DÉCAMPS Régis & JUÈS Thomas

110101
111011
111001
111100
100011
001111
001110
110111
111011
111111
011111
011111

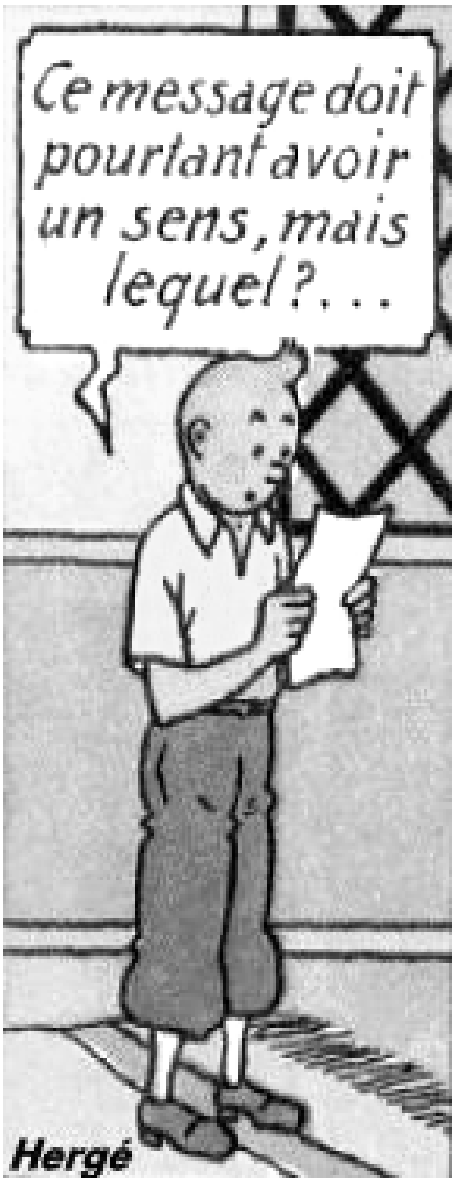
Travail d'intérêt personnel encadré : La cryptographie

Histoire et fonctionnement des techniques de cryptographie et aperçus de cryptanalyse.

I Introduction

Les procédés de cryptographie sont l'ensemble des méthodes de chiffrement qui permettent de rendre un message intelligible uniquement aux personnes auxquelles il est destiné. On a tendance à croire que la cryptographie est une technique récente mais vers 1900 av. JC un scribe égyptien avait codé un message en utilisant des hiéroglyphes originaux.

Cependant il est vrai que les techniques de cryptographie les plus perfectionnées sont aussi les plus récentes et que la cryptographie n'a pu réellement se développer qu'avec l'apparition de l'informatique.



1 Sommaire

2 Bibliographie

Encyclopédie Universalis, article intitulé cryptographie (Larousse)

Introduction aux méthodes de cryptologie, BECKETT Masson

A course in number Theory and cryptography KOBLITZ (Springer Verlay)

Quadrature n°9 (juillet août 91) La cryptographie à clefs publiques par M .D. INDJOUDJIAN

Pour la science n°220 (février 96) La confidentialité des communications (Thomas BETH)

Chiffrement & cryptographie http://dafne.mines.u-Nancy.fr/~tisseran/I33_Reseaux/cryptage/cryptage.htm

Groupe de Recherche en Complexité et cryptologie. http://www.dmi.ens.fr/equipes/grecc

Crypotlography Timeline http://www.clark.net/pub/cme/html/timeline.htm

3 Iconographie

Image page 2 : « Tintin et Milou » : *Le Lotus Bleu* par HERGE

Tous les schémas du dossier par Régis DECAMPS

II Concepts Généraux

Avant de voir les principes de plusieurs procédés célèbres de chiffrement, fixons les bases de la cryptographie et de son vocabulaire.

1 Définitions

message : ensemble de caractères, chiffres, ponctuation et autres symboles.
Le message clair (*plaintext*) est compréhensible par n'importe qui.
Le message crypté ou cryptogramme (*ciphertext*) est, en principe, intelligible uniquement aux personnes auxquelles il est destiné.

« casser » un code : trouver la faille dans la fonction de chiffrement qui permet de décrypter les messages sans y être autorisait dans un laps de temps acceptable.

décrypté un message : transformer le texte crypté en texte intelligible.

cryptanalyse : art de déchiffré des messages cryptés, de casser les codes.

cryptologie : branche mathématique née de la cryptographie qui repose sur la théorie des nombres.

