Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

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 Licence en Informatique

Thème

Développement d'une application pour l'échange des messages sécurisés

Réalisé par :

- **Kebir Bahia**
- **Rahmouni Samia**

Présenté le 27 Mai 2015 devant la commission d'examination composée de MM.

- *Mr BENAISSA* *(Encadreur)*
- *Mr MANA M* *(Examinateur)*
- *MR LEHSAINI M* *(Examinateur)*

Année universitaire : 2014-2015

Remerciements

Nous remercions ALLAH de nous avoir données la santé et le courage afin de pouvoir réussir ce travail.

Ce travail est l'aboutissement d'un long cheminement au cours duquel nous avons bénéficié l'encadrement, des encouragements et du soutien de plusieurs personnes, à qui nous tenons à dire profondément et sincèrement merci.

Nous exprimons notre grande gratitude à notre professeur encadrant « Mr Benaissa Med » , d'avoir accepté de suivre notre travail et pour ses précieux conseils et ses orientations.

Nous avons eu le privilège de travailler parmi votre équipe et d'apprécier vos qualités et vos valeurs, votre sérieux, votre compétence.

Nous remercions toutes les personnes surtout notre frère Djebbar Med Ali , d'une quelconque manière , nous ont apporté leur amitié, leur attention, leurs encouragements, leur appui et leur assistance pour que nous puissions mener à terme ce travail .

Nous tenons à exprimer nos sincères remerciements à tous les professeurs qui nous ont enseigné et qui par leurs compétences nous ont soutenu dans la poursuite de nos

études.

MERCI

Dédicaces

Je dédie avec respect ce modeste travail fruit de plusieurs années de réflexion à tout des qui ont inspiré aimé et aidé de proche ou de loin

Mon père qui ma prête attention et qui a fait de moi ce que je suis maintenant.

Ma très chère mère qui ma offert depuis toujours le plus belle cadeau de l'univers, le cœur d'une mère et à qui je serais éternellement reconnaissant pour son aide moral et son soutient matériel et affectif.

A ma grand frère Abdelhak qui je souhaite beaucoup de joie et de bonheur dans sa vie.

A ma petit frère Mouad et ma petite sœur Ammaria qui je souhaite la réussite dans ces études.

A ma tante Hanifa qui je souhaite une belle vie pleine de joie et de bonheur.

A tous les membres de ma famille, petits et grands, mes oncles, mes tantes, mes cousins et mes cousines.

A tous mes amies et mes camarades ... Merci pour tous ces moments qu'on avait vécus ensemble

A mon binôme Kebir Bahia Wahiba

A La Fin tous simplement je vous dites « J'aime Beaucoup »

Rahmouni Samia

Dédicaces

A mes chers parents symbole de sacrifice, de tendresse, qui m'ont a éclairée mon chemin et qui m'ont encouragé et soutenue toute au long de mes études.

A ma grand frère Mostefa et sa femme Samira à qui je souhaite beaucoup de bonheur dans la vie.
Sans oublier ma nièce Rahima Nour el Houda ma petite Ange.

A mon frère Abdou El Ilah et ma sœur Nouria qui je souhaite beaucoup de joie ,de réussites et de bonheur dans la vie,

A ma cousine Meriem qui je souhaite la réussite dans les études.

A tous les membres de ma famille, petits et grands, mes oncles, mes tantes, mes cousins et mes cousines.

A tous mes camarades et mes amies … Merci pour tous ces moments qu'on avait vécus ensemble.

A mon binôme Rahmouni Samia

A tous ceux qui m'aiment & que j'aime.

Kebir Bahia

Table des matières

Chapitre 03 : ***les algorithmes de chiffrements à clé secrète***

Chapitre 04 : ***développement d'une application de chiffrement***

Table des Figures

Introduction générale

Le besoin de dissimuler les informations préoccupe l'homme depuis le début de la civilisation La confidentialité apparait notamment nécessaire lors des luttes pour l'accès au pouvoir. Puis elle se développe énormément à des fins militaires et diplomatiques. Aujourd'hui, de plus en plus d'applications dites civiles nécessitent la sécurité des données transitant entre deux interlocuteurs ou plusieurs, via un vecteur d'information comme les réseaux de télécommunications actuels et futurs. Ainsi, les banques utilisent ces réseaux pour assurer la confidentialité des opérations avec leurs clients ; les laboratoires de recherche s'en servent pour échanger des informations dans le cadre d'un projet d'étude commun ; les chefs militaires pour donner leurs ordres de bataille, etc.
De nos jours, la nécessité de cacher ou de casser une information rentre dans un vaste ensemble appelé cryptographie.
Le mot cryptographie est un terme générique désignant l'ensemble des techniques permettant de chiffrer des messages, c'est-à-dire permettant de les rendre inintelligibles sans une action spécifique. Le verbe crypter est parfois utilisé mais on lui préfèrera le verbe chiffrer .
La cryptographie est l'art du secret à celle de la piraterie, sert non seulement à préserver la confidentialité des données mais aussi à garantir leur intégrité et leur authenticité.

L'objectif principal de notre projet de fin d'étude, est de développer une application qui va chiffrer des messages et qui assuré la confidentialité.

Notre mémoire est structuré comme suite :

- ✓ **Chapitre 1** : étude de l'architecteur client/serveur.
- ✓ **Chapitre 2** : introduction générale sur des algorithmes de chiffrements à clé secrète
- ✓ **Chapitre 3** : chiffrement des messages texte par des algorithmes de chiffrement à clé secrète
- ✓ **Chapitre 4** : Développement d'une application de chiffrement réparti selon le modèle client/serveur pour sécuriser le transfert des messages texte sur un réseau local

Chapitre 1

1.10. Introduction

Le développement de réseau informatique a conduit à l'apparition d'une architecture distribué basé sur le modèle Client/serveur.

Architecture Client/serveur est une technologie d'un ensemble d'évolutions survenus dans les 30 dernières années

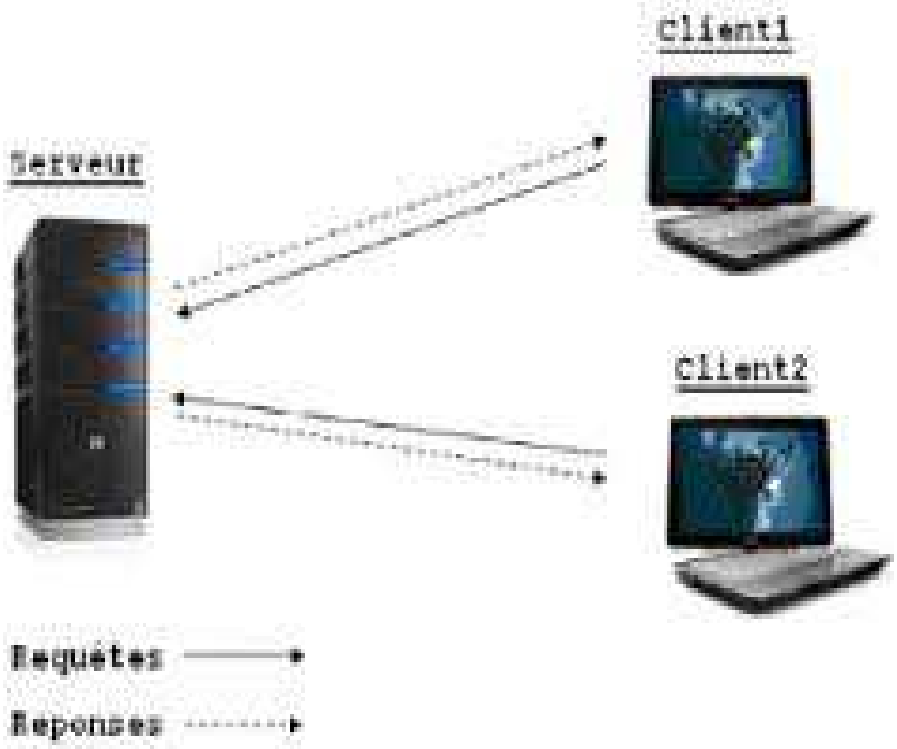


Fig. 1.1 : modèle client/serveur

1.11. Le serveur

On appelle logiciel serveur un programme qui offre un service sur le réseau.

Le serveur accepte des requêtes, les traite et renvoie le résultat au demandeur (client). Le terme serveur s'applique à la machine sur lequel s'exécute le logiciel serveur. [s1]

Un service est fourni sur un Port de communication identifié par un numéro. Certains numéros de Port (internationalement définis) identifient le service quelque soit le site. [s2]

Exemple :

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

Le service TELNET (émulation terminal) sur le port 23,

Le service SMTP (mail) sur le port 25.

Les types de serveurs :

Serveur itératifs : ne gèrent qu'un seul client à la fois

Serveur parallèles : fonctionnent en mode concurrent

1.12. Le client

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

1.13. L'architecture client/serveur

C'est la description du fonctionnement coopératif entre le serveur et le client. Les services internet sont conçus selon cette architecture. Ainsi, chaque application est composée de logiciel serveur et logiciel client. A un logiciel serveur, peut correspondre plusieurs logiciels clients développés dans différents environnements: Unix, Mac, PC...; la seule obligation est le respect du protocole entre les deux processus communicants. Ce protocole étant décrit dans un RFC (Request For Comment).

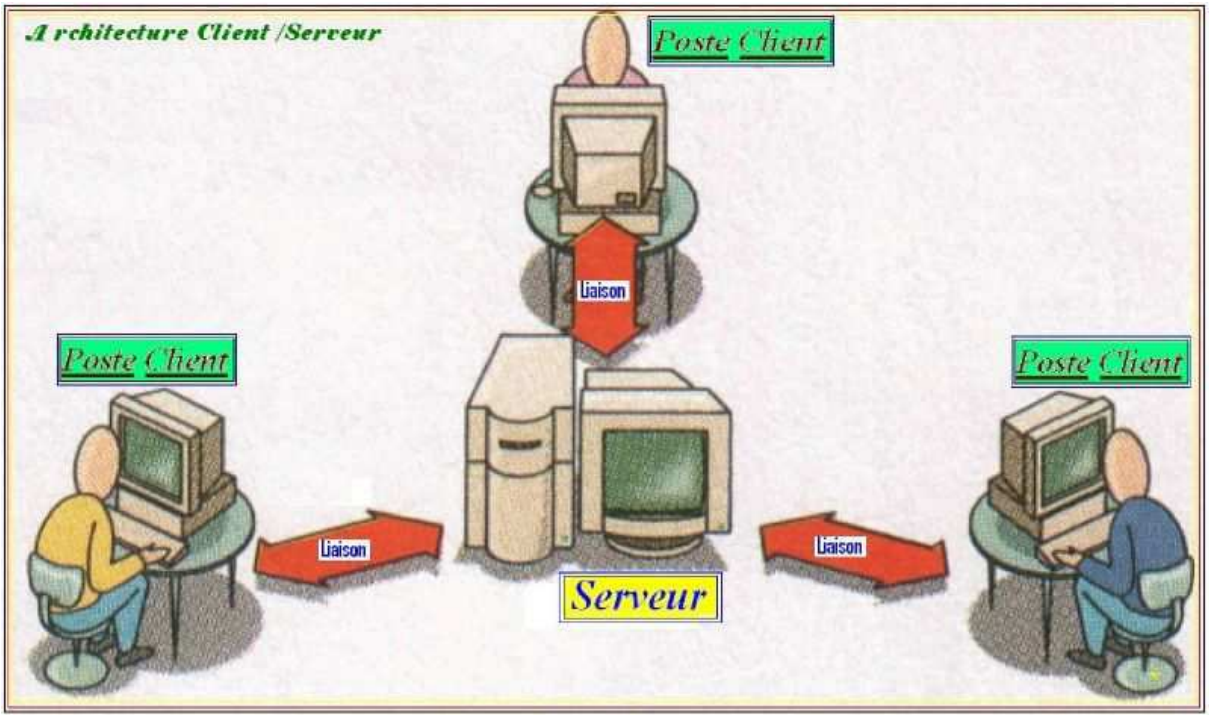


Fig. 1.2 : Architecture client/serveur

1.4.1. Caractéristiques de l'architecture client serveur

Les éléments qui caractérisent une architecture client serveur sont :

- **Service**

Le modèle client serveur est une relation entre des processus qui tournent sur des machines séparées. Le serveur est un fournisseur de services. Le client est un consommateur de services.

