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 système distribué (RSD)*

Thème

Etude et analyse des algorithmes de brouillage utilisé dans la sécurité des Signaux de parole

Réalisé par :

Guerroudj Lina.

Présenté le ***06 Juillet 2019*** *devant le jury composé de :*

- **Mme. LABRAOUI NABILA** *(Présidente)*
- **Mr. BENAISSA MOHAMMED** *(Encadreur)*
- **Mr. MANAA MOHAMMED** *(Examinateur)*

Année universitaire: 2018-2019

Remerciements

A Madame **LABRAOUI NABILA** qui nous a fait le grand honneur de présider le jury de ce mémoire.

A Monsieur **MANAA MOHAMMED** (Examinateur) qui a bien voulu me prêter son aide.

A Messieurs les membres de jury que nous remercions de l'intérêt qu'ils ont bien voulu apporter à notre travail par sa lecture et sa discussion.

Je tiens à remercier **BENAISSA MOHAMMED** *qui a bien voulu m'encadrer, me conseiller, me guider et me confier ce travail.*

Et je tiens à exprimer ma gratitude la plus sincère pour **DIEU TOUT PUISSANT** *sans l'aide duquel ce mémoire n'aurait pas eu lieu.*

Dédicace

Je dédie ce mémoire aux plus chers proches qui m'ont inspirée par leur soutient :

Mes parents.

Mes sœurs qui m'ont encouragé.

Ma famille.

Mes professeurs.

Toutes mes amies.

Et à toute personne pouvant apprécier les notions exposées dans ce mémoire.

Liste des figures

Chapitre1

Chapitre 2

Chapitre 3

Liste des figures

Chapitre 4

Liste des figures

Liste des tableaux

Chapitre 1

Chapitre 3

Chapitre 4

Liste des tableaux

Sommaire

Table de matières

Table de matières

Introduction générale

Dès le début de la civilisation, l'homme s'est toujours préoccupé de l'information, pour rendre sa vie quotidienne plus facile et plus confortable. Actuellement, depuis la révélation du réseau internet, l'information et la communication s'avèrent faciles et adéquates d'où la nécessité d'intégrer les réseaux informatiques dans différents systèmes d'informations et communication.

À présent chaque entreprise doit avoir un réseau informatique de taille plus ou moins important est mise en œuvre.

Le nombre de machines existant dans ces réseaux peut être très élevé, une simple panne peut entrainer des catastrophes indéniables d'où l'intérêt de la maintenance est la bonne gestion dans ces réseaux.

Cependant de nos jours, afin d'accéder au pouvoir à des fins militaires et diplomatique, l'homme doit dissimuler des informations confidentielles de façon sécurisée, cet aspect de dissimuler les informations rentre dans un vaste domaine appelé cryptologie ou science du secret celle-ci comporte deux branches : la cryptographie et la cryptanalyse.

La cryptographie est l'étude des méthodes qui assume la transmission des informations confidentielles en transformant le message tout en le rendant incompréhensible, c'est ce qu'on appelle le chiffrement, son but est donc de modifier un test en clair en un texte chiffré dit cryptogramme. En revanche, la cryptanalyse est une technique permettant de décrypter des textes chiffrés selon la méthode de décryptage qui a pour objectif de retrouver un texte en clair à partir d'un texte chiffré tout en utilisant une clé.

Notamment, l'information transmise n'est pas seulement des données textuelles mais Aussi audio images numérique et autres multimédia. Les fichiers audio jouent un rôle indispensable dans notre vie ordinaire et plus leur utilisation est croissante, plus leur sécurité est vitale.de surcroît, il est nécessaire de protéger plusieurs domaines.

À ce moment, la confidentialité est devenue un moyen nécessaire contre tous types d'attaques dans tous les secteurs (militaire, santé, diplomatique, enseignement etc.) D'état d'un pays, c'est pour cela la cryptologie est apparue pour cacher et casser toutes informations.

Introduction générale

L'objectif de ce mémoire est d'assurer le transfert des données sécurisé en utilisant des algorithmes de brouillage sur un fichier audio (format wav) tout en s'inspirant du modèle client/serveur.

La structure de ce mémoire se répartit comme suite :

Chapitre 1 : Cryptographie à clé secrète et à clé publique.

Chapitre 2 : les formats des fichiers audio.

Chapitre 3 : les applications client/serveur.

Chapitre 4 : Implémentation de l'application.

Chapitre 1

Cryptographie à clé secrète et à clé publique

1.1Introduction

La cryptographie ou le codage des messages dite aussi étude des méthodes est devenue aujourd'hui un moyen de sécurité permettant de transmettre des données de manière confidentielle en se basant sur certaines sciences telles que les mathématiques, de l'informatique, et parfois même de la physique dans l'intention d'éviter toute sorte d'attaque, de guerre et de protéger les messages tout en appliquant des transformations qui le rendent incompréhensible.

Le but de cette approche est de fournir un certain nombre de services de sécurité : confidentialité, intégrité et authentification de l'origine des données ou d'un tiers.

Ce chapitre se focalisera globalement sur les méthodes de chiffrement dites modernes selon la Seconde Guerre mondiale d'où la cryptographie est le centre d'intérêt. En commençant par la saga DES et d'AES, en passant par le fameux RSA, le protocole le plus reconnu dans notre siècle, toutefois utilisés par les militaires et des sociétés qui ont un grand investissement financier. Dans cette partie Toutes les méthodes de cryptographie vont être présentées d'une façon chronologique selon leur ordre d'apparition.

Général Lewal a dit que : « La cryptologie est un auxiliaire puissant de la tactique militaire » (Études de guerre).

La résolution de ce dicton nous pousse à dire que la cryptologie est la "science du secret" d'où il se compose de deux facteurs primordiaux : le premier est la cryptographie, cette méthode génère toutes sortes de codages des messages, le second est la fonction inverse dite la cryptanalyse qui permet le décodage.

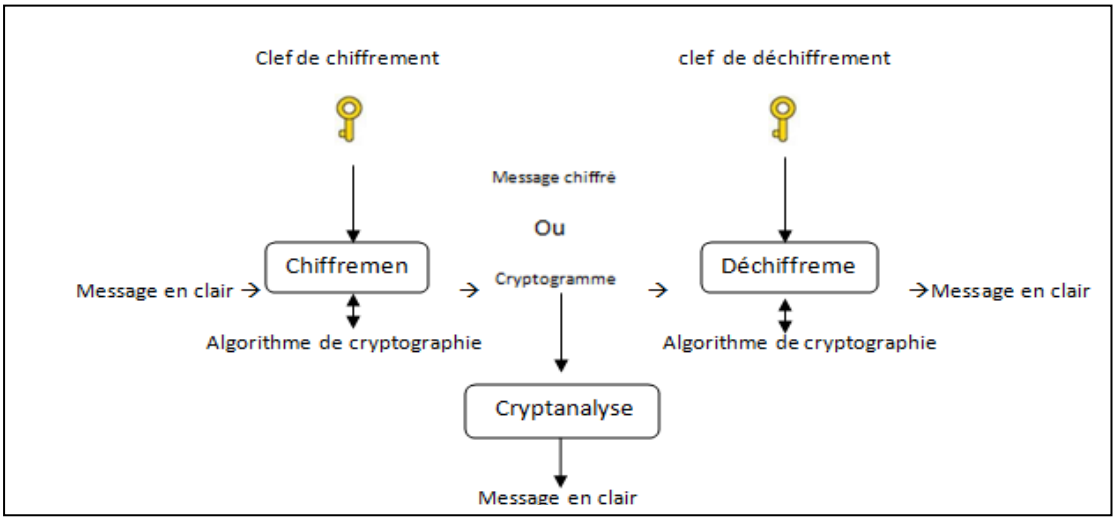


Fig1.1: protocole de chiffrement /déchiffrement.

Terminologie :

- **Cryptologie :** est une science mathématique comportant deux branches : la cryptographie et la cryptanalyse.
- **Cryptographie** : est l'étude des méthodes qui donne la possibilité d'envoyer les données de manière confidentielle sur un support donné.
- **Chiffrement :** Le chiffrement consiste à transformer une donnée (texte, message, ...) afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire. La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de déchiffrement.
- **Message chiffré ou 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.
- **Chiffrement :** l'algorithme utilisé afin de transformer le texte clair en texte chiffré.
- **Déchiffrement** : l'inverse du chiffrement d'où l'algorithme est utilisé pour transformer un texte chiffré en texte clair.

1.2Le Chiffre de César

Jules César était un général, homme politique et écrivain romain, né à Rome le 12 juillet ou le 13 juillet 100 av. J.-C. est mort le 15 mars 44 av. J.-C. Il aurait été assassiné par une conspiration, son propre fils Brutus lui portant le coup de grâce.

César s'est illustré lors de la guerre des gaules, ce qui a donné des siècles plus tard son personnage dans la bande dessinée Astérix le Gaulois. Il utilisait une méthode de chiffrement qui porte aujourd'hui son nom [a1].

1.2.1 Principe

Le chiffre de César est la méthode de cryptographie la plus ancienne communément admise par l'histoire. Il consiste en une substitution mono-alphabétique : chaque lettre est remplacée ("substitution") par une seule autre ("mono-alphabétique"), selon un certain décalage dans l'alphabet ou de façon arbitraire. D'après Suétone, César avait coutume d'utiliser un décalage de 3 lettres : A devient D, B devient E, C devient F, etc. Il écrivait donc son message normalement, puis remplaçait chaque lettre par celle qui lui correspondait :

Clair	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Chiffré	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C

Tab 1.1 : la table de césar.

Exemple :

D’après ce chiffrement, <<**Lina**>> devient << **QNSF**>> avec un décalage de 5[a2].

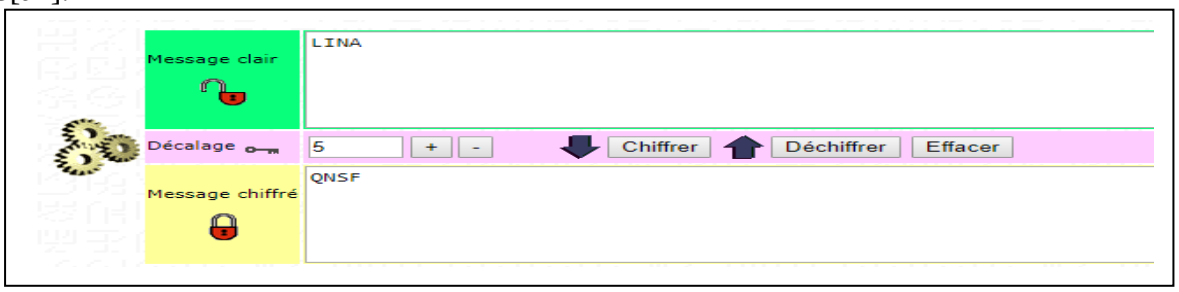


Fig1.2 : chiffrement du mot Lina.

1.2.2 Sécurité

Niveau sécurité, le chiffre de César ne sont pas fiables du tout, et ce pour deux raisons :

- Il n'existe que 26 façons différentes de crypter un message : puisqu'on ne dispose que de 26 lettres, il n'y a que 26 décalages possibles. Dès lors, des attaques exhaustives (tester toutes les décalages un à un) ne demanderaient que très peu de temps.
- Le chiffre de César est très vulnérable à l'analyse des fréquences.

1.3Chiffrement de vigenére

1.3.1 Principe

Le chiffrement de vigenére ressemble à celui de César, ce qui rend ce chiffrement différent est le fait d'utiliser un mot-clef long au lieu d'un simple caractère.

Afin de crypter, on choisit une clef (mot ou phrase), et on va correspondre chaque lettre claire à une autre lettre de la clef la lettre cryptée va être prise dans la colonne correspondante à la lettre du texte clair ainsi que la lettre de la clef sera pointé sur la ligne.

C : est le texte codé.

T : le texte.

K : est la clé.

On peut traduire ceci par la formule suivante :

C=T + K [mod 26].

Pour déchiffrer le message, on utilise l'opération inverse :

La lettre de la clé va correspondre à la ligne et on la suit jusqu'à tomber sur le caractère codé, la lettre décodée se positionne sur la première colonne.

On peut traduire ceci par la formule suivante:

T=C-K [mod 26].

1.4Chiffrement à clé secret [14][16][17]

1.4.1 Chiffrement et système cryptographique

Une fonction cryptographique, est une fonction qui permette la transformation des données est en général bijective :

F : est la fonction de chiffrement, M→ C d'où M'est le message en clair, C'est le message chiffré, vu que cette fonction est bijective l'opération peut-être faite dans le sens inverse ce qui signifie le passage de C→ M, noté sous la forme F^{-1} dit le déchiffrement .cette notion de chiffrement/déchiffrement est nommé "symétrique" ; car l'utilisation d'une bijection permet l'aller-retour très facilement.

L'emploi d'une bijection se justifie également parce qu'elle permet une correspondance univoque entre les éléments de l'ensemble. Autrement dit, à partir d'un message crypté, il n'est pas possible de tomber sur plusieurs possibilités de messages en clair.

Il est donc primordial, pour lire/écrire des messages cryptés :

- D'utiliser une fonction facilement inversible, en particulier pour un usage privé (il faut être capable de déchiffrer les messages sans avoir recours à des moyens techniques colossaux).
- De préserver cette fonction secrète, car si un "ennemi" se la procure, il lui suffira de l'inverser pour déchiffrer le message.

Ce dernier point implique un autre problème : l'utilisation d'une même fonction f pour crypter un message risque, à la longue, d'en dévoiler le secret. Ainsi, l'idéal serait de changer régulièrement de fonction de cryptage.

On définit donc un système cryptographique, ou de chiffrement, ou encore un chiffre, comme étant une famille finie 'F' de fonctions 'f' cryptographiques, chacune étant déterminée par un paramètre 'K', appelé clé secrète : $F = (f_K)$

On utilise ainsi la même "structure" de fonction de cryptage, mais avec chaque fois un paramètre K différent.[1] [2].

1.4.2 Le principe de Kerckhoffs

> La sécurité du chiffre ne doit pas dépendre de ce qui ne peut pas être facilement changé.

En d'autres termes, aucun secret ne doit résider dans l'algorithme mais plutôt dans la clé. Sans celle-ci, il doit être impossible de retrouver le texte clair à partir du texte chiffré. Par contre, si on connaît K, le déchiffrement est immédiat. On parle aussi de la Maxime de Shannon, dérivée du principe énoncé ci-dessus : L'adversaire connaît le système.

Remarque : Il faut distinguer les termes "Secret" et "Robustesse" d'un algorithme. Le secret de l'algorithme revient à cacher les concepts de celui-ci, ainsi que les

méthodes utilisées (fonctions mathématiques). La robustesse quant à elle désigne la résistance de l'algorithme à diverses attaques qui seront explicitées dans la suite de ces notes [a3].

1.4.3 Schémas de Feistel, ou chiffrement par blocs

Le chiffrement par bloc utilise des clefs secrètes petites au lieu des clefs à grands bits (de 80 à 128 bits), seulement l'utilisation de ces clefs est complexe au point qu'une attaque ne peut pas le déchiffrer.

Nonobstant, la taille de la clef est petite n'est-il pas facile d'essayer toutes les possibilités jusqu'à atteindre au décryptement voulu ? La réponse sera donc qu'aucun ordinateur ne pourra réaliser ça à un temps minime vu qu'une clef de 128 bits nous donne 2^{128} des clefs possibles que de « 0 » et «1 » successives.

L'objectif est donc de sélectionner en partant du message M une suite aléatoire de chiffres ou au moins les chiffres qui apparaissent aléatoirement d'où ils vont être déchiffrés selon la clef K.

Concrètement, il s'agit de construire une fonction bijective "pseudo-aléatoire" :

- En premier lieu, elle doit être une bijection, c'est-à-dire qu'il faut la correspondance entre chaque chiffre du message en clair et le chiffre du message codé ce qui signifie qu'on peut remonter d'une façon univoque en partant d'un chiffre C vers le chiffre en clair M.
- En second lieu, Elle doit être ou apparaître aléatoire, en cryptographie, la perfection même est l'aléatoire, le message codé doit sembler qu'il est généré au hasard afin de limiter les attaques liées aux analyses du texte chiffré ainsi ces redondances, etc.

La mise au point d'une fonction réunissant ces deux conditions posa problème aux cryptographes jusqu'aux années 1950, lorsque Feistel montra qu'une fonction pseudo-aléatoire se transformait, par une méthode simple, en bijection. Actuellement, c'est la méthode de chiffrement à clé secrète la plus utilisée.