Ce symbole renvoie à un complément sur transparent

Diagramme Alice, Oscar, Bob.

Alice (nom usuel de l'émetteur) veut transmettre un message à Bob (destinataire du message) par un canal public (ligne téléphonique, courrier, courrier électronique…). Oscar est un espion et a la possibilité d'intercepter les messages qui transitent sur le canal public. Mais les messages qu'il intercepte ont été cryptés par Alice et il ne peut les comprendre.

bit : Comme nous allons utilisé un peu le système binaire (/2), nous rappelons qu'un bit (*binary digit*) est l'unité de la numérotation binaire (0 ou 1). On munit (/2) de la loi d'addition $\oplus$ connue sous le nom XOR (*eXclusive OR*, ou exclusif).

octet : c'est un groupe de 8 bits.

2 Intérêts du chiffrement

2.1 Besoins

Besoin militaire & gouvernementaux évidents : le chiffrement s'est considérablement développé lors des deux guerres mondiales. Jusqu'au début de la décennie, la cryptographie était considérée comme une arme de guerre. Aujourd'hui encore, son usage est restreint. En France la loi n'autorisait pas le cryptage avant 1995. Besoin civils apparus dans les années 60 seulement : confidentialité de documents importants pour les entreprises, banques, institutions financières. Les particuliers en font un usage grandissement.

2.2 But

- confidentialité : le message crypté doit être incompréhensible, sauf pour les personnes autorisées.
- authentification : toute modification du document doit pouvoir être mise en évidence. Cette fonction peut ainsi servir à « signer » numériquement et de façon infalsifiable des contrats.

3 Le codage

Transforme le message clair en une suite de nombres qui seront plus facilement traités par l'algorithme de codage sur l'ordinateur.

Il repose sur une matrice de codage de dimension (2,N) (matrice d'une transposition)

> Exemples : alphabet : A=1, B=2, … Z=26 (N=26) , tableau ASCII (*American Standard Code for Information Interchange* publié par l' *American National Standards Institute* et utilisé sur les PC, N=256), (pour PC), Unicode (qui est un ASCII étendu à $N = 2^* 2^8 = 65536$)

Dans tout l'exposé, nous allons confondre le texte avec sa représentation en ASCII. Ainsi « Bonjour » est strictement équivalent à $(66\ 111\ 110\ 106\ 111\ 117\ 114)_{10}$ en base 10 et à $(1000010\ ;\ 1101111\ ;\ 1101110\ ;\ 1101010\ ;\ 1101111\ ;\ 1110101\ ;\ 1110010)_2$ en base 2.

Bien entendu, le codage n'existait pas avant l'invention des ordinateurs. Cependant nous l'utiliserons car il ne change en rien le principe des méthodes de codage concernées (et qu'il est plus logique d'additionner des nombres que des lettres…).

III Chiffrement classique

Le chiffrement classique décrit la période antérieure à l'informatique. Il est simple à mettre en œuvre mais ne représente pas une protection valable de nos jours.

1 Substitution

1.1 Substitution simple (ou monoalphabétique)

1.1.1 Historique de méthodes

Vers 1900 av JC un scribe égyptien utilise un alphabet non-usuel

> Utilisation de symboles. Bonjour se crypte alors

On peut objecter qu'il s'agit en fait d'un codage dont le code est maintenu secret plus que d'un système de chiffrement.

500 av JC : Les scribes hébreux utilisent un cyptage par substitution simple en « renversant » l'alphabet (le A correspond au Z, le B au Y etc.). Cette méthode est connue sous le nom **atbash**.

50 av JC. Jules **César** lui-même donne naissance à une méthode qui donne son nom pour les communications du gouvernement romain.

1.1.2 Principe

A chaque caractère du texte clair correspond un caractère du texte crypté.

M = ($x_1x_2…x_i…x_n$) x_i est le $i^{ème}$ caractère du message clair après codage. (il s'agit donc d'un nombre)

Le message codé est C = « $c(x_1)c(x_2)…c(x_i)…c(x_n)$ » avec c la fonction de cryptage. On remarquera que ces méthodes conservent le nombre de lettre du texte clair (ici, le texte clair et le texte crypté contiennent n lettres).

