Chapitre I

Quelques notions sur les codes

"Un esprit en mal de notions doit d'abord s'approvisionner d'apparences."

F. Ponge

La sécurité des schémas d'identification proposés dans cette thèse dépend d'un problème issu de la théorie des codes correcteurs d'erreurs. De plus, le chapitre vi est entièrement dédié à l'étude d'une sous classe des codes de Goppa. C'est pourquoi, dans cette première partie, les définitions et théorèmes de base de la théorie des codes sont rappelés. La liste des définitions pourra paraître assez "rébarbative" mais elle donnera au lecteur le "matériel" nécessaire à une bonne compréhension des notions qui seront mises en avant dans les chapitres suivants. Aucune démonstration ne sera faite, néanmoins on trouvera aisément ces dernières dans [MaSl]. Dans le dernier paragraphe, nous introduisons des définitions qui seront utilisées au chapitre iii.

CONSIDÉRONS le problème $\mathcal{P}$ suivant : Soient H une matrice $k \times n$ quelconque de rang k ($k \leqslant n$) à coefficients dans $\mathbb{F}_q$ (le corps fini à q éléments), i un élément de $\mathbb{F}_q^k$ et soit $S = \{z \in \mathbb{F}_q^n, Hz^{\mathrm{t}} = i\}$. Trouver un élément $s \in S$ possédant le plus petit nombre de composantes non-nulles.

En règle générale, ce problème est "difficile" (cf. Chap. ii). Si H ne possède pas de structures particulières, la seule solution consiste à parcourir les q^{n-k} éléments de l'ensemble S. Une telle recherche s'avère rapidement impraticable. A titre d'exemple, pour $q = 2$, si on suppose que $n = 2k$, alors il devient impossible de résoudre ce problème dès que $k > 50$.

La théorie des codes permet de construire facilement des instances particulières du problème $\mathcal{P}$, dont une solution peut être trouvée en temps polynomial (ceci sans poser de conditions sur la taille de la matrice H).

§1. Généralités

DÉFINITION 1.1. *Un* code *sur $\mathcal{A}$ de longueur n est un sous-ensemble C de $\mathcal{A}^n$. L'ensemble $\mathcal{A}$ est appelé* l'alphabet*, n la* longueur *du code C et les éléments de C sont appelés les* mots *du code.*

DÉFINITION ET PROPOSITION 1.2. *Soient $\mathcal{A}$ un alphabet, $x = (x_1, \dots, x_n)$ et $y = (y_1, \dots, y_n)$ deux éléments de $\mathcal{A}^n$. L'application*

$$\begin{array}{rccl} d_H\colon & \mathcal{A}^n \times \mathcal{A}^n & \longrightarrow & \mathbb{N} \\ & (x, y) & \longmapsto & d_H(x, y) = \#\{i \mid x_i \neq y_i, 1 \leqslant i \leqslant n\} \end{array}$$

est une distance appelée distance de Hamming

DÉFINITION 1.3. *On appelle* distance minimale *d'un code C sur $\mathcal{A}$ l'entier*

$$d_{\min} = \min\{d_H(x, y) \mid x, y \in C \text{ et } x \neq y\}.$$

Remarque : Il est important de noter ici la difficulté de recherche de la distance minimale d'un code ne comportant aucune structure particulière. On doit alors comparer deux à deux les mots du code, ce qui donne pour un code constitué de M mots, $\frac{(M-1)(M-2)}{2}$ comparaisons à effectuer.

On s'intéressera plus particulièrement aux cas où $\mathcal{A}$ est un corps fini et C est un $\mathcal{A}$-espace vectoriel. Ce sont les *codes linéaires*. Du fait de la structure d'espace vectoriel de C, la recherche de la distance minimale ne nécessite plus que M comparaisons.

