4,5)$	$(4,-5,3)$	$(-7,9,2)$	$(11,-14,2)$
(v_1, v_2, v_3)	(0,1,29)	$(1,-1,8)$	$(-3,4,5)$	$(4,-5,3)$	$(-7,9,2)$	$(11,-14,2)$	(-18,23,0)
q	1	3	1	1	1	1	2

Donc $x^{-1} \equiv -14 \equiv 23$ modulo 37.On peut vérifier tout en prenant 23*29≡ 1(modulo 37).

Knuth [25] donne le nombre moyen de tours dans la boucle que l'algorithme d'Euclide étendu effectue.

$$0.843n + 1.47 \quad \text{(III.2)}$$

Chaque tour dans la boucle comporte une division d'éléments de taille *n* bits et 3 multiplications. Ces trois multiplication peuvent se transformer en une seule puisque division nous donne déjà v_3 et que l'on n'a pas besoin de v dans le cas particulier de l'inversion modulaire.

$$C_{algo} \sim (0.843n + 1.47)(Divn + Muln)$$

Le cas le pire correspond à prendre pour *u* et *v* deux nombres de Fibonacci consécutifs. Ceux-ci sont créés à partir de la suite de Fibonacci régie par F0, F1 = 1 et Fi = Fi − 1 + Fi − 2.

Les premiers nombres de Fibonacci sont 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ... Au final, la complexité d'algorithme d'Euclide étendu est donnée par Lame en 1844 [25].

Selon le théorème de Lamé Si 0 ≤ a, b < N, le nombre de divisions dans l'algorithme d'Euclide appliqué à a et b est au plus de $\lfloor \log_{\varphi}\sqrt{5}\ N \rfloor - 2$où$\varphi$ est le nombre d'or φ = 1+2√5.

En pratique, la complexité du pire cas est approximativement 1.44n −0.328.

$$C_{algo} < (1.44n - 0.328)(Divn + Muln)$$

La différence de complexité entre le cas moyen et le pire cas fait que l'algorithme d'Euclide étendu est intéressant pour l'implémentation logicielle mais moins intéressant pour l'implantation matérielle. Les circuits intégrés sont construits en fonction du cas pire et ne tire aucun avantage d'un bon cas moyen [20].

On peut faire l'inverse modulaire le théorème de Fermat, ce qui est faite dans la partie qui va suivre.

III.3.2. Inversion modulaire via le théorème de Fermat.

Théorème de Fermat : Soit 0 < x < p avec p premier alors $x^{p-1} \equiv 1 \pmod{p}$ Ce théorème de Fermat permet de calculer l'inverse de *x* modulo *p*. En multipliant l'équation $x^{p-1} \equiv 1 \pmod{p}$ par $x^{-1} \pmod{p}$, nous obtenons un calcul de l'inverse. $x^{-1} \equiv x^{p-2} \pmod{p}$(III.3)

Le calcul de l'inverse sur un corps premier correspond à une exponentiation modulaire. Pour ce qui est de l'inversion sur un anneau (par exemple pour RSA), au 18`eme siècle, Leonhard Euler généralise le petit théorème de Fermat. Il utilise l'indicateur d'Euler.

Selon sa définition, l'indicateur d'Euler, noté φ(n), est le nombre d'entiers premiers avec n compris entre 1et n-1.L'indicateur d'Euler φ(n) correspond entre autres au nombre d'entiers inversibles sur l'anneau Z/*n* Z.

Le théorème d'Euler est le suivant :

Soit 0 < x < n, deux entiers premiers entre eux, alors $\mathbf{x}^{\varphi(n)} \equiv 1 \pmod{n}$ où φ(n) est l'indicateur d'Euler. De même que pour Fermat, cette formule permet de calculer l'inverse de *x*(si il existe) modulo n
Pour tout *x* tel que 0 < *x* < *n*, PGCD(*x*, *n*) = 1 $\Rightarrow \boldsymbol{x}^{-1} \equiv \boldsymbol{x}^{\varphi(n-1)} \pmod{n}$ (III.4)
La formule de Fermat se retrouve à partir de la formule d'Euler. Si *p* est premier alors $\varphi(p) = p - 1$.Le calcul de l'inversion sur un corps ou un anneau correspond `a une exponentiation modulaire. Il existe de nombreuses méthodes pour effectuer une exponentiation modulaire. On peut utiliser l'exponentiation modulaire binaire et ou l'exponentiation modulaire en base 2^k.L'exponentiation modulaire binaire est basé sur le principe "square and multiply".

L'algorithme d'exponentiation binaire est la suivante :

Entrée : 0<x, e<p avec e=$(e_{n-1}, \dots, e_0)$

Sortie : y≡ $x^e (mod\ p)$

Début

$y \leftarrow x^{e_{n-1}}$

Pour i$\leftarrow$ n $-$ 2 à 0 faire y$\leftarrow$ y^2modulo p

si $e_1 = 1$ alors y$\leftarrow$ xy modulo p

End

Fin

L'algorithme d'exponentiation en base 2^k

Entrée : 0<x, e<p avec e=$(e_{l-1}, \dots, e_0)$

Sortie : y$\equiv$ x^e(mod p)

Début

MEM(0) $\leftarrow$ 1

Pour i$\leftarrow$ 1 à 2 $-$ 1 faire

MEM(i) $\leftarrow$ MEM(i $-$ 1) $*$ x modulo p

Fin pour

b $\leftarrow$MEM(e_{l-1})

Pour i$\leftarrow$ l $-$ 2 à 0 faire

Pour j$\leftarrow$ 1 à k faire

y$\leftarrow$ y^{2k} modulo p

Fin pour

y$\leftarrow$ MEM(e_i) $*$ y mod p

Fin pour

Fin

La valeur de k intervient dans la complexité de cet algorithme. La longueur l correspond au nombre de chiffres utilisées pour écrire e en base 2^k

Voici un tableau comparatif de coût d'inversion modulaire via Euclide et Fermat [20].

Tableau 2 : Cout d'inversion [30]

	Via Euclide		Via Fermat			
	Algo10		Algo11		Algo12	
	~	<	~	<	K=4	K=5
n	0,843n+1,47	1,44n-0,328	1,5n-1,5	2n-2	1,25n+8	1,2n+23
160	137	231	239	318	208	215
192	164	277	287	382	248	254
224	191	323	335	446	288	292
256	218	369	383	510	328	331
288	245	415	431	574	368	369
320	272	461	479	638	408	407
352	299	507	527	702	448	446
384	326	553	575	766	488	484
416	353	599	623	830	528	523
448	380	645	671	894	568	561
480	407	691	719	958	608	599
512	434	737	767	1022	648	638

III.4 Multiplication modulaire

III.4.1 Introduction

Les algorithmes de multiplications modulaires généralistes sont des algorithmes qui fonctionnent pour n'importe quel modulo, c'est-`a-dire sans utiliser ni requérir aucune propriété sur le modulo. Les algorithmes que nous donnons dans cette partie sont légèrement modifiés par rapport à leur écriture initiale mais correspondent bien à l'idée originale. De plus, cette partie correspond à une liste non exhaustive d'algorithmes de multiplication modulaire généralistes [30].

Les algorithmes introduit ci-dessous, ne sont pas détaillés tous, mais j'ai mis les détails sur celui que j'ai utilisé pour faire la multiplication d'un point d'une courbe elliptique par un scalaire. Voici cette petite liste d'algorithme de multiplication modulaire :

- L'algorithme de G. R. Blakley [38] est le premier algorithme de multiplication modulaire intégrée : la réduction est intégrée dans les calculs de la multiplication. C'est en fait une adaptation en modulaire du "double and add".
- L'algorithme de multiplication modulaire de Peter Montgomery, crée en 1985 par Peter Montgomery [24], n'est plus une adaptation de la multiplication classique au modulaire. Pour effectuer la réduction modulaire, il n'utilise pas de division euclidienne par le modulo. Il remplace cette opération par une

division par une puissance de la base2^n. La réduction modulaire a alors un coût vraiment intéressant. Par contre, Montgomery utilise une représentation dédié à son algorithme.

- L'algorithme de Paul Barrett [26],il propose lui d'approcher le calcul du quotient. Cette approximation est suffisamment précise pour avoir un reste à une ou deux soustractions du reste exact. Elle accélère beaucoup les calculs : le coût de son algorithme est très proche de celui de l'algorithme de Montgomery. Il conserve une représentation classique.
- L'algorithme de N. Takagi [30] utilise son propre système d'addition modulaire en représentation redondante pour proposer une multiplication intégrée très efficace. Celle-ci est une fois de plus sous la forme d'un "double and add".[20]

III.4.2. Algorithme de Taylor avec mémorisation

F. J. Taylor, en1981, proposa un algorithme de multiplication modulaire en utilisant une représentation classique des nombres. Cet algorithme utilise une décomposition de la multiplication en carrées.

$$ab = \frac{(a+b)^2}{4} - \frac{(a-b)^2}{4} \quad \text{(III.5)}$$

Cette d´décomposition est adaptable au modulaire à condition que le modulo p soit impair pour pouvoir faire la division par 4. Cette décomposition est intéressante uniquement si deux mises au carrée modulaire (a^2 mod p) sont moins coûteuses qu'une multiplication modulaire. Taylor utilise une table mémoire, MEM, qui effectue cette opération couteuse. Cette table reçoit en entrée un entier x avec $0 \leq x < p$ et retourne en sortie un entier y avec $0 \leq y < p$ tel que $y \equiv x2 \pmod{p}$. Pour accélérer encore un peu plus les calculs, nous intégrons aussi la division par 4 modulo p. Ce qui donne pour MEM :

Pour tout x tel que $0 \leq x < p$, MEM(x) $\leftarrow 4^{-1}x^2 \pmod{p}$
Le signe n'a pas d'importance pour une mise au carrée ; nous prenons la valeur absolue. Ceci apour avantage de limiter la taille de $a + b$. En soustrayant p au calcul de $a + b$, nous transformons$0 \leq a + b < 2p$ en $-p < a + b - p < p$. Après avoir enlevé le signe, nous obtenons $0 \leq |a + b - p| < p$[20].

L'algorithme de multiplication modulaire de Taylor est le suivant :

Entrée :a, b, p avec $0 \leq a, b < p$

Données :MEM avec MEM(x)=$4^{-1}x^2 \ mod \ p$

Sortie :s avec s=ab modulo p

Début

u←$|a + b - p|$

v← $|a - b|$

s← MEM(u)- MEM(v)

si s<0 alors s←s+p

FIN

III.4.3. Algorithme de Blakley

En 1983 [27], G. R. Blakley adapte un “double and add” c classique en y intégrant des réductions après chaque étape de calcul. Cet algorithme fonctionne pour tous les moduli et utilise une représentation classique des nombre s.

