REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE
LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOULOUD MAMMERI DE TIZI-OUZOU
FACULTE DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT D'INFORMATIQUE

Mémoire

De fin d'études

En vue de l'obtention du diplôme de Master de recherche en informatique

Option : Réseaux Mobilités et Systèmes Embarqués

Thème

Sécurité Informatique basé sur un cryptosystéme

AES amélioré avec RSA

Proposée et diriger par :

M^{me} HADAOUI Eps Skendraoui

Réalisé par:

M$_r$ BEN SAID Rabah

M$_r$ ARABI YAZID

Membres du jury :

……..

…….....

…….....

Promotion :2014

Remerciements :

Nous remercions tout d'abord ALLAH qui nous a donné la foi pour accomplir ce travail.

Nous tenons aussi à exprimer notre reconnaissance à notre promoteur, M^{me} HADAOUI pour nous avoir encadrés durant cette année, et nous remercions les membres du jury pour avoir accepté d'honorer par leur jugement notre travail. Aussi, nous adressons nos remerciements à tous nos enseignants de l'UMMTO pour nous avoir appris le goût de l'effort et du travail pendant cette période qui présente une très grande valeur pour nous.

Enfin, nous dédions ce travail à toutes nos familles et nos amis en reconnaissance de leurs précieux soutiens et de leurs encouragements.

Table des matières

Chapitre IV : Conception et Réalisation

Table des figures

Introduction Générale

Le rôle de la cryptographie est de garantir la sécurité des communications c'est dire de permettre à des entités qui ne se font pas confiance en général de communiquer en toute sécurité en présence de potentiels adversaires (susceptibles entre autres d'intercepter et de modifier les informations échangées ou d'usurper des identités). Mais la cryptographie à elle seule ne prend pas en charge tous les besoins de sécurité et n'est donc pas la seule discipline qui intervienne dans la sécurité des communications. La cryptographie est composée de la cryptographie symétrique et de la cryptographie asymétrique.

La cryptographie symétrique consiste à échanger des données chiffrées à partir d'un secret (appelé clé) connu uniquement par les parties concernées, le chiffrement comme le déchiffrement nécessitant la clé. Pour cela, il paraît nécessaire d'avoir un protocole d'échange de clés sûr et efficace. De l'antiquité jusque dans les années 1970, tous les systèmes de cryptographie étaient symétriques (donc la principale préoccupation était la confidentialité) et il n'existait pas de méthode sûre pour échanger les clés secrètes.

C'est en 1976 que deux chercheurs White **Diffie** et **Martin Hellmann** à l'université de Stand-ford, dans leur papier "New Direction In Cryptography", ont donné une solution au problème de l'échange des clés et ont proposé en même temps le modèle théorique de la cryptographie à clé publique qui est appelé modèle de cryptographie asymétrique.

Notre problématique se penche sur l'idée d'une unification de deux algorithmes puissants pour augmenter la performance, l'efficacité et la solidité de la sécurité des données échangées sur un réseau, et pour cela notre travail consiste à réaliser une implémentation d'un algorithme de cryptographie asymétrique (RSA) sur une application AES (algorithme de cryptographie symétrique) dans le but de réaliser une application plus solide en termes de sécurité.

Notre travail consiste à concevoir et réaliser un logiciel qui permet de crypter des données en utilisant l'algorithme AES amélioré avec RSA pour sécuriser sa clé symétrique, et notre mémoire sera organisé de la manière suivante :

- **Chapitre I :** présentation de quelque Généralités sur les réseaux informatique et la sécurité réseau ;
- **Chapitre II :** On présentera les bases de la cryptographie ;
- **Chapitre III :** On va étudier l'algorithme AES.
- **Chapitre IV :** la réalisation de notre application en présentons les différents outils nécessaire pour la mise en œuvre de notre projet ainsi on présentera le fonctionnement de celle-ci

Partie I :

État de l'art

Chapitre I : Généralités sur les réseaux informatique et la sécurité réseau.

Chapitre II : La cryptographie.

CHAPITRE I.

GENERALITES SUR LES RESEAUX INFORMATIQUE ET LA SECURITE RESEAU.

Partie I :GENERALITES SUR LES RESEAUX INFORMATIQUE

1. Introduction :

Les réseaux informatiques occupent aujourd'hui une place prépondérante dans la vie professionnelle, sont inventés pour les utilisés dans l'échange de l'information pour le besoin de communication. Les types de données transmises sont très variés : parole, sons, photos, vidéo, fichiers de données... Les réseaux couvrent complètement la planète grâce à divers équipements et l'apparition de l'Internet et le Web offrent des services énormes.

Dans cette partie on présentera quelques généralités sur les réseaux informatiques et les notions de bases utiliser pour la création de ces réseaux.

2. Définition d'un Réseau : [25]

Un réseau en général est le résultat de la connections de plusieurs machines entre elles afin que les utilisateurs et les applications qui fonctionne sur ces dernières puissent échanges des informations.

Le terme réseau en fonction de son contexte peut designer plusieurs choses :

- désigner l'ensemble des machines ou l'infrastructure informatique d'une organisation avec les protocoles qui sont utilisés. Ce qu'est le cas lorsqu'on parle de l'Internet.
- décrire la façon dont les machines d'un site sont interconnectées
- spécifier les protocoles qui sont utilisés pour que les machines communiquent on peut parler de réseau TCP/IP.

3. Rôles des réseaux : [26]

Il est intéressant de mettre en place un réseau, local ou longue distance, pour des raisons techniques d'une part, et orientées vers l'utilisateur d'autre part.

3.1. Objectifs techniques :

Parmi les objectifs techniques des réseaux :

- Le partage des ressources entre plusieurs utilisateurs. Par exemple, mettre des programmes à la disposition de l'ensemble des utilisateurs connectés, peut réduire le nombre d'installations à une seule.
- La fiabilité peut être à la source de la mise en place d'un réseau .Cette architecture permet une duplication des données et limite ainsi les pertes de données.
- Il est évident que le partage de périphériques entraîne directement une réduction des couts. Par exemple, partager une imprimante par plusieurs ordinateurs.

3.2. Objectifs utilisateurs :

Parmi les objectifs utilisateurs des réseaux :

- La communication est sans nul doute l'aspect le plus intéressant pour un utilisateur. Elle peut prendre la forme de courrier électronique, de vidéoconférence,…etc.
- Citons encore des applications en cours d'amélioration : les jeux interactifs, la radio, la vidéo à la demande….

4. Différents types de réseaux :

Le classement des réseaux peut être en fonction de leur distance et de leurs topologies.

4.1. En fonction de la distance : [27]

On distingue en général quatre catégories de réseaux informatiques :

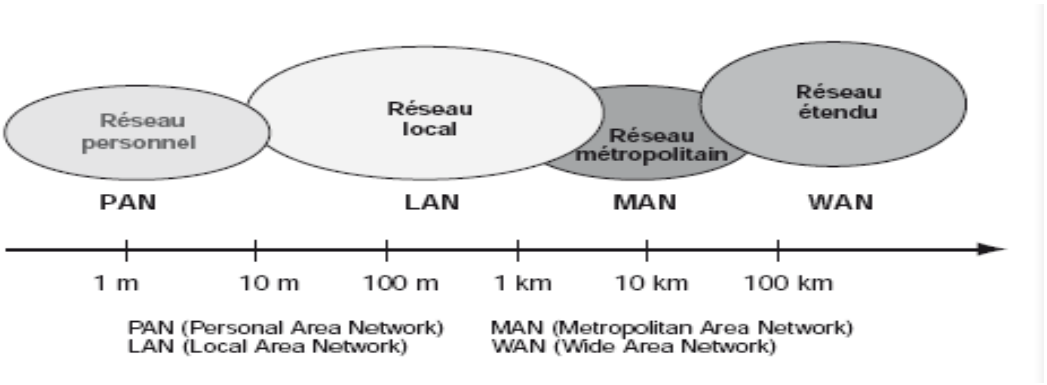


Figure 1 : Les grandes catégories de réseaux informatiques.

- **Les réseaux personnels ou PAN (Personal Area Network)** : Les équipements sont interconnectés par une distance très courte, voire sur quelques mètres, tels que des terminaux GSM, portables, organiseurs etc...
- **Les réseaux locaux, ou LAN (Local Area Network) :** Les réseaux LAN correspondent par leur taille aux réseaux intra-entreprises. Ils servent au transport de toutes les informations numériques de l'entreprise. En règle générale, les bâtiments à câbler s'étendent sur plusieurs centaines de mètres. Les débits de ces réseaux vont aujourd'hui de quelques mégabits à plusieurs centaines de mégabits par seconde.
- **Les réseaux métropolitains, ou MAN (Metropolitan Area Network) :** Les réseaux MAN permettent l'interconnexion des entreprises ou éventuellement des particuliers sur un réseau spécialisé à haut débit qui est géré à l'échelle d'une métropole. Ils doivent être capables d'interconnecter les réseaux locaux des différentes entreprises pour leur donner la possibilité de dialoguer avec l'extérieur.
- **Les réseaux étendus, ou WAN (Wide Area Network) :** Les réseaux WAN sont destinés à transporter des données numériques sur des distances à l'échelle d'un pays, voire d'un continent ou

de plusieurs continents, il utilise des infrastructures au niveau du sol, essentiellement de grands réseaux de fibre optique, ou hertzien, comme les réseaux satellite.

4.2. En fonction de la topologie:[28]

Topologie : Structure des réseaux en termes de lien d'interconnexion entre stations

On distingue généralement 04 types :

- **Topologie en BUS :**

Figure2 : Topologie en BUS

La topologie en bus définit une liaison entre les machines via un câble commun et dans le cas d'un réseau client/serveur les machines sont reliées directement à un serveur. La topologie en bus et largement répandue dans les réseaux locaux Ethernet et adoptée par les réseaux APPLETALK et TOKEN BUS D'IBM. On distingue deux types de topologie en bus :

- UNIDIRECTIONNEL (2 câbles distincts ou 2 canaux multiplexés),
- BIDIRECTIONNEL(Les données circulent dans les 2 sens mais non simultanément).

- **Topologie en ANNEAU :**

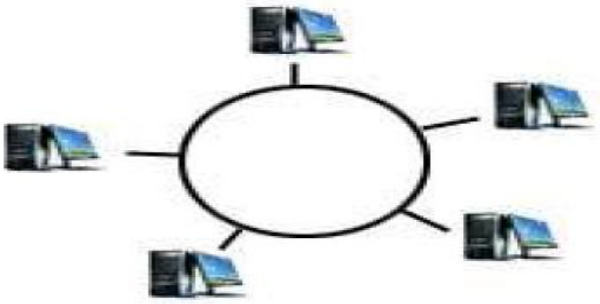


Figure3 : Topologie en ANNEAU

Dans la topologie en anneau, chaque machine est reliée à 2 équipements voisins (boucle fermée), et les données se transfèrent d'une station à l'autre jusqu'à la destination.

Les unités de connexions multiples (MAU : Medium Attachment unit) sont des éléments actifs chargés de recevoir les données et de les orienter (retransmission ou réception). Elle supporte la communication unidirectionnelle ou bidirectionnelle, et adoptée par les réseaux TOKEN, RINGFDDI.

- **Topologie en ETOILE :**

Dans une topologie en étoile, toutes les machines sont reliées à un serveur commun, et le transfert des données se fait à travers le nœud central, et Adoptée par les réseaux STARLAN ARCNET.

Figure4 : Topologie en ETOILE

- **Topologie en ARBRE :**

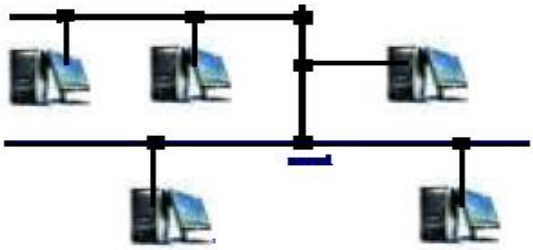


Figure5 :Topologie en ARBRE

La topologie en arbre est une topologie en BUS dans laquelle une nouvelle connexion donne naissance à un nouveau bus, elle nécessite ainsi un répéteur (appareil comprenant essentiellement un ou plusieurs amplificateurs ou régénérateurs et des organes associés).

- **Éléments d'une Topologie :**

Pour utiliser une topologie nous avons besoin d'une carte d'interface réseaux (relient les ordinateurs au support de transmission), un support de communication, des hubs (nœuds de câblage) distribuent les données entre le serveur et les stations.

- **Comment peut-on choisir une Topologie ?**

On peut choisir une topologie selon plusieurs critères :

- Les avantages et inconvénients par rapport aux équipements qu'on va utiliser,

- L'analyse des besoins avant et après notre choix,
- La disponibilité des équipements et des locaux.

5. Architecture des réseaux :[29]

5.1. Description :

Le transport des données d'une extrémité à une autre d'un réseau nécessite un support physique ou hertzien de communication. Pour que les données arrivent correctement au destinataire, avec la qualité de service exigée, il est nécessaire de réaliser une architecture logicielle chargée du contrôle des paquets dans le réseau, mais d'une autre part le développement du grand nombre de fonctionnalités implémentées dans les réseaux a causé une complexité particulière dans l'architecture de réseau, pour cela les architectes réseau ont décomposé les processus à l'œuvre en sept couches protocolaires plus un support physique, ce découpage permet au réseau de traiter en parallèle les fonctions attribuées aux différentes couches.

ISO (International Standardization Organization) a normalisé sa propre architecture sous le nom d'OSI (Open Systems Interconnection) pour réaliser l'interconnexion des architectures hétérogènes, ce modèle est relativement très complexe, car il contient de nombreuses options, destinées à couvrir l'ensemble des fonctionnalités proposées, quel que soit l'environnement d'adaptation.

En revanche, l'architecture TCP/IP répond à tous les besoins possibles, et des fonctionnalités en dehors de tout souci de réalisation. Ainsi on distingue deux grandes architectures qui se disputent actuellement dans le marché mondial des réseaux :

- L'architecture OSI (OPEN SYSTEMS INTERCONNECTION), ou interconnexion de systèmes ouverts, provenant de la normalisation de l'ISO (International Standardization Organization),
- L'architecture TCP/IP utilisée dans le réseau Internet.

5.2. Le modèle OSI :

Le **modèle OSI** (Open Systems Interconnection ou interconnexion de systèmes ouverts) a été mis en place par l'ISO (International Standardization Organization) afin de normaliser les communications entre les ordinateurs d'un réseau. En effet, aux origines des réseaux, chaque constructeur avait un système propre (Système propriétaire) et de nombreux réseaux incompatibles coexistant, et puis le modèle OSI a autorisé de rendre la communication entre les machines standard afin que les différents constructeurs puissent mettre en œuvre des produits (logiciels ou matériels) compatibles. Le modèle OSI est un modèle qui comporte sep couches.

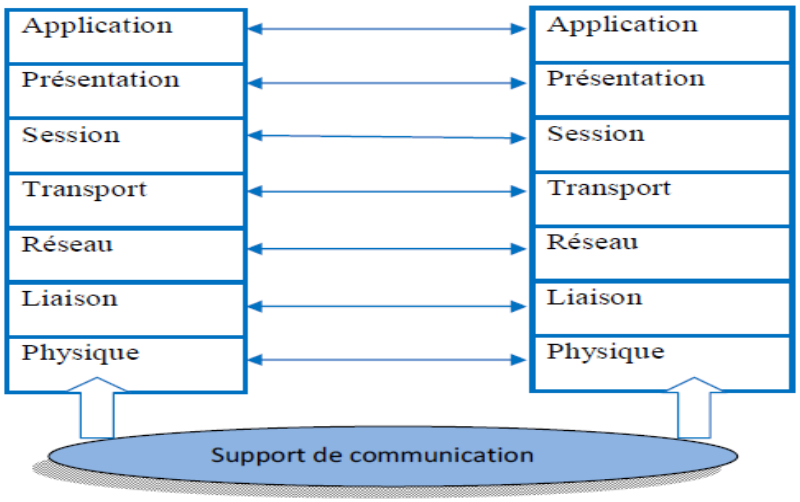


Figure6 : Architecture OSI

a.<u>Fonctionnement</u> :

Le terme de couche a été créé pour évoquer le fait que les données qui transitent sur le réseau passent par plusieurs niveaux de protocoles. Les données (paquets d'informations) qui circulent sont traitées successivement par chaque couche, qui vient rajouter un élément d'informations (en-tête) puis sont transmises à la couche suivante, et l'intérêt est de partager le problème en différentes parties(les couches), ainsi chaque couche communique avec une couche adjacente en utilisant les services de la couche inférieure et fournit des services à la couche de niveau supérieure.

b. <u>Rôles des différentes couches</u> :

- **La couche physique :** définit la façon dont les données sont physiquement converties en signaux numériques sur le media de communication (impulsions électriques, modulation de la lumière etc.).
- **la couche liaison de données** : définit l'interface avec la carte réseau et le partage du média de transmission.
- **la couche réseau** : permet de gérer l'adressage et le routage des données c'est-à-dire leur acheminement via le réseau.
- **la couche transport** : elle est chargée du transport des données, et de leur découpage en paquets ainsi de la gestion des éventuelles erreurs de transmission.
- **la couche session** : définit l'ouverture et la destruction des communications entre les machines du réseau.
- **la couche présentation** : définit le format des données manipulées par le niveau applicatif (leur représentation, éventuellement leur compression et leur chiffrement) indépendamment du système.

- **la couche application** : assure l'interface avec les applications, il s'agit donc du niveau le plus proche des utilisateurs, est géré directement par les logiciels.

5.3. Le modèle TCP/IP:[29]

TCP/IP (Transmission Control Protocol /Internet Protocol) est une suite de protocoles. Cette appellation provient des noms des deux protocoles majeurs TCP et IP.

Le modèle TCP\IP Permet la représentation de l'ensemble des règles de communication sur Internet et elle est basée sur la notion d'adressage IP, c'est-à-dire le fait de fournir une adresse IP à chaque machine du réseau afin de pouvoir acheminer des paquets de données. La suite de protocoles TCP/IP est conçue pour répondre à un certain nombre de critères parmi lesquels :

- le fractionnement des messages en paquets,
- l'utilisation d'un système d'adresses,
- l'acheminement des données sur le réseau (routage),
- le contrôle des erreurs de transmission de données.

Les principaux protocoles faisant partie de la suite TCP/IP sont : **TCP, UDP, IP, ARP, RARP, FTS, FDDI, PPP, Ethernet, Anneau à jeton (Token Ring)**…etc.

Le **modèle TCP/IP** reprend l'approche modulaire du modèle **OSI** (utilisation de modules ou de couches), mais ce modèle ne contient que quatre couches, ainsi que ces couches ont des taches beaucoup plus diverses étant donné qu'elles correspondent à plusieurs couches du modèle OSI.

Niveau	Modèle TCP/IP	Modèle OSI	Protocoles TCP/IP
Niveau 4	Couche Application	Couche Application	Application réseau (Telnet, SMTP, FTP…)
		Couche Présentation	
		Couche Session	
Niveau 3	Couche Transport(TCP)	Couche Transport	TCP ou UDP
Niveau 2	Couche Internet(IP)	Couche Réseau	IP, ARP, RARP
Niveau 1	Couche Accès Réseau	Couche liaison données	FTS, FDDI , PPP, Ethernet, Anneau à jeton (Token Ring)
		Couche Physique	

Tableau 1 : les couches du modèle TCP/IP

6. les protocoles : [25]

6.1. Définition : Un protocole est une méthode standard qui permet la communication entre deux machines c'est-à-dire un ensemble de règles et de procédures à respecter pour émettre et recevoir des données sur un réseau. Il en existe plusieurs selon que l'on attend de la communication. Certains protocoles seront par exemple spécialisés dans l'échange de fichiers (FTP), d'autres pourront servir à gérer simplement l'état de la transmission et des erreurs (protocole ICMP).

Sur Internet par exemple les protocoles utilisés font partie d'une suite de protocoles, c'est-à-dire un ensemble de protocoles reliés entre eux. Cette suite de protocole s'appelle TCP/IP.

6.2.Classement de protocoles :

- **Les protocoles orientés connexions** : il s'agit des opérants de contrôle de transmission des données pendant une communication établie entre deux machines. Exemple : TCP, http, FTP, TELNET…etc.
- **Les protocoles non orientés connexions** : il s'agit d'un mode de communication dans lequel la machine émettrice envoie des données sans prévenir la machine réceptrice et cette dernière reçoit les données sans envoyer d'avis de réception à la première. Exemple : UDP, DNS, NFS, SNMP,…etc.

7.L'architecture client-serveur : [30]

7.1. Introduction :

Ces vingt dernières années ont vu une évolution majeure des systèmes d'information, à savoir le passage d'une architecture centralisée à travers de grosses machines (des Mainframes) vers une architecture distribuée basée sur l'utilisation de serveurs et de postes clients grâce à l'utilisation des PC et des réseaux. Cette évolution a été possible essentiellement grâce à 2 facteurs qui sont :

- la baisse des prix de l'informatique personnelle,
- le développement des réseaux.

7.2. Définition :

L'architecture client-serveur est un modèle de fonctionnement logiciel qui peut se réaliser sur tout type d'architecture matérielle (petites ou grosses machines). On parle de fonctionnement logiciel dans la mesure où cette architecture est basée sur l'utilisation de deux types de logiciels, à savoir un logiciel serveur et un logiciel client qui s'exécutant normalement sur 2 machines différentes. L'importance dans cette architecture, est l'utilisation de mécanismes de communication entre les deux applications.

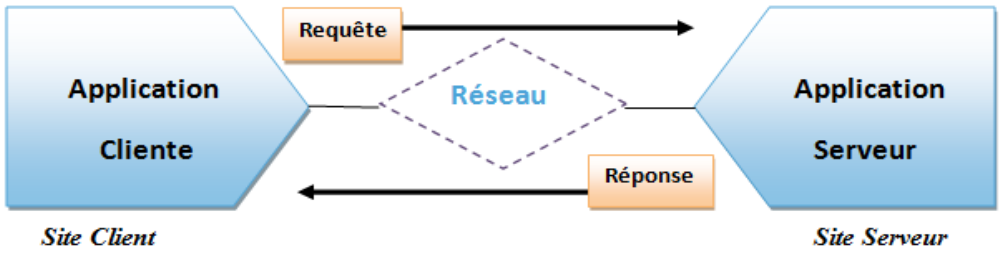


Figure 8 : Principes de fonctionnements du clients/serveur

Le dialogue entre les applications peut se résumer par :

- Le client demande un service au serveur,
- Le serveur réalise ce service et renvoie le résultat au client.

Un des principes fondamentaux de cette architecture est que le serveur réalise un traitement pour le client.

7.3. Définition du middleware :

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 différents composants d'une application répartie. Ce dialogue se base sur un protocole applicatif commun, défini par l'API du middleware.

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 et voici quelques services du middleware :

- **Conversion** : Service utilisé pour la communication entre machines,
- **Adressage** : Permet d'identifier la machine serveur sur laquelle est localisé le service demandé afin d'en déduire le chemin d'accès,
- **Sécurité** : Permet de garantir la confidentialité et la sécurité des données,
- **Communication** : Permet la transmission des messages entre les deux systèmes sans altération. Ce service doit gérer la connexion au serveur, la préparation de l'exécution des requêtes, la récupération des résultats et la déconnexion de l'utilisateur.

7.4. Avantages et Inconvénients de l'architecture Client-Serveur :[31]

○ **Avantages :**

Le modèle client/serveur est particulièrement recommandé pour les réseaux nécessitant un haut niveau de fiabilité, et voici quelques principaux avantages de ce modèle :

- **Des ressources centralisées :** étant donné que le serveur est au centre du réseau, il peut gérer des ressources communes à tous les utilisateurs, par exemple une base de données centralisée afin d'éviter les problèmes de redondance et de contradiction,

- **Une meilleure sécurité :** car le nombre de points d'entrée permettant l'accès aux données est moins important,
- **Une administration au niveau serveur :** l'administration se fait au niveau serveur,
- **Un réseau évolutif :** on peut supprimer ou rajouter des clients sans perturber le fonctionnement du réseau.

○ **Inconvénients :**

- **Un coût élevé :** dû à la technicité du serveur, un serveur doit être puissant et rapide, de pouvoir gérer le plus rapidement possible les requêtes d'un grand nombre de processus,
- **Un maillon faible :** le serveur représente un maillon faible dans ce modèle client/serveur, étant donné que tout le réseau est architecturé autour de lui.

PARTIE II :LA SECURITE INFORMATIQUE.

1. Introduction :

Les utilisateurs d'ordinateurs sont de plus en plus nombreux et ces ordinateurs sont généralement connectés à des réseaux, en particulier à l'internet. Si ces utilisateurs ne prennent pas un minimum de précautions, leurs ordinateurs peuvent être facilement attaqués.

Un système d'information est une organisation des activités consistant à acquérir, stocker, transformer, diffuser, exploiter, gérer les informations. Un des moyens techniques pour faire fonctionner un système d'information est d'utiliser un Système informatique être connues et mises en œuvre par tous les utilisateurs.

La sécurité informatique désigne un ensemble de techniques et de bonnes pratiques pour protéger les ordinateurs et les données qui y sont stockées. Donc Pour Assurer la sécurité de l'information implique d'assurer la sécurité des systèmes informatiques.

C'est l'objectif de cette partie qui propose d'acquérir les principes essentiels de la sécurité informatique.

2. Qu'est-ce que la sécurité d'un réseau ? [25]

La sécurité d'un réseau est un niveau de garantie que l'ensemble des machines du réseau fonctionnent de façon optimale et que les utilisateurs dites machines possèdent uniquement les droits qui leur ont été octroyé. Il peut s'agir :

- D'empêcher des personnes non autorisées d'agir sur le système de façon malveillante.
- D'empêcher les utilisateurs d'effectuer des opérations involontaires capables de nuire au système.
- De sécuriser les données en prévoyant les pannes.
- De garantir la non interruption d'un service.

3. Services principaux : [13],[14]

Une classification utile des services de sécurité est la suivante : confidentialité ; authenticité ; intégrité ; non-répudiation ; contrôle d'accès ; disponibilité. Et on va définir chacune de ces propriétés :[**13]**

- ✓ Confidentialité :C'est la propriété qui assure qu'une information ne peut être lue que par des entités habilitées (légales) et selon des contraintes précises.[**14]**

Exemple : Un mot de passe ne doit jamais pouvoir être lu par une autre personne que son possesseur.

- ✓ Authentification :C'est la propriété qui assure la reconnaissance sûre de l'identité d'une entité, L'authentification protège l'usurpation d'identité, on utilisant des Signatures.[**14]**
- ✓ Intégrité :C'est la propriété qui assure qu'une information n'est modifiée que par des entités habilitées (selon des contraintes précises) .[**14]**

Exemple : Une modification intempestive (même très temporaire) est à interdire sur une écriture comptable validée.

- ✓ Non-répudiation :La **non-répudiation** empêche tant l'expéditeur que le receveur de nier avoir transmis ou reçu un message. Ainsi, lorsqu'un message est envoyé, le receveur peut prouver que le message a bien été envoyé par l'expéditeur prétendu. De même, lorsqu'un message est reçu, l'expéditeur peut prouver que le message a bien été reçu par le receveur prétendu.[**13]**
- ✓ Contrôle d'accès :Dans le contexte de la sécurité des réseaux, le **contrôle d'accès** est la faculté de limiter et de contrôler l'accès à des systèmes et des applications via des maillons de communications. Pour accomplir ce contrôle, chaque entité essayant d'obtenir un accès doit d'abord être authentifiée, ou s'authentifier.[**13]**
- ✓ Disponibilité :De nombreuses attaques peuvent résulter en une perte ou une réduction de la **disponibilité** d'un service ou d'un système. Certaines de ces attaques sont susceptibles d'être l'objet de contre-mesures automatiques, telle que l'authentification et le chiffrement, alors que d'autres exigent une action humaine pour prévenir ou se rétablir de la perte de disponibilité des éléments d'un système.[**14]**

4. Objectifs des attaques : [15]

Tout ordinateur connecté à un réseau informatique est potentiellement vulnérable à une attaque.

Une « **attaque** » : est l'exploitation d'une faille d'un système informatique (système d'exploitation, logiciel ou bien même de l'utilisateur) à des fins non connues par l'exploitant des systèmes et généralement préjudiciables. Ces attaques sont pour la plupart lancées automatiquement à partir de machines infectées (par des virus, chevaux de Troie, vers, etc.), à l'insu de leur propriétaire. Plus rarement il s'agit de l'action de pirates informatiques.

Les motivations des attaques peuvent être de différentes sortes :

- Obtenir un accès au système ;
- Voler des informations, tels que des secrets industriels ou des propriétés intellectuelles ;
- Glaner des informations personnelles sur un utilisateur ;
- Récupérer des données bancaires ;
- S'informer sur l'organisation (entreprise de l'utilisateur, etc.) ;
- Troubler le bon fonctionnement d'un service ;
- Utiliser le système de l'utilisateur comme « rebond » pour une attaque ;

- Utiliser les ressources du système de l'utilisateur, notamment lorsque le réseau sur lequel il est situé possède une bande passante élevée
- Et le plus dangereux de tous, La vengeance.

