



Master 1 de Cryptologie et Sécurité Informatique
Équipe-projet GRACE

Stage estival 2017

Journal de bord

Étude du cryptosystème de Chor-Rivest

Rémi CLARISSE

Tuteurs : Daniel AUGOT et Luca DE FEO

Table des matières

1	Le problème du sac à dos	5
1.1	Densité d'un sac à dos	5
2	Le théorème Bose-Chowla	6
3	Le cryptosystème Chor-Rivest	7
3.1	Description	7
3.2	Exemple	9
3.3	Transformation des messages	12
4	Les attaques avec divulgation partielle de la clé privée	12
4.1	Symétrie dans la clé privée	13
4.2	Attaque due à B. Chor et R. Rivest (g et d connus)	17
4.3	Attaque due à S. Vaudenay (g connu)	17
4.4	Attaque due à B. Chor et R. Rivest (t et d connus)	18
4.5	Attaque due à O. Goldreich (t connu) [<i>version modifiée</i>]	18
4.6	Attaque due à A. Odlyzko (σ connue)	19
5	L'attaque de Brickell	20
6	La cryptanalyse de Vaudenay	21
6.1	Attaque sachant $g^{(p^h-1)/(p^r-1)}$ et σ	22
6.2	Attaque sachant $g^{(p^h-1)/(p^r-1)}$	22
6.3	Test pour trouver $g^{(p^h-1)/(p^r-1)}$	27
6.4	Utilisation des c_i	28
7	Le calcul de logarithmes dans un groupe	31
7.1	Recherche exhaustive	31
7.2	Méthode de Pohlig et Hellman	32
7.3	Algorithme pas de bébé, pas de géant	35
8	La méthode rho de Pollard	36
8.1	Discussion liminaire : partitionner, marcher, chercher	36
8.2	Méthode rho de Pollard originale dans $(\mathbb{Z}/r\mathbb{Z})^\times$	37
8.3	Méthode rho de Pollard originale généralisée	40
9	L'algorithme $L(1/4 + o(1))$ de Joux	41
9.1	Travaux précédents le logarithme discret en petite caractéristique	41
10	La réduction de réseau Lenstra-Lenstra-Lovász	41
11	L'attaque Lagarias-Odlyzko sur les sacs à dos à densité faible	41
A	Le b.a.-ba de la théorie des groupes	42

B	L'anatomie des corps	42
B.1	Homomorphismes et caractéristique	43
B.2	Endomorphisme de Frobenius	44
B.3	Sous-corps premiers	45
B.4	Extensions de corps	46
B.5	Extensions algébriques	47
B.6	Extensions de décomposition	49
B.7	Corps finis	51
C	La théorie de Kummer	51
D	La théorie d'Artin-Schreier	51
E	L'étude ad hoc du polynôme $\nu(x)$	51
F	Le paradoxe des anniversaires	51

Introduction

En 1984, Benny Chor et Ronald L. Rivest ont proposé un cryptosystème [2] à clé publique dont la sécurité repose sur un problème NP-complet : le problème du sac à dos. Il a été cryptanalysé en 1998 par Serge Vaudenay.

Nous nous proposons ici d'étudier ce cryptosystème.

1 Le problème du sac à dos

Le problème du sac à dos est le problème NP-complet suivant :

Définition 1.1 (Problème du sac à dos). Soient a_0, \dots, a_{n-1}, h et S des entiers naturels. Le *problème du sac à dos* est alors le problème de décision : existe-t-il n entiers naturels x_0, \dots, x_{n-1} de sorte que

$$\sum_{i=0}^{n-1} x_i a_i = S \quad \text{et} \quad \sum_{i=0}^{n-1} x_i \leq h \quad ?$$

h est appelé le *poids* du sac à dos.

Un autre problème, toujours NP-complet, demande seulement à ce que les x_i soient 0 ou 1, sans se soucier du poids du sac : c'est le problème *subset sum*.

Les cryptosystèmes se basant sur un problème de sac à dos reposent sur le fait qu'il est difficile de trouver une solution à un tel problème, même quand il en existe une.

Dans de tels cryptosystèmes, l'utilisateur publie un ensemble a_i , tel que $0 \leq i \leq n-1$, et un poids h . Un message clair consiste alors en un vecteur d'entiers $\mathbf{m} = (x_0, \dots, x_{n-1})$ dont le poids est inférieur à h . Le chiffré de \mathbf{m} est noté :

$$E(\mathbf{m}) = \sum_{i=0}^{n-1} x_i a_i.$$

L'utilisateur choisit les éléments a_i de telle façon à lui permettre de résoudre assez facilement l'instance du problème du sac à dos pour recouvrer le message clair à partir de $E(\mathbf{m})$. Il s'agit alors d'une *trapdoor* : une instance à laquelle il est aisé de trouver une solution.

Une propriété appréciée des cryptosystèmes de type sac à dos est la facilité de chiffrement : il n'y a qu'à faire des additions. Vu que nous souhaitons coder un message via une instance de sac à dos, il serait préférable qu'il y ait une seule et unique solution au problème, pour pouvoir recouvrer le message de manière univoque.