Algorithme

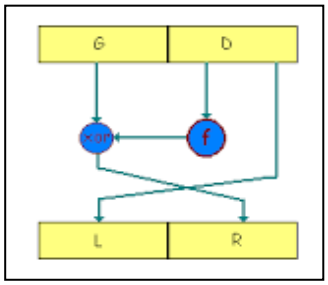


fig1.3 : Algorithme de Feistel.

Principe

Soit F une fonction dont elle a un argument de n bits.

L'algorithme de chiffrement va chiffrer 2n bits, d'ou il va être divisé en 2 parties n bit chacun.

Comme il est indiqué dans la figure G est la partie gauche, D est la partie droite.

Le miroir du bloc (G, D) est le bloc (L, R).

L=D, R=G XOR F(D).

A partir de (L, R) on obtient le couple (G, D) par les opérations D=L et G=R XOR f(L).Ce qui signifie que la transformation est bijective.

La partie droite n'a pas été transformée (juste envoyée à gauche). Il faut donc répéter le schéma de Feistel un certain nombre de fois (on parle de **tours**).

1.4.4 Le DES

Principe

Il utilise successivement les expansions de bits, d'un XOR, d'une réduction de bits, et d'une permutation de bits.

3) On recompose un bloc B'_{16} en "recollant" D_{16} et G_{16} dans cet ordre.

4) On effectue la permutation inverse de la permutation initiale 1)

Le décodage se fait en utilisant la même clé *K* mais en déroulant l'algorithme dans le sens inverse [a4].

Sécurité de DES

Au moment de son invention, ils sont focalisés sur le niveau de sécurité, l'étude était minutieuse d'où ils sont utilisés des techniques spéciales telles que la cryptanalyse différentielles ou linéaire, ont été inventées pour attaquer le DES, mais les attaques les plus efficaces proviennent d'une exploration exhaustive de l'espace des clefs, d'autant plus des matériels spécifiques ou des grands réseaux de stations de travail, alors il est possible de déchiffrer les cryptogrammes munis du DES en quelques jours, voire même quelques heures .Au train ou la puissance des ordinateurs augmente, on s'attend à ce que le DES puisse bientôt être cassé par un simple PC.

1.4.5 L'AES

L'AES (Advanced Encryption Standard) est, comme son nom l'indique, un standard de cryptage symétrique destiné à remplacer le DES (Data Encryption Standard) qui est devenu trop faible au regard des attaques actuelles.

Historiquement, le développement de l'AES a été instigué par le NIST (National Institue of Standards and Technology). Il est également approuvé par la NSA (National Security Agency) pour l'Encryption des informations dites très sensibles.

Cet algorithme suit les spécifications suivantes :

L'AES est un standard, donc libre d'utilisation, sans restriction d'usage ni brevet.

- C'est un algorithme du type symétrique
- C'est un algorithme de chiffrement par blocs
- Il supporte différentes combinaisons [longueur de clé]-[longueur de bloc] : 128-128, 192-128.
- 256-128 bits (en fait, l'AES supporte également des tailles de blocs variables, mais cela n'est pas retenu dans le standard).

En termes décimaux, ses différentes tailles possibles signifient concrètement que:

3.4 x 1038 clés de 128-bits possibles

6.2 x 1057 clés de 192-bits possibles

1.1 x 1077 clés de 256-bits possibles

Pour avoir un ordre d'idée, les clés DES ont une longueur de 56 bits (64 bits au total dont 8 pour les contrôles de parité), ce qui signifie qu'il y a approximativement 7.2 x 1016 clés différentes possibles. Cela nous donne un ordre de 1021 fois plus de clés 128 bits pour l'AES que de clés 56 bits pour le DES. En supposant que l'on puisse construire une machine qui pourrait cracker une clé DES en une seconde (donc qui puisse calculer 255 clés par seconde), alors cela prendrait environ 149 mille milliards d'années pour cracker une clé AES [a5].

1.4.6 Le OU exclusif

L'ou exclusif ou le XOR est un algorithme de cryptographie simple à manipuler sur un ordinateur. Cet algorithme est symétrique ainsi il fonctionne avec une clé qui doit être choisie selon le message à crypter.

Table de vérité de XOR		
A	B	R = A ⊕ B
0	0	0
0	1	1
1	0	1
1	1	0

Tab 1.2 : Tableau de XOR.

Principe

Considérons un document numérique qui consiste une suite de bits à chiffrer, dans cet algorithme qui est basé sur le chiffrement par flot on doit avoir deux suites de bits de même longueur, carrément aléatoire, on appelle cette suite la clé de chiffrement. On traite un à un les bits du document en clair, en le combinant avec le bit de même rang de la clé de chiffrement.

A et le bit en clair et C'est le bit de même rang de la suite aléatoire.

Le chiffrement consiste à calculer le bit D par :

D= A ⊕ C on dit que D est le chiffrer d'A.

Afin de déchiffrer D on utilise à nouveau le bit C de la suite aléatoire et on calcule :

D ⊕ C.

Dans ce cas le résultat sera A d'où le bit va être en clair vu que :

D ⊕ C = A ⊕ C ⊕ C = A⊕ 0 = A

On remarque que la même clé sert à chiffrer est à déchiffrer ont conclu que c'est un chiffrement symétrique.

Illustration

Le message en clair : *A* = 0110101011010100

La clé secrète : *K* = 0101011011100110

⊕ est le symbole qui représente l'opérateur XOR à chacun des bits Pour chiffre.

On utilise la table de vérité:

"A" le message, "K" la clé secrète.

Donc:

A→0110101011010100

⊕

K→ 0101011011100110

=

C→0011110000110010 le message chiffré:

Déchiffrement : $A = C \oplus K$ = 0110101011010100 (message déchiffré).

1.4.7 **Chiffrement par transposition** [7]

Principe

Ce chiffrement consiste à changer l'ordre des lettres ce qui permet de construire des anagrammes, une autre manière il consiste à découper le texte clair en bloc de taille identique ainsi qu'il doit garder la même permutation est l'applique sur tous les blocs, le texte doit éventuellement être complété pour permettre ce découpage .la clef de chiffrement est la permutation elle-même.

La longueur des données permutées est calculée par la factorielle dont son augmentation est rapide.

Cryptanalyse des chiffres de transposition

En général, la technique de transposition consiste en un chiffrement et une description. Le processus de chiffrement commence par la segmentation préalable des données vocales.[5]

Dans le chiffrement de transposition, la fréquence des lettres ne change pas, elle s'applique uniquement sur leurs ordres.

Le but d'une analyse des fréquences des lettres est de détecter l'utilisation probable de ce chiffrement seulement il ne peut pas le déchiffrer de plus, ce type de chiffrement non seulement garde toujours les mêmes indices de coïncidences.

Exemple

Message chiffré :

CARTM IELHX YEERX DEXUE VCCXP EXEEM OEUNM CMIRL XRTFO CXQYX EXISV NXMAH GRSML ZPEMS NQXXX ETNIX AAEXV UXURA FOEAH XUEUT AFXEH EHTEN NMFXA XNZOR ECSEI OAINE MRCFX SENSD PELXA HPRE

La clé de transposition :

8 4 9 14 1 2 16 10 3 17 15 19 11 5 20 6 7 12 13 18

Le déchiffrement :

Le déchiffrement se fait en remplissant les colonnes verticalement, dans l'ordre défini par la clé. On commence par remplir de haut en bas la colonne numérotée 01 avec les huit premiers caractères du message chiffré : CARTMIEL. On continue en remplissant

de la même façon la colonne numérotée 02 avec les huit caractères suivants HxYEERxD etc.

08	04	09	14	01	02	16	10	03	17	15	19	11	05	20	06	07	12	13	18
S	P	R	U	C	H	x	S	E	C	H	S	N	U	L	L	x	V	O	N
V	E	S	T	A	x	A	N	x	S	T	E	I	N	x	x	Q	U	E	E
N	x	M	A	R	Y	x	Q	U	E	E	N	x	M	A	R	Y	x	A	M
x	E	L	F	T	E	N	x	E	I	N	S	A	C	H	T	x	U	H	R
M	E	Z	x	M	E	Z	x	V	O	N	D	A	M	P	F	E	R	x	C
A	M	P	E	I	R	O	x	C	A	M	P	E	I	R	O	x	A	U	F
H	O	E	H	E	x	R	E	C	I	F	E	x	R	E	C	I	F	E	x
G	E	M	E	L	D	E	T	x	.	.	.	.	.	.	.	.	.	.	.

Tab 1.3 : matrice de déchiffrement.

Le texte clair est écrit horizontalement. Il s'avère être en Allemand :

Spruch 60. Von VESTA An STEIN.QUEEN MARY am Elften eins acht Uhr MEZ von Dampfer CAMPEIRO auf hoehe RECIFE gemeldet.

Soit, en Français :

Texte 60, de VESTA pour STEIN.

Queen Mary signalé au large de Recife le 11 à 18 heures HEC par le vapeur Campeiro. [a15]

1.4.8 Blowfish

Blowfish est un algorithme de chiffrement symétrique (i.e. “à clef secrète”) par blocs conçus par Bruce Schneier en 1993. Il tire son nom du poisson-lune japonais (ou fugu), qui en est également l’emblème.

Blowfish utilise une taille de bloc de 64 bits et la clé de longueur variable peut aller de 32 à 448 bits. Elle est basée sur l’idée qu’une bonne sécurité contre les attaques de cryptanalyse peut être obtenue en utilisant de très grandes clés pseudo-aléatoires.

Blowfish présente une bonne rapidité d’exécution excepté lors d’un changement de clé, il est environ 5 fois plus rapide que Triple DES et deux fois plus rapide qu’IDEA. Malgré son âge, il demeure encore solide du point de vue cryptographique avec relativement peu d’attaques efficaces sur les versions avec moins de tours. La version complète avec 16 tours est à ce jour entièrement fiable et la recherche exhaustive reste le seul moyen pour l’attaquer.

Il est utilisé dans de nombreux logiciels propriétaires et libres (dont GnuPG et OpenSSH).

Algorithme et principe générale

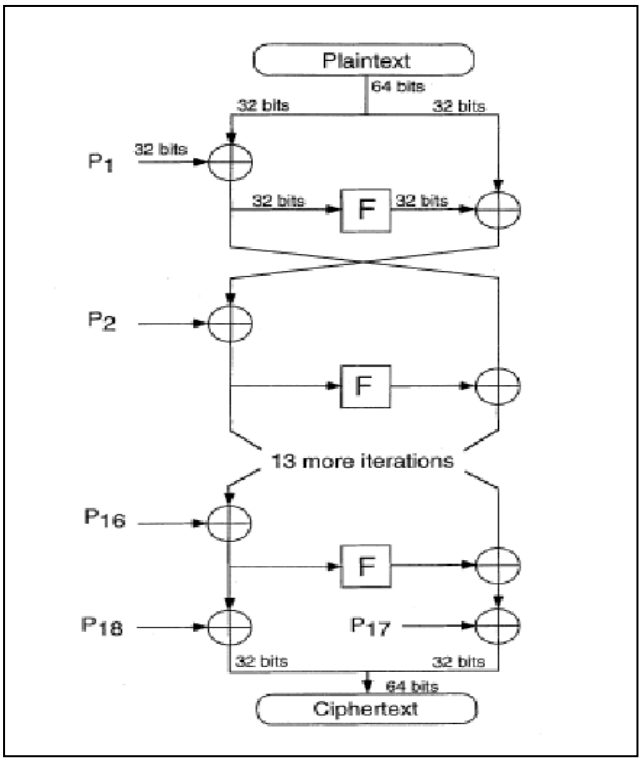


Fig 1.5 : Algorithme de Blowfish.

Le schéma montre la structure principale de Blowfish. Chaque ligne représente 32 bits. L'algorithme gère deux ensembles de clés : les 18 entrés du tableau P et les quatre S-Boxes de 256 éléments chacune.

Les S-Boxes acceptent un mot de 8 bits en entrée et produisent une sortie de 32 bits. Une entrée du tableau P est utilisée à chaque tour. Arrivée au tour final, la moitié du bloc de données subit un XOR avec un des deux éléments restants dans le tableau P.

Fonction de chiffrement

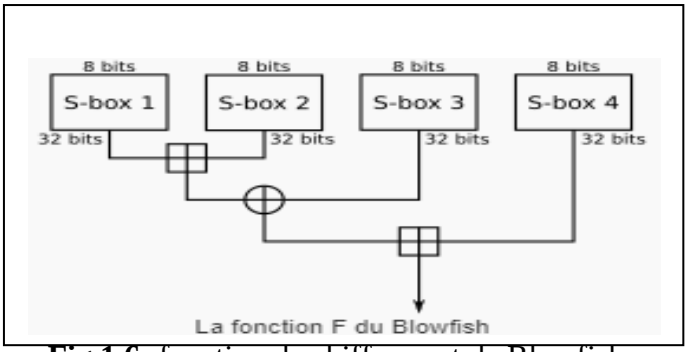


Fig 1.6: fonction de chiffrement de Blowfish.

La F-fonction de Blowfish sépare une entrée de 32 bits en quatre morceaux de 8 bits et les utilise comme entrées pour accéder aux S-Boxes. Les sorties sont additionnées avec une somme modulo 232 et l'algorithme effectue un XOR entre les deux sous-totaux pour produire la sortie finale de 32 bits.

En tant que schéma de Feistel, Blowfish peut être inversé simplement en appliquant un XOR des éléments 17 et 18 du tableau P sur le bloc chiffrés. Il faut ensuite utiliser les entrées du tableau P dans l'ordre inverse. [a5].

1.4.9 RC5

RC5 est un algorithme de chiffrement par bloc symétrique dont sa taille varie entre 32, 64,128 bits. il adresse deux blocs de mots à la fois.

Il contient une clé allant de 40 à 2040 bits et un nombre de tours de 0 à 255.

Le chiffrement original suggère un choix de paramètres avec une taille de bloc de 64 bits, une clef de 128-bits et 12 tours. il est possible de définir différentes instances du RC5.

Chaque instance est indiquée par RC5-w / r / b, d'où :

w = taille de mot en bits, r = nombre de tours et b = taille de la clé en octets.

- **les étapes de fonctionnement :**
  - **Etape1 :** Initialisation des constantes P et Q.
  - **Etape 2 :** conversion de la clé secrète K d'octets en mots.
  - **Etape 3 :** Initialisation de la sous-clé S
  - **Etape 4 :** Mélange de sous-clés.
  - **Etape 5 :** chiffrement.

Algorithme

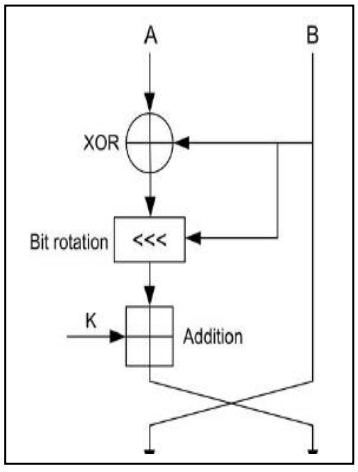


Fig 1.7 : opération de l'algorithme RC5.

1.5 Le chiffrement à clé publique :

1.5.1 Diffie et Hellman

En cryptographie, l'algorithme de Diffie et Hellman est un algorithme d'échange de clés, souvent utilisé dans les ouvertures de connexion des sites sécurisés, c'est le premier algorithme qui permet à deux personnes de mettre en place un système par lequel elles pourront échanger de façon secrète.

- Le protocole D-H est un protocole basé sur la cryptologie à clef publique d'où il intervient des valeurs publiques et privées, le partage des informations privées entre les deux pairs se fait sans que l'un reconnaisse l'autre.
- La sécurité est renforcée vue qu'il applique des logarithmes discrets et durs à calculer.
- Le résultat obtenu de cet algorithme permet de créer plusieurs clefs (clef secrète, clef de chiffrement de clefs....).

Principe

Alice et Bob veulent s'échanger une clé secrète par le protocole de Diffie-Hellman. Ils font des actions en parallèle, que l'on décrit dans le tableau suivant :

	Alice	Bob
Étape 1 :	Alice et Bob choisissent ensemble un grand nombre premier p et un entier $1 \leq a \leq p-1$. Cet échange n'a pas besoin d'être sécurisé.	
Étape 2 :	Alice choisit secrètement x_1.	Bob choisit secrètement x_2.
Étape 3 :	Alice calcule $y_1 = a^{x_1} \pmod{p}$.	Bob calcule $y_2 = a^{x_2} \pmod{p}$.
Étape 4 :	Alice et Bob s'échangent les valeurs de y_1 et y_2. Cet échange n'a pas besoin d'être sécurisé.	
Étape 5 :	Alice calcule $y_2^{x_1} = (a^{x_2})^{x_1} = a^{x_1 x_2} \pmod{p}$ et appelle ce nombre K, la clé secrète à partager avec Bob.	Bob calcule $y_1^{x_2} = (a^{x_1})^{x_2} = a^{x_1 x_2} \pmod{p}$ et appelle ce nombre K, la clé secrète à partager avec Alice.

