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

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

**Mémoire de fin d'études**

**Pour l'obtention du diplôme de Licence en Informatique**

*Thème*

# Implémentation d'algorithmes de Cryptographie

**Réalisé par :**

- *M.:* **Hacini Souleyman Boumedyen**
- *M. :* **Inal Mohamed Taha**

*Présenté le 09 Juin 2014 devant la commission d'examination composée de*

- M$^{me}$ *: Didi.F* *(Encadreur)*
- M$^{me}$ *: Labraoui Nabila* *(Examinateur)*
- *M. : Ziani Cherif Salim* *(Examinateur)*

Année universitaire : 2013-2014

# Sommaire

Introduction générale ..... 3
Chapitre I : Généralités sur la sécurité ..... 4
Introduction ..... 4
I. Mesure des risques ..... 4
II. Objectifs de la sécurité informatique ..... 4
III. Domaine d'application ..... 6
IV. Vocabulaire de base ..... 6
V. Historique ..... 7
1. Chiffrement de César (50 av. J-C) ..... 7
2. Chiffrement de Vigenère (1568) ..... 7
3. La machine Enigma ..... 8
4. Les principaux concepts cryptographiques ..... 9
Les différentes méthodes de cryptage ..... 9
a) La cryptographie à clé secrète (symétrique) ..... 10
I- D.E.S. - Data Encryption Standard ..... 10
II- A.E.S. - Advanced Encryption Standard ..... 12
III- Le chiffrement de Vernam ..... 14
IV- Les structures de Feistel ..... 15
V- Conclusion sur la cryptographie à clé secrète ..... 16
b) La cryptographie à clé public (asymétrique) ..... 17
Concept ..... 17
I. Le protocole de Diffie et Hellman ..... 17
II. RSA (Rivest - Shamir – Adleman) ..... 18
La signature électronique ..... 18
VI- Conclusion sur la cryptographie à clé publique ..... 19
Conclusion ..... 20
Chapitre III : Implémentation de notre Application ..... 21
Introduction ..... 21
I- Introduction sur JAVA ..... 21
II- IDE ( Interface Developement Environement) ..... 23
III- Modélisation de l'Application ..... 24
Conclusion ..... 29
Conclusion générale ..... 30
Bibliographie ..... 31

# Introduction générale

A l'heure actuelle, les besoins en matière de sécurité sont grandissants, et la tendance n'est certainement pas à la baisse. Mais pourquoi ?

Tout d'abord parce que le matériel informatique est omniprésent. En effet les logiciels tendent à se simplifier au niveau de l'utilisation et permettent une prise en main rapide.

D'un autre côté, les entreprises, elles aussi informatisées, nécessitent un réseau sécurisé pour le transfert des données, que ce soit entre les machines de cette entreprise, ou avec des machines externes, distantes de plusieurs milliers de kilomètres.

On assiste également à une évolution constante des techniques, qu'il s'agisse des techniques visant à sécuriser l'échange de ces données ou des techniques mises au point pour contourner ces systèmes sécurisés. Pour toutes ces raisons, notre intérêt s'est naturellement porté sur ce domaine très vaste et assez compliqué, on a voulu y rentrer en commençant par l'aspect cryptographique. Sachant tous les autres aspects qui restent à explorer, et dont on va parler brièvement, ce modeste travail de fin d'études consiste donc à programmer quatre algorithmes de chiffrement dont César, Vigenere et Vigenere étendu et RSA puis un algorithme de hachage MD5. Le tout est présenté via une interface graphique très simple d'utilisation.

# Chapitre I : Généralités sur la sécurité

## Introduction

L'avènement de l'informatique puis de l'internet fut une des plus importantes découvertes de tout le siècle, elle a bouleversé toutes nos habitudes en termes de communications, de travail collaboratif distant, même dans notre vie quotidienne. Qui n'utilise pas le net cent fois par jour, pour une raison ou pour une autre. L'internet a grandement facilité nos tâches les plus banales et rapprocher toute la population terrestre. Cependant il y a encore plein de verrous à cette technologie, dont l'aspect sécuritaire. Bien que des bases très solides ont déjà été posées, comme on va brièvement le voir dans ce chapitre.

## I. Mesure des risques

Le risque en termes de sécurité est généralement caractérisé par l'équation suivante :

$$\text{Risque} = \frac{\text{Menace x Vulnérabilité}}{\text{Contre-mesure}}$$

La **menace « threat »** représente le type d'action susceptible de nuire dans l'absolu, tandis que la **vulnérabilité « vulnerability »,** appelée parfois faille ou brèche représente le niveau d'exposition face à la menace dans un contexte particulier. Enfin la **contre-mesure** est l'ensemble des actions mises en œuvre en prévention de la menace.

Les **contre-mesures** à mettre en œuvre ne sont pas uniquement des solutions techniques mais également des mesures de formation et de sensibilisation à l'intention des utilisateurs, ainsi qu'un ensemble de règles clairement définies.

Afin de pouvoir sécuriser un système, il est nécessaire d'identifier les menaces potentielles, et donc de connaître et de prévoir la façon de procéder de l'ennemi.

## II. Objectifs de la sécurité informatique

Le système d'information est généralement défini par l'ensemble des données et des ressources matérielles et logicielles de l'entreprise permettant de les stocker ou de les faire circuler. Le système d'information représente un patrimoine essentiel de l'entreprise, qu'il convient de protéger.

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

La sécurité informatique vise généralement cinq principaux objectifs :

- **L'intégrité**, c'est-à-dire garantir que les données sont bien celles que l'on croit être ;

- **La confidentialité**, consistant à assurer que seules les personnes autorisées aient accès aux ressources échangées ;
- **La disponibilité**, permettant de maintenir le bon fonctionnement du système d'information ;
- **La non répudiation**, permettant de garantir qu'une transaction ne peut être niée ;
- **L'authentification**, consistant à assurer que seules les personnes autorisées aient accès aux ressources.

Avec la popularité grandissante des réseaux, des échanges de données, et donc des transmissions entre individus, de nombreuses menaces émergèrent. Parmi celles-ci, on trouve diverses catégories :

1. Les menaces accidentelles
2. Les menaces intentionnelles :
    - passives
    - actives

Les menaces accidentelles ne supposent aucune préméditation. Dans cette catégorie, sont repris les bugs logiciels, les pannes matérielles, et autres défaillances "incontrôlables".

Les menaces intentionnelles quant à elles, reposent sur l'action d'un tiers désirant s'introduire et relever des informations. Dans le cas d'une attaque passive, l'intrus va tenter de dérober les informations par audit, ce qui rend sa détection relativement difficile. En effet, cet audit ne modifie pas les fichiers, ni n'altère les systèmes. Dans le cas d'une attaque active, la détection est facilitée, mais il peut être déjà trop tard lorsque celle-ci a lieu. Ici, l'intrus aura volontairement modifié les fichiers ou le système en place pour s'en emparer.

