République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: *Réseaux et Systems Distribué (R.S.D)*

Thème

Cryptage des flux audio par les algorithmes à clé publique selon le modèle client/serveur

Réalisé par :

- Bouhassoun Abdelkader
- Beyoud Lout

Présenté le 24/06/2015 devant le jury:

- Labraoui N (Présidente)
- Benaissa S (Encadreur)
- Benmouna M (Examinateur)
- Belhoucine A (Examinateur)

Année universitaire : 2014-2015

Dédicace

Au nom du ALLAH le clément et le miséricordieux je dédie ce modeste travail :

Mes chers parents qui ont toujours été dévoués pour que je puisse réaliser ce travail de recherche dans les meilleures conditions.

A mes proches amis et toutes mes grandes familles.

Et toute la famille informatique et bien sur**,** ***Mes professeurs*** 2014-2015(spécialité R.S.D).

Remerciement

Nous nous sommes permis de consacrer se petit espaces pour remercie infiniment tous s'eux qui ont participé de prés ou de loin à la réalisation, et puis à la réussit de notre projet.

Un très grand merci à :

Mes parents qui m'en soutenu toute au lent de mon cursive d'étude. Touts mes enseignants qui mon éclairer le bon chemin, et surtout à Mr.Benaissa.Samir, pour l'encadrement technique et pour m'avoir guidé, encouragé et conseillé pendant toute la période de formation. Je tiens à mentionner le plaisir que j'ai eu à travailler avec lui.

A mes frères, mes oncles, tantes, cousins, cousines, grands parents. Et tous mes amis qui ont cru en moi, m'ont encouragé et m'on donné la force d'aller jusqu'au bout.

A tous mes collègues de promotion que j'ai eu le plaisir de côtoyer pendant cette période de formation. Une pensée va particulièrement à tous ceux d'entre nous qui n'ont pas eu la possibilité d'aller jusqu'au bout de leur formation.

Liste des figures

Chapitre 1

Chapitre 2

Chapitre 3 :

Chapitre 4:

Liste des tableaux

Chapitre 1

Chapitre 2

Chapitre 4

Table de matières :

Table de matières

Table de matières

Table de matières

Introduction général

Depuis sa création, le réseau internet a tellement évolué qu'il est devenu un outil essentiel de communication.

Il n'est plus d'actualité aujourd'hui de parler de la nécessaire intégration des réseaux informatiques dans les systèmes d'information : c'est désormais un fait acquis.

Mais, faire ce constat ne suffit pas à résoudre les différentes problématiques liées à cette nouvelle dimension et, entre autre, la prise en compte des aspects sécurité.

Actuellement aucune entreprise ne peut se passer d'outils informatiques, et très souvent un réseau informatique de taille plus ou moins importante est mis en œuvre. Le nombre des machines dans ces réseaux peut parfois devenir extrêmement élevé; La maintenance ainsi que la gestion de ces parcs informatiques deviennent alors des enjeux cruciaux, d'autant plus qu'une panne du réseau peut parfois avoir des conséquences catastrophiques.

Aujourd'hui, en particulier avec le développement d'internet, transmettre des informations confidentielles de façon sécurisée est devenu un besoin primordial... Aussi, bien qu'il s'agisse d'une science très ancienne, la cryptologie est toujours d'actualité. Décryptons l'un des algorithmes les plus utilisés, l'algorithme RSA, basé sur une propriété simple des nombres premiers.

La cryptographie est un sujet important, c'est notre monde soucieux de la sécurité, et il est particulièrement important pour les entreprises dont l'activité est la transmission et le stockage de données confidentielles. Le cryptage symétrique et le chiffrement asymétrique sont largement utilisés sur Internet, d'assurer la sécurité pour le commerce électronique et d'autres transactions.

La sécurité informatique, d'une manière générale, consiste à assurer que les ressources matérielles ou logicielles d'une organisation sont uniquement utilisées dans le cadre prévu.

Introduction général

L'objectif principal de notre projet de fin d'étude est d'assurer un transfert sécurisé de données représenté par des flux audio en utilisant des algorithmes de chiffrements à clé publique et à clé secrète.

Ce système de chiffrement et de sécurité des fichiers audio est implémenté dans une architecture client/serveur.

Notre mémoire est structurée comme suite :

Chapitre 1 : étude de l'architectures client/serveur.

Chapitre 2 : introduction aux algorithmes de chiffrement a clé publique et a clé secrète.

Chapitre 3 : présentation des Formats des fichiers audio.

Chapitre 4 : implémentation de l'application pour sécuriser et la protéger des fichiers son.

Chapitre1:

étude de l'architectures client/serveur.

1.1. Introduction

Dans l'informatique moderne, de nombreuses applications fonctionnent selon un environnement client-serveur.

Le modèle client-serveur s'articule autour d'un réseau auquel sont connectés deux types d'ordinateurs le serveur et le client. Ils communiquent via des protocoles.

L'environnement **client-serveur** désigne un mode de communication à travers un réseau entre plusieurs programmes ou logiciels :

L'un, qualifié de client, envoie des requêtes ; les serveurs, attendent les requêtes des clients et y répondent. Par extension, le client désigne également l'ordinateur sur lequel est exécuté le logiciel client, et le serveur, l'ordinateur sur lequel est exécuté le logiciel serveur.

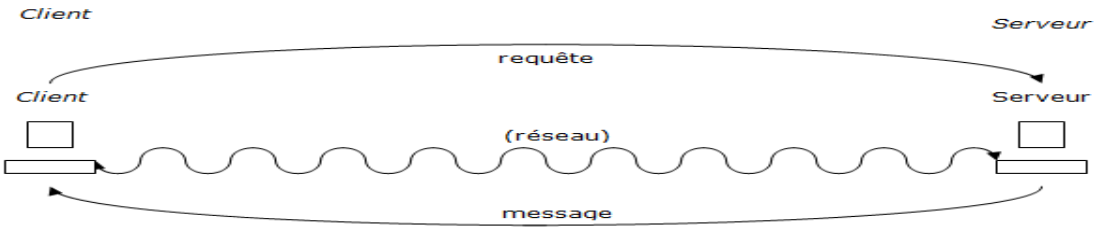


Fig. 1.1 : modèle client/serveur

1.2. Définition d'un serveur

On appelle logiciel serveur un programme qui **offre un service** sur le réseau. Le serveur accepte des requêtes, les traite et renvoie le résultat au demandeur (client). Le terme serveur s'applique à la machine sur lequel s'exécute le logiciel serveur. Le client/serveur représente un dialogue entre deux processus informatiques par l'intermédiaire d'un échange de messages.

Pour pouvoir offrir ces services en permanence, le serveur doit être sur un site avec **accès permanent** et s'exécuter en permanence.

Un site peut offrir plusieurs services. Chacun de ces services est fourni sur **un port de communication** identifié par un **numéro**. Ce numéro identifie le service quelque soit le site. [s1]

Exemple :

- Le service **FTP** est offert sur les ports numéros 21 (contrôle) et 20 (données),

- Le service **TELNET** (émulation terminal) sur le port 23,
- Le service **SMTP** (mail) sur le port 25.
    - Pour **accéder à un service**, il faut **l'adresse du site** et le **numéro du port**.

Les types de serveurs

Serveur itératifs

- ne gèrent qu'un seul client à la fois.
- traite séquentiellement les requêtes
- adapté aux requêtes qui peuvent s'exécuter rapidement
- souvent utilisé en mode non connecté (recherche de la Performance)

Serveur concurrent :

- Parallélisme réel- Pseudo parallélisme
- le serveur accepte les requêtes puis les "délègue" à un processus fils (traitement de plusieurs clients)
- adapté aux requêtes qui demandent un certain traitement (le coût du traitement est suffisamment
- important pour que la création du processus fils ne soit pas pénalisante)
- souvent utilisé en mode connecté

1.3. Définition d'un client

On appelle logiciel client un programme qui **utilise le service** offert par un serveur. Le client envoie une requête et reçoit la réponse. Le client peut-être raccordé par une liaison temporaire.

Les types de client :

- **Client léger :** Le poste client accède à une application située sur un ordinateur dit « serveur » via une interface et un navigateur Web. L'application fonctionne entièrement sur le serveur, le poste client reçoit la réponse « toute faite » à la demande (requête) qu'il a formulée.
- **Client lourd :** Le poste client doit comporter un système d'exploitation capable d'exécuter en local une partie des traitements. Le traitement de la réponse à la requête du client utilisateur va mettre en œuvre un travail combiné entre l'ordinateur serveur et le poste client.

1.4. Caractéristiques des systèmes client serveur

Les éléments qui caractérisent une architecture client serveur sont :

Services : Le serveur est un fournisseur de services. Le client est un consommateur de services.

Partage de ressources : Un serveur traite plusieurs clients et contrôle leurs accès aux ressources.

Protocole asymétrique : le protocole de communication est asymétrique le client déclenche le dialogue, le serveur attend les requêtes des clients.

Transparence de la localisation : Transparence par rapport aux systèmes d'exploitation et aux plates-formes matérielles. Idéalement, le logiciel client serveur doit être indépendant de ces deux éléments.

Message : Les messages sont les moyens d'échanges entre client et serveur.

Encapsulation des services : Le serveur décide de la façon de le rendre une mise à niveau du logiciel serveur doit être sans conséquence pour le client tant que l'interface message est identique.

Evolution : Une architecture client serveur doit pouvoir évoluer horizontalement (évolution du nombre de clients) et verticalement (évolution du nombre et des caractéristiques des serveurs).

1.5. Les architectures C\S

1.5.1 : Triptyque d'une application

Présentation : correspondant à l'interface homme-machine (IHM). Contient les différents types de clients (lourd et leger).

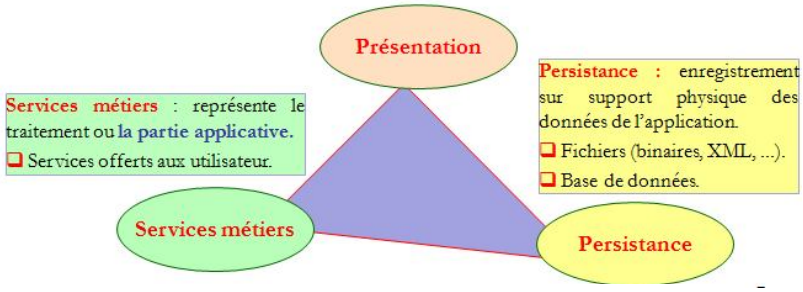


Fig. 1.2 : Triptyque d'une application

1.5.2 : Architecture 1 – tiers

- **Dans un contexte distribué :**
  - Les couches peuvent être exécutées sur des machines différentes.
  - Certains couches peuvent être sous découpées.
  - De nombreuses variantes de placement des couches et de leur distribution

Tout est sur la même machine :

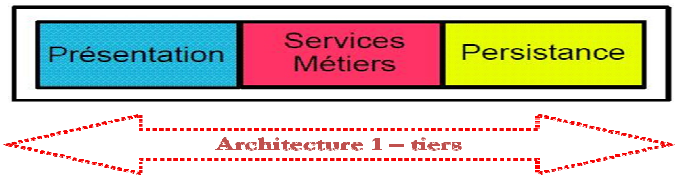


Fig. 1.3 : architecture 1-tiers

1.5.3 : Architecture 2 – tiers

- **Client :** présentation, IHM.
- **Serveur :** persistance, gestion physique des données.
- Les services métiers (partie applicative) peuvent être :
  - ***Soit entièrement coté client, intégrés avec la présentation.***
    - La partie serveur ne gère que les données.
  - ***Soit entièrement coté serveur.***
    - La partie client ne gère que l'interface utilisateur.
  - ***Soit découpés entre la partie serveur et la partie client.***
- **Client : présentation + traitement.**

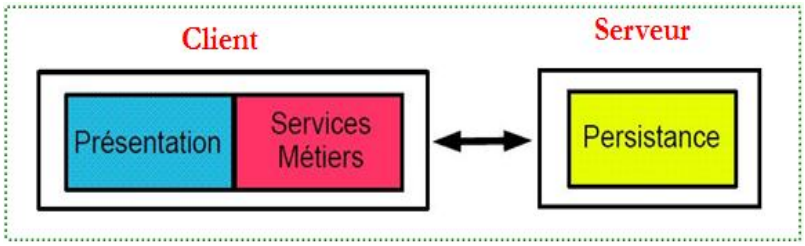


Fig. 1.4: architecture 2-tiers –a-

- **Serveur : traitement + gestion des données.**

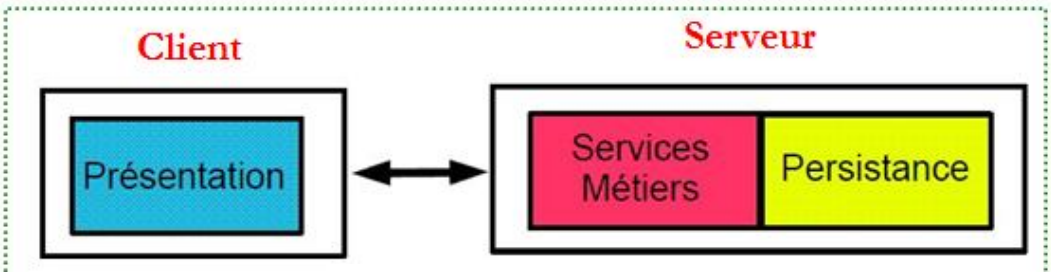


Fig. 1.5: architecture 2-tiers –b-

- **Applicatif : découpé entre client et serveur.**

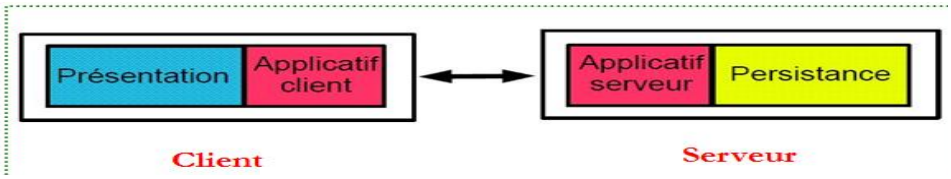


Fig.1.6: architecture 2-tiers c-

1.5.4 : Architecture 3 – tiers

- **Les 3 principales couches s'exécutent chacun sur une machine différente :**
  - **Présentation.**
    - Machine cliente.
  - **Applicatif / métier.**
    - Exemple : sur un serveur d'applications.
  - **Persistance.**
    - Exemple : sur un serveur de base de données

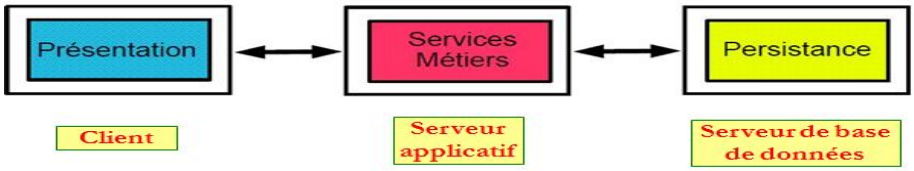


Fig. 1.7 : architecture 3-tiers

1.5.5 : Architecture n – tiers / n > 3

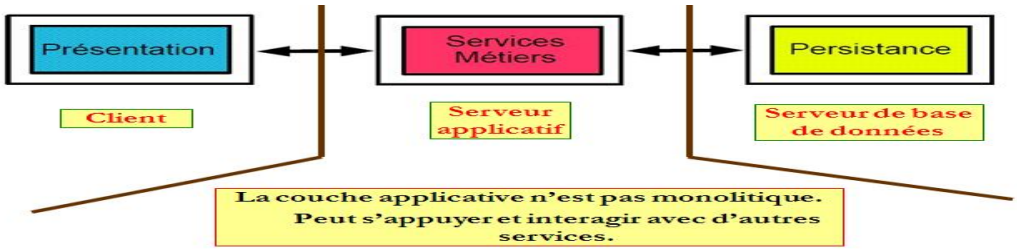


Fig. 1.8 : architecture n-tires

- Composition de la couche applicative.
  - **Composition verticale.**
    - Les services métiers peuvent s'appuyer sur des services techniques.
      - Sécurité.
        - ***Exemple : Serveur Radius pour l'authentification.***
      - ...
  - **Composition horizontale.**
    - Service métier utilise d'autres services métiers.
      - ***Exemple : Architecture 4 tiers dans la norme J2EE.***
- Chaque service correspond à une couche.
  - D'où le terme de **n-tiers**. [2]

1.6 : La notion de protocole et port

Lors d'une communication en réseau, les différents ordinateurs s'échangent des informations qui sont généralement destinées à plusieurs applications (le client mail et le navigateur internet par exemple).

