REPUBLIQUE ALGERIENNE DEMOCRATIQUE ETPOPULAIRE

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITÉ MOULOUD MAMMERI DE TIZI-OUZOU

FACULTÉ DE GÉNIE ELECTRIQUE ET D'INFORMATIQUE

DEPARTEMENT D'INFORMATIQUE

Mémoire De Fin d'études En Vue De l'Obtention D'un Master Académique

Filière : Informatique

Spécialité : Réseaux, Mobilités et Systèmes Embarqués

Thème

Application Mobile Android De Chat Sécurisée Avec DSA et RSA

Présenté par :

Bougdour Mohamed Khaled

Proposé et Encadré par :

Mme. Hadaoui Rebiha

Promotion 2019/2020

Sommaire

LISTE DES FIGURES :

Introduction générale :

De nos jours, les équipements mobiles connectés au réseau internet sont en constante évolution grâce à l'apparition de nouvelles technologies de communication et de nouveaux réseaux comme les réseaux sociaux. La prolifération de ces objets connectés engendre de nouvelles possibilités de partage de ressources. La mise en commun de ces ressources offre une puissance de calcul qu'on peut exploiter pour faire des calculs complexes, notamment dans la 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 Mon projet de fin d'étude, est de développer une application Android qui contribue à signer et crypter des messages dans un chat de bout en bout (End-to-End) et qui assure l'authentification, l'intégrité et la confidentialité.

Ce mémoire comporte quatre chapitres:

✓ Le premierchapitreest une introduction générale du mémoire quidéfinit la sûreté de fonctionnement, la sécurité informatique avec ses modèles et ses objectifs

✓ Le deuxième chapitre est consacré à la cryptographie : il évoquela cryptographie et la représentationdes différents algorithmes connus de cryptage.

✓ Le troisième chapitretraite l'algorithme de cryptage RSA et l'algorithme de signature DSAimplémentés dans notre solution. Il définit le fonctionnement des deux algorithmesRSA et DSA.

✓ Le quatrième chapitre décrit l'implémentation et la mise en œuvre de notre solution « Application mobile Android de Chat sécurisée, en utilisant l'algorithme RSA pour le cryptage des messages et DSA pour l'intégrité du message ».Ontermine le mémoire par une conclusion générale et des perspectives ouvertes pour des travaux futurs.

Chapitre I

La Sécurité Informatique

Introduction :

Ce chapitreintroduit la sûreté de fonctionnement d'un système, plus précisément la sécurité informatique qui réunit les trois objectifs de la sureté informatique à savoir : la confidentialité, l'intégrité et la disponibilité. Il aborde aussi les causes de l'insécurité et les mécanismes pour assurer la sureté informatique.

I.1 La Sécurité Informatique:

I.1.1 Définition: [1]

La sécurité informatique concerne la protection des systèmes informatiques et des informations contre les attaques, le vol et l'utilisation non autoriséede ces derniers. La principale raison pour laquelle les utilisateurs sont fréquemment attaqués est qu'ils n'ont pas de défenses adéquates pour empêcher les intrus, et les cybercriminels exploitent rapidement ces faiblesses. La sécurité informatique garantit la confidentialité, l'intégrité et la disponibilité des ordinateurs et de leurs données stockées. Elle fait aussi référence à l'ensemble des technologies, des processus et des pratiques conçus pour protéger les réseaux, les appareils, les programmes et les données contre les attaques, les dommages ou les accès non autorisés.

I.1.2 L'importance de la Sécurité Informatique :[2]

La sécurité informatique est importante car les organisations gouvernementales, militaires, commerciales, financières et médicales collectent, traitent et stockent des quantités sans précédent de données sur des ordinateurs et d'autres appareils. Une partie importante de ces données peut être des informations sensibles, qu'il s'agisse de propriété intellectuelle, de données financières, d'informations personnelles ou d'autres types de données pour lesquelles un accès ou une exposition non autorisé pourrait avoir des conséquences négatives. Ces organisations transmettent des données sensibles sur des réseaux et à d'autres appareils au cours de leurs activités, et la sécurité informatiquedécrit la discipline dédiée à la protection de ces informations et des systèmes pour les traiter ou les stocker. À mesure que le volume et la sophistication des cyberattaques augmentent, les entreprises et les organisations, en particulier celles qui sont chargées de protéger les informations relatives à la sécurité nationale, à la santé ou aux dossiers financiers, doivent prendre des mesures pour protéger leurs informations commerciales et personnelles sensibles. Dès mars 2013, les plus hauts responsables du renseignement du pays ont averti que les cyberattaques et l'espionnage numérique constituaient la principale menace pour la sécurité nationale, éclipsant même le terrorisme.

I.1.3 L'Objectif de la Sécurité Informatique : [3]

L'objectif général de la sécurité informatique consiste à assurer que les ressources matérielles et logicielles d'une organisation sont uniquement utilisées conformément aux missions qui leur sont assignées, notamment :

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

I.1.4 Défis de la Sécurité Informatique : [2]

Pour une sécurité informatique efficace, une organisation doit coordonner ses efforts dans l'ensemble de son système d'information. Leconcept de la cyber-sécuritéenglobe tous les éléments suivants:

- ✓ **Sécurité du réseau**: processus de protection du réseau contre les utilisateurs indésirables, les attaques et les intrusions.
- ✓ **Sécurité des applications**: les applications nécessitent des mises à jour et des tests constants pour garantir la sécurité de ces programmes contre les attaques.
- ✓ **Sécurité des terminaux**: l'accès à distance est un élément nécessaire de l'entreprise, mais peut également être un point faible pour les données. La sécurité des terminaux est le processus de protection de l'accès à distance au réseau d'une entreprise.
- ✓ **Sécurité des données**: à l'intérieur des réseaux et des applications se trouvent les données. La protection des informations sur l'entreprise et les clients est une couche de sécurité distincte.
- ✓ **Gestion de l'identité:** il s'agit essentiellement d'un processus de compréhension de l'accès que chaque individu a dans une organisation.
- ✓ **Sécurité des bases de données et des infrastructures:** tout dans un réseau implique des bases de données et des équipements physiques. La protection de ces appareils est tout aussi importante.
- ✓ **Sécurité du Cloud:** de nombreux fichiers se trouvent dans des environnements numériques ou «le Cloud». La protection des données dans un environnement 100% en ligne présente un grand nombre de défis.
- ✓ **Sécurité mobile:** les téléphones portables et les tablettes impliquent pratiquement tous les types de défis de sécurité en eux-mêmes.
- ✓ **Planification de la reprise après sinistre / continuité des activités:** en cas de violation, les données relatives aux catastrophes naturelles ou à d'autres événements doivent être protégées et les activités doivent se poursuivre. Pour cela, l'élaboration d'un plan s'impose :

  - la formation des utilisateurs finaux : les utilisateurs peuvent être des employés accédant au réseau ou des clients se connectant à une application de l'entreprise
  - l'éducation aux bonnes habitudes (changement de mot de passe, authentification à 2 facteurs, etc.) est un élément important de la sécurité informatique.

I.1.5 Les Objectifs Principaux de la Sécurité Informatique : [4]

- **Confidentialité**: Consiste à protéger les données transmises contre les attaques passives et les flux de données contre l'analyse, et préserver le secret des données ; seules les entités communicantes sont capables d'accéderaux données.
- **Intégrité**: ce terme recouvre deux concepts connexes:
  - Intégrité des données: garantit que les informations et les programmes sont modifiés uniquement d'une manière spécifiée et autorisée.

- Intégrité du système: garantit qu'un système remplit sa fonction prévued'une manière non affectée, libre de délibérer ou de non-autoriser la manipulation du système.

- **Disponibilité**: garantit que les systèmes fonctionnent immédiatement et que le service n'estpas refusé aux utilisateurs autorisés.

«Ces trois concepts forment ce que l'on appelle souvent la triade CIA »

- **Authenticité**: la propriété d'être authentique et de pouvoir être vérifié et digne de confiance; confiance dans la validité d'une transmission, message ou l'expéditeur du message. Cela signifie : vérifier que les utilisateurs sont bienles personnes qu'ils prétendent êtres et que chaque entrée arrivant dans le système provient d'une source sûre.
- **La non répudiation**: empêche l'émetteur ou le receveur de nier avoir transmis ou reçu un message.

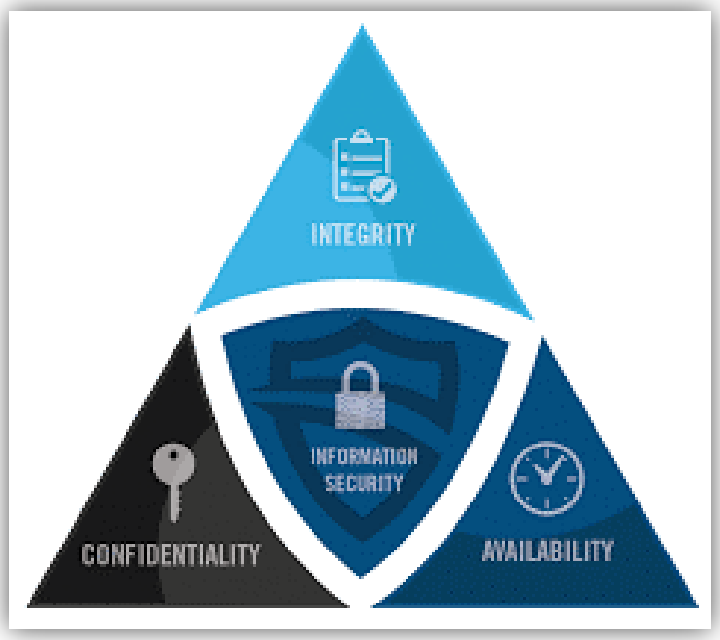

Figure 1.1La triade CIA

I.2 Causesde l'insécurité :[3]

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 Les causes des attaques : [5]

Avant de découvrir comment sécuriser les données contre les violations, on doit essayer de comprendre les motifs de ces attaques. En connaissant les motifs des attaques, il est facile pour les professionnels ducyber sécurité de sécuriser les systèmes. Les principalescauses d'attaque de l'ordinateur d'une organisation ou d'un individu sont les suivantes:

- **Perturber la continuité d'une entreprise**: si une entreprise est perturbée, cela cause un grand préjudice à l'organisation sous la forme de pertes de profits, de fraude et de dommages à sa réputation.
- **Vol d'informations et manipulation de données**: les pirates prennent des informations confidentielles qu'ils volent aux organisations et les vendent à des particuliers ou à des groupes sur le marché noir.
- **Créer le chaos et la peur en perturbant les infrastructures critiques**: les cyber-terroristes attaquent une entreprise ou un organisme gouvernemental pour perturber leurs services, causant des dommages qui peuvent potentiellement affecter toute une nation.
- **Perte financière pour la cible**: les pirates informatiques attaquent une organisation ou une entreprise et interrompent leurs services de telle sorte que la cible doit allouer des fonds substantiels pour réparer les dommages.
- **Atteindre les objectifs militaires d'un État**: les nations rivales se surveillent constamment et utilisent parfois des tactiques cybercriminelles pour voler des secrets militaires.
- **Demande de rançon**: les pirates utilisent des « Ransomwars» pour bloquer un site Web ou des serveurs, ne libérant le contrôle qu'après le paiement d'une rançon.
- **Atteinte à la réputation de la cible**: le pirate informatique peut avoir des raisons personnelles d'attaquer une organisation ou un individu afin que sa réputation en souffre.
- **Propager des croyances religieuses ou politiques**: les pirates informatiques peuvent s'infiltrer sur des sites Web pour promouvoir un dogme religieux ou un certain programme politique, généralement pour inciter les électeurs à voter d'une certaine manière.

I.3.2 Les Types D'attaque :

Les attaques peuvent être classées en attaques passives ou actives :

•Attaque passive :

- Ne modifie pas le contenu de l'information.
- Tente de collecter ou d'utiliser des informations relatives au système, mais elle n'affecte pas les ressources du système.
- Très difficile à détecter mais assez facile à sécuriser (cryptage).

Il y a deux catégories essentielles d'attaque passive :

- **Interception des messages** : En vue de tirer des informations pertinentes ou compromettantes (mots de passes, informations confidentielles, …etc)
- **Analyse du trafic** : En vue de comprendre l'architecture du réseau et de déceler les points faibles et les ressources importantes à attaquer.

•Attaque active :

- Entraine la modification de l'information ou la création de fausses informations.
- Il est difficile d'empêcher les attaques actives de façon absolue à moins de protéger physiquement tous les moyens et chemins de communications en même temps.

Il y a quatre catégories essentielles d'attaque passive :

- **Mascarade** : Une entité prétend être une entité différente afin d'obtenir des privilèges.
- **Re-jeu (Replay):** capture passive des données et leurs transmissions ultérieures en vue de réaliser des actions non autorisées.
- **Modification:** Altération, destruction, ou injection dans un message échangé en vue de produire un effet non souhaité ou non autorisé.
- **Déni de service:** Empêcher ou inhiber l'utilisation normale des moyens de communications.

I.3.3 Les Techniques d'attaque :

I.3.3.1 Attaque Virale :

- **Les virus :** Un virus informatique est un type de logiciel malveillant qui se propage en insérant une copie de lui-même dans un autre programme et en y faisant partie. Il se propage d'un ordinateur à un autre, laissant des infections en cours de route. La gravité des virus peut varier, allant d'effets légèrement gênants à des données ou des logiciels dommageables et provoquant des conditions de déni de service (DoS).

- **Les vers :** Les vers informatiques sont similaires aux virus dans la mesure où ils reproduisent des copies fonctionnelles d'eux-mêmes et peuvent provoquer le même type de dommages. Contrairement aux virus, qui nécessitent la propagation d'un fichier hôte infecté, les vers sont des logiciels autonomes et ne nécessitent pas de programme hôte ni d'aide humaine pour se propager. Pour se propager, les vers

exploitent une vulnérabilité sur le système cible ou utilisent une sorte d'ingénierie sociale pour inciter les utilisateurs à les exécuter. Un ver pénètre dans un ordinateur par une vulnérabilité du système et tire parti des fonctions de transport de fichiers ou de transport d'informations sur le système, lui permettant de voyager sans aide. Les vers plus avancés exploitent le chiffrement, les essuie-glaces et les technologies de Ransomwars pour endommager leurs cibles.

- **Cheval de Troie :** C'est un logiciel nuisible qui semble légitime. Les utilisateurs sont généralement amenés à le charger et à l'exécuter sur leurs systèmes. Une fois activé, il peut atteindre un nombre illimité d'attaques contre l'hôte, allant d'irriter l'utilisateur (faire apparaître des fenêtres ou changer de bureau) jusqu'à endommager l'hôte (supprimer des fichiers, voler des données ou activer et diffuser d'autres logiciels malveillants, tels que des virus). Les chevaux de Troie sont également connus pour créer des portes dérobées pour permettre aux utilisateurs malveillants d'accéder au système.

I.3.3.2 Attaque de reconnaissance :

- **Sniffer :** est un programme qui surveille et analyse le trafic réseau. Il est conçu pour détecter les goulots d'étranglement et les problèmes sur le réseau. Grâce à ces informations, un gestionnaire de réseau peut maintenir un trafic efficace.