La **cryptographie** est l'un moyens de protections des informations et des données des particuliers et des entreprises et c'est sur ce sujet qu'on va se focaliser sur ce PFE

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

1) Le but d'un système cryptographique est de chiffrer un texte clair **P** en un cryptogramme **C** au moyen d'une clé **K**.
2) Ce message crypté est ensuite transmis à un destinataire qui doit pouvoir le déchiffrer à l'aide de la clé **K**

# III. Domaine d'application

**1-** Les cartes bancaires

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

**3-** Secret militaire

**4-** Communications numériques

**5-** Droits d'auteurs

# IV. Vocabulaire de base

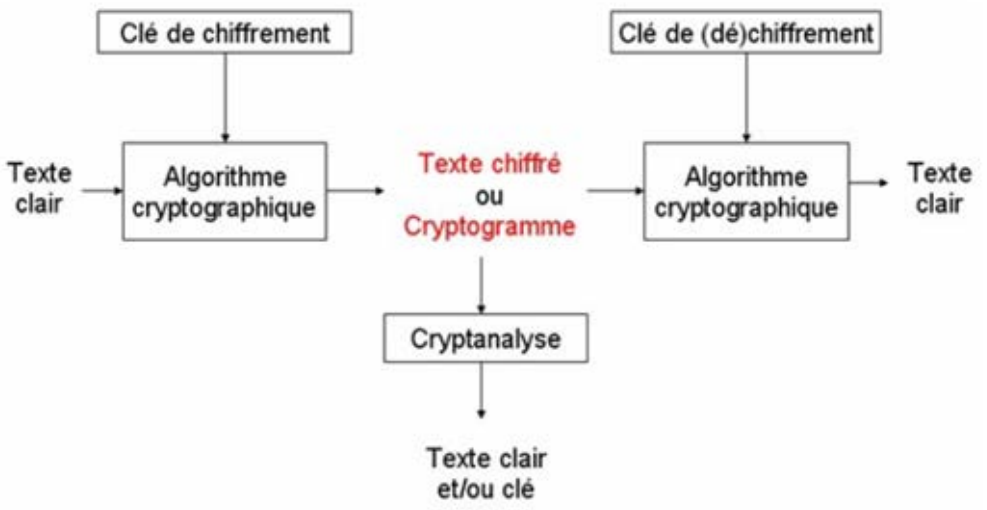



**Figure III.1 - Protocole de chiffrement**

- **Cryptologie :** Il s'agit d'une science mathématique comportant deux branches : la cryptographie et la cryptanalyse
- **Cryptographie :** La cryptographie est l'étude des méthodes donnant la possibilité d'envoyer des données de manière confidentielle sur un support donné.
- **Chiffrement :** Le chiffrement consiste à transformer une donnée (texte, message, ...) afin de la rendre incompréhensible par une personne autre que celui qui a créé le message et celui qui en est le destinataire. La fonction permettant de retrouver le texte clair à partir du texte chiffré porte le nom de déchiffrement.
- **Texte chiffré :** Appelé également cryptogramme, le texte chiffré est le résultat de l'application d'un chiffrement à un texte clair.

- **Clef :** Il s'agit du paramètre impliqué et autorisant des opérations de chiffrement et/ou déchiffrement. Dans le cas d'un algorithme symétrique, la clef est identique lors des deux opérations. Dans le cas d'algorithmes asymétriques, elle diffère pour les deux opérations.
- **Cryptanalyse :** Opposée à la cryptographie, elle a pour but de retrouver le texte clair à partir de textes chiffrés en déterminant les failles des algorithmes utilisés.
- **Crypto système :** Il est défini comme l'ensemble des clés possibles (espace de clés), des textes clairs et chiffrés possibles associés à un algorithme donné.

# V. Historique

## 1. Chiffrement de César (50 av. J-C)

**Principe**

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

Dans les formules ci-dessous, p est l'indice de la lettre de l'alphabet, k est le décalage.

Pour le chiffrement, on aura la formule : $\mathbf{C = E(p) = (p + k) \bmod 26}$

Pour le déchiffrement, il viendra : $\mathbf{p = D(C) = (C - k) \bmod 26}$

| CLAIR | A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -> décalage = 3 | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CODE | 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 |

**Figure IV.1 – Application du chiffre de César**

**Exemple :** d'après cette méthode, "VIVE LES MATHS" devient donc "YLYH OHV PDWKV" !

## 2. Chiffrement de Vigenère (1568)

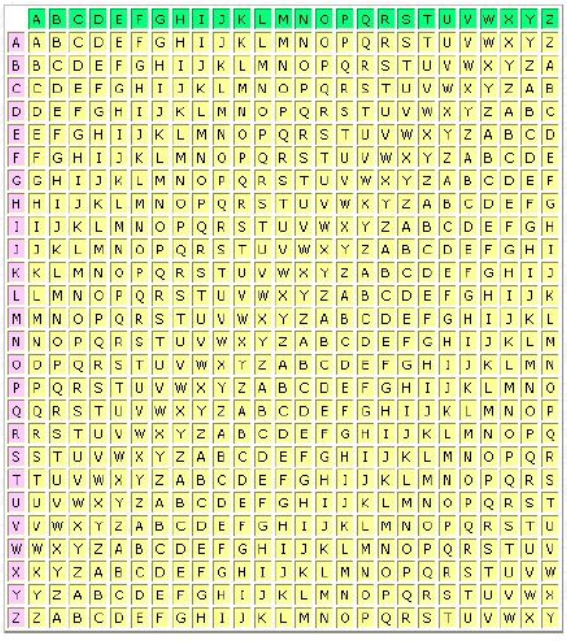
C’est une amélioration décisive du chiffre de César. Sa force réside dans l’utilisation non pas d’un, mais de 26 alphabets décalés pour chiffrer un message. On parle du carré de Vigenère. Ce chiffre utilise une clef qui définit le décalage pour chaque lettre du message (A : décalage de 0 cran, B : 1 cran, C : 2 crans, ..., Z : 25 crans).

**Exemple :** le chiffrement du texte "CHIFFRE DE VIGENERE" avec la clef "BACHELIER" (cette clef est éventuellement répétée plusieurs fois pour être aussi longue que le texte clair)

| Clair | C | H | I | F | F | R | E | D | E | V | I | G | E | N | E | R | E |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Clef | B | A | C | H | E | L | I | E | R | B | A | C | H | E | L | I | E |
| Décalage | 1 | 0 | 2 | 7 | 4 | 11 | 8 | 4 | 17 | 1 | 0 | 2 | 7 | 4 | 11 | 8 | 4 |
| Chiffré | D | H | K | M | J | C | M | H | V | W | I | I | L | R | P | Z | I |

**Figure IV.2 - Application du carré de Vigenère**

Pour utiliser le chiffrement de Vigenère, on a recours au Carré de Vigenère, illustré dans la figure suivante :




### 3. La machine Enigma

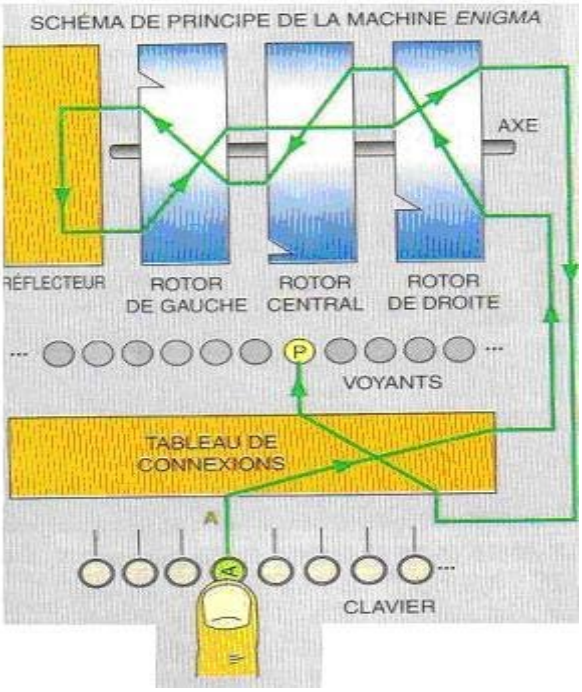
La machine allemande Enigma a joué un grand rôle pendant la guerre de l'Atlantique, et son décryptement par les Alliés leur a assuré bon nombre de victoires (notamment parce que les Allemands ne se doutaient pas que leurs messages étaient déchiffrés).

Enigma ressemble à une machine à écrire : on frappe le clair sur un clavier, et des petites lampes s'allument pour éclairer les lettres résultant du chiffrement.

Le principe de chiffrement qu'utilise Enigma est à la fois simple et astucieux. Simple, car il ne s'agit ni plus ni moins d'une substitution de lettres : par exemple, A devient Q, P devient N, etc. Et astucieux, parce que la substitution change d'une lettre à une autre : si la lettre A correspond à Q la première fois qu'on la saisit, elle pourrait correspondre à M, K, H, ou tout autre lettre différente de Q à la fois suivante (ce principe est possible grâce à un système de rotors).

