Chapitre 5 Application des codes cycliques à la cryptographie.

5.1 Introduction

L'application des codes correcteurs en cryptographie représente une convergence fascinante entre la correction d'erreurs et la sécurité de l'information. La cryptographie, qui concerne la science de la sécurisation des données et des communications, joue un rôle crucial dans notre monde numérique interconnecté. Les codes correcteurs d'erreurs, quant à eux, sont des outils mathématiques qui permettent de détecter et de corriger les erreurs dans les données transmises ou stockées. Lorsque ces deux domaines se rejoignent, cela crée une puissante alliance pour protéger les informations confidentielles, garantir l'intégrité des données et sécuriser les communications.

Les codes correcteurs d'erreurs sont essentiellement utilisés pour garantir que les données transmises ou stockées restent intactes, même dans des environnements sujets à des perturbations, des interférences ou des attaques. Ils fonctionnent en ajoutant une certaine quantité de redondance aux données d'origine, ce qui permet de détecter et, dans de nombreux cas, de corriger les erreurs. Cependant, leur application en cryptographie va au-delà de la simple correction d'erreurs accidentelles. Les codes correcteurs d'erreurs sont intégrés dans des protocoles cryptographiques pour diverses raisons :

1. **Confidentialité**: Lors de la transmission de données sensibles, il est essentiel de s'assurer qu'elles ne peuvent pas être interceptées ou comprises par des tiers non autorisés. Les codes correcteurs d'erreurs peuvent être utilisés pour ajouter un niveau de confidentialité supplémentaire en cryptant les données avant de les transmettre.
2. **Intégrité des données**: La modification ou la corruption de données est une menace courante en ligne. Les codes correcteurs d'erreurs permettent de détecter ces altérations et, dans certains cas, de les corriger automatiquement pour garantir l'intégrité des données.
3. **Authentification**: Les codes correcteurs d'erreurs peuvent être utilisés pour générer des signatures numériques, qui sont des empreintes digitales cryptographiques associées à des messages. Ces signatures permettent de vérifier l'authenticité des messages et de s'assurer qu'ils proviennent bien de la source prétendue.

4. **Résilience aux attaques**: Les codes correcteurs d'erreurs renforcent la résistance aux attaques par force brute et à d'autres attaques cryptographiques. Ils rendent plus difficile pour un attaquant de manipuler les données sans être détecté.
5. **Sécurisation des canaux de communication**: Lors de la transmission de données via des canaux non fiables, comme Internet, les codes correcteurs d'erreurs garantissent que les données atteignent leur destination de manière fiable, même en présence de bruit ou d'interférences.

En résumé, l'application des codes correcteurs d'erreurs en cryptographie est une discipline essentielle pour sécuriser les données et les communications dans le monde numérique. Elle repose sur les principes de la correction d'erreurs pour garantir la confidentialité, l'intégrité, l'authenticité et la fiabilité des informations échangées, renforçant ainsi la sécurité des transactions financières, des communications gouvernementales, de la confidentialité des données personnelles et de bien d'autres aspects de notre vie quotidienne numérique.

5.2 Notions de Cryptographie.

Définition 5.1.

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 est 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 est appelé **espace de messages chiffrés** ou **cryptogrammse**.
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 $f_d(c)=m \in M$.

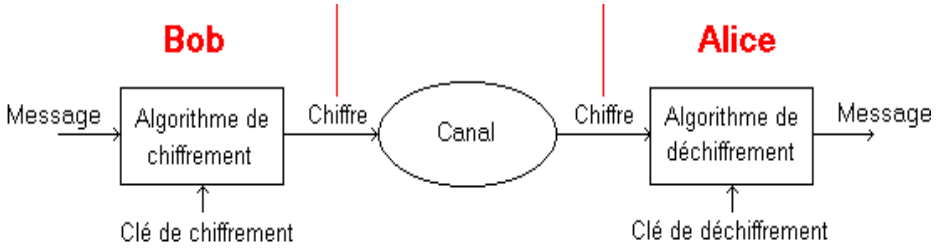

Fig 5.1 Schéma général d'un cryptosystème.

