Remerciements

Nous tenons à témoigner notre reconnaissance à DIEU tout puissant, qui nous a aidé et bénit par sa volonté durant toute cette période.

Notre profonde gratitude et sincères remerciements vont à notre promotrice Mme HADAOUI Epse SKENDRAOUI pour sa présence continuelle, son encouragement et sa patience tout au long de ce travail.

Nous adressons nos remerciements aux membres du jury, devant qui, nous avons l'honneur d'exposer notre travail, et qui ont pris la peine de lire ce mémoire pour juger son contenu.

Nous réservons ici une place particulière pour remercier vivement tous ceux qui, d'une manière ou d'une autre, nous ont aidés et encouragés à la réalisation de ce modeste travail.

Dédicaces:

Mes parents, qui n'ont jamais cesser de travailler pour mon confort et ma réussite, eux qui ont toujours eu confiance en moi, qui m'ont encouragé et cru en moi

Ainsi que toute ma famille SELLAH et ZAGHIEM, qui m'encourager lors des moments les plus difficiles.

Mes amis grâce à qui j'ai passé une année formidable (Moh, Chakib, Salim, HakimOff, Fateh, Monaime, Sofiane, Khaled, Nassim).
A toute la section master 2 RMSE

Promotion (2017/2018)

Et pour finir, une dédicace particulière à mon binôme Youcef.

Rafik

A ma mère, qui n'a jamais cessé de travailler pour mon confort et ma réussite, elle qui a toujours eu confiance en moi, qui m'a encouragé et cru en moi

Ainsi que toute la famille HAMID, qui m'encouragée lors des moments les plus difficiles.
A tous mes amis grâce à qui j'ai passé une année formidable (Moh, Chakib, Hakim, Amine, Fateh , Monaime, Djamel , nabil).
A toute la section master 2 RMSE

Promotion (2017/2018)

Et pour finir, une dédicace particulière à mon binôme Rafik et mon cousin billal

Youcef

Sommaire

Liste des figures:

Liste des Tableaux:

Introduction générale :

La sécurité Informatique est devenue une préoccupation importante des utilisateurs et des entreprises dans tous les domaines. Vu l'expansion et l'importance grandissante des réseaux informatiques, lesquels réseaux ont engendré le problème de sécurité des systèmes d'information; Dans la plupart d'organisations informatisées, partager les données directement entre machines est leur souci majeur. Il s'avère indispensable de renforcer les mesures de sécurités, dans le but de maintenir la confidentialité, l'intégrité et le contrôle d'accès au réseau pour réduire les risques d'attaques. De nos jours, la nécessité de cacher ou de casser une information rentre dans un vaste ensemble appelé cryptographie.

Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre inintelligibles sans une action spécifique. La signature électronique (parfois appelée signature numérique) est un mécanisme permettant de garantir l'intégrité d'un document électronique et d'en authentifier l'auteur. L'objectif principal de notre projet de fin d'étude, est de développer une application Android qui contribue à signer des messages et qui assure authentification et l'intégrité.

Ce mémoire comporte cinq chapitres:

- ✓ Le premier chapitre traite la sécurité informatique.
- ✓ Le deuxième chapitre consacré pour la cryptographie et ses méthodes.
- ✓ Le troisième définit le fonctionnement de l'algorithme DSA.
- ✓ Le quatrième expose une étude sur le système d'exploitation ANDROID.
- ✓ Le cinquième chapitre décrit l'implémentation de notre solution l'algorithme DSA sous-Android et la mise en œuvre de notre solution.

Et pour finir notre mémoire on a mis en place une conclusion générale pour conclure notre travail.

Chapitre I :

La Sécurité Informatique

Introduction :

Les réseaux et systèmes informatiques sont devenus des outils indispensables au fonctionnement des entreprises. Ils sont aujourd'hui déployés dans tous les secteurs professionnels : les banques, les assurances, la médecine ou encore le domaine militaire. Initialement isolés les uns des autres, ces réseaux sont à présent interconnectés et le nombre de points d'accès ne cessent de croître. Ce développement phénoménal s'accompagne naturellement de l'accroissement du nombre d'utilisateurs, qui ne sont pas forcément pleins de bonnes intentions vis-à-vis de ces réseaux. Dès lors que ces réseaux sont apparus comme des cibles d'attaques potentielles qui peuvent être exploitées par des hackers, et les sécuriser est indispensable.

I.1 Sécurité Informatique:

I.1.1 Définition: [1]

La sécurité informatique est un ensemble de moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir sa sécurité. En général, la sécurité d'un réseau englobe celle du système informatique sur lequel il s'appuie.

I.1.2 Objectif de la sécurité Informatique: [2]

L'objectif générale de la sécurité informatique consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu :

- Empêcher des personnes non autorisées d'agir sur le système de façon malveillante
- Empêcher les utilisateurs d'effectuer des opérations volontaire ou involontaires capables de nuire au système
- Garantir la non-interruption d'un service.

I.1.3 Services principaux de la sécurité Informatique

Pour remédier aux failles et pour contrer les attaques, la sécurité informatique se base sur un certain nombre de services qui permettent de mettre en place une réponse appropriée à chaque menace. Les principaux services sont :

La confidentialité: consiste à protéger les données transmises contre les attaques passives, et protéger les flux de données contre l'analyse, et préserver le secret des données transmises. Seulement les entités communicantes sont capables d'observer les données.

L'intégrité de données: Vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication (de manière fortuite ou intentionnelle).

La disponibilité: C'est la propriété qui permet a un système ou une ressource du système pour qu'il soit accessible et utilisable suite à la demande d'une entité autorisée.

La non répudiation: Empêche l'émetteur ou le receveur de nier avoir transmis ou reçu un message.

L'authentification: L'authentification consiste à assurer l'identité d'un utilisateur, c'est-à-dire de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être. [3]

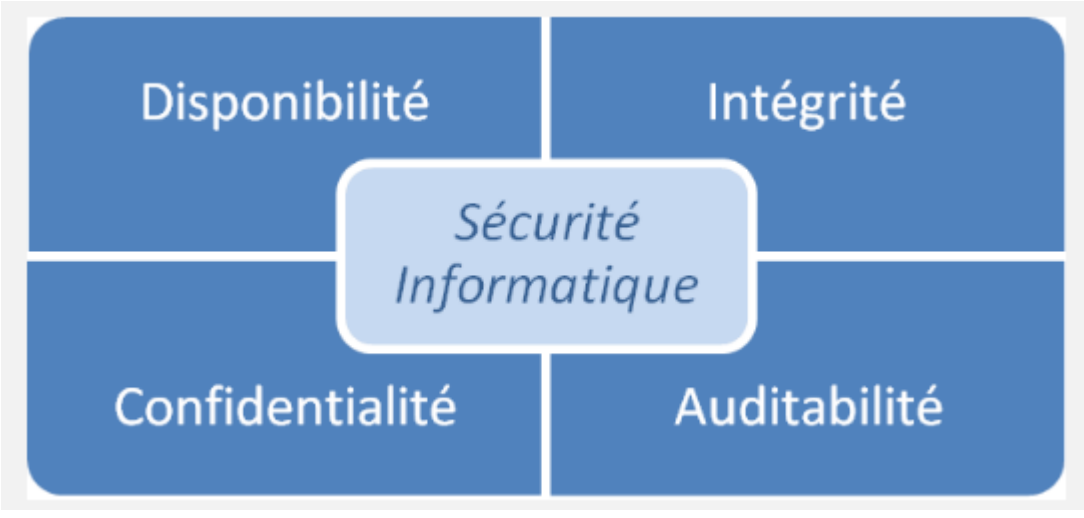

Figure I.1: Services de la sécurité Informatique

I.2 Les causes de l'insécurité [4]

On distingue généralement deux types d'insécurité :

- **L'état actif d'insécurité :** c'est-à-dire la non-connaissance par l'utilisateur des fonctionnalités du système, dont certaines pouvant lui être nuisibles (par exemple la non-désactivation de services réseaux non nécessaires à l'utilisateur)
- **L'état passif d'insécurité :** c'est-à-dire lorsque l'administrateur (ou l'utilisateur) d'un système ne connaît pas les dispositifs de sécurité dont il dispose.

I.3 Les Attaques

I.3.1 Motivation d'un attaquant :

Les motivations d'un attaquant que l'on appelle communément "pirate" ou "hacker» peuvent être multiples :

- L'attirance de l'interdit.
- Le désir d'argent (violer un système bancaire par exemple).
- Le besoin de renommée (impressionner des amis).
- L'envie de nuire (détruire des données, empêcher un système de fonctionner).
- Vérification de la sécurité d'un système.

I.3.2 Les types d'attaque [5][6]

Les systèmes informatiques mettent en œuvre différentes composantes, allant de l'électricité pour alimenter les machines au logiciel exécuté via le système d'exploitation et utilisant le réseau.
Les attaques peuvent intervenir à chaque maillon de cette chaîne, pour peu qu'il existe une vulnérabilité exploitable.

I.3.2.1 Les attaques directes.

C'est la plus simple des attaques. L'hacker attaque directement sa victime à partir de son ordinateur. La plupart des "script kiddies" utilisent cette technique. En effet, les programmes de hack qu'ils utilisent ne sont que faiblement paramétrables, et un grand nombre de ces logiciels envoient directement les paquets à la victime.

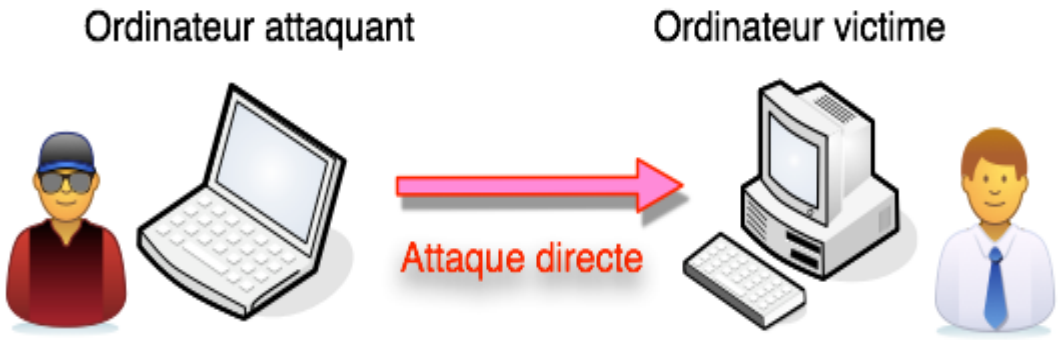

Figure I.2 : Attaque Directe

Dans ce type d'attaque, il y a de grandes chances d'intercepter l'origine de l'attaque, identifiant par la même occasion l'identité de l'attaquant.

I.3.2.2 Les attaques indirectes par rebond

Cette attaque est très prisée des hackers. En effet, le rebond a deux avantages :

- Masquer l'identité (l'adresse IP) du hacker.
- Éventuellement, utiliser les ressources de l'ordinateur intermédiaire car il est plus puissant (CPU, bande passante...) pour attaquer.

Le principe en lui-même, est simple : Les paquets d'attaque sont envoyés à l'ordinateur intermédiaire, qui répercute l'attaque vers la victime. D'où le terme de rebond.

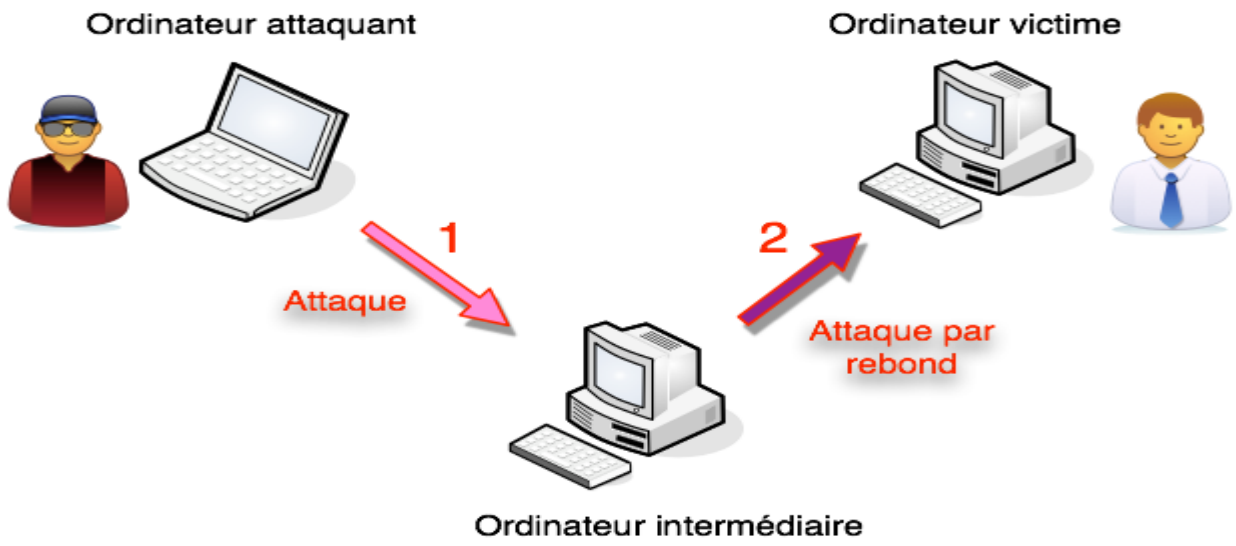

Figure I.3: Attaque indirecte par rebond

Dans ce type d'attaque, il est difficile de remonter à la source, mais il est plus simple de remontrez à l'ordinateur intermédiaire.

I.3.2.3 Les attaques indirectes par réponse

Cette attaque est un dérivé de l'attaque par rebond. Elle offre les mêmes avantages, du point de vue du hacker. Mais au lieu d'envoyer une attaque à l'ordinateur intermédiaire pour qu'il la répercute, l'attaquant va lui envoyer une requête.

L'ordinateur victime va exécuter la requête vis à vis d'un autre ordinateur ou d'un site auquel il a accès normalement.

Et c'est cette réponse à la requête qui va être envoyée à l'ordinateur du hacker.

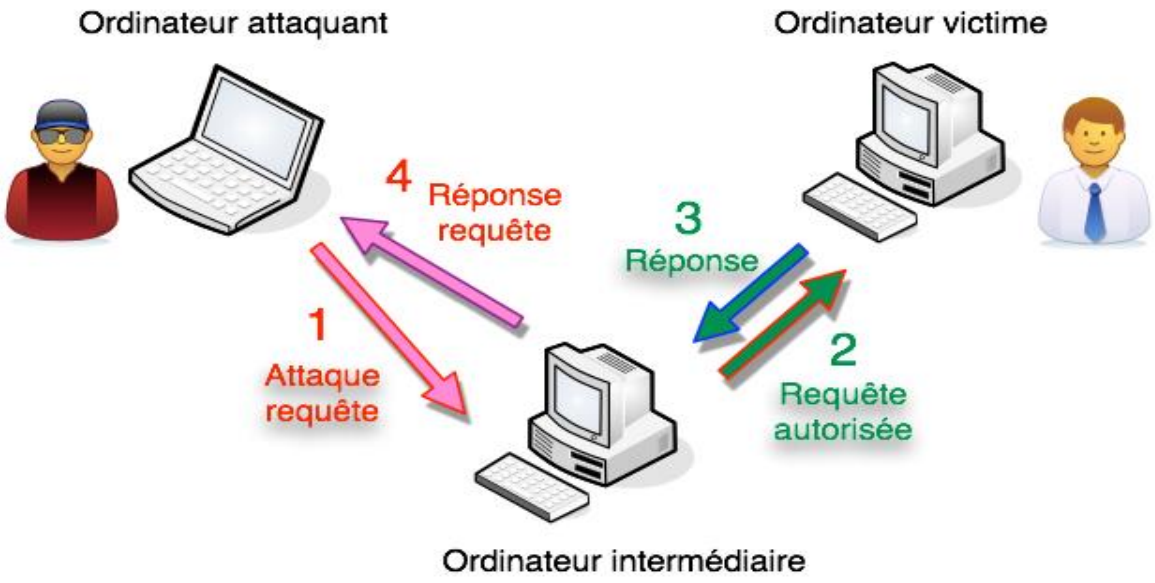

Figure I.4: Attaques indirecte par réponse

I.3.3 Les techniques d'attaques :

Dans cette section, nous classerons les différentes attaques informatiques connues à ce jour, évidemment il s'agit d'une liste non-exhaustive, car il existe des centaines, voire des milliers de techniques permettant de mettre en échec un système de sécurité informatique qui dépendent de l'intelligence du pirate, de celle de ceux qui mettent le système de sécurité en place, et de la complexité du système informatique lui-même.

I.3.3.1 Les attaques virales :

a Cheval de trois :

Un cheval de Troie (*Trojan horse*) est un programme qui exécute des instructions sans l'autorisation de l'utilisateur. Ces instructions sont généralement nuisibles à l'utilisateur, et qui une fois installé sur un ordinateur y effectue des actions cachées et pernicieuses.

Le cheval de Troie ne se réplique pas

b Virus :

Un segment de programme qui, lorsqu'il s'exécute, se reproduit en s'adjoignant à un autre programme (du système ou d'une application), et qui devient ainsi un cheval de Troie. Puis le virus peut ensuite se propager à d'autres ordinateurs (via un réseau) à l'aide du

programme légitime sur lequel il s'est greffé. Il peut également avoir comme effets de nuire en perturbant plus ou moins gravement le fonctionnement de l'ordinateur infecté.

c Ver :

Un programme autonome qui se reproduit et se propage à l'insu des utilisateurs. Contrairement aux virus, un ver n'a pas besoin d'un logiciel hôte pour se dupliquer. Le ver a habituellement un objectif malicieux, par exemple : espionner l'ordinateur dans lequel il réside, offrir une porte dérobée à des pirates informatiques, détruire des données sur l'ordinateur infecté ; envoyer de multiples requêtes vers un serveur internet dans le but de le saturer. Le ver Blaster avait pour but de lancer une attaque par déni de service sur le serveur de mises à jour de Microsoft.

I.3.3.2 Les attaques par réseau :

a Attaque de l'homme du milieu :

« L'attaque de l'homme du milieu » est une attaque qui a pour but d'intercepter les communications entre deux parties, sans que ni l'une ni l'autre ne puisse se douter que le canal de communication entre elles a été compromis. Le canal le plus courant est une connexion à Internet de l'internaute lambda. L'attaquant doit d'abord être capable d'observer et d'intercepter les messages d'une victime à l'autre.

b Attaque par déni de service :

Une « attaque par déni de service » est une attaque ayant pour but de rendre indisponible un service, d'empêcher les utilisateurs légitimes d'un service de l'utiliser. Il peut s'agir de:

- L'inondation d'un réseau afin d'empêcher son fonctionnement ;
- La perturbation des connexions entre deux machines, empêchant l'accès à un service particulier ;
- L'obstruction d'accès à un service à une personne en particulier ;
- Également le fait d'envoyer des millions de kilooctets à une box wi-fi.