5. Les pirates (Hackers): [18]

Le terme « **hacker** » est souvent utilisé pour désigner un pirate informatique. Les victimes de piratage sur des réseaux informatiques aiment à penser qu'ils ont été attaqués par des pirates chevronnés ayant soigneusement étudié leur système et ayant développé des outils spécifiquement pour en exploiter les failles.

5.1. Les différents types de pirates :

Il existe de nombreux types d'"attaquants" catégorisés selon leur expérience et selon leurs motivations :

- Les « **white hat hackers** » : hackers au sens noble du terme, dont le but est d'aider à l'amélioration des systèmes et technologies informatiques,
- Les « **black hat hackers** » : plus couramment appelés pirates, c'est-à-dire des personnes s'introduisant dans les systèmes informatiques dans un but nuisible ;

5.2. Objectifs poursuivis des hackers :

Les motivations des **black hat hackers** (pirates) peut être multiple :

- l'attrait de l'interdit ;
- l'intérêt politique ;
- l'intérêt éthique ;
- le désir de la renommée ;
- la vengeance ;
- l'envie de nuire (détruire des données, empêcher un système de fonctionner).

Par contre : Les objectifs des **white hat hackers** sont en règle générale un des suivants :

- l'apprentissage ;
- l'optimisation des systèmes informatiques ;
- la mise à l'épreuve des technologies jusqu'à leurs limites afin de tendre vers un idéal plus performant et plus sûr.

6. Les différents types d'attaques :[2], [16], [17]

L'informatique étant un domaine très vaste, le nombre de vulnérabilités présentes sur un système peut donc être important. Ainsi, les attaques visant ces failles peuvent être à la fois très variées et très dangereuses.

6.1. Anatomie d'une attaque :[16]

Fréquemment appelés « les 5 P » dans la littérature, ces cinq verbes anglophones constituent le squelette de toute attaque informatique : Probe, Penetrate, Persist, Propagate, Paralyze.

- **Probe :** consiste en la collecte d'informations par le biais d'outils. par exemple un scan de ports grâce au programme Nmap, ou encore un scan de vulnérabilités à l'aide du programme Nessus ;
- **Penetrate :** utilisation des informations récoltées pour pénétrer un réseau. Des techniques comme le brute force ou les attaques par dictionnaires peuvent être utilisées pour outrepasser les protections par mot de passe ;
- **Persist :** création d'un compte avec des droits de super utilisateur pour pouvoir se réinfiltrer ultérieurement ;
- **Propagate :** cette étape consiste à observer ce qui est accessible et disponible sur le réseau local ;
- **Paralyze :** cette étape peut consister en plusieurs actions. Le pirate peut utiliser le serveur pour mener une attaque sur une autre machine.

6.2. Les attaques réseau :

Ce type d'attaque se base principalement sur des failles liées aux protocoles ou à leur implémentation. Observons quelques attaques bien connues.

✓ **Les techniques de scan :**

Les scans de ports ne sont pas des attaques à proprement parler. Le but des scans est de déterminer quels sont les ports ouverts, et donc en déduire les services qui sont exécutés sur la machine cible (ex. : port 80/TCP pour un service HTTP).

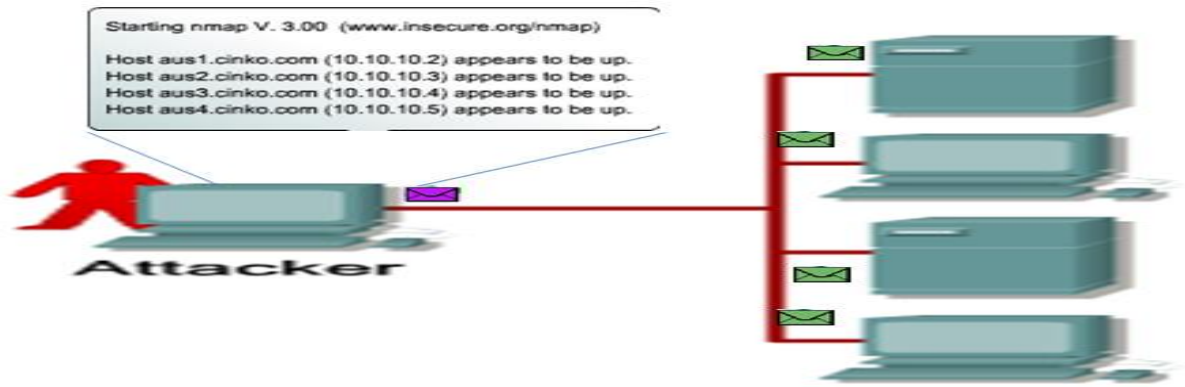


Figure9 :Scan avec Nmap

Et voici une description des techniques de scan les plus répandues :

- **le scan simple :** aussi appelé le scan connect (), il consiste à établir une connexion TCP complète sur une suite de ports.

- **Le scan furtif :** aussi appelé scan SYN, il s'agit d'une amélioration du scan simple. Ce scan essaie également de se connecter sur des ports donnés, mais il n'établit pas complètement la connexion (i.e. pas de commande ACK (acquittement))
- **Les scans XMAS, NULL et FIN :** se basent sur des détails de la RFC du protocole TCP pour déterminer si un port est fermé ou non en fonction de la réaction à certaines requêtes.
- **Le scan à l'aveugle :** s'effectue via une machine intermédiaire et avec du spoofing Le système attaqué pense que le scan est réalisé par la machine intermédiaire et non par le pirate ;
- **Le scan passif :** Consiste à analyser les champs d'en-tête des paquets (TTL, ToS, MSS…) et à les comparer avec une base de signatures qui pourra déterminer les applications qui ont envoyé ces paquets.

✓ **Spoofing :**

Le But est usurper l'adresse IP d'une autre machine pour se faire passer pour une autre machine en truquant les paquets IP. On utilisant des utilitaires qui permettent de modifier les paquets IP ou de créer nos propres paquets (ex. : hping2). Grâce à ces utilitaires, il est possible de spécifier une adresse IP différente de celle que l'on possède, et ainsi se faire passer pour une autre « machine ».

✓ **ARP Spoofing (ou ARP Redirect) :**

Le but est de rediriger le trafic d'une machine vers une autre pour que une personne mal intentionnée peut se faire passer pour une autre. De plus, le pirate peut rerouter les paquets qu'il reçoit vers le véritable destinataire, ainsi l'utilisateur usurpé ne se rendra compte de rien. La finalité est la même que l'IP spoofing, mais on travaille ici au niveau de la couche liaison de données.

✓ **DNS Spoofing :**

Fournir de fausses réponses aux requêtes DNS, c'est-à-dire indiquer une fausse adresse IP pour un nom de domaine afin de rediriger les internautes vers des sites pirates. Grâce à cette fausse redirection, l'utilisateur peut envoyer ses identifiants en toute confiance.

✓ **Fragments attacks :**

Le but est de passer outre les protections des équipements de filtrage IP. En passant outre les protections, un pirate peut par exemple s'infiltrer dans un réseau pour effectuer des attaques ou récupérer des informations confidentielles.

✓ **TCP Session Hijacking :**

Rediriger un flux TCP afin de pouvoir outrepasser une protection par mot de passe. Le contrôle d'authentification s'effectuant uniquement à l'ouverture de la session, un pirate réussissant cette attaque parvient à prendre possession de la connexion pendant toute la durée de la session.

6.3. Les attaques virales (logiciels malveillants) :[2]

Parmi les multiples procédés d'attaque contre le système d'information, les logiciels malveillants (*malware*) qui se répandent en général par le réseau, soit par accès direct à l'ordinateur attaqué, soit cachés dans un courriel ou sur un site Web attrayant, mais aussi éventuellement par l'intermédiaire d'une disquette, d'une clé USB ou d'un CD-Rom. La destination de ces logiciels est de s'installer sur l'ordinateur dont ils auront réussi à

violer les protections pour y commettre des méfaits. Et il existe plusieurs types de ces menaces :

✓ **Virus :**Un virus est un logiciel capable de s'installer sur un ordinateur à l'insu de son utilisateur légitime. Le terme virus est réservé aux logiciels qui se comportent ainsi avec un but malveillant, parce qu'il existe des usages légitimes de cette technique dite de **code mobile**: ex les appliquettes Java et les procédures JavaScript.

✓ **Virus réticulaire (botnet) :**La cible d'un virus informatique peut être indirecte : il y a des exemples de virus qui se propagent silencieusement sur des millions d'ordinateurs connectés à l'Internet, sans y commettre le moindre dégât. Puis, à un signal donné, ou à une heure fixée, ces millions de programmes vont se connecter à un même serveur Web, ce qui provoquera son effondrement. C'est ce qu'on appelle un déni de service distribué **(Distributed Denial of Service, DDoS).**Un tel virus s'appelle en argot SSI un **bot**, et l'ensemble de ces virus déployés un **botnet**.

✓ **Cheval de Troie :**Un cheval de Troie **(Trojan horse)** est un logiciel qui se présente sous un jour honnête, utile ou agréable, et qui une fois installé sur un ordinateur y effectue des actions cachées et pernicieuses.

✓ **Porte dérobée :**Une porte dérobée **(backdoor)** est un logiciel de communication caché, installé par exemple par un virus ou par un cheval de Troie, qui donne à un agresseur extérieur accès à l'ordinateur victime, par le réseau.

✓ **Bombe logique :**Une bombe logique est une fonction, cachée dans un programme en apparence honnête, utile ou agréable, qui se déclenchera à retardement, lorsque sera atteinte une certaine date, ou lorsque surviendra un certain événement.

✓ **Logiciel espion :**Un logiciel espion, comme son nom l'indique, collecte à l'insu de l'utilisateur légitime des informations au sein du système où il est installé, et les communique à un agent extérieur, par exemple au moyen d'une porte dérobée.

6.4. Les attaques applicatives :[16]

Les attaques applicatives se basent sur des failles dans les programmes utilisés, ou encore des erreurs de configuration. Toutefois, comme précédemment, il est possible de classifier ces attaques selon leur provenance.

✓ **Les problèmes de configuration :**

Il est très rare que les administrateurs réseau configurent correctement un programme. En général, ils se contentent d'utiliser les configurations par défaut. Celles-ci sont souvent non sécurisées afin de faciliter l'exploitation du logiciel.

✓ **Les bogues :**

Liés à un problème dans le code source, ils peuvent amener à l'exploitation de failles. Il n'est pas rare de voir l'exploitation d'une machine bloquée suite à une simple erreur de programmation. On ne peut toutefois rien faire contre ce type de problèmes, si ce n'est attendre un correctif de la part du développeur.

✓ **Les buffers overflows :**

Les buffers overflows, ou dépassement de la pile, sont une catégorie de bogue particulière. Issus d'une erreur de programmation, ils permettent l'exploitation d'un shellcode à distance. Ce shellcode permettra à une personne mal intentionnée d'exécuter des commandes sur le système distant, pouvant aller jusqu'à sa destruction.

✓ **Les scripts :**

Principalement web (ex. : Perl, PHP, ASP), ils s'exécutent sur un serveur et renvoient un résultat au client. Cependant, lorsqu'ils sont dynamiques (i.e. qu'ils utilisent des entrées saisies par un utilisateur), des failles peuvent apparaître si les entrées ne sont pas correctement contrôlées.

✓ **Man in the middle :[17]**

L'attaque « man in the middle » littéralement « attaque de l'homme au milieu » ou « attaques de l'intercepteur »), parfois notée MITM, est un scénario d'attaque dans lequel un pirate écoute une communication entre deux interlocuteurs et falsifie les échanges afin de se faire passer pour l'une des parties.

La plupart des attaques de type « man in the middle » consistent à écouter le réseau à l'aide d'un outil appelé sniffer exemple wireshark.

Figure10 :Attaque de l'homme au milieu

6.5. Le Déni de service :[16]

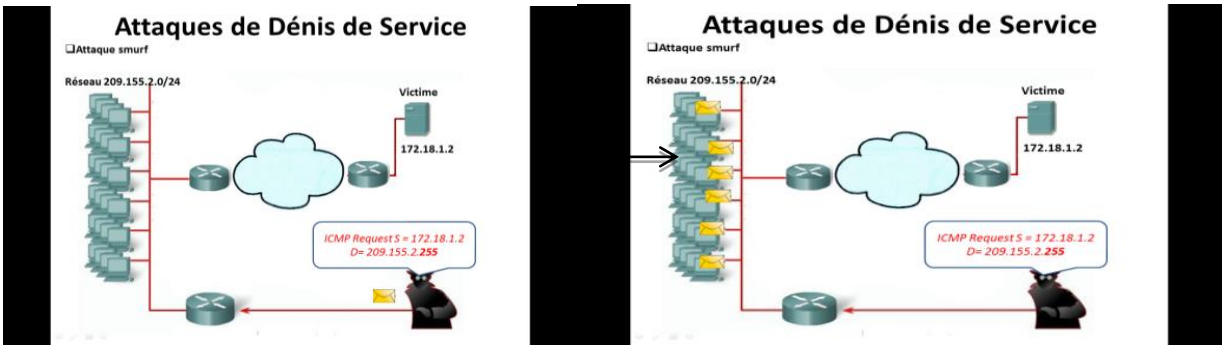
Le déni de service est une attaque visant à rendre indisponible un service. Ceci peut s'effectuer de plusieurs manières : par le biais d'une surcharge réseau, rendant ainsi la machine totalement injoignable ; ou bien de manière applicative en crashant l'application à distance.

Grâce à quelques instructions malicieuses et suite à une erreur de programmation, une personne mal intentionnée peut rendre indisponible un service (serveur web, serveur de messagerie…) voire un système complet.

Voici quelques attaques réseau connues :

- **SYN Flooding :** exploite la connexion en trois phases de TCP (Three Way Handshake : SYN/SYN-ACK/ACK). Le principe est de laisser un grand nombre de connexions TCP en attente. Le pirate envoie de nombreuses demandes de connexion (SYN), reçoit les SYN-ACK, mais ne répond jamais avec ACK. Les connexions en cours occupent des ressources mémoire, ce qui va entraîner une saturation et l'effondrement du système ;
- **UDP Flooding :** le trafic UDP est prioritaire sur TCP. Le but est donc d'envoyer un grand nombre de paquets UDP, ce qui va occuper toute la bande passante et ainsi rendre indisponibles toutes les connexions TCP.
- **Packet Fragment :** utilise une mauvaise gestion de la défragmentation au niveau ICMP. Par exemple : ping of death. La quantité de données est supérieure à la taille maximum d'un paquet IP.
- **Smurfling :** le pirate fait des requêtes ICMP ECHO à des adresses de broadcast en spoofant l'adresse source (en indiquant l'adresse de la machine cible). Cette machine cible va recevoir un nombre énorme de réponses, car toutes les machines vont lui répondre, et ainsi utiliser toute sa bande passante ;
- **Déni de service distribué :** le pirate va tenter de se rendre maître d'un nombre important de machines. Grâce à des failles (buffer overflows, failles RPC(4)…) il va pouvoir prendre le contrôle de machines à distance et ainsi pouvoir les commander à sa guise.

✓ **Exemple d'attaque déni de service :**



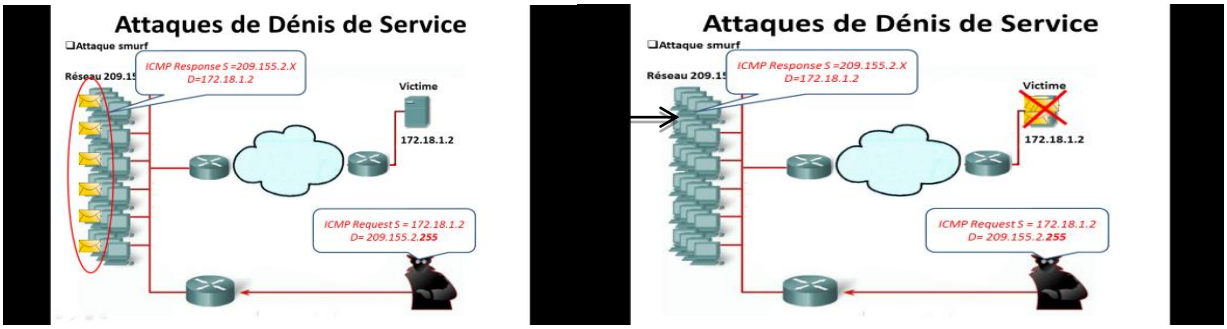


Figure11 :Attaque smurf

7. Mécanismes de sécurité :

La sécurité informatique se construit à l'aide de nombreux outils complémentaires et techniques existant sur le marché. Un seul ne suffit pas: la sécurité est assurée par une utilisation correcte d'un ensemble d'outils à choisir, paramétrer et/ou développer en fonction de l'objectif de sécurité fixé.**[20]**

7.1. Mots de passes :[19]

Lors de la connexion à un système informatique, celui-ci demande la plupart du temps un **identifiant** (en anglais **login**) et un **mot de passe (password**) pour y accéder. Ce couple **identifiant/mot de passe** forme ainsi la clé permettant d'obtenir un accès au système. Le choix du mot de passe est souvent laissé libre à l'utilisateur. Ainsi, la plupart des utilisateurs, estimant qu'ils n'ont rien de vraiment secret à protéger, se contentent d'utiliser un mot de passe facile à retenir (par exemple leur identifiant, le prénom de leur conjoint ou leur date de naissance).

Or, si les données sur le compte de l'utilisateur n'ont pas un **caractère stratégique**, l'accès au compte de l'utilisateur peut constituer une **porte ouverte** vers le système tout entier. En effet, dès qu'un pirate obtient un accès à un compte d'une machine, il lui est possible d'élargir son champ d'action en obtenant la liste des utilisateurs autorisés à se connecter à la machine

Les mots de passe des utilisateurs représentent donc la première défense contre les attaques envers un système.

7.2. Pare-feu (Firewall) :[2],[6]

La plupart des réseaux privés sont munis d'un pare-feu (firewall), ordinateur qui filtre les communications, un peu comme un routeur, d'ailleurs il est possible de configurer un routeur pour lui faire jouer le rôle d'un pare-feu simple. Un routeur doit décider au coup par coup du sort de chaque paquet, avec seulement une faible possibilité d'analyse historique, alors qu'un pare-feu efficace contre les attaques subtiles doit pouvoir faire des choses plus compliquées. La configuration d'un pare-feu consiste à rédiger des règles propres à déterminer les paquets autorisés et les paquets interdits ; chaque paquet est caractérisé par quelques paramètres :

- l'interface réseau sur laquelle le paquet est arrivé ; un pare-feu a au moins deux interfaces, l'une connectée au réseau privée et l'autre connectée au lien d'accès à l'Internet ;
- le fait que le paquet se présente sur l'interface depuis l'intérieur du pare-feu ou depuis le réseau ;
- le protocole auquel appartient le paquet, tel que mentionné dans son en-tête IP ;
- les adresses d'origine et de destination, mentionnées dans l'en-tête IP du paquet ;
- les numéros de port d'origine et de destination, mentionnés dans l'en-tête TCP ou UDP ;
- s'il s'agit d'un paquet TCP, les numéros de séquence et d'acquittement, qui permettent de reconstituer la séquence des paquets d'une connexion TCP.

Ces paramètres permettent d'identifier le type de communication auquel appartient le paquet, et éventuellement de reconstituer une séquence. Le simple filtrage par port se traduit par la rédaction de règles simples, qui peuvent prendre la forme de listes de contrôle d'accès (ACL) comme sur les routeurs Cisco**.[2]**

- **Les différentes catégories de firewall :[6]**

Il existe deux grandes catégories de firewall :

✓ **Sans état (stateless) :** ce sont ceux qui ne se souviennent pas du trafic déjà passé pour prendre les décisions de filtrage ; exemple ,au moment de laisser passer un paquet TCP avec les flags SYN ACK dans un sens, il ne pourront se baser sur le fait qu'un paquet avec le flag SYN est passé auparavant et cela donc correspondant à une réponse logique .

✓ **A état (stateful) :** Ils sont capables de mémoriser les trafics précédents et de se baser dessus pour laisser passer les paquets qui se présentent.

Les firewalls à état sont plus couteaux en ressources, mémoire et processeur, mais ce n'est plus un problème sur les architectures actuelles, sauf pour les débits plus importants.

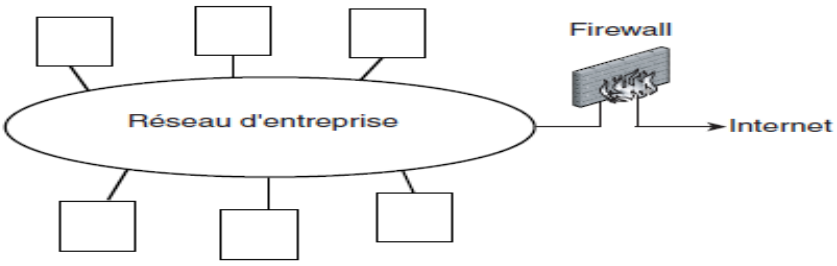


Figure12 : Protection avec un pare-feu

7.3. VPN(Réseau Privé Virtuel): [2][21]

Les techniques cryptographiques pour le chiffrement de messages individuels (i.e. qui seront détaillé dans le **chapitre II**), ne le sont en aucun cas le seul usage de cette technique. On peut imaginer, et de plus en plus c'est ce qui sera réalisé, le chiffrement systématique de toutes les

communications en réseau. Si l'on procède ainsi, chiffrer message par message serait très inefficace : on choisira plutôt de chiffrer le flux de l'ensemble du trafic sur un ou plusieurs itinéraires donnés, cela constituera un réseau privé virtuel, ou **VPN**, comme **Virtual Private Network**. Il s'agira par exemple d'établir un canal chiffré entre deux nœuds quelconques de l'Internet, ces nœuds pouvant eux-mêmes être des routeurs d'entrée de réseaux. On aura ainsi établi une sorte de tunnel qui, à travers l'Internet, reliera deux parties éloignées l'une de l'autre du réseau d'une même entreprise pour donner l'illusion de leur contiguïté. Mais le chiffrement permet aussi d'établir un VPN personnel pour un utilisateur, par exemple entre son ordinateur portable et le réseau local de l'entreprise. [**2]**

7.3.1. Tunnel :[21]

La création d'un VPN implique la création d'un tunnel. Cela signifie que les données échangées entre les deux entités sont encapsulées et/ou cryptées. Ainsi ces données transitent le plus souvent sur Internet de manière sécurisée en garantissant soit la confidentialité des données (grâce au cryptage), soit à l'intégrité des données (c'est-à-dire que les données n'ont pas étés modifiées entre l'émission et la réception), soit les deux en même temps.

7.3.2. Types de VPN :[21]

On peut dénombrer deux grands types de VPN, chacun d'eux caractérise une utilisation bien particulière de cette technologie.

- **LAN-to-LAN :**

Tout d'abord le LAN-to-LAN qui permet de relier deux réseaux d'entreprises entre eux de façon transparente. Généralement les deux sites ont des tranches IP différentes ce qui oblige les postes clients à passer par le routeur. Celui-ci est directement relié à l'équipement responsable du VPN ou implante directement les protocoles choisit pour la mise en place du VPN.

Ce type de VPN est installé de manière permanente.

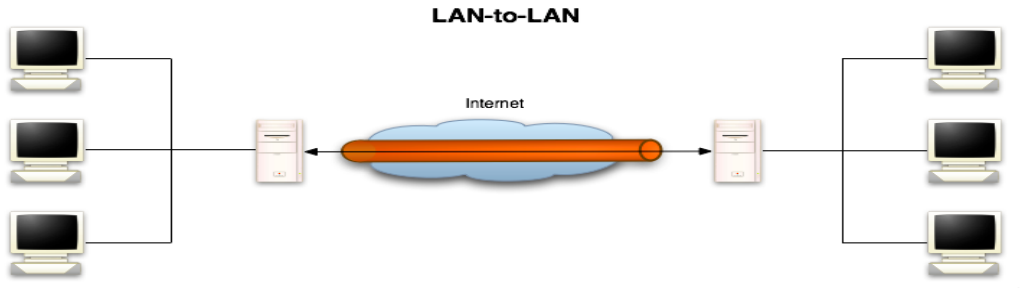


Figure13:VPN LAN-to-LAN

- **Nomade ou Road Warrior :**

Ensuite, il existe le type nomade, également appelé "Road Warrior" qui permet à un utilisateur distant de son entreprise de se connecter à celle-ci pour pouvoir profiter de ses services. Ainsi, il pourra lire ses mails, récupérer des fichiers présents sur le réseau de son entreprise, ...
Etant donné son utilisation, ce type de VPN est installé de manière occasionnelle.

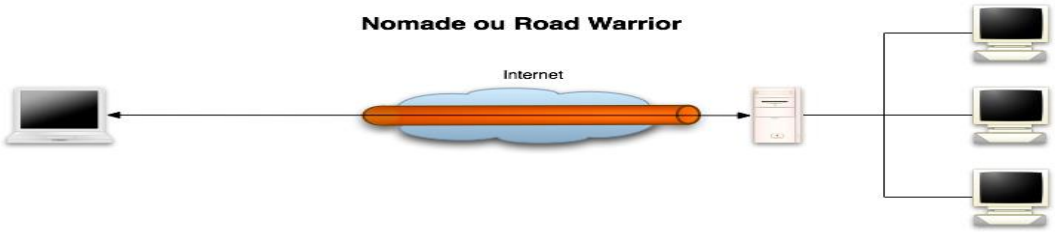


Figure14: VPN Nomade ou Road Warrior

7.4. Les fichiers historiques :

Des outils de traçabilité (logging) doivent être mis en œuvre pour garder une trace des événements, comme par exemple: qui est venu, quand, quelle a été la durée de la transaction? Qu'a-t-on consulté ou modifié? Quelle sont été les ressources utilisées?

La consultation régulière des fichiers historiques constitués doit notamment permettre de vérifier les anomalies dans le trafic des transactions (par exemple les messages répétitifs en provenance d'une même adresse extérieure et rejetés par le firewall peuvent être un signe d'essai d'intrusion).

7.5. Les copies de sauvegarde :

Les copies de sauvegarde (back-up) créées régulièrement et stockées dans des endroits sécurisés permettent de protéger les informations essentielles pour l'entreprise et permettent également de redémarrer rapidement en cas de problème.

7.6. Système de détection d'intrusion :[16]

Comme nous l'avons vu, les attaques utilisées par les pirates sont très variées. Certaines utilisent des failles réseau et d'autres des failles de programmation. Nous pouvons donc facilement comprendre que la détection d'intrusions doit se faire à plusieurs niveaux.

Ainsi, il existe différents types d'Système de détection d'intrusion, Citons quelques types :

- **Les systèmes de détection d'intrusions (IDS) :**

IDS est un ensemble de composants logiciels et matériels dont la fonction principale est de détecter et analyser toute tentative d'effraction (volontaire ou non) et fonction est sa détection des techniques de sondage (balayages de ports), des tentatives de compromission de systèmes, d'activités suspectes internes, des activités virales ou encore audit des fichiers de journaux (logs).

Certains termes sont souvent employés quand on parle d'IDS :

Faux positif : une alerte provenant d'un IDS, mais qui ne correspond pas à une attaque réelle ;

Faux négatif : une intrusion réelle qui n'a pas été détectée par l'IDS.

- **Les systèmes de détection d'intrusions « réseau » (NIDS) :**

NIDS est utilisé pour analyser de manière passive les flux en transit sur le réseau et détecter les intrusions en temps réel. Un NIDS écoute donc tout le trafic réseau, puis l'analyse et génère des alertes si des paquets semblent dangereux. Les NIDS étant les IDS plus intéressants et les plus utiles du fait de l'omniprésence des réseaux dans notre vie quotidienne.

Figure15: NIDS

7.7. Encryption, signature électronique et certificats :[20]

L'utilisation des techniques d'encryptions, de signature électronique et des certificats sont la base d'un commerce électronique sécurisé:

- **l'encryption:** elle consiste à transformer les informations électroniques au moyen d'un algorithme mathématique afin de les rendre inintelligibles, sauf pour celui qui possède le moyen (une clé) de les décoder. L'encryption des informations qui transitent par le réseau est utilisée pour assurer la confidentialité, l'intégrité et l'authenticité des transactions et du courrier électronique (ce qui sera détaillé dans la prochaine partie).
- **la signature électronique:** c'est un code digital (une réduction du document électronique à envoyer) qui, associé aux techniques d'encryption, garantit l'identité de la personne qui émet le message et assure la non-répudiation et l'intégrité de l'envoi;
- **le certificat:** document électronique (carte d'identité) émis par une autorité de certification. Il valide l'identité des interlocuteurs d'une transaction électronique, associe une identité à une clé publique d'encryption et fournit des informations de gestion complémentaires sur le certificat et le détenteur.