Tab 1.4 : étapes d'échange de la clé secret selon Diffie-Hellman.

A la fin du protocole, Alice et Bob sont donc en possession d'une même clé secrète K, qu'ils ne se sont pas échangés directement. Si quelqu'un a espionné leurs

conversations, il connait p, a, y1p, a, y1 et y2y2. Il ne peut pas retrouver KK comme le font Alice ou Bob, car il lui manque toujours l'une des informations nécessaires, à savoir x1x1 ou x2x2. Et il ne peut pas revers x1x1 connaissant y=ax1 (mod p) y=ax1 (mod p), aa et pp, puisque la résolution du logarithme discret est un problème difficile. [a7]

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

Inconvénients

La découverte de Diffie et Hellman est une grande révolution dans le monde de cryptographie. Le problème d'échange des clés est enfin résolu. Seulement dans ce protocole reconnaît un défaut d'où il exige la simultanéité des actions d'Alice et de Bob. Si Alice veut envoyer un e-mail à Bob alors que celui-ci dort ou n'est simplement pas connecté, elle ne pourra pas le faire immédiatement. C'est pourquoi ce protocole fut en réalité très vite supplanté par les méthodes de chiffrement à clé publique du type RSA, pour lesquels on met à la disposition de tout le monde une clé publique. Toutefois, il est utilisé pour les problèmes d'appariement de deux objets dans la technologie Bluetooth. [a7].

1.5.2 **RSA (Rivest, Shamir, Adleman)** [12]

Le chiffrement RSA est un algorithme de cryptographie asymétrique nommée selon ses trois inventeurs, souvent utilisé dans la vente en ligne, afin d'échanger les données confidentielles sur Internet. Il a été inventé en 1977 par Ronald Rivest, Adi Shamir et Leonard Adleman. Ce système cryptographique à clé publique est en fait basé sur deux clés **:**

- Une clé publique: pouvant être distribué librement, c'est le cadenas ouvert.
- Une clé secrète : connue uniquement du receveur, c'est le cadenas fermé.

Principe

Au début :

- Il est facile de fabriquer de grands nombres premiers p et q (+- 100 chiffres).
- Etant donné un nombre entier n = p.q, il est très difficile de retrouver les facteurs p et q.

a) Création des clés :

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

b) *Chiffrement* :

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

$C=M^e \bmod n$

c) Déchiffrement :

Il s'agit de calculer la fonction réciproque

$M=C^d \bmod n$ (tel que e.d =1 mod [(p-1) (q-1)]). [a8]

Exemple

Chiffrer BONJOUR

Etape 1 : Alice crée ses clés :

- La clé secrète : **p = 53, q = 97** (Note : en réalité, p et q devraient comporter plus de 100 chiffres !)
- La clé publique : **e = 7 (premier avec 52*96), n = 53*97 = 5141**

Etape 2 : Alice diffuse sa clé publique (par exemple, dans un annuaire).

Etape 3 : Bob ayant trouvé le couple **(n, e)**, il sait qu'il doit l'utiliser pour chiffrer son message. Il va tout d'abord remplacer chaque lettre du mot BONJOUR par le nombre correspondant à sa position dans l'alphabet:

B = 2, O = 15, N = 14, J = 10, U = 21, R = 18

BONJOUR= 2 15 14 10 15 21 18

Etape 4 : Bob découpe son message chiffré en blocs de même longueur représentant chacun un nombre plus petit que **n**. Cette opération est essentielle, car si on ne faisait pas des blocs assez longs (par exemple, si on laissait des blocs de 2 chiffres), on retomberait sur un simple chiffre de substitution que l'on pourrait attaquer par l'analyse des fréquences**.**

BONJOUR =002 151 410 152 118

Etape5 : Bob chiffre chacun des blocs que l'on note **B** par la transformation **$C = B^e \bmod n$** (où C est le bloc chiffré) :

$C_1 = 2^7 \bmod 5141 = 128$

$C_2 = 151^7 \bmod 5141 = 800$

$C_3 = 410^7 \bmod 5141 = 3761$

$C_4 = 152^7 \bmod 5141 = 660$

$C_5 = 118^7 \bmod 5141 = 204$

On obtient donc le message chiffré C : 128 800 3761 660 204.

1.6 Conclusion

Ce chapitre est une introduction globale sur la cryptographie d'où on a fait la comparaison entre les deux classes importantes des algorithmes de chiffrement symétrique à clé secrète et le cryptage asymétrique à clé publique ainsi les différences qui existent entre eux.

Le suivant chapitre va se baser sur les différents formats des fichiers audio.

Chapitre 2

Les formats des fichiers Audio

2.1 Introduction

Le son est une vibration mécanique dans l'air d'où elle se propage sous forme d'onde longitudinale grâce à la déformation élastique du fluide les êtres vivants le reconnaissent grâce au sens d'ouïe.

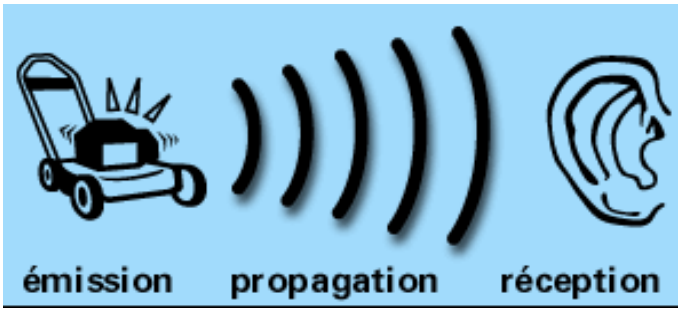


Fig2.1 : Propagation du son.

Ce phénomène est relié au temps d'autant plus il joue un rôle fondamental, il représente la variation de pression ainsi l'information sonore de variation de cette variation. Ce sonore dépend à plusieurs égards du temps il existe des relations étroites entre l'espace et le temps, tant dans l'étude du son que dans sa perception. Plus la pression acoustique est grande, plus le volume sonore est important.

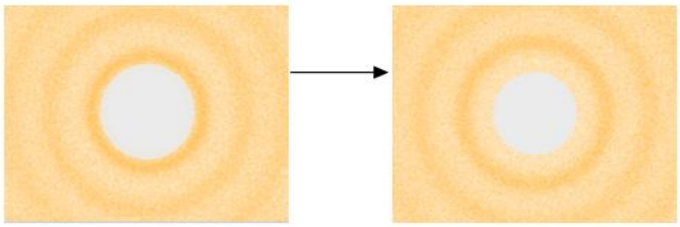


Fig2.2 : Propagation d'onde sphérique de pression dans un fluide.

NB : souvent on reconnaît trois types de son : la parole, le bruit et la musique.

2.2 Caractéristiques de son

Un son est défini par 3 paramètres : ***la Hauteur, le Volume et le Timbre***.

2.2.1 La hauteur (son grave/aigu) :

La hauteur est la fréquence de vibration de l'air, elle s'exprime en Hertz (Hz) (nombre de vibrations par seconde) et Elle permet de distinguer entre les :

- ✓ sons graves (à basses fréquences).
- ✓ Sons aigus (à hautes fréquences).

Il est clair que la qualité d'un son dépend de sa numérisation, et en particulier, de sa fréquence.

L'oreille humaine peut percevoir généralement les fréquences comprises entre 20 Hz et 20 000 Hz.[3]

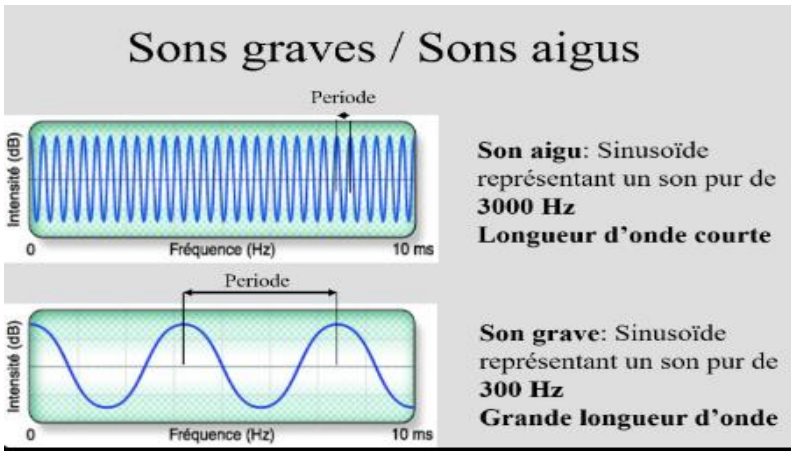


Fig2.3 : La taille de la période fixe de la hauteur du son.

2.2.2Le volume (ou intensité sonore)

Le volume est la force avec laquelle l'air frappe le tympan qui est une membrane au bout du conduit auditif de notre oreille, c'est aussi la hauteur de l'amplitude du signal.

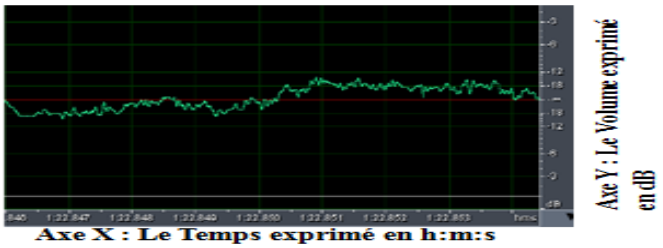


Fig2.4 : graphe de l'intensité sonore**.**

Exemple

L'intensité sonore

- d'un avion au décollage est de 160 db.
- Le coup de feu, seuil de douleur est de 140 db.
- Les bruits extérieurs la nuit est de 20 db.

2.2.3 Le Timbre (ou qualité de la sensation auditive)

C'est la caractéristique qui permet d'identifier un son d'une façon unique.

Afin de différencier les sons de même hauteur et même volume, le timbre dépend :

- Du nombre de composantes (harmoniques).
- De leurs intensités relatives.

- De la nature du corps vibratoire (peau, bois, métal..).
- Etc. [3]

Fig2.5 : graphe de qualité de la sensation auditive.

2.3 Définition d'un format de fichier audio

Un format de fichier audio est un format de données souvent utilisé en informatique afin de stocker des sons, des musiques ou bien de la voix sous forme numérique, il existe plusieurs standards certains s'appliquent au moment de production, de stockage ou à l'instant de diffusion, il y' a d'autres qui utilisent des algorithmes de compression de données ou de débit dont ils se basent uniquement sur le principe de diffusion.

2.4 Type de formats

Il existe deux types de formats le format de fichier et le codec la question qui se pose est, comment peut-on distinguer entre eux ?

Dans un codec le codage et le décodage des données vont être effectué sur des data audio brutes tandis que les données elles-mêmes sont stockées dans un fichier avec un format spécifique fichier audio.

La plupart des formats de fichier audio peuvent être créés avec un de deux ou plusieurs codecs.

Il y a 3 catégories principales de formats :

2.4.1 Format sans compression de données

Le format audio non compressé PCM (Pulse Code Modulation) est globalement stocké sous forme de WAV ou AIFF, c'est deux formats flexibles conçus afin de stocker un taux fini d'échantillonnage ou de bitrates (bit rate). Ainsi qu'ils sont appropriés pour le stockage et la réalisation d'enregistrements originaux. [a19]

a) **RAW :** (Real Audio Wrapper) est un format audio utilisé pour représenter les données de son en modulation d'impulsion codée sans en-tête ni métadonnées. [a19]

b) **WAV :** (ou WAVE), (WAVEform audio format) est une extension de fichiers audio, il s'agit d'un conteneur capable de recevoir des formats variés. Il est basé sur le format de fichier RIFF lequel est semblable au format IFF. Mono ou stéréo, il a été mis au point par Microsoft et IBM .Le suffixe des fichiers créés est. **Wav. [a19]**

c) **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. [a19]

d) **AIFF :** est un format de stockage de sons sur les ordinateurs d'Appel. C'est l'équivalent du format WAV dans le monde Macintoch. Les résolutions 8, 16, 20, 24 et 32 bits (à virgule flottante) sont acceptées. Le suffixe des fichiers créés est. **Aif [a17]**

e) **CAF :** (*Core 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. [a17]

2.4.2 Les formats audio compressés sans perte

Dans ce type de compression qui est dit aussi lossless, on utilise un algorithme dont il peut toujours retourner les données d'origine. Dans l'absolu, il existe toujours un fichier d'origine tel que l'algorithme ne ferait pas gagner d'espace disque.

Le principe de la compression sans perte est de diviser la taille des fichiers en deux ou trois, cette méthode est peu utilisée vu qu'elle est très faible en comparaison avec les algorithmes de compression avec perte (ce qui est un gros handicap pour les échanges de fichiers), ainsi que le temps de calcul est assez grand. Aucun standard n'a donc suffisamment convaincu pour devenir universellement lisible. [a19]

a) **ATRAC :** (Adaptive Transform Acoustic Coding) est une technique de compression audio avec et sans pertes développée par Sony en 1992. Ce format a subi plusieurs évolutions : ATRAC3, ATRAC3plus (familièrement écrit ATRAC3+) et ATRAC Advanced Lossless se sont succédé respectivement en 1999, 2002 et 2006. [a19]

b) **FLAC** : (Free Lossless Audio Codec), est un format libre. Maintenu par la fondation Xiph.org, il est apprécié pour conserver la qualité des fichiers sonores originaux en alternative aux formats de compression avec perte type **mp3**. **[a19]**

2.4.3 Les formats audio compressés avec perte

Ce type de compression est dit aussi lossy permet d'utiliser des algorithmes spécifiques afin de déterminer des transformations en simplifie la représentation du son en gardant le sens des fichiers de sort qu'une oreille humaine peut comprendre la contenue. Le but de ce format et se diminue la taille du fichier en éliminant les nuances perçues comme la moins utile. L'élimination est définitive, créer un fichier dans un format de haute qualité à partir d'un fichier compressé avec perte ne sert strictement à rien.

a) **MP3 :** ou (MPEG-1 Layer III), est l'abréviation de MPEG-1/2 Audio Layer 3. Cet algorithme est conçue en 1987 par L'ISO, On reconnait que La couche (Layer) III la plus complexe vu qu'elle est dédiée à des applications nécessitant des débits faibles (128 Kbits/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 Kib/s (V5) permet d'obtenir une qualité comparable au AAC encodé à 48 kbit/s.

   Le suffixe des fichiers créés est. mp3 [a19]

   Type de compression : constant ou variable (VBR).

b) **mp3PRO :** combine l'algorithme MP3 et un système améliorant la qualité des fichiers comprimés appelé (en)SBR pour Spectral Bandwidth Replication. C'est un travail collaboré entre Thomson Multimédia et l'Institut Fraunhofer d'où il a été publié la fin de 2001 ; le fichier MP3pro de 64 Kbit/s a une qualité équivalente qu'un MP3 à 128 Kbit/s.

   Le suffixe des fichiers créés est. mp3

c) **AC3 :** Audio Coding 3 ou Codage Audio 3,il est connu aussi sous le nom de Dolby Digital., une extension qui est utilisé pour les fichiers audio de son surround. Elle a été inventée en 1987 par les Laboratoires Dolby afin d'être utilisée sur les DVD, les lecteurs Blu-ray, et les systèmes domestiques de programmation et de divertissement en haute définition (HDTV). Le format AC3 peut inclure jusqu'à 6 canaux sonores. Les 5 canaux dont on se sert le plus communément sont destinés aux enceintes à gamme normale (allant de 20 à 20 000Hz) plus un canal dédié aux basses fréquences (de 20 à 120Hz) pour le caisson des graves, dit subwoofer. Il s'agit respectivement de l'avant gauche, de l'avant droit, du centre, de l'arrière gauche et de l'arrière droit, et enfin du canal à ultra basses fréquences dit 5.1, ce qui est la configuration de son audio surround (ou ambiophonique) standard utilisée le plus fréquemment dans les salles de cinéma et les systèmes de « home cinéma ».

