الجمهورية الجزائرية الديمقراطية الشعبية
REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
وزارة التعليم العالي و البحث العلمي
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
جامعة أبي بكر بلقايد– تلمسان
Université Aboubakr Belkaïd-Tlemcen
كلية التكنلوجيا
Faculté de Technologie

MEMOIRE

Présenté pour l'obtention du **diplôme** de **MASTER**

En : Télécommunications

Spécialité : Réseaux et Télécommunications

Par : MKURUZI Idrisa Hoseni et KUNENGE Abubakar Muhaji

Sujet

Sécurisation des données entre les Smartphones

Soutenu publiquement, le **01 /07/ 2019** , devant le jury composé de :

Mr M.HADJILA	Maître de Conférences	Univ. Tlemcen	Président
Mr R.MERZOUGUI	Maître de Conférences	Univ. Tlemcen	Directeur de mémoire
Mr H.ZERROUKI	Maître de Conférences	Univ. Tlemcen	Examinateur

Année Universitaire 2018-2019

DEDICACES

Mkuruzi Idrisa *:* Ce travail que je dédie à mes parents qui ont toujours été une source constante de motivation et d'encouragement au cours des défis de toute ma vie étudiante. Aussi à mes frères et sœurs que je suis vraiment reconnaissant pour leur soutien et pour les avoir dans ma vie.

Kunenge Aboubakary : Ce travail que je dédie à ma famille et des parents, des amis proches qui ont été favorables à moi dans tous les moyens, spirituellement, financièrement et d'inspiration pour me faire l'homme que je suis aujourd'hui, à ce niveau d'enseignement. Que ALLAH les bénisse tous.

REMERCIEMENTS

Nous tenons à remercier le gouvernement de l'Algérie par le ministère de l'Enseignement supérieur et de la recherche Scientifique, pour cette merveilleuse chance de rester et étudier dans ce pays et de nous donner tous les besoins, y compris la sécurité et la bonne santé.

M. Rachid Merzougui notre professeur unique dans ce projet qui nous a donné le courage et la confiance, l'aide matérielle pour voir ce travail à travers, seul **ALLAH** peut vous rembourser assez.

Nous voudrions également remercier les membres du jury, **Mr M. Hadjila**, Maitre de Conférence à l'université de Tlemcen et **Mr H. Zerrouki**, Maitre de Conférence à l'université de Tlemcen pour accepter d'examiner notre memoire.

Être un étranger n'est pas facile dans un pays, mais grâce à de bonnes personnes de l'Algérie en particulier nos camarades de classe depuis le début de la première année 2014/2015 à 2018/2019, nos merveilleux professeurs qui nous ont accueillis et traités comme un. On vous remercie !

Table de Matière

LISTE DES FIGURES

CHAPITRE 1

CHAPITRE 2

CHAPITRE 3

CHAPITRE 4

LISTE DES ABREVIATIONS

2G	Second Generation Network	Réseau de deuxième génération
3G	Third Generation Network	Réseau de troisième génération
AES	Advanced Encryption Standard	Standard d'encryptage avancé
ADT	Android Development Tool	Outil de développement Android
API	Application Programming Interface	Interface de programmation d'applications
ARPA	Advanced Research Project Agency	Advanced Research Project Agency
CRC	Cyclic Redundancy Check	Contrôle de redondance cyclique
DES	Data Encryption Standard	Norme de cryptage des données
DH	Diffie Hellman	Diffie Hellman
DMG	Disk iMaGe	Disque iMaGe
E2EE	End To(2) End Encryption	chiffrement de bout en bout
GCHQ	Government Communications Headquarters	Quartier général des communications gouvernementales
GnuPG	GNU Privacy Guard	

GSM	Global System Mobile	Système global mobile
IDE	Integrated development environment	Environnement de développement intégré
IETF	Internet Engineering Task Force.	Groupe de travail sur l'ingénierie Internet.
IMSI	International Mobile Subscriber Identity	Identité internationale d'abonné mobile
JSON	JavaScript Object Notation	Notation d'objet JavaScript
MMS	Multimedia Message Service	Service de messagerie multimédia
NSA	National Security Agency	Agence de Sécurité Nationale
OTR	**Off-the-Record** Messaging	
PIN	Personal Identity Number	Numéro d'identité personnelle
RAM	Random Access Memory	Memoire vive
RAND	Random	Aléatoire
RSA	Rivest-Shamir-Adleman	Rivest-Shamir-Adleman
SIM	Subscriber Identity Mobile	Identité d'abonné mobile
SMS	Short Message Service	Service de messages courts
SRES	Signed RESponse	réponse signée

TCP/IP	Transmission Control Protocol/Internet Protocol	Protocole de contrôle de transmission / protocole Internet
TETRA	Terrestrial Trunked Radio	Sécurité de la couche de transport
TLS	Transport Layer Security	Sécurité de la couche de transport
TMSI	Temporary Mobile Subscriber Identity	Identité d'abonné mobile temporaire
VoIP	Voice Over Internet Protocol	Protocole de voix sur Internet
VoLTE	Voice over LTE	Voix sur LTE

RÉSUMÉ

La sécurité des données entre smartphones concerne la sécurité des données d'un expéditeur au destinataire. Les données sont envoyées ou partagées selon différentes méthodes, comme Internet, et il y a toujours une vulnérabilité de piratage de ces données si elles ne sont pas sécurisées. Dans ce memoire, nous avons montré comment les données sont sécurisées en utilisant l'algorithme RSA.

Mots clés : chiffrement, déchiffrement, algorithme, RSA, authentification, Intégrité, clé publique, clé privée

ABSTRACT

The security of data between smartphones concerns the security of the data of a sender to the recipient. Data is sent or shared using different methods, such as the Internet, and there is always a vulnerability to hacking this data if it is not secure. In this Research, we have shown how data is secured by using RSA algorithm.

Key words: Encryption, Decryption, Algorithm, RSA, Authentication, integrity, public key, private key,

Introduction Générale

En 3 Avril 1973 ingénieur américain Martin Cooper a inventé le premier téléphone portable, qui a été utilisé seulement dans les affaires et le gouvernement. En 1992, le premier Smartphone a été inventé par IBM connu sous le nom Smartphone Simon. Ce fut l'une des plus grandes percées dans le domaine de la technologie qui a permis à des millions de personnes de communiquer en petite période de temps en faisant des appels, envoyer des messages courts et même des e-mails.

Un Smartphone peut être défini comme un ordinateur personnel (PC) en raison de ses capacités et fonctionnalités avancées. Un dispositif sans fil qui utilise radio bidirectionnelle, constitué d'un émetteur radio et un récepteur radio. Au fil des ans, un grand nombre de smartphones ont sorti avec différents systèmes d'exploitation, tous dans le but de partager ou de transférer des données. Des exemples de téléphones intelligents couramment utilisés ces jours-ci sont Apple, Samsung, Oppo, Nokia et ainsi de suite, chacun vient avec son propre système d'exploitation (par exemple Apple iOS, Java Android, Symbian, Windows) et des moyens de sécurité.

Aujourd'hui, les smartphones ont été la plupart du temps dominé par les applications médias sociaux comme Facebook, WhatsApp, Viber et ainsi de suite où grand nombre de données sont partagées. Sans oublier marketing et d'affaires sont largement liés à l'utilisation des smartphones.

Les données qui circulent et partagées par les téléphones intelligents devient énorme chaque jour et qui le rend vulnérable à être attaqué (volé) par des pirates et utiliser leurs avantages, d'où la sécurité de ces données est nécessaire.

Les attaques de données partagées par les smartphones dépendent du type de communication utilisé par les dispositifs, parmi les menaces à la sécurité des smartphones comprennent :

I. Des fuites de données (Data Leakage) - cela est causé par les applications gratuites installées par l'utilisateur (s) qui contient publicités ou programmes malveillants qui recueille des données à l'insu de l'utilisateur (s).

II. Wi-Fi non sécurisé - Wi-Fi gratuite sont normalement non sécurisé (non chiffré), d'où il devient facile de capturer des informations qui circule dans un réseau particulier.

III. Réseau Spoofing - dans ce cas, les pirates utilisent un point d'accès faux pour attirer les utilisateurs par exemple « Airport connexion Wi-Fi » ou « Coffee house » qui, lorsque l'utilisateur se connecte au réseau sera nécessaire pour créer un compte pour profiter du service gratuit, la plupart des utilisateurs préfèrent utiliser les mêmes informations pour les différents services et c'est lorsque le pirate profitera à l'information.

IV. Phishing Attack - communément appelé « spam », les pirates envoient des courriels non vérifiés à différents utilisateurs avec des liens inconnus qui lors de l'ouverture constituera une grande menace pour les utilisateurs.

V. Spyware - un programme installé en arrière-plan qui est utilisé pour collecter et surveiller les activités quotidiennes d'un utilisateur. En l'absence d'antivirus, le programme se déroulera non détecté.

VI. Cryptographie Brisé - se produit lorsqu'un développeur utilise un des algorithmes de chiffrement faible ou algorithme de chiffrement fort avec la mauvaise application. Alors que dans le premier cas, un pirate peut se fissurer facilement le mot de passe et l'accès gagner. Le second cas, le développeur utilise un algorithme puissant, mais laisse de porte dérobée qui laissent les pirates libres de modifier certaines fonctions et l'accès gagner.

VII. Une mauvaise manipulation de session - c'est quand jeton d'authentification ou des informations d'une utilisation ou plus sont fournis à un autre utilisateur (s) ou accidentellement divulgué à un autre utilisateur (s) qui leur permettra d'usurper l'identité des utilisateurs légitimes.

Dans ce projet, nous verrons ;

- Méthodes de communications par téléphones intelligents, leur vulnérabilité et de la sécurité (« Chapitre 1 »).
- Méthodes de sécurité des données dans les téléphones intelligents, comment les données sont sécurisées sous forme étant une fuite et comment ils sont chiffrés et déchiffrés (« Chapitre 2 »).
- Chiffrement de bout en bout (« Chapitre 3 »).
- Simulation (« Chapitre 4 »).

Puis on va terminer avec la conclusion générale de ce memoire.

CHAPITRE 1

MÉTHODES DE COMMUNICATIONS ENTRE LES SMARTPHONES

1,1 INTORDUCTION

En raison de leur capacité et des fonctionnalités avancées, Smartphones peut communiquer ou partager des données (information) par le biais de diverses façons en utilisant différents types de réseaux auxquels ils ont accès. Dans chaque type de réseau utilisé, il y a des vulnérabilités et des moyens de sécurité utilisés pour assurer la sécurité des données. Les points suivants sont des moyens dont les téléphones intelligents peuvent communiquer ;

1.1 Appels et SMS via GSM / 3G / VoLTE

L'évolution du réseau en télécommunications de GSM (2G) 4G ont permis le transfert d'appels vocaux (données vocales) et SMS d'un appareil à l'autre de manière efficace. Smartphones sont livré avec des qualités différentes, les prix diffèrent donc des types de réseau, ils peuvent prendre en charge.

GSM et 3G sont des réseaux à commutation de circuits qui crée dédier point à point de connexion pour établir un appel vocal alors que le réseau 4G est un paquet réseau commuté qui transfèrent des données en petits blocs et réassemblés à destination.

Un smartphone 4G de soutien devra soit passer à abaisser les types de réseau (2G / 3G) à affermira un appel vocal ou commutateur VoLTE qui est une norme spéciale pour établir les appels vocaux via LTE (4G). Tous ces réseaux partagent presque la même infrastructure pour la première (2G) Génération Network quand il vient d'établir à l'établissement d'un appel téléphonique, comme illustré dans la figure ci-dessous ;

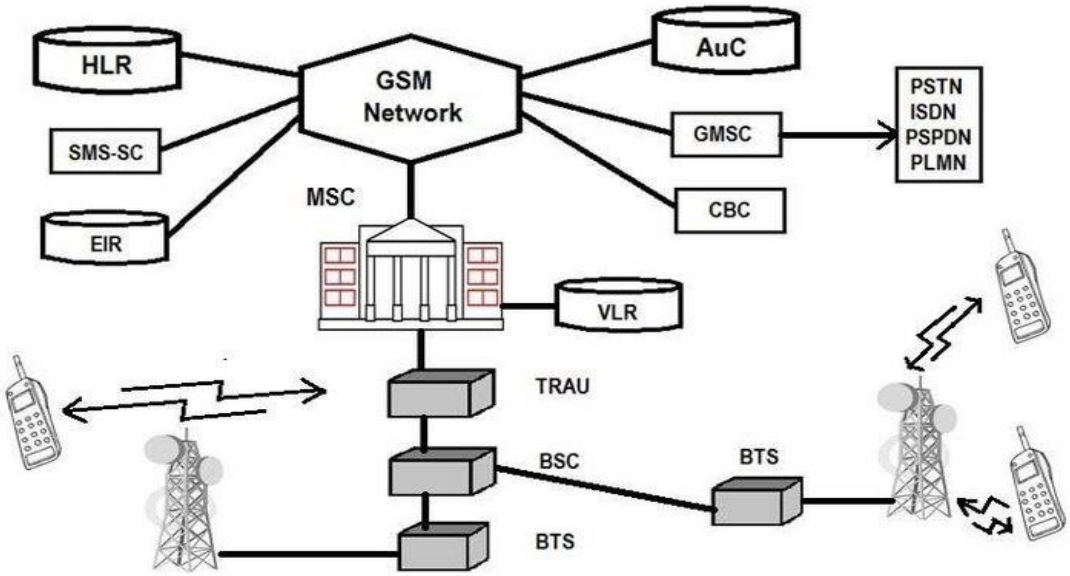


Figure 1 L'architecture du réseau GSM

1.1.1 Définition des termes

I. BTS (Base Transmission Station) - qui sont des émetteurs-récepteurs ayant un minimum d'intelligence,
II. BSC (Base Switching Control) - qui contrôle un ensemble de BTS et permet une première concentration des circuits.
III. VLR (Visitor Location Register) – est une base de donnes qui enregistre les numéros temporaires des abonnées par Une Présente zone géographique.
IV. MSC (Mobile Switching Center) - sont des commutateurs mobiles associés en général aux bases de données VLR *(Visitor Location Register),*
V. HLR (Home Location Register) - est une base de données de localisation et de caractérisation des abonnés.
VI. EIR (Equipment Identity Register) – est une base de données qui enregistre les identités des abonnes (IMEI) [1]

1.1.2 Gestion de la sécurité

Le réseau GSM utilise un canal radio pour gérer les communications entre les utilisateurs. Mais l'utilisation d'un canal radio rend les communications vulnérables aux écoutes d'où des problèmes de confidentialité, et aux utilisations frauduleuses d'où des problèmes de sécurité Par conséquent les systèmes de réseau (GSM) ont créé les mesures de sécurité à suivre avant que la communication entre les appareils est établie comme suit ;

- Confidentialité de l'IMSI,
- Authentification d'un abonné pour protéger l'accès aux services,
- Confidentialité des Données usager,
- Confidentialité des informations de signalisation.

1.1.2.1 Confidentialité de l'identité de l'utilisateur

Pour identifier les utilisateurs, le réseau utilise le numéro temporaire appelé TMSI (identité temporaire d'abonné mobile) qui est stocké au niveau de la carte SIM et VLR d'un utilisateur. Ce numéro est toujours crypté et il est couramment utilisé car il est facile pour l'utilisateur de cloner le numéro IMSI. [2]

L'utilisation de TMSI dépend d'un emplacement d'un utilisateur, si l'utilisateur change de lieu alors l'ancien TMSI doit être utilisé pour acquérir la nouvelle, et voilà comment s'identifie les utilisateurs par leurs réseaux. La figure illustre ci-dessous;

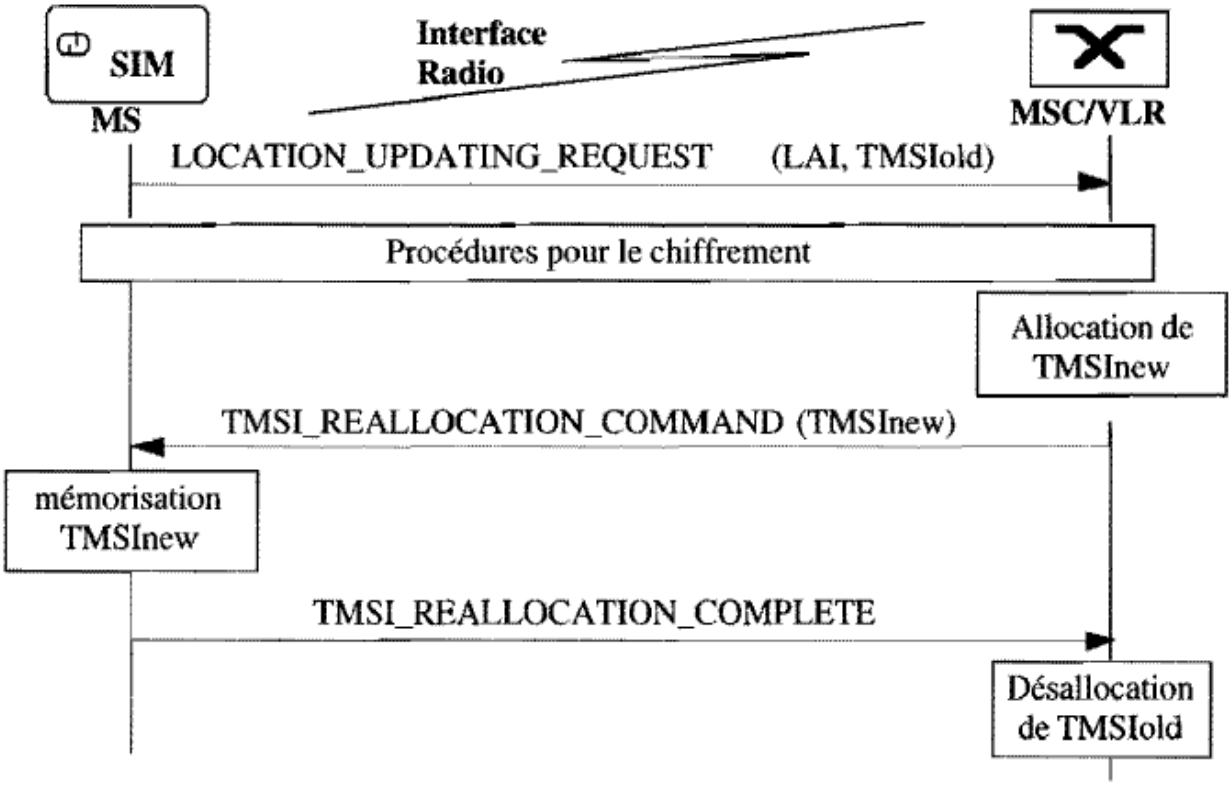


Figure 2 Exemple de l'Intervenir procédure allocation Faisant du TMSI

1.1.2.2 Principe général d'authentification et de chiffrement

GSM utilise les méthodes suivantes pour le chiffrement ;

- Des nombres aléatoires RAND,
- Une clé Ki pour l'authentification et la détermination de la clé de chiffrement Kc,
- Un algorithme A3 fournissant un nombre SRES à partir des arguments d'entrée RAND et la clé Ki pour l'authentification,
- Un algorithme A8 pour la détermination de la clé Kc à partir des arguments d'entrée RAND et Ki,
- Un algorithme A5 pour le chiffrement/déchiffrement des données à partir de la clé Kc.

Chaque utilisateur est attribué à la clé Ki, algorithmes A3, A5, A8 sont les mêmes pour tous les utilisateurs du même réseau. A3 et A8 sont parfois regroupés et ont appelé A38. RAND, SRES, Kc aussi appelé les triplés sont utilisés pour authentifier et crypter les communications. [2]

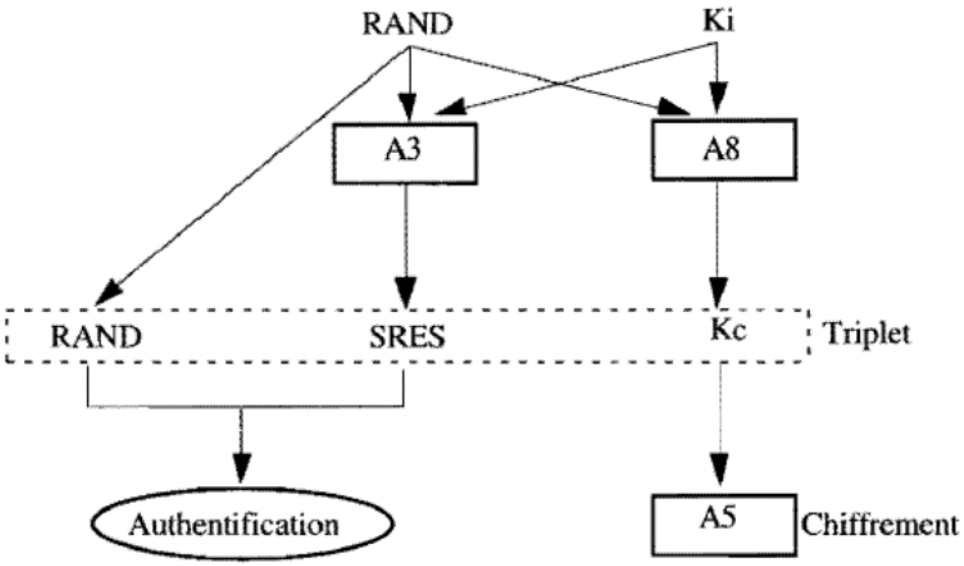


Figure 3 Utilisation des différents éléments de sécurité dans GSM

Lorsque l'échange du numéro TMSI a réussi le processus d'authentification commence, le réseau envoie nombre aléatoire RAND aux utilisateurs des cartes SIM et la génération du nombre SRES commence à utiliser A3 algorithme et clé Ki, si le nombre SRES généré par les deux cartes SIM et le réseau est le même, l'utilisateur est authentifié. [2]

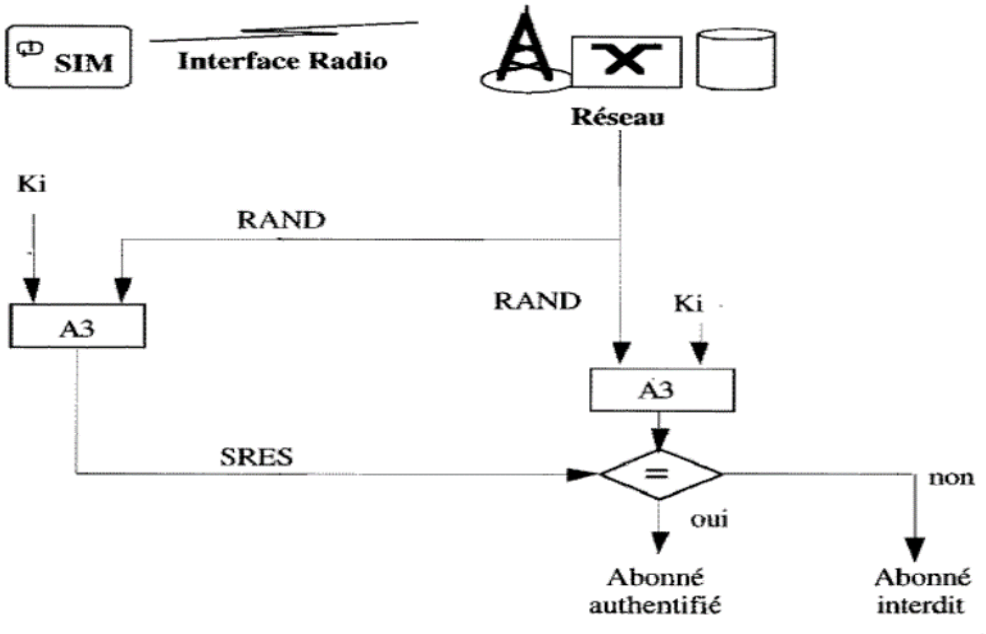


Figure 4 Utilisation des différents éléments de sécurité dans GSM

La même chose vaut pour le chiffrement de la communication, une fois que l'utilisateur (s) est authentifié pour acquérir le canal radio pour la communication, le réseau envoie nombre aléatoire RAND pour générer la clé de chiffrement Kc en utilisant A8 algorithme, si le même est généré à la fois par carte SIM et le réseau puis l'algorithme A5 est utilisé pour chiffrer les communications d'un utilisateur. [2]

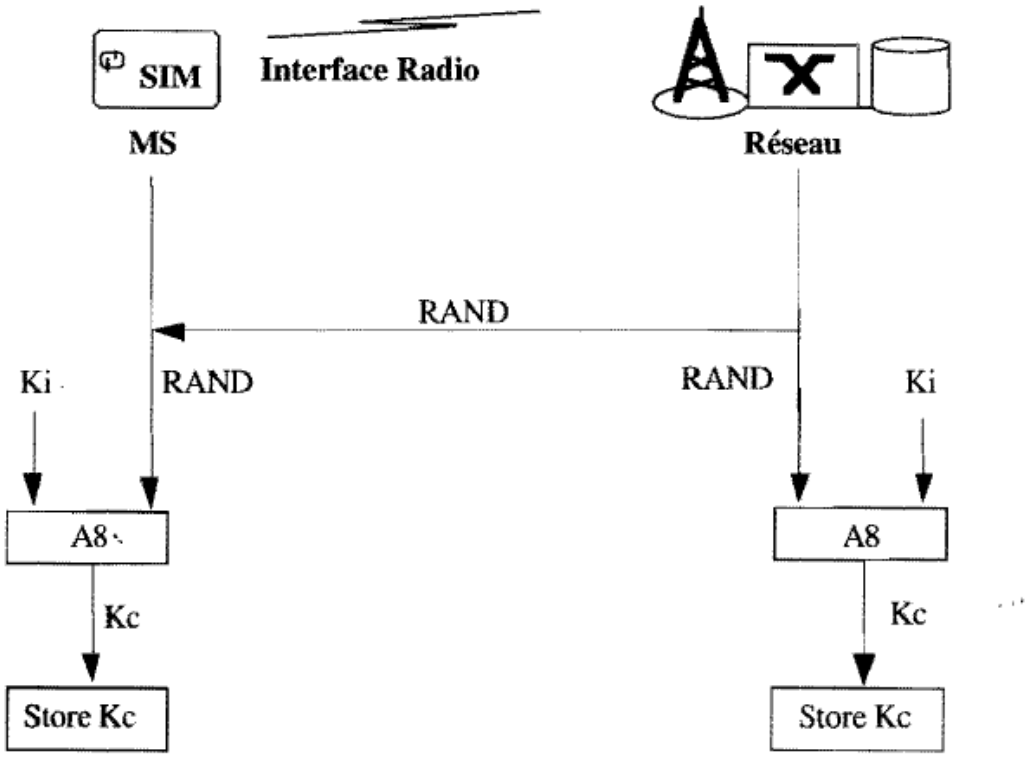


Figure 5 Déroulement global de la procédure d'authentification

Et voilà comment les appels (appels de voix/données) est sécurisé au réseau GSM pour éviter les écoutes par des pirates, comme pour SMS sa sécurité dépend de l'application installée dans les téléphones mobiles des utilisateurs. Ces applications sont mises en œuvre avec de nombreuses normes de cryptage, y compris DES, TRIPPLE DES, AES et RSA qui sont utilisés pour crypter des messages texte ou le contenu graphique envoyé par SMS (MMS).

