# Cahier des charges du Projet Nigma

CrypTeam : LAPÔTRE Guillaume (lapotr_g)
GANIVET Justin (ganive_j)
LADEVIE Stéphane (ladevi_s)
GISLAIS Sébastien (gislai_s)

21 novembre 2008

# Table des matières

**I Introduction** **3**

**1 Présentation de la CrypTeam** **3**

**2 Présentation individuelle** **3**
2.1 Guillaume LAPÔTRE . . . . . 3
2.2 Justin GANIVET . . . . . 3
2.3 Stéphane LADEVIE . . . . . 3
2.4 Sébastien GISLAIS . . . . . 3

**II État de l'art** **4**

**3 Cryptographie** **4**
3.1 Cryptage par substitution . . . . . 4
3.1.1 Le cryptage par substitution mono-alphabétique . . . . 4
3.1.2 Le codage par substitution poly-alphabétique . . . . . 5
3.2 Cryptage symétrique . . . . . 6
3.2.1 DES (*Data Encryption Standard*) . . . . . 6
3.2.2 AES (*Advanced Encryption Standard*) . . . . . 8
3.3 Cryptage asymétrique . . . . . 10
3.3.1 RSA (*Rivest Shamir Adleman*) . . . . . 10

**4 Stéganographie** **10**

**III Répartition des charges** **11**

**IV Planning de Réalisation** **12**

**V Conclusion** **13**

# Première partie
# Introduction

## 1 Présentation de la CrypTeam

## 2 Présentation individuelle

### 2.1 Guillaume LAPÔTRE

Actuellement étudiant en Info-spé à EPITA, j'entame avec réjouissance ce fabuleux projet à la fois utile et pédagogique. En effet, la cryptographie m'a toujours attiré. Mais je n'avais jamais l'occasion de m'y interresser de plus près. La stéganographie n'était pas mon idée mais je l'ai trouvée assez interressante.

### 2.2 Justin GANIVET

### 2.3 Stéphane LADEVIE

### 2.4 Sébastien GISLAIS

# Deuxième partie
# État de l'art

La cryptographie et la stéganographie sont deux techniques extrêmement anciennes permettant de transmettre des informations uniquement aux personnes voulues.

La cryptographie protège le message en le chiffrant, c'est-à-dire en le rendant incompréhensible sans connaître l'algorithme de cryptage. On peut citer comme procédé de cryptage historique le chiffre de César qui décale l'alphabet de $n$ rang suivant le chiffre choisi (ainsi si le chiffre choisi est 3, l'alphabet sera : DEFGHI...ZABC).

La stéganographie consiste à cacher le message à transmettre plutot que de le chiffrer. Comme exemple historique, on peut citer un procédé utilisé par César : il écrivait sur le crâne d'un esclave un message puis attendait que les cheveux de cet esclave repoussent. Il envoyait ensuite l'esclave à la personne à qui le message était destiné. Il suffisait donc de raser l'esclave pour récupérer le message.

Cependant, la stéganographie ainsi que la cryptographie étaient utilisées quasiment uniquement par les militaires avant la fin de la Seconde Guerre Mondiale. Depuis, il y a énormément d'application civile au chiffrement.

## 3 Cryptographie

### 3.1 Cryptage par substitution

Le cryptage par substitution est une des techniques les plus basique et les plus ancienne de chiffrement. Le chiffre de César est une technique de cryptage par substitution. Il existe plusieurs types de substitution pour chiffrer des données.

#### 3.1.1 Le cryptage par substitution mono-alphabétique

On remplace chaque lettres de l'alphabet par une autre lettre. Ainsi pour la première lettre il y a 26 possibilités, pour la seconde 25 possibilités, etc. Il existe donc 26! façons distinctes de coder. L'inconvénient de la substitution mono-alphabétique est qu'il faut se souvenir de chaque substitution pour chaque lettre. L'autre inconvénient est qu'en connaissant la langue du message codé on peut relativement facilement déchiffrer le message en se basant sur la fréquence d'apparition de chaque lettre dans la langue. Par exemple

en français la lettre apparaissant le plus souvent est le E. En analysant un texte chiffré avec cette technique de chiffrement, on peut trouver la lettre qui apparaît le plus souvent et l'associer donc à la lettre E. Ensuite on utilise le même raisonnement pour toutes les autres lettres.