- **Scanneur de ports** : L'analyse des ports est l'une des techniques les plus utilisées par les attaquants pour découvrir les services qu'ils peuvent exploiter pour pénétrer dans les systèmes. Bien que l'analyse de port ne soit pas intrinsèquement hostile, c'est souvent la première étape de reconnaissance utilisée par les pirates informatiques lorsqu'ils tentent d'infiltrer un réseau ou de voler / détruire des données sensibles.

I.3.3.3 Attaque d'accès :

- **Attaques de mot de passe :**en prenant « brut force » comme exemple : lorsqu'un hacker utilise un ensemble de valeurs prédéfinies pour attaquer une cible et analyser la réponse jusqu'à ce qu'il réussisse. Le succès dépend de l'ensemble des valeurs prédéfinies. S'il est plus grand, cela prendra plus de temps, mais il y a une meilleure probabilité de succès.
- **Man-in-the-middle :** est lorsqu'un attaquant intercepte les communications entre deux parties, soit pour espionner secrètement,

soit pour modifier le trafic circulant entre les deux. Les attaquants peuvent utiliser des attaques MITM pour voler des informations de connexion ou des informations personnelles, espionner la victime ou saboter les communications ou corrompre les données.

- **Phishing attack :** implique l'envoi par le cybercriminel d'e-mails qui semblent provenir de sources fiables. Le but de ce type d'attaque est d'acquérir des informations sensibles ou de les inciter à faire quelque chose.
- **Buffer overflow :** Les attaquants exploitent les problèmes de dépassement de mémoire tampon en écrasant la mémoire d'une application. Cela modifie le chemin d'exécution du programme, déclenchant une réponse qui endommage les fichiers ou expose des informations privées. Par exemple, un attaquant peut introduire du code supplémentaire, envoyant de nouvelles instructions à l'application pour accéder aux systèmes informatiques**.**

I.3.3.4 Attaque de déni de service :

- **Ping of Death :** est un type d'attaque par déni de service (DoS) dans lequel un attaquant tente de planter, de déstabiliser ou de geler l'ordinateur ou le service ciblé en envoyant des paquets mal formés ou surdimensionnés à l'aide d'une simple commande ping.

- **TCP SYN Flood** : Une attaque SYN flood (attaque semi-ouverte) est un type d'attaque par déni de service (DDoS) qui vise à rendre un serveur indisponible au trafic légitime en consommant toutes les ressources du serveur disponibles. En envoyant à plusieurs reprises des paquets de demande de connexion initiale (SYN), l'attaquant est en mesure de submerger tous les ports disponibles sur une machine serveur cible, ce qui oblige le périphérique ciblé à répondre au trafic légitime de manière lente ou pas du tout.

- **Attaque smurf :** L'intrus inonde la victime par un grand nombre de Pings pour le saturer. La technique la plus utilisée s'exécute en deux phases : Dans la première phase, l'intrus usurpe l'adresse IP de la cible pour l'utiliser comme adresse source, dans la deuxième phase, l'intrus envoie un maximum de Pings en diffusion directe à destination d'un réseau contenant un grand nombre d'hôtes. L'adresse source étant transformée en l'adresse de la victime. Si le routeur d'entrée accepte de faire passer les diffusions directes, tous les hôtes du réseau vont répondre à l'adresse source qui est l'adresse de la victime.

- **IpSpoofing :** L'usurpation d'adresse IP est la création de paquets IP (Internet Protocol) qui ont une adresse source modifiée afin de masquer l'identité de l'expéditeur, d'usurper l'identité d'un autre système informatique, ou les deux. Il s'agit d'une technique souvent utilisée par de mauvais acteurs pour invoquer des attaques DDoS contre un appareil cible ou l'infrastructure environnante.

I.4 Mécanisme de sécurité :

I.4.1 Two-Way Authentification (2FA) : [6]

Est un type spécifique d'authentification multi facteur (MFA) qui renforce la sécurité d'accès en exigeant deux méthodes (également appelées facteurs d'authentification) pour vérifier l'identité. Ces facteurs peuvent inclure quelque chose qu'on sait comme :

- un nom d'utilisateur et un mot de passe

- quelque chose qu'on possède.

- une application pour Smartphone

2FA protège contre le phishing, l'ingénierie sociale et les attaques par force brute de mot de passe et sécurise les connexions contre les attaquants exploitant les informations d'identification faibles ou volées.

I.4.1.1 Les Types de 2FA :

- **SMS 2FA :** L'authentification SMS à deux facteurs valide l'identité d'un utilisateur en envoyant un code de sécurité par SMS à son appareil mobile. L'utilisateur entre ensuite le code dans le site Web ou l'application auprès de laquelle il s'authentifie.

- **TOTP 2FA :** La méthode Time-Based One Time Password (TOTP) 2FA génère une clé localement sur le périphérique auquel un utilisateur tente d'accéder. La clé de sécurité est généralement un code QR que l'utilisateur scanne avec son appareil mobile pour générer une série de chiffres. L'utilisateur entre ensuite ces numéros dans le site Web ou l'application pour y accéder. Les codes d'accès générés par les authentificateurs expirent après un certain laps de temps et un nouveau sera généré la prochaine fois qu'un utilisateur se connectera à un compte. TOTP fait partie de l'architecture de sécurité Open Authentication (OAUTH).

- **Push-Based 2FA :** La 2FA basée sur les push améliore les SMS et la TOTP 2FA en ajoutant des couches de sécurité supplémentaires, tout en améliorant la facilité d'utilisation pour les utilisateurs finaux. La 2FA basée sur les push confirme l'identité d'un utilisateur avec plusieurs facteurs d'authentification que

les autres méthodes ne peuvent pas. Duo Security est le principal fournisseur de 2FA push.

- **U2F Tokens :** Les jetons U2F sécurisent l'authentification à deux facteurs en utilisant un port USB physique pour valider l'emplacement et l'identité d'un utilisateur qui tente de se connecter. Pour utiliser un jeton U2F, un utilisateur insère le jeton dans son appareil et appuie sur le bouton situé en haut de l'appareil. Une fois le jeton activé, l'utilisateur entre son code PIN et accède à ses comptes.

I.4.2 Antivirus :

Un logiciel antivirus est un type de programme conçu et développé pour protéger les ordinateurs contre les logiciels malveillants tels que les virus, les vers informatiques, les logiciels espions, les botnets, les rootkits, les enregistreurs de frappe et autres. Les programmes antivirus fonctionnent pour analyser, détecter et supprimer les virus de votre ordinateur. Il existe de nombreuses versions et types de programmes antivirus sur le marché. Cependant, l'objectif principal de tout programme antivirus est de protéger les ordinateurs et de supprimer les virus une fois détectés.

I.4.3 VPN : [7]

I.4.3.1 Définition:

Un VPN est un tunnel (peut 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.

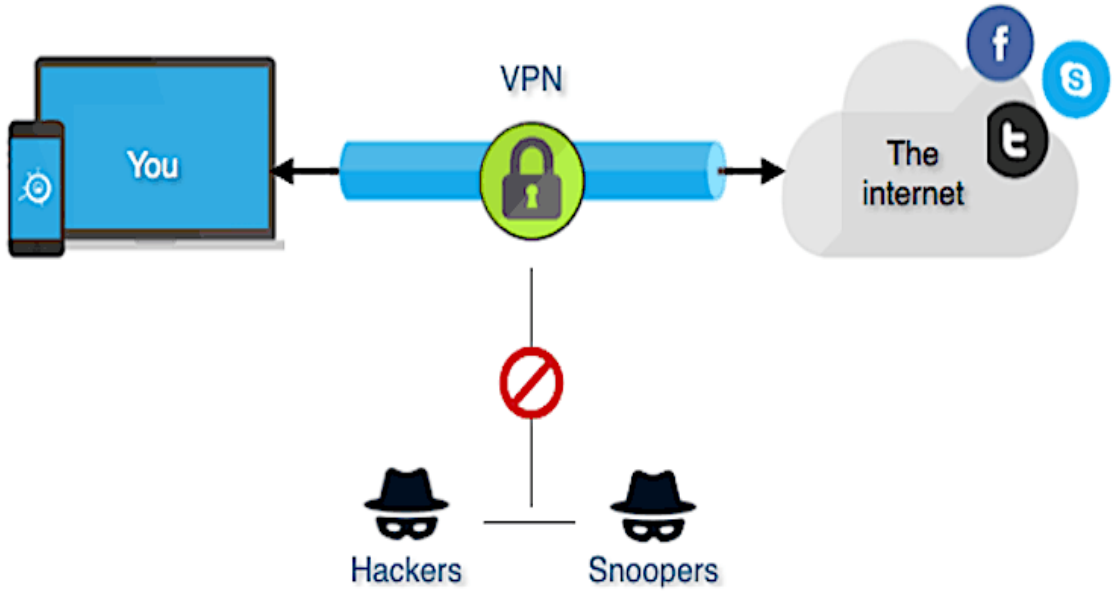

Figure 2 VPN

I.4.3.2 Sécurité VPN:

Les protocoles VPN sont conçus pour sécuriser les données sur des réseaux publics. La sécurité est assurée grâce à:

- **Confidentialité des données** : les données sont cryptées, ce qui les rend illisibles pour les utilisateurs du réseau public.
- **Intégrité des données :** les données sont signées numériquement, de sorte que le destinataire puisse reconnaître que les données ont été modifiées pendant la transmission. Cela ne chiffre pas les données, mais utilise une valeur de hachage des données pour déterminer si le contenu a été modifié. La valeur de hachage de toutes les données restera la même tant que le contenu des données n'est pas modifié.
- **Protection contre la réexécution** : garantit que les mêmes données ne peuvent pas être envoyées plusieurs fois. Dans une attaque de relecture, un attaquant capture puis renvoie les données, telles que les informations de connexion, pour tenter d'accéder au serveur. Grâce à l'utilisation du séquençage, les protocoles VPN s'assurent que les données ne sont pas relues.

- **Authentification de l'origine des données :** utilise des techniques d'authentification pour garantir l'origine des données transmises et reçues. Il s'assure que l'émetteur et le récepteur sont fiables.

I.4.3.3 Les types de configuration VPN:

Le VPN peut être configuré comme une connexion d'hôte à hôte, une connexion VPN de site à site et un VPN d'accès distant :

- **Host-to-host connection :**permet à un hôte individuel connecté à Internet, d'établir une connexion VPN à un autre hôte à l'autre extrémité. Avec une connexion VPN d'hôte à hôte, les deux appareils doivent être en mesure d'établir et de comprendre le protocole VPN utilisé. Les deux appareils doivent disposer du logiciel pour crypter les paquets et encapsuler les paquets avant de les envoyer via Internet. Le périphérique à l'autre extrémité doit utiliser le même protocole pour supprimer les informations d'encapsulation et décrypter les paquets.

- **Avec le site-to-site VPN**, il y aura une collection d'ordinateurs à différents endroits. N'importe quel ordinateur sur n'importe quel emplacement peut communiquer en toute sécurité avec tout autre ordinateur sur un emplacement différent. Plutôt que d'exiger une configuration VPN sur chaque ordinateur, on installe un seul appareil sur chaque emplacement qui agit comme un serveur de passerelle. Ce serveur VPN accepte les paquets non chiffrés de l'emplacement privé, et chiffre et encapsule ce paquet pour l'envoyer sur Internet au serveur VPN de destination. À l'autre emplacement, le serveur VPN supprime ensuite les informations de chiffrement et transfère les données vers le réseau privé à l'autre extrémité. Ainsi, avec cette configuration, seuls les serveurs VPN doivent être configurés pour le protocole VPN.

- **Remoteaccess VPN** remplace le serveur d'accès à distance. Dans ce cas, tout client peut établir une connexion VPN au site distant. L'ordinateur client doit pouvoir établir la connexion VPN avec le serveur qui se trouve sur le bord du réseau privé. Ce serveur est souvent appelé concentrateur VPN et son travail consiste à accepter plusieurs connexions VPN via Internet, avec plusieurs clients. Chaque client est configuré avec un logiciel qui lui permet de chiffrer les paquets, et le concentrateur VPN est configuré pour autoriser ou rejeter les connexions des utilisateurs, puis pour supprimer le chiffrement avant de transférer les paquets vers le réseau privé.

I.4.3.4 Les Protocoles de tunnelisationVPN:

Le protocole de tunneling ou le protocole VPN identifie les méthodes utilisées par les appareils pour établir la connexion VPN et crypter les données. Les clients exécutant Windows OS peuvent utiliser 4 protocoles VPN différents. Ces protocoles diffèrent par le type de cryptage et de protection des données qu'ils offrent. Ces protocoles sont:

- **PPTP (Point-to-Point Tunneling Protocol)** : Technologie VPN Microsoft, et c'était l'un des protocoles VPN d'origine. C'est le moins sécurisé de tous les protocoles VPN. Il ne nécessite pas d'infrastructure à clé publique (PKI), qui utilise des certificats.

- **L2TP / IPSec (Layer 2 Tunneling Protocol / IP Security) :** protocole standard ouvert. Fournit le plus haut niveau de sécurité en utilisant des certificats numériques. Il nécessite l'accès à l'infrastructure du service de certificats.

- **IPSec (Internet Protocol Security)** : fournit l'authentification et le cryptage. Il peut être utilisé avec L2TP ou seul en tant que solution VPN. Il peut crypter tout trafic pris en charge par le protocole IP.

- **SSTP (Secure Socket Tunneling Protocol)** : utilise le protocole SSL. Il peut être utilisé pour passer presque tous les pare-feu qui autorisent l'accès à Internet. C'est quelque chose qui n'est pas vrai avec les autres protocoles VPN.

- **IKEv2 (Internet Key Exchange version 2)** : disponible à partir de Windows 7. Il prend en charge IPv6, la nouvelle fonctionnalité de reconnexion VPN, ainsi que l'authentification par certificat de carte à puce. IKEv2 prend en charge l'authentification de l'origine des données, l'intégrité des données, la protection contre la relecture et la confidentialité des données.

I.4.4 Les Pare-feu (Firewall): [8]

I.4.4.1Définition :

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.4.4.2 Fonctionnement d'un pare-feu :

- Un pare-feu contient un ensemble de règles définies par l'administrateur de sécurité
- Chaque règle définit les caractéristiques d'un paquet (adresse IP, port, type de paquet, protocole, …) et la décision à appliquer à ce paquet (autoriser, rejeter, loguer, …).
- Chaque paquet traversant le pare-feu est soumis à l'ensemble de ces règles.
- Si le paquet correspond à une règle, la décision associée est prise (soit il est autorisé, soit il est rejeté).
- Si un paquet ne correspond à aucune règle, le paquet est soit autorisé ou rejeté en fonction de la politique de sécurité par défaut.
- La politique de sécurité par défaut peut :
    - Autoriser tous les paquets qui ne correspondent à aucune règle de pare-feu
    - Interdire tout paquet ne correspondant à aucune règle de pare-feu
    -

I.4.4.3 Types de Pare-feux :

- **Pare-feu à filtrage de paquet sans état (stateless firewall) :**
    - C'est le type de pare-feu le plus basique
    - Tout pare-feu doit fournir au minimum cette fonctionnalité
    - Les paquets sont filtrés à bases d'informations contenues dans la couche IP et TCP ou UDP de chaque paquet.
- **Pare-feu à filtrage de paquet avec état (statefull firewall) :**
    - En plus des règles de filtrage, le pare-feu tient à jour une table des connexions déjà établies
    - Quand un paquet fait partie d'une nouvelle connexion, les règles de filtrage sont appliquées. Une fois admis, le paquet est rajouté à la table de connexions.
    - Quand un paquet fait partie d'une connexion existante, les flags du paquet sont consultés pour vérifier s'il fait partie d'une connexion cohérente.
- **Pare-feu applicatif :**
    - En plus des informations de couches 3 et 4, les informations de couche application sont aussi examinées pour décider du sort du paquet.
    - Le module qui gère les informations de couche application est appelé proxy.

