1. Introduction

Les systèmes de cryptage à clé privée, appelés aussi systèmes de cryptage symétrique ou cryptage conventionnel, sont utilisés depuis plusieurs siècles déjà. C'est l'approche la plus authentique du chiffrement de données et mathématiquement la moins problématique. Dans ce chapitre nous présentons les différents algorithmes utilisés dans le chiffrement des fichiers audio (fichier son). Un expéditeur et un destinataire souhaitant communiquer de manière sécurisée à l'aide du cryptage conventionnel doivent convenir d'une clé et ne pas la divulguer. Dans la majorité des systèmes de cryptage symétrique la clé de chiffrement et la clé de déchiffrement sont identiques.

2. Les algorithmes de cryptage classique

2.1. Le cryptage XOR

Le cryptage XOR est un système de cryptage basique mais pas trop limité. Ainsi, il a beaucoup été utilisé dans les débuts de l'informatique et continue à l'être encore aujourd'hui car il est facile à sortes implémenter, dans toutes de programmes.

- **Mécanisme**

Le XOR est un opérateur logique qui correspond à un "OU exclusif" : c'est le (A OU B) qu'on utilise en logique mais qui exclue le cas où A et B sont simultanément vrais. Voici sa table de vérité :

A	B	(A XOR B)
FAUX	FAUX	FAUX
FAUX	VRAI	VRAI
VRAI	FAUX	VRAI
VRAI	VRAI	FAUX

Tableau 3.1 : Table de vérité du XOR

En informatique, chaque caractère du message à coder est représenté par un entier, le code ASCII. Ce nombre est lui-même représenté en mémoire comme un nombre binaire à 8 chiffres (les bits).

On choisit une clé que l'on place en dessous du message à coder, en la répétant autant de fois que nécessaire, comme dans le cryptage de Vigenère.

Le message et la clé étant converti en binaire, on effectue un XOR, bit par bit, le 1 représentant VRAI et le 0 FAUX. Le résultat en binaire peut être reconverti en caractères ASCII et donne alors le message codé.

L'algorithme est complètement symétrique : la même opération est réappliqué au message final pour retrouver le message initial.

Remarque : Parfois, on applique une permutation circulaire aux bits du message final pour donner le message codé. **[s22]**

Exemple

Voici le mot MESSAGE converti en binaire :

Lettres	**M**	**E**	**S**	**S**	**A**	**G**	**E**
Codes ASCII	77	69	83	83	65	71	69
Binaire	01001101	01000101	01010011	01010011	01000001	01000111	01000101

Tableau 3.2 : exemple avec la méthode de XOR

Le mot CLE en binaire est lui représenté par 01000011 - 01001100 - 01000101.

Message en binaire	01001101	01000101	01010011	01010011	01000001	01000111	01000101
Clé en binaire (répétée si nécessaire)	01000011	01001100	01000101	01000011	01001100	01000101	01000011
Message crypté en binaire	**00001110**	**00001001**	**00010110**	**00010000**	**00001101**	**00000010**	**00000110**

Tableau 3.3 : résultat de cryptage avec la méthode de XOR

2.2. Masque jetable

Le masque jetable est le seul algorithme de cryptage connu comme étant indécryptable. C'est en fait un chiffre de Vigenère avec comme caractéristique que la clef de chiffrement a la même longueur que le message clair. Le système du masque jetable fut inventé par Gilbert Vernam en 1917, puis perfectionné par le major Joseph O. Mauborgne en 1918, qui inventa le concept de clef aléatoire. **[s17]**

Principe

Le chiffrement par la méthode du masque jetable consiste à combiner le message en clair avec une clé présentant les caractéristiques très particulières suivantes :

- La clé doit être une suite de caractères au moins aussi longue que le message à chiffrer.
- Les caractères composant la clé doivent être choisis de façon totalement aléatoire.
- Chaque clé, ou « masque », ne doit être utilisée qu'une seule fois (d'où le nom de *masque jetable*). **[4]**

Exemple

Cette clé est choisie à l'avance entre les deux personnes souhaitant communiquer. Elle n'est connue que d'eux.