L'attaque par déni de service peut ainsi bloquer un serveur de fichiers, rendre impossible l'accès à un serveur web ou empêcher la distribution de courriel dans une entreprise.

L'attaquant hacker n'a pas forcément besoin de matériel sophistiqué. Ainsi, certaines attaques DoS (attaque par déni de service) peuvent être exécutées avec des ressources limitées contre un réseau beaucoup plus grand et moderne. On appelle parfois ce type

d'attaque « attaque asymétrique » (en raison de la différence de ressources entre les protagonistes). Un hacker avec un ordinateur obsolète et un modem lent peut ainsi neutraliser des machines ou des réseaux beaucoup plus importants.

c Balayage de port :

Le « balayage de port » (*port scanning* en anglais) est une technique servant à rechercher les ports ouverts sur un serveur de réseau.

Cette technique est utilisée par les administrateurs des systèmes informatiques pour contrôler la sécurité des serveurs de leurs réseaux. La même technique est aussi utilisée par les pirates informatiques pour tenter de trouver des failles dans des systèmes informatiques. Un balayage de port effectué sur un système tiers est généralement considéré comme une tentative d'intrusion, car un balayage de port sert souvent à préparer une intrusion.

d Usurpation d'IP :

L'usurpation d'adresse IP est une technique utilisée en informatique qui consiste à envoyer des paquets IP en utilisant une adresse IP source qui n'a pas été attribuée à l'ordinateur qui les émet. Le but peut être de masquer sa propre identité lors d'une attaque d'un serveur, ou d'usurper en quelque sort l'identité d'un autre équipement du réseau pour bénéficier des services auxquels il a accès.

e Usurpation d'ARP :

L'usurpation ARP est une technique qui modifie le cache ARP. Le cache ARP contient une association entre les adresses matérielles des machines et les adresses IP, l'objectif du pirate est de conserver son adresse matérielle, mais d'utiliser l'adresse IP d'un hôte approuvé. Ces informations sont simultanément envoyées vers la cible et vers le cache. A partir de cet instant, les paquets de la cible seront routés vers l'adresse matérielle du pirate.

I.3.3.3 Les attaques de système :

a L'écran bleu de la mort :

« L'écran bleu de la mort » se réfère à l'écran affiché par le système d'exploitation Windows lorsqu'il ne peut plus récupérer une erreur système ou lorsqu'il est à un point critique d'erreur fatale. Il y a deux types d'écrans d'erreur, dont l'un est l'écran bleu de la mort, qui a une signification d'erreur plus sérieuse que l'autre. En général la vue de cet écran

signifie que l'ordinateur est devenu complètement inutilisable. Pour certain Black Hats leur but et d'arriver à provoquer cet « écran bleu de la mort » sur le plus d'ordinateur possible.

b Fork Bomb :

Une « fork bomb » fonctionne en créant un grand nombre de processus très rapidement afin de saturer l'espace disponible dans la liste des processus gardée par le système d'exploitation. Si la table des processus se met à saturer, aucun nouveau programme ne peut démarrer tant qu'aucun autre ne termine. Même si cela arrive, il est peu probable qu'un programme utile démarre étant donné que les instances de la bombe attendent chacune d'occuper cet emplacement libre.

Non seulement les fork bombes utilisent de la place dans la table des processus, mais elles utilisent chacune du temps processeur et de la mémoire. En conséquence, le système et les programmes tournant à ce moment-là ralentissent et deviennent même impossibles à utiliser.

I.3.3.4 Les attaques de mots de passe :

a Attaque par dictionnaire :

L'« attaque par dictionnaire » est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Elle consiste à tester une série de mots de passe potentiels, les uns à la suite des autres, en espérant que le mot de passe utiliser pour le chiffrement soit contenu dans le dictionnaire. Si ce n'est pas le cas, l'attaque échouera.

Cette méthode repose sur le fait que de nombreuses personnes utilisent des mots de passe courants (par exemple : un prénom, une couleur ou le nom d'un animal). C'est pour cette raison qu'il est toujours conseillé de *ne pas utiliser de* mot de passe comprenant un mot ou un nom.

L'attaque par dictionnaire est une méthode souvent utilisée en complément de l'« attaque par force brute » qui consiste à tester, de manière exhaustive, les différentes possibilités de mots de passe. Cette dernière est particulièrement efficace pour des mots de passe n'excédant pas 5 ou 6 caractères.

b Attaque par force brute :

L' « attaque par force brute » est une méthode utilisée en cryptanalyse pour trouver un mot de passe ou une clé. Il s'agit de tester, une à une, toutes les combinaisons possibles.

Cette méthode est en général considérée comme la plus simple concevable. Elle permet de casser tout mot de passe en un temps fini indépendamment de la protection utilisée, mais le temps augmente avec la longueur du mot de passe. En théorie la complexité d'une attaque par force brute est une fonction exponentielle de la longueur du mot de passe, la rendant virtuellement impossible pour des mots de passe de longueur moyenne.

I.3.3.5 Attaque de site web :

a Defacement :

Un defacement (*defacing* en anglais) est un anglicisme désignant la modification non sollicitée de la présentation d'un site web, à la suite du piratage de ce site. Il s'agit donc d'une forme de détournement de site Web par un hacker. En général les hackers s'en servent pour laisser un message. Les defacements sont provoqués par l'utilisation de failles présentes sur une page Web ou tout simplement une faille du système d'exploitation du serveur web. La plupart du temps, les sites d'effacés le sont uniquement sur la page d'accueil. Le defacement n'entraîne pas en soi de perte de données.

b Cross-site Scripting :

Le Cross-site Scripting (abrégé XSS), est un type de faille de sécurité des sites web permettant d'injecter du contenu dans une page, permettant ainsi de provoquer des actions sur les navigateurs web visitant la page. Les possibilités des XSS sont très larges puisque l'attaquant peut utiliser tous les langages pris en charge par le navigateur (JavaScript, Java, Flash…) et de nouvelles possibilités sont régulièrement découvertes notamment avec l'arrivée de nouvelles technologies comme HTML 5. Il est par exemple possible de rediriger vers un autre site pour du Hameçonnage ou encore de voler la session en récupérant les cookies.

I.4 Les menaces

(En anglais threat) signe, indice qui laisse prévoir un danger. Action ou évènement susceptible de se produire, de se transformer en agression contre un environnement ou des ressources et de porter préjudices à leur sécurité. On peut également classer les menaces en deux catégories selon qu'elles ne changent rien (menaces *passives*) ou qu'elles perturbent effectivement le réseau (menaces *actives*).

c 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.

d Les menaces actives :

Sont de nature à modifier l'état du réseau.

I.5 Mécanismes de sécurité :

I.5.1 Cryptographie, Signature électronique et Certificat

I.5.1.1 Cryptographie : [7]

Le mot cryptographie est un terme générique désignant l'ensemble de techniques permettant de chiffrer des messages. Chiffrer un message consiste à le transformer au moyen d'un algorithme mathématique afin de le rendre inintelligible, sauf pour celui qui possède le moyen (une clé) de le déchiffrer. L'encryptions des informations électroniques transitent par le réseau est utilisée pour assurer la confidentialité et l'authenticité des transactions. Le chiffrement se fait généralement à l'aide d'une clé de chiffrement, le déchiffrement nécessite quant à lui une clé de déchiffrement.

I.5.1.2 La signature électronique [8]

C'est un code digital (une réduction du document électronique à envoyer) qui, associé aux techniques de cryptage, garantit l'identité de la personne qui émet le message et assure la non-répudiation et l'intégrité de l'envoi.

I.5.1.3 Le certificat [9]

Document électronique (carte d'identité) émis par une autorité de certification. Il valide l'identité des interlocuteurs d'une transaction électronique, associe une identité à une clé publique de cryptage et fournit des informations de gestion complémentaires sur le certificat et le détenteur.

I.6 VPN [10]

I.6.1 Définition :

Un VPN est un tunnel (nous pouvons aussi parler de liaison virtuelle) sécurisé permettant la communication entre deux entités y compris au travers de réseaux peu sûrs comme peut l'être le réseau Internet. Cette technologie, de plus en plus utilisée dans les entreprises, permet de créer une liaison virtuelle entre deux réseaux physiques distants de manière transparente pour les utilisateurs concernés. Les données envoyées au travers de ces liaisons virtuelles sont chiffrées, ceci garantit aux utilisateurs d'un VPN qu'en cas d'interception malveillante les données soient illisibles

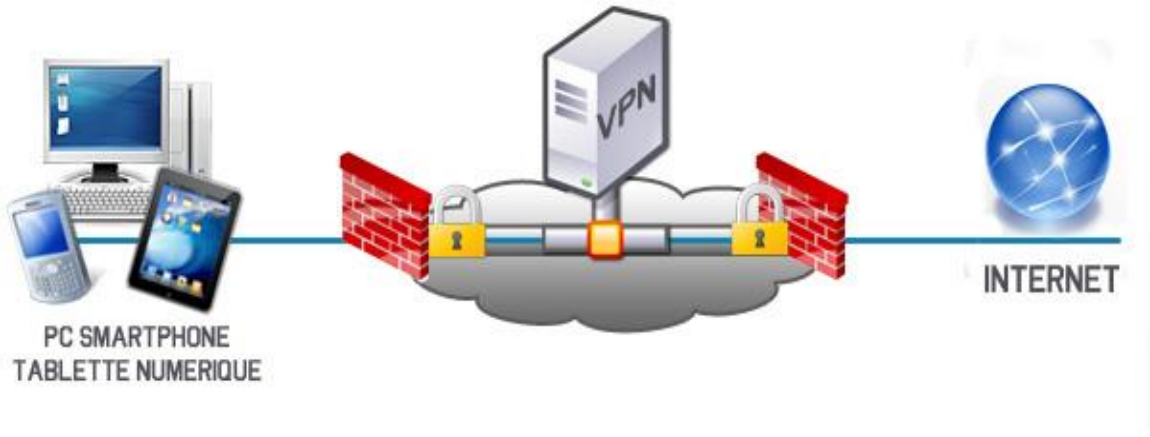

Figure I.5: VPN

I.6.2 Fonctionnement :

Un VPN repose sur un ou des protocoles, appelé protocoles de tunnelisation (ou tunneling), ce sont des protocoles permettant aux données passant entre deux réseaux physiques d'être sécurisées par des algorithmes de chiffrage. On utilise d'ailleurs le terme de « tunnel » pour mettre l'accent sur le fait qu'entre l'entrée et la sortie d'un VPN les données sont chiffrées et protégées. Lorsqu'un VPN est établi entre deux réseaux physiques, l'élément qui permet de chiffrer et de déchiffrer les données du coté client (ou utilisateur) est

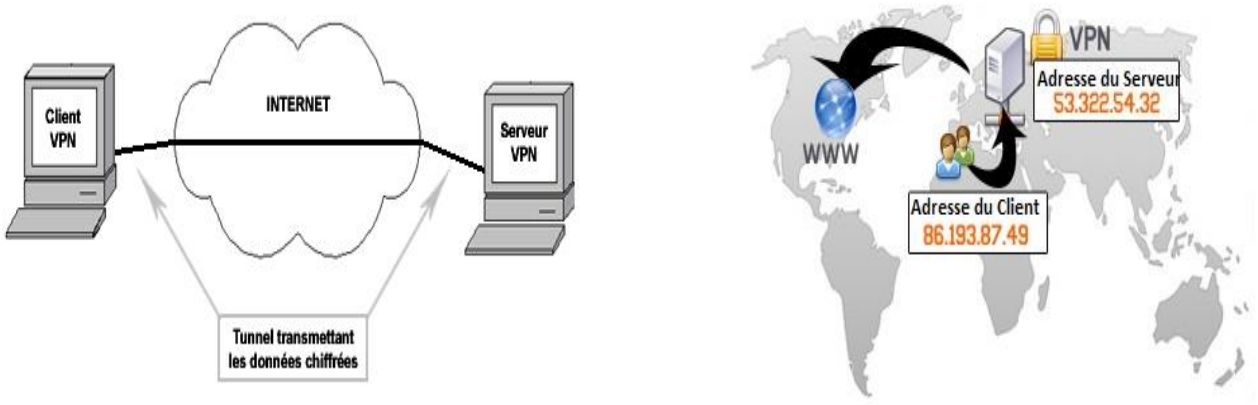

Figure I.6: Fonctionnement de VPN

nommé « Client VPN ». On appelle « Serveur VPN » l'élément qui chiffre et qui déchiffre les données du côté de l'organisation.

Dans les faits, nous établissons une connexion sécurisée avec le serveur qui nous propose le service VPN. Ce serveur VPN nous connecte sur Internet en masquant notre adresse IP par son adresse IP.

Une communication VPN peut donc être de client à serveur mais il est à prendre en considération qu'elle peut aussi se faire de serveur à serveur.

Il existe d'autres types d'utilisation des VPN :

L'intranet VPN: qui est utilisé pour relier deux intranets entre eux. Ce type de VPN est utile pour les entreprises possédant plusieurs sites distants. Le plus important avec ce type de VPN est de garantir la sécurité et l'intégrité des données.

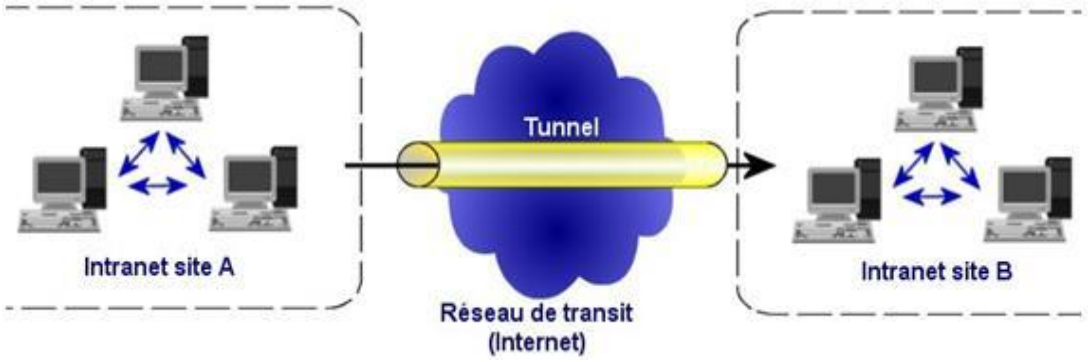

Figure I.7: Lintranet VPN

L'extranet VPN: là aussi utilisé par les entreprises car elles peuvent utiliser ce type de VPN pour communiquer avec ses clients. Dans les faits, elle ouvre son réseau local à ses clients ou à ses partenaires. Dans ce cadre, il est fondamental que l'administrateur du VPN puisse tracer les clients sur le réseau et gérer les droits de chacun sur celui-ci.

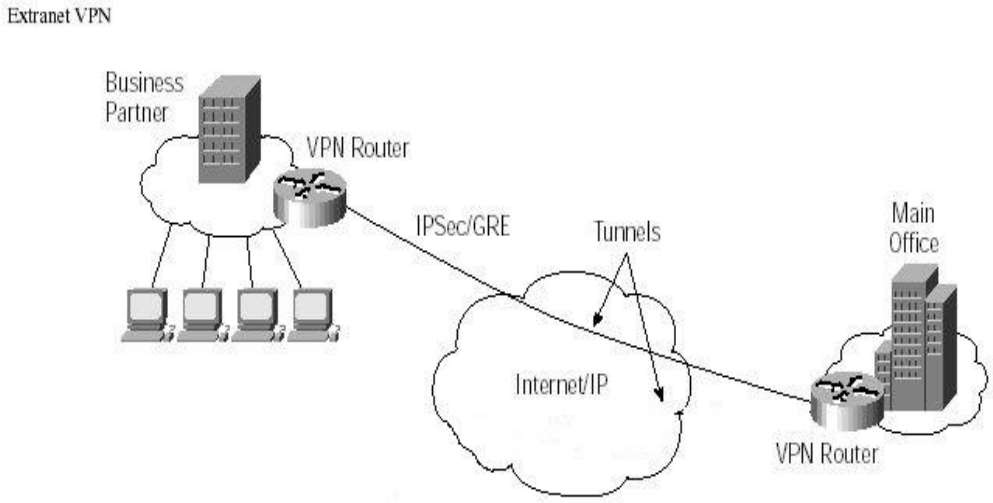

Figure I.8: L'extranet VPN

Intérêts : Lorsque nous utilisons un VPN, nous rendons notre connexion Internet privée, anonyme, protégée et celui-ci cache notre adresse IP sur Internet.

Un VPN laisse la possibilité de construire des réseaux overlay (ou réseaux superposés, réseau informatique bâti sur un autre réseau).

Un autre intérêt est le faible coût de l'accès à Internet, que ce soit à haut débit ou via une ligne téléphonique. C'est pour cela que les VPN sont de plus en plus répandus au sein des entreprises.

I.6.3 Les principaux protocoles de tunnelisation :

Concernant les principaux protocoles de tunnelisation :

L2F (Layer Two Forwarding) qui est un protocole de niveau 2 développé par Cisco, Northern Telecom et Shiva. A noter qu'il est désormais obsolète.

PPTP (Point-to-Point Tunneling Protocol) qui est aussi un protocole de niveau 2 développé par Microsoft, 3Com, Ascend, US Robotics et ECI Telematics.

L2TP (Layer Two Tunneling Protocol), protocole de niveau 2 s'appuyant sur PPP, et qui fait converger les fonctionnalités de PPTP et L2F.

GRE (Generic Routing Encapsulation), développé par Cisco, mais qui est souvent remplacé par L2TP.

IPSec , un protocole de niveau 3, issu des travaux de l' IETF(Internet Engineering Task Force, groupe participant à l'élaboration des standards Internet). Il permet de transporter des données chiffrées pour les réseaux IP.

SSL (Secure Sockets Layer), quant à lui, offre une très bonne solution de tunnelisation. L'avantage de cette solution est de permettre l'utilisation d'un navigateur Web comme client VPN : on peut accéder à ce type de VPN avec un navigateur web via « https ». Dans les faits, il permet aux utilisateurs de mettre en place une connexion sécurisée au réseau depuis n'importe quel navigateur Web.

I.7 Antivirus

Les antivirus 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 antivirus 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.), les données qui transitent sur les éventuels réseaux (dont internet), etc.

I.7.1 Pare-feu (firewall) [11]

Un pare-feu est un appareil de protection du réseau qui surveille le trafic entrant et sortant et décide d'autoriser ou de bloquer une partie de ce trafic en fonction d'un ensemble de règles de sécurité prédéfinies.

