République **A**lgérienne **D**émocratique et **P**opulaire
Ministère de l'**E**nseignement **S**upérieur et de la **R**echerche **S**cientifique

UNIVERSITE **M**OULOUD **M**AMMERI DE **T**IZI-**O**UZOU

FACULTE DE **G**ENIE **E**LECTRIQUE ET D'**I**NFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire de Fin d'Etudes de MASTER ACADEMIQUE

Domaine : **Mathématiques et Informatique**
Filière : **Informatique**
Spécialité : **Réseaux, Mobilité et Systèmes Embarqués**

Présenté par :
Amir BOUTOUCHENT

Thème

Crypto-système à courbe elliptique pour vote électronique.

Mémoire soutenu publiquement le 07/09/2016 devant le jury composé de :

Président : M. Mehammed DAOUI – UMM Tizi-ouzou
Encadreur : M^{me} Rebiha HADAOUI– UMM Tizi-ouzou
Examinatrice : M^{me} Rachida AOUDJIT – UMM Tizi-ouzou
Examinatrice : M^{me} Karima OUKFIF – UMM Tizi-ouzou

Table des matières

Résumé

Le vote est un acte de citoyenneté important dans une démocratie. Tout au long de l'histoire moderne, les différentes institutions responsables de l'organisation des élections ont reconnu l'existence de problémes dans leur systéme de vote (pertes de bulletins, votes de personnes non inscrite sur les listes électorale etc ...) cela engendre des résultats souvent contestés.

Le vote électronique (machine) est apparu pour répondre au besoin d'un systéme de vote plus sûr et plus crédible. Mais l'existence de certaines failles rendent ce systéme difficilement utilisable sur le internet, principalement à cause de la faible sécurité de chiffrement des votes.

Les systémes de vote en ligne sont alors apparue pour répondre ętte attente. a travers donc cette étude nous proposons le dévlopement d'une implémentation d'un systéme de chiffrement des votes électroniques plus précisèment en utilisant la cryptographie à courbes elliptiques. Cette étude permettra aussi d'étudier les questions de confidentialité, vérifiabilité et incoercibilité d'un vote notamment par le systéme de chiffrement d'El Gamal à base de courbes elliptiques.

Mots-clés : Cryptographie, Chiffrement, Courbes elliptique, Vote électronique, Sécutité informatique, El Gamal, Python, SageMath, Signature ECDSA.

abstract

Voting is one of the most important things an citizenship have in a democracy. Throughout modern history, the various institutions responsible for the organization of the elections recognized the probable existence of problems in their voting system (bulletins losses, people not included on the voting list ect .. .) that generates results often contested.

Electronic voting has come meet the need for a system of voting more secure, but there are certain flaws that make it difficult to use on the internet, mainly because of weak encryption votes.

The online voting systems like Helio are then appeared and in this work we propose an implementation of a cryptographc system of electronic votes more precisely using elliptic curve cryptography. This application will also investigate issues of confidentiality, verifiability and incoercibility in online voting including the encryption of El Gamal system based on elliptic curves.

Keywords : Cryptography, Elliptic curve cryptography, Network, EC El Gamal, IT security, e-Voting, ECDSA Signatur, Python, SageMath.

Remerciement

Je tiens à remercier en tout premier lieu ma promotrice Rebiha HADAOUI qui a dirigé ce travail de recherche de Master. Je la remercie pour tout ce qu'elle m'a apporté, pour ses conseils, sa présence, et sa patience, pour m'avoir fait confiance et m'avoir laissé la liberté nécessaire ł'accomplissement de mon travail, tout en y gardant un oeil critique et avisé.

Je remercie également toute l'equipe du département informatique de l'UMMTO et spécialement l'équipe RMSE, en particulier Monsieur M^r.DAOUI et M^r.DJAMAH qui ont toujours su me guider et m'aider dans mes choix d'avenir.

J'exprime ma reconnaissance aux membrex du jury, M.DAOUI, R.AOUDJIT, M.BELKADI pour la lecture de ce rapport de Master et pour leur jugement avisé.

Je remercie également tout mes ami(e)s : Amar, Yanis, Turkia, Sarah, Arezki, Moh, Katia et tous les autres, merci d'avoir été la pour moi dans les moments important.

Mes remerciements vont également à toute ma famille proche spécialement ma tente Lila et ses filles qui m'ont toujours apporté leurs soutiens et leurs encouragement et sur qui je peux toujours compter.

Mes plus grands remerciements sont réservés á mes parents et mon frére, ils m'ont toujours soutenu, encouragé, protégé et entouré de leurs affections. Je ne pourrais jamais les remercier assez.

Enfin je tiens à dédié ce modéste travail à la mémoire de mon grand pére, qui nous a quitté cette année, paix à son âme. Il m'a donné la volonté et la force pour affronter les nombreuses difficultées de la vie.

Introdution général

Tout au long de l'histoire moderne, les différentes institutions responsable de l'organisation des élections ont reconnu l'existence probable de problèmes dans leur système de vote comme par exemple : les pertes de bulletins, votes de personnes non inscrites sur les listes électorale, ect... .
a cela s'ajoute que parfois les électeures douter de la crédibilité des résultats, ce qui peux remetre en question les principes de la démocratie. C'est la raison pour laquelle notre interêt c'est porté sur l'existance eventuelle d'une autre procédure pour ôrganiser des élections plus sûr et plus crédible. Le vote électronique apparaît donc comme etant le meilleur moyen pour arriver à cet objectif.

Pour la mise en place de ce systéme de vote éctronique et plus précisement d'un systéme de vote en ligne, le niveau de sécurité du protocole de sécurité utilisé doit étre optimal. Une etude cryptographique en vue de l'élaboration d'un tel protocole de sécurité doit étre opréalable effectuée.

La cryptographie est une science au croisement des mathématiques, de l'informatique, et de la physique. Elle étudie l'ensemble de techniques permettant de chiffrer un message et de le rendre inintelligible sauf pour son destinataire : cette opération s'appelle le chiffrement. Elle fournit en oûtre un message chiffré ou cryptogramme [1] à partir d'un message en clair [2]. a l'inverse, le déchiffrement est l'action de reconstruire le texte en clair à partir du texte chiffré. Ces fonctions de chiffrement et déchiffrement sont des fonctions mathématiques appelées algorithmes cryptographiques [3], qui dépendent d'un paramètre appelé "clé de chiffrement".

L'objectif de notre étude est d'exposer les différentes problêmes que la cryptographie rencontre dans le cadre d'un vote électronique. Il y a lieu de mettre en évidence les différent procédés cryptographiques utilisées pour affronter ces problèmes liés au vote èlectronique, en utilisant les courbes elliptiques, du point de vue thèorique que pratique. Cette ètude a donné lieu à la conception d'un crypto-sytéme qui aura pour rôle de chiffrer, dèchiffrer, construire des preuves à connaissance nulle dans le domaine des courbes elliptiques et d'utiliser la signature électronique.

1. De l'anglais : Ciphertext
2. De l'anglais : Plaintext
3. Oú Crypto-systèmes

Pour mener à bien notre analyse, il été indispensable de décomposer notre travail en cinq chapitre : le premier est une introduction au monde des réseaux, et tout les concepts de sécurité en raport avec les réseaux. Le dexiéme à trait à lètat de l'art de la Cryptographie actuel, il traitera notament des principes nécessaire à la lecture de ce travail. Le Troisieme est consacré á létude théorique sur les systémes de chiffrement á courbes elliptique et ses fondements théoriques. Le quatriéme quant à lui porte sur le crypto-systéme de chiffrement à courbe elliptique élaboré et explique son fonctionnement. Enfin le dernier chapitre aura pour objectif de présenter l'implémentation du syséteme de vote et le module de chiffrement proposé sur SageMath.

Chapitre I

1 La sécurité réseaux

Introduction

Au cours de ce chapitre, il y a lieu tout d'abord d'expliquer les concepts fondamentaux sur les réseaux, puis d'une façon plus détaillée les différents aspects liés à la sécurité dans les réseaux pour traiter ensuite des mécanismes de protection contre les attaques.

La sécurité d'un système informatique est au coeur de la préoccupation des professionnels de informatiques. Ils se doivent de garantir la protection du réseau et l'intégrité des données à l'encontre des menaces qui évoluent en permanence. Il est important de faire face à l'évolution de ces menaces et de savoir y répondre.

1.1 Qu'es qu'un réseau ?

"Un réseau est un ensemble de moyens matériels et logiciels géographiquement dispersés destinés à offrir un service ou assurer le transport de données" [C.Servin Réseaux et Télécoms].

1.2 La naissance des réseaux

La justification de regrouper des ordinateurs en un réseau s'exprime avant tout par le désir de rompre l'isolement de ces ordinateurs. Le terme "réseau d'ordinateurs" évoque l'interconnexion de systémes isolés et indépendants ; il sous-entend que ces éléments communiquent par l'intermédiaire de lignes de transmission. Les réseaux ont évolué à partir des télécommunications terminal-ordinateur, l'objectif initial étant la connexion d'un terminal distant a un ordinateurcentral. Puis, le besoin de connexion ordinateurs - ordinateurs s'est fait sentir, besoin justifié par le désir d'un partage de ressources (fichiers, applications, etc ...). Dans le même temps, l'utilisation d'ordinateurs pour gérer les communications, dont les techniques se développaient et devenaient plus complexes, apparaissait. Ce cheminement a permis, á la fois, le développement des techniques de communication et le développement de la coopération entre systèmes d'exploitation.[29]

1.3 Les composants d'un réseau

Les composants d'un réseau sont les neuds et les lignes (ou voies) de communication. Les noeuds gérent les protocoles et fournissent des capacités de commutation. Un noeud est généralement un ordinateur (général ou spécial) qui exécute un protocole réseau. Les voies de communication ont plusieurs formes : paire torsadée, câble coaxial, fibre optique, ondes radio et lignes téléphoniques.[23]

1.4 Les principaux objectifs des réseaux

Objectifs des réseaux :

1. Partage des ressources : Rendre accessible à chacun les données, les programmes et équipements indépendamment de leur situation physique par rapport à l'utilisateur.[23]

2. Augmenter la fiabilité : Permettre des copies d'un même fichier sur plusieurs machines augmente la fiabilité face aux pannes d'une machine.[23]

3. Réduction des coûts : Plusieurs petits ordinateurs revient moins cher que de gros serveurs à performance égale.[18]

4. Médium de communications : Des personnes éloignées géographiquement peuvent travailler ensemble plus facilement.[23]

5. Travail coopératif

1.5 Structure des réseaux

Il existe dans le domaine des réseaux deux types structure de réseaux :

1.5.1 Réseau point à point

Il ont la caractéristique de pouvoir habrité un grand nombre de connexions entre machines. Les messages peuvent passer par plusieurs machines avant d'atteindre leur destination.La figure suivante montre les différentes façon de structuré les connexions.[23]

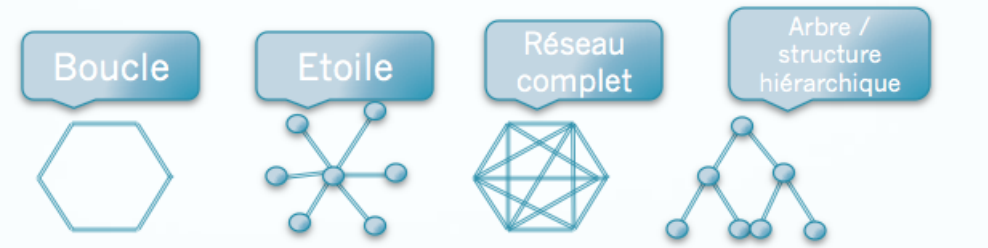


FIGURE 1.1 – Structure réseau point á point

1.5.2 Réseau à diffusion

Un réseau à diffusion (Broadcast ou à accés multiple) posséde un seul canal de communication, donc tout le monde entend le message de tout le monde. Un message est envoyé avec une adresse de destination et seul le destinataire répond. Dans ce cas aussi nous avons différentes structures possible illustré dans ce qui suit.[23]

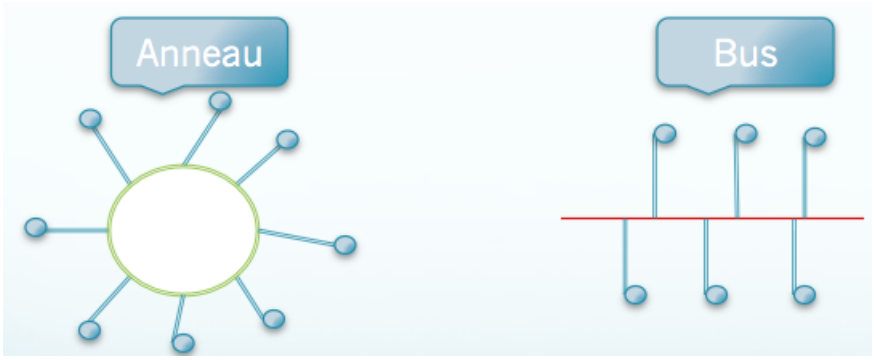


FIGURE 1.2 – Structure réseau á diffusion

1.6 Classifications des réseaux

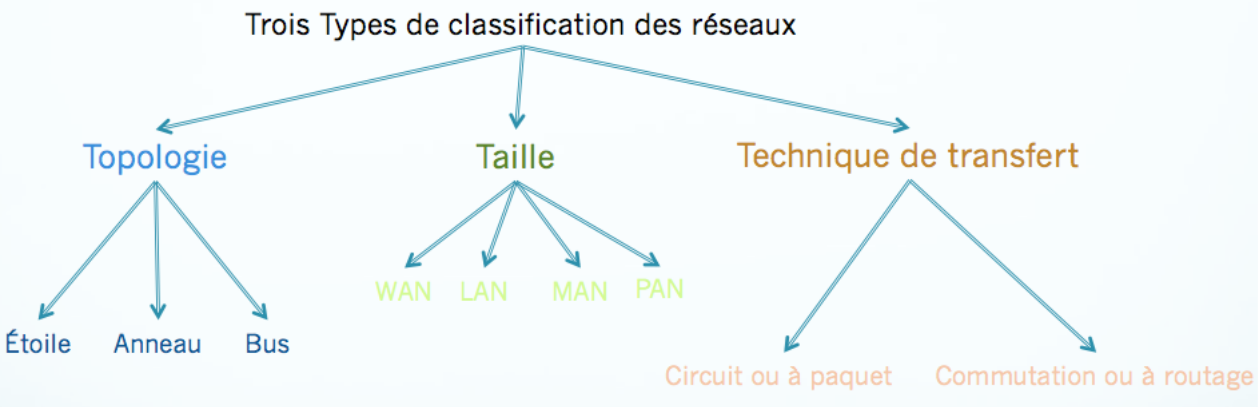


FIGURE 1.3 – arbre de classifications des réseaux

Nous pouvons donc classer les réseaux en 3 différentes catégories.

1.6.1 Classifications par tailles

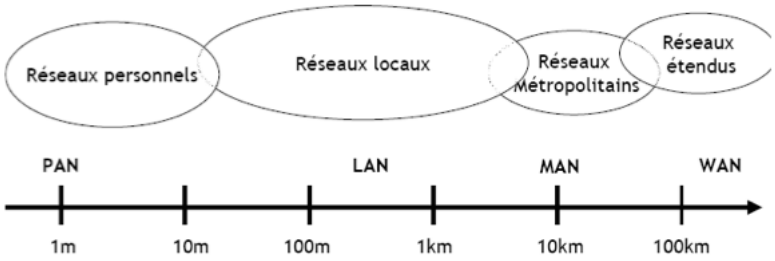


FIGURE 1.4 – Arbre de classifications des réseaux [23]

Un PAN [4] est un réseau personnel désigne un type de réseau informatique restreint en terme d'équipements, généralement mis en oeuvre dans un espace d'une dizaine de métres. D'autres appellations pour ce type de réseau sont : réseau domestique ou réseau individuel, car ils Interconnectent sur quelques métres des équipements personnels tels que les téléphones portables, PDA, oreillettes, auto-radio et posséde une couverture : de 10m à 100m avec débit : quelques Mbits/s.[29]

LAN [5]en français Réseau Local . Il s'agit d'un ensemble d'ordinateurs ap-

4. Personal Area Network
5. Local area Network

partenant à une même organisation et reliés entre eux dans une petite aire géographique par un réseau, souvent à l'aide d'une même technologie (la plus répandue étant Ethernet).Correspondent souvant aux réseaux d'entreprises, réseaux de campus ou équivalents.Les LaN posséde des tailles restreintes avec un débits de 10Mbps à 10Gbps et une couverture : de 100m à 1000m.les Topologies les plus utilisées sont : bus, anneau, étoile.[29]

Les MAN [6] interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de kms) á des débits importants. ainsi un MAN permet à deux noeuds distants de communiquer comme s'ils faisaient partie d'un même réseau local.Un MAN a une couverture de la taille d'une ville, et un débit de quelques dizaines de Mbits/s. sur tout utilié dans l'nterconnexion des entreprises, campus, et éventuellement de particuliers avec un haut débit qui sera redistribué en de moindres mesures aux extrémités, il est trés important dans le coeur de réseau d'une architecture.[29]

Un WAN [7] interconnecte plusieurs LAN á travers de grandes distances géographiques. Les WAN fonctionnent grâce à des routeurs qui permettent de choisir le trajet le plus approprié pour atteindre un noeud du réseau. C'est donc réseau longue distance (de l'ordre du pays) avec couverture Mondiale, peuvent être des réseaux terrestres (essentiellement de la fibre optique), ou hertziens (comme les réseaux satellitaires).Le plus connu des WaN est Internet.[29]

1.6.2 Classification par topologies

1. Topologie Logique : Prend en compte le mode d'échange des messages dans le réseau, elle est réalisée par un protocole d'accés comme par exemple Ethernet ou Token ring.[23]

2. Topologie Physique : Prend en compte cette fois le mode de raccordement des machines qui peux etre par Bus, étoiles etc ... et peux être comme expliquer précedanment de liaisons : point-á-point ou multipoints.[23]

1.6.3 Classification par la technique de transfert

Ce mode de classification comprend quant a lui deux mode de fonctionnement,pour faire transiter les informations : un mode Connecté (circuit) Non

6. Metropolitan area Network
7. Wide Area Network ou réseau étendu

connecté (routage).Le choix entre ces deux se fait en fonction du service demandé et du protocole utilisé.[23]

Services avec connexion (ou orienté connexion) : principe similaire au service téléphonique,le transfert d'information se déroule en 3 phases :

1. Établissement de la connexion.
2. Échange de données (avec ou sans séparation des frontières de messages).
3. libération de la connexion.