### 3.1.2 Le codage par substitution poly-alphabétique

Le chiffre de Vigenère est un bon exemple de codage par substitution poly-alphabétique. On crée un mot qui sert de clé et on le « colle » en dessous du texte à chiffrer. Pour chiffrer ou déchiffrer un message on utilise une matrice $26 \times 26$ avec l'alphabet sur la première ligne et sur la première colonne. Ensuite, on crypte chaque lettre à l'aide de la lettre de la clé que l'on a disposée juste en dessous avec la matrice ci-dessous.

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

Exemple : Chiffrons le mot `Poney` à l'aide de la clé `EPITA` :
`Poney`
`EPITA`
`Poney` chiffré avec cette clé :

- P crypté avec la lettre E : T
- O crypté avec la lettre P : D
- N crypté avec la lettre I : V
- E crypté avec la lettre T : X
- Y crypté avec la lettre A : Y

`Poney` crypté à l'aide de la clé `EPITA` avec le chiffre de Vigenère donne `TDVXY` !

## 3.2 Cryptage symétrique

### 3.2.1 DES (*Data Encryption Standard*)

Jusque dans les années 1970, seuls les militaires possédaient des algorithmes à clé secrète fiables. Devant l'émergence de besoins civils, le NBS (*National Bureau of Standards*) lança le 15 mai 1973 un appel d'offres dans le *Federal Register* (l'équivalent du *Journal Officiel* américain) pour la création d'un système cryptographique. Le cahier des charges était le suivant :

- l'algorithme repose sur une clé relativement petite, qui sert à la fois au chiffrement et au déchiffrement
- l'algorithme doit être facile à implémenter, logiciellement et matériellement, et doit être très rapide
- le chiffrement doit avoir un haut niveau de sûreté, uniquement lié à la clé, et non à la confidentialité de l'algorithme

Les efforts conjoints d'IBM, qui propose l'algorithme *Lucifer* fin 1974, et de la NSA (*National Security Agency*) conduisent à l'élaboration du DES (*Data Encryption Standard*), l'algorithme de chiffrement le plus utilisé au monde durant le dernier quart du XX$^{e}$ siècle.

La clé du DES est une chaîne de 64 bits, mais en fait seuls 56 bits servent réellement à définir la clé. Les bits 8, 16, 24, 32, 40, 48, 56 et 64 sont des bits de parité (bits de détection d'erreur). Le 8$^{e}$ bit est fait en sorte que sur les 8 premiers bits, il y ait un nombre impair de 1. Par exemple, si les 7 premiers bits sont 1010001, le 8$^{e}$ bit est 0. Ceci permet d'éviter les erreurs de transmission.

Il y a donc pour le DES $2^{56}$ clés possibles, soit environ... 72 millions de milliards possibilités. Les grandes lignes de l'algorithme sont :

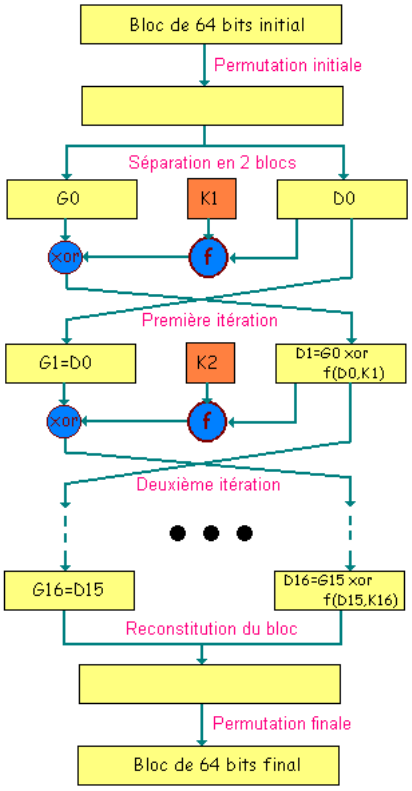
**Phase 1 : Préparation – Diversification de la clé.** Le texte est découpé en blocs de 64 bits. On diversifie aussi la clé $K$, c'est-à-dire qu'on fabrique à partir de $K$ 16 sous-clés $K_1, \ldots, K_{16}$ à 48 bits. Les $K_i$ sont composés de 48 bits de $K$, pris dans un certain ordre.