> Pour la **méthode de César** : $\mathbf{c(x_i)=x_i+k\ [N]}$
>
> Pour décoder $\mathbf{x_i = c(x_i) - k}$

Rem : si le texte est crypté avec k = 13, et en ne codant que l'alphabet (N=26), le texte se décrypte avec –k = k [26] ; c'est la méthode ROT13 (pour rotation 13 caractères).

Rem : Jules César utilisait k = 3

1.1.3 Fiabilité

Ces méthodes sont **peu fiables** car elles conservent la **fréquence d'apparition** des lettres. Si Oscar (un espion) se procure un long message il peut étudier la fréquence d'apparition des caractères. Par étude statistique, le « » (une espace) revient le plus souvent, suivi du « e » en français, etc. … Il en déduit la valeur de k et obtient immédiatement le texte clair.

D'autre part, s'il parvient à se procurer la position d'un caractère donné dans le texte clair, il est en mesure de décrypter le message immédiatement.

Plus simplement, il n'y a que N valeurs distinctes possibles pour k, avec la rapidité des machines actuelles, il ne pose aucun problème d'essayer toutes les clés (et même à la main, l'opération n'est pas très longue).

En fait, cette méthode a bien fonctionné, car personne, à l'époque, ne connaissait le « truc ».

1.2 Substitution incohérente (ou polyalphabétique)

1.2.1 Principe

Il s'agit en fait d'une amélioration de la méthode précédente. En faisant varier la valeur de k, on casse la fréquence d'apparition des lettres et on améliore grandement la difficulté de décryptage.

Substitution incohérente $c(x_i)=x_i+k(i)$ [N]avec $k = (k_1,k_2,k_3,\ldots k_m)$ la clé (*key*)

C'est le regroupement de m substitutions simples qui sont appliquées successivement suivant la place du caractère dans le message. La valeur de k(i) est donc donnée par la position i du caractère à chiffré ainsi que par la clé.

On peut également décrire cette méthode en utilisant l'espace vectoriel $(Z/NZ)^m$. Le message est coupé en blocs de m lettres qui correspondent chacun à 1 vecteur dans $(Z/NZ)^m$. **La transformation de Vigenère** que nous venons de décrire consiste simplement à effectuer la translation de vecteur $\overrightarrow{k}$ de coordonnées $\begin{pmatrix} k1 \\ k2 \\ \vdots \\ km \end{pmatrix}$.

Enfin, la méthode est encore une fois considérablement améliorée en ajoutant un coefficient multiplicateur. On obtient alors m **transformations affines** qui sont appliquées en cycle en fonction de la position du caractère à crypté dans le texte.

$c(x_i)=a(i)\,x_i+k(i)$ [N] avec $k = (k_1,k_2,k_3,\ldots k_m)$ et $a = (a_1,a_2,a_3,\ldots a_m)$

Autrement dit, on applique successivement $\begin{cases} c(x_i) = a_1\, x_i + k_1\ [N] \\ c(x_i) = a_2\, x_i + k_2\ [N] \\ \vdots \\ c(x_i) = a_m\, x_i + k_m\ [N] \end{cases}$

Il faudra prendre garde d'avoir **pgcd(a(i),N)=1** sinon on s'aperçoit immédiatement que la fonction n'est plus bijective (l'image est toujours k).

1.2.2 Historique

1466 : Invention de la méthode par Alberti. Il utilise alors un disque pour simplifier l'application de la méthode.

1553 : Giovan Batista Belaso inaugure avec la **notion de clé** de cryptage.

1585 : Blaide de Vigenère écrit un livre sur le cryptage et laisse son nom à la méthode inventée par Belaso.

1.2.3 Fiabilité

Apparemment, ces méthodes n'ont pas été cassées avant le XIX siècle.

2 Transposition

Les méthodes de transpositions sont beaucoup plus contraignantes à l'usage que les méthodes de substitution. Il faut écrire le texte dans une matrice de taille définie, ce qui pose rapidement des problèmes de stockage et de facilité de maniement.

2.1.1 Historique

1914 **algorithme allemand ADFGFX** utilisé pendant la première guerre mondiale a été cassé par une étudiante française pendant la guerre.

2.1.2 Méthodes

Utilisation de matrices de tailles définies à l'avance.

Par exemple, on peut écrire le texte horizontale et ensuite le lire verticalement.