Services sans connexion [8] : principe similaire au courrier postal, on envoie des messages sans se soucier de la disponibilité du destinataire (ce genre de service est généralement utilisé pour l'émission de messages courts et urgents). Chaque datagramme (message ou paquet) contient l'adresse du (ou des) destinataire(s),les réseaux IP (dont Internet) ont font partie.[29]

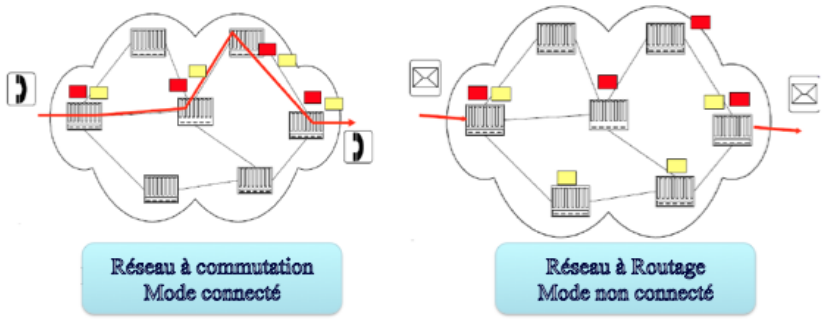


FIGURE 1.5 – Classification des éseaux par la technique de transfert

1.7 Le Modéle OSI

OSI [9] est un modéle en 7 couches pour linter-connexion des systémes ouverts. Il Permet à des équipements hétérogénes de communiquer entre eux.Chaque couche fournit des fonctions particulières pour préparer les informations á transmettre sur le réseau.

Le tableau suivant explique les rôles des différentes couche :[29]

8. ou Datagramme
9. Open System Interconnection

COUCHES	FONCTIONS
NIVEAU 1 **Couche Physique** ***Physical Layer***	La couche physique assure un transfert de **bits** sur le canal physique (support). À cet effet, elle définit les supports et les moyens d'y accéder : spécifications mécaniques (connecteur), spécifications électriques (niveau de tension), spécifications fonctionnelles des éléments de raccordement nécessaires à l'établissement, au maintien et à la libération de la ligne. Elle détermine aussi les moyens d'adaptation (ETCD).
NIVEAU 2 **Couche Liaison de données** ***Data Link Layer***	La couche liaison assure, sur la ligne, un service de transfert de blocs de données (**trames**) entre deux systèmes adjacents en assurant le contrôle, l'établissement, le maintien et la libération du lien logique entre les entités. Les protocoles de niveau 2 permettent, en outre, de détecter et de corriger les erreurs inhérentes aux supports physiques.
NIVEAU 3 **Couche Réseau** ***Network Layer***	La couche réseau assure, lors d'un transfert à travers un système relais, l'acheminement des données (**paquets**) à travers les différents nœuds d'un sous-réseau (routage). Les protocoles de niveau 3 fournissent les moyens d'assurer l'acheminement de l'appel, le routage, le contrôle de congestion, l'adaptation de la taille des blocs de données aux capacités du sous-réseau physique utilisé. Elle offre, en outre, un service de facturation de la prestation fournie par le sous-réseau de transport.
NIVEAU 4 **Couche Transport** ***Transport Layer***	La couche transport est la couche pivot du modèle OSI. Elle assure le contrôle du transfert de bout en bout des informations (**messages**) entre les deux systèmes d'extrémité. La couche transport est la dernière couche de contrôle des informations, **elle doit assurer aux couches supérieures un transfert fiable quelle que soit la qualité du sous-réseau de transport utilisé**.
NIVEAU 5 **Couche Session** ***Session Layer***	La couche session gère l'échange de données (**transaction**) entre les applications distantes. La fonction essentielle de la couche session est la synchronisation des échanges et la définition de points de reprise.
NIVEAU 6 **Couche Présentation** ***Presentation Layer***	Interface entre les couches qui assurent l'échange de données et celle qui les manipule, cette couche assure la mise en forme des données, les conversions de code nécessaires pour délivrer à la couche supérieure un message dans une syntaxe compréhensible par celle-ci. En outre, elle peut, éventuellement, réaliser des transformations spéciales, comme la compression de données.
NIVEAU 7 **Couche Application** ***Application Layer***	La couche application, la dernière du modèle de référence, fournit au programme utilisateur, l'application proprement dite, un ensemble de fonctions (entités d'application) permettant le déroulement correct des programmes communicants (transferts de fichiers, courrier électronique...).

FIGURE 1.6 – Bréve description des fonctionnalités des couche OSI [30].

1.8 Le Modéle TCP/IP

L'architecture TCP/IP a été développée, dans le milieu des années 1970, par la DARPA [10] pour les besoins d'interconnexion des systémes informatiques de l'armée DoD [11]. TCP [12]/IP [13], est un ensemble de protocoles permettant de résoudre les problémes d'interconnexion en milieu hétérogéne [30]

10. Defense advanced Research Project Agency-USA
11. Department of Defense
12. Transmission Control Protocol
13. Internet Protocol

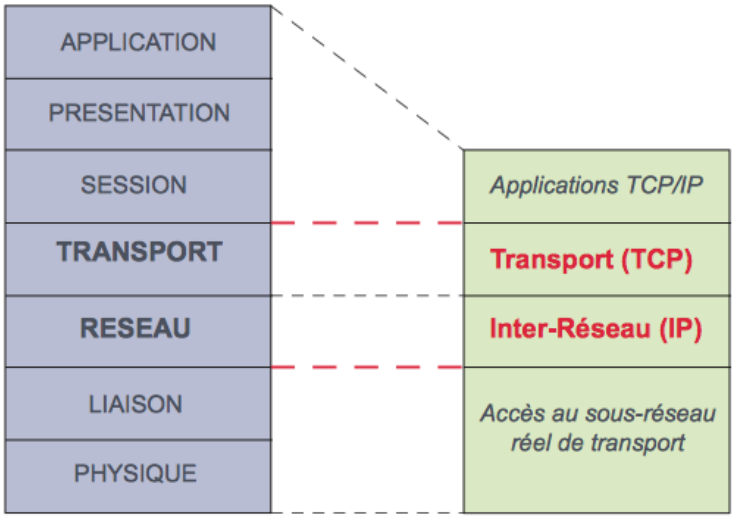


FIGURE 1.7 – Le modéle OSI et l'architecture TCP/IP [30].

1.9 La sécurité d'un réseau

La sécurité d'un réseau est un niveau de garantie que l'ensemble des machines du réseau fonctionnent de façon optimale et que les utilisateurs des machines possédent uniquement les droits qui leur ont été octroyés.

Il s'agit de garantir en tout temps que :

1. Empêcher les personnes non autorisées d'agir sur le systéme (le réseaux) de façon malveillante.
2. Empêcher les utilisateurs d'effectuer des opérations involontaires capables de nuire au systéme.
3. Securiser les données et prévoir un moyen de récupération en cas de panne.
4. Garantir la non-interruption d'un service.

1.9.1 Terminologie

Une attaque : n'importe quelle action qui compromet la sécurité des informations.[13]

Mécanismes de sécurité : un mécanisme qui est conçu pour détecter, prévenir et lutter contre une attaque de sécurité.[13]

Service de sécurité : un service qui augmente la sécurité des traitements et des échanges de données d'un système. Un service de sécurité utilise un ou plusieurs mécanismes de sécurité.[13]

1.9.2 L'humain et sa machine

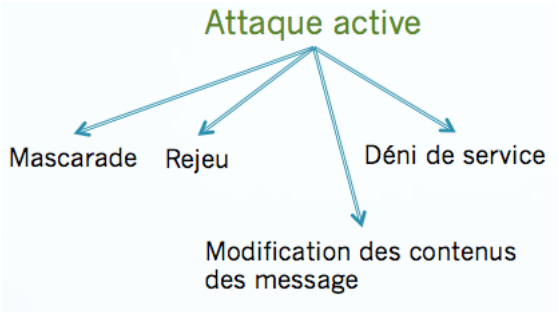
Environ 80% des problèmes de sécuritès ont pour origine les utilisateurs internes qui mettent le réseau en danger souvent par ignorance ou inconscience par leur comportement quelques exemples : installation intempestives de logiciels de sources douteuses (chevaux de troie, vers, virus, porte d'erobée), une mauvaise utilisation du lecteur de courrier (en ouvrant automatiquement les fichiers attachés par exemple), il y a aussi le fait qu'aujourd'hui les utilisateurs possède pour la plupart des mots de passe trop "basique".[28]

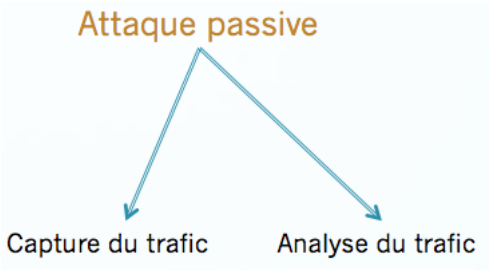
1.9.3 Le but des pirates informatique

Les motivations des pirates informatique que l'on appelle communément "Hakers" peuvent être multiples : l'attirance de l'interdit, le désir d'argent via le piratage des systéme bancaire, la réalisation de certaines attaque contre rémunairation, le besoin de renommée, l'envie de nuire c'est à dire : dtruire des données ou empêcher un systéme de fonctionner.

1.9.4 les types de menaces

On peut classer les menaces(attaques) en deux catégories selon qu'elles perturbent ou pas le réseau :





Les menaces passives consistent essentiellement à copier ou à écouter l'information sur le réseau, elles nuisent à la confidentialité des données. Dans ce cas, celui qui prélève une copie n'altère pas l'information elle-même. Il en résulte des difficultés à détecter ce type de malveillance, car elles ne modifient pas l'état du réseau. La méthode de prélèvement varie suivant le type de

réseau. Sur les réseaux câblés, on peut imaginer un branchement en paralléle grâce à des appareils de type analyseurs de protocole ou une induction [14]. Sur les faisceaux hertziens, des antennes captent les lobes secondaires des faisceaux dans les transmissions par satellites...

En second lieu les menaces actives qui nuisent à l'intágritá des données. Elles se traduisent par différents types d'attaques. On distingue le brouillage, le déguisement [15], l'interposition qui esr une création malveillante de messages en émission ou en réception.

1.9.5 Les attaques sur le réseau

Ecoute des connexions [16] **:** interception des messages et/ou mot de passe par des renifleurs, identification des services du réseaux, des machines qui communiquent, des comportements, etc.

Mystification [17] **:** prendre l'identité d'une autre machine, DNS, routeur, machine sécuriséee, etc.

Déni de service : Le DDoS [18] repose sur une parallélisation d'attaques DoS, simultanément menées par plusieurs systémes (machines rebond) contre un seul, cela empêcher donc le bon fonctionnement d'un systéme en empêchant un service/machine de fonctionner.

14. rayonnement électromagnétique
15. Modification des données au cours de leur transmission, modification de l'identité de l'émetteur ou du destinataire
16. Sniffing
17. Spoofing
18. Distributed Denial of Service

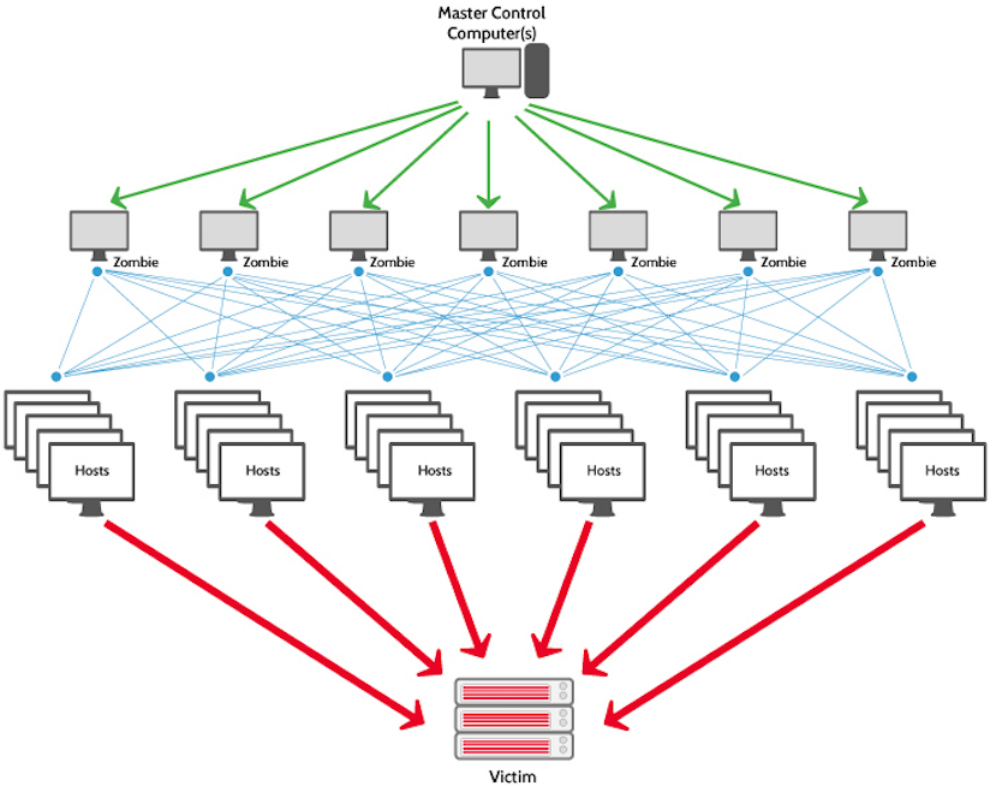


FIGURE 1.8 – Illustration d'une attaque DDos [21]

Le DDoS par saturation des ressources peux être réaliser de plusier façon :[28]

1. **Attaque ARP :** demandes incessantes ARP envoyées vers un routeur ou serveur d'un même réseau.
2. **Ping de la mort :** ICMP echo request de taille supérieure (65535 octets) au maximum spécifié.
3. **Attaque ping :** innondation de ICMP echo request vers un routeur ou serveur.
4. **Attaque Smurf :** innondation de ICMP echo request avec pour adresse IP source l'adresse de la victime et pour destination l'adresse de diffusion du réseau.
5. **Host Unreachable :** envoi de message ICMP "Host unreachable" à la victime (implique une déconnexion des session)
6. **Lan Attack :** envoi de paquet TCP d'IP et port source identiques á ceux de la victime.

7. **Teardrop Attack :** messages fragmentés construits pour que les fragments se chevauchent et soient impossibles á reconstituer (crash)
8. **SYN Flood :** inondation de demandes d'ouverture de session TCP.
9. **Evasive UDP Attack :** inondation de paquets UDP de longueur variable et d'adresse source IP aléatoire vers la victime.
10. **mail bombing :** avalanche d'e-mail sur le compte d'un utilisateur ou sur un serveur pour l'engorger.

Intrusion exploitant : Exploite les bugs des services (serveur de mail, routeurs ...) ou alors des portes déerobéees déjá mises en place par une intrusion précédente. Cela peux etre aussi une faille dans l'architecture du réseau, sur les systémes employés ou dans les éléments d'interconnexion.

1.9.6 Securisé un échange

Dés que plusieurs entités sont impliquées dans un échange de messages sécurisés, des régles doivent déterminer l'ensemble des opérations cryptographiques à réaliser, afin de sécuriser la communication.
Lorsque l'on parle de "sécuriser un échange", on souhaite prêter attention aux 4 services suivants : la confidentialité, l'intégrité, l'authentification et la non-répudiation.[14]

Confidentialité : principe selon lequel une information n'est accessible qu'á ceux ayant droit d'y accéder. Les systémes apportant la confidentialité rendent l'information inaccessible en l'absence de certains éléments. Leur perte signifie la perte de l'information.

Intégrité : concept selon lequel une information est reçu entiére, telle qu'elle a été transmise, non altéré. Le contrôle de l'intégrité est utilisé principalement lors des transmissions numériques, pour détecter les erreurs accidentelles. La plupart des contrôles d'intégrité sont simples à calculer (MD5 [19], CRC [20]...) et ne protégent donc pas contre une attaque.

19. Fonction de hachage (Message Digest 5)
20. Contrôle d'erreur (Cyclic Redundancy Check)

Authenticité : assure l'origine d'une information. En terme de sécurité, cette propriété n'a pas de sens sans l'intégrité (et vice-versa). Les deux ensembles (intégrité et authenticité) assure à la fois que le message provient bien de l'émetteur supposé, et qu'il n'a pas été altéré.

Non-répudiation : principe selon lequel une entité ne peut pas nier être l'origine d'une action. Dans le cadre de la signature numérique, cela signifie que l'auteur d'un message ne peut pas nier qu'il est l'auteur du message. De maniére plus général, cela s'applique également au destinataire d'une transmission, qui ne peut pas nier l'avoir reçue (par exemple, avec un systéme d'accusé de réception)

Les différentes type d'attaque sur chaque régle de sécurité :

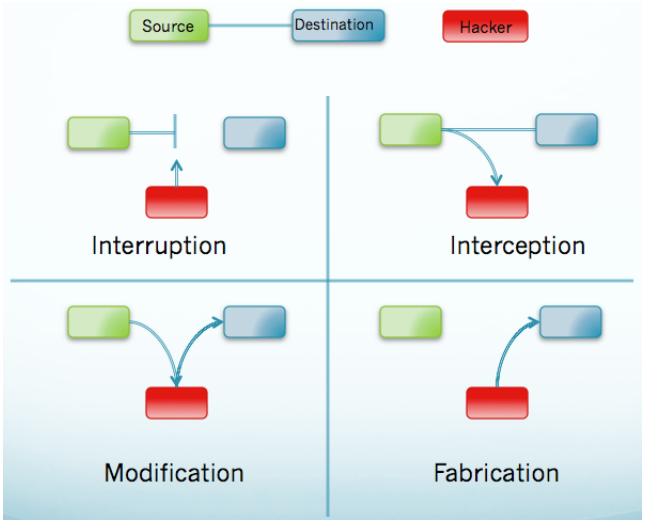


FIGURE 1.9 – Les diffrentes type d'attaque sur un réseaux

Interruption : vise la disponibilité des information.

Interception : vise la confidentialité des informations.

Modification : vise l'intégrité des informations.