**Phase 2 : Permutation initiale.** Pour chaque bloc de 64 bits $x$ du texte, on calcule une permutation finie $y = P(x)$. $y$ est représenté sous la forme $y = G0D0$, $G0$ étant les 32 bits à gauche de $y$, $D0$ les 32 bits à droite.

**Phase 3 : Itération.** On applique 16 rondes d'une même fonction. A partir de $G_{i-1}D_{i-1}$ (pour $i$ de 1 à 16), on calcule $G_iD_i$ en posant :
- $G_i = D_{i-1}$
- $D_{i-1} = G_{i-1}$ XOR $f(D_{i-1}, K_i)$

XOR est le OU Exclusif bit à bit, et $f$ est une fonction de confusion, suite de substitutions et de permutations.

**Phase 4 : Permutation finale.** On applique à $G_{16}D_{16}$ l'inverse de la permutation initiale. $Z = P^{-1}(G_{16}D_{16})$ est le bloc de 64 bits chiffré à partir de $x$.




Régulièrement, le DES a fait l'objet de polémiques. Toute sa sécurité repose sur la fonction de confusion $f$, et en particulier à l'intérieur de celle-ci sur des boîtes S, tableau $4 \times 16$ d'entiers compris entre 0 et 15, aux valeurs mystérieuses. Certains ont affirmé que la NSA, qui a finalisé l'algorithme, a placé dans ces boîtes S des trappes qui lui permettaient de tout décrypter, tout en affirmant que l'algorithme est sûr. Toutefois, rien n'a objectivement étayé cela. En particulier, le DES a toujours résisté aux travaux des cryptanalystes non basés sur la force brute.

En revanche, ce qui a signé l'arrêt de mort du DES est l'extraordinaire progression de la puissance des ordinateurs. Le 17 juin 1997, le DES est cassé en 3 semaines par une fédération de petites machines sur Internet. Et on estime très officiellement (dans un rapport présenté au Sénat Américain) à cette date à quelques secondes le temps nécessaire à un Etat pour percer les secrets d'un message chiffré avec le DES.

La solution a été dans un premier temps l'adoption du triple DES, trois applications de DES à la suite avec 2 clés différentes (d'où une clé de 112 bits) :


Si le 3DES est largement suffisant à l'heure actuelle, il est malheureusement trois fois plus lent que le DES. C'est pourquoi, en janvier 1997, le NIST (*National Institute of Standards and Technologies*) lance un nouvel appel pour créer un successeur au DES. L'histoire commence pour l'AES (*Advanced Encryption Standard*).

### 3.2.2 AES (*Advanced Encryption Standard*)

Avec le temps et les progrès de l'informatique, les $2^{56}$ clés possibles du DES n'ont plus représenté une barrière infranchissable. Il est désormais possible, même avec des moyens modestes, de percer les messages chiffrés par DES en un temps raisonnable. En janvier 1997, le NIST (*National Institute of Standards and Technologies*) des États-Unis lance un appel d'offres pour élaborer l'AES, *Advanced Encryption System*. Le cahier des charges comportait les points suivants :

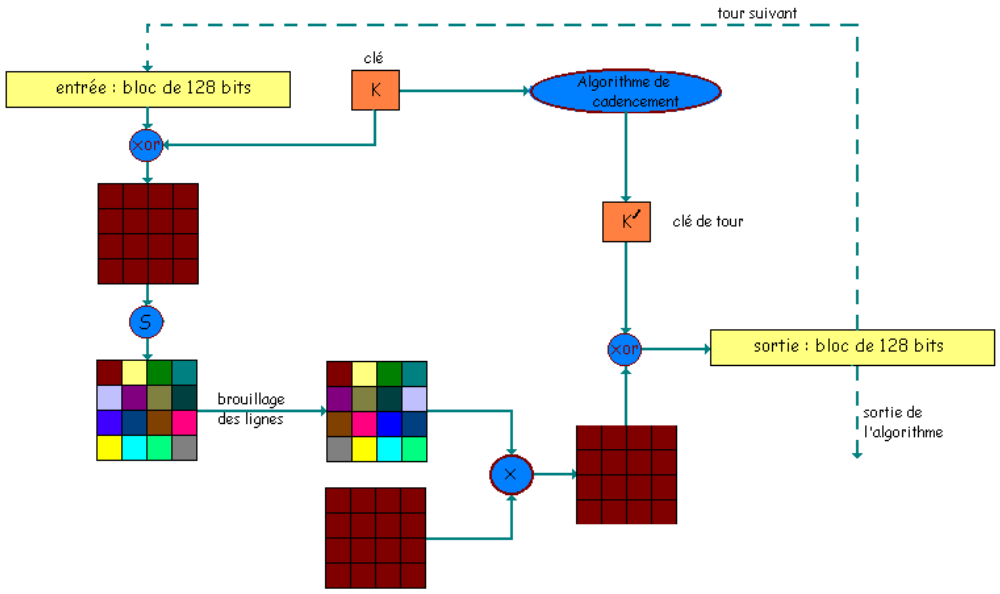
- évidemment, une grande sécurité
- une large portabibilité : l'algorithme devant remplacer le DES, il est destiné à servir aussi bien dans les cartes à puces, aux processeurs 8 bits peu puissants, que dans des processeurs spécialisés pour chiffrer des miliers de télécommunications à la volée
- la rapidité
- une lecture facile de l'algorithme, puisqu'il est destiné à être rendu public
- techniquement, le chiffrement doit se faire par blocs de 128 bits, les clés comportant 128, 192 ou 256 bits

