Cryptanalyse Algébrique des Systèmes Basés sur les Codes

Mohamed Amine Najahi

encadré par Jean-Charles Faugère, Ludovic Perret
Ayoub Otmani et Jean-Pierre Tillich

Équipes projets SALSA/LIP6 & SECRET/INRIA-Rocquencourt

22 octobre 2012

Le contexte général

Ce compte rendu est un rapport d'un stage effectué au sein des équipes SECRET de l'INRIA/Rocquencourt et SALSA du LIP6. Le but du stage est d'investiguer la sécurité de certaines primitives de chiffrement à clef publique. Contrairement à la majorité des cryptosystèmes déployés aujourd'hui, les algorithmes auxquels nous nous sommes intéressé s'appuient sur des problèmes issus de la théorie des codes.

La cryptographie basée sur les codes n'est pas récente puisqu'elle remonte à 1978. Néanmoins, certains inconvénients comme la taille de la clef publique constituent des obstacles à son déploiement.

Le problème étudié

Plusieurs travaux récents ont proposé de remédier au problème de la taille de la clef publique en ajoutant de la structure aux codes utilisés. Néanmoins, l'impact de ces solutions sur la sécurité des systèmes n'est pas complètement établi. D'ailleurs, des publications récentes ont exhibé des faiblesses dans ces variantes. Un des objectif du stage est d'étudier ces faiblesses et de les étendre à d'autres primitives.

La contribution proposée

Nous nous sommes intéressé à une variante proposée récemment. Nous avons effectué une étude approfondi des outils algébrique sous-jacents. Nous proposons aussi une attaque qui rend obsolètes certaines primitives basées de cette variante.

Les arguments en faveur de sa validité

Nous avons implémenté plusieurs algorithmes liés à cette variante. Notre attaque a aussi été implémenté.

Le bilan et les perspectives

Et après ? En quoi votre approche est-elle générale ? Qu'est-ce que votre contribution a apporté au domaine ? Que faudrait-il faire maintenant ? Quelle est la bonne *prochaine* question ?

Table des matières

1 Introduction

Depuis son introduction par Diffie et Hellman [1] en 1976, la cryptographie à clef publique n'a cessé de se diversifier. En effet, les cryptosystèmes à clef publique reposent sur des problèmes réputés difficiles, qui sont issus de théories diverses comme celles des codes correcteurs ou encore de la résolution de systèmes polynomiaux.

Néanmoins, la quasi totalité des algorithmes de chiffrement asymétrique en usage aujourd'hui reposent sur des problèmes issus de la théorie des nombres. Ainsi, la forte dépendance entre cryptographie à clef publique et théorie des nombres peut sembler alarmante. D'autant plus qu'en 1994, Shor a proposé un algorithme quantique [2] qui est capable de factoriser les entiers en temps polynomial. Le constat est sans appel : si un ordinateur quantique venait à être construit, une grande partie des primitives cryptographiques deviendrait obsolète. Depuis, la recherche autour des problèmes baptisés post-quantiques, s'est intensifiée. Dans cette course, les primitives basées sur les codes correcteurs d'erreurs semblent avoir une longueur d'avance.

Introduits par McEliece, en 1978, les chiffrements basés sur les codes ont très bien résisté aux attaques structurelles. Parmi leurs avantages figurent la rapidité du déchiffrement et surtout du chiffrement. Mais, leur principal inconvénient reste la taille des clefs publiques : le cryptosystème de McEliece offre avec une clef de 67 000 octet une sécurité inférieure à celle de RSA avec 128 octets.

En ajoutant de la structure sur la clef publique, plusieurs travaux ont tenté de réduire sa taille. C'est ainsi que dans [3], Berger *et al.* ont proposé d'utiliser comme clef publique une matrice quasi-cyclique. Barreto *et al.* ont quand à eux suggéré dans [4] d'utiliser une matrice quasi-dyadique.
Si d'un coté, ces propositions ont permis de baisser considérablement la taille des clefs, elles ont d'un autre côté exposé les systèmes aux attaques structurelles [5],[6].

Quelques unes de ces attaques seront exposées dans ce rapport. En effet, on commencera par un rappel sur les théories des codes et des systèmes polynomiaux. Nous présenterons ensuite les principales primitives basées sur les codes. Enfin, une partie sera consacrée à la variante dyadique du cryptosystème de McEliece, ainsi qu'aux attaques structurelles.

2 Rappels sur les codes

2.1 Codes linéaires

Un code linéaire $\mathcal{C}$ est un sous-espace vectoriel de dimension k de $\mathbb{F}_q^n$. On appelle n la longueur du code, k sa dimension. La distance de Hamming est la fonction qui à deux mots m_1 et m_2 du code, associe le nombre de positions où ses deux mots diffèrent, on la note $d(m_1, m_2)$. Le poids de Hamming d'un

mot m est la distance $d(m, \overrightarrow{0})$, où $\overrightarrow{0}$ est le mot dont toutes les composantes sont nulles. On note d la distance minimale du code. Ce code que l'on note $[n, k, d]_q$ détécte $d-1$ erreurs et en corrige $\lfloor \frac{d-1}{2} \rfloor$.
Pour définir un code, on en donne généralement une base sous forme d'une matrice notée G, de taille $k \times n$. Cette matrice s'appelle la matrice génératrice du code.
On peut aussi définir un code comme le noyau d'une matrice notée H, que l'on appelle matrice de parité du code. Par le théorème du rang, cette matrice est de dimension $(n-k) \times n$. G et H sont ainsi liées par la relation

$$GH^T = (0)^{(k \times n-k)} \tag{1}$$

La matrice G est souvent donnée sous la forme agréable suivante :

$$G = \begin{pmatrix} 1 & 0 & \cdots & 0 & g_{1,k+1} & \cdots & g_{1,n} \\ 0 & 1 & \cdots & 0 & g_{2,k+1} & \cdots & g_{2,n} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & g_{k,k+1} & \cdots & g_{k,n} \end{pmatrix} = \left(Id^{(k \times k)} \mid \hat{G} \right) \tag{2}$$