§2. Les codes linéaires

DÉFINITION 2.1. *Un* code linéaire *de longueur n sur $\mathbb{F}_q$ et de* dimension k *(on dira un* code $[n, k]$*) est un* sous-espace vectoriel *de $\mathbb{F}_q^n$ de* dimension k*.*

PROPRIÉTÉ 2.2. *Si C est un code linéaire $[n, k]$ sur $\mathbb{F}_q$, alors le nombre de mots de C est q^k.*

DÉFINITION 2.3. *Pour un code linéaire $[n, k]$, on parlera du* poids *d'un élément, défini comme suit*

$$\omega(x) = d_H(x, 0) = \#\{i \mid x_i \neq 0, 1 \leqslant i \leqslant n\}.$$

PROPRIÉTÉ 2.4. *Si C est un code linéaire,*

$$\{d_H(x, y), (x, y) \in C \times C\} = \{\omega(x), x \in C\}.$$

DÉFINITION 2.5. *Le* poids minimum *des mots non-nuls d'un code linéaire s'appelle le* poids du code*.*

Remarques :

- La recherche de la distance minimale d'un code linéaire revient donc à trouver un mot de poids minimum. Ce qui nécessite, à priori, le calcul des q^k mots du code,
- Plus généralement, la recherche d'un mot de poids p, dans un code linéaire binaire, est un problème $\mathcal{NP}$-dur. Une méthode heuristique pour estimer le nombre de mots de poids p dans un code $C[n,k]$, sans structure particulière, est de considérer le nombre N_p de mots de poids p dans $\mathbb{F}_q^n$ et de le ramener en proportion dans C (i.e. multiplier N_p par $\frac{q^k}{q^n}$), ce qui donne
$$\frac{(q-1)^p}{q^{n-k}}\binom{n}{p}.$$
Ainsi, dans la pratique et sans stratégie particulière, on doit s'attendre pour obtenir un tel mot à effectuer en moyenne $\frac{q^n}{(q-1)^p}\binom{n}{p}^{-1}$ tirages aléatoires parmi les mots du code. Le calcul d'un mot du code nécessitant environ $(2k-1)(n-k)$ opérations sur $\mathbb{F}_q$ (en considérant une matrice génératrice mise sous forme systématique), le nombre d'opérations, sur $\mathbb{F}_q$, nécessaires à la recherche d'un mot de poids p est dans ce cas
$$(2k-1)(n-k)\frac{q^{n-k}}{(q-1)^p}\binom{n}{p}^{-1}.$$
- Si la distribution des poids est voisin de la distribution binomiale, la méthode ci-dessus aboutira rapidement si le poids cherché est voisin de $\frac{n}{2}$,
- La complexité d'attaque des protocoles décrits au chapitre iii est équivalente à la complexité de recherche d'un mot de poids p dans un code linéaire binaire quelconque. Cependant, il existe des algorithmes probabilistes (cf. Chap. ii) qui, pour certains paramètres, s'avèrent être très efficaces pour résoudre ce problème.

Pour un code de grande dimension, il est impossible de donner sa description en dressant la liste de tous les mots qu'il contient. L'avantage des codes linéaires est qu'ils peuvent être entièrement déterminés par une de leurs bases en tant que sous-espace vectoriel.

2.1. Matrices génératrices

DÉFINITION 2.1.1. *Une* matrice génératrice *d'un code linéaire C sur le corps $\mathbb{F}_q$ est une matrice sur $\mathbb{F}_q$ dont les lignes forment une base de C.*

PROPRIÉTÉ 2.1.2. *Soit C un code linéaire $[n, k]$ sur le corps $\mathbb{F}_q$. Alors :*

i) Toute matrice génératrice est une matrice $(k \times n)$ sur $\mathbb{F}_q$, dont le rang est k.

ii) Inversement, toute matrice $(k \times n)$ sur $\mathbb{F}_q$ de rang k, est une matrice génératrice d'un code $[n, k]$ sur $\mathbb{F}_q$.