De plus, un autre avantage non négligeable que possède Enigma est la réversibilité : si on tape le message clair, on obtient le message code, et avec le message codé, on obtient le message clair.

L'inconvénient majeur est que jamais la lettre A ne sera codée par A....




# 4. Les principaux concepts cryptographiques

### Les différentes méthodes de cryptage

Il existe des tonnes d'algorithmes de cryptage, Si l'on ne peut pas connaître tous les algorithmes de cryptage, il faut par contre savoir que l'on peut les classer en deux catégories : le cryptage symétrique et le cryptage asymétrique.

### a) La cryptographie à clé secrète (symétrique)

**Caractéristiques**

- Les clés sont identiques : $\mathbf{K_E = K_D = K}$
- 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 texte clair 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 aller jusque 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, on préfèrera l'échange manuel. Malheureusement, pour de grands systèmes, le nombre de clés peut devenir conséquent. C'est pourquoi on utilisera souvent des échanges sécurisés pour transmettre les clés. En effet, pour un système à N utilisateurs, il y aura N. (N − 1)/2 paires de clés.

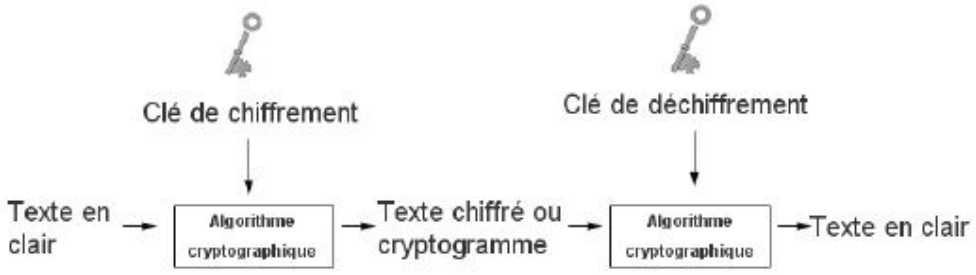



**Figure V.1 -Chiffrement symétrique**

**Principe de Kerckhoff**

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

En d'autres termes, aucun secret ne doit résider dans l'algorithme mais plutôt dans la clé. Sans celle-ci, il doit être impossible de retrouver le texte clair à partir du texte chiffré. Par contre, si on connaît K, le déchiffrement est immédiat.

### I- D.E.S. - Data Encryption Standard

## 1-Présentation

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, cet algorithme, devenu alors D.E.S., fut adopté au niveau fédéral le 23 novembre 1976.

## 2-Particularités

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 :

- il possède un haut niveau de sécurité,
- il est complètement spécifié et facile à comprendre,
- la sécurité est indépendante de l'algorithme lui-même,
- il est rendu disponible à tous, par le fait qu'il est public,
- il est adaptable à diverses applications (logicielles et matérielles),
- il est rapide et exportable,
- il repose sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement,
- il est facile à implémenter.

## 3-Algorithme de chiffrement

Le D.E.S. est un crypto système agissant par blocs. Cela signifie que D.E.S. ne chiffre pas les données à la volée quand les caractères arrivent, mais il découpe virtuellement le texte clair en blocs de 64 bits qu'il code séparément, puis qu'il concatène. Un bloc de 64 bits du texte clair entre par un coté de l'algorithme et un bloc de 64 bits de texte chiffré sort de l'autre coté. L'algorithme est assez simple puisqu'il ne combine en fait que des permutations et des substitutions.

C’est un algorithme de chiffrement à clef secrète. La clef sert donc à la fois à chiffrer et à déchiffrer le message. Cette clef a ici une longueur de 64 bits, c’est-à-dire 8 caractères, mais dont seulement 56 bits sont utilisés. On peut donc éventuellement imaginer un programme testant l’intégrité de la clef en exploitant ces bits inutilisés comme bits de contrôle de parité.

L’entière sécurité de l’algorithme repose sur les clefs puisque l’algorithme est parfaitement connu de tous. La clef de 64 bits est utilisée pour générer 16 autres clefs de 48 bits chacune qu’on utilisera lors de chacune des 16 itérations du D.E.S.. Ces clefs sont les mêmes quel que soit le bloc qu’on code dans un message.

Cet algorithme est relativement facile à réaliser matériellement et certaines puces chiffrent jusqu’à 1 Go de données par seconde. Pour les industriels, c’est un point important notamment face à des algorithmes asymétriques, plus lents, tels que l’algorithme R.S.A.

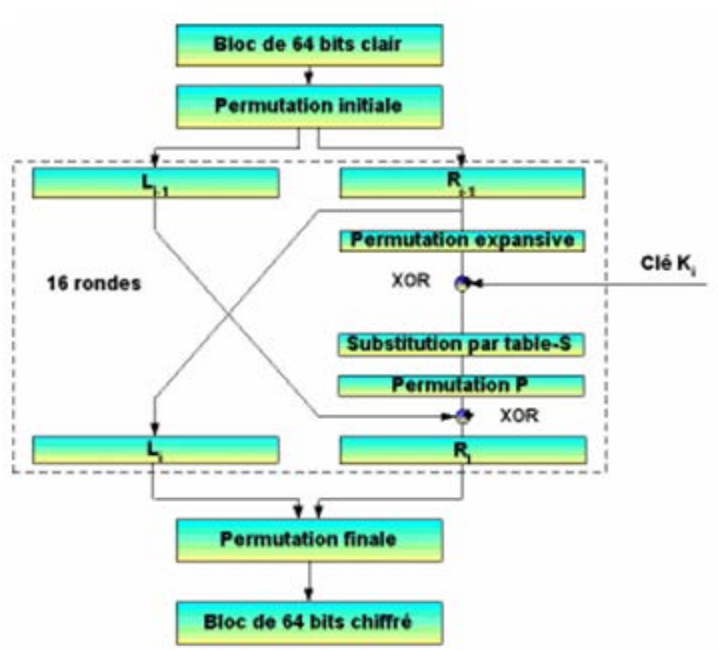



**Figure V.1.A - Algorithme principal du DES**

L’algorithme repose principalement sur 3 étapes, en plus de la gestion spécifique de la clé:

1. Permutation initiale
2. Calcul médian (16 fois) : application d’un algorithme complexe appliqué en fonction de la clé
3. Permutation finale

## II- A.E.S. - Advanced Encryption Standard

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, nom résultant de la contraction des noms de ses inventeurs : Rijmen et Deamen.

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.
- À chaque ronde, quatre transformations sont appliquées.
- 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.