On dit alors qu'elle est sous forme systématique, et on en déduit facilement la matrice de parité car :

$$H = \begin{pmatrix} -g_{1,k+1} & -g_{2,k+1} & \cdots & -g_{k,k+1} & 1 & 0 & \cdots & 0 \\ -g_{1,k+2} & -g_{2,k+2} & \cdots & -g_{k,k+1} & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ -g_{1,n} & -g_{2,n} & \cdots & -g_{k,n} & 0 & 0 & \cdots & 1 \end{pmatrix} = \left(-\hat{G}^T \mid Id^{(n-k \times n-k)} \right) \tag{3}$$

est bien de rang $n-k$ et on a bien :

$$GH^T = \left(Id^{(k \times k)} \mid \hat{G} \right) \begin{pmatrix} -\hat{G} \\ Id^{(n-k \times n-k)} \end{pmatrix} = \left(-\hat{G} + \hat{G} \right) = (0)^{(k,n-k)} \tag{4}$$

2.2 GRS et codes alternants

Les codes GRS et alternants sont des codes linéaires que l'on définit à partir d'une matrice de parité particulière.
On commence par choisir une longueur de code n, ainsi que deux valeurs m et t telles que $n > mt$. On se donne ensuite deux vecteurs distincts de $\mathbb{F}_{q^m}^n$: $x = (x_1, \ldots, x_n)$ tel que $x_i \neq x_j$ si $i \neq j$ et $y = (y_1, \ldots, y_n)$ tel que $y_i \neq 0 \, \forall i$ qui servent à construire la matrice de parité suivante :

$$H = \begin{pmatrix} y_1 & y_2 & \cdots & y_{n-1} & y_n \\ x_1 y_1 & x_2 y_2 & \cdots & x_{n-1} y_{n-1} & x_n y_n \\ x_1^2 y_1 & x_2^2 y_2 & \cdots & x_{n-1}^2 y_{n-1} & x_n^2 y_n \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{t-1} y_1 & x_2^{t-1} y_2 & \cdots & x_{n-1}^{t-1} y_{n-1} & x_n^{t-1} y_n \end{pmatrix} \tag{5}$$

Notons que l'on peut exprimer H comme produit de deux matrices plus simples : En effet, la matrice de Vandermonde à l'ordre t de x est définie ainsi :

$$Vdm(x,t) = \begin{pmatrix} 1 & 1 & \cdots & 1 & 1 \\ x_1 & x_2 & \cdots & x_{n-1} & x_n \\ x_1^2 & x_2^2 & \cdots & x_{n-1}^2 & x_n^2 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ x_1^{t-1} & x_2^{t-1} & \cdots & x_{n-1}^{t-1} & x_n^{t-1} \end{pmatrix} \tag{6}$$

Rappelons au passage que le fait que toute sous-matrice carré de la matrice de Vandermonde est inversible est un résultat classique d'algèbre.
Le deuxième type de matrice est la matrice diagonale d'un vecteur :

$$Diag(y) = \begin{pmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & y_n \end{pmatrix} \tag{7}$$

Il est maintenant facile de voir que :

$$H = Vdm(x,t) \cdot Diag(y) \tag{8}$$

Un code dont la matrice de parité est H et dont les mots sont des vecteurs de $F_{q^m}^n$ s'appelle un code de Reed-Solomon Généralisé (GRS en anglais).
Un code dont la matrice de parité est H et dont les mots sont des vecteurs de F_q^n s'appelle un code alternant.

Théorème 2.1. *La distance minimale d d'un code GRS est $\geq t+1$.*

Démonstration. La distance minimale d'un code est égale au nombre minimal de colonnes linéairement dépendantes dans sa matrice de parité H.
La matrice $Diag(y)$ et les sous-matrices carrées $t \times t$ de $Vdm(x,t)$ étants inversibles. On déduit que la distance minimale d est $\geq t+1$. □

On déduit de ce théorème que le rang $n-k$ de la matrice de parité H est égal à t. En fait, on a d'une part que l'inégalité $n-k \geq d-1$ est vraie pour tout code linéaire, et s'appelle la borne de Singleton. D'autre part, $n-k \leq t$ car H a t lignes. Les codes, tels que les GRS, vérifiants l'égalité $n-k = d-1$ s'appellent des codes MDS (Maximum Distance Separable).
Un code alternant est par définition un sous-code d'un GRS. On obtient un code alternant en ne gardant que les vecteurs du noyau de H qui sont dans $\mathbb{F}_q^n$. Un tel code s'appelle un *sous-code sur un sous-corps.*
En se donnant une base de $\mathbb{F}_{q^m}$ sur $\mathbb{F}_q$, et en projetant la matrice de parité

sur le petit corps, on obtient la matrice de parité $mt \times n$ suivante :

$$\hat{H} = \begin{pmatrix} \pi_1(y_1) & \pi_1(y_2) & \cdots & \pi_1(y_{n-1}) & \pi_1(y_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \pi_m(y_1) & \pi_m(y_2) & \cdots & \pi_m(y_{n-1}) & \pi_m(y_n) \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \pi_1(x_1^{t-1}y_1) & \pi_1(x_2^{t-1}y_2) & \cdots & \pi_1(x_{n-1}^{t-1}y_{n-1}) & \pi_1(x_n^{t-1}y_n) \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \pi_m(x_1^{t-1}y_1) & \pi_m(x_2^{t-1}y_2) & \cdots & \pi_m(x_{n-1}^{t-1}y_{n-1}) & \pi_m(x_n^{t-1}y_n) \end{pmatrix} \in \mathbb{F}_q^{mt \times n} \tag{9}$$

On a donc $n-k \leq mt$, et on obtient le code alternant $[n, \geq n-mt, \geq t+1]_q$. En mettant $\hat{H}$ sous forme systématique, on en déduit facilement une matrice génératrice du code.