Il existe deux types de cryptographie:

5.2.1 Cryptographie symétrique.

Définition 5.2.

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.

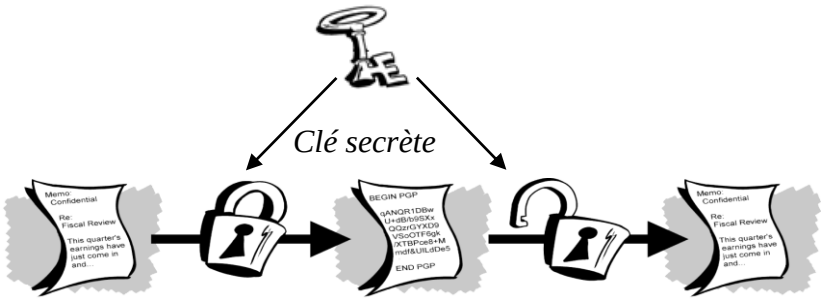

Fig 5.2 Chiffrement et déchiffrement symétrique.

5.2.2 Cryptographie asymétrique.

Définition 5.3.

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.

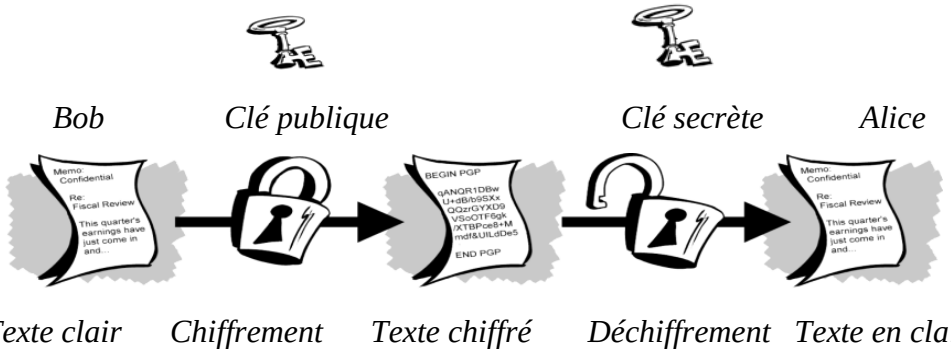

Fig 5.3 Chiffrement et déchiffrement asymétrique.

5.3 Cryptosystème de McEliece.

5.3.1 Historique et principe du Cryptosystème de McEliece.

Le cryptosystème de McEliece, développé par l'informaticien américain Robert J. McEliece en 1978, est l'un des premiers systèmes de chiffrement à clé publique basés sur la théorie de la correction d'erreur. Il diffère considérablement des systèmes de chiffrement à clé publique plus couramment utilisés, tels que le RSA ou le chiffrement basé sur les courbes elliptiques. Le cryptosystème de McEliece est apprécié pour sa résistance théorique aux attaques de déchiffrement quantique, ce qui en fait une option attrayante pour la sécurité à long terme.

L'idée fondamentale derrière le cryptosystème de McEliece repose sur les codes correcteurs d'erreurs, qui sont traditionnellement utilisés pour corriger les erreurs de transmission dans les communications numériques. Au lieu d'utiliser ces codes pour corriger des erreurs, McEliece a conçu son système de chiffrement en exploitant la difficulté de déterminer les erreurs dans les données chiffrées.