Entrée :a, b, p avec 0≤ $a, b < p$

Sortie :s avec s=ab mod p

Début

a=$[a_{n-1}, \dots, a_0]_2$

S← 0

Pour i← $n - 1$ à 0 $faire$

s← $2s$

If s≥ p alors s ← s − p

s← s + a_ib mod p

If s≥ p alorss ← s − p

Fin pour

Fin

Conclusion

Ce chapitre nous montre différents algorithmes utilisé pour faire l'addition modulaire, la multiplication modulaire et l'inverse modulaire. Ces opérations arithmétiques modulaires sont utilisées lors de l'étude de la cryptographie basée sur les courbes elliptiques. En effet, on aura à faire l'addition de deux points d'une courbe, le double d'un point d'une courbe et la multiplication d'un point par un scalaire. Par rapport au mode classique de calcul des opérations modulaire, on remarque qu'il d'autres modes efficaces et rapides qui peuvent être utilisés afin d'implémenter ces opérations dans des cas très concrets.

Quant à l'addition modulaire, j'ai choisi d'utiliser l'algorithme basé sur la retenue sortante d'Omura.

Quant à l'inverse modulaire, j'ai utilisé l'algorithme d'Euclide étendu, une variante de l'algorithme d'Euclide et par après j'ai essayé d'étudier l'algorithme modulaire en utilisant le théorème de Fermat.

Quant à la multiplication modulaire, on a vu qu'il existe plusieurs algorithme de multiplication modulaire comme l'algorithme de Taylor avec mémorisation l'algorithme de Blakley, L'algorithme de Montgomery, l'approximation du quotient par Barrett, l'utilisation d'une écriture redondante par Takagi même si la liste n'est pas exhaustive. Ces algorithmes n'ont pas été développés, le choix est conditionné de mon degré de compréhension de certains d'entre eux. En, effet les deux premiers ont été analysé tout en constant qu'il y a une transformation du modulo en binaire afin de déterminer la taille du modulo. Celle joue un rôle primordial de façon que si vous vous trompez dans le calcul du nombre de bit utilisé pour représenter le modulo, la solution sera incorrecte. Le chapitre ma beaucoup intéressé et m'a permis d'obtenir la base d'étude des courbes elliptiques appliquées aux signatures numériques.

CHAPITRE IV ETUDE DES COURBES ELLIPTIQUES APPLIQUÉES À L'ALGORITHME DE SIGNATURE NUMÉRIQUE(ECDSA)

IV.0. Introduction

La signature numérique DSA, comme est déjà décrite dans chapitre précédent, est basée sur le problème du logarithme discret. Elle est constituée de trois algorithmes qui sont l'algorithme de génération des clés, algorithme de signature et l'algorithme de vérification de la signature. Tous ces processus resteront inchangés du point de leur nomenclature, mais la différence réside au niveau des opérations. Dans DSA on travaille sur les entiers mais dans ECDSA on utilisera les points de la courbe elliptique. Les courbes elliptiques sont utilisées en cryptographie. Les applications conçues à base des courbes elliptiques sont, par exemple : ECDSA,ECDDH, ECDH, ECDHP,ECDLP,ECIES.

Neal Koblitz et Victor S. Miller ont été les premiers à proposer l'utilisation des courbes elliptiques dans la cryptographie. Cela a donné lieu aux courbes elliptiques dans la cryptographie. Cela a donné lieu à la cryptographie des courbes elliptiques (ECC, Elliptic Curve Cryptography), qui est une cryptographie asymétrique basée sur le problème du logarithme discret sur courbes elliptiques.

Le problème du logarithme discret sur courbes elliptiques est l'un des problèmes les plus importants en cryptographie. Après vingt ans de cryptanalyse infructueuse, ce problème semble beaucoup plus difficile à résoudre que les deux autres problèmes très utilisés en cryptographie (le problème de la factorisation d'entiers et le problème du logarithme discret sur les corps finis).Dans cette partie on va s'intéresser sur les courbes elliptiques appliquées aux algorithmes de signature numériques ECDSA.

IV.1 Généralité sur Courbes elliptiques appliquées à la cryptographie

Le logarithme discret déjà expliqué au chapitre 2, peut être décrit dans un groupe abstrait de cycle fini. Je vais introduire quelques concepts élémentaires de la théorie des groupes qui donnent une généralisation .On va alors voir le groupe d'une façon générale comme on l'a définie dans le chapitre I, mais aussi un groupe défini sur les courbes elliptiques.

IV.1.1 Groupe

Un groupe abélien noté (G,∗) est constitué par un paramètre G avec une opération binaire ∗ : G×G⟶G satisfaisant les propriétés suivantes :

- associativité :a∗ $(b * c)$=a∗ $(b * c)$ pour tout a,b, c, d ∈G.
- admet l'élément neutre e∈ tel que : a∗e=e∗a=a
- admet son inverse :pour a ∈G, ilexiste un élément b∈G appelé inverse de a tel que a∗b=b∗a=1.
- admet une commutativité :a∗b=b∗a pour tout a, b ∈G

D'une façon générale le groupe admet deux opérations dénommées respectivement opération d'addition et de multiplication.

L'élément neutre de l'addition est 0 et l'inverse de a est noté –a et pour la multiplication, l'élément neutre est 1 et l'inverse de a est noté a^{-1}.Le groupe est fini si G possède un nombre fini d'éléments. L'ordre d'un groupe est l'ensemble d'élément qui le constitue.

IV.2 Groupe basé sur les courbes elliptiques.

Soit p un nombre premier,$\mathbb{F}_p$ l'ensemble des entiers modulo p. Une courbe elliptique E défini sur,$\mathbb{F}_p$ est définie par une équation de la forme :

$y^2 \equiv x^3+ax+b$(IV.1)

Où a, b ∈ $\mathbb{F}_p$ avecΔ=$4a^3+27b^2 \not\equiv 0\ (mod\ p)$(IV.2)

La paire (x, y), où x,y ∈ $\mathbb{F}_p$,est un point sur la courbe si et seulement si elle vérifiée l'équation (IV.1).Le point à l'infini est symbolisé par ∞,on dit que lui aussi appartient sur courbe.

L'intérêt en cryptographie des courbes elliptiques est que si on ajoute le point à l'infini aux points de la courbe, les points d'une courbe elliptique forment un groupe abélien[9,10,11].

- Le point à l'infini est l'élément neutre du groupe.
- L'opposé d'un point est son symétrique par rapport à l'axe des abscisses.
- Dans [18], J. H. Silverman démontre qu'il existe une loi de groupe associative qui, à deux points, associe un troisième point

IV.3 Algorithme de génération de points

Plusieurs algorithmes et approches ont été proposé pour le de nombre de points d'une courbe. En effet, depuis le travail de Schoof [32], il est possible de calculer la cardinalité de la courbe E(K) en temps polynomial en la taille du corps. Toutefois, il a fallu les améliorations d'Atkin et Elkies pour rendre l'algorithme praticable sur des corps de taille raisonnable, en caractéristique p grande. Dans mon travail j'ai utilisé une approche naïve telle qu'elle est décrite de la façon suivante :

- Fait une liste possible de valeur de x ∈ $[0, p[$
- Fait une liste de résidus quadratiques x modulo p
- Pour tout x, calculer x^3+$ax + b$ (mod p)
- Déterminer les valeurs possibles de y qui, si on vérifie :

$y^2 Modulo\ p \equiv (x^3$+$ax + b)$ Modulo p

L'ensemble de points de la courbe est noté par $E(\mathbb{F}_p)$.

Le choix de p est aléatoire, mais plus p est plus grand, plus le nombre de points augmente, comme vous le verrez dans l'annexe.

Exemple : Soit une équation de la courbe elliptique $E_{11}(1,4)$:x^3+$x + 4$,p=11

Tableau 3 : Détermination des points de la courbe

x	x^2modulo 11	x^3+ $x + 4$(modulo 11)	Y
0	0	4	2 ; 9
1	1	6	-
2	2	2	5 ; 6
3	9	1	1 ; 10
4	5	5	-
5	3	9	-
6	3	4	-
7	5	2	-
8	9	7	-
9	4	5	4 ; 7
10	1	2	-

On a $E(\mathbb{F}_{11}(1,4)) = 9$={(0,2); (0,9); (2,5); (2,6); (3,1); (3,10); (9,4); (9,7)}

Pratiquement le nombre de points est 8, on a ajouté un point à l'infini, on donc $E(\mathbb{F}_{11}(1,4))$=9

IV.4 Opération sur les courbes elliptiques

Dans ce travail, je vais m'y attacher sur les courbes définies sur un champ fini F_P, mais il en existe d'autres qui dont définies sur un champ fini F_{2^m}.Cette dernière catégorie peut être vue comme un espace vectoriel de dimension m, définie sur F_2 formé de deux éléments 0 et 1.Il existe m éléments $\alpha_{0,\alpha_1 \ldots\ldots,\alpha_{m-1}}$ dans F_{2m}

Cette partie montre les différentes opérations que comprend la courbe elliptiques.

IV.4.1 Addition

Soit la figure suivante sur laquelle on désire additionner deux points *P* et *Q*.

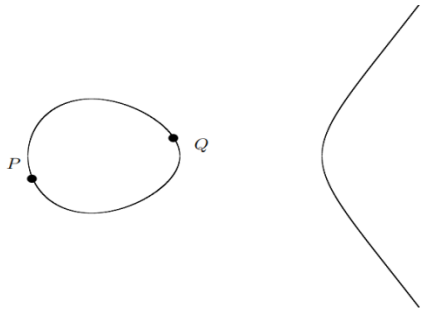


Figure 24 : Exemple d'une courbe elliptique E

Dans ce cas, il faut tracer la droite qui passe par ces deux points. Cette droite coupe la courbe en un troisième point, comme nous pouvons le voir sur la figure IV.2.Il faut alors prendre son point symétrique par rapport à l'axe des abscisses pour obtenir un point *R* tel que *R* = *P* + *Q* sur le groupe des points de la courbe (voir figure IV.3). Le point symétrique par rapport à l'axe des abscisses est en fait l'opposé d'un point dans le groupe. Les coordonnées sont dans un corps de base. Les corps de base des protocoles d'ECC sont soit des corps premiers soit des extensions de corps premiers. Un corps premier est un corps fini Z/*p* Z où *p* est premier. L'extension de degré *k* d'un corps premier est noté(Z/*p* Z) *k*. De nombreuses implémentations utilisent le corps (Z/2 Z) *k*. Mais pour plus de clarté, nous utilisons le corps de base R pour l'exemple de la figure IV.1. Pour calculer les coordonnées exactes du point *R*, nous utilisons l''́équation de la courbe et celle de la droite. Ainsi, nous pouvons directement calculer les coordonnées de *R*. Nous évaluons le coefficient directeur λ de la droite [20].