2.3 Codes de Goppa

Ces codes ont été proposés par Goppa en 1982 [7]. La définition classique des codes de Goppa est la suivante :
Étants donnés un polynôme de degré t, $G(X) \in \mathbb{F}_{q^m}[X]$ et un vecteur $x = (x_1, \ldots, x_n)$ de $F_{q^m}^n$: tel que $x_i \neq x_j$ et $(X - x_i) \not| \quad G(X) \quad \forall i$, les mots du code de Goppa sont les vecteurs $c \in \mathbb{F}_q^n$ tels que :

$$\sum_{i=1}^{n} \frac{c_i}{X - x_i} \equiv 0 \, mod \, G(X).$$

où $G(X) \in \mathbb{F}_{q^m}[X]$ est de degré t.
On note traditionnellement ce code $\Gamma(x, G)$, et il est alternant pour les raisons suivantes :

$$(X - x_i)^{-1} = -\frac{G(X) - G(x_i)}{X - x_i} G(x_i)^{-1} \text{ dans } \mathbb{F}_{q^m}[X]/G(X) \tag{10}$$

$$c \in \Gamma(x, G) \Longleftrightarrow \sum_{i=1}^{n} c_i \frac{G(X) - G(x_i)}{X - x_i} G(x_i)^{-1} = 0 \tag{11}$$

En écrivant le polynôme $G(X)$ sous la forme $G(X) = g_0 + g_1 X + \cdots + g_t X^t$, on a

$$\begin{array}{rcl}
\frac{G(X)-G(x_i)}{X-x_i} & = & \frac{g_1(X-x_i)+g_2(X^2-x_i^2)+\cdots+g_t(X^t-x_i^t)}{X-x_i} \\
& = & g_1 + g_2(X+x_i) + \cdots + g_t(X^{t-1} + x_i X_i^{t-2} + \cdots + x_i^{t-1}) \\
& = & g_1 + g_2 x_i + \cdots + g_t x_i^{t-1} \\
& + & (g_2 + g_3 x_i + \cdots + g_t x_i^{t-2})X \\
& \vdots & \vdots \\
& + & (g_{t-1} + g_t x_i)X^{t-2} \\
& + & g_t X^{t-1}
\end{array} \tag{12}$$

Et une matrice de parité est donné par la matrice $t \times n$ suivante :

$$\begin{array}{rcl}
\hat{H} & = & \begin{pmatrix}
g_t G(x_1)^{-1} & \cdots & g_t G(x_n)^{-1} \\
(g_{t-1} + x_1 g_t)G(x_1)^{-1} & \cdots & (g_{t-1} + x_n g_t)G(x_n)^{-1} \\
\vdots & \ddots & \vdots \\
(g_1 + g_2 x_1 + \cdots + g_t x_1^{t-1})G(x_1)^{-1} & \cdots & (g_1 + g_2 x_n + \cdots + g_t x_n^{t-1})G(x_n)^{-1}
\end{pmatrix} \\
& = & \begin{pmatrix}
g_t & 0 & \cdots & 0 \\
g_{t-1} & g_t & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
g_1 & g_2 & \cdots & g_t
\end{pmatrix}
\begin{pmatrix}
G(x_1)^{-1} & G(x_2)^{-1} & \cdots & G(x_n)^{-1} \\
x_1 G(x_1)^{-1} & x_2 G(x_2)^{-1} & \cdots & x_n G(x_n)^{-1} \\
\vdots & \vdots & \ddots & \vdots \\
x_1^{t-1} G(x_1)^{-1} & x_2^{t-1} G(x_2)^{-1} & \cdots & x_n^{t-1} G(x_n)^{-1}
\end{pmatrix} \\
& = & MH
\end{array}$$

où M est inversible (sinon $g_t = 0$ et le polynôme G serait de degré $< t$). Donc H est bien une matrice de parité du code de Goppa $\Gamma(x, G)$ et elle a bien la forme GRS voulue car :

$$H = Vdm(x,t) \cdot Diag(y)$$

avec $y = (G(x_1)^{-1}, \ldots, G(x_n)^{-1}) \in F_{q^m}^n$.

2.3.1 Codes de Goppa séparables

Théorème 2.2. *Soient $G(X) \in \mathbb{F}_{q^m}[X]$ un polynôme sans facteurs carrés, et $x = (x_1, \ldots, x_n)$ un vecteur de $F_{q^m}^n$ tel que $x_i \neq x_j$ et $(X - x_i) \nmid G(X) \quad \forall i$. Alors $\Gamma(x, G^{q-1}) = \Gamma(x, G^q)$.*

Démonstration. Sens 1 : Soit c un mot de $\Gamma(x, G^q)$, alors $\sum_{i=1}^{n} \frac{c_i}{X - x_i} \equiv 0 \, mod \, G(X)^q$. Or $G(X)^{q-1} \mid G(X)^q$ donc $\sum_{i=1}^{n} \frac{c_i}{X - x_i} \equiv 0 \, mod \, G(X)^{q-1}$ et on a bien $c \in \Gamma(x, G^{q-1})$.
Sens 2 : Soit $c \in \Gamma(x, G^{q-1})$. Il existe une extension k de $\mathbb{F}_{q^m}$ telle que G y est totalement décomposé. On a donc $\sum_{i=1}^{n} \frac{c_i}{X - x_i} = 0$ dans $k[X]/G(X)^{q-1}$

et pour toute racine z de G dans k, $\sum_{i=1}^{n} \frac{c_i}{X - x_i} = 0$ dans $k[X]/(X-z)^{q-1}$.
Considérons maintenant la somme de suite géomètrique suivante :
$S = \sum_i (\frac{X-z}{x_i - z})^i$, on a $S = \frac{z-x_i}{X-x_i}$ dans $k[X]/(X-z)^{q-1}$. Donc

$$\frac{1}{X - x_i} = -\frac{1}{x_i - z} - \frac{X-z}{(x_i-z)^2} - \frac{(X-z)^2}{(x_i-z)^3} - \dots \tag{13}$$

On en déduit que $\sum_i \frac{c_i}{(x_i - z)} = 0$, $\sum_i \frac{c_i}{(x_i-z)^2} = 0$, ..., $\sum_i \frac{c_i}{(x_i-z)^{q-1}} = 0$. En élevant l'équation $\sum_i \frac{c_i}{(x_i - z)} = 0$ à la puissance q et en remarquant que $c_i^q = c_i$, on obtient $\sum_i \frac{c_i}{(x_i-z)^q} = 0$. En ajoutant ce dernier terme à (13), et en remontant, on trouve que $\sum_{i=1}^{n} \frac{c_i}{X - x_i} = 0$ dans $k[X]/(X-z)^q$. Or $G^q = \prod (X-z)^q$, on a bien $\sum_i \frac{c_i}{(X-x_i)^q} = 0$ dans $k[X]/(G(X)^q)$ et donc dans $F_{q^m}[X]/(G(X)^q)$. □

Corollaire 2.1
Si $G \in F_{2^m}[X]$ est un polynôme de Goppa séparable alors $\Gamma(x, G) = \Gamma(x, G^2)$.

