

## Chapitre 4. Application des codes correcteurs en Cryptographie.

### 4.1 Notions de Cryptographie.

**Définition 4.1.** Un système de cryptographie ou cryptosystème est constitué de:

- 1- Un alphabet  $A$ . En pratique  $A = \{0, 1\}$ . Les éléments de  $A$  sont dits *symboles*.
- 2- Un ensemble  $M$  composé de chaînes de symboles de l'alphabet  $A$  appelé *espace de messages clairs*. Un élément de  $M$  est appelé *texte clair*.
- 3- Un ensemble  $C$  constitué de chaînes de symboles d'un alphabet  $B$ , qui peut être différente de l'alphabet  $A$ , appelé *espace de messages chiffrés* ou *cryptogrammes*.
- 4- Un ensemble  $K$  dit *espace des clés*. Un élément  $e$  de  $K$  est dit clé.
- 5- Pour chaque clé  $e$  de  $K$ , on définit une bijection  $f_e$  de  $M$  dans  $C$ , dite *fonction de chiffrement*. Si  $m \in M$  alors  $f_e(m) = c \in C$ .
- 6- Pour chaque clé  $d$  de  $K$ , on définit une bijection  $f_d$  de  $C$  dans  $M$ , dite *fonction de déchiffrement*. Si  $c \in C$  alors  $f_d(c) = m \in M$ .

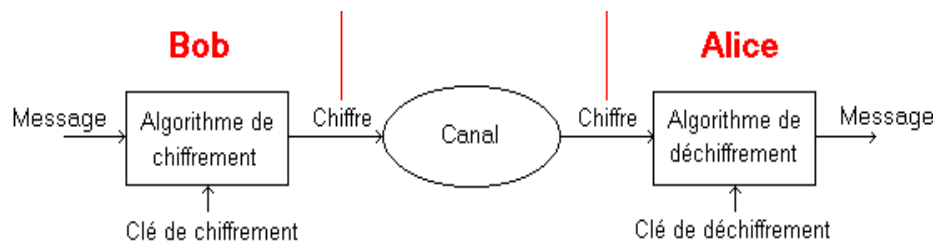


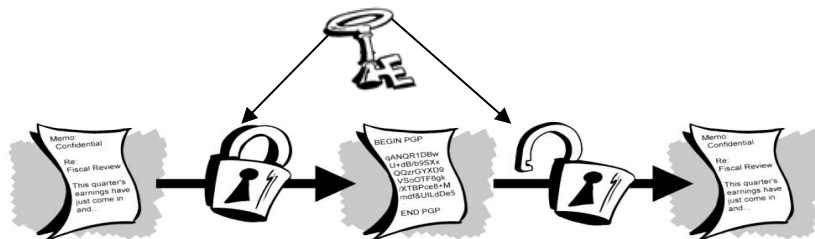
Schéma **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document..**1 Schéma général d'un cryptosystème.

Il existe deux types de cryptographie:

#### 4.1.1 Cryptographie symétrique.

**Définition 4.2.**

En cryptographie *symétrique*, également appelée cryptographie à *clé secrète*, une seule clé suffit pour le chiffrement et le déchiffrement. les clés  $e$  et  $d$  sont identiques.

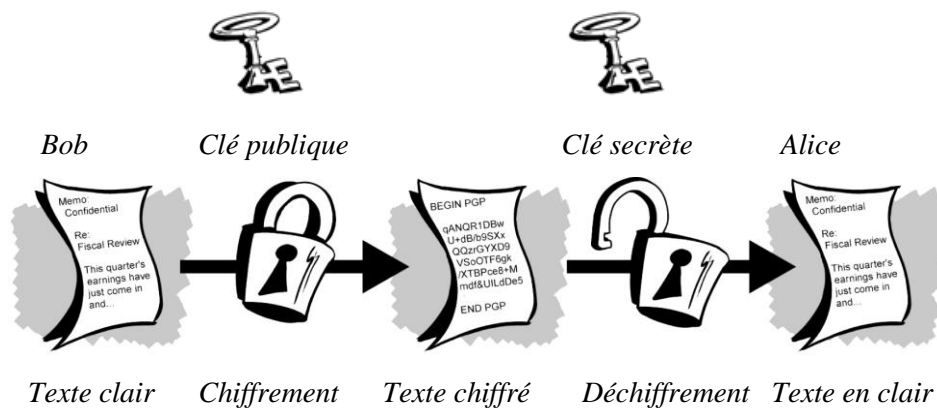


Shéma **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**2 Chiffrement et déchiffrement symétrique.