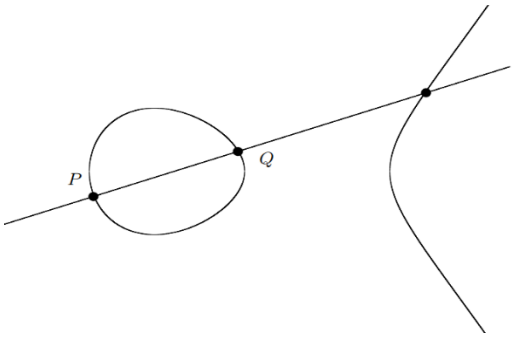


Figure 25 : Droite passant par P et Q

Soient $P = (x_1,y_1)$ et $Q= (x_2,y_2)$ deux points de E tels que $P \neq Q$. Pour trouver la somme de P et Q, on se sert de la droite qui passe par ces deux points et qui ne coupe la courbe qu'à un troisième point $R = (x_3,y_3)$.

La somme de P et Q devient le point $–R$;le point symétrique á R par rapport à l'axe horizontal. [28]

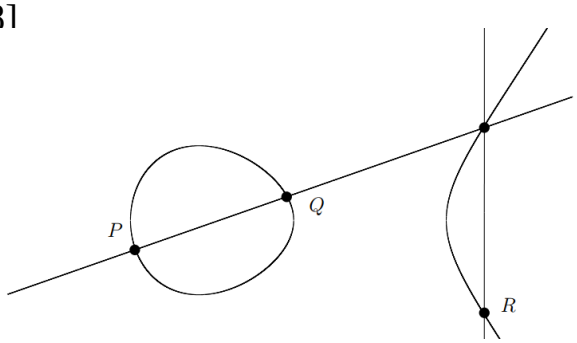


Figure 26 : Description géométrique de l'addition de deux points

Elliptique distincts P+Q=R

Le pont R (x_3,y_3) peut être calculé d'une façon algébrique comme le montre les formules suivantes :

$x_3 = \lambda^2 - x_1 - x_2$ et $y_3 = \lambda(x_1 - x_3) - y_1$ (IV.3)

Où : $\lambda = (y_2 - y_3)/(x_2 - x_1)$ (IV.4)

Si l'abscisse de P et celui de Q sont égaux, alors leur somme est le point à l'infini.

Exemple

Soit une équation de la courbe elliptique traitée$E_{11}(1,4)$:$y^2 = x^3+x+4$,p=11

Soient P(2,6) , Q(3,1)

$\lambda = (y_2 - y_1)/(x_2 - x_1)$ $=\frac{1-6}{3-2}$ $=\frac{-5}{1}$ =-5

$x_3=\lambda^2 - x_1 - x_2=(-5)^2$-2-3=25-2-3=20≡ 9 modulo 11

$y_3 = \lambda (x_1 - x_3) - y_1$=-5(2-9)-6=29≡ 7modulo11

P+Q=(9,7) vérifie l'égalité suivante $y^2 \text{modulo} 11 = x^3+x+4 \text{modolo} 11$

IV.4.2 Calcul du Doublement d'un point

Si l'on veut doubler un point, il faut choisir pour droite la tangente de la courbe en ce point et poursuivre le même protocole (Figure IV.4).

Soit $P = (x_1,y_1) \in E$ tel que $y_1 \neq 0$. Pour trouver la somme de *P* et *P*, on se sert de la tangente de la courbe au point *P*. La tangente ne coupe la courbe qu'a un point $-R = (x_3,y_3)$. Le point $P + P = 2P$ devient le point $R=(x_3,y_3)$ Algébriquement, le point *R*(*x* 3,*y*3) est calculé de la manière suivante :

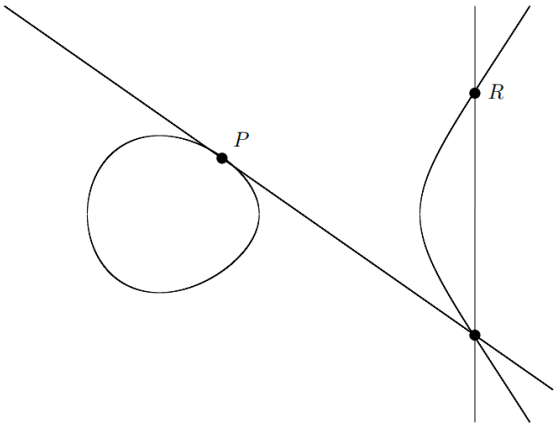


Figure 27 : Calcul du doublement d'un point

$x_3 = \lambda^2 - 2x_1$ IV.4

$y3 = \lambda(x_1 - x_3) - y_1$ IV.5

Où : $\lambda = (3x_1^2 + a)/(2y_1)$.IV.6

Dans le cas ou *y*1 = 0, la tangente devient une droite verticale qui ne coupe pas la courbe à un autre point. Dans ce cas, *P* + *P* = 2*P* = *O*.

Exemple : Soit une équation de la courbe elliptique traitée $E_{11}(1,4)$:

$y^2 = x^3+x+4$,p=11 ,P(2,6)

$\lambda = (3x_1^2 + a)/(2y_1) = \frac{(3*2^2+1)}{2*6} = \frac{13}{12} \equiv \frac{13}{12} \text{modulo } 11 \equiv 2 \text{modulo} 11$

$x_3 = \lambda^2 - 2x_1 = 2^2 - 2*2 = 4-4 = 0 \equiv 0 \text{modulo} 11$

$y3 = \lambda(x_1 - x_3) - y_1 = 2(2-0) - 6 = 4-6 = -2 \equiv 9 \text{modulo} 11$

2P=P+P=(0,9) vérifie l'égalité suivante $y^2 \text{modulo} 11 = x^3+x+4 \text{modolo} 11$

IV.4.3 Calcul de la négation d'un point

Soient $P = (x_1,y_1)$ et $Q = (x_2,y_2)$ deux points de E tels que $P = -Q$. Si on essaye d'appliquer la méthode géométrique utilisée pour l'addition, à la somme de P et Q, on voit que la droite qui passe par ces deux points ne coupe pas la courbe à un troisième point. On définit alors le point à l'infini comme étant le point infiniment éloigné dans la direction de l'axe y. Ce point, noté O, devient le troisième point d'intersection avec P et Q.[29]

Alors $P - P = O \Rightarrow (x_1,y_1) + (x_1, -y_1) = O$ (le point $(x\ 1, -y1) = P$ est appelé le négatif de P) Le point à l'infini O représente l'élément neutre du groupe additif. Pour tout point $P \in E$, on a $P + O = O + P = P$.

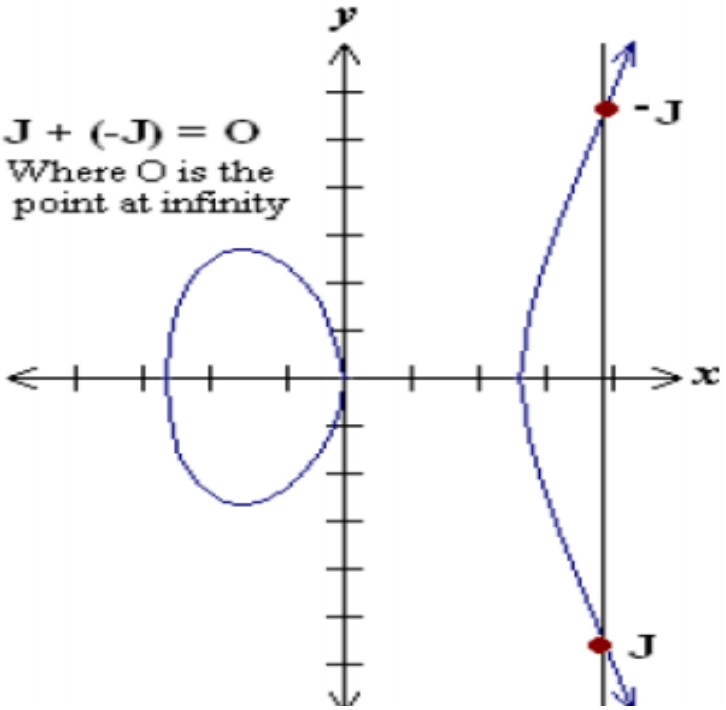


Figure 28: Description géométrique de l'addition de deux points de la courbe elliptique $P - P=O$

IV.4.4 Multiplication scalaire d'un point

La multiplication scalaire d'un point est basé principalement sur l'addition : étant donné un $k \in n$ et $P \in E$ le produit KP est : [29]

KP=$\underbrace{P + P \ldots \ldots P}_{Kfois}$

Exemple : Soit une équation de la courbe elliptique traitée $E_{11}(1,4): y^2 = x^3 + x + 4$,p=11 ,P(2,6) et k=7 ;

On peut vérifier si P appartient à la courbe.

Calculons d'abord P+P =2 P ce qui correspond à la solution précédente de la double d'un point.

D'où 2P= (0 ,9)

Calculons 3P=2P+P=$(0\ ,9) + (2,6)$=(3,1)

4P est calculé de deux manières :-En calculant le double de 2P

4P=2[2P]=2P+2P=(0,3) + (0,3)=(9,7)

-En calculant la somme de 3P et de P :

4P=3P+P=(3,1) + (2,6)=(9,7)

5P=4P+P ou 3P+2P =(9,7) +(2,6)=(9,4)

IV.5. Application des courbes elliptiques à l'algorithme de signature numérique

IV.5.1 Introduction

ECDSA est un algorithme de signature numérique à clé publique utilisant les courbes elliptiques. C'est une adaptation de DSA, qui est un algorithme utilisant un groupe multiplicatif d'entiers modulaires. Tous les deux sont basés sur le problème du logarithme discret. Cet algorithme a été proposé en 1992 par Scott Vanstone, en réponse à un appel d'offre pour les signatures numériques du NIST (National Institute of Standards and Technology). Vanstone fonda la société Certicom en 1985, et son entreprise détient la plupart des brevets des algorithmes à base de courbes elliptiques.