Ceci implique en particulier que la distance minimale est $\geq 2t+1$, à la condition que G soit sans facteurs carrés, et que le corps de base soit le corps binaire.
Remarque : On peut, pour s'assurer que cette condition soit vérifiée, prendre un polynôme irréductible comme polynôme de Goppa.

2.4 Matrices de Cauchy et codes de Goppa

Ètants donnés deux vecteurs $U = (u_0, \cdots, u_{m-1}) \in \mathbb{K}^m$ et $V = (v_0, \cdots, v_{l-1}) \in \mathbb{K}^l$, avec $u_i \neq v_j \forall i, j$, leur matrice de Cauchy associé $C(U, V)$ est la matrice $m \times l$ suivante :

$$C_{i,j} = 1/(u_i - v_j) = \begin{pmatrix} \frac{1}{u_0 - v_0} & \cdots & \frac{1}{u_0 - v_{l-1}} \\ \vdots & \ddots & \vdots \\ \frac{1}{u_{m-1} - v_0} & \cdots & \frac{1}{u_{m-1} - v_{l-1}} \end{pmatrix} \in \mathbb{K}^{m*l}.$$

Théorème 2.3. *Soit $\Gamma(X, G)$ un code de Goppa. Si le polynôme de Goppa G est scindé et sans facteurs multiples i.e. $G(Z) = (Z - z_0) \times \cdots \times (Z - z_{t-1})$ avec $z_i \in \mathbb{F}_{q^m}$ distincts, alors la matrice de Cauchy suivante : $C((z_0, \cdots, z_{t-1}), X)$ est une matrice de parité du code Γ.*

Remarque :Un code est un espace vectoriel de dimension k dans $\mathbb{F}_q$, toute matrice génératrice ou vérificatrice n'est jamais unique, elle l'est à multiplication par une matrice inversible près.

Démonstration. Soit C l'ensemble des mots du code Γ.

$$\forall c \in \mathbb{F}_q^n, c \in C \iff \sum_{i=0}^{n-1} \frac{c_i}{Z - x_i} \equiv 0 \, mod \, G(Z)$$

or $G(Z) = (Z - z_0) \times \cdots \times (Z - z_{t-1})$ donc $\sum_{i=0}^{n-1} \frac{c_i}{Z - x_i} \equiv 0 \, mod \, Z - z_j \forall j$ □

3 Cryptosystèmes basés sur les codes

3.1 Problèmes difficiles issus de la théorie des codes

La distance minimale d'un code linéaire. Décodage d'un code linéaire aléatoire.

3.2 Cryptosystème de McEliece

Le cryptosystème de McEliece est le premier chiffrement à clef publique basé sur un problème issu de la théorie des codes.
La clef privé est un code $[n, k, d]_q$ dont la matrice génératrice est $G \in \mathbb{F}_{q^{k \times n}}$, ainsi qu'une matrice inversible $S \in \mathbb{F}_{q^{k \times k}}$ et une matrice de permutation $P \in \mathbb{F}_{q^{n \times n}}$. La clef publique est le code $\hat{G} = SGP \in \mathbb{F}_{q^{k \times n}}$ ainsi que le nombre d'erreurs que le code peut corriger $t = \lfloor \frac{d-1}{2} \rfloor$.
Pour transmettre un message m à Bob, Alice lui envoie $m' = m\hat{G} + e$ où

	Cryptosystème de McEliece
Clef publique	$\hat{G} = SGP, t$
Clef privée	$S \in GL_k(\mathbb{F}_q), G, P \in \mathbb{F}_q^{n \times n}$

Tableau 1 – Le cryptosystème de McEliece.

$e \in \mathbb{F}_q^n$ tel que $\omega(e) = t$. Bob calcule alors $y = m'P^{-1} = mSG + eP^{-1}$. Comme P, est une matrice de permutation, on a que $\omega(eP^{-1}) = t$ et Bob peut retrouver mS en décodant y. Il retrouve alors m en multipliant par l'inverse de S.

	Cryptosystème de McEliece
Chiffrement	$c = m\hat{G} + e$, $m \in \mathbb{F}_q^k$, $w(e) = t$
Déchiffement	$m = Dec(cP^{-1})S^{-1}$

McEliece a proposé un code de Goppa $[1024, \geq 500, \geq 50]_2$ pour utiliser son cryptosystème. Néanmoins, la description reste inchangée si l'on souhaite utiliser une autre famille de code.

3.3 Cryptosystème de Niederreiter

Le but de Niederreiter était de réduire la taille de la clef publique de McEliece. Il a donc proposé d'utiliser le problème dual de McEeliece, à savoir, publier la matrice de parité et chiffrer en incorporant le message dans l'erreur.