Les pare-feu constituent la première ligne de défense des réseaux depuis plus de 25 ans. Ils établissent une barrière entre les réseaux internes sécurisés et contrôlés qui sont dignes de confiance et les réseaux externes non fiables tels qu'Internet.

Un pare-feu peut être un appareil physique, un logiciel ou les deux.

I.7.2 Types de pare-feu

I.7.2.1 Pare-feu proxy

Apparu tôt, le pare-feu proxy sert de passerelle entre deux réseaux pour une application spécifique. Les serveurs proxy peuvent offrir des fonctionnalités supplémentaires, comme la mise en cache du contenu ou la protection, en empêchant toute connexion directe provenant de l'extérieur du réseau. Ils peuvent toutefois avoir un impact sur le débit et sur les applications prises en charge.

I.7.2.2 Pare-feu a inspection « stateful »

Désormais considéré comme un pare-feu « classique », le pare-feu à inspection « stateful » autorise ou bloque le trafic en fonction de l'état, du port et du protocole. Il surveille toute l'activité entre le début et la fin d'une connexion. Les décisions de filtrage sont prises en fonction de règles définies par l'administrateur ainsi que du contexte, ce qui implique d'utiliser les informations sur les connexions précédentes et les paquets de la connexion.

I.7.2.3 Pare-feu de gestion unifiée des risques liés à la sécurité

Un pare-feu de gestion unifiée des risques liés à la sécurité conjugue partiellement les fonctions d'un pare-feu à inspection « stateful » avec celles de prévention des intrusions et d'antivirus. Il peut également prendre en charge des services supplémentaires et intègre souvent la gestion du cloud. Ce type de pare-feu favorise la simplicité et la facilité d'utilisation.

I.7.2.4 Pare-feu de nouvelle génération (NGFW)

Les pare-feu ont évolué pour aller au-delà du simple filtrage de paquets et de l'inspection « stateful ». De nombreuses entreprises déploient des pare-feu de nouvelle génération pour bloquer les malwares modernes tels que les programmes malveillants avancés et les attaques au niveau de la couche application.

Selon la définition de Gartner, Inc., un pare-feu de nouvelle génération doit inclure :

- Les capacités d'un pare-feu standard telles que l'inspection « stateful »
- Des fonctions intégrées de prévention des intrusions
- La reconnaissance et le contrôle des applications pour détecter et bloquer celles qui présentent un risque
- Des possibilités de mise à niveau pour prendre en compte les futurs flux d'informations
- Des techniques pour faire face à l'évolution des malwares

Ces capacités s'imposent de plus en plus comme la norme pour l'entreprise, mais les pare-feu de nouvelle génération peuvent en faire encore plus.

I.7.2.5 Pare-feu de nouvelle génération axés sur les menaces

Ces pare-feu offrent toutes les fonctionnalités des pare-feu de nouvelle génération classiques, tout en proposant des fonctions avancées de détection et d'élimination des attaques. Avec un pare-feu de nouvelle génération axé sur les menaces, vous pouvez :

- Savoir quelles ressources présentent le plus de risques grâce à une connaissance complète du contexte
- Réagir rapidement aux attaques avec une automatisation intelligente des systèmes de protection qui définit des politiques et renforce vos défenses de manière dynamique
- Mieux détecter les activités furtives ou suspectes grâce à une mise en corrélation des événements au niveau du réseau et des terminaux
- Réduire fortement les délais entre la détection et le nettoyage avec des fonctions de sécurité rétrospective qui surveillent en continu l'activité et les comportements, même après l'inspection initiale
- Simplifier l'administration et réduire la complexité avec des politiques unifiées qui vous protègent pendant tout le cycle de l'attaque

I.8 Les systèmes de détection d'intrusions « réseau » (NIDS) : [12]

N-IDS (*Network Based Intrusion Detection System*), ils assurent la sécurité au niveau du réseau, il est utilisé pour analyser de manière passive le flux en transit sur le réseau et détecter les intrusions en temps réel. Un NIDS écoute donc tout le trafic réseau, puis l'analyse et généré des alertes si des paquets semblent dangereux, les NIDS étant les IDS les plus intéressants et les plus utiles du fait de l'omniprésence des réseaux dans notre vie quotidienne.

I.9 Pots de miels [13]

Les pots de miels sont des systèmes qui simulent plusieurs services réseaux pour leurrer des intrus en exposant des vulnérabilités connues délibérément.

Un attaquant pense que ces services vulnérables sont actifs et qu'il peut les utiliser pour s'introduire dans le réseau. Il s'y colle pendant un certain temps. Pendant ce temps l'administrateur enregistre les activités de l'intrus pour découvrir ses actions et ses techniques.

Une fois ces techniques sont connues. L'administrateur emploie ces informations plus tard pour durcir la sécurité sur les serveurs réels.

I.10 Conclusion

Nous avons constaté dans ce chapitre, que la sécurité informatique est un point primordial. Les administrateurs déploient des solutions de sécurité efficaces, capables de protéger le réseau de l'entreprise. Dans ce contexte, on a présenté les principales notions et leurs concepts, dont on a mentionné l'objectif de la sécurité, les menaces et les attaques, ainsi les différents méthodes et technique utilisés pour les sécuriser et les protéger, et dans le deuxième chapitre nous allons détailler l'un des mécanismes de sécurité d'une information qui est la cryptographie et présenter ces méthodes et mentionner certains de ses algorithmes.

Chapitre II :

La Cryptographie

Introduction :

Les transactions faites à travers le réseau peuvent être interceptées, d'autant plus que les lois ont du mal à se mettre en place sur Internet, il faut donc garantir la sécurité de ces informations, c'est la cryptographie qui s'en charge. Le but de la cryptographie moderne est de traiter plus généralement des problèmes de sécurité des communications et de fournir un certain nombre de services de sécurité. L'objectif de ce chapitre, est de présenter les méthodes de la cryptographie. Il intéressera le lecteur qui connait peu ou pas au domaine et qui souhaiterait comprendre le fonctionnement et le mécanisme mis en œuvre en cryptographie prévu. Si un tiers intercepte les données chiffrées, il lui sera difficile de les déchiffrer.

II.1 La cryptologie:

II.1.1 Définition de la cryptologie [14]

La cryptologie est la science des codes secrets, elle comporte le domaine de la cryptographie (qui est le domaine où on cherche à protéger le secret) et de la cryptanalyse (qui est le domaine où on cherche à retrouver le message d'origine sans connaitre exactement tout le procédé). Ces deux domaines étant fortement liés, il n'est pas rare de voir des cryptologies et des cryptanalyses travailler ensemble.

II.1.2 Définition de la cryptanalyse : [15]

La cryptanalyse est d'origine arabe et consiste à utiliser des connaissances mathématiques et linguistiques pour décoder les messages. Cette méthode ne marche que si on a suffisamment de messages cryptés à analyser. Par exemple, il est probable que le message *« les zèbres zigzaguent »* posera quelques problèmes à cause de la forte proportion de lettres 'z'. Avec un grand nombre de message, les proportions d'apparitions des lettres s'approcheront de plus en plus des proportions d'apparitions des lettres dans la langue française.

II.1.3 Définition de la cryptographie : [16]

Le secret a toujours tenu une place importante dans la vie de l'homme, des informations militaires au concept de vie privée. La cryptographie est l'art de cacher l'information, de la rendre accessible uniquement à un nombre restreint de personnes.

Pour se faire on transforme le message pour le rendre illisible mais de manière à pouvoir

réobtenir le message d'origine.

On parle souvent de chiffrage, de cryptage de l'information.

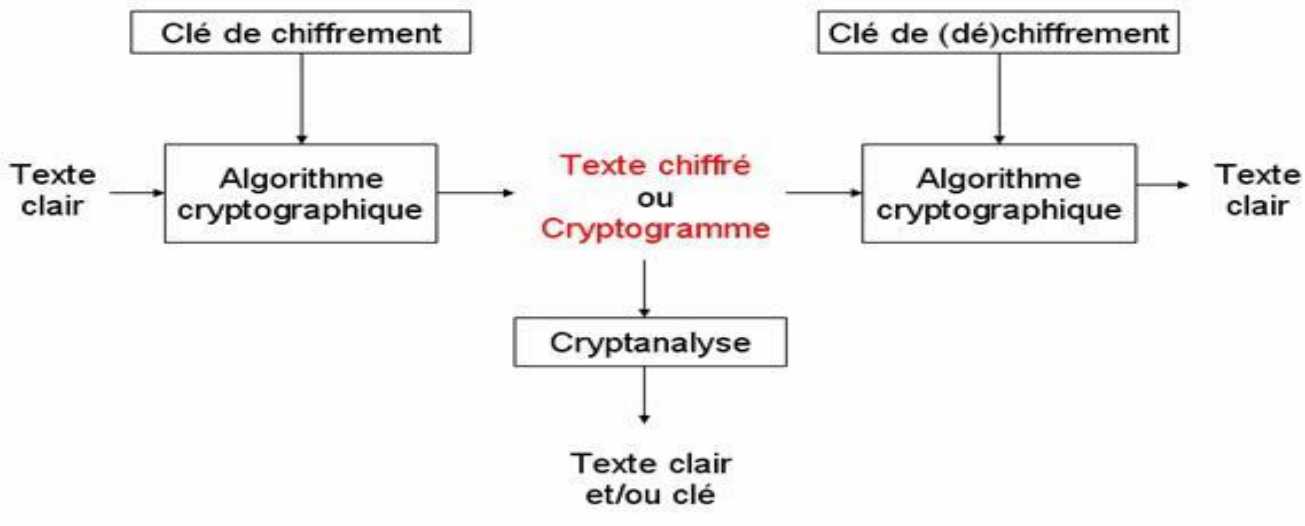

Figure II.1.: Protocole de chiffrement

II.1.4 Les services de la cryptographie [17]

La cryptographie est un outil de sécurité de l'information essentielle. Il fournit les quatre services les plus élémentaires de la sécurité de l'information

- **La confidentialité** : consiste à rendre l'information intelligible à d'autres personnes que les acteurs de la transaction.
- **L'intégrité** : vérifier l'intégrité des données consiste à déterminer si les données n'ont pas été altérées durant la communication.
- **L'authentification :** consiste à assurer l'identité d'un utilisateur, c.-à-d. de garantir à chacun des correspondants que son partenaire est bien celui qu'il croit être un contrôle d'accès peut permettre (par exemple par le moyen d'un mot de passe qui devra être crypté) l'accès à des ressources uniquement aux personnes autorisées.
- **La non répudiation :** de l'information est la garantie qu'aucun des correspondants ne pourra nier la transaction.

II.1.5 Terminologie de la Cryptographie

Nous utiliserons dans la cryptographie de différents termes que nous allons définir ici :

- ***Texte clair*** : le texte que l'on souhaite transmettre avant toute modification. Ça peut aussi bien être un texte dans une langue quelconque qu'une donnée autre (image, vidéo, son, ...).
- ***Un cryptogramme*** : une énigme basée sur un message chiffré.
- ***Texte chiffré*** : c'est le texte obtenu après avoir appliqué l'algorithme de chiffrement sur le texte clair.
- ***Chiffre*** : un chiffre est un algorithme permettant de substituer à chaque caractère du message clair un autre caractère.
- ***Clef*** : la clef est un paramètre permettant de calculer le message chiffré et/ou le message clair.

II.2 Les menaces : [18]

Les menaces accidentelles : Les menaces accidentelles ne supposent aucune préméditation. Dans cette catégorie, sont repris les bugs logiciels, les pannes matérielles, et autres défaillances "incontrôlables".

Les menaces intentionnelles : reposent sur l'action d'un tiers désirant s'introduire et relever des informations. Dans le cas d'une attaque passive, l'intrus va tenter de dérober les informations par audit, ce qui rend sa détection relativement difficile. En effet, cet audit ne modifie pas les fichiers, ni n'altère les systèmes. Dans le cas d'une attaque active, la détection est facilitée, mais il peut être déjà trop tard lorsque celle-ci a lieu. Ici, l'intrus aura volontairement modifié les fichiers ou le système en place pour s'en emparer.

Les menaces actives appartiennent principalement à quatre catégories :

- **Interruption :** problème lié à la disponibilité des données
- **Interception :** problème lié à la confidentialité des données
- **Modification :** problème lié à l'intégrité des données
- **Fabrication :** problème lié à l'authenticité des données

II.3 Les niveaux d'attaques [19]

L'intrus peut effectuer quatre niveaux d'attaques, l'attaque est une tentative de cryptanalyse.

- ***L'attaque par cryptogramme*** (par message chiffré seulement) : ou la cryptanalyse ne connait qu'un ensemble de message chiffrés, il peut soit retrouver seulement les messages en clair, soit retrouver la clef. En pratique, il est très souvent possible de deviner certaines propriétés du message en clair (format ASCII, présence d'un mot particulier, …), ce qui permet de valider ou non le décryptement.
- ***L'attaque à message en clair connu*** : ou la cryptanalyse connait non seulement les messages chiffrés mais aussi les messages en clair correspondants, son but est alors de retrouver la clef. Du fait de la présence, dans la plupart des messages chiffrés, de parties connue (en-têtes de paquets, champs communs à tous les fichiers d'un type donné,).
- ***L'attaque à message en clair choisi*** : ou la cryptanalyse peut, de plus choisir des messages en clair à chiffrer et donc utiliser des messages apportant plus d'informations sur la clef. Si la cryptanalyse peut de plus adapter ses choix en fonction des messages chiffrés précédents, on parle d'**attaque adaptative**.
- ***L'attaque à message chiffré choisi*** : qui l'inverse de la précédente, la cryptanalyse peut choisir des messages chiffrés pour lesquels il connaitra le message en clair correspondant ; sa tâche est alors de retrouver la clef. Ce type d'attaques est principalement utilisé contre les systèmes à clef publique, pour retrouver la clef privée

II.4 Les méthodes de cryptographie : [20]

Les méthodes de cryptographie ce compose de deux grandes parties : La cryptographie symétrique et la cryptographie asymétrique à base de clés.

Avant de les aborder, nous allons d'abord définir la notion de **clé** qui nous sera utile tout au long de ce chapitre :

Une clé : Paramètre constitué d'une séquence de symbole et utilisé par un algorithme cryptographique, pour transformer, valider, authentifier, chiffrer ou déchiffrer des données.

On distingue généralement deux types de clés :

Les clés symétriques: il s'agit de clés utilisées pour le chiffrement ainsi que pour le déchiffrement. On parle alors de chiffrement symétrique ou de chiffrement à clé secrète.
Les clés asymétriques: il s'agit de clés utilisées dans le cas du chiffrement asymétrique (aussi

appelé chiffrement à clé publique). Dans ce cas, une clé différente est utilisée pour le chiffrement et pour le déchiffrement.

II.4.1 La cryptographie symétrique :

La cryptographie à clé symétrique a très longtemps été utilisée pour le chiffrement de messages confidentiels. Son usage a été progressivement réduit depuis l'apparition de la cryptographie à clé publique (cryptographie asymétrique) même si les deux techniques sont encore parfois utilisées conjointement. Dans les algorithmes de chiffrement à clé symétrique ou clé secrète, c'est la même clé qui sert à la fois à chiffrer et à déchiffrer un message. C'est exactement le même principe qu'une clé de porte : c'est la même qui sert à ouvrir et à fermer une serrure.

Les algorithmes symétriques sont de deux types :

- Les algorithmes de chiffrement en continu, qui agissent sur le message en clair un bit à la fois ;
- Les algorithmes de chiffrement par bloc, qui opèrent sur le message en clair par groupes de bits appelés blocs.

II.4.2 Caractéristiques :

– Les clés sont identiques : KE = KD = K.

– La clé doit rester secrète.

– Les algorithmes les plus répandus sont le DES, AES, 3DES, Blowfish, IDEA, …

– Au niveau de la génération des clés, elle est choisie aléatoirement dans l'espace des clés.

– Ces algorithmes sont basés sur des opérations de transposition et de substitution des bits du texte clair en fonction de la clé.

– La taille des clés est souvent de l'ordre de 128 bits. Le DES en utilise 56, mais l'AES peut aller jusque 256.

– L'avantage principal de ce mode de chiffrement est sa rapidité

– Le principal désavantage réside dans la distribution des clés : pour une meilleure sécurité, on préfèrera l'échange manuel.

Malheureusement, pour de grands systèmes, le nombre de clés peut devenir conséquent. C'est pourquoi on utilisera souvent des échanges sécurisés pour transmettre les clés. En effet, pour un système à N utilisateurs, il y aura N. (N − 1)/2 paires de clés.

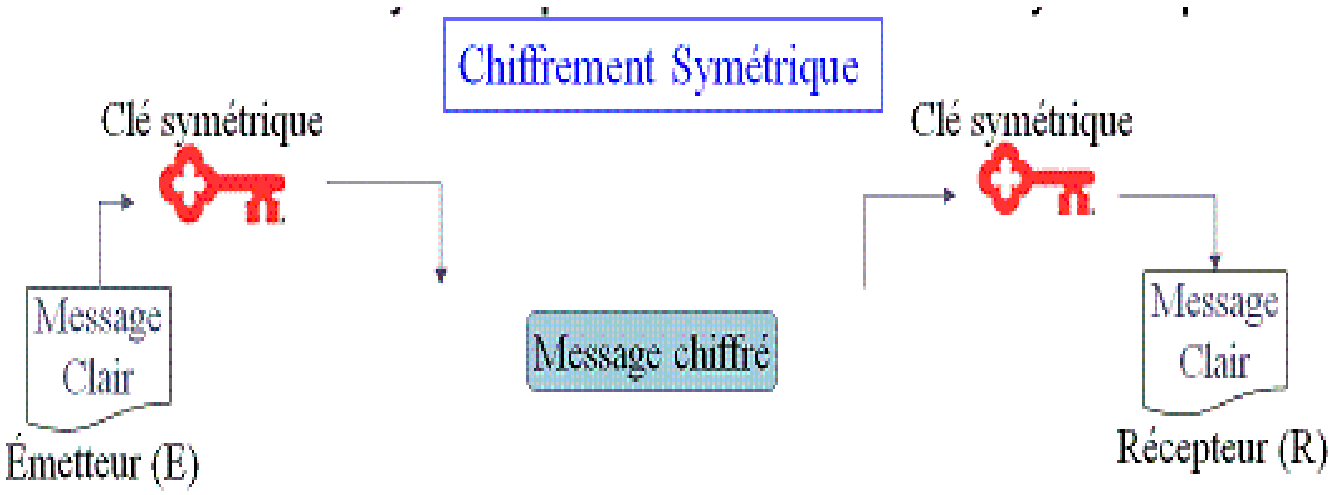

Figure II.2: Chiffrement symétrique

II.5 Exemple d'algorithmes symétriques

II.5.1 DES : [21]

Le 15 mai 1973 le **NBS** (*National Bureau of Standards*, aujourd'hui appelé *NIST - National Institute of Standards and Technology*) a lancé un appel dans le *Federal Register* (l'équivalent aux Etats-Unix du *Journal Officiel* en France) pour la création d'un algorithme de chiffrement répondant aux critères suivants :

- Posséder un haut niveau de sécurité lié à une clé de petite taille servant au chiffrement et au déchiffrement
- Être compréhensible
- Ne pas dépendre de la confidentialité de l'algorithme
- Être adaptable et économique
- Être efficace et exportable

Fin 1974, IBM propose « Lucifer », qui, grâce à la NSA (National Security Agency), est modifié le 23 novembre 1976 pour donner le **DES** (*Data Encryption Standard*). Le DES a finalement été approuvé en 1978 par le NBS. Le DES fut normalisé par l'*ANSI* (*American National Standard Institute*) sous le nom de *ANSI X3.92*, plus connu sous la dénomination *DEA* (*Data Encryption Algorithm*).