Le modèle client/serveur est relié avec Un système dans le réseau à partir de défection de protocole et port utilisé pour charger la communication.

Certains protocoles utilisent des échanges de flux en texte (ex : ASCII) et peuvent être compris par des humains (ex: SMTP, POP3, HTTP, etc.). D'autres utilisent des échanges de flux binaires (ex : DNS).

Chaque paquet réseau contient

- L'adresse IP de la machine d'origine (le client dans le cas d'une requête),
- L'adresse IP de la machine de destination (le serveur dans notre cas),
- Numéro de port.

1.6 .1 : La notion de port

Un service rendu par un programme serveur sur une machine est accessible par un port et permet de savoir à quel service vas analysée le paquet .Un port est identifié par un entier (16 bits).

- De 0 à 1023 : port reconnus ou réservés.
- De 1024 à 49 151 : ports registrés.
- De 49 152 à 65 535 : ports dynamiques ou privés. [3]

Le port	Les services
21	FTP
23	Telnet
25	SMTP
53	DNS
80	http
110	POP3
119	NNTP

Tab1.1 : Liste des services et les ports

1.6.2 : Notion de protocoles

Un protocole est un langage spécifique a un type de service particulier, le client et le serveur se communique et dialogue par ce langage.

1.6.3 : But protocoles de communication

-Compréhension entre machines /logiciels

-Communications indépendantes du système d'exploitation ou de la plate-forme

- Internet est un ensemble de protocoles regroupés sous le terme "TCP-IP" (**T**ransmission **C**ontrol **P**rotocol/**I**nternet **P**rotocol). [3]

1.6.4 : Les deux catégories protocoles

Les protocoles où les machines s'envoient des accusés de réception (pour permettre une gestion des erreurs). Ce sont les protocoles "orientés connexion"

Les autres protocoles qui n'avertissent pas la machine qui va recevoir les données sont les protocoles "non orientés connexion". [1]

1.7 : Modes de fonctionnement client/serveur

Il existe deux modes :

- **Mode connecté (comparable à une communication téléphonique) TCP,** utilisant le protocole TCP. Dans ce mode de communication, une connexion durable est établie entre les deux processus, de telle façon que l'adresse de destination n'est pas nécessaire à chaque envoi de données.
- **Mode non connecté (analogue à une communication par courrier) UDP,** utilisant le protocole UDP. Ce mode nécessite l'adresse de destination à chaque envoi, et aucun accusé de réception n'est donné. [s2]

1.7.1 : Caractéristiques du mode connecté

- Tout d'abord, la connexion doit être établie entre les sockets.
- La transmission est sans erreur à cause de protocole utulisé (TCP).
- Tant que la connexion est maintenue, la transmission peut se faire dans les 2 sens : sur chaque socket il est possible de recevoir et d'émettre, donc 2 flux par socket.
- Pour établir la connexion, il faut qu'un programme prenne l'initiative : ce sera le client, tandis que l'autre doit être à l'écoute et/ou l'attente : ce sera le serveur.

1.7.2 : Caractéristiques du mode non connecté

-pas d'établissement préalable d'une connexion

-Dans ce mode non connecté, chaque message est transmis avec l'adresse du socket destination et celui du socket émetteur.

-protocole de transport utilisé : UDP

-La transmission n'est pas sure et les messages peuvent arriver dans le désordre. [s1]

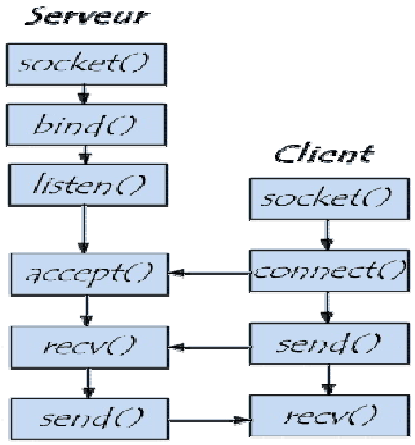


Fig.1.9 :
Mode connecté TCP

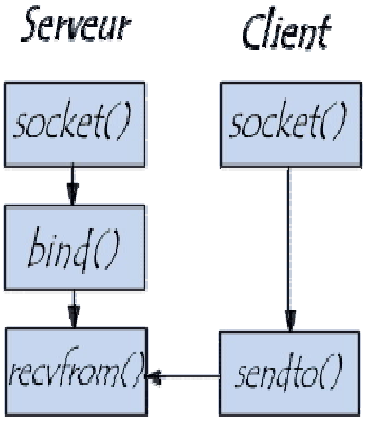


Fig.1.10 :
Mode non connecté UDP

1.8 : Les Sockets

- Les sockets permettent à deux processus de communiquer entre eux à travers d'une liaison identifiée par **une adresse IP et un numéro de port.**
- Les sockets utilisent directement les services de la couche transport du modèle OSI (protocoles UDP ou TCP), elle-même utilisant les services de la couche réseau (protocole IP).
- Socket c'est un Tuyau entre deux programmes

- Les sockets sont des portes d'entrées/sorties vers le réseau (la couche transport)
- Une socket est identifiée par une adresse de transport qui permet d'identifier les processus de l'application concernée [1]

1 .8.1 : Communication par socket

- La communication nécessite 2 sockets : 1 pour chacun des 2 programmes communicants via le réseau.
  - 1 socket pour le client.
  - 1 socket pour le serveur.

A chaque mode de communication IP utilisé : UDP ou TCP, correspond un type de socket**.**

Côté du serveur

- Le serveur utilisera deux types de sockets.
  - Le premier, appelé **socket de connexion** sert à attendre un client.
    - En Java, créer un socket de connexion peut se faire simplement en e java.net.
  - Le second, appelé **socket de communication** sert à dialoguer avec le client.
    - Une fois le socket de connexion créé, il suffit de lui demander d'attendre un client et de récupérer le socket de communication qui permettra de dialoguer avec le client.
    - Socket comm = Conn.accept();
    - On peut ensuite communiquer avec le client en utilisant les flux d'entrée et de sortie associés au socket.

Côté du client :

- Contrairement au serveur, le client n'utilise qu'un seul socket : **le socket de communication.**
- Connexion au serveur et obtention d'un socket de communication.
  - **Socket comm = new Socket ("localhost", 10080).**
  - On peut ensuite communiquer avec le serveur en utilisant les flux d'entrée et de sortie associés au socket. [2]

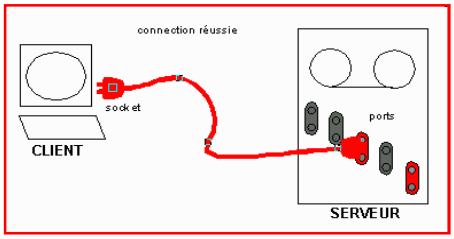


Fig. 1.12: trouve le port

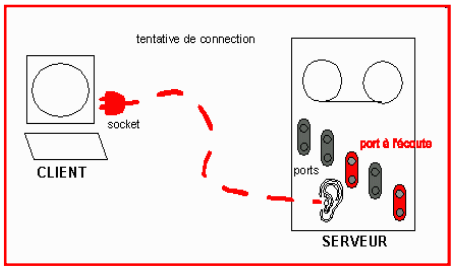


Fig. 1.11 : écouté sur le port

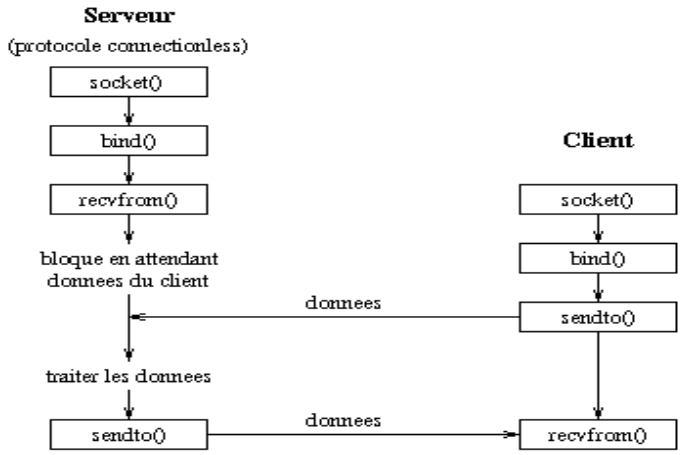


Fig. 1.13: Modèle OSI et sockets

1.9:Dialogue entre client et serveur

- **2 manières :**
  - **Bat niveau** : directement en appelant les services des couches TCP ou UDP.
    - Quand le programmeur souhaite implémenter son propre protocole applicatif de communication.
    - *Exemple :* utilisation des sockets en Java.
  - **Haut niveau** : définition de couches offrant des services plus complexes.
    - Couche réalisée en s'appuyant sur les couches TCP/UDP.
    - *Exemple :* appel d'une méthode chez une entité distante.
      - Notion de middleware (intergiciel). [2]

1.9.1 : Les middlewares

- Le middleware est une couche intermédiaire (couche logiciel) qui s'intercale entre l'infrastructure de communication d'un réseau et les éléments de l'application distribuée.
- l'ensemble des couches réseau et services logiciel qui permettent le dialogue entre les différents composent d'une application répartie.
- Middleware c'est une interface de communication universelle entre processus. Il représente véritablement la clef de voûte de toute application client/serveur.

1.9.2 : Les services des middlewares

Conversion : Services utilisé pour la communication entre machine mettant en œuvre des formats de données différentes

Adressage : Permet d'identifier la machine serveur sur laquelle est localisé le service demandé afin d'en déduire le chemin d'accès. Dans la mesure du possible.

Sécurité : Permet de garantir la confidentialité et la sécurité des données à l'aide de mécanismes d'authentification et de cryptage des informations.

Communication : Permet la transmission des messages entre les deux systèmes sans altération. Ce service doit gérer la connexion au serveur, la préparation de l'exécution des requêtes, la récupération des résultats et la déconnexion de l'utilisation.

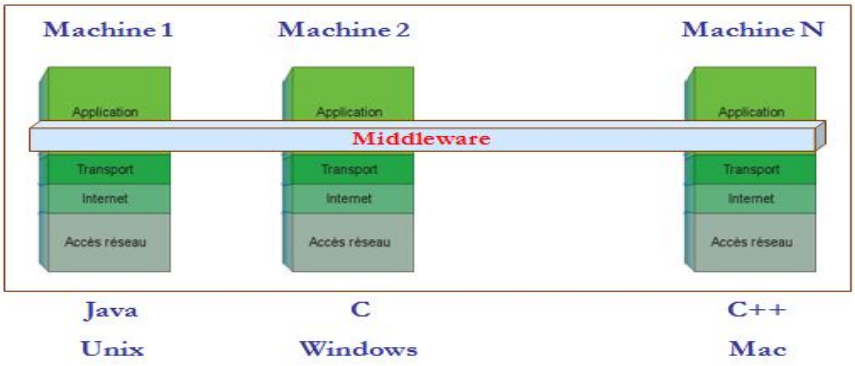


Fig. 1.14: notion de middleware

1.9.3 : Etapes d'un appel de méthode distante

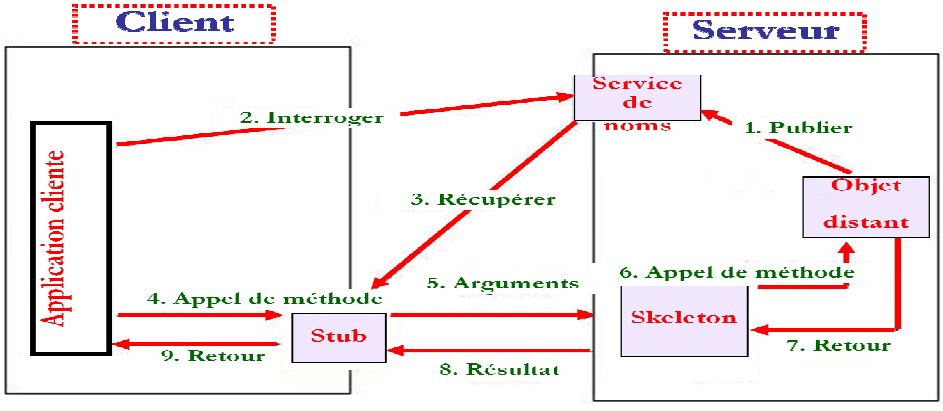


Fig1.15 : Etapes d'un appel de méthode distante

1.9.4 : Fonctions des middlewares

- Procédures d'établissement/fermeture de connexion
- Exécution des requêtes, récupération des résultats
- Initiation des processus sur différents sites
- Services de répertoire
- Accès aux données à distance
- Gestion d'accès concurrents
- Sécurité et intégrité (authentification, cryptage, …)
- Monitoring (compteurs, …)
- Terminaison de processus
- Mise en cache des résultats, des requêtes

1.9.5 : Rôles des middlewares

-négociation des connexions,

- conversion des types de données échangées,
- fiabilisation et sécurisation des échanges.
- Permet l'échange de requêtes et des réponses associées entre client et serveur de manière Transparente

1.9.6 : Les avantages des middlewares

- il offre des services de haut niveau aux applications.
- il rend portable les applications.
- il prend en charge les protocoles de conversion des caractères.
- il établit des sessions entre clients et serveurs

1.9.7 : Les objectifs de middlewares

L'objectif principal du middleware est d'unifier, pour les applications, l'accès et la manipulation de l'ensemble des services disponibles sur le réseau, afin de rendre l'utilisation de ces derniers presque transparente.

- les transports des requêtes et des réponses.
- la simplification de la session utilisateur.
- les performances et la fiabilité.

1.10. Les objectifs du client-serveur

Répartir les tâches entre le client et le serveur :

- décharger le serveur de l'exploitation des données.
- décharger la station cliente de la gestion des données.
- réduire le trafic sur le réseau. [s1]

1.11. Avantage du client /serveur

- Unicité de l'information
- Meilleure sécurité
- Meilleure fiabilité : les clients ne sont pas des ressources critiques.
- Facilité d'évolution : il est très facile de rajouter ou d'enlever des clients, et même des serveurs. [s1]

1.12. Architecture Peer to peer (P2P)

1.12.1. Introduction

Avec la croissance rapide de l'Internet, les réseaux P2P ont été largement étudiés et déployés. Selon CacheLogicResearch, le trafic P2P a dominé l'Internet en 2006, avec un pourcentage de 72% du trafic total.

Le réseau P2P est un réseau logique qui utilise un réseau physique. Le Peer to Peer est pour beaucoup de personnes synonyme de "partage de fichiers", cette idée préconçue donne une mauvaise image de ce domaine. En général, le terme P2P décrit un environnement où les machines communiquent entre elles sans l'utilisation d'un point de contrôle centralisé pour router le trafic de données.

1.12.2. Définition

Les systèmes pair à pair (P2P) sont composés d'un ensemble d'entités partageant un ensemble de ressources, et jouant à la fois le rôle de serveur et de client. Chaque nœud peut ainsi télécharger des ressources à partir d'un autre nœud, tout en fournissant des ressources à un troisième nœud.

Avec les réseaux Peer-to-Peer, les ordinateurs partagent les données et les ressources, en communiquant directement entre eux sans utiliser un serveur central.

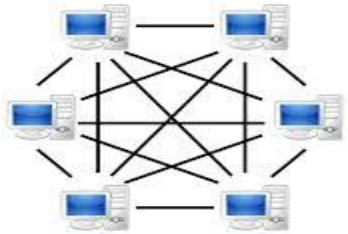


Fig. 1.16 : architecture P2P.

1.12.3 : P2P vs client/serveur

Architecture Peer to Peer	**Architecture client/serveur**
Auto-organisé	Management centralisé
Découverte des peers	Configuré
Evolution dynamique, Ad-hoc	Consultation de tables
Flux distribué (mesh)	Flux centralisé
Symétrie du réseau	Asymétrie du réseau
Communication par Messages	Orienté RPC
Adressage dynamique	Adressage statique @IP
Entités Autonomes	Entités dépendantes
Attaques difficiles (mobilité , anonymat)	Attaques plus simples

Tab1.2 : comparaison entre client/serveur et P2P

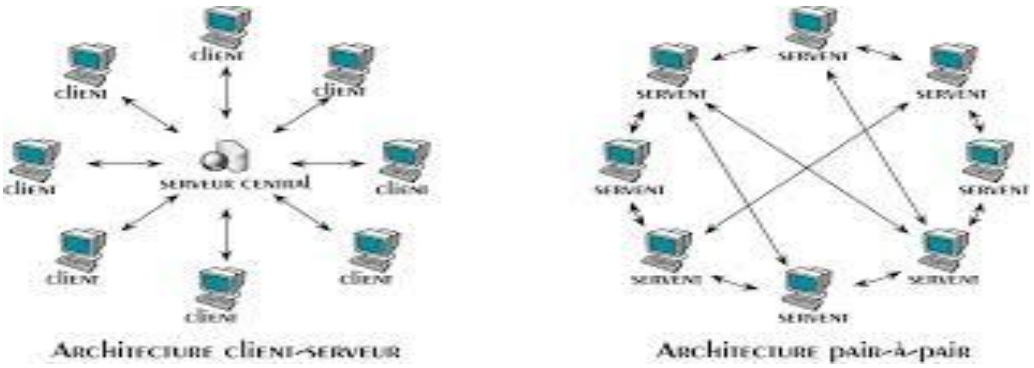


Fig. 1.17: comparaison entre client/serveur et P2P

1.13 : Conclusion

Dans ce chapitre, nous avons présentés une introduction générale du modèle client serveur et leur utilité dans le réseau informatique.et décrire les différentes notions de base de ce modèle comme le middleware, les protocoles, les sockets et l'appel de procédure à distance.