1.1 Densité d'un sac à dos

Pour attaquer les cryptosystèmes reposant sur le problème du sac à dos, Jeffrey C. Lagarias et Andrew M. Odlyzko ont introduit la définition :

Définition 1.2 (Densité d'un sac à dos). La *densité d'un sac à dos* est le rapport entre le nombre d'éléments dans le sac sur la taille binaire des ces éléments.

Concrètement, si un sac à n éléments, que ceux-ci sont positifs et que son élément le plus grand est $s \in \mathbb{N}^*$, alors la densité de ce sac à dos est $n/\log_2(s)$.

Il s'avère qu'il existe une attaque sur les sac à dos à densité faible (< 0.94). Elle a été proposée par J. Lagarias et A. Odlyzko en 1985, et utilise la réduction de réseaux, comme l'algorithme LLL.

B. Chor et R. Rivest se sont donc inspirés du théorème de Bose-Chowla pour construire leur cryptosystème.

2 Le théorème Bose-Chowla

Donnons rapidement une définition pour rendre le propos plus clair :

Définition 2.1 (*h-fold sum*). Soit h un entier naturel. On appelle *h-fold sum* d'une suite, la somme d'exactly h termes de celle-ci, répétition possible.

Nous nous posons alors la question : étant donné n et h deux entiers naturels, existe-t-il une suite d'entiers naturels (a_0, \dots, a_{n-1}) telle que toutes ces *h-fold sums* soient distinctes ?

Il est facilement de construire de telle suite si les a_i croissent exponentiellement en n . Par exemple, la suite $(1, h, h^2, \dots, h^{n-1})$ à cette propriété (mais cela ne marche pas pour $h + 1$ termes, puisque $h^2 + h \cdot 1 = (h + 1) \cdot h$). Cependant, le sac à dos associé a une densité faible, le rendant vulnérable à l'attaque de J. Lagarias et A. Odlyzko.

En 1962, Raj Chandra Bose et Sarvadaman Chowla ont trouvé une façon de construire une telle suite avec $1 \leq a_i \leq n^h - 1$ pour tout $0 \leq i \leq n - 1$. Cela permet d'instancier un sac à dos avec une densité acceptablement proche de 1. Nous donnons ici une version modifiée du théorème, qui instruit sur l'inspiration du cryptosystème de Chor-Rivest.

Théorème 2.2 (Théorème de Bose-Chowla). Soient p un nombre premier et $h \geq 2$ un entier. Il existe une suite $(a_i)_{0 \leq i \leq p-1}$ d'entiers telle que :

1. pour tout $0 \leq i \leq p - 1$,

$$1 \leq a_i \leq p^h - 1,$$

2. si (x_0, \dots, x_{p-1}) et (y_0, \dots, y_{p-1}) sont deux vecteurs distincts d'entiers naturels de poids inférieur à h , alors

$$\sum_{i=0}^{p-1} x_i a_i \neq \sum_{i=0}^{p-1} y_i a_i.$$

Démonstration. Plaçons nous dans les corps finis $\text{GF}(p)$ et $\text{GF}(p^h)$, extension de degré h de $\text{GF}(p)$. Numérotions les éléments de $\text{GF}(p) = \{\alpha_0, \dots, \alpha_{p-1}\}$.

Soit t un élément de $\text{GF}(p^h)$ algébrique de degré h sur $\text{GF}(p)$, i.e. le polynôme minimal de t dans $\text{GF}(p)[x]$ est de degré h .

Soit g un élément primitif de $\text{GF}(p^h)$, i.e. un élément générateur du groupe multiplicatif $\text{GF}(p^h)^\times$, à savoir

$$\text{GF}(p^h)^\times = \{g^e : 0 \leq e \leq p^h - 1\}.$$

Considérons l'ensemble des translatés de t par un élément de $\text{GF}(p)$, à savoir :

$$t + \text{GF}(p) = \{t + \alpha_i : \alpha_i \in \text{GF}(p)\}.$$

Soit $a_i := \log_g(t + \alpha_i)$, où $\alpha_i \in \text{GF}(p)$, le logarithme de $t + \alpha_i$ en base g dans $\text{GF}(p^h)$. Alors les a_i sont tous des entiers de l'intervalle $[1, p^h - 1]$ et, montrons par l'absurde que, toutes les *h-fold sums* de la suite $(a_i)_{0 \leq i \leq p-1}$ sont distinctes.