- Le serveur proxy peut autoriser, refuser ou tracer ces connexions à des fin d'analyse ou pour toute fin utile (contrôle parental, juridique, …etc.)
- Il peut aussi soumettre l'information pour un scan contre les virus, spams, …etc.

- **Pare-feu NAT et NATP :**
  - NAT pour Network Address Translation et NATP pour NAT and port AddressTranslation.
  - Les adresses IP et les numéros de ports des hôtes du réseau interne sont remplacées par d'autres adresses et ports au niveau du pare-feu.
  - La topologie et le plan d'adressage interne sont cachés pour l'extérieur.
  - Vis-à-vis de l'extérieur, toutes les machines internes sont vues comme une seule machine qui plus-est le pare-feu.

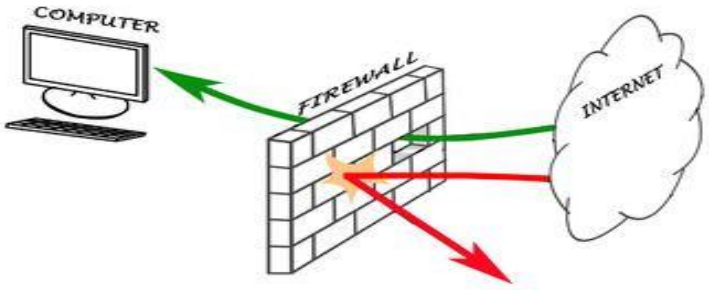

Figure 3 Pare-Feu

I.4.5 Les systèmes de détection d'intrusions « réseau » (NIDS) : [9] M

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 génére 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.4.6 Pots de miels : [9]M

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.5 Conclusion:

Dance ce chapitre, nous avons traité la sécurité informatique, qui assure la sureté des systèmes, des informations et des données personnelles. Nous avons particulièrement décrit les objectifs de la sureté informatique, les attaques et comment l'assurer. Dans qui ce suit, nous abordons la cryptographie qui est une technique fondamentale sur laquelle repose notre travail.

Chapitre II

LaCryptographie

Introduction :

Ce chapitre définit la cryptographie qui permet d'atteindre ses objectifs. On présente les types de cryptographie et ainsi que lesméthodes de cryptage les plus connues, en évoquant également la signature numérique et les fonctions de hachage.

II.1 La Cryptologie :

II.1.1 Définition de la Cryptologie : [10]

La cryptologie, l'étude des crypto systèmes (la science des codes secrets), peut être subdivisée en deux branches :

- Cryptographie : qui est le domaine où on cherche à protéger le secret
- Cryptanalyses : qui est le domaine où on cherche à retrouver le message d'origine sans connaitre exactement tout le procédé

II.1.2 Définition de la Cryptographie : [11]

La cryptographie est l'art et la science de créer un crypto système capable d'assurer la sécurité de l'information.
La cryptographie traite de la sécurisation effective des données numériques. Elle fait référence à la conception de mécanismes basés sur des algorithmes mathématiques qui fournissent des services fondamentaux de sécurité de l'information. On peut considérer que la cryptographiecomme l'établissement d'une grande boîte à outils contenant différentes techniques dans les applications de sécurité.

II.1.3 Définition de la Cryptanalyse : [11]

La cryptanalyse est la branche sœur de la cryptographie et les deux coexistent. Le processus cryptographique aboutit au texte chiffré pour la transmission ou le stockage. Cela implique l'étude de mécanisme cryptographique avec l'intention de les briser. La cryptanalyse est également utilisée lors de la conception des nouvelles techniques cryptographiques pour tester leurs forces de sécurité.

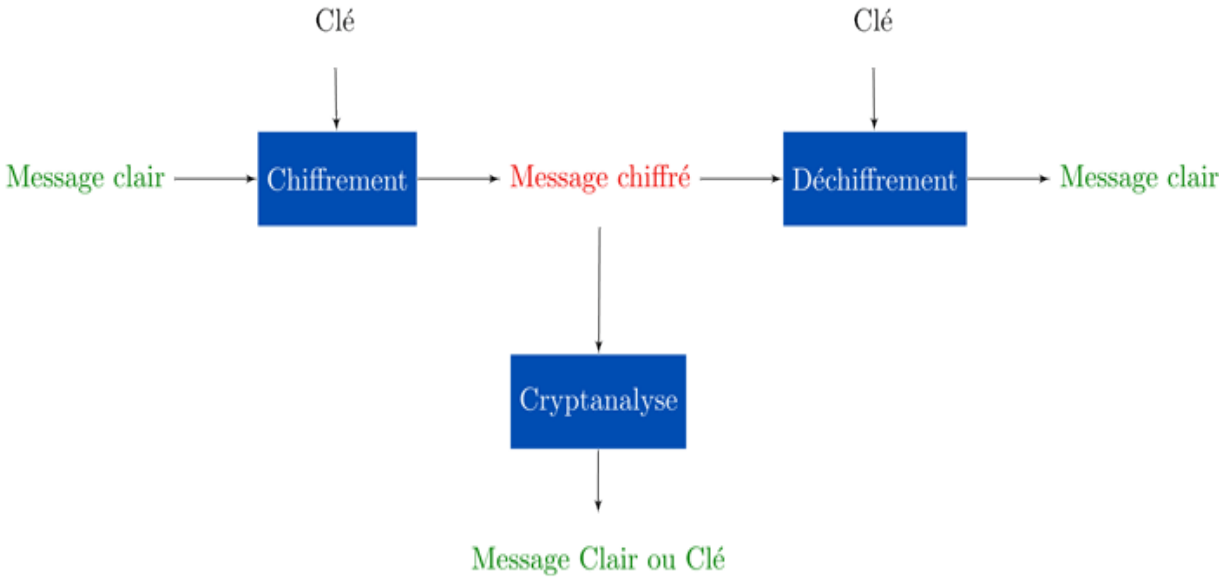

Figure 4Schéma de la cryptographie

II.1.4 L'objectif de la Cryptographie : [12]

L'objectif : L'objectif principal de la cryptographie est de sécuriser les données importantes sur le disque dur ou lors de leur passage sur un support qui peut ne pas être sécurisé lui-même. Habituellement, ce support est un réseau informatique.

Les Services :

- **Confidentialité :** S'assurer que personne ne peut lire le message à l'exception du destinataire prévu. Les données sont gardées secrètes de ceux qui n'ont pas les informations d'identification appropriées, même si les données transitent par un support non sécurisé.

- **Intégrité :** Assurer au destinataire que le message reçu n'a en aucun cas été modifié par rapport à l'original.

- **Authentification** : L'authentification fournit l'identification de l'expéditeur. Il confirme au destinataire que les données reçues n'ont été envoyées que par un expéditeur identifié et vérifié.

    - Le service d'authentification a deux variantes:
        - **L'authentification de message** identifie l'expéditeur du message sans aucun égard au routeur ou au système qui a envoyé le message.
        - **L'authentification d'entité** est l'assurance que les données ont été reçues d'une entité spécifique, par exemple un site Web particulier.

Outre l'expéditeur, l'authentification peut également fournir une assurance sur d'autres paramètres liés aux données telles que la date et l'heure de création / transmission.

- **Non-Répudiation :** Un mécanisme pour prouver que l'expéditeur a vraiment envoyé ce message.

II.1.5 Terminologie de la Cryptographie : [12]

La cryptographie utilise des différents termes qui vont être définis comme suit :

- **Texte en clair :** Ce sont les données à protéger lors de la transmission.
- **Algorithme de cryptage** : Il s'agit d'un processus mathématique qui produit un texte chiffré pour tout texte brut et clé de chiffrement donnés. Il s'agit d'un algorithme cryptographique qui prend du texte en clair et une clé de chiffrement en entrée et produit un texte chiffré.
- **Texte chiffré** : Il s'agit de la version brouillée du texte en clair produit par l'algorithme de cryptage en utilisant une clé de cryptage spécifique.
- **Algorithme de décryptage :** C'est un processus mathématique qui produit un texte en clair uniquepour n'importe quel texte chiffré et clé de déchiffrement. Il s'agit d'un algorithme cryptographique qui prend un texte chiffré et une clé de déchiffrement en entrée, et produit un texte en clair.
- **Clé de cryptage :** C'est une valeur connue de l'expéditeur. L'expéditeur entre la clé de chiffrement dans l'algorithme de chiffrement avec le texte en clair afin de calculer le texte chiffré.
- **Clé de décryptage :**C'est une valeur connue du récepteur. La clé de déchiffrement est liée à la clé de chiffrement, mais ne lui est pas toujours identique. Le récepteur entre la clé de déchiffrement dans l'algorithme de déchiffrement avec le texte chiffré afin de calculer le texte en clair.

II.2 Les niveaux d'attaques : [13]

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

- ***L'attaque par cryptogramme:*** (par message chiffré seulement) où la cryptanalyse ne connait qu'un ensemble de message chiffrés, il peut soit retrouver seulement les messages en clair, soit retrouver la clé. 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 :*** où 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 connues (en-têtes de paquets, champs communs à tous les fichiers d'un type donné,).

- ***L'attaque à message en clair choisi :***où 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 :*** à 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 clé. Ce type d'attaques est principalement utilisé contre les systèmes à clé publique, pour retrouver la clé privée

II.3 Les Types de la Cryptographie : [14]

Les méthodes de cryptographie se composent de deux grandes parties : La cryptographie symétrique et la cryptographie asymétrique à base de clés.
Avant de les aborder, il faudrait définir la notion de clé qui 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.

  On distingue aussi deux types de cryptographie selon le format des données :
  - **Cryptage par bloc** :
    - L'information à crypter est décomposée en blocs de taille fixe.
    - Chaque bloc est crypté séparément.
    - Chiffrement lent et nécessite beaucoup de mémoire et de puissance de calcul.
  - **Cryptage à la volée :**

- L'information à crypter est traitée comme un flux continu de bits ou de caractères.
- plus rapide et nécessite moins de ressources. Il est souvent réalisé par matériel.
- Utilisé dans les applications de diffusion audio et vidéo.

II.3.1 La cryptographie symétrique: [13]

Est un système de cryptage dans lequel l'expéditeur et le destinataire d'un message partagent une seule clé commune qui est utilisée pour crypter et décrypter le message.

- L'utilisation de l'algorithme est également connue sous le nom d'algorithme à clé secrète ou parfois appelée algorithme symétrique
- Une clé est un élément d'information (un paramètre) qui détermine la sortie fonctionnelle d'un algorithme cryptographique ou d'un chiffrement.
- La clé de chiffrement et de déchiffrement du fichier doit être connue de tous les destinataires. Sinon, le message ne pourrait pas être déchiffré par des moyens conventionnels.

II.3.1.1 Caractéristiques de la crypto symétrique :

- ✓ 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.

II.3.1.2 Désavantage de la crypto symétrique : [13]

Les systèmes à clé symétrique sont plus simples et plus rapides; leur principal inconvénient est que les deux parties doivent en quelque sorte échanger la clé de manière sécurisée et la garder en sécurité après cela.
La gestion des clés a provoqué un cauchemar pour les parties utilisant la cryptographie à clé symétrique. Ils s'inquiétaient de savoir comment transmettre les clés en toute sécurité à tous les utilisateurs afin que le décryptage du message soit possible. Cela a permis à des tiers d'intercepter les clés en transit pour décoder les messages top-secrets. Ainsi, si la clé était compromise, tout le système de codage était compromis et un «secret» ne resterait plus un «secret».
C'est pourquoi la «cryptographie à clé publique» a vu le jour.

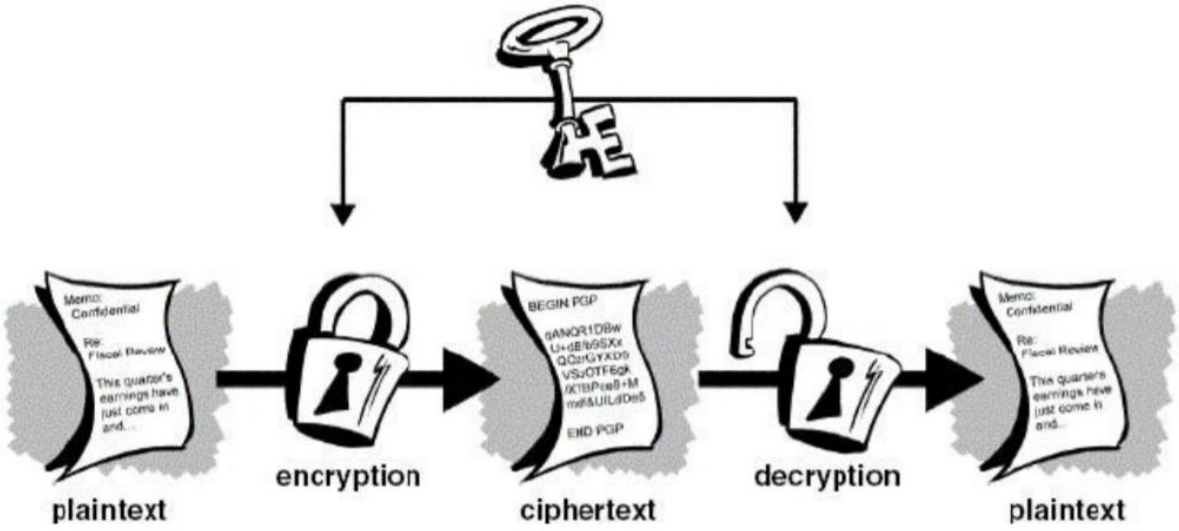

Figure 5Cryptographie symétrique

II.3.1.3 Exemples de la Crypto symétrique : [13]

- **Data Encryption Standard (DES)** :
  - Le Data Encryption Standard a été publié en 1977 par le National Bureau of Standards des États-Unis.
  - DES utilise une clé de 56 bits et mappe un bloc d'entrée de 64 bits de texte en clair sur un bloc de sortie de 64 bits de texte chiffré. 56 bits est une clé plutôt petite pour la puissance de calcul d'aujourd'hui.

- **Triple DES : Triple** DES était la réponse à de nombreuses lacunes du DES. Puisqu'il est basé sur l'algorithme DES, il est très facile de modifier le logiciel existant pour utiliser Triple DES. Il présente également l'avantage d'une fiabilité éprouvée et d'une longueur de clé plus longue qui élimine la plupart des attaques de raccourci qui peuvent être utilisées pour réduire le temps nécessaire pour interrompre le DES.