Le chapitre 2 a expliqué en détail comment les normes de cryptage DES, AES, RSA fonctionne.

1.2 Connexion Bluetooth

Bluetooth est une norme ouverte / technologie de communication de fréquence radio à courte portée (RF). La technologie Bluetooth est utilisée pour établir des réseaux sans fil personnel (WPAN), tels que

Les réseaux ad hoc (réseau de topologie moins) ou Peer-to-Peer (P2P).

En raison de son faible coût, de faible puissance, il fournit un mécanisme pour la création de petits réseaux sans fil sur une base ad hoc, connu sous le nom piconets.

La technologie Bluetooth est livrée avec différentes versions, la plus récente est la version 4 qui se trouve dans la plupart des appareils Smartphones car il peut se connecter à la plupart des autres appareils avec la version inférieure de Bluetooth.

1.2.1 Spécifications de Bluetooth

Le Bluetooth couramment installé dans les smartphones est la version 4.0 de Bluetooth, qui est la dernière version de Bluetooth. Bluetooth est également défini dans différentes classes en raison de la gestion de l'alimentation ou de la consommation ;

I. Classe 1 : niveau de puissance de 100 mW (20 dBm), actif jusqu'à 100 mètres, exemple de point d'accès Bluetooth.
II. Classe 2 : puissance moyenne de 25 mW (4dBm), active jusqu'à 10 mètres, elle est couramment utilisée dans les smartphones.
III. Classe 3 : puissance inférieure à 1mW (0dBm), active jusqu'à 1 mètre (adaptateurs Bluetooth).

L'architecture principale Bluetooth est constituée de la radio, de la fréquence de base et du gestionnaire de liens. Utilise une fréquence de 2,5 GHz avec une bande passante de 1 Mo / s qui est ralentie par la FEC (Forward Error Correction). Le type de modulation utilisé est également GFSK (Gaussien Frequency Shift Key).

Le gestionnaire de liens (Link manager) est la partie essentielle de la technologie Bluetooth dans les smartphones, car il configure, authentifie et gère la connexion des périphériques Bluetooth afin de gérer la consommation d'énergie.

1.2.2 Sécurité de Bluetooth (sécurité des données Bluetooth)

En raison de la simplicité de leur disponibilité, les utilisateurs Bluetooth de nombreux appareils ne comprennent pas à quel point Bluetooth peut être vulnérable aux attaques de pirates informatiques s'il n'est pas géré / utilisé correctement. La technologie Bluetooth installée sur plusieurs appareils intelligents, tels que Bluejacking et l'écoutes (eavesdropping), fait l'objet de nombreuses menaces, dont nous parlerons plus loin dans ce chapitre.

Pour assurer la sécurité des données circulant dans les périphériques Bluetooth, la technologie Bluetooth permet aux entités de maintenir la sécurité, à savoir :

I. Bluetooth Device Address(BD_ADD)- Il s'agit d'une adresse 48 bits unique pour chaque périphérique Bluetooth et définie par l'Institut des ingénieurs électriciens et électroniciens (IEEE).
II. Private Authentication- un nombre aléatoire de 128 bits utilisé à des fins d'authentification.
III. Private encryption key- 8-128 bits de longueur utilisés pour le cryptage.
IV. Random number RAND- un nombre aléatoire ou pseudo-aléatoire de 128 bits qui change fréquemment et qui est créé par le périphérique Bluetooth lui-même.

De plus, la sécurité Bluetooth est divisée en 3 modes :

I. Security Mode 1- Mode non sécurisé
II. Security Mode 2- un gestionnaire de sécurité (spécifié dans l'architecture Bluetooth) contrôle l'accès à des services et périphériques spécifiques.
III. Security Mode 3- un périphérique Bluetooth lance des procédures de sécurité avant que la liaison physique ne soit complètement établie. Les périphériques Bluetooth fonctionnant en mode de sécurité 3 requièrent une authentification et un cryptage pour toutes les connexions avec le périphérique. Ce mode prend en charge l'authentification (unidirectionnelle ou mutuelle) et le cryptage. [3]

Avant qu'une connexion soit établie et prête à partager des données entre appareils, trois procédures doivent être suivies pour assurer la sécurité des informations partagées entre les appareils concernés :

I. Gestion des clés (Key Management)
II. Authentification
III. Encryptions

1.2.3 Gestion des clés

Dans toutes les transactions de sécurité entre deux périphériques ou plus, la clé de liaison est utilisée lors du processus d'authentification et de la détermination de la clé de cryptage. Cette clé de liaison a une durée de vie. Il peut s'agir d'une clé semi-permanente utilisée après la fin de la session en cours pour authentifier les unités Bluetooth qui la partagent, ou d'une clé temporaire qui dure jusqu'à la fin de la session en cours et qui ne peut pas être réutilisée. Les clés temporaires sont couramment utilisées dans les connexions point à multipoint, où les mêmes informations sont transmises à plusieurs destinataires.

Dans la technologie Bluetooth, il existe différents types de clé de liaison ;

I. The unity key (La clé de l'unité) - généré en un seul appareil lorsqu'il est installé.
II. Combination Key - tirées des informations de deux appareils et il est généré pour chaque nouvelle paire de périphériques Bluetooth.
III. Master key and initialization key - La clé principale est une clé temporaire, qui remplace la clé de la liaison actuelle. Il peut être utilisé lorsque l'unité maître veut transmettre

l'information à plusieurs destinataires. La clé d'initialisation est utilisée comme clé de liaison au cours du processus d'initialisation lorsqu'il n'y a pas encore les clés de toute unité ou combinaison. Il est utilisé uniquement lors de installation.

IV. Encryption key - générée à partir de la clé de liaison actuelle, un 96 bits chiffrement Offset nombre (COF) et un nombre aléatoire de 128 bits. Le COF est basé sur l'authentifié chiffrement Offset (ACO), qui est générée pendant le processus d'authentification. Lorsque le lien Manager (LM) active le chiffrement, la clé de chiffrement est générée. Il est automatiquement changé chaque fois que le périphérique Bluetooth est en mode Chiffrement.

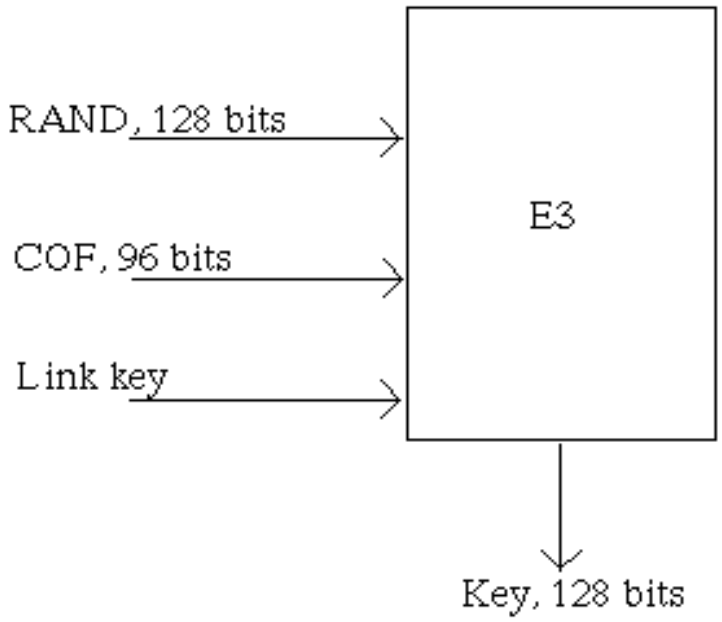


Figure 6 algorithme de génération de clé E3 pour la clé de chiffrement

1.2.4 Authentification

Avant de partager des données, dispositifs passe des processus d'authentification pour vérifier les utilisateurs. L'authentification mis en place en Bluetooth est stratégie de stimulation / réponse où un protocole 2-move est utilisé pour vérifier si l'autre partie sais la clé secrète. Si l'authentification réussite, authentifié chiffrement Offset (Authenticated Ciphering Offset ,ACO) est calculée et stockés dans les deux dispositifs de génération de clé de chiffrement plus tard.[3][4]

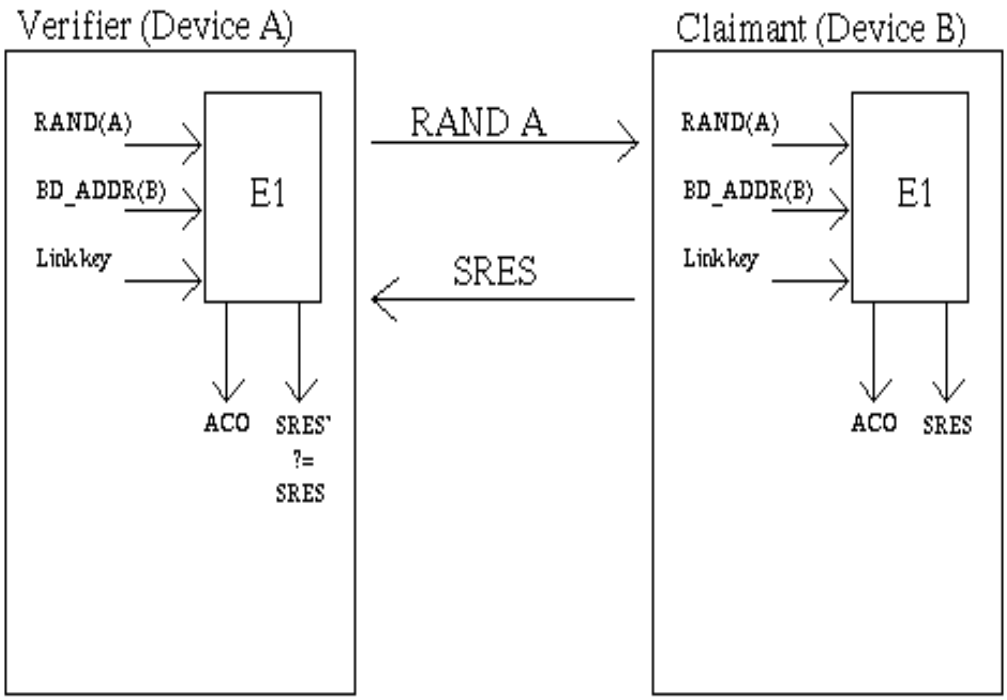


Figure 7 Description du processus d'authentification

Tout d'abord, le vérificateur envoie au requérant un nombre aléatoire d'être authentifié. Ensuite, les deux participants utilisent la fonction de l'authentification E1 avec le nombre aléatoire, les requérants adressent de périphérique Bluetooth et la clé de liaison actuel pour obtenir une réponse. Le prestataire envoie la réponse au vérificateur, qui ensuite s'assure que les réponses correspondent. Si l'authentification échoue, il y a une période de temps qui doit s'écouler jusqu'à ce qu'une nouvelle tentative d'authentification peut être faite. La période de temps double pour chaque échec de tentative ultérieur de la même adresse, jusqu'à ce que le temps d'attente maximal est atteint. Le temps d'attente diminue exponentiellement au minimum lorsque aucun échec de l'authentification des tentatives au cours d'une période de temps.

1.2.6 Encryptions

Le système de cryptage Bluetooth crypte les charges utiles des paquets. Cela se fait avec un chiffrement de flux E0, qui est resynchronisée pour chaque charge utile. Le chiffrement de flux E0 comprend le générateur de clé de charge utile, le générateur de flux de clés et la partie cryptage/décryptage. [4]

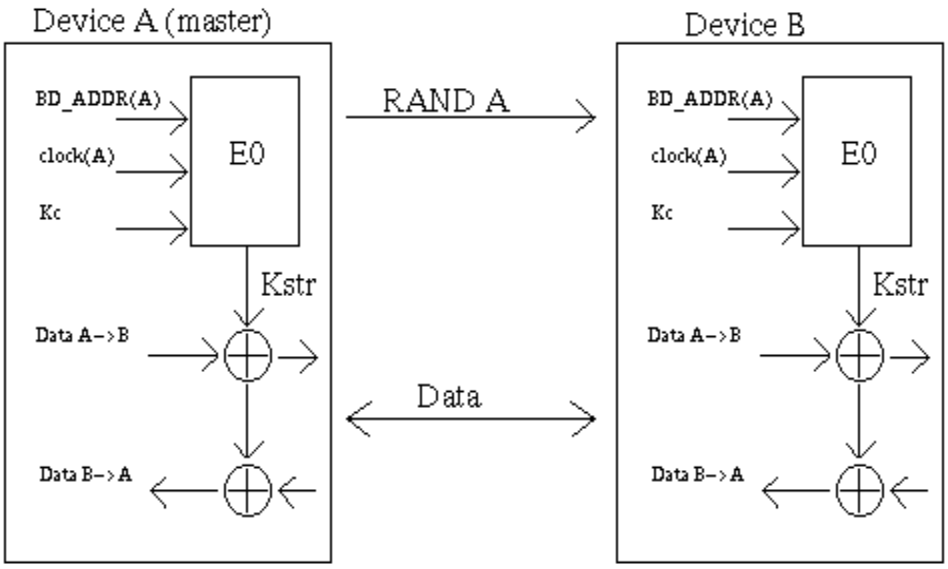


Figure 8 Description du processus de chiffrement

Le générateur de clé de charge utile combine les bits d'entrée dans un ordre approprié et les déplacements pour les quatre Linear Feedback Shift Registers (LSFR) du générateur de flux de données clés. Comme la taille de clé de chiffrement varie de 8 bits à 128 bits, la taille de la clé de chiffrement utilisée entre deux périphériques doit être négociée. Le même cryptage utilisé dans le dispositif A pour crypter les données est utilisé dans l'appareil B pour déchiffrer les données.

Selon que dispositifs utilise clé de liaison semi-permanente ou passe-partout, cryptage en Bluetooth varie dans 3 modes principaux ;

I. Le Mode de chiffrement 1 — aucun cryptage n'est effectuée sur tout le trafic.
II. Mode de cryptage 2 — adressé individuellement le trafic est crypté à l'aide de clés de chiffrement basés sur des clés de liaison individuels ; trafic de diffusion n'est pas chiffré.
III. Le Mode de chiffrement 3 — tout le trafic est crypté à l'aide d'une clé de chiffrement basée sur la clé de liaison maître. Modes de chiffrement 2et 3 utilisent le même mécanisme de chiffrement. [5]

1.2.6 Menaces Bluetooth (bleu » Menaces)

A. Bluesnarfing - Force une connexion à un dispositif Bluetooth, ce qui permet l'accès aux données stockées sur l'appareil et même l'identité internationale d'équipement mobile de l'appareil (IMEI)
B. Bluejacking - Initié par un attaquant d'envoyer des messages non sollicités à un utilisateur d'un appareil compatible Bluetooth pour inciter l'utilisateur à répondre.

Ressemble attaques de spam et le phishing menées contre les utilisateurs de messagerie.

C. Bluebugging - défaut de sécurité dans le firmware permet attaquant d'utiliser les commandes du dispositif sans en informer l'utilisateur. [5]

1.2.8 Contremesures

A. Codes PIN complexes
B. Introuvables par défaut
C. Chiffrement de taille maximale de clé
D. Authentiquassions mutuelle
E. Mode de sécurité de niveau de service 3 (le plus sûr)
F. Installer des correctifs logiciels et mises à niveau

1.3 Communication Internet

Le monde change à un rythme très rapide, ce qui était de prendre les quelques mois pour envoyer, prend maintenant secondes. Aujourd'hui, plus que jamais, la communication sur Internet a fait en sorte de relier les gens de deux côté opposé du monde sans problème. Communication sur Internet est le partage des informations, des idées ou des mots sur Internet. Contrairement à avant, les gens peuvent rester à la maison et être connecté à sa famille, ses amis et même des collègues de partout dans le monde.

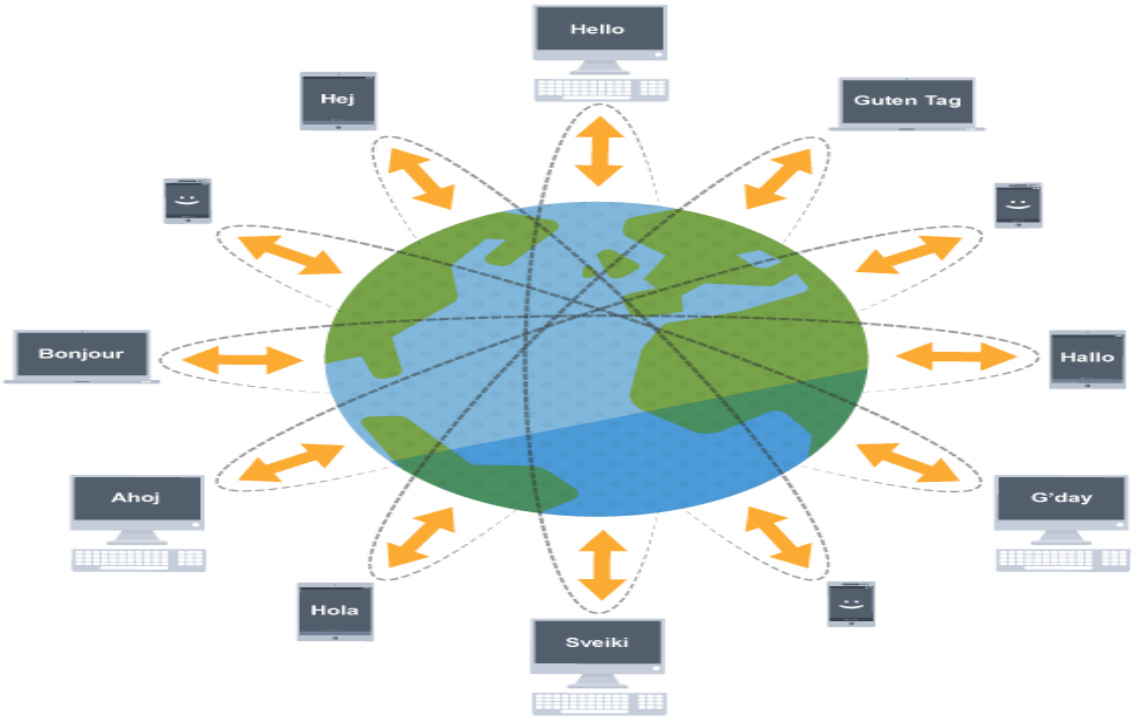


Figure 9 Internet en tant que moyen d'échange dans le monde entier

1.3.1 Origine de l'Internet

Dans les années 1950, le Département de la Défense des États-Unis formé plusieurs organismes, comme l'Advanced Research Projects Agency (ARPA, maintenant connu sous le nom DARPA) dans le but de développer la technologie. Toutefois, étant donné qu'ils étaient basés dans les universités du pays, les scientifiques de ARPA ne pouvaient pas facilement communiquer ou partager des informations. Pour résoudre ce problème, ARPA a créé un réseau d'ordinateurs, qu'ils ont appelé ARPANET. Se rendant compte comment était ARPANET utile, d'autres organisations ont construit leurs propres réseaux. Cependant, ces réseaux individuels ne pouvaient pas communiquer facilement entre eux.

Dans les années 1970, un protocole a été développé appelé TCP / IP, ce protocole a permis aux réseaux séparés pour communiquer entre eux. La jonction de ces réseaux individuels a créé un vaste réseau étendu (Wide Area Network, WAN) qui est venu à être connu comme Internet.

Depuis lors, l'utilisation de l'Internet par les organisations et les particuliers a augmenté année après année. Au début, ARPANET ne comptait que quatre ordinateurs. Maintenant, des milliards d'ordinateurs et de périphériques intelligents tels que les smartphones sont connectés à Internet.

1.3.2 Types de communication internet

A. Messagerie instantanée :

Messagerie instantanée, envoie un message en temps réel d'un utilisateur à l'autre. Cela se fait normalement sur les applications de chat avec l'utilisation de l'Internet. Jusqu'à présent, il y a beaucoup d'applications de chat développées qui peuvent être utilisés pour envoyer des messages instantanés certaines de ces applications de chat sont : Facebook Messenger, WhatsApp, Telegram et plusieurs autres.

B. Des sites sociaux:

Les sites de réseaux sociaux tels que Instagram, Twitter, Snapchat, Facebook et Google Plus permet aux utilisateurs de créer des identités d'eux-mêmes en ligne et communiquer avec des personnes qu'ils connaissent. L'utilisation de ces sites comme un moyen et Internet en tant que canal de communication, les utilisateurs peuvent faire plusieurs choses. Ils peuvent partager des nouvelles et des mises à jour, ils deviennent bien informés de ce que tout le monde est à tout moment à temps, ils peuvent partager des vidéos, des photos et des idées les unes aux autres.

L'utilisation des sites de réseaux sociaux pour rester en contact avec les personnes que vous connaissez et de faire de nouveaux contacts, est devenu l'un des plus grandes applications de communication Internet.

C. Courrier électronique:

Les sites de messagerie tels que Gmail et Yahoo Mail permettent aux gens d'envoyer du courrier électronique sur Internet en quelques secondes. Par rapport aux anciens systèmes postaux où les lettres ont été envoyées par la poste aux autres et a pris plusieurs jours pour atteindre le destinataire. En raison du développement du courrier électronique, les systèmes postaux dans le monde entier sont devenus obsolètes.

D. Forums en ligne:

Forums permettent aux personnes ayant des intérêts similaires de se réunir et de communiquer entre eux sur les sujets qu'ils aiment. Il y a littéralement des milliers de sites du forum qui se consacrent à des thèmes spécifiques tels que les voyages, les voitures, les sports et autres.

E. Conférence Internet:

Conférence sur Internet peut être divisé en deux parties - audio et vidéo. En conférence audio, les gens échangent des dialogues via des microphones et haut-parleurs. Ceci est l'équivalent d'un appel téléphonique, mais est plus flexible tenant compte des appels téléphoniques charge plus élevés quand il est au-delà des applications locales, alors que la conférence audio tels que Skype, peuvent le faire sans frais.

Les autres moyens de faire des conférences sur Internet est la vidéo conférence. Il est pratiquement le même que les conférences audio, mais avec les deux parties se voir à travers une caméra web. Des applications comme Skype, et bien d'autres fournir à la fois des conférences audio et vidéo dans leurs services, qui sont bons pour les pourparlers de famille à longue distance et entreprises.

1.3.3 Avantage de la communication internet

A. Très vite :

Dans une communication Internet, diverses informations envoyées à un destinataire sont reçus en quelques secondes, par exemple, les messages sont reçus en un instant, des e-mails ont frappé la boîte de réception d'un récepteur en l'espace de quelques secondes, les dernières nouvelles atteignent le public visé presque le temps de leur publication.

B. Moins cher:

Le coût de la communication Internet est très faible par rapport à d'autres moyens de communication comme le visage aux réunions du visage et la livraison du courrier. La technologie vous connecte à vos partenaires, collègues, clients et fournisseurs de n'importe quel emplacement pour une fraction du coût nécessaire pour accueillir un-à-tête. Et que la technologie continue de devenir plus efficace, le coût de la communication en ligne continue de baisser de manière significative. Avec le visage traditionnel pour faire face à la réunion, vous avez besoin de gagner du temps, d'argent pour voyager et ainsi de suite. Communication Internet permet à vous et votre équipe de vous connecter sans avoir à quitter vos bureaux.

C. Permet le transfert de toutes types des données :

Communication sur Internet est mieux que d'autres méthodes traditionnelles car elle permet une plus large gamme de données-types tels que des images, des vidéos et des textes. Comparer avec des journaux et des magazines hors ligne qui ne peut pas permettre à des vidéos ou des programmes de télévision qui ne peuvent pas se concentrer sur le texte.

D. Améliorer la collaboration:

Communication Internet apporte des équipes ensemble à travers le monde. Le personnel peut collaborer facilement sans limites et prendre des décisions plus éclairées instantanément. Cela conduit à une réduction des délais du projet, réduisant le temps nécessaire pour lancer un nouveau produit / service. Ce morceau de technologie est également utile dans l'éducation. Non seulement les étudiants de collaborer entre eux et avec les étudiants étrangers, ils peuvent échanger des idées et en apprendre davantage sur les diverses cultures là-bas. Les parents peuvent aussi participer activement à l'éducation de leurs enfants en liant l'école de leurs enfants avec les bibliothèques, les maisons, et plus encore. Des millions d'écoles à travers le monde utilisent déjà cette technologie pour améliorer l'apprentissage.

E. Couvre Une Grande surface:

Avec une communication internet un billet de blog peut obtenir des milliers de lectures. Vous pouvez communiquer et partager des idées avec des gens à travers les continents. Les marqueteurs peuvent atteindre un public mondial.