2.4.4 Formats multipistes

Les formats multipistes sont une innovation récente. Ils permettent d'encapsuler dans un fichier des différentes pistes sonores, dont ils peuvent être combinés par l'utilisateur dans les proportions qui lui conviennent. Le but de ce format est de proposer, pour un morceau de musique, la piste correspondant à chaque instrument (et la voix) de manière séparée. Ce qui aide L'utilisateur a créé sa propre version.

a) **IKlax :** Format de la catégorie des conteneurs, il a été développé par la société iKlax Media et le LaBRI (Laboratoire Bordelais de Recherche Informatique).

Extension : .iklax.

Avantages : L'iKlax permet d'organiser la musique en différents groupes et de leur appliquer des contraintes.

Inconvénients : C'est un format peu connu et peu répandu.

b) **U-MYX :** Format multipiste créé par l'entreprise du même nom, il a été Utilisé pour fournir des morceaux en tant que bonus.

Avantages : Ce format permet d'augmenter ou d'atténuer le volume de chacune des pistes. On peut obtenir une version instrumentale de bien meilleure qualité qu'avec certains logiciels de filtrage.

Inconvénients : Il n'est lisible que par une application dédiée, développée par le même éditeur. Ce n'est pas un format standard, cela implique de fournir constamment le logiciel de lecture avec les fichiers. Les logiciels permettant son enregistrement ne sont pas distribués publiquement.[a20]

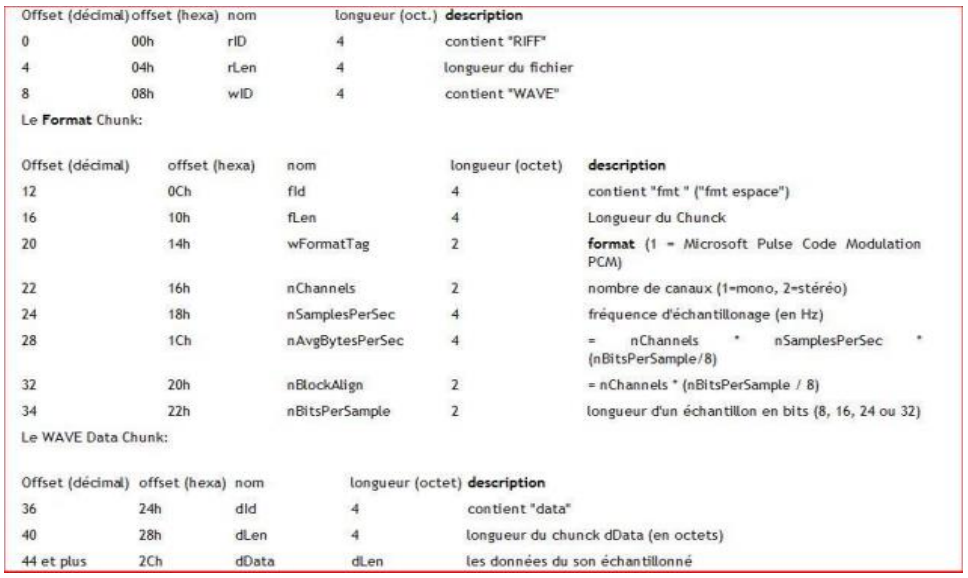
a) **MXP4** : MXP4 est un format conteneur, multipiste dans lequel les pistes sont au format Ogg.

Extension : .mpx4.

Avantage : Ce format permet de facilement passer d'une version acoustique à une version électrique pour une même chanson.

Inconvénient : C'est un format très peu connu et encore peu utilisé. [a20]

2.5 Structure générale d'un fichier wav



Offset (décimal)	offset (hexa)	nom	longueur (oct.)	**description**
0	00h	rID	4	contient "RIFF"
4	04h	rLen	4	longueur du fichier
8	08h	wID	4	contient "WAVE"

Le **Format** Chunk:

Offset (décimal)	offset (hexa)	nom	longueur (octet)	**description**
12	0Ch	fId	4	contient "fmt " ("fmt espace")
16	10h	fLen	4	Longueur du Chunck
20	14h	wFormatTag	2	**format** (1 = Microsoft Pulse Code Modulation PCM)
22	16h	nChannels	2	nombre de canaux (1=mono, 2=stéréo)
24	18h	nSamplesPerSec	4	fréquence d'échantillonage (en Hz)
28	1Ch	nAvgBytesPerSec	4	= nChannels * nSamplesPerSec * (nBitsPerSample/8)
32	20h	nBlockAlign	2	= nChannels * (nBitsPerSample / 8)
34	22h	nBitsPerSample	2	longueur d'un échantillon en bits (8, 16, 24 ou 32)

Le WAVE Data Chunk:

Offset (décimal)	offset (hexa)	nom	longueur (octet)	**description**
36	24h	dId	4	contient "data"
40	28h	dLen	4	longueur du chunck dData (en octets)
44 et plus	2Ch	dData	dLen	les données du son échantillonné

Fig2.7 : Structure générale d'un fichier wav.[a18]

2.6 Caractéristique de fichier wav

2.6.1 L'amplitude : Un sample se compose d'une courbe continue dont elle a une valeur bi-polaire (signée).le 1 er paramètre 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.

Son unité de grandeur est le décibel (dB);" une mesure logarithmique donnant le degré d'amplification d'une vibration."

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 embêtants à manipuler et qui sont très rarement utilisés. 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.[a18]

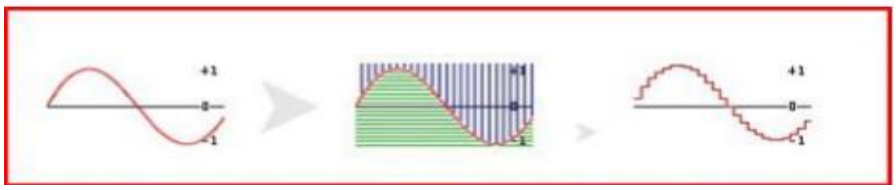


Fig2.8: amplitude, fréquence. [a18]

2.6.2 La fréquence : En bleu (fig 2.8), 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.[a18]

2.6.3 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. À partir de la taille du fichier (donnée par file Size), et les caractéristiques du sample, il est facile de calculer le temps total en seconde (ou ms) de lecture d'un sample. [a18]

2.6.4 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. [a18]

2.6.5 Fichier wav en hexadécimal

Le fichier wav se compose de trois blocs (riff, fmt et data), la structure suivante nous montre en détaille cette architecture.

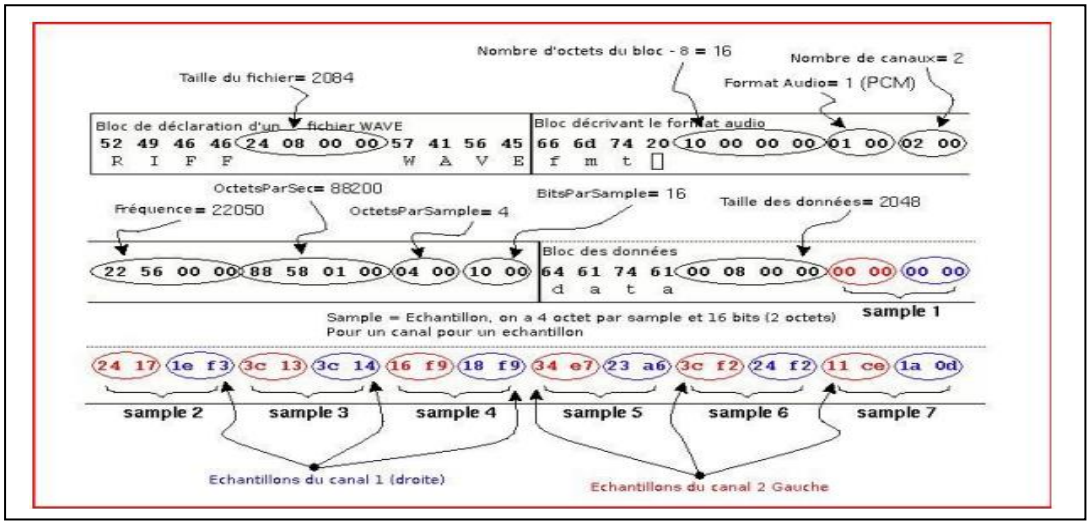


Fig2.9 : Fichier wav en hexadécimal. [a18]

2.7 Conclusion

Le but de ce chapitre est d'étudier toute sorte de fichier audio, les différents types ainsi leurs utilités en se basant sur l'architecture et la structure du format wav.

Le chapitre ci-dessous vise les algorithmes de brouillage.

Chapitre 3

Introduction aux architectures Client/Serveur

3.1 Introduction

L'environnement client/serveur désigne un mode de communication organisé par l'intermédiaire d'un réseau et d'un interface Web entre plusieurs ordinateurs. " cela signifie que des machines clientes (machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacités d'entrées-sorties, qui leur fournit des services. Lequels services sont exploités par des programmes, appelés programmes clients, s'exécutant sur les machines clientes." De nombreuses applications fonctionnent selon un environnement client/serveur, cela signifie que des machines clientes (des machines faisant partie du réseau) contactent un serveur, une machine généralement très puissante en terme de capacités d'entrée-sortie, qui leur fournit des services. [a21]

3.2 Définition des systèmes répartis

Les systèmes répartis sont, dans leur définition la plus simpliste, des logiciels dont l'exécution se déroulent sur un ensemble d'ordinateurs distribués géographiquement et reliés entre eux par des interconnexions réseau. [4][5]

3.3 Définition d'un serveur

Un serveur informatique est un dispositif informatique matériel ou logiciel qui offre des services, à différents clients. Les services les plus courants sont :

- ✓ le partage de fichiers.
- ✓ l'accès aux informations du World Wide Web.
- ✓ le courrier électronique.
- ✓ le commerce électronique.
- ✓ le stockage en base de données.
- ✓ le jeu et la mise à disposition de logiciels applicatifs.

Autre signification :

- ➢ Un serveur est une machine physique.
- ➢ Connectée à un réseau.
- ➢ Qui stocke des données.
- ➢ Effectue diverses opérations en réponse à une requête. [a16]

3.3.1 Les types de serveurs

- ➢ **Serveur itératif :** Permet de traiter une seule requête selon son implémentation.

- **Un serveur concourant** : Un serveur concourant désigne une implémentation capable de gérer plusieurs tâches en apparence simultanées. Attention, cette fonctionnalité n'implique pas nécessairement que ces tâches concourantes doivent toutes s'exécuter en parallèle. [a8]

3.4 Définition d'un Client

Un client informatique est un logiciel qui envoie des requêtes à un serveur, Il peut être automatique ou manipulé par un utilisateur. Par extension, on parle de client pour l'ordinateur personnel et jusqu'à l'utilisateur.[a16]

3.4.1 Les types de client

- ✓ **Client "léger" :** Le poste client accéde 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 sa demande qu'il a formulée. (Appelée : "requête»).
- ✓ **Client "lourd" :** Le poste client doit comporter un système d'exploitation capable d'exécuter en local une partie des traitements. Car 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.
- ✓ **Client "riche" :** Un interface graphique plus évolué permet de mettre en œuvre des fonctionnalités comparables à celles d'un client "lourd". Les traitements sont effectués majoritairement sur le serveur, la réponse "semi-finie" étant envoyée au poste client, où le client "riche" est capable de la finaliser et de la présenter. [a14]

3.5 Modèle client / serveur

- Complète les systèmes distribués**.**
- Fournit l'intégration des données et des services.
- Le traitement des demandes fourni par plusieurs niveaux :
    1. Serveur de base de données.
    2. Serveur d'application.
    3. Station de travail PC (client).

3.6 Les architectures Client/serveur

Dans le monde d'internet l'architecture client/serveur est très répandue pour but de servir les pages web vue qu'elle permet de les servir.

Parmi ses avantages :

- ✓ Elle est simple à sécuriser.
- ✓ Elle centralise l'information.
- ✓ Elle est très sensible à la charge.

Fig 3.1 :l'architecture C/S.

Interprétation du schéma (figure)

Soit A le client dont son rôle est le suivent :

- ➢ Maître.
- ➢ Soumet les requêtes.
- ➢ Interprète les réponses.

Soit B le serveur dont son rôle est le suivent :

- ➢ Passif, en attente.
- ➢ Répond aux requêtes.

3.7 Triptyque d'une application

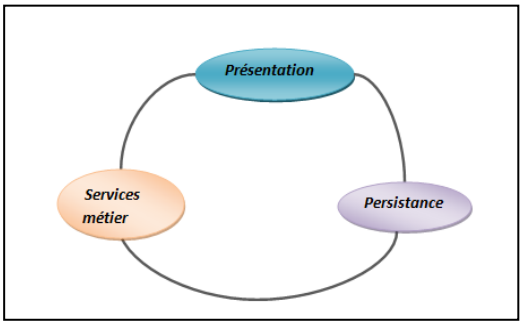


Fig 3.2 : Triptyque d'une application.

Présentation

- Interface utilisateur pour interagir avec l'application.
- Interface classique ; traitement de texte (type GUI).
- Une interface Web qui est plus légère.

Persistance

Permet d'enregistrer le support physique de l'application pour :

- Fichiers (binaires, XML, ...).
- Base de données :
    - Simple.
    - Avec redondance pour fiabilité.
    - Multiples : fédération de bases de données ...

Services métier :

- La partie qui Intègre le logique métier :
    - Exemple: un document est composé de sections, elles-mêmes composaient de sous-sections ...
- Offre les services aux utilisateurs :
    - Exemple: créer un document, le modifier, ajouter des sections, l'enregistrer ...
- Vise la partie applicative. [a10]

Ces trois parties sont intégrées et coopérées afin de fonctionner l'application.

Différents types de modèles existent :

3.7.1 Principe

La division des parties se fait **:**

- Les tiers peuvent être exécutés sur des machines différentes.
- Certains tiers peuvent être sous découpés.
- De nombreuses variantes de placement des tiers et de leur distribution.

a) Architecture 1 – tiers : dit aussi (Modèle centralisé), se modèle permet d'intégrer les trois parties en une seule machine.

Fig 3.3 : Architecture 1 – tiers.

b) Architecture 2– tiers : l'architectures client / serveur à deux niveaux a 2 composants essentiels :

- Un PC client.
- Un serveur de base de données.

Considérations à 2 niveaux:

➢ **Le programme client accède directement à la base de données :**

- Nécessite un changement de code pour transférer vers une base de données différente.
- Goulot d'étranglement potentiel pour les demandes de données.
- Volume de trafic élevé dû à l'envoi de données.

➢ **Le programme client exécute la logique d'application :**

- Limité par la capacité de traitement du poste de travail client (mémoire, processeur).
- Requiert que le code de l'application soit distribué à chaque poste de travail client.

❖ **Client : présentation + applicatif:**

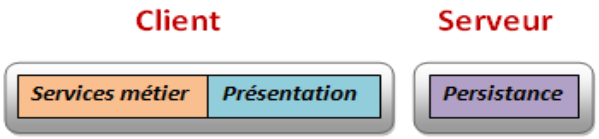


Fig 3.4 : Architecture 2– tiers (cas 1)

❖ **Terminal : client intègre un minimum de la partie présentation :**

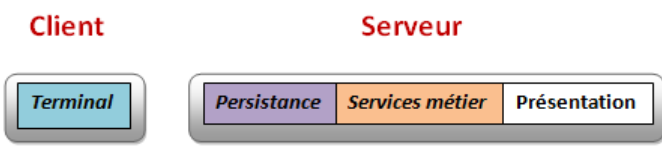


Fig 3.5: Architecture 2– tiers (cas 2)

❖ **Serveur : applicatif + gestion données :**

Fig 3.6: Architecture 2– tiers (cas 3)

❖ **Applicatif : découpé entre client et serveur :**

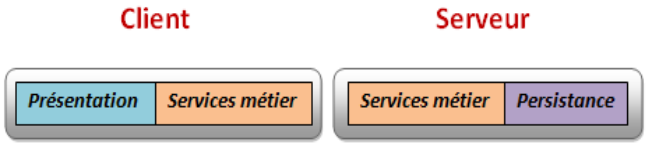


Fig 3.7 : Architecture 2– tiers (cas 4)

❖ **Avantages et inconvénients :**

Avantages	Inconvénients
Problèmes de développement: ➢ Structure simple. ➢ Facile à installer et à entretenir.	**Problèmes de développement:** ➢ Règles d'application complexes difficiles à implémenter dans le serveur de base de données – nécessite plus de code pour le client. ➢ Règles d'application complexes difficiles à mettre en œuvre dans le client et ont pauvre performance. ➢ Les modifications du logique métier ne sont pas appliqué automatiquement par un serveur les modifications nécessitent un nouveau logiciel côté client être distribué et installé. ➢ Non portable vers un autre serveur de base de données plateformes.
Performance: ➢ Performance adéquate pour un environnement de volume faible à moyen. ➢ Le logique métier et la base de données sont physiquement proches, ce qui fournit plus de performance**.**	**Performance:** ➢ Performance inadéquate pour les environnements à volume moyennes à élevé, vu qu'un serveur de base de données est nécessaire pour effectuer la logique d'entreprise. Cela ralentit la base de données opérations sur le serveur de base de données.

c) **Architecture 3– tiers :** Les architectures client-serveur à 3 niveaux ont 3 composants essentiels:

1. Un PC client.
2. Un serveur d'application.
3. Un serveur de base de données.

- **Considérations à 3 niveaux:**

1. **Le programme client contient uniquement la partie présentation :**
   - ✓ Peu de ressources ont besoin de poste de travail client.
   - ✓ Aucune ou modification va être faite sur client dans le cas de changement d'emplacement de la base de données.
   - ✓ Moins de code a distribué aux postes de travail clients.
2. **Un serveur gère plusieurs demandes de clients :**
   - ✓ Plus de ressources disponibles pour le programme serveur.
   - ✓ Réduit le trafic de données sur le réseau.