Voici une brève introduction au fonctionnement du cryptosystème de McEliece :

1. **Génération des clés** : Tout d'abord, l'utilisateur génère une paire de clés, une clé publique et une clé privée. La clé publique est destinée à chiffrer les messages, tandis que la clé privée est utilisée pour déchiffrer les messages.
2. **Construction de la matrice de génération de code** : Un élément clé du cryptosystème de McEliece est une matrice binaire de grande taille appelée matrice de génération de code. Cette matrice est générée de manière aléatoire et est utilisée pour encoder les messages en texte chiffré.
3. **Chiffrement** : Pour chiffrer un message, l'émetteur encode le message en utilisant la matrice de génération de code et ajoute ensuite un vecteur d'erreur aléatoire. Le résultat est le texte chiffré, qui est envoyé au destinataire.
4. **Déchiffrement** : Pour déchiffrer le message, le destinataire utilise la clé privée, qui consiste en une description de la matrice de génération de code ainsi que des informations pour déterminer et corriger les erreurs introduites lors du chiffrement. En appliquant des techniques de correction d'erreur, le destinataire peut retrouver le message d'origine.

Ce qui distingue le cryptosystème de McEliece, c'est sa résistance présumée aux attaques quantiques, notamment aux attaques de factorisation quantique qui menacent la sécurité du systèmes cryptographiques traditionnels comme le RSA. Cependant, le principal inconvénient de ce cryptosystème réside dans la taille relativement grande de la clé publique, ce qui peut rendre les opérations de chiffrement et de déchiffrement plus lentes par rapport à d'autres méthodes. Malgré cela, le cryptosystème de McEliece reste un sujet de recherche actif en cryptographie, en particulier pour ses avantages en matière de sécurité quantique.

5.3.2 Algorithme Cryptosystème de McEliece.

1. Génération de clé

On commence par générer un code cyclique $C(n, k, d)$ en donnant son polynôme générateur $g(X)$ et donc 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_σ de taille $n \times n$ associée à une permutation σ 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_σ 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* ε 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_σ, 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 5.1.

Soit C(*7*, 4, *d*) un code de Hamming (cyclique) de longueur *n*=7, de polynôme générateur $g(X) = X^3 + X + 1$, donc sa matrice génératrice est donnée par : $G=\begin{pmatrix} 1\,1\,0\,1\,0\,0\,0 \\ 0\,1\,1\,0\,1\,0\,0 \\ 0\,0\,1\,1\,0\,1\,0 \\ 0\,0\,0\,1\,1\,0\,1 \end{pmatrix}$.

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

$$G_N=\begin{pmatrix} 1\,1\,0\,1\,0\,0\,0 \\ 0\,1\,1\,0\,1\,0\,0 \\ 0\,0\,1\,0\,0\,1\,0 \\ 0\,0\,0\,0\,0\,0\,1 \end{pmatrix}$$

et donc C admet comme matrice génératrice

$$H=\begin{pmatrix} 1\,0\,0\,1\,0\,0\,0 \\ 0\,1\,0\,1\,1\,0\,0 \\ 0\,0\,1\,0\,1\,1\,0 \end{pmatrix}$$

En utilisant la matrice *H*, on peut montrer que 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_σ de type 7×7, une matrice inversible et aléatoire *S* de type 4×4 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 $G'=S.G_N.P_\sigma$. Donc

$$G'=\begin{pmatrix} 0\,0\,1\,1\,0\,0\,1 \\ 1\,1\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,1\,1 \\ 0\,0\,0\,0\,1\,0\,0 \end{pmatrix}$$

et la capacité *e*=1.

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

c'=0011010+0100000 = 0111010.

Le déchiffrement

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

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

On calcule r'=c'. P_σ^{-1}= 0111010.$\begin{pmatrix} 0&0&0&1&0&0&0\\ 1&0&0&0&0&0&0\\ 0&0&0&0&1&0&0\\ 0&1&0&0&0&0&0\\ 0&0&0&0&0&0&1\\ 0&0&0&0&0&1&0\\ 0&0&1&0&0&0&0 \end{pmatrix}$ =1100110. r' est un mot entaché d'erreurs.

En représentation polynomiale le mot r'=1100110 correspond au polynôme $r'(X) = X^5 + X^4 + X + 1$. En utilisant la méthode de décodage de Meggitt. On peut corriger le mots $r'(X)$ d'erreur ε'.