	Cryptosystème de Niederreiter
Clef publique	$\hat{H} = SHP, t$
Clef privée	$S \in GL_{n-k}(\mathbb{F}_q), H, P \in \mathbb{F}_q^{n \times n}$
Chiffrement	$c = \hat{H}m^T$, $m \in \mathbb{F}_q^n$, $\omega(m) = t$
Déchiffement	$m^T = P^{-1}Dec.Syn(S^{-1}c)$

Tableau 2 – Le cryptosystème de Niederreiter

Théorème 3.1. *Les cryptosystème de McEliece et Niederreiter sont équivalents :*

Démonstration. Supposons que l'on dispose d'un algorithme pour résoudre McEliece. Soit $c = \hat{H}m^T$ un chiffrement Niederreiter. On peut trouver par l'algèbre linéaire un vecteur r tel que $c = \hat{H}r^T$ qui ne vérifie pas forcément la contrainte $\omega(r) = t$. On a alors $u = r - m \in \hat{G}$. On résout alors l'instance de McEliece suivante : $r = u + m$ qui permet de recouvrer u. On retrouve alors $m = r - u$.
Supposons que l'on dispose d'un algorithme pour résoudre Niederreiter.
Soit $c = m\hat{G} + e$ un chiffrement McEliece. On a $cH^T = eH^T$, donc $Hc^T = He^T$. On utlise alors notre algorithme pour trouver e tel que $\omega(e) = t$ et $Hc^T = He^T$. On dispose ainsi de $c - e = \hat{G}m$ et on peut résoudre et trouver m en utilisant de l'algébre linéaire. □

3.4 La signature de Courtois, Finiazs et Sendrier (CFS)

Il y'a $s = 2^{n-k}$ syndromes possibles avec une matrice de parité H de taille $n - k \times n$. Néanmoins, il n'y a que $l = \sum_{i=0}^{t} \binom{n}{i}$ syndromes correspondants à des erreurs de poids inférieur à t. L'idée de CFS est de hasher le message à signer concaténé à un entier i. En faisant varier l'entier, on finit par tomber sur un hashé h qui correspond bien à un syndrome d'une erreur e de poids t *i.e.* $h = He$. On utilise alors l'algorithme de décodage pour calculer e. La signature du message m est alors le couple (i, e). Pour la vérifier, il suffit de hasher le message concaténé à i et de vérifier que ce hashé est bien égal à He.

La densité des syndromes décodables est donc : $\frac{l}{s} \approx \frac{n^t}{t!2^{mt}}$. En prenant

	Signature CFS
Clef publique	$\hat{H} = SHP, t$
Clef privée	$S \in GL_{n-k}(\mathbb{F}_q), H, P \in \mathbb{F}_q^{n \times n}$
Signature	$h = hash(m,i)$, $e \in \mathbb{F}_q^n$, $\omega(e) = t$ $h = He^T$
Vérification	$hash(m,i) \stackrel{?}{=} He^T$

Tableau 3 – Le schéma de signature *CFS*.

donc $n = 2^m$. En obtient la meilleur densité qui est de $\frac{1}{t!}$.

4 Version dyadique du cryptosystème de McEliece

4.1 Matrices dyadiques

Définition 4.1. *Soient* $\mathbf{R}$ *un anneau, et* $m = (m_0, \ldots, m_{n-1}) \in \mathbf{R}^n$ *un vecteur. Une matrice* $M \in \mathbf{R}^{n \times n}$ *est dite dyadique si* $M_{i,j} = (m_{i \oplus j})$. *On dit que* m *est la signature de* M.

Proposition 4.1. *Une matrice dyadique* M *est entièrement déterminée par sa première ligne (ou colonne). Elle est de plus symètrique.*

Démonstration. On a $M_{0,j} = m_{0 \oplus j} = m_j$, $\forall j \leq n-1$. Pour la symètrie, il suffit de montrer que $M_{i,j} = M_{j,i} = m_{i \oplus j}$, or ceci est vrai car $i \oplus j = j \oplus i$. □

Proposition 4.2. *Si* $M \in \mathbf{R}^{n \times n}$ *est dyadique, il existe* $s \in \mathbb{N}$ *tel que* $n = 2^s$.

Démonstration. Supposons $n := 2^t \times m$ avec $m > 1$ impair. Alors il existe i tel que la représentation binaire de $n-1$ admet 0 à l'indice i *i.e.* $n - 1 = (1 \cdots 0 \cdots)_2$. Soit alors l'élèment $M_{n-1,2^{i-1}}$. Cet élèment est bien défini car $2^{i-1} \leq n-1$, mais $n - 1 \oplus 2^{i-1} = n - 1 + 2^{i-1} > n - 1$. Contradiction. □

Remarque : Cette dernière proposition permet de donner une nouvelle définition des matrices dyadiques. Si M de taille $2^s \times 2^s$ est dyadique alors, soit $s = 0$, soit $M = \begin{pmatrix} A & B \\ B & A \end{pmatrix}$ avec A et B dyadiques de taille $2^{s-1} \times 2^{s-1}$.

Proposition 4.3. *Soit* $n = 2^s$, *le produit de deux matrices dyadiques* $n \times n$ *est une matrice dyadique* $n \times n$.

Démonstration. Prouvons cette proposition par récurrence sur s. Pour $s = 0$, l'assertion est triviale. Soient M et N deux matrices dyadiques de taille

$2^s \times 2^s$. Il existe alors 4 matrices dyadiques A, B, C et D de taille $2^{s-1} \times 2^{s-1}$ telles que $M = \begin{pmatrix} A & B \\ B & A \end{pmatrix}$ et $N = \begin{pmatrix} C & D \\ D & C \end{pmatrix}$. On a

$$M \cdot N = \begin{pmatrix} AC + BD & AD + BC \\ BC + AD & BD + AC \end{pmatrix} \tag{14}$$

En utilisant l'hypothèse de récurrence, qui stipule que $AC+BD$ et $AD+BC$ sont dyadiques, on conclut que $M \cdot N$ l'est aussi. □

Proposition 4.4. *L'ensemble* $\mathbf{D}$ *des matrices dyadiques carrées* $n \times n$ *à coefficients dans* $\mathbf{R}$ *est un sous-anneau de* $\mathbf{R}^{n \times n}$.