Le modèle client /serveur est la base de tous les services réseaux informatique, c'est pour cela nous sommes intéressé par l'étude de se modèle.

Chapitre 2 :

introduction aux algorithmes de chiffrement

a clé publique et a clé secrète.

2.1 : Introduction

Depuis l'Egypte ancienne, l'homme a voulu pouvoir échanger des informations de façon confidentielle.

• Il existe de nombreux domaines où ce besoin est vital :

– **militaire** (sur un champ de bataille ou bien pour protéger l'accès à l'arme atomique) ;

– **commercial** (protection de secrets industriels) ;

– **bancaire** (protection des informations liées à une transaction financière);

– **de la vie privée** (protection des relations entre les personnes) ;

– **diplomatique** (le fameux « téléphone rouge » entre Etats-Unis et Union soviétique)

Le cryptage est historiquement l'une des premières applications de l'informatique.

Ce domaine, qui était il y a encore quelques années, réservé aux militaires et aux grandes entreprises, concerne aujourd'hui tous ceux qui souhaitent transmettre des données protégées, qu'ils soient professionnels ou particuliers. Pour cela, il existe de nombreuses méthodes de cryptage, mais peu d'entre elles sont reconnues comme sûres. La méthode RSA fait depuis longtemps partie de cette catégorie. [4]

2.2 : Définition de la cryptographie

Le mot cryptographie vient des mots en grec ancien *kruptos* « caché » et *graphein* « écrire ».

Le mot **cryptographie** est un terme générique désignant l'ensemble des techniques permettant de **chiffrer** des messages, c'est-à-dire permettant de les rendre inintelligibles sans une action spécifique. Le verbe **crypter** est parfois utilisé mais on lui préfèrera le verbe *chiffrer*. [s7]

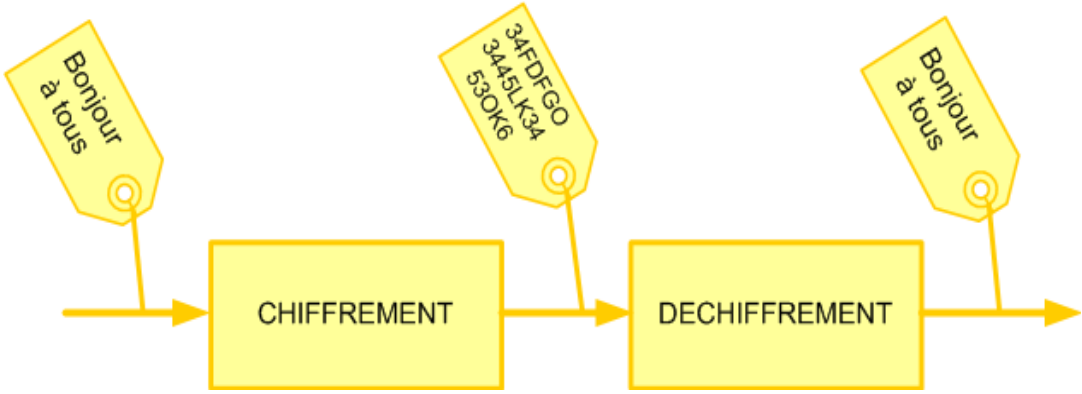


Fig2.1 : Cryptage et décryptage

2.3. Vocabulaire de base de la cryptographie

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

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

• **Chiffrement** : algorithme généralement basé sur des clefs et transformant les données. Sa sécurité est dépendante du niveau de sécurité des clefs.

•**Déchiffrer** consiste à retrouver le texte original (aussi appelé clair) d'un message chiffré dont on possède la clé de déchiffrement.

• **Signature numérique**: données ajoutées pour vérifier l'intégrité ou l'origine des données.

• **Notarisation** : utilisation d'un tiers de confiance pour assurer certains services de sécurité.

• **Contrôle d'accès** : vérifie les droits d'accès d'un acteur aux données. N'empêche pas l'exploitation d'une vulnérabilité.

Distribution de clefs : distribution sécurisée des clefs entre les entités concernées.

• **Un chiffre** concerne plutôt une technique de cryptage portant sur des éléments de taille fixe (caractères alphabétiques par exemple).

• **Un code** désigne plutôt un cryptage portant sur des éléments de taille variable (mots ou phrases)

• La possibilité de crypter repose sur la connaissance de: la clé (algorithme E, secret k)

•l'ensemble des paramètres permettant la réalisation des opérations de cryptage ou de chiffrement.

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

•**Clef :** Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations.

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

• **Crypto système :** Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné. [4]

En cryptographie, la propriété de base est que

$$M = D(E(M))$$

où

– M représente le texte clair,

– C est le texte chiffré,

– E(x) est la fonction de chiffrement.

– D(x) est la fonction de déchiffrement.

Ainsi, avec un algorithme à clef symétrique :

$$M = D(C) \text{ si } C = E(M).$$ [s5]

2.4 : La clé de chiffrement

On appelle clé une valeur utilisée dans un algorithme de cryptographie, afin de chiffrer une donnée. Il s´agit en fait d´un nombre complexe dont la taille se mesure en bits. On peut imaginer que la valeur numérique correspondant à 1024 bits est absolument gigantesque. Voir aussi Bits and bytes. Plus la clé est grande, plus elle contribue à élever la sécurité à la solution. Toutefois, c´est la combinaison d´algorithmes complexes et de clés importantes qui seront la garantie d´une solution bien sécurisée.

Les clés doivent être stockées de manière sécurisée et de manière à ce que seul leur propriétaire soit en mesure de les atteindre et de les utiliser.

2.5 : Les objectifs de la cryptographie

– **L'intégrité des données** : Le contrôle d'intégrité d'une donnée consiste à vérifier que cette donnée n'a pas été altérée, frauduleusement ou accidentellement.

– **Le contrôle d'accès** : Il s'agit d'authentifier les utilisateurs de façon à limiter l'accès aux données, serveurs et ressources aux seules personnes autorisées.

– **La confidentialité** : Il s'agit de rendre l'information inintelligible à tous les opposants tant lors de sa conservation qu'au cours de son transfert par un canal de communication.

– **L'identification**: Le contrôle d'identification consiste à s'assurer que le destinataire est bien celui qui prêtant être (authentification des partenaires) et d'obtenir une garantie que l'expéditeur a bien signé l'acte (authentification de l'origine des informations).

– **La non répudiation**: Il s'agit de garantir l'authenticité de l'acte. L'expéditeur ne peut nier le dépôt d'information, le réceptionneur ne peut nier la remise d'information, ni l'un ni l'autre ne peut nier le contenu de cette information.

– **L'authentification :** consistant à assurer que seules les personnes autorisées aient accès aux ressources. [s3]

2.6 : Mécanismes de la cryptographie

Un algorithme de cryptographie ou un chiffrement est une fonction mathématique utilisée lors du processus de cryptage et de décryptage. Cet algorithme est associé à une clef (un mot, un nombre, ou une phrase). Afin de crypter une donnée avec des clés différentes le résultat du cryptage variera également. La sécurité des données cryptées repose entièrement sur deux éléments : l'invulnérabilité de l'algorithme de cryptographie et la confidentialité de la clef.

Un système de cryptographie est constitué d'un algorithme de cryptographie, ainsi que de toutes les clefs et tous les protocoles nécessaires à son fonctionnement.

2.7 : Les familles de la cryptographie

• Il existe deux familles d'algorithmes :

1. La cryptographie symétrique
2. La cryptographie asymétrique

2.7.1 : La cryptographie symétrique (à clé sécrète)

-2.7.1 .1 : Principe

-La même clé doit être employée pour chiffrer ou déchiffrer le message.

- Le chiffrement consiste alors à effectuer une opération entre la clé privée et les données à chiffrer.

- Le déchiffrement se fait à l'aide de cette même clé secrète.

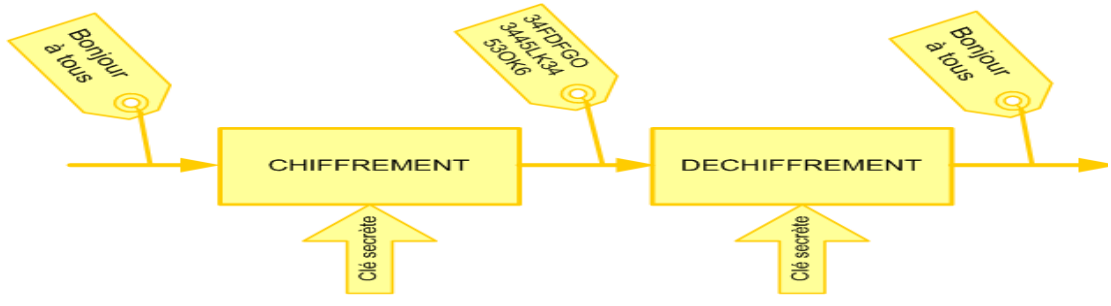


Fig2.2: La cryptographie symétrique

2.7.1 .2 : Les limites de la cryptographie Symétrique

La multiplication des clés

Pour établir un canal de communication entre deux individus :

– Il faut qu'il soit chiffré avec une clé partagée entre les deux individus.

– Il est ainsi confidentiel pour ceux qui ne possèdent pas la clé de chiffrement.

Lors d'échange entre plusieurs intervenants : une clé est partagée que par 2 interlocuteurs, donc pour N interlocuteurs il faut N*(N-1)/2 clés. [4]

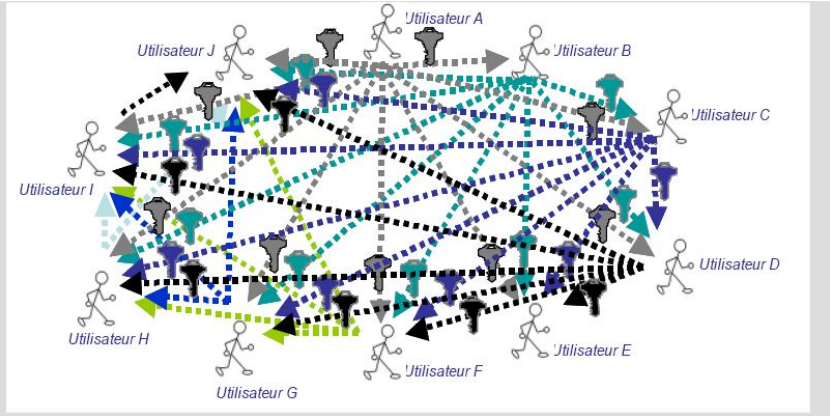


Fig2.3 : Les limites du chiffrement

- Pas d'intégrité et d'identification de l'auteur

-Si Alice, Bob et Cédric partage le même lien de communication alors ils partagent la même clé de chiffrement symétrique.

-Chacun peut intercepter et modifier les messages qui s'échangent. [4]

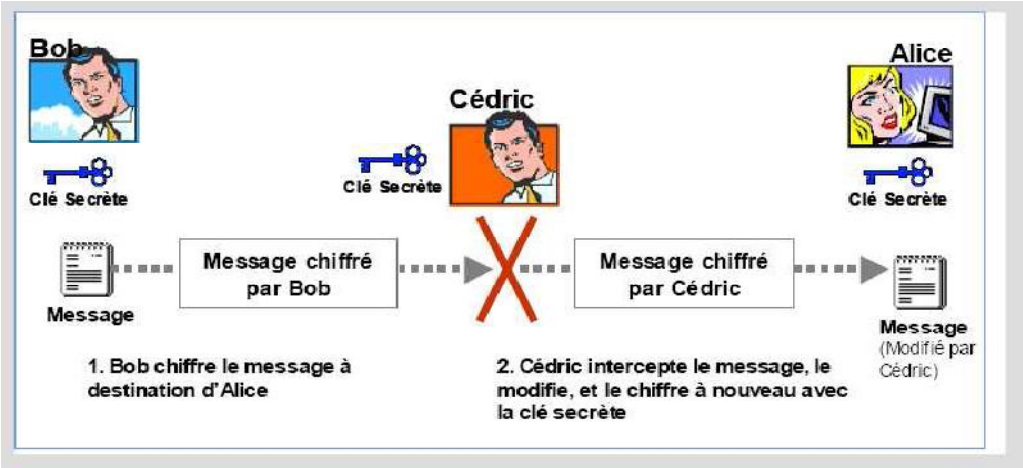


Fig2.4: Les limites du chiffrement Symétrique

2.7.2 : La cryptographie asymétrique (à clé publique)

Les problèmes de distribution des clés sont résolus par la *cryptographie de clé publique*. Ce concept a été introduit par Whitfield Diffie et Martin Hellman en 1975.

La cryptographie de clé publique est un procédé asymétrique utilisant une *paire* de clés pour le cryptage : une clé publique qui crypte des données et une clé privée ou secrète correspondante pour le décryptage.

Vous pouvez ainsi publier votre clé publique tout en conservant votre clé privée secrète. Tout utilisateur possédant une copie de votre clé publique peut ensuite crypter des informations que vous êtes le seul à pourvoir lire. Même les personnes que vous ne connaissez pas personnellement peuvent utiliser votre clé publique.

D'un point de vue informatique, il est impossible de deviner la clé privée à partir de la clé publique. Tout utilisateur possédant une clé publique peut crypter des informations, mais est dans l'impossibilité de les décrypter. Seule la personne disposant de la clé privée correspondante peut les décrypter.

On peut classer l'utilisation des algorithmes à clé publique en 3 catégories :

– **Chiffrement/déchiffrement** : cela fournit le secret.

– **Signatures numériques** : cela fournit l'authentification.

– **Échange de clés** (ou des clefs de session).

La sécurité de tels systèmes repose sur des problèmes calculatoires :

– RSA : factorisation de grands entiers

– ElGamal : logarithme discret

– Merkle-Hellman : problème du sac à dos (knapsacks)

– ...

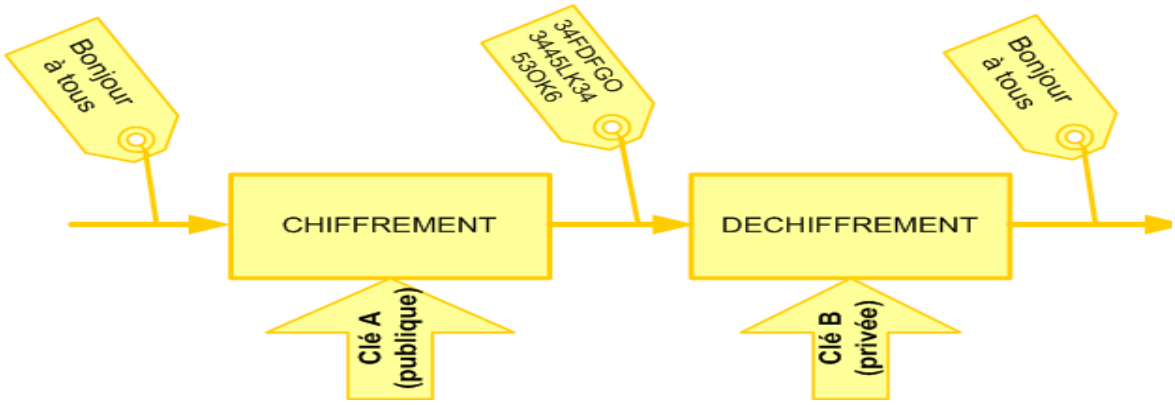


Fig2.5: La cryptographie asymétrique

2.7.2.1. Rapports entre les clés

La recherche de la clé privée à partir de la clé publique revient à résoudre un problème mathématique notoirement très compliqué, c.-à-d. demandant un grand nombre d'opérations et beaucoup de mémoire pour effectuer les calculs ---> infaisable !