Le syndrome de *r'(X)* est le reste de la division Euclidienne de *r'(X)* par $g(X)$. En utilisant un registre à décalage circulaire, on trouve S(*r'(X)*)=*X* et w(S(*r'(X)*)) =1. D'apres l'algorithme de Meggitt L'erreur est $\varepsilon(X) = S\big(r'(X)\big) = X$. Donc le mot code est $r(X) = r'(X) + \varepsilon(X) = X^5 + X^4 + 1$, le mot correspondant au polynôme $r(X)$ est $r = 1000110$= $m.S.G_N$.

Le décodage de *r* (en enlevant la redondance les trois bits de gauche) 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. Qui est le message clair transmis.

Exemple 5.2. Soit C (*n*=15, *k*=7, d) un code cyclique de polynôme générateur

$$g(X) = X^8 + X^4 + X^2 + X + 1.$$

et comme matrice de contrôle

H=(I_8/M) 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 que 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 envoyer ce message à Alice.

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

La clés secrète : les éléments de la première ligne de la matrice génératrice sont les coefficients du générateur et les autres lignes sont les shifts de la première ligne donc :

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

La matrice normalisée est comme suit :

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

et comme matrice de contrôle

H=(I_8/M) tel que M= $\begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix}$

Une matrice de permutation P_σ de type 15×15, associée à la permutation

σ = (3,11,4,6,13,5,14,2,10,9,12,7,8,1,15).

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

Sa matrice inverse est :

$$\mathrm{P}_\sigma^{-1} = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 1&0&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0&0&1 \end{pmatrix}.$$

Une matrice inversible et aléatoire S de type 7×7 par exemple:

$$S = \begin{pmatrix} 0&1&0&0&0&0&0 \\ 1&0&0&0&0&1&1 \\ 0&1&1&0&0&0&0 \\ 1&0&0&0&0&1&0 \\ 1&0&0&0&1&0&1 \\ 1&1&1&1&0&1&1 \\ 1&1&1&0&1&1&1 \end{pmatrix}$$

Sa matrice inverse est :

$$S^{-1} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 \end{pmatrix}$$

Apres calcul, on trouve que la clé publique ($G'=S.G_N.P_\sigma$) est la suivante:

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

On choisit comme vecteur d'erreur par exemple ε=100000000000000, tel que w(ε)=1 ≤ *e*.

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

Apres calcul, on trouve le message chiffré

c'= *m.G'*+ε=101111111010100+100000000000000=001111111010100.

Le déchiffrement.

On a c'=*m.G'*+ε=*m.S.G.*P_σ+ε ⟹ r'=c'. P_σ^{-1}=*m.S.G*+ε. P_σ^{-1}=*m' G* +ε' tel que w (ε′)=1 (car P_σ est une permutation et conserve donc le poids des mots).

On trouve *r'*=c'. P_σ^{-1}=111111000101100. En représentation polynomiale

$r'(X) = X^{12} + X^{11} + X^9 + X^5 + X^4 + X^3 + X^2 + X + 1$, mot entaché d'erreur $\varepsilon'(X)$.

En utilisant la méthode de décodage par syndrome polynomial. On peut corriger le mot r' d'erreur ε'.

Le syndrome de $r'(X)$ est $S(r'(X)) = X^7 + X^6 + X^3 + X^2 + X$, qui représente le mot

$$h(r') = 01110011 = C_{14} = h(000000000000010) = h(\varepsilon').$$

Et w(ε')=1. L'erreur est donc ε'=000000000000010. Et le mot code est *r*=*m.S.G* = r'+ε',

r = 111111000101100 + 000000000000010 = 111111000101110.

On a *: r= m.S.G*=111111000101110. On enlève la redondance c.à.d. les huit premier bits, on trouve *m.S* =0101110= *m*' et donc le message initial :

$m=m'.S^{-1}=(0101110)\begin{pmatrix}0&0&1&1&1&0&1\\1&0&0&0&0&0&0\\1&0&1&0&0&0&0\\0&1&1&0&0&1&0\\0&1&1&0&0&0&1\\0&0&1&0&1&0&1\\0&1&0&1&0&0&0\end{pmatrix}$=1010110, qui est le mot clair.