Mais, pour améliorer le système, on peut lire les colonnes verticalement les colonnes avec un ordre donné par une clé. Ainsi,

Le message clair : « Bar plazza minuit » avec la clé (2143)

B	A	R	P
L	A	Z	Z
A	M	I	N
U	I	T	E

(on a complété le tableau avec un E, mais on peut y mettre les caractères que l'on veut)

donne AAMIBLAURZITPZNE.

2.1.3 Fiabilité

Les méthodes de transpositions sont plus fiables que les substituions. Cependant, elles requièrent de gros moyens techniques (notamment mémoire), c'est pourquoi elles sont assez peu utilisées et développées.

La Fiabilité augmente avec la taille de la matrice.

3 Redondance

3.1 Principe

Il s'agit de « noyer » le texte dans un message plus grand.

Il est ainsi possible de ne considérer que le i ème mot de chaque ligne d'un texte, ou seulement les 3 premières lettres de chaque mot (comme dans TINTIN : *Le Lotus Bleu*).

La sécurité est assez bonne, car l'espion ne sait même pas qu'il s'agit d'un message codé. Mais s'il cherche un sens au message qu'il a intercepté, il trouvera assez rapidement le texte originel. D'autre part, cette méthode présente rapidement des problèmes de longueur de message.

IV Chiffrement Moderne

Reposant sur la puissance de calcul des ordinateurs, ces méthodes apparaissent comme infaillibles.

1 Clé privée

1.1 Méthode DES

1.1.1 Historique

Le *Data Encryption System* est né dans les laboratoires de la firme IBM Corp. Ses caractéristiques ont été entièrement publiées le *US Federal Bureau of Standards* en 1977. Il s'agit donc d'un **algorithme public**. La confidentialité des messages est assurée par une **clé secrète**. C'est la même clé qui sert pour le codage et pour le décodage. DES est donc à **usage restreint**, il faut avoir une clé pour chacun de ses correspondants.

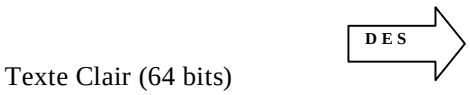
Le DES est certainement le système de crytage **le plus utilisé** au Monde, c'est pourquoi nous allons particulièrement bien développé son fonctionnement.

Précisons enfin que DES est prévu pour fonctionner sur un équipement *hardware* assez coûteux et grâce auquel il est extrêmement rapide ; mais qu'il existe aussi des émulations *softwares* pour PC, environ 1000 fois plus lentes, ce qui reste pourtant tout à fait acceptable.

1.1.2 Fonctionnement du cryptage

Le DES se compose de 16 opérations identiques, sauf la dernière qui est légèrement différente pour permettre le décryptage (DES^{-1}).

Le DES est un **code produit** car il se compose de substitutions (ce qui crée une **confusion**) et de transpositions (ce qui crée une **diffusion**). Cela permet d'obtenir un degré de complexité maximal. Le texte est d'abord coupé par paquets de 64 bits (8 octets).



Rem : En fait, seuls 56 bits sont significatifs : le $8^{ème}$ bit de chaque octet est un bit de parité. Cela permet de vérifier qu'il n'y a pas eu d'erreur de transmission, de calcul…

1.1.2.1 Permutation initiale

On effectue sur le Texte clair entrant une première permutation, appelée **Permutation initiale** (PI), de matrice :

```
58  50  42  34  26  18  10  2
60  52  44  36  28  20  12  4
62  54  46  38  30  22  14  6
64  56  48  40  32  24  16  8
57  49  41  33  25  17  9   1
59  51  43  35  27  19  11  3
61  53  45  37  29  21  13  5
63  55  47  39  31  23  15  7
```

On appelle T_0 le résultat de cette permutation.

1.1.2.2 Séparation en deux

La matrice de PI n'est pas aléatoire. Elle effectue un certain tri : les quatre premières lignes (c'est à dire les 32 premiers bits) correspondent aux **bits pairs** de TC et les 4 dernières lignes aux **bits impairs**. La dernière colonne contient les bits du 1er octet de TC, l'avant dernière colonne contient le 2ème octet de TC, etc.

DES coupe donc T_0 en deux $\mathbf{T_0=G_0 \ \& \ D_0}$; la partie gauche G_0 (resp. droite D_0) contenant les bits pairs (resp. impairs).