2.2. Matrices de contrôle

DÉFINITION 2.2.1. *Le* code orthogonal *de C est l'espace vectoriel orthogonal de C pour le produit scalaire usuel.*

PROPRIÉTÉ 2.2.2. *Le code orthogonal d'un code $[n, k]$ est un code $[n, n-k]$.*

DÉFINITION 2.2.3. *On appelle* matrice de contrôle *de C, toute matrice génératrice de son orthogonal.*

PROPRIÉTÉ 2.2.4. *Soit H une matrice de contrôle d'un code $C[n, k]$. Un mot $x = (x_1, \ldots, x_n)$ appartient au code C* si et seulement si $Hx^{\mathrm{t}} = 0$.

Remarque : par rapport au problème $\mathcal{P}$, défini dans l'introduction, pour $i = 0$, rechercher dans S un vecteur possédant le plus petit nombre de composantes non nulles revient à trouver un mot de poids minimum dans le code C.

DÉFINITION 2.2.5. *Soit $C[n, k]$ un code linéaire sur $\mathbb{F}_q$, soit $y = (y_1, \ldots, y_n) \in \mathbb{F}_q^n$; on appelle* syndrome *de y le vecteur de $\mathbb{F}_q^{n-k}$ défini par*

$$S = Hy^{\mathrm{t}}.$$

Le problème $\mathcal{P}$ revient donc à trouver un élément de poids minimum admettant pour syndrome le vecteur i. La théorie des codes donne, dans certains cas, le nombre d'éléments de poids minimum dont le syndrome vaut i.

PROPOSITION 2.2.6. *Soient* $x \in C$ *et* $B(x,t) = \{y \in \mathbb{F}_q^n, d_H(x,y) \leqslant t\}$ $(t > 0)$. *Alors* $\forall (x,y) \in C \times C$ *(*$x \neq y$*),*

$$B(x,t) \cap B(y,t) = \emptyset \iff \forall i \in \mathbb{F}_q^k,\ \mathrm{card}\{z \mid \omega(z) \leqslant t \text{ et } Hz^{\mathrm{t}} = i\} \leqslant 1.$$

On dit alors que le code C vérifie la condition de décodage d'ordre t. C'est-à-dire que pour tout élément y, de $\mathbb{F}_q^n$, il existe au plus un élément x du code C, tel que $d_H(x,y) \leqslant t$. En d'autres termes, si G est la matrice génératrice du code C et si on pose $x = mG$, il existe au plus un mot e, de poids inférieur ou égal à t, tel que $y = mG + e$. Si effectivement, ce mot existe, alors la théorie des codes permet de retrouver de façon déterministe la quantité mG, connaissant seulement y et G. Dans la terminologie usuelle, e est appelée le *mot erreur*.

Un des domaines d'étude, de la théorie du codage, est la construction de codes pour lesquels, il existe un algorithme polynomial permettant de retrouver cette solution unique.

Dans le cadre des schémas d'identification, que nous proposons, un problème plus général que le problème $\mathcal{P}$ est utilisé. La solution recherchée n'est pas forcément de poids minimum, mais de poids p, p étant un paramètre fixé du schéma. Le problème se transforme donc en la recherche d'un mot de poids donné et de syndrome donné. Ceci est équivalent (cf. Chap. ii) à trouver un mot de poids $p+1$ dans un certain code linéaire.

PROPRIÉTÉ 2.2.7. *Soient* $h_1, \dots, h_n$ *les colonnes de* H. *On a*

$$H(c_1, \dots, c_n)^{\mathrm{t}} = h_1c_1 + h_2c_2 + \cdots + h_nc_n.$$

Ainsi il existe un mot du code de poids r *si et seulement s'il existe une* combinaison linéaire *à* coefficients non-nuls*, de* r *colonnes de* H*, qui soit elle-même nulle.*

Conséquence : Le poids minimum de C est le plus petit entier r non-nul tel qu'il existe une combinaison linéaire à coefficients non-nuls, de r colonnes de H, qui soit elle-même nulle.