5.4 Cryptosystème de Niederreiter

5.4.1 Historique et principe du Cryptosystème de Niederreiter

Le cryptosystème de Niederreiter est un système de chiffrement à clé publique, inventé par Harald Niederreiter en 1978 et qui a été mis au point en 1986. Ce système appartient à la famille des cryptosystèmes basés sur les codes à clé publique et est similaire au cryptosystème de McEliece, mais il diffère dans les détails de sa mise en œuvre. Le cryptosystème de Niederreiter est réputé pour sa robustesse théorique contre les attaques quantiques et sa sécurité basée sur des problèmes mathématiques difficiles.

Le cryptosystème de Niederreiter fonctionne de la même manière que celui de McEliece:

1. Génération des clés
2. Construction de la matrice de décodage.
3. Chiffrement.
4. Déchiffrement.

La force principale du cryptosystème de Niederreiter réside dans sa résistance supposée aux attaques quantiques, notamment aux attaques de factorisation quantique qui menacent la sécurité de systèmes cryptographiques classiques. De plus, contrairement au cryptosystème de McEliece, la clé publique du cryptosystème de Niederreiter est généralement plus petite, ce qui le rend plus rapide et plus efficace en termes de performances de chiffrement et de déchiffrement.

Cependant, la mise en œuvre du cryptosystème de Niederreiter peut être complexe et nécessite des calculs mathématiques avancés, ce qui peut limiter son adoption dans certaines applications. Néanmoins, en raison de sa sécurité théorique et de sa résistance aux attaques quantiques, il reste un sujet de recherche actif en cryptographie, en particulier à mesure que les technologies quantiques continuent de se développer.

5.4.2 Algorithme du Cryptosystème de Niederreiter

1. Génération de clé

On commence par générer un code cyclique $C(n, k, d)$ et une matrice de contrôle H de taille $n\text{-}k \times n$. On va mélanger cette matrice pour la rendre indistinguable d'une matrice aléatoire, pour cela on a besoin de :

1. La clé secrète est composée des trois matrices :
   a. Une matrice de contrôle H (et donc la matrice normalisée H_N) du code C.
   b. Une matrice de permutation aléatoire P_σ de taille $n \times n$ associée à une permutation σ du groupe symétrique S_n.
   c. Une matrice inversible aléatoire M de taille $n\text{-}k \times n\text{-}k$.
2. La clé publique sera le couple (H', t) où $H' = M.H_N.P_\sigma$ et $t \leq e$ tel que e est la capacité de correction de C.

2. Chiffrement.

Soit y un message de n bits que l'émetteur Bob veut chiffrer et de poids $w(y)= t \leq e$. Le mot chiffré est le mot $z = y.{}^tH'$ de type $1 \times n\text{-}k$.

3. **Déchiffrement.**

Pour déchiffrer en connaissant P_σ, M et H_N, Alice suit les étapes suivantes :

a. Calcule: $s= z.{}^tM^{-1}=(y.{}^tP_\sigma).{}^tH_N$ de type $1 \times n\text{-}k$ qui est un mot syndrome.
b. En utilisant un algorithme de décodage des codes cycliques, Alice retrouve le mot $z'= y.{}^tP_\sigma$ correspondant au syndrome s.
c. Le récepteur retrouve enfin le mot $y=z'.{}^tP_\sigma^{-1}$.