- **Advanced Encryption Standard (AES) :**
  - est une norme de cryptage adoptée par le gouvernement américain. La norme comprend trois chiffrements par blocs, AES-128, AES-192 et AES-256, adoptés à partir d'une plus grande collection publiée à l'origine sous le nom de Rijndael.
  - Chaque chiffrement AES a une taille de bloc de 128 bits, avec des tailles de clé de 128, 192 et 256 bits, respectivement. Les chiffrements AES ont été analysés de manière approfondie et sont maintenant utilisés dans le monde entier, comme ce fut le cas avec son prédécesseur, le Data Encryption Standard (DES)

- **IDEA :**
  - L'algorithme international de chiffrement des données a été développé en 1991.
  - Il utilise une clé de 128 bits pour crypter un bloc de 64 bits de texte en clair en un bloc de 64 bits de texte chiffré.
  - La structure générale d'IDEA est très similaire à DES, il effectue 17 tours, chaque tour prenant 64 bits d'entrée pour produire une sortie de 64 bits, en utilisant des clés par tour générées à partir de la clé de 128 bits.

II.3.2 La Cryptographie Asymétrique : [13]

La cryptographie asymétrique, également connue sous le nom de cryptographie à clé publique, fait référence à un algorithme cryptographique qui nécessite deux clés distinctes, dont l'une est privée et l'autre publique. La clé publique est utilisée pour crypter le message et la clé privée est utilisée pour décrypter le message.

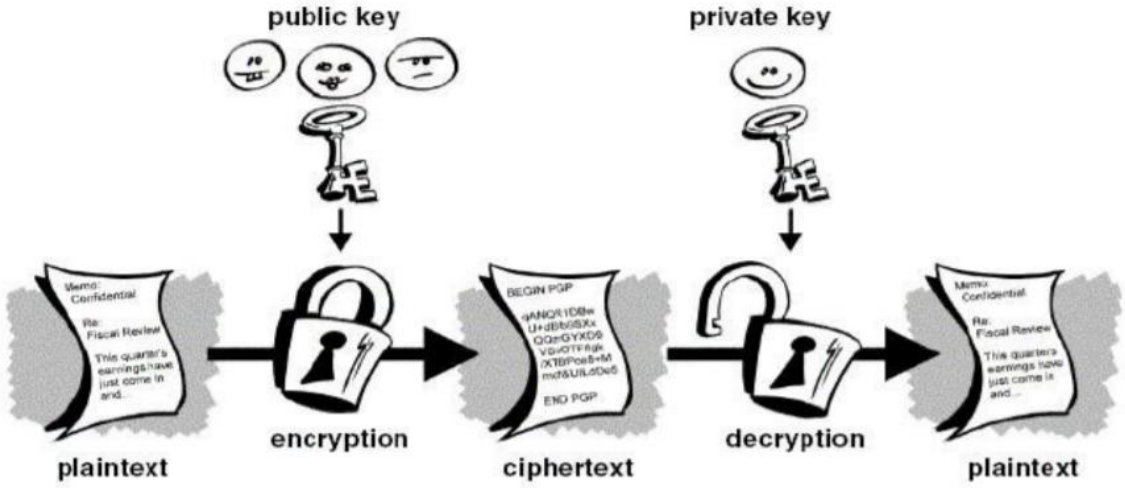

Figure 6Cryptographie Asymétrique

II.3.2.1 Les Principes de la cryptographie asymétrique :

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.

II.3.2.2 Exemples de la cryptographie asymétrique :

- **RSA :**
  - RoneRivest, Adi Shamir et Leonard Adelman.
  - Repose sur la complexité de factorisation de grands nombres.
  - Adapté à la fois pour la cryptographie et la signature numérique.
- **ElGamal :**
  - Tahar El Gamal
  - Repose sur la difficulté de calcul du logarithme discret.
  - utilisé pour l'échange de clés secrètes.
- **Rabin** :
  - Michael Rabin.
  - Repose sur la factorisation comme le RSA

.

II.4 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).

L'utilisation de ces fonctions a pour but d'assurerl'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.5 La signature Numérique :

II.5.1 Définition : [16]

Une signature numérique est un cachet d'authentification électronique, crypté, sur des informations numériques telles que des messages électroniques, des macros ou des documents électroniques. Une signature confirme que les informations proviennent du signataire et n'ont pas été modifiées.

II.5.2 Fonctionnement de la signature numérique : [17]

Supposons que Alice a un message à envoyer à James n'est pas confidentiel; cependant, James devrait savoir que le message vient vraiment d'Alice. Alicepeut utiliser sa clé privée pour chiffrer le message, puis, James peut utiliser la clé publique d'Alicepour déchiffrer le message. Cependant, cela n'est pas possible si le message est trop long, car le message crypté serait doublé et c'est une opération qui prend du temps. Pour résoudre ce problème, Alicepeut envoyer le message à une fonction de hachage à sens unique pour produire un résultat (résumé de message) de même taille toujours (en fonction de l'algorithme).

Caractéristiques de la fonction de hachage à sens unique:

- Extrêmement rapide.
- Impossible de créer un message à partir d'un résumé.

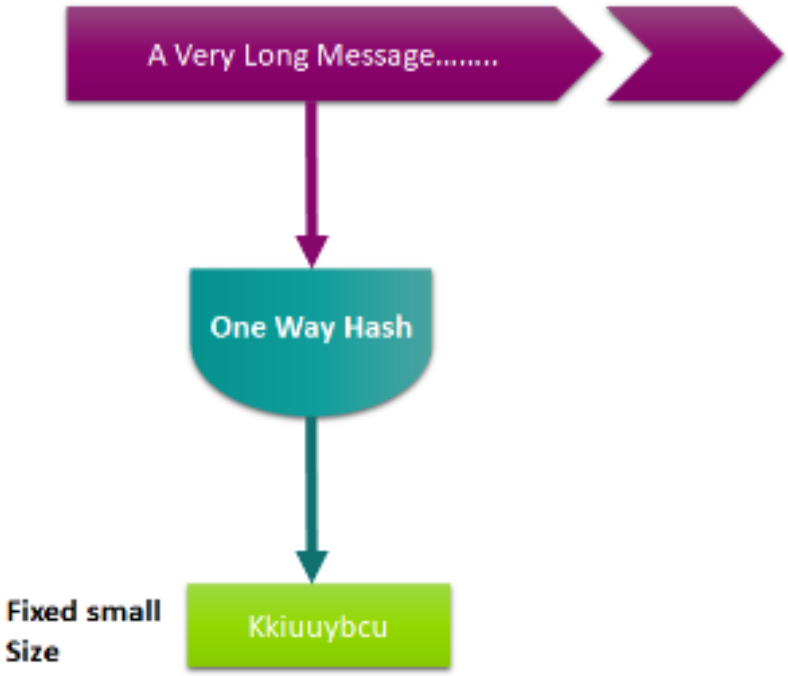

Figure 7Schéma Hachage

Maintenant, pour que James soit sûr qu'Alice est celle qui prétend être, il peut utiliser sa clé privée pour crypter le résumé du message, puis envoyer le message et le résumé à James.
James peut déchiffrer le résumé à l'aide de la clé publique d'Alice, calculer le résumé à partir du message d'origine et enfin comparer les résumés de message.

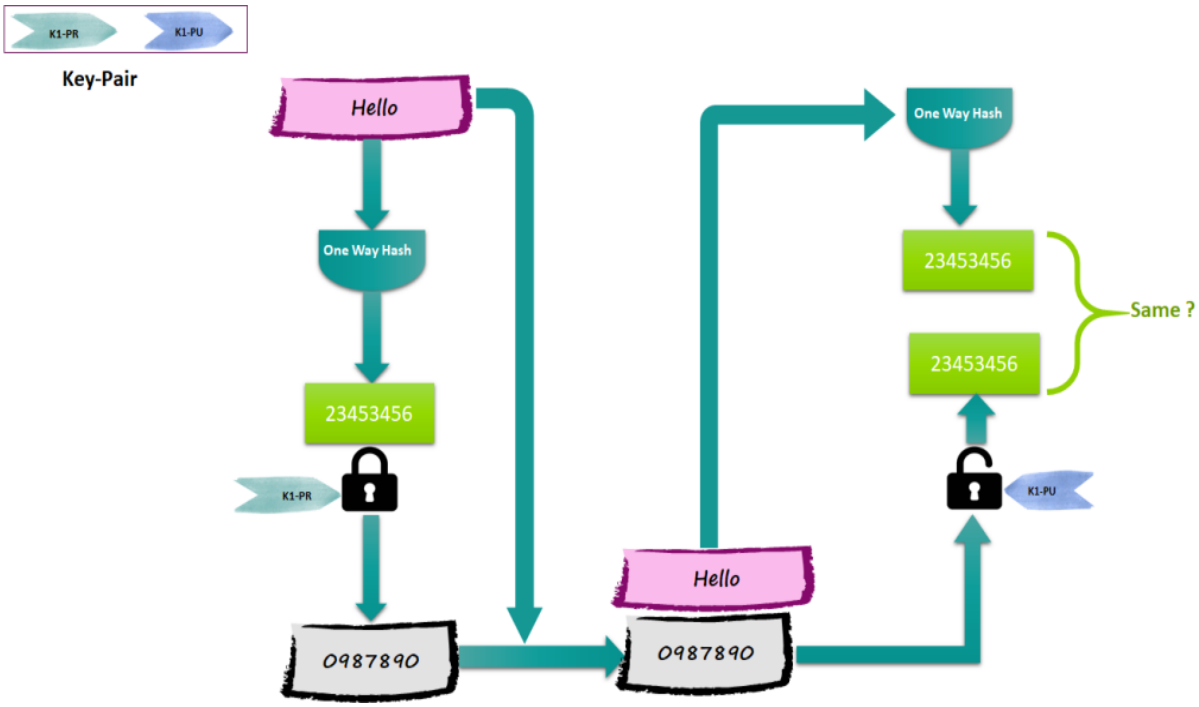

Figure 8Schéma Signature Digital

Le résumé chiffré est appelé «signature numérique», et tout le processus de calcul du résumé, puis de chiffrement, est appelé «signature du message».

II.6 Conclusion

Dans ce chapitre, nous avons traité la cryptographie qui remplit les objectifs suivants de la sécurité informatique : la confidentialité, l'intégrité et l'authenticité des données. Nous avons particulièrement décrit des algorithmes cryptographiques, les fonctions de hachages, les signatures numériques. Dans ce qui suit, nous abordons l'algorithme de cryptage asymétrique RSA et l'algorithme de signature DSA pour l'intégrité et l'authentification du messagesur lequel repose notre travail de mémoire.

Chapitre III

DSA & RSA

III Introduction :

Àprésent, dans ce chapitre, nous nous intéresserons à l'Algorithme de Signature Digitale DSA et l'Algorithme RSA (Rivest-Shamir-Adleman) où nous soulignerons le fonctionnement et les motivations qui nous ont poussés à choisir ces algorithmes.

III.1 DSA :

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.1.2 L'algorithme DSA :

III.1.2.1 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.

Entrée : taille de la clé (choisir longueur **I** et **J** divisibles par 64) //exprimée en bits.
Sortie : la clé publique (**p, q, g, y**) et la clé privée (**x**)

1. Choisir un nombre premier **p** de longueur **I**,
2. Choisir un nombre premier **q** de longueur **J**, de telle façon que **q-1= qz**, où **z** est un entier.
3. Choisir **h**, avec **1<h<p-1** de manière que **g=hzmode p >1**,
4. Générer aléatoirement un **x**, avec **0 < x < q**,
5. Calculer **y** = g^x**modp,**
6. **Returng,y et x**

III.1.2.2 Signature :

Entrée : k tel que 1 < k < q. **Sortie :** la signature (**r,s**)
1. Calculer **r = (g k mod p) mod q,**
2. Si **r = 0** recommencer avec un autre **k**.
3.Calculer **s= (H(m) + r.x)** $\boldsymbol{k}^{-1}$**mod q**, où **H(m)** est le résultat d'un hachage cryptographique, par exemple avec SHA-256, sur le message m.
4. Si s **= 0** recommencer avec un autre **k**
5. **Return r,s**

III.1.2.3 Vérification :

Entrée : r,s : si 0< r < q ou 0 < s <q. **Sortie :** (**v**)
1. Calculer **w =** $\boldsymbol{s}^{-1}$**mod q**
2. Calculer **u1 = H(m). W mod q**
3.Calculer **u2 = r. w mod q.**
4. Calculer (**v =**$\boldsymbol{G^{u1}}$**.** $\boldsymbol{Y^{u2}}$ **modp**)**modq**
5. **Return True si v=r else Return False**

III.1.3 Fonctionnement du DSA :

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

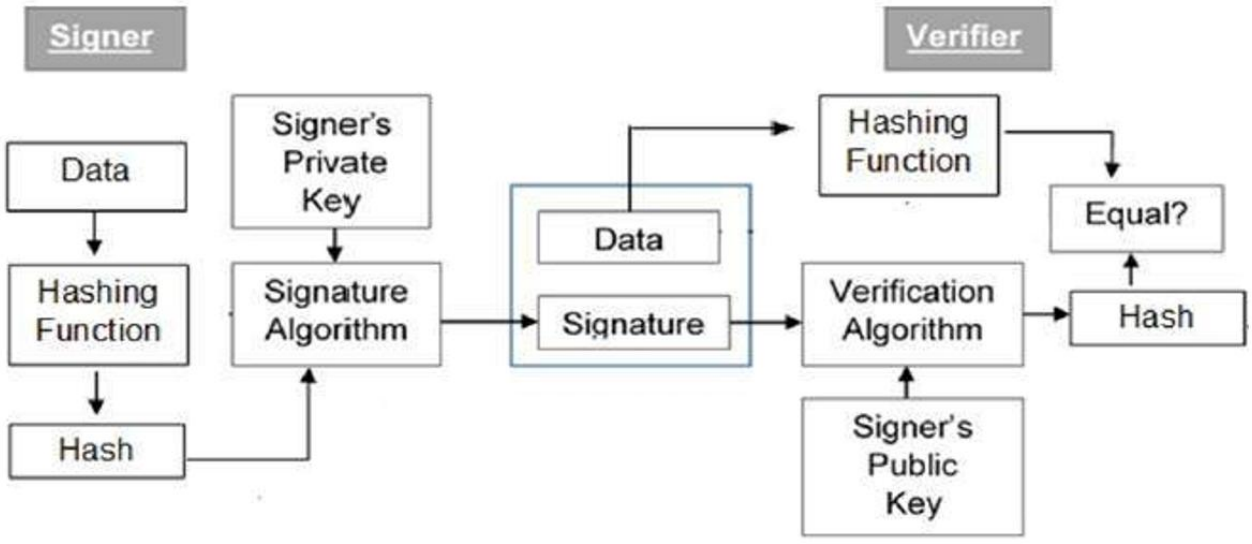

Figure 9Fonctionnement DSA

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 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.1.4 LA différence entre DSA et autres signatures numériques [20] :

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.1.5 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 demessage** : 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, on peut 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.2 RSA :

L'algorithme a été introduit par trois chercheurs en 1976, nommés Ronald Rivest, Adi Shamir et Leonard Adleman, et est basé sur le cryptage des messages à l'aide d'une exponentiation modulaire, et le partage des informations publiques et des clés privées. Dans un algorithme de

cryptage symétrique, il existe une clé secrète qui est utilisée à la fois pour le cryptage et décrypter les données. Le cryptage symétrique est relativement rapide, mais cette méthode de cryptage peut poser des problèmes en termes de distribution de la clé secrète, puisqu'elle doit être utilisée à la fois pour le cryptage et le décryptage des données.
Le RSA est aujourd'hui utilisé dans toute une série de navigateurs web, de services de chat et de courrier électronique, de VPN et d'autres moyens de communicationde l'Union européenne. Il est couramment utilisé simplement parce que les gens font confiance à l'algorithme pour fournir un cryptage suffisamment bon pour leurs objectifs, et il a été prouvé qu'il était sûr.