Démonstration. Il est clair que D, muni de l'addition des matrices est un groupe commutatif dont l'élèment neutre est la matrice dyadique de signature $s = (0, \ldots, 0) \in \mathbf{R}^n$.
D est aussi stable pour la multiplication matricielle. L'élèment neutre pour la multiplication est la matrice identité, de signature $i = (1, 0, \cdots, 0)$. Les autres propriétés de distributivité et associativité sont facilement vérifiables. □

Proposition 4.5. *L'inverse d'une matrice dyadique* $2^s \times 2^s$ *inversible est aussi une matrice* $2^s \times 2^s$ *dyadique.*

Démonstration. La preuve est faite par récurrence : Pour des matrices de taille 1×1 l'assertion est triviale. Prouvons-la au rang s. Soit $M = \begin{pmatrix} M_1 & M_2 \\ M_2 & M_1 \end{pmatrix}$ une matrice dyadique $2^s \times 2^s$ et $S = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ son inverse. On a que

$$AM_1 + BM_2 = Id \qquad AM_2 + BM_1 = (0) \tag{15}$$

et

$$CM_1 + DM_2 = (0) \qquad CM_2 + DM_1 = Id \tag{16}$$

Or, par (15), en prenant $D = A$ et $C = B$ dans (16), ces équations sont vérifiées. Par unicité de l'inverse, on conclut alors que $A = D$ et $B = C$. □

Définition 4.2. *On note* $\Delta(m)$ *la matrice dyadique de signature* m *et* $\Delta(m, t)$ *la matrice* $\Delta(m)$ *tronquée à la* $t^{ième}$ *ligne. On définit également une matrice quasi-dyadique comme une matrice dyadique par bloc. Finalement, on définit une matrice de permutation dyadique comme une matrice dyadique dont la signature admet un seul élèment égal à 1, les autres éléments étants nuls.*

Remarque : Une matrice dyadique carrée, on l'a déja vu est forcément dyadique par bloc, mais le contraire n'est pas forcément vrai.
Une matrice de permutation dyadique est totalement déterminée par l'indice de l'élément non nul de sa première ligne.

4.2 Codes de Goppa dyadiques

Le but dans cette partie est de voir sous quelles conditions, une matrice de Cauchy d'un code de Goppa Γ est dyadique. Soit $H = (h_{ij})$.

$$h_{i,j} = \frac{1}{z_i - x_j} = \frac{1}{z_j - x_i} = h_{j,i} = h_{0,i\oplus j} \Rightarrow H \text{ est dyadique} \tag{17}$$

En particulier, $h_{i,i} = \frac{1}{z_i - x_i} = \frac{1}{z_0 - x_0} = h_{0,i\oplus i} = h_{0,0}$ donc $z_i = z_0 - x_0 + x_i$.

Or selon la première condition, $z_i = z_j - x_i + x_j$, en remplaçant z_j par sa valeur de la deuxième condition, on obtient $z_i = z_0 - x_0 + x_j - x_i + x_j$. Donc $z_0 - x_0 + x_i = z_0 - x_0 + x_j - x_i + x_j$ et $2x_i = 2x_j$. En cacatéristique impaire, ceci est impossible car $x_i = x_j$ ce qui contredit le fait que les élèments du support d'un code de Goppa doivent être distincts.
En caratéristique paire, cette condition est trivialement vérifiée. Donc $\frac{1}{h_{i,j}} = \frac{1}{h_{j,i}} = z_i + x_j = z_0 + x_i + x_0 + x_j = \frac{1}{h_{0,0}} + x_i + x_j = \frac{1}{h_{0,0}} + x_i + z_0 + x_j + z_0 = \frac{1}{h_{0,0}} + \frac{1}{h_{i,0}} + \frac{1}{h_{j,0}}$.
Notons $h = (h_{0,0}, \dots h_{0,n-1})$ la première ligne de la matrice H, par ce qui précéde, si

$$\frac{1}{h_{i\oplus j}} = \frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0} \tag{18}$$

alors H est une matrice dyadique d'un code de Goppa séparable.

L'idée, donc, est de générer la signature h en respectant la condition (18).

(ici, se referer à l'article de Barreto).

5 Attaque algèbrique de la version dyadique

L'objectif de l'attaque est de recouvrer le vecteur $\tilde{h}$ qui a servi à la construction de la matrice publique.
On dispose de la matrice génératrice G sous la forme systématique suivante :

$$G = \begin{pmatrix} & 1 & 0 & \cdots & \cdots & 0 \\ & 0 & 1 & \cdots & \cdots & 0 \\ P \in \mathbb{F}_2^{k\times k} & \vdots & \vdots & \ddots & \cdots & 0 \\ & \vdots & \vdots & \ddots & \cdots & 0 \\ & 0 & 0 & \cdots & \cdots & 1 \end{pmatrix} \in \mathbb{F}_2^{k\times n}$$

La relation fondamentale que l'on va utiliser est la suivante

$$G \cdot H^T = (0)^{k,n-k}$$

Comme H est dyadique, on s'interessera particulièrement à sa première ligne. En notant ce premier vecteur $X := (x_0, \cdots, x_{n-1}) \in \mathbb{F}_{q^m}[x_1, \cdots, x_{n-1}]$.

Le produit matrice vecteur de G par X donne k équations de la forme suivante :

$$\sum_{j=0}^{n-k-1} p_{i,j}x_j + x_{n-k+i} = 0 \quad \forall\, 0 \leq i \leq k-1$$

Or $n-k = mt$ est petit par rapport à k et n. Ce produit matrice vecteur permet donc d'exprimer linéairement k variables inconnues en fonction de $n-k = mt$ variables.

L'étape suivante consiste à exhiber les relations vérifiées entre les x_i et qui découlent de la construction dyadique par bloc. Au départ, le vecteur h vérifiait la condition suivante : $\frac{1}{h_{i\oplus j}} = \frac{1}{h_i} + \frac{1}{h_j} + \frac{1}{h_0}$.
Or la permutation des blocs fait que cette relation n'est plus vérifiée qu'à l'intérieur des blocs eux-même. À l'intérieur du $b^{\text{ième}}$ bloc de taille t, on a les relations suivantes :

$$\begin{aligned}
\frac{1}{h_{bt+i\oplus j}} &= \frac{1}{h_{(bt\oplus i)\oplus j}} \\
&= \frac{1}{h_{bt\oplus i}} + \frac{1}{h_j} + \frac{1}{h_0} \\
&= \frac{1}{h_{bt\oplus i}} + \frac{1}{h_{bt}} + \frac{1}{h_j} + \frac{1}{h_0} + \frac{1}{h_{bt}} \\
&= \frac{1}{h_{bt\oplus i}} + \frac{1}{h_{bt\oplus j}} + \frac{1}{h_{bt}}
\end{aligned}$$

Or, chaque bloc a subi une multiplication par une matrice de permutation dyadique. On va exhiber l'effet de cette transformation.