Au 15 juin 1998, date de la fin des candidatures, 21 projets ont été déposés. Certains sont l'œuvre d'entreprises (IBM), d'autres regroupent des universitaires (CNRS), les derniers sont écrits par à peine quelques personnes. Pendant deux ans, les algorithmes ont été évalués par des experts, avec forum de discussion sur Internet, et organisation de conférences. Le 2 octobre 2000, le NIST donne sa réponse : c'est le *Rijndael* qui est choisi, un algorithme mis au point par 2 belges, Vincent Rijmen et Joan Daemen.

Le *Rijndael* procède par blocs de 128 bits, avec une clé de 128 bits également. Chaque bloc subit une séquence de 5 transformations répétées 10 fois :

1. Addition de la clé secrète (par un OU Exclusif)
2. Transformation non linéaire d'octets : les 128 bits sont répartis en 16 blocs de 8 bits (un octet !), eux-même dispatchés dans un tableau $4 \times 4$. Chaque octet est transformé par une fonction non linéaire S
3. Décalage de lignes : les 3 dernières lignes sont décalées cycliquement vers la gauche : la $2^{e}$ ligne est décalée d'une colonne, la $3^{e}$ ligne de 2 colonnes, et la $4^{e}$ ligne de 3 colonnes
4. Brouillage des colonnes : chaque colonne est transformée par combinaisons linéaires des différents éléments de la colonne (ce qui revient à multiplier la matrice $4 \times 4$ par une autre matrice $4 \times 4$). Les calculs sur les octets de 8 bits sont réalisés dans le corps à $2^{8}$ éléments
5. Addition de la clé de tour : à chaque tour, une clé de tour est générée à partir de la clé secrète par un sous-algorithme (dit de cadencement). Cette clé de tour est ajoutée par un ou exclusif au dernier bloc obtenu

Schéma de l'algorithme AES (ou *Rijndael*) :



## 3.3 Cryptage asymétrique

### 3.3.1 RSA (*Rivest Shamir Adleman*)

Le RSA est un algorithme méthode de cryptographie inventée en 1977 par Ron Rivest, Adi Shamir et Len Adleman (d'où le nom de RSA). C'est encore le système cryptographique à clé publique le plus utilisé de nos jours.

Petite anecdote : au départ, Rivest, Shamir et Adleman voulaient prouver que tout système à clé plublique possède une faille, c'est ainsi qu'ils ont créé le RSA !

Son principe de fonctionnement suit 4 étapes :
Tout d'abord il y a la création des clés (que l'on nommera $P$, $Q$, $E$ et $D$). $P$ et $Q$ sont deux grands nombres premiers distincts. Leur génération se fait au hasard en utilisant un algorithme de test de primalité probabiliste. C'est un algorithme qui determine si un nombre est probablement premier selon le degré de probabilité que l'on a fixé dans l'algorithme. En cryptographie, on se « contente » d'avoir un nombre dont on sait qu'il est premier avec une probabilité supérieure à $1 - \frac{1}{2^{100}}$. $E$ est un entier premier avec le produit $(P-1)(Q-1)$. $D$ est tel que $ED = 1 \bmod (P-1)(Q-1)$ donc que $ED - 1$ est un multiple de $(P-1)(Q-1)$. On peut fabriquer $D$ à partir de $E$, $P$ et $Q$ en utilisant l'algorithme d'Euclide.