III.2.2 L'algorithme RSA :

L'algorithme RSA fait appel à des clés privées et publiques. La clé publique peut être connue et publiée par n'importe qui, car elle est utilisée pour chiffrer les messages du texte en clair au texte chiffré. Les messages qui sont cryptés avec cette clé publique spécifique ne peuvent toutefois être décryptés qu'avec la clé privée correspondante. Le processus de génération de la clé de l'algorithme RSA est ce qui le rend si sûr et fiable aujourd'hui, car il contient un niveau de complexité élevé par rapport aux autres algorithmes cryptographiques.

III.2.2.1 Génération des clés :

Entrée : taille de la clé //exprimée en bits. **Sortie :** la clé publique (N,E) et la clé privée (N,D)
1. Prendre deux nombres premiers **p** et **q** suffisamment grands (de taille à peu près égale). 2. Calculer **N = p*q,** 3. Calculer **α= (p-1) (q-1)**, 4. Choisir un nombre **E**tel que ***1 <E <α*** etle **PGDC *(e, α)=1***, 5. Prendre un nombre **E** qui n'a aucun facteur en commun avec **α**, 6. Calculer **D** tel que **D*E**modα=**1**.
7. **Return**N, E, D

III.2.2.2 Chiffrement :

Entrée : (N, E) et M // la clé publique et le texte en clair M avec M∈ [0.N-1]. **Sortie :** C // texte chiffré.
1. Calculer **$C=M^E \bmod N$**
2. **Return**C

III.2.2.3 Déchiffrement :

Entrée : (D, E) et C // la clé privée et le texte chiffré. **Sortie :** M // texte clair.
1. Calculer**$M=C^D \bmod N$**
2. **Return**M

III.3 Sécurité RSA :

La sécurité de RSA repose sur la difficulté que représente la factorisation de grands entiers. Cependant, à mesure que la puissance de traitement augmente et que des algorithmes de factorisation plus efficaces sont découverts, il devient possible de factoriser des nombres de plus en plus élevés. La puissance du chiffrement est directement liée à la taille de la clé. De ce fait, un doublement de la longueur de la clé renforce le chiffrement de façon exponentielle, au détriment toutefois des performances.

Les clés RSA font généralement 1024 ou 2048 bits, mais les experts pensent que les clés de 1024 bits pourraient être déchiffrées à brève échéance. C'est la raison pour laquelle l'administration et le secteur privé commencent à adopter des clés d'une longueur minimale de 2048 bits. A moins d'une percée imprévue en informatique quantique, denombreuses années devraient s'écouler avant que des clés plus longues soient nécessaires.

III.4Conclusion:

Dans ce chapitre, nous nous sommes intéressés à l’algorithme DSA dans lebut d’assurer l’intégrité et l’authentification et à l’algorithme RSA pour la confidentialité. Dans cette partie nous avons expliqué le fonctionnement des deux algorithmes et l’importance de ces derniers dans notresolution qui sera détaillée et expliquée dans ce qui suit.

Chapitre IV

Outils de développement et Implémentation

Introduction :

Dans ce chapitre, on va parler des technologies et des outils utilisés, et l'implémentation des deux algorithmes DSA et RSA,(signature et cryptographie respectivement) dans une application Android de chat « messages cryptés et signés de bout en bout (END-to-END) »

IV.1 Système D'Exploitation Android :

IV.1.1 HISTORIQUE [22]

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

IV.1.2 Architecture d'Android [23]

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 :

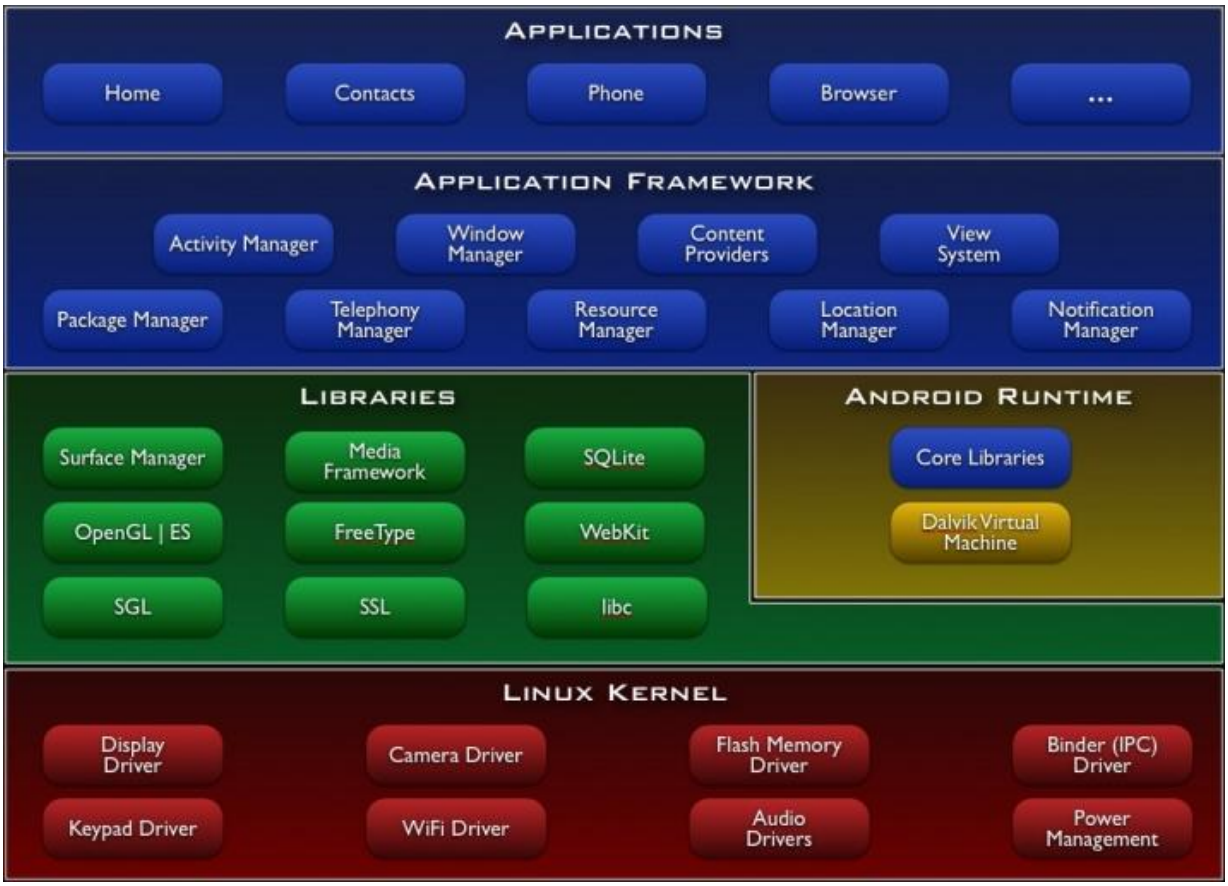

Figure 10 Architecture Android

IV.1.2.1 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.1.2.2 Deuxième niveau:

- ✓ **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 de l'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.

✓ ***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 est 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.1.2.3 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 packetmanager 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.1.2.4 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.1.3 Environnement d'exécution Android:

L'environnement d'exécution d'Android est la machine virtuelle Dalvik, qui est incorporée 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:

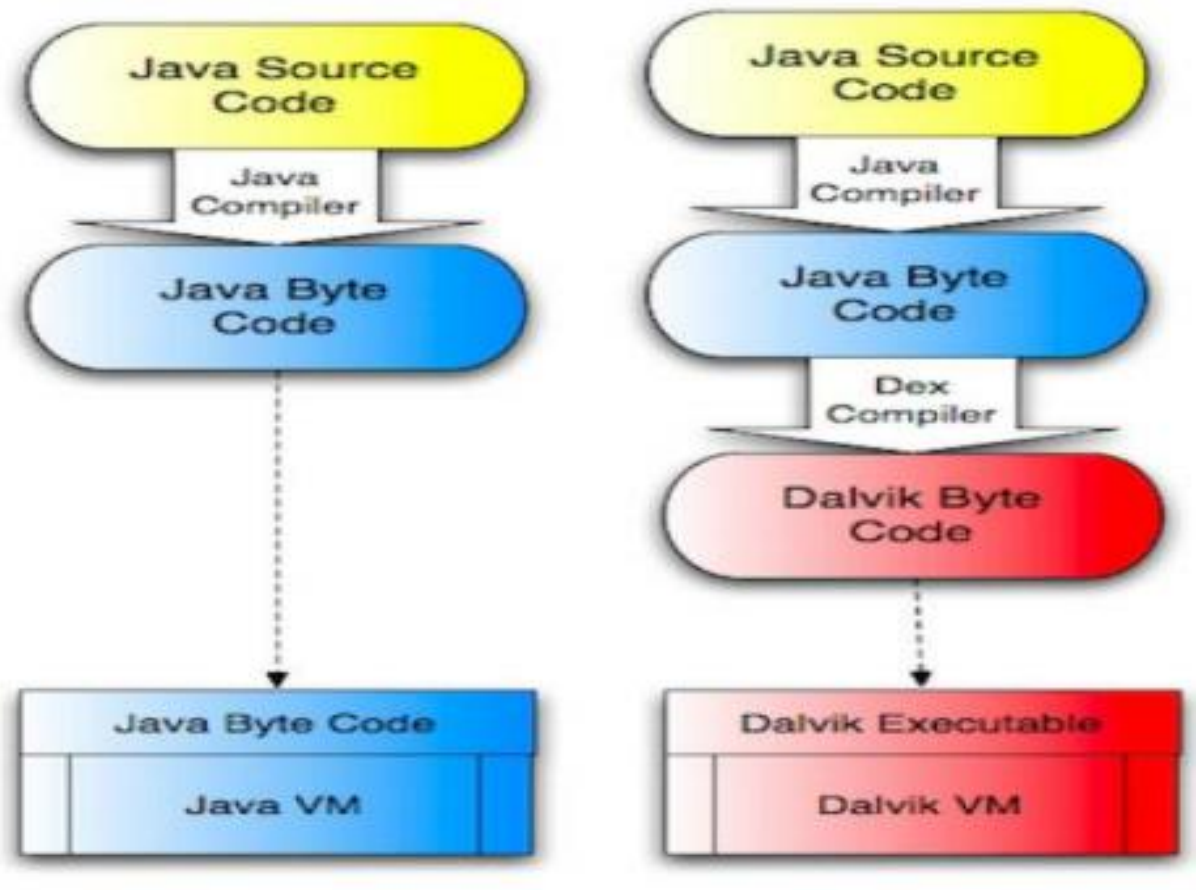

Figure 11 Environnement Android

IV.1.4 Les composants d'une application Android :

IV.1.4.1 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.1.4.2 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.1.4.3 Intent /BroadcastReceiver :

- ✓ **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.
- ✓ **BroadcastReceiver :**
  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.1.4.4 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.1.5 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 permettre de suivre l'état d'une activité au fur et mesure de son existence dans le système Android.

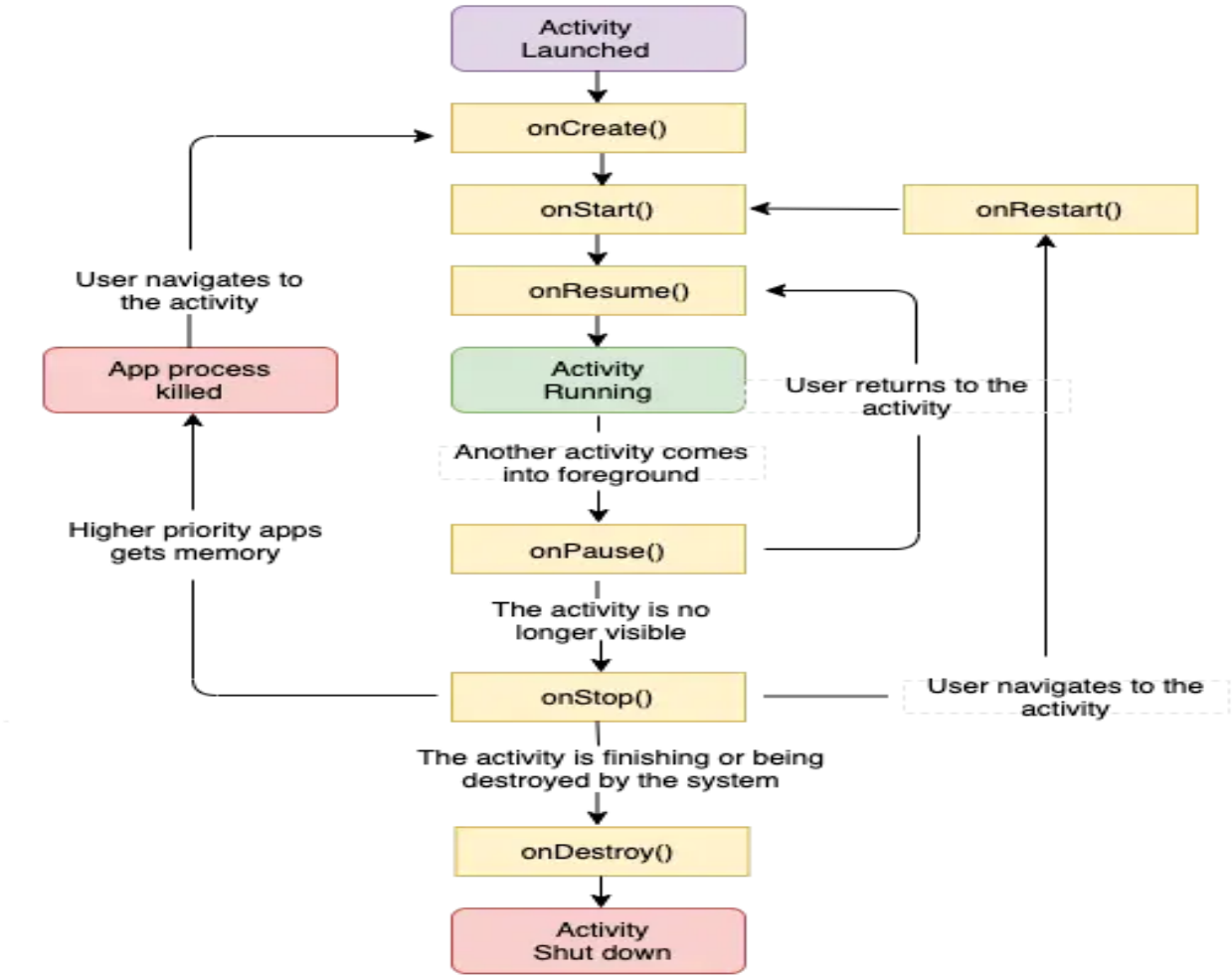

Figure 12 Android Life Cycle

IV.1.5.1 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.1.5.2 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.1.6 Les versions d'Android [18] :

Android 1.0 G1 (2008) :
Lors de son lancement, Android 1.0 proposait 35 applications via Android Market. Son Google Maps utilisait le GPS et le Wi-Fi du téléphone et un navigateur Android y était intégré.