1.1.2.3 Fonction d'expansion

D_0 (32 bits) ——— Table E ▶ $E[D_0] = D_0'$ (48 bits)

La sortie est plus grande que l'entrée car il y a redondance de certains bits, comme nous pouvons le constater sur la **table E** :

```
32  1   2   3   4   5
4   5   6   7   8   9
8   9   10  11  12  13
12  13  14  15  16  17
16  17  18  19  20  21
20  21  22  23  24  25
24  25  26  27  28  29
28  29  30  31  32  1
```

1.1.2.4 Combinaison avec la clé

DES applique le ou exclusif XOR avec la première clé K1. Le résultat obtenu s'écrit matriciellement $\mathbf{K1 \oplus E[D_0]}$

1.1.2.5 fonction de sélection

Le résultat, que nous continuons d'appeler D_0, est divisé en 8 blocs de 6 bits (D_{01},D_{02},…,D_{08}) qui entrent chacun dans la **fonction de sélection** (*S-Box*) ou boîte de substitution.

Tableau des S-Box.

Soit $D_{0i} = (a_1\ a_2\ a_3\ a_4\ a_5\ a_6)$ un bloc quelconque.

Le résultat est lu dans le tableau à la ligne n° $a_1\ a_6$ et colonne n° $a_2\ a_3\ a_4\ a_5$

Exemple : 011011 entre. La sortie se trouve ligne $(01)_2 = (1)_{10}$ et colonne $(1101)_2=(15)_{10}$ soit $(8)_2=(111)_2$.

1.1.2.6 Permutation P

```
16  7   20  21
29  12  28  17
1   15  23  26
5   18  31  10
2   8   24  14
32  27  3   9
19  13  30  6
22  11  4   25
```

1.1.2.7 Conclusion

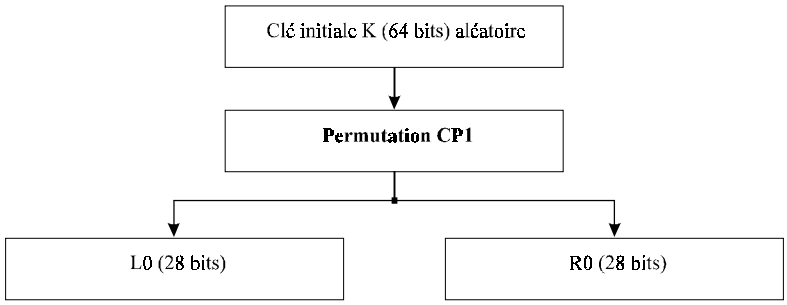
Matriciellement, le résultat s'écrit **: $D_1 \leftarrow P[S[K1 \oplus E[D_0]]]$.** Bien entendu, il ne s'agit pas de multiplications matricielles mais de permutations définies par les matrices correspondantes.

DES réitère ensuite l'opération, en changeant simplement de S-Box et de clé K.

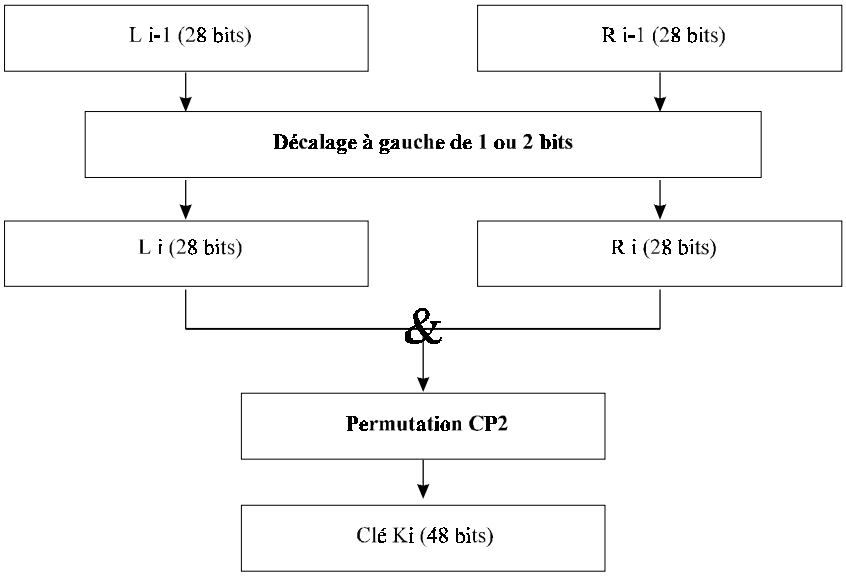
1.1.3 Construction des clés

On construit les clés Ki à partir de la clé K.