On veut **chiffrer** le message « HELLO ». Pour cela, on attribue un nombre à chaque lettre, par exemple le rang dans l'alphabet, de 0 à 25. Ensuite on additionne la valeur de chaque lettre avec la valeur correspondante dans le masque; enfin si le résultat est supérieur à 25 on soustrait 26 (calcul dit "modulo 26") :

7 (H) 4 (E) 11 (L) 11 (L) 14 (O) message
+ 22 (W) 12 (M) 2 (C) 10 (K) 11 (L) masque
= 29 16 13 21 25 masque + message
= 3 (D) 16 (Q) 13 (N) 21 (V) 25 (Z) masque + message modulo 26

Le texte reçu par le destinataire est « DQNVZ ».

Le **déchiffrement** s'effectue de manière similaire, sauf que l'on soustrait le masque au texte chiffré au lieu de l'additionner. Ici encore on ajoute éventuellement 26 au résultat pour obtenir des nombres compris entre 0 et 25 :

3 (D) 16 (Q) 13 (N) 21 (V) 25 (Z) message chiffré
- 22 (W) 12 (M) 2 (C) 10 (K) 11 (L) masque
= -19 4 11 11 14 message chiffré - masque
= 7 (H) 4 (E) 11 (L) 11 (L) 14 (O) message chiffré - masque modulo 26

On retrouve bien le message initial **« HELLO ». [s11]**

2.3. Le chiffrement par substitution

Le chiffrement par substitution consiste à remplacer dans un message une ou plusieurs entités (généralement des lettres) par une ou plusieurs autres entités.

On distingue généralement plusieurs types de crypto systèmes par substitution:

- La ***substitution monoalphabétique*** *:* consiste à remplacer chaque lettre du message par une autre lettre de l'alphabet

- La ***substitution polyalphabétique*** : consiste à utiliser une suite de chiffres monoalphabétique réutilisée périodiquement
- La ***substitution homophonique*** : permet de faire correspondre à chaque lettre du message en clair un ensemble possible d'autres caractères
- La ***substitution de polygrammes :*** consiste à substituer un groupe de caractères (polygramme) dans le message par un autre groupe de caractères . **[3]**

2.4. Chiffrement par transposition

Les méthodes de chiffrement par transposition consistent à réarranger les données à crypter de telle façon à les rendre incompréhensibles. Il s'agit généralement de réarranger géométriquement les données pour les rendre visuellement inexploitables.

❖ La technique assyrienne

Cette technique de cryptage est vraisemblablement la première preuve de l'utilisation de moyens de chiffrement en Grèce dès 600 avant Jésus Christ pour dissimuler des messages écrits sur des bandes de papyrus.

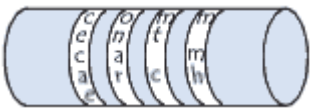

Figure 3.1 : La technique assyrienne **[s13]**

La technique consistait à:

- enrouler une bande de papyrus sur un cylindre appelé **scytale**
- écrire le texte longitudinalement sur la bandelette ainsi enroulée (le message dans l'exemple ci-dessus est "comment ça marche")

Le message une fois déroulé n'est plus compréhensible ("cecaeonar mt c m mh "). Il suffit au destinataire d'avoir un cylindre de même rayon pour pouvoir déchiffrer le message. en réalité un casseur (il existait des casseurs à l'époque...) peut déchiffrer le message en essayant des cylindres de diamètre successifs différents, ce qui revient à dire que la méthode peut être cassée statistiquement (il suffit de prendre les caractères un à un, éloignés d'une certaine distance). **[s13]**

3. Algorithme de chiffrement par Les Nombres Principaux Nobles (NPN)

NPN peut être employé pour chiffrer les fichiers audio. Le procédé de chiffrage se compose de trois étapes. La première étape est de trouver le NPN droit par le prise en compte de taille de données dans le fichier audio. Est en second lieu choisir le NPN dont les données seront mélangées. La troisième étape est de tourner les données par un nombre moins que le NPN courant. Le mélange en résultant alors est écrit au fichier de cible. Au-dessous, il y a une explication détaillée. **[s24]**

Exemple

Le nombre principal noble choisi est ***19***. **A** est la rangée originale et contient quelques caractères.