Conclusion :

Au cours de ce premier Chapitre, nous avons présenté le moyen qui permet l'échange de l'information qui est le réseau informatique, et on a définit l'architecture Client/serveur utilisé par le réseau internet. Ces différents concepts traités dans cette partie, nous aiderons à mieux comprendre les notions fondamentales pour mener à bien notre projet et en deuxième partie on a pu voir un peu

comment sécurisé cette information en utilisant les moyens appropriés pour cette tâche et en regardant les différentes menaces De l'information échangé alors en deuxième chapitre on va détailler l'un de ses moyens de sécurisé une information qui est la cryptographie .

CHAPITRE II.

La cryptographie

1. Histoire :

L'histoire est une suite d'événements mis bout à bout. Sans les acteurs qui créent cette dynamique,il ne serait possible de voir une progression. Les personnes suivantes ont toutes eu une incidence dans la cryptographie que l'on connaît aujourd'hui. Voici donc un rapide descriptif de quelques faits marquants et des hommes qui ont permis cette évolution.

- Vers 1900 av. J.-C., un scribe égyptien a employé des hiéroglyphes non conformes à la langue correctedans une inscription. Kahn le qualifie de premier exemple documenté de cryptographie écrite.
- Quatre siècles plus tard, vers 1500 av. J.-C., une tablette mésopotamienne contient une formulechiffrée pour la fabrication de vernis pour les poteries.
- Cinq siècles avant notre ère, des scribes hébreux mettant par écrit le livre de Jérémie ont employéun simple chiffre de substitution connu sous le nom d'Atbash. C'était un des quelques chiffres hébreux de cette époque, avec Albam et Atbah.
- En 487 av. J.-C., les grecs emploient un dispositif appelé la scytale, un bâton autour duquel unebande longue et mince de cuir était enveloppée et sur laquelle on écrivait le message.

La cryptographieévolue petite a petite jusqu'à en arriverà notre époqueoù on peut parler des différents algorithmes de cryptage qui utilises de différents concepts par exemple les algorithme de cryptage a clé public comme (**RSA** et **ElGamal ..)** ainsi des algorithme a clé privé comme (**RC6**,**DES**, et **AES** ….) et beaucoup d'algorithme sont mis en œuvre afin de protéger l'information transmise, età cause des menaces qui les mis en danger et qui évolue en parallèle aussi.

2. Introduction :

La cryptographie concerne l'humanité depuis longtemps, en effet, la cryptographie est une science très ancienne, de ce temps-là et au long de l'histoire, la cryptographie a été utilisée exclusivement à des fins militaires. Aujourd'hui, les réseaux informatiques exigent une phase de cryptographie comme mécanisme fondamental afin d'assurer la confidentialité de l'information numérique.

Dans ce chapitre nous présenterons les notions de base de la cryptographie ainsi quelques algorithmes de cryptages connues.

3. Les bases de la cryptographie : [1][2]

La cryptographie utilise des concepts issus de nombreux domaines (Informatique, Mathématiques, Electronique). Toutefois, les techniques évoluent et trouvent aujourd'hui régulièrement racine dans d'autres branches (Biologie, Physique, etc.).

3.1. Naissance de la cryptographie :[2]

L'invention de l'ordinateur a bien sûr donné un essor considérable à la cryptographieet à la cryptanalyse. Ce n'est d'ailleurs pas un hasard si le créateur dumodèle théorique de l'ordinateur, Alan Turing, a été aussi pendant la guerre unformidable concepteur de machines à déchiffrer les codes allemands chiffrés parles automates Enigma. Les machines de Turing, appelées Bombes, étaient fondéessur une réalisation originale du logicien polonais Marian Rejewski. La courbe quitrace le succès des attaques de sous-marins allemands contre les convois transatlantiquesqui acheminaient les fournitures américaines à la Grande-Bretagne subitdes fluctuations importantes qui correspondent au délai à l'issue duquel l'équipe d'Alan Turing à Bletchley Park en Angleterre parvenait à déchiffrer plus ou moinsparfaitement le code allemand après un changement de combinaison des Enigma. Lorsque l'on sait l'importance militaire qu'ont eue ces fournitures, on ne sauraitsous-estimer la contribution de Turing à la victoire alliée.

3.2. Vocabulaire de base :[1]

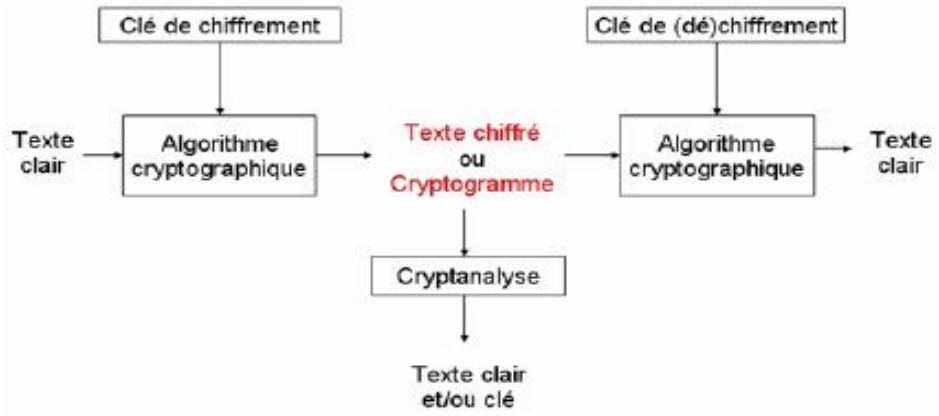


Figure16: Protocole de chiffrement

- **Cryptologie :** Il s'agit d'une science mathématique comportant deux branches : la cryptographie et lacryptanalyse
- **Cryptographie :** La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des donnéesde manière confidentielle sur un support donné.
- **Chiffrement :** Le chiffrement consiste à transformer une donnée (texte, message, …) afin de la rendreincompréhensible 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.

- **Texte chiffré :** Appelé également cryptogramme, le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair.
- **Clef :** Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations.
- **Cryptanalyse :** Opposée à la cryptographie, elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés.

➢ **Cryptosystème :** Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné.

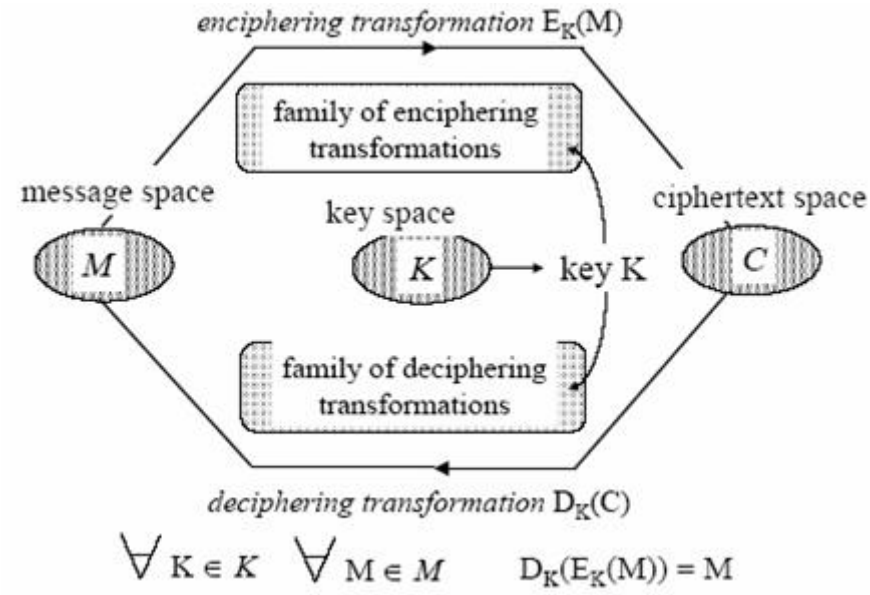


Figure17: Cryptosysteme

L'algorithme est en réalité un triplé d'algorithmes :

- l'un générant les clés K,
- un autre pour chiffrer M, et
- un troisième pour déchiffrer C.

Remarque : On parle de « décryptage » pour désigner l'action permettant de retrouver le texte clair sans connaître la clef de déchiffrement. On emploie également parfois les termes « cryptage » et « crypter » pour qualifier l'action de chiffrer un message. Les mots « encryptage » et « (en) cryptement » sont des anglicismes dérivés du verbe « to encrypt ».

4. Les principes de chiffrement :

4.1. Chiffrement à clé symétrique (dits à clé secrète) : [7] [1]

Si la clé est unique, elle sert à chiffrer et à déchiffrer le message ; on parle alors de chiffrement symétrique. Les algorithmes symétriques actuels utilisent une succession de transpositions et de substitutions complexes des valeurs du message, basées sur des opérations mathématiques et réalisées en plusieurs passes. La clé faisant partie intégrante de la fonction, il est impossible d'inverser l'algorithme sans elle, et les seules attaques envisageables consistent souvent à essayer toutes les valeurs de clés possibles. C'est la raison pour laquelle une clé symétrique suffisamment importante (128 bits) et bien choisie est considérée comme sûre. L'algorithme symétrique le plus célèbre est le **DES** (Data Encryption Standard, qui fonctionnait avec des clés de 64 bits) remplacé depuis par l'**AES** (Advanced Encryption System, qui fonctionne avec des clés allant jusqu'à 256 bits). Les chiffrements symétriques exigent toutefois que les deux correspondants échangent au préalable la clé secrète par un canal sûr, ce qui est quasiment impossible à grande échelle. **[7]**

4.1.1. Caractéristiques : [1]

- Les clés sont identiques : KE = KD = K,
- La clé doit rester secrète,
- Les algorithmes les plus répandus sont le DES, AES, 3DES, ...
- Au niveau de la génération des clés, elle est choisie aléatoirement dans l'espace des clés,
- Ces algorithmes sont basés sur des opérations de transposition et de substitution des bits du texteclair en fonction de la clé,
- La taille des clés est souvent de l'ordre de 128 bits. Le DES en utilise 56, mais l'AES peut allerjusque 256,
- L'avantage principal de ce mode de chiffrement est sa rapidité,
- Le principal désavantage réside dans la distribution des clés : pour une meilleure sécurité, onpréfèrera l'échange manuel. Malheureusement, pour de grands systèmes, le nombre de clés peutdevenir conséquent. C'est pourquoi on utilisera souvent des échanges sécurisés pour transmettre lesclés. En effet, pour un système à N utilisateurs, il y aura $N.(N-1)/2$ paires de clés.

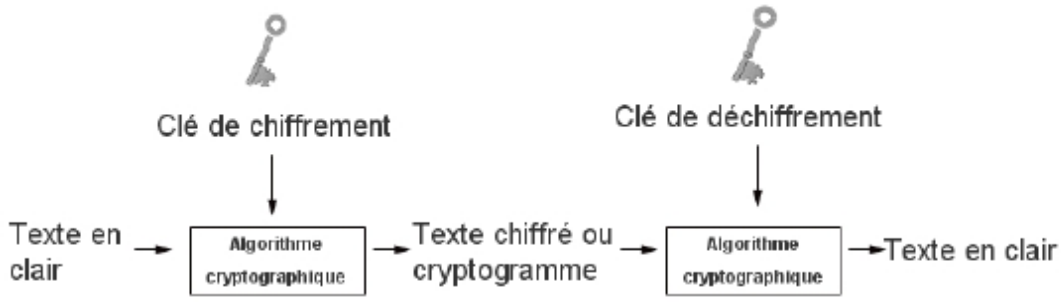


Figure18: Chiffrement symétrique

4.1.2. Systèmes de chiffrements par blocs (Block Cipher) : [6] [9]

4.1.2.1. Les différents modes de chiffrement : [9]

Le mode de chiffrement correspond à la manière dont on va utiliser un algorithme de chiffrement donné. Ce mode de chiffrement consiste par exemple à rajouter de la contre-réaction entre l'entrée et la sortie de l'algorithme afin de lui rajouter des caractéristiques bien précises. Les différentes modes de chiffrement utilisés sont les suivants :

- Le mode ECB pour **Electronic Code Book**
- Le mode CBC pour **Cipher Block Chaining**
- Le mode **chiffrement en continu**
- Le mode CTAK pour **Cipher Text Auto Key**
- Le mode CFB pour **Cipher Feed Back**
- Le mode KAK pour **Key Auto Key**
- Le mode OFB pour **Output Feed Back**
- Le mode CTR pour **CounTeR**
- Le mode BC pour **Block Chaining**
- Le mode PCBC pour **Propagating Cipher Block Chaining**
- Le mode CBCC pour **Cipher Block Chaining with Checksum**
- Le mode OFBNLF pour **Output Feed Back mode with a Non Linear Function**
- Le mode PBC pour **Plaintext Block Chaining**
- Le mode PFB pour **PlaintextFeed Back**
- Le mode CBCPD pour **Cipher Block Chaining of Plaintext Difference**
- Le mode CTS pour **Cipher Text Stealing**

Dans la suite nous allons regarder un peu plus en détail les modes ECB, CBC, CFB et OFB. Les autres modes sont cités mais sont rarement utilisés :

- **Le mode ECB** :

Ce mode est le plus simple : un même bloc est toujours codé de la même manière. Il n'y a pas de rétroaction de l'entrée ou de la sortie sur la fonction de chiffrement.

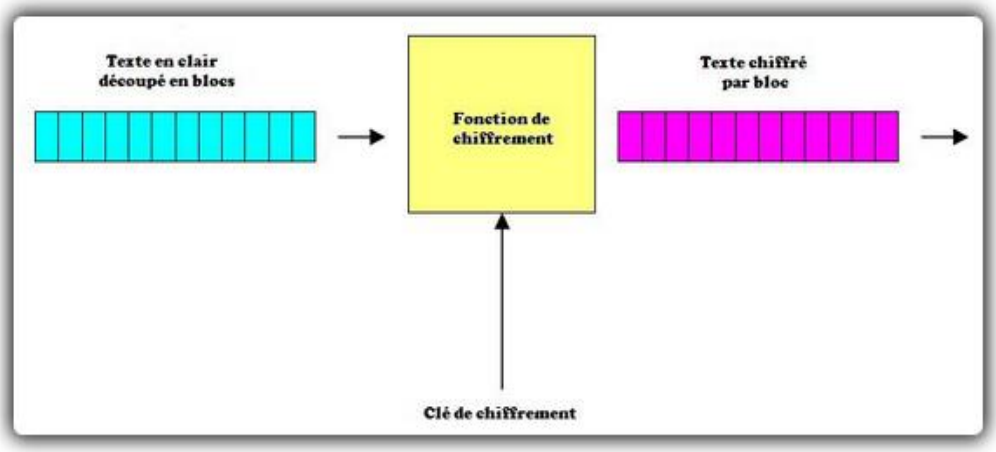


Figure19: Le mode ECB

- **Le mode CBC :**

Dans ce mode de chiffrement, chaque bloc de texte en clair est d'abord combiné par un ou exclusif avec le dernier bloc du texte chiffré. La sortie de ce ou exclusif est ensuite appliquée à la fonction de chiffrement. Ce mode de chiffrement dispose en plus d'un vecteur d'initialisation appelée IV (pour Initialisation Vector) qui permet d'initialiser le processus quand aucun bloc n'a encore été chiffré.

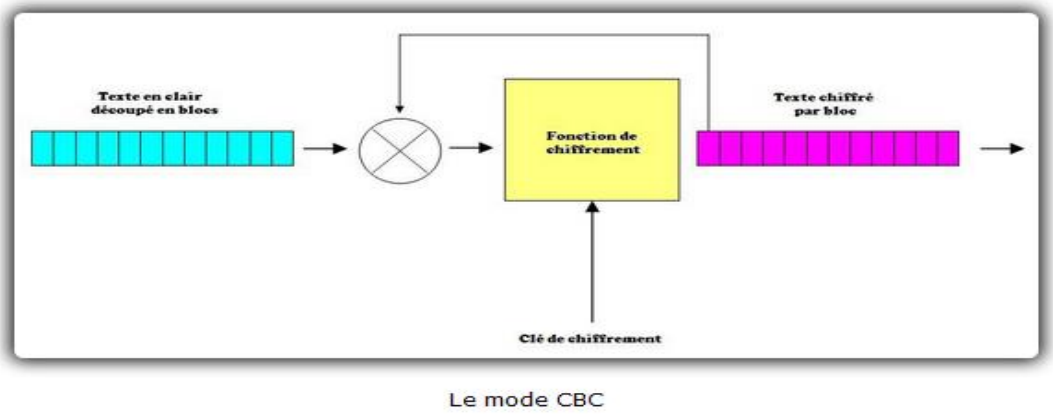


Figure20: Le mode CBC

- **Le mode CFB :**

Les modes ECB et CBC travaillent avec sur des blocs de texte en clair (64 bits par exemple). Ces modes ne sont pas utilisables lorsque le chiffrement ne peut débuter que lorsqu'un bloc est complet. Sur des applications réseau, cela peut poser des problèmes car les valeurs à chiffrer arrivent de manière asynchrone sous forme d'octets et doivent être transmises immédiatement (cas du protocole Telnet par exemple).

Le registre à décalage est initialisé avec un vecteur d'initialisation. Le bloc complet est alors chiffré. L'octet de poids faible du texte chiffré est combiné par un ou exclusif avec l'octet de texte en clair. Le résultat de cette opération est alors transmis en même temps qu'il est injecté dans le registre à décalage.

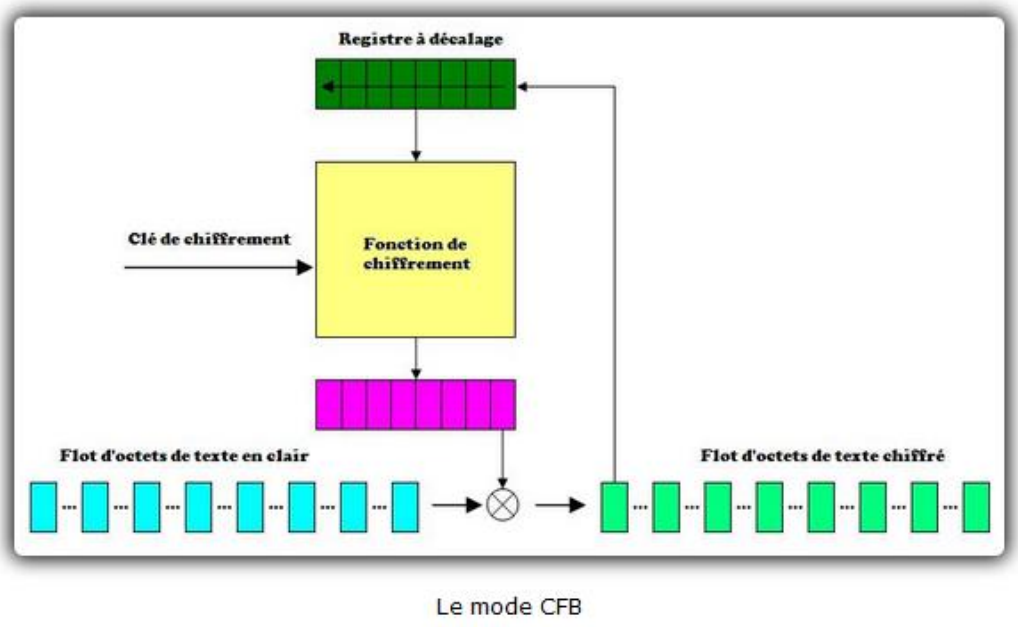


Figure21: Le mode CFB

- **Le mode OFB :**

Le mode OFB ressemble au mode CFB. La seule différence est que l'octet injecté dans le registre à décalage est l'octet de poids faible du texte chiffré.

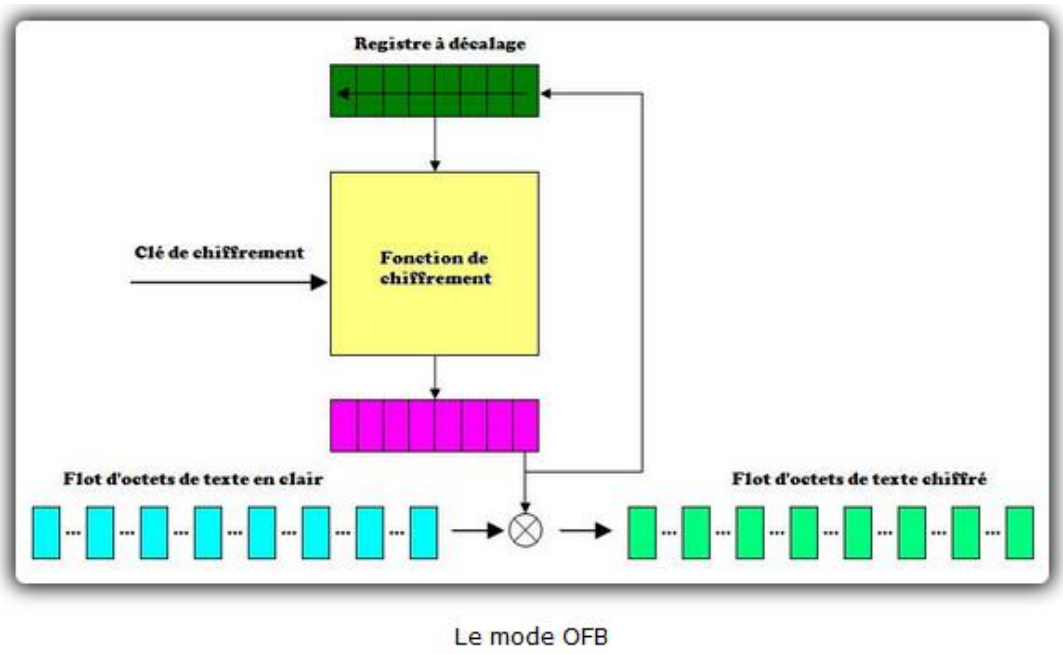


Figure22: Le mode OFB

4.1.3. Chiffrement par permutation :[6]

Le chiffrement par permutation se contente de change l'ordre des lettres du messagedésigne de nouveau l'ensemble des permutations π .On prend :

$\{1,\ldots, m\} \rightarrow \{1,\ldots, m\}$

$E_k(x_1,\ldots,x_m) \rightarrow (x_{\pi(1)},\ldots,x_{\pi(m)})$

$d_k(y_1,\ldots,y_m) \rightarrow (y_{T(1)},\ldots,y_{T(m)})$

Avec T la permutation inverse de π.

Présentation :

On définit en générale π par la suite des valeurs π(1),…, π(m) donnée par une table lue ligne par ligne.

4.1.4. Chiffrement par substitution :[6]

On suppose le message constitué d'éléments pris dans un ensemble A (l'alphabet).

Soit π une substitution de A, c.-à-d. qu'à un élément x de A, on associe un autre élément y de façon que deux x distincts donnent deux y distincts ; forcement, tout éléments de A peut être vu comme associé à un x (unique).Une telle substitution fournit un système de chiffrement.

4.1.5. Chiffrement monoalphabétiques :[6]

On parle de chiffrements monoalphabétiques lorsque chaque lettre du message clair est toujours remplacée par le même symbole. On obtient ainsi une bijection entre les lettres claires et les symboles de l'alphabet de chiffrement.

4.2. Chiffrement à clé asymétrique (dits à clé public) :[7]

L'algorithme asymétrique dissocie les fonctions de chiffrement et de déchiffrement en deux clés. Ce que l'une chiffre, seule l'autre peut le déchiffrer. Aucune autre clé même celle qui a réalisé le chiffrement, ne peut y parvenir. Ainsi, chacun peut diffuser librement l'une de ses deux clés (dite publique) afin que n'importe qui puisse chiffrer un message à son attention. Seule la clé gardée secrète (dite privée) permet d'en prendre connaissance. C'est là le principal avantage sur le chiffrement symétrique : les clés privées ne circulent pas et les clés publiques peuvent être publiées dans un annuaire (exemple : keyserver.com) ou demandées directement au propriétaire.

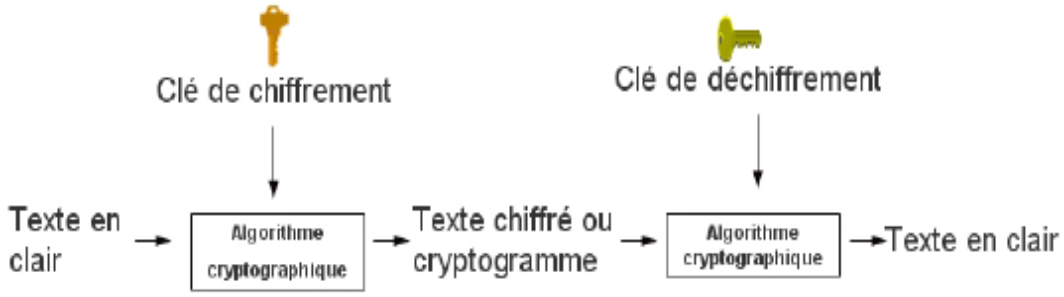


Figure23: Chiffrement asymétrique

L'un des principaux avantages de la cryptographie de clé publique est qu'elle offre une méthode d'utilisation des signatures numériques.**[7]**

4.2.1. La signature :

La signature numérique est un mécanisme qui permet d'authentifier un message, autrement dit de prouver qu'un message provient bien d'un expéditeur donné. La signature d'un document utilise à la fois la cryptographie asymétrique et les fonctions de hachage. C'est en effet par l'association de ces deux techniques que nous pouvons obtenir les cinq caractéristiques d'une signature (authentique, infalsifiable, non réutilisable, inaltérable, irrévocable).

4.2.2. Caractéristiques :[1]

- Une clé publique PK (symbolisée par la clé verticale),
- Une clé privée secrète SK (symbolisée par la clé horizontale),
- Propriété : La connaissance de PK ne permet pas de déduire SK,
- L'algorithme de cryptographie asymétrique le plus connu est le RSA,
- Le principe de ce genre d'algorithme est qu'il s'agit d'une fonction unidirectionnelle à trappe. Unetelle fonction à la particularité d'être facile à calculer dans un sens, mais difficile voire impossibledans le sens inverse. La seule manière de pouvoir réaliser le calcul inverse est de connaître unetrappe. Une trappe pourrait par exemple être une faille dans le générateur de clés...
- Les algorithmes se basent sur des concepts mathématiques tels que l'exponentiation de grandsnombres premiers (RSA),
- La taille des clés s'étend de 512 bits à 2048 bits en standard. Dans le cas du RSA, une clé de512 bits n'est plus sûre au sens "militaire" du terme, mais est toujours utilisable de particulier àparticulier.
- Au niveau des performances, le chiffrement par voie asymétrique est environ 1000 fois plus lent quele chiffrement symétrique.
- Cependant, à l'inverse du chiffrement symétrique où le nombre de clés est le problème majeur, ici,seules n paires sont nécessaires. En effet, chaque utilisateur possède une paire (SK, PK) et tous lestransferts de message ont lieu avec ces clés.
- La distribution des clés est grandement facilitée car l'échange de clés secrètes n'est plus nécessaire.Chaque utilisateur conserve sa clé secrète sans jamais la divulguer. Seule la clé publique devra êtredistribuée.

4.3. Fonction de hachage :[5]

Lors d'échanges de messages cryptés, il est important de pouvoir s'assurer que le message n'a pas été altéré ou modifié par un tiers pendant l'envoi. Les fonctions de hachage permettent alors de s'assurer de l'intégrité du message.

4.3.1. Principe :

Une fonction de hachage calcule l'empreinte y (ou digest) d'un message x. Cette fonction F doit être une fonction à sens unique c'est-à-dire qu'il doit être facile de trouver y à partir de x, mais très difficile de trouver x à partir de y. Elle doit aussi être très sensible pour qu'une petite modification du message entraîne une grande modification de l'empreinte. En envoyant

le message accompagné de son empreinte, le destinataire peut ainsi s'assurer de l'intégrité du message en recalculant le résumé à l'arrivée et en le comparant à celui reçu. Si les deux résumés sont différents, cela signifie que le fichier n'est plus le même que l'original : il a été altéré ou modifié par une tierce personne. Les fonctions de hachage les plus répandus sont **MD5** et **SHA-1** qui sont basés tous les deux sur MD4, MD5 générant des empreintes de 128 bits et SHA-1 de 160 bits (seul MD5 sera décrit, ces deux fonctions ayant un fonctionnement similaire).

4.3.2. MD5 :

MD5 (« Message Digest ») est un des plus connus algorithme de hachage. C'est une version améliorée de MD4 tous deux conçus par Ron Rivest, un des créateurs de RSA. MD5 fabrique une empreinte d'une taille de 128 bits et voici les étapes utilisé pour l'avoir :

- **Padding**

Soit un message m d'une longueur de n bits. MD5 manipulant des blocs de 512 bits, l'algorithme complète le message avec un 1 suivi d'autant de 0 que nécessaires jusqu'à ce que la longueur de message soit congrue à 448 modulo 512. L'opération de padding a toujours lieu même si la longueur du message est déjà congrue à 448 modulo 512.

- **Ajout de la taille**

On ajoute à ce message la valeur de n, codée en binaire sur 64 bits. On obtient donc un message dont la longueur est un multiple de 512 bits. Chaque bloc de 512 bits est décomposé en 16 blocs de 32 bits.