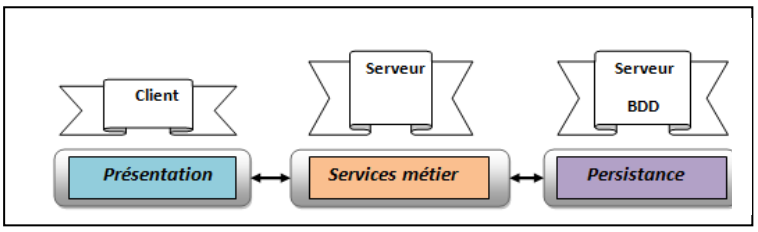


Fig3.8: Architecture 3– tiers

d) **Architecture n– tiers :**

- ❖ Rajoute des étages / couches en plus.
- ❖ La couche applicative n'est pas monolithique.
  - ✓ Peut s'appuyer et interagir avec d'autres services.
  - ✓ Composition horizontale :
    - Service métier utilise d'autres services métiers
  - ✓ Composition verticale :
    - Les services métiers peuvent aussi s'appuyer sur des services techniques : Sécurité, Transaction ...

❖ Chaque service correspond à une couche d'où le terme de N-tiers. [a10].

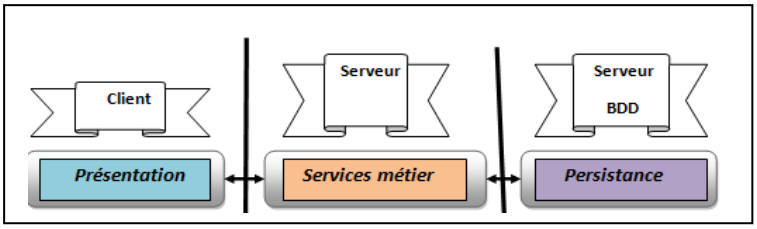


Fig3.9: Architecture n– tiers

3.8 Communication entre client et serveur

3.8.1 Notions de ports et protocoles

La communication entre un serveur et un client se fait avec un langage particulier dit protocole, ainsi que deux machines communiquent entre eux avec des adresses IP.de plus, il y'a la notion de port d'où on aura une disponibilité de plusieurs services

Chaque paquet réseau contient :

- ✓ Une adresse IP pour la machine d'origine : cas d'une requête (machine client).
- ✓ Une adresse IP pour la machine destination : pour le serveur.
- ✓ Numéro de port.

3.8.2 Définition du Port

Souvent, le mot port est dédié pour les sockets. D'ou elles jouent un rôle d'un d'identifiant unique dans un réseau donné résultant de la concaténation de l'adresse internet et du numéro de port. Le but d'un port est d'identifier localement un processus.

- ➢ Les ports 0 à 1023 sont les «**ports reconnus**» ou réservés («**Well Known Ports**»).
- ➢ Les ports 1024 à 49151 sont appelés «**ports enregistrés**» («**Registered Ports**»).
- ➢ Les ports 49152 à 65535 sont les «**ports dynamiques et/ou privés**» («**Dynamics and/or Private Ports**».[a12]

Port	Service ou Application
21	FTP
23	Telnet
25	SMTP
53	Domain Name System
63	Whois
70	Gopher
79	Finger
80	HTTP
110	POP3
119	NNTP

Tab 3.1 : Liste des services et les ports.

3.8.3 Les protocoles

Pour pouvoir envoyer de l'information entre deux équipements, les deux appareils doivent parler le même langage. Ce langage est appelé protocole.

Les protocoles qui apparaissent dans la couche application du model TCP/IP sont:

- File Transfer Protocol (FTP).
- HyperText Transfer Protocol (HTTP).
- Simple Mail Transfer Protocol (smtp).
- Domain Name Service (DNS).
- Trivial File Transfer Protocol (TFTP).

Les protocoles de la couche transport sont:

- Transport Control Protocol (TCP).
- User Datagram Protocol (UDP).

Les protocoles de la couche Internet sont:

- Internet Protocol (IP).

Le protocole le plus souvent utilisé dans la couche d'accès réseau est:

- Ethernet. [a13]

3.8.4 Middleware

Simplifie le développement et l'administration d'application à 3 niveaux en fournissant une couche de serveur d'applications supplémentaires pour gérer la communication entre les composants.

a) Caractéristiques du middleware:

- Simplifie le partitionnement du traitement des applications entre clients et serveurs.
- Gère les transactions distribuées entre plusieurs bases de données.

- Communique avec des produits de base de données hétérogènes au sein d'une même application.
- Prend en charge la scalabilité de l'application.
- Prend en charge la hiérarchisation des demandes de service, l'équilibrage de la charge, le routage et la mise en file d'attente en fonction des données.

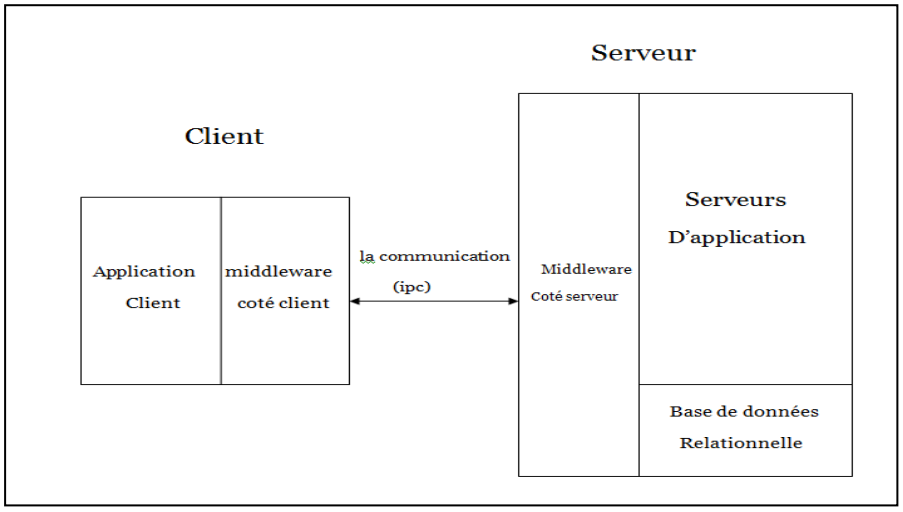


Fig 3.10 : Middleware.

3.9 Communication interprocessus

- ✓ Une Base pour le client / serveur.
- ✓ Le processus client communique avec le processus serveur.
- ✓ Chaque processus exécute des fonctions distinctes.
- ✓ Les données sont transmises entre les processus à l'aide de fonctions IPC.

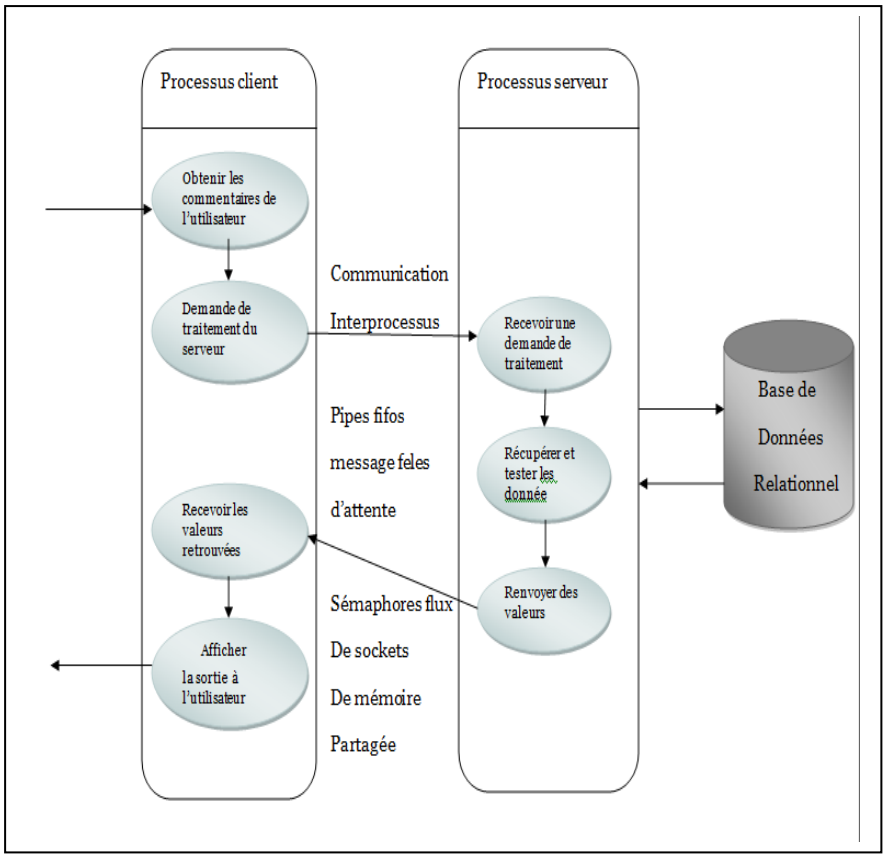


Fig 3.11 : Communication interprocessus

3.10 Modes de fonctionnement client/serveur

Il existe deux modes :

- **Mode connecté (comparable à une communication téléphonique) TCP :**
  On reconnaît avec ce type de protocole, une connexion durable qui relie les deux pairs tels que les adresses peuvent être négligeables.
- **Mode non connecté (analogue à une communication par courrier) UDP :**
  Contrairement au protocole TCP, l'UDP exige la nécessité de l'adresse de destination à chaque envoi ainsi que qu'il y'a pas d'accusé de réception des data.

3.11 Sockets

Est un point d'accès entre les 2 applications du réseau, dont elles peuvent échanger de données en utilisant le mécanisme d'E/S (java.io).

Différents types de sockets

- **Stream Sockets (TCP) :**

- Permet d'établir la communication (mode connecté).
- Dans le cas d'interruption de connexion : application sera informé.

❖ **Datagram Sockets (UDP) :**

- Communication se fait en mode non connecté.
- Les données seront envoyées sous forme de paquets.

3.12 Avantages du modèle Client/serveur

✓ Divise le traitement des applications sur plusieurs machines:
  - Les données et fonctions non critiques sont traitées sur le client.
  - Les fonctions critiques sont traitées sur le serveur.

✓ Optimise les postes de travail clients pour la saisie de données et la présentation.

✓ Optimise le serveur pour le traitement et le stockage des données (par exemple, une grande quantité de mémoire et d'espace disque).

✓ Échelles horizontales : Plusieurs serveurs, chaque serveur ayant des capacités et une puissance de traitement, peuvent être ajoutées pour répartir la charge de traitement.

✓ Échelles verticalement - Peut être déplacé vers des machines plus puissantes, tel qu'un mini-ordinateur ou un ordinateur central peut avoir des performances vastes sur le système.

✓ Réduit la réplication des données - Les données stockées sur les serveurs au lieu de chaque client réduisent la quantité de réplication de données pour l'application.

3.13 Architecture du modèle P2P

3.13.1 Définition du P2P

✓ Peer-to-peer signifie littéralement pair à pair. Ce concept introduit ainsi une relation d'égal à égal entre deux ordinateurs.

✓ Dans son essence, l'informatique pair à pair se définit comme le partage des ressources et des services par échange direct entre systèmes. Ces échanges peuvent porter sur les informations, les cycles de traitement, la mémoire cache ou encore le stockage sur disque des fichiers.

- ✓ Contrairement au modèle client / serveur, chaque système est une entité réseau complète qui remplit à la fois le rôle de serveur et celui de client. Avec le peer-to-peer, les ordinateurs personnels ont le droit de faire partie du réseau.
- ✓ Le peer-to-peer désigne donc une classe d'application qui tire parti des ressources matérielles ou humaines qui sont disponibles sur le réseau Internet. [a11]

3.13.2 Architecture décentralisée

Afin d'obtenir l'accès aux informations dans un réseau centralisé, il suffirait de se connecter au serveur. Contrairement à un réseau P2P il faudrait :

Contrairement aux réseaux centralisés, où il suffisait de se connecter au serveur pour avoir accès aux informations, afin d'accéder à une information décentralisée il faut :

- Apprendre la topologie du réseau sur lequel le client est connecté.
- Rechercher l'information sur tous les nœuds.
- Recevoir une réponse d'un nœud répondant aux critères.

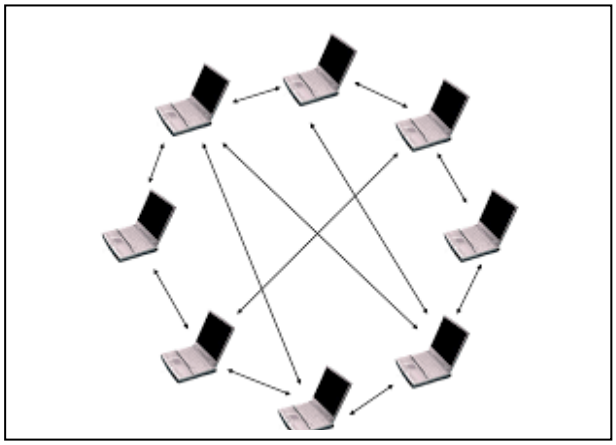


Fig3.12 : l'architecture P2P.

3.14 Client/serveur vs P2P

	Client-Serveur	Peer-to-Peer
De base	Il y a un serveur spécifique et des clients spécifiques connectés au serveur.	Le client et le serveur font le même travaille. chaque nœud agit en tant que client et serveur.
Service	Le client demande le service et le serveur offre le service.	Chaque nœud peut demander des services et peut également fournir des services.
La stabilité	Modèle Client-Serveur est plus stable et évolutif.	Peer-to Peer souffre si le nombre de pairs augmente dans le système.
Le coût	Le client-serveur est coûteux à implémenter.	Peer-to-peer sont moins chers à mettre en œuvre.
Coté Serveur	Lorsque plusieurs clients demandent les services simultanément, un serveur peut être encombré.	Comme les services sont fournis par plusieurs serveurs répartis dans le système peer-to-peer, un serveur n'est pas encombré.
Les données	Les données sont stockées dans un serveur centralisé.	Chaque pair a ses propres données.

Tab 3.2 : comparaison entre C/S et P2P.

3.15 Conclusion

Dans ce chapitre, nous avons intégré quelque notion primordiale sur le modèle Client/serveur tel que :

Les sockets et l'appel des procédures à distance, les protocoles, Les middlewares. Ce type de modèle préoccupe une grande importance dans les services des réseaux informatiques, c'est pour cela, nous avons inspiré de ce dernier pour réaliser notre étude.

Chapitre 4

L'implémentation de L'application

4.1Introduction

Ce chapitre permet d'englober le travail ainsi les outilles que j'ai utilisé (l'environnement du travail..).

L'application est programmée en java avec un EDI NetBeans.de plus je l'accompagne avec des captures d'écran de l'interface graphique.

Le but de mon étude est de faire une comparaison entre les différents algorithmes de chiffrement sur un fichier audio du type wav, cette comparaison sera au niveau de la qualité et la vitesse de l'opération de chiffrement et de déchiffrement.

J'ai utilisé un ordinateur Intel(R) core(Tm) i3-2330M CPU @ 2.20GHZ, 2200MHZ, 2 cœurs, avec une mémoire physique (RAM) de 4GO et un SE windows7.

4.2Paramètres de comparaison

La comparaison va se baser sur deux facteurs :

- ✓ La qualité du chiffrement.
- ✓ Le temps de chiffrement et de déchiffrement.
- ✓

4.3 Introduction sur le langage JAVA

JAVA est un langage développé par SUN **.**ce type de langage est interprété, il permet d'interpréter le programme compilé par un interpréteur puis l'exécuter par le système d'exploitation.