Fabrication : vise l'authenticité des informations.

1.9.7 Mécanismes de sécurité

A cause des menaces provenant des logiciels malveillants, Il faut mettre en place des mécanismes pour s'assurer la confidentialité, l'intégrité et la disponibilité des services. Parmi ces mécanismes, on peut citer :[7]

Cryptage : Le chiffrement est un procédé de cryptographie grâce auquel on souhaite rendre la compréhension d'un document impossible à toute personne qui n'a pas la clé de (dé)chiffrement. Ce principe est généralement lié au principe d'accès conditionnel.Bien que le chiffrement puisse rendre secret le sens d'un document, d'autres techniques cryptographiques sont nécessaires pour communiquer de façon sûre [7].

Pare-Feu : C'est un ensemble de différents composants matériels (physique) et logiciels (logique) qui contrôlent le trafic intérieur/extérieur selon une politique de sécurité. Un système pare-feu fonctionne la plupart du temps grâce à des règles de filtrage indiquant les adresses IP autorisées a communiquer avec les machines aux réseaux, il s'agit ainsi d'une passerelle filtrante. Il permet d'une part de bloquer des attaques ou connexions suspectes d'accéder au réseau interne. D'un autre côté, un firewall sert dans de nombreux cas également à éviter la fuite non contrôlée d'informations vers l'extérieur. Il propose un véritable contrôle surle trafic réseau de l'entreprise, Il permet donc d'analyser, de sécuriser et de gérer le trafic réseau [7].

Anti-virus : Les anti-virus sont des logiciels conçus pour identifier, neutraliser et éliminer des logiciels malveillants. Ceux-ci peuvent se baser sur l'exploitation de failles de sécurité, mais il peut également s'agir de programmes modifiant ou supprimant des fichiers, que ce soit des documents de l'utilisateur de l'ordinateur infecté, ou des fichiers nécessaires au bon fonctionnement de l'ordinateur. Un anti-virus vérifie les fichiers et courriers électroniques, les secteurs de boot (pour détecter les virus de boot), mais aussi la mémoire vive de l'ordinateur, les médias amovibles(clefs USB, CD, DVD, etc.) et les données qui transitent sur les éventuels réseaux (dont internet)[7].

VPN : Le virtual private network permet la transmission d'informations de façon sécurisé, c'est-á-dire encapsulant les données á transmettre de

façon chiffrée.

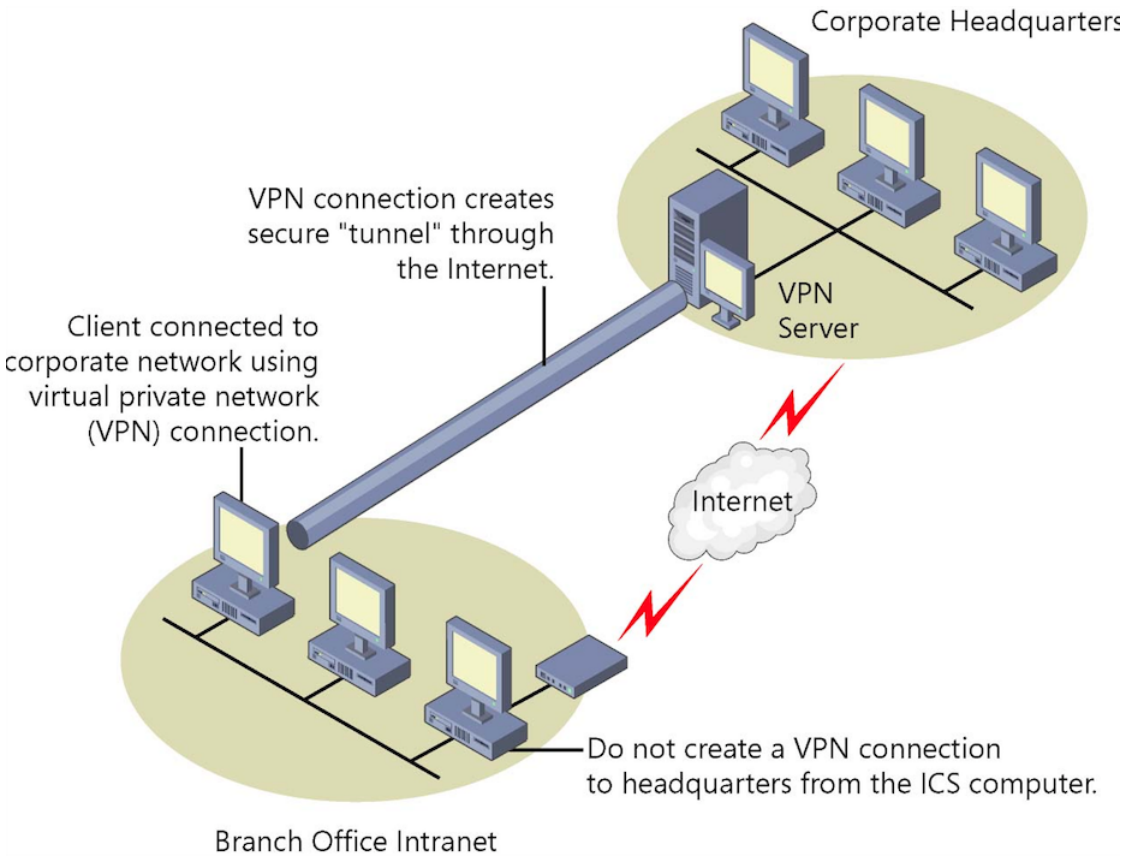


FIGURE 1.10 – Fonctionnement d'un tunnel VPN [1]

Un réseau privé virtuel repose sur un protocole, appelé protocole de tunnelisation (tunneling), c'est-á-dire un protocole permettant aux données passant d'une extrémité du VPN á l'autre d'être sécurisées par des algorithmes de cryptographie.Le terme de "tunnel" est utilisé pour symboliser le fait qu'entre l'entrée et la sortie du VPN les données sont chiffrées (cryptées) et donc incompréhensible pour toute personne située entre les deux extrémités du VPN, comme si les données passaient dans un tunnel.

IDS : Un système de détection d'intrusion(IDS [21]) est un mécanisme destiné à repérer des activités anormales ou suspectes sur la cible analysée (un

21. Intrusion Detection System

réseau ou un hôte). Il permet ainsi d'avoir une connaissance sur les tentatives réussies comme échouées des intrusions.Il faut distinguer deux aspects dans le fonctionnement d'un IDS : le mode de détection utilisé et la réponse apportée par l'IDS lors de la détection d'une intrusion.[7]

IPS : Un système de prévention d'intrusion (ou IPS, Intrusion Prevention System) est quant a lui un outil similaire aux IDS, sauf que ce système peut prendre des mesures afin de diminuer les risques d'impact d'une attaque. C'est un IDS actif, il posséde un balayage automatisé, exemple :l'IPS peut bloquer les ports automatiquement.[7]

1.9.8 Les IDS

Un IDS a quatre fonctions principales : l'analyse, la journalisation, la gestion et l'action.[7]

Analyse : analyse des journaux du système pour identifier des intentions dans la masse de données recueillie par l'IDS. Il y a deux méthodes d'analyse : L'une basée sur les signatures d'attaques, et l'autre sur la détection d'anomalies.

Journalisation : Enregistrement des évènements dans un fichier de log. par Exemples d'évènements : arrivée d'un paquet, tentative de connexion.

Gestion : Les IDS doivent être administrés de manière permanente. On peut assimiler un IDS à une caméra de sécurité.

Action : alerter l'administrateur quand une attaque dangereuse est détectée.

1.9.9 Les types d'IDS

1. **IDS Réseaux :**
   Le rôle essentiel d'un IDS réseau (NIDS [22]) est l'analyse et l'interprétation des paquets circulant sur ce réseau. L'implantation d'un NIDS sur un réseau se fait de la façon suivante :
   des capteurs sont placés aux endroits stratégiques du réseau et génèrent des alertes s'ils détectent une attaque. Ces alertes sont envoyées à une console sécurisée, qui les analyse et les traitent éventuellement. Cette

22. Network IDS

console est généralement située sur un réseau isolé, qui relie uniquement les capteurs et la console.[6]

2. **IDS Host :**
   Les HIDS [23] analysent le fonctionnement et l'état des machines sur lesquelles ils sont installés. L'intégrité des systèmes est alors vérifiée périodiquement et des alertes peuvent êtres déclanchées.[6]

3. **IDS Hybride :**
   Les IDS hybrides rassemblent les caractéristiques des NIDS et HIDS. Ils permettent, en un seul outil, de surveiller le réseau et les terminaux. Les sondes sont placées en des points stratégiques, et agissent comme NIDS et/ou HIDS suivant leurs emplacements. Toutes ces sondes remontent alors les alertes à une machine qui va centraliser le tout, et agréger(lie) les informations d'origines multiples. Ainsi, on comprend que les IDS hybrides sont basés sur une architecture distribuée, où chaque composant unifie son format d'envoi. Cela permet de communiquer et d'extraire des alertes plus pertinentes. Les avantages des IDS hybrides sont multiples :[6]

   (a) Moins de faux positif.

   (b) Meilleure corrélation (la corrélation permet de générer de nouvelles alertes à partir de celles existantes).

   (c) Possibilité de réaction sur les analyseurs.

1.10 Deep packet inspection

1.10.1 OWASP

Open Web application Security Project (OWASP) est une communauté en ligne travaillant sur la sécurité des applications Web. Sa philosophie est d'étre á la fois libre et ouverte à tous. Elle a pour vocation de publier des recommandations de sécurisation Web et de proposer aux internautes, administrateurs et entreprises des méthodes et outils de réfrénce permettant de contrôler le niveau de sécurisation de ses applications Web.[11]

Le but de ce projet est de fournir une liste des dix risques de sécurité applicatifs Web les plus critiques. Ce classement fait référence aujourd'hui dans le

23. Host IDS

domaine de la sécurité : il est cité par de nombreux organismes d'audits et de sécurisation des systémes d'information (DoD, PCI Security Standard).cela a pour objéctif de sensibiliser les développeurs sur les failles Web les plus importantes. La majorité des audits de sécurité informatique Web sont basées sur ce Top Ten.
Les dix risques du Top Ten par ordre de dangerosité :[11]

1. **The Injection :** correspond au risque d'injection SQL, SHELL ...
2. **Broken authentification and Session Management :** correspond au risque de casser la gestion de l'authentification et de la session. Comprend notamment le vol de session ou la récupération de mots de passe.
3. **Cross-Site Scripting :** correspond au XSS soit l'injection de contenu dans une page, ce qui provoquent des actions non désirées sur une page Web. Les failles XSS sont particuliérement répandues parmi les failles de sécurités Web.
4. **Insecure Direct Object References :** correspond aux failles de sécurité des ID de données visualisées. Nécessite de mettre en place un contrôle d'accés aux données.
5. **Security Misconfiguration :** correspond aux failles de configuration liés aux serveurs Web, applications, base de données ou framework.
6. **Sensitive Data Exposure :** correspond aux failles de sécurité liées aux données sensibles comme les mots de passe, les numéros de carte de crédit ou encore les données personnelles et la nécessité de crypter ces données.
7. **Missing Function Level access Control :** correspond aux failles de sécurité liés aux accés de fonctionnalité.
8. **Cross-Site Request Forgery (CSRF) :** correspond aux failles liées á l'exécution de requêtes á l'insu de l'utilisateur.
9. **Using Components with Known Vulnerabilities :** correspond aux failles liées à l'utilisation de composants tiers.
10. **Unvalidated Redirects and Forwards :** correspond aux faillies liées aux redirect et forward générique des applications.

Conclusion

Au cours de ce chapitre nous avons eu pour objectif de cerné le fonctionnement et l'architectures des réseaux. Cela nous a permi d'approfondir nos recherches dans ce domaine et de ce fait de nous intéressé à la sécurité réseaux. Aprés avoir parcouru les différentes attaques possible de se produire sur un réseau, nous avons proposer, les différents mécanismes existant pour une assuré une meilleur protection contre ces menaces. la cryptographie en est une, c'est dans ce domaine là que nous avons choisie d'approfondir nos recherches. C'est justement le sujet traité dans le prochain chapitre.

Chapitre II

2 État de l'art

Introduction

A travers ce chapitre, nous avons tenté de nous familiariser avec les principes de base de la cryptographie(Symétrique, Asymétrique, Signature ect...) Nous avons étudié les algorithmes cryptographiques les plus utilisés et les prérequis nécessaire à la compréhension de notre étude ont été abordés.

2.1 La cryptographie

2.1.1 Le vocabulaire de base

Cryptologie : Il s'agit d'une science mathématique comportant deux branches : la cryptographie et la cryptanalyse

Cryptographie : La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de maniére confidentielle sur un support donné.[14]

Chiffrement : Le 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.[14]

Texte chiffré : appelé également cryptogramme, le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair.[14]

Clé : Il s'agit du paramétre 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. Dans le cas d'algorithmes asymétriques, elle différe pour les deux opérations.[14]

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.[14]

Cryptosystéme : Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné.[14]

Remarque. *On parle de "décryptage" pour désigner l'action permettant de retrouver le texte clair sans connaître la clef de déchiffrement et on parle de "déchiffrement" pour désigner l'action permettant de retrouver le text claire avec la clé .*

2.1.2 La publication des algorithmes

Selon l'endroit où réside le secret, on peut parler d'algorithme secret ou d'algorithme publié. Chacun posséde ses atouts et inconvénients.

Algorithme secret :[14]

A) La cryptanalyse, souvent basée sur le secret de la clé, doit ici en plus retrouver l'entiéreté de l'algorithme (mécanisme de récupération).

B) Souvent, de tels algorithmes 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 á le casser.

C) De tels algorithmes sont rarement distribués par delà les frontiéres, afin de garder un nombre d'utilisateurs restreint.

Algorithme publié :[14]

A) Puisque l'algorithme est publié, tout le monde a le droit de l'explorer. Ainsi, les failles (laissées intentionnellement ou non par les concepteurs) peuvent être plus facilement découvertes. La sécurité en est donc améliorée.

B) Comme la publication est autorisée, il n'est pas nécessaire de chercher à protéger le code contre le reverse-engineering.

C) Cette publication permet d'étendre les travaux sur l'algorithme au niveau mondial. Toute une série d'implémentations logicielles peuvent donc être réalisées.

D) Tout le monde utilise la même version publique ce qui permet une standardisation générale.

En conséquence, on préférera les algorithmes publiés, souvent plus sûrs.

2.1.3 Principe de Kerckhoff

Definition. *La sécurité du chiffré ne doit pas dépendre de ce qui ne peut pas être facilement changé.[Principe de Kerckhoff]*

En d'autres termes, aucun secret ne doit résider dans l'algorithme mais plutôt dans la clé. Sans celle-ci, il doit être impossible de retrouver le texte clair à partir du texte chiffré. Par contre, si on connaît K, le déchiffrement est immédiat.

2.1.4 Les algorithme cryptographiques

Pour chiffrer les données, il existe deux types de méthodes de chiffrement/déchiffrement :

1. La cryptographie symétrique.
2. La cryptographie asymétrique.

Dans le chiffrement symétrique (aussi appelé cryptographie à clef secréte), la même clef est utilisée pour le chiffrement et pour le déchiffrement. Parmi les standards les plus célébres de ce type de chiffrement, on trouve le DES [24][1.B] (et ses variantes comme le triple-DES) et l'AES [25][2.B]. En chiffrement asymétrique (appelé aussi cryptographie à clef publique), une clef publique est utilisée pour le chiffrement et une autre clef dite clef privée est utilisée pour le déchiffrement. Le RSA [3.B] est l'un des algorithmes les plus utilisés car son implantation matérielle est l'une des moins coûteuses pour ce type de chiffrement,on ajoutera donc le chiffrement á base de courbe elliptique la la liste des algorithme a clé public. .

2.1.5 La cryptographie à clé privée

Definition. *Un algorithme de chiffrement à clé privé (Symétrique) transforme un message clair P avec une clé secréte K. Le rśultat est un message chiffré C.*
Pour le déchiffrement la fonction de chiffrement doit être inversible, le chiffré C et donc déchifré grace a clé K.
Il existe deux catégorie de chiffrement à clé publique, le chiffrement par bloc et le chiffrement pas flot.[25]

24. Data Encryption Standard
25. advanced Encryption Standard

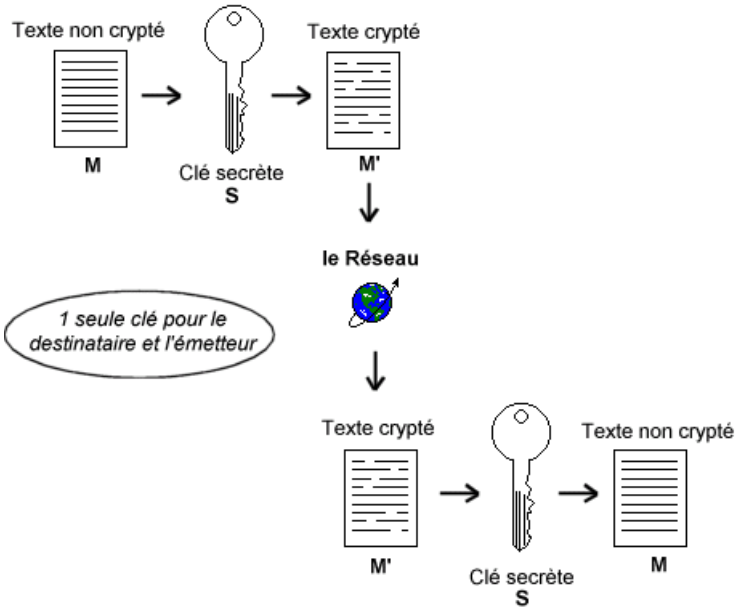


FIGURE 2.1 – Cryptographie à clé privée [25]

-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 [26], mais l'AES peut allerjusque 256.

-L'avantage principal de ce mode de chiffrement est sa rapidité d'execution, car ils peuvent être implement/'e avec des circuit directement.