II.5.2 Principe du DES

Il s'agit d'un système de chiffrement symétrique par blocs de 64 bits, dont 8 bits (un octet) servent de test de parité (pour vérifier l'intégrité de la clé). Chaque bit de parité de la clé (1 tous les 8 bits) sert à tester un des octets de la clé par parité impaire, c'est-à-dire que chacun des bits de parité est ajusté de façon à avoir un nombre impair de '1' dans l'octet à qui il appartient. La clé possède donc une longueur « utile » de 56 bits, ce qui signifie que seuls 56 bits servent réellement dans l'algorithme.

L'algorithme consiste à effectuer des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé, en faisant en sorte que les opérations puissent se faire dans les deux sens (pour le déchiffrement). La combinaison entre substitutions et permutations est appelée code produit.

La clé est codée sur 64 bits et formée de 16 blocs de 4 bits, généralement notés *k1* à *k16*. Etant donné que « seuls » 56 bits servent effectivement à chiffrer, il peut exister 256 (soit 7.2*1016) clés différentes !

II.5.3 L'algorithme du DES

Les grandes lignes de l'algorithme sont les suivantes :

- Fractionnement du texte en blocs de 64 bits (8 octets).
- Permutation initiale des blocs.
- Découpage des blocs en deux parties: gauche et droite, nommées G et D.
- Etapes de permutation et de substitution répétées 16 fois (appelées rondes).
- Recollement des parties gauche et droite puis permutation initiale inverse.

Inconvénient :

Aujourd'hui, le D.E.S. est fortement menacé par les puissances de calcul des ordinateurs. Il n'est en effet pas impossible de balayer la plupart des clés pour casser le code. Un nouveau système, le A.E.S. (Advanced Encryption Standard) est prévu pour le remplacer.

II.5.4 AES: [22]

Avec le temps, et les progrès de l'informatique, les 256 clés possibles du DES n'ont plus représenté une barrière infranchissable. Il est désormais possible, même avec des moyens modestes, de percer les messages chiffrés par DES en un temps raisonnable. En janvier 1997, le NIST (National Institute of Standards and Technologies) des Etats-Unis lance un appel d'offres pour élaborer l'AES, Advanced Encryption System. Le cahier des charges comportait les points suivants :

- Évidemment, une grande sécurité.
- Une large portabilité : l'algorithme devant remplacer le DES, il est destiné à servir aussi bien dans les cartes à puces, aux processeurs 8 bits peu puissants, que dans des processeurs spécialisés pour chiffrer des milliers de télécommunications à la volée.
- La rapidité.
- Une lecture facile de l'algorithme, puisqu'il est destiné à être rendu public.
- Techniquement, le chiffrement doit se faire par blocs de 128 bits, les clés comportant 128,192 ou 256 bits.

Au 15 juin 1998, date de la fin des candidatures, 21 projets ont été déposés. Certains sont l'œuvre d'entreprises (IBM,..), d'autres regroupent des universitaires (CNRS,...), les derniers sont écrits par à peine quelques personnes. Pendant deux ans, les algorithmes ont été évalués par des experts, avec forum de discussion sur Internet, et organisation de conférences. Le 2 octobre 2000, le NIST donne sa réponse : c'est le Rijndael qui est choisi, un algorithme mis au point par 2 belges, Joan Daemen et Vincent Rijmen. Depuis, le Rijndael, devenu AES, a été largement déployé et a remplacé progressivement le DES.

II.5.4.1 Principe de fonctionnement de l'AES

Le Rijndael procède par blocs de 128 bits, avec une clé de 128 bits également. Chaque bloc subit une séquence de 5 transformations répétées 10 fois :

- Addition de la clé secrète (par un ou exclusif).
- Transformation non linéaire d'octets : les 128 bits sont répartis en 16 blocs de 8 bits (8 bits=un octet), eux-mêmes dispatchés dans un tableau 4×4. Chaque octet est transformé

par une fonction non linéaire S. S peut être simplement vu comme une substitution sur les entiers compris entre 1 et 256. En particulier, elle peut être implantée sur ordinateur par un simple tableau.

- Décalage de lignes : les 3 dernières lignes sont décalées cycliquement vers la gauche : la 2ème ligne est décalée d'une colonne, la 3ème ligne de 2 colonnes, et la 4ème ligne de 3 colonnes.
- Brouillage des colonnes : Chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne (ce qui revient à multiplier la matrice 4×4 par une autre matrice 4×4). Les calculs sur les octets de 8 bits sont réalisés dans le corps à 2^8 éléments.
- Addition de la clé de tour : A chaque tour, une clé de tour est générée à partir de la clé secrète par un sous-algorithme (dit de cadencement). Cette clé de tour est ajoutée par un ou exclusif au dernier bloc obtenu.

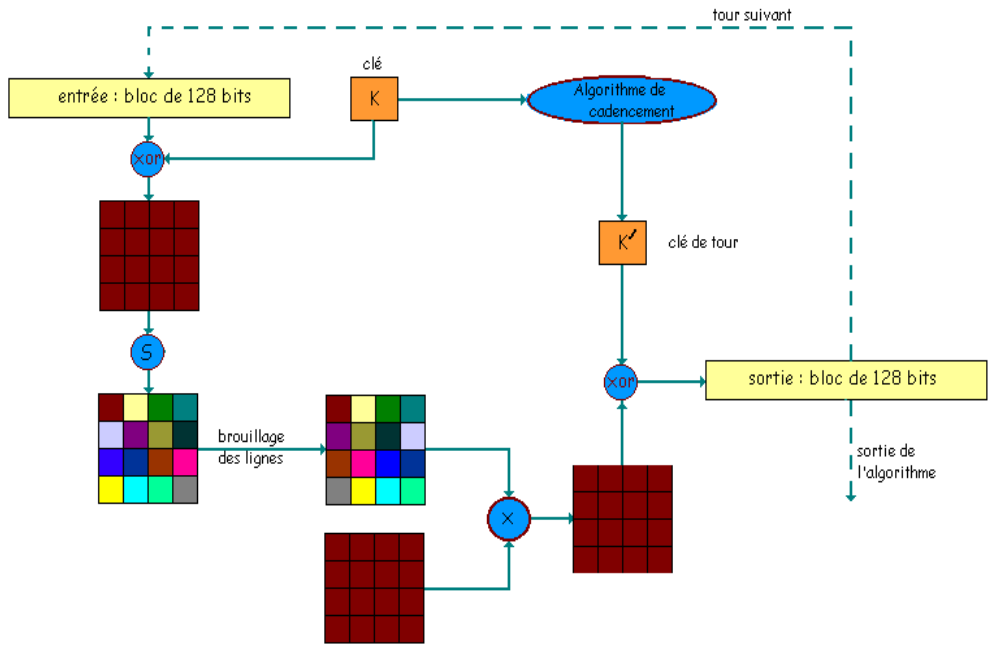

Figure II.3 : Diagramme de fonctionnement AES

En conclusion, l'AES est plus sûr que le DES car il présente, entre autres, une plus grande résistance aux attaques par dictionnaires de clés. Les autres attaques ne sont pas appliquables dans son cas.

II.6 Les algorithmes à clef publique ou algorithmes asymétriques :

La cryptographie asymétrique est une méthode utilisée pour transmettre et échanger des messages de façon sécurisée en s'assurant de respecter les principes suivants :

- Authentification de l'émetteur
- Garantie d'intégrité
- Garantie de confidentialité

Cette technique repose sur le principe de « paire de clés » (ou bi-clés) composée d'une clé dite « privée » conservée totalement secrète et ne doit être communiquée à personne et d'une clé dite « publique » qui, comme son nom l'indique peut est transmise à tous sans aucune restriction. Les clés dites asymétriques sont des clés de chiffrement. Le chiffrement étant le nom général donné aux techniques mathématiques de codage ou de décodage des données.

Les principes généraux de la cryptographie à clé publique sont les suivants :

- Un message codé avec une clé privée ne peut être décodé que par la clé publique associée.
- Un message codé avec une clé publique ne peut être décodé que par la clé privée associée.
- Une clé publique donnée ne peut être associée qu'à une seule clé privée.
- Plusieurs clés privées différentes ne peuvent pas avoir la même clé publique comme clé complémentaire.
- Une clé privée donnée ne peut être associée qu'à une seule clé publique.
- Plusieurs clés publiques différentes ne peuvent pas avoir la même clé privée comme clé complémentaire.

Les principaux algorithmes à clé publique sont :

* protocole de Deffie-Hellman

*chiffrement de RSA

*chiffrement de Rabin

*chiffrement Elgamal

*chiffrement DSA

II.7 Exemple d'algorithmes Asymétriques

II.7.1 RSA :[23]

II.7.2 Définition :

L'algorithme RSA (du nom de ses inventeurs Ron Rivest, Adi Shamir et Len Aldeman, qui ont imaginé le principe en 1978) est utilisé pour la cryptographie à clé publique et est basé sur le fait qu'il est facile de multiplier deux grands nombres premiers mais difficile de factoriser le produit. C'est l'exemple le plus courant de cryptographie asymétrique, toujours considéré comme sûr, avec la technologie actuelle, pour des clés suffisamment grosses (1024, 2048 voire 4096 bits).

Usages :

RSA, du nom de ces inventeurs, est un algorithme de chiffrement appartenant à la grande famille "Cryptographie asymétrique".

RSA peut être utilisé pour assurer :

- La confidentialité : seul le propriétaire de la clé privée pourra lire le message chiffré avec la clé publique correspondante.
- La non-altération et la non-répudiation : seul le propriétaire de la clé privée peut signer un message (avec la clé privée). Une signature déchiffrée avec la clé publique prouvera donc l'authenticité du message.

Sa robustesse réside dans la difficulté à factoriser un grand nombre.

II.7.3 Génération des clés :

- Soient deux grands nombres premiers « aléatoirement » choisis : p et q.
- Notons : $n = p*q$ et $\varphi = (p-1)*(q-1)$
- Soient d'un grand entier « aléatoirement » choisi, premier avec φ. Et e l'inverse de d modulo φ.
- La clé publique de chiffrement est le couple (n,e), la clé privée de déchiffrement le couple (n,d).

II.7.4 Chiffrement

- Avant d'être chiffré, le message original doit être décomposé en une série d'entiers *M* de valeurs comprises entre 0 et *n-1*.
- Pour chaque entier *M* il faut calculer *C≡M^e [n]*.
- Le message chiffré est constitué de la succession des entiers *C*.

II.7.5 Déchiffrement

- Conformément à la manière dont il a été chiffré, le message reçu doit être composé d'une succession d'entiers *C* de valeurs comprises entre 0 et *n-1*.
- Pour chaque entier *C* il faut calculer *M≡C^d [n]*.
- Le message original peut alors être reconstitué à partir de la série d'entiers *M*.

II.7.6 Fiabilité :

La sécurité de l'algorithme RSA repose sur la difficulté à factoriser *n*. Pour décrypter le message, il est nécessaire de trouver *d* connaissant *e*, ce qui nécessite de recalculer *φ*, et donc de connaître *p* et *q*, les deux facteurs premiers de *n*.

Or, la factorisation d'un entier (de très grande taille) en facteurs premiers est extrêmement difficile, cette opération nécessitant une capacité de calcul très importante. Pour exemple : en 2010, l'INRIA et ses partenaires ont réussi à factoriser une clé de 768 bits. Il leur a fallu deux ans et demi de recherche, et plus de 10^20 calculs. C'est à ce jour le meilleur résultat connu de factorisation.

Afin de se prémunir contre les puissances de calculs grandissantes, il est régulièrement recommandé d'utiliser des tailles de clés de plus en plus grandes (actuellement de 2048 bits).

II.8 Les avantages et inconvénients de la cryptographie à clé publique comparé à la cryptographie à clé privée [24]

Le premier avantage de la cryptographie à clé publique est d'améliorer la sécurité elle permet d'échanger des messages de manière sécurisée sans aucun dispositif de sécurité. L'expéditeur

et le destinataire n'ont plus besoin de partager des clés secrètes via une voie de transmission sécurisée. Les communications impliquent uniquement l'utilisation de clés publiques et plus aucune clé privée n'est transmise ou partagée.

Avec un système à clé secrète, au contraire, il existe toujours le risque de voir la clé récupérée par une personne tierce quand elle est transmise d'un correspondant à l'autre. Toute personne interceptant la clé lors d'un transfert peut ensuite lire, modifier et falsifier toutes les informations cryptées ou authentifiées avec cette clé. De la norme de cryptage de données DES au code secret de ***Jules César***, la distribution des clés reste le problème majeur du cryptage conventionnel. (Autrement dit, comment faire parvenir la clé à son destinataire sans qu'aucune personne ne l'intercepte) les moyens à déployer pour garantir la distribution sécurisée des clés correspondants sont très onéreux, ce qui constitue un inconvénient supplémentaire.

Le cryptage à clé publique représente une révolution technologique qui offert à tout citoyen la possibilité d'utiliser une cryptographie robuste. En effet, la cryptographie conventionnelle était auparavant la seule méthode pour transmettre des informations secrètes. Les couts d'institutions disposants de moyens suffisants, telles que gouvernements et banque.

Un autre avantage majeur des systèmes à clé publique est qu'ils permettent l'authentification des messages par signature électronique, ce qui peut aussi servir devant un juge, par exemple.

L'inconvénient des systèmes à clé publique est leur vitesse contrairement aux méthodes à clé secrète qui sont plus rapide. Ils sont particulièrement adaptés à la transmission de grandes quantités de données. Mais les deux méthodes peuvent être combinées de manière à obtenir le meilleur de leurs systèmes. Pour le cryptage, la meilleure solution est d'utiliser un système à clé publique pour crypter une clé secrète qui sera alors utilisée pour crypter fichiers et message**.**

II.9 La signature électronique : [25]

II.9.1 Définition :

La signature électronique (parfois appelée signature numérique) est un mécanisme permettant de garantir l'intégrité d'un document électronique et d'en authentifier l'auteur, par analogie avec la signature manuscrite d'un document papier.

Elle se différencie de la signature écrite par le fait qu'elle n'est pas visuelle, mais correspond à une suite de caractères.

II.9.2 Fonctionnement de la signature numérique : [26]

Les concepts de signature numérique sont principalement basés sur la cryptographie asymétrique. Cette technique permet de chiffrer avec un mot de passe et de déchiffrer avec un autre, les deux étant indépendants.

Par exemple, imaginons que Bob souhaite envoyer des messages secrets à Alice. Ils vont pour cela utiliser la cryptographie asymétrique.

Alice génère tout d'abord un couple de clés. Une clé privée (en rouge) et une clé publique (en vert). Ces clés ont des propriétés particulières vis à vis des algorithmes utilisés. En effet, un message chiffré avec une clé ne peut être déchiffré qu'avec l'autre clé. Il s'agit de fonction à sens unique.

Alice transmet ensuite la clé publique (en vert) à Bob.

Grâce à cette clé, Bob peut chiffrer un texte et l'envoyer à Alice.

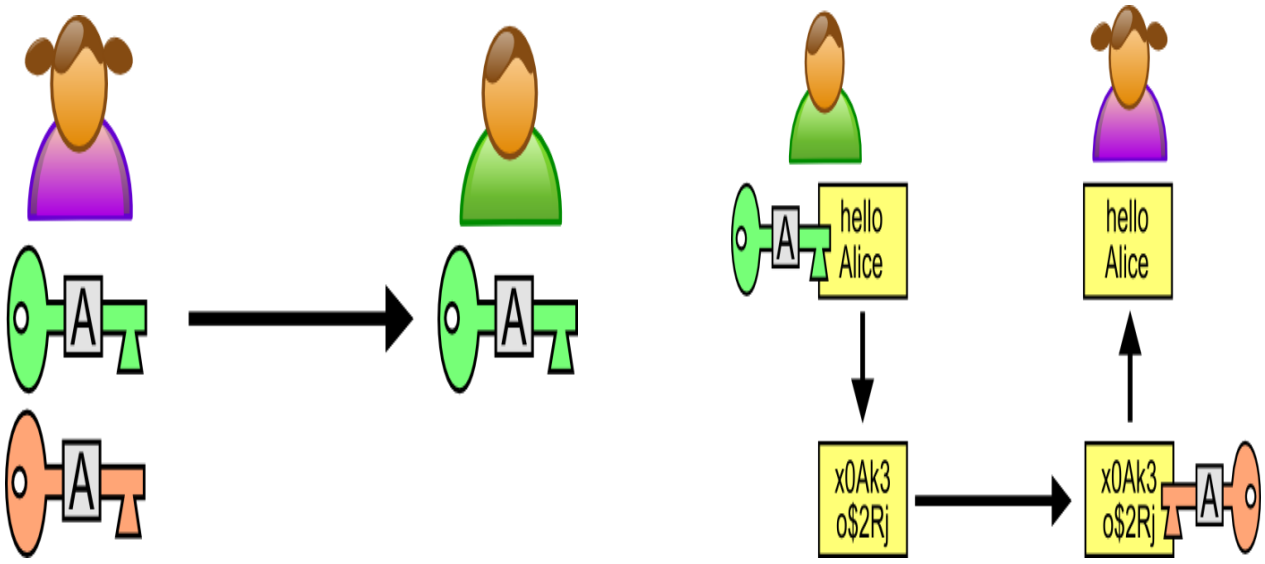

Figure II.4 : Génération des clés & crypter le message

En utilisant la clé publique d'Alice, Bob est certain de deux choses :

- Personne ne peut lire le message, puisqu'il est crypté
- Seule Alice peut déchiffrer le message, car elle est la seule à posséder la clé privée.

Nous venons de répondre au besoin de confidentialité des données. Mais la cryptographie asymétrique peut être utilisée d'une autre façon. En effet, on peut

également utiliser la clé privée pour chiffrer, la clé publique servant alors à déchiffrer.