1.3.4 Inconvénients (défis) de la communication Internet

- **Problème de confidentialité**

La plupart des données utilisateur partagées sur Internet sont stockés quelque part dans une base de données. Les informations que vous partagez aujourd'hui restera pendant une longue période de temps dans ces bases de données quelque part hors de votre portée. De temps en temps, nous avons vu les pirates qui tentent de pénétrer dans les bases de données pour accéder aux informations de l'utilisateur, cela a été un problème sans fin. Si quelqu'un avoir accès à la base de données, il peut lire et examiner toutes les données utilisateur partagées dans le passé proche et lointain et qui pose un problème de confidentialité.

- **Questions de sécurité**

Il ne manque pas de spammeurs et les fraudeurs sur Internet. Vous devez vous assurer que vous ne tombez pas en proie à leurs régimes.

Vous devez être conscient des techniques couramment utilisées telles que le internet phishing, Malwares, Spywares, et beaucoup d'autres logiciels espions comme décrit ci-dessus.

Alors que certains sites peuvent être carrément dangereux, d'autres tentent de connaître simplement votre adresse e-mail afin de pouvoir faire une grande liste d'adresses e-mail et le vendre à d'autres personnes, qui peuvent ensuite utiliser pour envoyer des messages de spam pour vous afin qu'ils puissent accéder à des informations personnelles importantes de votre appareil.

1.3.5 Les solutions aux problèmes de confidentialité et de sécurité

Une solution concrète à appliquer un problème de la vie privée et la sécurité des communications Internet, le besoin de sécuriser les données de l'utilisateur quand ils sont à la fois en transits sur leur chemin vers le récepteur après avoir été envoyé et dans les dispositifs finaux avant d'être envoyés et après avoir été reçu. Dans ce cas, nous devons appliquer les opérations de sécurité suivantes.

- Implémentations sécurisées dans les dispositifs finaux.
- Cryptographie de Bout en Bout (E2EE) qui doit assurer les services suivants :
    1. Confidentialité des données : par l'encryptions.
    2. Intégrité : utilisant les fonctions de Hachage (SHA ou MD5).
    3. Non-répudiation : par la signature numérique.
    4. Authentification : par l'identité et mot de passe

A. La mise en œuvre sécurisée dans les dispositifs finaux

La mise en œuvre sécurisée du dispositif final est une opération de sécurité essentielle qui assure la sécurité des données dans le dispositif utilisateur. Par exemple, un smartphone ou un ordinateur de l'utilisateur doivent avoir un antivirus installé de telle sorte que Malwares qui peuvent être utilisés pour voler des données de l'utilisateur ne seront pas en mesure de le faire. Avec antivirus nous nous attendons à d'autres menaces Internet destinées à du matériel du dispositif utilisateur seront sans danger et les données utilisateur restent toujours en sécurité dans l'appareil.

- **Cryptographie de Bout en Bout (E2EE)**

E2EE est un cryptage des données utilisateur avant d'être envoyé au récepteur, l'application chat ou par courrier Crypter les données utilisateur dans le dispositif de l'expéditeur avant d'être envoyé et les données restent chiffrées à travers tous les intermédiaires jusqu'à ce qu'ils atteignent le récepteur. Après la réception, les données sont décryptées dans le dispositif de l'utilisateur. En d'autres termes, E2EE est un type de cryptage où les données ne sont accessibles que par l'expéditeur et de recevoir. Même les gestionnaires de bases de données et les détenteurs de clés de la base de données ne seront pas en mesure de lire ces données puisqu'elles sont chiffrées, et que l'émetteur et le récepteur ont accès aux clés utilisées pour le chiffrement et le déchiffrement. Avec ce type de données de cryptage partagées entre les smartphones sont garantis avec la sécurité.

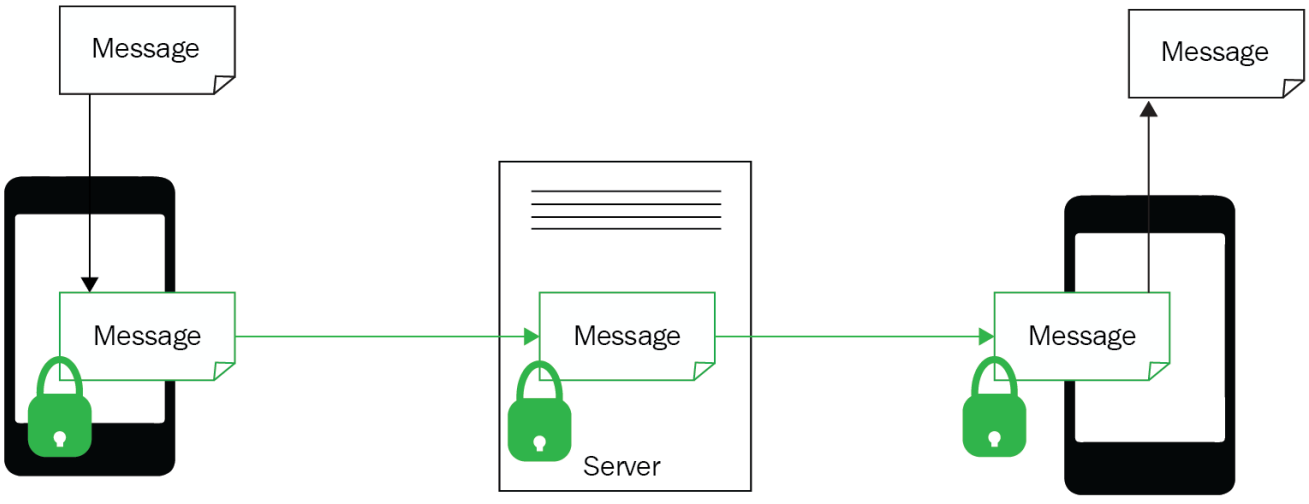


Figure 10 Cryptographie de Bout en Bout

1.4 Conclusion

Nous avons vu différentes façons où les smartphones peuvent communiquer, et dans chaque façon a sa propre méthode de sécurisation des informations afin qu'ils ne tombent pas dans de mauvaises mains des pirates. Les méthodes utilisées ont des faiblesses qui peuvent être utilisés bien par les pirates et la seule façon de repousser les pirates est d'ajouter de la complexité des méthodes utilisées pour sécuriser les informations. Le chapitre suivant va discuter différentes méthodes utilisées pour crypter des informations en détail.

CHAPITRE 2

MÉTHODES DE CRYPTOGRAPHIE UTILISÉES DANS LES COMMUNICATIONS

2.1 INTRODUCTION

Comme nous l'avons vu dans le premier chapitre, chaque méthode de communication a plusieurs méthodes de sécurisation des informations d'un appareil à l'autre, dans ce chapitre, nous verrons d'autres méthodes de sécurité / cryptage des informations utilisées dans le SMS, la communication internet/réseau en général.

Le chiffrement des données (Data Encryption) peut être défini comme l'application principale de la cryptographie ; il rend les données incompréhensibles afin de garantir leur confidentialité. Le chiffrement utilise une valeur secrète appelée **clé** ; Si vous ne connaissez pas la clé secrète, vous ne pouvez ni déchiffrer, ni apprendre des informations sur le message chiffré et aucun n'attaquant non plus.

Le cryptage moderne est divisé en deux catégories ;

- Symmetric encryption
- Asymmetric encryption

2.2 SYMMETRIC CRYPTAGE

Dans ce type de cryptage une seule clé est utilisée pour le chiffrement et le déchiffrement des données. Il est considéré par beaucoup comme le mode de chiffrement le plus faible, car il peut facilement être attaqué par des pirates.

La plupart des Techniques utilisés dans ce type de cryptage AES comprend DES et RS5.

2.2.1 BLOC CIPHER

Un chiffrement par bloc est constitué d'un algorithme de chiffrement et un déchiffrement algorithme :

- L'algorithme de chiffrement (E) prend une clé, K, et un bloc de texte en clair, P et produit un bloc de texte chiffré, C. Nous écrivons une opération de cryptage C = E (K, P).

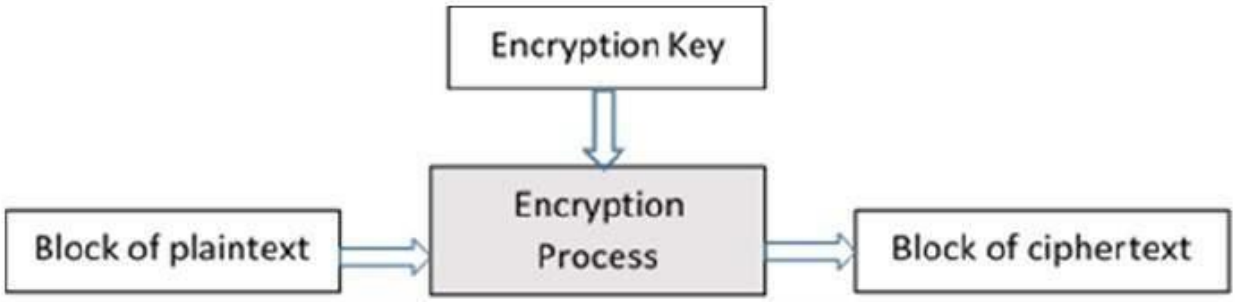


Figure 11 Exemple de procédé de chiffrement

● L'algorithme de déchiffrement (D) est l'inverse de l'algorithme de chiffrement et décrypte un message de texte en clair d'origine, P. Cette opération est écrite sous la forme P = D (K, C).

Comme ils sont inverses l'une de l'autre, les algorithmes de chiffrement et de déchiffrement impliquent généralement des opérations similaires.

Chiffrement par bloc comprend 3 opérations principales ;

- Permutation - changement de la position des bits
- Remplacement - changement d'un bit par un autre, par exemple 0-> 1.
- Ou Exclusif - l'ajout d'une clé secrète K.

2.2.1.1 Différents modes de fonctionnement

A. Electronic Codebook Mode (ECB)

Mode électronique est le livre de code façon la plus évidente d'utiliser un chiffrement par bloc.

Chiffrant.

Entrée : *k*bits principaux blocs de texte en clair de n bits K = M1M2…Mt.

Algorithme :

cj= EK (Mj).

Sortie : *n*-bit blocs de texte de chiffrement C = C1C2 ... Ct.

Décrypter.

Contribution : *k* clés blocs de bits texte de chiffrement K n bits C = C1C2 ... Ct.

Algorithme :

Mj= DK (Cj).

Sortie : *n*-bit blocs M = M1M2 plaintext ... Mt.

Pour expliquer le nom, on doit penser à ce mode comme étant défini par une table de consultation ou codebook. Considérons, par exemple, le DES, qui fonctionne sur 64 chaînes de bits (binaire). Ceux-ci décrivent, par exemple, 8 caractères ASCII 8 bits (ou en ASCII 7 bits avec un bit de contrôle de parité). Pour chaque clé K, le livre de code contient l'image cryptogramme de chacune de ces 8 chaînes de caractères comme une table de consultation. Pour chiffrer le message, le document électronique est consulté pour le codage de

Cryptogramme de chaque bloc. Notez que le nombre de ces hypothétiques lui-même est codebooks énorme - pour DES, il y a 256 clés possibles, chacune avec son propre répertoire.

Nous considérons maintenant certaines des propriétés et les limites du mode ECB. Les catégories ci-dessous sont choisis à titre de comparaison avec les modes de fonctionnement qui suivent.

- **Proprieties:**

I. **Identiques texte brut.** Le même bloc de texte en clair cartes toujours au même cryptogramme 30 bloc Cryptographie élémentaire.
II. **Dépendances chaînage.** Réordonner les blocs induit un texte clair réordonnancement des mêmes blocs de cryptogramme.
III. **Propagation d'erreur.** Une erreur dans un résultat de bloc de texte chiffré en une seule erreur de déchiffrement du bloc de texte en clair correspondant.

B. Cipher Block mode Enchaînement (CBC)

Mode de chaînage de bloc de chiffrement implique une opération de somme de bits de vecteur du bloc de message avec le bloc de texte chiffré précédent avant le chiffrement. Les blocs de texte chiffré sont initialisés avec un message choisi au hasard qui peut être transmis ouvertement, à savoir la sécurité du système de chiffrement est basé sur le secret de la clé, non pas sur le secret du vecteur d'initialisation.

Chiffrant.

Entrée: *c*lé K –bit ,*N* vecteur d'initialisation des blocs de texte en clair bits C0 n bits M = M1M2 ... Mt.

Algorithme :

cj= EK (Cj-1 ⊕Mj).

Sortie : *n* blocs -bit de cryptogramme C = C0C1 ... Ct.

Décrypter.

Contribution : *k*bits principaux blocs de texte chiffré K n bits C = C0C1 ... Ct.

Algorithme :

Mj= Cj-1 ⊕DK (Cj).

Sortie :

n-bit blocs M = M1M2 plaintext ... Mt.

❖ **Proprieties:**

I. **Identiques texte brut.** La même séquence de blocs de texte chiffré résultat lorsque la même touche et le même vecteur d'initialisation sont utilisés.

II. **Dépendances chaînage.** Le mécanisme de chaînage provoque Cj à dépendre Cj-1 et Mj, alors chiffrement n'est pas indépendante de réordonnancèrent.

III. **Propagation d'erreur.** Une erreur dans un bloc de texte chiffré Cj Cj affecte de déchiffrement et Cj + 1. Pour un algorithme de chiffrement raisonnable, une seule erreur sur les bits affecte 50% des bits dans le bloc de texte en clair décryptéM'_j, Alors que l'erreur binaire affecte seulement peu deM'_{j+1}. 3. récupération d'erreur. Le système de chiffrement est dit auto-récupération, en ce sens que, si une erreur Cj résulte en clair décrypté de manière incorrecte Mj0 et Mj0 + 1, le texte chiffré Cj + 2 déchiffre correctement à Mj0 + 2 = Mj + 2.

C. Cipher Feedback Mode (CFB)

Le mode de rétroaction de chiffrement permet de traiter des blocs de taille r <n à la fois. La valeur typique pour r est 1, alors n peut être de taille 64, en utilisant DES.

Chiffrant.

Entrée: K *c*lé K –bit ,n *v*ecteur d'initialisation I1 bits des blocs de texte en clair r bits M = M1M2 ... Mt.

Algorithme :

Cj= Mj ⊕Lr (EK (Ij)), Ij + 1 = Rn-r (Ij) || Cj,

Où Lr et Rn-r sont les opérateurs qui prennent les plus à gauche r bits et le plus à droite n - r bits, et || est l'opérateur de concaténation.

Le vecteur Ij doit être considéré comme un registre à décalage, un bloc de n bits de memoire qui stocke un état de l'algorithme. La formation de Ij + 1 est un décalage à gauche par r de ce bloc, en écartant les bits les plus à gauche R, avec les bits les plus à droite r remplacés par Cj.

Décrypter.

Entrée: *k* clé K –bit, *n* vecteur d'initialisation -bit I1 blocs de texte chiffré r bits C = C1C2 ... Ct.

Algorithme :

Computer I1, ..., comme dans l'algorithme de chiffrement, qui peut être généré indépendamment du texte du message déchiffré, puis calculer

mj= Cj ⊕Lr (EK (Ij)).

Notez que déchiffrage CFB ne nécessite que le chiffrement par bloc EK, pas la carte déchiffrage bloc inverse DK.

❖ **Proprieties:**

I. **Identiques texte brut.** La même séquence de blocs de texte chiffré produit lorsque là même 32 Cryptographie élémentaire clé et vecteur d'initialisation est utilisé. Modification du vecteur d'initialisation change le cryptogramme.

II. **Dépendances chaînage.** Bloc cryptogramme Cj dépend des blocs précédents Mj-plaintext 1, ..., M1, ainsi que Mj, de sorte que les blocs de texte chiffré ne sont pas Réorganiser indépendants.

III. **Propagation d'erreur.** Une erreur dans Cj affecte le déchiffrement des prochaines [n / r] des blocs de texte en clair. Le texte brut récupéré Mj0 sera différent de Mj à exactement les bits pour lesquels Cj était dans l'erreur. Ces erreurs de bit apparaissent dans les blocs suivantsM'_{j+k} à des positions traduites.

IV. **Récupération d'erreur.** Déchiffrement correct exige que le registre à décalage pour être correcte, pour laquelle les précédentes [n / r] des blocs de texte chiffré sont nécessaires. Le déchiffrement est auto-récupération d'erreurs, mais seulement après que les blocs [n / r] (à peu près les mêmes n bits du bloc de texte chiffré en erreur).

V. **Débit.** Le taux de chiffrement et déchiffrement est réduite d'un facteur n / r, qui est, pour tous les r bits' de sortie de l'algorithme doit effectuer une opération de chiffrement n bits.

D. Open Feedback Mode (OFB)

Le mode de contre-réaction de sortie a une utilisation similaire que le mode de chiffrement à rétroaction, mais est pertinent pour des applications pour lesquelles la propagation d'erreur doit être évitée. Le mode de contre-réaction de sortie est un exemple de chiffrement de flux synchrone (construit à partir d'un chiffrement par bloc), dans lequel la séquence clé est créée indépendamment du flux de texte en clair.

Chiffrant.

Entrée: *c*lé K –bit , *n* vecteur d'initialisation bits I0 blocs de texte en clair r bits M = M1M2 ... Mt.

Algorithme :

ij= EK (Ij-1) Cj = Mj ⊕Lr (Ij)

Décrypter.

Contribution: *k*clé K –bit, *n*vecteur d'initialisation -bit I0 blocs de texte chiffré r bits C = C1C2 ... Ct.

Algorithme :

Computer I1, ..., comme dans l'algorithme de chiffrement.

mj= Cj ⊕Lr (Ij)

❖ **Proprieties:**

I. **Identiques texte brut.** Les mêmes commentaires pour Radio-Canada et à la BFC appliquent.

II. **Dépendances chaînage.** La sortie cryptogramme est l'ordre à charge, mais le keystream I1, I2, ... est indépendant plaintext.

III. **Propagation d'erreur.** Une erreur dans un bit de cryptogramme affecte seulement que peu de texte clair. 4. récupération d'erreur. Le chiffrement est auto-synchronisation et les erreurs de bit dans un bloc de texte chiffré affecte seulement que peu du clair récupéré. Il récupère immédiatement des erreurs de bit, mais les pertes de bits affectent l'alignement.

IV. **Débit.** Comme BFC, le taux de chiffrement et déchiffrement est réduite d'un facteur n / r, mais les vecteurs Ij peuvent être pré calculées à partir de K et I0, indépendamment des blocs de texte chiffré.

2.2.2 Advanced Encryption Standard (AES)

Algorithme Advanced Encryption Standard (AES) est l'un sur l'algorithme de chiffrement par bloc le plus commun et largement symétrique utilisé dans le monde entier. Cet algorithme a une structure propre pour chiffrer et déchiffrer les données sensibles et est appliquée dans le matériel et les logiciels partout dans le monde. Il est extrêmement difficile pour les pirates pour obtenir les données réelles lors du cryptage par algorithme AES. Jusqu'à ce jour n'est pas une preuve de Crake cet algorithme. AES a la capacité de traiter trois tailles différentes clés tels que AES 128 bits, 192 et 256 et chacun de ces chiffrements a la taille de bloc de 128 bits. Ce document donne un aperçu de l'algorithme AES et expliquer plusieurs caractéristiques essentielles de cet algorithme dans les détails et la démonstration des recherches antérieures qui ont fait à ce sujet avec comparant à d'autres algorithmes tels que DES, 3DES, Blowfish, etc.

2.2.2.1 STRUCTURE DE BASE D'ALGORITHME AES

AES utilise deux techniques communes pour chiffrer et déchiffrer des données appelées réseau de substitution et de permutation (Substitution and Permutation Network, SPN). SPN est un certain nombre d'opérations mathématiques qui sont effectuées dans les algorithmes de chiffrement par bloc . AES a la capacité de traiter de 128 bits (16 octets) comme une taille de bloc de texte en clair fixe. Ces 16 octets sont représentés dans la matrice 4x4 et AES fonctionne sur une matrice d'octets. En outre, une autre caractéristique essentielle en AES est le nombre de tours. Le nombre de tours est fondé sur la longueur de la clé. Il existe trois tailles de clés différentes sont utilisées par l'algorithme AES pour chiffrer et déchiffrer des données telles que (128, 192 ou 256 bits). Les tailles de clé décident du nombre de tours tels que AES utilise 10 tours pour les clés 128 bits, 12 tours pour les clés de 192 bits et 14 tours pour clés de 256 bits .

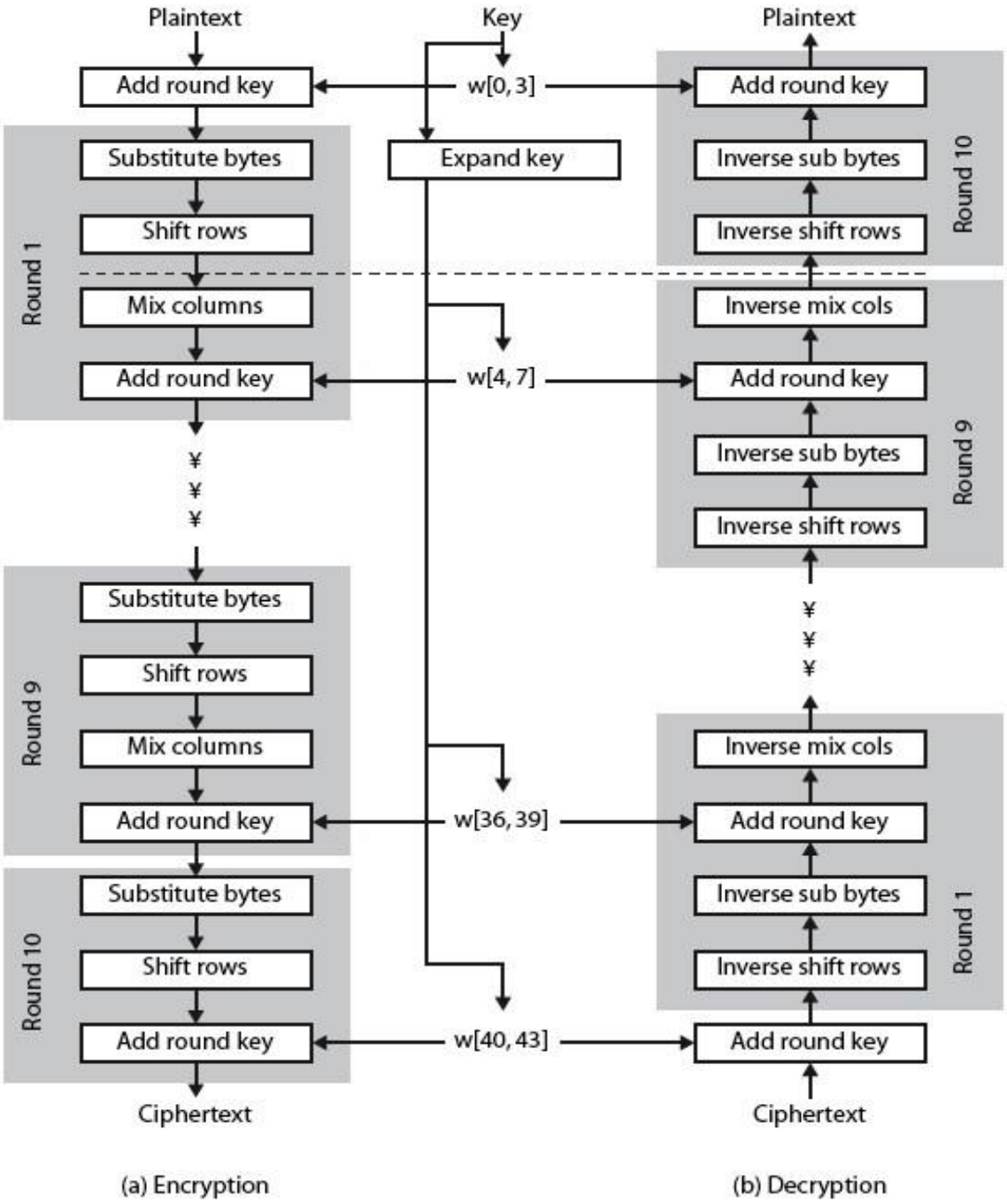
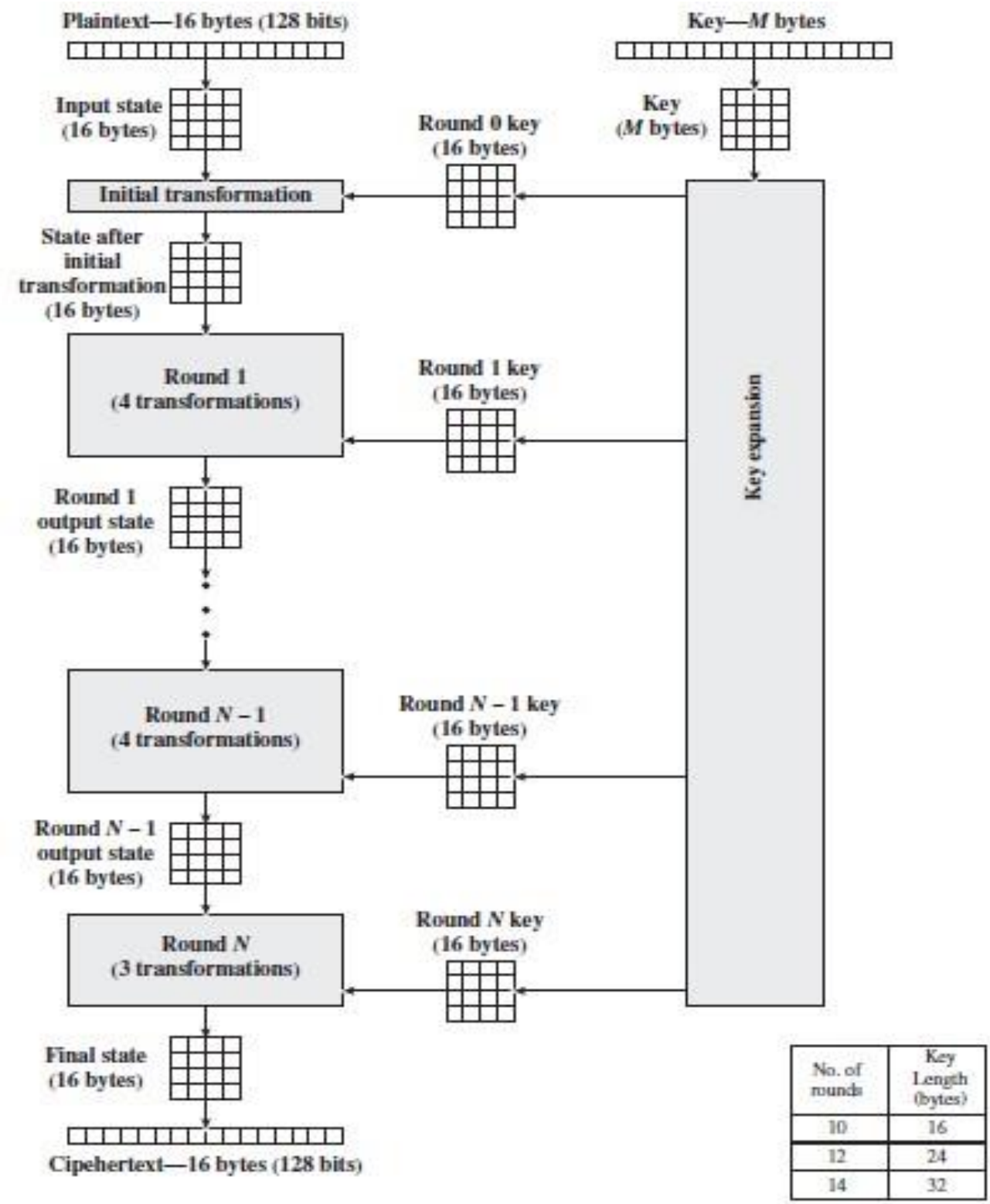