A =

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
A	B	C	D	E	F	G	H	I	J	K	L	M	N	P	Q	R	S	T

1 mod 19 = {1,10,5,12,6,3,11,15,17,18,9,14,7,13,16,8,4,2,1} **(s)**

Employer **(s)** la rangée de mélange construite est {1,10,5,12,6,3,11,15,17,18,9,14,7,13,16,8,4,2,19}

Rangée mélangée **A** est : {A,J,E,L,F,C,K,P,R,S,I,N,G,M,Q,H,D,B,T}

Alors la position de rotation est choisie. Ici elle est 9

Nous pouvons dire $\left\lfloor \frac{19}{2} \right\rfloor = 9$ .Tellement 9 est choisis pour la rotation.
Rangée tournée **A** = {S, I, n, g, m, q, h, d, b, t, a, j, e, l, f, c, k, p, r}

La rangée tournée **A** est la rangée chiffrée. **[s24]**

3.1. La structure de l'algorithme de NPN

Variables

Asil[] : tableau des nombres principales nobles NPN.
src[] : ficher audio source.
dst[] : fichier audio sortie
Asil_2[] : contient les sélective résidus nombres NPN.
inD[] : donnée original de fichier audio.
mix1[] : Contient les données mélange dans le processus de cryptage.
mix2[] : contient les données mélanges de l'opération de rotation.

Asil_Asal : le nombre NPN dans le tableau Asil[].

as : le nombre NPN sélectionné.
d_len : la taille de la partie dans le fichier audio. **[s24]**