Ainsi, il existe deux algorithmes probabilistes, "naïfs", permettant de chercher un mot de poids p dans un code $C[n,k]$ sans structure particulière :

- Le premier consiste à générer aléatoirement des mots du code et conduit en moyenne à

$$(2k-1)(n-k)\frac{q^{n-k}}{(q-1)^p}\binom{n}{p}^{-1}$$

opérations sur $\mathbb{F}_q$ pour aboutir (si un tel poids existe !),

- Le second consiste à générer aléatoirement des mots de poids p. On vérifie alors à l'aide de la matrice H, s'ils appartiennent au code. Ceci nécessite

$$q^{(n-k)}(2p-1)(n-k)$$

opérations sur $\mathbb{F}_q$.

Exemple : Notons Ω, le nombre d'opérations effectuées pour l'une ou l'autre de ces méthodes, voici l'évolution de ce paramètre, en fonction de p, pour $q = 2, n = 600, k = 300$:

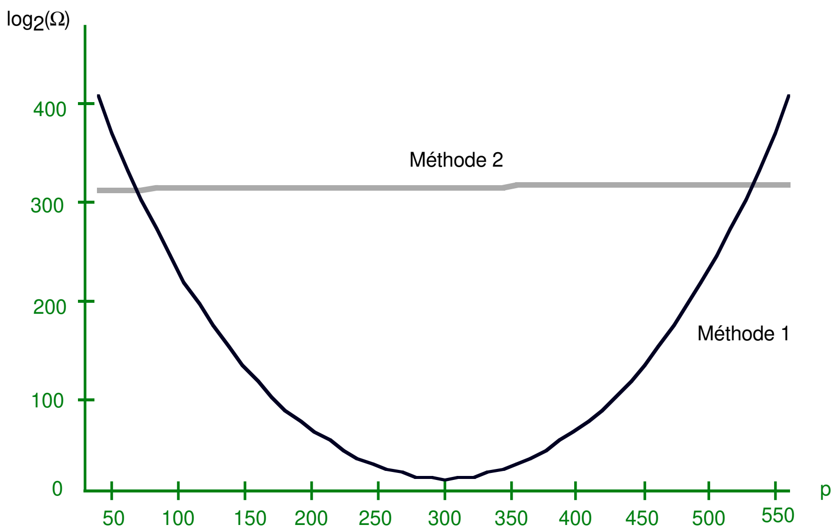

Fig. 1. Complexité des algorithmes naïfs

Dans le chapitre ii, des algorithmes probabilistes, permettant de réduire la complexité des calculs donnés ci-dessus, sont décrits. Ces derniers sont particulièrement performants lorsque p est petit.

Le paragraphe suivant présente les notions essentielles qui seront utilisées au chapitre vi.

§3. Les codes cycliques

DÉFINITION 3.1. *Un code C est dit* cyclique *si :*

i) *C est un code linéaire,*

ii) *Si $(c_0, \dots, c_{n-1}) \in C$, alors $(c_{n-1}, c_0, \dots, c_{n-2}) \in C$.*

On peut aussi représenter un code cyclique sous forme polynomiale en faisant correspondre à chaque mot c de C, le polynôme

$$c(x) = c_0 + c_1 x + \cdots + c_{n-1} x^{n-1}.$$

Le shift de $c(x)$ est le polynôme $xc(x)$ calculé modulo $x^n - 1$

Théorème 3.2. *Un code linéaire C sur $\mathbb{F}_q$ est cyclique si et seulement si sa représentation polynomiale $C(x)$ est un idéal de $\mathbb{F}_q[X]/(X^n - 1)$.*

Théorème 3.3. *La représentation polynomiale d'un code cyclique est formée par tous les multiples (modulo $x^n - 1$) d'un même polynôme. Un tel polynôme s'appelle un* générateur *du code cyclique.*

Les codes cycliques forment une classe de codes possédant de bonnes propriétés, et sont les plus utilisés dans les communications. Parmi eux, on distingue certains codes dont une estimation de la distance minimale est facilement calculable avant la construction du code, et pour lesquels il existe un algorithme polynomial de décodage. C'est le cas des codes BCH (Bose, Chauduri et Hoquenghem), des codes de Reed-Solomon Généralisés (codes GRS) et des codes de Goppa. Dans les deux prochains paragraphes, une brève description des codes BCH et des codes GRS est donnée. Ceci afin de préparer le lecteur à l'étude du chapitre vi concernant les codes de Goppa. En effet, d'une part, ces derniers peuvent être présentés comme étant une "généralisation des codes BCH" et d'autre part ils sont la restriction à $\mathbb{F}_q$ d'un code GRS défini sur $\mathbb{F}_{q^m}$.

