# Leçon 170 : Méthodes de chiffrement ou de codage. Illustrations.

## 1) Introduction

Il faut distinguer les 2 notions proposées par cette leçon (même si le langage courant a tendance à les confondre) :

- ☞ Le **codage** considère le fait de **représenter l'information**. L'objectif n'est pas de la cacher, toute personne connaissant ce codage peut obtenir l'information.
- ☞ Le **chiffrement** est une des possibilités permettant de **cacher l'information**, au même titre que la *stéganographie* (dissimuler l'information en pleine vue) ou l'*offuscation* (obscurcir le sens d'un message pour décourager, exemple : *TrackMeNot*, extension Firefox). Ici on modifie le message de manière à ce que (dans l'idéal), seul le destinataire soit en mesure de comprendre l'information.

Remarquons que ces deux sujets sont extrêmements vastes et qu'il est donc nécessaire de faire des choix.

## 2) Codage

### a. Premiers exemples

- ☞ ASCII : 128 codes à 7 bits, dont 95 caractères imprimables (0-9, a-z, A-Z, symboles mathématiques et ponctuation)
- ☞ Images (pixels, RGB)
- ☞ Modèles 3D (surfaces, triangles)

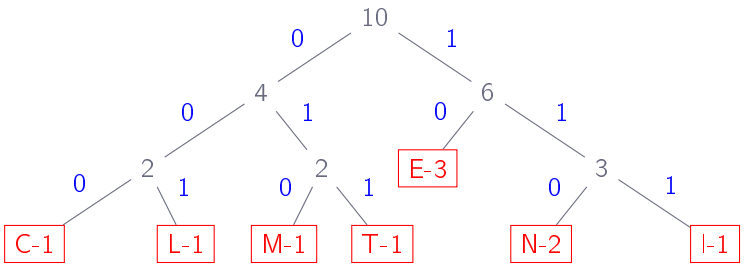
### b. Compression

- ☞ JPEG : algorithme de compression des images avec perte.
- ☞ **Huffman** : algorithme de compression des données sans perte. Le code est déterminé à partir d'une estimation probabiliste de la fréquence d'apparition des différents caractères, basée sur une structure d'arbre.

☞ **Exemple(s) :**

Dans le mot « CLEMENTINE » :

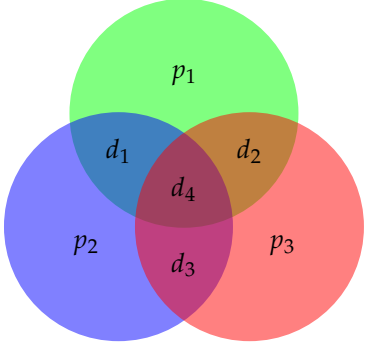
| Caractère | C | L | E | M | N | T | I |
|---|---|---|---|---|---|---|---|
| Effectif | 1 | 1 | 3 | 1 | 2 | 1 | 1 |




« CLEMENTINE » est donc codé « 000 001 10 010 10 110 011 111 110 10 » donc avec 27 bits (au lieu de 70 en ASCII).

### c. Code détecteur et/ou correcteur d'erreurs

#### i. Hamming (1950, DÉV)




Permet la détection et la correction d'1 bit erroné (si plus d'une erreur, le résultat sera incohérent). On s'intéresse en particulier au cas (7, 4) : transmission d'un message de longueur 4 bits avec 3 bits de parité.

L'idée se représente via le diagramme ci-contre : dans chaque cercle, la somme des 4 bits doit être nulle (paire). Si ce n'est pas le cas, on peut trouver le bit incorrect selon les valeurs des 3 cercles. Par exemple si le bit $d_2$ est erroné, ce sont les sommes des cercles vert et rouge qui seront impactés, mais pas le bleu.

### ii. Reed-Solomon (1960)

Permet la détection et la correction d'erreurs. Utilisé par les QR-Codes. Un code $RS(n,k,t)$ avec $n = k+2t$ contient $n$ bits au total, dont $k$ bits de message, et peut corriger $t$ erreurs (exemple transmission vidéo par satellite : RS(204, 188, 8)), et on détecte (sans pouvoir corriger) s'il y a plus de $t$ erreurs.

Pour un message $m = (m_1,\dots,m_k)$, on définit le polynôme $P_m(X) = \sum_{i=1}^{k} m_i X^{i-1}$ puis on évalue ce polynôme en $n$ points $x = (x_1,\dots,x_n)$ qui formeront le message codé.

## 3) Chiffrement

### a. Substitution mono et poly-alphabétique

L'exemple le plus simple est le chiffrement par décalage, par exemple ROT $a$ consiste à prendre $f(x) = x+a$ où $a \in \mathbb{Z}/26\mathbb{Z}$. Par exemple :

$$\text{CLEMENTINE} \xrightarrow{\text{ROT 13}} \text{PYRZRAGVAR}$$

Un exemple de chiffrement poly-alphabétique est donné par le chiffre de Vigenère, où chaque lettre est décalée selon une clé :

$$\begin{matrix}\text{CLEMENTINE} \\ \text{CLECLECLEC}\end{matrix} \xrightarrow{\text{Vigenère de clé CLE}} \text{EWIOPRVTRG}$$

### b. RSA (Rivest, Shamir, Addleman, 1977)

**Propriété 1 : Algorithme de cryptographie asymétrique RSA**

On note $\varphi$ l'indicatrice d'Euler. Soient $p,q$ premiers distincts et $n = pq$.

S'il existe $e,d \in \mathbb{Z}$ tels que $ed \equiv 1[\varphi(n)]$, alors pour tout $m \in \mathbb{Z}$, $m^{ed} \equiv m[n]$.

☞ **Exemple(s) :**

Alice veut envoyer un message $m$ à Bob. Bob détermine sa clé privée $d$ et sa clé publique $(e,n)$, qu'il communique. On note qu'il est le seul à pouvoir connaître facilement $\varphi(n)$.

$$\begin{matrix}\text{Alice} \\ c \equiv m^e[n]\end{matrix} \longrightarrow \begin{matrix}\text{Bob} \\ m \equiv c^d[n] \equiv m^{ed}[n]\end{matrix}$$

Voir annexe « Illustration RSA »

### c. El Gammal

Dans le même principe que RSA, on utilise le fait que le *logarithme discret* (réciproque de l'exponentielle dans un groupe cyclique) est difficile à calculer (contrairement à l'exponentielle).

**Sources :**

☞ [DUM] Dumas, Roch, Tannier, Varette. (2018). *Théorie des codes : Compression, cryptage, correction* (3e édition). Dunod.

☞ Wikipédia...