-Le principal désavantage réside dans la distribution des clés : pour une meilleure sécurité, on pratiquera à l'échange de maniére manuelle. Malheureusement, pour de grands systémes, le nombre de clés peut devenir conséquent. C'est pourquoi on utilisera souvent des échanges sécurisés pour transmettre les clés. En effet, pour un systéme à N utilisateurs, il y aura $N * (N-1)/2$ paires de clés.[14]

2.1.6 La cryptographie à clé publique

Definition. *Un algorithme de chiffrement à clé publique (asymétrique) nécessite la génŕation de deux clé, une privé et une publique, le chiffrement se fera avec la clé publique.*
Pour le déchiffrement seul les détenteur de la clé privé peuvent déchiffrer le message P.[25]

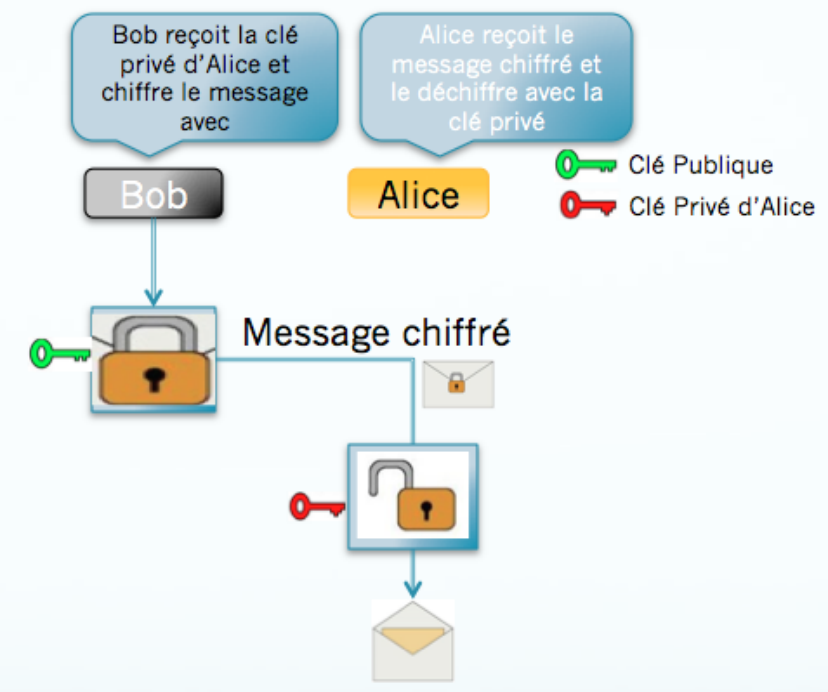


FIGURE 2.2 – La cryptographie á clé publique

- Le principe de ce genre d'algorithme est qu'il s'agit d'une fonction unidirectionnelle à trappe. Une telle fonction à la particularitd'être facile à calculer dans un sens, mais difficile voire impossible.Dans le sens inverse. La seule maniére de pouvoir réaliser le calcul inverse est de connaître une trappe. Une trappe peut par exemple être un faille dans le générateur de clés. Cette faille peut être soit accidentelle ou intentionnelle de la part du concepteur.[14]

- Les algorithmes se basent sur des concepts mathématiques tels que l'exponentiation de grands nombres premiers (RSA), le probléme des logarithmes discrets (ElGamal, EC), ou encore le probléme du sac à dos.[14]

-Au niveau des performances, le chiffrement par voie asymétrique est environ 1000 fois plus lent que le chiffrement symétrique.[14]

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

-La distribution des clés est grandement facilitée car l'échange de clés secrétes n'est plus nécessaire. Chaque utilisateur conserve sa clé secréte sans

jamais la divulguer. Seule la clé publique devra être distribuée.

2.1.7 Fonction de hachage

Il s'agit de la troisiéme 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 á celle de départ. Le message réduit portera le nom de "Haché". L'intérêt est d'utiliser ce Haché comme empreinte digitale du message original afin que ce dernier soit identifié de maniére unique.[14]

Deux caractéristiques importantes :

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 calculatoirement difficile de trouver un M' tel que $M' \neq M$ et $H(M') = H(M)$.

2.1.8 La signature

Les signatures numériques fonctionnent en majorité grâce à la cryptographie asymétrique et aux fonctions de hachage. Il existe également des algorithmes de signatures basé sur de la cryptographie symétrique, mais ceux ci sont plus compliqué à mettre en oeuvre.

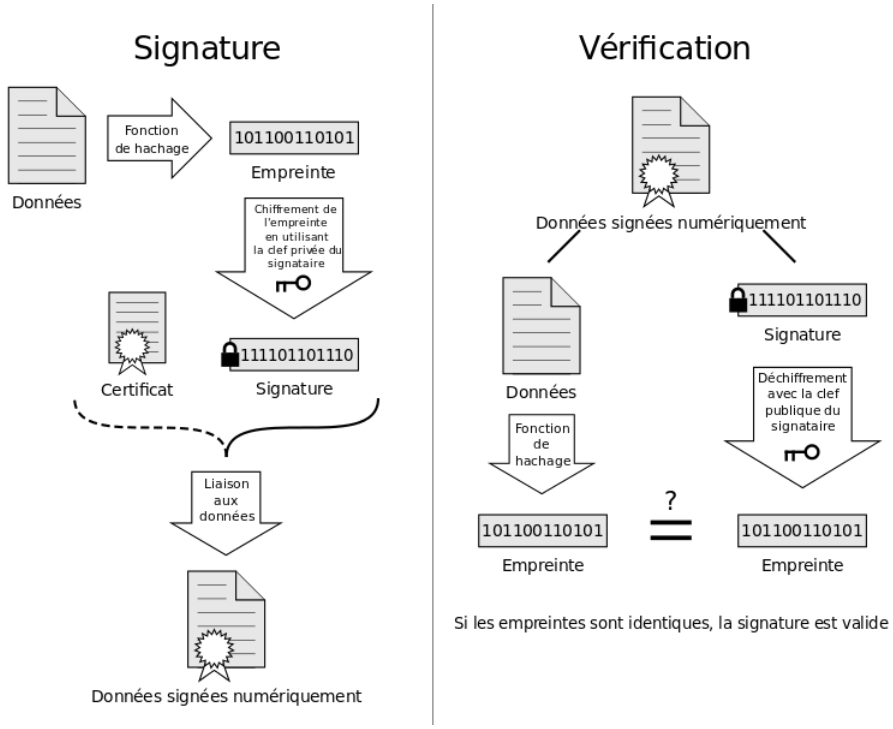


FIGURE 2.3 – Signature et vérification

Une signature numérique doit dépendre du message et de l'émetteur. Par ailleurs, l'algorithme utilisé doit être simple à utiliser (simple à produire en connaissance des clés, difficile sinon). De plus, la vérification d'une signature numérique doit être faisable par n'importe qui.

On constate des similitudes dans l'évolution des fonctions de hachage et des chiffrements symétriques. A cause de constante amélioration des attaques par force brute, les algorithmes doivent eux aussi évoluent constamment. On est passé du DES à l'AES dans les chiffrements symétriques et de MD5 à SHA et à dans les algorithmes de hachage. Il est aujourd ?hui conseillé d ?utiliser le SHA-2 [27], suite aux récentes attaques trouvée contre SHA-1.[14]

2.2 La cryptanalyse

Il s'agit de l'étude des mécanismes théoriques ou techniques visant à briser (casser) un algorithme de chiffrement, c'est-á-dire le fait de retrouver le message en claire M à partir de du chifré C, sans connaître la clé K . Dans certains cas, il s'agira également de retrouver cette clé K. On parlera

27. SAH-256

donc d'attaque cryptanalytique. Il en existe 4 grands types, chacun pouvant utiliser différentes techniques.

2.2.1 La théorie de la complexité des algorithmes cryptorgraphique

Cette théorie fournit une méthodologie pour analyser la complexité de calcul de différents algorithmes et techniques cryptographiques. Elle compare les algorithmes et les techniques cryptographiques pour déterminer leur niveau de sécurité.
Jusqu'ici, la théorie de l'information nous apprenait que tous les algorithmes peuvent être cassés. Avec la théorie de la complexité, on apprend s'ils peuvent être cassés avant la fin du monde.[14]

Classe	Complexité	Nombre d'ops pour $n=10^6$	Temps pour 10^6 ops/sec
Linéaires	$O(n)$	10^6	1 seconde
Quadratiques	$O(n^2)$	10^{12}	11,6 jours
Cubiques	$O(n^3)$	10^{18}	32.000 ans
Exponentiels	$O(2^n)$	$10^{301\,030}$	$10^{301\,006}$ fois l'age de l'univers

FIGURE 2.4 – Temps pour "Casser" un algorithme selon sa complexité[14]

La complexité des problémes détermine les temps et espaces minimaux nécessaires pour résoudre l'instance la plus difficile du problème sur un ordinateur [28]. Les problèmes qui peuvent être résolus avec des algorithmes polynomiaux en temps sont appelés solubles, car ils peuvent généralement être résolus en un temps raisonnable. Les problémes qui ne peuvent être résolus en temps polynomial sont appelés non solubles car calculer leur solution devient vite impossible. Il s'agit souvent des problèmes résolus par des algorithmes exponentiels.[14]

Nous citerons deux ensembles de problémes :[14]

La classe P , qui contient les problémes résolus en temps polynomial sur une MT [29].

28. Théoriquement Appelé : Machine de Turing
29. Machine de turing

La classe NP , qui contient les problémes résolus en temps polynomial sur une MTND [30].

2.2.2 Les 4 attaques cryptanalytiques

1. Attaque sur le texte chiffré uniquement (ciphertext-only)
   a partir d'un texte chiffré, on recherche le texte clair et/ou la clé. On procéde par analyse de fréquence des lettres utilisées dans le texte chiffré. Cette technique ne fonctionne que pour la plupart des chiffrements classiques basiques, les seuls permettant l'utilisation de l'analyse de fréquence. On peut aussi également procéder par force brute pour briser de tels chiffrements.

2. Attaque à texte clair connu (known-plaintext attack)
   Etant donné un texte chiffré et un fragment de texte clair associé, on recherche le texte clair restant et/ou la clé. On utilise la technique dite de la cryptanalyse linéaire présentée dans la section suivante.

3. Attaque sur un texte clair sélectionné(chosen-plaintext attack)
   Etant donné la capacité de chiffrer un fragment de texte clair choisi arbitrairement, on recherche la clé par la technique de la cryptanalyse différentielle.

4. Attaque sur le texte chiffré uniquement (chosen-ciphertext attack)
   Etant donné la capacité de déchiffrer un fragment de texte chiffré choisi arbitrairement, on recherche la clé.

2.2.3 Quelques autres techniques

1. **La force brute** :
   La force brute ou recherche exhaustive a pour but de tester toutes les clés possibles de maniére exhaustive. La limite maximale est donnée par T^N avec T la taille de l'alphabet utilisé, N est la taille de la clé.
   Par exemple, pour une clé de 128 bits, il y a 2^{128} clés possibles. Cette technique n'est efficace que pour des textes chiffrés avec une clé relativement courte.

30. Une Machine de Turing Non Déterministe est une variante de la machine de Turing normale qui devine les solutions soit par chance, soit en essayant toutes les possibilités en parallèle.

Remarque. *Un algorithme est dit cassé quand il est possible de retrouver la clé en effectuant moins d'opérations qu'en utilisant la force brute.*

2. **Attaque par dictionnaire** :
   Lorsque la clé est un mot (p.ex. un mot de passe), on peut tenter de court-circuiter la Force Brute. Le principe est ici d'utiliser un recueil de mots possibles (le dictionnaire), et de tester tous les mots de ce dictionnaire. attention à bien distinguer les deux attaques : on teste tous les mots du dictionnaire mais celui-ci ne contient pas toutes les possibilités, ce qui nous raméne a une recherche plus courte qu'une recherche exhaustivemais qui ne couvre pas tout les cas possibles.

3. **Cryptanalyse différentielle** :

   Il s'agit de l'étude (modélisation) des transformations subies par le message durant son passage dans l'algorithme de chiffrement. Le principe est de modéliser ce qu'une modification en entrée induira sur le résultat de l'algorithme.[15]

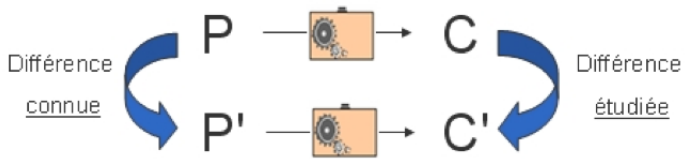


FIGURE 2.5 – Fonctionnement de la cryptanalyse différentielle

4. **Cryptanalyse linéaire** :

   Le but est d'effectuer une approximation linéaire de l'algorithme de chiffrement. Il n'y a ici aucune possibilité de choisir le texte clair à chiffrer, on dispose tout au plus d'un ensemble de couples (M, C).
   Cryptanalyses différentielle et linéaire sont des outils parfaits pour comparer la résistance de divers chiffrements. aucun algorithme cryptographique n'est dit valable s'il ne résiste pas à ce type de cryptanalyse. La résistance à ces attaques ne signifie pas résistance contre toutes les autres méthodes inconnues. Sans une connaissance approfondie des

techniques de cryptanalyse, il n'est pas possible de créer un algorithme de chiffrement performant, sûr et robuste.[15]

5. **Man-In-The-Middle** :

   Son déroulement est illustré par la figure ci-dessous : le pirate se fait passer pour B auprés de A et pour A aupérs de B dans le but de récuperer les informations echangées.

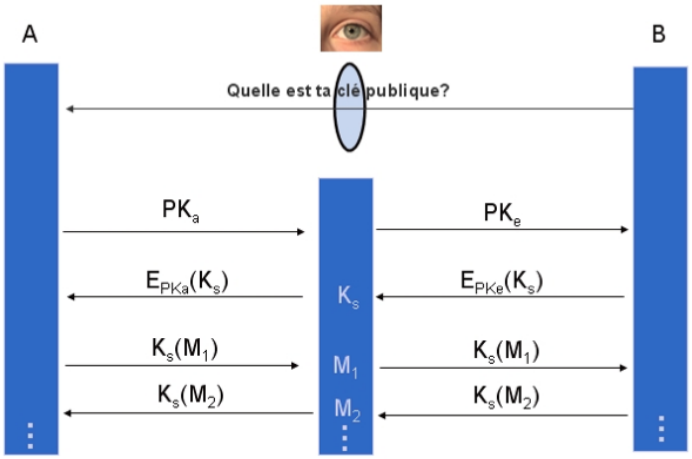


FIGURE 2.6 – Fonctionnement du Man-In-The-Middle

2.2.4 Les attaques physiques

On appelle attaque physique, toute attaque menée non contre la structure mathématique d'un algorithme mais plutôt contre un ou plusieurs composants, en général électroniques, du dispositif physique mettant en oeuvre l'algorithme. Elle vise à exploiter des défauts du systéme mise en oeuvre. Ce genre d'attaque s'est développé de plus en plus pour devenir aujourd'hui l'une des composantes essentielles de la quête de renseignements.[15]

En effet, une sécurité "mathématique" ne garantit pas forcément une sécurité physique lors de l'utilisation du chiffrement. Ces attaques communément appelées attaques par canaux cachés, peuvent être invasives, par la mise à nu du circuit physiquement, ou non-invasives, par la mesure de paramétres physiques extérieurs au matériel pendant son activité, sans toucher à son intégrité physique. Ces attaques se basent généralement sur une connaissance

approfondie de l'architecture du matériel et portent sur différents types de données :[15]

Attaque temporelle : elle est basée sur une comparaison du temps mis pour effectuer certaines opérations critique de l'algorithme.

Attaque par faute : elle consiste en l'injection volontaire d'erreurs dans le systéme pour provoquer certains comportements révélateurs sur les données de l'algorithme.

Analyse de rayonnement électromagnétique : Elle est basée sur le fait qu'une particule chargée de grande énergie émet un rayonnement électromagnétique. Ce rayonnement est analysé pour déduire l'information secréte.

Analyse de consommation : Elle consiste à analyser la consommation en courant pendants l'exécution de l'algorithme, un pique de consommation indiquera par exemple une utilisation d'une donnée critique.

Conclusion

Dans ce chapitre nous avons traité le sujet de la cryptographie. Nous avons expliquer qu'il y avait deux concepts qui se livraient une bataille sans relâche.
D'un côté nous avons donc la cryptographie qui tend à rendre le chiffrement des données de plus en plus sûr et d'un autre côter la cryptanalyse qui "casse" les systémes de chiffrement mis au point par la cryptographie.
C'est donc un mal pour un bien car le fait de trouver des failles dans des systémes de chiffrement nous poussent a en trouvé de meilleurs.
La cryptographie ļé privé va être utiliser dans ce travail. La question qui mérite d'être soulvée est de savoir quel programme cryptographique doit-on utiliser pour notre chiffrement ?.
C'est justement à cette question que nous tenterons de répondre dans le prochain chapitre.

Chapitre III

3 Etude théorique

Introduction

Tout au long de ce chapitre notre étude avait pour objectif, dans un premier temps, de présenter les systémes de vote existant. Puis dans un second temps, expliquer les différents problèmes rencontrés dans le vote électronique ainsi que la partie théorique de ses solutions. Enfin Nous avons abordé tous les concepts théoriques nécessaires à la mise en place d'un système de cryptage à courbe elliptique.

3.1 Les systémes de vote en ligne

3.1.1 Helios

Helios est une plate-forme open-source pour créer des élections sur internet et qui posséde les propriétés de : confidentialité, vérifiabilité et incoercibilité. Il est basé sur la thèse de Ben adida et il est implementé par Ben adida, Olivier de Marneffe, Olivier Pereira avec les conseils de Jim adler, Lawrence Lessig, Josh Benaloh, andy Neff et Dan Wallach. Il compte aujourd'hui plus de 100.000 votes organisé et il a vu sa plate-forme utilisé par plusieurs universités notamment par le Catholique de Louvain pour ses élections présidentielles.[5]

Parmi les points forts de Helios :[5]

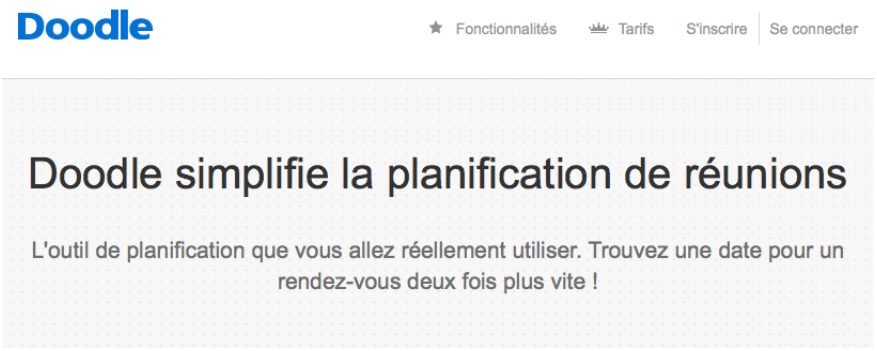
1. La possibilité pour un électeur de suivre son vote et de s'assurer de sa bonne prise en compte par le serveur. Ceci signifie que personne ne peut modifier les votes, mêmes pas les administrateurs.