Ensuite il faut distribuer les clés. Le couple $(n, e)$ constitue la clé publique. Elle est disponible pour toute personne voulant crypter un message afin de nous l'envoyer ensuite. Le couple $(n, d)$ constitue notre clé privée que l'on garde secrète. Si une personne désire nous envoyer un message codé, elle le représente sous la forme de plusieurs entiers $M$ compris entre 0 et $n-1$. Elle possède notre clé publique $(n, e)$ et calcule $C = M^e \bmod n$. C'est ce dernier nombre qu'elle nous envoie.

Nous recevons donc $C$ et on calcule $D$ grâce à notre clé privée : $D = C^d \bmod n$. D'après un théorème d'Euler $D = M^{de} = M \bmod n$. On a donc reconstitué le message.

# 4 Stéganographie

## Troisième partie

# Répartition des charges

Nous allons coder le logiciel *Nigma* qui permettra de crypter un fichier via un algorithme de cryptographie au choix puis l'incrustation dans une image par des méthodes de stéganographie. Ainsi, le fichier sera crypté puis camouflé dans une image quelconque. Il y a deux étapes importantes dans la réalisation du logiciel : le cryptage et la stéganographie. Nous considérons que les deux étapes sont liées.

Les langages de programmation que nous allons utiliser sont le `C` et le `Caml`. Notre programme sera compilé dans l'environnement Unix FreeBSD d'EPITA. Il possèdera un mode console ainsi qu'une interface graphique.

Nous allons coder la partie cryptographie majoritairement avec le langage `Caml` et la partie stéganographie majoritairement avec le langage `C`. Notre interface graphique sera quant à elle codée en `C`.

Les différentes parties de programmation sont réparties dans le groupe de la manière suivante :

| Tâches | Guillaume | Justin | Stéphane | Sébastien |
|---|---|---|---|---|
| Cryptographie | $\oplus$ | | | $\oplus$ |
| Stéganographie | | $\oplus$ | $\oplus$ | |
| Interface Graphique | | $\oplus$ | $\oplus$ | |

Cette répartition est à titre indicatif ; chacun de nous compte en réalité s'intéresser et participer toutes ces tâches. Ainsi, selon notre avancée dans chaque partie, nous pourrons aider l'autre partie du groupe.

## Quatrième partie

# Planning de Réalisation

Pour la première soutenance, nous allons démarrer dans la partie cryptographie avec le cryptage RSA. Pour la stéganographie, nous coderons la création d'une image avec des niveaux de gris, le nuage de points sera organisé. Notre logiciel sera pour l'instant en mode console uniquement. Comme type fichier à crypter, nous nous occuperons excusivement d'un fichier texte afin de contrôler le cryptage et le décryptage plus facilement. Nous mettrons aussi en place notre site Web, il permettra de nous présenter ainsi que notre projet. L'intégralité du code source sera disponible au téléchargement.

Pour la deuxième soutenance, nous ajouterons le cryptage DES dans la partie cryptographie. L'utilisateur aura donc au choix les algorithmes de cryptage RSA et DES. Du côté de la stéganographie, nous présenterons notre progression de l'intégration d'un fichier dans une image. Nous aurons une interface graphique ou une utilisation en mode console au choix pour l'utilisateur. Nous diversifierons les types de fichier à crypter en ajoutant la possibilité d'utiliser une image.

Enfin, pour la soutenance finale, nous implémenterons l'algorithme de cryptage AES que nous ajouterons aux algorithmes déjà réalisés. La partie stéganographie sera terminée, nous aurons alors notre fichier crypté qui sera incrusté dans une image. Notre interface aura sans doute évoluée au regard de notre utilisation régulière.

Tout au long du projet, notre site Web sera mis à jour et décrira l'avancée du Projet Nigma.

| Soutenance | Cryptographie | Stéganographie | Interface | Do |
|---|---|---|---|---|
| 1 | RSA | création d'une image / nuage de points en niveau de gris | mode console | |
| 2 | + DES | progression vers l'intégration du message crypté dans une image | + mode graphique | + |
| finale | + AES | intégration dans une image | | |

# Cinquième partie
# Conclusion

# Source

- http ://www.bibmath.net/crypto/