HAÏUN MEIR
IFERGAN HAÏM

CRYPTOLOGIE

SOUS

SCILAB

SOMMAIRE

INTRODUCTION

La cryptologie, étymologiquement la science du secret, ne peut être vraiment considérée comme une science que depuis peu de temps. Cette science englobe la cryptographie (l'écriture secrète) et la cryptanalyse , c'est à dire le décryptage.
La cryptographie est née du besoin pour les rois et les gouverneurs de pouvoir communiquer sans que les messages secrets tombent dans les mains de l'ennemi. Ainsi depuis l'antiquité cet art ne cesse de progresser et de se développer allant du code de César qui consiste à décaler les lettres d'un message de 3 crans dans l'alphabet, jusqu'à de nos jours à assurer la sécurité des cartes bancaires.
Dans notre étude nous allons principalement nous intéresser à trois méthode de cryptage :
le code de Vigenère, le principe d'échange de clés de Diffie-Hellman, et pour finir le système RSA.
Ainsi nous essaierons de les implémenter avec le langage Scilab.

I) LE CODE DE VIGENERE

1) Biographie de Vigenère

Blaise de Vigenère (5 avril 1523 - 1596) était un diplomate et cryptographe français. Le chiffrement de Vigenère doit son nom au chiffrement qui lui a été attribué au XIXe siècle.

Vigenère est né dans le village de Saint-Pourçain. À 17 ans, il embrassa une carrière diplomatique, qu'il poursuivit pendant plus de trente ans, prenant sa retraite en 1570. À 24 ans, il entra au service du Duc de Nevers. En 1549 il visita Rome au cours d'une mission diplomatique de deux ans, et il y retourna en 1566. Pendant ces deux séjours, il entra en contact à la fois avec des livres traitant de la cryptographies, et les cryptologues eux-mêmes. Quand Vigenère prit sa retraite, à 47 ans, il offrit sa pension annuelle de 1 000 livres aux pauvres de Paris. Il épousa une certaine Marie Varé.

Au cours de sa retraite, il a écrit plus de vingt livres, dont :

le Traicte de Cometes

le Traicte des Chiffres (1585).

En 1584 il devient secrétaire de la chambre du roi Henri III de France.

Vigenère est mort d'un cancer de la gorge en 1596. Il est inhumé en l'église Saint-Étienne-du-Mont, dans le V^{e} arrondissement de Paris.

2) Comment crypter un message avec le code de Vigenère

C'est une méthode de cryptage de texte qui utilise un tableau appelé « Carré de Vigenère » et une clef qui est un mot choisit ou plus précisément une chaîne de caractères d'une longueur quelconque. Tout d'abord le mot clef est épelé bien clairement au dessus du message ,et répété en boucle de sorte que chaque lettre du message soit associée à une lettre de la clef. Si nous voulons chiffrer la phrase « **appeler nord troupes ville** » avec comme mot clef « **ROUGE** », Le texte crypté est alors construit comme suit :Pour chiffrer la première lettre a, commençons par identifier la lettre de la clef placée juste au dessus,à savoir R, qui détermine la ligne du carré de Vigenère commençant par R (soit la 17ème ligne du tableau),et c'est elle qui va définir l'alphabet à utiliser pour substituer une lettre a la lettre originale a. A partir de là on repère la colonne commençant par a et l'on voit qu'elle coupe la ligne 17 sur R.

Schématisation :

Mot_clé	**R**	**O**	**U**	**G**	**E**	**R**	**O**	**U**	**G**	**E**	**R**	**O**	**U**	**G**	**E**	**R**	**O**	**U**	**G**	**E**	**R**	**O**	**U**
Clair	**a**	**p**	**p**	**e**	**l**	**e**	**r**	**n**	**o**	**r**	**d**	**t**	**r**	**o**	**u**	**p**	**e**	**s**	**v**	**i**	**l**	**l**	**e**
Crypté	**R**	**D**	**J**	**K**	**P**	**V**	**F**	**H**	**U**	**V**	**U**	**H**	**L**	**U**	**Y**	**G**	**S**	**M**	**B**	**M**	**C**	**Z**	**Y**
décalage	**17**	**14**	**20**	**6**	**4**	**17**	**14**	**20**	**6**	**4**	**17**	**14**	**20**	**6**	**4**	**17**	**14**	**20**	**6**	**4**	**17**	**14**	**20**