Exemple 5.3. Soit C(7, 4, d) un code de Hamming (cyclique) de longueur n=7, de polynôme générateur $g(X) = X^3 + X + 1$, donc son polynôme de control est :

$$h(X) = X^4 + X^2 + X + 1.$$

Le polynôme générateur du code dual du code C est :

$$h_1(X) = X^4 h(X^{-1}) = X^4 + X^3 + X^2 + 1.$$

et donc C admet comme matrice de contrôle

$$H = \begin{pmatrix} 1\,0\,1\,1\,1\,0\,0 \\ 0\,1\,0\,1\,1\,1\,0 \\ 0\,0\,1\,0\,1\,1\,1 \end{pmatrix}$$

La matrice normalisée H_N associée à H est :

$$H_N = \begin{pmatrix} 1\,0\,0\,1\,0\,1\,1 \\ 0\,1\,0\,1\,1\,1\,0 \\ 0\,0\,1\,0\,1\,1\,1 \end{pmatrix}$$

En utilisant la matrice H_N, on peut montrer que 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.

3. **Génération des clés.** Alice génère les clés suivantes :

c. La clés secrète : La matrice normalisée H_N, une matrice de permutation P_σ de type 7×7, une matrice inversible et aléatoire M de type 3×3 par exemple on choisit:

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

L'inverse de M est elle-même, car M est une matrice de permutation symétrique.

d. La clé publique est le couple (H', e) tel que $H'=M.H_N.P_\sigma$.

Apres calcul on trouve

$$H' = \begin{pmatrix} 1&0&1&1&0&1&0 \\ 1&1&0&0&1&1&0 \\ 0&0&1&0&1&1&1 \end{pmatrix}$$

et la capacité e=1.

Le Chiffrement. Soit $y(X) = X^4 + X^3 + X + 1$ qui correspond au mot y=0001000 le mot reçu dont le poids de l'erreur ne dépasse pas 1. Bob chiffre le message y, en calculant le mot chiffré $z = y.{}^tH'$. On trouve que le texte chiffré est :

$$z = 1\,0\,0$$

Le déchiffrement

Pour le déchiffrement on calcule: $s = z.{}^tM^{-1} = z.{}^tM = 1\,0\,0 \begin{pmatrix} 0&1&0 \\ 1&0&0 \\ 0&0&1 \end{pmatrix} = 0\,1\,0 \neq 000$.

Théoriquement $s=y.{}^t P_\sigma.{}^tH_N$. c.à.d. s représente le syndrome du mot $y' = y.{}^tP_\sigma$ et comme $s \neq 0$, donc y' n'est pas un mot de C et donc nécessite une correction.

Pour la correction de y', on va utiliser la méthode de décodage par syndrome. On a

s = C_2 (s est la deuxième colonne de H_N) donc s=h(ε) tel que ε=010000000000000 et le poids w(ε)=1=e. Donc les mots $y' = y.{}^tP_\sigma$ et ε ont le même syndrome et le même poids=1 alors, d'après la Proposition 2.8, on déduit que $y' = \varepsilon$ d'où $y.{}^tP_\sigma=\varepsilon$, ce qui donne que $y = \varepsilon.{}^tP_\sigma^{-1}$, comme P est une matrice de permutation symétrique alors, $P_\sigma^{-1} = {}^tP_\sigma = P_\sigma$ et le mot déchiffré est : $y = \varepsilon.P_\sigma$=0001000, qui est bien le mot transmis.

Exemple 5.4. Considérons le code (n = 15; k = 7; d = 5) de longueur n = 15 sur le corps de Galois F_{16}, de racine primitive α et de polynôme primitif $M_\alpha(X) = X^4+X+1$.

On a $F_{16} = \{0, \alpha^i / 0 \le i \le 14\}$ avec $\alpha^4 = \alpha + 1$. Considérons la matrice de contrôle H:

$$H=\begin{pmatrix} 1&1&0&1&0&0&0&1&0&0&0&0&0&0&0\\ 0&1&1&0&1&0&0&0&1&0&0&0&0&0&0\\ 0&0&1&1&0&1&0&0&0&1&0&0&0&0&0\\ 0&0&0&1&1&0&1&0&0&0&1&0&0&0&0\\ 0&0&0&0&1&1&0&1&0&0&0&1&0&0&0\\ 0&0&0&0&0&1&1&0&1&0&0&0&1&0&0\\ 0&0&0&0&0&0&1&1&0&1&0&0&0&1&0\\ 0&0&0&0&0&0&0&1&1&0&1&0&0&0&1 \end{pmatrix},$$