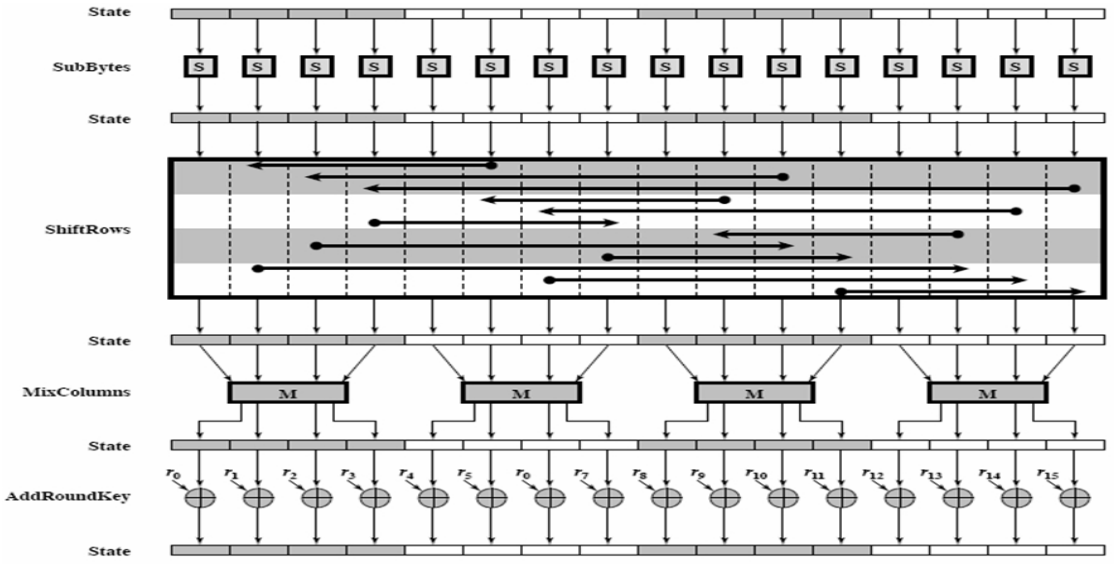



**Figure V.1.B- Schéma général de Rijndael (A.E.S)**

## III- Le chiffrement de Vernam

En 1917, Gilbert Vernam mit au point un algorithme de chiffrement -basé sur une clé secrète- parfaitement sûr, tellement sûr qu'il a longtemps protégé le fameux "téléphone rouge", qui reliait la Maison Blanche au Kremlin.

- L'algorithme de chiffrement et de déchiffrement est un simple XOR
- La longueur de la clé est égale à celle du message
- Une clé n'est utilisée qu'une seule fois
- La clé doit être la plus aléatoire possible
- C'est le crypto système le plus sûr !
- Mais il n'est pas très pratique

### 1-Utilisation

- La clé est de la taille du message à envoyer
- Les lettres de cette clé sont choisies de façon totalement aléatoire
- La clé ne doit servir qu'une seule et unique fois

### 2-Illustration du principe par un exemple

Alice veut transmettre à Bob un message M. A l'aide de la clé secrète K (convenue avec B), elle va crypter M pour arriver à sa version cryptée C. Ecrivons :

**C = K * M** où "*" est une loi de groupe. L'utilisation d'une loi de groupe se justifie car elle possède une propriété nécessaire : quels que soient deux nombres a et b, il existe un unique nombre x tel que : **a = b * x**

**Exemple** Soient M = 1000011 et K = 1101000,

Le message crypté C est donc : C = K XOR M = 1101000 XOR 1000011 = 0101011

### 3-Sécurité "inconditionnelle"

Si la clé choisie est soumise aux conditions citées plus haut, l'utilisation d'une loi de groupe garantit une sécurité dite inconditionnelle. En effet, si on admet qu'un cryptanalyste intercepte le message crypté C, il ne pourra rien en déduire, si ce n'est la taille du message en clair M. Il lui est impossible d'établir une corrélation entre M et C sans connaître K, car étant donné qu'on utilise pour crypter une loi de groupe, il n'existe pour M et C qu'un seul nombre K tel que **M = K * C !**

Même si l'ennemi possédait des ordinateurs ultra puissants, il ne pourrait jamais en trouver la solution. C'est dans ce cas-là que le mot "sécurité inconditionnelle" prend son sens : une puissance de calcul infinie ne décrypterait pas le message.

**IV-Les structures de Feistel**

Cette structure fut décrite en 1973 (par Feistel, employé chez IBM). La plupart des chiffrements de la fin du XXè siècle sont basés sur cette structure. Elle découle des réseaux S-P de Shannon. Il adapte la structure de Shannon afin de la rendre inversible ce qui permet de réutiliser le matériel de chiffrement pour déchiffrer un message. La seule modification s'opère dans la manière dont la clé est utilisée.

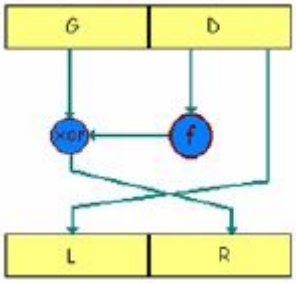



**Figure V.1.D (1) Structure de Feistel**

Dans une construction de Feistel, le bloc d'entrée d'un round est séparé en deux parties.

La fonction de chiffrement est appliquée sur la première partie du bloc et l'opération binaire OU-Exclusif (+) est appliquée sur la partie sortante de la fonction et la deuxième partie. Ensuite les deux parties sont permutées et le prochain round commence.

L'avantage est que la fonction de chiffrement et la fonction de déchiffrement sont identiques. Ainsi la fonction n'a pas à être inversible, c'est la structure qui l'est.

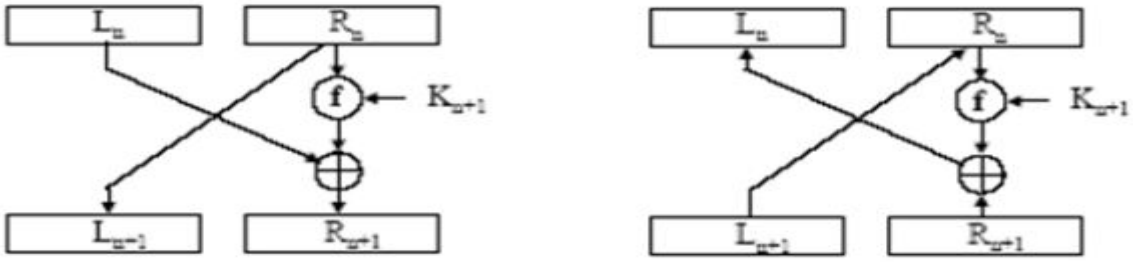



**Figure V.1 .D (2) -La structure est inversible**

**Exemple** A partir d'une table de correspondance, on peut déterminer le résultat du chiffrement d'un bloc après passage dans une structure de Feistel.

**Table de correspondance des fonctions**

| f1 | | f2 | |
|---|---|---|---|
| **Entrée** | **Sortie** | **Entrée** | **Sortie** |
| **00** | **01** | **00** | **11** |
| **01** | **11** | **11** | **00** |
| **10** | **10** | **10** | **00** |
| **11** | **01** | **11** | **01** |

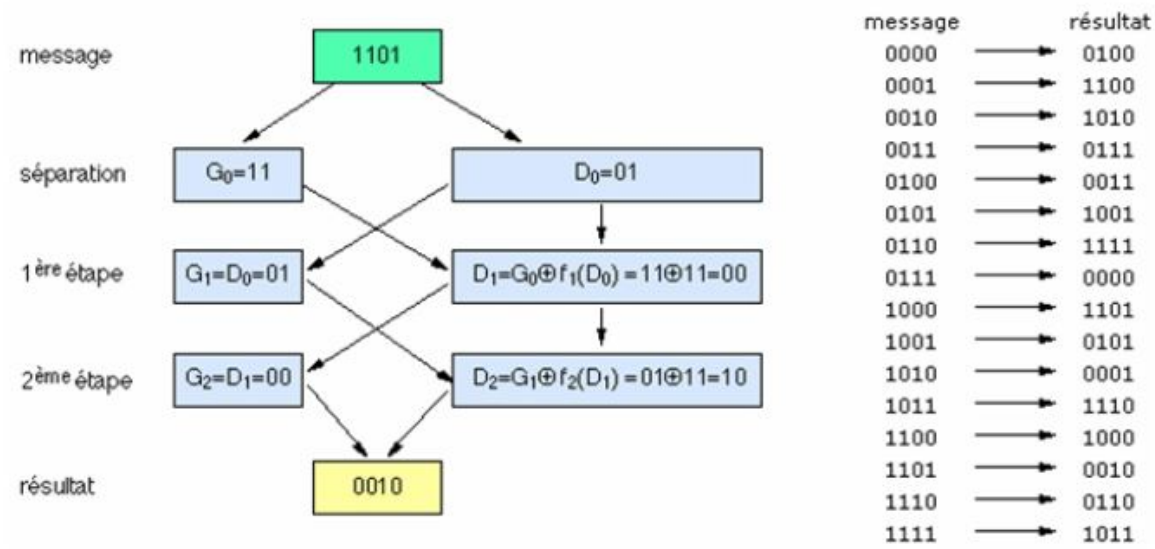



**Figure V.1.D (3) Exécution d'un schéma de Feistel**

## V- Conclusion sur la cryptographie à clé secrète

Malgré toutes ses évolutions et ses mises en œuvre, la cryptographie à clé secrète est toujours entravée par un défaut : le secret de sa clé (principe de Kerckhoffs). Bien qu'ayant pu au fil du temps réduire sa taille, les cryptographes ont toujours été confrontés au problème de la transmission de cette clé... Mais le progrès ne s'arrête jamais ! Si le problème est de conserver le secret de la clé, pourquoi ne pas le contourner... en inventant un système qui la rend publique ?