ECDSA a été admis en 1998 comme une norme ISO (International Standards Organization) norme (ISO 14888-3), accepté en 1999 comme ANSI (American National Standards Institute) et a été publié comme norme internationale dans le document ANSI X9.62, et accepté en 2000 comme IEEE (Institute of Electrical and Electronics Engineers) standard (IEEE 1363-2000) et les normes FIPS (Federal

Information Processing Standard) (FIPS 186-2). Il est également à l'étude pour inclusion dans d'autres normes ISO [29].

IV.5.2 Principe de fonctionnement ECDSA

Pour utiliser le protocole ECDSA, les parties doivent s'accorder sur un ensemble d'éléments qui définissent la courbe elliptique (p, $\mathbf{E_p}$(a, b), G, n). Chaque participant génère ses paramètres de domaine et les envoie avec sa clé publique [31].

Comme on a déjà vu dans le chapitre II, les différentes étapes qui nous permettent de signer un message, lors de l'étude de DSA. En effet, il y a la génération de la clé de signature, étape de signature et vérification de la signature. Pour le cas d'ECDSA, les trois opérations restent inchangées, mais s'ajoutent une opération de génération des points de la courbe.

IV.5.3 Génération des points de la courbe.

Pour déterminer les points de courbes, j'ai utilisé la méthode de l'approche naïve comme je l'ai démontré dans IV.1.3.Pour le cas présent la courbe est$E_{1877}(100,155)$tel que: $y^2=x^3+100x+155$.Vous trouverez la liste des points dans les annexes.

Pour générer la signature, j'ai pris un point au hasard point au hasard .Le nombre de points est de n=1926.Vous les trouverez en annexe.

Dans la suite de étude la signature j'ai considéré les points P(1788,895) et Q(1569,524) appartenant respectivement sur la courbe.

IV.5.4 Génération de clé de signature

Le système d'ECDSA est centré, comme les autres algorithmes traités, sur une clé publique et une clé privée. Pour fabriquer ses deux clés on suive les étapes suivant :

- choisir une courbe elliptique(Les paramètres de la courbe doivent être choisis de façon à respecter les critères de sécurité usuels) définie sur le corps F_p. L'ordre de E(F_p) doit être divisible par un grand nombre premier n .
- Choisir un élément G de E(F_p) d'ordre premier n . Les éléments du sous-groupe
  sont :{P, 2P, 3P, ..., (n-1) P, nP=O}.

– Choisir une valeur aléatoire d dans [1,n − 1]. Et par la suit calculer Q = dP. Alors
  La clé public : c'est Q.
  La clé privée : c'est d[1].

IV.5.5 Génération de signature

Pour signer un message m, un expéditeur avec les paramètres du domaine (p, E_p(a, b), P, n) et la paire de clés associée (d, Q) effectue les opérations suivantes :

– Choisir une valeur aléatoire k dans [1, n − 1].
– Calculer k P = (x 1, y1).
– Calculer r= x 1 mod n.
– Calculer e = h(m), où m le message à signer et h la fonction de hachage intégré dans netbeans qui s'exécute sur les données à signer.
– Calculer s=k−1(e+dr)modn.
  La signature de message m est (r, s)

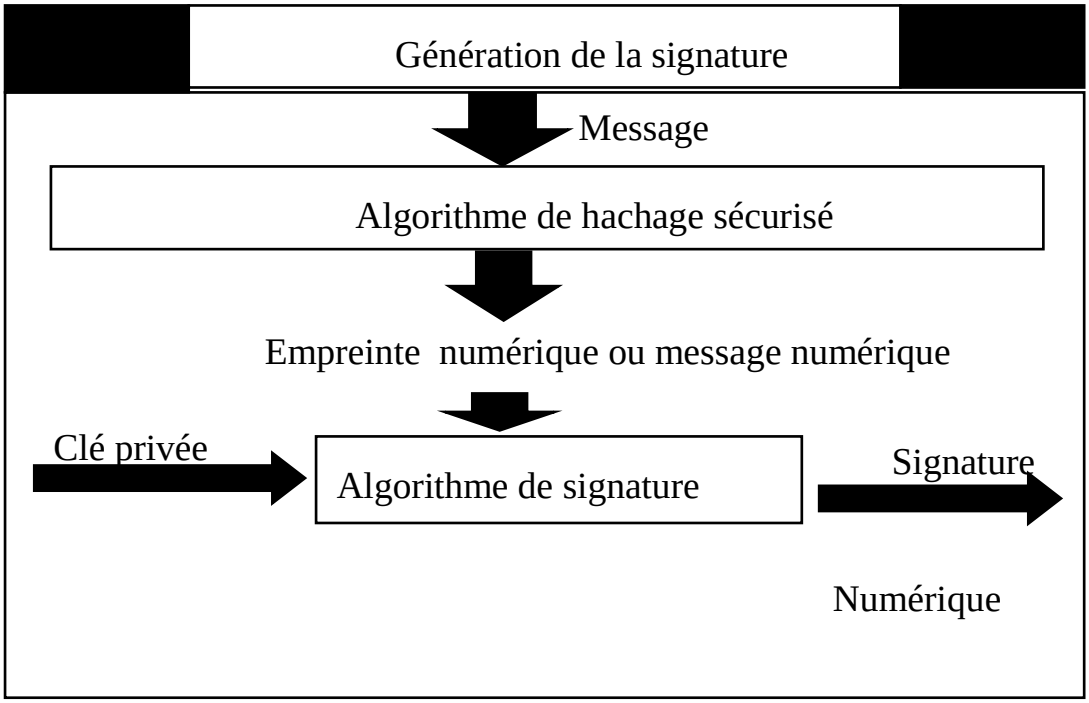


Figure 29 : Génération de la signature

IV.5.6 Vérification de signature

Pour vérifier une signature (*r*, *s*) sur *m*, un récepteur obtient une copie authentique du paramètre de domaine (*p*,(*a, b*),*G, n*) et la clé publique *Q*. et effectue par la suit les opérations suivantes :

- Calculer $e = h(m)$.
- Calculer $u_1 = e \cdot s^{-1} \bmod n$.
- Calculer $u_2 = r \cdot s^{-1} \bmod n$.
- Calculer $u_1 \cdot P + u_2 \cdot Q = (x_2, y_2)$.
- Calculer $v = x_2 \bmod n$.

La signature est acceptée si et seulement si $v = r$

On peut vérifier le fonctionnement de la signature de la façon suivante

Si l'on considère que la signature (*r*, *s*) d'un message *m* fut bel et bien générée par le signataire légitime. Alors $s = k - 1(e + d*r) \bmod n$. Donc le réarrangement nous donne :$k = s - 1(e + d*r) \bmod n \equiv s - 1 \cdot e + s - 1 \cdot d*r \equiv u\,1 + u\,2 \cdot d \bmod n$.

Par conséquent, nous avons

$X = u\,1 \cdot G + u\,(2) \cdot Q = e*s - 1P + r*s - 1dP = s - 1(e + r*d)*P = k*P = (x\,1, y1)$.

Et donc, $r = x\,1 \bmod n = v$ comme voulu.

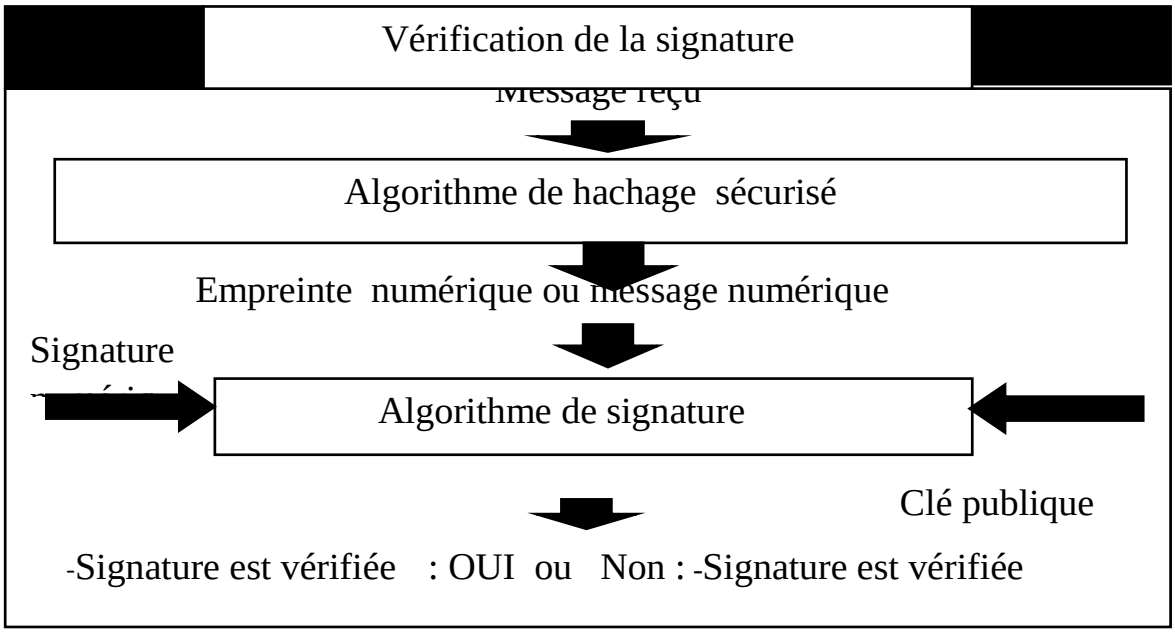


Figure 30: Vérification de la signature

IV.5.7Comparaison entre DSA et ECDSA

Conceptuellement, le ECDSA est simplement obtenue à partir de la DSA en remplaçant le sous-groupe de l'ordre q de $z_p$généré par g avec le sous-groupe de points sur une courbe elliptique qui sont générés par *G*. La seule différence significative entre ECDSA et DSA est dans la génération de *r*. Le DSA fait en prenant l'aléatoire $X = g^k \bmod p$, et les réductions modulo *q*, en obtenant ainsi un nombre entier dans l'intervalle $[1, q-1]$. L'ECDSA génère *r* dans l'intervalle $[1, n-1]$ en prenant la x-coordonnée du point aléatoire *KG* et en la réduisant modulo *n* [13] .

Voici deux tableaux montrant la correspondance entre les notations d'ECDSA et DSA du point du vu des paramètres de signatures.

Tableau 4 : Correspondance entre les notations de DSA et ECDSA [33]

Notations DSA	Notations ECDSA
q, g, x, y	n, P, d, n, Q

Maintenant, comme les courbes sont définies sur un groupe fini et respectant la loi du groupe, le tableau suivant montre la correspondance entre Correspondance entre F_p^*et le groupe *E* (Z_P).

Tableau 5 : Correspondance entre $\boldsymbol{F_p^*}$et le groupe *E* ($\boldsymbol{Z_P}$) [33].