Android 1.5 Cupcake (2009) :
Cupcake fut la première mise à jour majeure d'Android avec l'introduction de widgets pour l'écran d'accueil, un clavier virtuel, l'enregistrement vidéo pour l'appareil photo et une option copier/coller pour le navigateur web.

Android 2.0 Eclair (2009) :
Avec Android Eclair, les utilisateurs Microsoft ont pu bénéficier de la prise en charge d'Exchange, de plusieurs comptes Google, d'un moteur de recherche dans les SMS, des gestes multitouch et d'améliorations au niveau de l'appareil photo avec un flash et un zoom numérique.

Android 2.2 Froyo (2010) :
Froyo a apporté Flash Player 10.1 qui a permis aux smartphones de pouvoir lire des contenus vidéo et audio en streaming. Le flash de l'appareil photo fonctionne avec la vidéo, la compatibilité Bluetooth est élargie et l'on peut transformer son mobile en point d'accès Wi-Fi.

Android 2.3 Gingerbread (2011) :
Arrivée de la technologie Near Field Communication (NFC) qui permet aux terminaux de se connecter avec d'autres appareils compatibles à proximité immédiate. Les appels vidéo sont désormais disponibles via la caméra frontale et l'OS hérite d'un gestionnaire de téléchargement.7

Android 3.0 Honeycomb (2011) :
Cette mise à jour fut la première dédiée uniquement aux tablettes Android. Elle introduit la prise en charge des graphismes en 3D, le chat vidéo avec Google Talk, le partage de connexion via Bluetooth et un mode plein écran dans la galerie photo.

Android 4.0 IceCream Sandwich (2011) :
IceCream Sandwich a réconcilié les versions smartphones et tablette de l'OS en ajoutant au passage, la reconnaissance faciale pour déverrouiller le mobile, des sms préprogrammés pour décliner les appels téléphoniques et des effets en live pour l'enregistrement vidéo.

Android 4.1 Jelly Bean (2012) :

Grâce au « Projet Butter », Jelly Bean gagne en performance et en fluidité. Les notifications s'enrichissent, le navigateur Chrome est adopté par défaut, les widgets sont redimensionnables et Google Now est préinstallé.

Android 4.4 KitKat (2013) :
KitKat apporte des émoticônes au clavier Google, l'impression à distance avec Google Cloud Print et une utilisation plus parcimonieuse de la mémoire vive pour pouvoir tourner sur des smartphones d'entrée de gamme.

Android 5.0 Lollipop (2014) :
Grosse évolution esthétique avec l'introduction du « Material Design » et son interface aplanie. Les notifications s'affichent désormais sous forme de bannières sur l'écran de verrouillage et d'alertes pop-up.

Android 6.0 Marshmallow (2015) :
Arrivée du mode Doze qui préserve l'autonomie de la batterie, prise en charge native des lecteurs d'empreintes digitales, de l'USB-C et de l'Ultra HD pour les applications.

Android 7.0 Nougat (2016) :
Avec Nougat, il devient enfin possible de fermer toutes les applications depuis l'aperçu d'un seul geste. La mise à jour permet de modifier le teint des émoticônes, le menu des réglages rapides d'enrichit et l'OS peut fonctionner avec la plateforme de réalité virtuelle Google Daydream.

Android 8.0 Oreo (2017) :
Android Oreo renforce le multitâche, améliore le copier/coller, la sécurité et la gestion de la batterie.

Android 9.0 Pie (2018) :
La mise à jour d'Android 9.0 a rendu les smartphones Android plus rapides et économes en énergie. Pour y arriver, Android a eu recours à l'intelligence artificielle pour suggérer des applications et des raccourcis qui anticipent les besoins de l'utilisateur, charger des aperçus ciblés sur l'information recherchée sans avoir à ouvrir une application ou une page web.

Android 10 Q (2019) :
Avec Android 10, Google abandonne les dénominations de desserts pour ses mises à jour système. Il faudra se contenter dorénavant d'un "simple" chiffre à partir de cette dixième version. Cette mise à jour, sortie en septembre 2010, a notamment mis l'accent sur les fonctionnalités de confidentialité et de sécurité.

Android 11 Beta (2020) :

Cette nouvelle version se concentre toujours autant sur les fonctionnalités de confidentialité et de sécurité. Le contrôle des autorisations est plus poussé, un nouveau système de capture d'écran fait son apparition tout comme le système de bulles de notification.

IV.1.7 Taux d'utilisation des versions Android [19] :

Android Platform Version (API Level)	Distribution (as of April 10, 2020)
Android 4.0 "Ice Cream Sandwich" (15)	0.2%
Android 4.1 "Jelly Bean" (16)	0.6%
Android 4.2 "Jelly Bean" (17)	0.8%
Android 4.3 "Jelly Bean" (18)	0.3%
Android 4.4 "KitKat" (19)	4%
Android 5.0 "Lollipop" (21)	1.8%
Android 5.1 "Lollipop" (22)	7.4%
Android 6.0 "Marshmallow" (23)	11.2%
Android 7.0 "Nougat" (24)	7.5%
Android 7.1 "Nougat" (25)	5.4%
Android 8.0 "Oreo" (26)	7.3%
Android 8.1 "Oreo" (27)	14%
Android 9 "Pie" (28)	31.3%
Android 10 (29)	8.2%

Figure 13 Android Versions

IV.2 Firebase :[20]

IV.2.1 Introduction :

Depuis quelques années, les applications mobiles sont très utilisées aussi bien à des fins professionnelles que personnelles.

De nouvelles applications ne cessent de faire leur apparition et tous les systèmes mobiles sont concernés (Android, iOS, Windows Phone).Aujourd'hui, on trouve également de nombreuses applications dédiées au web.

Pour les créer, les développeurs ont souvent recours à des plateformes de développement d'applications comme FIREBASE.

IV.2.2 Plateforme Firebase :

Firebase est le nom d'une plateforme mobile de Google qui facilite la création de back-end à la fois scalable et performant. En d'autres termes, il s'agit d'une plateforme qui permet de développer rapidement des applications (Coté Back-end) pour mobile et pour le web.

L'objectif de la création de Firebase en 2011 par James Tamplin et Andrew Lee est d'éviter aux professionnels et aux particuliers de s'engager dans un processus complexe de création et de maintenance d'une architecture serveur.

De plus, la plateforme peut être exploitée par plusieurs utilisateurs en même temps sans connaître un quelconque bug. La praticité est également au rendez-vous grâce à ses fonctionnalités intuitives. Depuis le rachat de la plateforme par Google en 2014, Firebasesdks a connu de nombreuses améliorations et n'a de cesse de satisfaire ses utilisateurs.

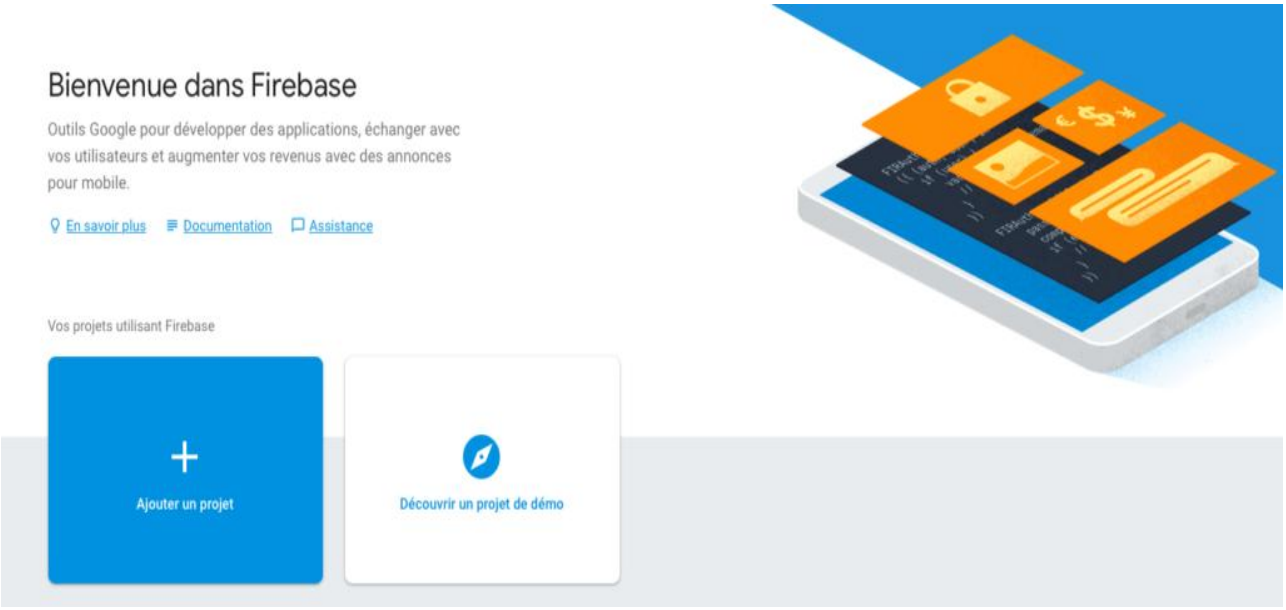

Figure 14Firebase

IV.2.3 Atouts de Firebase :

Dans Firebase, on trouve des API intuitives regroupées dans un SDK unique. Ces API, en plus de faire gagner du temps, permettent de réduire le nombre d'intégrations qu'on doit gérer par le biais de l'application.

On profite ainsi d'une offre sur mesure ainsi qu'une intégration étroite entre les différents produits qu'on exploite. Étant donné que Firebase utilise l'infrastructure de Google, la plateforme n'a aucun mal à s'adapter à l'évolution de votre application. Ainsi, On peut développer l'applicationFirebase dans les meilleures des conditions, d'autant plus que la plateforme met à disposition une solution complète, évolutive et boostée par Google.

IV.2.4 Services Firebase :

Firebase met à disposition différents services, on détaillera 3 de ces services qu'on a utilisés dans notre projet :

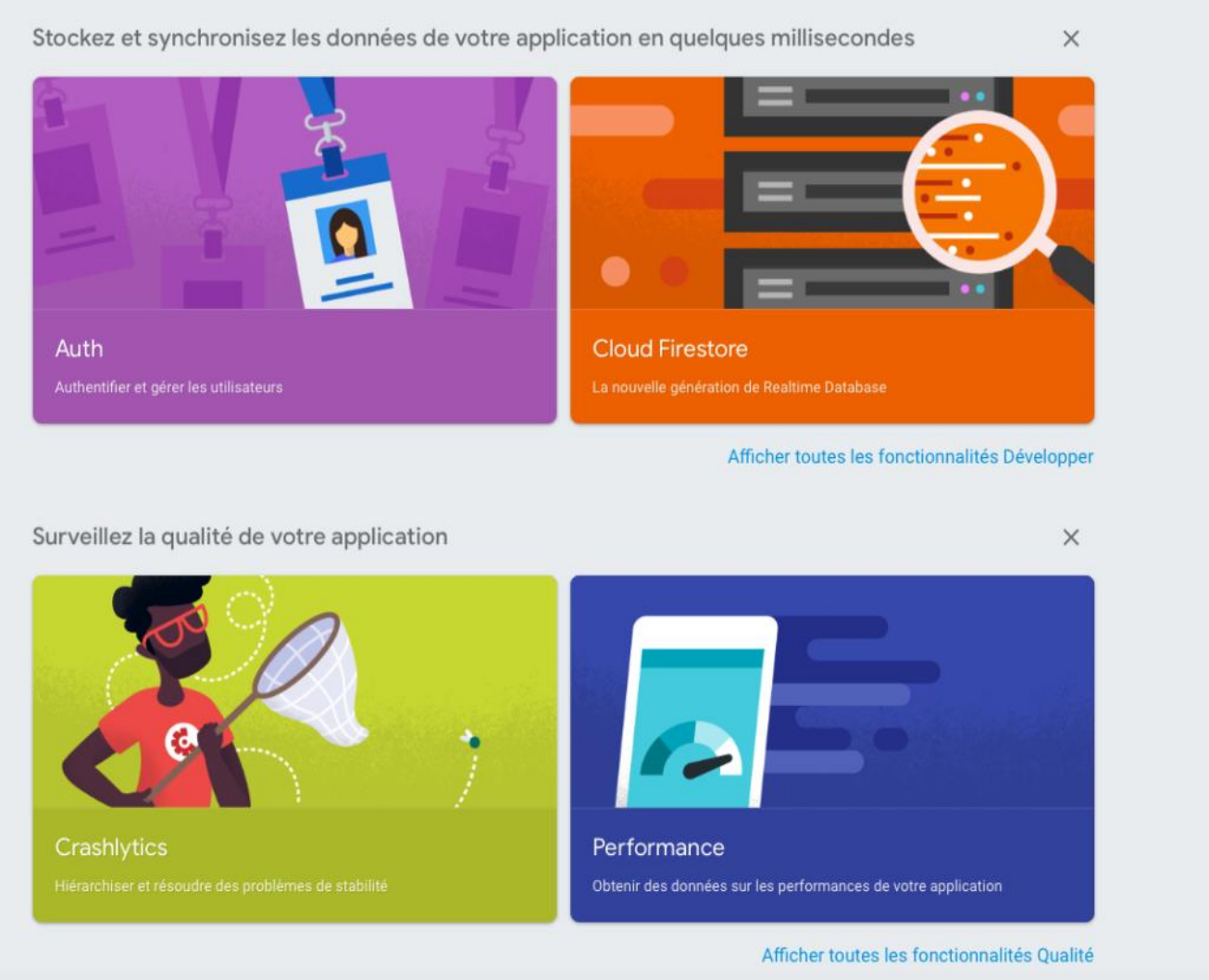

Figure 15 Services Firebase

IV.2.4.1 RealTimeDataBase :

FirebaseRealtimeDatabase n'est autre qu'une base de données NoSQL, bénéficiant d'un hébergement « Cloud » et permettant le stockage et la synchronisation de données de vos utilisateurs. Les développeurs peuvent gérer cette base de données en temps réel.

FirebaseRealtimeDatabase est capable de fournir à votre application la valeur des données et les mises à jour appliquées sur ces dernières avec à une simple API. Grâce à la synchronisation en temps réel, les utilisateurs de votre application peuvent consulter leurs données depuis n'importe quel terminal (sur le web ou depuis leur mobile). Notez que cette base de données est livrée avec des SDK mobiles et web et permet la création d'amplifications sans utiliser de serveurs.

Quand les utilisateurs passent en mode hors ligne, les SDK de base de données en temps réel utilisent le cache pour enregistrer les modifications. Quand l'appareil est en ligne, les données locales connaissent une synchronisation automatique. Dernière chose, FirebaseDatabase peut rejoindre l'authentification Firebase pour un processus d'authentification plus simple et plus rapide.