Théorème 5.1. *Soit $M \in \mathbb{K}^{n\times n}$, une matrice dyadique de signature r i.e. $M_{i,j} = r_{i\oplus j}$, et $P \in \mathbb{K}^{n\times n}$ une matrice de permutation dyadique dont la signature s admet un unique élèment non nul à l'indice d, $s = (0,\cdots,1,\cdots,0)$. Alors $M\cdot P$ est une matrice dyadique de terme général $(M\cdot P)_{i,j} = r_{i\oplus j\oplus d}$.*

Démonstration. On a $(M\cdot P)_{i,j} = \sum_{k=0}^{n-1} m_{i,k}\cdot p_{k,j}$.

Or $p_{k,j} = \begin{cases} 1 & \text{si } k\oplus j = d \\ 0 & \text{sinon} \end{cases}$, donc $(MP)_{i,j} = m_{i,j\oplus d} = r_{i\oplus j\oplus d}$ □

Supposons que le bloc auquel on s'intéresse a été multiplié par une matrice de permutation dyadique d'indice d_b, on a alors les relations suivantes à l'intérieur d'un bloc :

$$\begin{aligned}
\frac{1}{h_{bt+i\oplus j\oplus d_b}} &= \frac{1}{h_{bt+i\oplus d_b}} + \frac{1}{h_{bt+j}} + \frac{1}{h_{bt}} \\
&= \frac{1}{h_{bt+i\oplus d_b}} + \frac{1}{h_{bt+j}} + \frac{1}{h_{bt+d_b}} + \frac{1}{h_{bt}} + \frac{1}{h_{bt+d_b}} \\
&= \frac{1}{h_{bt+i\oplus d_b}} + \frac{1}{h_{bt+j\oplus d_b}} + \frac{1}{h_{bt+d_b}}
\end{aligned}$$

qui est un système d'équations de degré 3. On pose alors $h_{bt+i\oplus d_b} = x_{bt+i}$, et en mettant au même dénominateur, on obtient les équations intra-bloc

suivantes :

$$\begin{aligned} x_{bt+i}x_{bt+j}x_{bt} &+ \\ x_{bt+i\oplus j}x_{bt+j}x_{bt} &+ \\ x_{bt+i\oplus j}x_{bt+i}x_{bt} &+ \\ x_{bt+i\oplus j}x_{bt+i}x_{bt+j} &= 0 \end{aligned}$$

Un deuxième type d'équations, les équations inter-bloc pourrait être exploité : On remarque que l'on a :

$$\frac{1}{h_{bt+i}} = \frac{1}{h_{bt}} + \frac{1}{h_i} + \frac{1}{h_0}$$

La relation suivante reste donc invariante (elle ne dépend pas du bloc choisi) :

$$\frac{1}{h_{bt+i}} + \frac{1}{h_{bt}} = \frac{1}{h_i} + \frac{1}{h_0}$$

Après la permutation dyadique, cette relation devient :

$$\frac{1}{h_{bt+i\oplus d_b}} + \frac{1}{h_{bt\oplus d_b}} = \frac{1}{h_{i\oplus d_0}} + \frac{1}{h_{0\oplus d_0}}$$

en effectuant le changement de variable, on a :

$$\frac{1}{x_{bt+i}} + \frac{1}{x_{bt}} = \frac{1}{x_i} + \frac{1}{x_0}$$

.

5.1 Attaque quadratique

Le point de départ de cette attaque sont les « équations inter-bloc » :

$$\frac{1}{x_{bt+i}} + \frac{1}{x_{bt}} = \frac{1}{x_i} + \frac{1}{x_0} \tag{19}$$

où b est l'index du bloc et $0 \leq i \leq (t-1)$. On remarque que $b = 0$ et $i = 0$ donnent des équations triviales. L'idée principale est de supposer que l'attaquant connaît le premier bloc, donc qu'il connaît les valeurs de $x_0, \ldots, x_{t-1}$. Les équations (19) se réécrivent alors :

$$(x_i x_0)(x_{bt} + x_{bt+i}) + (x_i + x_0)(x_{bt+i}x_{bt}) = K_i(x_{bt} + x_{bt+i}) + C_i(x_{bt+i}x_{bt}) = 0 \tag{20}$$

où $K_1, \ldots, K_{t-1}$ et $C_1, \ldots, C_{t-1}$ sont des constantes connues.
La structure dyadique de la matrice publique offre un deuxième type d'équations, les « équations intra-bloc » :

$$\frac{1}{x_{bt+i\oplus j}} + \frac{1}{x_{bt+i}} + \frac{1}{x_{bt+j}} + \frac{1}{x_0} = 0 \tag{21}$$

où b est l'index du bloc, et $0 \leq i, j \leq (t-1)$.
Mises au même dénominateur, on obtient des équations polynomiales cubiques. Or si le but est de résoudre le système par linéarisation, avoir des équations de degré 3 est plutôt prohibitif. Néanmoins, on peut utiliser ces équations pour réduire les contraintes sur le premier bloc.
En effet, une hypothèse de cette attaque est que l'attaquant dispose du premier bloc. Mais, les équations (21) nous disent que pour connaître les t élèments du premier bloc, il suffit de connaître les éléments dont l'indice est une puissance de $\bullet$, *i.e.* $x_0, x_1, \ldots, x_{2^{\log(t)-1}}$. Il suffit donc de connaître $c = \log(t) + 1$ éléments pour monter cette attaque.