• Par exemple dans RSA, l'algorithme le plus utilisé actuellement, la déduction de la clé rivée à partir de la clé publique revient à résoudre un problème de factorisation de grand nombre que lequel travaille les mathématiciens depuis plus de 2000 ans !

• Le choix des clés doit être fait de la manière la plus imprédictible possible :

Éviter les mots du dictionnaire, nombres pseudo-aléatoires à germe de génération difficile à deviner, etc…

• Il repose sur la connaissance d'une fonction mathématique unidirectionnelle

• Une fonction unidirectionnelle est une fonction y = f(x) telle que, si l'on connaît la valeur y, il est pratiquement impossible de calculer la valeur x (c'est-à-dire d'inverser la fonction f).

• On dit que cette fonction est munie d'une porte arrière s'il existe une fonction x = g(y, z) telle que, si l'on connaît z, il est facile de calculer x à partir de y. Z est appelée trappe. [4]

2.7.2.2: Chiffrement d'un message

Lorsqu'un utilisateur désire envoyer un message à un autre utilisateur, il lui suffit de chiffrer le message à envoyer au moyen de la clé publique du destinataire (qu'il trouvera par exemple dans un serveur de clés tel qu'un annuaire ou bien en signature d'un courrier électronique).

Le destinataire sera en mesure de déchiffrer le message à l'aide de sa clé privée

(qu'il est seul à connaître). [4]

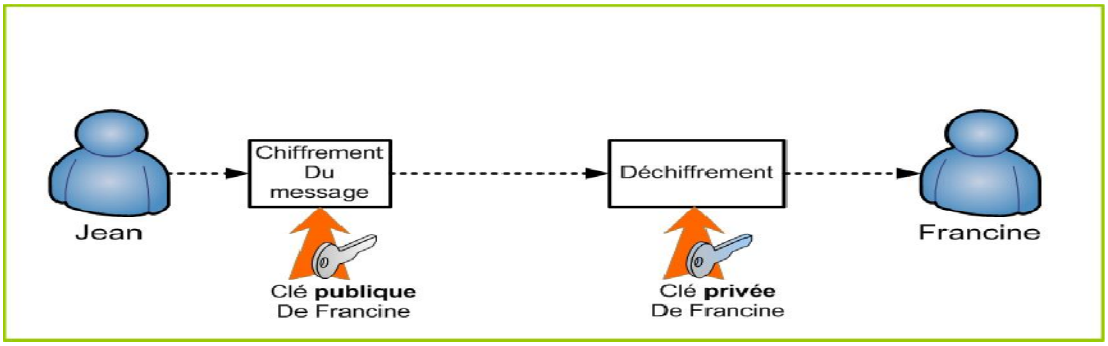


Fig.2.6 : Principe de crypto asymétrique

2.8 : Domaine d'application de la cryptographie

1- Les cartes bancaires

2- Les navigateurs, ou browsers, tels que Mozilla Firefox ou Internet Explorer, utilisent le protocole de sécurité SSL (Secure Sockets Layers), qui repose sur un procédé de cryptographie par clé publique

3- Secret militaire

4- Communications numériques

5- Droits d'auteurs

2.9 : Différence entre symétrique et asymétrique

Attribut	**Clé secrète**	**Clé publique/privé**
Année d'existence	Des milliers	Moins de 50
Utilisation actuelle principale	Chiffrement des données en gros	échange des clés, signature numérique
Standard actuel	DES, AES, Triple DES	RSA, Diffie-Hellman
Vitesse de chiffrement/déchiffrement	Rapide	Lent
Clé	Secret partagé par au moins deux personnes	**Privée :** gardée cachée par une personne**.** **Publique :** largement distribuée.
Echange de clé	Transfert risqué et difficile pour une clé secrète	Simple, moins risqué de remettre une clé publique jamais partagée
Longueur de la clé	56 bits	1024 bits
Confidentialité, authentification	Oui	Oui
Intégrité du message	Non	Oui
Non-répudiation	Besoin d'un tiers de confiance servant de témoin	Signature numérique : pas besoin de tiers
Attaque	Oui	Oui
	Très facile	Difficile par rapport au cryptage symétrique

Tab 2.1 : symétrique contre asymétrique

2.10 : Avantages chiffrement asymétrique

- l'élimination de la problématique de la transmission de clé
- la possibilité d'utiliser la signature électronique
- l'impossibilité de décrypter le message dans le cas de son interception par une personne non autorisée.

-Le problème consistant à se communiquer la clé de déchiffrement n'existe plus, dans la mesure où les clés publiques peuvent être envoyées librement. Le chiffrement par clés publiques permet donc à des personnes d'échanger des messages chiffrés sans pour autant posséder de secret en commun, seule la clé secrète à besoin d'être conservée de manière secrète.

- Selon l'usage, une paire de clé (publique/secrète) peut être utilisée plus longtemps qu'une clé symétrique.
- La cryptographie à clé publique permet de réaliser des schémas de signature électronique assurant un service de non répudiation.
- Dans un grand réseau, le nombre de clés est beaucoup plus petit que dans un système symétrique car seulement 2n clés sont nécessaires s'il y a n utilisateurs dans le réseau.

2.11 : Inconvénients chiffrement asymétrique

- le temps d'exécution : plus lent que le cryptage symétrique
- le danger des attaques par substitution des clés (d'où la nécessité de valider les émetteurs des clés)

-Tout le challenge consiste à s'assurer que la clé publique que l'on récupère est bien celle de la personne à qui l'on souhaite faire parvenir l'information chiffrée.

-Les performances des systèmes asymétriques sont beaucoup moins bonnes que celles des systèmes symétriques car ces systèmes nécessitent de pouvoir calculer sur des grands nombres.

-La taille des clés est généralement plus grande pour ces systèmes que pour les systèmes à clé secrète

2.12 : Classement des attaques de cryptanalyse

On distingue plusieurs types d'attaques suivant les informations que peut obtenir le cryptanalyste, Ce sont :

***L'attaque à message chiffré seulement**, où le cryptanalyste ne connaît qu'un ensemble de messages chiffrés ; il peut soit retrouver seulement les messages en clair, soit retrouver la clef. En pratique, il est très souvent possible de deviner certaines propriétés du message en clair (format ASCII, présence d'un mot particulier,...), ce qui permet de valider ou non le décryptement.

***L'attaque à message en clair connu**, où le cryptanalyste connaît non seulement les messages chiffrés, mais aussi les messages en clair correspondants ; son but est alors de retrouver la clef. Du fait de la présence, dans la plupart des messages chiffrés, de parties connues (en-têtes de paquets, champs communs à tous les fichiers d'un type donné,...), ce type d'attaques est très courant.

***L'attaque à message en clair choisi**, où le cryptanalyste peut, de plus, choisir des messages en clair à chiffrer et donc utiliser des messages apportant plus d'informations sur la clef. Si le cryptanalyste peut de plus adapter ses choix en fonction des messages chiffrés précédents, on parle d'**attaque adaptative**.

***L'attaque à message chiffré choisi**, qui est l'inverse de la précédente : le cryptanalyste peut choisir des messages chiffrés pour lesquels il connaîtra le message en clair correspondant ; sa tâche est alors de retrouver la clef. Ce type d'attaques est principalement utilisé contre les systèmes à clef publique, pour retrouver la clef privée. *[s4]*

2.13 : Algorithme Diffie-Hellman

2.13.1 : Introduction

En cryptographie, l'échange de clés **Diffie-Hellman**, du nom de ses auteurs Whitfield Diffie et Martin Hellman. C'est un algorithme d'échange de clefs : il permet à deux personnes de mettre en place un système par lequel elles pourront échanger de façon secrète. C'est le premier algorithme créé dans ce but, en 1976, par Whitfield Diffie et Martin E. Hellman dans leur article New Directions in Cryptography.

- Le protocole D-H permet à deux tiers de générer un secret partagé sans avoir aucune information préalable l'un sur l'autre. Il est basé sur la cryptologie à clef publique (dont il est d'ailleurs à l'origine), car il fait intervenir des valeurs publiques et des valeurs privées.

- Sa sécurité dépend de la difficulté de calculer des logarithmes discrets sur un corps fini.
- Le secret généré à l'aide de cet algorithme peut ensuite être utilisé pour dériver une ou plusieurs clefs (clef secrète, clef de chiffrement de clefs...).

2.13.2 : Principe

 Alice et Bob ont choisi un groupe fini (soit un corps fini, dont ils n'utilisent que la multiplication, soit une courbe elliptique) et un générateur g de ce groupe.

 Alice choisit un nombre au hasard a, élève g à la puissance a, et dit à Bob g^a .

 Bob fait de même avec le nombre b ;

 Alice, en élevant le nombre reçu de Bob à la puissance a, obtient g^{ba} (toujours calculé modulo p par exemple).

 Bob fait le calcul analogue et obtient g^{ab}, qui est le même, mais puisqu'il est difficile d'inverser l'exponentiation dans un corps fini (ou sur une courbe elliptique), c'est-à-dire de calculer le logarithme discret, Ève ne peut pas découvrir, donc ne peut pas calculer g^{ab} [mod p] ; [s6]

 Alice et Bob connaissent donc tous les deux le nombre g^{ab} [mod p] dont Ève n'a pas connaissance

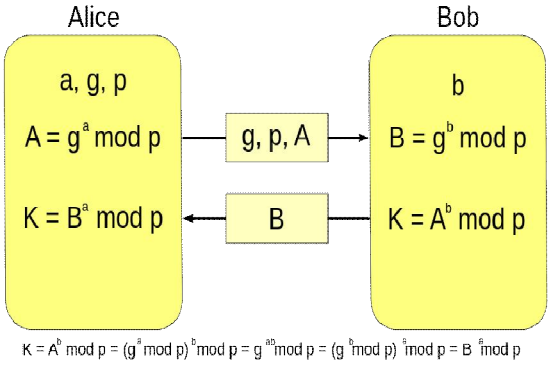


Fig2.7 : Principe d'un échange de clés Diffie-Hellman

2.13.3 : Exemple :

1. Alice et Bob choisissent un nombre premier p et une base g. Dans notre exemple, p=23 et g=3
2. Alice choisit un nombre secret a=6
3. Elle envoie à Bob la valeur $A = g^a$ [mod p] = 3^6 [23] = 16

4. Bob choisit à son tour un nombre secret b=15
5. Bob envoie à Alice la valeur $B = g^b$ [mod p] = 3^{15} [23] = 12
6. Alice peut maintenant calculer la clé secrète : $(B)^a$ *[mod* p*]* = 12^6 *[23] = 9*
7. Bob fait de même et obtient la même clé qu'Alice : $(A)^b$ *[mod* p*]* = 16^{15} *[23] = 9*

2.13.4 : Inconvénients

-Quand Alice veut transmettre un message chiffré à Bernard, elle doit d'abord entrer en contact avec lui afin de fixer la clé lors d'un premier échange ; chaque transmission de message s'en trouve ainsi fortement ralentie et perd toute instantanéité (sans parler de l'augmentation du nombre de communications engendrée par le système).

-Une tierce personne ne peut intervenir lors du processus d'échange des clés publiques en les remplaçant par des clés de son choix ; elle sera alors en mesure de déchiffrer les messages transmis sans que les correspondants ne s'en aperçoivent.

Lors de l'échange des clés, l'intermédiaire choisit sa propre clé c et calcule $\gamma = g^c \bmod p$. Il remplace a et b cette valeur. [s6]

2.14 : Algorithme RSA (Rivest,Shamir,Adleman)

2.14.1 : Introduction

Le chiffrement RSA (nommé par les initiales de ses trois inventeurs) est un algorithme de cryptographie asymétrique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman.

Procédures de chiffrement et de déchiffrement à 2 clés :

-Une clé **publique :** révélée de tous les correspondants

-Une clé **privée** : gardée secrète par son détenteur.

2 .14.2 : Création des clés

Le RSA fonctionne à partir de deux nombres premiers, que l'on appellera **p** et **q**. Ces deux nombres doivent être très grands, car ils sont la clé de voûte de notre cryptage.

Aujourd'hui, on utilise des clés de 128 à 1024 bits, ce qui représente des nombres décimaux allant de 38 à 308 chiffres !

Choisir deux nombres p et q de l'ordre de 10100

1-on fixe n = p*q et on publie n

Il est difficile de retrouver p et q à partir de n

2-Calculer f (n) = (p - 1)(q - 1)

3-Choisir e (Kc) tel que pgcd(e, f (n)) = 1 et on le publie e < f (n) et e = Kc

4-Calculer d (Kd) tel que d.e mod f (n) d = Kd

Le couple **(e, n)** est la clé publique, et **(d, n)** est la clé privée. [s6]

2 .14.3 : Cryptage et décryptage

Chiffrement :

1. Le message en clair : M (0 < M < n)

2. Le message chiffré C est obtenu en calculant: [5]

$$C = M^{Kc} \bmod n$$

Déchiffrement :

1. Le message chiffré: C

2. Le message en clair M sera obtenu en calculant:[5]

$$CKd \bmod n = M^{Kc*Kd} \bmod n = M$$

2.14.4 : Exemple

Prenons 2 nombres premiers au hasard: **p = 29**, **q = 37**

1-On calcul **n = pq = 29 * 37 = 1073**

2-On doit choisir **e** au hasard tel que **e** n'ai aucun facteur en commun avec **(p-1)(q-1)**:

(p-1)(q-1) = (29-1)(37-1) = 1008

On prend **e = 71**

3-On choisit **d** tel que **71*d mod 1008 = 1**

On trouve **d = 1079**

On a maintenant nos clés :

- La clé publique est **(e,n) = (71,1073)** (=clé d'encryptage)
- La clé privée est **(d,n) = (1079,1073)** (=clé de décryptage)

On va encrypter le message '**HELLO**'. On va prendre le code ASCII de chaque caractère et on les met bout à bout:

m = 7269767679

Ensuite, il faut découper le message en blocs qui comportent moins de chiffres que **n**. **n** comporte 4 chiffres, on va donc découper notre message en blocs de 3 chiffres:

726 976 767 900

(on complète avec des zéros)

Ensuite on encrypte chacun de ces blocs:

726^71 mod 1073 = 436

976^71 mod 1073 = 822

767^71 mod 1073 = 825

900^71 mod 1073 = 552

Le message encrypté est **436 822 825 552**. On peut le décrypter avec d:

436^1079 mod 1073 = 726

822^1079 mod 1073 = 976

825^1079 mod 1073 = 767

552^1079 mod 1073 = 900

C'est à dire la suite de chiffre **726976767900**.

On retrouve notre message en clair **72 69 76 76 79** : '**HELLO**'.

2.15 : Conclusion

Dans ce chapitre, nous avons présenté une introduction générale sur la cryptographie.

Nous avons distingué deux classes importantes des algorithmes de chiffrement, c'est le cryptage symétrique à clé secrète et le cryptage asymétrique a clé publique. Nous avons aussi montré La différence entre qui existe entre les deux classes.

Dans la suite de ce chapitre, nous présentons les différents formats des fichiers audio.

Chapitre 3 :

présentation des Formats des fichiers audio.

3.1 Introduction

Le son est une vibration de l'air, c'est-à-dire une suite de surpressions et de dépressions de l'air par rapport à une moyenne, qui est la pression atmosphérique.

Le son est une chose familière dans notre vie quotidienne. Le son, c'est donc ce que l'oreille perçoit de la vibration d'un corps. Généralement il se propage sous la forme d'une onde dans l'air jusqu'à notre oreille, mais il se transmet aussi dans les liquides et dans les corps solides. Cet ébranlement de la matière se caractérise par une variation de pression se propageant de proche en proche. Plus la pression acoustique est grande, plus le volume sonore est important.

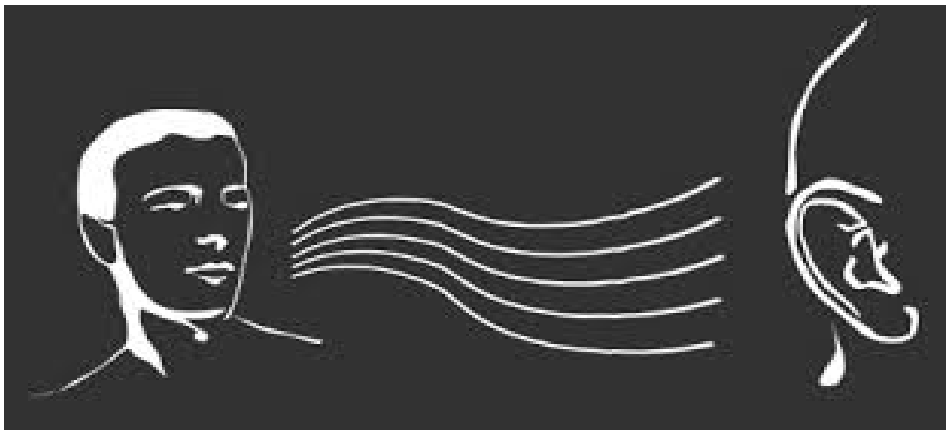


Fig. 3.1 : Émission réception de son

3.2: Historique

On admet en général qu'Edison fut le premier à reproduire un son enregistré, avec l'invention du phonographe en 1877. Le premier enregistrement sonore conservé a lui eu lieu en 1860 et on le doit à un savant français, Édouard-Léon Scott de Martin ville.