3.1. Les codes BCH

Définition 3.1.1. *Soit C un code cyclique de longueur n sur $\mathbb{F}_q$ et soit $g(x)$ son polynôme générateur. Soient $\mathbb{F}_{q^m}$ le corps de décomposition de $x^n - 1$ et $\beta \in \mathbb{F}_{q^m}$, une racine primitive $n^{\text{ième}}$ de l'unité. Alors, C est un code BCH de distance δ si δ est le plus grand entier vérifiant*

$$\exists\, b \in \mathbb{Z},\ g(\beta^b) = g(\beta^{b+1}) = \cdots = g(\beta^{b+\delta-2}) = 0.$$

Définition 3.1.2. *Un code BCH est dit BCH au* sens strict *si $b = 1$.*

Théorème 3.1.3. (borne BCH) *Soit un code BCH de distance δ, sa distance minimale d vérifie*

$$d \geqslant \delta.$$

3.2. Les codes de Reed-Solomon généralisés

DÉFINITION 3.2.1. *Soit* $\alpha = (\alpha_1, \ldots, \alpha_n)$*, où les* α_i *sont des éléments distincts de* $\mathbb{F}_{q^m}$*, et soit* $v = (v_1, \ldots, v_n)$ *où les* v_i *sont des éléments non-nuls (mais non nécessairement distincts) de* $\mathbb{F}_{q^m}$*. Alors le* code de Reed-Solomon Généralisé, *de longueur* n *et de dimension* k*, noté* $GRS_k(\alpha, v)$*, est déterminé par tous les vecteurs de la forme*

$$(v_1 F(\alpha_1), \ldots, v_n F(\alpha_n))$$

où F(x) décrit l'ensemble des polynômes de degré $< k$ *à coefficients dans* $\mathbb{F}_{q^m}$*.*

PROPOSITION 3.2.2. *Une matrice de contrôle d'un* $GRS_k(\alpha, v)$ *est :*

$$H = \begin{pmatrix} y_1 & y_2 & \cdots & y_n \\ y_1\alpha_1 & y_2\alpha_2 & \cdots & y_n\alpha_n \\ y_1\alpha_1^2 & y_2\alpha_2^2 & \cdots & y_n\alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ y_1\alpha_1^{n-k-1} & y_2\alpha_2^{n-k-1} & \cdots & y_n\alpha_n^{n-k-1} \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 1 & \cdots & 1 \\ \alpha_1 & \alpha_2 & \cdots & \alpha_n \\ \alpha_1^2 & \alpha_2^2 & \cdots & \alpha_n^2 \\ \vdots & \vdots & \vdots & \vdots \\ \alpha_1^{n-k-1} & \alpha_2^{n-k-1} & \cdots & \alpha_n^{n-k-1} \end{pmatrix} \begin{pmatrix} y_1 & & & & 0 \\ & y_2 & & & \\ & & \cdot & & \\ & & & \cdot & \\ & & & & \cdot \\ 0 & & & & y_n \end{pmatrix}$$

où $y = (y_1, \ldots, y_n)$ *(*$y_i \in \mathbb{F}_{q^m}$*,* $y_i \neq 0$*), vérifie* $GRS_k(\alpha, v)^\perp = GRS_{n-k}(\alpha, y)$*. C'est-à-dire que cette matrice est la* matrice génératrice *du code* $GRS_{n-k}(\alpha, y)$*.*

§4. L'opérateur Trace

Dans ce paragraphe, les propriétés essentielles de l'opérateur trace sont rappelées. Un des schémas d'identification présenté dans le chapitre ii repose essentiellement sur les caractéristiques de cet opérateur. Il en est de même, pour les résultats obtenus au chapitre vi concernant les codes de Goppa.

