

Codes correcteurs et cryptosystème de Mc Eliece

Auclair Pierre

23 mars 2014

Table des matières

1	Codes correcteurs	2
1.1	Définition d'un code correcteur	2
1.2	Utilisation des codes correcteurs	2
2	Codes de Goppa	4
2.1	Définition d'un code de Goppa	4
3	Cryptosystème de Mc Eliece	6
3.1	Les clefs	6
3.2	Le chiffrement	7
3.3	Le déchiffrement	7
3.4	Le principe de sécurité	7
4	Implémentation des structures du code et du cryptosystème	8
4.1	Structures algébriques	8
4.1.1	Les matrices	8
4.1.2	Les polynômes	8
4.1.3	Les éléments dans un corps de Galois \mathbb{F}_{2^m}	9
4.2	Structures de support d'information	9
4.2.1	Fichiers enregistrés	9
4.2.2	Messages	9
4.2.3	Clefs publiques et privées	9
5	Principaux algorithmes	10
5.1	Algorithme de Strassen	10
5.2	Pivots de Gauss et applications	10
5.3	Algorithme de Berlekamp	10
5.4	Algorithme de décodage	10

Chapitre 1

Codes correcteurs

1.1 Définition d'un code correcteur

Définition 1. [[Pan04](#)] Un code en bloc \mathbf{C} est l'image d'une application injective \mathbf{f} de \mathbb{F}_q^k dans \mathbb{F}_q^n . Ce code est dit linéaire si \mathbf{f} est une application linéaire.

$$\mathbf{f} : \mathbb{F}_q^k \rightarrow \mathbb{F}_q^n$$

Définition 2. La distance de Hamming est l'application \mathbf{d} de $\mathbb{F}_q^n \times \mathbb{F}_q^n$ dans \mathbb{N} définie par :

$$\mathbf{d} : (a, b) \in \mathbb{F}_q^n \times \mathbb{F}_q^n \mapsto \text{card}(i \in \mathbb{N} / a_i \neq b_i)$$

$\mathbf{d}(a, 0) = \omega(a)$ est appelé poids de Hamming. On appelle d le poids minimum des mots de \mathbf{C} .

Définition 3. On note t la capacité de correction d'un code correcteur, par définition :

$$t = \max_{r \in \mathbb{N}} \bigcap_{x \in \mathbf{C}} B(x, r) = \emptyset$$

De manière évidente on a l'inégalité : $2t + 1 \leq d$

Définition 4. On appelle un décodage l'application $\mathbf{D} : \mathbb{F}_q^n \rightarrow \mathbb{F}_q^k$ telle que $\mathbf{D} \circ \mathbf{f} = \text{Id}_{\mathbb{F}_q^k}$. \mathbf{D} est de vraisemblance maximale si :

$$\forall x \in \mathbf{C}, \mathbf{D}(B(x, t)) = x$$

1.2 Utilisation des codes correcteurs

L'intérêt d'utiliser des applications linéaires est de pouvoir définir l'ensemble \mathbf{C} avec un minimum d'informations. La seule donnée de l'image des vecteurs de base définit entièrement \mathbf{f} et \mathbf{C} et donc le codage.

Définition 5. On appelle G matrice génératrice du code \mathbf{C} la matrice de \mathbf{f} dans la base canonique.

$$G \in M_{n,k}(\mathbb{F}_q), G = \text{Mat}(\mathbf{f})$$

Définition 6. On appelle H matrice de parité du code \mathbf{C} une matrice telle que :

$$H \in M_{n-k,n}(\mathbb{F}_q), \text{Ker}(H) = \mathbf{C}$$

L'application $x \in \mathbb{F}^n \rightarrow Hx$ s'appelle le syndrome de x .