Scott inventa le *phonautographe*, un appareil permettant de transcrire des ondes sonores sur un rouleau de papier enduit de noir de fumée. La particularité de l'appareil est qu'il permettait d'enregistrer les sons mais pas de les reproduire. Scott dépose un brevet en 1857 pour le phonautographe. [s10]

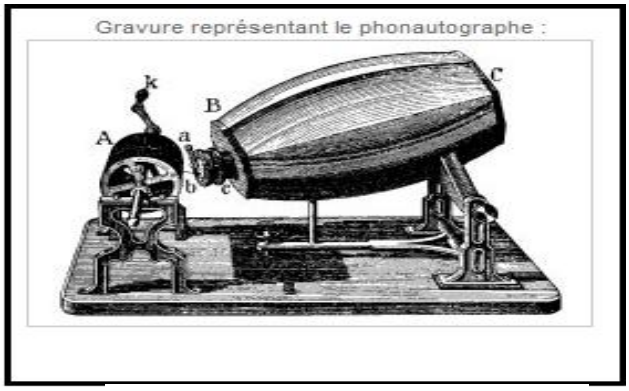


Fig. 3.2: phonautographe

3.3 : Définition

Un format de fichier audio est un format de données utilisé en informatique pour stocker des sons, (de la musique, des voix, etc.) sous forme numérique. L'industrie a produit de nombreux formats destinés à une application principale ou exclusive, soit à la production, soit à la conservation, soit à la diffusion.

Les fichiers WAV sont devenus un standard dans le monde du PC. Il est donc primordial de savoir travailler avec ce type de fichier si l'on veut pouvoir transformer le son

3.4 : Formats audio

On recense environ **50 formats audio** différents dans le monde, plus ou moins utilisés.

Aujourd'hui, **98%** des fichiers audio sont de l'un des **6 principaux formats** :

3.4.1 : RIFF

Plusieurs formats audio étant basés sur la norme RIFF (Resource Inter change File Format), format d'échange de fichiers ressources, il convient de le décrire en premier, bien qu'il s'agisse d'une norme d'usage général. Le principe est de définir des *conteneurs* et des *bouts*, les uns et les autres identifiés par leurs quatre premiers octets, qui sont, généralement, des caractères ASCII, de façon à pouvoir être lus par les techniciens qui ouvrent le fichier avec un éditeur hexadécimal, suivi par quatre octets qui indiquent où se trouve la fin du conteneur (ce nombre d'octets limite la taille d'un fichier RIFF à 4 Go).

Un fichier RIFF commence par « RIFF » et la longueur totale; puis on trouve l'identifiant du premier sous-conteneur, par exemple « INFO » suivi de la longueur de ce bout, à la fin duquel commence le suivant, par exemple « WAVE » suivi de la longueur de ce bout.

3.4.2 : WAV

Le format WAV (ou WAVE), (« WAVEform audio format ») est un conteneur basé sur le format de fichier3 RIFF, dans lequel son identifiant est « WAVE ». Il peut contenir des codages audio avec réduction de débit ou non, mono ou stéréo, il a été mis au point par Microsoft et IBM. Les informations nécessaires au décodage se trouvent au début du fichier. Le WAV permet de stocker des métadonnées dans le fichier.

Le plus souvent, il contient de l'audio sans réduction de données, avec des fréquences d'échantillonnage et des résolutions variées.

Le suffixe des fichiers créés est **.wav**

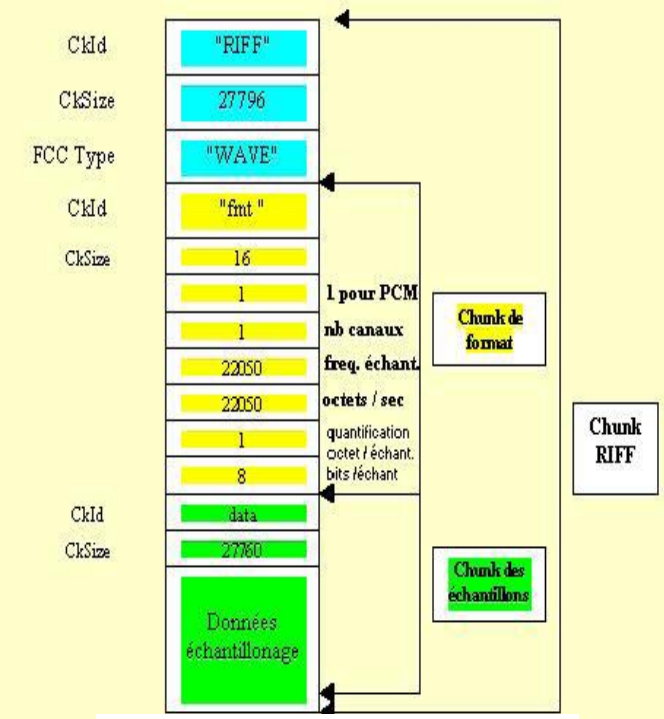


Fig.3.3: principe Wav et Riff

3.4.3 : BWF

Le BWF (*Broadcast Wave Format*) est un format audio standard créé par l'Européen à partir du WAV à l'usage des professionnels. Les Fichiers BWF incluent une référence standardisée Timestamp qui permet et facilite la synchronisation avec un élément d'image distinct. C'est le format d'enregistrement usuel de nombreuses stations de travail audio professionnelles de la télévision et du cinéma.

Stand-alone, basé sur des fichiers, multi-enregistreurs de Sound Devices, Zaxcom, HHB USA, (en) en:Fostex, et Aaton tous utilisent BWF comme leur format préféré.

3.4.4 : Ogg

Le format Ogg Vorbis est un format libre, fruit de la fondation Xiph.org. Ogg est un conteneur qui peut contenir des pistes sonores audio sans perte (FLAC), codées avec le codec psycho acoustique Vorbis, audio parlées (Speex) et vidéo (Theora). Un 'fichier Ogg' peut donc contenir l'un ou l'autre (ou une combinaison) de pistes

3.4.5 : AIFF

L'AIFF est un format de stockage de sons sur les ordinateurs de Apple. C'est l'équivalent du format WAV dans le monde Windows.

Les résolutions 8, 16, 20, 24 et 32 bits (à virgule flottante) sont acceptées.

Le suffixe des fichiers créés est **.aif**

Une variante l'AIFF-C permet de compresser la taille jusqu'à 6x.

3.4.6 : CAF

Le CAF (*Corê audio format*) a été développé par Apple pour s'affranchir des limitations de conteneur audio plus ancien comme le AIFF ou le WAV.

Il est compatible avec le système Mac OS X d'Apple depuis la version 10.3 et est lisible par QuickTime 7 **[s11]**

3.5. Les formats multipistes

Les formats multipistes sont une innovation récente. Ils consistent à encapsuler dans un fichier différent pistes sonores, qui peuvent alors être combinées par l'utilisateur dans les proportions qui lui conviennent. L'idée est de proposer, pour un morceau de musique, la piste correspondant à chaque instrument (et la voix) de manière séparée. L'utilisateur peut alors créer sa propre version.

Le U-MYX avait été utilisé pour inclure des parties bonus dans les albums.

Ces formats ont l'inconvénient de nécessiter un ordinateur pour être utilisés. Cependant l'apparition de Smartphones faisant office de baladeurs et capables d'exécuter des applications indépendantes permet à ces formats de devenir transportables, y compris pour la lecture avec mixage.

3.5.1. iKlax

Le format iKlax est un format numérique d'écoute active développé par la société iKlax Media et le LaBRI. Ce format de fichier de type conteneur permet d'organiser la musique en différents groupes et de leur appliquer des contraintes.

Le format a bénéficié d'une parution scientifique lors de l'ICMC 2008 (Internation Computer Music Conférence) à Belfast.

3.5.2.U-MYX

Le U-MYX est un format multipiste lisible avec une application dédiée, éditée par la même entreprise qui a développé le format. Ce format a été utilisé pour fournir des morceaux en version multipiste à titre de bonus dans les albums, l'application étant fournie en même temps que les morceaux, le tout sur une session de CD visible uniquement si on lisait le disque sur un ordinateur.

Malgré ces exemples d'utilisation, le futur de U-MYX est incertain, car l'entreprise qui le produit est assez floue sur ses projets futurs.

3.5.3.MXP4

Le MXP4 est un format multipiste dans lequel les pistes encapsulées sont au format Ogg. Produit par une société française, le format a bénéficié d'une forte médiatisation dans la presse française, présenté comme un « successeur du MP3 ». Une expression qui peut sembler exagérée, les formats multipistes jouant tout simplement un rôle différent [6]

3.6. Les formats audio compressés avec perte

- **AC3**

Officialisée en 1992, la compression AC3 permet d'utiliser jusqu'à 6 canaux sonores indépendants avec un taux d'échantillonnage de 32, 44,1 ou 48 kHz et avec un taux de transfert allant de 32 à 640 kbit/s. Le Dolby Digital utilise ce principe de codage, c'est pourquoi on le désigne souvent sous ce nom. Format très courant dans les DVD.

- **MP3**

MP3 est l'abréviation de MPEG-1/2 Audio Layer 3. Cet algorithme de compression prend naissance en 1987. L'ISO en fera un standard dans les années 92-93. La couche (*Layer*) III est la couche la plus complexe. Elle est dédiée à des applications nécessitant des débits faibles (128 kbit/s) d'où une adhésion très rapide du monde Internet à ce format de compression. Les taux de compression (*ratio*) sont d'ordinaire de 1 pour 10 (1:10) (1:4 à 1:12). Très rapide à

l'encodage. Des royalties importantes sont à payer pour exploiter la licence MP3. Utiliser l'encodeur MP3 LAME dernière version, encodé à 130 kbit/s (V5) permet d'obtenir une *qualité comparable* au AAC (Advanced Audio Coding) encodé à 48 kbit/s.

Le suffixe des fichiers créés est **.mp3**

Type de compression : constant ou variable (VBR).

- **mp3PRO**

Le format mp3PRO, fruit de la collaboration entre Thomson Multimédia et l'Institut Fraunhofer, combine l'algorithme MP3 et un système améliorant la qualité des fichiers comprimés appelé **(en)**SBR pour *Spectral Bandwidth Replication*.

Ce format a été publié à la fin de 2001; un fichier MP3pro 64 kbit/s a une qualité équivalente à celle d'un MP3 à 128 kbit/s.

Le suffixe des fichiers créés est **.mp3 [6]**

3.7 : Utilisation des formats

Un format est dit :

- De **production** quand le signal peut être restitué après décodage identique à ce qu'il était avant le codage en dehors des métadonnées encapsulées.
- De **diffusion** quand le codage vise à restituer l'impression sonore, sans garantir la restitution du signal à l'identique, en général au bénéfice d'une réduction du débit des données, ce qui rend incertaine la qualité d'un traitement ultérieur.

3.8. Caractéristiques des audio

- **Débit numérique** : taille du fichier par rapport à la durée du son.

  La taille L d'un fichier audio non compressé est donnée par la formule suivante :

  L=F*T*Q*P

  où **F** est la fréquence d'échantillonnage, **T** le temps d'enregistrement, **Q** la quantification et **P** le nombre de pistes (mono, stéréo, … etc.). Il faut diviser la taille par **8** pour convertir en octets. Résolution de chaque échantillon en bits. Le rapport signal sur bruit dépend étroitement de cette caractéristique.

- ***Fréquence d'échantillonnage*** : nombre d'échantillons par seconde utilisés pour décrire numériquement le signal qui représente l'onde sonore pour chaque canal. La bande passante dépend étroitement de cette caractéristique.
- **Puissance de calcul nécessaire au codage.**
- **Puissance de calcul nécessaire au décodage**.
- **Nombre de canaux sonores codés** : mono, stéréo, multicanaux.
- **compression de données** ou réduction de débit
- **Structure** permettant ou non
    - de commencer à jouer le fichier alors qu'on en connaît pas encore la fin,
    - de jouer un fichier à partir du milieu sans connaître le début,
    - de sauter sur un emplacement déterminé,
    - d'enregistrer des données ancillaires et auxiliaires (Métadonnées),
    - de gérer les droits de reproduction numérique (DRM),
    - d'adapter automatiquement le niveau au local d'écoute.
    - l'impression sonore (codage psycho acoustique).

Selon l'usage auquel le fichier est destiné, certaines caractéristiques ont plus d'importance que d'autres**.** [7]

3.9. Caractéristique de fichier wav

3.9.1. L'amplitude

Un sample est composé d'une courbe continue ayant une valeur bi-polaire (signée) . Le 1er élément d'un son est l'amplitude: C'est le point le plus élevé (et le plus bas) de la courbe. Plus l'amplitude est élevée, plus le son est fort, bruyant. L'unité de grandeur de l'amplitude est le décibel (dB). "C'est une mesure logarithmique donnant le degré d'amplification d'une vibration." Nous n'irons pas plus loin dans la description du décibel, puisque ce n'est pas indispensable dans la programmation. L'amplitude est digitalisée avec l'ADC de la carte son. Par exemple, en 8 bits, l'amplitude possède une résolution de 256 valeurs. En 16 bits, 65536 valeurs, etc... Il existe aussi le 24 et 32 bits. Pour le moment, toutefois, SoudEditor ne supporte pas les données 24 bits qui sont très embêtantes à manipuler et qui sont très rarement utilisées. Plus la résolution est élevée, plus l'échantillon est proche du son original. Dans la

figure 4.1, l'amplitude digitalisée est illustrée en vert. En 8 bits, la valeur de l'amplitude est non signée, et en 16 bits, l'amplitude est signée.

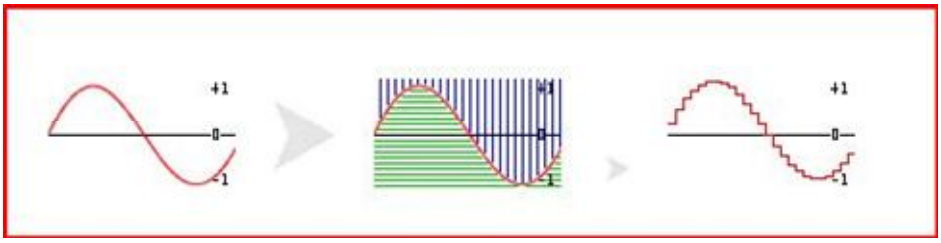


Fig.3.4: Amplitude, fréquence

3.9.2. La fréquence

En bleu (figure 3.4), c'est la fréquence d'échantillonnage, le nombre de valeurs définissant l'amplitude pour une seconde d'enregistrement. Ainsi 44100 Hz signifie 44100 échantillons pour une seconde de son mémorisé. Plus la fréquence d'échantillonnage est élevée, meilleure est la qualité du sample, plus les données digitales sont proches de l'original.

3.9.3. Le débit

On peut calculer le "débit" (ko/s) d'un sample avec ces paramètres: L'amplitude (format 8 ou 16 bits), le mode (mono ou stéréo) et la fréquence. Par exemple, en 8 bits mono 44100Hz, cela nous donne pour une seconde d'enregistrement: 44100 octets (1 octet=8 bits). En stéréo, c'est le double. En 16 bits stéréo, c'est le quadruple. Ainsi, avec la qualité du CD audio (16 bits stéréo 44,1 kHz), on obtient 176400 octets par seconde. A partir de la taille du fichier (donnée par file Size), et les caractéristiques du sample, il est facile de calculer le temps total en secondes (ou ms) de lecture d'un sample.