DÉFINITION 4.1. *Soit* q *une puissance d'un nombre premier* p*, l'opérateur* trace de $\mathbb{F}_{q^m}$ sur $\mathbb{F}_q$ *est défini par :*

$$\begin{array}{rccc} \mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}\colon & \mathbb{F}_{q^m} & \longrightarrow & \mathbb{F}_q \\ & z & \longmapsto & \sum_{i=0}^{m-1} z^{q^i} \end{array}.$$

PROPRIÉTÉ 4.2. $\forall \lambda \in \mathbb{F}_q$, $\forall x, y \in \mathbb{F}_{q^m}$,

$$\mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(\lambda x + y) = \lambda \mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(x) + \mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(y).$$

Remarque : C'est essentiellement cette propriété qui sera utilisée dans l'un des schémas d'identification proposé au chapitre iii.

PROPRIÉTÉ 4.3. $\forall j = 0, \ldots, q-1$,

$$\mathrm{card}\{z \in \mathbb{F}_{q^m}, \mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(z) = j\} = q^{m-1}.$$

PROPRIÉTÉ 4.4. $\forall \beta \in \mathbb{F}_{q^m}$ $\mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(\beta^q) = \mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(\beta)^q = \mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(\beta)$. *De plus, si* $q = p^j$ $(j > 1)$, *alors* $\mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(\beta^p) = \mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(\beta)^p$.

PROPOSITION 4.5. $\mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(1) \equiv m \pmod p$.

PROPOSITION 4.6. *Si* $\{\beta_1, \ldots, \beta_m\}$ *est une base de* $\mathbb{F}_{q^m}$ *sur* $\mathbb{F}_q$, *alors il existe un entier* i *tel que* $\mathrm{Tr}_{\mathbb{F}_{q^m}:\mathbb{F}_q}(\beta_i) \neq 0$.

PROPOSITION 4.7. *Si* $q = 2^m$, *alors* $\{z \in \mathbb{F}_{q^2}, \mathrm{Tr}_{\mathbb{F}_{q^2}:\mathbb{F}_q}(z) = 0\} = \mathbb{F}_q = \{\varepsilon^q + \varepsilon, \varepsilon \in \mathbb{F}_{q^2}\}$ *(à un isomorphisme près).*

§5. Complexité des calculs dans les corps finis

Nous rappelons, dans ce paragraphe, le nombre d'opérations binaires (additions, multiplications) devant être effectuées pour multiplier deux éléments du corps $\mathbb{F}_{2^m}$. Ceci afin de pouvoir donner une estimation de la complexité des calculs des différents protocoles d'identification proposés.

De plus, certaines notations utilisées dans le chapitre iii sont introduites.

Notation: Soit $P(x) = \sum_{i=0}^{m} P_i x^i$ un polynôme irréductible sur $\mathbb{F}_2$, de degré m. On note W_P le poids de Hamming du vecteur $(P_0, \ldots, P_m)$.

DÉFINITION 5.1. *Soient* $\{\beta_1, \ldots, \beta_m\}$ *une base de* $\mathbb{F}_{2^m}$, *et* $\gamma = \sum_{i=1}^{m} \gamma_i \beta_i$ *un élément quelconque de* $\mathbb{F}_{2^m}$, *le* β*-poids de* γ, $\omega_\beta(\gamma)$, *est l'entier défini par*

$$\omega_\beta(\gamma) = \omega((\gamma_1, \ldots, \gamma_m)).$$

DÉFINITION 5.2. *Soit* $\beta = \{1, \alpha, \ldots, \alpha^{m-1}\}$ *une base de* $\mathbb{F}_{2^m}$, *où* α *est une racine d'un polynôme irréductible de degré* m. *Soit* $\gamma = \sum_{i=0}^{m-1} \gamma_i \alpha^i$ *un élément quelconque de* $\mathbb{F}_{2^m}$, *le* β*-exposant de* γ, $I_\beta(\gamma)$, *est l'entier défini par*

$$I_\beta(\gamma) = \max\{i \mid \gamma_i \neq 0, \quad 0 \leqslant i \leqslant m-1\}.$$