Java se compose des concepteurs dont ils sont :

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

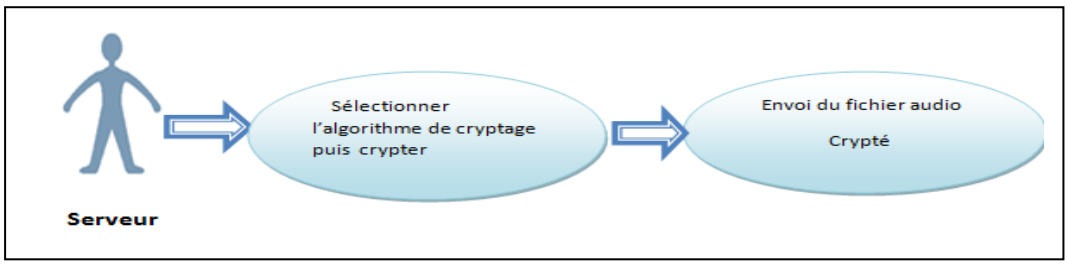
4.4 Fonctionnement de l'application

Le but de mon application est de crypter un fichier audio sécurisé sous l'extension wav selon des différents types d'algorithmes de cryptage**.** Cette application est basée sur le modèle client/serveur. Le serveur permet

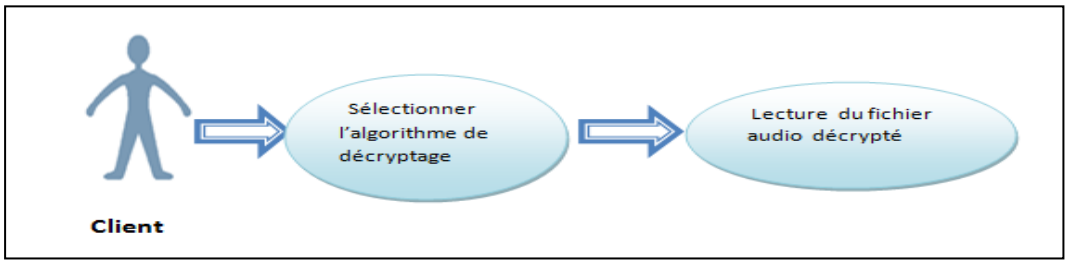
d'enregistrer ou de sélectionner un fichier audio, le crypter puis l'envoyer au client .à ce point-là, le client va faire l'opération inverse d'où il va le récupérer et le décrypter puis le lire.

Le choix d'algorithme va être acquit selon l'accord des deux membres tout on envoyant un message indicatif.

❖ **Interface Serveur**



❖ **Interface Client :**



4.5 Réalisation de l'application

Cette étude vise la performance et l'efficacité de chaque type d'algorithmes de cryptage appliqués pour chiffrer les fichiers audio wav.

❖ **Classe connexionCli**

La communication entre le client /Serveur se fait dans la classe connexionCli dont la méthode est nommée sous le nom connexion indiqué si dessus.

```
public connexionCli (String ip, int port){
    try {
      s = new Socket (ip, port);
      input = s.getInputStream ();
      output = s.getOutputStream ();
    }
```

❖ **Classe ServEnvoyer et envoyerCli**

L'objectif de ces deux classes est de transférer les messages secret et sécurisé afin de ce mètre d'accord ainsi qu'ils vont être dans le même sous réseau.

- **Classe recoiServ et recoiCli :**

Ces classes permettent de recevoir les messages sécurisés de l'autre paire ainsi qu'ils ont été dans le même sous réseau.

4.6. L'interface graphique de l'application

4.6.1 Serveur

Fig 4.1: interface serveur.

- **Le contenu du serveur**

a) Le lancement du serveur

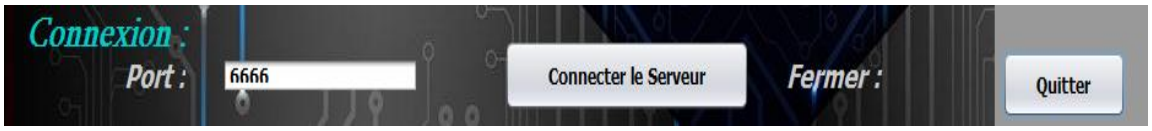


Fig *4.2*: lancement du serveur

Le bouton connecter le serveur: permet l'etablisement de connexion selon le port d'entrer.

Le bouton quitter: permet de fermer l'interface.

b) La partie chat

Fig4.3: le chat sécurisé

Le bouton Envoyer Msg: permet d'envoyer le message sécurisé.

c) L'enregistrement de l'audio

Fig 4.4: enregistrer le son

Le bouton enregistrer: permet d'enregister la voix sous le fichier wav.

Le bouton stoper: permet d'arrêter L'enregistrement.

Le bouton +: permet d'ajouter un fichier audio.

Le bouton ecouter: permet d'écouter le fichier audio.

Le bouton pause: permet d'arrêter temporairement le fichier audio.

Le bouton arrêter: permet D'arrêter définitivement la lecture du fichier audio.

d) La zone de chiffrement

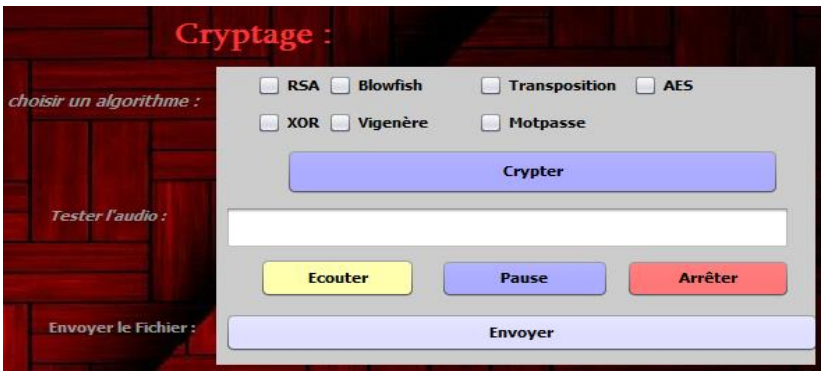


Fig *4.5:* la sélection de l'algorithme de cryptage.

Le bouton crypter: faire l'opération de cryptage sur le fichier audio selon le choix de l'algorithme de cryptage (vigenére, RSA, AES, Motdepasse, Blowfish, Transposition, XOR).

Le bouton ecouter: permet d'écouter le fichier audio crypté.

Le bouton pause: permet d'arrêter temporairement le fichier audio crypté.

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

Le bouton Envoyer: permet d'envoyer le fichier audio crypté au client**.**

4.6.2 Client

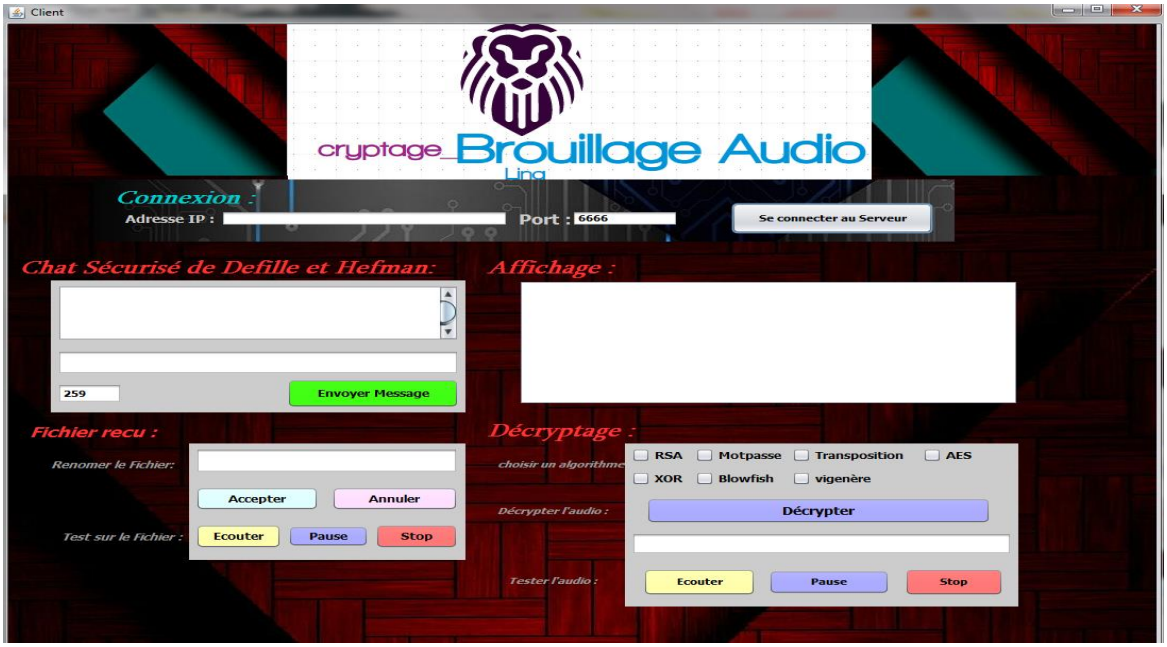


Fig 4.6: interface client.

- **Le contenu du Client**

a) Le lancement du Client

Fig 4.7: Établissement de connexion avec le serveur.

Le bouton se connecter au serveur: la communication avec le serveur selon son adresse et le numéro de port.

b) La partie de réception du fichier crypté

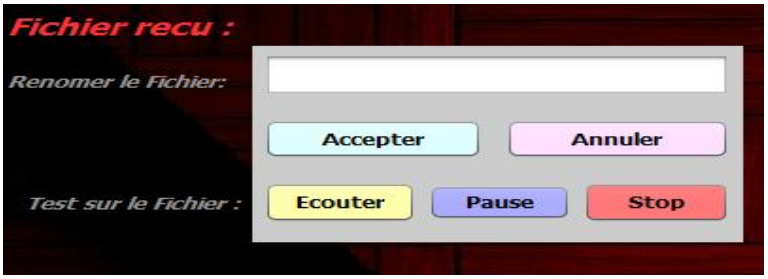


Fig 4.8: Réception et enregistrement du fichier audio**.**

Le bouton accepter*:* permet de récupérer et d'enregistrer le fichier audio qui a été reçu par le serveur.

Le bouton annuler: permet d'annuler la réception du fichier audio.

Le bouton ecouter: permet d'écouter le fichier audio.

Le bouton pause: permet d'arrêter temporairement le fichier audio.

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

c) La zone de déchiffrement

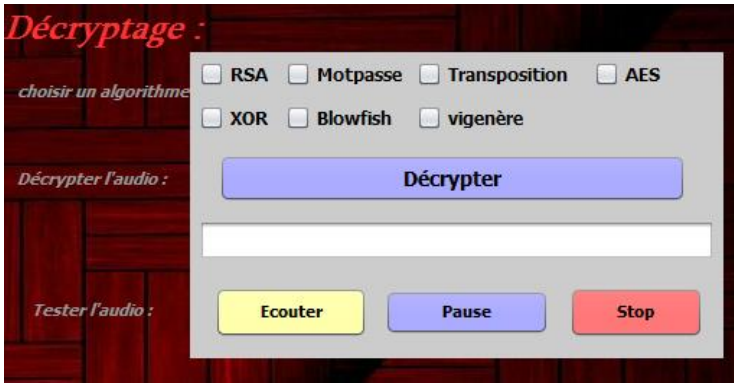


Fig 4.9: la sélection de l'algorithme de décryptage.

Le bouton décrypter: faire l'opération de décryptage sur le fichier audio selon le choix de l'algorithme de cryptage (vigenére, RSA, AES, Motdepasse, Blowfish, Transposition, XOR).

Le bouton ecouter: permet d'écouter le fichier audio décrypté.

Le bouton pause: permet d'arrêter temporairement le fichier audio décrypté.

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

- **Fenêtre de cryptage:**

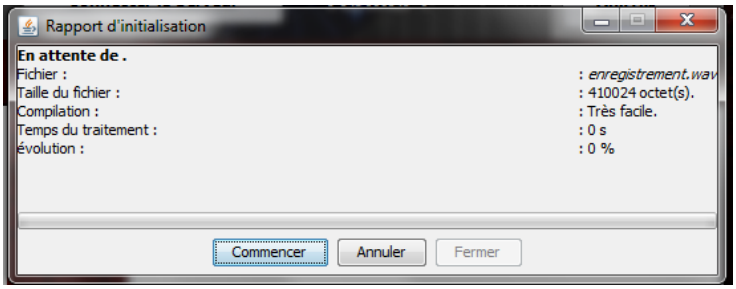


Fig 4.10: traitement de chiffrement et déchiffrement.

Le bouton commencer: permet de commencer le traitement.

Le bouton fermer: permet de fermer la fenêtre.

Le bouton arrêter: permet de suspender le traitement**.**

4.6.3 les exceptions:

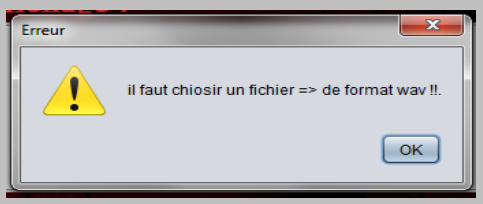


Fig A: Exception sur le choix de fichier.

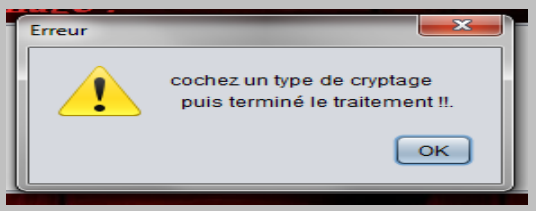


Fig B: Exception sur la sélection de l'algorithme de cryptage.

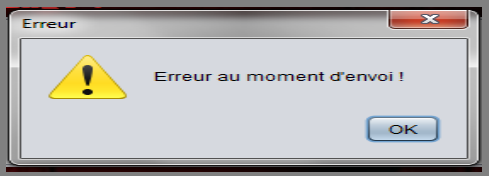


Fig C : Exception de l'envoi.

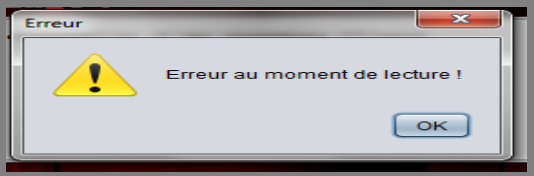


Fig D: Exception de la lecture.

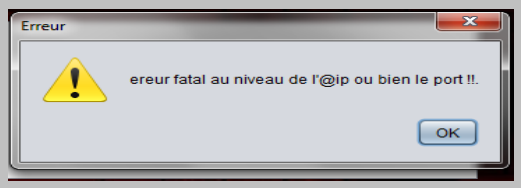


Fig E: Exception au niveau de port ou @ip.

Tab 4.1: les exceptions.

4.7 Résultat de chiffrement des différents algorithmes de cryptage en fonction du temps

Algo / Fichier (.wav)	Transposition	Vigenére	Xor	Blowfish	AES	Mot de passe	RSA	Taille des fichier
Audio 1	0.005 s	0.005s	0.004s	0.319s	0.46s	0.851s	2.025s	360 ko
Audio 2	0.007s	0.008s	0.005s	0.346s	0.491s	1.88s	2.737s	380ko

Tab 4.2: temps de chiffrement de l'audio selon les différents types d'algorithmes.

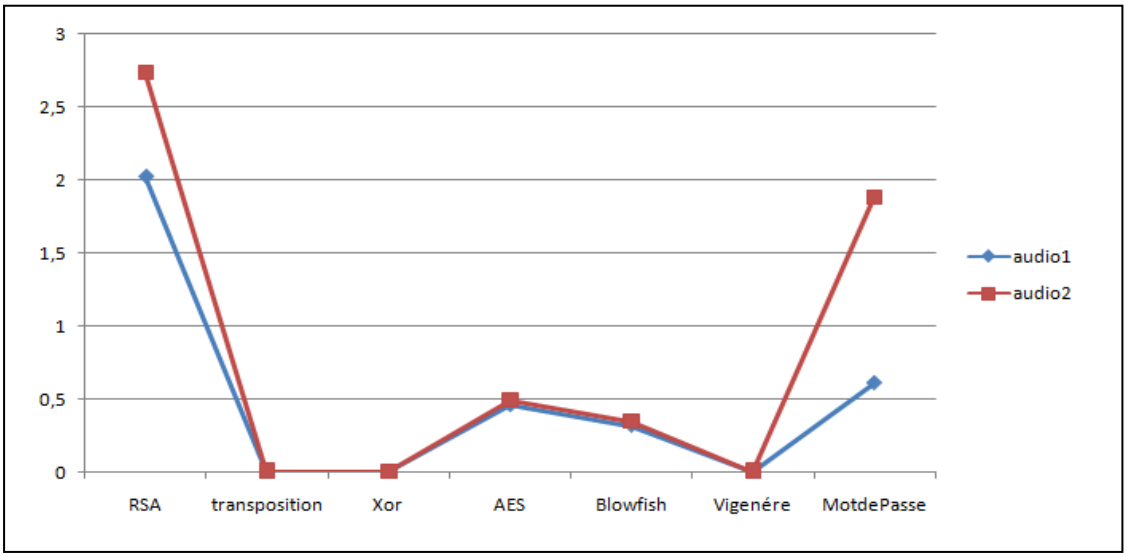


Fig 4.10: le temps de chiffrement des fichiers audio selon les algorithmes de cryptage.