3.9.4. L'ordre des données

Après les 44 octets de l'en-tête, viennent les données. Les données ont un ordre bien défini. Dans le cas d'un sample **8 bits mono,** c'est **1** seul octet par sample, donc les données se suivent normalement. Pour un sample **8 bits stéréo** , ce sont **2** octets par sample, l'octet de la voie de gauche, puis l'octet de la voie de droite: L,R, L,R, L,R, etc... Dans le cas d'un sample **16 bits mono** , ce sont **2** octets par sample, l'octet de poids faible, puis l'octet de poids fort. Par exemple pour une donnée qui vaudrait 15000, nous aurions: $98 $3A (les octets sont toujours inversés). Dans le cas d'un sample **16 bits stéréo** , ce sont **4** octets par sample; Deux

octets pour la voie de gauche, et deux pour la voie de droite: L,L,R,R, L,L,R,R, L,L,R,R, etc...
Le format des données : Le sample **8 bits** (mono ou stéréo) possède des données non-signées, c'est-à-dire que le point (l'amplitude) le plus bas vaut **zéro,** le point du milieu vaut **127** , et le point le plus haut vaut **255** .
Pour pouvoir travailler sur ces données 8 bits, (par exemple pour modifier le volume du sample, ou le mixer avec un autre, etc...) les données 8 bits doivent subir une petite manipulation afin d'être présentées de la même façon que les données 16 ou 32 bits. [8]

Nom	Nombre d'octets	Type	Endian	Description
Bloc de déclaration d'un fichier au format WAVE				
Chunk ID	4	4 char	big	4 caractères constants RIFF : 0x52494646
Chunk Size	4	int 32	little	Taille du fichier - 8 octets des deux premiers blocs
Format	4	4 char	big	4 caractères constants WAVE : codé 0x57415645
Bloc décrivant le format audio				
SubChunk 1 ID	4	4 char	big	4 caractères constants "fmt " : codé 0x666d7420
SubChunk 1 Size	4	int 32	little	Taille du sous-bloc "SubCHunk 1" - 8 octets, vaut 16 pour le PCM
Audio Format	2	int 16	little	Type de compression audio, 1 pour PCM
Num Channels	2	int 16	little	Nombre de canaux (1 = Mono, 2 = Stéréo, ...)
Sample Rate (Frequence)	4	int 32	little	Fréquence d'échantillonnage (nombre d'échantillons par seconde)
Byte Rate (BytePerSec)	4	int 32	little	Nombre d'octets par seconde = Frequence * BitsPerSample/8 * NumChannels
Block Align (BytePerSample)	2	int 16	little	Nombre d'octets par échantillon (tous canaux confondus) = Nombre de canaux * Nombre d'octets par échantillons
Bits per Sample	2	int 16	little	Nombre de bits par échantillon
Bloc des données				
SubChunk 2 ID	4	4 char	big	4 caractères constants : "data" : 0x64617461
SubChunk 2 Size	4	int 32	little	Taille des données = NbCannaux * NbSamples * BitsPerSample/8 = taille du fichier - 44 octets (taille de l'en-tête, rappelez-vous-en !)
Datas (données)	FileSize - 44	...	little	Les données !

Fig3.5 : Exemple détaillé de la structure des fichiers audio wav

Fichier wav en hexadécimal

Voila une représentation détaillée d'un fichier wav avec les trois blocs (riff, fmt et data)

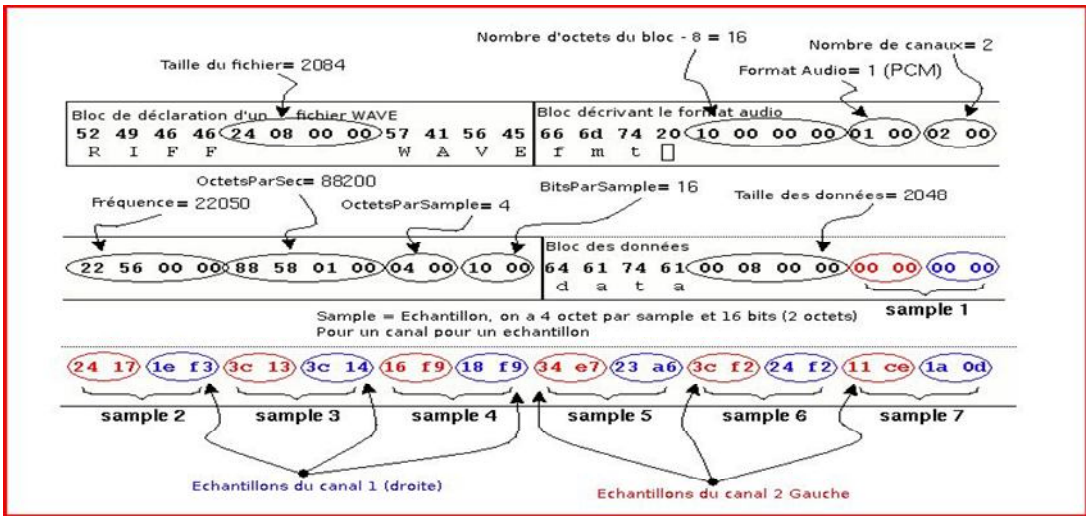


Fig.3.6: Fichier wav en hexadécimal [8]

Remarque

Un fichier .WAV comporte 44 octets d'en-tête.

Il y a une différence entre la taille fichier Sous Windows et « taille sur le disque ».

Cette différence provient de la taille des clusters (bloc mémoire ou secteur) occupés par le fichier. Les Ko sont en fait des Ko, c'est-à-dire qu'un Ko=1024 octets et non 1000.

3.10 : conclusion

Dans ce chapitre, nous avons montré les différents types de format audio les plus utilisé.

Nous sommes intéressés par le format de fichiers WAVE.

Pour cela, nous avons traité et étudié les différents algorithmes de chiffrement afin de sécuriser et protéger ce type des fichiers audio qui seront circulés dans notre réseau de communication.

Chapitre 4 :

implémentation de l'application pour sécuriser et protéger des fichiers son.

4.1 : Introduction

Nous présentons dans ce chapitre, premièrement l'environnement de notre travail.

Notre application est écrite en JAVA, et donc on a naturellement utilisé Netbeans. Puis nous montrons quelques captures d'écran de l'interface graphique de notre application.

Nous effectuons une étude comparative entre les différents algorithmes de chiffrement appliqué sur les fichiers son de type WAV, nous présentons une comparaison sur la qualité et la vitesse de l'opération de chiffrement et déchiffrement des fichiers son. Nous avons utilisé un ordinateur Intel (R) core (TM) i3-2330 CPU 2.20 MHz avec une RAM de 4.00 Go et un système d'exploitation windows7.

4. 2. Paramètres de comparaison

La comparaison va se faire sur plusieurs points et suivant certains critères :

- Qualité de chiffrement.
- Le temps de chiffrement et de déchiffrement.

4.3: Introduction sur JAVA

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C.

JAVA est un langage développé par SUN et qui selon ses concepteurs est :

- Simple
-Orienté objets
-Robuste et sûr
-Indépendant des architectures matérielles
-Multitâches.

Simple

Cette notion est relative mais par rapport au C++, JAVA est plus simple à utiliser. En particulier, il n'existe pas de pointeurs explicites et la gestion de la mémoire est transparente pour le programmeur.

Orienté objets

Un objet informatique est une entité munie de propriétés et de méthodes capables d'agir sur ses propriétés ou de réaliser certaines actions. Un objet contient des données et du code permettant de les manipuler. Dans JAVA, cette notion est poussée à l'extrême puisque dans ce langage tout est objet y compris le programme.

Robuste et sûr

Le typage des données est extrêmement strict. Aucune conversion de type implicite pouvant provoquer une perte de précision n'est possible. Comme les pointeurs ne sont pas accessibles les erreurs de gestion mémoire sont impossibles. Pour les applets, il est en principe impossible d'accéder aux ressources de la machine hôte. Enfin lors de l'exécution, on vérifie que le code généré par le compilateur n'a pas été altéré.

Indépendant des architectures matérielles

Le compilateur génère un code universel le « byte-code ». Un interpréteur spécifique à l'ordinateur hôte appelé « machine-virtuelle » permet l'exécution des programmes. La représentation des données étant indépendante de la machine qui exécute le code, les résultats des calculs sont indépendants de cette machine.
Dans les versions récentes des machines virtuelles, un compilateur génère un code directement exécutable sur la machine ce qui permet d'obtenir des performances comparables à celle d'un langage directement compilé.

Multitâches

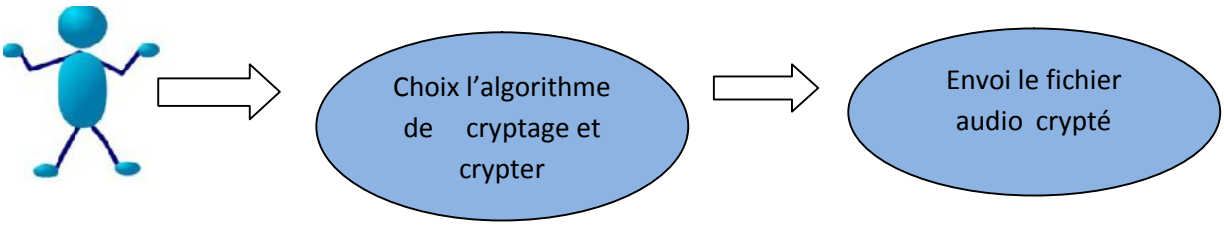
JAVA permet l'exécution en apparence simultanée de plusieurs processus. En réalité on accorde de façon séquentielle un peu du temps processeur à chaque processus. On dit aussi multithread.

4.4 : Schéma de l'Application

Notre application est un transfert des fichiers audio sécurisé (grâce au cryptage) selon le modèle client/serveur pour l'utilisé il suffit le serveur capte (ou choix) un fichier audio l'extension .wav puis le crypter à l'aide d'un des algorithmes proposé et enfin l'envoyer , le client décrypte le fichier qui a reçu en utilisant le même l'algorithme et la même clé que le serveur.

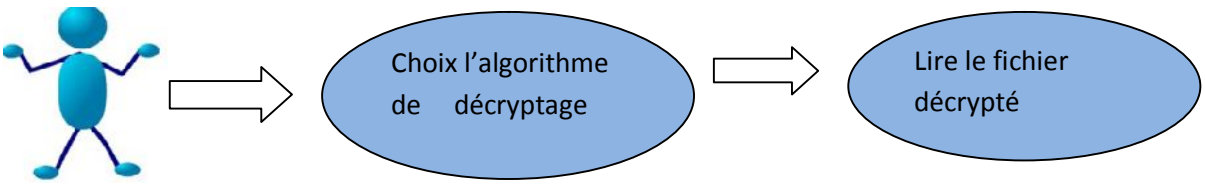
le client et le serveur sont d'accord sur l'algorithme utilisé à l'aide de transfert de message

coté serveur :



Serveur

Coté client :



Client

4.5 : Réalisation de l'application

Notre objectif de cette implémentation se base sur la performance et l'efficacité de chaque type d'algorithme de cryptage appliqué dans le chiffrement des fichiers son (fichier wav).

- **Classe connect :**

La classe connect permet faire la connexion entre le client et le serveur grâce à la méthode suivante :

```
public connect(String ip, int port){
try { s = new Socket(ip, port);
        in = s.getInputStream();
out = s.getOutputStream();  }
```

- **Classes client_envoi & serveur_envoi**

  Permet de transfert des messages sécurisé entre le client et le serveur bien sûr qu'ils se trouvent dans le même réseau.

- **Classes client_recoi & serveur_recoi**

  Permet de recevoir des messages sécurisé entre le client et le serveur bien sûr qu'ils se trouvent dans le même réseau.

4.6 :L'interface de l'application

A-Serveur

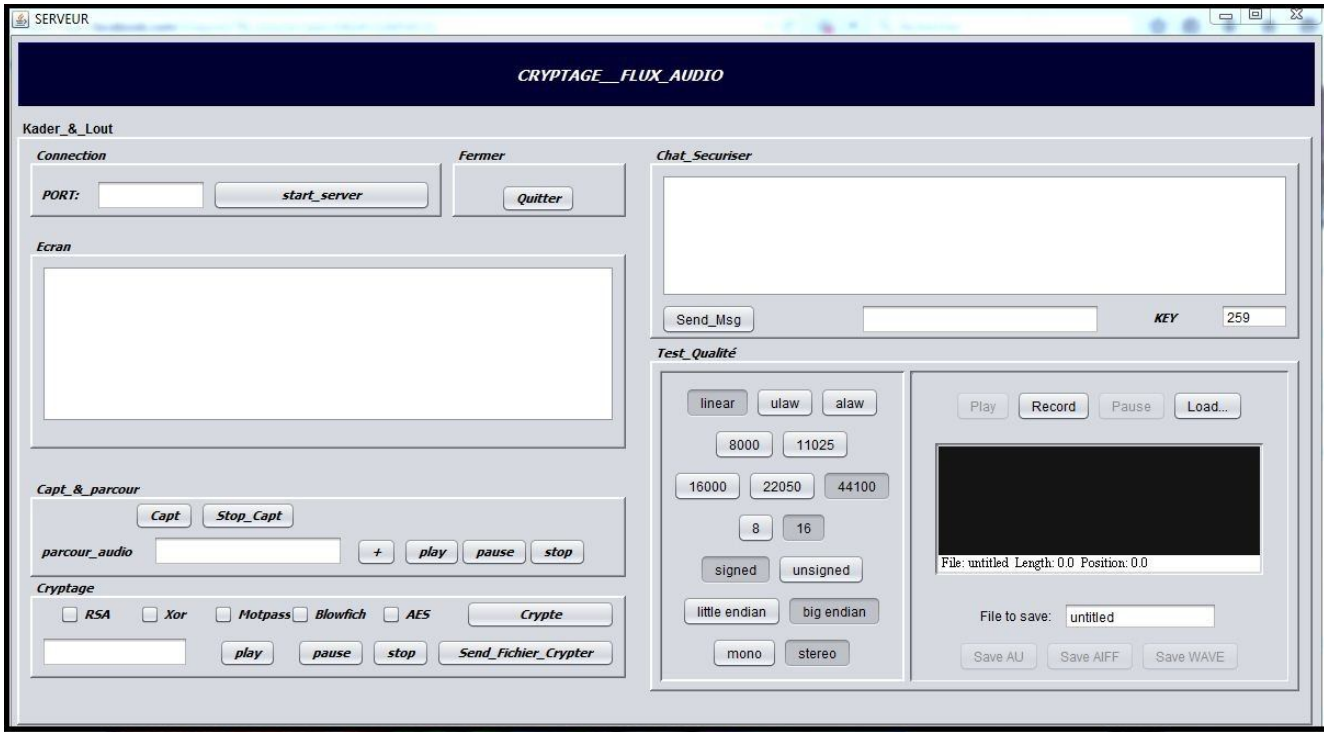


Fig.4.1 : Interface Serveur

4.6.1. Le menu de serveur

4.6.1.1. Démarrer le serveur

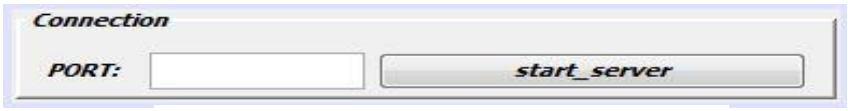


Fig.4.2 : démarrer le serveur

Le bouton démarre Serveur : permet d'établir une connexion avec port entré

4.6.1.2. Partie capteur

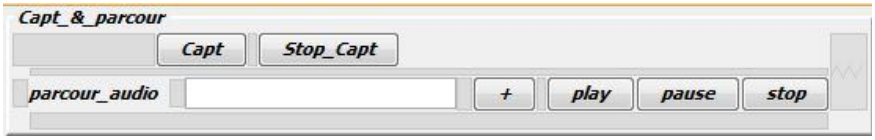


Fig.4.3 : capteur de son

Le bouton capt: permet capter la voix et enregistrer dans un fichier .Wav

Le bouton Stop Capt: permet du stoppé l'enregistrement de la voix.

Le bouton + : parcourir un fichier audio.

Le bouton Play : permet d e jouer des fichiers audio

Le bouton Pause : permet du suspendre les fichiers audio

Le bouton stop : permet d'arrêter la lecture du fichier audio

4.6.1.3. Partie Cryptage

Fig.4.4 : choix l'algorithme de cryptage

Le bouton crypter : faire appelle a la fenêtre du chiffrement pour crypter le fichier audio selon l'algorithme sélectionné (RSA, XOR, MOTPASS, BLOWFICH, AES)

Le bouton Play : permet d e jouer des fichiers audio cryptés

Le bouton Pause : permet du suspendre les fichiers audio

Le bouton stop : permet d'arrêter la lecture du fichier audio

Le bouton send fichier crypter : permet d'envoi le fichier crypté au client

Le bouton Arrêter : permet du suspendez le cryptage

Le bouton Fermer : permet du quitter la fenêtre du chiffrement.