- **Partage de ressources**

Un serveur traite plusieurs clients et contrôle leurs accès aux ressources

- **Protocole asymétrique**

Conséquence du partage de ressources, le protocole de communication est asymétrique
Le client déclenche le dialogue ; le serveur attend les requêtes des clients.

- **Transparence de la localisation**

L'architecture client serveur doit masquer au client la localisation du serveur (que le service soit sur la même machine ou accessible par le réseau). Transparence par rapport aux systèmes d'exploitation et aux plates-formes matérielles. Idéalement, le logiciel client serveur doit être indépendant de ces deux éléments

- **Message**

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

- **Encapsulation des services2**

Un client demande un service. Le serveur décide de la façon de le rendre une mise à niveau du logiciel serveur doit être sans conséquence pour le client tant que l'interface message est identique.

- **Evolution**

Une architecture client serveur doit pouvoir évoluer horizontalement (évolution du nombre de clients) et verticalement (évolution du nombre et des caractéristiques des serveurs). [s4]

1.4.2. Les objectifs du client-serveur

- Répartir les tâches entre le client et le serveur :
- décharger le serveur de l'exploitation des données.
- décharger la station cliente de la gestion des données.
- réduire le trafic sur le réseau. [1]

1.4.3. Avantages de l'architecture client/serveur

- **Unicité de l'information :** pour un site web dynamique par exemple (comme vulgarisation-informatique.com), certains articles du site sont stockés dans une base de données sur le serveur. De cette manière, les informations restent identiques. Chaque utilisateur accède aux mêmes informations.
- **Meilleure sécurité** : Lors de la connexion un PC client ne voit que le serveur, et non les autres PC clients. De même, les serveurs sont en général très sécurisés contre les attaques de pirates.
- **Meilleure fiabilité** : En cas de panne, seul le serveur fait l'objet d'une réparation, et non le PC client.
- **Facilité d'évolution :** Une architecture client/serveur est évolutive car il est très facile de rajouter ou d'enlever des clients, et même des serveurs. . [s3]

1.4.4. Inconvénients de l'architecture client/serveur

Architecture inadaptée au cas de traitements longs, que cette longueur soit due à :

- Temps de traitement long au niveau du serveur
- Nombreux échanges requête/réponse entre client et serveur avant que le client ne se déconnecte [2]
- Un coût d'exploitation élevé (bande passante, câbles, ordinateurs surpuissants). [s3]

1.14. Évolution des architectures C\S

1ére génération

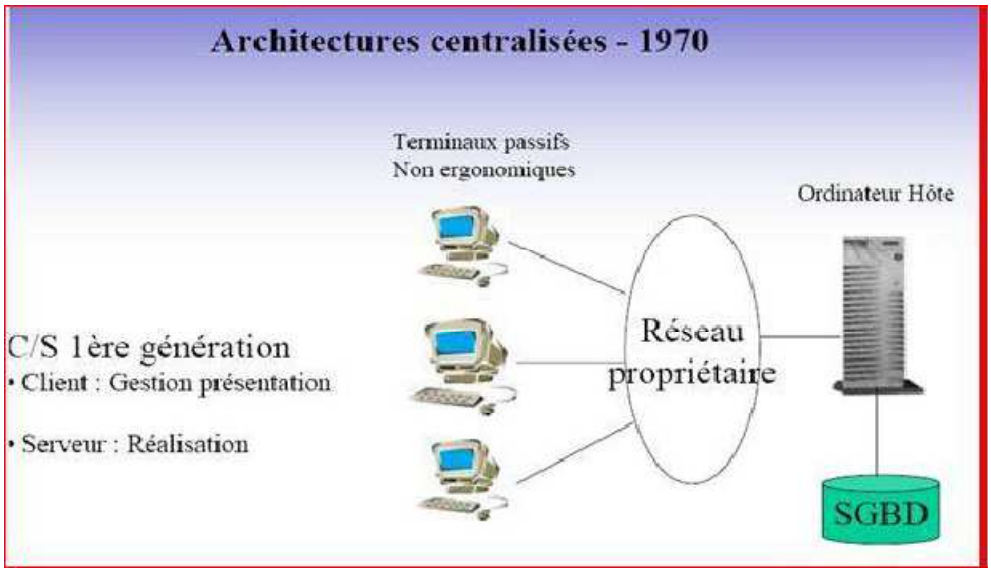


Fig. 1.3 : 1ère génération client/serveur

2ième génération

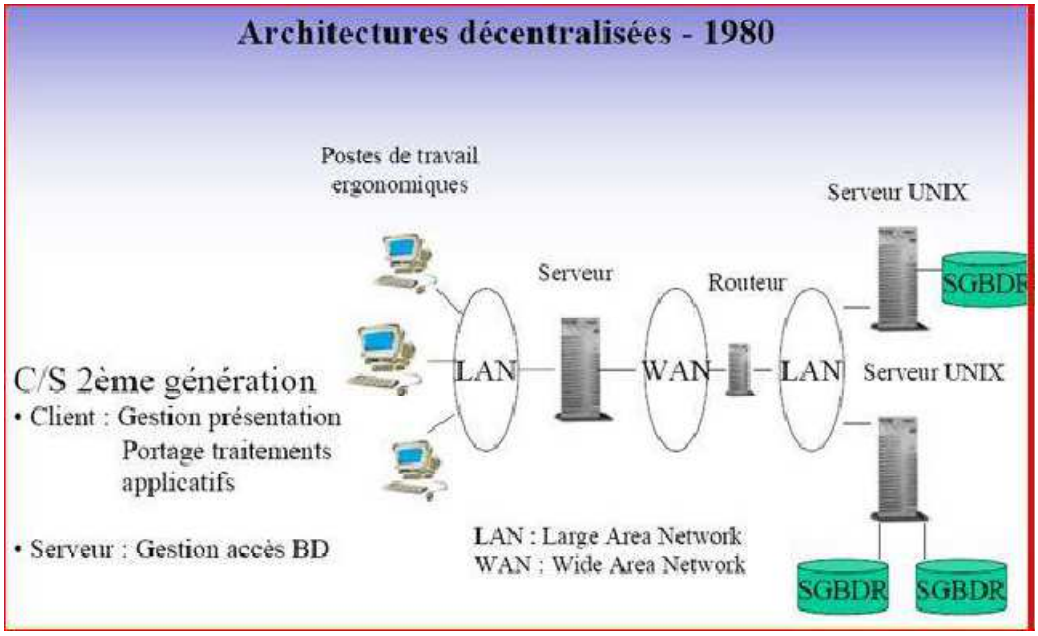


Fig. 1.4 : 2ième génération client/serveur

3éme génération

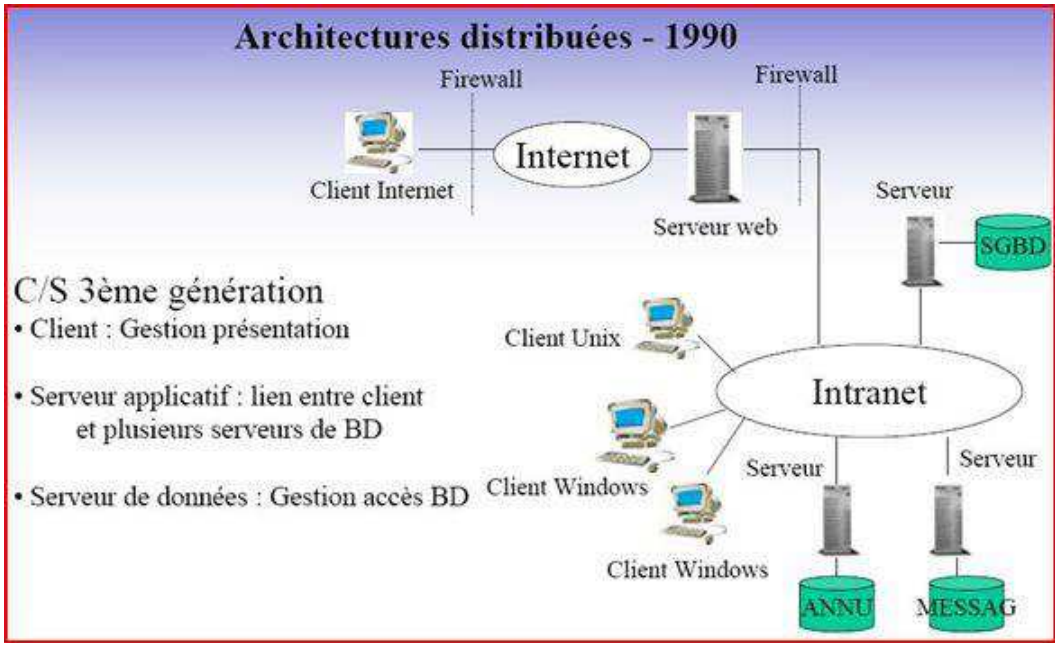


Fig. 1.5 : 3ième génération client/serveur

1.15. Notion de port et de protocole

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

Les informations échangées par le système Sant des flux ou bien des paquets en mode texte (ex : ASCII) et peuvent être compris par des humains (ex: SMTP, POP3, HTTP, etc.). D'autres utilisent des échanges de flux binaires (ex : DNS).

Chaque paquet réseau contient :

L'adresse IP de la machine d'origine (le client dans le cas d'une requête),

L'adresse IP de la machine de destination (le serveur dans notre cas),

Numéro de port qui permet de savoir à quel service vas analysée le paquet

Un Port :

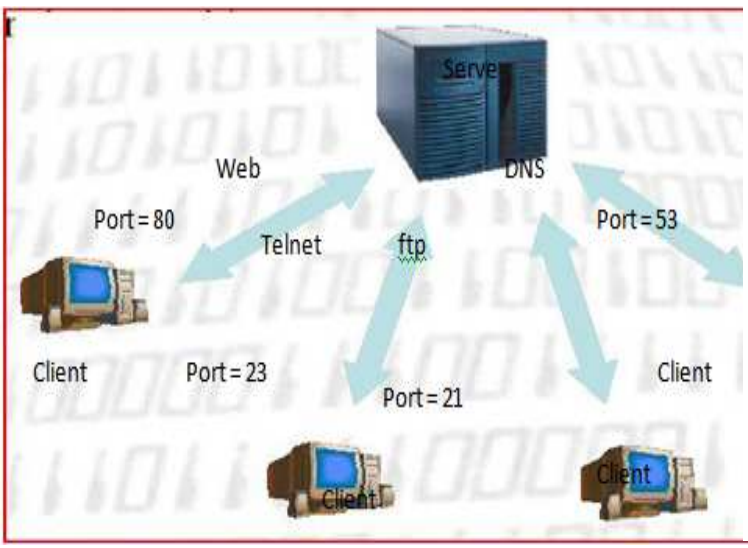
Un numéro entier (16 bits) va identifier à quel service ou un programme va exécuter le traitement de paquet reçus

De 0 à 1023 : port reconnus ou réservés.