IV.2.4.2Firebase Authentification :

Cet outil fournit des SDK faciles à exploiter, des services back-end ou encore des bibliothèques d'interface utilisateur. Ces bibliothèques permettent d'authentifier les utilisateurs de l'application.

En général, la configuration manuelle d'un système d'authentification prend plusieurs mois. Par la suite, il faut engager une équipe pour la maintenance. Avec Firebase, les choses se déroulent autrement. La configuration du système ne prend que quelques heures même s'il faut prendre en charge des opérations délicates comme la fusion de comptes.

Plusieurs méthodes s'offrent à nous pour authentifier les utilisateurs notamment l'exploitation de :

- ➢ Leur e-mail et de leur mot de passe ;
- ➢ Google Cloud ;
- ➢ Twitter ;
- ➢ Facebook ;
- ➢ Numéro de téléphone ;

Grâce à Firebase Authentification, la création de systèmes d'authentification sécurisés devient un véritable jeu d'enfant. Cet outil permet également à vos utilisateurs finaux de profiter d'une meilleure expérience d'intégration et de connexion.

IV.2.4.3 Firebase Cloud Storage :

Firebase Storage permet de partager ou encore de stocker du contenu produit par les utilisateurs comme les images, les vidéos ou encore les fichiers audio. C'est une solution de stockage d'objets puissante qui se démarque par sa simplicité et son caractère économique.

Firebase propose d'autres fonctionnalités comme : ML kits (MachineLearning),FirebaseAnalytics,ect … .

IV.2.5Avantages Firebase:

Firebase se démarque d'autres plateformes de développement d'application grâce notamment à ses nombreuses fonctionnalités. Voici quelques-uns des avantages à exploiter via l'exploitation de cette plateforme :

- ✓ **Développement rapide d'application :**
  Firebase renferme des API intuitives rassemblées dans un SDK unique. Avec ces API, onpeut développer rapidement et efficacement même des applications haut de gamme. La plateforme renferme également des outils permettant d'attirer de nombreux utilisateurs et par là même d'augmenter les revenus. Pour ce faire, ona juste à combiner les fonctionnalités Firebase qui répondent aux attentes, et qui correspondent à aux besoins.

- ✓ **Plus besoin d'infrastructures complexes :**
  Grâce à Firebase, onn'a plus besoin de mettre en place des infrastructures complexes en interne ou encore d'exploiter un tableau de bord complexe pour concevoir des applications répondant aux attentes des utilisateurs. Quand bien même ces derniers sont exigeants, Firebase met les outils nécessaires à disposition afin que nous puissions satisfaire nos utilisateurs.

- ✓ **Des décisions raisonnées :**
  La plateforme Firebase intègre une option d'analyse gratuite et illimitée dédiée aux mobiles. Cette option est personnalisable à souhait afin de permettre d'obtenir des résultats correspondant à nos attentes. Avec la fonctionnalité Google Analytics pour Firebase, onpeut trouver des informations importantes sur les utilisateurs.

- ✓ **Exploiter une compatibilité multiplateforme :**
  Grâce à Firebase, tous les besoins seront satisfaits, qu'importe leur nature. Onpeut, par exemple, proposer une application mobile sur différentes plateformes pour ne citer qu'iOS, Android, C++ ou encore JavaScript. L'accès à Firebase peut également se faire via des API REST ou à l'aide des bibliothèques disponibles côté serveur.

- ✓ **Une évolution constante et sûre :**
  Une fois que l'application se retrouve en tête de classement, onn'aura pas à faire une adaptation du côté du serveur ou encore à optimiser la capacité de l'application. Firebase s'occupe de tout automatiquement.
  De plus, onpeut profiter de nombreuses fonctionnalités Firebase gratuites et ceci est valable, qu'importe l'envergure de votre appli.

- ✓ **Un service d'assistance totalement gratuit :**
  Pour profiter d'une assistance gratuite et personnalisée quant à l'utilisation de Firebase, onpeut envoyer un e-mail à l'assistance technique. Sachant également que les experts en développement de Google et l'équipe de Firebase sont particulièrement réactifs sur de nombreux forums en ligne comme GitlHub ou encore StackOverflow.

- ✓ **Une plateforme soutenue par Google :** Firebase est une plateforme créée par Google ainsi elle est soutenue et bien maintenue par Google.

IV.3 Implémentation :

IV.3.1 Introduction :

Dans cette partie 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.

IV.3.2Environnement de développement

IV.3.2.1 Android Studio :

Android Studio est l'environnement de développement intégré (IDE) officiel pour le développement d'applications Android, basé sur IntelliJIDEA. En plus du puissant éditeur de code et des outils de développement d'IntelliJ, Android Studio offre encore plus de fonctionnalités qui améliorent la productivité lors de la création d'applications Android, telles que :

- Un système de compilation flexible basé sur Gradle.
- Un émulateur rapide et riche en fonctionnalités.
- Un environnement unifié où onpeut développer pour tous les appareils Android.
- Appliquer les modifications pour pousser les changements de code et de ressources à l'application en cours sans redémarrer l'application.
- Modèles de code et intégration Git Hub pour aider à créer des fonctionnalités communes et à importer des exemples de code.
- Des outils et des cadres d'essai étendus.
- Outils de peluche pour détecter les performances, la convivialité, la compatibilité des versions et d'autres problèmes.

IV.3.2.2Installation Android Studio :

Pour installer Android Studio, il est nécessaire d'avoir le logiciel du 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

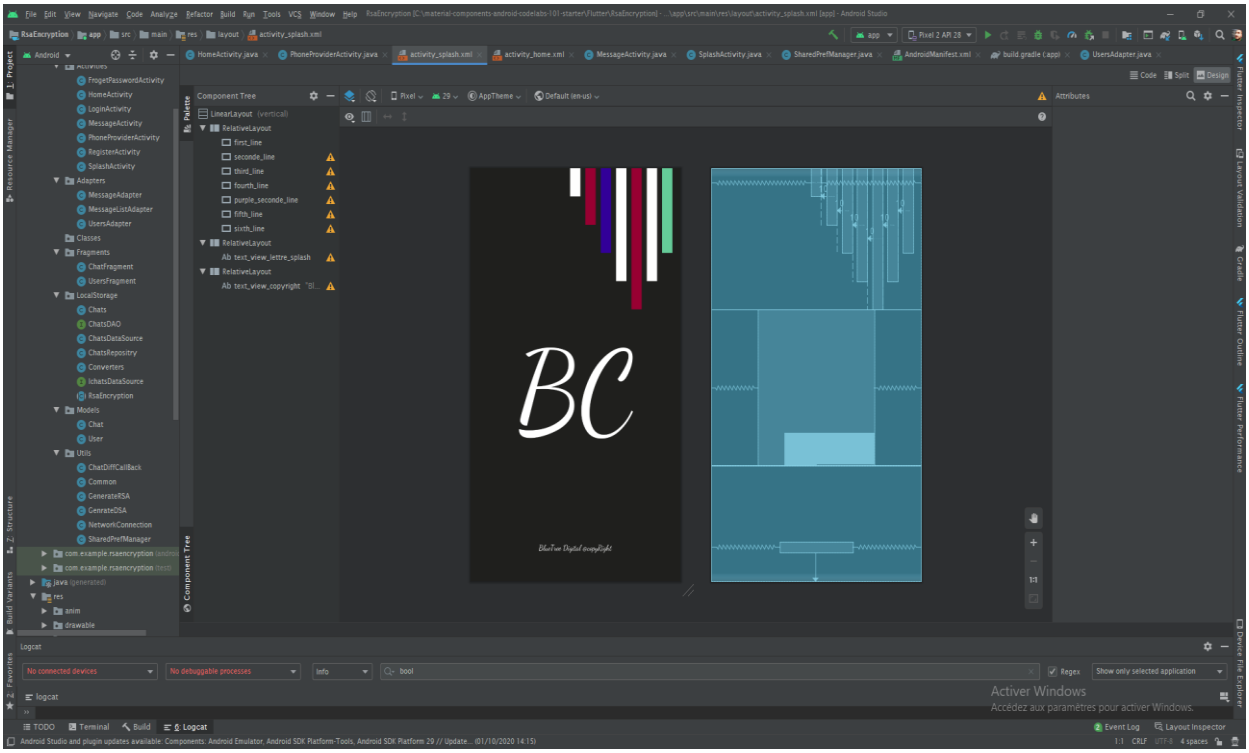

Figure 16 Android Studio

IV.3.2.3Langage 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 à 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 …

IV.3.2.4intégration de Firebase :

- ✓ En haut dans la barre d'outils, on clique sur « tools » :

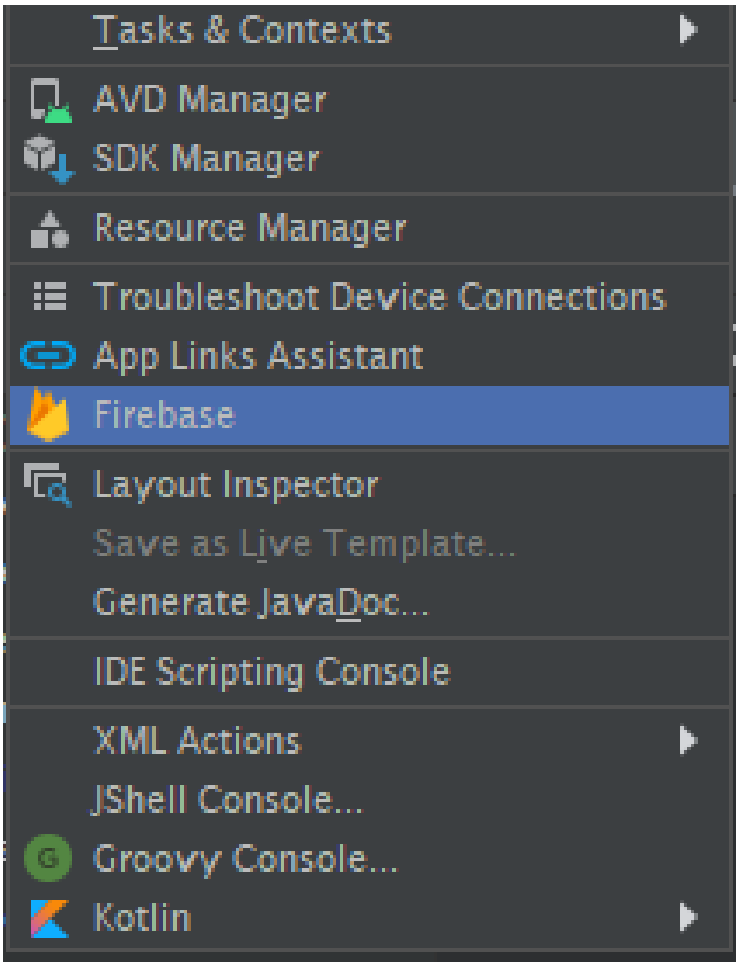

Figure 17 Affichage Tools

- ✓ On cliquant sur Firebase, une seconde fenêtre sera ouverte sur la droite de l'IDE Android Studio :

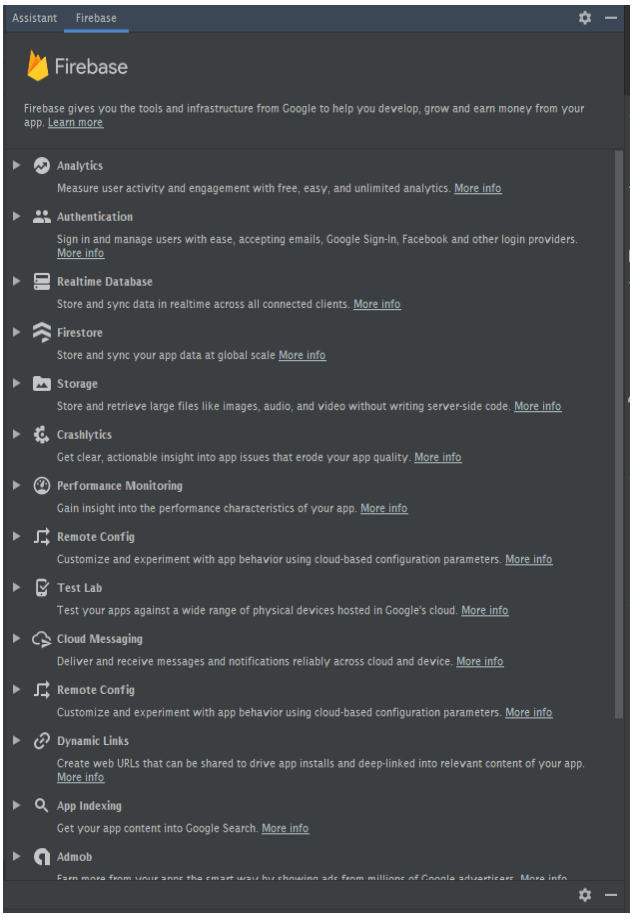

Figure 18Firebase dans Android Studio

- ✓ Afin d'ajouter lesservices qu'on utilisera dans notre projet (Storage,Authentification,RealTimeDatabase),on clique sur « Connect to Firebase » pour ajouter et synchroniser notre application avec Firebase et ajouter les trois fonctionnalités avec le bouton « AddFirebase Authentification / storage / realtimedatabase to yourapp».

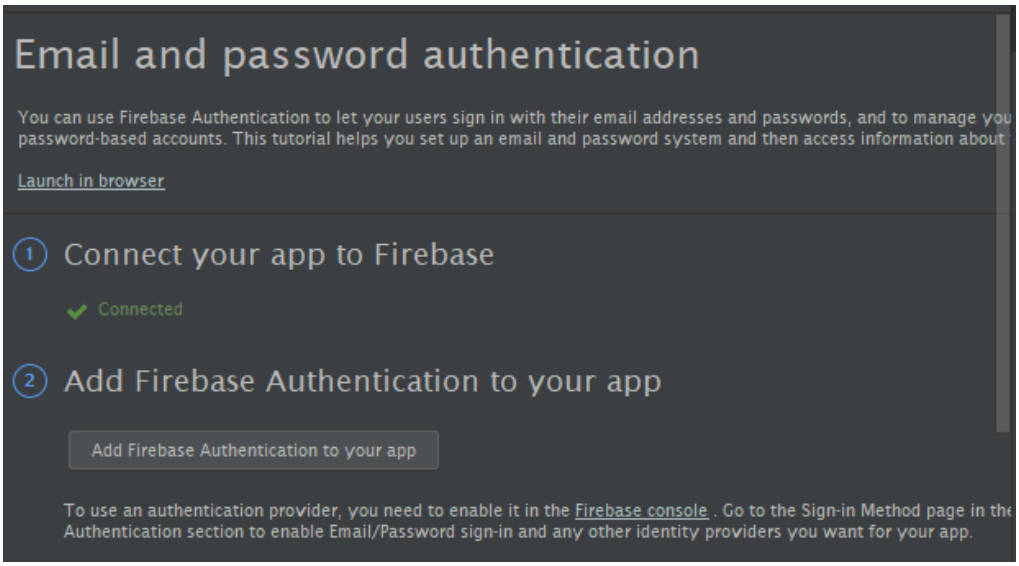

Figure 19Connecter Firebase et ajouter les Services

IV.3.3 Présentation de l'application :

IV.3.3.1 Introduction :

Maintenant qu'on a présenté notre environnement de travail, cette partie est consacrée à la présentation des interfacesde l'application, en expliquant comment les messages sont signés et chiffrés de bout en bout en utilisant des captures d'écrans.