B-Client

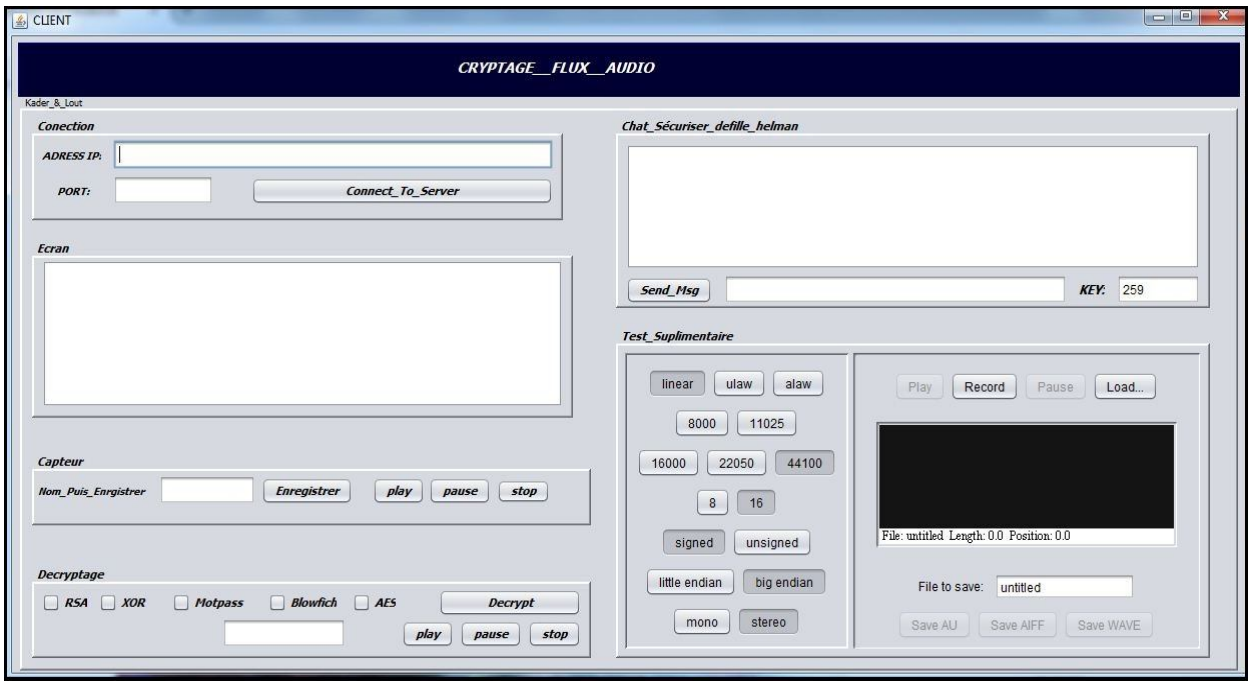


Fig.4.5 : Interface Client

4.6.2. Le menu de Client

4.6.2.1. Partie connexion

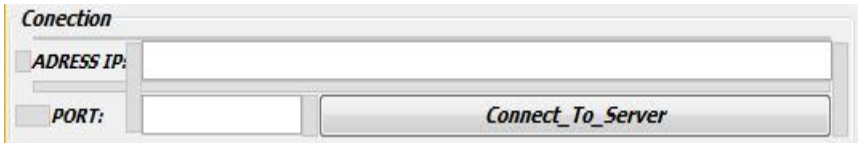


Fig.4.6 : connexion avec le serveur

Le bouton Connect_to_server: permet d'établir une connexion avec port entré et l'adresse du serveur

4.6.2.2. Partie d'accepté le fichier crypter :

Fig.4.7 : enregistrement le fichier audio

Le bouton Enregistrer: permet d'enregistrer le fichier reçu par serveur

Le bouton Play : permet de jouer du fichier audio crypté

Le bouton Pause : permet du suspendre les fichiers audio

Le bouton stop : permet d'arrêter la lecture du fichier audio crypté

4.6.2.3. Partie Décryptage

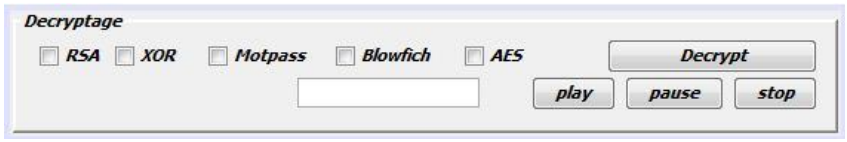


Fig.4.8 : choix l'algorithme de décryptage

Le bouton crypter : faire appelle a la fenêtre du déchiffrement pour décrypter le fichier audio selon l'algorithme sélectionné (RSA, XOR, MOTPASS, BLOWFICH, AES)

Le bouton Play : permet d e jouer du fichier audio décrypté

Le bouton stop : permet d'arrêter la lecture du fichier audio décrypté

4.6.2.4. Partie du chat

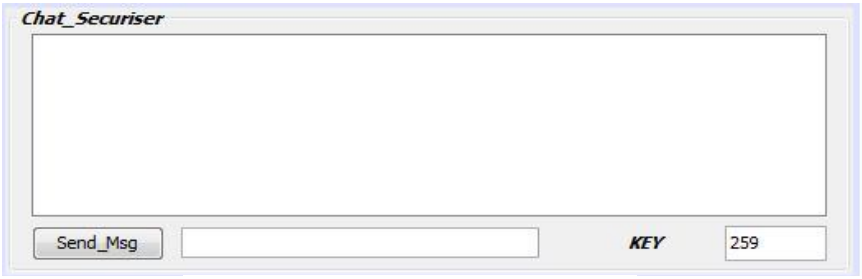


Fig.4.9 : partie discussion

Le bouton Send_Msg : permet d'envoi le message crypter selon l'algorithme defile helman

. Fenêtre du chiffrement et déchiffrement

Fig.4.10 : Fenêtre du chiffrement et déchiffrement

Le bouton Commencer : permet du démarrer le décryptage

Le bouton Arrêter : permet du suspendez le décryptage

Le bouton Fermer : permet du quitter la fenêtre du déchiffrement

4.6.3. Partie Test qualité

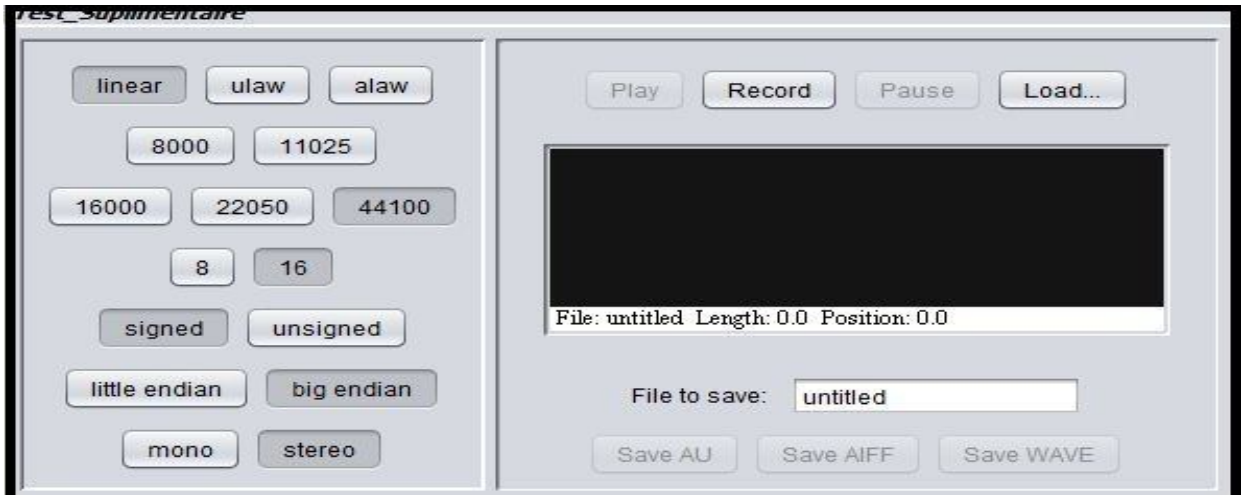


Fig.4.11 : test qualité

Le bouton capter: permet capter la voix et enregistrer dans un fichier

Le bouton Stop: permet du stoppé l'enregistrement de la voix.

Le bouton load : permet du parcourir n'import quelle fichier audio

Le bouton Play : permet d e jouer des fichiers audio

Le bouton stop : permet d'arrêter la lecture du fichier audio

Le bouton Save AU : permet d'enregistrer le fichier capter sous forme .au

Le bouton Save AIFF : permet d'enregistrer le fichier capter sous forme .aiff

Le bouton Save wav : permet d'enregistrer le fichier capter sous forme .wav

Partie gauche : donne les Caractéristique de fichier parcourir sous forme des bouton click

Fenêtre D'affichage de l'amplitude

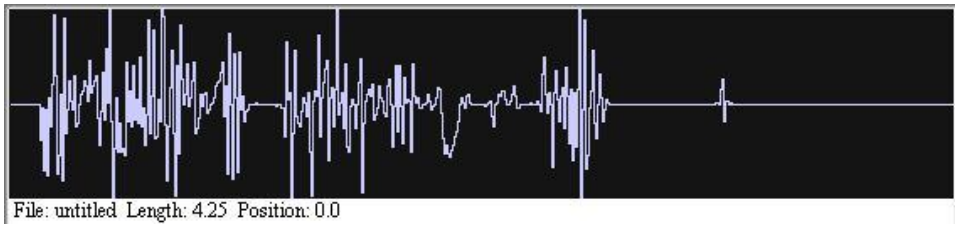


Fig.4.12 : Affichage de l'amplitude

Permet d'afficher la qualité de fichier audio même la taille de fichier

4.6.4. Les Exceptions

Fig.4.13 : Exception sur le PORT

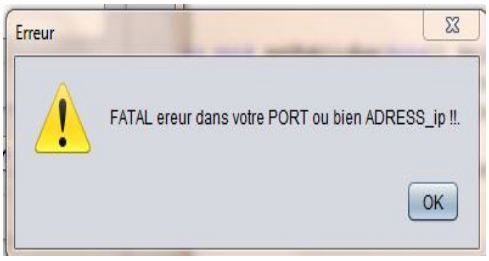


Fig.4.14 : Exception sur le PORT ou

Fig.4.15 : Exception sur le choix de l'algorithme

Fig.4.16 : Exception sur la lecture de fichier

4.7 : Résultats de temps de chiffrement de son par les différents algorithmes de cryptage

			Temps de chiffrement par algorithme				
	taille	**Type**	**RSA**	**XOR**	**Mot de passe**	**Blowfich**	**AES**
Fichier 1	**480 ko**	**.wav**	**5.03 s**	**0.004 s**	**0.220 s**	**0.060 s**	**0.078 s**
Fichier 2	**264 ko**	**.wav**	**1.93 s**	**0.003 s**	**0.192 s**	**0.029 s**	**0.034 s**

Tab.4.1 : temps de chiffrement du son par les différents algorithmes de cryptage

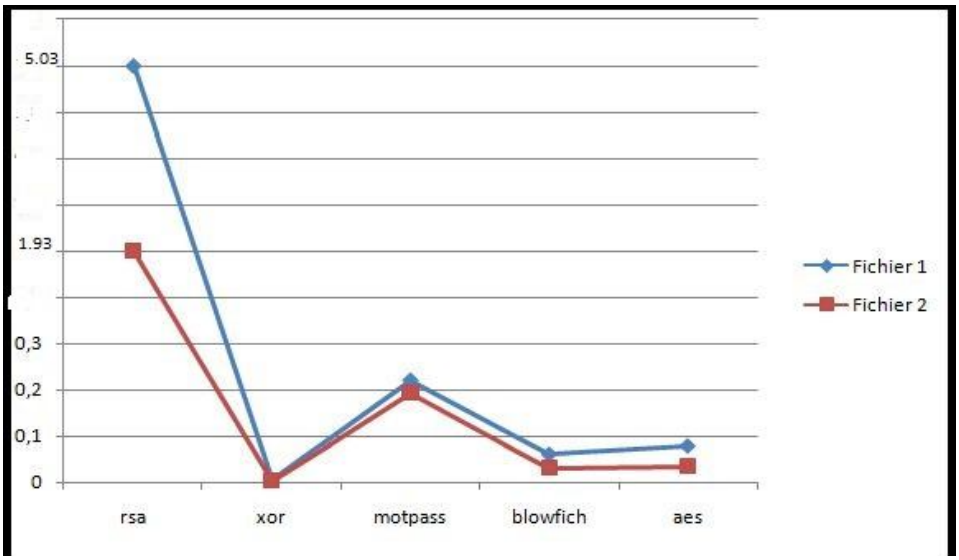


Fig. 4.17 : le temps de chiffrement audio par les différents algorithmes de cryptage

Remarque :

- ✓ Le temps de chiffrement est changé par le changement de la taille de fichier à chiffrer.
- ✓ Chaque fois en augmente dans la taille de fichier, la vitesse de l'opération de chiffrement sera automatiquement augmenté
- ✓ D'après les résultats obtenue, l'algorithme ou-exclusif c'est l'algorithme le plus rapide par

  contre l'algorithme RSA à clé publique c'est l'algorithme le plus lent.

4.8 : Résultats de temps de déchiffrement de son par les différents algorithmes de décryptage

			Temps de déchiffrement par algorithme				
	Taille	**Type**	**RSA**	**XOR**	**Mot de passe**	**Blowfich**	**AES**
Fichier 1	**480 ko**	**.wav**	**11.03 s**	**0.260 s**	**0.323 s**	**0.050 s**	**0.064 s**
Fichier 2	**260 ko**	**.wav**	**5.03 s**	**0.176 s**	**0.244 s**	**0.027 s**	**0.022 s**

Tab.4.2 : temps de déchiffrement du son par les différents algorithmes de décryptage

R

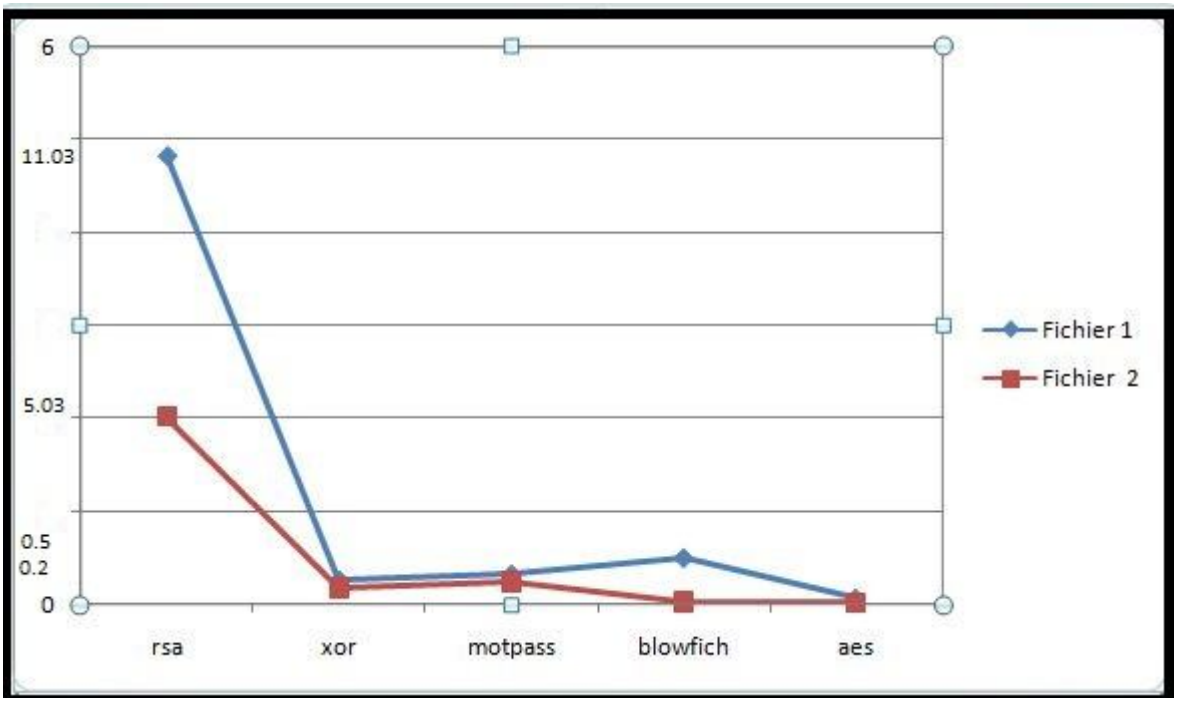


Fig.4.18 : Le temps de déchiffrement de son par les différents algorithmes de cryptage.