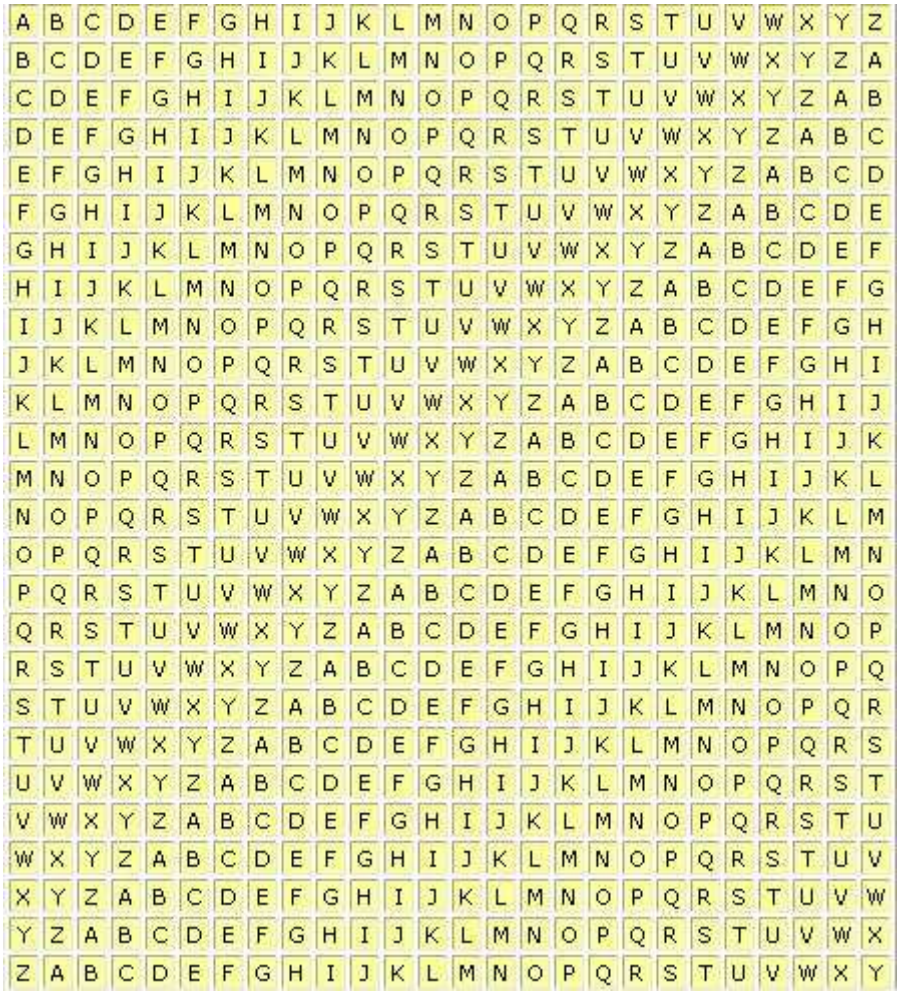
Carré de Vigenère

Texte claire

|

\/

CLEF →



3) Implémentation dans scilab

Tout d'abord avec les fonctions d'algèbre linéaire déjà définies dans scilab, on crée une matrice de nombres représentant le carré de vigenère par des nombres, c'est à dire en remplaçant les lettres par leur indice dans l'alphabet,ainsi a=1,b=2 etc...On appelle cette matrice ‘matrice’.

```
m1= ones(26,1)*[1:26];
m2=m1';  // m2 est donc la transposé de m1
matrice = modulo(m1+m2-2,26)+1;
```

Ensuite on définit un tableau à une ligne et 26 colonnes contenant toutes les lettres de l’alphabet :

```
alphabet=['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'];
```