Remarque:

D'après les courbes on remarque que:

- Le temps de chiffrement d'un fichier audio varie selon sa taille.

- Le temps de chiffrement augmente proportionnellement avec la taille du fichier.
- L'algorithme XOR est le plus rapide, en opposé le RSA à clé publique est le plus lent.

4.8 Résultat de déchiffrement des différents algorithmes de cryptage en fonction du temps

Algo / Fichier (.wav)	Transposition	Vigenére	Xor	Blowfish	AES	Mot de passe	RSA	Taille des fichier
Audio 1	**0.31 s**	**0.295s**	**0.202s**	**0.331s**	**0.418s**	**0.733s**	**3.054s**	**360 ko**
Audio 2	**0.318s**	**0.325s**	**0.286s**	**0.356s**	**0.385s**	**1.167s**	**4.179s**	**380ko**

Tab 4.3: temps de déchiffrement de l'audio selon les différents types d'algorithmes**.**

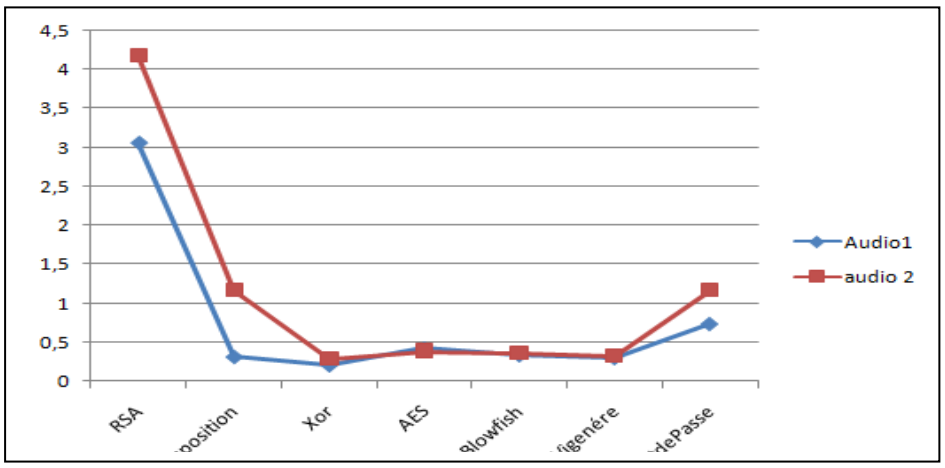


Fig 4.11: le temps de déchiffrement des fichiers audio selon les algorithmes de cryptage.

Remarque:

- Le traitement de déchiffrement est plus lourd que celui de chiffrement.
- Le temps de chiffrement est plus lent que celui du déchiffrement.

4.9 Résultats de la qualité du son selon les algorithmes de cryptage

Fichier	Vigenére	Transposition	Xor	Blowfish	AES	Mot de Passe	RSA	Taille
Audio	Très mauvaise	Mauvaise	Mauvaise	bonne	bonne	bonne	excellente	360KO

Tab 4.3: la qualité du son selon les algorithmes de cryptage.

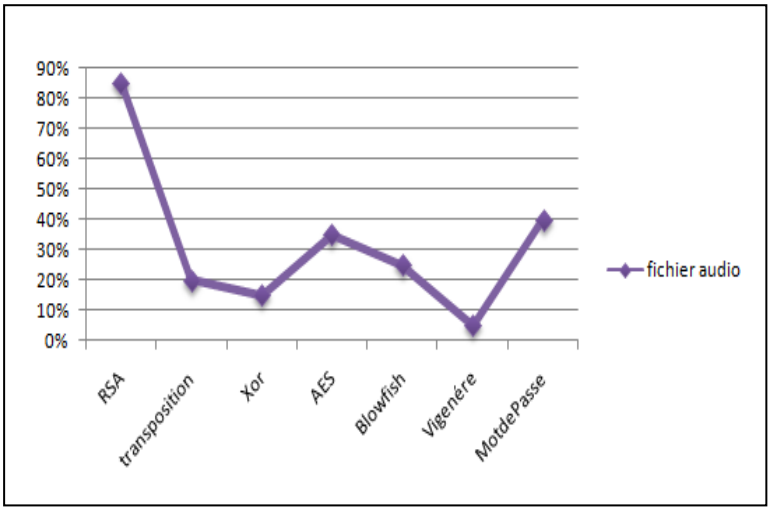


Fig 4.12: la qualité du son selon les algorithmes de cryptage.

4.10 L'amplitude temporelle et fréquentielle des fichiers crypté

Afin de réaliser cette tâche on a utilisé le Matlab.

4.10.1 Introduction sur Matlab

- ✓ Matlab est un puissant outil de calcul numérique, de développement ainsi de visualisation graphique dont il est dédié aux applications scientifiques vu qu'il reconnaît une très grande puissance de calcul.
- ✓ C'est un interpréteur: il permet d'interpréter et exécuter ses instructions ligne par ligne, d'autant plus il fonctionne dans plusieurs environnements tells que: Windows, xwindows, Macintosh.
- ✓ IL est dévloppé par la société The MathWorks.
- ✓ Le langage Matlab est un langage de programmation de quatrième génération (apparu en 1980), il peut s'interfacer avec autres langage comme: C, C++, Java, fortran.

4.10.2 Le code lié à l'amplitude et son explication

Le code suivant concrétise le tracer d'un signal wav dans le domaine temporel et fréquentiel (Fft)

Editor - C:\Users\hp\Desktop\matlab\testwav.m

testwav.m

```
%-------------------- domaine temporel-----------------------------
figure
    %y : les échantillons
    %fs : frequences d'échantillons
[y,fs]=audioread('G:\\Users\\seven\\Desktop\\audio1.wav');
    %t : vecteur de temps
t=linspace(0,length(y)/fs,length(y));
plot(t,y,'r');
%--------------------domaine frequetiel ---------------------------
figure
    % Nfft: la taille de la fft (signal frequentiel)
Nfft=1024;
    % G:c'est le tracer du module de la fft
    %fft: c'est la fonction qui calcul la fft (transformer
%de fourier rapide)
G=abs(fft(y,Nfft));
f=linspace(0,fs,Nfft);
plot(f,G,'b');
```

Fig 4.13: le code de l'amplitude**.**

4.10.3 L'amplitude des fichiers cryptés selon les algorithmes de cryptage

- **L'amplitude du fichier original (audio1.wav)**

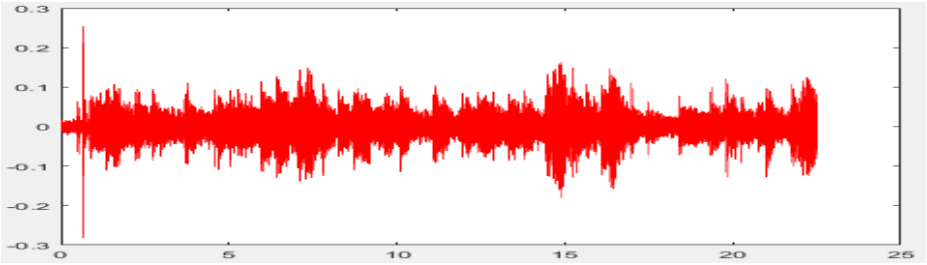


Fig 4.14: l'amplitude temporelle du fichier original.

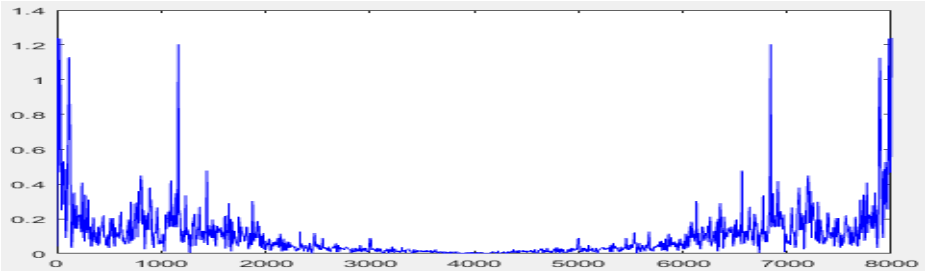


Fig 4.15: l'amplitude fréquentielle du fichier original.

❖ **L'amplitude du fichier crypté par vigenére**

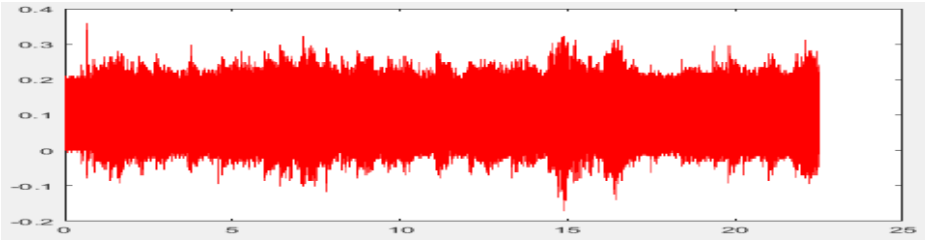


Fig 4.16: l'amplitude temporelle du fichier crypté par vigenére**.**

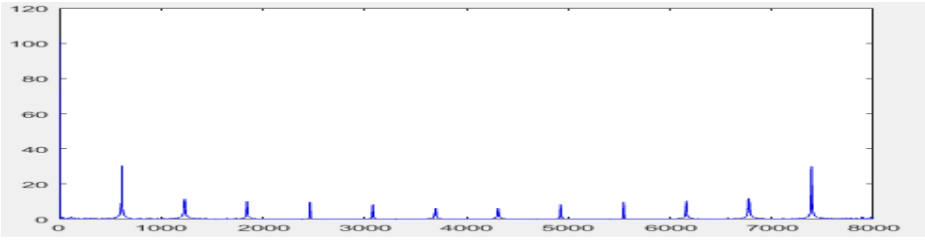


Fig 4.17: l'amplitude fréquentielle du fichier crypté par vigenére.