Groupe	$\mathbb{Z}_p^*$	$\mathrm{E}(\mathbb{Z}_p)$
Elément du groupe	Ensemble des entiers $\{1,2,3,\dots,p-1\}$	Les points (x,y) de $\mathrm{E}(\mathrm{F}_p)$
La loi de composition	Multiplication modulo p	L'addition des points
Notation	Eléments : g, h Multiplication :g*h Inverse :g^{-1} Exponentiation :g^a	Points : P, Q
Problème du logarithme discret	Soient $\mathbb{Z}_p^*$ et h=g^amodulo p, trouver a	Soient P appartenant à $\mathrm{E}(\mathrm{F}_p)$ et Q= a*P, trouver a

L'ECDSA empêche la faiblesse trouvée dans DSA qui permet la falsification sélective d'un message si l'adversaire peut choisir les paramètres du système. En ECDSA, il est obligatoire de vérifier, au cours de la génération de signature, si le numérique signature (*r*, *s*) est non nul. Or, dans DSA, il est facultatif. On peut aussi comparer entre DSA et ECDSA en fonction de la taille de la clé ou du temps de signature ou de vérification ou du problème sur lequel repose la sécurité de l'algorithme utilisé. Le schéma ECDSA utilise des clés plus courtes et est le plus rapide en vérification et en signature par apport au schéma de DSA. L'avantage du schéma DSA par rapport au schéma ECDSA est que le problème du logarithme discret sur courbe elliptique est un problème plus récent et donc moins étudié que celui sur corps ni. De plus, le schéma DSA n'a pas de preuve des sécurités, contrairement à RSA et ECDSA.

Le tableau suivant présente une comparaison des temps de calcul en millisecondes pour une signature et une vérification de signature sur un ordinateur équipé d'un processeur Pentium *III* 900 MHz avec 256 Mo de RAM.

Tableau 6 : Comparaison des temps de calcul. [1]

	DSA			ECDSA		
Taille de clés en bits	680	1368	2704	112	160	224
Signature	10	20	80	5	5	25
Vérification	10	30	110	25	75	95

CONCLUSION

Ce chapitre m'a permis de comprendre les opérations arithmétiques utilisées par une courbe elliptique définie sur un champF_p. En effet, la multiplication d'un point par un scalaire est une opération élémentaire dans l'implémentation de la signature numérique basée sur les courbes elliptiques. Elle intervient lors de la génération de la clé publique, dans la génération de la signature ainsi que dans vérification de la signature. Des standards de ECDSA existent comme ANSI X9.62, FIPS 186-2, IEEEP1363, ISO /IEC 1488-3.La cryptographie basé sur les courbes elliptiques est une autre méthode de chiffrement des fichiers. Les points de la courbe sont utilisés pour le cryptage. Malgré les avantages d'ECDSA, des inconvénients ne manquent pas surtout il augmente la taille du message à crypter et son implémentation est difficile par rapport à la RSA .Ce qui peut engendrer des erreurs et l'algorithme devient moins sécurisant par rapport à ce qu'on y attendait du point de vue du niveau de sécurité. Du fait de la petite consommation de l'espace mémoire, il peut être utilisé dans des composants à faible consommation d'énergie électriques.

CONCLUSION ET RECOMMANDATIONS

Le travail était très intéressant, malgré la complexité de certains théorèmes mathématiques. En effet ; il m'a permis de revoir l'impact des théories mathématiques sur les systèmes informatiques en général, mais surtout en dans l'élaboration des algorithmes cryptographiques. Le premier chapitre m'a permis de comprendre le fonctionnement des systèmes informatiques et de découvrir différentes menaces de ceux-ci ainsi différentes techniques de sécurisation d'où l'intervention de la cryptographie comme outil de sécurisation du point de vue logique. Le deuxième chapitre est centré surtout sur la compréhension de la structure de groupes et leurs dérivées .En partant de la définition d'un ensemble, j'ai pu découvrir les propriétés d'un groupe abélien, le problème du logarithme discret, de factorisation ainsi que des exemples concrets. Le troisième chapitre est centre sur les opérations arithmétiques modulaires et en fin après avoir reçu les bases, le dernier chapitre est articulé sur l'opération des courbes elliptiques, l'étude de la signature numérique basé sur une courbe définie surF_P.Comme vous le trouverez dans l'annexe plus le nombre premier est grand, plus le nombre de points de la courbes augmente.

Mes recommandations reviennent au gouvernement d'investir dans la signature numérique basée numériques comme vu la tendance des nouvelles technologies de l'information et ses avantages du point de la consommation de l'espace mémoire. Effet, différentes services sont entrées de passer du monde physique vers le monde virtuel (e-Banking, e-gouverment, e-marketing, e-medecine,e-learning,..). L'investissement dans la signature numérique permettra de réduire les fraudes existant lors de l'utilisation de la signature manuscrite.

A l'université du Burundi, comme étant l'université de référence de s'investir de plus dans les nouvelles technologies de l'information, en amenant des matériels nécessaires au bon apprentissage des nouvelles technologies de l'information et de la communication. Mais aussi comme c'est une technologie récente, de créer une filière de sécurité informatique, afin de fournir à l'état des lauréats capable de faire face aux différents attaques informatiques. Il peut intégrer la technologie ECDSA dans le système d'échange électronique des informations entre les services.

Au lieu d'utiliser des signatures manuelles, qu'il utilise des signatures numériques basées sur les courbes elliptiques. J'invite à d'autres étudiants qu'ils puissent continuer la recherche sur l'implémentation de signatures numériques basées sur les courbes elliptiques dans des systèmes embarqués.

REFERENCES BIBLIOGRAPHIQUES

1. NASSER YASSINE, OUYOUS MINA, Rapport sur l'étude et l'implémentation de quelques algorithmes de chiffrent et de signature, UNIVERSITÉ MOHAMMED V-AGDAL, 2013-2014
2. Cours, Pierre Lavaurs- DEUG MIAS., Théories des ensembles, Université de Lyon, année 2001-2002
3. M.Gouy, G.Huvent, A.Ladureau : Groupe calculatrice, Irem de LILLE,12/mai/2002
4. National Institute of Standards and Technology. FIPS PUB 46-3: Data Encryption Standard (DES). National Institute for Standards and Technology, October 1999.
5. National Institute of Standards and Technology. FIPS PUB 186-2: Digital Signature Standard (DSS). National Institute for Standards and Technology, Gaithersburg, MD, USA, January2000.
6. W. Diffie and M.E.Hellman. New directions in cryptography. IEEE Transactions on Information Theory, IT-22(6): 644–654, November 1976.
7. V. Shoup. Lower bounds for discrete logarithms and related problems. InEUROCRYPT, pages 256–266, 1997.
8. U.M.Maurer. Towards the equivalence of breaking the Diffie-Hellman protocol and computing discrete algorithms. In Yvo G. Desmedt, editor, Advances in Cryptology—CRYPTO '94, volume 839 of Lecture Notes in Computer Science, pages 271–281. Springer-Verlag, 21–25 August 1994.
9. N. Koblitz. Elliptic curve cryptosystems. Mathematics of Computation, 48(177):203–209, January 1987.
10. J. H. Silverman. The Arithmetic of Elliptic Curves, volume 106 of Graduate Texts in Mathematics. Springer-Verlag, 1986.
11. J. H. Silverman and J. Tate. Rational Points on Elliptic Curves. Springer-Verlag, 1992.
12. T. ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In G. R. Blakley and David Chaum, editors, Advances in Cryptology: Proceedings of CRYPTO '84, volume 196 of LNCS, pages 10–18. Springer-Verlag, 1985.
13. M. H. Sherif .Paiements électroniques sécurisés .PPUR presses polytechniques, 2007
14. R.A.Mollin.An Introduction to Cryptography, Second Edition. CRC Press, 12/décembre/2012

15. P.Barthélemy, R, R. Rolland,P. Véron. Cryptographie. Collection informatique dirigée par Jean-Charles Pomero.
16. M.S. Rhee, B.Lee. Information Security and Cryptology-ICISC 2006.Springer, 14 November. 2006.
17. K. M. Martin. Everyday Cryptography: Fundamental Principles RSA and Applications. Oxford University Press, 1 mars 2012.
18. S. Katzenbeisser .Recent Advances in RSA Cryptography. Springer, 30 Sept.2001
19. T. Favre .Cryptographie et courbes elliptiques. Master's thesis, Ecole polytechnique fédérale de LAUSANNE, 2011
20. Thomas Plantard. Arithmétique modulaire pour la cryptographie. Autre [Cs. OH]. Université Montpellier II - Sciences et Techniques du Languedoc, 2005. Français
21. R. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public key cryptosystems. Communications of the ACM, 21(2):120-126,Feb 1978
22. A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, editors. Handbook of Applied Cryptography. CRC Press, 1996
23. J. Omura. A public key cell design for smart card chips. In IT Workshop, Hawaii, USA, November 27–30, pages 983–985, 1990.
24. P. L. Montgomery. Modular multiplication without trial division. Mathematics of Computation,44(170) :519–521, Apr 1985.
25. D. E. Knuth. The Art of Computer Programming, Vol. 2. Seminumerical Algorithms. Addison Wesley, Reading, Mass., 1981
26. P. Barrett. Implementing the Rivest Shamir and Adleman public key encryption algorithm on a standard digital signal processor. In A. M. Odlyzko, editor, Advances in Cryptology –CRYPTO *'86*, volume 263 of *LNCS*, pages 311–326. Springer-Verlag, 1986.
27. G. R. Blakley. A Computer Algorithm for Calculating the Product AB modulo M. IEEE Transactions on Computers, C-32: 497–500, 1983.
28. H. Cohen. A course in computational algebraic number theory, volume 138 of Graduate Texts in Mathematics. Springer-Verlag, 1993.
29. A. Khalique, K. Singh, S. Sood. Implementation of Elliptic Curve Digital Signature Algorithm .International Journal of Computer Applications (0975 - 8887).

30. N. Takagi. A radix-4 modular multiplication algorithm for modular exponentiation. IEEE Transactions on Computers, 41(8) :949–956, August 1992
31. H. Liao, Y.Shen. On the Elliptic Curve Digital Signature Algorithm.Tunghai Science Vol, July, 2006.
32. R.Schoof, Elliptic curve over finite field and the computation of square roots mod p, Math. comp.44 (1985), 483-494.
33. M.dion .Implantation d'ECDSA sur une carte à puce. Université de Montréal, MAI 1999.