### b) La cryptographie à clé public (asymétrique)

**Concept**

Dans le cas des systèmes asymétriques, chaque personne possède 2 clés distinctes (une privée, une publique) avec impossibilité de déduire la clé privée à partir de la clé publique. De ce fait, il est possible de distribuer librement cette dernière.

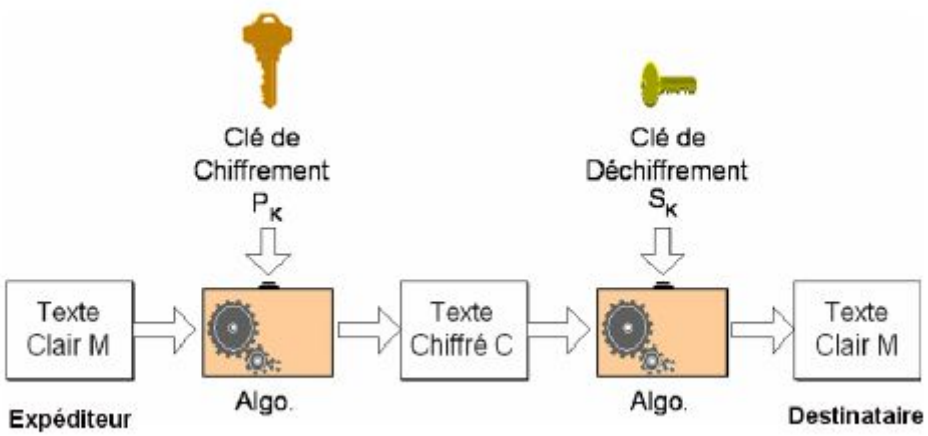



**Figure V.2 : Chiffrement à clé publique**

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

- Chiffrement/déchiffrement : cela fournit le secret.
- Signatures numériques : cela fournit l'authentification.
- Échange de clés (ou des clefs de session).

Quelques algorithmes conviennent pour tous les usages, d'autres sont spécifiques à un d'eux.

## I. Le protocole de Diffie et Hellman

Parallèlement à leur principe de cryptographie à clé publique, Diffie et Hellman ont proposé un protocole d'échanges de clés totalement sécurisé, basé sur des fonctions difficiles à inverser.

1- Alice et Bob se mettent d'accord publiquement sur un très grand nombre premier "p" et sur un nombre "n" inférieur à "p".

2- Alice engendre une clé secrète "a" et Bob une clé secrète "b".

3- Alice calcule l'élément public $k_a$ et Bob l'élément public $k_b$: $\mathbf{k_a = n^a \bmod p}$

$\mathbf{k_b = n^b \bmod p}$

4- Alice transmet sa clé publique ka à Bob, et Bob transmet sa clé publique kb à Alice.

5- Alice et Bob profitent ensuite de la commutativité de la fonction exponentielle pour établir leur secret commun : $\mathbf{K_{Alice} = (k_b)^a = (n^b)^a \bmod p}$ $\mathbf{K_{Bob} = (k_a)^b = (n^a)^b \bmod p}$

**=> $K_{Alice} = K_{Bob} = n^{ab} \bmod p$**

## II. RSA (Rivest - Shamir – Adleman)

**Le principe**

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

**L'algorithme de chiffrement**

1. **Départ**

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

2. **Création des clés**

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

3. **Chiffrement**

   Le chiffrement d'un message M en un message codé C se fait suivant la transformation suivante : $\mathbf{C = M^e \bmod n}$

4. **Déchiffrement :** il s'agit de calculer la fonction réciproque $\mathbf{M = C^d \bmod n}$
   **tel que $e.d = 1 \bmod [(p-1)(q-1)]$**

**La signature électronique**

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

Rappelons les données :

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

1- Alice accompagne son message chiffré de sa signature, qui correspond à : $\mathbf{M^d}$

2- Bob va donc voir si l'égalité $(M^d)^e$ mod n = M est vérifiée. Si c'est le cas, Alice est bien l'émettrice du message**.**

**<u>Exemple :</u> Chiffrer BONJOUR**

1) Alice crée ses clés :

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

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

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

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

**BONJOUR = 2 15 14 10 15 21 18**

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

**BONJOUR = 002 151 410 152 118**

5) Bob chiffre chacun des blocs que l'on note B par la transformation $\mathbf{C = B^e mod\ n}$ (où C est le bloc chiffré) :

**C1 = 27mod 5141 = 128  C2 = 1517mod 5141 = 800  C3 = 4107mod 5141 = 3761**

**C4 = 1527mod 5141 = 660  C5 = 1187mod 5141 = 204**

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

## VI-Conclusion sur la cryptographie à clé publique

Après avoir vu les avantages indéniables de la cryptographie à clé publique (transmission de la clé, sécurité, authentification), on est en raison de se demander quel pourrait être l'avenir des systèmes à clé secrète.
Le RSA et ses comparses ont-ils relégué le DES et l'AES aux oubliettes ? Faut-il renier Vernam et Feistel ?

La réponse évidente est : **NON !**

En effet, à l'heure actuelle, on utilise des crypto systèmes hybrides, qui couplent les avantages des deux principes, alliant la souplesse de gestion des clés de la cryptographie asymétrique et les performances (vitesse) de la cryptographie symétrique.

Il existe deux types de crypto systèmes hybrides :

- Soit, la cryptographie à clé publique sécurise le transport d'une clé symétrique
- Soit, les entités émettrice et destinatrice se mettent publiquement d'accord sur un secret commun et l'utilisent ensuite pour chiffrer les données grâce à un algorithme symétrique classique.

## Conclusion

Dans ce chapitre, nous avons présenté les notions de base de la sécurité de manière très résumée, puis on a focalisé sur le chiffrement et ses techniques et en particulier les algorithmes les plus connus, qui sont devenus des standards comme DES, AES, RSA et autres, pour mieux comprendre la complexité de tels algorithmes ainsi que leur faiblesses

# Chapitre III : Implémentation de notre Application

## Introduction

On va présenter dans ce chapitre, premièrement l'environnement de travail et les outils utilisés pour se faire. Notre application est écrite en JAVA, et donc on a naturellement utilisé Netbeans. Puis on va donner quelques captures d'écran de l'interface graphique de notre application.

## I- Introduction sur JAVA

Java est un langage de programmation à usage général, évolué et orienté objet dont la syntaxe est proche du C. Ses caractéristiques ainsi que la richesse de son écosystème et de sa communauté lui ont permis d'être très largement utilisé pour le développement d'applications de types très disparates. Java est notamment largement utilisée pour le développement d'applications d'entreprises et mobiles.

Quelques chiffres et faits à propos de Java:

- 97% des machines d'entreprises ont une JVM installée
- Java est téléchargé plus d'un milliards de fois chaque année
- Il y a plus de 9 millions de développeurs Java dans le monde
- Java est un des langages les plus utilisé dans le monde
- Tous les lecteurs de Blue-Ray utilisent Java
- Plus de 3 milliards d'appareils mobiles peuvent mettre en oeuvre Java
- Plus de 1,4 milliards de cartes à puce utilisant Java sont produites chaque année

**Les caractéristiques**

Java possède un certain nombre de caractéristiques qui ont largement contribué à son énorme succès :

- **Java est interprété :** le code source est compilé en pseudo code ou bytecode puis exécuté par un interpréteur Java : la Java Virtual Machine (JVM). Ce concept est à la base du slogan de Sun pour Java : WORA (Write Once, Run Anywhere : écrire une fois, exécuter partout). En effet, le bytecode, s'il ne contient pas de code

spécifique à une plate-forme particulière peut être exécuté et obtenir quasiment les mêmes résultats sur toutes les machines disposant d'une JVM.