2. l'addition homomorphe (les votes ne sont pas d'echiffrés un par un mais le seul déchiffrement effectué est à l'issue du vote)

3. l'utilisation de plusieurs clés pour effectuer le déchiffrement. Ceci permet de s'assurer qu'un administrateur ne peut pas déchiffrer le résultat ou un vote tout seul.

3.1.2 Doodel

Doodel est une application WEB qui permet d'effectuer des sondages en ligne, elle permet notamment : la création, modification et suppréssion d'un sondage en ligne.



Ce service WEB permet notamment de faire des planifications en ligne, en soumettant des suggestions puis en demander à un certain public choisie, de voter pour le meilleur choix. cela permet entre autre de fixer des rendez-vous entre plusieurs personnes trés facilement, ce qui dordinaire et une tâche compliquer au vue des différents planning de chacun.

3.2 Pourquoi voter en ligne est-il dur ?

Le vote en ligne est très particulier du fait qu'il demande un grand nombre de contraintes qui doivent être mises en place pour réaliser un système de chiffrement fiable et fonctionnel.
Tout d'abord un électeur doit avoir la possibilité de vérifier que son vote est bien correct.
D'un autre côté le votant doit être sûr que son vote est bien confidentiel et que lui seul connaît son contenu, ce qui introduit ici la propriété de confidentialité.
De plus le système qui doit être mis en place doit pouvoir vérifier la validité

des votes et aussi l'existence et l'authenticité des votants.
Le système doit pouvoir prouver à chaque instant que les opérations et traitement effectués sont valides, et cela sans pour autant divulguer d'informations sur les données traitées.

3.3 Pourquoi la cryptographie à courbe elliptique ?

Il y a de nombreux avantages à utiliser cette dernière. Tout d'abord, la longueur des clés utilisées, en effet une clé de 256 bits équivaut à une clé 4096 bits RSA. La sécurité de la clé repose sur la difficulté à résoudre le problème du logarithme discret [4.8], sachant que parmi les meilleurs algorithmes de nos jours qui résolvent ce problème sur les courbes elliptiques sont : le Baby Step, le Giant Step et le Rho de Pollard qui ont une complexité de $O(\sqrt{n})$ et $O(n \log n)$ avec n l'ordre du groupe, ce qui nous assure une bonne sécurité de chiffrement.

Un autre élément à prendre en compte est que la cryptographie à courbe elliptique possède la propriété d'homomorphisme par l'addition, ce qui facilite les traitements sur les données chiffrées.
Cette propriété entraîne de bien meilleures performances que les autres algorithmes.

3.4 La cryptographie à courbe elliptique

Definition. *Une courbe elliptique est un couple (E, φ) où E désigne une courbe projective lisse de genre 1, et φ un point de E, appelé point de base ou origine.*

Plusieurs formes de courbes elliptiques ont été étudiées ces dernières années. Parmi les plus connues : [20]

La forme de Weierstrass : $y^2 = x^3 + ax + b$
La forme de Montgomery : $y^2 = x^3 + ax^2 + x$
La forme de Edwards : $x^2 + y^2 = 1 + dx^2y^2$

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

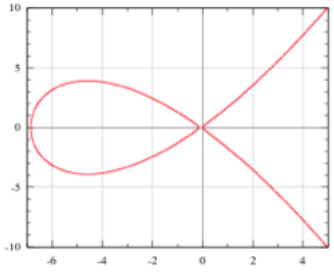


FIGURE 3.1 – Une courbe de Montgomery[20]

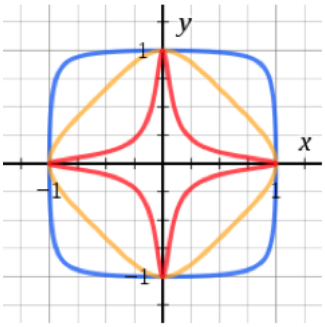


FIGURE 3.2 – Trois courbes sous la forme d'Edwards[20]

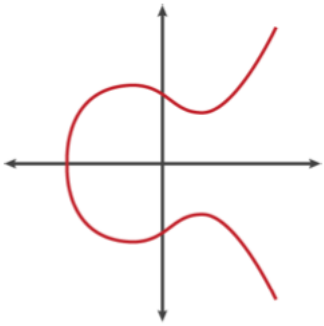


FIGURE 3.3 – Courbe de Weierstrass[20]

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

avec $a_i \in K$. *Une courbe donnée par une telle équation est dite lisse si le système suivant n'admet pas de solution :*

$$a_1y = 3x^2 + 2a_2x + a_42y + a_1x + a_3 = 0$$

autrement dit si les dérivées partielles en x *et* y *de* $f(x, y) = y^2 + a_1xy + a_3y - x^3 - a_2x^2 - a_4x - a_6$ *ne s'annulent pas.*

Definition. *Une courbe elliptique* E *définie sur* K *est une courbe lisse donnée par une équation de Weierstrass définie sur* K *à laquelle on a rajouté un point à l'infini, noté* φ.

$$E = \{y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6\} \cup \{\varphi\}$$

Si la caractéristique de K (char(K)) n'est ni 2 ni 3, alors en faisant les deux changements de variables successifs : $y \to \frac{1}{2}(y - a_1x - a_3)$ et ensuite $(x, y) \to ((\frac{x-3b_2)}{36}, \frac{y}{216})$ dans E, où $b_2 = a_1^2 + 4a_2$, nous obtenons :

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

avec $b_4 = 2a_4 + a_1a_3, b_6 = a_{23} + 4a_6, c_4 = b_2 - 24b_4, c_6 = -b_{32} + 36b_2b_4 - 216b_6$. ainsi si char(K)= 2, 3, nous pouvons toujours travailler avec des courbes elliptiques de la forme :

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

Dans ce cas la courbe est lisse si :

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

Propriété. *Soient E une courbe elliptique définie sur un corps K, et deux points $P, Q \in E(K)$, L la droite reliant P à Q (la tangente à E si $P = Q$) et R le troisième point d'intersection de L avec E.*

Soit L' la droite verticale passant par R. On définit $P+Q \in E(K)$ comme étant le deuxième point d'intersection de L' avec E. Muni de cette loi de composition $(E(K),+)$ est un groupe abélien dont l'élément neutre est le point à l'infini (φ).[20]

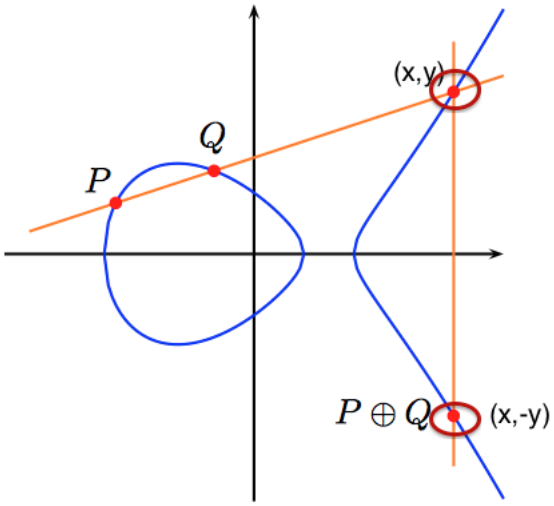


FIGURE 3.4 – Exemple graphique sur une courbe [20]

3.5 Nombre de points dans une courbe elliptique dans un corps fini

Nous allons prendre un exemple pour illustrer le nombre de points possibles dans une courbe elliptique .[4]
Exemple : E est une courbe elliptique telle que $E : y^2 = x^3 + 2$ sur le corps finie $\mathbb{F}_7$
Nous aurons donc :

$$E(\mathbb{F}_7)=\{\varphi,(0,3),(0,4),(3,1),(3,6),(5,1),(6,1),(6,6)\}$$

Les points de la courbe

Une question nous vient tout de suite à l'esprit : sommes-nous capable de calculer la cardinalité de la courbe sans compter tous les points ?

3.5.1 Le théorème de Hasse-Weil

Theorem 1. *Soit N est le nombre de points d'une courbe elliptique E sur un corps fini $\mathbb{F}_q$ avec q la cardinalité du groupe :[4]*

$$|N-(q+1)| \leqq 2\sqrt{q}$$

au final nous avons bien là une relation entre le nombre de points et la cardinalité du corps fini sur laquelle est construite la courbe.

3.6 Choix du corps de définition

Il faut avant tout choisir le corps dans lequel on va définir la courbe elliptique.Pour éviter certaines attaques, il est préférable de choisir :[18]

1. Soit un corps premier $\mathbb{F}_p$ oú p est un grand nombre premier.
2. Soit un corps $\mathbb{F}_{p^r}$ de caractéristique p petite (par exemple p = 2) oú r est un nombre premier tel que l'ordre de 2 dans $\mathbb{F}_p^*$ est grand. En particulier, il faut éviter les nombres premiers de Fermat[1.A] et de Mersenne[1.B].

3.7 Le chiffrement avec des courbes elliptiques

En cryptographie on s'intéresse surtout aux courbes elliptiques sur des corps finis. En particulier, il est crucial de savoir calculer $E(\mathbb{F}_q)$: une courbe elliptique E définie sur le corps $\mathbb{F}_q$.[10]

Prenons deux points A et B sur cette courbe, la courbe passant par a et B recoupe la courbe en un troisième point de coordonnées (x, y). Son symétrique $(x, -y)$ est lui aussi sur la courbe et on le désigne par $A + B$ pour signifier qu'il a été construit comme la somme de A et B. Cette opération + possède toutes les propriétés de l'addition des nombres. C'est-à-dire que l'on peut faire tous les calculs de type addition, soustraction et division euclidienne que nous faisons sur la droite des nombres réels sur cet objet que constitue une courbe elliptique.[10]

Il est possible, comme l'ont démontré Miller [27] et Koblitz [24], de coder avec cette opération au lieu de travailler avec l'addition usuelle. Il en résulte une plus grande complexité des calculs, un chiffrement par méthode des courbes elliptiques avec une clef de 192 bits possédant le même niveau de sécurité qu'une clé de 1024 bits pour la méthode RSa.

3.7.1 Multiplication Scalaire

On remplace la multiplication par une série d'additions. Prenons un exemple.
Soit la courbe $y^2 \mod 11 = (x_3 + x + 2) \mod 11$.
Calculons $d * P$, avec $d = 3$ et $P = (4, 2)$. On peut vérifier que le point P appartient bien à la courbe elliptique.
On peut remplacer $3 * P$ par $P + P + P$. Calculons d'abord $P + P$.
D'après la règle 5, $P + P = (4, 2) + (4, 2) = (8, 4) = 2 * P$.
D'après la règle 4, $2 * P + P = (8, 4) + (4, 2) = (2, 10) = 3 * P$.

3.7.2 Echange de clés

Alice et Bob se mettent d'accord (publiquement) sur une courbe elliptique $E(a, b, p)$ avec a,b deux éléments du corps fini et p le nombre d'éléments du corps c'est-à-dire qu'ils choisissent une courbe elliptique $y^2 \mod p = (x^3 + ax^2 + b) \mod p$. Ils se mettent aussi d'accord, publiquement, sur un point P situé sur la courbe.

Secrètement, Alice choisit un entier d_a, et Bob un entier d_B. alice envoie à Bob le point $d_a * P$, et Bob envoie à alice $d_B * P$. Chacun de leur côté, sont capables de calculer $d_a * (d_B * P) = d_B(d_a * P) = (d_a * d_B) * P$, qui est un point de la courbe, et constitue leur clef secrète commune.

3.8 Propriétés de la cryptographie à courbe elliptique

3.8.1 Sécurité

Supposant que Eve a espionné l'échange précédent, elle connait $E(a, b, p)$, P, $d_a * P$ et $d_B P$. Pour pouvoir calculer $d_a * d_B * P$, il faut pouvoir calculer d_a connaissant P et $d_a * P$. C'est ce que l'on appelle résoudre le logarithme discret sur une courbe elliptique [4.8].

3.8.2 Probléme du logarithme discret

Soit k un corps et E une courbe elliptique définie sur k. Les points k-rationnels formant un groupe abélien, celui-ci donne un cadre pour le probléme du logarithme discret.

Definition. *Soit E une courbe elliptique déefinie sur k et $G \in E(k)$. Connaissant le point $P \in E(k)$, le probléme du logarithme discret consiste à trouver $n \in N$, s'il existe, tel que $P = nG$.*

Si k est un corps fini, ce probléme est réputé être un probléme "difficile" en principe. á ce jour aucun algorithme sous-exponentiel général n'est connue pour le résoudre.

Cependant, il existe des attaques sous-exponentielles pour certaines courbes dite "faibles" du point de vue de la sécurité (par exemple les courbes supersinguliéres). Il faut donc faire attention au choix de E. Réciproquement, connaissant E, k, G et n, il est facile de calculer $P = nG$ en utilisant un algorithme déxponentation rapide.

3.8.3 Inconvénients

La théorie des fonctions elliptiques est complexe, et encore relativement récente. Il n'est pas exclu que des trappes permettent de contourner le problème du logarithme discret. La technologie de la cryptographie par courbe elliptique à fait l'objet du dépôt de nombreux brevets à travers le monde. Cela peut rendre son utilisation très coûteuse.

3.9 La méthode d'El Gamal à courbe elliptique

alice veut envoyer un message secret à Bob. Tout d'abord, Bob fabrique une clé publique de la manière suivante :
Il choisit une courbe elliptique E définie sur un corps fini $\mathbb{F}_q$ de telle manière que le problème du logarithme discret [4.8] soit plus difficile à résoudre sur $E(\mathbb{F}_q)$ que sur $\mathbb{F}_q$. Il choisit aussi un point P sur E tel que l'ordre de P soit un grand nombre premier. Il choisit un nombre entier secret s et calcule $B = sP$. La courbe E, le corps fini $\mathbb{F}_q$ et les points P et B sont donc la clé publique de Bob, la clé secrète de Bob est donc s.
Pour envoyer le message, alice fait comme suit :

1. Alice télécharge la clé publique de Bob.

2. Alice transforme son message en un point $M \in E(\mathbb{F}_q)$

3. Alice choisit un nombre entier secret k et calcule $M_1 = kP$.

4. Alice calcule $M_2 = M + kB$.

5. Alice envoie M_1 et M_2 à Bob.

6. Bob déchiffre le message en calculant $M = M_2 - sM_1$.

Nous avons cette égalité parce que

$$M_2 - sM_1 = (M + kB) - s(kP) = M + k(sP) - skP = M.$$

Supposons maintenant que Eve connait la clé publique et les points M_1 et M_2. Si elle savait résoudre le problème du logarithme discret 4.8, elle aurait pu utiliser P et B pour trouver s et ainsi calculer $M_2 - sM_1$. Eve pourrait aussi utiliser P et M_1 pour trouver k et calculer $M = M_2 - kB$.

actuellement, on ne connait pas de moyen plus rapide pour retrouver le message initial en ne sachant que ce qui est rendu public du système de cryptage. Donc, a priori, la fiabilité de ce genre de crypto-systèmes dépend fortement des progrès fait en matière de résolution du logarithme discret [4.8].

Remarque. *Il est important qu'alice utilise un k différent à chaque fois qu'elle envoie un message crypté à Bob avec la même clé. En effet, si elle utilise le même k pour deux messages différents M et M', alors $M_1 = M_1'$. Eve ayant intercepté les deux messages codés s'en apercevra et pourra calculer :*

$$M_2' - M_2 = M' - kB - (M - kB) = M' - M$$

alors Eve calculera sans peine M' qui vaut $M - M_2 + M_2'$

3.10 Signature électronique d'El Gamal

Comment prouver à Bob cette fois que le message a bien été envoyé par alice, ce qui est le principe de base de la signature. En effet, nous ne sommes pas sûrs de l'authenticité du message. Eve pourrait très bien se faire passer pour alice en créant elle-même un système de clés privées et publiques et dire que ce sont les clés d'alice.
L'idée est de joindre au message une signature électronique, l'équivalent de

la signature dans le monde physique, qui certifie au destinataire l'identité de l'expéditeur. Nous allons présenter un modèle de signature basé sur les courbes elliptiques et qui à la réputation d'être difficilement falsifiable.
Ce modèle utilise les fonctions de hachages, nous allons donc commencer par donner la définition de ces fonctions.

Definition. *Une fonction de hachage H est une fonction telle que l'image par H de n'importe quel élément est un élément ayant une longueur plus petite, rappelons qu'une fonction de hachage doit satisfaire les trois propriétés suivantes : [22]*

1. *Pour un nombre x donné, $H(x)$ se calcule très rapidement.*
2. *Pour un nombre x_1 donné, il est très difficile de trouver un nombre x_2 tel que*

$$H(x_2) = x_1.$$

3. *Il est très difficile de trouver deux nombres distincts x_1 et x_2 tels que*

$$H(x_1) = H(x_2).$$

(Dans ce cas, on dit que H est fortement sans collision).

Remarque. *Les conditions 2 et 3 empêchent Eve de falsifier la signature. Il existe plusieurs bonnes fonctions de hachages. Par exemple, la fonction MD5 est une fonction de hachage inventée par Ron Rivest. Elle donne des hachés de 128 bits. Des faiblesses ont été trouvées et son utilisation se raréfie . Une autre fonction de hachage est la fonction SHA-2. Elle renvoie une empreinte de 256 bits.*

Supposition. *Alice envoie un message à Bob et elle veut signer électroniquement son message.*

Si elle utilise la signature El Gamal voici comment elle doit s'y prendre : Alice doit tout d'abord créer une clé publique. Pour cela, elle choisit une courbe elliptique E définie sur un corps fini $\mathbb{F}_q$, de manière que le problème du logarithme discret [4.8] soit difficile sur $E(\mathbb{F}_q)$. Elle choisit aussi un point $A \in E(\mathbb{F}_q)$, tel que l'ordre n de P est un grand nombre premier. De plus, elle choisit un nombre secret a et calcule $B = A * P$. Finalement, alice choisit encore deux fonctions, une fonction de hachage H et une fonction f tel que :[22]

$$f : E(\mathbb{F}_q) -> Z.$$

Par exemple, si q est un nombre premier, elle peut prendre $f(x, y) = x \mod q$. La fonction f doit avoir les propriétés suivantes :

1. La cardinalité de $f(E(\mathbb{F}_q))$ doit être grande.

2. Un élément de l'image de f n'a qu'un petit nombre d'antécédents. (Par exemple, pour $f(x, y) = x$, il y a au plus deux points (x, y) qui ont pour image x.)