- **Initialisation**

MD5 prend 4 tampons de 32 bits en entrée initialisés de la manière suivante (en hexadécimal)

A=01234567

B=89abcdef

C=fedcba98

D=76543210

- **Rondes**

MD5 est composé de quatre rondes qui exécutent chacune 16 opérations. Pour chaque ronde, une seule fonction prenant 3 arguments codés sur 32 bits et renvoyant une valeur sur 32 bits est utilisée pour les 16 opérations. Les 4 fonctions sont les suivantes :

F(X,Y,Z) = (X and Y) or (not(X) and Z)

G(X,Y,Z) = (X and Z) or (Y and not(Z))

H(X,Y,Z) = X xor Y xor Z

I(X,Y,Z) = Y xor (X or not(Z))

Pour chaque bloc de 512 bits, on effectue les opérations suivantes :

- on sauvegarde les valeurs des tampons (A, B, C et D) dans des registres AA, BB, CC et DD.
- On calcule les nouvelles valeurs pour A, B, C et D à partir de leurs anciennes valeurs, des bits du bloc qu'on étudie et une des quatre fonctions F, G, H ou I selon la ronde.
- On effectue A=AA+A, B=BB+B, C=CC+C, D=DD+D

- **Ecriture de l'empreinte**

L'empreinte sur 128 bits est obtenue en mettant bout à bout les quatre tampons finaux A, B, C et D.

5. La cryptographie classique : [4] [3]

Dans la cryptographie classique, la méthode et la clé de chiffrement ainsi que celles de déchiffrement sont connues par l'émetteur et le destinataire. L'émetteur et le destinataire doivent se mettre préalablement d'accord sur la clé. Ce qui pose le problème de l'échange des clés, par exemple, dans un réseau de N entités susceptibles de communiquer secrètement, il faut distribuer C2 = N(N-1)/2 N clés. La plupart des méthodes de chiffrement classiques reposent sur deux principes essentiels: la substitution et la transposition. La substitution consiste àremplacer certaines lettres par d'autres ou par des symboles. La transposition signifie qu'on permute les lettres du message afin de le rendre inintelligible. Au cours des siècles, de nombreux systèmes cryptographiques ont été mis au point, de plus en plus perfectionnés et de plus en plus astucieux. **[4]**

5.1. Exemple de quelques codes classiques :

5.1.1. Code de César : [3]

Le code de **Cesar** est la méthode de cryptographie la plus ancienne communément admise par l'histoire. Il consiste à substituer chaque lettre de l'alphabet par celle obtenue après un décalage des lettres. Par exemple, si on remplace A par D, on remplace alors B par E, C par F, D par G, etc.

Texte claire	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
Texte Codé	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

Table2: Chiffrement de César

Il n'y a que 26 façons différentes de chiffrer un message avec le code de César. Donc, il est très facile de le casser, en testant de façon exhaustive toutes les possibilités.

5.1.2.Le carré de Polybe :[3]

Polybe (150 av .JC) était un écrivain grec. C'est lui qui a inventé le premier chiffrement de substitution. Le carré de Polybe est basé sur une grille comme suit :

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	IJ	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

Table3: Chiffrement de carré de Polybe

Chaque lettre est codée par 2 chiffres. Exemple : S= 43.Les lettres I et J ne sont pas différenciés. « BONJOUR » sera chiffré par « 12343324344542 »

5.1.3. Chiffrement de Delastelle :[4]

Le chiffre de Delastelle est un mélange de codage par substitution et par transposition. On commence par découper le message en des blocs de 5 lettres, puis on utilise le carré de Polybe. On écrit verticalement pour chaque lettre la position dans le tableau. Par exemple, si on souhaite coder "OMARYFOUZ", on procède comme suit :

Le groupe des 5 premières lettres

O	M	A	R	Y
3	3	1	4	5
4	2	1	2	4

F	O	U	Z	
2	3	4	5	0
1	4	5	5	0

Table 3.3 : Un chiffrement de Delastelle

:

Table4: Chiffrement de Delastelle

(Remarquons que les lettres vides sont remplacées par un double 00).

Ensuite, on effectue la transposition qui consiste à regrouper les chiffres deux par deux, de la gauche vers la droite, puis du haut vers le bas. On obtient alors : 33145421242345014550. La dernière opération consiste à transformer le code précédent en message littéral, en utilisant à

nouveau le carré de Polybe. Toutefois, il peut apparaître des nombres du type 00 01 02 10 20 etc. Qu'on remplace par des lettres spéciales ou des chiffres. Par exemple :

00——————————0 01——————————1
02——————————2 03——————————3
04——————————4 05——————————5
10——————————6 20——————————7
30——————————8 40——————————9
50——————————%

On trouve ici finalement le message chiffré : "NDYFIHU1U%"

5.1.4. Le chiffrement de Vigenère :[4]

En 1586, Blaise de Vigenère a développé son Tracté des chiffres ou Secrètes manièresd'écrire, qui ressemble au chiffrement de César, à la différence près qu'il utilise un mot clé au lieu d'un simple caractère. Pour chiffrer un message, on choisit une clé qui sera un mot de longueur arbitraire. On écrit ensuite cette clé sous le message à chiffrer, en la répétant aussi souvent que nécessaire pour que sous chaque lettre du message à chiffrer, on trouve une lettre de la clé. Pour chiffrer, on fait la somme modulo 26 de chaque caractère du message et du caractère de la clé correspondant. Exemple : On veut chiffrer le texte "CRYPTOGRAPHIE DE VIGENERE" avec la clé "SECRET". On commence par écrire la clé sous le texte à chiffrer :

Texte	C	R	Y	P	T	O	G	R	A	P	H	I	E	D	E	V	I	G	E	N	E	R	E
Clé	S	E	C	R	E	T	S	E	C	R	E	T	S	E	C	R	E	T	S	E	C	R	E
Nbres	2	17	24	15	19	14	6	17	0	15	7	8	4	3	4	21	8	6	4	13	4	17	4
x_1 x_2	18	4	2	17	4	19	18	4	2	17	4	19	18	4	2	17	4	19	18	4	2	17	4
+Mod26	20	21	0	6	23	7	24	21	2	6	11	1	22	7	6	12	12	25	22	17	6	8	8
Texte	U	V	A	G	X	H	Y	V	C	G	L	B	W	H	G	M	M	Z	W	R	O	I	I

Table5: Un chiffrement de Vigenère

Bien que ce chiffrement soit beaucoup plus sûr que le chiffrement de César, il peut encore être facilement casse.

5.1.5. Le chiffrement par transposition (par permutation) :[4]

La technique de chiffrement par transposition est vraisemblablement la première preuve de l'utilisation de moyens de chiffrement en Grèce dès 400 avant Jésus Christ pour dissimuler des messages écrits sur des bandes de papyrus.

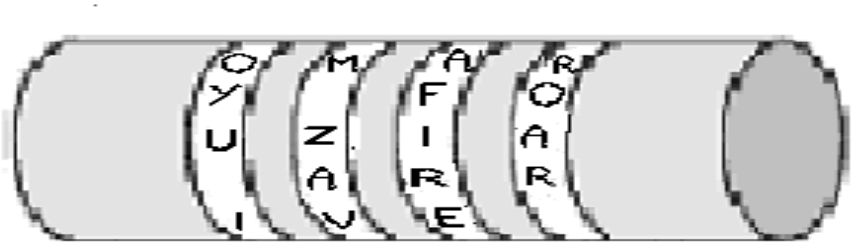


Figure24: Le cylindre de Scytale.

La technique consistait à:

– enrouler une bande de papyrus sur un cylindre appelé scytale
– écrire le texte longitudinalement sur la bandelette ainsi enroulée (le message dansl'exemple ci-dessus est "OMARY FOUZIA ARRIVE")

Le message une fois déroulé n'est plus compréhensible (dans l'exemple ci-dessus "OYU IM ZAVAFIREROAR "). Il suffit au destinataire d'avoir un cylindre de même rayon pour pouvoir déchiffrer le message. En réalité un casseur (il existait des casseurs à l'époque...) peut déchiffrer le message en essayant des cylindres de diamètre successifs différents, ce qui revient à dire que la méthode peut être cassée statistiquement (il suffit de prendre les caractères un à un, éloignés d'une certaine distance).

Formellement : On fixe un entier m≠0 et une permutation de {1, 2 …m} qui seront la clé de chiffrement, par exemple avec m=5 et la permutation

$$\begin{pmatrix} 1\,2\,3\,4\,5 \\ 5\,3\,1\,2\,4 \end{pmatrix}$$

Le message "CRYPTOGRAPHIE CLASSIQUE" sera chiffré en "TYCRPPROGACEHI ISLAS EQU". En effet, le texte en clair est décomposé en blocs de cinq caractères chacun. Chaque bloc est transformé en en utilisant la permutation ci-dessus. Par exemple CRYPT et OGRAP sont chiffrés respectivement en TYCRP et PROGA et ainsi de suite. La clé de déchiffrement est le nombre 5 et la permutation inverse :

$$\begin{pmatrix} 1\,2\,3\,4\,5 \\ 3\,4\,2\,5\,1 \end{pmatrix}$$

6. Cryptographie moderne :

Si le but de la cryptographie classique est d'élaborer des méthodes permettant de transmettre des données de manière confidentielle, la cryptographie moderne s'intéresse en fait plus généralement aux problèmes de sécurité des communications. Le but est d'offrir un certain nombre de services de sécurité comme la confidentialité, l'intégrité, l'authentification des données transmises. La cryptographie moderne se compose de deux grandes parties : La cryptographie asymétrique et la cryptographie symétrique présenter précédemment. Dans la partie qui suit on verra quelques algorithmes asymétriques ainsi quelques algorithmes symétriques connus.

6.1. Algorithme asymétrique :

6.1.1. RSA : Rivest - Shamir – Adleman :[7] [1] [6]

Le cryptage RSA, du nom de ses concepteurs, Ron Rivest, Adi Shamir et Leonard Adleman, est le premier algorithme de chiffrement asymétrique. Il a été découvert en 1977 au Massachusetts Institute of Technology. RSA est devenu un système universel servant dans une multitude d'applications : systèmes d'exploitation (Microsoft, Apple, Sun. . .), cartes à puces bancaires et bien sûr le réseau Internet pour assurer la confidentialité du courrier électronique et authentifier les utilisateurs et aussi la protection de dossiers hautement confidentiels.

Le RSA est basé sur la théorie des nombres premiers, et sa robustesse tient du fait qu'il n'existe aucun algorithme de décomposition d'un nombre en facteurs premiers. Alors qu'il est facile de multiplier deux nombres premiers, il est très difficile de retrouver ces deux entiers si l'on en connaît le produit.

6.1.1.1. Principe de fonctionnement :

Le RSA c'est un système à clé publique, ce qui signifie que l'algorithme de calcul n'est pas caché, ni la clé de codage (appelée de ce fait clé publique).La connaissance de la clé publique du destinataire permet à tous les émetteurs de crypter les messages qui ne pourront être décryptés que par le destinataire, grâce à sa clé secrète.

6.1.1.2. Génération des clés :

Le RSA fonctionne à partir de deux nombres premiers, que l'on appellera p et q. Ces deux nombres doivent être très grands, car ils sont la clé de voûte de notre cryptage. Une fois ces deux nombres déterminés, multiplions-les. On note n le produit :

n = p **x** q, et on calcule l'indicatrice d'Euler Ø(n) = Ø(p). Ø(q) = (p - 1) (q - 1).

Cherchons maintenant un nombre e (inférieur à Ø (n)), qui doit nécessairement être premier avec Ø (n). Calculons ensuite l'inverse de e modulo Ø (n), que nous noterons d. avec d =e^{-1}mod Ø (n).

Alors :

- La clé publique : c'est le couple **(n,e).**
- La clé privée : c'est le couple**(n,d).**

6.1.1.3.<u>Chiffrement :</u>

- Avant d'être chiffré, le message original doit être décomposé en une série d'entiers M de valeurs comprises entre 0 et n-1.
- Pour chaque entier M il suffit de le mettre à la puissance e. C =M^emod n. (C = M^e modulo n est calculé grâce à la méthode d'exponentiation modulaire).
- Le message chiffré est constitué de la succession des entiers C.

6.1.1.4.<u>Déchiffrement :</u>

- Conformément à la manière dont il a été chiffré, le message reçu doit être composé d'une succession d'entiers C de valeurs comprises entre 0 et n-1.
- Pour chaque entier C il faut calculer M =C^dmod n.
- Le message original peut alors être reconstitué à partir de la série d'entiers M.

6.1.1.5.<u>Exemple d'application :</u>[1]

Soient **p** = 31, **q** = 53 c'est-à-dire **n**=**1643**. **Ø(n)** = **1560** (nombre d'éléments relativement premiers à **n** et <**n**).

Soit **e** = **11** (par exemple, et on a bien (**e**,**Ø**(**n**))=**1**)**.**

On détermine que **d** = **851** (inverse modulaire de e sur $\mathbf{Z_{Ø(n)}}$).

La clé publique est donc (**11,1643**) et la clé privée est (**851,1643**).

Soit le codage par la position dans l'alphabet du mot «ANEMONE». Il vient

01 14 05 13 15 14 05

On procède selon deux conditions :

1. Découpage en morceaux de même longueur, ce qui empêche la simple substitution :

011 405 131 514 05_

On ajoute un padding initial si nécessaire.

001 140 513 151 405

Cela provoque la perte des patterns (« NE »).

2. Découpage en morceaux de valeur inférieure à n, car opération modulo n.

Lors du chiffrement, on a :

$001^{11} \bmod 1643$	0001
$140^{11} \bmod 1643$	0109
$513^{11} \bmod 1643$	0890
$151^{11} \bmod 1643$	1453
$405^{11} \bmod 1643$	0374

Table6: Chiffrement

Et pour le déchiffrement **:**

$0001^{851} \bmod 1643$	001
$0109^{851} \bmod 1643$	140
$0890^{851} \bmod 1643$	513
$1453^{851} \bmod 1643$	151
$0374^{851} \bmod 1643$	405

Table7: Déchiffrement

Lors du déchiffrement, sachant qu'il faut obtenir des blocs de 2 éléments (grâce au codage particulierde l'exemple), on a bien :

01	14	05	13	15	14	05
A	N	E	M	O	N	E

Table8: Déchiffrement du texte

6.1.1.6.Efficacité et robustesse de RSA :[6]

- Il est facile de générer des grands nombres premiers. tout au moins en acceptant un taux d'erreur Dans le cas de RSA, l'erreur n'est pas trop grave : en effet, si l'on commet une erreur en croyant que p et q sont premiers, le destinataire se rendra rapidement compte que les nombres ne sont pas premiers : soit la clé d n'est pas inversible, soit certains blocs du message décrypté sont incompréhensibles. Dans ce cas, on peut procéder à un changement de système RSA (recalcule de p et q) :

- le calcul du couple (e, d) est extrêmement facile : il suffit d'appliquer l'algorithme d'Euclide étendu ;

- enfin, chiffrement et déchiffrement sont réalisés par exponentiation modulaire.

- La sécurité fournie par RSA repose essentiellement sur la difficulté à factoriser de grands entiers. En effet, si un attaquant peut factoriser le nombre n = pq de la clé publique, il peut alors déduire directement Ø (n) = (p - 1) (q - 1) et donc calculer la clé privée à partir de la clé publique par l'algorithme d'Euclide étendu. Donc, si l'on dispose d'un algorithme rapide pour factoriser de grands entiers, casser RSA devient facile aussi.
- Après vingt ans de recherche, aucun moyen plus efficace que la factorisation de n n'a été publié pour casser RSA. Cependant, la réciproque : « si factoriser de grands entiers est dur, alors casser RSA est dur » n'a pas été prouvée. On peut cependant remarquer que si l'on choisit une petite clé publique e (par exemple e <= log n), alors casser RSA permet de factoriser n en temps polynomial.

6.2. Algorithme symétrique :

6.2.1. DES (Data Encryption Standard) :

6.2.1.1. Présentation : [1]

Le D.E.S. (Data Encryption Standard, c'est-à-dire Standard de Chiffrement de Données) est un standard mondial depuis la fin des années 1970. Au début de cette décennie, le développement des communications entre ordinateurs a nécessité la mise en place d'un standard de chiffrement de données pour limiter la prolifération d'algorithmes différents ne pouvant pas communiquer entre eux. Pour résoudre ce problème, L'Agence Nationale de Sécurité américaine (N.S.A.) a lancé des appels d'offres. La société I.B.M. a développé alors un algorithme nommé Lucifer, relativement complexe et sophistiqué. Après quelques années de discussions et de modifications (applications de S-Boxes et réduction à des clés de 56 bits), cet algorithme, devenu alors D.E.S., fut adopté au niveau fédéral le 23 novembre 1976.

6.2.1.2. Particularités : [1]

Le DES comporte plusieurs avantages qui en ont fait l'algorithme de chiffrement symétrique standard pendant longtemps, jusqu'il y a quelques années. En voici quelques-uns :

- Possède un haut niveau de sécurité,
- Complètement spécifié et facile à comprendre,
- La sécurité est indépendante de l'algorithme lui-même,
- Disponible à tous, par le fait qu'il est public,
- Adaptable à diverses applications (logicielles et matérielles),
- Rapide et exportable,

- Repose sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement,
- Facile à implémenter.

Conclusion :

Cette partie a été procédé pour voir les notions de bases de la cryptographie de son histoire jusqu'au dernier algorithme moderne symétrique**DES**(Data Encryption Standard) ainsi quelques algorithmes classiques utilisés à l'époque pour sécuriser leurs informations on utilisant la méthode de chiffrement .Jusqu'à l'année1997 le DES était l'algorithme dominant et le plus utilisé mais après plusieurs attaques réussies sur ce dernier le Data Encryption Standard a été trouvé trop faible à cause de l'évolution technologique très rapide alors dans la partie suivante on verra l'algorithme qui va remplacer le **DES** et comment a été choisie et pourquoi.

CHAPITRE III.

AES (Advanced Encryption Standard)

1. Introduction à l'Advanced Encryption Standard:

Advanced Encryption Standard, dans la suite référencée comme **AES**, est le gagnant du concours, qui s'est tenue en 1997 par le gouvernement américain, après le **Data Encryption Standard** a été trouvé trop faible en raison de sa petite taille de la clé et les progrès technologiques dans la puissance du processeur. Quinze candidats ont été acceptés en 1998 et basées sur les commentaires du public de la piscine a été réduite aux finalistes tapa dans la main en 1999 En Octobre 2000, l'un de ces cinq algorithmes a été choisi comme la future norme. Une version légèrement modifiée de la **Rijndael**.

Le **Rijndael**, dont le nom est basé sur le nom de ses deux inventeurs belges, **Joan Daemen** et **Vicent Rijmen,** est un chiffrement de bloc, ce qui signifie qu'il fonctionne sur le groupe de longueur fixe de bits, qui sont appelés blocs. Il prend un bloc d'entrée d'une certaine taille, le plus souvent 128, et produit un bloc de sortie correspondant de la même taille. La transformation exige une deuxième entrée, qui est la clé secrète. Il est important de savoir que la clé secrète peut être de n'importe quelle taille (selon le chiffrement utilisé) et qu'**AES** utilise trois tailles de clés différentes: 128, 192 et 256 bits.

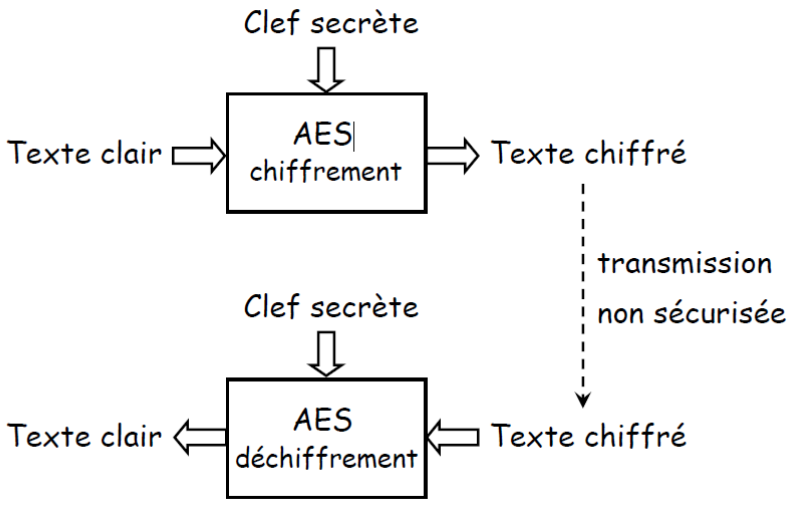


Figure 25 : Chiffrement AES

Alors qu'**AES** prend en charge uniquement les tailles de blocs de 128 bits et les clés de 128, 192 et 256 bits, le **Rijndael** d'origine prend en charge des tailles de clés et blocs en multiples de 32, avec un minimum de 128 et un maximum de 256 bits.

Autres lectures: Contrairement à **DES**, qui est basé sur un réseau de **Feistel, AES** est un réseau de substitution-permutation, qui est une série d'opérations mathématiques qui utilisent des substitutions (aussi appelé **S-Box**) et de permutations (**p-boxes**) et leur définition précise implique que chaque bit de sortie dépend de chaque bit d'entrée.

2. Les résultats d'un concours : [1]

La progression de la puissance des ordinateurs a causé la mort du DES. Ce dernier n'est plus jamais utilisé lorsque la sécurité demandée est forte (**utilisation militaire**, **documents "secrets"**, etc.). Pour cette tâche, on préfère utiliser l'algorithme connu sous le nom générique d'**AES** (Advanced Encryption Standard), issu d'un concours créé en raison des faiblesses avérées du DES. Le véritable nom de l'**AES** est le **Rijndael**.

Le **Triple DES** demeure toutefois une norme acceptée pour les documents gouvernementaux. Pour l'instant, il n'y a pas de projet ou d'obligation de rechiffrer les documents existants.

Cahier des charges : 1

Au second tour du concours, les jurys devaient juger différents critères :

- La sécurité générale,
- Le coût en termes de calculs (rapidité),
- La simplicité de l'algorithme et ses facilités d'implémentation,
- Une lecture facile de l'algorithme, puisqu'il est destiné à être rendu public,
- La résistance aux attaques connues,
- Flexibilité - Portabilité : l'algorithme devant remplacer le **DES**, il est destiné à servir aussi bien dans les cartes à puces, aux processeurs 8 bits peu puissants, que dans des processeurs spécialisés pour chiffrer des milliers de télécommunications à la volée.
- Techniquement, le chiffrement doit se faire par blocs de 128 bits, les clés comportant 128, 192 ou256 bits.

Au niveau du chiffrement/déchiffrement, les résultats varient assez fortement. Cependant, **Serpent** reste le moins bon pour la majorité des plates-formes, **Rijndael** et **RC6** étant les meilleurs.

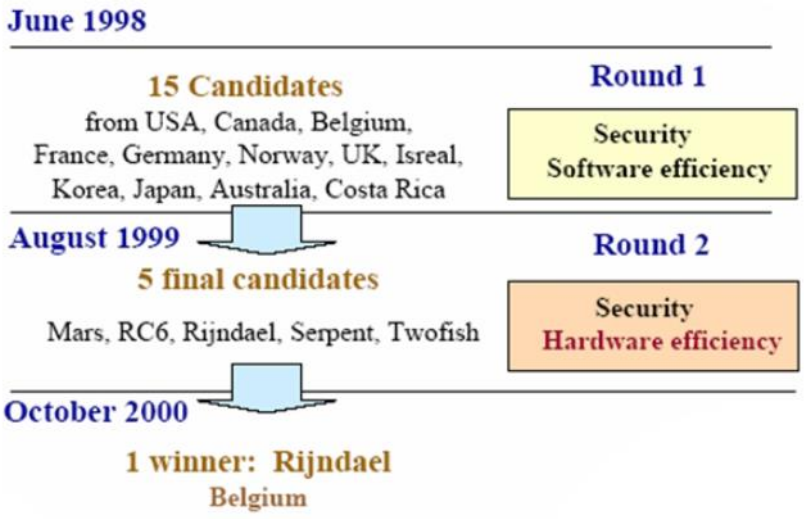


Figure 26 : Différents participants au concours AES

3. Caractéristiques et points forts de l'AES :[23]

Le choix de cet algorithme répond à de nombreux critères plus généraux dont nous pouvons citer les suivants :

- sécurité ou l'effort requis pour une éventuelle cryptanalyse,
- facilité de calcul : cela entraine une grande rapidité de traitement,
- besoins en ressources et mémoire très faibles,
- flexibilité d'implémentation: cela inclut une grande variété de plateformes et d'applications ainsi que des tailles de clés et de blocs supplémentaires,
- hardware et software : il est possible d'implémenter l'**AES** aussi bien sous forme logicielle que matérielle (câblé),
- simplicité : le design de l'**AES** est relativement simple.

Si l'on se réfère à ces critères, on voit que l'**AES** est également un candidat particulièrement approprié pour les implémentations embarquées qui suivent des règles beaucoup plus strictes en matière de ressources, puissance de calcul, taille mémoire, etc... C'est sans doute cela qui a poussé le monde de la 3G (3ème génération de mobiles) à adopter l'algorithme pour son schéma d'authentification "Millenage".

4. Le choix : Rijndael[1][8]

A la suite de nombreux tests, c'est finalement **Rijndael** qui a remporté la médaille, et est ainsi devenule remplaçant officiel du **DES**.

Il possède les propriétés suivantes :

- Plusieurs longueurs de clef et de bloc sont possibles : 128, 192, ou 256 bits ;

- Le nombre de cycles ("rondes") varie en fonction de la longueur des blocs et des clés (de 10 à 14) ;
- La structure générale ne comprend qu'une série de transformations/permutations/sélections ;
- Il est beaucoup plus performant que le **DES** ;
- Il est facilement adaptable à des processeurs de 8 ou de 64 bits ;
- Le parallélisme peut être implémenté.

	Pas de Réponse	OUI	?	NON	Abs.	RANG
Rijndael	7	77	19	1	76	1
RC6	4	79	15	6	73	2
Twofish	9	64	28	3	61	3
MARS	5	58	35	6	52	4
Serpent	6	52	39	7	45	5
E2	11	27	53	13	14	6
CAST-256	12	16	58	18	-2	7
SAFER+	13	20	47	24	-4	8
DFC	12	22	43	27	-5	9
Crypton	14	16	43	31	-15	10
DEAL	10	1	22	71	-70	11
HPC	12	1	13	78	-77	12
MAGENTA	9	1	10	84	-83	13
Frog	11	1	6	86	-85	14
LOKI97	10	1	7	86	-85	14

Figure 27 : Vœux de présence des algorithmes au second tour.[8]

5. Présentation de l'algorithme : [8][1]

L'algorithme se présente en deux temps, tout d'abord une procédure d'expansion de la clef, puisla fonction principale de chiffrement.

La fonction de chiffrement se divise en trois : une transformation initiale avec la clé, une série detours puis une transformation finale.

Le nombre de tours s'établit en fonction de la taille des blocs et de la clé :

Block length	128 bits Nk=4	192 bits Nk=6	256 bits Nk=8
128 bits Nb=4	10	12	14
192 bits Nb=6	12	12	14
256 bits Nb=8	14	14	14

Figure 29 : Nombres de rondes à effectuer

À chaque ronde, quatre transformations sont appliquées :**[1]**

- substitution d'octets dans le tableau d'état ;
- décalage de rangées dans le tableau d'état ;
- déplacement de colonnes dans le tableau d'état (sauf à la dernière ronde) ;
- addition d'une "clef de ronde" qui varie à chaque ronde.