Le message ainsi chiffré est lisible par toute personne disposant de la clé publique. Ceci n'est pas très utile si l'on cherche la confidentialité. En revanche, une seule personne est susceptible d'avoir chiffré ce message : Alice. Ainsi, si l'on peut déchiffrer un message avec la clé publique d'Alice, c'est forcément la personne à avoir chiffré ce message.

II.10 Fonction de hachage :

Une fonction de hachage est un procédé à sens unique permettant d'obtenir une suite d'octets (une empreinte) caractérisant un ensemble de données. Pour tout ensemble de données de départ, l'empreinte obtenue est toujours la même.

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

- Ce sont des fonctions unidirectionnelles : A partir de H(M) il est impossible d retrouver M.
- Ce sont des fonctions sans collisions :
- A partir de H(M) et M il est impossible de trouver M≠M tel que H(M)=H(M).

Nous pouvons donc utiliser ces fonctions pour nous assurer de l'intégrité d'un document.

Les deux algorithmes les plus utilisées sont MD5 et SHA. A noter que MD5 n'est plus considéré comme sûr par les spécialistes. En effet, une équipe chinoise aurait réussi à trouver une collision complète, c'est à dire deux jeux de données donnant la même empreinte, sans utiliser de méthode de force brute.

II.11 Singer un document :

La signature d'un document utilise à la fois la cryptographie asymétrique et les fonctions de hachage. C'est en effet par l'association de ces deux techniques que nous pouvons obtenir les 5 caractéristiques d'une signature (authentique, infalsifiable, non réutilisable, inaltérable,

irrévocable).

Imaginons que Alice souhaite envoyer un document signé à Bob.

- Tout d'abord, elle génère l'empreinte du document au moyen d'une fonction de hachage.
- Puis, elle crypte cette empreinte avec sa clé privée.

Figure II.5: Signer un document

- Elle obtient ainsi la signature de son document. Elle envoie donc ces deux éléments à Bob

Figure II.6: Ennoie pubkey + document signé

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

- Ensuite, Bob génère l'empreinte du document qu'il a reçu, en utilisant la même fonction de hachage qu'Alice (On supposera qu'ils suivent un protocole établi au préalable).
- Puis, il compare l'empreinte générée et celle issue de la signature.

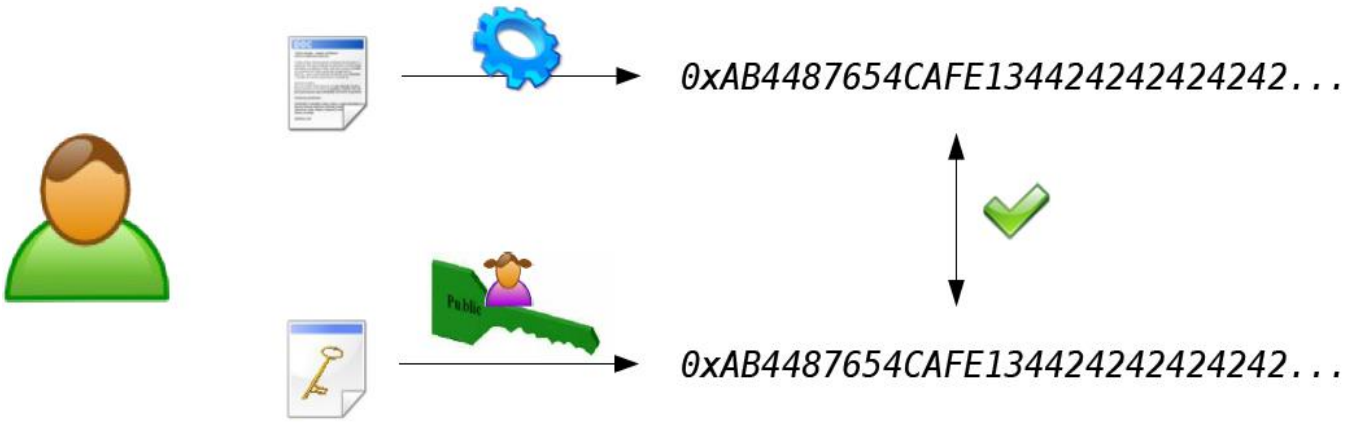

Figure II.8: Vérification de la signature

- Si les deux empreintes sont identiques, la signature est validée. Nous sommes donc sûr que :
    - C'est Alice qui a envoyé le document,
    - Le document n'a pas été modifié depuis qu'Alice l'a signé.
- Dans le cas contraire, cela peut signifier que :
    - Le document a été modifié depuis sa signature par Alice,
    - Ce n'est pas ce document qu'Alice a signé.

Un mécanisme de signature numérique doit présenter les propriétés suivantes :

- Il doit permettre au lecteur d'un document d'identifier la personne ou l'organisme qui a apposé sa signature (propriété d'identification).
- Il doit garantir que le document n'a pas été altéré entre l'instant où l'auteur l'a signé et le moment où le lecteur le consulte (propriété d'intégrité).

Pour cela, les conditions suivantes doivent être réunies :

- **Authentique** : l'identité du signataire doit pouvoir être retrouvée de manière certaine.

- **Infalsifiable** : la signature ne peut pas être falsifiée. Quelqu'un ne peut se faire passer pour un autre.
- **Non réutilisable** : la signature n'est pas réutilisable. Elle fait partie du document signé et ne peut être déplacée sur un autre document.
- **Inaltérable** : un document signé est inaltérable. Une fois qu'il est signé, on ne peut plus le modifier.
- **Irrévocable** : la personne qui a signé ne peut le nier.

II.12 Conclusion

Dans ce chapitre nous avons présenté une introduction générale sur la cryptographie, on a retrouvé deux grandes classes des méthodes de chiffrement, la cryptographie symétrique à clé secrète et le cryptographie asymétrique à clé publique, on a aussi décrit quelques algorithmes de chiffrements les plus utilisés, mais bien sûr il en existe beaucoup d'autres.
Et dans le chapitre suivant nous allons nous focaliser plus précisément sur l'algorithme de Signature DSA (Digital signature Algorithm) qui est l'objet de notre travail.

Chapitre III

L'Algorithme DSA (Digital Signature Algorithm)

Introduction

Après avoir présenté les notions générales de la cryptographie, à présent nous nous intéresserons à l'Algorithme de Signature Digitale DSA où nous soulignerons le fonctionnement et les motivations qui nous ont poussé à choisir cet algorithme.

III.1 Présentation: [27]

DSA signifie Digital Signature Algorithm (Algorithme de Signature Digitale). Il s'agit d'un algorithme inventé en 1991 aux Etats-Unis par le **National Institute of Standards and Technology** (NIST) et adopté par le **Federal Information Processing Standard** (FIPS) en 1993. L'algorithme DSA fut utilisé en premier lieu pour signer électroniquement des données, mais on l'utilise désormais à la fois comme algorithme de signature et de chiffrement dans les certificats SSL.

La méthode DSA est essentiellement utilisée par les services publics américains, car il s'agit d'une méthode approuvée par les agences fédérales, et qui correspond donc aux critères de sécurité requis pour les services publics.
Il est tout à fait possible de **combiner les algorithmes RSA et DSA** sur un même serveur (c'est le cas notamment des serveurs sous Apache) pour garantir un niveau de sécurité optimal.

DSA est considéré comme l'un des algorithmes de signature numérique les plus utilisés aujourd'hui.

III.2 L'algorithme DSA

III.2.1 Génération des clés :

Chaque signataire dispose d'une paire de clés: une clé privée *x* et une clé publique *y* qui sont mathématiquement liés les uns aux autres. La clé privée doit être utilisé que pour une période de temps fixe (par exemple, la crypto période clé privée), dans lequel les signatures numériques

peuvent être générées; la clé publique peut continuer à être utilisé aussi longtemps que les signatures numériques qui ont été générés en utilisant la nécessité clé privée associée à vérifier (c.-à-la clé publique peut continuer à être utilisé au-delà de la crypto période de la clé privée associée).

Leur sécurité repose sur la difficulté du problème du logarithme discret dans un groupe fini.

- Choisir des longueurs L et N avec L divisible par 64. Ces longueurs définissent directement le niveau de sécurité de la clef. NIST 800-57 recommande de choisir $L=3072$ et $N=256$ pour une sécurité équivalente à 128 bit.
- Choisir un nombre premier P de longueur L.
- Choisir un nombre premier q de longueur N , de telle façon que $q-1 = qz$, avec z un entier
- Choisir h, avec $1 < h < p-1$ de manière que $g = h^z \bmod p > 1$
- Générer aléatoirement un x , avec $0 < x < q$
- Calculer $y = g^x \bmod p$
- La clé publique est (p, q, g, y). La clé privée est x .

III.2.2 Signature :

- Choisir un nombre aléatoire k tel que $1 < k < q$
- Calculer $r = (g^k \bmod p) \bmod q$
- Si $r = 0$ recommencer avec un autre k .
- Calculer $s = (H(m) + r.x) k^{-1} \bmod q$, où $H(m)$ est le résultat d'un hachage .cryptographique, par exemple avec SHA-256, sur le message m.
- Si $s = 0$ recommencer avec un autre k.
- La signature est (r,s).

III.2.3 Vérification :

La vérification de signature peut être effectuée en utilisant la clé publique du signataire.

Les étapes de la vérification :

- Rejeter la signature si $0 < r < q$ ou $0 < s < q$ n'est pas vérifié
- Calculer $w = s^{-1} \bmod q$

- Calculer ***u1 = H(m) . w mod q***
- Calculer ***u2 = r . w mod q***
- Calculer ***(v = g^{u1}. y^{u2} mod p) mod q***
- La signature est valide si ***v = r***

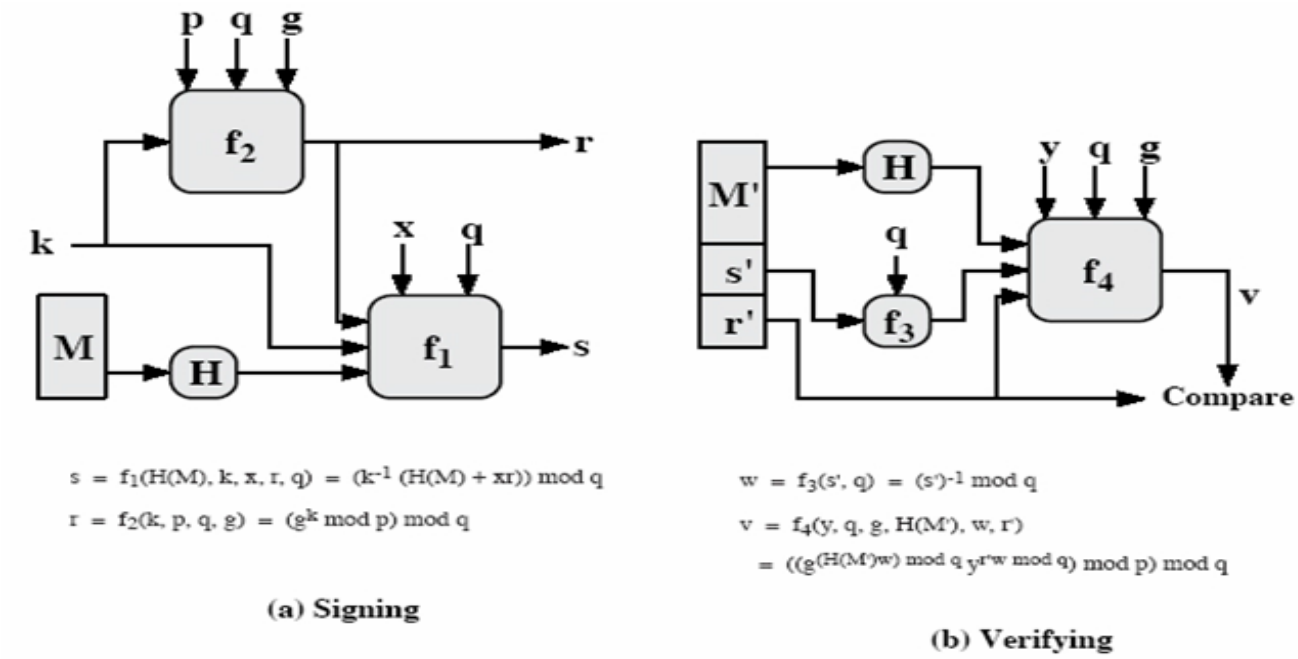

Figure III.3: Illustration des procédés de signature et de vérification de DSA

III.2.4 Validité de l'algorithme :

Ce principe de signature est correct dans le sens où le vérificateur acceptera toujours des signatures authentiques. Ceci peut être démontré comme suit avec un exemple pratique :

A partir de ***p-1 = qz*** et ***$g = h^z$*** mod p découle :

$g^q \equiv h^{qz} \equiv h^{p-1} \equiv 1$ mod p selon le petit théorème de Fermat. Puisque ***$g > 1$*** et ***q*** est premier, il s'ensuit que g a un ordre égal a q.

Celui qui procède a la signature obtient :

$s = k^{-1}$ (H(m) +xr) mod q

Ainsi

$S \equiv H(m)\, s^{-1} + x.r.s^{-1}$

$\equiv$ ***H (m) w + x.r.w mod q***

Comme g est d'ordre q on a :

$$g^k \equiv g^{H(m)w} g^{xrw}$$

$$g^{H(m)w} y^{r.w}$$

$$g^{u1} y^{u2} \ (mod\ p).$$

Finalement, on aboutit à la validité de DSA :

$$r = (g^k \ mod\ p) \ mod\ q = (g^{u1} y^{u2} \ mod\ p) \ mod\ q = v .$$

III.3 Fonctionnement du DSA :

Le modèle de schéma de signature numérique est représenté dans l'illustration suivante -

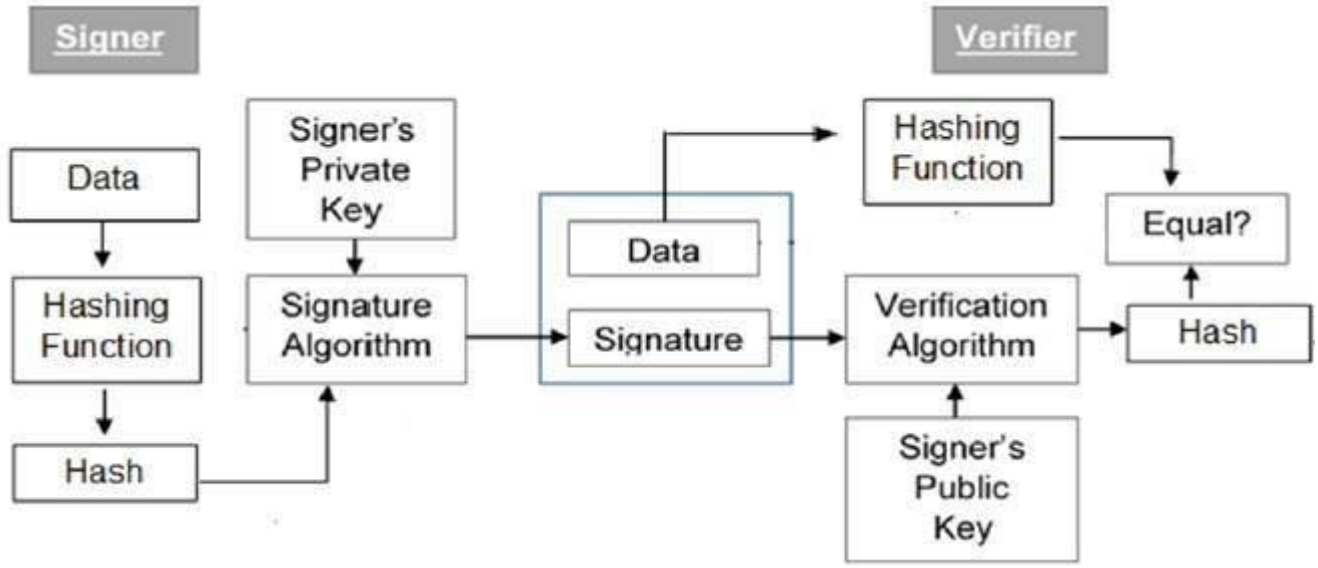

Figure III.4: Fonctionnement de DSA

Les points suivants expliquent l'ensemble du processus en détail -

- Chaque personne adoptant ce schéma a une paire de clés publique-privée.
- Généralement, les paires de clés utilisées pour le chiffrement / déchiffrement et la signature / vérification sont différentes. La clé privée utilisée pour la signature est appelée clé de signature et clé publique comme clé de vérification.

- Le signataire envoie des données à la fonction de hachage et génère un hachage de données.
- La valeur de hachage et la clé de signature sont ensuite transmises à l'algorithme de signature qui produit la signature numérique sur un hachage donné. La signature est ajoutée aux données, puis les deux sont envoyées au vérificateur.
- Le vérificateur introduit la signature numérique et la clé de vérification dans l'algorithme de vérification. L'algorithme de vérification donne une valeur en sortie.
- Le vérificateur exécute également la même fonction de hachage sur les données reçues pour générer une valeur de hachage.
- Pour la vérification, cette valeur de hachage et la sortie de l'algorithme de vérification sont comparées. Sur la base du résultat de la comparaison, le vérificateur décide si la signature numérique est valide.
- Puisque la signature numérique est créée par la clé « privée » du signataire et personne d'autre ne peut avoir cette clé; le signataire ne peut pas répudier la signature des données à l'avenir.

III.4 LA différence entre DSA et autres signatures numériques [28]

Contrairement à DSA, la plupart des types de signatures numériques sont générés en signant des condensés de message avec la clé privée de l'expéditeur. Cela crée une empreinte numérique des données. Puisque seul le résumé du message est signé, la signature est généralement beaucoup plus petite par rapport aux données qui ont été signées. En conséquence, les signatures numériques imposent moins de charge aux processeurs au moment de l'exécution de la signature, utilisent de petits volumes de bande passante et génèrent de petits volumes de texte chiffré destinés à la cryptanalyse.

DSA, d'autre part, ne crypte pas les condensés de message en utilisant la clé privée ou décrypte les digests de message en utilisant la clé publique.Au lieu de cela, il utilise des fonctions mathématiques uniques pour créer une signature numérique composée de deux nombres de 160 bits, qui proviennent des résumés de message et de la clé privée. Les DSA utilisent la clé publique pour authentifier la signature, mais le processus d'authentification est plus compliqué par rapport à RSA.

Les procédures de signature numérique pour RSA et DSA sont généralement considérées comme étant de force égale. Comme les DSA sont exclusivement utilisés pour les signatures numériques et ne contiennent aucune disposition pour le cryptage des données, ils ne sont généralement pas soumis à des restrictions d'importation ou d'exportation, qui sont souvent appliquées sur la cryptographie RSA.