Propriété 1. A la réception d'un message de \mathbb{F}^n , en considérant que le nombre d'erreurs est inférieur à t , donc que le vecteur d'erreur ϵ a un poids inférieur ou égal à t , on a : $\forall m \in \mathbb{F}^n, \exists!(x, \epsilon) \in \mathbf{C} \times \mathbb{F}^n$ tels que :

$$\begin{cases} m &= x + \epsilon \\ \mathbf{d}(\epsilon) &\leq t \\ Hm &= H\epsilon \end{cases}$$

Le principe est de déterminer ϵ . On retrouve ainsi le mot de code d'origine $x = m - \epsilon$

Propriété 2. $\forall m \in \mathbb{F}^n$, son syndrome est caractéristique de l'erreur, dans l'hypothèse d'un maximum de t erreurs. Supposons deux vecteurs d'erreurs avec le même syndrome :

$$H\epsilon = H\eta \Rightarrow H(\epsilon - \eta) = 0 \Rightarrow \epsilon - \eta \in \mathbf{C}$$

D'après l'inégalité triangulaire, $d(\epsilon - \eta) \leq 2t < d$ donc $\epsilon - \eta = 0$

Chapitre 2

Codes de Goppa

2.1 Définition d'un code de Goppa

Les codes de Goppa classiques sont des codes linéaires dont on connaît une borne inférieure de la distance minimale. Ce sont les codes utilisés dans le cryptosystème de McEliece qui va nous intéresser dans une deuxième partie.

Définition 7. On définit un code de Goppa $\mathbb{F}_q^k \rightarrow \mathbb{F}_q^n$ par son support L et un polynôme g défini par :

$$\begin{cases} g & \in \mathbb{F}_{q^m}[X] \text{ unitaire irréductible sur } \mathbb{F}_{q^m} \text{ de degré } t \\ L & = (\alpha_1, \dots, \alpha_n) \text{ d'éléments de } \mathbb{F}_{q^m} \end{cases}$$

A partir de ces éléments, on définit le code de Goppa \mathbf{C} par son syndrome \mathbf{S} comme :

$$\mathbf{C} = \{y = (y_1, \dots, y_n) \in \mathbb{F}_q^n / \mathbf{S}(y) = \sum_{i=1}^n \frac{y_i}{x - \alpha_i} = 0 \text{ mod } g(x)\}$$

Propriété 3. Le code de Goppa ainsi défini est de distance minimale $\mathbf{d} \geq t + 1$. En effet en supposant qu'il existe un message $y = (y_1, \dots, y_n) \in \mathbf{C}$ avec t ou moins coordonnées non nulles :

$$\sum_{i=1}^n \frac{y_i}{x - \alpha_i} = \frac{a(x)}{b(x)} = 0 \text{ mod } g(x) \text{ fraction irréductible}$$

Donc $g|a$, ainsi a a de degré supérieur ou égal à t . Or a a de degré = nombre de coordonnées non nulles de y - 1 $\leq t - 1$

Par l'absurde le code de Goppa a une distance minimale $\mathbf{d} \geq t + 1$.

Définition 8. Nous allons définir une famille de polynômes \mathbf{f}_i inverses des $(x - \alpha_i)$:

$$\mathbf{f}_i(x)(x - \alpha_i) = 1 \text{ mod } g(x)$$

On a donc que :

$$\forall y \in \mathbb{F}^n \quad \mathbf{S}(y) = \sum_{i=1}^n \frac{y_i}{x - \alpha_i} = \sum_{i=1}^n \mathbf{f}_i(x) y_i$$

Propiété 4. *Evaluons les \mathbf{f}_i :*

$$\mathbf{f}_i(x) = \frac{1}{\alpha_i} \frac{g(x) - g\alpha_i}{x - \alpha_i}$$

Chapitre 3

Cryptosystème de Mc Eliece

Le cryptosystème de Mc Eliece mis au point en 1978 est l'un des premiers systèmes de cryptage à clef publique. Son principe repose sur l'utilisation de codes correcteurs d'erreurs, plus particulièrement des codes de Goppa. Bien que l'utilisation de codes correcteurs linéaires facilement décodables le rende extrêmement rapide, ce système a été pénalisé par la taille importante de ses clefs publiques. Néanmoins, des projets comme celui de l'OTAN¹ remettent au jour ce système pouvant résister à l'avènement d'ordinateurs quantiques. C'est ce cryptosystème que nous avons étudié.

3.1 Les clefs

Un cryptage asymétrique repose sur un principe simple : tout le monde a accès à une clef publique permettant le chiffrement et seul le créateur de cette clef possède la clef privée permettant le déchiffrement.