#### 4.1.2 Cryptographie asymétrique.

##### Définition 4.3.

En cryptographie *asymétrique*, également appelée cryptographie à *clé publique*, admet deux clés, une clé  $e$  (dite *publique*) sert pour le chiffrement et une autre clé  $d$  différente de la clé  $e$  (dite *secrète*) sert pour le déchiffrement.



Shéma **Erreur ! Il n'y a pas de texte répondant à ce style dans ce document.**3 Chiffrement et déchiffrement asymétrique.

#### 4.2. Cryptosystème de McEliece.

C'est le plus ancien cryptosystème à clé publique utilisant des codes correcteurs d'erreurs. Il a été imaginé par McEliece en 1978, à peu près en même temps que le chiffrement RSA. Comme tous les cryptosystèmes à clé publique, ce cryptosystème est constitué de trois algorithmes:

1. La génération des espace de clés,
2. Le chiffrement (utilisant la clé publique) et fonctions de chiffrement.
3. Le déchiffrement (utilisant la clé secrète) et fonctions de déchiffrement.

##### 1. Génération de clé

On commence par générer un code linéaire  $C(n, k, d)$  et sa matrice génératrice  $G$  de taille  $k \times n$ . On va mélanger cette matrice pour la rendre indistinguable d'une matrice aléatoire, pour cela on a besoin de :

- a. Une matrice de *permutation aléatoire*  $P_\sigma$  de taille  $n \times n$  associée à une permutation  $\sigma$  de  $S_n$ .
- b. Une matrice inversible *aléatoire*  $S$  de taille  $k \times k$ .

La clé publique sera le couple  $(G', e)$  tel que  $G' = S.G$ .  $P_\sigma$  qui est indistinguable d'une matrice aléatoire et  $e$  la capacité de correction de C.

La clé secrète est composée des trois matrices  $S$ ,  $P_\sigma$  et  $G$  qui permettent de retrouver la structure du code C et donnent donc accès à l'algorithme de décodage.

## 2. Chiffrement.

Soit  $m$  un message de  $k$  bits que l'on veut chiffrer. On ne dispose pour cela que de la clé publique  $G'$ . On commence par calculer le mot de code C de longueur  $n$  associé à  $m$  :  $c = m.G'$ . Ensuite on génère une *erreur aléatoire*  $\varepsilon$  de longueur  $n$  et de poids  $t \leq e$ . Le texte chiffré sera simplement le mot bruité :  $c' = c + \varepsilon$ .

**3. Déchiffrement.** Pour déchiffrer le texte  $c'$ , en connaissant  $P_\sigma$ ,  $S$  et  $G$ , il suffit de calculer :  $r' = c'. P_\sigma^{-1} = m.G'. P_\sigma^{-1} + \varepsilon. P_\sigma^{-1} = m.S.G + \varepsilon. P_\sigma^{-1}$ . Le mot  $r = m.S.G$  est un mot du code C et  $\varepsilon' = \varepsilon. P_\sigma^{-1}$  est une erreur de poids  $t$  (car  $P$  est une permutation et conserve donc le poids des mots), donc on peut corriger cette erreur et retrouver le message initial  $m' = m.S$  et le message clair  $m = m'.S^{-1}$ .

**Exemple 4.5.1.** Soit  $C(n, 4, d)$  un codes linéaire binaire de longueur  $n=7$  de matrice génératrice

$$G = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

En appliquant la méthode de Gauss sur les lignes de  $G$  on trouve la matrice génératrice normalisé  $G_N$ ,

$$G_N = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix} \text{ et donc C admet comme matrice génératrice } H = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 1 \end{pmatrix}$$

On déduit la distance  $d=3$  et donc la capacité  $e=1$ .

Soit  $m = 1010 \in \mathbb{F}_2^4$  un message clair. Supposons que Bob veut envoyer ce message à Alice.

1. **Génération des clés.** Alice génère les clés suivants :

a. La clés secrète : La matrice normalisée  $G_N$ , une matrice de permutation  $P_\sigma$  de type  $n \times n$ , une matrice inversible et aléatoire  $S$  de type  $k \times k$  par exemple on choisit:

$$P_\sigma = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix} \text{ et } S = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

b. La clé publique est le couple  $(G', e)$  tel que  $G' = S.G_N$ .  $P_\sigma = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 0 & 1 & 0 & 0 \end{pmatrix}$  et la capacité  $e=1$ .

**Le Chiffrement.** Bob chiffre le message  $m$  en calculant  $c' = mG' + \varepsilon$  avec par exemple l'erreur  $\varepsilon = 0100000 \in \mathbb{F}_2^7$  de poids  $w(\varepsilon) = 1 \leq e = 1$ . Donc le texte chiffré est  $c' = mG' + \varepsilon = 0111001 + 0100000 = 0111001$ .