3.2. Algorithme NPN

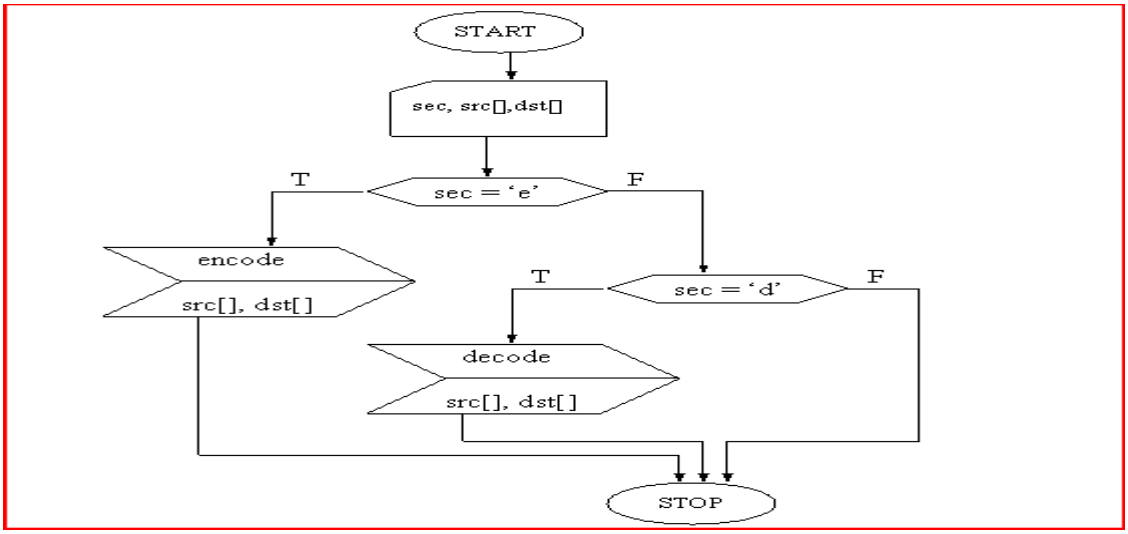

Figure 3.2 : algorithme NPN. **[s24]**

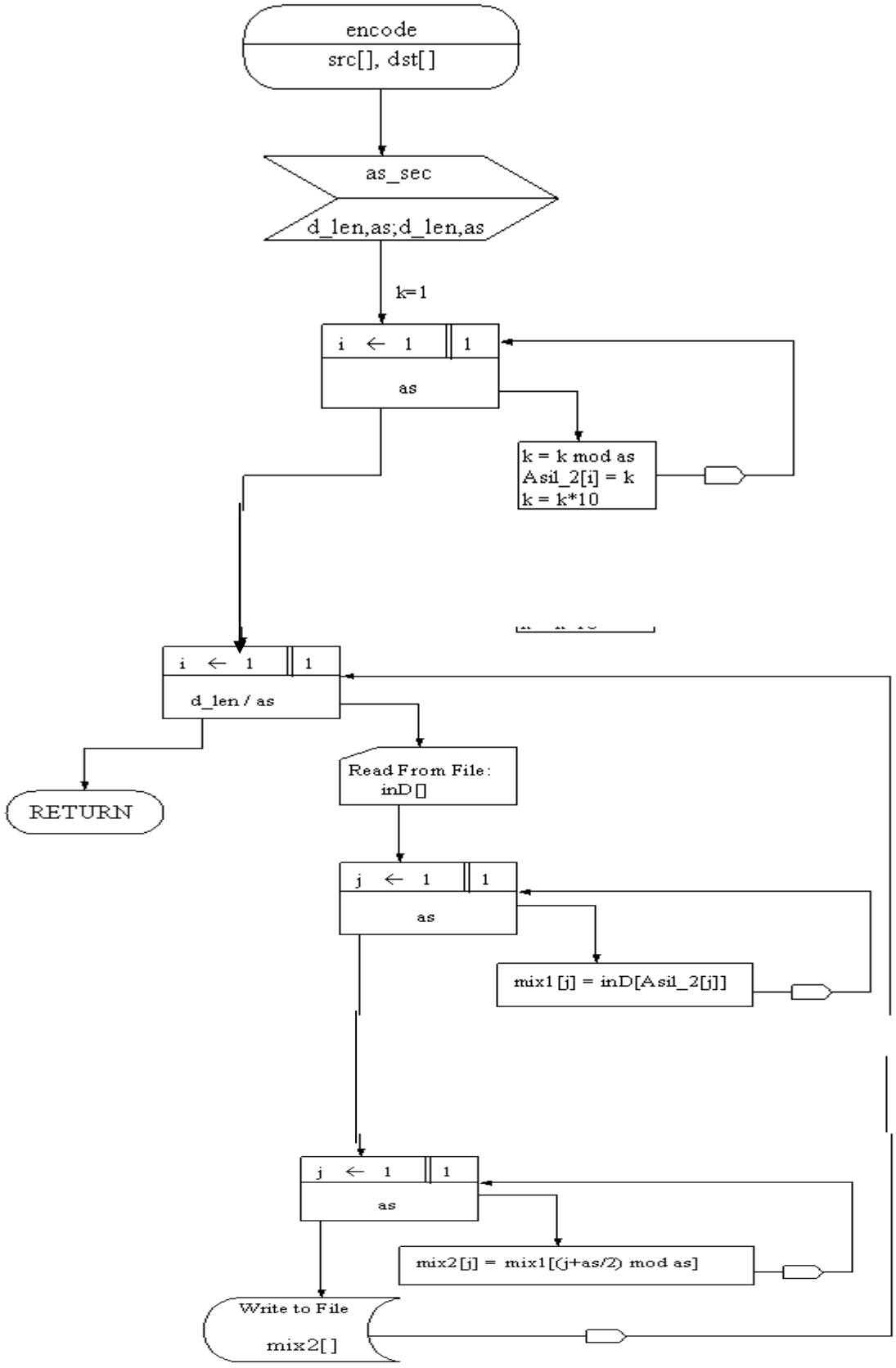

Figure 3.3 : opération de cryptage par l'algorithme NPN. **[s24]**

4. Algorithme de chiffrement à clé secrète DES (DATA Encryption Standard)

Le 15 mai 1973 le **NBS** (*National Bureau of Standards*, aujourd'hui appelé *NIST - National Institute of Standards and Technology*) a lancé un appel dans le *Federal Register* (l'équivalent aux Etats-Unix du *Journal Officiel* en France) pour la création d'un algorithme de chiffrement répondant aux critères suivants :