- **Java est portable :** il est indépendant de toute plate-forme : il n'y a pas de compilation spécifique pour chaque plate forme. Le code reste indépendant de la machine sur laquelle il s'exécute. Il est possible d'exécuter des programmes Java sur tous les environnements qui possèdent une Java Virtual Machine. Cette indépendance est assurée au niveau du code source grâce à Unicode et au niveau du bytecode.
- **Java est orienté objet :** comme la plupart des langages récents, Java est orienté objet. Chaque fichier source contient la définition d'une ou plusieurs classes qui sont utilisées les unes avec les autres pour former une application. Java n'est pas complètement objet car il définit des types primitifs (entier, caractère, flottant, booléen,...).
- **Java est simple :** le choix de ses auteurs a été d'abandonner des éléments mal compris ou mal exploités des autres langages tels que la notion de pointeurs (pour éviter les incidents en manipulant directement la mémoire), l'héritage multiple et la surcharge des opérateurs, ...
- **Java est fortement typé :** toutes les variables sont typées et il n'existe pas de conversion automatique qui risquerait une perte de données. Si une telle conversion doit être réalisée, le développeur doit obligatoirement utiliser un cast ou une méthode statique fournie en standard pour la réaliser.
- **Java assure la gestion de la mémoire :** l'allocation de la mémoire pour un objet est automatique à sa création et Java récupère automatiquement la mémoire inutilisée grâce au garbage collector qui restitue les zones de mémoire laissées libres suite à la destruction des objets.
- **Java est sûre :** la sécurité fait partie intégrante du système d'exécution et du compilateur. Un programme Java planté ne menace pas le système d'exploitation. Il ne peut pas y avoir d'accès direct à la mémoire. L'accès au disque dur est réglementé dans une applet.
  Les applets fonctionnant sur le Web sont soumises aux restrictions suivantes dans la version 1.0 de Java :
  - aucun programme ne peut ouvrir, lire, écrire ou effacer un fichier sur le système de l'utilisateur

- aucun programme ne peut lancer un autre programme sur le système de l'utilisateur
- toute fenêtre créée par le programme est clairement identifiée comme étant une fenêtre Java, ce qui interdit par exemple la création d'une fausse fenêtre demandant un mot de passe
- les programmes ne peuvent pas se connecter à d'autres sites Web que celui dont ils proviennent.

– **Java est économe :** le pseudo code a une taille relativement petite car les bibliothèques de classes requises ne sont liées qu'à l'exécution.

– **Java est multitâche :** il permet l'utilisation de threads qui sont des unités d'exécutions isolées. La JVM, elle même, utilise plusieurs threads.

## II- IDE ( Interface Developement Environement)

NetBeans est un environnement de développement intégré (EDI), placé en open source par Sun en juin. En plus de Java, NetBeans permet également de supporter différents autres langages, comme Python, C, C++, JavaScript, XML, Ruby, PHP et HTML. Il comprend toutes les caractéristiques d'un IDE moderne (éditeur en couleur, projets multi-langage, refactoring, éditeur graphique d'interfaces et de pages Web).

Conçu en Java, NetBeans est disponible sous Windows, Linux, Solaris (sur x86 et SPARC), Mac OS X ou sous une version indépendante des systèmes d'exploitation (requérant une machine virtuelle Java). Un environnement Java Development Kit JDK est requis pour les développements en Java.

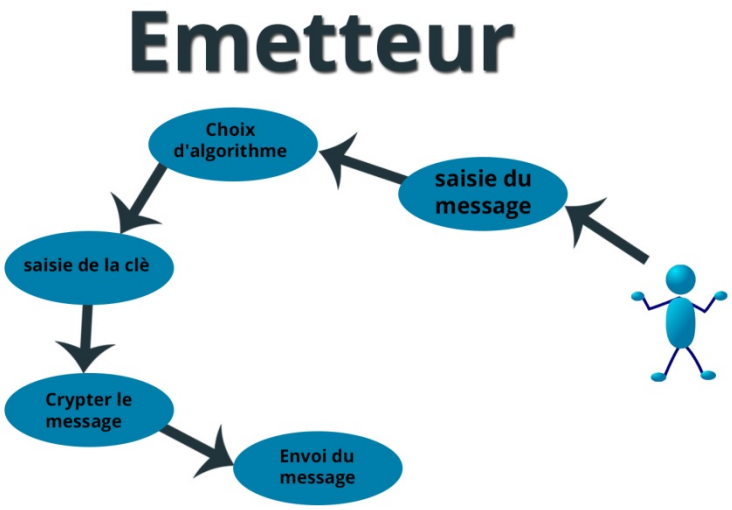
| | |
|---|---|
| **Logo :** |  |
| **Développeur :** | Oracle |
| **Première version :** | 1996, sous le nom de Xelfi1 |
| **Dernière version :** | 8.0 (18 mars 2014) |
| **Environnements :** | Plateforme Java |
| **Langues :** | Multilingue |
| **Type :** | IDE pour Java, PhP, C/C++, Fortran, JavaScript, Python, Ruby |

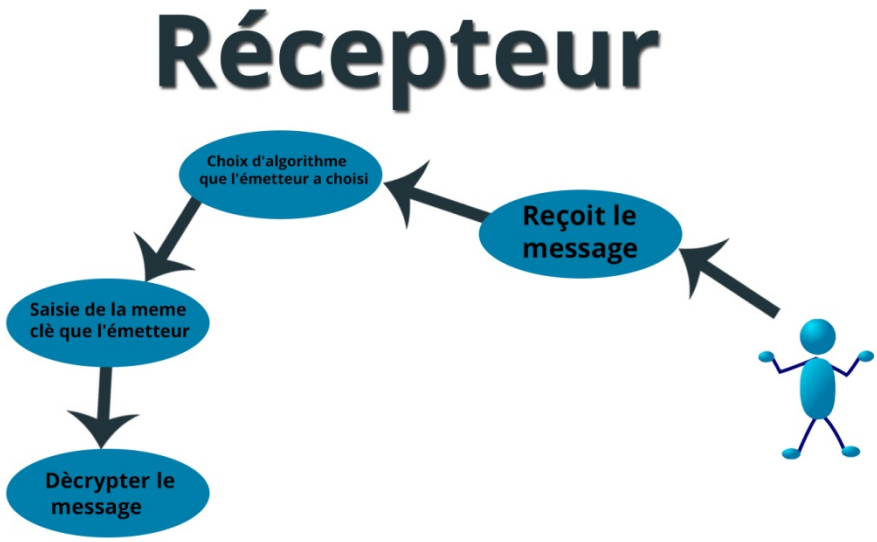
| | |
|---|---|
| **Licence :** | CDDL/ GPL |
| **Site web :** | netbeans.org |

## III- Modélisation de l'Application

Notre application est un Chat sécurisé (grâce au cryptage) entre 2 personnes, pour l'utiliser il suffit à l'émetteur de saisir le message puis le crypter à l'aide d'un des algorithmes proposé et enfin l'envoyer, le récepteur quant à lui doit déchiffrer le message crypté en utilisant le même l'algorithme et la même clé que l'émetteur.