Un protocole

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

Le but d'un protocole est pour facilité la compréhension entre machines /logiciels [4]



Port	Service ou Application
21	FTP
23	Telnet
25	SMTP
53	DNS
80	HTTP
110	POP3
119	NNTP

Fig. 1.6 : notion service et port

1.16. Les Sockets

Un socket est une abstraction à travers laquelle une application peut envoyer et recevoir des données, en de la même manière comme un fichier ouvert permet une application de lire et écrire des données à stable le stockage. Une prise permet à une application de "Plug-in" sur le réseau et de communiquer avec d'autres applications qui sont également branchés sur le même réseau. Informations écrit à la prise de courant par une application sur une machine peut être lue par une application sur un autre Machine. [5]

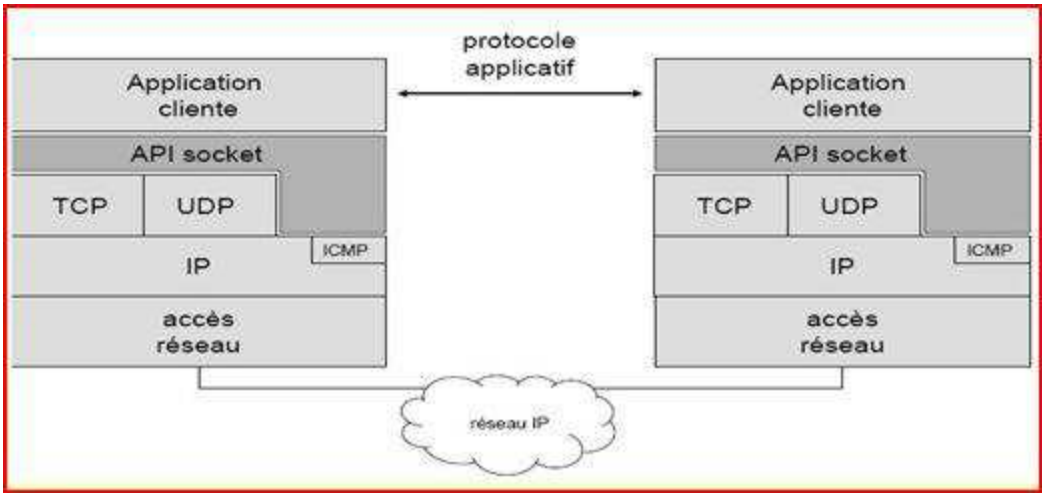


Fig. 1.7 sockets

1.17. Les middlewares

On appelle middleware (ou logiciel médiateur en français), littéralement “ élément du milieu“, l’ensemble des couches réseau et services logiciel qui permettent le dialogue entre les différent composent d’une application répartie. Ce dialogue se base sur un protocole applicatif commun, défini par l’API du middleware.

Middleware c’est une interface de communication universelle entre processus. Il représente véritablement la clef de voûte de toute application client/serveur.

L’objectif principal du middleware est d’unifier, pour les applications, l’accès et la manipulation de l’ensemble des services disponibles sur le réseau, afin de rendre l’utilisation de ces derniers presque transparente. [6]

Rôles des middlewares :

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

Les avantages :

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

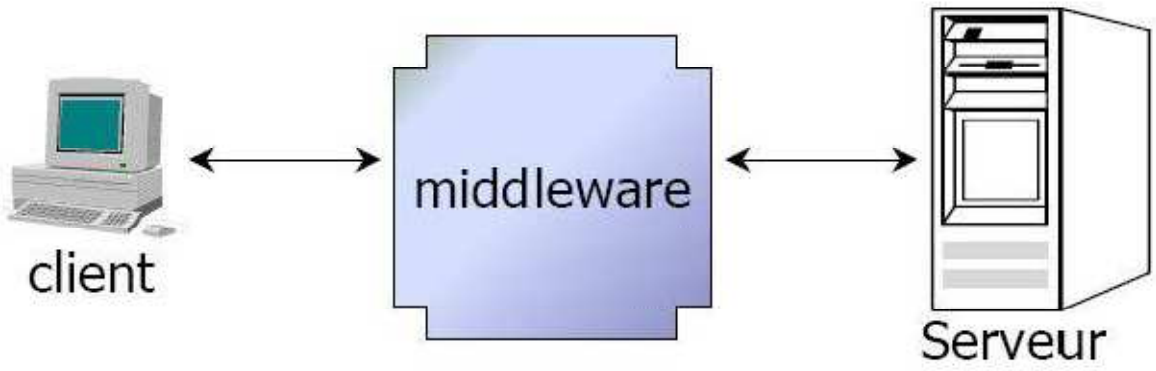


Fig. 1.8 Middlewares

1.18. Conclusion

Dans ce premier chapitre, nous avons présenté une étude sur l'architecture client serveur, ainsi nous avons indiqués l'importance de la notion de protocoles, ports, sockets et middleware pour simplifier l'implémentation des applications client/serveur dans l'Internet. Dans le chapitre suivant nous avons présenté une introduction générale sur des algorithmes de chiffrements à clé secrète.

Chapitre 2

2.1. Introduction

L’objectif fondamental de la cryptographie est de permettre à deux personnes de communiquer à travers un canal peu sûr de telle sorte qu’un opposant, un troisième personne, qui a accès aux informations qui circulent sur le canal de communication, ne puisse pas comprendre ce qui est échangé. Le canal peut être par exemple une ligne téléphonique ou tout autre réseau de communication.

2.2. Définition de cryptologie

La cryptologie est la science qui étudie les aspects scientifiques de ces techniques, c'est-à-dire qu'elle englobe la cryptographie et la cryptanalyse. [s5]

- **la cryptographie**

  Est l'étude des principes, méthodes et techniques mathématiques reliées aux aspects de sécurité de l'information telles la confidentialité, l'intégralité des données, authentification d'entités, et l'authentification de l'originalité des données. C'est un ensemble de techniques qui fournit la sécurité de l'information.

  La cryptographie vous permet de stocker des informations sensibles ou de les transmettre à travers des réseaux non sûrs (comme Internet) de telle sorte qu'elles ne peuvent être lues par personne à l'exception du destinataire convenu.

- **La cryptanalyse**

  Étudie la sécurité des procédés de chiffrement utilisés en cryptographie. Elle consiste alors à casser des fonctions cryptographiques existantes, c'est-à-dire à démontrer leur sécurité. La cryptanalyse mêle une intéressante combinaison de raisonnement analytique, d'application d'outils mathématiques, de découverte de redondances, de patience, de détermination et de chance. [7]

2.3. Définition de la cryptographie

La cryptographie est l'**art de chiffrer**, coder les messages est devenue aujourd'hui une science à part entière. Au croisement des **mathématiques**, de l'**informatique**, et parfois même de la **physique**, elle permet ce dont les civilisations ont besoin depuis qu'elles existent :
le maintien du secret. Pour éviter une guerre, protéger un peuple, il est parfois nécessaire de cacher des choses.

- **A quoi sert la cryptographie ?**

  Les communications échangées entre Alice et Bob sont sujettes à un certain nombre de menaces. La cryptographie apporte un certain nombre de fonctionnalités permettant de pallier ces menaces, pour Confidentialité, Authentification, Intégrité, Non-répudiation :

1. **Confidentialité**

   La confidentialité consiste à empêcher l'accès aux informations qui transitent à ceux qui n'en sont pas les destinataires. Ils peuvent lire les messages cryptés transmis sur le canal mais ne peuvent pas les déchiffrer.

2. **Authentification**

   Il faut pouvoir détecter une usurpation d'identité. Par exemple, Alice peut s'identifier en prouvant à Bob qu'elle connaît un secret S qu'elle est la seule à pouvoir connaitre.

3. **Intégrité**

   Il s'agit de vérifier que le message n'a pas subi d'altérations lors de son parcours.

   Elle concerne plutôt une modification volontaire et malicieuse e l'information provoquée par un tiers lors de transfert sur le canal. Ces modifications sont en générales masquées par le tiers pour être difficilement détectables.

4. **Non-répudiation**

   C'est une protection protagonistes d'un échange entre les informations, et non plus contre un tiers. Si Alice envoie un message M, elle ne doit pas pouvoir prétendre ensuite devant Bob qu'elle ne l'a pas fait, ou alors qu'elle a envoyé M'et que le message a été mal compris. [8]

2.4. Cryptage

Le **cryptage** est donc un moyen de transmettre les informations confidentielles de telle sorte qu'elles puissent être lues uniquement par des personnes autorisées. On distingue

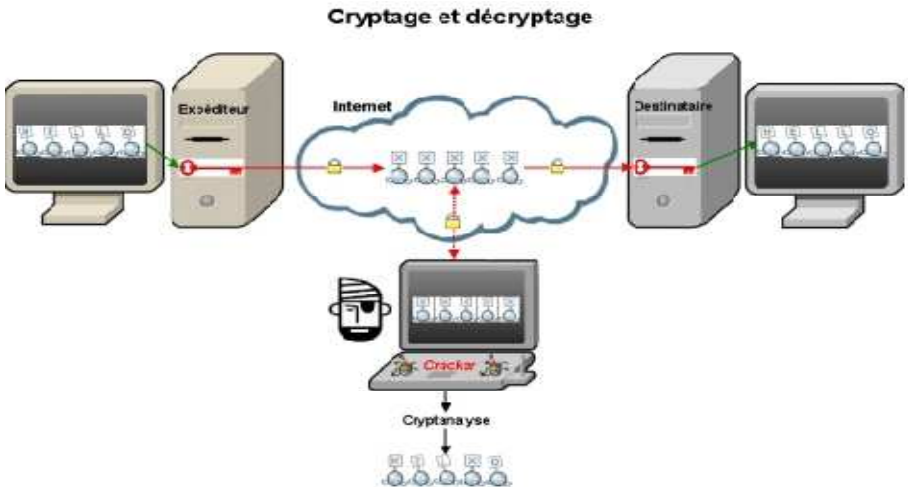


Fig.2.1 : Schéma de cryptage

Deux principaux types de cryptage : symétrique et asymétrique.

2.4.1. Cryptage symétrique

Autrement appelée cryptage à clé privée, ce type se base sur l'utilisation d'une clé pour crypter et décrypter les messages. La sécurité de cette solution repose sur le fait que la clé est connue uniquement par l'émetteur et le récepteur du message.
L'exemple historique de l'utilisation du cryptage symétrique est fameux téléphone rouge qui reliait le Kremlin à la Maison Blanche. La clé privée était alors transmise dans une valise diplomatique. Pour une meilleure sécurité, elle était détruite et réinitialisée après chaque conversation.
Le cryptage symétrique fonctionne selon deux procédés différents :

- le cryptage par flot : le cryptage s'effectue en continu, bit par bit
- le cryptage par bloc : le cryptage s'effectue sur les blocs de bits

Les avantages du cryptage symétrique

-la rapidité d'exécution (une seule clé utilisée).
-la simplicité d'implémentation (gestion d'une seule clé).
- Permet de concevoir différents mécanismes cryptographiques (fonctions de hachage, etc...)
- Clés relativement courtes.