Pour la suite on aura besoin de deux fonctions de conversion :
La première list_to_string permet de convertir une liste de nombres allant de 1 à 26 en une chaîne de caractères, par exemple la liste [16,1,16,1] sera convertie en ‘papa’ puisque ‘p’ est la 16ème lettre de l’alphabet et ‘a’ est la 1ère .
La deuxième fonction string_to_list permettra de faire le contraire, c'est-à-dire de convertir une chaîne de caractères en une liste de nombres représentant les positions des lettres du mot dans l’alphabet ainsi ‘papa’ est converti en [16,1,16,1].

```
//list_to_string ([16,1,16,1])->papa
function[r]=list_to_string(mot_list)
r = '';
for i =1:length(mot_list)
r = r + alphabet(mot_list(i));
end
endfunction

//string_to_list('papa') -> [16,1,16,1]
function[liste]=string_to_list(mot_string)
liste=zeros(1,length(mot_string));
x=strsplit(mot_string,[1:length(mot_string)-1])
for i=1:prod(size(x))
liste(i)=find(x(i)==alphabet)
end
endfunction
```

Fonction coder

Maintenant avec tout ce qu'on a fait on peut enfin définir la fonction **coder** qui prend en paramètre 2 chaînes de caractères : le mot à coder et la clef

```
function[r]=coder(mot,clef)
//On convertit le mot en une liste de nombres
mot_liste=string_to_list(mot)
//On convertit la clef en une liste de nombres aussi
clef_liste=string_to_list(clef)
//On initialise une liste de la même longueur que le mot à coder avec que des zéros, qu'on
//appelle « crypte »
crypte = zeros(1,length (mot_liste))

//On fait une boucle, pour i allant de 1 à longueur(mot_liste), et on remplit la i ème case
//par la valeur contenu dans la matrice de vigenère à l'emplacement
//marice( clef_liste( (i-1 modulo longueur(clef_liste)) +1) , mot_liste(i))
for i=1:length(mot_liste)
  crypte(i)=matrice(clef_liste(modulo(i-1, length(clef_liste))+1),mot_liste(i))
end

//ensuite on convertit la liste "crypte" en "chaîne de caractères " qui est donc le mot codé
que la fonction renvoie.
r=list_to_string(crypte)
endfunction
```

Notons que la difficulté de cette fonction réside dans le fait que la clef peut être plus petite que le mot à coder, en effet dans ce cas la clef est répétée jusqu'à la fin du mot, c'est pourquoi on utilise les **modulo**.

Essayons de comprendre avec un exemple concret

Par exemple si on veut coder le mot 'bonjour ' avec la clef 'ja', logiquement on code la 3ème lettre de bonjour 'n' avec la 1ère lettre de la clef 'j' , et la 4ème lettre de bonjour 'j' avec la 2ème lettre de la clef 'a'.

Pour i=3

A la 3ème position de la liste **crypte** on place l'élément de la **matrice** de vigenère, se trouvant à la ligne L =clef_liste(3-1 modulo longueur(clef_liste)+1)

=clef_liste ((3-1) mod 2 +1)=cle_liste(1)=10

Et à la colonne C=mot_liste(3) = 14

Ainsi le **n** de bonjour est converti en **14,** est codé en **matrice(10,14)=24** est est ensuite converti en **w** :

Schéma : n→14→matrice(10,14)=24→w

Ainsi dans une fenêtre scilab cela donne :

```
-->coder('bonjour','ja')
 ans  =

 kowjxua
```

Le **n** est bien codé en **w**

Fonction decoder

```
function[r]=decoder(mot_code,clef)
//conversion du mot en liste
liste_mot=string_to_list(mot_code)
//conversion de la clef en liste
clef_liste=string_to_list(clef)
//initialisation d'une liste de zéros de la même longueur que le mot à décoder  qu'on va
//remplir
decrypte=zeros(1,length(liste_mot))
//boucle ressemblant à celle utilisée pour la function coder
for i=1:length(liste_mot)
clef_ligne=matrice(clef_liste(modulo(i-1, length(clef_liste))+1),:)
decrypte(i)=matrice( 1 , find(liste_mot(i) = =clef_ligne))
end
//conversion de decrypte en chaîne de caractères qui est le mot décodé
r=list_to_string(decrypte)
endfunction
```