Figure 12 Structure AES

2.2.2.2 PROCESSUS DE CHIFFREMENT

Pour fournir la meilleure sécurité possible, AES a une structure particulière pour chiffrer les données. Pour ce faire, il repose sur un certain nombre de tours et à l'intérieur de chaque cycle de quatre comprennent des sous-processus. Chaque tour se compose des quatre étapes suivantes pour chiffrer le bloc 128 bits



No. of rounds	Key Length (bytes)
10	16
12	24
14	32

Figure 13 Processus de chiffrement

A. Substitute Bytes Transformation

La première étape de chaque cycle commence par la transformation sous des octets. Cette étape est en fonction de S-box non linéaire de substituer un octet dans l'état à un autre octet. Selon la diffusion et de la confusion des principes de Shannon pour la conception de l'algorithme cryptographique, il a un rôle important pour obtenir beaucoup plus de sécurité. Par exemple, en AES si nous avons hexa 53 dans l'état, il doit remplacer à hexa ED. ED créé à partir de l'intersection de 5 et 3. Pour octets restants de l'Etat doivent effectuer cette opération. [6]

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	63	7C	77	7B	F2	6B	6F	C5	30	01	67	2B	FE	D7	AB	76
1	CA	82	C9	7D	FA	59	47	F0	AD	D4	A2	AF	9C	A4	72	C0
2	B7	FD	93	26	36	3F	F7	CC	34	A5	E5	F1	71	D8	31	15
3	04	C7	23	C3	18	96	05	9A	07	12	80	E2	EB	27	B2	75
4	09	83	2C	1A	1B	6E	5A	A0	52	3B	D6	B3	29	E3	2F	84
5	53	D1	00	ED	20	FC	B1	5B	6A	CB	BE	39	4A	4C	58	CF
6	D0	EF	AA	FB	43	4D	33	85	45	F9	02	7F	50	3C	9F	A8
7	51	A3	40	8F	92	9D	38	F5	BC	B6	DA	21	10	FF	F3	D2
8	CD	0C	13	EC	5F	97	44	17	C4	A7	7E	3D	64	5D	19	73
9	60	81	4F	DC	22	2A	90	88	46	EE	B8	14	DE	5E	0B	DB
A	E0	32	3A	0A	49	06	24	5C	C2	D3	AC	62	91	95	E4	79
B	E7	C8	37	6D	8D	D5	4E	A9	6C	56	F4	EA	65	7A	AE	08
C	BA	78	25	2E	1C	A6	B4	C6	E8	DD	74	1F	4B	BD	8B	8A
D	70	3E	B5	66	48	03	F6	0E	61	35	57	B9	86	C1	1D	9E
E	E1	F8	98	11	69	D9	8E	94	9B	1E	87	E9	CE	55	28	DF
F	8C	A1	89	0D	BF	E6	42	68	41	99	2D	0F	B0	54	BB	16

Table 1 AES S-box

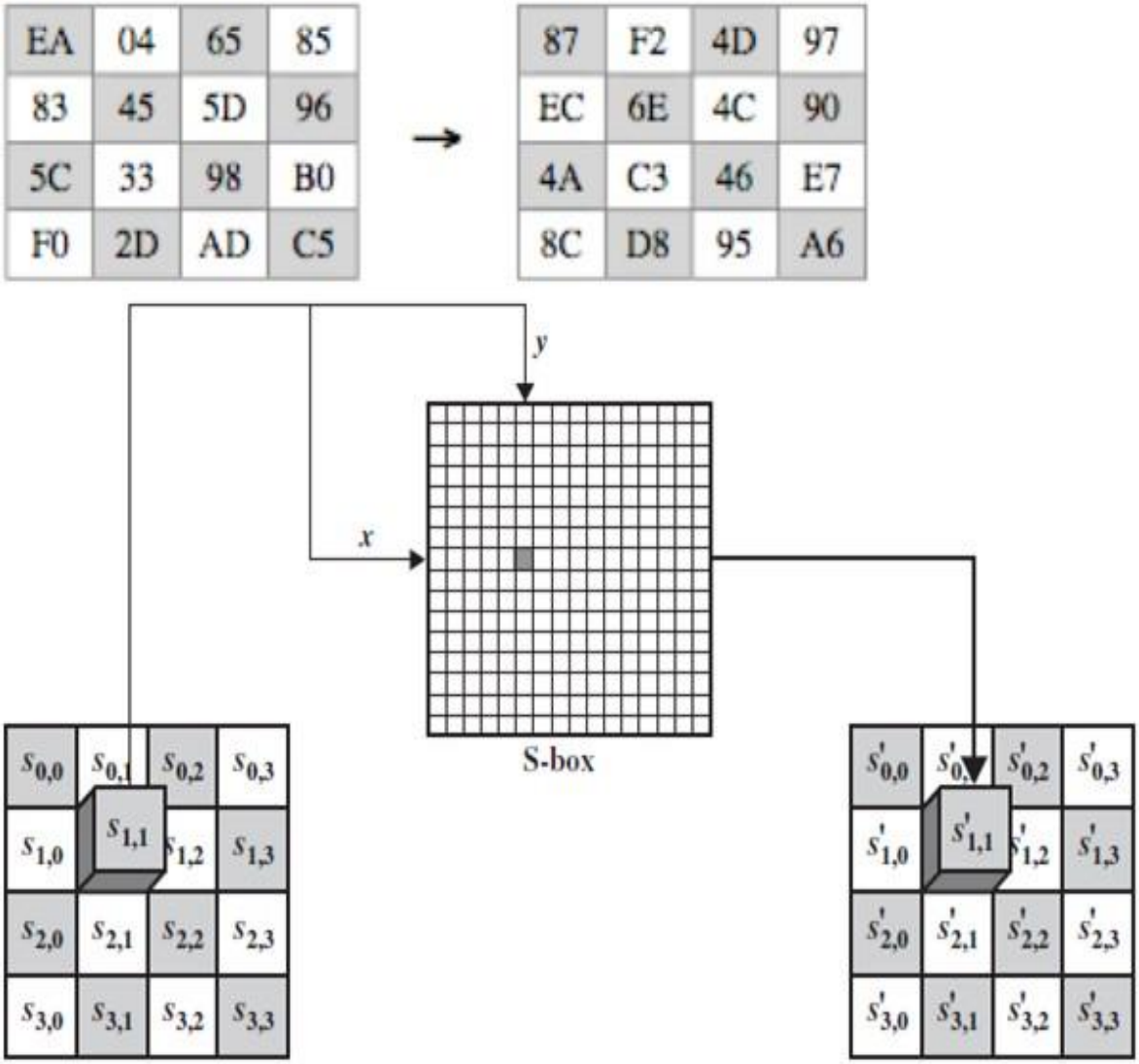


Figure 14 transformation d'octet de substitution

B.Shift-Row Transformation

L'idée principale de cette étape consiste à décaler les octets de l'état de manière cyclique à la gauche de chaque rangée plutôt que le numéro de rangée zéro. Dans ce processus, les octets du numéro de la ligne zéro restent et ne porte pas toute permutation. Dans la première rangée un seul octet est décalé circulaire à gauche. La deuxième rangée est décalée de deux octets vers la gauche. La dernière rangée est décalée de trois octets à la gauche. La taille du nouvel état ne change pas qui reste la même taille d'origine de 16 octets, mais décalée de la position des octets dans l'état comme illustré sur la figure 15. [7][10]

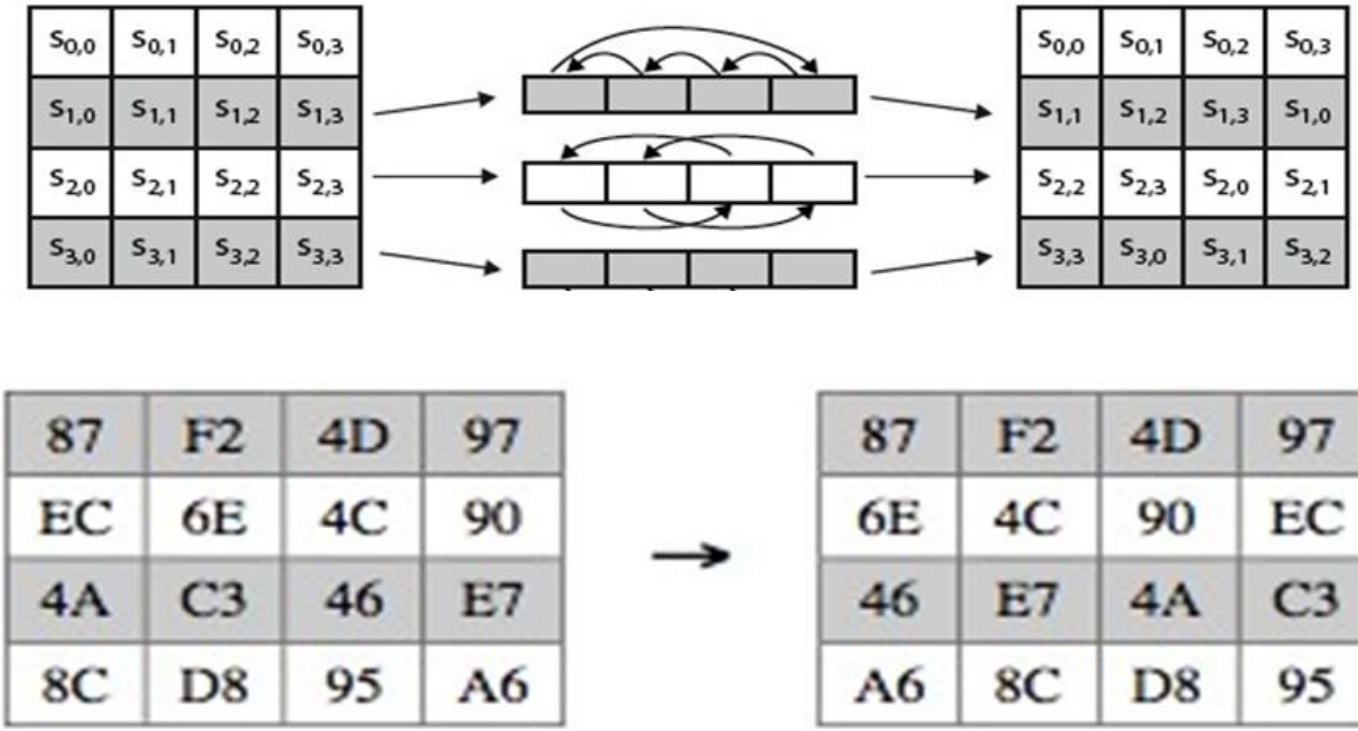


Figure 15 Les lignes de décalage

C. Transformation MixColumn

Dans le MixColumn la multiplication se fait de l'Etat. Chaque octet d'une ligne de transformation de la matrice par multiplication de chaque valeur (octet) de la colonne d'état. Dans un autre mot, chaque ligne de la matrice de transformation doit multiplier par chaque colonne de l'état. Les résultats de ces multiplications sont utilisés avec XOR pour produire un nouveau quatre octets pour l'état suivant. Dans cette étape, la taille de l'état ne change pas qui est restée à la taille originale 4x4 comme représenté sur la Fig. 16.

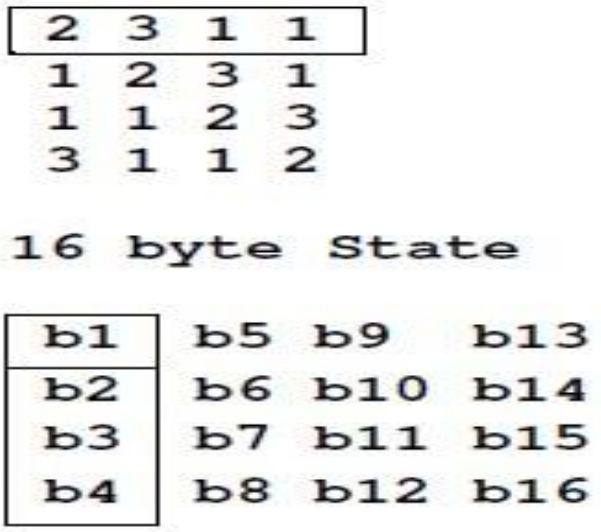


Figure 16 Multiplication Matrice

b1 = (b1 * 2) XOR (b2 * 3) XOR (b3 * 1) XOR (b4 * 1)

Et ainsi de suite jusqu'à ce que toutes les colonnes de l'état sont épuisées.

D. Transformation AddRoundKey

AddRoundKey est l'étape la plus essentielle dans l'algorithme AES. La clé et les données d'entrée (appelées aussi l'état) sont structurées dans une matrice de 4x4 d'octets [8][10]. La Figure 20 montre comment les données de clé et d'entrée de 128 bits sont répartis dans les matrices d'octets. AddRoundKey a la capacité de fournir beaucoup plus de sécurité lors de cryptage des données. Cette opération est basée sur la création de la relation entre la clé et le texte chiffré. Le texte chiffré est arrivant de la précédente étape. Le AddRoundKey sortie repose exactement sur la clé qui est indiquée par les utilisateurs [9][10]. En outre, à l'étape de la sous-clé est également utilisé et combiné avec l'état. La principale clé est utilisée pour calculer la sous-clé à chaque tour en utilisant le calendrier clé de Rijndael. La taille de sous-clé et de l'état est le même. La sous-clé est ajoutée en combinant chaque octet de l'état de l'octet correspondant de la sous-clé en utilisant XOR au niveau du bit.

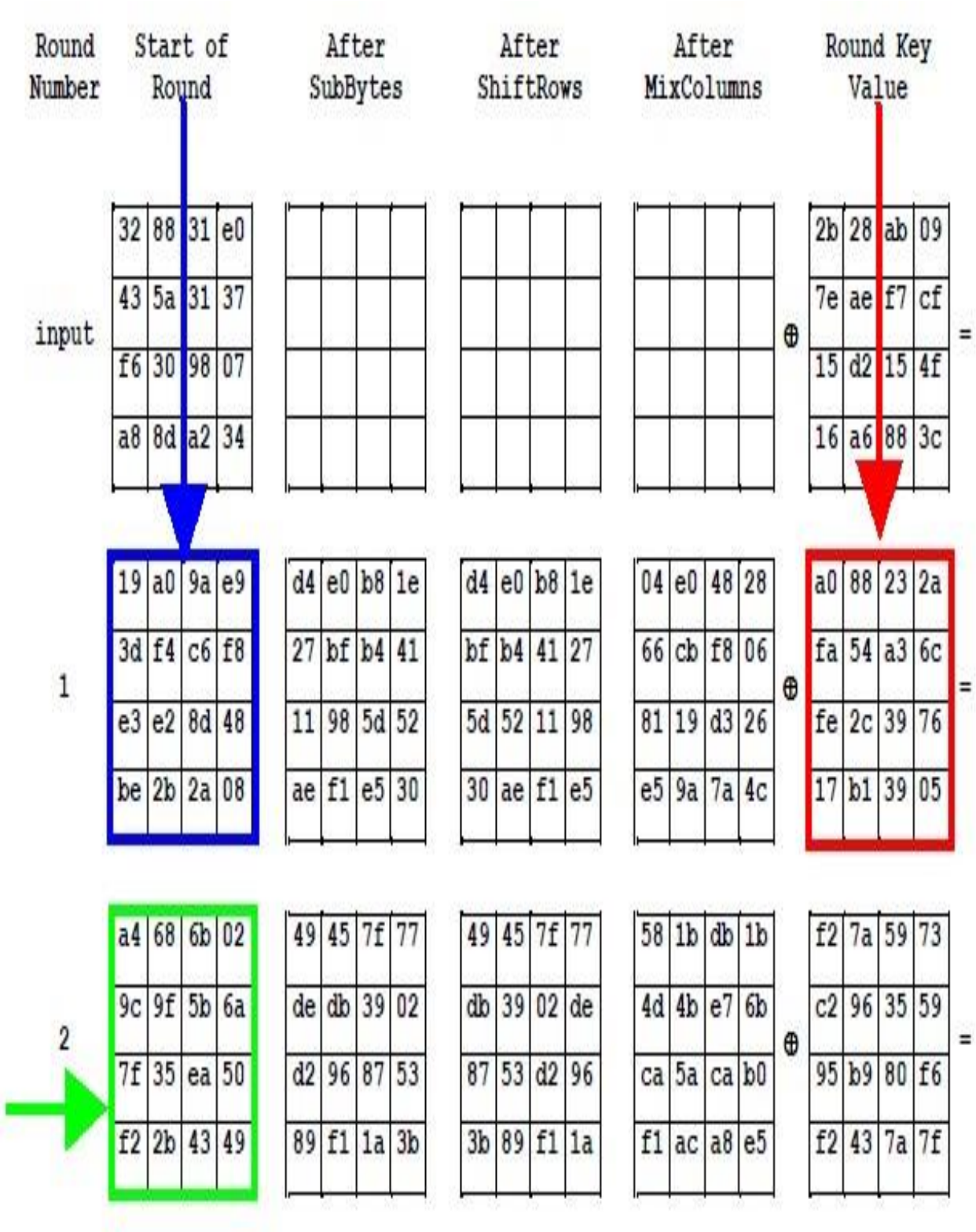


Figure 17 Ajouter ronde clé

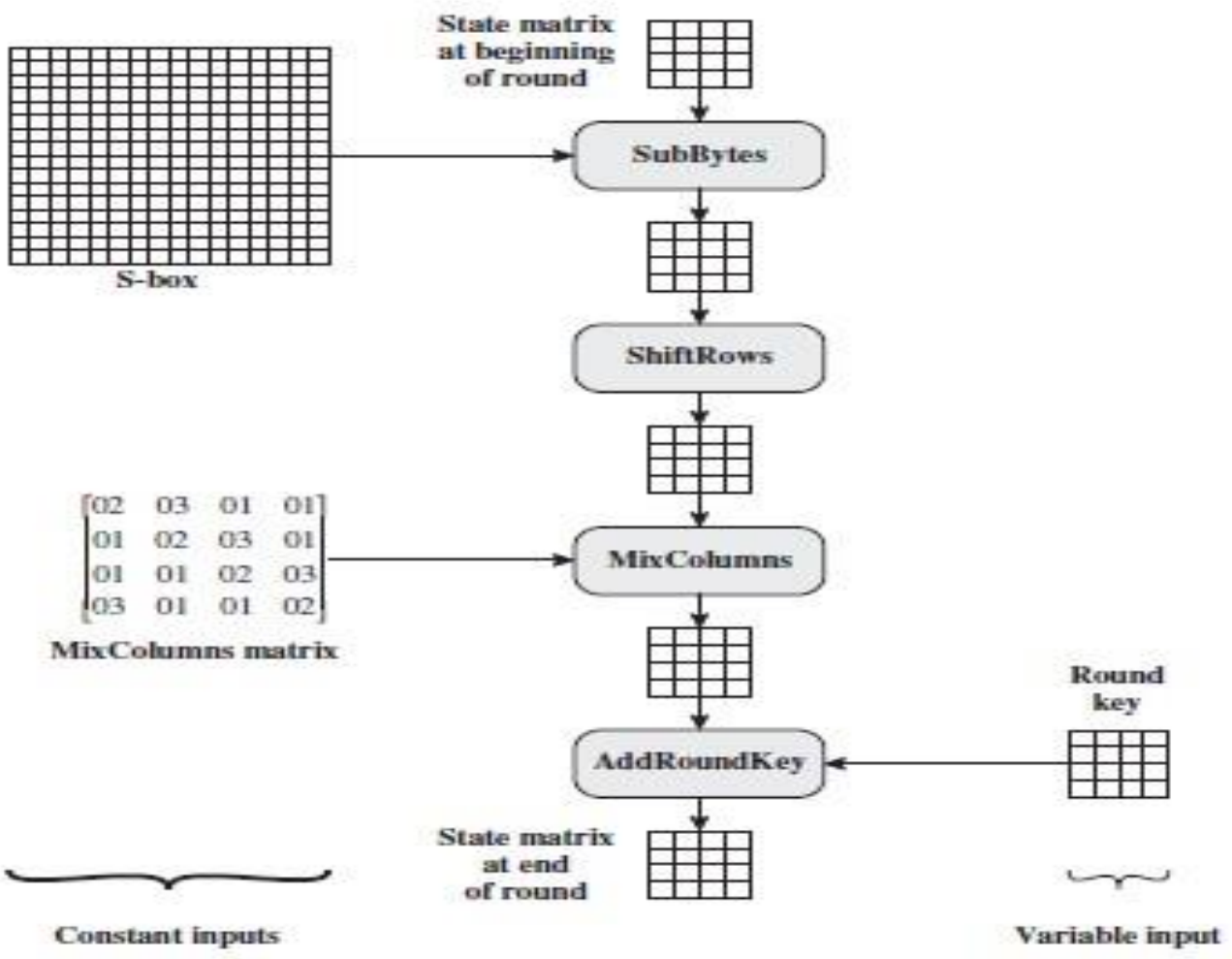


Figure 18 Entrées pour Occupation Simple AES Round

E. AES KEY EXPANSION

Algorithme AES est basée sur l'expansion de la clé AES pour chiffrer et déchiffrer les données. Pour chaque ronde il y a une nouvelle clé. Dans cette section se concentre sur AES technique clé d'extension. La routine clé d'expansion crée mot clés tour par mot, où un mot est un tableau de quatre octets. La routine crée 4x (Nr + 1) mots. Où est Nr le nombre de rondes. Le processus est le suivant :

La clé de chiffrement (clé initiale) est utilisée pour créer les quatre premiers mots. La taille de la clé se compose de 16 octets (K0 à K15), comme illustré sur la figure 19 qui représente dans un tableau. Les quatre premiers octets (k0 à k3) représentent comme w0, les quatre octets suivants (K4 à K7) dans la première colonne représente comme w1, et ainsi de suite. On peut utiliser l'équation particulière pour calculer et trouver les clés dans chaque ronde facilement comme suit :

- K [n]: w [i] = K [n-1]: w [i] XOR k [n]: w [i].

Cette équation utilise pour trouver une clé pour chaque tour plutôt que w0. Pour w0, nous devons utiliser l'équation particulière qui est différente de l'équation ci-dessus.

- K [n]: w0 = k [n-1]: w0 XOR Sub Byte (k [n-1]: w3 >> 8) XOR Rcon [i].

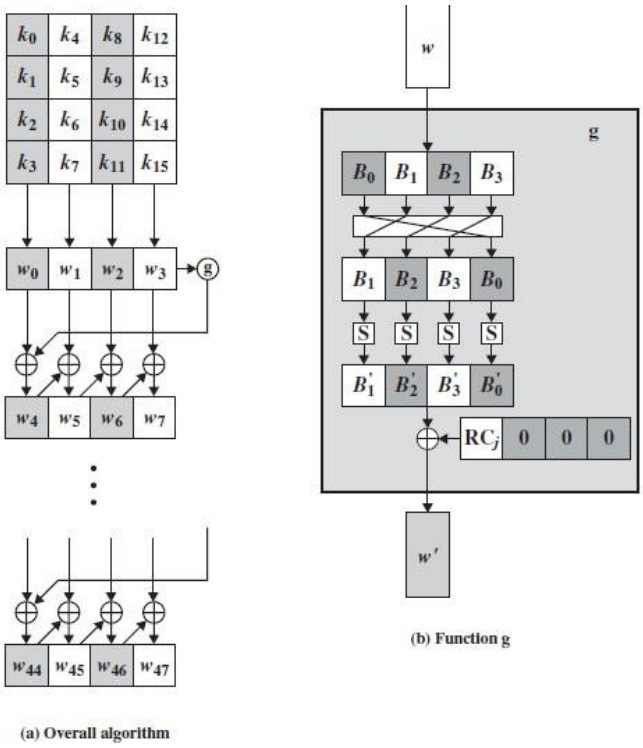


Figure 19 Processus d'AES Key Expansion

• AES Key Expansion Example

K1:

W0 = 0f 15 71 c9

W1 = 47 D9 e8 59

W2 = 0C b7 ad e8

W3 = af 7F 67 98

Comment trouver K2?