Les inconvénients du cryptage symétrique :

- la complexité de fonctionnement : une obligation d'avoir le nombre de clés privées égal au nombre de destinataires.
- la sécurisation de la chaîne de transmission de la clé.
- Impossibilité de garantir la propriété de non-répudiation dans les schémas de signature électronique. [s6]

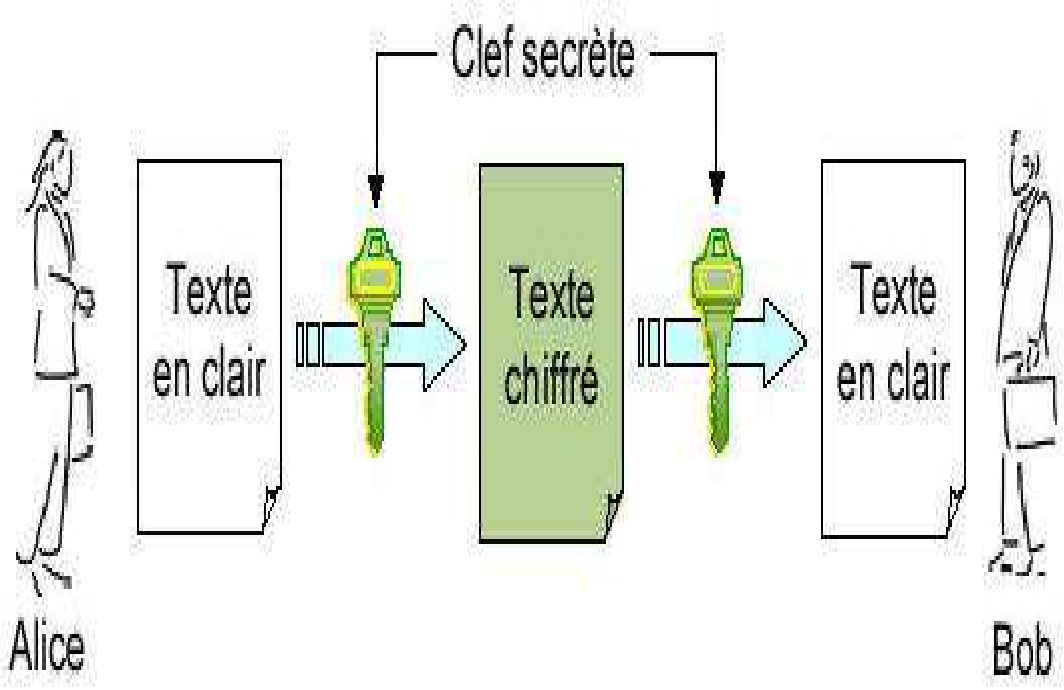


Fig.2.2 : chiffrement symétrique

2.4.2. Cryptage asymétrique

Le cryptage asymétrique, contrairement au symétrique, se base sur l'utilisation des 2 clés : publique (pour crypter, elle est accessible publiquement) et privée (pour décrypter le message, elle est gardée secrète). Ce type de cryptage élimine la problématique de la transmission de la clé (fameuse valise pour le Téléphone Rouge). Ce mode de cryptage est également nommé le cryptage à clé publique. Il est essentiel

Que l'on ne puisse pas déduire la clé privée de la clé publique.

Pour bien comprendre le principe, on peut l'illustrer avec l'échange d'une lettre entre un émetteur et un destinataire :

- l'émetteur possède deux clés : privé et publique. Il envoie sa lettre contenant la clé publique au destinataire.
- le destinataire utilise la clé publique pour crypter son message; il envoie tout à l'émetteur initial
- l'émetteur utilise sa clé privée pour décrypter le message

Un exemple d'utilisation du cryptage asymétrique est la transmission d'une clé secrète dans SSL. Dans la première phase de l'échange, le serveur envoie sa clé publique au client. Ensuite le client valide sa fiabilité. Si la validation est correcte, il génère un pré clé principale avec l'utilisation de la clé publique du serveur. Le résultat de cette génération est ensuite envoyé au serveur.

Les avantages du cryptage asymétrique

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

-Une paire de clés (publique/secrète) peut être utilisée plus longtemps qu'une clé Symétrie.

Les inconvénients du cryptage asymétrique

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

-Taille des clés, plus grand que celle des systèmes symétriques. [s6]

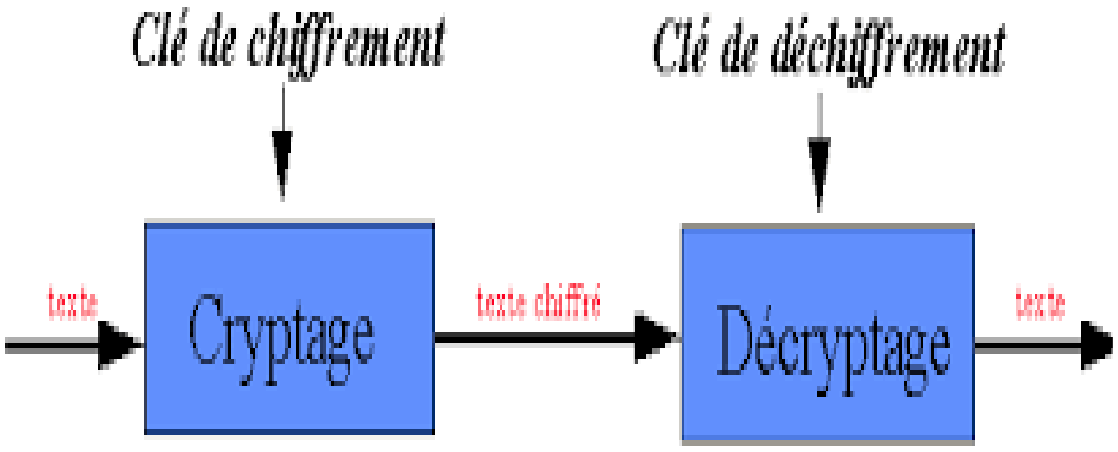


Fig.2.3 : chiffrement asymétrique

2.5. Conclusion

Dans ce chapitre, nous avons présenté les deux différents types de chiffrement symétrique et asymétrique et on a fait une comparaison (avantages et inconvénients) entre ces deux types. Dans le chapitre suivant on va détailler le principe de chiffrement des messages texte par des algorithmes à clé secrète.

Chapitre 3

3.1. Introduction

Les systèmes de cryptage à clé secrète, appelés aussi systèmes de cryptage symétrique ou cryptage conventionnel, sont utilisés depuis plusieurs siècles déjà. C'est l'approche la plus authentique du chiffrement de données et mathématiquement la moins problématique.

Dans ce chapitre, nous présentons les différents algorithmes de chiffrement à clé secrète des messages texte.

3.2. Algorithme de chiffrement à clef secrète

La cryptographie symétrique est très utilisée et se caractérise par une grande rapidité (opérations simples, chiffrement à la volée) et par des implémentations aussi bien software que hardware ce qui accélère nettement les débits et autorise son utilisation massive.

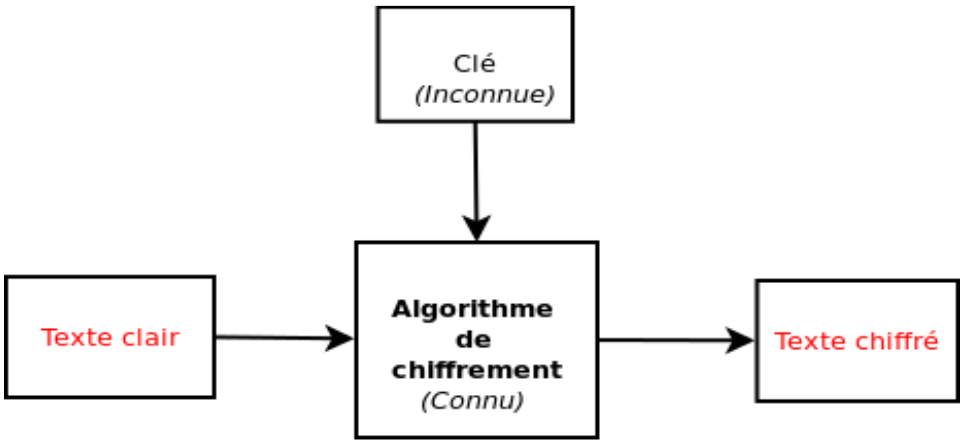


Fig. 3.1 : Algorithme de chiffrement

3.2.1. Algorithme de chiffrement par bloc

Dans un schéma de chiffrement par blocs, le message est divisé en blocs de bits, de longueur fixe. Les blocs sont chiffrés l'un après l'autre. Le chiffrement peut être effectué par substitutions (les bits d'un bloc sont substitués par d'autres bits) et par transpositions (les bits d'un bloc sont permutés entre eux). La substitution permet d'ajouter de la confusion, c'est-à-dire de rendre la relation entre le message et le texte chiffré aussi complexe que possible. La

transposition permet d'ajouter de la diffusion, c'est-à-dire de réarranger les bits du message afin d'éviter que toute redondance dans le message ne se retrouve dans le texte chiffré.

On distingue le chiffrement par blocs itératifs. Une fonction constituée de combinaisons complexes de substitutions et/ou de transpositions, appelée fonction de tour ou fonction de ronde, est appliquée itérativement. Une itération est appelée un tour ou une ronde. Chaque ronde prend en entrée la sortie de la ronde précédente et chiffre cette entrée à l'aide de la fonction de ronde et d'une sous-clé de ronde générée à partir de la clé secrète K. La fonction de chiffrement n'est pas la fonction de ronde, mais elle est constituée par l'ensemble de toutes les rondes. [9]

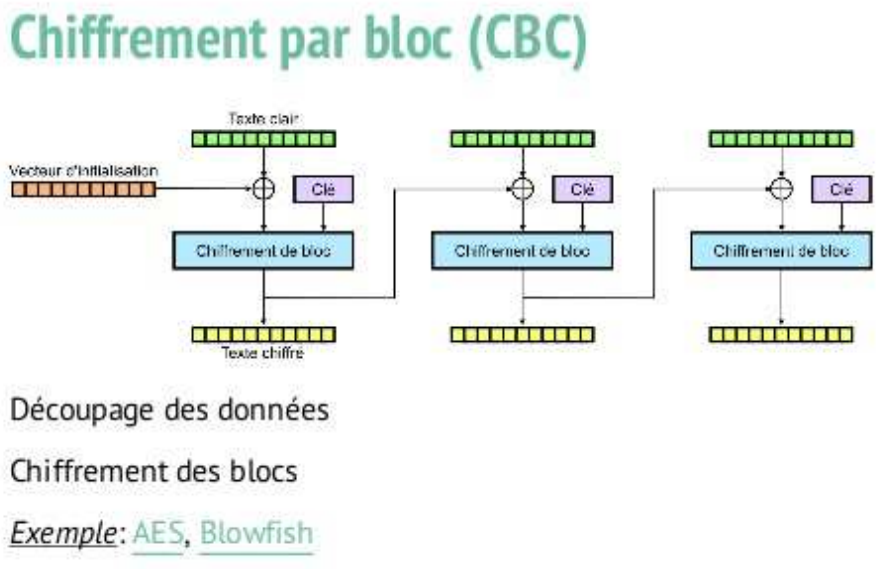


Fig. 3.2 : chiffrement par bloc

Le cryptage XOR

Le cryptage XOR est un système de cryptage basique mais pas trop limité. Ainsi, il a beaucoup été utilisé dans les débuts de l'informatique et continue à l'être encore aujourd'hui car il est facile à implémenter, dans toutes sortes de programmes. [s7]

Mécanisme

Le XOR est un opérateur logique qui correspond à un "OU exclusif" : c'est le (A OU B) qu'on utilise en logique mais qui exclue le cas où A et B sont simultanément vrais. Voici sa table de vérité :

Table de vérité du XOR		
A	**B**	**(A XOR B)**
FAUX	FAUX	FAUX
FAUX	VRAI	VRAI
VRAI	FAUX	VRAI
VRAI	VRAI	FAUX

En informatique, chaque caractère du message à coder est représenté par un entier, le code ASCII. Ce nombre est lui-même représenté en mémoire comme un nombre binaire à 8 chiffres (les bits). On choisit une clé que l'on place en dessous du message à coder, en la répétant autant de fois que nécessaire, comme dans le cryptage de Vigenère. Le message et la clé étant converti en binaire, on effectue un XOR, bit par bit, le 1 représentant VRAI et le 0 FAUX. Le résultat en binaire peut être reconverti en caractères ASCII et donne alors le message codé.