Pour comprendre ce que fait la boucle, si par éxemple je reprend l'exemple précedent et je veux décoder le mot 'kowjxua' avec la clef 'ja'.
On décode la 3ème lettre **w** du mot avec la 1ère lettre de la clef **j**, puisque la clef se répète le long du message si elle est plus courte que ce dernier.
Ainsi pour i=3 :
clef_ligne=matrice(clef_liste((3-1 mod longueur(clef_liste)) +1), :)
=matrice(clef_liste (1), :) = matrice(10, :)
= le tableau qui contient que la dixième ligne de la matrice de Vigenère
A la 3ème position de la liste decrypte on place l'élément se trouvant dans la **matrice** de vigenère à la ligne L=1 et à la colonne C=find(liste_mot(3) = = dixieme ligne de la matrice)
=première occurrence de 24 dans la dixième ligne de la matrice=14.
Ainsi on peut schematiser le decodage de w par : w→24→ matrice(1,14)→14→n

Testons le programme avec scilab :

```
-->decoder('kowjxua','ja')
 ans  =

 bonjour
```

Le message clair est bien bonjour

4) Points forts du code de Vigenère

Cet algorithme de cryptographie, qui est en fait une amélioration du code de César est une méthode de substitution polyalphabétique où l'alphabet chiffré change au cours du chiffrement selon une clef, contrairement au code de César qui est une méthode de chiffrement par substitution monoalphabetique ou l'alphabet chiffré reste inchangé au cours du temps. Ce code comporte beaucoup de points forts. Il est très facile d'utilisation, et le déchiffrement est tout aussi facile si on connaît la clé. La grande caractéristique du chiffre de Vigenère est qu'il est impossible par une analyse des fréquences simple de retrouver où sont certaines lettres. En effet dans le tout premier exemple utilisé,quand on a crypté le message : « appeler nord troupes ville » par « RDJKPVF HUVU HLUYGSM BMCZY »,la lettre U qui apparaît trois fois dans le texte chiffré est mise successivement pour la lettre o ou d, mais jamais pour la lettre e. Par ailleurs, une même lettre apparaissant à plusieurs reprises dans le texte chiffré, alors qu'elle remplace des lettres différentes engendre des ambiguïtés pour le cryptanalyste

Car si j'utilise un alphabet monoalphabétique dans lequel je remplace tous mes E par des U,si je fait une analyse de fréquences, et que je vois que la lettre la plus utilisé c'est le U,j'en conclus que les u remplacent les e car en français la lettre la plus utilisée est le e. Notons que cette méthode de décryptage dont on attribue l'origine au philosophe arabe Al-Kindi du IX^{e} siècle a ses limites , en effet en 1969 ,Georges Perec écrivit un roman de 200 pages : *La Disparition* sans utiliser une seule fois la lettre e.

Un autre avantage du code de Vigenère réside dans le fait que l'on peut produire une infinité de clés.

5) Le point faible de cet algorithme

Le principal point faible de ce système est qu'il est attaquable par une analyse fréquentielle rigoureuse. De plus, il faut bien transmettre la clé au destinataire à un moment ou à un autre avec tous les risques que cela comporte.

II) PRINCIPE D'ECHANGE DE CLES DE DIFFIE-HELLMAN

Inventé en 1976, l'échange de clés Diffie-Hellman, du nom de ses auteurs Whitfield Diffie et Martin Hellman, est une méthode par laquelle deux personnes nommées conventionnellement Alice et Bob peuvent se mettre d'accord sur un nombre qui servira de clé pour correspondre, sans que quelqu'un d'autre puisse l'intercepter en l'écoutant par exemple.

1) Principe

- Alice et Bob ont choisi un groupe (soit un corps fini, dont ils n'utilisent que la multiplication, soit une courbe elliptique) et une génératrice g de ce groupe.
- Alice choisit un nombre au hasard a, élève g à la puissance a, et transmet à Bob g^a.
- Bob fait de même avec le nombre b, et transmet g^b à Alice.
- Alice, en élevant le nombre reçu de Bob à la puissance a, obtient g^{ba}.
- Bob fait le calcul analogue et obtient g^{ab}, qui est le même. Mais puisqu'il est difficile d'inverser l'exponentiation dans un corps fini, c'est-à-dire de calculer le logarithme discret, Eve ne peut pas découvrir a et b, donc ne peut pas calculer g^{ab} .