L'information publique d'Alice est $(E, \mathbb{F}_q, P, B, H, f)$. Elle garde secret le nombre a. L'ordre n de P n'est pas forcément gardé secret, cela n'entrave pas la sécurité du système. Pour signer son document, alice fait comme suit :

1. Elle représente son document sous forme d'un nombre entier m et le hache, c'est-à-dire calcule $H(m)$ (n étant un grand nombre premier, $H(m) \leqq n$. Si tel n'est pas le cas, on sépare le message en blocs $m1, ..., mk$ tels que chaque $H(mi) \leqq n, 1 \leqq i \leqq k$).

2. Elle choisit un nombre entier k avec $PGDC(k, n) = 1$ et calcule $R = k * P$.

3. Elle calcule $s \equiv k^{-1}(H(m) - a * f(R)) \mod n$.
   Le message signé est (m, s, R). Si alice veut garder son message secret, elle peut par exemple le crypter avec RSA et utiliser le message crypté au lieu de m.

Pour vérifier l'authenticité de la signature d'Alice, Bob procède de la manière suivante :

1. Bob télécharge l'information publique d'Alice.

2. Bob calcule

$$V_1 = f(R)B + s * R$$
$$V_2 = H(m) * P.$$

3. Si $V_1 = V_2$ alors la signature est valide.

Montrons que si la signature est valide, alors $V_1 = V_2$.

$V_1 = f(R)B + sR$
$= f(R)aA + (k - 1(H(m) - af(R)) + zn)kA$
$= f(R)aA + (H(m) - af(R))A = H(m)A$
$= V_2$

En fait, $V_1 = V_2$ n'implique pas forcément que la signature soit valide mais il est très difficile de trouver un nombre s' tel que :

$$f(R)B + s' * R = H(m)P$$

sans connaitre ni a, ni k. C'est l'utilisation de la fonction de hachage qui nous garantit ceci.

3.11 Preuve à divulgation nulle de connaissance

Une preuve à divulgation nulle de connaissance ou aussi appelé Zero Knowledge proof par les anglophones est un concept utilisé en cryptologie dans le cadre de l'authentification et de l'identification. Cette expression désigne un protocole sécurisé dans lequel une entité nommée "fournisseur de preuve", prouve mathématiquement ňe autre entité, le "vérificateur", qu'une proposition est vraie sans révéler une autre information que la véracité de la proposition.[14]

En pratique, ce schéma se présente souvent sous la forme d'un protocole de type "stimulation/réponse" (challenge-response). Le vérificateur et le fournisseur de preuve s'échangent des informations et le vérificateur contrôle si la réponse finale est positive ou négative.

Trois propriétés doivent être satisfaites :[14]

1. Consistance [31] : si le fournisseur de preuve et le vérificateur suivent le protocole alors le vérificateur doit toujours accepter la preuve

2. Solidité [32] : si la proposition est fausse, aucun fournisseur de preuve malicieux ne peut convaincre un vérificateur "honnête" que la proposition est vraie et ceci avec une forte probabilité

31. Aussi applé : Completeness
32. Aussi applé : Soundness

3. Sans divulgation de connaissance (zero knowledge) : le vérificateur n'apprend de la part du fournisseur de preuve rien de plus que la véracité de la proposition, il n'obtient aucune information qu'il ne connaissait déjá sans l'apport du fournisseur de preuve.

   Si le vérificateur ne suit pas la procéédure, cette définition reste valable aussi longtemps que le fournisseur de preuve suit la procédure.Les deux premiéres propriétés sont les mêmes qui servent à définir un systéme de preuve interactive, qui est un concept plus général. C'est la troisiéme propriété qui fait le zéro knowledge.

Le principe peut être résumé en 3 phases :[14]

1. Le prouveur prétend connaître une secret.
2. Il veut convaincre le vérifieur qu'il connaît ce secret sans pour autant le lui révéler.
3. Le vérifieur veut être certain que le prouveur ne le trompe pas.

Conclusion

A la lumiére des informations cernées dans ce chapitre, nous avons envisagé de passer à l'étape suivante concernant la conception de notre systéme de chiffrement. Pour le reste de notre étude nous avons penser à utiliser principalement la cryptographique à courbes elliptiques.
Néanmois, un protocole de chiffrement et de vote reste à établir. il y a lieu donc de passer d'une étude thórique à une étude pratique.

Chapitre IV

4 Etude Pratique

Introduction

Cette partie porte sur l'étude pratique du crypto-sytéme, les points : chiffrement, signature, traitement des données, Zero knowlegde Proof [33] et sécurité des concepts utilisés pour notre étude seront traités dans ce chapitre.

4.1 Déroulement d'une élection

Dans un premier temps, nous devrons générer les données nécessaires : chiffrements, signature (courbes, clés, etc.). Les clés publiques et les courbes sont rendus publiques, et les clés privées sont distribués aux participants.

Dans un deuxième temps, les participants sont appelés à voter : un votant a le choix entre "oui", "non" et "blanc", qu'il chiffre avec la clé publique de l'élection, et qu'il signe avec sa clé privée. On vérifie que la signature est correcte, et on vérifie que le bulletin est bien constitué à l'aide d'une preuve à divulgation nulle de connaissance (Zero Knowledge Proof). La signature et les preuves sont publiées, le bulletin encrypté est stocké dans un chiffré contenant la somme des chiffrés de tous les participants.

Lorsque la phase de vote est finie, le paquet de votes est déchiffré et on décompte le nombre de "oui", "non" et "blanc". Les résultats sont publiés.

4.2 Courbe elliptique d'El Gamal

L'implémentation d'un crypto-système à base de courbe elliptique basé sur le schéma de chiffrement expliqué dans la partie précédente, serait difficile à mettre en place. C'est la raison pour laquelle nous utiliserons le crypto-système d'El Gamal pour les courbes elliptiques.

Soit un groupe finie $\mathbb{F}_p$, nous avons a et b qui définissent une courbe sur $\mathbb{F}_p$ d'après l'expression : $E : y^2 = x^3 + ax + b$ et la cardinalité de $E(\mathbb{F}_p) = q$

Un point P d'ordre q est choisie et les informations (p, E, P, q) sont public.

Pour la génération de la clé privé, on choisie aléatoirement un entier d dans l'intervalle $[1, q-1]$ et on calcule $Q = d * P$, la clé privée sera donc Q.

33. Preuve à divulgation nulle de connaissance

4.3 Chiffrement et déchiffrement

Un message m, qui sera dans notre cas la nature du vote émis, devra être chiffré avant d'être envoyer au Polling station pour les traitements. Le chiffrement se fera en utilisant la clé publique Q :

$$Chiff = EQ(m) = (A, B) = (r * P, M + r * Q)$$

Un r aléatoire et choisi lors du chiffrement, et le message chiffré $EQ(m)$ deviens le couple (A, B) qui sera envoyé.

Une fois le message envoyer, seul les détenteurs de la clé privée d pourrons déchiffrer le message m.

Pour le déchiffrement nous aurons besoin du facteur de multiplication d qui a servi a générer la clé $Q = d * P$.

L'équation de déchiffrement : $Dd(C) = B - d * A$

Vérification :
$Dd(Chiff) = B - d * A$
$= (M + r * Q) - d * (r * P)$
$= M + r * Q - (d * P) * r$
$= M + r * Q - r * Q$
$= M$

Nous retrouvons effectivement notre message chiffré M.

4.4 Traitement des données

Les données sont traitées par différents acteurs :
Le participant : c'est celui qui émet le vote, chaque participant possède sa clé.
Polling Station : c'est celui qui récolte les votes chiffré et qui va calculer le résultat et le publie.
Bulletin Board : c'est ici que le polling Station va publier les votes.
Key Storage Trust Party : c'est l'acteur qui va créer les paires de clé publique et de clé privée.

Afin que le polling station puissent calculer le résultat avec une opération d'addition (Vote1 + Vote2 ...), le crypto-système mis en place doit avoir la propriété d'homomorphisme de groupe.

4.4.1 Propriété d'homomorphisme

L'homomorphisme permet d'obtenir le chiffré de la somme de deux chiffrés, sans pour autant déchiffrer ses derniers.
Soit deux messages claires (A_1, B_1) (A_2, B_2) avec C_1 le chiffré de (A_1, B_1) et C_2 le chiffre de (A_2, B_2).

$$C = C_1 + C_2 = (A_1 + a_2, B_1 + B_2)$$

Cette propriété est retrouvée dans le chiffrement par courbe elliptique et est donc une propriété de notre crypto-système (El Gamal).

4.5 Protocole de vote

Afin de pouvoir affirmer que notre système de vote est sécurisé nous devons pouvoir prouver quatre points :
- L'authenticité du vote : seul une personne ayant été autorisé a voter peut voter.
- L'unicité : chaque participant doit pouvoir voté qu'une seul fois
- L'anonymat : personne doit savoir qui à voté pour qui sauf le votant lui même
- L'impartialité : le vote ne doit pas être truqué.

Le respect de ces propriétés doit donc nécessité l'utilisation d'un système cryptographique.

4.5.1 Zero knowledge proof

Le ZKP permet de vérifier qu'un vote est conforme sans divulguer d'information sur celui-ci. afin de vérifier qu'un vote soit conforme il doit vérifier qu'un votant ne peux voter que pour un seul candidat.

Première idée : Pour ce faire nous avons deux listes : une liste de votes classiques et une liste de votes redondants.
- Un vote $V_i = (V_ic, V_ir)$
- S_c est un ensemble contenant les votes classiques.
ZKP_c représente la preuve qu'un vote V_ic soit dans l'ensemble S_c.
- S_r est un ensemble contenant les votes redondants.
ZKP_r représente la preuve qu'un vote V_ir soit dans l'ensemble S_r.
- S_+ est un ensemble contenant l'addition des votes classiques et redondants.
L'addition $V_ic + V_ir$ doit faire partie de cet ensemble.

- ZKP_+ représente la preuve qu'il ne peut y avoir de vote classique "oui" et de vote redondant "non" pour un même vote.
- ZKP_t doit prouver qu'un vote soit pour un seul candidat ("oui", "non" ou "blanc")

Deuxième idée : Nous aurions pu déchiffrer ZKP_+ et vérifier qu'il soit dans l'ensemble S_+ si nous avions plusieurs groupes. Cela aurait révélé la position dans le groupe mais pas le groupe lui-même, mais nous n'avons qu'un groupe donc cela reviens à déchiffrer le vote.
Dans les deux cas la preuve n'aurait pas été réellement "Zero knowledge" vu qu'elle divulguait une information.

4.6 Le protocole de signature ECDSA

Le protocole de signature ECDSA (Elliptic Curve Digital Signature Algorithm), est un algorithme de signature à clé publique utilisant la cryptographie à courbe elliptique.

C'est un des meilleurs algorithme de signature actuelle car il est non seulement plus rapide que les autres algorithmes tel que DSa ou RSa-signature, de plus il utilise un logeur de clé plus petite.

Pour générer et vérifier une signature par ECDSa, nous supposons que l'on a une paire de clé publique et privée Q et S, avec $Q = S * G$, G étant le point de base de la courbe elliptique. On commence tout d'abord par hacher le message que l'on souhaite signer, nous utilisons ici SHa-256.

$$h = hash(m)$$

On génère ensuite un nombre aléatoire k, qui doit être différent pour chaque signature, tel que : $0 < k < n$. On multiplie ensuite k par le point G, et on récupère la coordonnée x que l'on prend modulo n :

$$R = (r_x, r_y) = k * G$$
$$x = rx \mod n$$

Si $x = 0$, on doit prendre un k différent.
Si $x \neq 0$, on calcule y : $y = (h + S * x)/k - 1$ mod n
Si $y = 0$, on doit prendre un k différent.
Si $y \neq 0$, on obtient la signature qui est (x, y).

4.6.1 Vérification de la signature

Les éléments à vérifier sont :

1. Q doit être différent de φ (le point à l'infinie)
2. Q appartient bien à la courbe elliptique généré pour la signature et vérifier que nQ donne bien φ.
3. Vérifier que $0 < x < y$ et $0 < y < n$
4. Inverser $w = 1/y$
5. Calculer u_1 et u_2 avec $u_1 = hash(m) * S \mod n$ et $u_2 = x * w \mod n$
6. Calculer $P = (p_x, p_y) = u_1 * G + u_2 * Q$
7. Pour montrer que la clé est valide, montrer que $p_x = x \mod n$

4.6.2 Sécurité de la signature

Tout les algorithmes connu pour résoudre le logarithme discret [4.8] sur les courbes elliptiques ont une complexité de $O(\sqrt{n})$. Pour avoir une sécurité acceptable nous prenons donc une courbe sur un corps finie $\mathbb{F}_q$ où $q = 2^{256}$ comme suggère le NITS [18] [34].

4.7 Dépouillement

Il est facile pour le Polling Station de calculer le résultat des votes et cela grâce à la propriété d'homomorphisme.

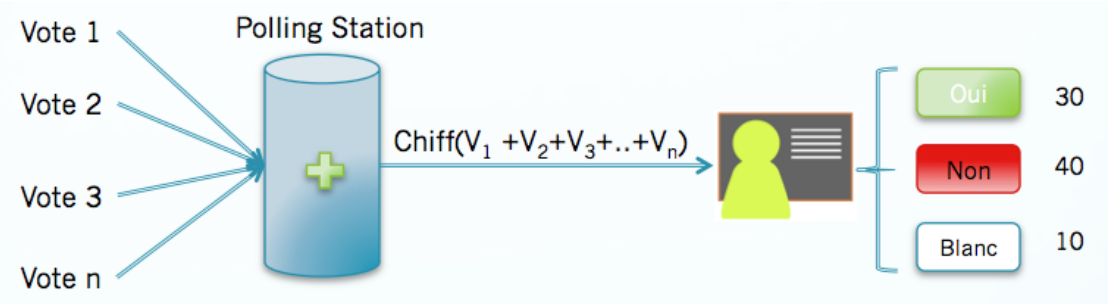


FIGURE 4.1 – Fonctionnement du Dépouillement

34. National Institute of Standards and Technology

Il lui suffit juste de faire la somme de tous les votes chiffrés et de l'envoyer au bulletin Board qui se charge de déchiffrer la somme et de publier le résultat.
Grâce à cette propriété, aucun vote n'est déchiffré, seul l'addition des votes l'est. ainsi les votes conservent leurs anonymats et les traitements sur les données chiffrées est facilité.

4.7.1 Déchiffrement du résultat

Calculer le logarithme discret [4.8] pour chaque paquet, connaitre le nombre de votes par paquet, et comme chaque paquet contient 100 votes le calcul du logarithme discret sera facile à calculer. L'addition des résultats prend en compte seulement les "oui" et les "blancs". On retrouvera le nombre de "non" grâce au nombre de participant réelle. Finalement, il y a publication des résultats du vote.

4.8 Sécurité

Dans cette partie nous nous intéressons aux failles auxquelles notre système est susceptible.
La manière dont sont stockés les votes blancs oblige à la machine de connaitre les messages en clair, cela nous cause donc un problème de confidentialité.
Un bulletin erroné pourrait être inséré car la bonne constitution du bulletin n'est pas vérifiée.
La construction de notre courbe elliptique fait apparaitre une faille de sécurité qui permettrai les timing attaque et qui exploite le fait que P=Q, mais nous n'avons pas pu prouver celle-ci.
Pour l'implémentation nous avons utiliser des courbes elliptiques de petit taille (ordre 2^{16}) et cela par souci d'optimisation des test car notre environnement de développement (sagemathcloud) n'a pas la puissance nécessaire pour faire ces calculs.

La sécurité des courbes elliptiques est directement liée au problème du logarithme discret : pour 2 points de la courbe elliptique $P, Q \in E(\mathbb{F}_q)$ nous avons $Q = kP$, pour retrouver k il faut résoudre le logarithme discret $k = log_p Q$. Pour des nombres suffisamment grand, nous ne connaissons pas de méthode efficace pour résoudre ce problème en un temps raisonnable.

Conclusion

Notre étude pratique a donnée donc naissance a un systéme de chiffrement à base de courbe elliptique d'El Gamal. Ce dernier aura donc une propriété d'homomorphisme de groupe avec l'addition. Il possédera aussi un mécanisme de signature utilisant le protocol ECDSA [35].Tout ce qu'il nous reste à accomplir maintenant c'est la conception et la réalisation de notre crypto-systém. Nous allons donc travailler dans le prochain chapitre sur environnement de dévloppement pour ce systéme et les technologies à utiliser.

35. Elliptic Curve Digital Signature Algorithm

Chapitre V

5 Conception et réalisation

5.1 SageMath

Introduction

La cryptographie à courbe elliptique comme nous l'avons vu précédemment utilise des groupes de nombre très grand. De ce fait il nous faillait un outil de programmation qui puisse manipuler facilement les grands groupes de nombre et possédant des fonctions déjà implémentés pour la manipulation des courbes elliptiques. SageMath viendra donc apporter un début de réponse à nos préocupations.

5.1.1 Qu'es ce que SageMath ?

SaGE est un environnement mathématique libre complet. Il permet de programmer en Python mais aussi d'utiliser des outils libres extérieurs (Maxima en particulier) et des bibliothéque de calcul, sur tout utiliser pour faire du calcul formel et du calcul sur les grands nombres.

Le choix s'est donc orienté vers le logiciel libre SageMath. En effet ce dernier combine la puissance de nombreux programmes libres dans une interface commune basée sur le langage de programmation Python. Cela nous permettra d'implémenter un système cryptographique plus facilement qu'avec un outil de programmation classique.

5.1.2 Serveur WEB SAGE

SaGE se présente aussi sous la forme d'un serveur web Linux. Il ne peut tourner sous d'autres plateformes que par le biais de machines virtuelles Li-

nux. Un utilisateur lancera un navigateur sur un poste quelconque du réseau et poura accéder au serveur web pour utiliser SAGE.