- posséder un haut niveau de sécurité lié à une clé de petite taille servant au chiffrement et au déchiffrement
- être compréhensible
- ne pas dépendre de la confidentialité de l'algorithme
- être adaptable et économique
- être efficace et exportable

Fin 1974, IBM propose « Lucifer », qui, grâce à la NSA (National Security Agency), est modifié le 23 novembre 1976 pour donner le **DES** (*Data Encryption Standard*). Le DES a finalement été approuvé en 1978 par le NBS. Le DES fut normalisé par l'*ANSI* (*American National Standard Institute*) sous le nom de *ANSI X3.92*, plus connu sous la dénomination *DEA* (*Data Encryption Algorithm*). **[s20]**

4.1. Principe du DES

Il s'agit d'un système de chiffrement symétrique par blocs de 64 bits, dont 8 bits (un octet) servent de test de parité (pour vérifier l'intégrité de la clé). Chaque bit de parité de la clé (1 tous les 8 bits) sert à tester un des octets de la clé par parité impaire, c'est-à-dire que chacun des bits de parité est ajusté de façon à avoir un nombre impair de '1' dans l'octet à qui il appartient. La clé possède donc une longueur « utile » de 56 bits, ce qui signifie que seuls 56 bits servent réellement dans l'algorithme.

L'algorithme consiste à effectuer des combinaisons, des substitutions et des permutations entre le texte à chiffrer et la clé, en faisant en sorte que les opérations puissent se faire dans les deux sens (pour le déchiffrement). La combinaison entre substitutions et permutations est appelée code produit.

La clé est codée sur 64 bits et formée de 16 blocs de 4 bits, généralement notés k_1 à k_{16}. Etant donné que « seuls » 56 bits servent effectivement à chiffrer, il peut exister 2^{56} (soit $7.2*10^{16}$) clés différentes !

4.2. L'algorithme du DES

Les grandes lignes de l'algorithme sont les suivantes :

- Fractionnement du texte en blocs de 64 bits (8 octets) ;
- Permutation initiale des blocs ;
- Découpage des blocs en deux parties: gauche et droite, nommées *G* et *D* ;
- Etapes de permutation et de substitution répétées 16 fois (appelées **rondes**) ;
- Recollement des parties gauche et droite puis permutation initiale inverse. **[s23]**