2) Exemple et implémentation dans Scilab

1. Alice et Bob choisissent un nombre premier p et une base g. Dans notre exemple, p=23 et g=3
2. Alice choisit un nombre secret a=6
3. Elle envoie à Bob la valeur A=g^a mod p = 3^6 mod 23 = 16
4. Bob choisit à son tour un nombre secret b=5
5. Bob envoie à Alice la valeur B=g^b mod p = 3^5 mod 23 = 13
6. Alice peut maintenant calculer la clé secrète : (g^b mod p)^a mod p = 13^6 mod 23 = 6
7. Bob fait de même et obtient la même clé qu'Alice : (g^a mod p)^b mod p = 16^5 mod 23 = 6

Si on implémente ça dans scilab, cela donne :

```
p=23
g=3

//Alice choisit a calcule A, et envoie A à Bob
a=6
A=modulo(g**a,p)

//Bob choisit b, calcule B et envoie ce B à Alice
b=5
B=modulo(g**b,p)
//Alice calcule la clé k1 avec B, a, et p
k1=modulo(B**a ,p )

//Bob calcule la clé k2 avec A, b, et p
k2=modulo(A**b ,p )
```

Et lorsqu’on exécute cela, on a :

```
--> p  =

   23.
g  =

   3.
a  =

   6.
A  =

   16.
b  =

   5.
B  =

   13.
k1  =

   6.
k2  =

   6.
```

On a bien k1=k2=6=clé secrète créée

III) RSA

RSA est un algorithme asymétrique de cryptographie à clé publique, très utilisé dans le commerce électronique, et plus généralement pour échanger des données confidentielles sur Internet. Cet algorithme a été décrit en 1977 par Ron Rivest, Adi Shamir et Len Adleman, d'où le sigle RSA. RSA a été breveté par le MIT en 1983 aux États-Unis d'Amérique, mais le brevet a expiré le 21 septembre 2000.

Le fonctionnement du cryptosystème RSA est basé sur la difficulté de factoriser de grands entiers comme nous allons voir.

1) Principe du cryptosystème RSA.

Chaque utilisateur de ce cryptosystème procède de la façon suivante :
1) il choisit deux grands nombres premiers p et q et calcule n = pq.
2) Il choisit un entier e premier avec $\varphi(n)$, tel que $1 < e < \varphi(n)$.
$\varphi(n)$ étant la fonction indicatrice d'Euler telle que $\varphi(n) = (p-1).(q-1)$ (car p et q sont des nombres premiers)

3) Il détermine l'entier d tel que $1 < d < \varphi(n)$ et $e.d = 1 \bmod \varphi(n)$, Ce calcul peut être effectuer en utilisant l'algorithme d'Euclide.
4) Il publie ensuite le couple (n, e), qui est sa clé publique, et il conserve secret le couple $(\varphi(n),d)$, qui est sa clé secrète.

4) Supposons qu'un utilisateur (nommé Bob) désire envoyer un message M à une personne (nommée Alice), il lui suffit de se procurer la clé publique (n, e) publiée par Alice, et de calculer le message chiffré C :

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

Bob envoie ensuite le message chiffré C à Alice, qui est capable de le déchiffrer à l'aide de sa clé privée $(\varphi(n), d)$ que elle seule connaît (si p et q assez grands pour être trouvés,en effet la difficulté réside dans le calcul de $\varphi(n)$ qui nécessite la connaissance de p et q) :

$M = C^d \bmod (n)$

2) Exemple concret dans scilab

Nous allons tester le principe de RSA avec une petite application dans scilab dans laquelle :

```
p=2
q=5
n=p*q
fi_n=(p-1)*(q-1)
e=3
d=7 //car e*d=3*7=21 = 1 mod 4 =1 mod fi_n

//soit m le message à envoyer
m=2

//m_prime le message codé
m_prime = modulo(m**e,n)

//m_decod etant le message décodé c'est à dire le message clair
m_decod = modulo(m_prime**d,n)
```

Ainsi après exécution scilab affiche bien le message codé qui est m_prime=8 et le message décodé qui est le message claire m_decod=m=2 (voire ci-dessous) :

```
p  =

   2.
q  =

   5.
n  =

   10.
fi_n  =

   4.
e  =

   3.
d  =

   7.
m  =

   2.
m_prime  =

   8.
m_decod  =

   2.
```