5.2 Le système de vote

5.2.1 Création de l'élection

Tout d'abords un nom et un nombre N de participants est donné, une courbe elliptique aléatoire est générée à partir d'un corps fini de taille randomisée p, nous procédons ensuite au stockage des paramètres :

p La taille du corps fini.

E La courbe.

q Un grand facteur premier.

P un point de la courbe.

Les représentations des votes possibles sont générées comme suit :

"oui" est représenté par P.

"non" par le point nul ϕ de la courbe E.

"blanc" par $(n+1) * P$.

avec n le nombre de bulletins qui sont additionnés dans un paquet.

Le KSTP est ensuite créé, il représentera le partie de confiance qui génère les clés. Il créé le couple clé publique/privée de l'élection, puis jusqu'à **n** couples de clés par courbe elliptique (qu'il créé lui même).

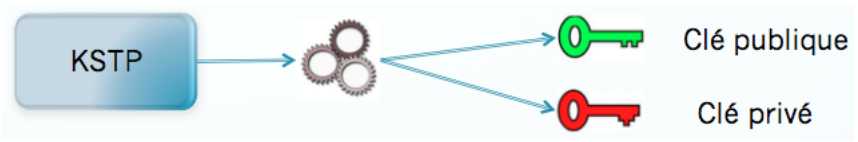


FIGURE 5.1 – Fonctionnement du KSTP

La création de la "Pooling Station" viens ensuite, elle contient une liste qui contiendra les paquets des votes "oui" et "non", qui grâce aux propriétés des courbes contiendra des sommes de n votes chiffrés. Un vote nul pour contenir les votes blancs pour les sommer. Une liste contient les votes signés et une autre est sensé contenir les données pour la Zero Knowledge Proof.

Le Bulletin Board consiste en une fonction d'affichage des données publiques, elle poura etre consulté par n'importe qui et contiendra les donnée publique de l'élection, la Bulletin Board permettra entre autre la transparence des élections réaliser, vu que les donnés pour la vérifications serons disponible publiquement.

5.2.2 Protocol de vote

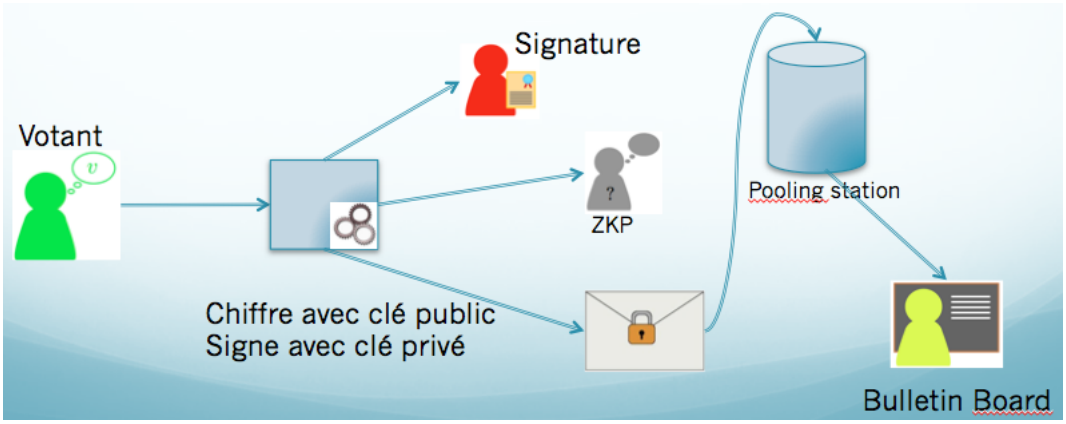


FIGURE 5.2 – Fonctionnement du Protocol

Pour la phase de vote, le participant choisit oui/non/blanc via les fonctions faites pour,et il fournit son numéro de participant, qui est associé à sa signature. Si un vote a déjà été signé alors il a déjà voté, il lui sera donc impossible de voté une seconde fois. Le vote est chiffré en deux exemplaires, l'un est signé pour vérification et publication(pour le Bulletin Board), l'autre doit passer le Zero Knowledge Proof et est ensuite ajouté à la liste des votes si les deux vérifications précédente sont validées.

a chaque vote on vérifie si tous les participants ont voté, si oui on finit l'élection et les résultats sont enregistrés. On compte les "oui" en cassant

le logarithme discret sur le message clair qui est la somme des messages, on additionne tous les résultats pour compter les "oui", idem pour les votes "blancs". Les votes "non" sont comptés en prenant le nombre de participant qui ont voté auquel on soustrait les votes oui et blanc. Toutes les données de l'élection sont ensuite rendu publique sur le Bulletin Board.

La vérification de la signature nous garantit que la même entié ne peux en aucun cas voter plusieurs fois. Le ZKP (Zero knowledge Proof) quand łui est une preuves sur laquelle s'appuie le crypto-stéme pour prouver que le vote d'une personne est bien valide et que cette personne a bien le droit de votér, sans pour autant dévoiler aucune information sur la nature de son vote.

5.3 Les Principales fonctions du Crypto-systém

```
                #Generation de la crourbe.

def random_E_curve():

#Fonction de generation d'une courbe elliptique
#de cardinal premier sur un corps fini premier variable.

#Return un 4-uples forme de :
    # 0, le nombre d'elements du corps fini.
    # 1, la courbe elliptique.
    # 2, un point de la courbe.
    # 3, l'ordre du point



                #Fonction de generation des clefs.

def key_generate(point , order):

#Prend en parametres :
#  point, Un point d'une courbe elliptique.
#  Ordre, L'ordre de ce point.

return : un couple cles publique & privee pour EC El Gamal.



              #Fonction de chiffrement
```

```
def chiffrement (msg , clef_public , point , order ):

#Prend en parametre:
#  msg, un point d'une courbe elliptique representant un message.
#  clef_publique, la clef publique, doit etre un point
#  de la courbe elliptique tel clef_publique = clef_privee * point.
#  point, un point qui a servi a generer la clef publique.
#  order, l'ordre de ce point

return : le message chiffre sous la forme d un couple de points



            #Fonction de dechiffrement

#Prend en parametre:
#  msg_chiffre, un couple de points representant un message chiffre.
#  clef_privee, la clef privee est un entier.

def dechiffrement ( msg_chiffre , clef_privee):
return : le message en clair sous la forme d un point



            #Test le chiffrement homomorphique d'un oui/non

def test_homomorphisme_oui_non (o,n, k, e):

#Prend en parametre:
# o, la representation du oui = point de la courbe elliptique
# passe en parametre dans e.
# n, representation du non = point nul de la courbe elliptique
# passe en parametre dans e.
# k, le couple cles publique/privee
# e, un 4-uplets contenant :
#   0, le nombre d'elements du corps fini definnissant la courbe.
#   1, la courbe elliptique sur laquelle est definie le point.
#   2, un point de la courbe ayant genere le couple de cles k.
#   3, l'ordre du point

return : La reponse du test
```

```
                    #Signature

 def signature_ecdsa(msg,S,e):

#on importe SHA256 pour pouvoir utiliser la fonction de hashage
#Prend en parametre:
#msg, les coordonnees des points.
#S, la cle secrete
#e, la courbe elliptique

return la cle (x,y)



                    #Verfication de la signature

def verif_ecdsa(msg,Q,sig,e):

#retourne un booleen, OUI si la signature est verifiee, NON sinon


                    #Creation KSPT

def new_KSTP(nbParticipant):



                    #Creation du pooling station

def new_PS(e, nbParticipant, clePub):

#Creation du PS : Polling Station
#param :
# e, un 4-uplets contenant :
#  0, le nombre d'elements du corps fini definnissant la courbe en 1.
#  1, la courbe elliptique sur laquelle est definie le point en 2.
#  2, un point de la courbe ayant genere le couple de cles k.
#  3, l'ordre du point
```

```
#  nbParticipant, le nombre de participants

return : un dictionnaire contenant les listes et le nombre de personnes
            ayant actuellement votees.



                    #Le Bulttin board

def bulletinBoard(elec):

#Correspond au panneau d'affichage en cours d'election.
#Affiche donc toutes les donnees pouvant etre
#rendu publique pendant celle-ci.
#Affiche les donnees publique d'une election.
#Parametres:
#elect , l'election a afficher.



                    #Creation d'une election.


def nouvelle_Election(objet, nbParticipant):

#Prend en parametres:
#objet : une chaine de caractere qui defini l'objet de l'election.
#nbParticipant, le nombre de participants participant a l'election.

return : un dictionnaire contenant l objet, le nombre de participants,
    les parametres sous forme de 4-uplets
    la representation du oui, du non, du vote blanc, la PS, le KSTP,
    l etat de l election, le nombre de oui, de non , de blanc.



                    #Fonction de test qui affiche tout

def affiche_Elec_All(elec):

#Affiche toutes les donnees d'une election y compris les privees.
#Toutes les donnees sont rendues publique pour
#verification et consultation.
```

```
#parametres :
# elect , l'election a afficher.
#Correspond a l'affichage en fin d'election.



                    #Depouillement

def depouillement(elec):

#Fonction de depouillement en fin d'election, compte les votes.
#parametres :
# elec, l'election a depouiller.
#Pas de retour, modifie les donnees de l'election.
```

5.4 Les fonctions de test

5.4.1 Fonction de test 1

Nous allons céer une élection, et faire voter des personnes.

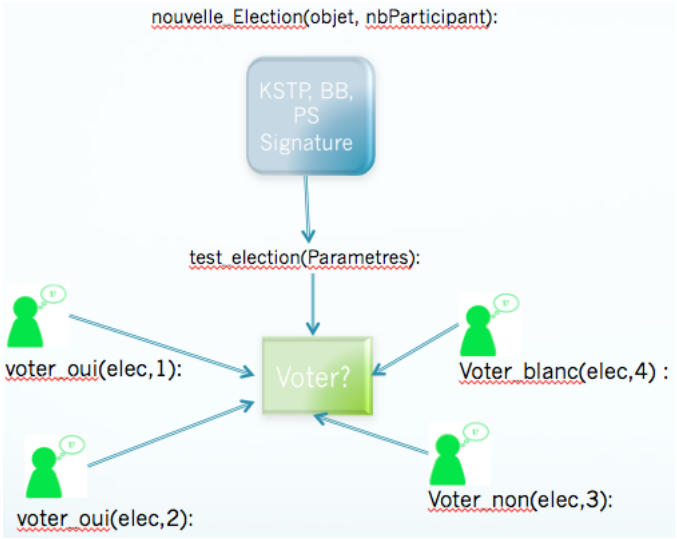


FIGURE 5.3 – Fonctionnement du test Normal

3 fonction ont été devloppé pour : Voter-Oui, Voter-Non, Voter-Blanc. Les votes sont ensuite traités par le systéme. Le Bulltin Board publie les résultats de l'élection et les donnés publique de l'élection.
‘ Cette fonction a permit de mettre en place la solidité du chiffrement réaliser.

Elle démontre aussi le bon fonctionnement de notre systéme.

Exemple de donnée publier par le Bulletin Board :

```
elec = nouvelle_Election("votez-vous  pour le candidat N°1 ?", 3) #nouvelle election
bulletinBoard(elec)
ret = voter_oui(elec,0)    #button  voter oui
ret = voter_non(elec,1)    #button  voter non
ret = voter_oui(elec,2)    #button  voter oui
```

FIGURE 5.4 – Election

```
Objet de l'élection :
votez-vous  pour le candidat N°1 ?

Nombre participant :
3

Taille du corps fini :
122963
Courbe elliptique de l'élection :
Elliptic Curve defined by y^2 = x^3 + 17823*x + 29161 over Finite Field of size 122963
Point de la Courbe elliptique :
(107028 : 49185 : 1)
Ordre du Point :
122753
cle public de l'election :
(121398 : 109988 : 1)

cles public signature :
[(177 : 40524 : 1), (12489 : 30744 : 1), (35074 : 62570 : 1)]

vote signé si nul alors pas encore voté :
[None, None, None]

Election terminée
Nombre de oui :
2
Nombre de non :
1
Nombre de Blanc :
0
```

FIGURE 5.5 – Publication du Bulletin board

5.4.2 Fonctionnement du test à grande échelle

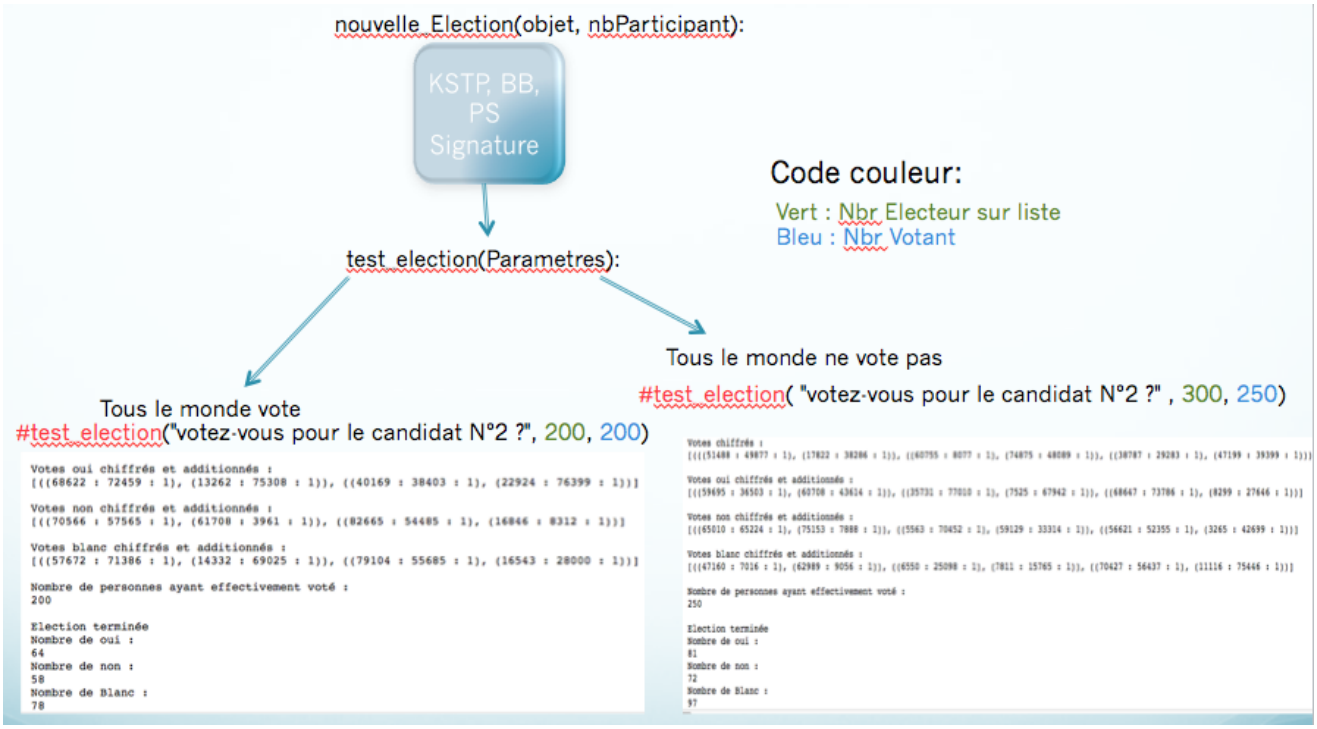


FIGURE 5.6 – Fonctionnement du test a grande échelle

Nous avons un test sur la population qui à effectuer le vote. dans ce cas 200 èlecteur sur 200 ont voter. Les résultat escompter sont qu'une élection doit s'arréter méme si tout le monde n'a pas voter.
Le test génére automatiquement 200 votant et les fait voter "Oui", "Non", "Blanc", avec un certain pourcentage, l'affichage du Bultin Board nous permetra de vérifier les données public publiés.

Le deuxiéme fonction quant a elle prend par exemple 300, il n y a que 250 qui vote, comment le syséme régis a l'addition des votes nos complet ?
Cette fonction de test nous a donc permis d'ťablir un mécanisme pour combler le vide des votes restant. la solution proposée est de les remplacer dans les traitements par des votes "blanc", néanmois l'état de l'électeur sur la liste ne change pas (statut : "n'a pas voter").

Conclusion

Nous arrivons donc à la fin de cette étude, où nous pouvons conclure que nous avons réaliser un systéme de chiffrement à courbes elliptiques pour vote électronique à partir de l'environement SageMath.
Des tests sur notre crypto-systéme ont été réalisé. Nous avons donc analysé dans ce dernier chapitre les détails de la réalisation du sytéme, les différents acteurs [36] et leurs rôles dans le protocol de vote proposé.

36. Pooling Station, Bulletin Board, KSTP ect.

Conclusion géneral et perspectives

Conclusion

Le vote est un sujet particulièrement sensible dans certaines démocraties. Ces derniers années des taux trés faible de participation des électeurs aux différentes élections ont été remarquer cela est dû essentiellement au manque de confiance dans le systéme de vote "classique" ou que le vote papier est trop archaïque pour la nouvelle génération que nous appelons tous " la génération connectée".
Il fallait donc faire appel aux nouvelle technologies de l'informatique pour une meilleur contribution à solutionné ce probléme. Le vote éléctronique est devenu aujourd'hui une partie de la réalité.Puisque quelque pays ont déjá tester les élections pas voie électronique. Néanmois le vote en ligne reste quand lui au stade embryonnaire comme bien même que des systémes comme Helio propose actuellement des votes en ligne avec une bonne sécurité, mais ce dernier n'est malheureusement pas adapté aux votes à grande échelle. L'introduction de ce nouveau systéme de vote poura sans aucun doute séduire les électeurs par sa vitesse, sa simplicité et pour la sûreté des éléctions organisé.

La généralisation d'un tel systéme et la mise en place dans les années à venir d'infrastructures nécessaires à l'élaboration d'un systéme de vote en ligne serait pour n'importe quelle état une révolution technologique et démocratique énorme.

C'est justement dans ce contexte que notre étude est venue apporter une solution aux problémes rencontrés dans les votes en ligne. Garantir la sécurité totale des votes des électeur n'a pas été une mince affaire. Elle a requit une étude approfondie sur la sécurité du chiffrement à utiliser pour le cryptosystéme et le protocole proposé.

L'exploration de la cryptographie à courbes elliptiques a donné des resultats trés intéressents comparés aux autres systémes de chiffrement tel que RSA : un gain important sur la taille des clés, traitement des donnée plus performant (Homomorphiseme), meilleur sécurité grâce au probléme du logarithme discrèt.