déchiffrement le plus lourd

- ✓ Nous remarquons aussi que le temps de déchiffrement et plus rapide que le temps de chiffrement

4.9. Résultats de la qualité de chiffrement de son par les différents algorithmes de cryptage

	Taille	RSA	XOR	Mot de passe	Blowfich	AES
Fichier 1	**480 ko**	Très bonne	mauvaise	Bonne	bonne	bonne

Tab.4.3 : la qualité de chiffrement du audio par les différents algorithmes de cryptage

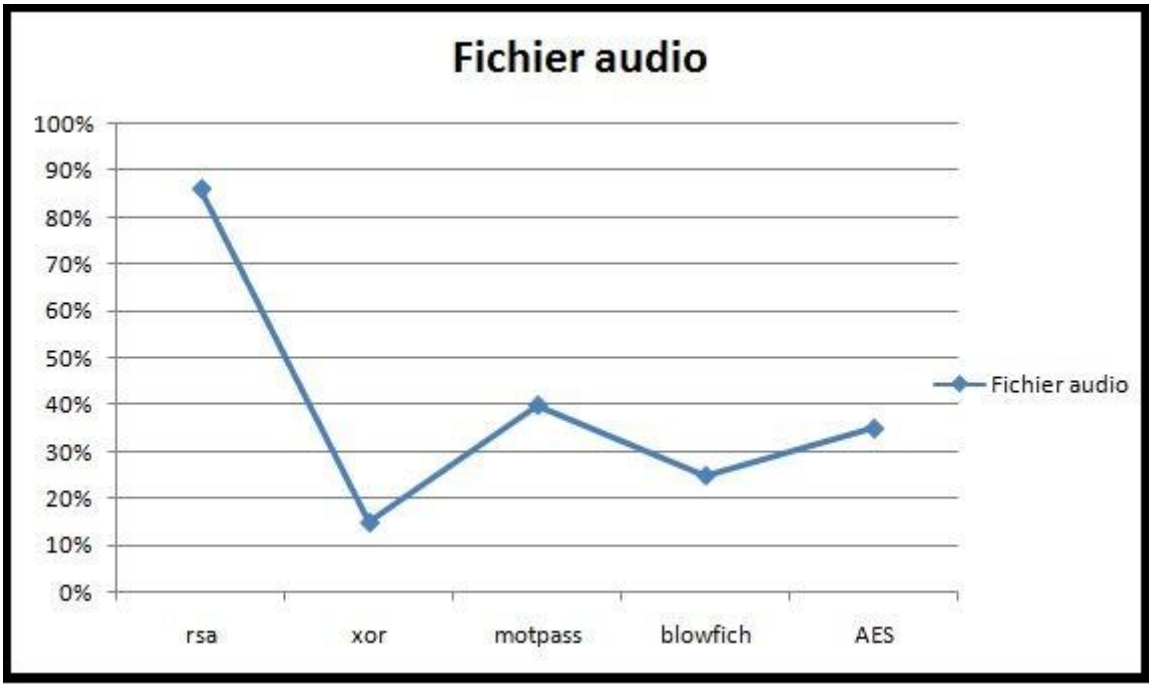


Fig.4.19 : la qualité de chiffrement du son par les différents algorithmes de cryptage

4.10. L'amplitude de chaque fichier crypté de chaque algorithme

4.10.1 : L'amplitude de fichier origine (capteur.wav)

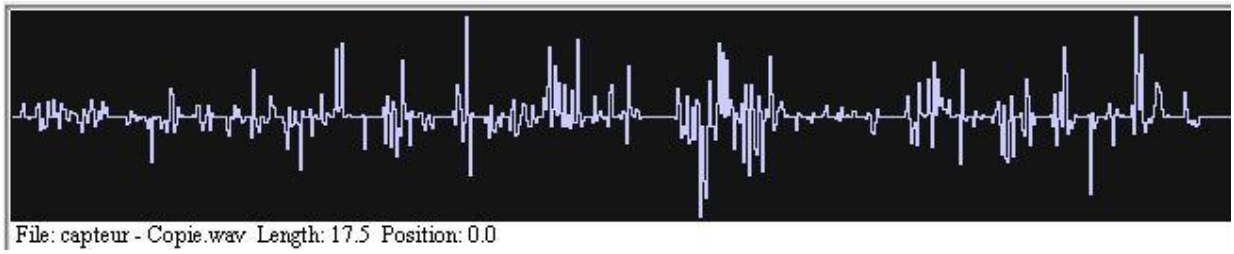


Fig.4.20 : L'amplitude de fichier origine

4.10.2 :L'amplitude de fichier crypté par l'algorithme RSA

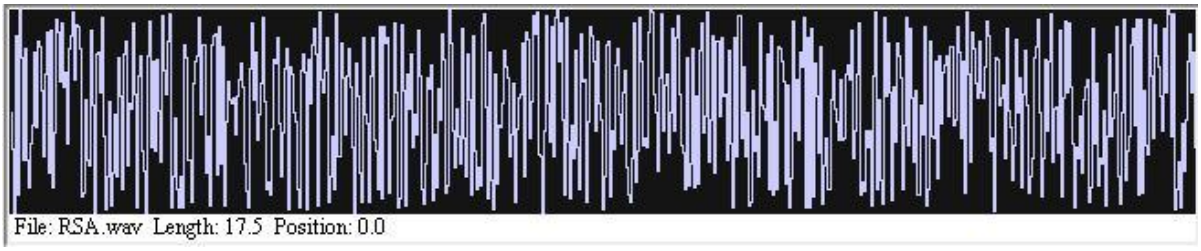


Fig. .4.21 :L'amplitude de fichier crypté par RSA

4.10.3 :L'amplitude de fichier crypté par Xor

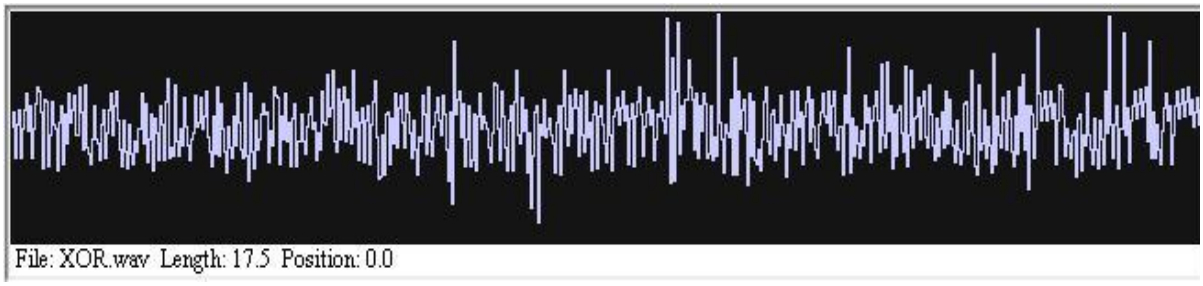


Fig.4.22 :L'amplitude de fichier crypté par XOR

4.10.6 :L'amplitude de fichier crypté par AES :

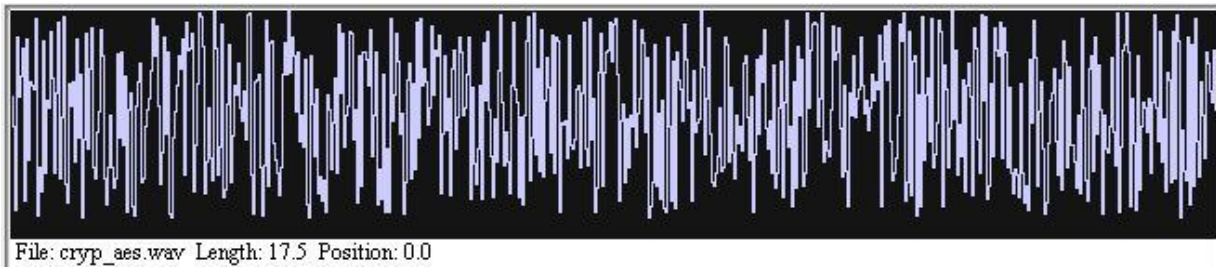


Fig.4.23 :L'amplitude de fichier crypté par AES

4.10.5 :L'amplitude de fichier crypté par blowfich

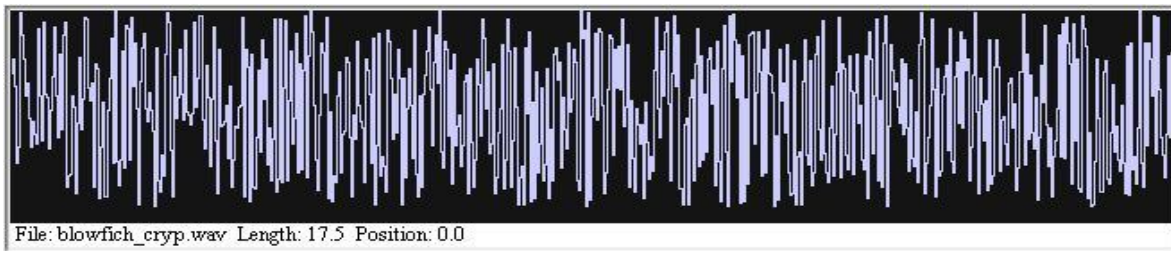


Fig.4.24 :L'amplitude de fichier crypté par Blowfich

4.10.4 :L'amplitude de fichier crypté par Motpass

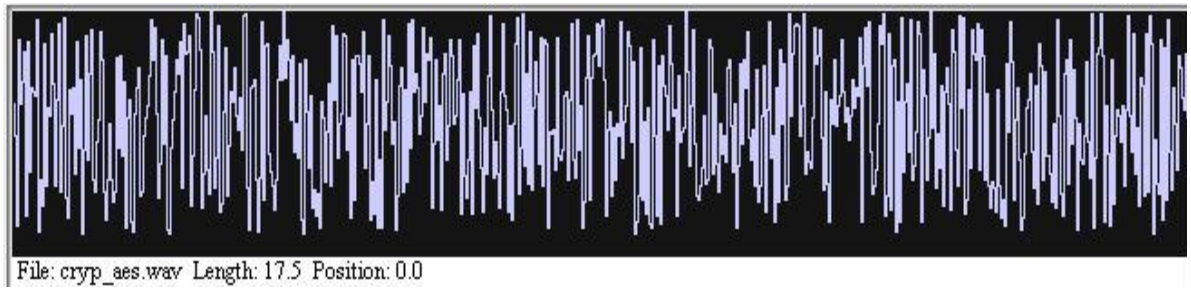


Fig.4.25 :L'amplitude de fichier crypté par Motpass

Remarque

- ✓ D'après le tableau et les résultats obtenus, nous avons remarqué que la qualité de cryptage l'audio est très bonne dans l'algorithme RSA. Par contre l'algorithme de chiffrement par XOR donne une mauvaise qualité de chiffrement

- ✓ Même l'algorithme AES et l'algorithme du MOTPASS, Blowfich donnent une bonne qualité de chiffrement des fichiers audio.

4.11 :Conclusion

Dans ce chapitre, nous avons vu les différents résultats obtenus par l'opération de cryptage des fichiers audio de format .wav par quelques algorithmes de chiffrement comme XOR, RSA, Motpass, Blowfich, et AES. .

Les critères de performance sont basés sur deux facteurs : le temps et la qualité de son crypté.

L'algorithme RSA, Blowfich et AES donnent une meilleur qualité dans le chiffrement des fichiers son. Par contre l'algorithme XOR est considéré comme l'algorithme le plus rapide dans le temps de chiffrement/déchiffrement mais avec une mauvaise qualité de chiffrement.

Conclusion générale

Au cours de cette mémoire, nous avons étudié et implémenté les différents algorithmes de chiffrement à clé publique et à clé secrète comme RSA, AES, mot de passe, et par XOR afin de crypter les fichiers son (fichier audio).

La sécurisation des informations échangées dans un réseau, est devenue très intéressant dans le domaine Informatique.

Les critères de comparaisons pris en considération dans le cryptage des fichiers audio sont :

- ✓ La vitesse et le temps de l'opération de chiffrement par les différents algorithmes étudiés dans notre projet de fin d'étude.
- ✓ La qualité (bruité) et la performance de l'opération de cryptage appliqué sur des fichiers son.

D'après l'étude comparative que nous avons effectués, nous remarquons que l'algorithme RSA donne une meilleur qualité de chiffrement des fichiers audio par rapport aux autres algorithme de chiffrement comme l'algorithme classique XOR en termes de bruité.

Nous avons observés aussi, que l'algorithme ou exclusif (XOR) est très rapide dans la vitesse de chiffrement et déchiffrement par rapport aux autres algorithmes. Cela revient à la structure simple de cet algorithme.

Références bibliographiques

[1] Network Programming for Microsoft Windows Second Edition ,MSPress2002

[2] Mr Benmammar Badr , Construction d'applications réparties Architecture n – tiers et Frameworks globaux (J2EE vs .Net).

[3] Mr benaissa Samir , chapitre1-architecture-client-serveur

[4] Mme Labraoui Nabila , Sécurité des réseaux sans fil Chapitre 2: Introduction à la cryptographie

[5] Mr Lehsaini M , cours_RSA,

[6] See R. Steinmetz, K. Nahrstedt. Multimedia: Computing, Communications & Applications. Innovative Technology Series. Prentice Hall P T R. 1995.

[7] David Salomon , D*ata-compression-the-complete-reference, Springer 2007*

[8] *Audio* Tubes - *caractéristiques* et utilisation » de Francis Ibre ... *caractéristiques* & utilisation .... Version *PDF* commander ce *livre* .

Les site web

[s1] - w3.polytech.univ-montp2.fr/~karen.godary/M1/Trans_Client_Serveur.pdf

[s2] - igm.univ-mlv.fr/~duris/NTREZO/20022003/ClientLeger.pdf

[s3] - www.clusir-rha.fr/sites/default/files/upload/s2/ClusirCryptographie.pdf

[s4] - www.student.montefiore.ulg.ac.be/~s091678/files/resume_crypto.pdf

[s5] -http://www.cryptage.org/

[s6]- http://www.cryptage.org/cle-publique.html

[s7] - http://fr.wikipedia.org/wiki/Wikipédia:Portail_Cryptologie

[s8] - http://ram-0000.developpez.com/tutoriels/cryptographie/?page=page_8

[s9] -docs/french/IntroToCrypto.pdf

[s10] http://www.laptitemaison.com/ptitemaison/article.php3?id_article=649

[s11] http://fr.wikipedia.org/wiki/Format_de_fichier_audio

Résumé

Dans l'informatique moderne, de nombreuses applications fonctionnent selon un environnement client-serveur.
A l'heure où le multimédia est de plus en plus présent dans la communication des entreprises Une identité audio personnalisée ainsi que le son deviennent indispensables.
Dans ce projet nous avons eu l'opportunité d'attaquer une des notions indispensables dans le domaine informatique : la cryptographie.
La cryptographie est de plus en plus utilisée. Elle permet de résoudre de nombreux problèmes de sécurité.
L'objectif principal de notre mémoire est d'assurer un transfert sécurisé de données représenté par des flux audio en utilisant les algorithmes de chiffrements à clé publique et à clé secrète comme RSA et AES.
Mot clé : client/serveur, audio, clé publique, clé privé, chiffrer et déchiffré.

Abstract

In modern computers, many applications operate in a client-server environment. At a time when multimedia is increasingly present in the business communication A customized audio identity and the sound are essential.
In this project we have had the opportunity to tackle one of the essential concepts in IT: cryptography.
Cryptography is increasingly used. It solves many security problems. The main purpose of our brief is to ensure secure transfer of data represented by the audio streams using ciphers public key and secret key as RSA and AES.
Keyword: client / server, audio, public key, private key, encrypt and decrypted.

خلاصة

في مجال الإعلام الآلي الحديث اعتمدت مختلف التطبيقات الجديدة على نظام الشبكات في وقت شهد كثرة وسائل الاتصال و اعتبرت الصورة و الصوت إحدى أولوياته

فمن خلال هذا المشروع أتيحت لنا الفرصة لمعالجة واحدة من المفاهيم الأساسية في مجال تكنولوجيا المعلومات و هي التشفير

و استعمال التشفير شهد تطور متزايد بحيث قام بحل مختلف المشكلات الأمنية

و الغرض الرئيسي من عملنا هذا هو ضمان نقل آمن الصوتية باستخدام الشفرات المفتاح العام والمفتاح السري أو كما يعرفها و العامة RSA/AES

الكلمات المفتاحية : الصوت, التشفير و فك التشفير , المفتاح العام, المفتاح الخاص , العميل/الخادم.