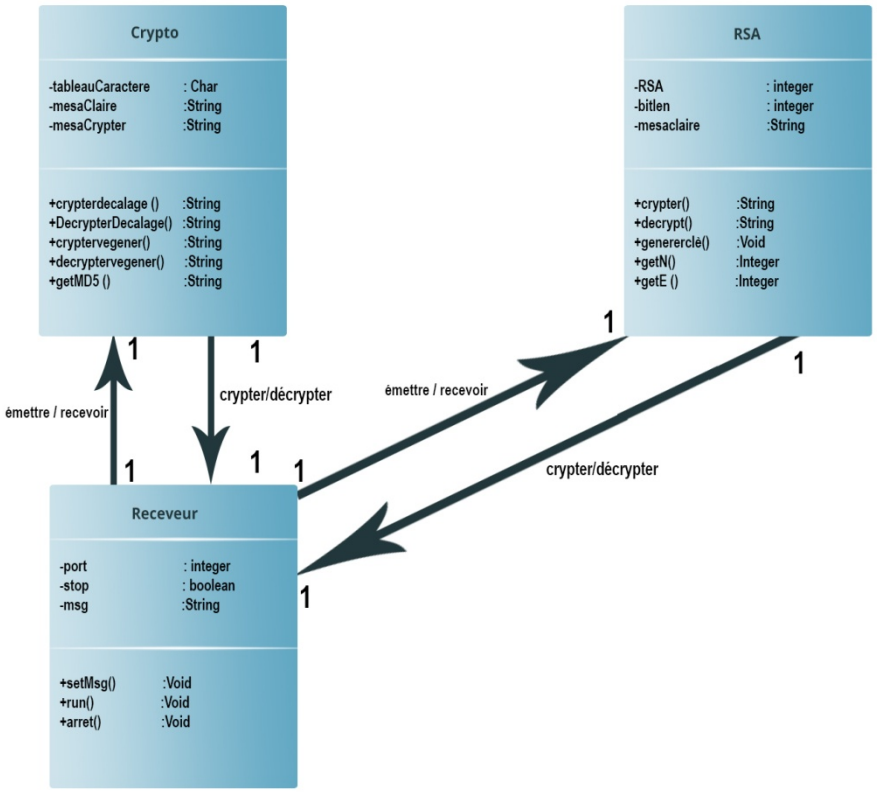
### 1- Diagramme cas d'utilisation






---

**2-** Diagramme des classes **:**




---

## 3- Réalisation de l'application

- **Classe crypto () :**

La classe crypto permet de crypter le message claire avec 4 algorithmes différents (vigenère ,par décalage, vigenère++ et md5)

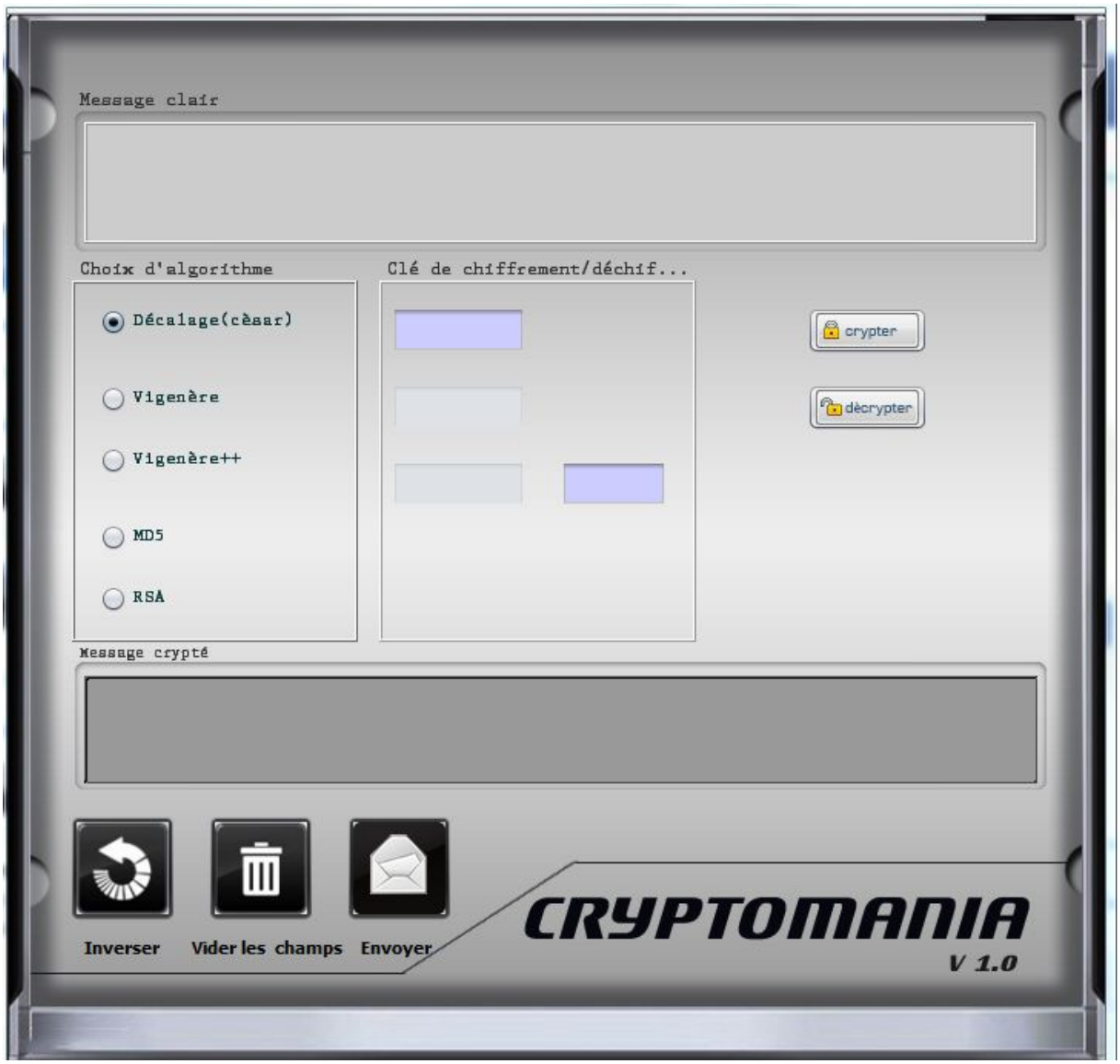
- **Classe RSA () :**

La classe RSA permet de crypter le message claire avec l'algorithme RSA avec une génération des clés **automatiquement**

- **Classe Receveur() :**

La classe Receveur prend en charge le transfert des messages entre les différents utilisateurs dans le même réseau

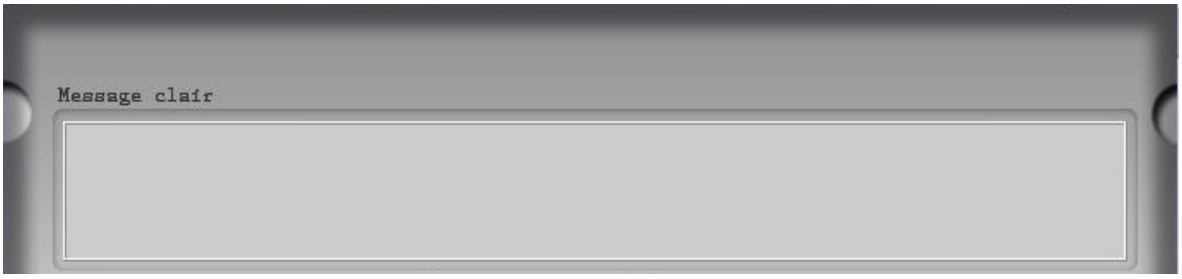
### A- Vue d'ensemble



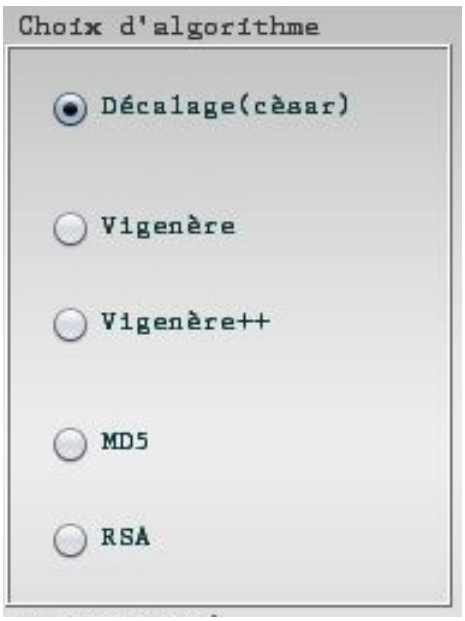

### B- Détails :

L'envoi d'un message crypté passe par les étapes suivantes :