III.5 Importance de la signature numérique

Parmi toutes les primitives cryptographiques, la signature numérique utilisant la cryptographie à clé publique est considérée comme un outil très important et utile pour assurer la sécurité de l'information.

Outre la possibilité de fournir une non-répudiation du message, la signature numérique fournit également l'authentification du message et l'intégrité des données. Voyons brièvement comment cela est réalisé par la signature numérique.

- **Authentification de message** - Lorsque le vérificateur valide la signature numérique à l'aide de la clé publique d'un expéditeur, il est assuré que la signature n'a été créée que par l'expéditeur qui possède la clé secrète privée correspondante et personne d'autre.
- **Intégrité des données** - Dans le cas où un attaquant a accès aux données et les modifie, la vérification de la signature numérique à la fin du récepteur échoue. Le hachage des données modifiées et la sortie fournie par l'algorithme de vérification ne correspondent pas. Par conséquent, le destinataire peut refuser le message en toute sécurité en supposant que l'intégrité des données a été violée.
- **Non-répudiation** - Comme il est supposé que seul le signataire a la connaissance de la clé de signature, il peut seulement créer une signature unique sur une donnée Ainsi, le destinataire peut présenter des données et la signature numérique à un tiers comme preuve si un différend survient à l'avenir.

En ajoutant le chiffrement à clé publique au schéma de signature numérique, nous pouvons créer un système de chiffrement qui peut fournir les quatre éléments essentiels de la sécurité, à savoir: la confidentialité, l'authentification, l'intégrité et la non-répudiation.

III.6 Conclusion

Dans ce chapitre nous avons détaillé le fonctionnement de l'algorithme DSA (Digital Signature Algorithm), et nous avons vu toutes les étapes du déroulement de cet algorithme pour la signature et la vérification, et finalement nous avons vu l'importance de la signature numérique.

Dans le chapitre suivant nous allons présenter le système d'exploitation Android.

Chapitre IV

Le Système D'exploitation Android

IV.1 HISTORIQUE [29]

A l'origine, Android était le nom d'une PME américaine, Android Incorporated, créée en 2003 puis rachetée par Google en 2005.

L'objectif était de développer un système d'exploitation mobile plus intelligent qui devrait permettre à l'utilisateur d'interagir avec son environnement (son emplacement géographique).

Avant 2007 les constructeurs concevaient tous un système d'exploitation spécifique pour leurs téléphones, et il n'y avait aucune base commune entre les systèmes d'exploitation mobiles de deux constructeurs différents. Ce système entravait la possibilité de développer facilement des applications qui s'adapteraient à tous les téléphones, surtout entre constructeurs, puisque la base était complètement différente. Mais durant cette année, la marque Apple a présenté une véritable révolution : Aphone. L'annonce de ce dernier était un désastre pour les autres constructeurs, qui doivent s'aligner sur cette nouvelle concurrence.

C'est pourquoi est créée en novembre de l'année 2007 l'Open Handset Alliance, et qui comptait à sa création 35 entreprises évoluant dans l'univers du mobile, dont Google. Cette alliance a pour but de développer un système ***open source*** (c'est-à-dire dont le code source est accessible à tous) pour l'exploitation sur mobile et ainsi concurrencer les systèmes propriétaires.

Depuis sa création, la popularité d'Android a toujours été croissante. C'est au quatrième trimestre 2010 qu'Android devient le système d'exploitation mobile le plus utilisé au monde

L'OHA compte à l'heure actuelle 80 membres.

IV.2 Architecture d'Android [30]

L'architecture de la plateforme Android se décline selon une démarche bottom up en quatre principaux niveaux que sont le noyau linux, les librairies et la plateforme d'exécution, le module de développement d'applications et enfin les différentes applications. Chacun de ces niveaux est décrit plus en détail ci-dessous :

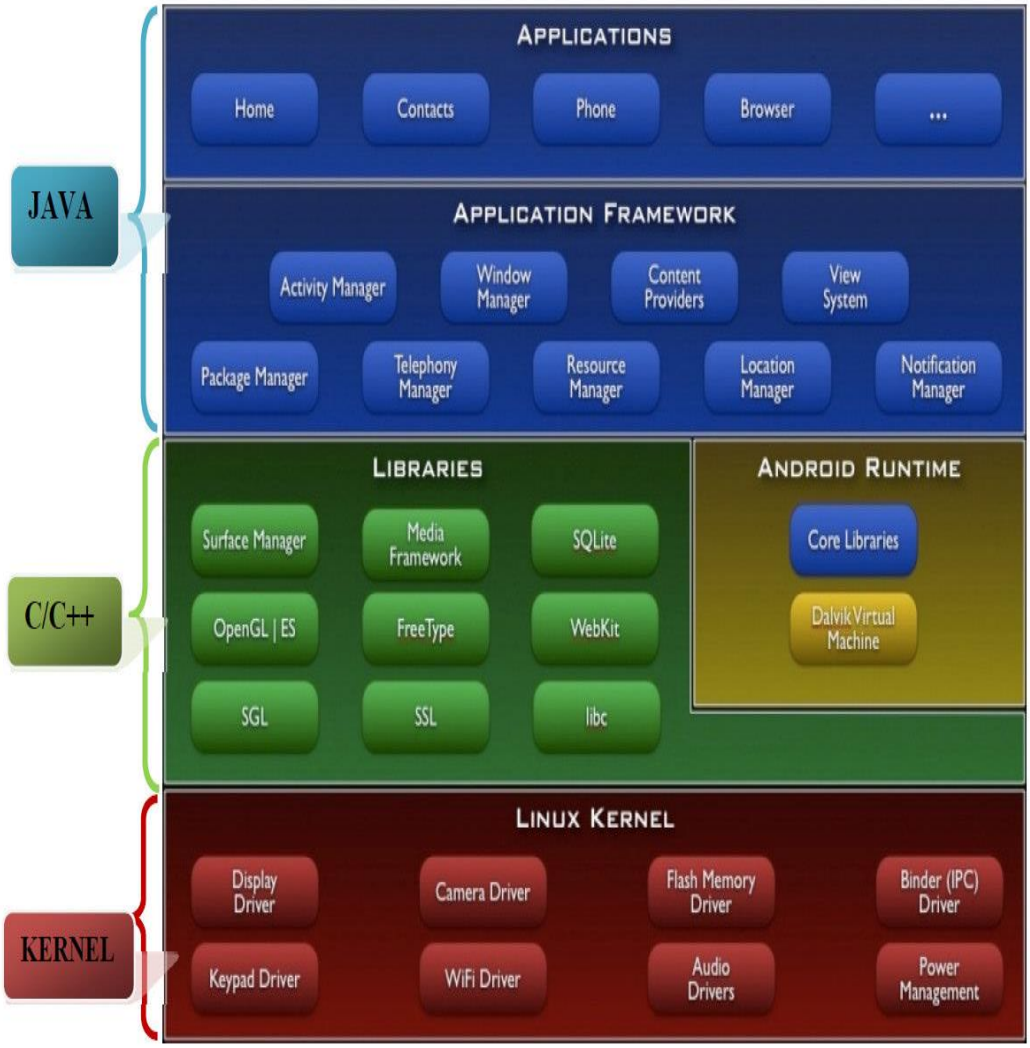

Figure IV.1 : Architecture d'Android

IV.2.1 Premier niveau: Les noyaux Linux:

Android s'appuie sur un noyau Linux 2.6 qui agit également comme une couche d'abstraction entre le matériel et le reste de la pile logicielle sur laquelle vient s'intégrer aux différents services tels que la sécurité, le gestionnaire de mémoire, le gestionnaire des processus et la pile réseau.

IV.2.2 Deuxième niveau:

a Les librairies:

Les librairies natives sont écrites en langage C et C++.

- La Surface Manager est chargée de la composition des items sur l'écran, de la gestion du dispositif d'affichage. Il permet de s'assurer que les pixels s'affichent bien à l'écran.
- OpenGL/ES quant à lui gère le graphisme en 3D tandis que SGL gère l'affichage en 2D. Ainsi une même application peut combiner du 2D avec du 3D.
- Le Media Framework fourni par la société PacketVideo (membre du OHA) contient des codecs audio et média (Mpeg 4, H.264, AAC, MP3…).
- Le Free type est une bibliothèque logicielle open source qui implémente un moteur de rendu de police de caractère.
- Le SQLite est une bibliothèque open source écrite en C permettant d'implémenter un moteur de base de données relationnelle.

b L'environnement d'exécution:

- Le «Runtime» est conçu spécifiquement pour des environnements embarqués (batterie, mémoire, CPU limités).
- Le Dalvik Virtual Machine exécute des fichiers de type «.dex» qui sont en fait le résultat en bytecodes de la conversion de fichier «.class» et «.jar». Il permet un usage de la mémoire, un partage entre processus plus efficace. C'est un interpréteur de bytecode optimisé. Il est possible d'avoir plusieurs instances de DVM s'exécutant au même moment.
- Les «Core Librairies» écrit en java est un ensemble de collection, de classes, d'utilitaires d'entrée/ sortie.

IV.2.3 Troisième niveau : Le module de développement d'application

Un framework fournit un ensemble de fonctions facilitant la création de tout ou d'une partie d'un système logiciel, ainsi qu'un guide architectural en partitionnant le domaine visé en module.

L'« Application Framework» développée en java contient un certain nombre d'applications dédiées (application téléphonique, des applications écrites par Google ou par un tiers). Toutes les applications peuvent utiliser le même API.

L'« Activity Manager» permet de gérer le cycle de vie d'une application (application en tâche de fond par exemple).

- Le packet manager garde une trace des applications installés dans l'équipement. Si on télécharge une nouvelle application par exemple, le packet manager informe sur la capacité du système
- Le «windows manager» s'occupe de gérer la fenêtre d'affichage.
- Le «Telephony manager» contient des API pour la construction d'une application téléphonique.
- Le «Content provider» permet le partage de données, l'interaction avec d'autres applications (repertoire, numéro de téléphone, dont on a besoin les autres applications).
- Le «Ressource Manager» quant à lui stocke les bitmaps locaux.

IV.2.4 Quatrième niveau: Les applications:

Niveau d'abstraction dans lequel on peut trouver toutes les applications spécifiques au fonctionnement d'un appareil mobile (téléphone, répertoire, navigateur web…).

IV.3 Environnement d'exécution Android:

L'environnement d'exécution d'Android est la machine virtuelle Dalvik, qui est incorporé dans le système d'exploitation Android et son rôle est de permettre l'exécution simultanée de plusieurs applications sur un appareil de faible capacité.

Les programmes sont écrits en JAVA puis compilés avec des outils Java afin d'obtenir un byte code qui sera lui-même recompilé avec l'outil (dex) pour obtenir un code adapté à la machine Dalvik, comme le montre le schéma si dessous:

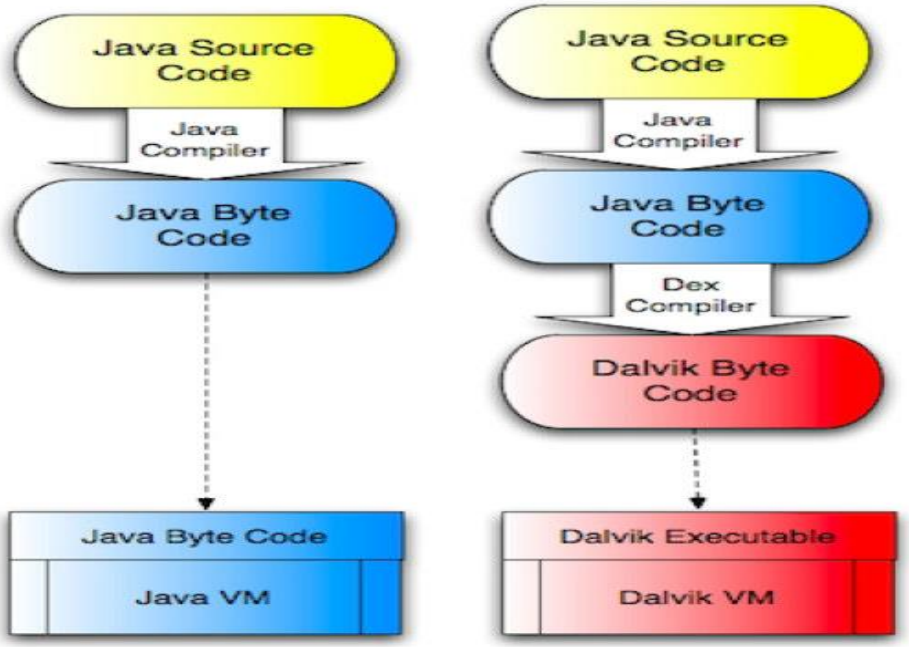

Figure IV.2: Environnement de développement Android

IV.4 Les composants d'une application Android :

IV.4.1 L'Activé (Activity) :

L'activité est l'élément le plus fréquent dans une application Android, elle correspond à un écran de l'interface de l'application. Par exemple, une application email peut avoir une activité qui affiche la liste des nouveaux emails, une autre activité pour écrire un nouvel email et une autre activité pour lire un email particulier, …etc.

Chaque activité est indépendante et peut constituer un point d'entrée de l'application. Une application peut faire appel à une activité particulière d'une autre application. Par exemple, l'appareil photo peut lancer l'activité de création de nouvel email pour envoyer une photo.

IV.4.2 Les services

Les services sont des tâches qui peuvent être lancées avec ou sans intervention de l'utilisateur.

Elles s'exécutent en background de l'application et peuvent se terminer soit après la finalisation de la tâche, soit à travers une intervention externe. Les services représentent également une

fonctionnalité d'une application exposée à d'autres applications. Il est important de mentionner que le service ne fournit pas d'interface graphique (User Interface).

Notre Player audio (lecteur de musique), par exemple, permet d'écouter la musique tout en consultant nos emails, etc.… Cette fonctionnalité n'est possible qu'à l'aide des Services.

IV.4.3 Intent /Broadcast Receiver

a Intent :

Les Intents sont des objets permettant de faire passer des messages contenant de l'information entre composants principaux. La notion d'Intents peut être vue comme une demande de démarrage d'un autre composant, d'une action à effectuer. La raison d'être des Intents provient du modèle de sécurité d'Android.

Chaque application est en effet sandboxée, cela veut dire qu'une application A ne peut accéder aux données d'une application B. Grace aux Intents, les applications ont la possibilité de fournir leurs services ou données si elles le souhaitent.

b Broadcast Receiver :

Les broadcasters sont les diffuseurs d'évènements/messages via des intentions. Les messages ainsi diffusés pourront être réceptionnés par plusieurs applications, les applications qui se seront abonnées à ces broadcastes (diffusions).

IV.4.4 Content Providers

Les content providers sont, comme l'exprime leurs noms, des gestionnaires de données. Ils permettent de partager l'information entre applications. Imaginons une application qui permet de conserver les cartes de visite virtuelles d'un ensemble de personne. Ces cartes de visite contiennent généralement le nom, le prénom, et un moyen de contact de la personne. Un tel programme peut être créé sous forme de content providers ce qui lui permettra de fournir à d'autres applications présentes sur le système les informations sur une personne. Une application tierce d'envoi de courriel d'un contact.

IV.5 Le cycle de vie d'une application activité

Le cycle de vie d'une activité correspond aux différents états d'une activité lors de sa gestion par le système Android.il est très important car il va vous permettre de suivre l'état de votre activité au fur et mesure de son existence dans le système Android.

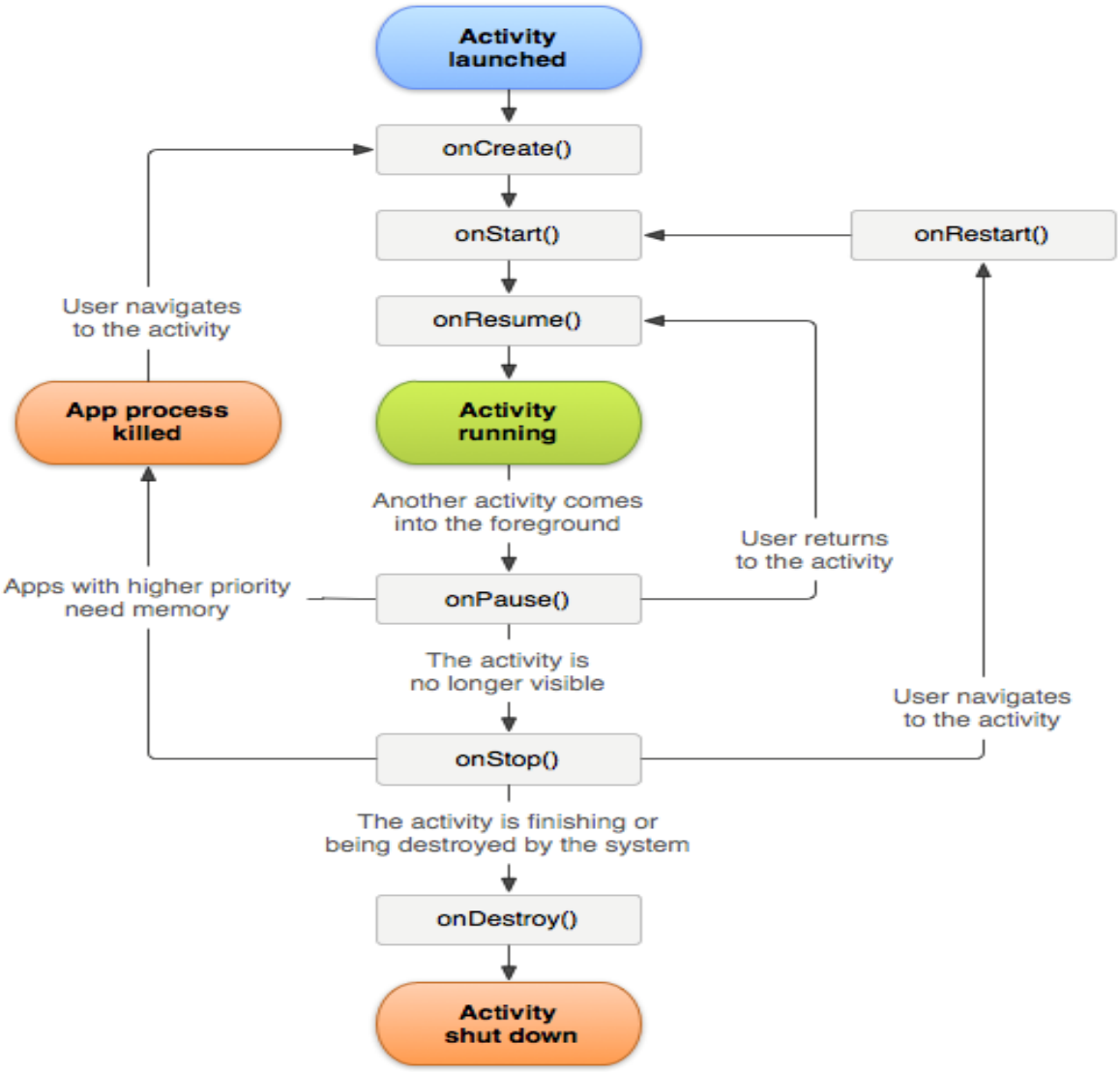

Figure IV.3:Le cycle de vie d'une activité

IV.5.1 Les états d'une Activité

Une activité peut se trouver sur trois états qui se différencient par leur visibilité :

- **Active (Resumed)**

- L'activité est visible en totalité.
- Elle est sur le dessus de la pile.
- Elle a le focus, ce qui veut dire que l'utilisateur agit directement sur l'activité et peut utiliser toute son intégralité.