ANNEXE

Run:

(0;753)
(0;1124)
(1;16)
(1;1861)
(3;222)
(3; 1655)
(4;109)
(4;1768)
(8;623)
(8;1254)
(9;341)
(9;1536)
(10;634)
(10;1243)
(14;275)
(14;1602)
(15;833)
(15;1044)
(16;859)
(16;1018)
(19;480)
(19;1397)
(22;324)
(22;1553)
(24;372)
(24;1505)
(25;202)
(25;1675)
(32;805)
(32;1072)
(33;685)
(33;1192)
(35;335)
(35;1542)
(36;823)
(36;1054)
(38;791)
(38;1086)
(40;200)
(40;1677)
(42;387)
(42;1490
(43;224)
(43;1653)
(44;161)
(44;1716)
(48;384)
(48;1493)
(51;153)
(51;1724)
(52;354)
(52;1523)
(54;263)
(54;1614)
(56;865)
(56;1012)
(57;629)
(57;1248)
(58;409)
(58;1468)
(59;853)
(59;1024)
(60;146)
(60;1731)
(68;461)
(68;1416)
(70;866)
(70;1011)
(75;270)
(75;1607)
(77;80)
(77;1797)
(79;156)
(79;1721)
(81;741)
(81;1136)
(82;242)
(82;1635)
(84;109)
(84;1768)
(85;600)
(85;1277)
(86;419)
(86;1458)
(89;506)
(89;1371)
(91;506)
(91;1371)
(92;183)
(92;1694)
(94;736)
(94;1141)
(95;233)
(95;1644)
(96;54)
(96;1823)
(97;535)
(97;1342)
(101;254)
(101;1623)
(102;349)
(102;1528)
(105;719)
(105;1158)
(106;180)
(106;1697)
(108;431)
(108;1446)
(110;10)
(110;1867)
(112;396)
(112;1481)
(113;149)
(113;1728)
(114;279)
(114;1598)
(118;211)
(118;1666)
(119;169)
(119;1708)
(124;153)
(124;1724)
(126;403)
(126;1474)
(128;500)
(128;1377)
(129;879)
(129;998)
(131;933)
(131;944)
(133;262)
(133;1615)
(135;12)
(135;1865)
(137;846)
(137;1031)
(140;687)
(140;1190)
(142;758)
(142;1119)
(144;82)
(144;1795)

(145;4)
(145;1873)
(146;875)
(146;1002)
(148;583)
(148;1294)
(152;368)
(152;1509)
(153;577)
(153;1300)
(156;548)
(156;1329)
(159;199)
(159;1678)
(160;435)
(160;1442)
(163;184)
(163;1693)
(164;399)
(164;1478)
(170;521)
(170;1356)
(177;406)
(177;1471)
(179;513)
(179;1364)
(180;895)
(180;982)
(182;364)
(182;1513)
(183;815)
(183;1062)
(185;513)
(185;1364)
(186;195)
(186;1682)
(188;151)
(188;1726)
(189;775)
(189;1102)
(190;52)
(190;1825)
(191;53)
(191;1824)
(192;887)
(192;990)
(199;19)
(199;1858)
(200;167)
(200;1710)
(202;625)
(202;1252)
(205;183)
(205;1694)
(206;67)
(206;1810)
(209;3)
(209;1874)
(210;438)
(210;1439)
(211;712)
(211;1165)
(213;823)
(213;1054)
(220;71)
(220;1806)
(221;687)
(221;1190)
(222;60)
(222;1817)
(223;402)
(223;1475)
(225;736)
(225;1141)
(226;114)
(226;1763)
(230;569)
(230;1308)
(231;902)
(231;975)
(234;399)
(234;1478)
(235;934)
(235;943)
(236;621)
(236;1256)
(239;583)
(239;1294)
(240;735)
(240;1142)
(241;572)
(241;1305)
(242;727)
(242;1150)
(244;669)
(244;1208)
(246;421)
(246;1456)
(251;858)
(251;1019)
(254;914)
(254;963)
(256;873)
(256;1004)
(257;545)
(257;1332)
(258;885)
(258;992)
(259;174)
(259;1703)
(260;735)
(260;1142)
(261;423)
(261;1454)
(262;232)
(262;1645)
(263;312)
(263;1565)
(265;576)
(265;1301)
(267;915)
(267;962)
(270;749)
(270;1128)
(271;568)
(271;1309)
(272;938)
(272;939)
(274;913)
(274;964)
(275;727)
(275;1150)
(278;79)
(278;1798)
(280;195)
(280;1682)
(284;349)
(284;1528)
(285;666)
(285;1211)
(286;417)
(286;1460)
(288;208)
(288;1669)

(295;440)
(295;1437)
(297;869)
(297;1008)
(300;722)
(300;1155)
(301;253)
(301;1624)
(303;937)
(303;940)
(304;444)
(304;1433)
(305;885)
(305;992)
(309;144)
(309;1733)
(310;363)
(310;1514)
(311;836)
(311;1041)
(313;898)
(313;979)
(314;172)
(314;1705)
(316;429)
(316;1448)
(319;796)
(319;1081)
(320;661)
(320;1216)
(322;350)
(322;1527)
(324;505)
(324;1372)
(326;34)
(326;1843)
(327;194)
(327;1683)
(328;803)
(328;1074)
(332;284)
(332;1593)
(335;239)
(335;1638)
(337;235)
(337;1642)
(338;741)
(338;1136)
(339;795)
(339;1082)
(341;540)
(341;1337)
(343;905)
(343;972)
(346;655)
(346;1222)
(350;189)
(350;1688)
(351;601)
(351;1276)
(353;825)
(353;1052)
(355;673)
(355;1204)
(357;761)
(357;1116)
(359;504)
(359;1373)
(362;160)
(362;1717)
(363;603)
(363;1274)
(365;604)
(365;1273)
(366;380)
(366;1497)
(372;87)
(372;1790)
(374;277)
(374;1600)
(375;650)
(375;1227)
(377;692)
(377;1185)
(379;225)
(379;1652)
(380;226)
(380;1651)
(385;771)
(385;1106)
(386;285)
(386;1592)
(387;748)
(387;1129)
(388;264)
(388;1613)
(390;341)
(390;1536)
(391;478)
(391;1399)
(393;168)
(393;1709)
(395;331)
(395;1546)
(397;514)
(397;1363)
(399;319)
(399;1558)
(400;146)
(400;1731)
(402;559)
(402;1318)
(404;437)
(404;1440)
(405;846)
(405;1031)
(406;522)
(406;1355)
(407;164)
(407;1713)
(408;335)
(408;1542)
(409;854)
(409;1023)
(411;4)
(411;1873)
(413;603)
(413;1274)
(417;605)
(417;1272)
(420;404)
(420;1473)
(421;326)
(421;1551)
(422;187)
(422;1690)
(425;248)
(425;1629)
(426;19)
(426;1858)
(427;245)
(427;1632)
(428;838)
(428;1039)

(431;95)
(431;1782)
(434;450)
(434;1427)
(435;166)
(435;1711)
(439;682)
(439;1195)
(440;82)
(440;1795)
(441;38)
(441;1839)
(444;701)
(444;1176)
(448;128)
(448;1749)
(449;132)
(449;1745)
(451;878)
(451;999)
(452;655)
(452;1222)
(453;558)
(453;1319)
(455;693)
(455;1184)
(457;530)
(457;1347)
(460;928)
(460;949)
(462;502)
(462;1375)
(466;831)
(466;1046)
(469;674)
(469;1203)
(470;27)
(470;1850)
(471;222)
(471;1655)
(475;776)
(475;1101)
(476;596)
(476;1281)
(477;613)
(477;1264)
(480;438)
(480;1439)
(482;749)
(482;1128)
(484;244)
(484;1633)
(489;140)
(489;1737)
(490;68)
(490;1809)
(491;194)
(491;1683)
(492;463)
(492;1414)
(498;275)
(498;1602)
(504;166)
(504;1711)
(505;419)
(505;1458)
(507;753)
(507;1124)
(508;863)
(508;1014)
(509;291)
(509;1586)
(510;179)
(510;1698)
(512;488)
(512;1389)
(514;339)
(514;1538)
(516;368)
(516;1509)
(518;83)
(518;1794)
(520;841)
(520;1036)
(521;277)
(521;1600)
(522;93)
(522;1784)
(524;681)
(524;1196)
(525;927)
(525;950)
(526;459)
(526;1418)
(527;905)
(527;972)
(530;623)
(530;1254)
(531;645)
(531;1232)
(533;258)
(533;1619)
(534;432)
(534;1445)
(541;151)
(541;1726)
(542;216)
(542;1661)
(544;618)
(544;1259)
(545;398)
(545;1479)
(546;429)
(546;1448)
(547;630)
(547;1247)
(550;522)
(550;1355)
(556;599)
(556;1278)
(557;288)
(557;1589)
(558;890)
(558;987)
(561;77)
(561;1800)
(562;631)
(562;1246)
(564;181)
(564;1696)
(565;927)
(565;950)
(568;38)
(568;1839)
(570;95)
(570;1782)
(574;661)
(574;1216)
(575;836)
(575;1041)
(576;935)
(576;942)
(578;253)
(578;1624)

(579;915)
(579;962)
(583;540)
(583;1337)
(584;323)
(584;1554)
(586;645)
(586;1232)
(588;465)
(588;1412)
(589;387)
(589;1490)
(590;301)
(590;1576)
(591;783)
(591;1094)
(592;450)
(592;1427)
(594;179)
(594;1698)
(596;263)
(596;1614)
(598;407)
(598;1470)
(599;776)
(599;1101)
(604;601)
(604;1276)
(607;702)
(607;1175)
(608;258)
(608;1619)
(609;66)
(609;1811)
(610;716)
(610;1161)
(613;65)
(613;1812)
(614;794)
(614;1083)
(615;582)
(615;1295)
(617;574)
(617;1303)
(619;81)
(619;1796)
(622;863)
(622;1014)
(623;288)
(623;1589)
(624;276)
(624;1601)
(625;764)
(625;1113)
(629;639)
(629;1238)
(630;586)
(630;1291)
(631;827)
(631;1050)
(632;841)
(632;1036)
(634;934)
(634;943)
(637;804)
(637;1073)
(640;791)
(640;1086)
(642;392)
(642;1485)
(643;273)
(643;1604)
(644;714)
(644;1163)
(645;320)
(645;1557)
(648;155)
(648;1722)
(649;916)
(649;961)
(650;342)
(650;1535)
(652;786)
(652;1091)
(654;515)
(654;1362)
(655;330)
(655;1547)
(656;589)
(656;1288)
(661;398)
(661;1479)
(662;876)
(662;1001)
(666;346)
(666;1531)
(670;832)
(670;1045)
(671;398)
(671;1479)
(672;16)
(672;1861)
(673;793)
(673;1084)
(674;660)
(674;1217)
(675;562)
(675;1315)
(676;679)
(676;1198)
(677;440)
(677;1437)
(678;635)
(678;1242)
(679;561)
(679;1316)
(680;177)
(680;1700)
(681;287)
(681;1590)
(683;682)
(683;1195)
(687;315)
(687;1562)
(688;604)
(688;1273)
(692;856)
(692;1021)
(693;892)
(693;985)
(694;255)
(694;1622)
(696;629)
(696;1248)
(697;288)
(697;1589)
(699;731)
(699;1146)
(700;930)
(700;947)
(701;88)
(701;1789)
(703;403)
(703;1474)