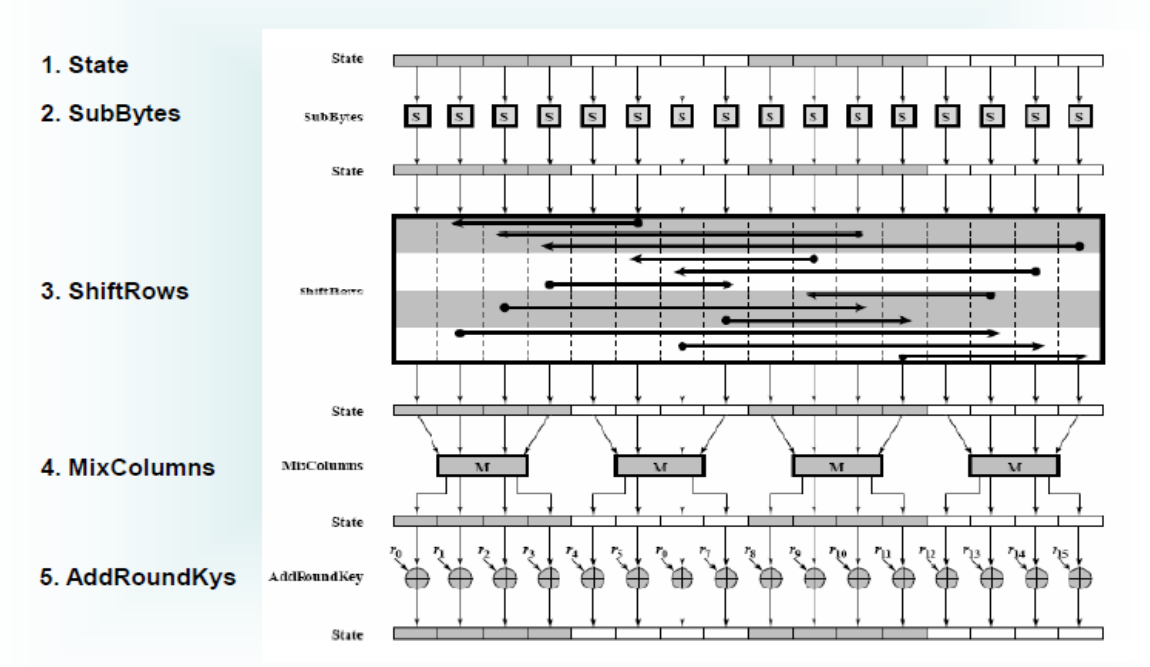


Figure 30 : Schéma général de Rijndael

6. Le Chiffrement AES :[6][1][24]

AES opère sur une matrice **$4xN_b$** d'éléments de $\mathbf{F_{256}}$, notée State. Le chiffrement **AES** consiste en une addition initiale de clé, notée **AddRoundKey**, suivie par $\mathbf{N_{r-1}}$ rondes, chacune constitué de quatre étapes (qui seront détaillées dans la section suivante) :

a. **SubBytes :** il s'agit d'une substitution non-linéaire lors de laquelle chaque octet est remplacé par un autre octet choisi dans une table particulière une Boite **S-Box**.

b. **ShiftRows :** est une étape de transposition ou chaque éléments de la matrice est décalée cycliquement à gauche d'un certain nombre de colonnes.

c. **MixColumns :** effectue un produit matriciel en opérant sur chaque colonne (vu alors comme un vecteur) de la matrice.

d. **AddRoundKey :** qui combine par addition chaque octet avec l'octet correspondant dans une clé de ronde obtenue par diversification de la clé de chiffrement.

Enfin, une ronde finale est appliquée (elle correspond à une ronde dans laquelle l'étape MixColumns est omise).

L'ordonnancement des étapes est illustré à la figure suivante :

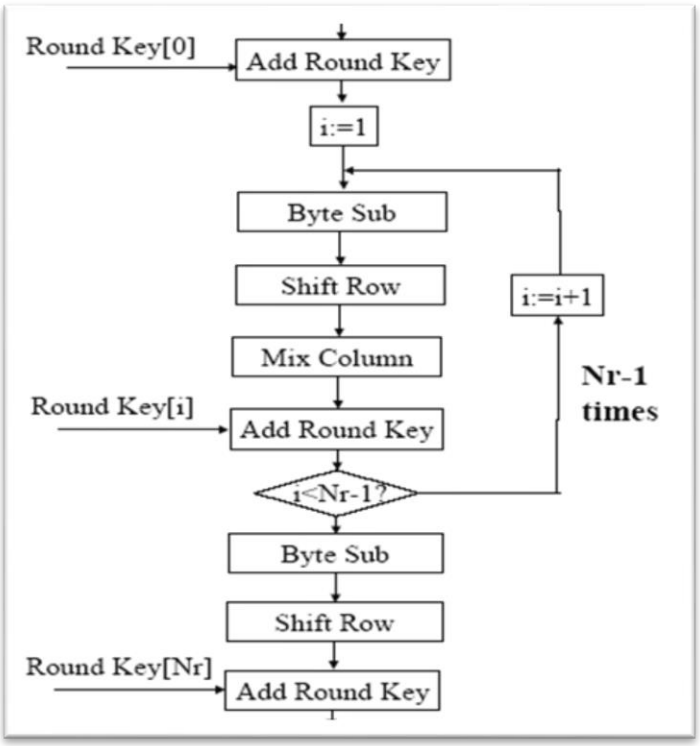


Figure 31 : Schéma des déférentesétapes de chiffrement [1]

6.1. Table d'état du texte et des clés :[1]

Le message et la clé sont conservés sous forme de tables représentées respectivement aux figures Suivante. Le nombre de colonnes dépend des tailles des textes et clés :

- N_b = Lbloc/32
- N_k = Lclef /32

Une colonne du tableau correspond à un mot de 32 bits. Ainsi, chaque petit bloc représente 8 bits, donc
1 octet. L'input et l'output sont donc gérés comme des séquences linéaires d'octets.

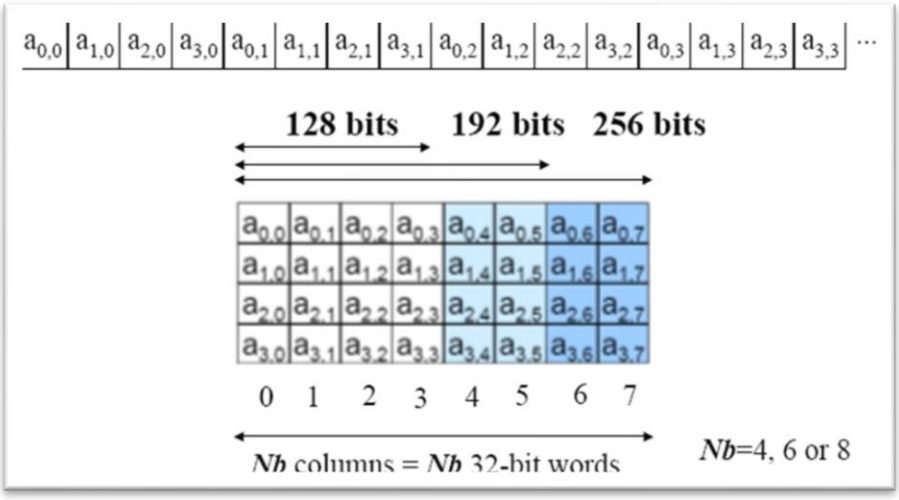


Figure 32 : Table d'état du texte

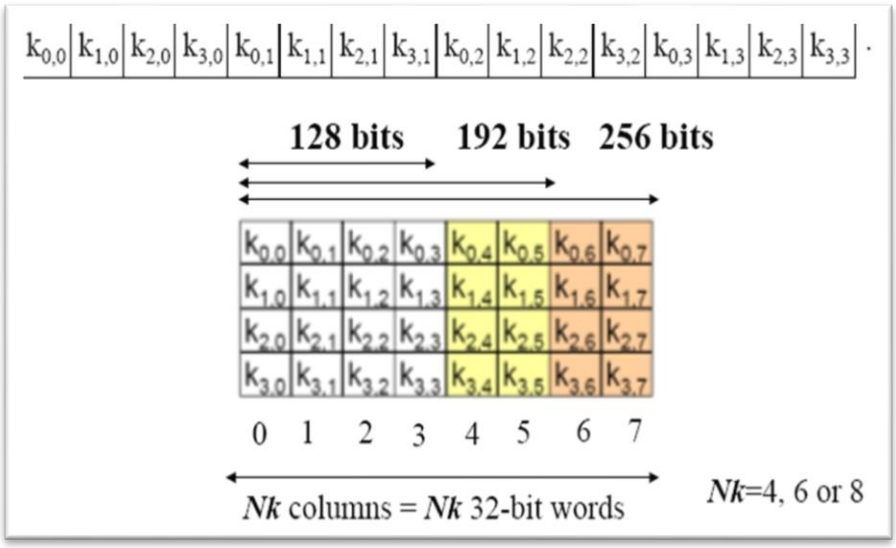


Figure 33 : Tables d'états des clés

6.2. L'étape SubByte :[6]

L'étape SubBytes correspond à la seule transformation non linéaire de l'algorithme .Dans cette étape, chaque élément de la matrice State (état) est permuté selon une table de substitution inversible notée **S-Box** .la figure suivante Illustre une transformation d'un élément en utilisant **S-box** :

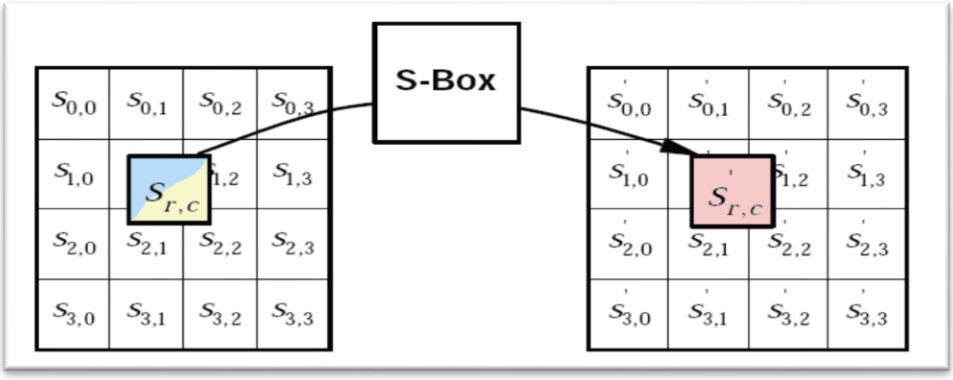


Figure 34 : Fonction SubByte

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

Figure 35 : Table S-Box

Exemple (Transformation d'un élément de la matrice state) :

Pour $s_{1,1}$= {53}

$s'_{1,1}$= SubBytes($s_{1,1}$) = {ED}.

6.3. L'étape ShiftRow:[1]

L'étape ShiftRow opère sur les lignes de la matrice State et effectue pour chaque élément d'une ligne un décalage cyclique de n éléments vers la gauche. L'offset **n** de décalage dépend de la ligne considérée.

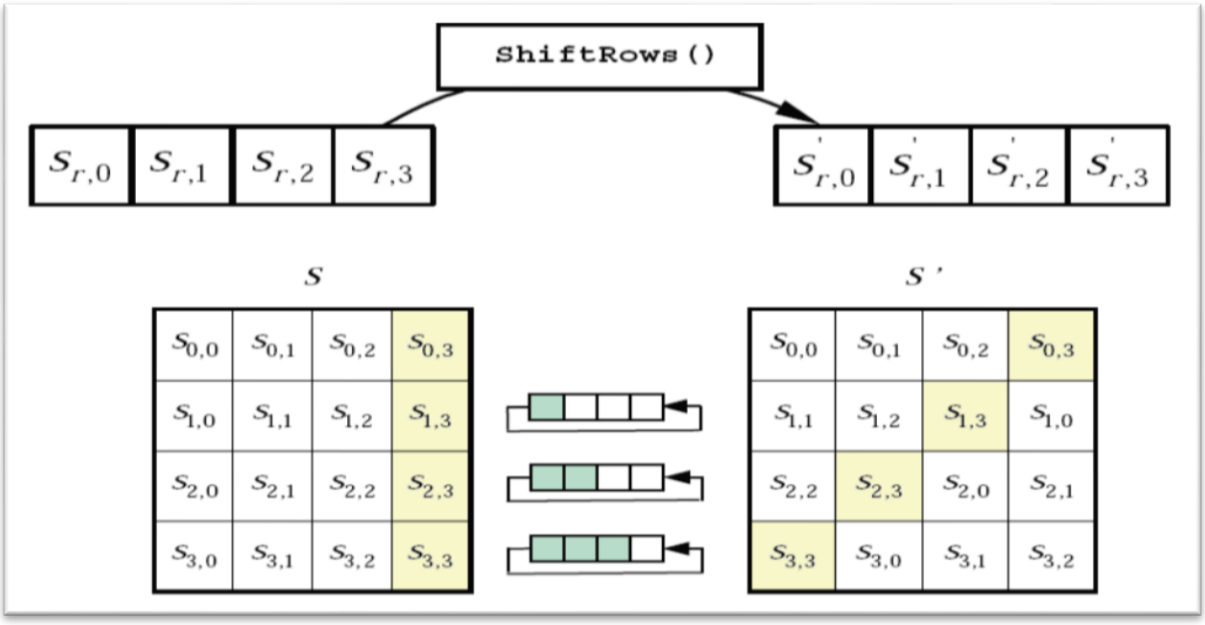


Figure 36 : Schéma de l'étape ShiftRow

Selon la taille des blocs de message (c'est-à-dire la valeur de N_b), les décalages ne seront pas toujoursidentiques.

- La ligne **0** n'est jamais décalée,
- La ligne **1** est décalée de **C1**,
- La ligne **2** est décalée de **C2**,
- La ligne **3** est décalée de **C3**.

- Le décalage défère selon les blocs des messages :

	C_1	C_2	C_3
N_B=4	1	2	3
N_B=6	1	2	3
N_B=8	1	3	4

Décalage selon la taille des blocs de messages.

6.4. L'étape MixColumns : [6]

La transformation **MixColumns**opère sur les colonnes **c** de la matrice state en le traitant comme un polynôme**a(x)** de degré 3 a coefficients dans $\mathbf{F_{256}}$.

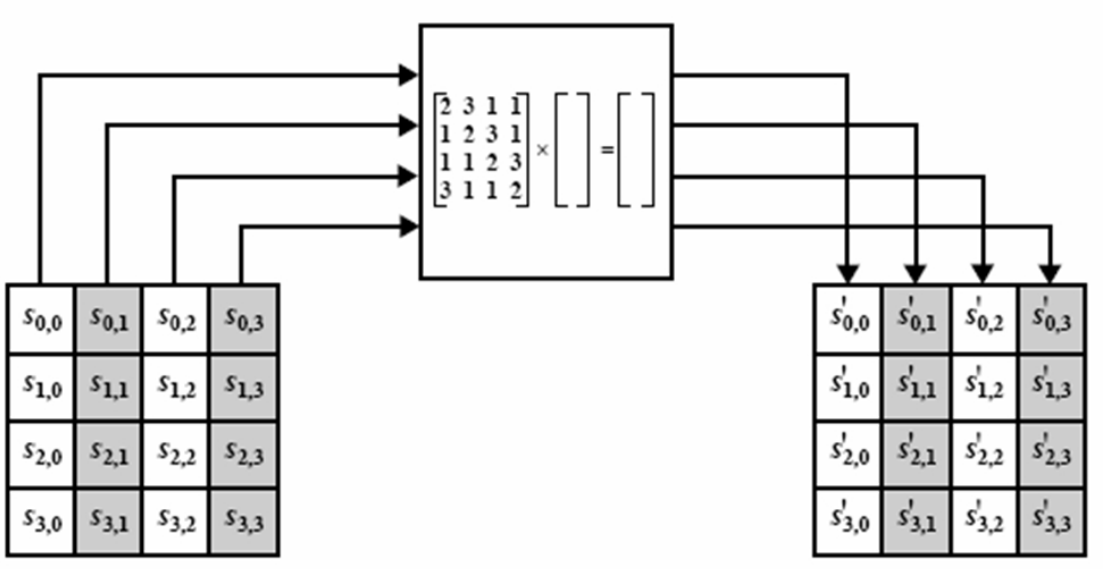


Figure 37 : Etape du MixColumns

L'étape**MixColumns** consiste alors àeffectuer, pour chaque colonne, une multiplication par un polynôme**c(x)**fixé, suivie d'une réduction Modulo le polynôme$\mathbf{x^4+1}$.

Dans MixColumns, on réalise donc l'opération :

$(03x^3 + x^2 + x + 02)$ **x**a(x) mod $(x^4 + 1)$ Matriciellement, cette opération s'écrit :

$$b(x) = c(x) \times a(x) \bmod (x^4+1) \Leftrightarrow$$

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix}$$

$$s'_{0,c} = (\{02\} \bullet s_{0,c}) \oplus (\{03\} \bullet s_{1,c}) \oplus s_{2,c} \oplus s_{3,c}$$

$$s'_{1,c} = s_{0,c} \oplus (\{02\} \bullet s_{1,c}) \oplus (\{03\} \bullet s_{2,c}) \oplus s_{3,c}$$

$$s'_{2,c} = s_{0,c} \oplus s_{1,c} \oplus (\{02\} \bullet s_{2,c}) \oplus (\{03\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{03\} \bullet s_{0,c}) \oplus s_{1,c} \oplus s_{2,c} \oplus (\{02\} \bullet s_{3,c})$$

6.5. L'étape Add Round Key:[1]

C'est un simple **XOR**. Il s'agit d'additionner des sous-clés aux sous-blocs correspondants.

- **L'operateur XOR** :

XOR		
0	0	0
0	1	1
1	0	1
1	1	0

$X \oplus 0 = X$

$X \oplus 1 = not(X)$

$X \oplus X = 0$ $\qquad X \oplus a \oplus X = a$

$X \oplus not(X) = 1$

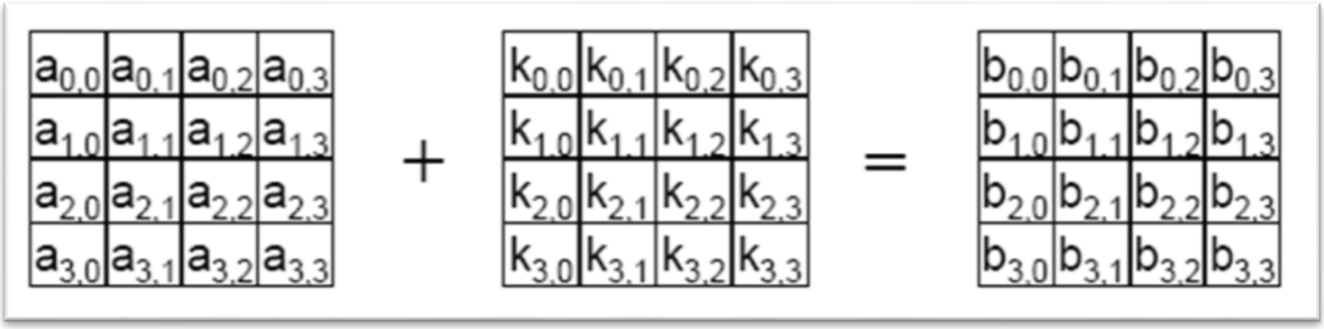


Figure 38 : Etape AddRoundKey

Exemple (Addition exclusifs de deux colonnes):

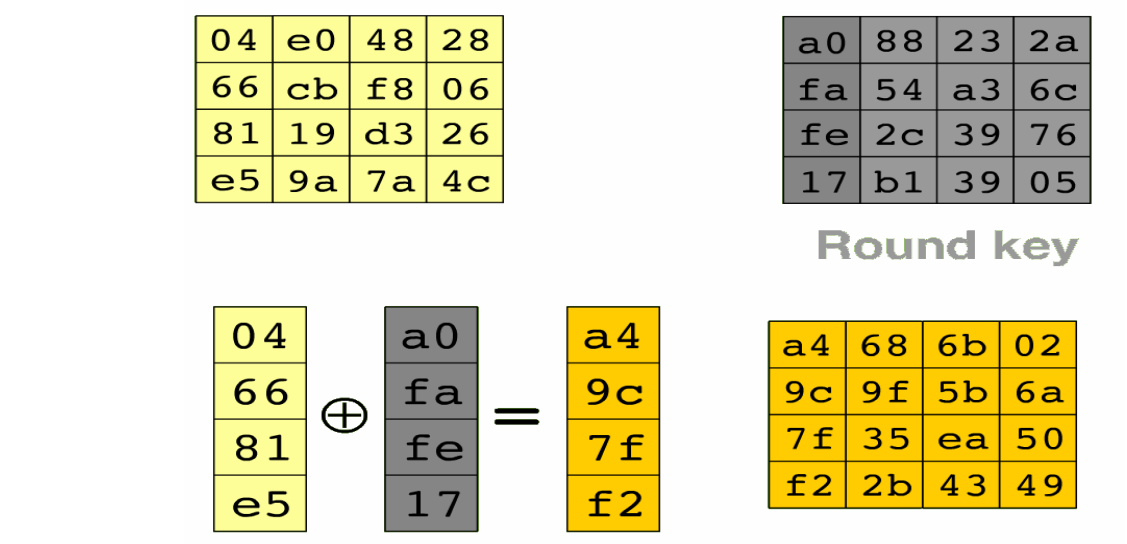


Figure 39 : Exemple de l'addition de deux colonnes

6.6. Calcul de la clé (Key Expansion) : [1] [24]

Cette étape, noté KeyExpansion, permet de diversifier la clé de chiffrement K (de $\mathbf{4N_k}$ octets) dans une clé étendue W de $\mathbf{4N_b\ (N_r+1)}$ octets. On disposera ainsi de $\mathbf{N_r+1}$clés de rondes (chacune de $\mathbf{4N_b}$octet). [**24]**

Après avoir subi une extension (Key Expansion), la clé sera découpée en sous-clés (appelées clés de rondes), comme indiqué à la figure :

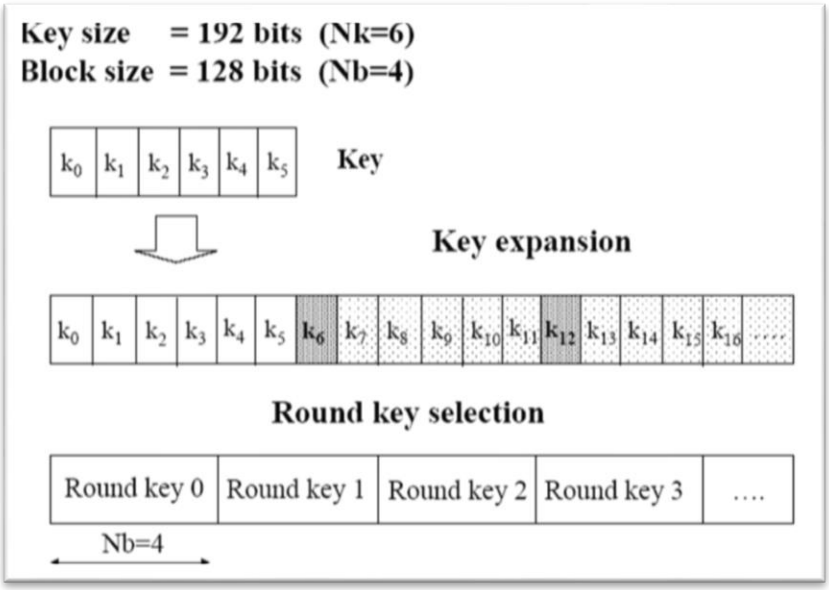


Figure 40 : Schéma des opérations effectuées sur la clé

Le nombre de sous-blocs $\mathbf{k_i}$ dépendra bien sûr de la taille des clés et bloc du message.

6.6.1. Extension de la clé :[1]

Le calcul de l'expansion de la clé se fait de deux manières distinctes selon le sous-bloc de la cléconcerné, comme l'illustrent les deux figures.

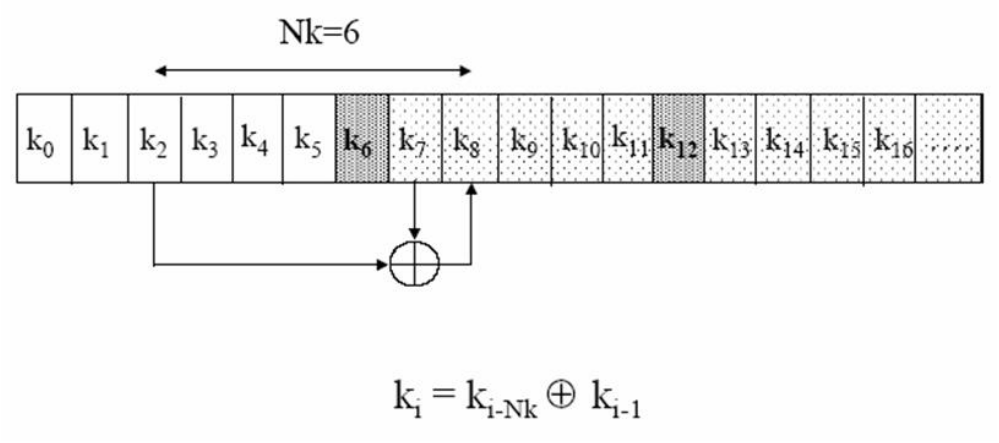


Figure 41 : Expansion de la clé avec bloc "commun"

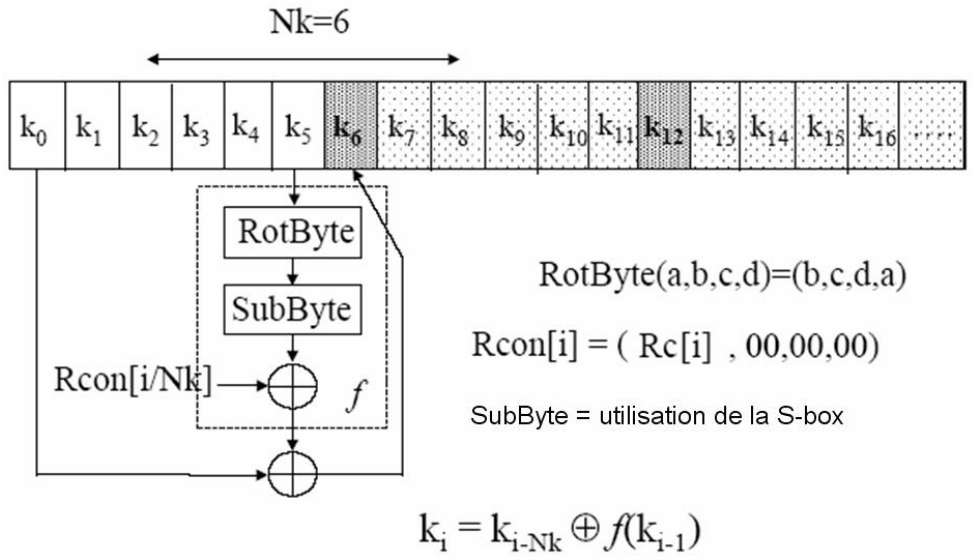


Figure 42 : Expansion de la clé avec les blocs "multiples de N_k"

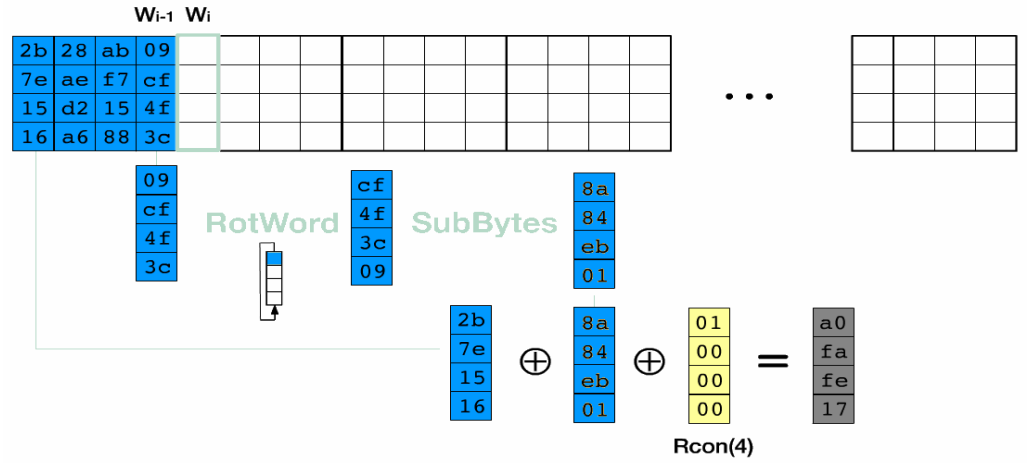
Remarque :

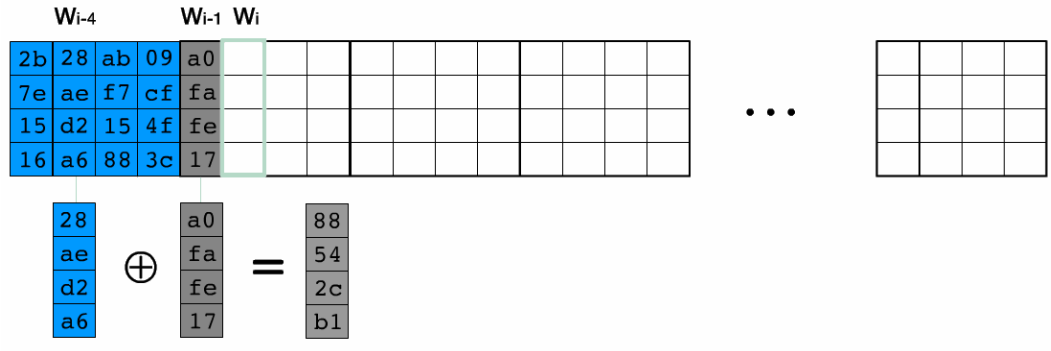
L'ajout de "Rcon[x]" donne comme résultat un **xor** sur les bits les plus significatifs. La table utilisée :

01	02	04	08	10	20	40	80	1b	36
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00
00	00	00	00	00	00	00	00	00	00

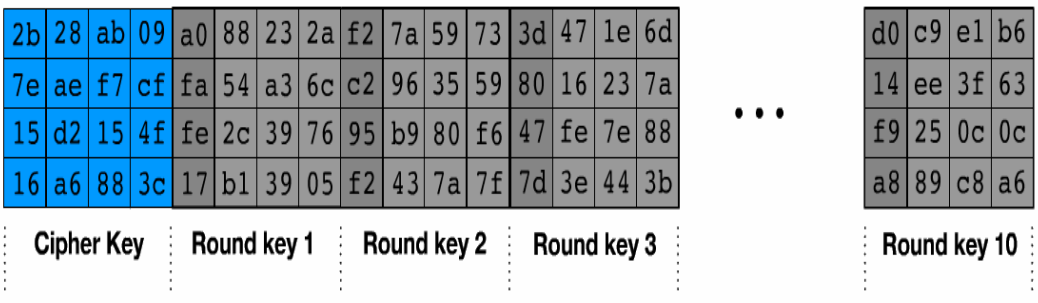
Figure 43 : Table de correspondance des Rcon[]

Exemple d'expansion des clés avec $N_k = 4$:





Et voila le résultat final on calculant tout les rondes :



7. Déchiffrement : [1] [22] [24]

La routine de chiffrement peut être inversée et réordonnée pour produire un algorithme de déchiffrement utilisant les transformations InvSubBytes, InvShiftRows, InvMixColumns, et AddRoundKey. Une modélisation formelle de l'algorithme de déchiffrement en pseudo-C pourrait être:

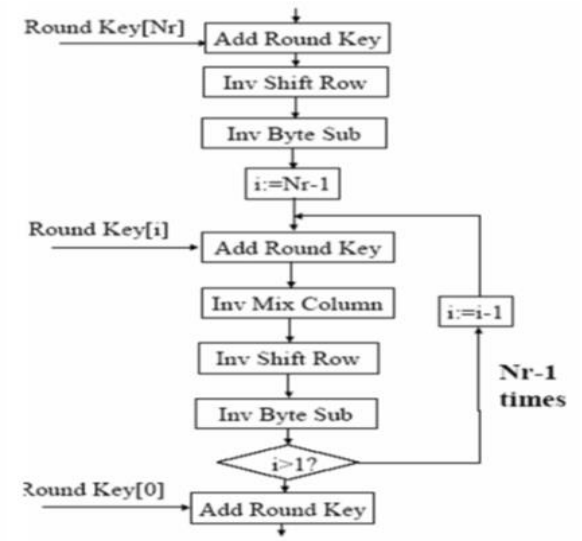


Figure 44 : Schéma des déférentes étapes de déchiffrement [1]

7.1. InvShiftRows () Transformation : [22]

InvShiftRows () est l'inverse de l'étape ShiftRows () transformation. Les octets dans les trois dernières lignes de l'État sont décalés cycliquement sur différents nombres d'octets. La première ligne, r = 0, n'est pas déplacée. Les bas trois lignes sont décalées cycliquement par Nb -shift (r, Nb) octets, où le changement de valeur de décalage (r, Nb) dépend du nombre de lignes, et est donnée dans l'équation.

$$s'_{r,(c+shift(r,Nb)) \bmod Nb} = s_{r,c} \quad \text{for } 0 < r < 4 \quad \text{and} \quad 0 \le c < \boldsymbol{Nb}$$

Plus précisément, les InvShiftRows () transformation se déroule comme suit:

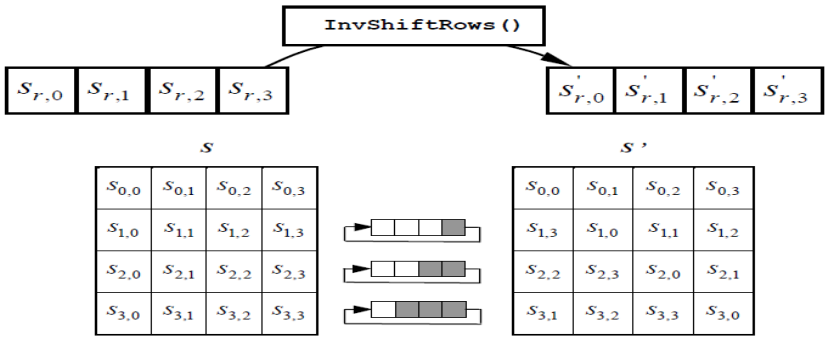


Figure 45 : InvShiftRows () déplace cycliquement les trois dernières lignes de l'État.