- **Suspendue (Paused) :**

L'activité est partiellement visible à l'écran. C'est le cas lors de la réception d'un SMS et qu'une fenêtre semi-transparente se pose devant l'activité pour afficher le contenu du message. Ce n'est pas sur l'activité suspendue qu'agit l'utilisateur. L'application n'a plus le focus, c'est l'application sous-jacente qui l'a. Pour que notre application récupère le focus, l'utilisateur devra se débarrasser (stopper) de l'application qui l'obstrue, puis il pourra à nouveau interagir avec elle.

- **Arrêtée (Stopped) :**

L'activité est tout simplement masquée par une autre activité, on ne peut plus la voir. L'application n'a évidemment plus le focus, et on ne peut pas agir dessus. Le système retient son état pour pouvoir reprendre, mais il peut arriver que le système tue l'application pour libérer de la mémoire système. Les transitions d'états d'une activité sont captées par les méthodes suivantes :

IV.5.2 Fonctions pour la gestion d'une activité:

- **onCreate () :**

  Est appelée au début de la création de l'activité et n'est appelée qu'une seule fois. Elle joue le rôle du constructeur en permettant d'initialiser des variables, affecter des listener...

- **onRestart () :**

Appelée après un nouveau démarrage de la même activité (quand l'activité était arrêtée).

- **onStart () :**

L'activité va devenir visible. Cette méthode sert à lancer les animations, ou généralement tout ce qui est lié à l'affichage graphique, car elle est également appelée lors d'un retour de focus sur l'activité (dans ce cas onRestart est appelé avant).

- **onResume ():**

L'activité est maintenant visible, Cette méthode sera exécutée lorsque l'activité résume son exécution après la suspension (pause) et que l'activité commence à interagir avec l'utilisateur.

- **onPause () :**

Méthode qui sert à arrêter une activité temporairement.

- **onStop () :**

L'activité ne sera plus visible, cachée par une autre activité qui est en premier plan. Une activité stoppée est aussi en vie, elle est encore en mémoire mais elle n'est pas rattachée au gestionnaire des fenêtres du système Android. Elle peut être tuée par le système Android en cas de besoin en mémoires.

- **onDestroy() :**

L'activité va être détruite. La destruction opère quand quelqu'un appelle cette méthode ou quand c'est le système qui décide de tuer l'activité pour économiser de l'espace.

IV.6 Les versions d'Android [31]

Le système de Google n'aurait pas connu un tel succès s'il était resté le même. C'est là que l'on voit la puissance d'un tel OS qui a su s'adapter aux besoins des utilisateurs à chaque version majeure et qui s'enrichit de nouveautés.

Voici les différentes versions que se sont succédées et un bref aperçu de chacune d'elles :

Version	Nom	API Level	Fonctionnalités
Android 1.0	**Apple pie**	1	Recherche sur internet avec le moteur de recherche Google. Envoi de SMS et de MMS.

			Possibilité de personnaliser le fond d'écran. Application YouTube. Autres applications incluses: alarme, calculatrice, menu d'appel…etc Support du Wi-Fi et du Bluetooth.
Android 2.0	**Eclair**	5	Interface graphique améliorée. Support d'HTML 5. Support Bluetooth 2.1. Clavier virtuel amélioré. Refonte de l'installation et des mises à jour du kit de développement.
Android 3.0	**Honeycomb**	11	Optimisé pour les tablettes et équipement à écran large. Multitâche et système de notification amélioré. Corrections de bugs et améliorations du Wi-Fi, de la sécurité et de la stabilité.
Android 4.0	**Ice cream Sandwich**	14	Nouvelle interface graphique. Amélioration de la sécurité. Beaucoup de raccourcis (Appareil photo, accès sdcard, …etc).
Android 4.2	**Jelly Bean**	16	Permet d'activer/désactiver le Wi-Fi et Bluetooth avec un appui long dans le menu rapide de paramètres. Amélioration des performances et corrections de bugs Correction des problèmes de streaming avec l'audio BluetoothA2DP

Android 4.4	**KitKat**	19	Nouvelle interface translucide. Amélioration du système de notification. Gestion système des sous-titres. Amélioration des performances.
Android 5.0	**Lollipop**	21	Nouvelle interface / design ("Material design"). Supporte plusieurs cartes SIM. Amélioration de la rapidité. Amélioration de la gestion.
Android 5.1	**Lolipop**	22	Supporte plusieurs cartes SIM. Raccourci pour joindre un réseau Wi-Fi ou contrôler un appareil Bluetooth. Protection par blocage en cas de perte ou vol Appel voix en Haute Définition. Amélioration de la stabilité et des performances. Corrections de bugs.
Android 6.0	**Marshmallow**	23	Support de l'USB Type-C. Support de l'authentification par empreinte Digitale. Amélioration de la durée dela batterie avec un mode "deepsleep". Panneau pour contrôler les permissions des applications Améliorations de Google Now
Android 7.0	**Nougat**	24	Support de l'USB Type-C Support de l'authentification par empreinte digitale. Amélioration de la durée de la batterie avec un mode "deepsleep". Panneau pour controler les permissions des applications.

Android 8.0	**Oreo**	26	Améliorations de Google Now. PIP: Picture-in-Picture avec fenêtre Redimensionnable. Amélioration du système de notifications. Redesign de l'écran de verrouillage.

Tableau IV.1:Les versions d'Android

IV.7 Taux d'utilisation des versions d'Android [32]

Le tableau montre comme chaque mois plusieurs variations dans la répartition des parts de marchés entre les différentes versions du système d'exploitation.

Nouvelle venue dans la famille Android, la déclinaison Oreo présente pour le moment une part de marché très modeste, à 1,1 % (en additionnant les statistiques des versions 8.0 et 8.1).

Version	Codename	API	Distribution
2.2	Froyo	8	0.002%
2.3.3 - 2.3.7	Gingerbread	10	0.10%
3.1.0 - 3.2.6	Honeycomb	13	0.02%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	0.4%
4.1.x	Jelly Bean	16	1.3%
4.2.x	Jelly Bean	17	2.4%
4.3	Jelly Bean	18	0.9%
4.4	KitKat	19	12.2%
5.0	Lollipop	21	6.9%
5.1	Lollipop	22	14.0%
6.0	Marshmallow	23	25.1%
7.0	Nougat	24	25.2%
7.1	Nougat	25	7.9%
8.0	Oreo	n/a	2.6%
8.1	Oreo	n/a	1.0%

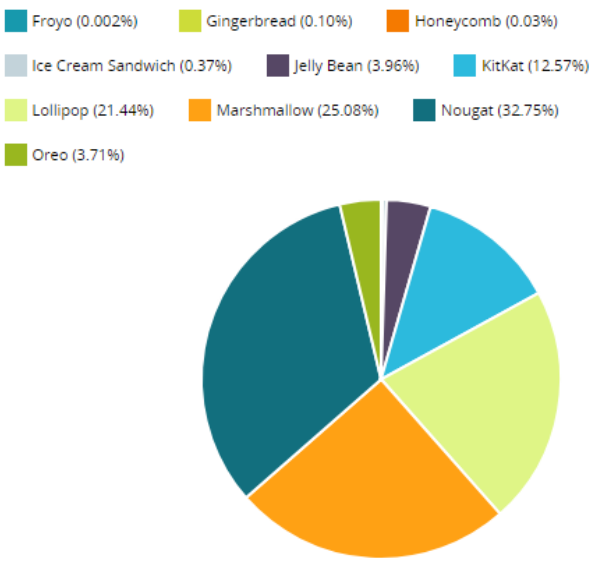

Tableau IV.2: Taux d'utilisation Versions d'Android

Au coude-à-coude, on retrouve Nougat (version 7.x de l'OS) et Marshmallow (6.0), dont la distribution tourne à un peu plus de 28 % pour les deux, 28,5 % pour la première branche et 28,1 % pour la seconde. Ce sont actuellement les deux moutures les plus répandues du système d'exploitation et d'assez loin, même si Lollipop n'est qu'à quelques coups de rame des deux premiers.

L'écart risque toutefois de se creuser pour **Lollipop**, sa présence sur les terminaux étant en recul mois après mois : elle se trouve actuellement sur un peu moins d'un quart des terminaux (24,6 %). Elle est toutefois la troisième version d'Android la plus déployée **sur les smartphones** et les tablettes qui ont été aperçus récemment sur Google Play au cours des derniers jours.

Les versions suivantes ne se partagent guère plus que des miettes. KitKat fait encore bonne figure, avec une part de marché à 12 %, mais en déclin, tandis que Jelly Bean ne se trouve plus que sur 5 % des terminaux. Les vieilles versions tendent à disparaître au profit des nouvelles et c'est tant mieux, ne serait-ce que pour des raisons de sécurité et de mises à jour des systèmes.

Enfin, quant aux autres moutures du système d'exploitation qui sont encore recensées par Google, elles sont pratiquement hors radar : Gingerbread se situe à 0,3 %. De son côté, Ice Cream Sandwich est légèrement en dessous, avec une part de marché à 0,4 %. Elles ont chuté de 0,1 point en janvier. Lorsqu'elles passeront en-dessous de 0,1 %, elles sortiront des statistiques de Google.

Ces statistiques sont collectées à partir de la nouvelle application Play Store, qui fonctionne avec Android 2.2 et plus, donc les appareils fonctionnant avec des versions plus anciennes ne sont pas inclus. Cela étant, en août 2013, les versions plus anciennes qu'Android 2.2 représentaient environ 1 % des appareils qui ont pointé sur les serveurs de Google «écrit l'entreprise américaine.

IV.8 Conclusion :

Tout au long de ce chapitre nous avons abordé le système d'exploitation conçus pour fonctionner sur les appareils ANDROID qui nous a permis de le positionner par rapport aux autres systèmes en spécifiant ses notions de bases, ses caractéristiques et ses possibilités d'utilisation. Dans le chapitre suivant nous allons implémenter notre solution en créant une application Android qui va signer des messages et qui assure authentification et l'intégrité.

Chapitre V

Implémentation

Introduction :

Dans ce chapitre nous allons présenter notre environnement de développement de notre application ainsi que les différents outils utilisés pour sa réalisation puis expliquer son fonctionnement en présentant quelques interfaces illustratives.

V.1 Description de l'environnement de travail

- Système d'exploitation Microsoft Windows 10.
- Environnement de développement : Android Studio et SDK.

V.1.1 Outils de développement :

Pour la réalisation de notre projet nous avons utilisé les outils de développement que nous verrons en détails dans ce qui suit:

V.1.2 Environnement de développement Android Studio et sa SDK : [33]

Android Studio est un nouvel environnement pour développement et programmation entièrement intégré qui a été récemment lancé par Google pour les systèmes Android. Il a été conçu pour fournir un environnement de développement et une alternative à Eclipse qui est l'IDE le plus utilisé.

Il est open source et disponible gratuitement, permettant de réaliser des projets sur différents types de support, tablette ou Smartphone.

La SDK Signifie Software Development Kit, c'est un ensemble d'outils d'aide à la programmation pour concevoir des logiciels, jeux, applications mobiles, etc. pour un terminal et/ou un système d'exploitation spécifique. Un SDK contient du code, permettant de concevoir une interface ou une partie d'une interface numérique (web, mobile, jeux, logiciels de recherches, widget météo…). Ce code est conçu avec le langage de programmation correspondant au terminal (ordinateur, téléphone, tablette…).

Installer ANDROID Studio :

Pour installer Android Studio, il est nécessaire d'avoir le logiciel du kit de développement Android (SDK), avec aussi le kit de développement (JDK) qui désigne un ensemble de bibliothèques logicielles de base du langage de programmation Java, ainsi que les outils avec lesquels le code Java peut être compilé.

On peut la télécharger du lien :

http://www.oracle.com/technetwork/java/javase/downloads/index.html

• Télécharger Android Studio du lien :

https://developer.android.com/studio/index.html

• Double clic sur l'exécutable téléchargé.

• Dans l'IDE en haut à droite cliquer sur l'icône "SDK Manager" afin de télécharger les versions utilisées pour le projet.

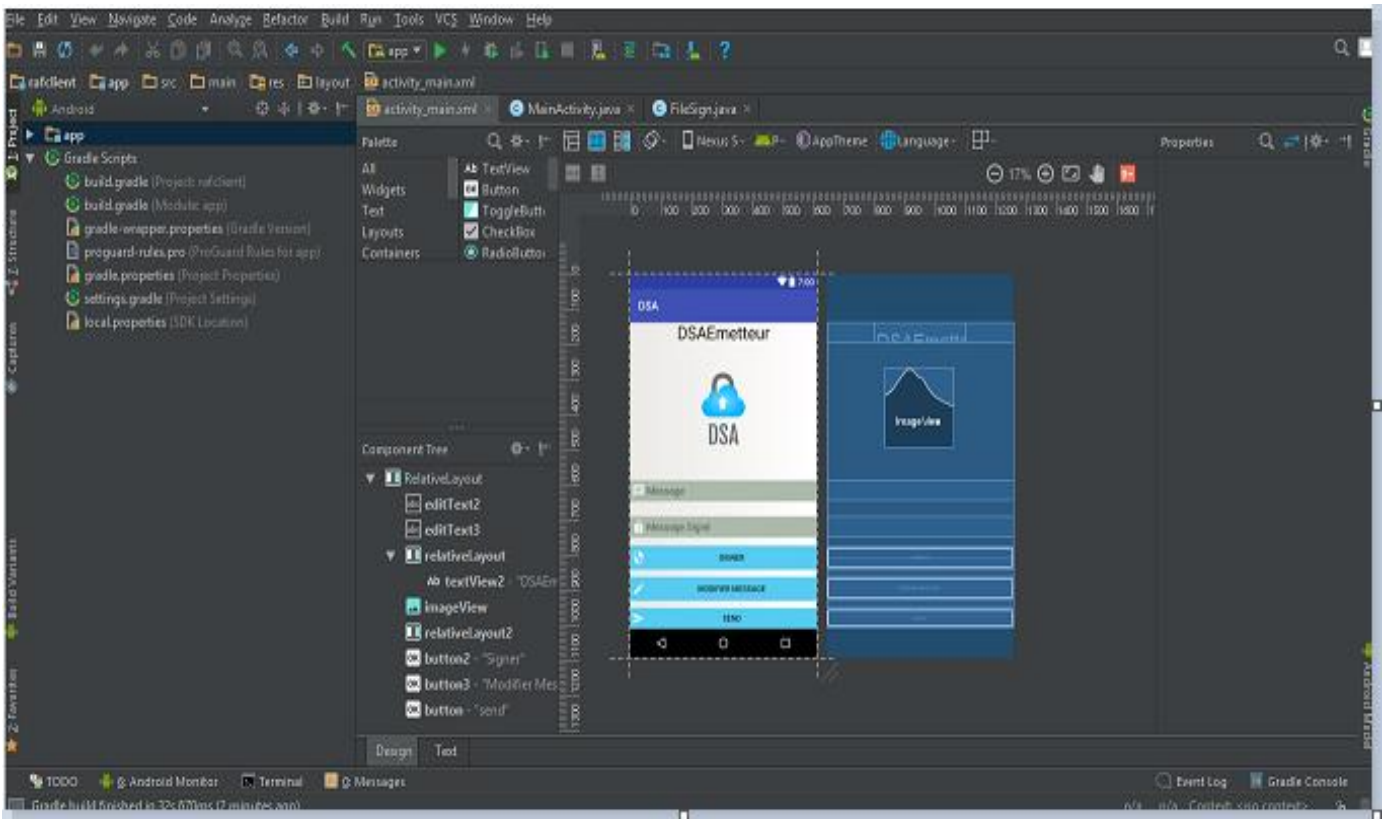

Figure V.1: Interface d'Android Studio.

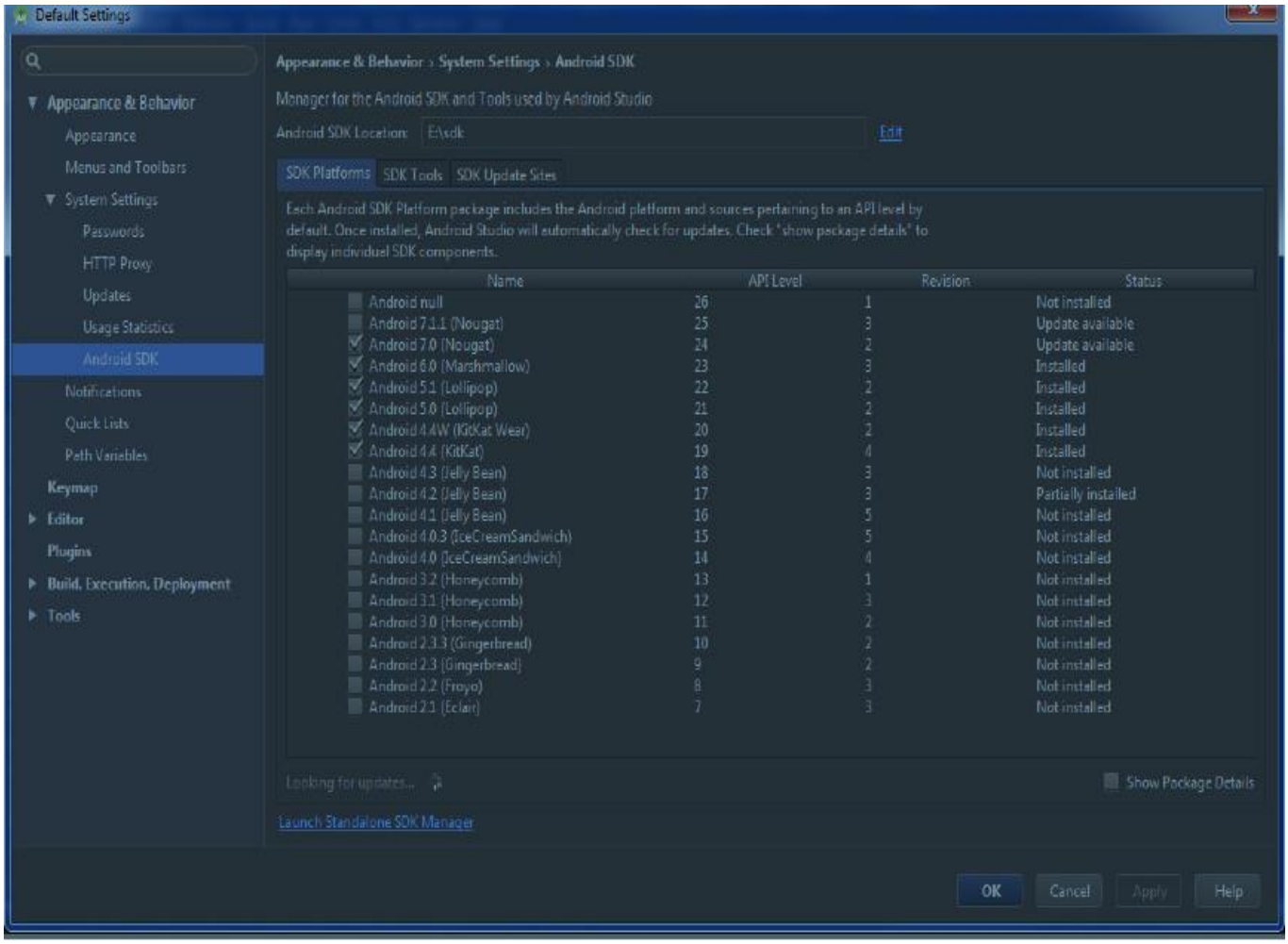

Figure V.2 Interface de l'Android SDK Manager.