Pour ce faire, on commence par créer G la matrice génératrice d'un code de Goppa de paramètres :

- k dimension du bloc de message à coder
- n dimension du bloc de message à envoyer
- t la capacité de correction

La clef privée représente l'ensemble :

- G matrice génératrice (n,k)
- P une matrice (n,n) de permutation
- Q une matrice (k,k) inversible

La clef publique partagé est l'ensemble :

- $G' = PGQ$
- La capacité de correction t

1. Projet OTAN : SPS 984520 Secure implementation of post-quantum cryptography

3.2 Le chiffrement

Soit $m \in \mathbf{F}_2^k$ le message à chiffrer, l'envoyeur calcule l'image du message par le code G puis ajoute un vecteur $e \in \mathbf{F}_2^n$ de poids inférieur à t .

$$x = G'm + e = PGQm + e$$

C'est ce vecteur $x \in \mathbf{F}_2^n$ qui correspond au message chiffré.

3.3 Le déchiffrement

Soit $x \in \mathbf{F}_2^n$ le message chiffré reçu, on calcule :

$$P^{-1}x = GQm + P^{-1}e$$

Le vecteur $GQm \in \mathbf{F}_2^n$ est un mot du code de Goppa et donc $P^{-1}e \in \mathbf{F}_2^n$ est un vecteur d'erreur de poids inférieur à t car P matrice de permutation. On utilise notre algorithme efficace de décodage du code de Goppa, on obtient donc Qm , connaissant Q inversible on a :

$$m = Q^{-1}Qm$$

D'où le déchiffrement.

3.4 Le principe de sécurité

Le cryptosystème de Mc Eliece repose sur 2 problèmes difficiles mathématiques.

Premièrement, la difficulté à partir d'une matrice $G' = PGQ$ de retrouver la matrice G en temps polynômial. Secondement, de décoder un code correcteur qui paraît aléatoire, c'est le problème de reconnaissance du syndrome que nous avons évité en utilisant un code de Goppa dont on connaît la structure et donc un algorithme efficace de décodage.

Chapitre 4

Implémentation des structures du code et du cryptosystème

4.1 Structures algébriques

4.1.1 Les matrices

```
1
2  def __init__(self, nbligne, nbcolonne, tableau):
3      """Definition de la matrice avec gestion du cas erreur de taille"""
4      try:
5          if len(tableau) != nbligne * nbcolonne :
6              raise IndexError("Erreur de taille de la matrice durant l'initialisation")
7          self.nbligne=nbligne
8          self.nbcolonne=nbcolonne
9          self.tableau=tableau[::]
10     except IndexError as ex:
11         print(ex)
12         print [nbligne, nbcolonne, tableau]
13     except AttributeError as ex:
14         print(ex)
```

Les matrices sont représentées par un tableau linéaire dont le comportement est caractérisé par le nombre de lignes et le nombre de colonnes. La flexibilité de Python nous permet d'utiliser cette classe aussi bien pour des réels que pour des éléments de corps fini.

4.1.2 Les polynômes

```
1      """Classe de polynomes"""
```

```

2
3     def __init__(self, liste):
4         """Initialisation de l'objet"""
5         try:
6             self.liste=liste[::]

```

Les polynômes à coefficients aussi bien dans \mathbb{R} que dans des corps finis.

4.1.3 Les éléments dans un corps de Galois \mathbb{F}_{2^m}

```

1
2     def __init__(self, n, p=2):
3         """Initialisation"""
4         diff = d(n, p)
5         while diff >= 0:
6             n ^= p << diff

```

Un élément d'un corps de Galois sur \mathbb{F}_{2^m} est un polynôme de $\mathbb{Z}/2\mathbb{Z}[X]/P$ avec P un polynôme irréductible de $\mathbb{Z}/2\mathbb{Z}[X]$. Notre implémentation repose sur la compréhension du binaire par l'interpréteur Python : un entier est stocké sous sa forme binaire sur laquelle on peut effectuer des opérations binaires (xor, and, or et décalages de bits). Grâce à cette caractéristique de Python on peut utiliser des entiers comme des polynômes de $\mathbb{Z}/2\mathbb{Z}$. On obtient une structure d'éléments de \mathbb{F}_{2^m} très rapide.

4.2 Structures de support d'information

4.2.1 Fichiers enregistrés

4.2.2 Messages

4.2.3 Clefs publiques et privées

Chapitre 5

Principaux algorithmes

5.1 Algorithme de Strassen

5.2 Pivots de Gauss et applications

5.3 Algorithme de Berlekamp

5.4 Algorithme de décodage

Bibliographie

- [Pan04] A.A Pantchichkine. *Mathématiques des codes correcteurs d'erreurs*. Institut Fourier, 2004.