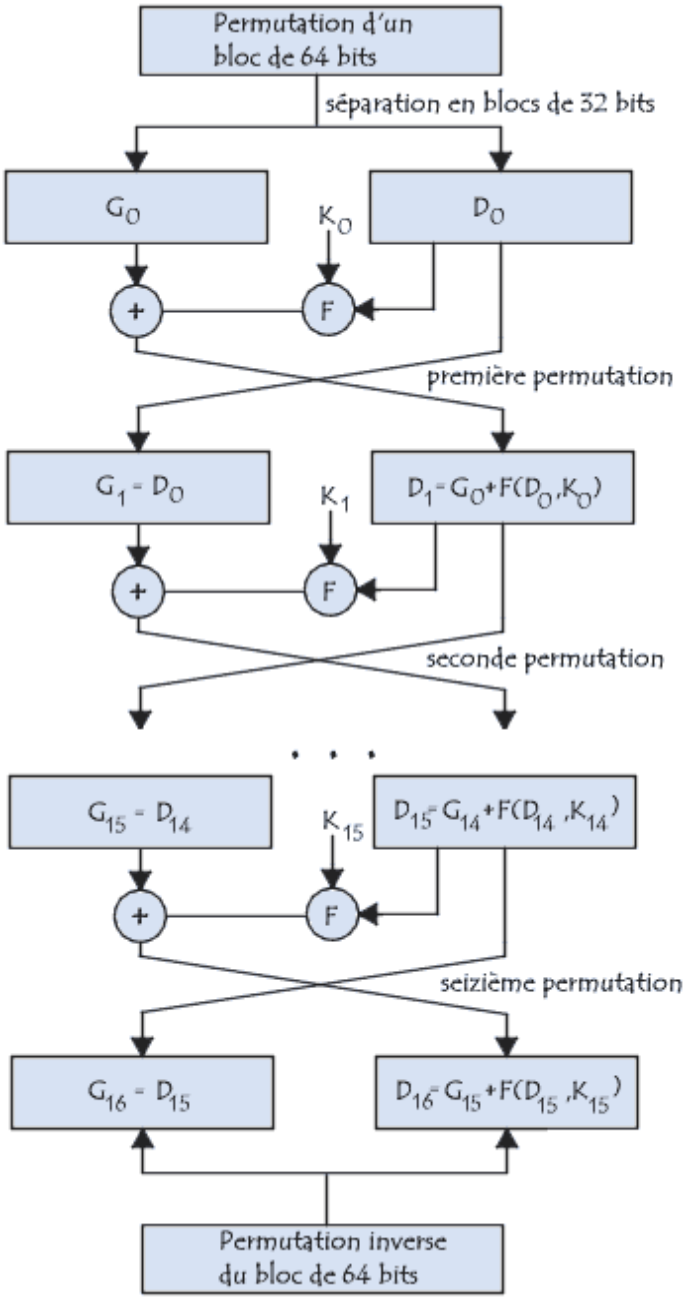

Figure 3.4 : cryptage par DES **[s23]**

5. algorithme à clé publique RSA

5.1. 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 **R**ivest, Adi **S**hamir et Leonard **A**dleman de l'Instituion de technologie du Massachusetts, 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**

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

1- 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)

2 - Chiffrement : le chiffrement d'un message M en un message codé C se fait suivant la transformation suivante :

$C = M^e \bmod n$

3 - Déchiffrement : il s'agit de calculer la fonction réciproque

$M = C^d \bmod n$

tel que $e.d = 1 \bmod [(p-1)(q-1)]$

5.2. Fonctionnement de RSA

Le fonctionnement du cryptosystème RSA est basé sur la difficulté de factoriser de grands entiers.

Soit deux nombres premiers *p* et *q*, et *d* un entier tel que *d* soit premier avec *(p-1)*(q-1)*). Le triplet *(p,q,d)* constitue ainsi la clé privée.

La clé publique est alors le doublet (n,e) créé à l'aide de la clé privée par les transformations suivantes :

n = p * q
e = 1/d mod((p-1)(q-1))

Soit *M*, le message à envoyer. Il faut que le message *M* soit premier avec la clé *n*. En effet, le déchiffrement repose sur le théorème d'Euler stipulant que si *M* et *n* sont premiers entre eux, alors :

$M^{phi(n)} = 1 \bmod(n)$
Phi(n) étant l'indicateur d'euler, et valant dans le cas présent *(p-1)*(q-1)*.

Il est donc nécessaire que M ne soit pas un multiple de *p*, de *q* , ou de *n*. Une solution consiste à découper le message *M* en morceaux M_i tels que le nombre de chiffres de chaque M_i soit strictement inférieur à celui de *p* et de *q*. Cela suppose donc que p et q soient grand, ce qui est le cas en pratique puisque tout le principe de RSA réside dans la difficulté à trouver dans un temps raisonnable *p* et *q* connaissant *n*, ce qui suppose *p* et *q* grands**. [s18]**

Exemple : 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 $\mathbf{C = B^e \bmod n}$ (où C est le bloc chiffré) :

$C_1 = 2^7 \bmod 5141 = 128$
$C_2 = 151^7 \bmod 5141 = 800$
$C_3 = 410^7 \bmod 5141 = 3761$
$C_4 = 152^7 \bmod 5141 = 660$
$C_5 = 118^7 \bmod 5141 = 204$

On obtient donc le message chiffré C : 128 800 3761 660 204. **[s23]**

6. Conclusion

Dans ce chapitre nous avons présenté quelques algorithmes de chiffrements les plus utilisés, mais bien sure il en existe d'autres. Nous somme intéressées dans notre projet de fin d'étude par les méthodes de chiffrement classique comme substitution et les nombres principales nobles (NPN) plus le chiffrement a clé sécrète et le chiffrement à clé publique comme RSA. Ces différents algorithmes de chiffrement seront appliqués pour crypter des fichiers audio non compressé de type wave. Le prochain chapitre sera consacré pour les caractéristiques des formats des fichiers wave.