V.1.3 Le langage de programmation:

Java est un langage de programmation orienté objet et reprend une syntaxe très proche de celle du langage C, développé par Sun Microsystems en 1995. Il est caractérisé comme étant un langage:

- Modulaire: on peut écrire des portions de code utilisables par plusieurs applications
- Rigoureux: les erreurs se produisent la compilation et non a l'exécution.
- Et l'une de ses plus grandes forces est son excellente portabilité, car une fois un programme a été créé il fonctionnera automatiquement sous Windows, Mac, Linux, UNIX …

V.2 Présentation des interfaces de notre application:

L'application prend en charge deux interfaces principales :

- Interface Emetteur
- Interface récepteur

Nous allons présenter dans ce qui suit les principales interfaces illustrant le fonctionnement de l'application :

V.2.1 Interface Emetteur

Cette partie permet à l'utilisateur d'envoyer un message signé par sa propre clef privée.

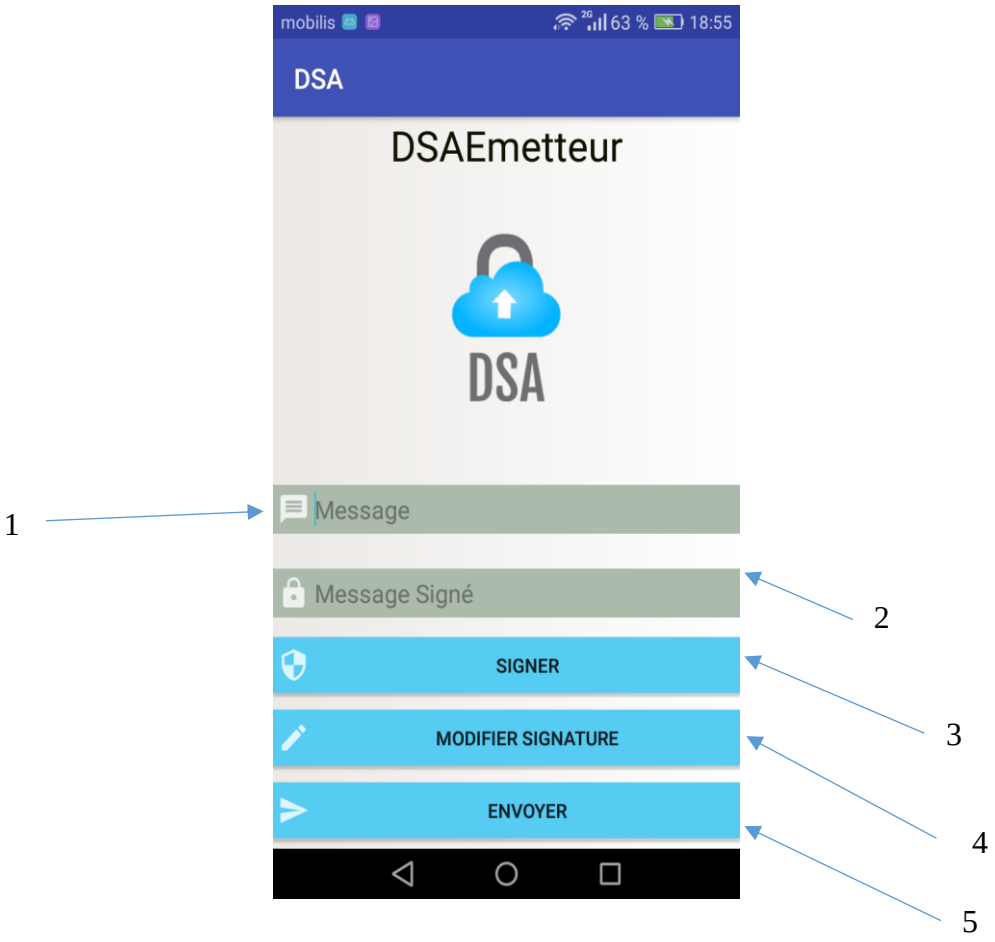

Figure V.3: Interface Emetteur

1: Saisir texte(message) en clair

2: Afficher le texte signé

3: Signer le message

4: modifier la signature

5: Envoyer le message

❖ Signer le message

Après avoir rempli le champ message (bonjour), On clique sur le bouton Signer

Figure V.4: Signer le Message

Après avoir cliqué sur le bouton Signer, on remarque un texte long (message signé) s'affichera dans la zone correspond au champ (2) (message signé).

❖ Modifier la signature

Après avoir cliqué sur le bouton Modifier Signature on remarque que le contenu de la zone N°2 a été modifier.

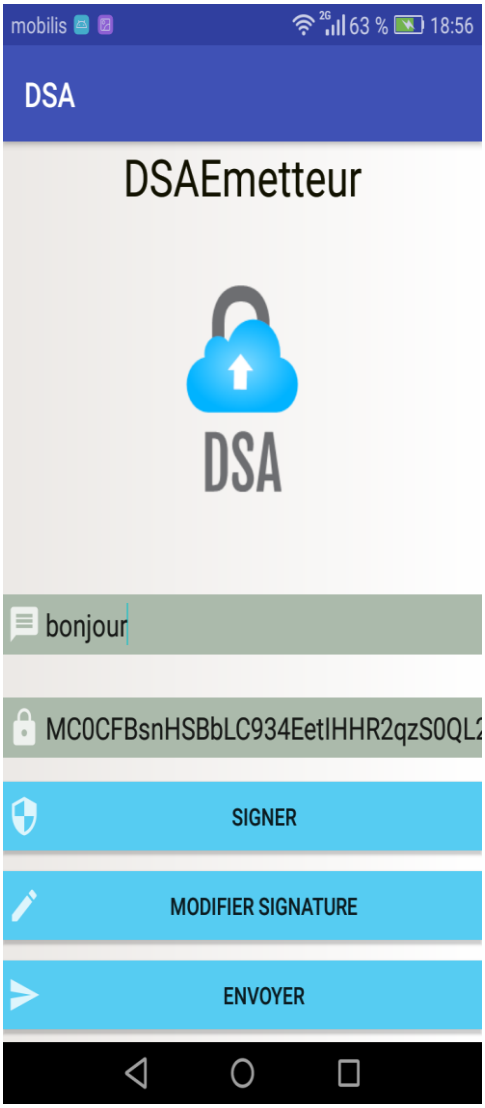

Figure V.5: Modifier La Signature

V.2.2 Interface Récepteur

Cette deuxième partie s'exécute sur un autre smartphone mais connecter au même point d'accès que la première (DSAEmetteur), elle permet de vérifier l'authentification de message, l'intégrité, et la non répudiation

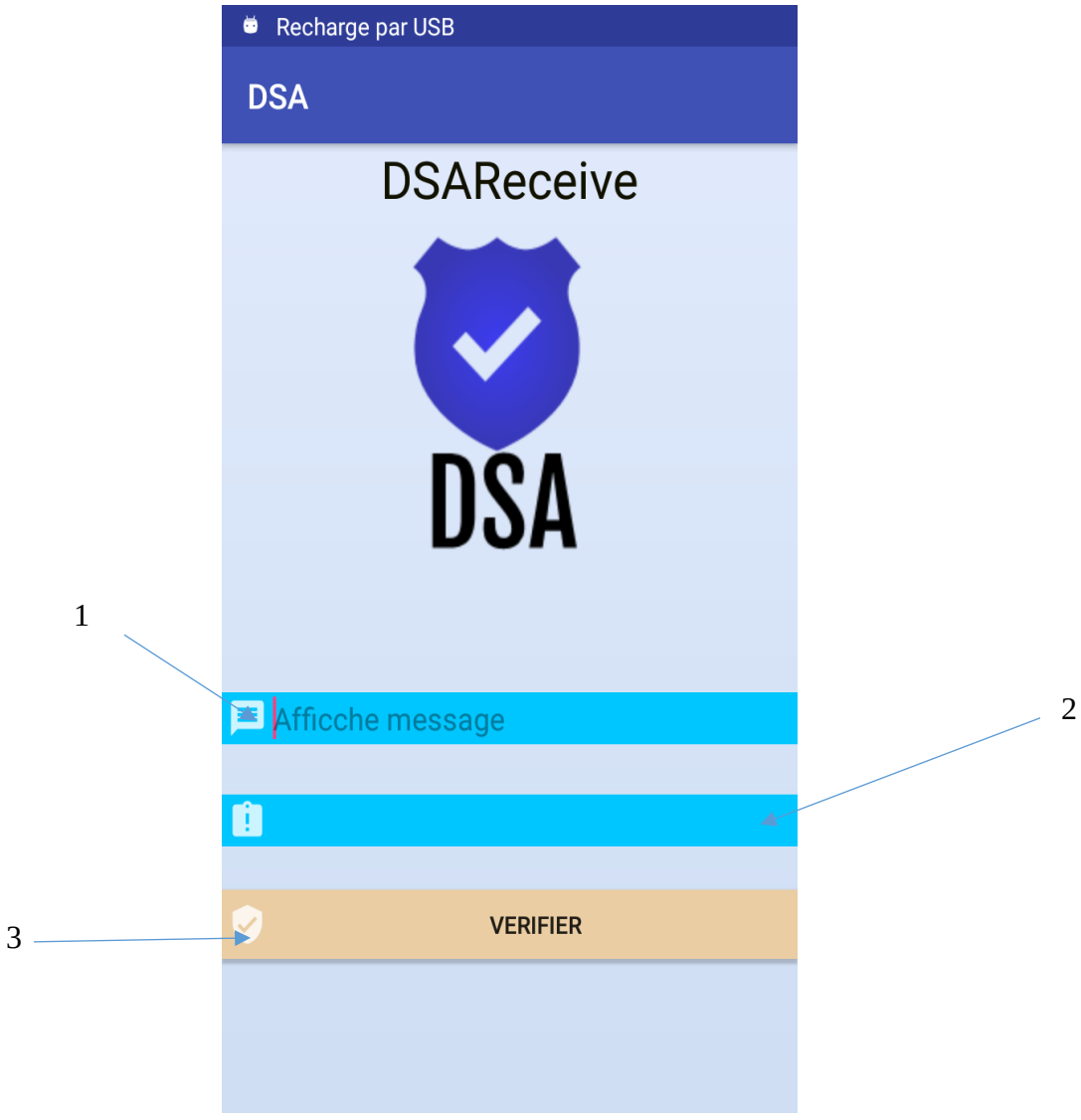

Figure V.6: Interface Récepteur

1: Afficher le message envoyé par l'émetteur.

2: Afficher la réponse de vérification.

3: Vérifier le message.

❖ Vérification de message

Dans la vérification on distingue deux partie :

La 1ère partie est: lorsque le récepteur reçoit le message sans modification.

Figure V.7: Vérification de Message

Après avoir clique sur le bouton vérifier, un message (le message est authentique) s'affichera, au niveau de deuxième champ.

La 2eme partie est: lorsque le récepteur reçoit le message modifié.

Figure V.8: Vérification de message

Après avoir cliqué sur le bouton vérifier un message (le message signature a été modifiée) s'affichera, au niveau de deuxième champ

VI Conclusion :

Ce dernier chapitre a été consacré à la présentation de l'étape réalisation de notre application ainsi, nous avons présenté les outils logiciels qui nous ont permis la réalisation de notre travail à savoir l'environnement de développement et les langages de programmation.

Puis, nous sommes passés aux présentations de notre application en décrivant ses fonctionnalités et présentant plusieurs interfaces.

Conclusion générale

La sécurité informatique est l'ensemble des moyens techniques, organisationnels, juridiques et humains nécessaires et mis en place pour conserver, rétablir, et garantir la sécurité des systèmes informatiques. Elle est intrinsèquement liée à la sécurité de l'information et des systèmes d'information d'où la sécurité doit être un processus transverse à tout système présentant des risques et supportant des menaces et pour cela au cours de notre mémoire on a réalisé un crypto système basé sur l'implémentation de l'algorithme DSA sur les systèmes Android.

Pour cela nous avons dans un premier temps commencé par présenter des généralités sur la sécurité informatique, ensuite nous avons vu quelques notions de base sur la cryptographie, ainsi nous avons détaillé le fonctionnement de l'algorithme DSA, et dans la partie suivante nous avons abordés quelques outils de développement Android qui nous servir à réaliser notre application.

Au cours de la réalisation de ce projet nous avons enrichis notre savoir et développé nous connaissances informatique notamment dans la programmation et la cryptographie. En effet l'application a exigé des connaissances du langage JAVA et des outils de développement indispensables à sa réalisation. La mise en œuvre de notre travail a exigé des connaissances très approfondies en la matière ainsi qu'une bonne maitrise de la configuration d'Android et de l'environnement Android.

Et enfin, notre travail ainsi présenté reste un prototype sur lequel nous espérant apporter plus de connaissances pour le rendre meilleur et plus fonctionnel, ainsi nous aimerons améliorer cette application pour qu'elle puisse contribuer à certifier et signer tous types de donnés (images, vidéos…etc).

Résumer :

La sécurité Informatique est devenue une préoccupation importante des utilisateurs et des entreprises dans tous les domaines. Vu l'expansion et l'importance grandissante des réseaux informatiques donc il est indispensable de renforcer les mesures de sécurités, dans le but de maintenir la confidentialité, l'intégrité et le contrôle d'accès au réseau pour réduire les risques d'attaques.

On a commencé par présenter des généralités sur la sécurité informatique, ensuite nous avons vu quelques notions de base sur la cryptographie, ainsi nous avons détaillé le fonctionnement de l'algorithme DSA, Et enfin, notre travail ainsi présenté reste un prototype sur lequel nous espérant apporter plus de connaissances pour le rendre meilleur et plus fonctionnel, ainsi nous aimerons améliorer cette application pour qu'elle puisse contribuer à certifier et signer tous types de donnés (images, vidéos…etc).

Bibliographie

[1] http://www.marche-public.fr/Terminologie/Entrees/Reseau-securite.html

[2] https://www.commentcamarche.com/contents/1033-introduction-a-la-securite-informatique.

[3] https://www.securiteinfo.com/conseils/introsecu.shtml

[4] https://www.commentcamarche.com/contents/995-protection-introduction-a-la-securite-des-reseaux

[5] www.mathgon.com/Cours/ESMISAB/CM4.pdf

[6] https://tpesecuriteinformatique.wordpress.com/les-differentes-attaques-informatiques/

[7] https://www.commentcamarche.com/contents/203-cryptographie

[8] http://www.montefiore.ulg.ac.be/~dumont/pdf/crypto09-10.pdf

[9]https://developer.mozilla.org/fr/docs/Introduction_%C3%A0_la_cryptographie_%C3%A0_clef_publique/Certificats_et_authentification

[10] https://www.supinfo.com/articles/single/41-vpn-virtual-private-network-fonctionnement-interets

[11] https://www.cisco.com/c/fr_fr/products/security/firewalls/what-is-a-firewall.html

[12] www.univ-bejaia.dz/.../Détection%20et%20Prévention%20des%20Intrusions%20dans...

[13]http://dlibrary.univboumerdes.dz:8080/bitstream/.../1/Hadaoui%20Rebiha%20magister.pdf

[14] https://cryptoloblog.wordpress.com/la-cryptologie-cest-quoi/

[15] https://cryptoloblog.wordpress.com/la-cryptologie-cest-quoi/#cryptanalyse

[16] https://cryptoloblog.wordpress.com/la-cryptologie-cest-quoi/#cryptographie

[17] https://www.sciences.unilim.fr/informatique/master-cryptis/

[18]http://s7deff5c7b202eeed.jimcontent.com/download/version/1457462978/module/10962550057/name/Chapitre%2B1%2BIntroduction.pdf

[19] http://dspace.univ-tlemcen.dz/bitstream/112/1076/5/chapitre1.pdf

[20] http:://dspace.univ-tlemcen.dz/bitstream/112/.../Etude-comparative-entre-la-cryptographie.pdf

[21] RAHMOUN SAMIA, Développement d'une application pour l'echange des messages sécurisés, Université Abou Bakr Belkaid- Tlemcen, 2015

[22] http://www.bibmath.net/crypto/index.php?action=affiche&quoi=moderne/aes

[23] https://www.futura-sciences.com/tech/definitions/tech-rsa-1787/

[24]http://dspace.univ-tlemcen.dz/bitstream/112/.../Etude-comparative-entre-la-cryptographie.pdf

[25]http://sieil37.fr/images/Actualites/...11.../CR_V02_signature_electronique_24_11_2016.pdf

[26] http://www-igm.univ-mlv.fr/~dr/XPOSE2006/depail/fonctionnement.html

[27] https://csrc.nist.gov/csrc/media/.../fips/186/3/...06.../fips_186-3.pdf

[28]https://developer.mozilla.org/fr/docs/Introduction_%C3%A0_la_cryptographie_%C3%A0_clef_publique/Signatures_num%C3%A9riques

[29] http://www.phonandroid.com/toute-l-histoire-et-la-chronologie-d-android-dossier.html

[3] http://www-igm.univ-mlv.fr/~dr/XPOSE2008/android/archi_comp.html

[31] https://www.android.com/intl/fr_fr/histoy

[32] https://www.numerama.com/tech/132165-les-versions-dandroid-les-plusutilisees.html

[33] http://android-studio.fr.uptodown.com/windows