Perspectives

Au cour de notre étude, nous avons eu l'occasion de nous confronter à plusieurs problèmes dans le domaine de la cryptographie, à la fois théoriques et pratiques. La possibilité de traiter un sujet moderne comme le vote électronique nous à énormément passionné. C'était également l'opportunité de nous familiariser avec les courbes elliptiques, les preuves à divulgation nulles de connaissance et la signature d'EC El gamal.

Le prototype réaliser fait donc partie d'un plus gros systéme qui servira à régire le vote en ligne du début à la fin, des modules complémentaires doivent donc être ajoutés au crypto-systéme pour avoir une application cohérante, comme par exemple une application WEB qui prendra en charge le coté client fera appel au serveur SageMath qui ce chargera quant à lui du protocole de chiffrement. Ce qui n'a malheureusement pas été realisé dans ce travail.

L'innovation pourrait donc nous amené à imaginer un vote en ligne depuis son Smartphone, pour cela un systéme Android par exemple pourrait faire appel au serveur SageMath via un serveur WEB.

L'autre point important n'ayant pas été realisé réside dans la maniére dans laquelle les clés secrétes v'ont être associées à des comptes utilisateurs, pour que l'électeur n'est pas a retenir sa clé secréte de lui même. Une base de donnée chiffrée pourrais donc faire office d'annuaire. Une personne désirent voter, pourrait alors se déplacer vers la mairie (comme pour la carte d'élection), pour recevoir un identifiant et un mot de passe. Une fois dans son compte d'electeur, une clé secret sera génerée pour le chiffrement de son vote. il faudra néanmois faire attention a ce que l'identité du votant et sa clé secret ne soit pas établie. cela pourrait engendrer des failles de sécurité potentiel.

Dans notre éture malgré une description pécise du travail à effectuer,Le Zero Knowledge Proof n'as pas pu être achevé car les fondements mathématiques pour passer de la théorie à la pratique dépasse largement le cadre d'un travail de reherche de Master 2 et requière des conaissances mathématiques beaucoup plus approfondi.

Table des figures

6 Annex

Annex A

1.A Les nombres de Fermat

Un nombre de Fermat est un nombre qui peut s'écrire sous la forme $2^{2^n}+1$, avec n entier naturel. Le n-ième nombre de Fermat, $2^{2^n}+1$, est noté F_n.

Ces nombres doivent leur nom à Pierre de Fermat, qui émit la conjecture que tous ces nombres étaient premiers. Cette conjecture se révéla fausse, F_5 étant un nombre composé, de même que tous les nombres de Fermat jusqu'à F_{32}. On ne sait pas si les nombres à partir de F_{33} sont premiers ou composés. Les seuls nombres de Fermat premiers connus sont donc F_0, F_1, F_2, F_3, F_4.[4]

2.A Les nombres de Mersenne

En mathématiques et plus précisément en arithmétique, les nombres de Mersenne sont les nombres de la forme : 2^n-1. Ils constituent la suite d'entiers

$$M_n = 2^n - 1, \quad n \geq 1$$

Ces nombres doivent leur nom à un religieux érudit et mathématicien français du $xvii^e$ siécle, Marin Mersenne.[4]

Un nombre premier de Mersenne, est un nombre qui est à la fois de Mersenne et premier. Pour que le n-iéme nombre de Mersenne M_n soit premier, il est nécessaire mais non suffisant que son indice n le soit.[4]

Par exemple :
M_4 n'est pas premier puisque 4 ne l'est pas (d'ailleurs, $24 - 1 = 15 = 3 \times 5$)
M_{11} n'est pas premier non plus bien que 11 le soit : $M_{11} = 211 - 1 = 2047 = 23 \times 89$.

Annex B

1.B Data Encryption Standard (DES)

Le DES ou Data encryption Standard a été l'algorithme cryptographique le plus important de ces derniers 30 ans, il est un Feistel Cipher avec des blocs de 64 bits (taille nominale) et la taille effective de la clé est de 56 bits (Un total de 64 bits avec 8 bits de parité).
Le DES est constitué de 16 étapes avec 16 sous-clés de 48 bits générées (une clé par étape).Il posséde les 4 modes de chiffrement par bloc : ECB, CBC, CFB et OFB.

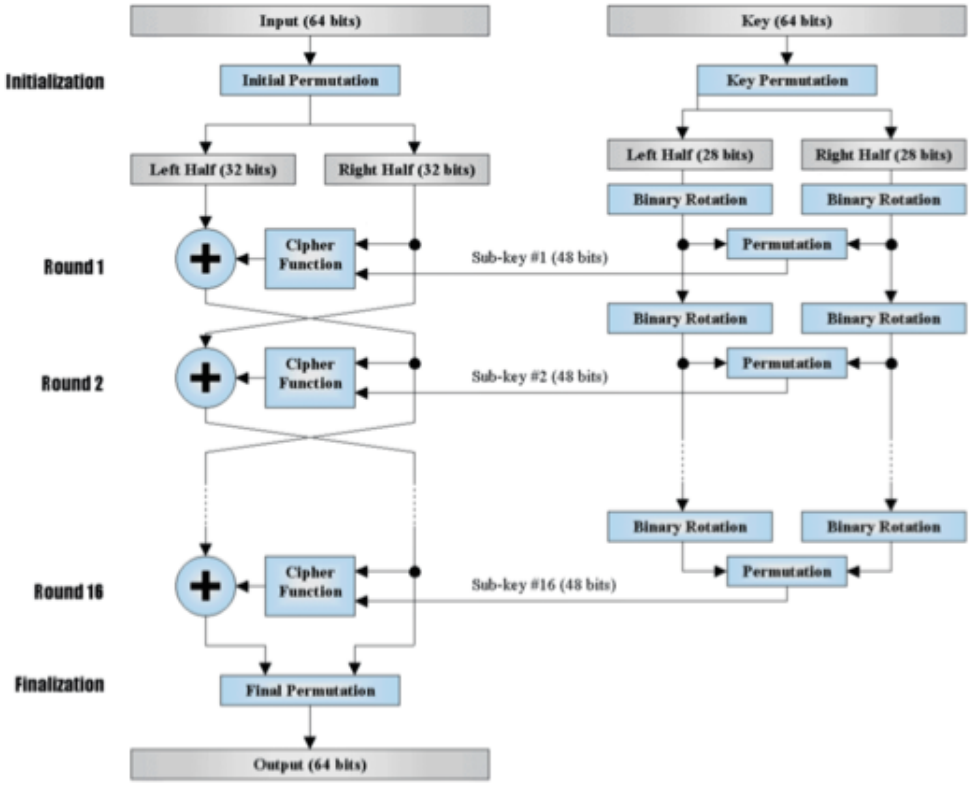


FIGURE 6.1 – Schéma de fonctionnement de DES

2.B Le Systéme AES

L'Advanced Encryption Standard (AES) est l'un des algorithmes de cryptage aujourd'hui le plus fréquemment utilisé et le plus sécuritaire. Son histoire

de succès a débuté en 1997, lorsque le NIST [37] à annoncé la recherche pour un successeur au standard de cryptage vieillissant DES. Un algorithme, appelé "Rijndael", développé par les cryptographes belges Daemen et Rijmen, a excellé dans la sécurité aussi bien dans la performance et la flexibilité. Il a été annoncé officiellement comme le nouveau standard de cryptage AES en 2001. Les algorithmes sont basés sur plusieurs substitutions, permutations et transformations linéaires, chacune réalisée sur des blocs de données de 16 octets. Ces opérations sont répétées plusieurs fois, appelées "rondes". Durant chaque ronde, une clé de ronde unique est calculée de la clé de cryptage, et incorporée dans les calculs. Basé sur cette structure de bloc de l'aES, le changement d'un seul bit soit dans la clé, ou soit dans le bloc de texte brut donne un bloc de texte de chiffrement complétement différent ce qui est un avantage clair sur les chiffrements de flux traditionnels.La différence entre AES-128, AES-192 et AES-256 , c'est la longueur de la clé : 128, 192 ou 256 bits.

37. National Institute of Standards and Technology

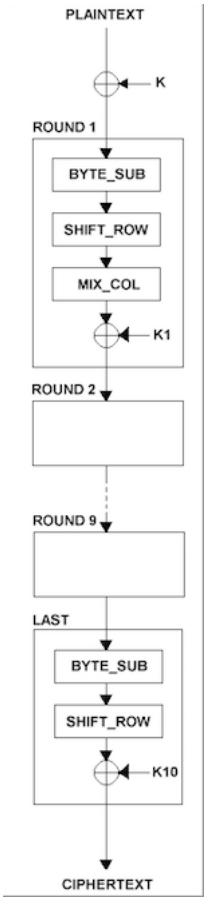


FIGURE 6.2 – Principe du chiffrement à clé Privée AES

Chiffrement : Le schéma suivant décrit le déroulement du chiffrement :

1. **BYTE-SUB (Byte Substitution)** est une fonction non-linéaire opérant indépendament sur chaque bloc à partir d'un table dite de substitution.

2. **SHIFT-ROW** est une fonction opérant des décalages (typiquement elle prend l'entrée en 4 morceaux de 4 octets et opére des décalages vers la gauche de 0, 1, 2 et 3 octets pour les morceaux 1, 2, 3 et 4 respectivement).

3. **MIX-COL** est une fonction qui transforme chaque octet d'entrée en une combinaison linéaire d'octets d'entrée et qui peut étre exprimée mathématiquement par un produit marticiel.

4. **XOR** : le + entouré d'un cercle désigne l'opération de OU exclusif (XOR).
   Ki est la ième sous-clé calculée par un algorithme à partir de la clé principale K.

Déchiffrement : consiste à appliquer les opérations inverses, dans l'ordre inverse et avec des sous-clés également dans l'ordre inverse.

3.B Le Systéme RSA

Le systéme RSA à été Publié en 1977 par Ron Rivest, Adi Shamir et Leonard Adleman de l'Institut de technologie du Massachusetts (MIT),il est Le premier systéme à clé publique solide à avoir été inventé,le RSA est fondé sur la difficulté de factoriser des grands nombres.
Principe :

1. La clé secrète : 2 grands nombres premiers p et q
2. La clé publique : n = pq ; un entier e premier avec (p-1)(q-1)

Il est facile de fabriquer de grands nombres premiers p et q (+100 chiffres),Un nombre n de la forme pq ou‘ p et q sont deux grands nombres premiers est appelé un module RSa. Il est très difficile de retrouver les facteurs p et q à partir de n seul .

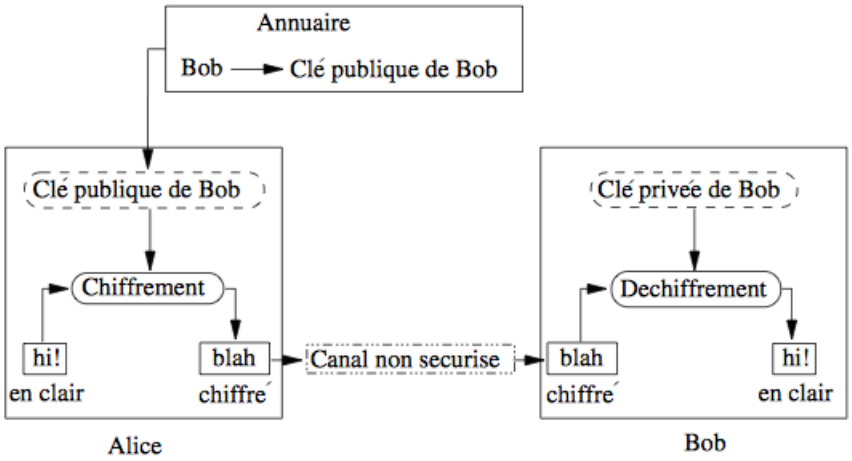


FIGURE 6.3 – Principe du chiffrement à cle publique RSa

Chiffrement : le chiffrement d'un message M en un message codé C se fait suivant la transformation suivante :

$$C = M^e \mod n$$

Déchiffrement : il s'agit de calculer la fonction réciproque

$$M = C^d \mod n$$
$$\text{tel que } e.d = 1 \mod [(p-1)(q-1)]$$

Annex C

1.C Secure Hash Algorithm

SHA-2 (Secure Hash algorithm) est une famille de fonctions de hachage qui ont été conÇues par la National Security agency des États-Unis (NSA), sur le modéle des fonctions SHa-1 et SHa-0, elles-mêmes fortement inspirées de la fonction MD4 de Ron Rivest.

Telle que décrite par le National Institute of Standards and Technology (NIST), elle comporte les fonctions, SHA-256 et SHA-512 dont les algorithmes sont similaires mais opérent sur des tailles de mot différentes (32 bits pour SHa-256 et 64 bits pour SHa-512).

Comme toutes les fonctions de hachage les fonctions SHA-2 prennent en entrée un message de taille arbitraire, et produisent un résultat appelé "hash" qui est lui de taille fixe.

Annex D

1.D La loi de groupe des courbes de Weierstrass

La forme de Weierstrass : $y^2 = x^3 + ax + b$

1. addition $P(x_1, y_1) + Q(x_2, y_2)$:

   $x = (y_2 - y_1)^2/(x_2 - x_1)^2 - x_1 - x2$
   $y = (2x_1 + x_2)(y_2 - y_1)/(x_2 - x_1) - (y_2 - y_1)^3/(x_2 - x_1)^3 - y1$

2. Multiplication 2P
   $x = (3x_1^2 + a)^2/(2y_1)2 - x_1 - x_1$
   $y = (2x_1 + x_2)(3x_1^2 + a)/(2y_1) - (3x_1^2 + a)^3/(2y_1)^3$

Références bibliographiques

[1] Explication sur le vpn `http://14abril.tk/soqyk/tunnel-vpn-7843.php`.

[2] Python documentation `https://www.python.org/doc/`.

[3] Safecurves : Choosing safe curves for elliptic-curve cryptography `http://safecurves.cr.yp.to`.

[4] Wikipédia - `https://fr.wikipedia.org/`.

[5] Ben Adida. Helios : Web-based open-audit voting. In *USENIX Security Symposium*, volume 17, pages 335–348, 2008.

[6] Michaël AMAND and Mohamed NSIRI. Etude d'un système de détection d'intrusion comportemental pour l'analyse du trafic aéroportuaire. *Rapport de projet LENAC*, 2011.

[7] Zineb BENDELLA. *Gestion de la sécurité d'une application Web à l'aide d'un IDS comportemental optimisé par l'algorithme des K-means*. PhD thesis, 2014.

[8] David Bernhard, Véronique Cortier, Olivier Pereira, Ben Smyth, and Bogdan Warinschi. Adapting helios for provable ballot privacy. In *European Symposium on Research in Computer Security*, pages 335–354. Springer, 2011.

[9] Kaouthar Bousselam. *Résistance des circuits cryptographiques aux attaques en faute*. PhD thesis, Université Montpellier II-Sciences et Techniques du Languedoc, 2012.

[10] Michael Brown, Darrel Hankerson, Julio López, and Alfred Menezes. *Software implementation of the NIST elliptic curves over prime fields*. Springer, 2001.

[11] Bylaws. The open web application security project `https://www.owasp.org`.

[12] M Àngels Cerveró, Víctor Mateu, Josep M Miret, Francesc Sebé, and Javier Valera. An efficient homomorphic e-voting system over elliptic curves. In *International Conference on Electronic Government and the Information Systems Perspective*, pages 41–53. Springer, 2014.

[13] Bernard Cousin. Sécurité des réseaux informatiques.

[14] Renaud Dumont. introduction à la cryptographie et à la sécurité informatique. *seconde partie du cours de système de conduite des ordinateurs*, 2006.

[15] Abdelaziz Elaabid. *Attaques par canaux cachés : expérimentations avancées sur les attaques template.* PhD thesis, Université Paris VIII Vincennes-Saint Denis ; Ecole nationale supérieure des telecommunications-ENST, 2011.

[16] Catherine Goldstein. L'arithmétique de pierre fermat dans le contexte de la correspondance de mersenne : une approche microsociale. *Sciences et techniques en perspective*, 8(1) :14–47, 2004.

[17] Darrel Hankerson, Julio López Hernandez, and Alfred Menezes. Software implementation of elliptic curve cryptography over binary fields. In *Cryptographic Hardware and Embedded Systems—CHES 2000*, pages 1–24. Springer, 2000.

[18] Darrel Hankerson and Alfred Menezes. Nist elliptic curves. In *Encyclopedia of Cryptography and Security*, pages 843–844. Springer, 2011.

[19] Patrick Ingram. Multiples of integral points on elliptic curves. *Journal of Number Theory*, 129(1) :182–208, 2009.

[20] Orestis Ioannou. Procédés cryptographiques pour le vote électronique. 2015.

[21] Innovation is Freedom. Ddos `https://www.ovh.com/fr/anti-ddos/principe-anti-ddos.xml`.

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

[23] Sondes Kallel Khemiri. *C1 Introduction : Réseaux Locaux.* PRISM/HPC-NETS, sondes.kallel@prism.uvsq.fr.

[24] Ann Hibner Koblitz, Neal Koblitz, and Alfred Menezes. Elliptic curve cryptography : The serpentine course of a paradigm shift. *Journal of Number theory*, 131(5) :781–814, 2011.

[25] linux France. Les mécanismes de chiffrement : Chapter 23. le chiffrement.

[26] Santi Martínez Rodríguez, M Cerveró Abelló, Víctor Mateu Messeguer, Josep Maria Miret, and Francesc Sebé Feixas. Elliptic curve array ballots for homomorphic tallying elections. *Lecture Notes in Computer Science (LNCS), 2015, vol. 9265, p 334-347*, 2015.

[27] Victor S Miller. Use of elliptic curves in cryptography. In *Advances in Cryptology—CRYPTO'85 Proceedings*, pages 417–426. Springer, 1985.

[28] Tuyet Tram DANG NGOC. *Introduction a la sécurité réseaux*. Université de Cergy-Pontoise, dntt@u-cergy.fr, 2012-2013 edition.

[29] Université Kasdi Merbah Ouargla. *Introduction aux réseaux et modèle OSI*, Février 2013.

[30] Claude Sevrin. *Réseaux et Télecom*. 2003.

[31] Pierre Verneuil. *Cryptographie à base de courbes elliptiques et sécurité de composants embarqués*. PhD thesis, Bordeaux 1, 2012.

[32] Marie Virat. *Courbes elliptiques sur un anneau et applications cryptographiques*. PhD thesis, Université Nice Sophia Antipolis, 2009.