qu'on peut mettre sous forme systématique : $H_N=(I_8 /A)$ tel que

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

On considère la matrice

$$M=\begin{pmatrix} 1&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&0\\ 1&1&1&1&0&0&0&0\\ 1&1&1&1&1&0&0&0\\ 1&1&1&1&1&1&0&0\\ 1&1&1&1&1&1&1&0\\ 1&1&1&1&1&1&1&1 \end{pmatrix},$$

d'inverse

$$M^{-1}=\begin{pmatrix} 1&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0\\ 0&0&1&1&0&0&0&0\\ 0&0&0&1&1&0&0&0\\ 0&0&0&0&1&1&0&0\\ 0&0&0&0&0&1&1&0\\ 0&0&0&0&0&0&1&1 \end{pmatrix}$$

et la matrice de permutation $P = P_\sigma$, tel que σ est la transposition τ_{17}.

Après calcul de $H' = M.H.P$, on trouve

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

La clé publique est (H', e=2). Soit le texte clair qu'on veut chiffrer et envoyer : y=010000000000001 de poids t=2.

Le chiffrement de y est le mot z donné par : $z = y.{}^tH'$. En remplaçant y et H' on trouve: z=10000000.

Pour le déchiffrement on calcule: $s = z.{}^tM^{-1}$=11000000 ≠ 0. Théoriquement $s=y.{}^tP.{}^tH_N$. c.à.d. s représente le syndrome du mot $y' = y.{}^tP$ et comme $s \neq 0$ donc y' n'est pas un mot de C.

Pour la correction de y', on va utiliser la méthode de décodage par syndrome. On a $s = C_1+C_2$ (s est la somme de la première et la deuxième colonne de H_N) donc $s=h(\varepsilon)$ tel que ε=110000000000000 et le poids $w(\varepsilon)=2=e$. Donc les mots $y' = y.{}^tP$ et ε ont le même syndrome et le même poids alors, d'après la Proposition 2.8, on déduit que $y' = \varepsilon$ d'où $y.{}^tP=\varepsilon$ ce qui donne que $y = \varepsilon.{}^tP^{-1}$, comme P est une matrice de permutation symétrique alors, $y = \varepsilon.P$=010000000000001, qui est bien le mot transmis.

5.5 Comparaison des cryptosystèmes McEliece, Niederreiter et RSA.

	McEliece	Niederreiter	RSA
Taille de la clé publique.	*Kn*	*k(n-k)*	*2n*
	67072 octets	*32750 octets*	256 octets
Nombre de bits d'information transmis par chiffrement.	*K*	$a=log2(C_n^e)$	*N*
	512	*276*	1024
Taux de transmission.	*k/n*	$Log_2(C_n^e)/n-k$	1
	51,17%	56,81%	100%
Nombre d'opérations binaires du chiffrement par bit d'information.	n/2+ *n/k*	*(n-k)ke/an+n/a*	$125.3^{m-1}/2^m$
	513,9	50,1	2402,7
Nombre d'opérations binaires du déchiffrement par bit d'information.	*B/k*	*C/a*	$25.3^{m-1}/2$
	5140	7863,3	738112,5
$B=n+mnt+4m^2t^2+2mt+mn(2t+1)+k^2/2$ et $C=2n+4m^2t^2+2m^2t+mn(2t+1)+(n-k)^2/2$.			

Tableau 5.1 Tableau de comparaison des cryptosystèmes McEliece, Niederreiter et du RSA.

Du tableau ci-dessus on déduit que la taille de la clé au cryptosystème RSA est meilleur que celle de Niederreiter et cette dernière et meilleur que celle de McEliece.

Le taux de transmission dans RSA est paré et il est deux fois meilleur que celui des deux autres cryptosystèmes qui se rapprochent.

Concernant le nombre d'opérations binaires du chiffrement est très couteux dans RSA est le moins couteux dans Niederreiter.