K2 = w0 = k1: w0 XOR Sub Byte (k1: w3 >> 8) XOR Rcon [2]

0f 15 71 c9 XOR Sub Byte (af 7f 67 98 >> 8) XOR Rcon [2]

Rcon [2] de la fonction auxiliaire = 02 00 00 00

0f 15 71 c9 XOR Sub Byte (7f 67 98 af) XOR 02 00 00 00

0f 15 71 c9 XOR D2 85 46 79 XOR 02 00 00 00

0f 15 71 c9 XOR d0 85 46 79 K2 = w0 = df q0 37 b0

K2: w1 = k1: w1 XOR K2: w0

47 D9 e8 59 XOR df q0 37 b0

K2: w1 = 98 49 df eq

K2: w2 = k1: w2 XOR K2: w1

Dans cet exemple, nous avons trouvé W0 et W1. Dans la même manière que nous pouvons trouver W2 et W3.

w0	w1	w2	w3
0f	47	0c	af
15	d9	b7	7f
71	e8	ad	67
c9	59	e8	98

W0	W1	W2	W3
df	98		
q0	49		
37	df		
b0	eq		

Figure 20 AES Key Expansion

Key Words	Auxiliary Function
w0 = 0f 15 71 c9 w1 = 47 d9 e8 59 w2 = 0c b7 ad w3 = af 7f 67 98	RotWord(w3)= 7f 67 98 af = x1 SubWord(x1)= d2 85 46 79 = y1 Rcon(1)= 01 00 00 00 y1 ⊕ Rcon(1)= d3 85 46 79 = z1
w4 = w0 ⊕ z1 = dc 90 37 b0 w5 = w4 ⊕ w1 = 9b 49 df e9 w6 = w5 ⊕ w2 = 97 fe 72 3f w7 = w6 ⊕ w3 = 38 81 15 a7	RotWord(w7)= 81 15 a7 38 = x2 SubWord(x4)= 0c 59 5c 07 = y2 Rcon(2)= 02 00 00 00 y2 ⊕ Rcon(2)= 0e 59 5c 07 = z2
w8 = w4 ⊕ z2 = d2 c9 6b b7 w9 = w8 ⊕ w5 = 49 80 b4 5e w10 = w9 ⊕ w6 = de 7e c6 61 w11 = w10 ⊕ w7 = e6 ff d3 c6	RotWord(w11)= ff d3 c6 e6 = x3 SubWord(x2)= 16 66 b4 8e = y3 Rcon(3)= 04 00 00 00 y3 ⊕ Rcon(3)= 12 66 b4 8e = z3
w12 = w8 ⊕ z3 = c0 af df 39 w13 = w12 ⊕ w9 = 89 2f 6b 67 w14 = w13 ⊕ w10 = 57 51 ad 06 w15 = w14 ⊕ w11 = b1 ae 7e c0	RotWord(w15)= ae 7e c0 b1 = x4 SubWord(x3)= e4 f3 ba c8 = y4 Rcon(4)= 08 00 00 00 y4 ⊕ Rcon(4)= ec f3 ba c8 = 4
w16 = w12 ⊕ z4 = 2c 5c 65 f1 w17 = w16 ⊕ w13 = a5 73 0e 96 w18 = w17 ⊕ w14 = f2 22 a3 90 w19 = w18 ⊕ w15 = 43 8c dd 50	RotWord(w19)= 8c dd 50 43 = x5 SubWord(x4)= 64 c1 53 1a = y5 Rcon(5)= 10 00 00 00 y5 ⊕ Rcon(5)= 74 c1 53 1a = z5
w20 = w16 ⊕ z5 = 58 9d 36 eb w21 = w20 ⊕ w17 = fd ee 38 7d w22 = w21 ⊕ w18 = 0f cc 9b ed w23 = w22 ⊕ w19 = 4c 40 46 bd	RotWord(w23)= 40 46 bd 4c = x6 SubWord(x5)= 09 5a 7a 29 = y6 Rcon(6)= 20 00 00 00 y6 ⊕ Rcon(6)= 29 5a 7a 29 = z6
w24 = w20 ⊕ z6 = 71 c7 4c c2 w25 = w24 ⊕ w21 = 8c 29 74 bf w26 = w25 ⊕ w22 = 83 e5 ef 52 w27 = w26 ⊕ w23 = cf a5 a9 ef	RotWord(w27)= a5 a9 ef cf = x7 SubWord(x6)= 06 d3 df 8a = y7 Rcon(7)= 40 00 00 00 y7 ⊕ Rcon(7)= 46 d3 df 8a = z7
w28 = w24 ⊕ z7 = 37 14 93 48 w29 = w28 ⊕ w25 = bb 3d e7 f7 w30 = w29 ⊕ w26 = 38 d8 08 a5 w31 = w30 ⊕ w27 = f7 7d a1 4a	RotWord(w31)= 7d a1 4a f7 = x8 SubWord(x7)= ff 32 d6 68 = y8 Rcon(8)= 80 00 00 00 y8 ⊕ Rcon(8)= 7f 32 d6 68 = z8
w32 = w28 ⊕ z8 = 48 26 45 20 w33 = w32 ⊕ w29 = f3 1b a2 d7 w34 = w33 ⊕ w30 = cb c3 aa 72 w35 = w34 ⊕ w32 = 3c be 0b 38	RotWord(w35)= be 0b 38 3c = x9 SubWord(x8)= ae 2b 07 eb = y9 Rcon(9)= 1B 00 00 00 y9 ⊕ Rcon(9)= b5 2b 07 eb = z9
w36 = w32 ⊕ z9 = fd 0d 42 cb w37 = w36 ⊕ w33 = 0e 16 e0 1c w38 = w37 ⊕ w34 = c5 d5 4a 6e w39 = w38 ⊕ w35 = f9 6b 41 56	RotWord(w39)= 6b 41 56 f9 = x10 SubWord(x9)= 7f 83 b1 99 = y10 Rcon(10)= 36 00 00 00 y10 ⊕ Rcon(10)= 49 83 b1 99 = z10
w40 = w36 ⊕ z10 = b4 8e f3 52 w41 = w40 ⊕ w37 = ba 98 13 4e w42 = w41 ⊕ w38 = 7f 4d 59 20 w43 = w42 ⊕ w39 = 86 26 18 76	

Figure 21 Fonction auxiliaire

• Cryptage AES -Exemple

Pour plus d'expliquer les principales étapes de cryptage AES, prendre un exemple pour la première ronde pour montrer comment chiffrer les données en utilisant l'algorithme AES. Nous avons un texte en clair : AES utilise une matrice.

Tout d'abord, nous devons transformer ce texte en hexadécimal.

Plaintext	Hexadecimal
A	**00**
E	**04**
S	**12**
U	**14**
S	**12**
E	**04**
S	**12**
A	**00**
M	**0C**
A	**00**
T	**13**
R	**11**
I	**08**
X	**23**
Z	**19**
Z	**19**

Table 2 : Texte en clair (Plaintext) Convertir en Hexadécimal

- D'autre part, la création d'une matrice qui repose sur les octets qui sont obtenus à partir du tableau ci-dessus, comme indiqué ci-dessous :

00	12	0C	08
04	04	00	23
12	12	13	19
14	00	11	19

Table3 État de la matrice (matrix state)

- Troisièmement, SubByte : Cette étape repose sur AES S-box, mais avant d'utiliser SubByte la clé et cette matrice (aussi appelée l'état) sont structurés dans une matrice 4x4 d'octets en utilisant l'opération XOR comme suit :

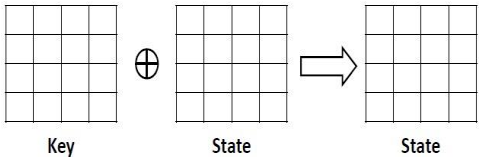


Figure 22Ajouter ronde clé étape

- La deuxième étape est ShiftRows. Il a expliqué plus haut. L'étape la plus importante est MixColumns. Chaque valeur de la colonne est finalement multipliée contre toutes les valeurs de la matrice dans un domaine particulier (Galois Field).

63	C9	FE	30		02	03	01	01
F2	63	26	F2	X	01	02	03	01
7D	D4	C9	C9		01	01	02	03
D4	FA	63	82		03	01	01	02

Figure 23 Multiplier deux états

Calculate:

63 * 02 + F2 *03 + 7D * 01 + D4 * 01

63 * 02 = 0110 0011 *02 = 1100 0110

F2 * 03 = F2 * 02 + F2 * 01

= 1111 0010 * 02 = 11100101 XOR 1B = 11100101

XOR 0001 1011

F2 * 02 = 1111 1111

F2 = 1111 * 01 0010 * 01 = 1111 0010

F2 * 02 + F2 * 01 = 0000 = 1101 F2 * 03

7D * 01 = 0111 1101

D4 * 01 = 1101 0100

63 * 02 + F2 * 03 + 7D * 01 + D4 * 01

11000110 + 00001101 + 01111101 + 11010100 01100010 = 62

Après avoir calculé tous les octets, on peut obtenir l'état comme suit. Dans cet exemple, nous avons calculé un seul octet de l'état, les octets restants ont les mêmes procédures.

62	02	27	26
CF	92	91	0D
0C	OC	F4	D
99	18	30	74

Figure 24 Multiplier deux Etats

• La dernière étape de la première ronde est AddRound Key. Cette étape crée sous forme de nouvel état de MixColumn avec 128 bits de la clé ronde par XOR.

2.2.2.3 PROCESSUS DE DÉCRYPTAGE

Le déchiffrement est le processus permettant d'obtenir les données d'origine qui ont été chiffrées. Ce processus est basé sur la clé reçue de l'expéditeur des données. Les processus de déchiffrement d'un système AES sont similaires au processus de chiffrement dans l'ordre inverse et l'expéditeur et le destinataire disposent de la même clé pour chiffrer et déchiffrer les données. La dernière étape d'une étape de décryptage comprend trois étapes, telles que InvShiftRows, InvSubBytes et AddRoundKey, comme illustré à la Fig. 26.

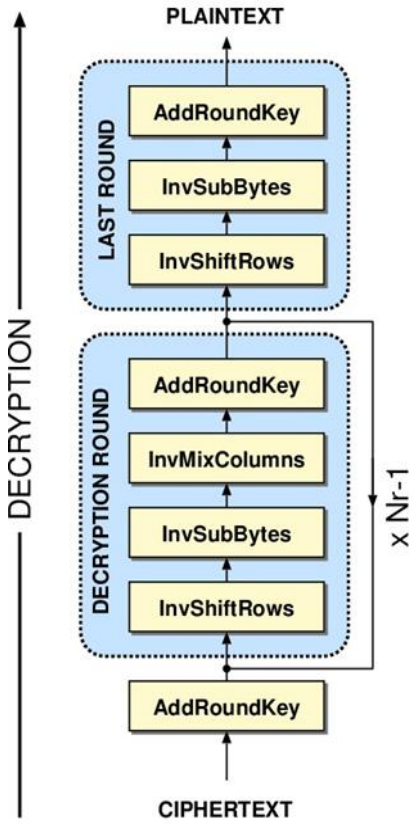


Figure 25 Processus de décryptag

- **ZONES DE MISE EN ŒUVRE**

Algorithme AES est l'un de l'algorithme le plus puissant qui sont largement utilisés dans différents domaines partout dans le monde. Cet algorithme permet plus rapide que les algorithmes DES et 3DES pour crypter et décrypter les données. En outre, il est utilisé dans de nombreux protocoles de cryptographie tels que Socket

Layer Security (SSL) et Transport Layer Security Protocol pour fournir beaucoup plus de sécurité des communications entre le client et le serveur sur Internet. Avant algorithme AES a publié deux protocoles pour chiffrer et déchiffrer les données sur l'algorithme DES compté, mais après avoir comparu devant une vulnérabilité de cet algorithme l'Internet Engineering Task Force (IETF) a décidé de remplacer le DES à l'algorithme AES. AES peut également être trouvée dans la plupart des applications et des appareils modernes qui ont besoin d'une fonctionnalité de cryptage tels que WhatsApp Messenger Facebook et Intel et AMD processeur et les périphériques Cisco comme routeur, commutateur,

Algorithme (AES) est l'un de l'algorithme efficace et il est largement soutenu et adopté le matériel et les logiciels. Cet algorithme permet de traiter différentes tailles de clés tels que 128, 192 et 256 bits avec chiffrement bloc 128 bits. Dans cet article, explique un certain nombre de caractéristiques importantes de l'algorithme AES et présente des recherches précédentes qui ont fait sur elle pour évaluer les performances d'AES pour chiffrer les données sous différents paramètres. Selon les résultats obtenus à partir des recherches

montre que AES a la capacité d'assurer la sécurité beaucoup plus par rapport à d'autres algorithmes tels que DES, 3DES, etc.

2.3 CYPTOGRAPHY ASYMETRIQUE

Dans le chiffrement asymétrique, il y a deux clés : l'une pour crypter et un autre pour décrypter. La clé de chiffrement est appelé une clé publique et est généralement considéré comme la disposition du public à tous ceux qui veulent vous envoyer des messages cryptés.[10]

La clé de déchiffrement, cependant, doit rester secret et est appelé une clé privée.

Il existe plusieurs algorithmes de chiffrement asymétrique ;

- Ceux basé à factorisation (RSA)
- Ceux basés dans l'algorithme discret (DH, ELGAMAL)

2.3.1 RSA

Inventé par Rivest, Shamir et Adleman du MIT en 1977, il est connu et largement utilisé système clé publique. Il offre la meilleure sécurité en raison du coût de l'affacturage grand nombre

Nombres premiers

Les nombres premiers ont seulement diviseurs de 1 et de soi, ils ne peuvent pas être écrits en tant que produit d'autres numéros.

Note : 1 est premier, mais il n'est généralement pas d'intérêt

Exemple de liste des nombres premiers est inférieur à 200 :

3 5 7 2 11 13 17 19 23 29 31 37 41 43 47 53 59 61

67 71 73 79 83 89 97 103 107 109 101 113 127 131

139 149 151 137 157 163 167 173 179 181 191 193 197 199

2.3.1.1 RSA Key Setup

Chaque utilisateur génère une paire de clés publique / privée par:

- La sélection de deux grands nombres premiers au hasard : p, q
- Un module de système informatique N = pq
  - RSA recommande actuellement un module qui est au moins 768 bits' à long
- calculi $\phi = (p-1)(q-1)$
- Sélectionner au hasard la clé de cryptage e
  - Où $1 < e < \phi$, $pgcd(e, \phi) = 1$
  - Dans la pratique, commune choix pour *e* sont 3, 17 et 65537

(Fermat numbers premiers)

- **Clé de chiffrement publique KU = {e, N}**

- Le calcul d'une clé privée d de telle sorte que ed laisse un reste de 1 lorsqu'elle est divisée par ϕ.
  - Nous disons ed est congru à 1 modulo ϕ
- La résolution de l'équation suivante pour trouver la clé de déchiffrement d
  - ed = 1 (mod ϕ) => ed = 1 k.ϕ, et 0≤d≤ N, d = e-1 (mod ϕ) (modulaire inverse AE peut être calculée en utilisant l'algorithme d'Euclide étendu.
  - **Clé de chiffrement privée KR = {d, p, q}**
- Notez que d est facile de calculer que si l'on connaît la valeur de ϕ. Ceci est essentiellement la même que la connaissance des valeurs de p et q.

2.3.1.2 RSA - cryptage / décryptage

- Si M est un nombre qui est non divisible par N, puis en divisant Me.d par N et en prenant le reste donne la valeur d'origine M.
- Ceci est un théorème mathématique relativement profond, que nous pouvons écrire comme mod N = M Me.d

- Si M est un codage numérique d'un bloc de texte brut, le cryptogramme est $C = M^e \bmod N$.
- Ensuite mod N = Cd (M e mod N) d mod N = (Me) d mod N = M ed mod N = M., on peut ainsi récupérer le texte en clair M avec la clé privée d.

2.3.1.3 mise en œuvre de chiffrement / déchiffrement RSA

- Disons que nous voulons calculer :
  - calcule: C = Me mod N, où 0≤M <N • pour déchiffrer le texte chiffré C le propriétaire :
  - Utilise sa clé privée KR = {d, p, q}
  - calculer: $M = C^{ré} \bmod N$
- c = md mod n

Notez que nous ne devons pas calculer la valeur totale des m à la puissance d ici. Nous pouvons utiliser le fait que: a = mod bc n = (b mod n) (c mod n) n mod, donc nous pouvons décomposer un nombre potentiellement important dans ses composants et combiner.Les résultats des calculs plus simples, plus petits pour calculer la valeur finale.

- Par exemple : (m = 13, d = 7, n = 33)
- c = 137 mod 33 = 13 (3 + 3 + 1) mod 33 = 13 3.133.13 mod 33 = (mod 133 33). (13 3 mod 33). (13 mod 33) mod 33

  = (2.197 mod 33). (2197 mod 33). (13 mod 33) mod 33

  = Mod 19.19.13 33 = 4.693 mod 33

  = 7.

2.3.1.4 RSA résumé

- N = pq, où p, q sont des nombres premiers distincts
- ϕ, ϕ = (p-1). (q-1)
- e <N, pgcd (e, ϕ) = 1 et d = e-1 (mod ϕ)
- **Publique** clé de chiffrement : KU = {e, N}
- **Privé** clé de déchiffrement : KR = {d, p, q}

❖ Notez que le message M doit être inférieur au module N (bloc si nécessaire)

2.3.1.5 Pour quoi le RSA fonctionne ?

- La multiplication de p par q est simple : le nombre d'opérations dépend du nombre de bits (nombre de chiffres) en p et q.
- Par exemple, la multiplication de deux nombres de 384 bits prend environ 3842 = 147456 opérations de bit
- Si on ne connaît que N, trouver p et q (factorisation) est difficile : En substance, le nombre d'opérations dépend de la ***valeur*** de N.
  - La méthode la plus simple pour la factorisation d'un nombre 768 bits prennent environ $2^{384} \approx 3{,}94 \times 10115$ divisions d'essai.
  - Une méthode plus sophistiquée prend environ 2 85 ≈ 3.87 x 1025 divisions d'essai.
  - Une méthode encore plus sophistiquée prend environ 241 ≈219,000,000,000 divisions d'essai
- Personne n'a trouvé un algorithme très rapide pour la factorisation d'un grand nombre N.
- Personne n'a prouvé qu'un tel algorithme rapide n'existe pas (ou même que l'on est peu susceptible d'exister)

par example RSA

a) Sélectionnez les nombres premiers : p = 17 & q = 11

b) Calculer $n = Pq = 17 \times 11 = 187$

c) Calculer $\phi = (P-1)(q-1) = 16 \times 10 = 160$

d) Sélectionner e : pgcd (e, 160) = 1; choisir $e = 7$

e) Determiner d: de = 1 (mod 160)

d = 23 depuis le 23×7 = 161 = 10×160 + 1

f) Publier clé publique KU = {} 7187

g) Gardez la clé privée secrète KR = {} 23,17,11

par example RSA

- Échantillon de chiffrement / déchiffrement RSA est:
- message donné M = 88 (nb. 88 <187)
- chiffrement:

$C = 88^{7} \bmod 187 = 11$

- décryptage:

$M = 11^{23} \bmod 187 = 88$

❖ **sécurité RSA**

Trois approches pour attaquer RSA :

- Recherche clé de la force brute (taille infaisable donné de chiffres)
- Attaques mathématiques (basé sur la difficulté du calcul de ϕ, par module factoriser N)
- Les attaques de synchronisation (sur la course de décryptage)

2.4 Conclusion

De nombreuses méthodes de cryptage sont utilisées dans la communication, mais chacune d'elles est utilisée en fonction des besoins, de la rentabilité et de l'efficacité. Par exemple, l'algorithme AES est couramment utilisé dans le cryptage Wi-Fi pour sécuriser les informations entourant un réseau Wi-Fi particulier, car il est plus efficace dans la technologie Wi-Fi. Mais en général, le cryptage RSA s'est avéré plus efficace, plus robuste et presque impossible à craquer. Il est donc recommandé de l'utiliser dans le cryptage des données.

CHAPITRE 3

CRYPTOGRAPHIE DE BOUT EN BOUT (E2EE)

3.1. Introduction

La procédure connue lors de l'utilisation d'une application est qu'un client fournit d'abord un nom d'utilisateur et un mot de passe pour pouvoir accéder à l'application. Ensuite, la plupart des activités de l'application sont basées sur un contrôle d'accès. Par exemple, un utilisateur sera autorisé à télécharger des fichiers s'il a cet accès, etc.

Cela ressemble à la façon dont le travail de sécurité de construction, un contrôle de visite à la réception, un visiteur reçoit un badge, ce badge lui ouvrira des portes, et lorsque le badge expirera, il ne sera plus autorisé dans cet immeuble.

Le problème c'est qu'il y a des visiteurs qui n'entrent pas par la porte principale, nous appelons ces personnes HACKERS. Hackers essaiera d'accéder aux serveurs via le back-end et réussira d'une manière ou d'une autre à voler les données des clients. Ou parfois par des erreurs des gestionnaires de base de données les données fuient.

Par conséquent, il doit exister un meilleur moyen de protéger les données client, en introduisant des clés dans les applications client. Les données sont chiffrées et déchiffrées dans la machine utilisateur avant leur envoi et après leur réception, laissant la base de données Cloud et tous les intermédiaires totalement aveugles. Nous appelons cela un Chiffrement de bout en bout (E2EE).

Façon connue aujourd'hui de protéger la vie privée des utilisateurs et nous l'appelons de chiffrement de bout en bout (E2EE).

3.2 Généralité de chiffrement de bout en bout

Chiffrement de bout en bout (E2EE) est l'un des moyens les plus couramment utilisés pour envoyer des informations en toute sécurité sur Internet. En principe, E2EE est un moyen d'envoyer des informations sur un réseau de telle sorte que seul le destinataire et l'expéditeur y ont accès. E2EE contient les composants suivants:

- Identité et protocoles
- Algorithme
- La mise en œuvre et le fonctionnement sécurisé

3.2.1 Identité et protocoles

Identité et protocoles décrivent la façon dont le processus de chiffrement aura lieu et les algorithmes qui seront utilisés.

Dans la composante d'identité, un utilisateur ou d'un serveur qui envoie des données via E2EE se vérifie par un nom d'utilisateur et mot de passe, ou dans le cas d'un serveur, un certificat de sécurité délivré par un site pour prouver son identité en cours de validité. Ce

composant est très important à vérifier que l'expéditeur et le destinataire dans un échange E2EE sont les partis prévus. Par exemple, si un utilisateur envoyait des informations chiffrées sur un site web bancaire, le serveur hébergeant le site aurait un certificat de sécurité indiquant l'utilisateur que ce site est le site qu'il prétend être.

Dans le composant suivant, un protocole de transport est utilisé pour assurer une connexion est privée et préparer les données à chiffrer. L'Internet Engineering Task Force décrit un exemple d'un de ces protocoles, la sécurité Transport Layer (TLS) Protocole Handshake, le serveur et le client pour authentifier l'autre et de négocier un algorithme de chiffrement et des clés de chiffrement avant que le protocole d'application transmet ou reçoit son premier octet de données. Pour l'essentiel, un protocole décrit ce que l'algorithme va être utilisé et assure qu'aucune partie non désirée participe à l'échange de données. Cela fait partie intégrante du processus E2EE car il permet aux deux parties d'avoir un lien sécurisé et de recevoir des clés pour décoder les données cryptées. Le serveur et le client de s'authentifier et de négocier un algorithme de chiffrement et des clés de chiffrement avant que le protocole d'application transmet ou reçoit son premier octet de données. Pour l'essentiel, un protocole décrit ce que l'algorithme va être utilisé et assure qu'aucune partie indésirable participe à l'échange de données [11]

3.2.2 Algorithme

L'algorithme utilise des processus mathématiques pour brouiller les données de telle sorte qu'il est presque impossible de déchiffrer sans la clé prédéterminée.

3.2.3. La mise en œuvre et le fonctionnement sécurisé

La mise en œuvre et le fonctionnement sécurisé est un élément essentiel pour assurer que le processus E2EE est sûr du côté matériel. Par exemple, si les logiciels malveillants est installé sur l'ordinateur ou smartphone d'un client qui est capable d'avoir volé les données de la memoire de l smartphone, le processus de chiffrement entier est inutile, car le pirate a accès aux données après qu'il a déjà été décrypté. Il y a aussi des Malwares capables de volés les mots de passe et noms d'utilisateurs et utilisés pour usurper l'identité de l'utilisateur. En raison de cette vulnérabilité, les entreprises qui utilisent souvent E2EE exigera des utilisateurs dans un réseau d'avoir un service antivirus installé sur leur ordinateur pour prévenir contre les attaques de logiciels malveillants compromettre la sécurité que E2EE offres. Dans l'ensemble, E2EE est un moyen extrêmement sûr et sécurisé d'envoyer des informations sur Internet et est la meilleure solution pour améliorer la vie privée en ligne. Cela est dû à sa capacité d'adaptation, l'utilisation des clés privées générées, et des algorithmes complexes qui permettent la sécurité maximale possible.

3.3 Comment E2EE fonctionne

Lorsque deux utilisateurs partagent des informations telles que les messages texte, des photos, des vidéos ou des documents, ils chiffrent ces informations dans le dispositif de clients, et est ensuite les données chiffrées vont à la base de données, cloud, et quel que soit le transit qu'ils peuvent faire. Il n'y a aucun moyen que le gestionnaire de base de données ou les fournisseurs de services cloud peuvent voir les données en plus de ces deux utilisateurs qui partagent des informations les uns aux autres. Et cela est crypto graphiquement appliquée. E2EE de cette manière ajoute une autre couche de protection des données utilisateur, une erreur humaine qui peut être fait par les gestionnaires de bases de données ne sera pas suffisante pour provoquer un agent de blanchiment de données d'utilisateur à partir d'une extrémité à l'autre application cryptée.