L'algorithme est complètement symétrique : la même opération est réappliquée au message final pour retrouver le message initial.

Remarque : Parfois, on applique une permutation circulaire aux bits du message final pour donner le message codé.

Exemple1 :

Voici le mot MESSAGE converti en binaire :

Lettres	**M**	**E**	**S**	**S**	**A**	**G**	**E**
Codes ASCII	77	69	83	83	65	71	69
Binaire	01001101	01000101	01010011	01010011	01000001	01000111	01000101

Le mot CLE en binaire est lui représenté par 01000011 - 01001100 - 01000101.

Message en binaire	01001101	01000101	01010011	01010011	01000001	01000111	01000101
Clé en binaire (répétée si nécessaire)	01000011	01001100	01000101	01000011	01001100	01000101	01000011
Message crypté	**00001110**	**00001001**	**00010110**	**00010000**	**00001101**	**00000010**	**00000110**

en binaire							

Substitution :

La substitution consiste effectuer des dérivations pour que chaque caractère du message chiffré soit différent des caractères du message en clair. Le destinataire légitime du message applique la dérivée inverse au texte chiffré pour recouvrer le message initial. La complexité des systèmes à substitutions dépend de trois facteurs :

- la composition spécifique de l'alphabet utilisé pour chiffrer ou pour communiquer,
- le nombre d'alphabets utilisés dans le cryptogramme,
- la manière spécifique dont ils sont utilisés.

On distingue couramment quatre types de substitutions différentes :

Substitution simple ou substitution monoalphabétique :

chaque caractère du texte en clair est remplacé par un caractère correspondant dans le texte chiffré. Les exemples les plus célèbres sont les algorithmes de César, Rot13, et bien évidemment le code morse. Ils sont encore utilisés aujourd'hui pour cacher le sens de certains messages (par exemple la solution de certains jeux dans des journaux), mais bien sûr elles sont très peu sûrs. En effet avec ce principe, les lettres les plus fréquentes dans le texte en clair restent les plus fréquentes dans le texte chiffré, il ne cache donc pas les fréquences d'apparition des caractères. C'est une faiblesse importante puisque des techniques statistiques peuvent être utilisés pour associer aux lettres les plus fréquentes, une lettre probable et en appliquant une technique sémantique récursive, les algorithmes à base de substitutions monoalphabétiques sont facilement cassés par les spécialistes.

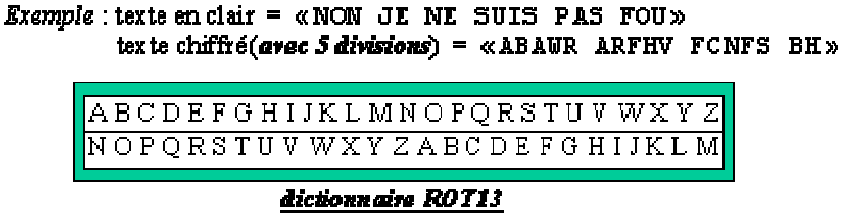


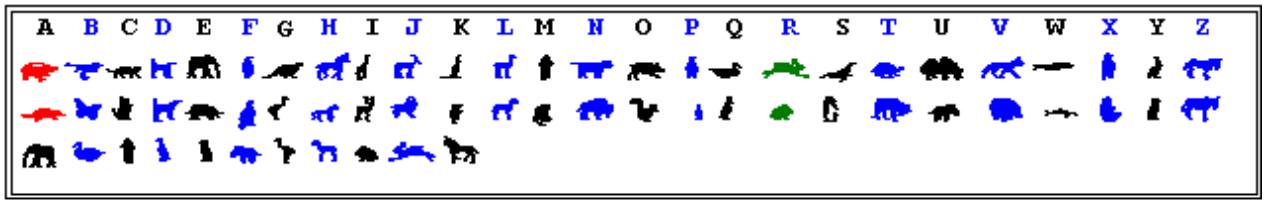
Fig. 3.3 : substitution

Substitution homophonique :

comme pour le principe précédent, sauf qu'à un caractère du texte en clair on fait correspondre plusieurs caractères dans le texte chiffré. Par exemple, " A " peut correspondre à 5, 13, 25 ou 56 ; " B " 7, 19, 31, ou 42 ; etc. Ce procédé est plus sûr , mais aussi craqué par les cryptanalystes ou des espions expérimentés.

Exemple : texte en clair = « CHANGEONS LES MENTALITES FRANCAISES »

texte chiffré = «  »



dictionnaire de substitution homophonique

Substitution polyalphabétique :

le principe consiste à remplacer chaque lettre du message en clair par une nouvelle lettre prise dans ou plusieurs alphabets aléatoires associés. Par exemple, on pourra utiliser n substitutions monoalphabétiques ; celle qui est utilisée dépend de la position du caractère à chiffrer dans le texte en clair. On choisit une clé qui sert d'entrée dans la grille polyalphabétique incluant autant de symboles qu'il y a de lettres différentes à chiffrer. Chaque caractère de la clé désigne une lettre particulière dans la grille de codage. Pour coder un caractère, on doit lire le caractère correspondant du texte en clair en utilisant la grille polyalphabétique et le mot clé associé dans l'ordre séquentiel (on répète la clé si la longueur de celle-ci est inférieure à celle du texte de départ). L'exemple le plus célèbre est l'algorithme de VIGENERE et de BEAUFORT. L'illustration la plus simple qui corresponde à ce principe est l'utilisation d'une fonction à base de ou exclusif (XOR).

Exemple : texte en clair = «ABCBACCBA ACBB»
clé = «DBBCBAACD »

	ABCD
A	CBDA
B	DCAB
C	CABD
D	BDAC

Fig.3.4 : polyalphabetic grid

Substitution par polygrammes :

les caractères du texte en clair sont chiffrés par blocs. Par exemple, " ABA " peut être chiffré par " RTQ " tandis que " ABB " est chiffré par " SLL ". Les exemples les plus célèbres sont les

algorithmes de PLAYFAIR et de HILL inventés en 1854 et utilisés pendant le première guerre mondiale par les anglais. [s8]

Exemple : texte en clair = « POUR LA LEGALISATION DE LA CRYPTO »
texte chiffré = « CESLASOCOCROCOQUIPIKASLEKUSS »
(sans division)

fig.3.5 : Substitution par **poly grammes**

Transposition :

Avec le principe de la transposition toutes les lettres du message sont présentes, mais dans un ordre différent. Il utilise le principe mathématique des **permutations**. Plusieurs types différents de transpositions existent : [s8]

- **Transposition simple par colonnes** :

on écrit le message horizontalement dans une matrice prédéfinie, et on trouve le texte à chiffrer en lisant la grille verticalement (cf. la figure ci-dessous). Le destinataire légal pour décrypter le message réalise le procédé inverse. L'algorithme allemand ADFGFX est fondé sur ce principe et fut utilisé pendant la première guerre mondiale. Il fut cassé par une jeune étudiante française.

Exemple . texte à chiffre: – « I LOVE MY ENGLISH TEACHER »
utilise une matrice [6;4]

I	L	O	V
E	M	Y	E
N	G	L	I
S	H	T	E
A	C	H	E
R			

Texte chiffré = « IENESA RLMGH COYLT HVETE E »

Fig.3.6 : transposition simple par colonnes

- **Transposition complexe par colonnes** :

Un mot clé secret (avec uniquement des caractères différents) est utilisé pour dériver une séquence de chiffres commençant à 1 et finissant au nombre de lettres composant le mot clé. Cette séquence est obtenue en numérotant les lettres du mot clé en partant de la gauche vers la droite et en donnant l'ordre d'apparition dans l'alphabet. Une fois que le séquence de transposition est obtenue, on chiffre en écrivant d'abord le message par lignes dans un rectangle (comme le dessin ci-dessous le montre), puis on lit le texte par colonnes en suivant l'ordre déterminé par la séquence.

Exemple : texte en clair = «I LOVE MY ENGLISH TEACHER »
utilise le mot clé **SERGIO.**

Clé :

S	E	R	G	I	O
6	1	5	2	3	4
I	L	O	V	E	M
Y	E	N	G	L	I
S	H	T	E	A	C
H	E	R			

Texte chiffré = « LEHEV GEELA MICON TRIYS H »

Fig.3.7 : transposition complexe parcolonnes

- **Transposition par carré polybique** :

un mot clé secret est utilisé pour construire un alphabet dans un tableau. Les coordonnées des lignes et des colonnes correspondant au lettres du texte à chiffrer sont utilisés pour transcrire le message en chiffres. Avec ce procédé chaque lettre du texte en clair est représenté par deux chiffres écrit verticalement. Ces deux coordonnées sont ensuite transposées en les recombinant par deux sur la ligne ainsi obtenue.

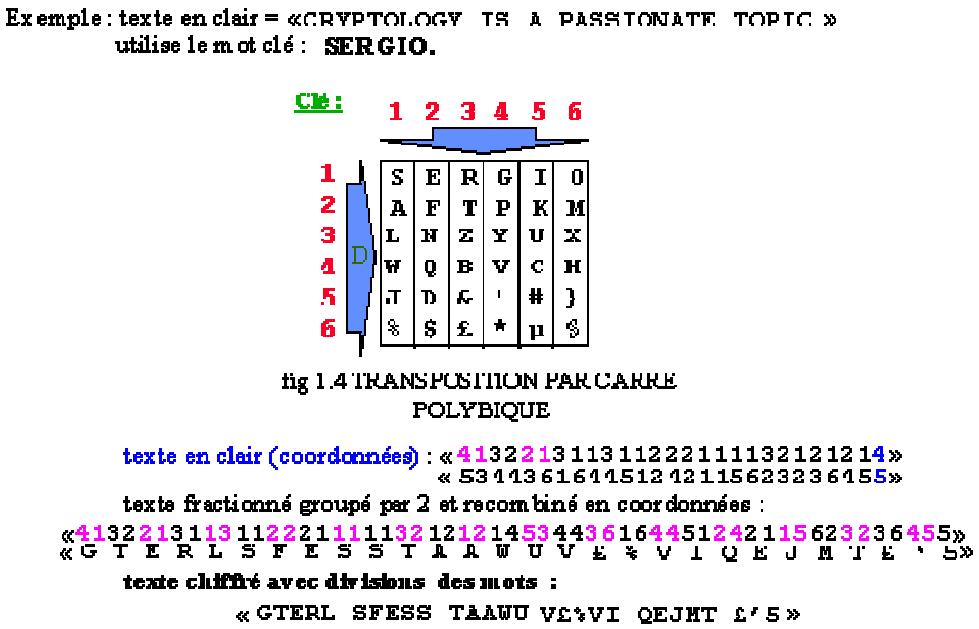


Fig.3.8 : transposition par carre polybique

Il est important de faire remarquer que les transpositions sont plus contraignantes que les substitutions, car elles ont besoin de plus de mémoire et ne fonctionnent que sur des messages à chiffrer d'une longueur limitée ; c'est pourquoi elles sont moins utilisées dans les algorithmes bien que pourtant un peu plus sûres que les substitutions.