(705;689)
(705;1188)
(706;493)
(706;1384)
(708;547)
(708;1330)
(709;172)
(709;1705)
(712;239)
(712;1638)
(713;563)
(713;1314)
(715;642)
(715;1235)
(717;627)
(717;1250)
(718;666)
(718;1211)
(720;380)
(720;1497)
(722;367)
(722;1510)
(723;838)
(723;1039)
(724;362)
(724;1515)
(725;841)
(725;1036)
(726;838)
(726;1039)
(728;37)
(728;1840)
(729;849)
(729;1028)
(730;879)
(730;998)
(731;672)
(731;1205)
(732;422)
(732;1455)
(733;337)
(733;1540)
(735;600)
(735;1277)
(736;258)
(736;1619)
(741;163)
(741;1714)
(742;685)
(742;1192)
(743;467)
(743;1410)
(744;280)
(744;1597)
(747;863)
(747;1014)
(750;455)
(750;1422)
(751;364)
(751;1513)
(753;737)
(753;1140)
(754;198)
(754;1679)
(755;682)
(755;1195)
(757;815)
(757;1062)
(760;645)
(760;1232)
(761;217)
(761;1660)
(764;444)
(764;1433)
(765;640)
(765;1237)
(767;507)
(767;1370)
(768;818)
(768;1059)
(769;751)
(769;1126)
(770;920)
(770;957)
(771;452)
(771;1425)
(773;179)
(773;1698)
(774;353)
(774;1524)
(778;231)
(778;1646)
(782;180)
(782;1697)
(785;654)
(785;1223)
(786;13)
(786;1864)
(787;927)
(787;950)
(791;380)
(791;1497)
(796;43)
(796;1834)
(800;80)
(800;1797)
(802;227)
(802;1650)
(803;776)
(803;1101)
(807;822)
(807;1055)
(808;307)
(808;1570)
(809;444)
(809;1433)
(810;421)
(810;1456)
(811;28)
(811;1849)
(812;0)
(813;238)
(813;1639)
(814;203)
(814;1674)
(817;96)
(817;1781)
(818;628)
(818;1249)
(820;659)
(820;1218)
(821;421)
(821;1456)
(824;604)
(824;1273)
(826;634)
(826;1243)
(827;1)
(827;1876)
(828;53)
(828;1824)
(830;239)
(830;1638)
(831;923)

(831;954)
(832;267)
(832;1610)
(833;740)
(833;1137)
(834;774)
(834;1103)
(835;684)
(835;1193)
(840;454)
(840;1423)
(844;834)
(844;1043)
(845;223)
(845;1654)
(849;246)
(849;1631)
(850;394)
(850;1483)
(851;450)
(851;1427)
(852;220)
(852;1657)
(854;172)
(854;1705)
(858;53)
(858;1824)
(859;683)
(859;1194)
(862;63)
(862;1814)
(863;347)
(863;1530)
(864;241)
(864;1636)
(865;901)
(865;976)
(868;38)
(868;1839)
(870;788)
(870;1089)
(873;458)
(873;1419)
(874;666)
(874;1211)
(875;512)
(875;1365)
(876;95)
(876;1782)
(877;415)
(877;1462)
(878;519)
(878;1358)
(880;361)
(880;1516)
(881;70)
(881;1807)
(886;617)
(886;1260)
(888;678)
(888;1199)
(889;584)
(889;1293)
(896;344)
(896;1533)
(897;658)
(897;1219)
(898;670)
(898;1207)
(899;598)
(899;1279)
(900;936)
(900;941)
(903;57)
(903;1820)
(904;746)
(904;1131)
(905;440)
(905;1437)
(906;240)
(906;1637)
(907;46)
(907;1831)
(908;196)
(908;1681)
(911;303)
(911;1574)
(913;877)
(913;1000)
(916;86)
(916;1791)
(919;732)
(919;1145)
(921;522)
(921;1355)
(922;601)
(922;1276)
(924;709)
(924;1168)
(925;324)
(925;1553)
(927;725)
(927;1152)
(928;44)
(928;1833)
(929;297)
(929;1580)
(930;324)
(930;1553)
(931;926)
(931;951)
(932;592)
(932;1285)
(933;549)
(933;1328)
(937;815)
(937;1062)
(938;166)
(938;1711)
(942;490)
(942;1387)
(944;364)
(944;1513)
(946;50)
(946;1827)
(947;768)
(947;1109)
(952;768)
(952;1109)
(953;540)
(953;1337)
(956;411)
(956;1466)
(958;556)
(958;1321)
(962;105)
(962;1772)
(963;497)
(963;1380)
(967;152)
(967;1725)
(972;830)
(972;1047)
(973;236)

(973;1641)
(975;152)
(975;1725)
(976;757)
(976;1120)
(977;91)
(977;1786)
(980;289)
(980;1588)
(981;15)
(981;1862)
(982;277)
(982;1600)
(983;661)
(983;1216)
(986;552)
(986;1325)
(989;180)
(989;1697)
(990;580)
(990;1297)
(991;836)
(991;1041)
(993;14)
(993;1863)
(994;142)
(994;1735)
(996;443)
(996;1434)
(997;824)
(997;1053)
(998;253)
(998;1624)
(999;499)
(999;1378)
(1000;80)
(1000;177)
(1001;728)
(1001;119)
(1003;136)
(1003;171)
(1005;653)
(1005;124)
(1006;17)
(1006;180)
(1007;905)
(1007;972)
(1008;934)
(1008;943)
(1013;99)
(1013;178)
(1015;429)
(1015;148)
(1016;371)
(1016;156)
(1018;879)
(1018;998)
(1019;283)
(1019;154)
(1024;458)
(1024;149)
(1027;676)
(1027;121)
(1028;553)
(1028;134)
(1031;915)
(1031;962)
(1036;588)
(1036;129)
(1038;697)
(1038;110)
(1039;206)
(1039;161)
(1041;634)
(1041;123)
(1047;460)
(1047;147)
(1048;403)
(1048;144)
(1049;283)
(1049;154)
(1054;816)
(1054;101)
(1055;700)
(1055;117)
(1056;293)
(1056;154)
(1057;600)
(1057;127)
(1059;194)
(1059;163)
(1060;828)
(1060;109)
(1063;847)
(1063;100)
(1064;150)
(1064;177)
(1067;293)
(1067;154)
(1068;829)
(1068;108)
(1069;919)
(1069;958)
(1070;792)
(1070;105)
(1074;762)
(1074;115)
(1075;896)
(1075;981)
(1076;554)
(1076;133)
(1077;415)
(1077;142)
(1078;491)
(1078;136)
(1079;655)
(1079;122)
(1082;329)
(1082;158)
(1083;441)
(1083;146)
(1084;647)
(1084;120)
(1086;185)
(1086;162)
(1087;360)
(1087;157)
(1091;560)
(1091;137)
(1092;117)
(1092;170)
(1095;678)
(1095;119)
(1096;806)
(1096;101)
(1099;852)
(1099;105)
(1100;385)
(1100;142)
(1101;603)
(1101;124)
(1102;685)
(1102;112)
(1103;626)

(1103;121)
(1106;218)
(1106;169)
(1107;190)
(1107;167)
(1109;343)
(1109;154)
(1110;886)
(1110;991)
(1112;212)
(1112;165)
(1113;829)
(1113;108)
(1115;145)
(1115;172)
(1116;333)
(1116;154)
(1119;912)
(1119;965)
(1121;725)
(1121;112)
(1123;907)
(1123;970)
(1124;629)
(1124;128)
(1125;749)
(1125;118)
(1126;549)
(1126;138)
(1129;457)
(1129;140)
(1132;491)
(1132;136)
(1137;633)
(1137;124)
(1141;464)
(1141;143)
(1142;659)
(1142;128)
(1143;470)
(1143;147)
(1145;483)
(1145;134)
(1147;683)
(1147;114)
(1148;151)
(1148;176)
(1149;191)
(1149;166)
(1150;170)
(1150;177)
(1151;8)
(1151;189)
(1152;680)
(1152;117)
(1153;434)
(1153;143)
(1154;8)
(1154;189)
(1155;845)
(1155;102)
(1156;633)
(1156;124)
(1157;185)
(1157;162)
(1158;466)
(1158;141)
(1159;136)
(1159;171)
(1161;139)
(1161;178)
(1162;171)
(1162;176)
(1163;286)
(1163;151)
(1165;460)
(1165;147)
(1166;922)
(1166;955)
(1170;925)
(1170;952)
(1180;446)
(1180;141)
(1181;737)
(1181;110)
(1187;438)
(1187;149)
(1191;766)
(1191;111)
(1192;476)
(1192;141)
(1195;303)
(1195;154)
(1196;23)
(1196;184)
(1197;778)
(1197;109)
(1198;826)
(1198;101)
(1199;791)
(1199;106)
(1200;830)
(1200;107)
(1202;143)
(1202;174)
(1204;16)
(1204;181)
(1205;793)
(1205;104)
(1208;188)
(1208;169)
(1209;368)
(1209;159)
(1210;25)
(1210;182)
(1213;14)
(1213;183)
(1218;90)
(1218;177)
(1220;186)
(1220;161)
(1222;336)
(1222;151)
(1224;508)
(1224;139)
(1227;263)
(1227;164)
(1228;575)
(1228;132)
(1230;554)
(1230;133)
(1237;635)
(1237;122)
(1239;774)
(1239;113)
(1245;680)
(1245;117)
(1246;387)
(1246;140)
(1250;28)
(1250;189)
(1251;290)
(1251;157)
(1252;19)