5.1. Multiplication classique

PROPOSITION 5.1.1. *Soit $\beta = \{1, \alpha, \ldots, \alpha^{m-1}\}$ une base de $\mathbb{F}_{2^m}$, où α est une racine d'un polynôme irréductible $P(x)$. Soient γ et ρ deux éléments quelconques de $\mathbb{F}_{2^m}$ tels que $\omega_\beta(\gamma) \leqslant \omega_\beta(\rho)$. Le produit $\rho\gamma$ nécessite au plus $m(\omega_\beta(\gamma) - 1) + I_\beta(\gamma)(W_P - 2)$ additions binaires.*

Preuve. On a $\gamma = \sum_{i=0}^{m-1} \gamma_i \alpha^i$. Il suffit de remarquer que, d'une part

$$\rho\gamma = \gamma_0\rho + \gamma_1(\rho\alpha) + \cdots + \gamma_{m-1}(\rho\alpha^{m-1})$$

et que, d'autre part la multiplication par α nécessite au plus $W_P - 2$ additions binaires. □

COROLLAIRE 5.1.2. *Si le polynôme $P(x)$ est un trinôme irréductible, alors le nombre d'opérations binaires nécessaires au calcul de $\rho\gamma$ dépend essentiellement de $\omega_\beta(\gamma)$.*

Remarque : Cette propriété sera souvent utilisée au cours du chapitre iii.

En pratique, cette multiplication nécessite l'utilisation de deux registres à décalage. Le premier permet de calculer, pour i variant de 0 à $m-1$, le produit $\rho\alpha^i$. Le second registre est vide au début du processus, puis, à chaque étape, il effectue la somme entre $\rho\alpha^i$ et son contenu (si $\gamma_i \neq 0$) ; c'est un accumulateur. De plus, une zone mémoire doit être réservée afin de stocker γ.

Remarque : Etant donné que la multiplication par α nécessite un décalage et au plus une addition avec l'élément α^m, il suffit de connaître α^m, afin d'effectuer une multiplication classique entre deux éléments du corps $\mathbb{F}_{2^m}$.

5.2. La méthode de Berlekamp

En 1982, E. R. Berlekamp [BER2] a proposé une méthode de multiplication très efficace, et particulièrement adaptée à une implémentation sur circuits intégrés, lorsque l'un des deux opérandes est une constante. Il a utilisé, pour cela, la notion de base *trace duale*.

DÉFINITION 5.2.1. *Soit $\beta = \{\beta_1, \ldots \beta_m\}$ une base de $\mathbb{F}_{2^m}$. $\beta^* = \{\beta_1^*, \ldots \beta_m^*\}$ est appelée* base trace duale *de β, si*

$$\forall i, j \quad \mathrm{Tr}_{\mathbb{F}_{2^m}:\mathbb{F}_2}(\beta_i\beta_j^*) = \delta_{ij}.$$

PROPOSITION 5.2.2. *Soit γ un élément fixe de $\mathbb{F}_{2^m}$, et soit ρ un élément quelconque de ce corps. Dans la base β^*, on a*

$$\rho\gamma = \sum_{i=0}^{m-1} \mathrm{Tr}_{\mathbb{F}_{2^m}:\mathbb{F}_2}(\rho\gamma\beta_i)\beta_i^*.$$