Il y a tout d'abord création de L_0 et R_0 à partir de K



Ensuite, pour la création de la ième clé Ki utilise dans l'algorithme principal à la ième itération, on applique :



Pour chaque itération de la fonction DES mais le décalage à gauche se fait successivement de {1 1 2 2 2 2 2 2 1 2 2 2 2 2 2 } bit(s).

1.1.4 Bilan sur DES

En général,

Le résultat de la ième itération est $\mathbf{T_i = L_i \& R_i}$ avec

$\mathbf{G_i = D_{i-1}}$

$\mathbf{D_i = G_{i-1} \oplus F[K_i, D_{i-1}]}$ avec F la fonction principale décrite en 1.1.2.

Mais T_{16} fait exception comme nous l'avions dit.

T_{16} admet $D_{16} = G_{15} \oplus F[K_{16},D_{15}]$ comme partie gauche et $G_{16}=D_{15}$ comme partie droite.

Pour terminer **DES appliquer PI-1** afin que le décryptage utilise le même algorithme.

1.1.5 Fonctionnement du décryptage DES^{-1}

La fonction de décryptage est identique à DES, si ce n'est que les itérations se font « à l'envers ».

Le texte crypté entre, subi la permutation initiale PI. On obtient T_0 composé de G_0 (qui correspond à D_{16} du chiffrement DES) et G_0 (qui correspond à $G_{16}=D_{15}$ de DES). Les 16 itérations se font ainsi dans l'ordre inverse.

Finalement, le résultat est soumis à PI^{-1} et on retrouve le texte originel.

1.1.6 Fiabilité

Le DES était destiné au documents non classés secret défense, ce qui fait immédiatement réfléchir quand à la **sécurité réelle du système**. Il est certain que la NSA (*National Secret Agency*, les services secrets américains) est capable de capter n'importe quel message téléphonique dans le Monde. Est-elle en mesure de déchiffrer ces messages dans un laps de temps raisonnable ?

A priori, il faut tester toutes les clés. Pour une clé de n bits, il y a **2n possibilités**. En utilisant un ordinateur futuriste très rapide (et très cher), capable de tester 1 clé en 1 µs (ce qui 10 fois plus rapide que les machines actuelles) il faut **2285 années** au plus. Mais il ne faut plus que 4,4 min. maximum avec une clé de 28 bits. En effet, à chaque fois que l'on agrandit la clé d'un bit, le temps de recherche double (puisqu'il y a 2^n possibilités). On peut donc considérer que DES est sûr, sa crypto-période est de 1142 ans, temps moyen pour décrypter un message, et au bout de cette période, le message n'a plus aucune valeur.

Parmi les pistes qui ont été explorées pour **casser DES**, il y a la recherche d'une faille dans l'algorithme. En effet, il pourrait très bien exister des **relations linéaires** ou quasi-linéaires entre le message clair et le message crypté. Ce type de relation rendrait DES extrêmement vulnérable. La NSA a demandé à un institut américain spécialiste des algorithmes d'explorer cette voie. Les résultats ont été classés secret défense.

Une autre possibilité serait de faire une **consultation de table**. Le principe est simple : l'espion code un texte qui a une forte probabilité d'être crypté (tel que le nom de la société, l'adresse…) avec toutes les clés possibles qu'il regroupe dans une table. Il ne lui reste ensuite qu'a consulter sa table et comparer avec des messages interceptés, et tester la clé correspondante. Cette méthode pose un très lourd problème de stockage : pour stocker toutes les clés il faudrait déjà $56*2^{56} = 5.10^8$ giga-octets (soit 250 millions de disque-dur de 2 giga-octets, c'est à dire 250 milliards de francs). Cette méthode est donc beaucoup trop chère, même pour la NSA.

Finalement, DES est utilisé depuis plus de 20 ans et il semblerait qu'aucun message n'est jamais été cassé, malgré toutes les tentatives des uns et des autres.

1.1.7 DES 2 ?

Face à l'âge de DES et aux petits problèmes de fiabilité qui sont apparus, un DES 2 pourrait être mis en place. Les améliorations consisteraient à **augmenter le nombre d'itérations** (il n'y en a que 16 pour le moment). Il serait également possible **d'allonger la longueur de la clé**. Mais dès lors, un facteur humain entre en compte. Si la clé (qui doit forcément être pseudo-aléatoire) est trop longue, les utilisateurs risquent de l'écrire sur un support quelconque, augmentant ainsi la possibilité (catastrophique pour la sécurité des données) d'interception de clé par l'ennemi.