(1252;188)
(1253;899)
(1253;978)
(1254;446)
(1254;141)
(1256;832)
(1256;105)
(1257;580)
(1257;127)
(1258;148)
(1258;179)
(1259;427)
(1259;140)
(1261;911)
(1261;966)
(1265;286)
(1265;151)
(1269;464)
(1269;143)
(1271;302)
(1271;155)
(1272;215)
(1272;162)
(1274;892)
(1274;985)
(1276;196)
(1276;161)
(1277;129)
(1277;178)
(1278;762)
(1278;115)
(1279;921)
(1279;956)
(1280;834)
(1280;103)
(1281;342)
(1281;155)
(1286;419)
(1286;148)
(1288;827)
(1288;100)
(1289;327)
(1289;150)
(1290;112)
(1290;175)
(1293;82)
(1293;175)
(1294;709)
(1294;118)
(1295;653)
(1295;124)
(1296;750)
(1296;117)
(1297;100)
(1297;177)
(1300;670)
(1300;127)
(1302;617)
(1302;120)
(1304;716)
(1304;111)
(1306;872)
(1306;105)
(1307;728)
(1307;119)
(1308;220)
(1308;167)
(1310;467)
(1310;140)
(1311;234)
(1311;163)
(1313;842)
(1313;105)
(1314;885)
(1314;992)
(1318;251)
(1318;166)
(1320;446)
(1320;141)
(1321;4)
(1321;183)
(1326;286)
(1326;151)
(1327;411)
(1327;146)
(1331;63)
(1331;184)
(1333;755)
(1333;112)
(1334;392)
(1334;145)
(1335;846)
(1335;101)
(1336;849)
(1336;108)
(1338;65)
(1338;182)
(1339;623)
(1339;124)
(1340;609)
(1340;128)
(1342;281)
(1342;156)
(1343;259)
(1343;168)
(1344;464)
(1344;143)
(1348;269)
(1348;168)
(1350;788)
(1350;109)
(1353;529)
(1353;138)
(1354;763)
(1354;114)
(1357;680)
(1357;117)
(1359;210)
(1359;167)
(1360;727)
(1360;110)
(1364;101)
(1364;176)
(1365;275)
(1365;162)
(1366;92)
(1366;175)
(1368;294)
(1368;153)
(1370;753)
(1370;114)
(1372;783)
(1372;104)
(1374;760)
(1374;117)
(1375;819)
(1375;108)
(1377;735)
(1377;112)
(1379;488)
(1379;139)
(1383;534)
(1383;133)
(1385;644)

(1385;123)
(1386;628)
(1386;129)
(1389;688)
(1389;119)
(1393;110)
(1393;177)
(1399;223)
(1399;164)
(1400;397)
(1400;140)
(1402;762)
(1402;115)
(1403;222)
(1403;165)
(1408;131)
(1408;176)
(1411;195)
(1411;162)
(1413;702)
(1413;115)
(1414;657)
(1414;120)
(1417;146)
(1417;171)
(1418;794)
(1418;103)
(1419;237)
(1419;160)
(1420;881)
(1420;996)
(1421;351)
(1421;156)
(1423;598)
(1423;129)
(1424;358)
(1424;159)
(1428;597)
(1428;120)
(1432;598)
(1432;129)
(1433;295)
(1433;152)
(1434;335)
(1434;152)
(1435;632)
(1435;125)
(1437;323)
(1437;154)
(1441;250)
(1441;167)
(1444;731)
(1444;116)
(1446;728)
(1446;119)
(1448;554)
(1448;133)
(1449;8)
(1449;189)
(1451;764)
(1451;113)
(1452;868)
(1452;109)
(1453;855)
(1453;102)
(1454;653)
(1454;124)
(1458;741)
(1458;116)
(1461;633)
(1461;124)
(1463;848)
(1463;109)
(1465;582)
(1465;125)
(1466;599)
(1466;128)
(1467;639)
(1467;128)
(1471;411)
(1471;146)
(1472;216)
(1472;161)
(1475;408)
(1475;149)
(1476;27)
(1476;180)
(1477;928)
(1477;949)
(1478;341)
(1478;156)
(1479;399)
(1479;148)
(1480;811)
(1480;106)
(1482;620)
(1482;127)
(1487;319)
(1487;158)
(1489;390)
(1489;147)
(1490;583)
(1490;124)
(1491;349)
(1491;158)
(1495;546)
(1495;131)
(1496;790)
(1496;107)
(1497;518)
(1497;139)
(1500;9)
(1500;188)
(1502;565)
(1502;132)
(1507;580)
(1507;127)
(1510;223)
(1510;164)
(1511;185)
(1511;162)
(1513;513)
(1513;134)
(1515;528)
(1515;139)
(1516;687)
(1516;110)
(1520;299)
(1520;158)
(1524;219)
(1524;168)
(1527;780)
(1527;107)
(1528;767)
(1528;110)
(1530;182)
(1530;165)
(1533;871)
(1533;106)
(1534;788)
(1534;109)
(1536;709)
(1536;118)
(1537;213)

(1537;164)
(1540;382)
(1540;145)
(1542;460)
(1542;147)
(1543;326)
(1543;151)
(1544;491)
(1544;136)
(1545;313)
(1545;154)
(1548;14)
(1548;183)
(1550;628)
(1550;129)
(1551;706)
(1551;111)
(1553;607)
(1553;120)
(1556;670)
(1556;127)
(1558;736)
(1558;111)
(1561;63)
(1561;184)
(1562;724)
(1562;113)
(1566;617)
(1566;120)
(1567;492)
(1567;135)
(1568;799)
(1568;108)
(1569;524)
(1569;133)
(1570;196)
(1570;161)
(1571;897)
(1571;980)
(1573;829)
(1573;108)
(1575;377)
(1575;150)
(1579;164)
(1579;173)
(1580;183)
(1580;164)
(1582;830)
(1582;107)
(1585;48)
(1585;189)
(1588;87)
(1588;170)
(1590;567)
(1590;130)
(1592;136)
(1592;171)
(1593;285)
(1593;152)
(1594;220)
(1594;167)
(1597;831)
(1597;106)
(1601;531)
(1601;136)
(1603;496)
(1603;131)
(1608;903)
(1608;974)
(1609;474)
(1609;143)
(1611;731)
(1611;116)
(1614;779)
(1614;108)
(1622;348)
(1622;159)
(1623;570)
(1623;137)
(1624;638)
(1624;129)
(1626;807)
(1626;100)
(1627;699)
(1627;118)
(1628;823)
(1628;104)
(1629;843)
(1629;104)
(1630;834)
(1630;103)
(1631;293)
(1631;154)
(1634;308)
(1634;159)
(1636;893)
(1636;984)
(1638;748)
(1638;119)
(1644;119)
(1644;178)
(1647;802)
(1647;105)
(1648;303)
(1648;154)
(1650;124)
(1650;173)
(1651;78)
(1651;179)
(1652;796)
(1652;101)
(1657;30)
(1657;187)
(1658;639)
(1658;128)
(1666;442)
(1666;145)
(1669;228)
(1669;169)
(1670;698)
(1670;119)
(1672;869)
(1672;108)
(1674;582)
(1674;125)
(1677;24)
(1677;183)
(1678;764)
(1678;113)
(1681;774)
(1681;113)
(1684;722)
(1684;115)
(1685;2)
(1685;185)
(1686;283)
(1686;154)
(1687;501)
(1687;136)
(1689;849)
(1689;108)
(1690;481)
(1690;136)
(1691;831)

(1691;106)
(1693;28)
(1693;189)
(1695;549)
(1695;138)
(1696;752)
(1696;115)
(1697;506)
(1697;131)
(1700;593)
(1700;124)
(1701;467)
(1701;140)
(1702;153)
(1702;174)
(1706;725)
(1706;112)
(1710;516)
(1710;131)
(1711;909)
(1711;968)
(1714;498)
(1714;139)
(1716;271)
(1716;166)
(1717;533)
(1717;134)
(1719;325)
(1719;152)
(1722;794)
(1722;103)
(1724;58)
(1724;189)
(1729;748)
(1729;119)
(1730;730)
(1730;117)
(1731;798)
(1731;109)
(1732;599)
(1732;128)
(1733;323)
(1733;154)
(1734;702)
(1734;115)
(1736;743)
(1736;114)
(1738;413)
(1738;144)
(1740;216)
(1740;161)
(1743;889)
(1743;988)
(1744;428)
(1744;149)
(1745;94)
(1745;173)
(1747;120)
(1747;177)
(1748;683)
(1748;114)
(1749;918)
(1749;959)
(1754;395)
(1754;142)
(1756;39)
(1756;188)
(1758;300)
(1758;157)
(1759;624)
(1759;123)
(1761;298)
(1761;159)
(1766;102)
(1766;175)
(1768;164)
(1768;173)
(1769;416)
(1769;141)
(1770;722)
(1770;115)
(1771;678)
(1771;119)
(1775;285)
(1775;152)
(1778;392)
(1778;145)
(1780;566)
(1780;131)
(1781;55)
(1781;182)
(1782;610)
(1782;127)
(1783;796)
(1783;101)
(1785;869)
(1785;108)
(1786;895)
(1786;982)
(1787;892)
(1787;985)
(1788;895)
(1788;982)
(1789;109)
(1789;178)
(1790;326)
(1790;151)
(1791;783)
(1791;104)
(1792;659)
(1792;128)
(1794;87)
(1794;170)
(1795;527)
(1795;130)
(1798;316)
(1798;151)
(1800;415)
(1800;142)
(1803;65)
(1803;182)
(1804;229)
(1804;168)
(1807;817)
(1807;100)
(1808;27)
(1808;180)
(1811;214)
(1811;163)
(1812;152)
(1812;175)
(1813;352)
(1813;155)
(1817;928)
(1817;949)
(1818;356)
(1818;151)
(1820;737)
(1820;110)
(1821;317)
(1821;150)
(1823;342)
(1823;155)
(1825;649)

(1825;128)
(1827;808)
(1827;109)
(1828;832)
(1828;105)
(1830;268)
(1830;169)
(1833;487)
(1833;130)
(1834;334)
(1834;153)
(1835;827)
(1835;100)
(1836;637)
(1836;120)
(1837;59)
(1837;188)
(1839;635)
(1839;122)
(1840;716)
(1840;111)
(1843;646)
(1843;121)
(1845;375)
(1845;152)
(1846;384)
(1846;143)
(1847;107)
(1847;170)
(1848;612)
(1848;125)
(1849;41)
(1849;186)
(1855;768)
(1855;119)
(1857;458)
(1857;149)
(1858;272)
(1858;165)
(1860;384)
(1860;143)
(1861;62)
(1861;185)
(1863;488)
(1863;139)
(1864;477)
(1864;140)
(1866;765)
(1866;112)
(1868;319)
(1868;158)
(1870;379)
(1870;148)
(1876;793)
(1876;104)

Le nombre de points est 1926

BUILD SUCCESSFUL (total time: 2 second