7.2. InvSubBytes () Transformation :[22]

InvSubBytes () est l'inverse de la transformation de la substitution d'octets, dans lequel la boîte de S-inverse est appliqué à chaque octet de l'Etat. Ceci est obtenu en appliquant l'inverse de la transformation, puis en prenant l'inverse multiplicatif de GF (28). Le S-box inverse utilisé dans les InvSubBytes () transformation est présentée dans la figure suivante:

x \ y	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	52	09	6a	d5	30	36	a5	38	bf	40	a3	9e	81	f3	d7	fb
1	7c	e3	39	82	9b	2f	ff	87	34	8e	43	44	c4	de	e9	cb
2	54	7b	94	32	a6	c2	23	3d	ee	4c	95	0b	42	fa	c3	4e
3	08	2e	a1	66	28	d9	24	b2	76	5b	a2	49	6d	8b	d1	25
4	72	f8	f6	64	86	68	98	16	d4	a4	5c	cc	5d	65	b6	92
5	6c	70	48	50	fd	ed	b9	da	5e	15	46	57	a7	8d	9d	84
6	90	d8	ab	00	8c	bc	d3	0a	f7	e4	58	05	b8	b3	45	06
7	d0	2c	1e	8f	ca	3f	0f	02	c1	af	bd	03	01	13	8a	6b
8	3a	91	11	41	4f	67	dc	ea	97	f2	cf	ce	f0	b4	e6	73
9	96	ac	74	22	e7	ad	35	85	e2	f9	37	e8	1c	75	df	6e
a	47	f1	1a	71	1d	29	c5	89	6f	b7	62	0e	aa	18	be	1b
b	fc	56	3e	4b	c6	d2	79	20	9a	db	c0	fe	78	cd	5a	f4
c	1f	dd	a8	33	88	07	c7	31	b1	12	10	59	27	80	ec	5f
d	60	51	7f	a9	19	b5	4a	0d	2d	e5	7a	9f	93	c9	9c	ef
e	a0	e0	3b	4d	ae	2a	f5	b0	c8	eb	bb	3c	83	53	99	61
f	17	2b	04	7e	ba	77	d6	26	e1	69	14	63	55	21	0c	7d

Figure 46 : S-box inverse

Pour notre exemple précèdent dans le chiffrement :

Pour $s_{i,j}$= {ed}

$s'_{i,j}$ = Inv Sub Bytes($s_{i,j}$) = {53}

7.3. InvMixColumns () Transformation : [22]

InvMixColumns () est l'inverse de la MixColumns () transformation. (InvMixColumns) opère sur l'état colonne par colonne, le traitement de chaque colonne comme un à quatre polynôme. Les colonnes sont considérés comme des polynômes sur GF (28) et multiplier modulo $x^4 + 1$ avec un polynôme fixe, donnée par :

$$a^{-1}(x) = \{0b\}x^3 + \{0d\}x^2 + \{09\}x + \{0e\}.$$

Ce qui peut être écrit comme une multiplication de matrice comme suit :

$s'(x) = a^{-1}(x) \otimes s(x)$:

$$\begin{bmatrix} s'_{0,c} \\ s'_{1,c} \\ s'_{2,c} \\ s'_{3,c} \end{bmatrix} = \begin{bmatrix} 0e & 0b & 0d & 09 \\ 09 & 0e & 0b & 0d \\ 0d & 09 & 0e & 0b \\ 0b & 0d & 09 & 0e \end{bmatrix} \begin{bmatrix} s_{0,c} \\ s_{1,c} \\ s_{2,c} \\ s_{3,c} \end{bmatrix} \quad \text{for } 0 \le c < \boldsymbol{Nb}.$$

En tant que résultat de cette multiplication, les quatre octets dans une colonne sont remplacés par ce qui suit:

$$s'_{0,c} = (\{0e\} \bullet s_{0,c}) \oplus (\{0b\} \bullet s_{1,c}) \oplus (\{0d\} \bullet s_{2,c}) \oplus (\{09\} \bullet s_{3,c})$$

$$s'_{1,c} = (\{09\} \bullet s_{0,c}) \oplus (\{0e\} \bullet s_{1,c}) \oplus (\{0b\} \bullet s_{2,c}) \oplus (\{0d\} \bullet s_{3,c})$$

$$s'_{2,c} = (\{0d\} \bullet s_{0,c}) \oplus (\{09\} \bullet s_{1,c}) \oplus (\{0e\} \bullet s_{2,c}) \oplus (\{0b\} \bullet s_{3,c})$$

$$s'_{3,c} = (\{0b\} \bullet s_{0,c}) \oplus (\{0d\} \bullet s_{1,c}) \oplus (\{09\} \bullet s_{2,c}) \oplus (\{0e\} \bullet s_{3,c})$$

7.4. Inverse de laAddRoundKey() Transformation :[22]

L'inverse de l'étape AddRoundKey(), est la mêmequi a été décritdans la Section précédente (**L'étape AddRoundKey**), est sonpropre inverse, puisqu'il s'agitseulementd'une applicationdel'opération **XOR**.

8. Avantages et limites de AES :[1]

Les principaux avantages sont :

- des performances très élevées,
- la possibilité de réalisation en "Smart Card" avec peu de code,
- la possibilité de parallélisme,
- il ne comprend pas d'opérations arithmétiques : ce sont uniquement des décalages et des XOR,
- il n'utilise pas de composants d'autres cryptosystèmes,
- il n'est pas fondé sur des relations obscures entres opérations,
- le nombre de rondes peut facilement être augmenté si c'est requis,
- il ne possède pas de clés faibles,

- il est résistant à la cryptanalyse différentielle et linéaire.

Quelques inconvénients et limites :

- le code et les tables sont différents pour le chiffrement et déchiffrement,
- le déchiffrement est plus difficile à implanter en "Smart Card",
- dans une réalisation matérielle, il y a peu de réutilisation des circuits de chiffrement pour effectuerle déchiffrement.

9. Comparaisons des algorithmes au niveau de la sécurité (AES et 3DES):[23]

9.1. Attaques par dictionnaires :

Nous allons comparer ici l'AES au 3DES qui est son concurrent le plus direct (le DES n'étant pratiquement plus utilisé dans sa forme simple). Le 3DES est, comme son nom l'indique, l'enchainement de 3 DES simples dans l'ordre DES / DES^{-1} / DES. Il est évident à prime abord que chaque opération utilise une clé distincte, car sans cela les 2 premières s'annuleraient (DES / DES^{-1}). Mais en pratique, on n'utilise que 2 clés différentes (que l'on alterne) car l'utilisation d'une troisième clé ne rajoute aucune sécurité.

En effet, l'attaque la plus courante contre le triple DES consiste à créer des dictionnaires multiples de façon à scinder(Diviser) le schéma en 2 parties et diminuer ainsi d'autant le nombre de possibilités à tester. En pratique, on séparera les 2 premières opérations DES de la 3ième et dernière.

- **La première partie :** conduit à l'élaboration d'un dictionnaire dont la taille est définie par le calcul suivant: le premier DES utilise une clé de 56 bits, il y a donc 2^{56} cas possibles. C'est pareil pour le deuxième DES, sauf que qu'il faut le multiplier au premier cas, soit un total de 2^{112}possibilités.
- **La deuxième partie :** ne comporte qu'un seul DES, donc 2^{56} possibilités pour la clé. Il suffit ensuite de faire correspondre ces 2 dictionnaires pour trouver la valeur qui est commune aux 2, nous donnant ainsi la bonne combinaison de clés. De manière générale et arrondie, la sécurité de l'algorithme peut donc être évaluée à 2^{113}.En ce qui concerne l'AES, c'est un algorithme qui ne présente qu'une seule étape, donc le calcul est simple : comme cité précédemment, il y 2^{128} clés possibles (dans la version minimale ou la clé ne fait "que" 128 bits de long). C'est directement la force de l'algorithme.

9.2. Attaques par cryptanalyse différentielle (DC) :

L'attaquant choisit des textes clairs présentant une différence fixe, calcule les chiffrés (en ayant accès au système) et leurs différences puis assigne des probabilités à certains types de clés. Plus le nombre

d'essais augmente, plus la probabilité de la bonne clé ne devientforte. Dans le cas du DES simple, cette attaque nécessite 2^{47} textes clairs et 2^{47} chiffrements pour retrouver la clé; néanmoins, les textes clairs doivent être, soigneusementchoisis. L'AES est lui résistant à ce type d'attaque.

9.3. Attaques par cryptanalyse linéaire (LC) :

L'attaquant utilise des approximations linéaires pour décrire les opérations conduisant au chiffré. Comme précédemment, plus le nombre d'essais augmente, plus la probabilité de la bonne clé ne devientforte. Cette attaque est actuellement la plus performante puisqu'elle ne nécessite que 2^{43} textes clairs et 2^{43} chiffrements pour retrouver une clé DES (simple).L'AES est lui résistant à ce type d'attaque.

Remarque :

En conclusion, l'AES est plus sûr que le 3DES car il présente, entre autres, une plus grande résistance aux attaques par dictionnaires de clés. Les autres attaques ne sont pas applicables dans son cas.

Conclusion :

A travers ce chapitre, on a détaillé le fonctionnement de l'algorithme **AES** (Advanced Encryption Standard) et on a vu tous les étapes utilisé de cet algorithme pour le **chiffrement** et le **déchiffrement,** et on a vu quelques avantages et inconvénients, finalement une petite comparaison entre les deux algorithmes **AES** et **3DES** qui reste un algorithme utilisé et une version développé du**DES**.

Dans la prochaine partie on va passer à l'étape de conception pour la programmation de L'algorithme**AES** avec le langage JAVA et comment intégrer l'algorithme **RSA** (l'algorithme a clé public) dans AES pour le transfert de la clé secrèted'**AES**en public.

CHAPITRE IV.

Conception et Réalisation

1. Introduction :

Apres avoir vu les notionsde bases sur la sécurité réseauprécisément sur la sécurité de l'information et les notions de bases de la cryptographie qui présente une des méthodes de la sécurité de l'information ,ainsi l'algorithme AES qui présente aujourd'hui comme le plus utilisé et l'algorithme de cryptage de l'actualité après avoir remplacé le DES par ce dernier .

Dans cette partie on passe à la réalisation de cet algorithme avec un langage qu'on a choisie bref le **JAVA**, on commence par une petite introduction de ce langage qui est vraiment vaste et reconnue ainsi on passe à la conception de notre algorithme en présentant son fonctionnement ainsi quelques méthodesutilisé,ainsi on présentera le problème de cet algorithme connu (**i.e.** Le problème de transfert de clé), et comment le résoudre avec un algorithme asymétrique qu'on a choisi le **RSA**.Enfin on présentera notre application et les différentes interfaces développés, et l'environnement de développement, en dernier un exemple d'application.

2. Langage JAVA : [33] [32]

On peut faire remonter la naissance de Java à 1991. À cette époque, des ingénieurs de chez SUN ont cherché à concevoir un langage applicable à de petits appareils électriques (on parle de code embarqué). Pour ce faire, ils se sont fondés sur une syntaxe très proche de celle de C++, en reprenant le concept de machine virtuelle déjà exploité auparavant par le Pascal UCSD. **[33]**

Java est à la fois un langage de programmation et un environnement d'exécution. Le langage Java a la particularité principale que les logiciels écrits avec ce dernier sont très facilement portables sur plusieurs systèmes d'exploitation tels qu'**Unix**, Microsoft **Windows**, **Mac OS**où Linux avec peu ou pas de modifications... C'est la plate-forme qui garantit la portabilité des applications développées en Java**. [32]**

2.1. Java et la programmation orientée objet : [32]

En programmation structurée, un programme est formé de la réunion de différentes procédures et de différentes structures de données généralement indépendantes de ces procédures.

En **P.O.O.,** un programme met en œuvre différents objets. Chaque objet associe des données et des méthodes agissant exclusivement sur les données de l'objet. Notez que le vocabulaireévolue quelque peu : on parlera de méthodes plutôt que de procédures ; en revanche, onpourra utiliser indifféremment les mots donnésou le mot champ.

3. Advanced Encryption Standard :

3.1. Vue d'ensemble :

Advanced Encryption Standard (AES) est un algorithme de cryptographie à clé symétrique et il utilise un chiffrement par blocs itéré avec une taille de bloc fixe de 128 bits et une longueur de clé qui peut être de 128, 192 ou 256 bits. Les différentes transformations fonctionnent sur le résultat

intermédiaires, appelé état. L'état est un tableau rectangulaire d'octets et depuis le taille de bloc est 128 bits, ce qui est 16 octets, la matrice rectangulaire est de dimensions 4x4. (Dans la version Rijndael avec la taille de bloc variable, la taille de rangée est fixée à quatre et le nombre de colonnes varie. Le nombre de colonnes est de la taille d'un bloc, divisé par 32 et **Nb** notée). La clé de chiffrement est de même décrite comme un tableau rectangulaire avec quatre rangées. Le nombre de colonnes de la clé de chiffrement, notée **Nk**, est égale à la longueur de la clé divisé par 32.

AES utilise un nombre variable de tours, qui sont fixe: Une clé de taille 128 à 10 tours. Une clé de taille 192 à 12 tours. Une clé de taille 256 à 14 tours ; Dans le cryptage chacun des tours fait quatre opérations **SubBytes**, **ShiftRows**, **MixColumns**, et **AddRoundKey**et pour le décryptage il utiliser **inverse** de ces fonctions. Toutes ces fonctions sont décrites dans le **chapitre III**et seront décrites en générale au-dessous.

3.2. Processus global algorithme AES :

1. AES n'utilise pas une structure de Feistel mais traite l'ensemble du bloc de données en parallèle au cours de chaque tour en utilisant des substitutions et permutation.

2. La clé qui est fourni en entrée est élargie en une matrice de quarante-quatre 32 bits mots. Quatre mots distincts (128 bits) servent de clé ronde pour chaque tour.

3. Quatre étapes différentes sont utilisées, l'une de permutation et trois de substitution:

- **SubBytes**: Utilise une table, dénommé S-box, d'effectuer un octet par substitution d'octets du bloc.
- **ShiftRows**: Une simple permutation qui est effectuée ligne par ligne.
- **MixColumns**: Une substitution qui modifie chaque octet dans une colonne comme une fonction de l'ensemble des octets de la colonne.
- **AddRom1dKey**: A XOR au niveau du bit simple du bloc courant avec une partie de la clé élargie.

4. La structure est assez simple. Pour le chiffrement et le déchiffrement, le chiffrement commence par un complément de scène ronde Key, suivie par neuf tours que chacun comprend les quatre étapes, suivie d'un dixième cycle de trois étapes.

5. Seule la ronde Ajouter scène clé fait usage de la clé. Pour cette raison, le procédé de chiffrement commence et se termine avec un **AddRoundKey** étape clé. Toute autre étape, appliquée au début ou à la fin, est réversible sans connaissance de la clé.

6. L'ajout d'étape Round clé en elle-même ne serait pas formidable. Les trois autres étapes se bousculent ainsi les bits, mais par eux-mêmes, ils ne fournissent aucune sécurité, car ils n'utilisent pas la clé.

7. Chaque étape est facilement réversible. Pour l'octet suppléant, **ShiftRows**, et **MixColumns** étapes, une fonction inverse est utilisée dans l'algorithme de déchiffrement. Pour l'ajoutde clé dans une ronde, l'inverse est réalisé par **XOR**, la même clé rond pour le bloc, en utilisant le résultat :

A **XOR** B **XOR**B = A.

8. Comme avec la plupart des algorithmes de chiffrement par blocs, l'algorithme de déchiffrement permet l'utilisation de la clé élargie dans l'ordre inverse. Cependant, l'algorithme de déchiffrement n'est pas identique à l'algorithme de chiffrement. Ceci est une conséquence du particulier de la structurede **l'AES**.

9.Une fois qu'il est établi que les quatre étapes sont réversibles, il est facile de vérifier que décryptage s'il récupère le texte en clair après l'avoir crypté.

10. La ronde finale de chiffrement et du déchiffrement se compose de seulement trois étapes.

Encore une fois, ceci est une conséquence de la structure particulière d'**AES** et est requis pour rendre le chiffrement réversible.

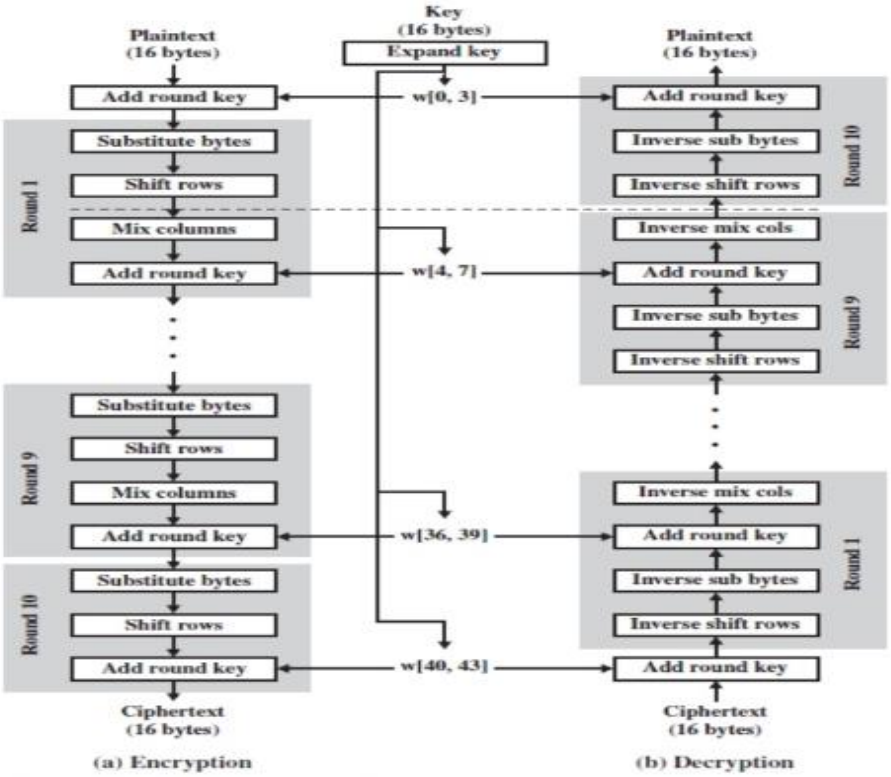


Figure 47 : illustration des structures d'AES

3.3. Détail de la Conception de l'algorithme AES:

3.3.1. Le chiffrement AES :

Voici l'algorithme AES est sous forme de lignes, en utilisant la syntaxe Java pour le pseudo code, et une grande partie de la notation standard AES:

Constants:int Nb=4; //mais il pourrait changer

int Nr= 10, 12, or 14; // ronds, Pour Nk=4, 6, or 8

Inputs: array in of 4*Nb bytes // plaintext entrée

array out of 4*Nb bytes // texte chiffré de sortie

array w of 4*Nb*(Nr+1) bytes // clef augmenter

Internal work array:state, 2-dim array of 4*Nb bytes, 4 ligneetNbcolonnes

Algorithm:

```
void Cipher (byte[] in, byte[] w) {
        byte[][] state= new byte[4][Nb];
        state= in;
        AddRoundKey (state, w, 0, Nb-1);
        for (int round =1; round < Nr; round++)
        {
            SubBytes(state);
            ShiftRows(state);
            MixColumns(state);
            AddRoundKey(state, w, Round*Nb,(round+1)*Nb-1);
        }

SubBytes(state);
            ShiftRows(state); // voir la Section 3 ci-dessous
            AddRoundKey(state, w, Nr*Nb,(Nr+1)*Nb-1); // Section 4
            out= state; // copie composant
            }
```

Il y aquartes opérations de cryptage AES utilisé dans l'extrait ci-dessus. Ils sont :

- **SubBytes**
- **ShiftRows**
- **MixColumns**
- **AddRoundKey**

3.3.1.1. SubBytes :

Beaucoup de différents algorithmes de chiffrement par blocs utilisent une substitution spéciale appelée "S-box". **L'AES** a aussi ces boîtes S, qu'il appelle la «SubBytes Transformation". Boîtes-S fourni une (réversible) transformation de segments de texte en clair pendant le chiffrement, avec

l'inverse lors du déchiffrement.Les SubBytes transformation modifient une seule entrée comme le montre ce code.

Le pseudo-code Java pour cette partie est maintenant très simple, en utilisant le tableau S-box déjà vu :

```
void SubBytes(byte[][] state) {
for (int row =0; row <4; row++)
for (int col=0; col <Nb;col++)
state[row][col]= Sbox[state[row][col]];
}
```

3.3.1.2. ShiftRows :

L'action de**ShiftRows** est particulièrement simple, il suffit de la scène circulaire gauche des décalages de lignes l, 2, et 3, selon des quantités de l, 2, et 3 octets. Et la ligne‘0’ n'est pas modifiée.

Le code java utilisé :

```
voidShiftRows (bytestate [] []) {
byte[]t= new byte[4];
for (int r=1; r<4;r++) {
for (int c=0; c<Nb; c++)
t[c]=state[r][(c+r)%Nb];
for (int c=0; c<Nb; c++)
state[r][c]=t[c];
}
}
```

3.3.1.3. MixColumns :

L'action de colonnes de mélange fonctionne sur les colonnes de la matrice de l'État, mais il est beaucoup plus compliqué que l'action des colonnes de décalage. Comme décrit dans **l’AES**cahier de charges, il traite chaque colonne comme un polynôme à quatre termes à coefficients. Tout cela est similaire à la description de la matière elle-même, à l'exception d'un supplément couche de complexité. Ces polynômes sont ajoutés et multiplié en utilisant simplement le fonctionnement de GF (2^8) sur les coefficients, à l'exception que le résultat d'une multiplication, qui est un polynôme de degré jusqu'à 6, doit être réduite en divisant par le polynôme $\mathbf{x^4 + l}$ et prendre le reste. Chacune des colonnes est multiplié parle polynôme fixe:

$$\mathbf{A(x) = (03)\ x^3 + (0l)\ x^2 + (01)\ x + (02)}$$

Où (03) représente l’élément de champ 0x03. Cela semble horrible, mais des manipulations mathématiques peut réduire tout à l'algorithme simple suivant, où la multiplication dans le domaine

est représenté ci-dessous par #. Le principal changement nécessaire pour convertir en Java pour remplacer # se fait par un appel à **FFMul().**et voici un exemple de code **JAVA** :

```
voidMIXCOLUMNS (byte [] []) {
byte[]sp=new byte[4];
for (int c=0; c<4; c++) {
sp[0]=(0x02#s[0][c])^(0x03#s[1][c])^s[2][c]^s[3][c];
sp[1]=s[0][c]^(0x02#s[1][c])^(0x03#s[2][c])^s[3][c];
sp[2]=s[0][c]^s[1][c]^(0x02#s[2][c])^(0x03#s[3][c]);
sp[3]=(0x03#s[0][c])^s[1][c]^s[2][c]^(0x02#s[3][c]);
for (int i=0; i<4; i++)
s[i][c]=sp[i];
}
}
```

3.3.1.4. AddRoundKey :

Comme décrit précédemment, des parties de la clé élargie w sont additionné avec un **XOR** sur la matrice d'état 'Nr + 1 fois' (une fois pour chaque tour ainsi une fois de plus). Il y a 4 * Nb octets d'état, et puisque chaque octet de la clé élargie est utilisée exactement une fois, la taille de la clé élargie de 4 * Nb * (Nr + 1) octets au juste. La clé élargie est utilisé, octet par octet, à partir du plus bas vers le plus haut indice, de sorte qu'il n'est pas nécessaire de compter les octets tels qu'ils sont utilisé à partir de w.