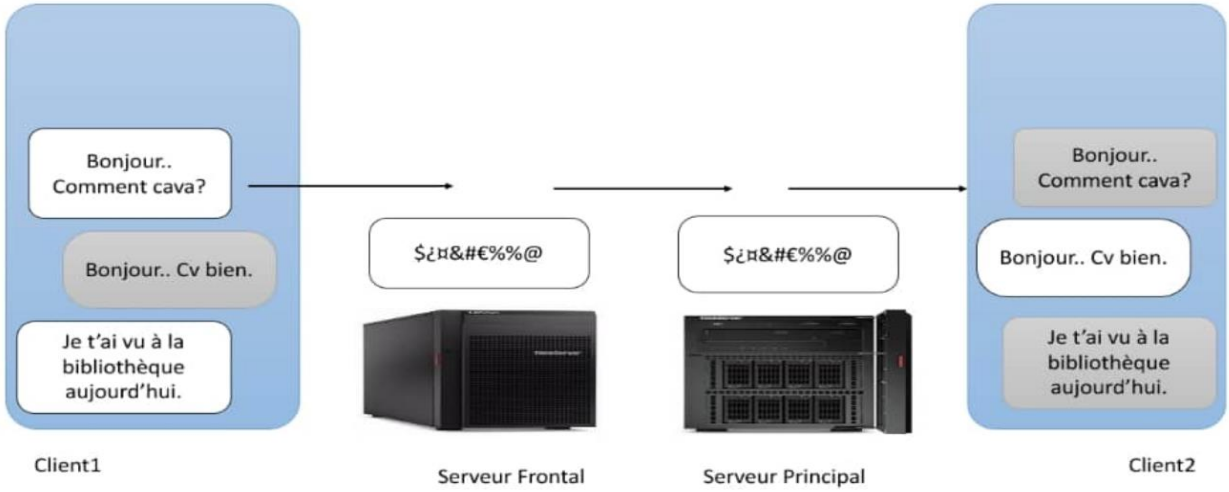


Figure 26 chiffrement de bout en bout

Par rapport à https où les données sont sécurisées sur un fil de l'application à l'extrémité avant de votre application, puis la sécurité met un terme, les données utilisateur reste clair dans la memoire des serveurs. Les serveurs sont protégés par des mots de passe dont la plupart d'entre eux sont maintenus à côté de ces bases de données, de sorte que tous ceux qui peuvent avoir accès aux mots de passe et accéder aux serveurs peuvent avoir accès aux données des utilisateurs en ce que les serveurs. E2EE d'autre part introduire des clés sur les périphériques clients, la clé est utilisée pour chiffrer les messages et autres données afin lorsque les données sont envoyées à la base de données restera crypté ne peut décrypter les données depuis les clés utilisées pour le décryptage sont disponibles dans le client dispositifs.

Le chiffrement et le déchiffrement des informations sur les périphériques des utilisateurs finaux des travaux similaires à celle de cryptage de cryptographie asymétrique dans l'algorithme RSA. Chaque utilisateur possède une clé publique et privée, il publie ensuite la clé publique dans le cloud et garder pour lui la clé privée.

Si un utilisateur A veut envoyer des informations à l'utilisateur B, elle utilise la clé publique de l'utilisateur B pour chiffrer ces informations, et l'utilisateur B utilise sa clé privée pour déchiffrer ces informations. La clé privée est toujours maintenue dans le dispositif des utilisateurs et nulle part ailleurs.

Avec E2EE comme exemple par de la simulation, nous avons utilisé, l'information est hautement sécurisée, et presque impossible que quelqu'un puisse intercepter les informations partagées entre les utilisateurs, et ceci est réalisé par les caractéristiques suivantes peu appliquées dans la simulation.

a. L'algorithme utilisé pour crypter les données est non déterministe, est crypté différemment un mot similaire de texte. Par exemple, le mot «oui» envoyé deux fois, montrera deux textes différemment chiffrés.
b. Dès que les messages sont livrés à l'utilisateur final, ils disparaissent de la base de données, et donc pas attaquant futur peuvent avoir accès à ces données de la base de données.

3.4 Communication sur les plates-formes de messagerie

La plupart des plates-formes de messagerie utilisé maintenant jours de E2EE comme un moyen de sécurisation des données de l'utilisateur sur Internet. L'utilisation de E2EE a commencé officiellement après la découverte de PGP par Phil Zimmerman, à partir de là sur de nombreuses versions de PGP ont été développées à partir de la source PGP ouverte.

3.4.1 PGP (Pretty Good Privacy)

PGP signifie Pretty Good Privacy. Il est un logiciel de chiffrement utilisé largement conçu pour assurer la confidentialité, la sécurité et l'authentification pour les systèmes de communication en ligne. PGP a été publié en 1991 par Phil Zimmerman, selon lui, il a été mis gratuitement à disposition en raison de la demande sociale croissante de la vie privée. Depuis sa création en 1991, beaucoup de versions de logiciels PGP ont etait créé. En 1997, Phil Zimmerman a fait une proposition à l'Internet Engineering Task Force (IETF) pour la création d'une norme PGP open-source. La proposition a été acceptée et a conduit à la création du protocole OpenPGP, qui définit les formats standard pour les clés de chiffrement et des messages. Bien qu'initialement utilisé pour sécuriser les courriels, les messages et les pièces jointes, PGP est maintenant appliqué dans une large gamme de cas d'utilisation, y compris la signature numérique, le chiffrement intégral du disque et de protection du réseau. PGP a d'abord été détenue par la société PGP Inc, qui a ensuite été acquis par de Network Associates Inc.

Le flux de travail de PGP implémente la cryptographie à clé publique. Il est un hybride qui crypto système utilise le chiffrement symétrique et asymétrique pour atteindre un haut

niveau de sécurité. Dans un processus de base de cryptage de texte, la plupart des systèmes de PGP effectue une compression de données avant de convertir les données en cryptogramme, cette façon PGP permet d'économiser l'espace disque et le temps de transmission tout en améliorant la sécurité. Après la compression, le fichier est crypté avec plaintext une clé à usage unique, connue sous le nom de la clé de session. Cette clé est générée de façon aléatoire par l'utilisation de la cryptographie symétrique, et chaque session de communication de PGP a une clé de session unique. La clé de session est ensuite chiffrée par la clé publique du récepteur; cette étape permet Alice de partager en toute sécurité la clé de session avec Bob à travers l'Internet, indépendamment des conditions de sécurité. Le cryptage asymétrique de la clé de session se fait généralement par l'utilisation de l'algorithme RSA. Lorsque le cryptogramme du message et la clé de session cryptée sont transmises, Bob peut utiliser sa clé privée pour déchiffrer la clé de session, qui est ensuite utilisée pour déchiffrer le texte chiffré de nouveau dans le texte brut d'origine. Mis à part le processus de base de chiffrement et de déchiffrement, PGP prennent également en charge les signatures numériques qui sert au moins trois fonctions :

a. Authentification : Bob peut vérifier que l'expéditeur du message était Alice.
b. Intégrité : Bob peut être sûr que le message n'a pas été modifié.
c. La non-répudiation : après que le message est signé numériquement, Alice ne peut prétendre qu'elle n'a pas envoyé

3.4.2 iMessage et FaceTime d'Apple Inc.

Lorsque Apple a lancé iMessage en 2011, le service contient E2EE intégré lors du lancement de la plate-forme. Apple utilise E2EE pour protéger la conversation iMessage et FaceTime sur tous leurs appareils. Avec watchOS et iOS, vos messages sont cryptés sur votre appareil afin qu'ils ne peuvent pas être accessibles sans votre mot de passe. Ils ont conçu iMessage et FaceTime pour que Apple ne peut pas déchiffrer les messages du client quand il est en transit entre les appareils. Vous pouvez choisir de supprimer automatiquement vos messages de votre appareil au bout de 30 jours ou un an ou pour les garder sur votre appareil pour toujours.

Des applications tierces qui utilisent iMessage n'ont pas accès à l'information de contact ou les conversations des participants. iOS fournit à chaque application d'un identificateur aléatoire pour chaque participant, qui est remis à zéro lorsque l'application est désinstallé. iMessage et les messages SMS sont sauvegardés sur iCloud pour le confort de l'utilisateur, mais l'utilisateur peut transformer iCloud sauvegarde hors quand il veut. Et Apple n'a jamais stocker le contenu de FaceTime appelle tous les serveurs.

3.4.3 WhatsApp.

Une autre des principales plates-formes pour offrir chiffrement de bout en bout est

WhatsApp, un service de messagerie populaire appartenant à Facebook, qui a mis en œuvre le système en 2014. WhatsApp estime que la sécurité est dans leur cœur et qui est la raison pour laquelle ils ont mis en œuvre E2EE dans leur plate-forme. Lorsque, chiffrement de bout en bout, messages, photos, vidéos, documents, appels et mises à jour de statut sont garantis de tomber dans de mauvaises mains.

Chiffrement de bout en bout WhatsApp assure que le client et la personne qu'il communique avec peut lire ce qui est envoyé, et personne entre les deux, même pas WhatsApp. Les messages du client sont sécurisés avec des serrures, et seul le destinataire et le client ont des touches spéciales nécessaires pour déverrouiller et lire ces messages. Pour une protection supplémentaire, chaque message envoyé a un verrou unique et clé. Tout cela se fait automatiquement : Pas besoin de tourner sur les réglages ou mis en place des conversations secrètes spéciales pour des messages sécurisés. Le chiffrement de bout en bout est toujours activé. Il n'y a pas moyen de désactiver le chiffrement de bout en bout.

3.4.3.1 Vérification de chiffrement de bout en bout WhatsApp

1. Ouvrez le chat.
2. Appuyez sur le nom du contact pour ouvrir l'écran d'information de contact.
3. Appuyez sur Chiffrement pour afficher le code QR et le numéro 60 chiffres.

Si vous et votre contact sont physiquement à côté de l'autre, l'un d'entre vous pouvez numériser l'autre de QR code ou comparer visuellement le nombre de 60 chiffres. Si vous scannez le code QR, et le code est bien le même, une coche verte apparaît. Comme ils correspondent, vous pouvez être sûr que personne n'intercepte vos messages ou appels.
Si les codes ne vous, il est peu probable correspondent scannant le code d'un autre contact ou un numéro de téléphone différent. Si votre contact a récemment réinstallé WhatsApp ou les téléphones modifiés, il vous est recommandé d'actualiser le code en leur envoyant un nouveau message, puis scannant le code.

Si vous et votre contact ne sont pas physiquement près de l'autre, vous pouvez les envoyer le numéro 60 chiffres. Laissez votre interlocuteur savoir qu'une fois qu'ils reçoivent votre code, ils doivent écrire et comparer visuellement au numéro 60 chiffres qui apparaît dans l'écran d'information de contact sous chiffrement. Pour Android, iPhone et Windows Phone, vous pouvez utiliser le bouton Partager de l'écran Vérification du code de sécurité pour envoyer le numéro 60 chiffres par SMS, e-mail, etc.

3.4.3.2 Pourquoi E2EE et comment est-il important ?

La sécurité est essentielle au service WhatsApp offre. « Nous avons terminé la mise en œuvre de chiffrement de bout en bout en 2016 pour tous les messages et appelant WhatsApp pour que personne ne nous même pas, a accès au contenu de vos conversations. Depuis lors, la

sécurité numérique est devenue encore plus important. Nous avons vu de multiples exemples où les pirates criminels ont obtenu illégalement des sommes énormes de données privées et de la technologie abusé de blesser les gens avec leurs informations volées. Alors que nous avons introduit plus de fonctionnalités - comme les appels vidéo et le statut -. Nous avons étendu le chiffrement de bout en bout à ces caractéristiques aussi bien » organisation WhatsApp dit.

WhatsApp n'a pas la capacité de voir le contenu des messages ou écouter des appels sur WhatsApp. En effet, le chiffrement et le déchiffrement des messages envoyés sur WhatsApp se produit entièrement sur l'appareil de l'utilisateur. Avant qu'un message quitte jamais dispositif utilisateur, il est sécurisé avec une serrure cryptographique, et seul le destinataire a les clés. De plus, les touches changent avec chaque message qui est envoyé. Bien que tout cela se passe dans les coulisses, un utilisateur peut confirmer ses conversations sont protégées en vérifiant le code de vérification de la sécurité sur son appareil.

D'autres plates-formes populaires cette fin hôte pour mettre fin chiffrement comprennent Facebook Messenger (bien qu'il ne soit pas activé par défaut), Threema, Google Allo, Signal, télégramme et OTR.

3.5 échange de clés en E2EE.

Avant le processus réel de chiffrement et le déchiffrement des données utilisateur dans leurs appareils, il doit y avoir un moyen pour les clés de partage. En cas de chiffrement symétrique une clé est bien partagée entre les utilisateurs finaux, alors que dans le chiffrement asymétrique avec les clés publiques-privées, la clé publique est partagée en clair sur Internet et la clé privée est stockée dans la zone sécurisée dans le dispositif de l'utilisateur final.

3.5.1 échange de clés Symétrique (Diffie-Hellman Key Exchange)

Échange de clés Diffie-Hellman(DH) est une méthode d'échange sécurisée des clés de chiffrement sur une chaîne publique et a été l'un des premiers protocoles à clé publique initialement conceptualisé par Ralph Merkle et porte le nom Whitfield Diffie et Martin Hellman. DH est l'un des premiers exemples pratiques d'échange de clés publique mis en œuvre dans le domaine de la cryptographie.

Traditionnellement, la communication cryptée sécurisée entre deux parties, il leur fallait premières clés d'échange par un certain canal physique sécurisé, comme les listes de clés de papier transportés par un courrier de confiance. La méthode d'échange de clés Diffie-Hellman permet à deux parties qui ont aucune connaissance préalable de l'autre pour établir conjointement une clé secrète partagée sur un canal non sécurisé. Cette clé peut

être utilisée pour chiffrer les communications ultérieures utilisant un algorithme de chiffrement à clé symétrique.

Diffie-Hellman est utilisé pour sécuriser une variété de services Internet. Cependant, la recherche publiée en Octobre ici à 2015 suggère que les paramètres utilisés pour de nombreuses applications DH Internet à ce moment-là ne sont pas assez forts pour éviter compromis
des attaquants très bien financés, tels que les services de sécurité des grands gouvernements.

Le système a d'abord été publié par Whitfield Diffie et Martin Hellman en 1976, mais en 1997 il a été révélé que James H. Ellis, Cocks Clifford, et Malcolm J. Williamson de GCHQ, les signaux britannique agence de renseignement, avait déjà, en 1969, représentée comment la cryptographie à clé publique pourrait être atteint.

Bien que l'accord de clé Diffie-Hellman lui-même est un protocole d'accord non-authentifié Key-, il fournit la base pour une variété de protocoles authentifiées,
et est utilisé pour fournir le secret avant dans les modes éphémères de Transport Layer Security (appelé ou DHE EDH en fonction de la suite de chiffrement).
La méthode a été suivie peu après par RSA, une implémentation de la cryptographie à clé publique en utilisant des algorithmes asymétriques.

3.5.1.1 Illustration de l'idée derrière l'échange de clés Diffie-Hellman.

Échange de clés Diffie-Hellman établit un secret partagé entre deux parties qui peuvent être utilisés pour les communications secrètes pour échanger des données sur un réseau public. Le schéma conceptuel qui suit illustre l'idée générale de l'échange de clés en utilisant des couleurs au lieu de très grands nombres.

Le processus commence par avoir les deux parties, Alice et Bob, sont d'accord sur une couleur de départ arbitraire qui n'a pas besoin d'être gardé secret (mais devrait être différent à chaque fois); dans cet exemple, la couleur est jaune. Chacun d'eux choisit une couleur secrète qu'ils gardent à eux-mêmes - dans ce cas, le rouge et le bleu-vert. La partie essentielle du processus est que Alice et Bob chaque mélange leur propre couleur secrète ainsi que leur couleur mutuellement partagée, ce qui en orange-tan et les mélanges bleu clair respectivement, puis d'échanger publiquement les deux couleurs mélangées. Enfin, chacun des deux mélanges la couleur qu'ils ont reçu du partenaire avec leur propre couleur privée. Le résultat est un mélange de couleur finale (jaune-brun dans ce cas) qui est identique au mélange de couleur finale du partenaire.

Si un tiers a écouté l'échange, il ne connaît la couleur commune (jaune) et les premières couleurs mélangées (orange- tan et bleu clair), mais il serait informatiquement difficile pour cette partie de déterminer la couleur finale secrète (jaune marron). En fait, lorsque vous utilisez un grand nombre plutôt que de couleurs, cette action est informatiquement cher: Il est impossible de le faire dans un laps de temps raisonnable même pour les super-ordinateurs modernes.[18]

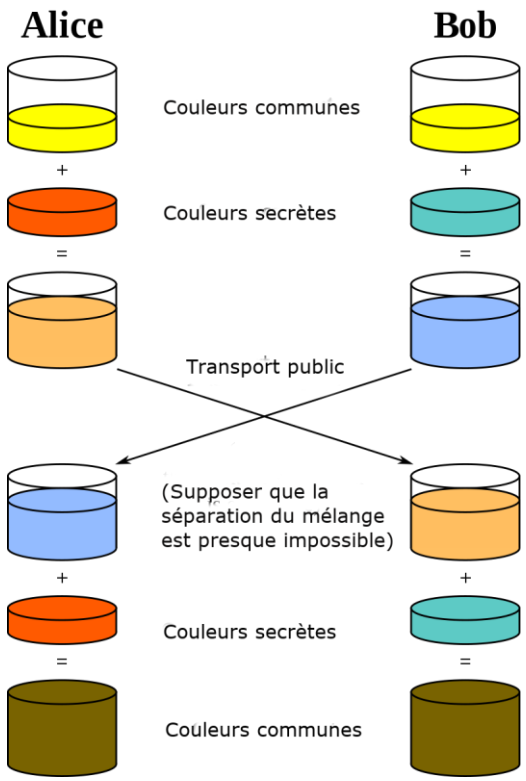


Figure 27 Illustration de l'idée derrière l'échange de clés Diffie-Hellman

3.5.1.2 explication Cryptographic

Le plus simple et la mise en œuvre initiale du protocole utilise le groupe multiplicatif des entiers modulo p, où p est un nombre premier et g est une racine primitive modulo p. Ces deux valeurs sont choisies de cette manière à faire en sorte que le secret partagé résultant peut prendre toute valeur de 1 à p-1. Voici un exemple du protocole,

1. Alice et Bob d'accord publiquement d'utiliser un module p = 23 et une base g = 5 (qui est une racine primitive modulo 23).

2. Alice choisit un nombre entier secret, a = 4, puis envoie Bob A =g^amod p, *A*= $\mathbf{5^4}$Mod23 = 4
3. Bob choisit un nombre entier secret, b = 3, puis envoie Alice B = g^bmod p, *B*= $\mathbf{5^3}$Mod23 = 10
4. Alice calcule s = B^amod p, *s*= $\mathbf{10^4}$mod23 = 18
5. Bob calcule s = A^b mod23, s = $\mathbf{4^3}$mod23 = 18
6. Alice et Bob partagent maintenant un secret (le numéro 18).

Notez que seulement a, b, et (g^{ab}mod p = g^{ba}mod p) sont gardées secrètes. Toutes les autres valeurs - p, g, g^amodp et g^b mod p - sont envoyés en clair. Une fois que Alice et Bob ont calculé le secret partagé, ils peuvent l'utiliser comme clé de cryptage, connue uniquement d'eux, pour l'envoi de messages via le même canal de communication ouvert.

Bien sûr, des valeurs beaucoup plus de a, b et p seraient nécessaires pour que cet exemple devient sûr, car il n'y a que 23 résultats possibles de *n* mod23. Cependant, si p est premier d'au moins 600 chiffres, alors même les ordinateurs les plus rapides modernes ne peuvent pas trouver *a* a partir de g, p et g^amod p. Un tel problème est appelé le problème du logarithme discret. Le calcul de g^a*modp* est connu comme exponentiation modulaire et peut être fait efficacement même pour un grand nombre. Notez que g n'a pas besoin d'être grand , et dans la pratique est généralement un petit nombre entier (comme 2, 3, ...).

3.5.1.3 Opération avec plus de deux parties

Accord clé Diffie-Hellman ne se limite pas à la négociation d'une clé partagée par seulement deux participants. Un nombre quelconque d'utilisateurs peuvent prendre part à un accord en effectuant des itérations du protocole d'accord et l'échange de données intermédiaires (qui ne se pas besoin d'être gardé secret). Par exemple, Alice, Bob et Carol pourrait participer à un accord Diffie-Hellman comme suit, avec toutes les opérations prises pour être modulo p:

1. Les parties conviennent de paramètres de l'algorithme p et g.
2. Les parties génèrent leurs clés privées, du nom a, b et c.
3. Alice calcule et envoie à Bob. g^a
4. Bob calcule et envoie à Carol. g^{ab}
5. Carol calcule et utilise comme son secret. g^{abc}
6. Bob calcule et envoie à Carol. g^b
7. Carol calcule et envoie à Alice. g^{bc}
8. Alice calcule et utilise comme son secret. g^{bca}
9. Carol calcule et envoie à Alice. g^c
10. Alice calcule et envoie à Bob. g^{ca}
11. Bob calcule g^{cab} et l'utilise comme son secret. $g^{cab} = g^{bca} = g^{abc}$

Un indiscret (Eavesdropper) a pu voir$g^a, g^b, g^c, g^{ab}, g^{ca} et\ g^{bc}$, mais il ne peut utiliser toute combinaison de ceux-ci pour reproduire efficacement g^{abc}.

3.5.1.4 Pour étendre ce mécanisme à des groupes plus importants, deux principes de base doivent être respectées:

a. A partir d'un « vide » clé composé uniquement de g, le secret est fait en augmentant la valeur actuelle de l'exposant privé de chaque participant une fois, dans un ordre quelconque (la première exponentiation donne cette propre clé publique du participant).

b. Toute valeur intermédiaire (ayant jusqu'à N-1 exposants appliquée, où N est le nombre de participants dans le groupe) peut être révélé publiquement, mais la valeur finale (ayant eu tous les exposants N appliqué) constitue le donc secret partagé et ne doit jamais être révélé publiquement. Ainsi, chaque utilisateur doit obtenir leur copie du secret en appliquant leur propre clé privée dernier (sinon il n'y aurait aucun moyen pour le dernier contributeur de communiquer la clé finale à son destinataire, car ce dernier contributeur aurait tourné la clé dans la très secret, le groupe a souhaité protéger).

   Ces principes laissent des options diverses ouvertes pour le choix dans lequel les participants de commande contribuent aux clés. La plus simple et la solution la plus évidente est d'organiser les participants N dans un cercle et ont des touches N tournent autour du cercle, éventuellement chaque clé a été contribué par tous les participants N (se terminant par son propriétaire) et chaque participant a contribué à clés N (se terminant par leur propre). Cependant, cela exige que chaque participant effectue N exponentiation modulaire

En choisissant un ordre plus optimale, et se fondant sur le fait que les clés peuvent être dupliquées, il est possible de réduire le nombre de exponentiations modulaires effectuées par chaque participant log_2 (N) + 1 en utilisant une approche de style division et conquer-, étant donné ici pour huit participants:

1. Les participants A, B, C, et D effectuent chacun une exponentiation, en donnant lieu g^{abcd}; cette valeur est envoyée à E, F, G et H. En retour, les participants A, B, C, et D reçoivent g^{efgh}.
2. Les participants A et B effectuent chacun une exponentiation, ce qui donne g^{efghab}, qu'ils envoient à C et D, tandis que C et D font la même chose, ce qui donne g^{efghcd}, qu'ils envoient à A et B.

3. Participant A exécute une exponentiation, ce qui donne $g^{efghcda}$, qui l'envoie à B; De même, B envoie $g^{efghcdb}$ à A. C et D font la même façon.
4. Participant A exécute une exponentiation finale, ce qui donne le secret $g^{efghcdba}$ = $g^{abcdefgh}$, Tandis que B fait la même chose pour obtenir $g^{efghcdab}$= $g^{abcdefgh}$; encore, C et D font la même façon.
5. Les participants E à H exécutent simultanément les mêmes opérations en utilisant g^{abcd} comme point de départ.

Une fois cette opération terminée tous les participants posséderont le secret $g^{abcdefgh}$, mais chaque participant aura effectué seulement quatre exponentiations modulaires, plutôt que les huit sous-entendus par un arrangement circulaire simple. [19]

3.5.1.5 Attaques pratiques sur le trafic Internet

Le n*umber field sieve algorithm*, qui est généralement la plus efficace pour résoudre le problème du logarithme discret, composé de quatre étapes de calcul. Les trois premières étapes ne dépendent que de l'ordre du groupe G, et non sur le nombre spécifique dont le journal fini est souhaitée. Il se trouve que beaucoup de trafic Internet utilise un d'une poignée de groupes qui sont de l'ordre de 1024 bits ou moins. En précalcul les trois premières étapes du tamis de champ numérique pour les groupes les plus communs, un attaquant doit seulement pour mener à bien la dernière étape, ce qui est beaucoup moins cher que les informatiquement trois premières étapes, pour obtenir un logarithme spécifique.

L'attaque Logjam (The Logjam Attack) utilisé cette vulnérabilité pour compromettre une variété de services Internet qui ont permis l'utilisation de groupes dont l'ordre était un nombre premier de 512 bits, ce qu'on appelle les grades d'exportation. Les auteurs avaient besoin de plusieurs milliers de cœurs de processeurs pour une semaine pour précalculer données pour un premier 512 bits. Une fois cela fait, logarithmes individuels pourraient être résolus en une minute en utilisant deux 18-core Intel Xeon processeurs.
Comme estimé par les auteurs de l'attentat Logjam, le précalcul beaucoup plus difficile nécessaire pour résoudre le problème du logarithme discret pour un premier 1024 bits coûterait de l'ordre de 100 millions $, et dans le budget de grande agence de renseignement national, comme les Etats-Unis Agence nationale de sécurité (NSA). Les auteurs spéculent que logjam précalcul contre les nombres premiers DH 1024 bits est largement réutilisé derrière demandes dans les documents de la NSA que la NSA est une fuite capable de briser une grande partie de la cryptographie actuelle.