IV.3.3.2 Différents interfaces de l'application :

Notre application prend en charge plusieurs interfaces que nous allons présenter en illustrant le fonctionnement de cette dernière :

IV.3.3.2.2 Interface de démarrage (SplashScreen) :

Cette interface prépare le lancement de l'application, en chargeant les données internes (stockageinterne) et en vérifiant si un utilisateur est déjà connecté ou non.

Figure 20 Interface de démarrage

IV.3.3.2.2 Interface de connexion (Login) :

Après que l'interface de démarrage ait terminé le chargement, l'interface Login est ouverte automatiquement s'il n'y a aucun utilisateur déjà connecté. Cette partie permet à l'utilisateur de se connecter s'il possède déjà un compte. Notonsqu'à la première connexion, il est nécessaire de se connecter via « Email &Password», ensuite l'utilisateur aura le choix d'ajouter un numéro de téléphone qui lui sera lié au compte.
Si l'utilisateur a déjà relié son compte avec un numéro de téléphone, il peut se connecter via son numéro.

Figure 21 Interface Login via Email

IV.3.3.2.3 Interface d'inscription (Register) :

Ici, l'utilisateur peut s'inscrire à l'aide d'une adresse mail et un mot de passe, avec confirmation d'email après enregistrement, tout en insérant son nom et prénom.

Toutes ces informations seront enregistrées dans la base de données. Quant à l'email et le mot de passe, ils vont être générés dans le service «Auth Provider using Email and password» de FireBase

Figure 22 Interface inscription

IV.3.3.2.4Fenêtreprincipal (Home Screen) :

Après Inscription et vérification d'email, l'utilisateur peut à présent se connecter. En se connectant, l'utilisateur va être dirigé à la fenêtre principale. Dans cette partie, on vérifie si c'est la première connexion de l'utilisateur, si c'est le cas ; les paires de clés RSA et DSA seront générées puis stockées dans lamémoire interne du périphérique en utilisant le service SharedPreferences.Les deux

clés publiques (DSA et RSA) seront envoyées et enregistrées au serveur (FireBaseDatabase) dans le document Users correspondant.

Figure 23 Fenêtre Principale

```
//RSA
KeyPairGenerator keyPairGenRSA = KeyPairGenerator.getInstance("RSA");
keyPairGenRSA.initialize( keysize: 1024);
KeyPair pairRSA = keyPairGenRSA.generateKeyPair();

PublicKey publicKey = pairRSA.getPublic();
PrivateKey privateKey = pairRSA.getPrivate();
Log.i( tag: "pkey",  msg: "bind: "+privateKey.toString());
//    Log.i("DONE", "generateRsaKeys: "+publicKey.toString());
SharedPrefManager.getmInstance(this).savePublic(publicKey, key: "");
SharedPrefManager.getmInstance(this).savePrivate(privateKey, key: "");
```

Figure 24RsaGeneration Keys

```
//DSA
String pubkeystr = SharedPrefManager.getmInstance(this).getPublicKeyString("");

KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("DSA");
keyPairGen.initialize( keysize: 1024);
KeyPair pairDSA = keyPairGen.generateKeyPair();

PrivateKey privKeyDSA = pairDSA.getPrivate();
PublicKey pubkeyDSA = pairDSA.getPublic();
```

Figure 25 Code génération DsaKeypairGeneration

- ✓ La figure ci-dessous,représente les documents «Users» après le processus d'inscription et de génération des clés publiques et privées (DSA, RSA).

Figure 26 Document Utilisateurs (Users)

- **Pkey** : représente la clé publique RSA.
- **PkeyDSA** : représente la Clé publique DSA.

IV.3.3.5 Interface discussion :

Cette partie est le noyau de l'application, puisque c'est ici que les messages serontsignés et cryptés puis envoyés de bout en bout au récepteur.
Le mécanisme d'envoi et de réception d'un message est comme suit :

- avant que Bob chiffre et envoie un message à Alice, il doit d'abord hacher le message puis le signer en utilisant sa clé privéeDSA, puis il fusionne le message signé avec le même message en clair avec une séparation entre les deux en utilisant le caractère « ///». Quand cette dernière est terminée, le message doit être crypté en utilisant la clé publique RSA d'Alice puis Bob envoie le message à Alice dans la RealTimeDatabase de Firebase.

```
//Sign message :
String MyPrivDSA = SharedPrefManager.getmInstance(this).getPrivateKeyDSA("");
Log.i( tag: "MessageActivity",  msg: "priv key dsa is : "+MyPrivDSA);
byte[] MybinsDSA = java.util.Base64.getMimeDecoder().decode(MyPrivDSA);
KeyFactory MykeyFactoryDSA = KeyFactory.getInstance("DSA");
EncodedKeySpec MyPrivKeySpecDSA = new PKCS8EncodedKeySpec(MybinsDSA);
PrivateKey MpkeyDSA = MykeyFactoryDSA.generatePrivate(MyPrivKeySpecDSA);

Signature Mysign = Signature.getInstance("SHA256withDSA");
Mysign.initSign(MpkeyDSA);
byte[] bytes = java.util.Base64.getMimeDecoder().decode(message);
Mysign.update(bytes);
byte[] signature = Mysign.sign();
String messagedSigned =  java.util.Base64.getMimeEncoder().encodeToString(signature);
Log.i( tag: "MessageActivity",  msg: "Message signed is : "+messagedSigned);
String finalMessageToEncrypt = (messagedSigned+"///"+message);
```

Figure 27 Signature du message

```
//INITIAT Ecryption to cipher message ;

byte[] HisbinCpk = java.util.Base64.getMimeDecoder().decode(pubString);
KeyFactory HiskeyFactory = KeyFactory.getInstance("RSA");
EncodedKeySpec HispublicKeySpec = new X509EncodedKeySpec(HisbinCpk);
PublicKey Hispkey = HiskeyFactory.generatePublic(HispublicKeySpec);

//CIPHER THE MESSAGE :

byte[] ThisbyteMessage = finalMessageToEncrypt.getBytes();
Cipher ThiscipherTEXT = Cipher.getInstance("RSA/ECB/PKCS1Padding");
ThiscipherTEXT.init(Cipher.ENCRYPT_MODE, Hispkey);
byte[] thisciphered MSG = ThiscipherTEXT.doFinal(ThisbyteMessage);
String finalMessageToSend = Base64.getMimeEncoder().encodeToString(thisciphered MSG);
```

Figure 28 Code pour Crypter les messages.

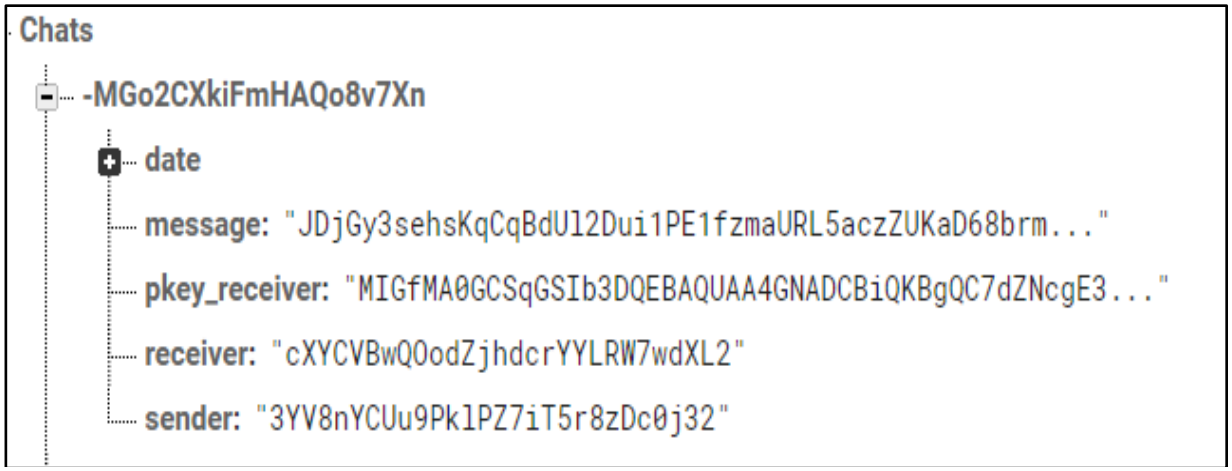

Figure 29 Document Chats pour les messages dans FireBase

La figure ci-dessus représente le document « Chats », où les messages signés et chiffrésseront envoyés et enregistrés.

- **Message** : est le message signé/crypté envoyé par l'émetteur.
- **Sender** : est l'ID de l'émetteur du message chiffré.
- **Receiver** : est l'ID du récepteur.

- Alice reçoit le message chiffré, elle doit le décrypter avec sa clé privée RSA, ensuite elle aura un message qui comporte le message en clair et le message signé, elle décompose le message en utilisant la méthode prédéfinie par java qui est **split.(str)** avec le paramètre '**str**' qui correspond au caractère de décomposition qui est «///» . Maintenaient qu'Alice a pu décomposer le message, elle peut vérifier la signature du message en utilisant la clé publique DSA de BOB. Si la signature est vérifiée alors le message sera affiché sinon le message d'alerte (intrusion dans le message sera affiché)

```
String privateKeystring = SharedPrefManager.getmInstance(context).getPrivateKeyString("");
byte[] binCpk = Base64.getMimeDecoder().decode(privateKeystring);
KeyFactory keyFactory = KeyFactory.getInstance("RSA");
EncodedKeySpec privKeySpec = new PKCS8EncodedKeySpec(binCpk);
PrivateKey pkey = keyFactory.generatePrivate(privKeySpec);

Cipher cipher = null;
try {
    cipher = Cipher.getInstance("RSA/ECB/PKCS1Padding");
    cipher.init(Cipher.DECRYPT_MODE, pkey);

    byte[] decodedText = Base64.getMimeDecoder().decode(chat.getMessage());
    byte[] decipheredText = cipher.doFinal(decodedText);
    String message = new String(decipheredText);
```

Figure 30 Code pour le déchiffrement de message

Figure 31 Fenêtre d'une discussion

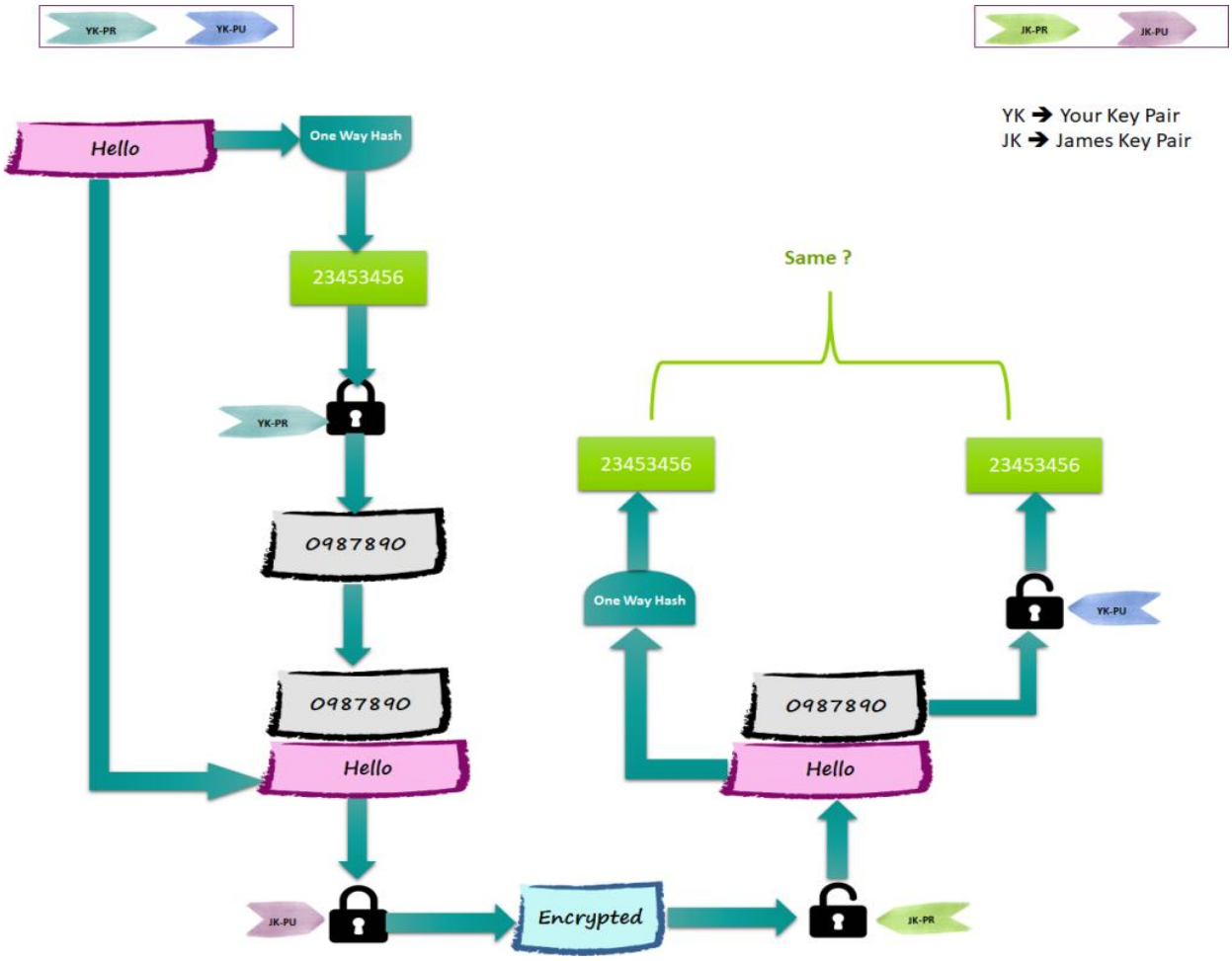

Figure 32 Schéma résument le mécanisme d'envoi du message

IV.4 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 :

En cours de notre mémoire on a réalisé une application Android de chat où les messages sont sécurisés par un crypto système basé sur l'algorithme DSA pour la signature qui vérifie l'intégrité et l'authentification et l'algorithme RSA pour le chiffrementdes messages.

Pour cela, dans un premier temps,nous avons 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 l'algorithme RSA, et dans la dernièrepartie nous avons abordé quelques outils de développement Android et le service Firebase qui est un 'backend as a service', qui nous ont servi à implémenter notre solution.

Notre solution consiste à échanger des messages entre des personnes, en temps réel en utilisant le servie RealTimeDatabase de Firebase. Les messages sont signés grâce à l'algorithme DSA puis crypté grâce à l'algorithme RSA.Les messages sont chiffrés de bout en bout, cela veut dire que les messages restent cryptés dès leurs émissions jusqu'à leur réception.

Enfin, notre travail ainsi présenté reste un prototype sur lequel nous espérons apporter plus de fonctionnalités pour le rendre performant et plus fonctionnel. C'est pourquoi nous envisageons d'améliorer dans cette perspective l'application à même de pouvoir crypter et signer tous types de donnés (images, vidéos…etc).