Ce code suppose que la clé a déjà été développée dans le tableau w, et il suppose un compteur **wCount** globale initialisée à **'0'** ; La fonction **AddRoundKey** utilise jusqu'à **4*Nb**= 16 octets de la clé élargi à chaque fois qu'il est appelé :

```
void AddRoundKey(byte[][]state){
for (int c=0; c<Nb; c++)
for (int r=0; r<4; r++)
state[r][c]= state[r][c]^w[wCount++];
}
```

3.3.2. Le déchiffrement AES :

Dans le déchiffrement les différentes étapes doivent être effectuées dans l'ordre inverse. Ceux-ci sont disposés en rondelles comme le cryptage, mais les fonctions de chaque tour sont dans un ordre légèrement différent de l'ordre utilisé dans cryptage. Le pseudo-code Java pour le chiffrement inverseest :

Constants:int Nb=4; //mais il pourrait changer

int Nr=10, 12, or 14; // ronds, pour Nk=4, 6, ou 8 ;

Inputs: array in of 4*Nb bytes // texte chiffré d'entrée

array out of 4*Nb bytes // plaintext de sortie

array w of 4*Nb*(Nr+1) bytes // clef augmentée

Internal work array: state, 2-dim array of 4*Nb bytes,

Algorithm:

```
voidInvCipher (byte[] in, byte[] out, byte[]w) {
byte[][] state= new byte[4][Nb];
state= in;
AddRoundKey (state, w, Nr*Nb, (Nr+1)*Nb-1);
for (int round =Nr-1; round >=1; round--)
{
InvShiftRows(state);
InvSubBytes(state);
AddRoundKey(state, w, Round*Nb,(round+1)*Nb-1);
InvMixColumns(state);
}
InvShiftRows(state);
InvSubBytes(state);
AddRoundKey(state, w, 0,Nb-1);
out= state;
}
```

Il ya quatre opérations en AES décryptage. Ce sont exactement l'inverse des opérations de chiffrement :

- **InvSubBytes**
- **InvShiftRows**
- **InvAddRoundKey**
- **InvMixColumns**

3.3.2.1. InvSubBytes :

Le tableau de la transformation des SubBytes inverse pourrait être généré à l'aide de l'inverse de la formule qui est utilisée dans la section précédente. Le pseudo-code Java pour cette partie est maintenant très simple, en utilisant le tableau Inv-Sbox déjà vu :

```
void InvSubBytes(byte[][] state){
for (int row=0; row<4; row++)
for (int col=0; col<Nb; col++)
state[row][col]=InvSbox[state[row][col]];
}
```

3.3.2.2. InvShiftRows :

Ce fait exactement l'inverse de ShiftRows: faire un décalage circulaire gauche des lignes 1, 2 et 3, en quantités de l, 2, et 3 octets,mais toujours pas pour la ligne '0' ; Et le décalage se fait un peu défèrent comme vu dans le **chapitreIII.**

Le code Java :

```
void InvShiftRows(byte[][]state){
byte[]t=new byte[4];
for (int r=1; r<4; r++){
for (int c=0; c<Nb; c++)
t[(c+r)%Nb]=state[r][c];
for (int c=0; c<Nb; c++)
state[r][c]=t[c];
}}
```

3.3.2.3. InvAddRoundKey :

Depuis la spécification AES utilise une fonction paramétré**AddRoundKey(),** elle est aussi son inverse, à l'aide des paramètres dans l'ordre inverse.

3.3.2.4. InvMixColumns :

La fonction MixColumns() a été soigneusement construite de sorte qu'elle a un inverse. Il suffit de dire que la fonction de chaque colonne multipliée par l'inverse d'un polynôme a(x):

$$\mathbf{A^{-1}(x) = (0b)x^3 + (0d)x^2 + (09)x + (0e)}$$

La fonction résultante, après simplification, prend la forme suivante en pseudo-code Java.

Code JAVA :

```
Void InvMixColumns(byte[][]s){
byte[]sp=new byte[4];
for (int c=0; c<4; c++) {
sp[0]=(0x0e#s[0][c])^(0x0b#s[1][c])^(0x0d#s[2][c])^(0x09#s[3][c]);
sp[1]=(0x09#s[0][c])^(0x0e#s[1][c])^(0x0b#s[2][c])^(0x0d#s[3][c]);
sp[2]=(0x0d#s[0][c])^(0x09#s[1][c])^(0x0e#s[2][c])^(0x0b#s[3][c]);
sp[3]=(0x0b#s[0][c])^(0x0d#s[1][c])^(0x09#s[2][c])^(0x0e#s[3][c]);
```

```
for (int i=0; i<4; i++)
s[i][c]=sp[i];
}}
```

3.3.3. Fonctionnement d'extension de la clé :

Dans un chiffrement simple, on pourrait utiliser **XOR** sur la clé avec le texte en clair. Une telle étape est facilement inversée par une autre ou-exclusif de la même clé avec le texte chiffré. Dans le cas de l'**AES**, il ya un certain nombre de tours, chacun ayant besoin de sa propre clé, de sorte que la clé réelle est "tendu" et transformé pour donner des parties de clé pour chaque tour, c'est l'élargissement de la clé qui est le sujet de cet article. Cette section a été détaillée dans le **chapitre III (section 6.6.1.),** et voici le code java :

Constants:int Nb=4; //mais il pourrait changer

Inputs: intNk= 4, 6, or 8; // le nombre de mots dans la clef

array Key of 4*Nk bytes or Nk words // clef d'entrée

Output: array w of Nb*(Nr+1) words or 4*Nb*(Nr+1) bytes // clef augmentée

Algorithm:

```
void KeyExpansion(byte[] Key,word[] w, intNw){
int Nr=Nk+6;
w= new byte[4*Nb*(Nr+1)];
int temp;
int i=0;
while (i<Nk) {
w[i]= word(Key[4*i], Key[4*i+1], Key[4*i+2], Key[4*i+3]);
i++;
}
i=Nk;
while (i<Nb*(Nr+1)) {
temp=w[i-1];
if (i%Nk==0)
temp=SubWord(RotWord(temp))^Rcon[i/Nk];
else if (Nk>6 && (i%Nk)==4
temp=SubWord(temp);
w[i]=w[i-Nk]^temp;
i++;
}}
```

3.4. Problème du transfert de clé AES:

Avec l'algorithme AES on peut chiffrer des documents et les envoyés à travers le réseau soit local ou le réseau internet mais le seul problème apparue c'est la clé, ainsi que on utilise la même clé pour le chiffrement et le déchiffrement alors cette clé doit être transférer pour le destinataire soit par email ou soit par messages etc. Ce qui rend l'obtention de cette clé très facile pour les pirates ou les dentinaires non désirés, et avoir cette clé implique avoir le fichier envoyer en le décryptant donc dans ce cas l'utilisation de l'algorithme AES avère inefficace.

3.4.1. Comment résoudre ce problème de clé :

3.4.1.1. Vue d'ensemble :

Dans la section précédente on a étudié l'algorithme RSA ainsi son fonctionnement pour le chiffrement et le déchiffrement des fichiers .Contrairement à l'algorithme AES ou en générale les algorithmes symétrique,l'algorithme RSA utilise des clés déférentes une pour chiffrer(**la clé public**) et l'autre pour déchiffrer(**la clé privé**).ce qui rend que la clé public circule en toute liberté dans les réseaux sans créer le moindre problème pour l'efficacité de cet algorithme tant que ça sert juste pour le chiffrement,la figure suivante illustre le principe de ces clés :

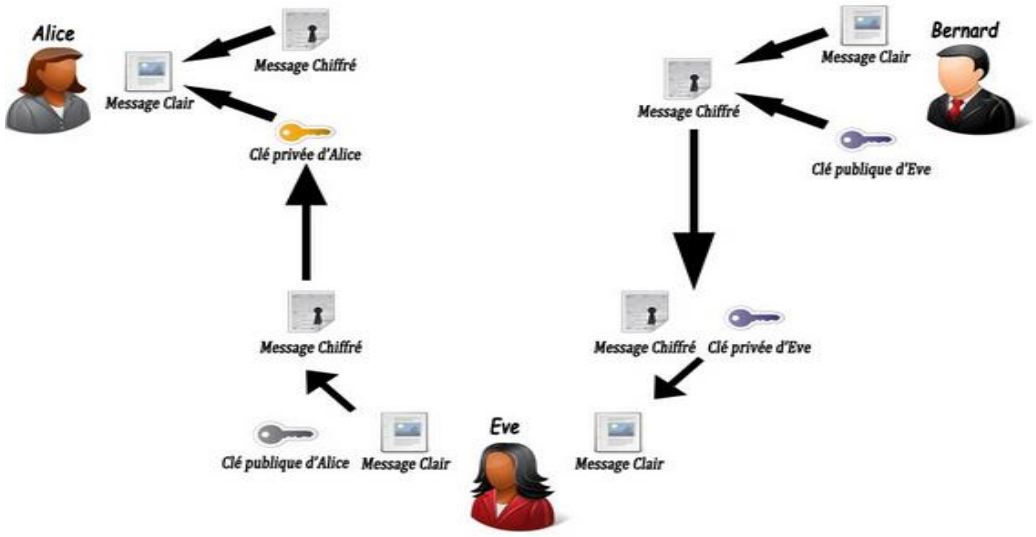


Figure 48 : Chiffrement et déchiffrement RSA

3.4.1.2. Principe pour résoudre le problème de clé AES :

Pour résoudre ce problème on doit intégrer l'algorithme RSA dans l'algorithme AES pour chiffrer le fichier à envoyer avec AES et en parallèle chiffrer la clé de AES en utilisant l'algorithme RSA,ainsi la clé de AES circule dans le réseau mais en texte crypter et donc nos données seront sécurisé plus et voici la procéduregénérale a implémenter et qui sera détaillé dans la prochain section :

- Le destinataire créer un couple de clé **RSA** (Clé **publique**, Clé **privé**) ;

- Le destinataire envoi la clé publique **RSA** au destinateur et garde sa clé privé secrète ;
- Le destinateur Crypte le fichier à envoyer avec AES et crypte la clé de chiffrement AES avec RSA ;
- Le destinateur Envoi le fichier crypter et la clé crypter sur le réseau pour le destinataire ;
- Le destinataire reçoit les deux fichiers ;
- Le destinataire Décrypte la clé AES avec l'algorithme RSA en utilisant sa clé privé ; et enfin le destinataire décrypte le fichier avec la clé AES décrypté.

3.5. Implémentation AES avec RSA:

3.5.1. Principe:

Pour démontrer l'algorithme **AES** on lui intégrant l'algorithme **RSA** pour chiffrer la clé, nous avons mis une application client-serveur. Les caractéristiques de l'application client-serveur sont :

- Le serveur doit générer deux clé pour RSA, une clé public**(e,n)** et une privé**(d,n),**on a vu le premier chapitre (algorithme RSA) comment créer ces nombres**(d,e,n)** à partir de deux nombres premiers **(p,q)** ,alors Le serveur envoi la clé public**(e,n)** au client par email ou message etc…
- client doit se connecter à l'aide de rompre Socket () en entrant l'adresse IP du serveur et un port particulier.
- client doit parcourir un fichier à envoyer et chiffrer l'aide de sa clé de choix. pour crypter il appelle la méthode **encrypt()** de la classe AES. Puis de l'envoyer au serveur en byte [] format.
- Client chiffre la clé de AES avec RSA à l'aide de sa clé public en utilisant la méthode **chiffrer(byte mess[])**,qui prend la clé en byte comme paramètre et envoyer cette clé crypter au serveur.
- serveur lit les octetsenvoyées et stocke dans deux fichiersspécifiques un pour le fichier et l'autre pour la clé AES crypté. Ensuite, décrypter la clé à l'aide de la méthode **dechiffrer(byte mess[])** et ensuite décrypter le fichier à l'aide de la méthode **decrypt()** de la classe AES.
- Et tout ça se fait à l'aide des deux interfaces **Client** et **Serveur**.

3.5.2. Les méthodes principales utilisées :

Pour l'implémentation de l'application on a utilisé beaucoup de méthodes mais dans cette partie on va citer et définir quelques-unes qu'on a jugées principale et on va voir les classes créer en java :

❖ **Classe AES.java :**

Cette classe contient toutes les méthodes utilisé soit pour le chiffrement AES ou le déchiffrement et voici les méthodes principales de cette classe et celles qui seront appeler par la suite de la classe Client et Serveur :

La méthode**encrypt(byte[] in,byte[]key)** utilise les fonctions suivantes pour générer texte chiffré :

- Méthode **generateSubkeys(byte[] key)** utilisé pour l'expansion clé.
- Méthode **encryptBloc(byte[] bloc)** utilisé pour le cryptage de bloc de données.
- **encryptBloc(byte[] bloc)**utilise à son tour les méthodes **SubBytes()**, **ShiftRows()**, **MixColumns()**, et **AddRoundKey()**. Sauf pour la dernière round on utilise pas **MixColumns().**
- La sortie de l'opération globale est un texte chiffré dans le format byte [].

La méthode**decrypt (byte[] in,byte[]key)** utilise les fonctions suivantes pour générer texte brut d'origine :

- Méthode **generateSubkeys(byte[] key)** utilisé pour l'expansion clé.
- Méthode **decryptBloc(byte[] bloc)** utilisé pour décrypter le bloc de données.
- Méthode **decryptBloc(byte [] bloc)** à son tour utilise les méthodes **InvSubBytes(), InvShiftRows()**, **InvAddRoundKey()** et **InvMixColumns()**. Et pour la dernière ronde, on utilise pas **InvMixColumns()**.

➢ La sortie de l'opération globale est texte clair dans le format de byte [].

❖ **Classe Client.java :**

Contient la méthodeprincipale**main()** alors elle sera parmi celle qui vont exécuter et la classe Client contient la fenêtre client et une méthode**chiffrer(byte mess[])**utiliser pour chiffrer la clé AES avec RSA après avoir donnéles deux nombres de la clé public**(e,n).**

✓ **Principe de la méthode chiffrer(byte mess[]) :**

❖ **Classe Serveur.java :**

Contient la méthode principale **main()** alors elle sera parmi celle qui vont exécuter et la classe Serveur contient la fenêtre Serveur et une méthode **dechiffrer(byte mess[])** utiliser pour déchiffrer la clé AES avec RSA après avoir donné les deux nombres de la clé privé**(d,n).**

✓ **Principe de la méthode dechiffrer(byte mess[]) :**

4. Réalisation :

4.1. Créer une application client/serveur en java :

Comme notre application se base sur le modèleClient/serveuroù un client envoi des données au serveur et vis-versa, l'architecture client/serveur définit dans le premier chapitre qui sert àtransférer ces donnéesà travers le réseau. Dans la programmation java pour créer une application Client/serveur on aura besoin d'utilisé les sockets qu'on a utilisé dans notre application Client/serveur et qui seront définitau-dessous.

4.1.1. Les sockets : [33]

Elles utilisent la notion de flux puisqu'elle désigne n'importe quel "canal" susceptible de transmettre de l'information sous forme d'une suite d'octets. Notamment, cette notion s'applique aux connexions TCP/IP entre ordinateurs utilisant le protocole Telnet. Dans ce cas, un des ordinateurs est considéré comme serveur et le service offert est caractérisé par :

• l'adresse IP de l'ordinateur, par exemple : 127.0.0.1;

• le numéro de port sur lequel on a choisi d'ouvrir le service ex : 8008 ;

La figure suivante illustre une connexion avec socket java entre un serveur et un client :

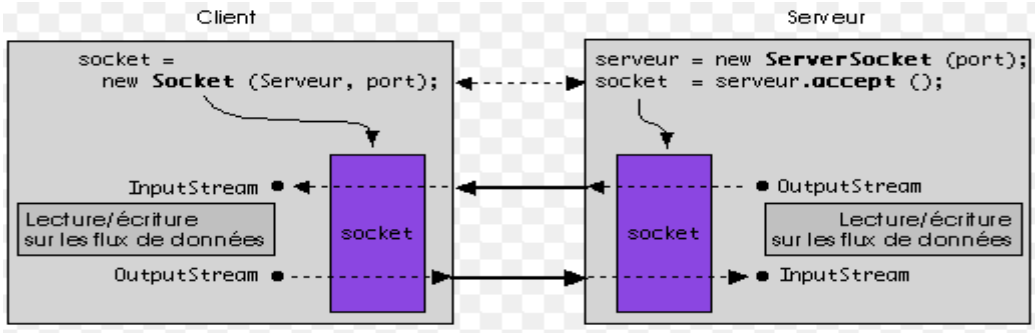


Figure 49 : Connexion par sockets

4.2. Présentation l'environnement de travail (matériels et logiciels) :

- **Environnement logiciels :**

Le langage JAVA cité dans la section précédente.

- **Environnement matériel :**

L'application client/serveur que nous avons réalisée a été développée sur un micro-ordinateur Fujitsu qui possède les caractéristiques suivantes :

- ✓ Fabricant FUJITSU SIEMENS
- ✓ Modèle ESPRIMO Mobile V6535
- ✓ Quantité totale de mémoire système Mémoire vive 2,00 Go
- ✓ Type du système Système d'exploitation 32 bits
- ✓ Nombre de cœurs de processeur 2
- ✓ Compatible 64 bits Oui
- ✓ Processeur Intel(R) Pentium(R) Dual CPU T3400 @ 2.16GHz

4.3. Environnement de développement (éclipse) :

Eclipse est un projet, décliné et organisé en un ensemble de sous-projets de développements logiciels, de laFondation Eclipse visant à développer un environnement de production de logiciels libre qui soit extensible, universel et polyvalent, en s'appuyant principalement sur Java.

Son objectif est de produire et fournir des outils pour la réalisation de logiciels, englobant les activités de programmation (notamment environnement de développement intégré et Framework).

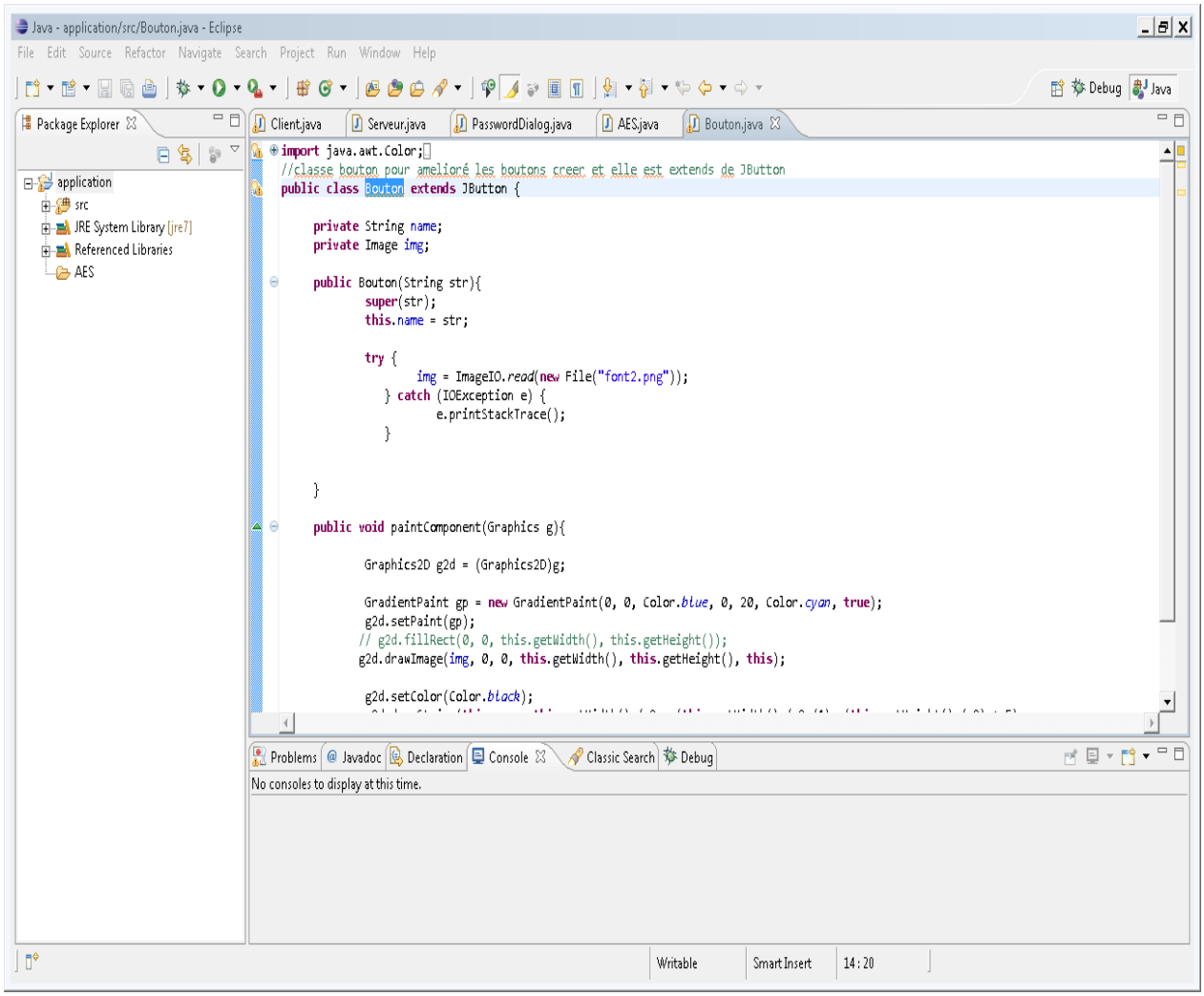


Figure 50 : Interface éclipse

4.4. Présentation denotre logiciel et lesdifférentes interfaces :

➢ Interface Client :

L'interface **Client**qu'on utilise pour le chiffrement de fichiers avec l'algorithme AES bien sur amélioré en termesdesécurité avec l'algorithme RSA.

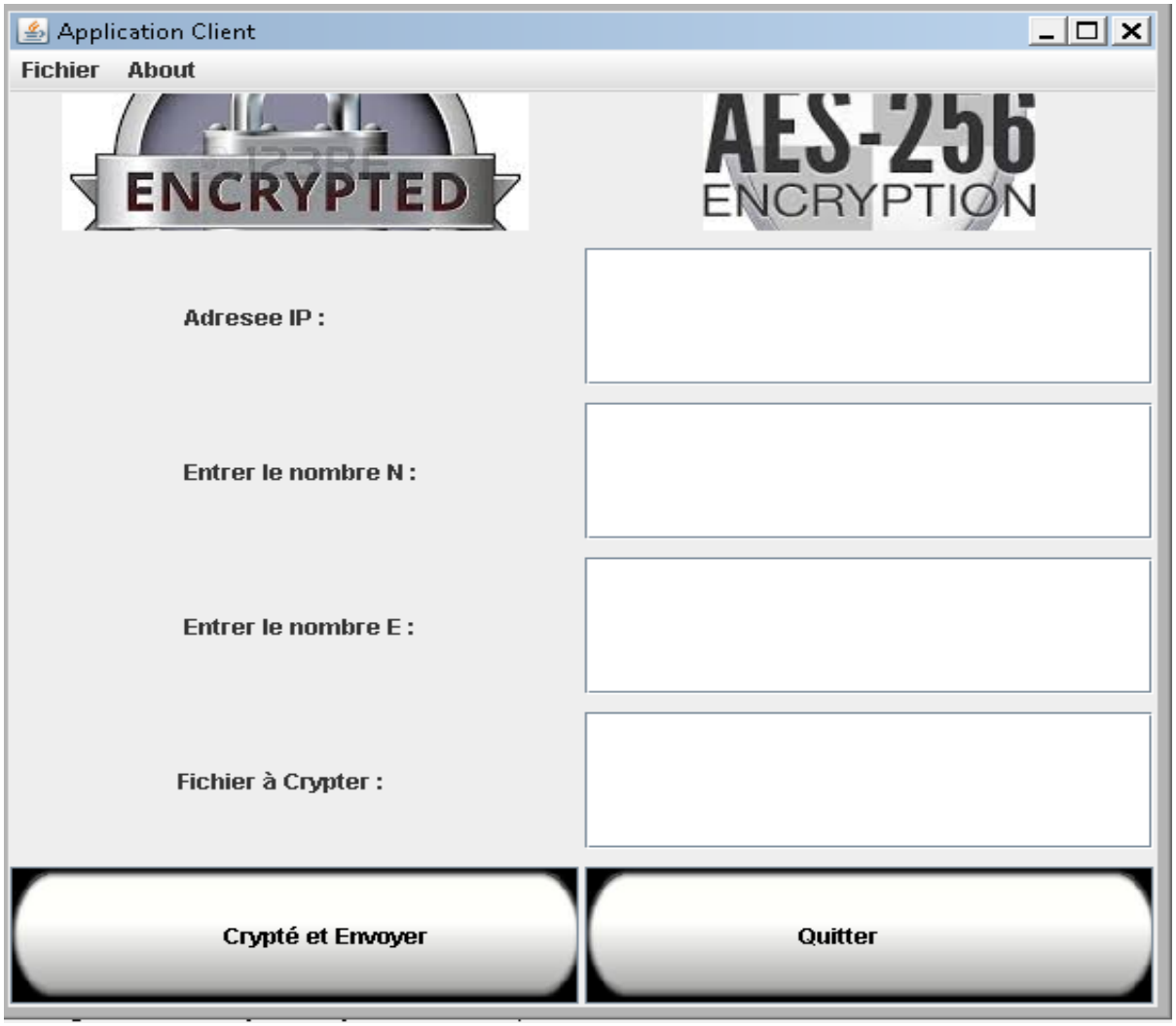


Figure 51 : Interface Client

- ✓ **Adresse IP :** pour enter l'adresse IP du serveur ;
- ✓ **Enter le nombre N :** N de la clé public ;
- ✓ **Entrer le nombre :** E de la clé public ;
- ✓ **Fichier à Crypter :** après avoir choie le fichier en cliquant sur le bouton le chemin du fichier s'affiche automatiquement dans ce champ ;
- ✓ Deux boutons pour crypter et quitter l'application.

Figure 52 : Boite de dialogue de la clé

- **Interface Serveur :**

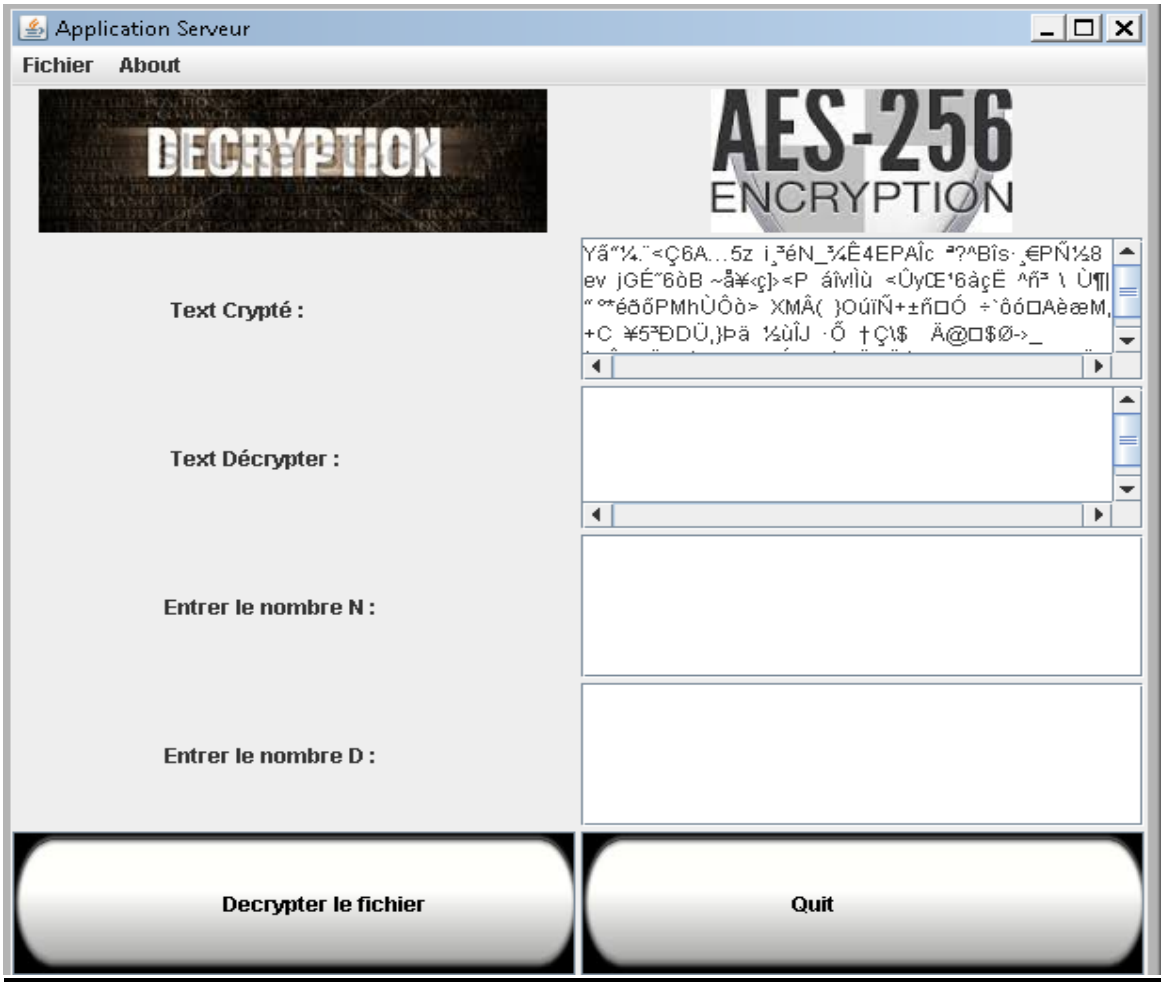
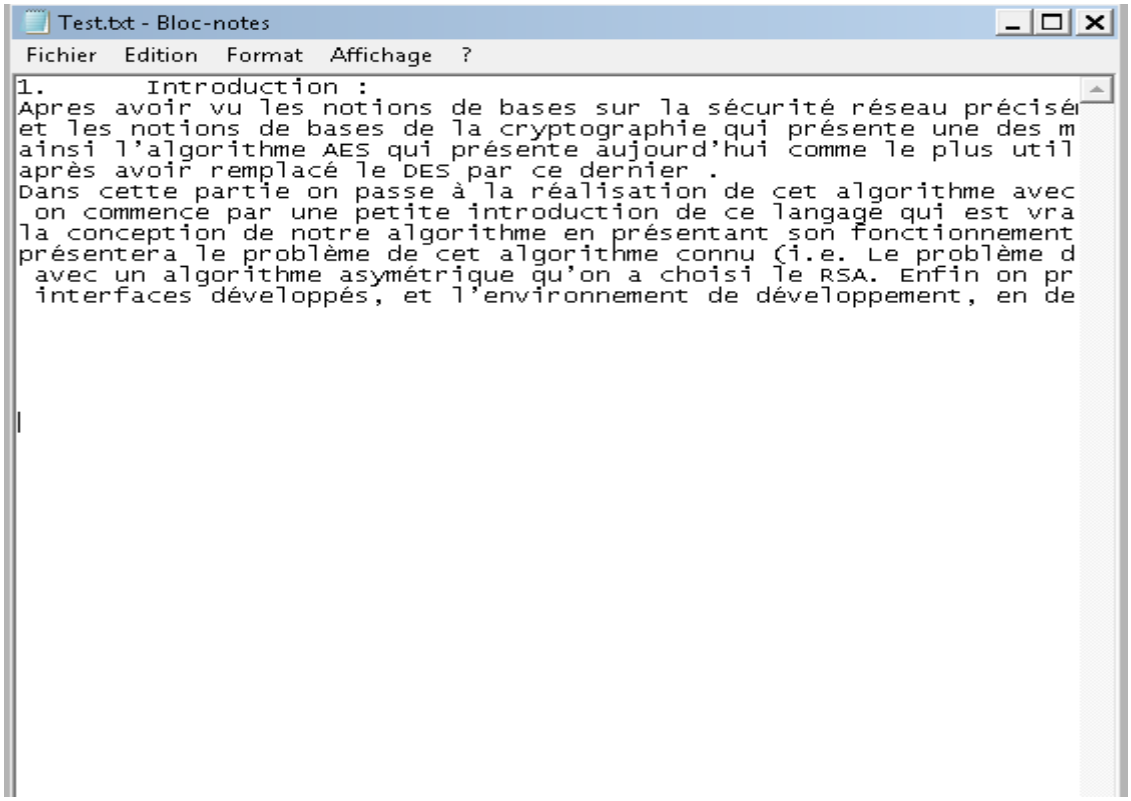