2 Clé publique

2.1 Avantages & inconvénients

L'inconvénient du DES et des autres systèmes à clé secrète (ou à usage restreint) et la quantité de clé nécessaire à la confidentialité des échanges. Si Bob a x correspondants, il devra mettre en place x clés afin d'empêcher chacun de ses correspondants de lire les messages émis par les autres correspondants. Globalement la communauté utilisera **x^x clés**.

La solution est apportée par les systèmes à clé publique. Il n'y a que 2 clés : l'une est publique et sert au cryptage, l'autre clé est secrète et sert au décryptage. Globalement, la communauté utilise **2 x clés** (ce qui est très inférieur à x^x).

2.2 Le code RSA

2.2.1 Historique

Le code RSA a été inventé en 1977 au MIT (*Massachusets Institute of Technology*) par les mathématiciens Ronald RIVEST, Adi SHAMIR et Leonard ADLEMAN.

2.2.2 Principe

RSA repose entièrement sur la difficulté qui existe lorsque l'on veut factoriser de grands nombres.

Cryptage $\mathbf{y_i = x_i^{\,e}\ [n]}$

Décyptage $\mathbf{x_i = y_i^{\,d}\ [n]}$

Mais en fait, dans la réalité, on chiffre, non pas caractère par caractère, mais par blocs de caractères. RSA ne conserve donc pas la fréquence d'apparition des lettres, comme notre exemple pourrait le laisser croire.

preuve

$y = x^e\ [n]$

$y^d = x^{ed}\ [n]$

$y^d = x^{(1+k\,n')}$, $k \in$

Théorème de Fermat (petit) : p premier => $\forall a,\ a^{p-1} \equiv 1\ [p]$

Donc $x^{p-1}=1\ [p]$ et $x^{q-1}=1\ [q]$

et donc $x^{(page\ 1)(q-1)}=1\ [pq=n]$

Comme $y^d = x^{1+k\,n'}\ [n]$

$y^d = x * x^{k(\,p-1)(\,q-1)}\ [n]$

$y^d = x$

CQFD

Démonstration du Théorème de Fermat p premier => $a^{p-1} \equiv 1\ [p]$

Montrons tout d'abord que $p \mid C_p^k$, p premier

$\forall k < p,$

$k!(p-k)!\ C_p^k = p!$

$Donc \quad p \mid k!(p-k)!C_p^k$

$\forall l \le k,\ p \wedge l = 1, \forall l \le k, (p-l) \wedge p = 1$

$Donc \quad p \mid C_p^k$

Pour p=2

Il n'y a que 2 classes d'équivalences, les nombres pairs et impairs.

Si n pair, Alors n^p est pair aussi et donc donc n^p de même classe que n.

Si n impair, Alors n^p est impair (car p=2) et donc n^p de même classe que n.

Pour $p \ge 3$

Par récurrence sur n,

pour n=0, c'est évident : 0=0,

On suppose la propriété vraie jusqu'à n. Montrons la pour n+1.

$(n+1)^p = n^p + \sum_{k=1}^{p-1} C_p^k n^k + 1$

$\equiv n^p + 1 \ [p]$ car $p | C_p^k$

$\equiv n + 1 \ [p]$ par hypothèse de récurrence

Généralisation sur Z

$\forall n \in -N, \ n^p = -(-n)^p \equiv -(-n)[p] = n$, car $-n \in N$ et donc $(-n)^p \equiv -n \ [p]$

CQFD

2.2.3 Exemple

On choisit :

p = 101 et q = 103 n= pq = 10403 (public)

n' = 10200

d = 127 (top secret)

Il faut maintenant déterminer e

ed=1[n']

Autrement dit, e d = 1 + k n', k ∈ N

Il faut résoudre l'équation de BEZOUT : k n' – e d = 1 , ici k * 10200 – e * 127

Avec l'algorithme d'Euclide,
10200 = 127*80 + 40
127 = 40*3 + 7
40 = 7 * 5 + 5
5 = 2*2 +1