Chiffre de César

Le **chiffre de César** (ou *chiffrement par décalage*) est un algorithme de chiffrement très simple que Jules César utilisait pour chiffrer certains messages qu'il envoyait.
Il s'agit d'une **substitution mono-alphabétique** car il remplace chaque lettre par une autre lettre de l'alphabet, toujours la même. [s9]

Principe de chiffrement

Ce cryptosystème consiste à remplacer chaque lettre du texte clair, par une lettre différente, située xlettres après dans l'alphabet, où x est la valeur de la clé passée en argument.
Si l'on considère que chaque lettre de l'alphabet est numérotée de 0 à 25 (A = 0, B = 1, etc.), cela revient à additionner la valeur de la lettre du texte clair avec la valeur de la clé.
Ensuite, pour éviter que la valeur dépasse le 'Z', on fait *modulo 26* sur le résultat précédent.
Rappelons que le modulo est une opération mathématique permettant d'obtenir le reste d'une division. Si vous n'avez peut-être eu que peu d'occasions d'étudier celle-ci, l'ordinateur quant à lui la considère comme une opération de base, au même titre que l'addition, la soustraction, la multiplication et la division.
Notez que faire un modulo 26 revient à soustraire le résultat de 26 lorsqu'il dépasse cette dernière valeur.

Par exemple, pour coder la première lettre de l'expression « J'adore les maths ! » (le 'J') avec une clé de 4, on fera :

(J + 4) modulo 26 = (9 + 4) modulo 26 = 13 modulo 26 = 13

La valeur de 14 correspondant à la lettre 'N', on peut d'ores et déjà noter que la première lettre du texte chiffré sera un 'N'.
Si l'on suit cette logique, à la fin du chiffrement vous devriez obtenir sur votre feuille de papier ce qui suit.

J'ADORE LES MATHS !
N'EHSVI PIW QEXLW !

Le texte chiffré est donc : « N'EHSVI PIW QEXLW ! ».

Principe de déchiffrement

Le déchiffrement fonctionne sur le même principe que le chiffrement, on prend la lettre située quatre lettres avant dans notre exemple.

N'EHSVI PIW QEXLW !
J'ADORE LES MATHS !

chiffrement avec DES (Data Encryption System)

Le 15 mai 1973 le **NBS** (National Bureau of Standards, aujourd'hui appelé NIST - National Institute of Standards and Technology) a lancé un appel dans le Federal Register (l'équivalent aux Etats-Unis du Journal Officiel en France) pour la création d'un algorithme de chiffrement répondant aux critères suivants :
posséder un haut niveau de sécurité lié à une clé de petite taille servant au chiffrement et au déchiffrement être compréhensible ne pas dépendre de la confidentialité de l'algorithme
être adaptable et économique être efficace et exportable
Fin 1974, IBM propose « Lucifer », qui, grâce à la NSA (National Security Agency), est modifié le 23 novembre 1976 pour donner le DES (Data Encryption Standard). Le DES a finalement été approuvé en 1978 par le NBS. Le DES fut normalisé par l'ANSI (American National Standard Institute) sous le nom de ANSI X3.92, plus connu sous la dénomination DEA (Data Encryption Algorithm).

- **Principe du DES**

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

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

La clé est codée sur 64 bits et formée de 16 blocs de 4 bits, généralement notés k_1 à k_{16}. Etant donné que « seuls » 56 bits servent effectivement à chiffrer, il peut exister 2^{56} (soit $7.2s10^{16}$) clés différentes !

- **L'algorithme du DES**

Les grandes lignes de l'algorithme sont les suivantes :

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

Recollement des parties gauche et droite puis permutation initiale inverse. [s8]

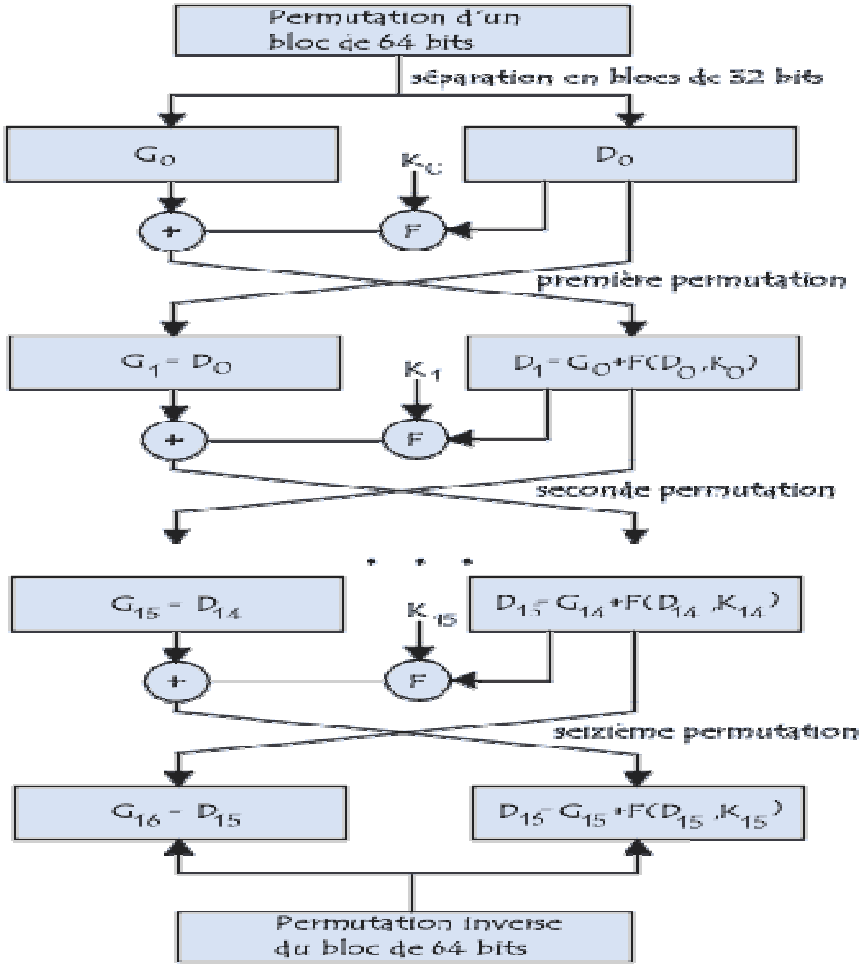


Fig.3.9 : DES (data encryption standard)

3.2.2. Algorithme de chiffrement par flot

Les schémas de chiffrement par flot [26] et appelé aussi chiffrement en continu, traitent l'information bit à bit, et sont très rapides. Ils sont parfaitement adaptés à des moyens de calcul et de mémoire (cryptographie en temps réel) comme la cryptographie militaire, ou la cryptographie entre le téléphone portable GSM et son réseau.

Leur principe est d'effectuer un chiffrement de Vernam en utilisant une clé pseudo-aléatoire, c'est à dire une clé qui ne soit pas choisie aléatoirement parmi tous les mots binaires de longueur *n*. Cette clé (qu'on appellera par la suite pseudo-aléatoire) est générée par différents procédés à partir d'une clé secrète d'une longueur juste suffisante pour résister. [9]

RC5

Le RC5 a été conçu en 1995. Il a l'avantage d'avoir une longueur de bloc de données variable, un nombre de rounds variable et une clé de longueur variable. Ainsi, l'utilisateur a le contrôle sur le

rapport entre la vitesse d'exécution et la sécurité de son chiffrement. En général, une longue clé et un nombre élevé de rounds assurent une plus grande sécurité. La taille des blocs de données pour sa part accommode différentes architectures de systèmes.

La simplicité de l'algorithme du RC5 rend son implémentation facile et, le plus important, rend son analyse plus aisée. De plus, la forte utilisation des décalages de bits (appelés rotations) dans le chiffrement prévient l'usage de la cryptanalyse linéaire et différentielle.

Chifffement

Il y a deux parties dans l'algorithme, soit une procédure d'expansion de la clé et une procédure de chiffrement. Les opérations utilisées sont l'addition modulo $2^{(\text{nombre de bits des blocs})}$ (+), le OU-Exclusif (XOR) et le décalage de bits vers la gauche (<<<).

En équations :

Soient K[0], K[1], ..., K[n] les sous-clés dérivées de la procédure d'expansion de la clé et A et B les deux parties d'un bloc de texte clair à chiffrer.

A = A + K[0]

B = B + K[1]

Pour i allant de 1 jusqu'au nombre de rounds

A = ((A XOR B) <<< B) + K[2i]

B = ((B XOR A) <<< A) + K[2i + 1]

Fin Pour

Déchiffrement

Le déchiffrement est exactement l'inverse du chiffrement. [s10]

Le RSA (algorithme de chiffrement asymétrique)

Le principe

Le premier système à clé publique solide à avoir été inventé, et le plus utilisé actuellement, est le système RSA. Publié en 1977 par Ron **R**ivest, Adi **S**hamir et Leonard **A**dleman de l'Institut de technologie du Massachusetts (MIT), le RSA est fondé sur la difficulté de factoriser des grands nombres, et la fonction à sens unique utilisée est une fonction "puissance".

L'algorithme de chiffrement

Départ :

Il est facile de fabriquer de grands nombres premiers p et q (+- 100 chiffres)

Etant donné un nombre entier n = pq, il est très difficile de retrouver les facteurs p et q

(1) Création des clés

La clé secrète : 2 grands nombres premiers p et q

La clé secrète : 2 grands nombres premiers p et q

- La clé publique : n = pq ; un entier e premier avec (p-1)(q-1)

(2) Chiffrement : le chiffrement d'un message M en un message codé C se fait suivant la transformation suivante :

$$C = M^e \bmod n$$

(3) Déchiffrement : il s'agit de calculer la fonction réciproque

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

La signature électronique

Après la confidentialité de la transmission d'un message subsiste un problème : son authenticité. Alice voudrait bien envoyer un message M à Bob de telle façon que celui-ci soit sûr qu'elle est réellement l'émettrice du message, et qu'un intrus ne tente pas de venir semer la confusion.
Le système RSA fournit une solution à ce problème :

Rappelons les données :

- Alice seule détient la clé secrète d et diffuse la clé publique (n,e)
- Alice va se servir de la clé publique pour chiffrer le message M

(1) Alice accompagne son message chiffré de sa **signature**, qui correspond à :

$$M^d$$

(2) Bob va donc voir si l'égalité $(M^d)^e \bmod n = M$ est vérifiée. Si c'est le cas, Alice est bien l'émettrice du message.[s11]

3.3. Conclusion

Dans ce chapitre on a décrit quelques algorithmes de chiffrements à clé secrète les plus utilisés, mais bien sur il en existe beaucoup d'autres. Ces algorithmes jouent un rôle important aussi bien au niveau de la sécurité que des performances.