Figure 53 : Interface Serveur

- ✓ **Text Crypté :**Affiche le texte chiffrer ;
- ✓ **Text décrypter :** après le décryptage, affiche le texte clair ;
- ✓ **Entrer le nombre N :** Pour enter le nombre N de la clé privé ;
- ✓ **Enter le nombre D :** Pour entrer le nombre D de la clé privé ;
- ✓ **Deux boutons pour décrypter le fichier et un pour quitter.**

4.5. Exemple d'utilisation :

Afin d'évaluer la performance de notre logiciel, nous allons faire un test d'un fichier nommée « **Test.txt** » et en le cryptant puis l'envoyer au serveur pour le décrypter :

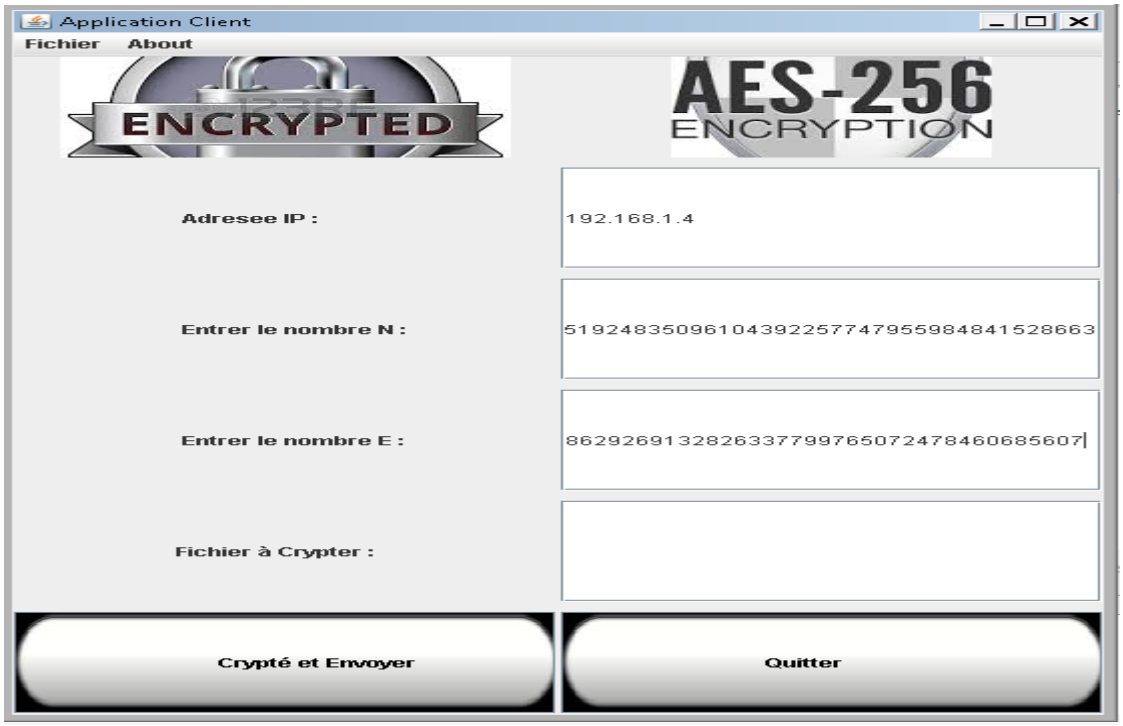


Le fichier à crypté

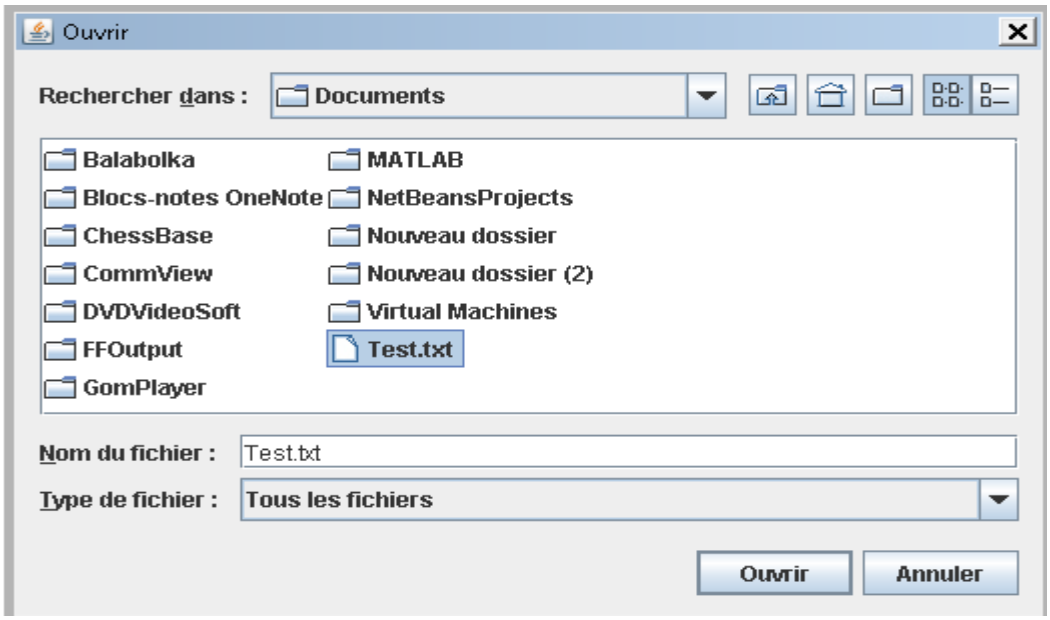
- **<u>Démarrer la fenêtre Client :</u>** on remplissant les champs
- **IP=192.168.1.4.**
- **la clé publique**

e=86292691328263377997650724784606856 07

n=151924835096104392257747955984841528663

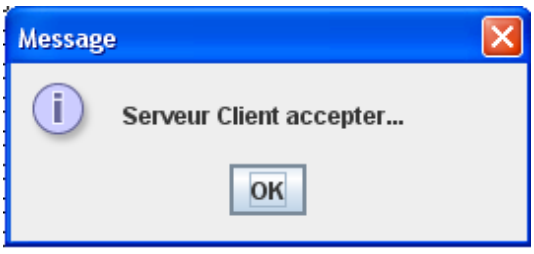


- Choisir le fichier **Test.txt**

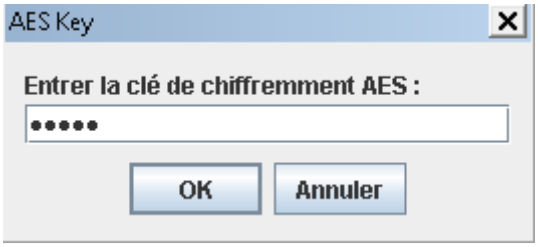


- Apres avoir cliqué sur le bouton Crypter le client demande au serveur d'ouvrir une connexion aprèsle démarrage du serveur.

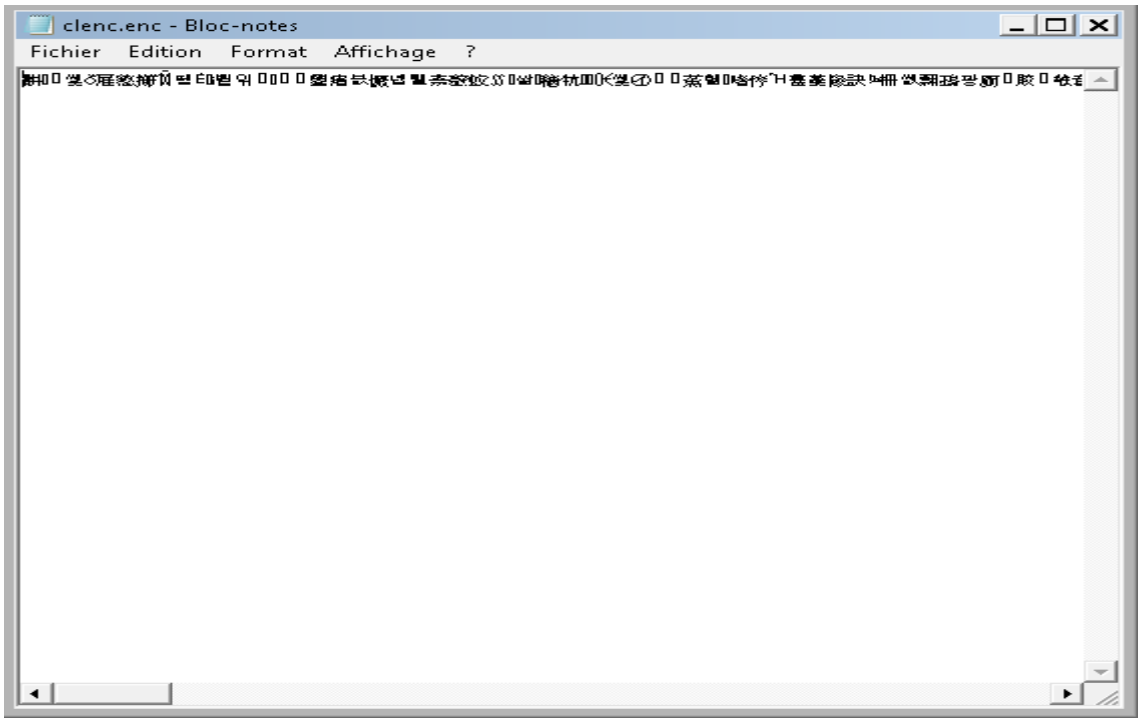
- Le serveur accepte le client.



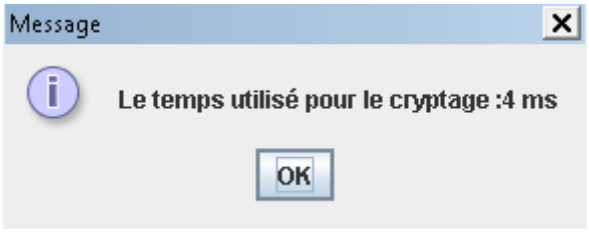
- Saisir la clé de chiffrement **AES**



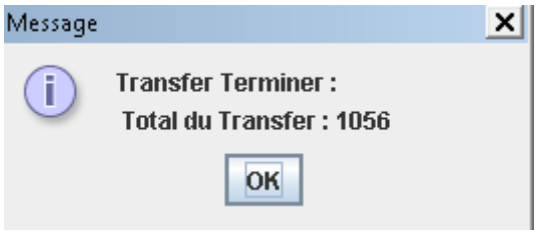
- Voila le fichier crypter qui se trouve dans la racine**(dossier AES).**



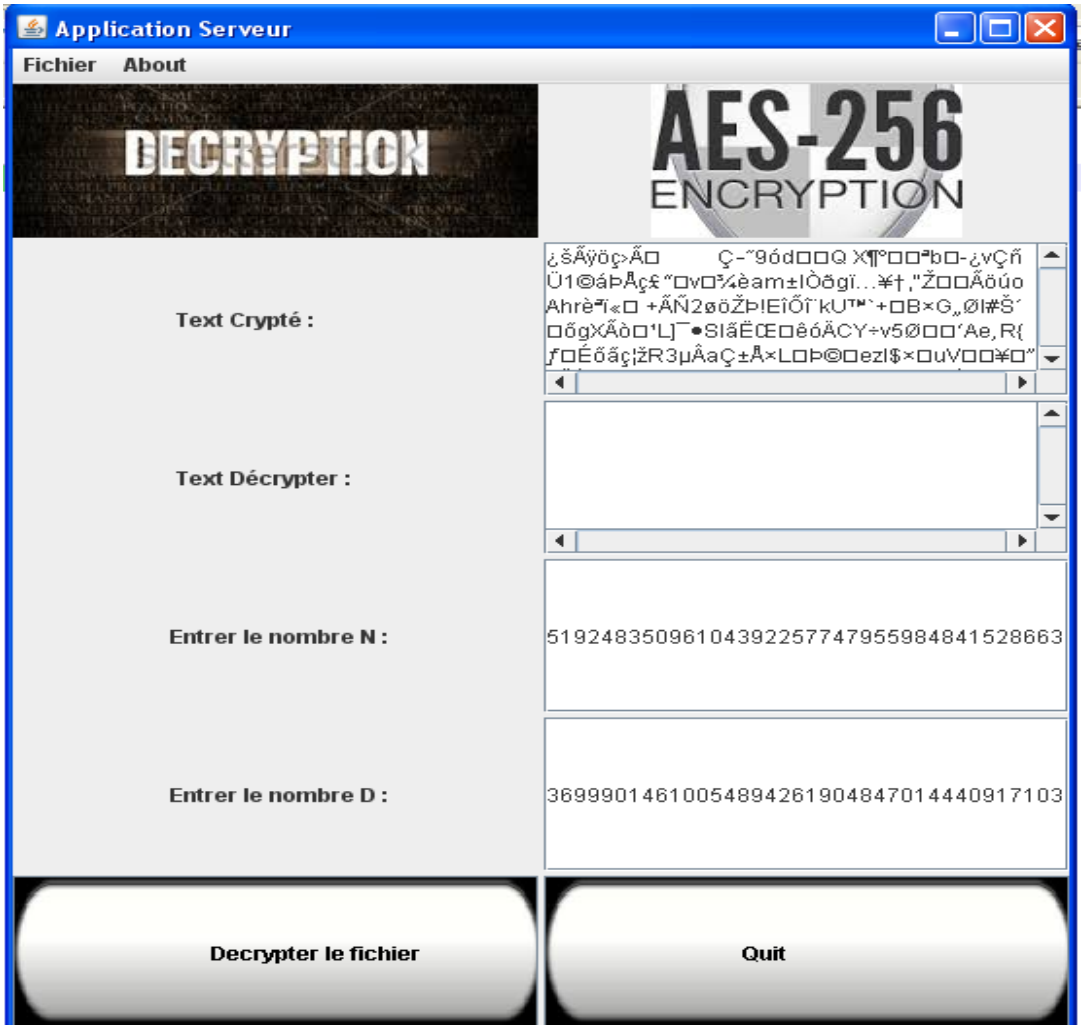
- Calcule du temps de cryptage on récupérant le temps système au démarrage du cryptage et après le cryptage et faire une soustraction.



- Le client transfert le fichier **Test.txt** et le fichier de la clé crypter avec **RSA** au serveur et le serveur mettre ces deux fichiers dans son dossier racine **AES**.



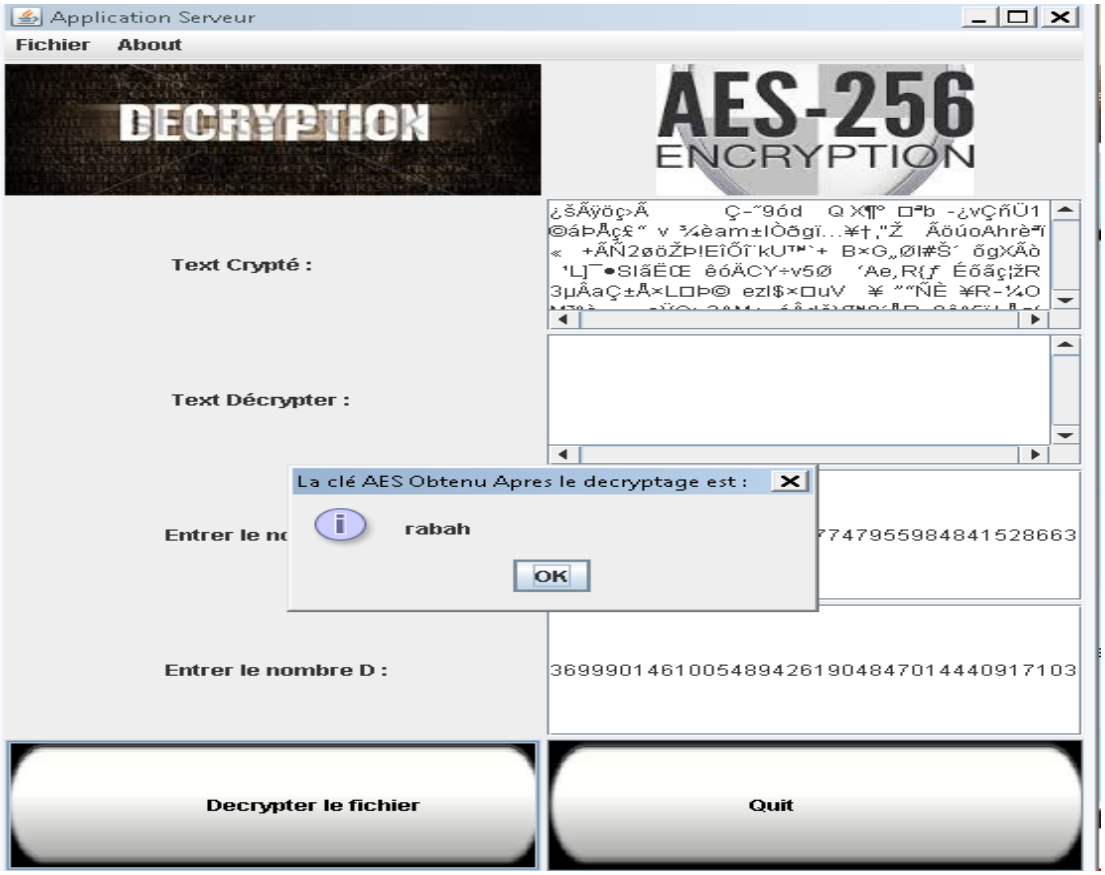
- La fenêtre serveur s'ouvre automatiquement et récupère le fichier crypté et la clé AES crypté avec **RSA**.



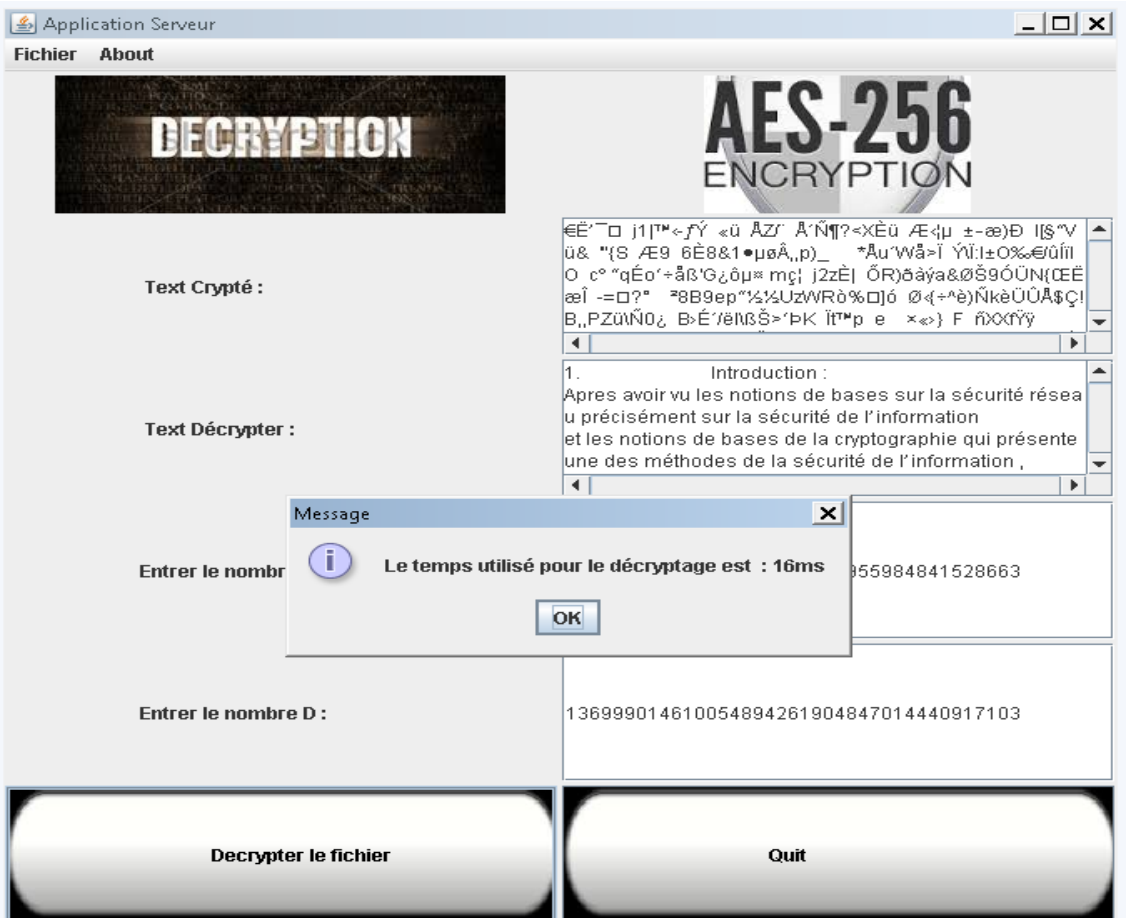
- Apres avoir rempli la clé privé **(d,n)** et on cliquant sur le bouton décrypter le fichier, le serveur décrypte la clé AES et après il va décrypter le fichier **Test.txt**.

  d = 13699901461005489426190484701444091710 3

  n = 15192483509610439225774795598484152866 3



- Et voici le texte brut d'origine en claire et le temps de décryptage :



Conclusion :

Dans ce chapitre nous avons analysé et étudié notre projet, au premier lieu, présenté les différents outils et langage que nous avons utilisé pour implémenter notre application. Par la suite, nous avons présenté quelques interfaces de notre application, enfin un exemple de cryptage d'un fichier et le déroulement du chiffrement et déchiffrement entre le Client et Serveur.

Conclusion générale

Depuis longtemps les informations transmises par les réseaux inquiètes les gens, à cause de la sécurité de cette dernière alors petite à petite ils commencent à chercher une solution pour résoudre ce problème de sécurité ,beaucoup de moyen son développé et parmi ces moyen et le cryptage ,bref chiffrer le contenu de ces données ainsi que personne ne pourra les lire à par le destinataire désiré qui peut les déchiffrer en utilisant des codes et actuellement ce que on appelles des algorithmes de cryptages.

Ce travail avait pour but de créer une application Client/serveur pour sécuriser ces informations transmises à travers le réseau, et pour cela nous avons implémenté un cryptosystème AES en l'améliorant avec l'algorithme RSA pour plus de sécurité.

Notre projet a été partagé par plusieurs étapes et on peut distinguer deux étapes essentiels la partie théorique qui consiste à comprendre mieux le fonctionnement des réseaux et la base de la sa sécurité, et la base de la cryptographie ; et la deuxième partie c'est la réalisation où on a étudié et développé notre application en en utilisant les outils et le langage nécessaire à savoir l'outil de développement Eclipse et le langage JAVA.

. Enfin, la réalisation de ce travail nous a permis d'acquérir des connaissances très importantes et très utiles dans le domaine de développement des applications en JAVA et surtout des applications Client/serveur, et aussi d'améliorer nos connaissances en conception logicielle.

Perspectives de recherche :

- dans notre application les clés RSA (Public et Privé) sont utilisé d'une manière directe sans les générer alors on propose d'implémenter une partie qui génère ces clés ainsi de les envoyés au destinataire par email.
- ces deux clés on l'utilise en nombres ce qui avère un peu difficile, alors on propose d les convertir en chaine de caractère.

Bibliographie

[1] Cryptographie et Sécurité informatique, Renaud Dumont, Université de Liège, 2009 – 2010 ;

[2] Sécurité informatique, Laurent Bloch et Christophe Wolfhugel, Edition EYROLLES, 2007 ;

[3] Sécurité informatique basé sur un cryptosystéme cas AES RMSE, Mémoire Master recherche, UMMTO, FGEI, 2012/2013 ;

[4] Applications des algorithmes évolutionnistes à la cryptographie, OMARY FOUZIA, THÈSE DE DOCTORAT D'ETAT, UNIVERSITÉ MOHAMMED V – AGDAL, FACULTÉ DES SCIENCES Rabat, 2006 ;

[5] TECHNIQUES DE CRYPTOGRAPHIE, Jonathan BLANC et Adrien DE GEORGES, Licence Informatique, 2003/2004 ;

[6] Cryptographie et sécurité des systeme et réseaux, Touradj Ebrahimi, Franck Leprévost et Bertrand Warusfel, Edition LAVOISIER,2006. ;

[7] Rapport sur l'étude et l'implémentation de quelques algorithmes de chirement et de signature, Nasser Yassine et Ouyous Mina, Master Informatique et Télé communications, UNIVERSITÉMOHAMMED V-AGDAL, Faculté des Sciences RABAT, 2013-2014 ;

[8] Mémoire de synthèse soumis dans le cadre d'un probatoire en vue de l'acquisition d'un diplôme en Ingénierie et Intégration Informatique Systèmes d'Information réaliser par Jean-Philippe Gaulier ;

[10] THÈSE sous le thème Vésication de protocoles cryptographiques en présence de théories équationnelles de M. Pascal La fourcade. (année 2006). ;

[11] Cours de Cryptographie (version préliminaire 2005/2006) de Daniel Barsky (année 2006) ;

[12] http//www.bibmath.netcryptoindex.phpaction=affiche&quoi=moderneelgamal ;

[13] UMR 7030 - Université Paris 13 - Institut Galilée Cours " Sécrypt " Laurent Poinsot ;.

[14] Sécurité des Systèmes Informatiques Legond-Aubry Fabrice - 20/11/2005 ;

[15] www.meziamus.com/attaques.php.html

[16] httpdbprog.developpez.com/securite/ids/ids.pdf ;

[17] www.meziamus.com/man_in_the_middle.php.html

[18] www.meziamus.com/typologie_pirates.php.html

[19] www.meziamus.com/mot_de_passe.php.html

[20] http://www.awt.be/web/sec/index.aspx?page=sec,fr,fic,045,004;

[21] http://www-igm.univ-mlv.fr/~dr/XPOSE2007/cchamp01_VPN/;

[22] Announcing the ADVANCED ENCRYPTION STANDARD (AES) Novembre 2001;

[23]httpswww.securiteinfo.comcryptographieaes.shtml

[24]httpsvarrette.gforge.uni.ludownloadteachingcryptoenonce_Projet_AES_HTMLnode4.html

[25] Sécurité Réseaux UST Oran Mr LARBI Miloud Maître Assistant IT Oran ;

[26] ARCHITECTURE DES RESEAUX, 2° édition, BERTRAND PETIT, 2006 ;

[27] Les Réseaux - 5ème édition, GUY PUJOLLE, Edition EYROLLES, 2004;

[28] http://www.lirmm.fr/~wpuech/enseignement/licence_MI/L1/classifications.pdf;

[29] Tout sur les Réseaux et Internet, JEAN-FRANÇOIS PILLOU, 2006 ;

[30]http://perso.modulonet.fr/~placurie/Ressources/BTS1-ALSI/Chap-12-%20Le%20clientserveur.pdf ;

[31] Initiation à la Conception Web, JOËL SKLAR, Edition EYROLLES ,2005 ;