CONCLUSION

Les gouvernements ont compris depuis des centaines d'années que l'art de la cryptologie était vital. En effet cette science est la base de l'espionnage, ainsi avec l'aide de la cryptologie on a pu déjouer de nombreux plans historiques, comme la tentative d'assassinat de la reine Elisabeth d'Angleterre en 1586 grâce au cryptanalyste Thomas Phellipes qui déchiffra le chiffre de Marie Stuart reine d'Écosse qui voulu assassiner la reine. De meme durant la seconde guerre mondiale, le décryptage des messages allemands vers 1942 émis par des machines enigma aidèrent beaucoup les alliés à gagner la guerre et à la raccourcir. Ce qui est très surprenant c'est que le chiffre de Vigenère a tenu près de 300 ans, ce qui est comparable au chiffre du Roi Louis XIV, développé par Les Rossignol (famille de cryptologues), qui a finit par être déchiffré vers 1893 par Étienne Bazeries, cryptologue militaire français.

Ainsi ce fut fort intéressant pour nous de nous plonger dans le monde de l'espionnage et de la cryptographie, et d'implémenter quelques cryptosystèmes dans scilab comme le code de Vigenère, le principe d'échange de clés de Diffie-Hellman ou encore le système RSA car le sujet est inépuisable et on ne cesse d'apprendre. De plus concernant le code de Vigenère, Scilab était adéquat car il contient beaucoup de fonctions permettant de manipuler des matrices (**ones, find**…). Concernant l'algorithme RSA on a du manipuler des petits nombres car Scilab ne prend en charge que des nombres inférieures à 64 bits(2^64).

LEXIQUE

Algorithme de chiffrement : processus d'opérations à effectuer pour le chiffrement, qui doit être lié à une clef qui le précise.

Alphabet chiffré : redisposition des lettres de l'alphabet usuel, qui permet ensuite de déterminer comment chaque lettre du message sera chiffrée. L'alphabet chiffré peut aussi se composer de nombres ou de n'importe quels autres caractères, mais, dans tous les cas, il impose de changer les lettres du message d'origine.

Chiffre : n'importe quel système appliqué pour dissimuler le sens d'un message en remplaçant chaque lettre du message d'origine par une autre.

Algorithme asymétrique (ou à clé publique) : fonction cryptographique de codage à clé secrète dont la clé de chiffrement est différente de la clé de déchiffrement (cette dernière pouvant être difficilement calculée à partir de la clé de chiffrement).

Chiffre de substitution : cryptosystème dans lequel chaque lettre du message est remplacée par un autre caractère, mais garde sa place dans le message.

Chiffre de transposition : cryptosystème dans lequel chaque lettre du message reste inchangée,mais mise à une autre place dans le message.

Chiffrer : transformer le message d'origine en message chiffré (codé)

Clef : paramètre qui transforme l'application de l'algorithme général de chiffrement en le spécifiant. Généralement l'ennemi peut connaître l'algorithme utilisé, mais il doit en aucun cas connaître la clef

Stéganographie : technique pour cacher l'existence d'un message, par opposition à la cryptographie qui en dissimule le sens.

Substitution homophonique (chiffre de) : chiffre qui offre plusieurs substitutions possibles pour chaque lettre du texte en clair. S'il y a, disons, six substitutions possibles pour la lettre a, ces six caractères ne représenteront que la lettre a. C'est une variante du chiffre de substitution monoalphabétique.

Substitution monoalphabétique (chiffre de) : chiffre de substitution où l'alphabet chiffré reste le même au cours de tout le chiffrement.

Substitution polyalphabétique (chiffre de) : chiffre de substitution où l'alphabet chiffré change au cours du chiffrement, comme dans le chiffre de Vigenère. Ce changement s'exécute selon une clef.
Texte chiffré : message obtenu après chiffrement. Texte en clair : message original avant chiffrement.

Bibliographie

SINGH (Simon),*Histoire des codes secrets*,Éditions Jean-Claude Lattès pour la traduction francaise ,1999

L'Art du secret : Dossier pour la Science . N°36 , Paris , Pour La Science -- juillet/oct. 2002

KAHN (David),*La guerre des codes secrets*,Paris,InterEditions,1980

Sites Internet :

http://cryptage.free.fr/html/intro/intro.html
http://fr.wikipedia.org/wiki/Cryptographie
http://www.apprendre-en-ligne.net/crypto/menu/index.html