Pour éviter ces vulnérabilités, les auteurs recommandent l'utilisation de la cryptographie à courbe elliptique, dont on connaît aucune attaque similaire. A défaut, ils recommandent que l'ordre, p, du groupe Diffie-Hellman doit être d'au moins 2048 bits. Ils estiment que le pré-calcul nécessaire pour un premier 2048 bits est 109 plus difficile que pour les nombres premiers 1024 bits. [20]

3.5.2 Échange de clés asymétrique E2EE (RSA algorithme)

Dans une application de chat où les messages sont cryptés de bout en bout, il devrait y avoir un moyen de gérer l'échange de clés utilisées dans le chiffrement et le déchiffrement des informations entre les utilisateurs. Lors de l'enregistrement de l'utilisateur dans une application de chat une paire de clés est générée, une clé publique et privée. La clé publique est toujours utilisée pour le chiffrement de l'information avant d'être envoyé à la destination, et la clé privée est utilisée pour décrypter l'information dans les dispositifs. Il Fore avant d'envoyer un message texte par exemple, on a fait Chiffrer ce message. Dans un algorithme RSA, quand un expéditeur laisser dire Alice veut envoyer un message texte à Bob, le message doit être crypté avec la clé publique de Bob et Bob utilisera sa clé privée pour déchiffrer le message.

En chiffrement de bout en bout avec des paires de clés de chiffrement asymétrique sont générés dans l'utilisateur de concevoir, contrairement au chiffrement symétrique où la même clé est partagée entre les utilisateurs pour le chiffrement et le déchiffrement, dans le chiffrement asymétrique que la clé publique est partagée entre les utilisateurs. La clé privée est toujours maintenu dans un endroit sûr en toute sécurité du dispositif utilisateur, par conséquent est beaucoup plus facile d'assurer la sécurité des clés de chiffrement asymétrique que dans le chiffrement symétrique.

Comme il a été connu dans le chiffrement asymétrique avec la référence de l'algorithme RSA toutes les informations sont chiffrées à l'aide des clés publiques et est alors la clé privée utilisée pour déchiffrer ces informations, donc si un utilisateur veut vous envoyer un message texte, il utilisera votre clé publique pour crypter cette information pour vous, et quand les informations entrer dans votre appareil l'application va les décrypter en utilisant la clé privée.

Chiffrement de bout en bout, un message texte est crypté dans le périphérique de l'utilisateur et envoyé à Internet le message reste crypté et ne sera pas vu par tout le monde à travers tous les transits qu'il fait jusqu'à au récepteur de ce message texte. Même le

développeur de l'application aura une connaissance zéro de l'information qui est partagée entre les utilisateurs finissent les clés privées stockées dans des dispositifs d'utilisateur.

3.5.2.1 Échange de clés asymétrique dans un Group Chat

Dans une conversation impliquant plusieurs utilisateurs, une autre étape est ajouté pour aider à l'échange des clés entre les membres du chat. Lorsqu'un troisième utilisateur se joindre à la conversation, une clé de fil est généré, cette clé de fil est crypté par l'application à l'aide des clés publiques de chaque membre du chat, il y a donc la clé publique de chaque membre sera utilisé pour chiffrer la clé de fil, puis ces clés de fil chiffrées sont publiées sur le cloud du fournisseur de services de l'application. Chaque membre à temps envoyer un message au groupe de discussion, il a d'abord ce message en utilisant encrypte la clé de fil de la discussion. D'autres membres du groupe décryptent la clé de fil qui a été chiffrée à l'aide de leurs clés publiques, ce décryptage se fait avec les clés privées des membres. Ils utilisent alors la clé de fil pour déchiffrer le message envoyé au groupe de discussion.

3.6 Autres caractéristiques de E2EE

En collaboration avec le chiffrement des données dans une application E2EE, certaines fonctionnalités nécessaires peuvent être ajoutées pour assurer la sécurité parfaite et complète des données entre les utilisateurs. Ces caractéristiques peuvent être différentes d'une application à une autre en fonction du développeur de cette application ou d'un logiciel. Certaines de ces caractéristiques peuvent être les suivantes:

3.6.1 Intégrité des données

Lorsque des informations sensibles sont échangées, le récepteur doit avoir l'assurance que le message est venu intact de l'émetteur prévu et n'est pas modifiée par inadvertance ou autrement. Il existe deux types de menaces d'intégrité des données, à savoir passives et actives.

3.6.1.1 Menaces passives

Ce type de menaces existe en raison de changements dans les données accidentelles.

- Ces erreurs de données sont susceptibles de se produire en raison du bruit dans un canal de communication. En outre, les données peuvent être corrompues pendant que le fichier est stocké sur un disque.
- codes correcteurs d'erreurs et checksums simples comme contrôles de redondance cyclique (CRC) sont utilisés pour détecter la perte d'intégrité des données. Dans ces techniques, un résumé des données est calculée mathématiquement et ajoutée aux données

3.6.1.2 Menaces actives

Dans ce type de menaces, un attaquant peut manipuler les données avec une intention malveillante.

- Au niveau le plus simple, si les données sont sans digest, il peut être modifié sans détection. Le système peut utiliser des techniques de l'ajout de données de CRC pour détecter une modification active.
- Au niveau de la menace plus, l'attaquant peut modifier les données et essayer de dériver de nouvelles digérer pour les données modifiées de sortie digest. Ceci est possible si le résumé est calculé à l'aide des mécanismes simples tels que CRC.
- Mécanisme de sécurité telles que les fonctions de hachage sont utilisées pour faire face aux menaces de modification actifs.

Les fonctions de hachage sont extrêmement utiles et apparaissent dans presque toutes les applications de sécurité de l'information.

Une fonction de hachage est une fonction mathématique qui convertit une valeur d'entrée numérique en une autre valeur numérique comprimé. L'entrée de la fonction de hachage est d'une longueur arbitraire, mais est toujours sortie de longueur fixe.

Les valeurs renvoyées par une fonction de hachage sont appelés condensé de message ou simplement des valeurs de hachage.

3.6.2 Cryptographie Signature numérique (digital signature)

Les signatures numériques sont les primitives de clé publique d'authentification des messages. Dans le monde physique, il est courant d'utiliser des signatures manuscrites sur les messages écrits à la main ou dactylographiées. Ils sont utilisés pour lier signataire du message.

De même, une signature numérique est une technique qui se lie à une personne / entité aux données numériques. Cette liaison peut être vérifiée de façon indépendante par le récepteur ainsi que d'un tiers.

La signature numérique est une valeur cryptographique qui est calculée à partir des données et une clé secrète connue uniquement par le signataire.

Dans le monde réel, le récepteur du message doit garantir que le message appartient à l'expéditeur et il ne devrait pas être en mesure de répudier l'origine de ce message. Cette exigence est cruciale dans les applications d'affaires, étant donné que la probabilité d'un conflit sur les données échangées est très élevé.

3.6.3 Forward Secrecy

Afin de maintenir les communications passées en toute sécurité, les clients entameront ressaisie une fois une clé a été utilisée pour décrypter et crypter un nombre donné de message, ou a été utilisé pendant une période donnée, à condition que la clé a été utilisée

pour chiffrer au moins un message. Anciennes clés sont ensuite solidement mis au rebut et ne peuvent pas être reconstruites, même avec l'accès aux nouvelles clés actuellement en cours d'utilisation.

Tout client participant à un chat secret peut lancer la saisie re-dès qu'il perçoit que la clé actuelle a été utilisé trop longtemps ou pour chiffrer trop de messages. Un client ne devrait jamais lancer une nouvelle instance du protocole ressaisie si une instance existe inachevés, initié par l'une des parties.

3.6.4 Supprimer le contenu dans un chat secret,

Au cours des 10-20 dernières années, chacun d'entre nous ont échangé des millions de messages avec des milliers de personnes. La plupart de ces journaux de communication sont stockés quelque part dans les boîtes de réception des autres et des serveurs hors de notre portée. Les relations de début et de fin, mais les histoires de messagerie avec les ex-amis et ex-collègues restent disponibles pour toujours,

Un vieux message que vous avez déjà oublié au sujet peut être pris hors contexte et utilisé contre vous décennies plus tard. Un texte hâtif que vous avez envoyé à une petite amie à l'école peut venir vous hanter en 2030 lorsque vous décidez d'exécuter à la mairie. Pour cette raison, la supprimer tout message dans les deux extrémités et les serveurs dans un chat privé, fonction à tout moment est nécessaire d'ajouter dans toute application de chat. Une caractéristique comme celui-ci a été ajouté dans les applications de chat comme Télégramme.

Cette fonction de suppression supprime le contrôle de tout le monde dans un chat privé, permettant à quiconque dans un thread privé à choisir de modifier ou même supprimer complètement l'histoire de chat s'ils le souhaitent à tout moment.

3.7 Chiffrement sur d'autres plateformes

En plus des différentes applications de messagerie, il existe également des systèmes différents qui utilisent le chiffrement de bout en bout sur différentes plates-formes. Fin de cryptage final est souvent trouvé sur le courrier électronique, téléphone, et même la radio. Les différentes plates-formes qui sont couramment utilisés sur le courrier électronique comprennent, GnuPG, protonmail, Mailfence, etc ... En fin de téléphone pour mettre fin à un cryptage, un logiciel comme FaceTime utilisation Voice over Internet Protocol, ou VoIP, basée sur le protocole de transport en temps réel. Le logiciel qui utilise le chiffrement à la cryptographie de bout en bout pour la radio est appelée TETRA. Il y a beaucoup d'autres applications ou services qui utilisent le chiffreent de bout en bout pour la communication et la messagerie. Le cryptage sur toutes ces plates-formes est importante car elle permet aux méthodes de communication, de sécurité privées entre deux utilisateurs ou plus.

3.8 Le chiffrement des dossiers médicaux et des données des patients

Les données personnelles sensibles ne se trouve pas seulement dans les plates-formes de communication ; Il se trouve également dans les hôpitaux où les données du patient sont utilisées pour garder une trace de l'histoire, les problèmes et la documentation. Il y a souvent des voleurs, qui profitent de l'information qui reste à l'air libre, qui tentent de voler des informations, ou pour avoir accès à des médicaments ou tout autre chose que les gens n'ont généralement pas accès. Pour éviter cela, les documents et les informations sont chiffrées avec un code initialement, de sorte qu'il ne peut être intercepté par les mauvais yeux. Le récepteur a également prévu une clé qui est capable de déchiffrer les informations et est en mesure d'utiliser les informations qui ont été chiffrées et est en mesure d'aider le patient. Ce processus est basé sur le chiffrement de bout en bout, et est également directement en mesure d'aider les gens,

3.9 Les défis de cryptage de bout en bout

Avec toutes les possibilités de faire E2EE le système le plus indestructible de sécurisation des données de l'utilisateur, il y a encore des défis qui peuvent rendre E2EE moins efficace. La plupart de ces défis entre en jeu en raison de moins confiance des fournisseurs de services e2ee.

3.9.1 backdoors

Les sociétés mettent en œuvre une backdoors à des plates-formes de chiffrement, et parfois ne dire à leurs clients d'avoir une backdoors. Une backdoors est un système en place par le fournisseur de services où ils peuvent aller et obtenir les messages qui ont été envoyés, les décrypter, et de les remettre à un tiers, comme les organisations gouvernementales. Ceci est généralement mal vu par les consommateurs, parce qu'ils ne reçoivent pas un service qui est vraiment chiffrement de bout en bout.

Un exemple très courant d'un bout à l'service de chiffrement avec une backdoors comme une partie du système est Skype de Microsoft. Le service a été que l'on croyait être complètement E2EE sans un moyen pour Microsoft de voir ce qui est envoyé. En 2013, Edward Snowden a partagé que la plate-forme avait en fait la backdoors, dans ce qui a permis l'accès Microsoft à tous les messages et communications sur le service. Microsoft et Skype nié cette déclaration plusieurs fois, mais à la fin, il a été montré que la messagerie Skype et la voix avaient backdoors. Microsoft a déjà déchiffré leur propre technologie, ce qui signifie qu'il est plus facile de le décrypter.

3.9.2 L'homme dans l'attaque du milieu (Man In The Middle Attack,MITMA)

Dans la cryptographie et de la sécurité informatique, Man In The Middle Attack (MITMA) est une attaque où l'attaquant relaie secrètement et modifie éventuellement les communications entre deux parties qui croient qu'ils communiquent directement entre eux. L'attaquant doit être en mesure d'intercepter tous les messages pertinents qui passent entre les deux victimes et injecter de nouveaux. Ceci est simple dans de nombreuses circonstances par exemple, un attaquant à portée de réception d'un point d'accès sans fil non crypté (Wi-Fi) pourrait s'insérer comme un MITMA.

Comme une attaque qui vise à contourner l'authentification ou son absence mutuelle, une attaque MITM ne peut réussir que lorsque l'attaquant peut usurper l'identité de chaque point final à leur satisfaction comme prévu des fins légitimes. La plupart des protocoles de chiffrement comprennent une certaine forme d'authentification des terminaux spécifiquement pour prévenir les attaques MITM. Par exemple, TLS peut authentifier une ou les deux parties à l'aide d'une autorité de certification mutuellement confiance.

- *Exemple*

Supposons Alice souhaite communiquer avec Bob. Pendant ce temps, Mallory souhaite intercepter la conversation pour écouter et éventuellement de fournir un faux message à Bob.

Tout d'abord, Alice demande Bob pour sa clé publique. Si Bob envoie sa clé publique à Alice, mais Mallory est capable d'intercepter, une attaque MITM peut commencer. Mallory envoie un message à Alice forgé qui prétend venir de Bob, mais inclut la place clé publique de Mallory.

Alice, croyant que cette clé publique appartient à Bob, son message avait cryptée par la clé de Mallory et envoie le message à Bob crypté. Mallory intercepte à nouveau, déchiffre le message en utilisant sa clé privée, peut-être modifie si elle veut, et re-il crypte avec la clé publique de Bob, elle a intercepté quand il a essayé à l'origine de l'envoyer à Alice. Lorsque Bob reçoit le nouveau message chiffré, il croit qu'il est venu d'Alice.

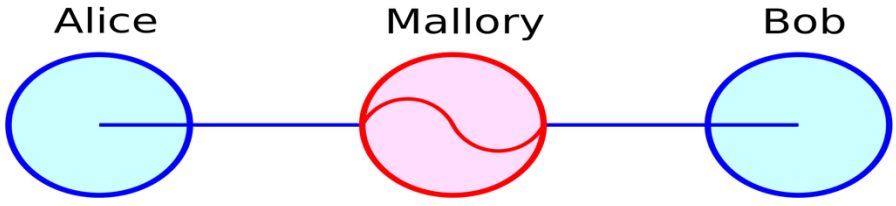


Figure 28 MITM scénario d'attaque dans une application de chat.

3.9.3 La défense et la détection

Les attaques de MITM peuvent être évitées ou détectées par deux moyens: l'authentification et la détection de sabotage. L'authentification fournit un certain degré de certitude qu'un message donné est venu d'une source légitime. Sabotage détection montre simplement la preuve qu'un message peut avoir été modifié.

3.10 Conclusion

Le chiffrement de bout en bout a longtemps été un outil formidable qui nous a aidé plusieurs fois, et l'avenir de cryptage de bout en bout comme un outil dans le public dépend vraiment des décisions d'aujourd'hui. Si la chiffrement de bout en bout est généralement acceptée comme normale, alors il y a une quantité sauvage de possibilités ouvertes pour le chiffrement et la cryptographie.

CHAPITRE 4

SIMULATION : IMPLEMENTATION D'ALGORITHME RSA

4.1 Introduction

Comme nous l'avons suggéré plus tôt dans ce memoire que E2EE est le type de cryptage le plus sûr qui peut garantir la sécurité et la confidentialité des informations partagées entre les smartphones, le reste de ce chapitre sera une partie pratique de ce memoire. Dans cette partie pratique, nous avons préparé une simple application de chat que nous utiliserons pour expliquer comment fonctionne E2EE dans la vie réelle.

4.1 Matériel nécessaire

4.1.1 ordinateur

Pour commencer à créer une application de chat qui peut effectuer les fonctionnalités de partage de l'information entre les smartphones et la sécurité de ces informations nous avons besoin d'avoir un ordinateur avec suffisamment de puissance de traitement, dans ce cas, nous avons besoin d'un ordinateur d'au moins 4 Go de RAM.

4.1.2 Système d'exploitation

La plupart du travail et à jour des systèmes d'exploitation tels que Windows, MacOS et Linux peuvent être utilisés pour développer l'application, tout dépendra du choix du développeur, et en fait le type d'ordinateur qu'il utilise. Par exemple, un développeur avec un ordinateur MacBook utilisera un MacOS et ainsi de suite.

4.2 Logiciel utilisé

4.2.1 Android studio

Le logiciel utilisé pour développer l'application est Android Studio la langue officielle pour le studio Android est Kotlin, dans cette simulation nous avons utilisé Java.

Android Studio est l'IDE officiel pour le système d'exploitation Android de Google, et comprend tout ce dont vous avez besoin pour créer des applications Android. Construit sur InteliJ IDEA de JetBrain , et conçu spécifiquement pour le développement des applications Android, il est disponible en téléchargement sur Windows, macOS, les systèmes d'exploitation Linux. Il est un remplacement pour l'Eclipse ADT (Eclipse Android Development Tool) comme IDE primaire pour le développement d'applications Android native.
Android Studio a été annoncé le 16 mai 2013 à la conférence Google. Il était au début de l'étape de prévisualisation d'accès à partir de la version 0.1 en mai 2013, puis est entré en version bêta à partir de la version 0.8 qui a été publié en Juin 2014. La première version stable a été publié en Décembre 2014, à partir de la version 1.0.

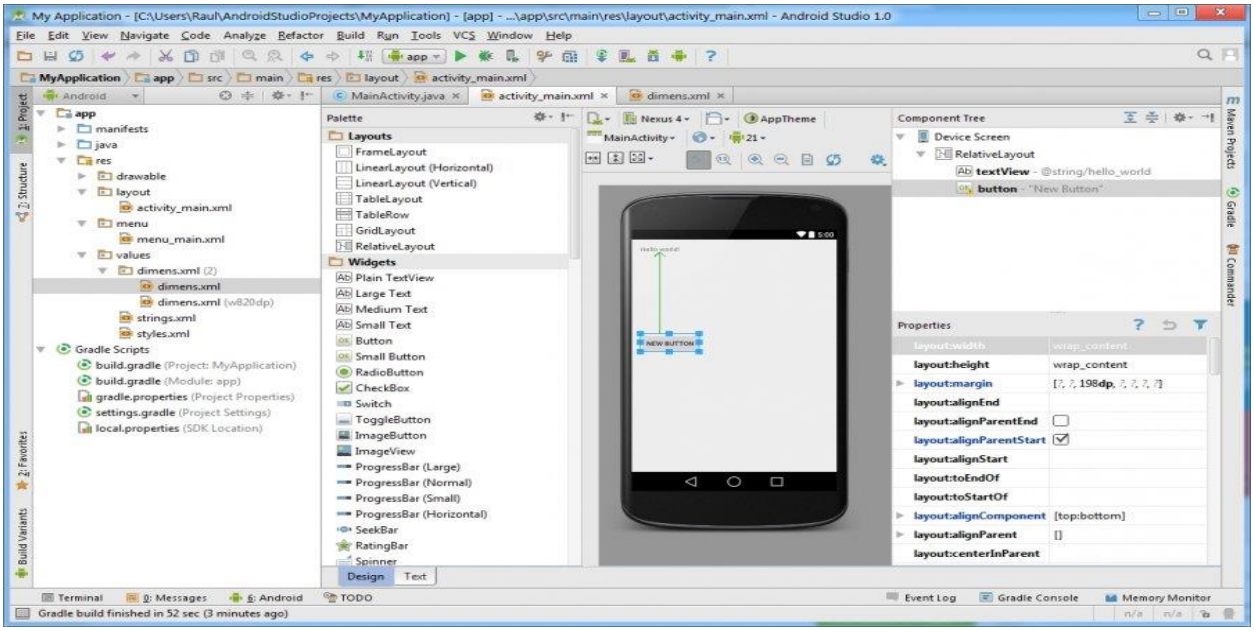


Figure 29 Android Studio

4.2.1.1 Système requis pour les versions actuelles d'Android studio

Critère	La description
OS version	Microsoft Windows 7/8/10 (32 bits ou 64 bits), 64 bits requis pour le débogage natif Mac OS X 10.10 (Yosemite) ou plus, jusqu'à 10.13 / 10.14 (MacOS supérieur Sierra / macOS Mojave) GNOME ou KDE Linux (64 bits capable d'exécuter l'application 32_bit) (Bibliothèque C GNU (glibc) 2.19+) Chrome OS (8 Go de RAM minimum ainsi que d'autres restrictions matérielles).
RAM	minimum 3 Go de RAM, 8 Go de RAM recommandés; plus 1 Go pour les applications de l'émulateur.
Espace disque	6 Go de minimum d'espace disque disponible, 16 Go recommandé (1700 Mo pour IDE + 4,4 Go pour Android SDK et de l'image du système Emulator).
Java version	Java Development Kit (JDK) 8, l'utilisation de Open JDK fourni est recommandé.
Résolution d'écran	1280x800 résolution d'écran minimale.

Tableau 4. Configuration minimale du système pour le studio Android.

4.2.1.2 Installation du studio Android

Windows
Pour installer Android Studio sur Windows, procédez comme suit :

1. Si vous avez téléchargé un fichier.exe (recommandé), double-cliquez dessus pour le lancer.
   Si vous avez téléchargé un fichier .zip, décompressez le fichier ZIP, copiez le dossier studio Android dans votre dossier Program Files, puis ouvrez l'androïde-studio> bin et studio64.exe de lancement (pour les machines 64 bits) ou studio.exe (pour les machines 32 bits).

2. Suivez l'assistant d'installation dans Android Studio installer tous les packages SDK qu'il recommande.

Mac
Pour installer Android Studio sur votre Mac, procédez comme suit :

1. Lancez le fichier Android studio DMG.
2. Glissez et déposez Android studio dans le dossier Applications, puis lancer Android Studio.
3. Sélectionnez si vous voulez importer les paramètres précédents Android Studio, puis cliquez sur OK.
4. L'Assistant Studio Setup Android vous guide à travers le reste de l'installation, qui comprend le téléchargement de composants Android SDK qui sont nécessaires pour le développement.

Linux
Pour installer Android Studio sur Linux, procédez comme suit :

1. Décompressez le fichier .zip téléchargé à un emplacement approprié pour vos applications, telles que dans les / usr / local / pour votre profil d'utilisateur, ou / opt / pour les utilisateurs partagés.
2. Pour lancer Android Studio, ouvrez un terminal, accédez à l'androïde studio / bin / et exécuter studio.sh.
3. Sélectionnez si vous voulez importer les paramètres précédents Android Studio ou non, puis cliquez sur OK.
4. L'Assistant Studio Setup Android vous guide à travers le reste de l'installation, qui comprend le téléchargement de composants Android SDK qui sont nécessaires pour le développement.

Bibliothèques nécessaires pour les machines 64 bits :
Si vous utilisez une version 64 bits d'Ubuntu, vous devez installer des bibliothèques 32 bits avec la commande suivante :
$ Sudo apt-get install libc6: i386 libncurses5: i386 libstdc ++ 6: i386 lib32z1 libbz2-1.0: i386
Si vous la commande est en cours d'exécution Fedora 64 bits,:

$ Sudo yum install zlib. i686 ncurses-libs.i686 bzip2-libs.i686

Après avoir installé Android Studio Lancez le logiciel et suivez les étapes simples :

4.3 Choisissez votre projet:

Dans le choix de votre écran de projet qui apparaît, vous pouvez sélectionner le type de projet que vous souhaitez développer, par exemple, téléphone et tablette, TV, auto ou Android choses Android. Dans notre cas pour le développement Android, nous choisissons téléphone et tablette, puis cliquez sur suivant pour continuer. En sélectionnant le type de projet et templet dans le choix de votre écran de projet, Android Studio comprendra des codes pour vous aider à démarrer.

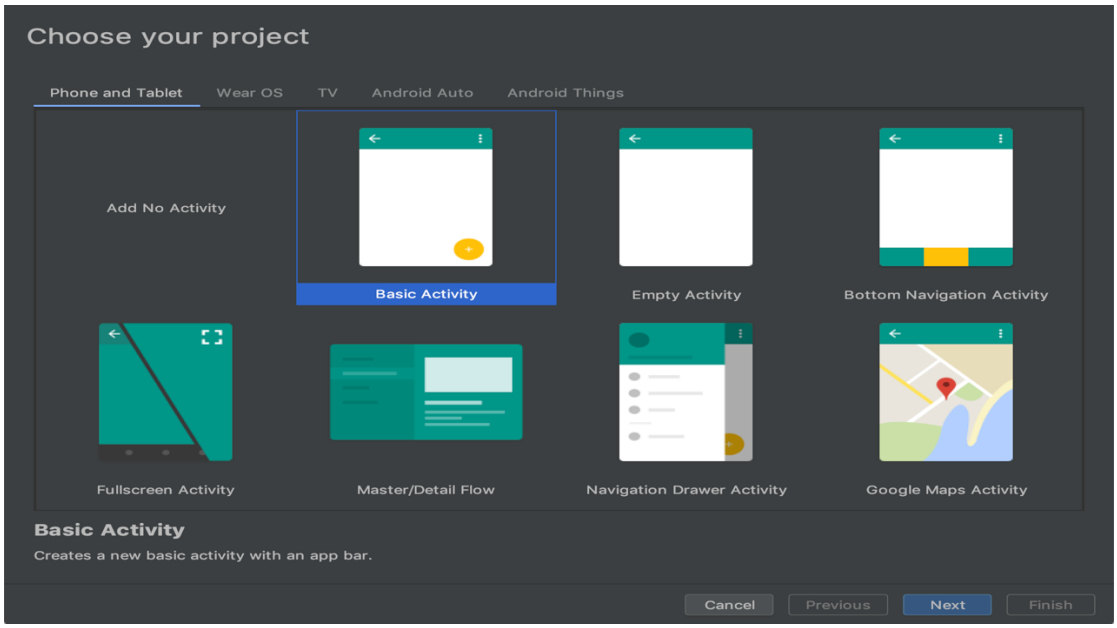


Figure 30 choisir le type de projet et TEMPLET vous voulez créer.