### Le déchiffrement

On a  $c' = mG' + \varepsilon = mS.G_N$ .  $P_\sigma + \varepsilon \Rightarrow r' = c'$ .  $P_\sigma^{-1} = m.S.G_N + \varepsilon$ .  $P_\sigma^{-1} = m' + \varepsilon'$  tel que  $m' \in C$  et

$w(\varepsilon') = 1$  (car  $P_\sigma$  est une permutation et conserve donc le poids des mots)

On calcule  $r' = c'$ .  $P_\sigma^{-1} = 0111001$ . 
$$\begin{pmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{pmatrix} = 1110100. \quad r' \text{ est un mot entaché d'erreurs.}$$

En utilisant la méthode de décodage par syndrome. On peut corriger le mots  $r'$  d'erreur  $\varepsilon'$ .

Le syndrome de  $r'$  est  $h(r') = C_1 + C_2 + C_3 + C_5 = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix} = C_1 = h(\varepsilon_1) =$

$h(1000000)$  et  $w(\varepsilon_1) = 1$ . L'erreur est donc  $\varepsilon_1 = 1000000$ , et le mot code est  $r = m.S.G_N = r' + \varepsilon_1 = 1110100 + 1000000 = 0110100$ .

Le décodage de  $r$  (en enlevant la redondance) on obtient le mot  $m' = mS = 0110$  et donc le message

initial  $m = m'.S^{-1} = 0110$ . 
$$\begin{pmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = 1010. \quad \text{Qui est le message clair.}$$

**Exemple 4.5.2.** ( $G$  une matrice génératrice quelconque) Soit  $C$  ( $n=15, k=7, d$ ) un code linéaire binaire de matrice génératrice

$$G = \begin{pmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{et comme matrice de contrôle}$$

$$H=(M/I_3) \text{ tel que } M=\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$$

On trouve la distance  $d \geq 3$  et donc la capacité  $e \geq 1$ .

Soit  $m = 1010110 \in \mathbb{F}_2^7$  un message clair. Supposons que Bob veut envoyé ce message à Alice.

**2. Génération des clés.** Alice génère les clés suivants :

La clés secrète : constituée de la matrice  $G$ , une matrice de permutation  $P_\sigma$  de type  $15 \times 15$ , une matrice inversible et aléatoire  $S$  de type  $7 \times 7$  par exemple on choisit:

Pour  $P_\sigma$  la matrice de permutation associee à la permutation

$$\sigma = (3,11,4,6,13,5,14,2,10,9,12,7,8,1,15) \text{ et on choisit } S=\begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \end{pmatrix}$$

$$\text{On trouve la clé publique: } G'=\begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 \end{pmatrix}$$

et on choisit comme vecteur d'erreur par exemple  $\varepsilon=000000000010000$  tel que  $w(\varepsilon)=1 \leq e$ .

**Le Chiffrement.** Bob chiffre le message  $m$  en calculant  $c'=mG'+\varepsilon$  de poids  $w(\varepsilon)=1$ .

On a  $c'=m.G'+\varepsilon=100100110100000$  ( $c'$  est le message chiffré).

**Le déchiffrement.**

On a  $c'=m.G'+\varepsilon=m.S.G.P_\sigma+\varepsilon \Rightarrow c=c'. P_\sigma^{-1}=m.S.G+\varepsilon. P_\sigma^{-1}=m'+\varepsilon'$  tel que  $m' \in C$  et  $w(\varepsilon')=1$  (car  $P_\sigma$  est une permutation et conserve donc le poids des mots). On a  $r'=c'. P_\sigma^{-1}=001000001001110$ .

En utilisant la méthode de décodage par syndrome. On peut corriger le mot  $r'$  d'erreur  $\varepsilon'$ .

Le syndrome de  $r'$  est  $h(r') = C_3 + C_9 + C_{12} + C_{13} + C_{14} = C_2 = h(\varepsilon') = h(0100000000000000)$

Et  $w(\varepsilon') = 1$ . L'erreur est donc  $\varepsilon' = 0100000000000000$ . Et le mot code est  $r = m.S.G = c + \varepsilon'$ ,

$r = 001000001001110 + 0100000000000000 = 011000001001110$ .

Or  $r = m.S.G = 011000001001110$ .  $r = m.S.(A/B)$  d'où  $m' = m.S.A$ .

Le décodage de  $m'$ . On a:  $m' = m.S.A = 0110000$  et donc et donc  $m = m'.A^{-1}.S^{-1}$ , le message initial

$$m = m'.A^{-1}.S^{-1} = (0110000) \cdot \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix} = 1010110. \text{ Qui est le mot clair.}$$