❖ **L'amplitude du fichier crypté par Blowfish**

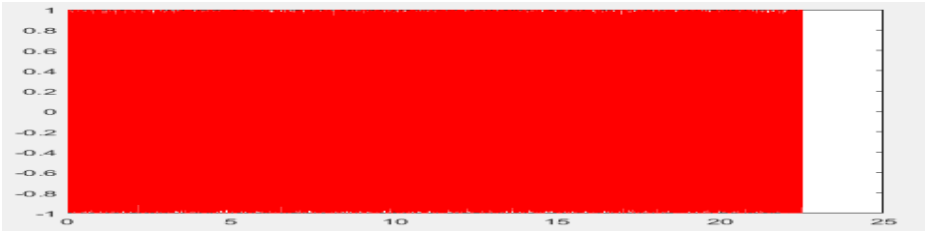


Fig 4.18: l'amplitude temporelle du fichier crypté par Blowfish**.**

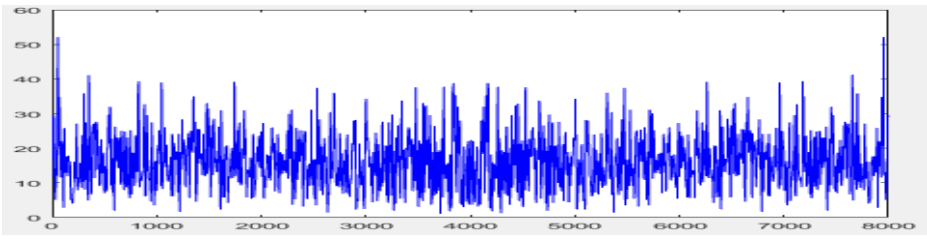


Fig 4.19: l'amplitude fréquentielle du fichier crypté par Blowfish**.**

❖ **L'amplitude du fichier crypté par MotDePasse**

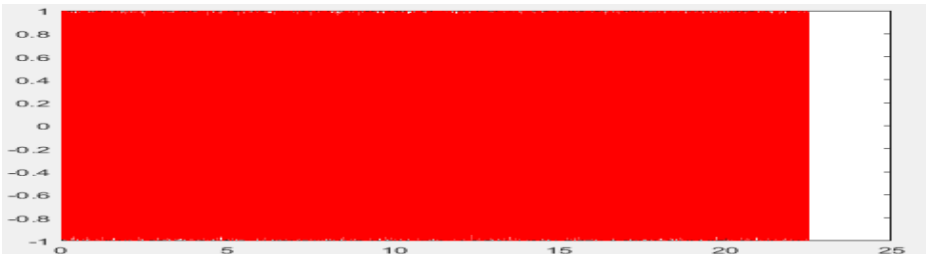


Fig 4.20: l'amplitude temporelle du fichier crypté par MotDePasse**.**

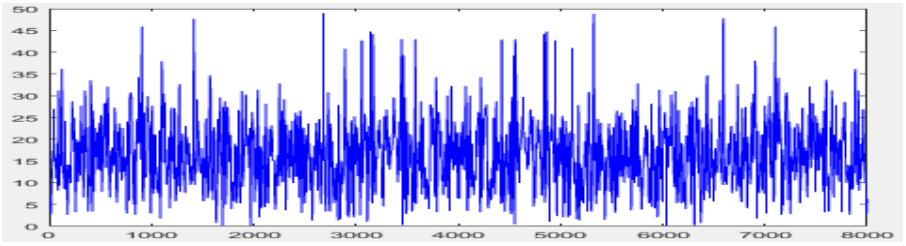


Fig 4.21: l'amplitude fréquentielle du fichier crypté par MotDePasse**.**

❖ **L'amplitude du fichier crypté par Xor**

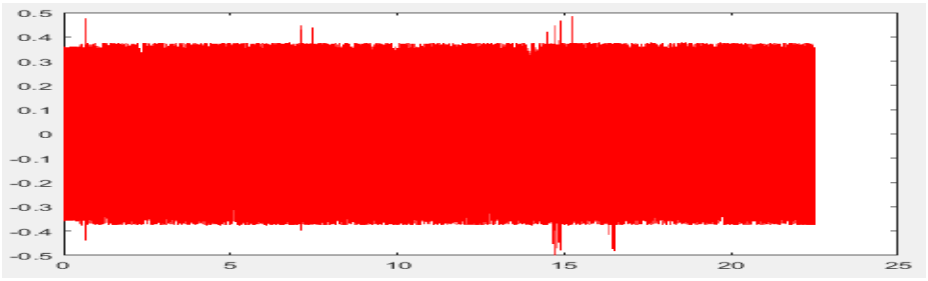


Fig 4.22: l'amplitude temporelle du fichier crypté par Xor**.**

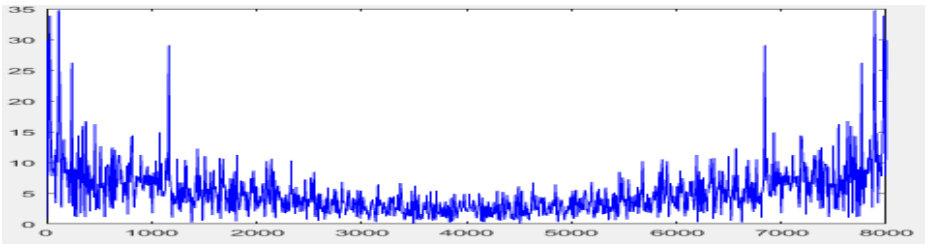


Fig 4.23: l'amplitude fréquentielle du fichier crypté par Xor**.**

❖ **L'amplitude du fichier crypté par AES**

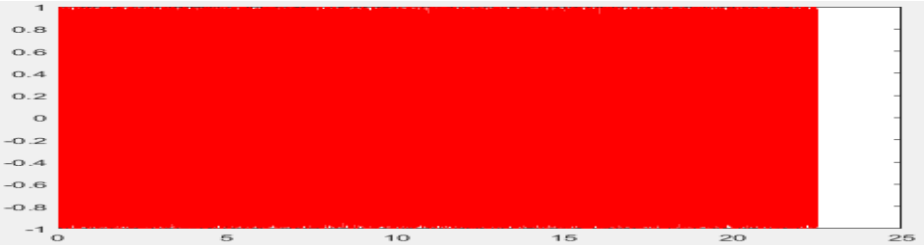


Fig 4.24: l'amplitude temporelle du fichier crypté par AES**.**

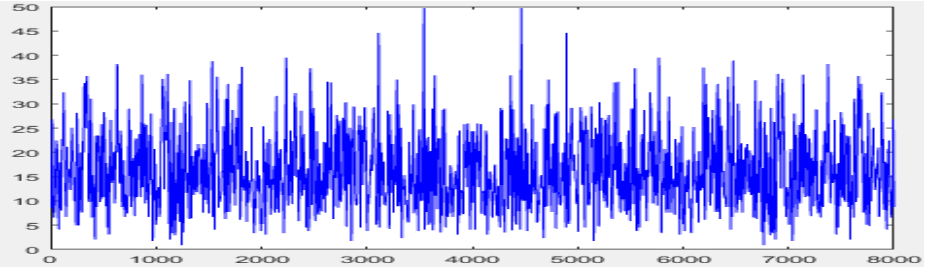


Fig 4.25: l'amplitude fréquentielle du fichier crypté par AES**.**

❖ **L'amplitude du fichier crypté par Transposition**

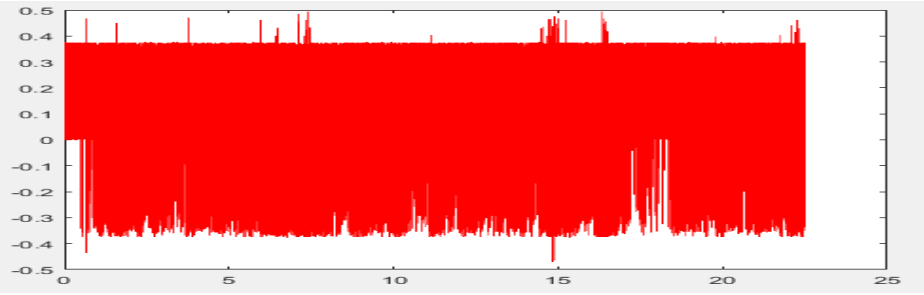


Fig 4.26: l'amplitude temporelle du fichier crypté par Transposition**.**

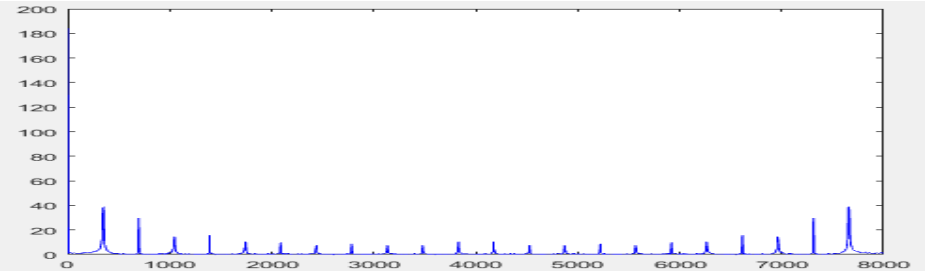


Fig 4.27: l'amplitude fréquentielle du fichier crypté par Transposition**.**

❖ **L'amplitude du fichier crypté par RSA**

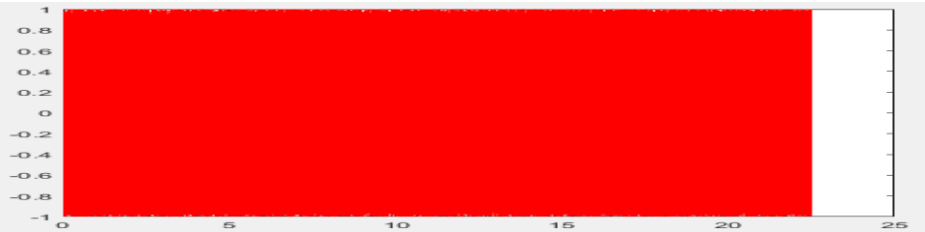


Fig 4.28: l'amplitude temporelle du fichier crypté par RSA**.**

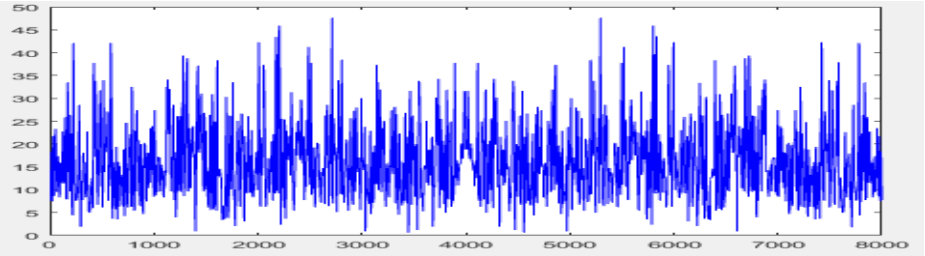


Fig 4.29: l'amplitude fréquentielle du fichier crypté par RSA.

Remarque

- D' après les résultats obtenus dans cette étude, on remarque que la qualité de cryptage dans l'algorithme de **Vigenére** est très mauvaise, Contrairement à celle de **RSA** d'où elle est excellente**.**
- On remarque aussi que les algorithmes: **Blowfish, MotDePasse, AES** ont une bonne qualité de cryptage sur un fichier audio.
- Les algorithmes de cryptage par **Transposition** et **Xor** ont une mauvaise qualité de chiffrement sur les fichiers audio.

4.11 Conclusion

Le but de ce chapitre est d'étudier et d'analyser la qualité et le temps de chiffrement selon les différents types d'algorithmes de cryptage tels que: AES, Blowfish, RSA, Xor, vigenére, Transposition, Motdepasse sur un fichier audio wav.

À la fin de cette étude nous avons obtenu les résultats suivants:

- Les algorithmes les plus performants vis-à-vis de la qualité (RSA, AES, blowfish).
- L'algorithme qui a une très mauvaise qualité: vigenére.
- L'algorithme avec une mauvaise qualité et un temps de traitement rapide: Xor.

Conclusion générale

De nos jours, la cryptographie basée sur la théorie du brouillage s'est largement évolué.

Récemment, grâce aux caractéristiques des signaux de brouillage tells que: les bonnes propriétés cryptographiques, l'hypersensibilité à la clé secrète et reproductibilité à l'identique (caractère déterministe des systems de brouillage) plusieurs recherché se font sur l'utilisation du brouillage dans des cryptosystèmes afin d'améliorer le temps de chiffrement et sécurité par rapport aux méthodes standard de brouillage (Mot de passe, Transposition, RSA, AES, XOR. Blowfish, vigenére).

Dans ce modeste travail : l'étude et l'analyse des algorithmes de brouillage sur les signaux de paroles ont été présentées en quatre chapitres.

Le premier chapitre, consacré à une présentation générale sur les différents algorithmes de brouillage qui a pour but de nous donner le principe de fonctionnement de ces derniers.

Le second chapitre, permet de nous rapprocher de l'idée et la structure des formats des fichiers audio.

Le troisième chapitre, nous donne une explication générale sur le modèle client serveur ainsi son utilité dans les réseaux.

Le dernier chapitre, regroupe tous sorts d'outils dont on s'est inspiré afin d'entamer notre étude qui a pour but la simulation et l'analyse de l'impact des algorithmes de brouillage sur les fichiers audio, tels que: Mot de passe, Transposition, RSA, Vigenére, AES, XOR. Blowfish. Ces deniers sont dotés de sécuriser et rendre l'information plus confidentielle.

Au cours de notre étude, nous nous sommes préoccupés sur deux facteurs principaux:

Le temps de traitement des chiffrements de ces algorithmes de cryptage.

La qualité décryptage sur le fichier audio.

Conclusion générale

Nos résultats nous ont mené à conclure et à dire que:

- l'algorithme Xor est plus rapide en termes temps avec une faible qualité de cryptage.
- L'algorithme vigenére a une très mauvaise qualité de cryptage.
- L'algorithme RSA est plus performant vis-à-vis de sa qualité de cryptage seulement il Est plus lent.

Comme perspectives de notre travail, nous souhaiterons que le prochain projet de fin d'études traite le cas de brouillage sur la parole on utilisant les fichiers audio compressés tels que Mp3 tous on le compare avec les fichiers non compressés tels que le wav dont en a fait cette étude, à la fin on analyse les résultats.

Bibliographies

[1]: B.Prenee, Understanding Cryptography, Springer 2010.

[2]: D.Stinson, Cryptography: Theory and Practice, CRC Press 1995.

[3]: M^{me} Meziane Tani Souad, **Le son en MultiMedia.**

[4]: George Coulouris et al., Distributed Systems Concepts and Design, Prentice-Hall, 2001.

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

[6]: Jihad Nadir, Ashraf Abu Ein and Ziad Alqadi_A Technique to Encrypt-decrypt Stereo Wave File_Computer Sngineering Department Albalqa Applied University Amman – Jordan,2011.

[7]: Ahmad Jawahira,1,*, Haviluddinb, An audio encryption using transposition method, International Journal of Advances in Intelligent Informatics,2012 .

[8]: Shital C. Patil, R. R. Keole, “Cryptography, Steganography & Network Securities”, "International Journal of Pure and Applied Research in Engineering and Technology”, 2012; Volume 1(8): pp. 9-15

[9]: The Advanced Encryption Standard, *Federal Information Processing Standards Publication (FIPS 197)*, pp. 92-96, 2001.

[10]: T. McDevitt and T. Leap. “Multimedia cryptology,” *Cryptologia* (Taylor & Francis), vol. 33, no. 2, 142-150, 2009. DOI: 10.1080/01611190802300408

[11]: R. Gnanajeyaraman, K. Prasadh, and Ramar “Audio encryption using higher dimensional chaotic map,” *Int. J. Recent Trends Eng.*, no. Academy Publisher), vol. 1, no. 2, 103-107, 2009.

[12]: Md. M. Rahman, T. K. Saha, and Md. A.-A. Bhuiyan. “Implementation of RSA algorithm for speech data encryption and decryption,” *Int. J. Comput. Sc. & Netw. Secur.*, vol. 12, no. 3, 74-82, 2012.

[13]: S. Sharma, L. Kumar, and H. Sharma. “Encryption of an audio file on lower frequency band for secure communication,” *Int. J. Adv. Res. Comput. Sc. & Software Eng.*, 3, no. 7, 79-84, 2013.

Bibliographies

[14]: B. Gadanayak, C. Pradhan, and U. C. Dey. “Comparative study of different encryption techniques on MP3 compression,” *Int. J. Comput. Appl.*, vol. 26, no. 3, 28-31, 2011.

[15]: B. Gadanayak, C. Pradhan, and N. Baranwal. “Secured partial MP3 encryption technique,” *Int. J. Comput. Sci. & Inform. Technol.*, vol. 2, no. 4, 1584-1587, 2011.

[16]: R. Gnanajeyaraman, K. Prasadh, and Ramar, "Audio encryption using higher dimensional chaotic map,"*International Journal of Recent Trends in Engineering,* vol. 1, pp. 103-107, 2009.

[17]: M. Kaur and S. Kaur, "Survey of Various Encryption Techniques for Audio Data," *International Journal of Advanced Research in Computer Science and Software Engineering,* vol. 4, pp. 1314-1317, 2014.

Reference site web

[a1]: http://www.cryptage.org/chiffre-cesar.html.

[a2] : l'outil utiles selon: http://www.nymphomath.ch/crypto/cesar/index.html.

[a3]: http://www.montefiore.ulg.ac.be/~dumont/pdf/crypto09-10.pdf.

[a4]: http://www.primenumbers.net/Renaud/fr/crypto/DES.htm.

[a5]:https://www.inbconcept.com/site/medias/documents/documentation_acelocker.pdf.

[a6]: http://sevat.fr/damien/M1S1/Theorie_Info/Rapport/

[a7]: http://www.bibmath.net/crypto/index.php?action=affiche&quoi=moderne/difhel

[a8]: http://www.cryptage.org/rsa.html

[a9] :http://wawadeb.crdp.accaen.fr/iso/tmp/ressources/linux/reseau/www.laissus.fr/cours/node240.html

[a10]: http://ecariou.perso.univ-pau.fr/cours/web/cours-architecture-par6.pdf

[a11]: http://igm.univ-mlv.fr/~duris/NTREZO/20022003/Peer-to-peer.pdf

[a12]: https://www.commentcamarche.net/contents/528-port-ports-tcp-ip

[a13] :http://www.hackerhighschool.org/lessons/HHS_fr3_Les_Ports_et_Protocoles.pdf

[a14]: https://www.supinfo.com/articles/single/2519-architecture-client-serveur

[a15]:https://fr.wikipedia.org/wiki/Chiffrement_par_transposition#Cryptanalyse_des_chiffres_de_transposition

[a16] : http://www.yannvidal.com/wordpress/pdf/kit2survie.pdf

[a17]: https://fr.wikipedia.org/wiki/Format_de_fichier_audio

[a18]: http://dspace.univ-tlemcen.dz/bitstream/112/1046/10/chapitre4.pdf.

[a19]: http://dspace.univ-tlemcen.dz/bitstream/112/1046/7/chapitre1.pdf.

[a20]: https://www.supinfo.com/articles/single/2680-mieux-connaitre-formats-audio

[a21]: https://www.supinfo.com/articles/single/2519-architecture-client-serve

Résumé

Le son est une chose familière dans notre vie quotidienne que l'on en oublie souvent la signification physique qui est loin d'être facile à comprendre. On le trouve partout surtout avec l'apparition de l'Internet. La cryptographie est traditionnellement utilisée pour dissimuler des messages aux yeux de certains utilisateurs. Cette utilisation a aujourd'hui un intérêt d'autant plus grand que les communications via Internet circulent dans des infrastructures dont on ne peut garantir la fiabilité et la confidentialité. L'objectif principal de notre travail est basé sur le cryptage et la protection des flux audio représenté par des fichiers en format Wave en utilisant des algorithmes de chiffrement à clé secrète et à clé publique comme AES (Advanced Encryption Standard), Blowfish et RSA (Rivest, Shamir, Adleman). Cette réalisation est suivie par une étude comparative entre ces différents algorithmes selon un ensemble de critères comme le temps de chiffrement/déchiffrement et la qualité et la performance de l'opération de cryptage.

Mots clés chiffrement à clé secrète, chiffrement à clé publique, flux audio, XOR, AES, Blowfish, RSA, Transposition, vigenére, format Wave

Abstract

The sound is a familiar thing in our daily lives that we often forget the physical meaning which is not easy to understand. In the found throughout, especially with the advent of the Internet. Cryptography is traditionally used to conceal messages to some users. This use today has an interest of greater as Internet communications flowing in for infrastructure which it cannot guarantee the reliability and confidentiality. The main objective of our work is based on encryption and secure audio represented by Wav files flow using secret key encryption algorithms and public key as AES (Advanced Encryption Standard), RSA(Rivest, Shamir, Adleman) and Blowfish. This achievement is followed by a comparative study of these algorithms on a set of criteria, such as encryption/ decryption time and the quality and performance of the encryption operation.

Keys words: private key encryption; public key encryption audio stream, XOR; AES; Blowfish; RSA, Transposition, vigenére, wav file; Cryptography, Audio security

خلاصة

يلعب الصوت دورا هاما في حياتنا اليومية على الرغم من صعوبة استيعاب منطقه العلمي .حيث يغزو كل المجالات

خصيصا عند اكتشاف الشبكة العنكبوتية

يستخدم التشفير عادة لإخفاء الرسائل في أعين بعض المستخدمين

في وقتنا الحاضراصبح استخدامها جد مهم حيت تنتشر في شبكات الاتصالات المعلوماتية التي لا يمكن ضمان سريتها و

حمايتها

الهدف الرئيسي من عملنا هو تشفير و حماية تدفقات الصوت الممثلة بملفات Wave *و ذلك باستخدام خوارزميات تشفير بالمفتاح السري والمفتاح العام مثل* :

RSA/blowfish/AES

تعتمد دراستنا على المقارنة بين مختلف الخوارزميات وفقا لمجموعة من المعايير مثل :

وقت التشفير / فك التشفير ونوعية وأداء عملية التشفير

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