1 = 5 –2*2
= 3*5 – 2*7
= 3*(40-7*5) – 2*7
= 3*40 –17*(7=127-40*3)
=54*(40=10200-127*80)-17*127
=54*10200 – 4337*127

Donc (k,e)=(54,-4337) est solution

Donc e = -4337 = 5863 [n']

e=5863 (public)

On chiffre « La Terre est bleue comme une orange » :

76 97 32 84 101 114 114 101 32 101 115 116 32 98 108 101 117 101 32 99 111 109 109 101 32 117 110 101 32 111 114 97 110 103 101

en élevant à la puissance e=5863 modulo n=10403 et on obtient (n'importe qui peut donc m'écrire):

9489 10383 6609 1955 1313 5602 5602 1313, etc.

Pour décrypter, on élève à la puissance d=127 modulo n=10403

Effectivement 9489127=76 [10403], etc.

Pour un espion, il est impossible de factoriser n, donc impossible d'obtenir n', et donc impossible de déterminer d en inversant e en modulo n'.

2.2.4 Vitesse

Le code RSA est assez long à mettre en œuvre à cause des puissances modulaires, qui demande beaucoup de calculs.

2.2.5 Sécurité

Tout repose sur la difficulté à factoriser les nombres. L'algorithme de Richard SHROEPPEL, reconnu comme étant le plus performant, demande pour décomposer l'entier n : $\exp\left(\sqrt{\ln(n)\ln(\ln(n))}\right)$ opérations.

Ainsi, il est recommandé d'utiliser des nombres premiers de 200 bits pour obtenir un rapport sécurité/temps de codage optimal.

MAIS :

- Un algorithme de factorisation meilleur peut encore être inventé, la recherche mathématique est assez active dans ce domaine. D'autre part, les ordinateurs quantiques, s'il voient le jour, sont théoriquement capables de factoriser les nombres beaucoup plus rapidement
- Enfin, rien ne prouve qu'il n'existe pas un moyen de décrypter le message sans passer par la factorisation.

Pour conclure, RSA est un système très performant mais également très lent. C'est pourquoi RSA est utilisé pour des informations extrêmement précieuses. Ainsi, le logiciel PGP (*Pretty Good Privacy*) utilise un chiffrement à clé privée rapide (il s'agit de IDEA) et utilise RSA pour communiquer les clés ! Cette méthode est très efficace, car elle permet de changer à chaque étape la clé utilisée par IDEA.

2.3 Authentification

Un autre atout des systèmes à clé publique est qu'ils permettent la signature numérique. Pour cela, on effectue un double codage.

Supposons que Alice envoie un ordre de bourse à sa banque B.

Soient la fonction de cryptage pour Alice (publique) et sa réciproque f_A^{-1} (secrète)

Soient f_B la fonction de cryptage pour la Banque (publique) et sa réciproque f_B^{-1}(secrète)

Alice écrit son ordre de bourse M auquel elle ajoute l'authentification f_A^{-1}(« Alice »). Elle code le tout par f_B et l'envoie à la banque.

La banque reçoit f_B(M& f_A^{-1}(« Alice »)). La banque applique sa fonction secrète f_B^{-1}(f_B(M& f_A^{-1}(« Alice ») et obtient l'ordre de bourse M suivi de l'authentification (qui a l'apparence d'un vrai charabia) c'est à dire f_A^{-1}(« Alice »). Pour s'assurer que c'est bien Alice qui a envoyé le message, la banque effectue le chiffrement grâce à la fonction publique d'Alice. f_A(f_A^{-1}(« Alice »)) donne « Alice ». C'est bien Alice qui a envoyé le message car elle est la seule à connaître la fonction f_A^{-1}.

V Conclusion

Aujourd'hui, seules les méthodes modernes de cryptographie sont utilisées. Mais les méthodes classiques ne sont pas oubliées puisqu'un système comme DES en est une combinaison.

Actuellement, les meilleurs systèmes combinent un chiffrement à clé privée, rapide, dont la clé, pseudo-aléatoire, changée pour chaque message, est transmise de manière sûr après chiffrement par RSA.

Mais le futur repose peut-être sûr le chiffrement quantique, né au début des années 1970. Il repose sur le principe d'incertitude d'Heisenberg, selon lequel la mesure d'un système quantique perturbe ce système. Il serait alors possible de transmettre une clé en étant sûr qu'elle n'a pas été « écoutée », et de l'utiliser ensuite avec un chiffrement habituel.