Chapitre 4

4.1. Introduction

Dans ce chapitre, nous présentons notre application qui permet de chiffrer des messages sécuriser par un model client/ serveur entre deux ordinateurs.

le premier : Intel(R) core (TM)i3-3217U CPU à 1.8 GHz avec une RAM 2.00 Go et un système d'exploitation Windows 7.

Le deuxième : AMD E1-2100APU with Radeom(TM) HD CPU à 1.00 GHz avec une RAM 2.00 et un système d'exploitation Windows 7.

L'application de chiffrement à été développé par le compilateur c++ builder version 6

Notre application est basée sur le modèle client / serveur.

4.2. Présentation de l'application

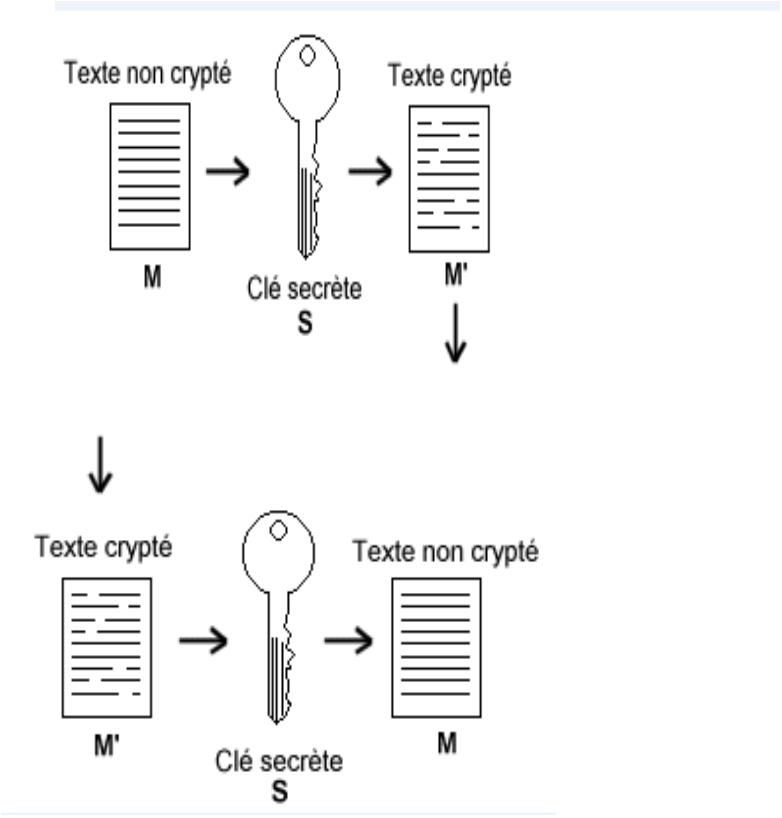


Fig 4.1 : idée général pour réaliser interface de client

4.3. L’interface de l’application

Client

La fenêtre d’interface de notre client est présentée ci-dessus par la figure :

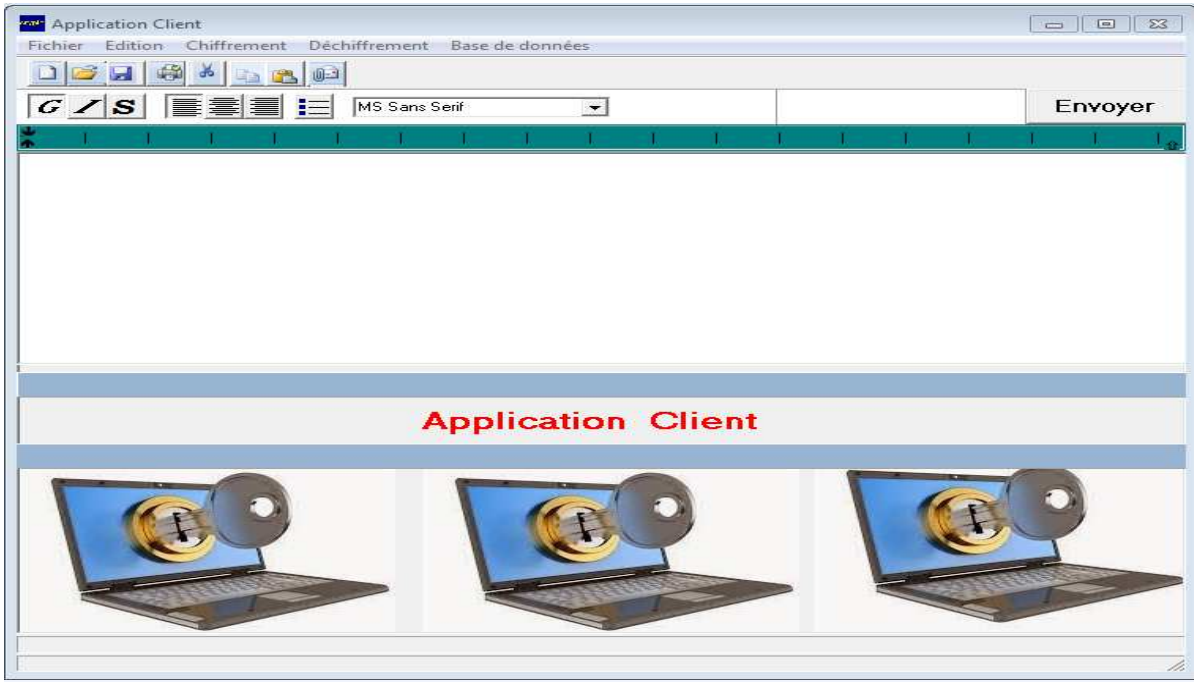


Fig.4.2 : interface de client

Le menu de programme

Le button envoyer : permet d'envoyer des messages a le serveur.

le menu de chiffrement contient les algorithmes suivants :

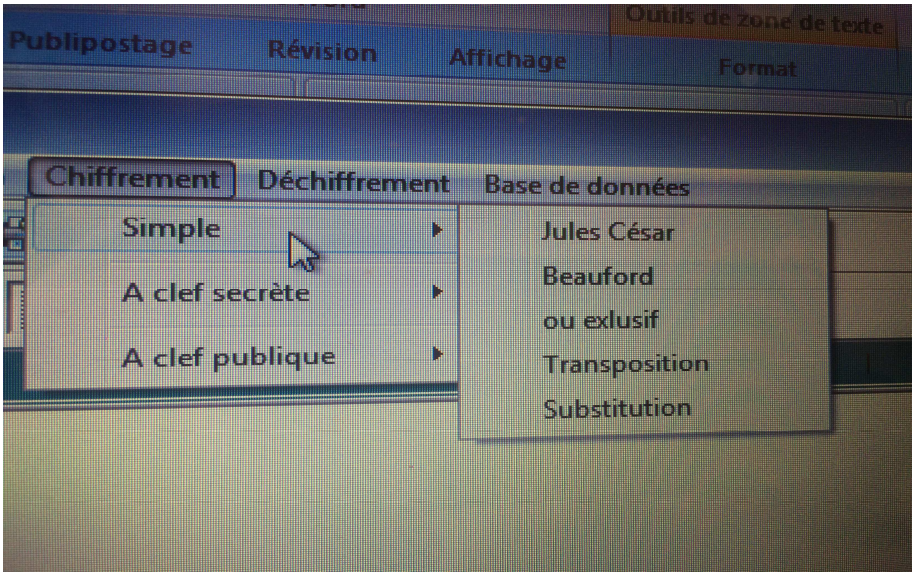


Fig.4.3 : chiffrement

Serveur

La fenêtre d'interface de notre serveur est présentée ci-dessus par la figure :

Fig.4.4 : interface de serveur

le menu de déchiffrement contient les algorithmes suivants

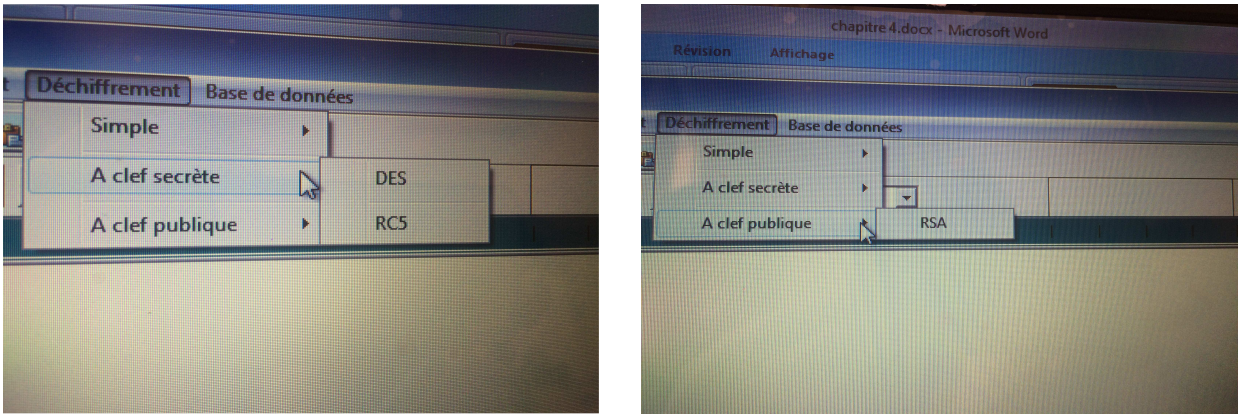


Fig.4.5 : déchiffrement

4.4. Exemple sur chiffrement d'un message

Message claire

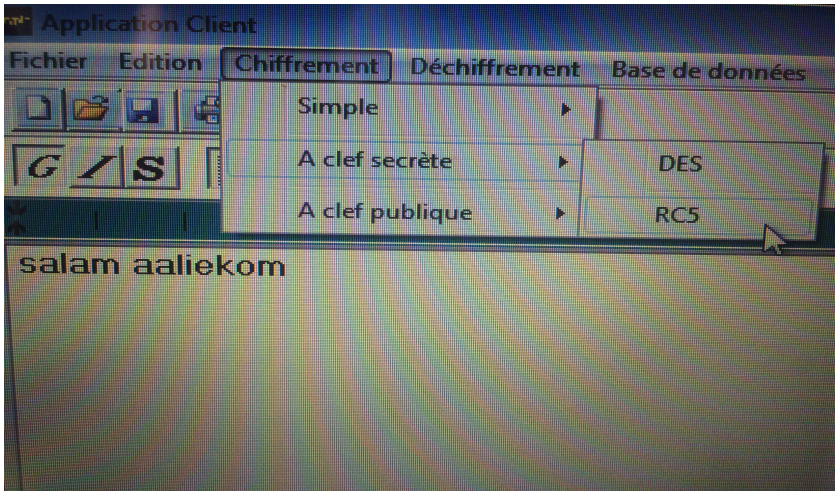


Fig.4.6 : message claire

Pour chiffré ce message il faut un code de sécurité (dans l'exemple on prend 14 comme code de sécurité)