1. **Saisie du message :** On écrit le message qu'on veut crypter dans un premier temps avant de l'envoyer dans le champ « Message Clair » situé en haut de l'application



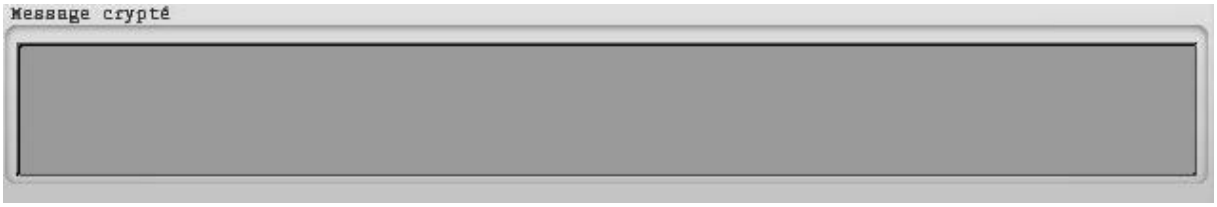

2. **Choix d'algorithme :** On choisit l'algorithme qu'on veut utiliser parmi 5 algorithmes de cryptage suivants :
    - Chiffrement par décalage (César)
    - Vigenère
    - Vigenère ++ : version qu'on a amélioré de Vigenère
    - MD5
    - RSA




3. Saisie de la clé associée à l'algorithme choisi

4. **Cryptage:** Le cryptage se fait selon l'algorithme choisi en cliquant sur le bouton crypté : 

Le message crypté s'affichera alors dans le champ « Message crypté » situé en bas de l'application :




5. **Envoi du message :** le message crypté est envoyé au destinataire grâce au bouton Envoyer :


Le récepteur quant à lui devra sélectionner le même algorithme et la même clé que l'émetteur, ensuite il n'aura qu'à appuyer sur le bouton 

pour déchiffrer le message

**NB :**

Si L'utilisateur a choisi l'algorithme par décalage la clé doit être inferieur à 26 sinon une boîte d'information qui s'ouvre comme la figure suivante :


# Conclusion

Notre application, comme on peut le voir marche correctement, donc les objectifs qu'on s'était posé sont atteints, bien que le temps nous a un peu manqué pour en améliorer l'aspect esthétique, et on pourrait ajouter plus d'algorithmes. Mais comme une première expérience, on a eu beaucoup de difficultés avec RSA surtout, ceci dit on tient à faire remarquer que sur internet, on peut trouver tous les algorithmes expliqués, détaillés et même programmés, le tout est de savoir comment organiser le tout.

# Conclusion générale

Les cryptographes n'ont cessé de redoubler d'ingéniosité, faisant se succéder des dizaines de systèmes de chiffrement plus recherchés les uns que les autres. Se livrant bataille pour la gloire ou l'argent, ils n'ont cessé de faire évoluer cette science qu'est la cryptographie. Avec d'abord une mécanisation (notamment la machine Enigma), puis grâce à l'avènement des ordinateurs, et avec eux une puissance de calcul surpassant de loin le niveau humain, la cryptographie a su trouver son chemin dans les dédales du progrès.

Progrès qui ne s'arrête jamais ! Déjà maintenant, d'autres voies, encore obscures, se dessinent :

Les ordinateurs quantiques : ils procureraient une puissance de calcul colossale, et une sécurité infaillible ! Cependant, la théorie est là, mais la mise en œuvre semble très difficile pour l'instant

- La cryptographie multi variable quadratique : des multiples équations faisant intervenir jusqu'à 120 variables
- Le chaos chiffrant : "noyer" le message dans un signal chaotique et, connaissant les caractéristiques du signal, le retrouver.
- Des méthodes fondées sur les courbes elliptiques
- La stéganographie

Ce sujet passionnant n’a pas de limites, mais on a essayé de donner les notions de base essentielles à la compréhension de ce qu’est la sécurité, avec des détails pour certains algorithmes. L’interface graphique qu’on a développée est simple mais atteint les objectifs initialement posés.

# Bibliographie

## 1- Documents écrits


[1] William Stallings. Cryptography and Network Security : Principles and Practice, 3rd ed. Prentice Hall, 2003.

[2] Didier Müller. Les Codes secrets décryptés. City Editions, 2007.

[3] Alfred J. Menezes, Scott A. Vanstone, and Paul C. Van Oorschot. Handbook of Applied Cryptography. CRC Press, Inc., Boca Raton, FL, USA, 1996.

[4] Travis Spann. Fault induction and environmental failure testing, 2005.

[5] Sergei P. Skorobogatov and Ross J. Anderson. Optical fault induction attacks, 2002.

[6] Adi Shamir and Aran Tromer. Acoustic cryptanalysis : On nosy people and noisy machines, 2004.

[7] Jan C. A. van der Lubbe. Basic methods of cryptography. Cambridge University Press, 1998.

[08] Douglas R. Stinson. Cryptography : Theory and Practice, 2nd ed. CRC Press, Inc., 2002.

[09] B. Lampson. Computer security in the real world, 2001.

[10] Peter Gutmann. Secure deletion of data from magnetic and solid-state memory. 1996


## 2- Site internet


[01] http://www.apprendre-en-ligne.net/crypto/activities/index.html

[02] http://www.security-labas.org/

[03] http://www.bibmath.net/crypto/index.php3

[04] http://www.bibmath.net/crypto/index.php3

[05] http://villemin.gerard.free.fr/Crypto/RSA.htm

[06] http://www.securite.org/

[07] http://users.skynet.be/projetgn/

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

[09] http://www.commentcamarche.net/crypto/crypto.php3

# Remerciements

« Nous tenons tout d'abord à remercier ALLAH de nous avoir donné le courage et la patience pour accomplir ce travail. »

Nous tenons à exprimer notre plus grand remerciement à nos Très chers parents pour leur soutien moral et leurs encouragements.

Nous tenons tout particulièrement à remercier Madame Didi Fedoua pour l'encadrement de ce mémoire et pour la confiance qu'elle nous accordée, sa disponibilité, son aide et ses bons conseils.

Nous voulons remercier toutes les personnes qui, de près ou de loin, ont contribué à l'accomplissement de ce travail.

## Résumé :

Dans ce projet nous avons eu l'opportunité d'attaquer une des notions indispensables dans le domaine informatique : la cryptographie. En effet, nous avons fait une étude théorique sur ce concept, à savoir les deux principales classes de cryptographie symétrique et asymétrique.
Nous avons opté par la suite d'étudier et d'implémenter des algorithmes les plus connus : DES et AES

## تلخيص:

**في هذا المشروع كان لدينا الفرصة لمعالجة واحدة من المفاهيم الأساسية في مجال تكنولوجيا المعلومات آلا و هو التشفير و في هذا الصدد قمنا بدراسات نظرية حول الفرعين الأساسيين و هما التشفير المتماثل والغير المتماثل ثم دراسة وتنفيذ خوارزميات أشهرها DES و AES**

## Summary

**In this project we had the opportunity to address one of the fundamental concepts in the computer field: cryptography. In fact, we did a theoretical study of this concept, namely the two main classes of symmetric and asymmetric cryptography.**
**We decided later to study and implement the best known algorithms: DES and AES**