La valeur de c peut même être réduite à $\log(t) - 1$ en utilisant la théorie des codes. En effet, un code de Goppa admet plus qu'un support et qu'un polynôme de Goppa. Ainsi, on peut fixer x_0 et x_1.
On va donc donner un récapitulatif des données du système à résoudre : Les équations (20) sont non-triviales pour tous les blocs sauf le premier, et chaque bloc donne $t-1$ equations. Le nombre total d'équations est donc

$$nb\_equations = \left(\frac{n}{t} - 1\right) \times (t-1) \tag{22}$$

Quant au nombre de variables,on commence par n variables qui représentent le vecteur $h = (x_0, \ldots, x_{n-1})$. La relation $GH^T = 0$ permet d'écrire les k dernière variables en fonction $n - k = mt$ premières variables. Ensuite, en fixant le premier bloc, on élimine encore t variables du système. Le nombre final de variables du système est donc :

$$nb\_variables = n - k - t = (m-1)t \tag{23}$$

5.1.1 Résolution par linéarisation

La linéarisation est une technique de résolution des systèmes polynomiaux surdéterminés. Linéariser revient a remplacer chaque monôme apparaissant dans le système par une nouvelle variable.
Une réduction de Gauss permet ensuite de triangulariser le système et donc de déduire la solution. Néanmoins, ces équations doivent être linéairement indépendantes et le système doit être de rang plein. Il faut donc plus d'équations que de variables, or il y'a autant de variables que de monômes de degré inférieur ou égal à d où d est le degré des équations. Le nombre de monôme en n variables, de degré au plus d étant égal à $\binom{n+d}{d}$, on en conclut qu'il faut environ $\binom{n+d}{d}$ équations indépendantes pour linéariser le système. Dans notre cas, pour pouvoir linéariser, nous avons besoin de choisir m et t (On prend toujours $n = 2^m - t$)tels que :

$$\left(\frac{2^m - t}{t} - 1\right) \times (t-1) \geq \binom{(m-1)t + 2}{2} \tag{24}$$

Tableau (4) résume les valeurs de m et t pour lesquelles l'inégalité (24) est vérifiée.

	m
$t = 4$	≥ 10
$t = 8$	≥ 13
$t = 16$	≥ 15

Tableau 4 – Valeurs de m et t pour lesquelles l'inégalité (24) est vérifiée.

Néanmoins, expérimentalement, on a été capable de résoudre les cas $t = 4, m = 9$ et $t = 8, m = 12$ par linéarisation. Le tableau (5) résume les valeurs pour lesquelles la linéarisation a permis de résoudre le système. La complexité d'une linéarisation est exactement celle de l'algèbre linéaire

m	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
$t = 4$	•	•	•	•	•	•	•	•	•	•	• [a]	• [a]	• [a]		
$t = 8$				•	•	•	•	•	•	• [a]	• [a]	• [a]	• [a]	• [a]	• [a]
$t = 16$							•	• [a]	• [a]						

Tableau 5 – Valeurs pour lesquelles on a résolu le système expérimentalement en le linéarisant.

utilisée pour trianguliariser le système. Dans notre cas en l'occurrence, il s'agit d'échelonner une matrice $(m-1)t \times (m-1)t$, la complexité de cette étape est donc en $\mathcal{O}(((m-1)t)^\omega)$ où ω est la constante de l'algèbre linéaire. On suppose aussi que l'attaquant connaît les valeurs de $\log(t) - 1$ éléments de $\mathbb{F}_{2^m}$, ce qui donne une complexité totale de $\mathcal{O}(2^{(m(\log(t)-1))}((m-1)t)^\omega)$.

m	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$t = 4$	2^{23}	$2^{24,5}$	$2^{25,9}$	$2^{27,3}$	$2^{28,6}$	$2^{29,9}$	$2^{31,2}$	$2^{32,5}$	$2^{33,8}$	2^{35}	$2^{36,2}$	$2^{37,4}$	$2^{39,8}$	$2^{38,7}$	2^{41}	$2^{42,2}$	$2^{43,4}$
$t = 8$				2^{42}	$2^{44,4}$	$2^{46,7}$	2^{49}	$2^{51,3}$	$2^{53,6}$	$2^{55,8}$	2^{58}	$2^{60,2}$	$2^{62,5}$	$2^{64,7}$	$2^{66,8}$	2^{69}	$2^{71,2}$
$t = 16$							$2^{66,8}$	$2^{70,1}$	$2^{73,4}$	$2^{76,6}$	$2^{79,8}$	$2^{83,1}$	$2^{86,3}$	$2^{89,5}$	$2^{92,6}$	$2^{95,8}$	2^{99}

Tableau 6 – Complexité de résolution par linéarisation des systèmes dyadiques ($\omega = 2.8$).

5.1.2 Résolution par bases de Gröbner

Le tableau (6) résume la complexité dans les cas où une linéarisation est possible et où elle permet de résoudre le système. Mais qu'en est-il des paramètres où une linéarisation n'est pas possible. Le calcul d'une base de Gröbner dans ces cas devrait permettre de les résoudre. On peut même se permettre de rajouter les équations (21) de degré 3. Puisqu'une linéarisation n'est pas possible, le degré de régularité doit être ≥ 3.

a. Le système a été tronqué. Juste assez d'équations pour linéariser.

m	5	6	7	8	9	10	11	12	13	14
$t = 4$	4	3	3	3						
$t = 8$				3	3	3	3			
$t = 16$					3	3	3			

Tableau 7 – Degré de régularité de certains systèmes résolus par l'algorithme F_4

Références

[1] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, pp. 644–654, November 1976.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM J. Comput.*, vol. 26, pp. 1484–1509, October 1997.

[3] T. P. Berger, P.-L. Cayrel, P. Gaborit, and A. Otmani, "Reducing key length of the mceliece cryptosystem," in *Proceedings of the 2nd International Conference on Cryptology in Africa : Progress in Cryptology*, AFRICACRYPT '09, (Berlin, Heidelberg), pp. 77–97, Springer-Verlag, 2009.

[4] R. Misoczki and P. S. Barreto, "Selected areas in cryptography," ch. Compact McEliece Keys from Goppa Codes, pp. 376–392, Berlin, Heidelberg : Springer-Verlag, 2009.

[5] J.-C. Faugère, A. Otmani, L. Perret, and J.-P. Tillich, "Algebraic cryptanalysis of mceliece variants with compact keys," in *EUROCRYPT*, pp. 279–298, 2010.

[6] A. Otmani, J.-P. Tillich, and L. Dallot, "Cryptanalysis of two mceliece cryptosystems based on quasi-cyclic codes," *Mathematics in Computer Science*, vol. 3, no. 2, pp. 129–140, 2010.

[7] F. MacWilliams and N. Sloane, *The theory for error-correcting codes*. North-Holland mathematical library, North-Holland, 2006.