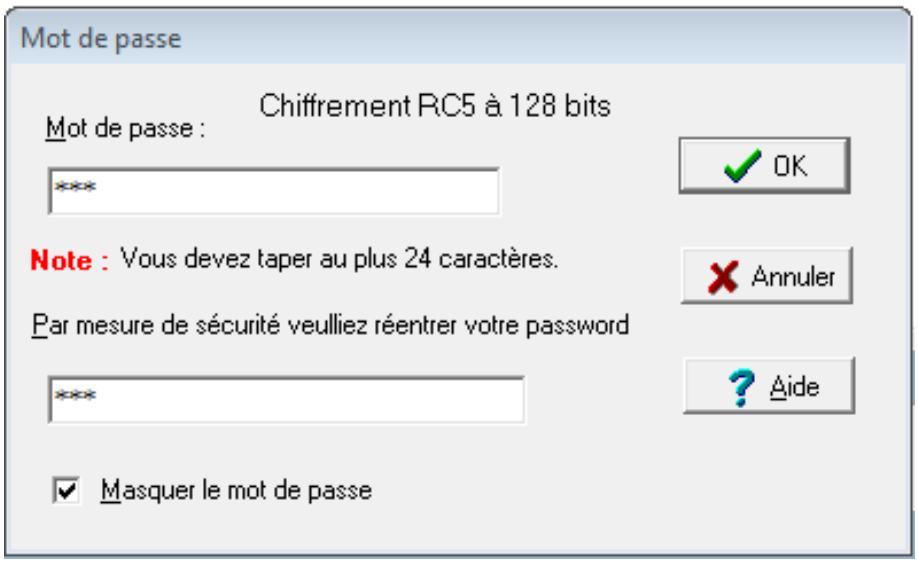


Fig.4.7 : zone de mot de passe

Message chiffré

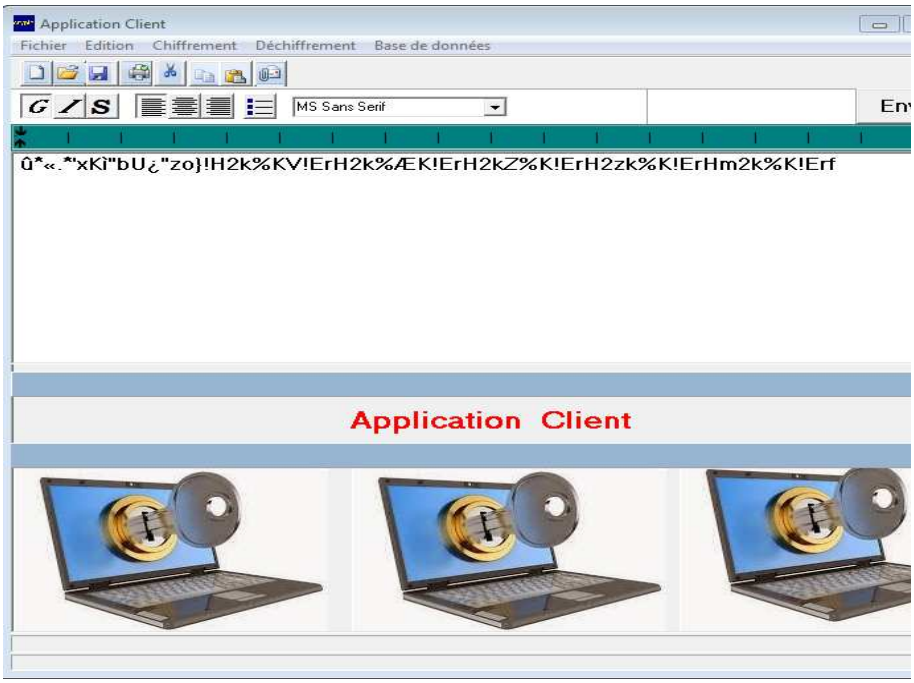


Fig.4.8 : message crypté

Donc on click sur bouton **Envoyer**

Message déchiffré

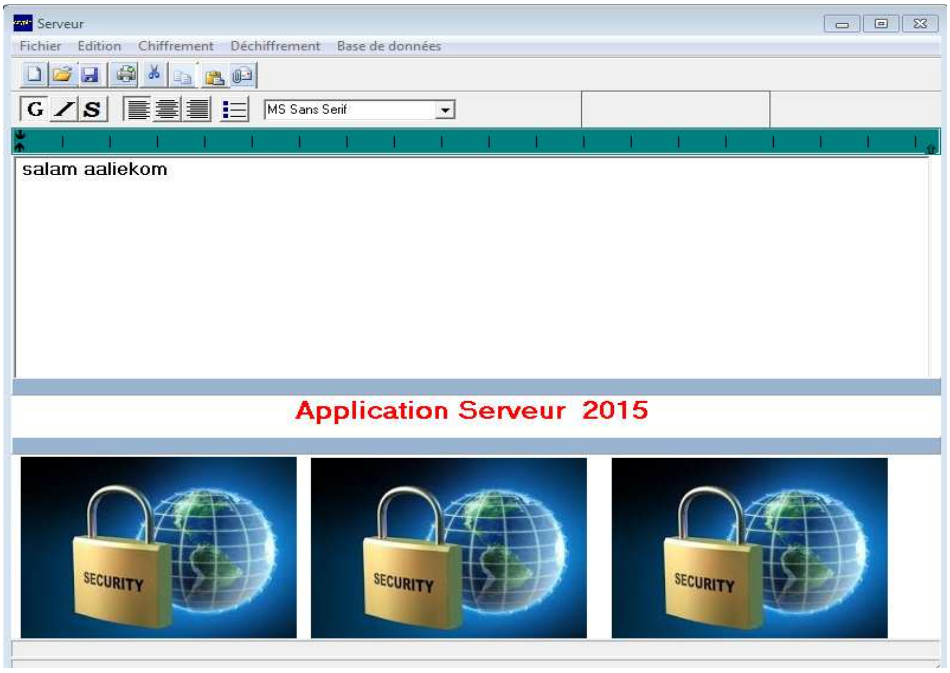


Fig.4.9 : message décrypté

4.5. Conclusion

La cryptographie est une science fondamentale et importante dans la sécurisation des informations transmission dans un réseau.

Notre objectif principale est de développer une application pour sécuriser la transmission des messages en utilisons une architecteur client et serveur.

Nous avons testés plusieurs algorithmes de chiffrement afin d'assurer une sécurité et une protection de notre données qui est circulées dans notre réseau.

Conclusion générale

D’après la préparation de notre projet, nous avons appris beaucoup d’informations et aussi nous avons utilisé les connaissances acquises durant nos études universitaires, les livres, les mémoires des années passés et les ressources internet.

D’âpres notre étude approfondie, on a remarqué que les recherches ont été basées au premier lieu sur la sécurité d’un crypto système. En pratique un système est sur tant que personne ne l’a cassé et par conséquent le défi actuel est de fournir des paramètres efficace garantissant de manière si possible prouvée une très forte sécurité qui ne sera pas détourné a des fin malhonnête

La cryptographie est une science fondamentale et importante dans la sécurisation des informations transmission dans un réseau.

Au cours de ce projet, nous sommes intéressés au chiffrement des messages sur un réseau de communication.

Donc, Notre objectif principale est de développer une application pour sécuriser la transmission des messages en utilisant une architecture client et serveur.

Nous avons testés plusieurs algorithmes de chiffrement afin d’assurer une sécurité et une protection de notre données qui est circulées dans notre réseau.

Référence

[1]- Mr BENAISSA Mohamed Cours master-RSD-M1-2015 **Architecture Client/Serveur**

[2]-Mr MICHEL SIMATIC Cours master 2- TELECOM (Sud Paris module CSC4508/M2 avril2012

[3]- Ghefir Mohamed El Amine et Bendoukha Sidi Ahmed, Développement d'une Application Distribuée par la méthode java RMI, Juillet 2006.

[4]- Mr Laouedj Mounir Mémoire master en Informatique Cryptage et sécurité des flux vidéo (modèle client/serveur) septembre 2014

[5]- Network Programming for Microsoft Windows Second Edition, MSPress2002

[6]- Mme BELKHOUCHE Souheyla Mémoire pour l'Obtention du Diplôme D'Ingénieur d'Etat en Informatique « Etude et Administration des Systèmes de Supervision dans un Réseau Local » **2011**

[7]- Simon Guillem –Lessard projet de fin d'étude 2001-2002 Département des mathématiques et de l'informatique Université du Québec à Trois-Rivières.

[8]- Jean-Guillaume Dumas, Jean-Louis Roch, Éric Tannier, Sébastien Varrette LIVRE (THÉORIE DES CODES compression, cryptage, correction).

[9]- NKAPKOP Jean De Dieu ,Mémoire de Master en EEA. Cryptage chaotique des images basé sur le modèle du perceptron Université de Ngaoundéré.

Référence site web

[s1] -http://www.httr.ups-tlse.fr/pedagogie/cours/internet/services/servclie.htm

[s2] -w3.polytech.univ-montp2.fr/~karen.godary/M1/Trans_Client_Serveur.pdf

[s3] - http://www.vulgarisation-informatique.com/ports.php

[s4] -http://dspace.univ-tlemcen.dz/bitstream/112/3457/1/belkhouche.pdf

[s5] - http://sarasad1987.centerblog.net/2-cryptographie

[s6]-http://www.bart-konieczny.com/fr/blog/securite-des-applications-web/cryptage-symetrique-et-asymetrique

[s7]- http://www.primenumbers.net/Renaud/fr/crypto/XOR.htmhttp://

[s8]- http://nopb.chez.com/crypto2.html

[s9]- http://openclassrooms.com/courses/les-premiers-algorithmes-de-chiffrement

[s10]- http://www.uqtr.uquebec.ca/~delisle/Crypto/prives/blocs_rc.php

[s11]- http://www.cryptage.org/rsa.html

Résumé :

La cryptographie est une science fondamentale et importante dans la sécurisation des informations transmission dans un réseau.

Notre objectif principale est de développer une application pour sécuriser la transmission des messages en utilisant une architecture client et serveur.

Nous avons testés plusieurs algorithmes de chiffrement afin d'assurer une sécurité et une protection de notre données qui est circulées dans notre réseau.

Mots clés : réseau, sécurité, cryptographie, chiffrement symétrique ; chiffrement asymétrique, modèle client/serveur

Summary:

The cryptography is a fundamental and important science in the reassurance of the information transmission in a network.

Our objective essential to develop an application to secure the transmission of messages use customer architecture and a server.

We made out a will several algorithms of encryption to assure security and a protection of our data which is circulated in our network.

Keywords: network, security, cryptography, symmetric encryption; asymmetric encryption, models customer / server.

ملخص:

التشفير هو احد العلوم الأساسية و الهامة في التأكد من نقل المعلومات في الشبكة.

هدفنا الرئيسي هو تطوير تطبيق لضمان أمن إرسال رسائل وذلك باستخدام عميل و خادم.

لقد قمنا باختبار العديد من خوارزميات التشفير و ذلك لضمان أمن و حماية لمعلوماتنا الخاصة التي يتم تداولها في الشبكة.

الكلمات المفتاحية : الشبكة،الأمن،الترميز،التشفير المتناظر،،التشفير الغير المتناظر ،ونموذج العميل و الخادم.