PROPOSITION 5.2.3. *Posons* $\gamma = \sum_{j=0}^{m-1} \gamma_j \beta_j$ *et, pour tout* i, $\rho\beta_i = \sum_{j=0}^{m-1} \mu_{ij}\beta_j^*$, *alors* $\mathrm{Tr}_{\mathbb{F}_{2^m}:\mathbb{F}_2}(\gamma(\rho\beta_i))$ *ne dépend que de la position des bits non nuls de* γ. *Plus précisément, on a*

$$\mathrm{Tr}_{\mathbb{F}_{2^m}:\mathbb{F}_2}(\gamma(\rho\beta_i)) = \sum_k \mu_{ij_k},\ j_k \text{ vérifiant } \gamma_{j_k} \neq 0.$$

Le principe de Berlekamp est alors le suivant :

entrée : - un polynôme irréductible sur $\mathbb{F}_2$, $P(x)$, de degré m,
- la base $\beta = \{1, \alpha, \ldots, \alpha^{m-1}\}$ (α racine de $P(x)$),
- la base trace duale, β^*, associée à β,
- $\alpha^m = \sum_{i=0}^{m-1} P_i \alpha^i$,

début
$\nu \leftarrow (\nu_{m-1}, \ldots, \nu_0)$, $\nu_i = \mathrm{Tr}_{\mathbb{F}_{2^m}:\mathbb{F}_2}(\alpha^i \rho)$,
(décomposition de ρ dans la base β^*) ;
$V \leftarrow \sum_{i=0}^{m-1} \nu_i \beta_i^*$, ;
Pour $i = 0$ **à** $m-1$ **faire**
début
$\varepsilon_i \leftarrow \mathrm{Tr}_{\mathbb{F}_{2^m}:\mathbb{F}_2}(\gamma V) = \sum_k \nu_{i_k}$, i_k vérifiant $\gamma_{i_k} \neq 0$;
Calculer $\mathrm{Tr}_{\mathbb{F}_{2^m}:\mathbb{F}_2}(\alpha^m V) = \sum_k \nu_{i_k}$, i_k vérifiant $P_{i_k} \neq 0$;
$\nu \leftarrow (\mathrm{Tr}_{\mathbb{F}_{2^m}:\mathbb{F}_2}(\alpha^m V), \nu_{m-1}, \ldots, \nu_1)$,
(représentation de $\rho\alpha^{i+1}$, dans la base β^*) ;
$V \leftarrow \sum_{i=0}^{m-1} \nu_i \beta_i^*$;
fin
$\rho\gamma \leftarrow \sum_{i=0}^{m-1} \varepsilon_i \beta_i^*$;
fin.

sortie : - le produit $\rho\gamma$

Fig. 2. Multiplication de Berlekamp

PROPOSITION 5.2.4. *Le calcul de* $\rho\gamma$ *nécessite exactement* $m(\omega_\beta(\gamma) - 1 + W_P - 2)$ *additions binaires.*

COROLLAIRE 5.2.5. *Si* $P(x)$ *est un trinôme irréductible, alors la multiplication de* $\rho\gamma$ *ne dépend que du* β*-poids de* γ.

Cette méthode ne nécessite l'utilisation que d'un seul registre à décalage, R, qui contient successivement les différentes valeurs de ν. Il est inutile de stocker γ, il suffit pour effectuer les calculs de relier, à l'aide de portes xor, certaines cellules de R.

Remarque Importante : Il faut bien distinguer la mise en œuvre pratique de ces deux méthodes de multiplication.

La multiplication classique permet de multiplier deux éléments quelconques du corps $\mathbb{F}_{2^m}$. Pour cela, deux registres à décalage sont nécessaires, ainsi qu'une zone mémoire où α^m est stocké.

La multiplication de Berlekamp, permet de multiplier un élément quelconque de $\mathbb{F}_{2^m}$ par un élément constant γ. Il suffit d'un seul registre à décalage dont certaines cellules sont connectées par des portes `xor`, ce câblage dépendant du poids de γ.

En d'autres termes, la multiplication de Berlekamp permet d'effectuer un produit par une constante β, sans avoir à stocker cette dernière. Ceci sera utilisé dans les schémas d'identification décrits au chapitre iii.