L'étape suivante consiste à configurer certains paramètres et créez votre nouveau projet, comme décrit ci-dessous et illustré à la figure 32. Indiquez le nom du projet, le nom du package, sauf l'emplacement et le niveau de l'API minimum que vous voulez que votre application pour soutenir, puis cliquez sur terminer pour commencer Coding.

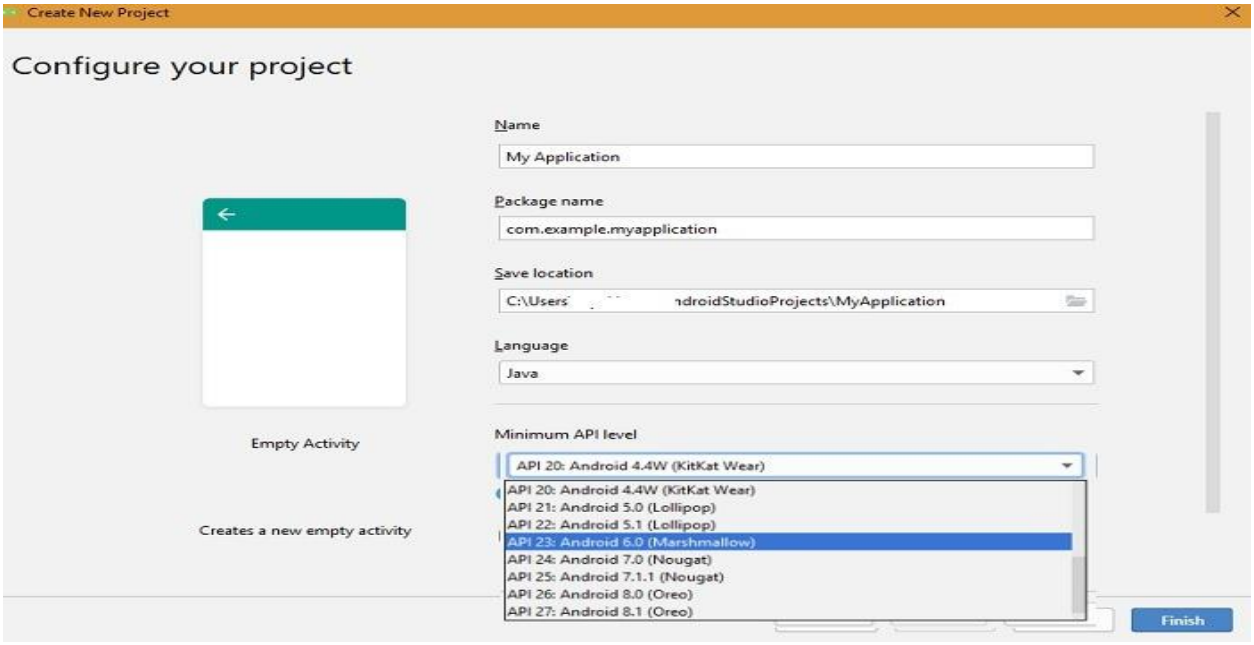


Figure 31 configuration d'un projet

4.3.1 Base de données Firebase

Firebase est la base de données Google qui fournissent les fonctionnalités nécessaires pour Android, iOS et Java Script fin de retour tels que le stockage de données, l'authentification des utilisateurs, l'hébergement statique et plus. Avec Firebase un développeur se concentrera sur la création d'une expérience extraordinaire utilisateur. Soutien Firebase croisées plates-formes natives des applications web et mobiles en raison de la présence d'Android, iOS et Java SDK de script.

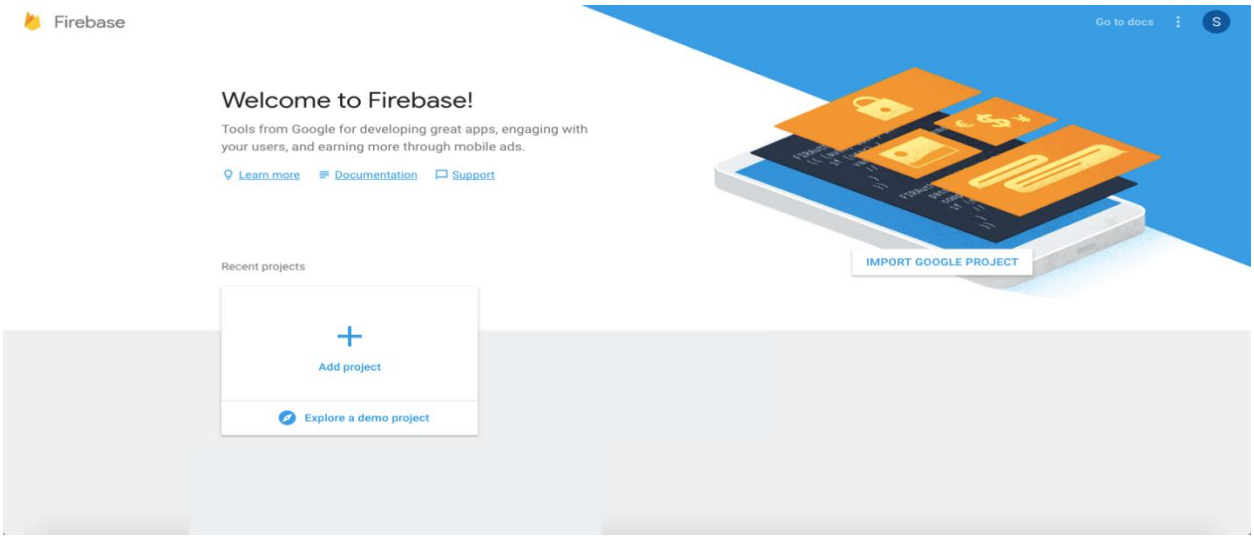


Figure 32 Firebase Base de données

4.3.1.1 caractéristiques Firebase

- **Base de données en temps réel :**

La base de données Firebase en temps réel est une base de données de cloud hébergé. Les données sont stockées sous forme de JSON et synchronisé en temps réel à chaque client connecté. Lorsqu'un développeur construit des applications multiplateforme avec iOS, Firebase Android et SDKs JavaScript, tous ses clients partagent une instance de base de données en temps réel et recevoir automatiquement des mises à jour avec les données les plus récentes.

- **L'authentification Firebase**

La plupart des applications ont besoin de connaître l'identité d'un utilisateur. Connaissant l'identité d'un utilisateur permet une application pour enregistrer en toute sécurité des données utilisateur dans le cloud et de fournir la même expérience personnalisée sur l'ensemble des appareils de l'utilisateur.
Firebase authentification fournit des services de back-end, faciles à SDKs utiliser, et les bibliothèques de l'interface utilisateur toutes faites pour authentifier les utilisateurs à une application. Il prend en charge les mots de passe à l'aide d'authentification, les numéros de téléphone, populaires fournisseurs d'identité fédérée comme Google, Facebook et Twitter, et plus encore.

- **Hébergement Firebase**

L'hébergement est Firebase contenu web production de qualité d'hébergement pour les développeurs. Avec une seule commande, vous pouvez déployer rapidement des applications Web et servir à la fois le contenu statique et dynamique à un (réseau de diffusion de contenu) global CDN. Un développeur peut également coupler un hébergement Firebase avec des fonctions Cloud ou Cloud Exécuter pour construire et héberger des micro services sur Firebase.

4.4 Caractéristiques de la simulation Chat App

Dans cette partie, nous allons expliquer les caractéristiques de la simulation qui apportent et assure la sécurité des données partagées entre les smartphones utilisant cette application de chat.
Dans la Simulation que nous avons préparée, nous avons utilisé des bibliothèques spécialisées de la sécurité de Virgile qui sont capables de générer clés (clés public/privé) pour le cryptage et le décryptage des données [23].

4.4.1 Enregistrement de l'utilisateur

Enregistrement de l'utilisateur est le premier long métrage au début de l'application, un utilisateur sera tenu de préciser ses lettres de créance d'enregistrement, peut être le nom d'utilisateur, adresse e-mail ou numéro de téléphone et mot de passe. Informations d'enregistrement sont utilisés pour créer l'identité de l'utilisateur et une clé du cerveau dans

l'application et dans la base de données qui sera utilisée pour l'identification et l'authentification de l'utilisateur à tout moment, il / elle vous connecter à l'application.

4.4.2 génération de clés

Au cours de la première inscription de l'utilisateur une paire de clés (clé privée et publique) est créé qui sera ensuite associée à l'identité de l'utilisateur. Cette paire de clés est responsable pour le chiffrement et le déchiffrement des données partagées avec d'autres utilisateurs.

Dans cette application, le chiffrement et le déchiffrement fonctionne de la même manière que l'algorithme de chiffrement asymétrique RSA expliqué dans le chapitre 2 et 3. Un expéditeur va crypter les données en utilisant la clé publique du récepteur et le récepteur va décrypter les données en utilisant sa propre clé privée.

4.4.3 La distribution et stockage des clés

Après la génération de la paire des clés, la clé publique est publiée dans le cloud. Clé publique de chaque utilisateur est publié dans le cloud afin que chacun puisse avoir accès à ceux clé publique. Quand Bob veut envoyer des informations à Alice par exemple, même s'il n'est pas dans sa liste de contacts, il peut accéder à sa clé publique et chiffrer les données avant de les envoyer à elle.

La clé privée d'autre part est stockée dans la zone sécurisée en toute sécurité de l'appareil de l'utilisateur. La clé privée est celle qui est utilisée pour le déchiffrement des données reçues par conséquent, il doit être secret dans la zone sécurisée en toute sécurité de l'appareil de l'utilisateur.

4.4.4 récupération des clés pour un téléphone perdu

Lors de l'enregistrement de l'utilisateur une clé du cerveau (Brain key) est généré en fonction du nom d'utilisateur et mot de passe utiliser pour l'enregistrement. Cette clé du cerveau est utilisée pour chiffrer la clé privée qui est ensuite stocké dans le cloud. Lorsqu'un utilisateur se connecte avec un autre appareil, le nom d'utilisateur et mot de passe créer une clé de cerveau correspondant à celui utilisé pour chiffrer la clé privée. Cette nouvelle clé de serveur télécharge la clé privée chiffrée du le cloud vers l'appareil de l'utilisateur, décrypter la clé privée et le stocker dans l'appareil de l'utilisateur prêt à être utilisé pour le décryptage des données.

4.4.5 chiffrement de bout en bout (End-to-End cryptography)

Le type de cryptage utilisé dans cette simple application de chat est E2EE où les données partagées entre les périphériques sont cryptées et décryptées dans le dispositif de l'utilisateur final. Informations restent cryptées dans tous les intermédiaires, y compris les bases de données. Ni le développeur de l'application, ni le gestionnaire de base de données, ni les pirates peuvent déchiffrer ces données car les clés utilisées pour le décryptage ne se trouvent que dans le dispositif de l'utilisateur final.

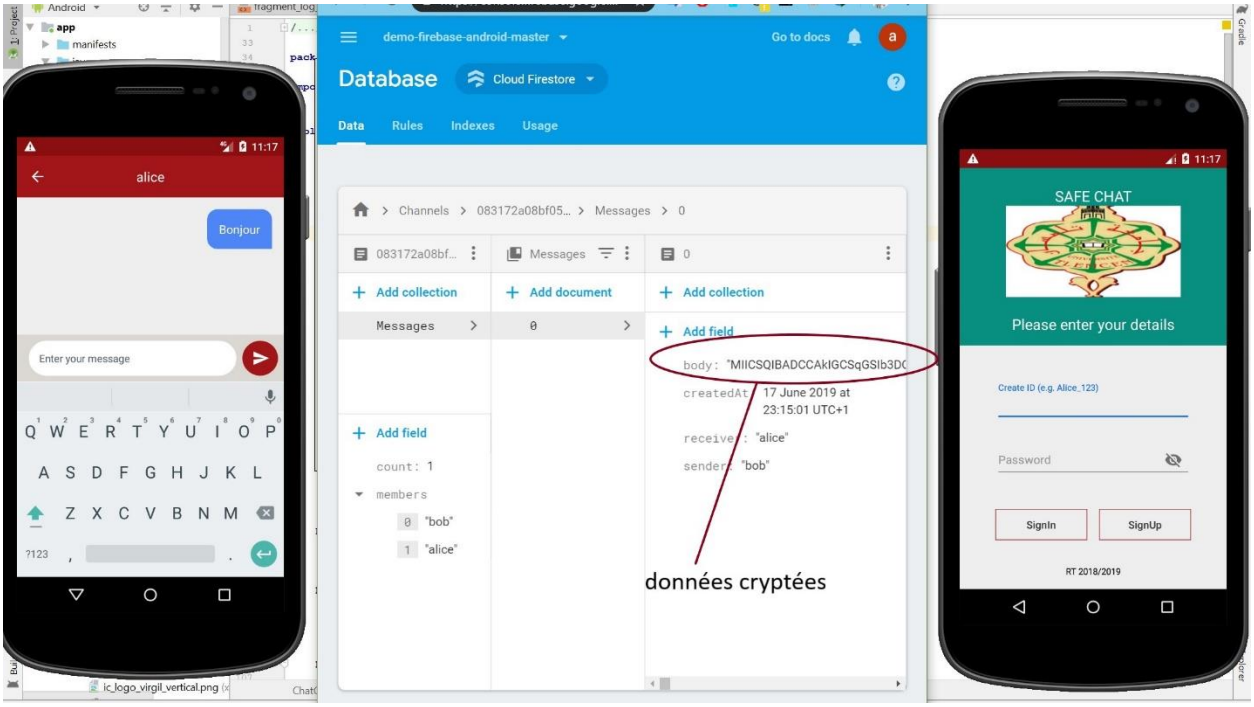


Figure 33 Données cryptées lorsque la destination est hors-ligne

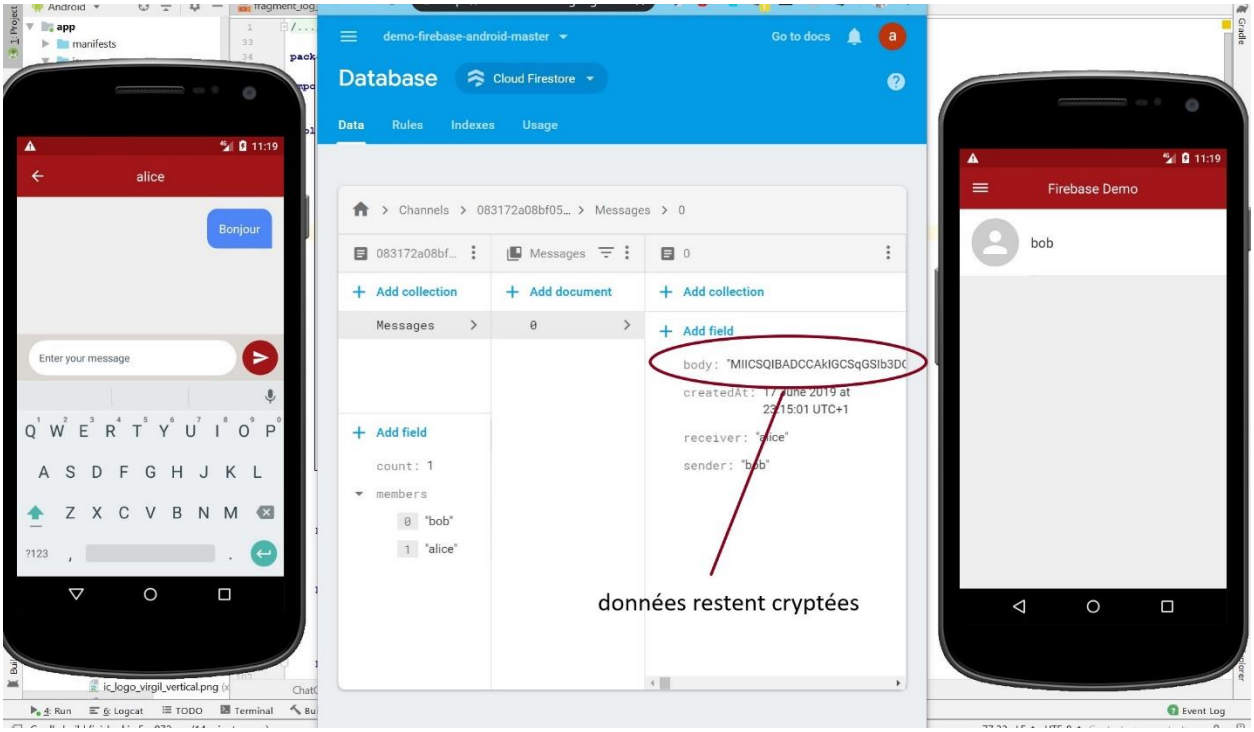


Figure 34 Données restent chiffrées lorsque elles n'ont pas encore lu

4.4.6 Disparition automatique des données

En plus de la E2EE des données partagées entre les utilisateurs, les données sont effacées également de celles de la base de données qu'ils sont reçus aux récepteurs. Cette fonction ajoute une autre couche importante de la sécurité où non seulement que les données sont toujours cryptées dans les serveurs et autres intermédiaires, mais aussi toutes les données sont supprimées de tous les intermédiaires après la réception réussie. Les informations partagées ne se trouve que dans le serveur lorsque le destinataire prévu n'a pas lu le message ou est en ligne, une fois que le dispositif est en ligne, il reçoit l'information, mais il reste non-supprimer dans le serveur jusqu'à ce qu'il ait été lu.

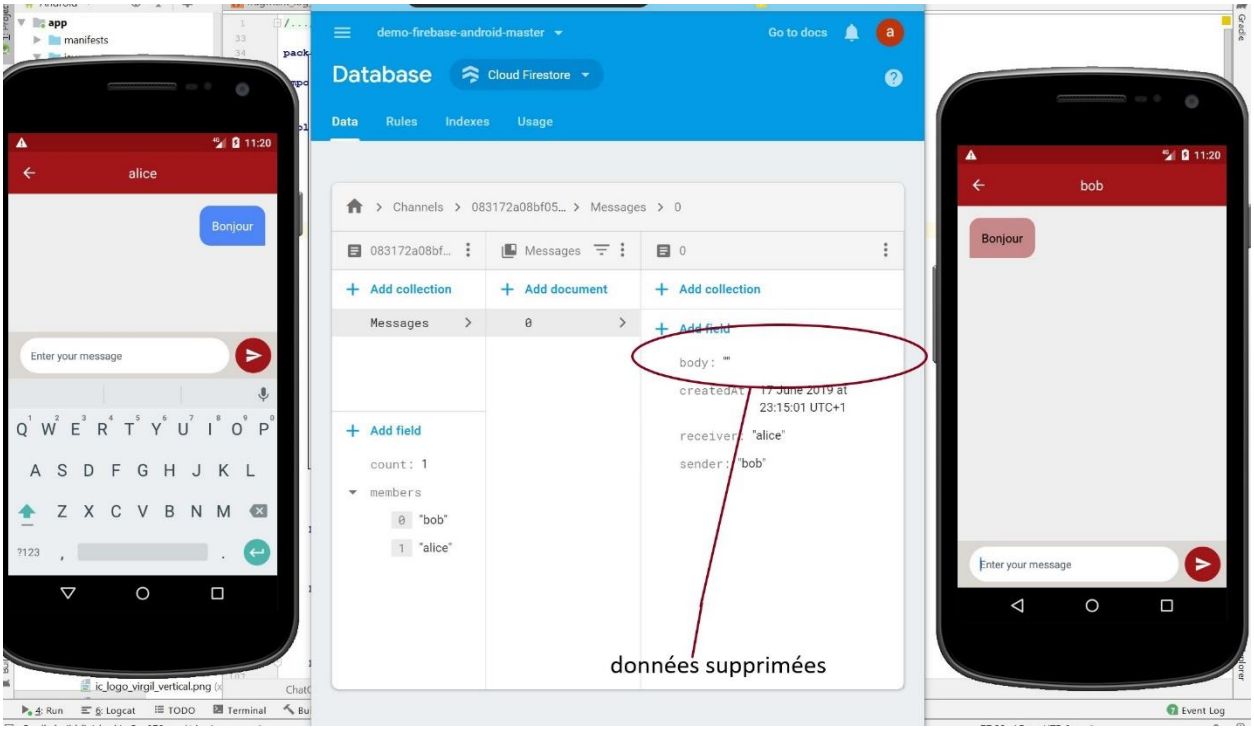


Figure 35 Données supprimées quand est déjà lu

4.5 Conclusion

Cryptographie de bout en bout avec RSA est la méthode la plus sécurisée utilisé dans la communication sur internet. Il est la méthode appliquée la plupart dans des applications de smartphones comme WhatsApp t'chat, Facebook Messenger, Viber et etc. Il est la méthode recommandée en raison de la complexité de fissurer les informations partagées entre les utilisateurs.

CONCLUSION GENERALE

Selon l'évolution actuelle de la science et de la technologie dans le monde entier, il est clair qu'à un moment donné dans l'avenir, la plupart des aspects de la vie dépendront de la technologie. Compte tenu de l'échange de volume importants d'informations sur Internet lors de la communication entre humains, nous avons également constaté la connexion d'un grand nombre d'équipements et de machines (Internet of Things, IoT) à internet qui échangeront entre eux sans aucun doute une quantité importante de données sur internet. La plupart de ces appareils et machines dépendent des données transférées entre elles pour effectuer les activités souhaitées. Certaines de ces machines, par exemple les voitures et les stimulateurs cardiaques, touchent directement la vie humaine. Par conséquent, ils ont besoin d'une sécurité élevée des données qu'ils utilisent pour mener à bien leurs activités comme prévu.

Par conséquent, après avoir examiné de nombreux types et formes de cryptographie, nous sommes arrivés à la conclusion que cryptographie de bout en bout (avec RSA comme une technique de cryptage) des données sur internet est le seul espoir de sécurisation des données sur internet si elles sont correctement appliquées.

PERSPECTIVE

Le domaine de la communication continue de changer d'une génération à l'autre et pour assurer la sécurité des données dans la communication, la sécurité doit évoluer ainsi. Pour le moment, le cryptage RSA est la méthode la plus sûre de sécurité pour sécuriser les données, mais les pirates travaillent dur aussi pour le casser et manipuler les données à leur volonté. C'est pourquoi les recherches pour de nouvelles méthodes de sécurité les plus robuste que RSA doit être fait pour assurer l'avenir de la sécurité des données. Dans ce travail, nous avons expliqué le fonctionnement de la RSA. Nous encourageons les générations futures à essayer d'autres méthodes telles que la courbe elliptique et à comparer la différence.

REFERENCES

[1] Gunar Heine, GSM Networks: protocols, Terminology and implementation, 31-37 (1998).

[2] Claude Castellucia INRIA, GSM Security, *Article,* 2016

[3] Juha T. Vainio, Bluetooth Security, *Article, 2000*

[4] Christian Gehrmann,Joakim Persson and Ben Smeets, Bluetooth Security, 54-80, 2004

[5] Nelli Gordon and Sean Vakili, Bluetooth Security, *Article*, May 10th 2011

[6] Jain R.Jejurkar, R. Chopade, S.Vaidya, & Sanap M. ,AES Algorithm Using 512 Bit Key Implementation for Secure Communication. International journal of innovative Research in Computer and Communication Engineering, 2014.

[7] Selmane N., Guilley S., & DangerJ. L. ,Practical setup time violation attacks on AES. In Dependable Computing Conference, 2008. EDCC 2008. Seventh European (pp. 91- 96). IEEE ,2008, May.

[8] Kretzschmar, U. AES128–AC Implementation for Encryption and Decryption. TI-White Paper, 2009.

[9] Benvenuto, C. J. Galois field in cryptography. University of Washington, 2012.

[10] Jean-Philippe Aumasson, SERIOUS CRYPTOGRAPHY A Practical Introduction to Modern Encrption,2014

[11] Erik Wehner, Evan Moran: End-to-end encryption, an answer to security concerns in private sectors. 2018

[12] Katriel Cohn-Gordon, Cas Cremers, Luke Garrat, John Millican and Kevin Milner: On end to end encryption, Asynchronous Group Messaging With Strong Security Guarentee: 2018

[13] Nadim Kobeissi: An analysis of the ProtonMail cryptography architecture: 2018

[14] David Szabo: end to end encryption with firebase: javebratt.com/firebase-end-to-end-encryption/ August 15, 2018

[15] End to end encryption, Glossary/virgil security.

[16] Whatsapp FAQ-End to end encryption: faq.whatsapp.com/en/android/

[17] Takanori Isobe et Kazuhiko Minematsu: Breaking Message Integrity of an End-to-End Encryption Scheme of LINE

[18] T. Kivinen, M. Kojo, *More Modular Exponential (MODP) Diffie–Hellman groups for Internet Key Exchange (IKE)*. SSH Communications Security. May 2003.

[19] Wikipedia, the free encyclopedia: *Diffie–Hellman Key Exchange 2019*

[20] Adam C. Champion, Ph.D: Threats and Attacks, Information Security

[21] https://developer.virgilsecurity.com/docs/sdk-and-tools , 2019 Virgil Security, Inc.

RÉSUMÉ

La sécurité des données entre smartphones concerne la sécurité des données d'un expéditeur au destinataire. Les données sont envoyées ou partagées selon différentes méthodes, comme Internet, et il y a toujours une vulnérabilité de piratage de ces données si elles ne sont pas sécurisées. Dans ce memoire, nous avons montré comment les données sont sécurisées en utilisant l'algorithme RSA.

Mots clés : chiffrement, déchiffrement, algorithme, RSA, authentification, Intégrité, clé publique, clé privée

ABSTRACT

The security of data between smartphones concerns the security of the data of a sender to the recipient. Data is sent or shared using different methods, such as the Internet, and there is always a vulnerability to hacking this data if it is not secure. In this Research, we have shown how data is secured by using RSA algorithm.

Key words: Encryption, Decryption, Algorithm, RSA, Authentication, integrity, public key, private key,