Soient (x_0, \dots, x_{p-1}) et (y_0, \dots, y_{p-1}) deux vecteurs d'entiers naturels satisfaisant à :

$$(x_0, \dots, x_{p-1}) \neq (y_0, \dots, y_{p-1}),$$

$$\sum_{i=0}^{p-1} x_i \leq h, \quad \sum_{i=0}^{p-1} y_i \leq h,$$

$$\sum_{i=0}^{p-1} x_i a_i = \sum_{i=0}^{p-1} y_i a_i.$$

Il s'en suit l'égalité dans $\text{GF}(p^h)$:

$$g^{\sum x_i a_i} = g^{\sum y_i a_i},$$

et donc :

$$\prod_{i=0}^{p-1} (g^{a_i})^{x_i} = \prod_{i=0}^{p-1} (g^{a_i})^{y_i}.$$

En utilisant l'égalité $g^{a_i} = t + \alpha_i$ et en ne considérant que les x_i et y_i non-nuls, nous obtenons :

$$(t + \beta_1)^{x_{i_1}} (t + \beta_2)^{x_{i_2}} \cdots (t + \beta_l)^{x_{i_l}} = (t + \gamma_1)^{y_{j_1}} (t + \gamma_2)^{y_{j_2}} \cdots (t + \gamma_m)^{y_{j_m}},$$

où $\{\beta_1, \dots, \beta_l\}$ et $\{\gamma_1, \dots, \gamma_m\}$ sont deux sous-ensembles non-vides de $\text{GF}(p)$ de cardinal au plus h . Les deux polynômes de part et d'autre de la dernière égalité, sont donc distincts, unitaires, de degré au plus h et à coefficients dans $\text{GF}(p)$. Ainsi, en faisant la différence de ces deux polynômes, nous obtenons que t est racine d'un polynôme non-nul de degré au plus $h-1$ et dont les coefficients sont dans $\text{GF}(p)$. Cela contredit le fait que t est algébrique de degré h sur $\text{GF}(p)$. \square

Remarque 2.3. D'après la preuve, il est clair que, pour $l \leq h$, les l -fold sums de la suite $(a_i)_{0 \leq i \leq p-1}$ sont distinctes dans \mathbb{Z} , mais aussi modulo $p^h - 1$.

La supposition “ p est un nombre premier” peut être remplacée par “ p est une puissance de nombre premier”, sans que cela n'affecte le théorème et sa preuve.

Nous exposerons en section 7 diverses façon de calculer les logarithmes dans un corps fini.

3 Le cryptosystème Chor-Rivest

3.1 Description

Soit p^h une puissance de nombre premier. Considérons le corps fini $\text{GF}(p^h)$ dont il est supposé que sa représentation est publique, i.e. il existe un polynôme unitaire $P(x)$ public de degré h irréductible sur $\text{GF}(p)$ et les éléments de $\text{GF}(p^h)$ sont vus comme des polynômes modulo $P(x)$. Ainsi :

$$\text{GF}(p^h) = \frac{\text{GF}(p)[x]}{(P(x))}.$$

Notons $a = x \pmod{P(x)}$, la classe de x modulo $P(x)$ dans $\text{GF}(p^h)$, et choisissons $(1, a, a^2, \dots, a^{h-1})$ comme base du $\text{GF}(p)$ -espace vectoriel $\text{GF}(p^h)$. Ainsi $\text{GF}(p^h) = \text{GF}(p)[a]$. Considérons aussi une numérotation publique α du sous-corps premier $\text{GF}(p)$, i.e.

$$\{\alpha_0, \dots, \alpha_{p-1}\} = \text{GF}(p) \subset \text{GF}(p^h).$$

Les éléments, choisis aléatoirement, de la clé privée – à garder secrète – consistent en :

- un élément $t \in \text{GF}(p^h)$ de degré algébrique h sur $\text{GF}(p)$ (nous notons $\mu(x) \in \text{GF}(p)[x]$ le polynôme minimal de t sur $\text{GF}(p)$),
 - un générateur g du groupe $\text{GF}(p^h)^\times$,
 - un entier d tel que $0 \leq d \leq p^h - 2$,
 - une permutation σ de l'ensemble $\{0, \dots, p-1\}$.
- La clé publique – à être publiée – est alors les :

$$c_i = d + \log_g(t + \alpha_{\sigma(i)}) \pmod{p^h - 1},$$

quel que soit i tel que $0 \leq i \leq p-1$.

Étant donné qu'il faut calculer des logarithmes discrets pour fabriquer la clé publique, les paramètres choisis doivent permettre d'effectuer ces logarithmes facilement dans $\text{GF}(p^h)$. B. Chor et R. Rivest ont suggéré de prendre un entier premier p relativement petit et un exposant h friable, de sorte que nous puissions appliquer l'algorithme de Pohlig-Hellman (voir section 7). En l'occurrence, ils ont proposé de se placer dans les corps tels que $\text{GF}(197^{24})$, $\text{GF}(211^{24})$, $\text{GF}(243^{24})$ et $\text{GF}(256^{25})$.

L'espace des messages est l'ensemble des chaînes de p -bits de poids de Hamming égal à h . C'est-à-dire que le message à chiffrer doit d'abord être encodé en une chaîne de bits $m = [m_0 \dots m_{p-1}]$ telle que $m_0 + \dots + m_{p-1} = h$.

L'espace des chiffrés est $\mathbb{Z}/(q-1)\mathbb{Z}$ et le chiffré de m est :

$$E(m) = \sum_{i=0}^{p-1} m_i c_i \pmod{q-1}.$$

Pour déchiffrer, nous calculons :

$$G(t) := g^{E(m)-hd}$$

comme un polynôme en t à coefficients dans $\text{GF}(p)$ et de degré au plus $h-1$. Comme g est primitif dans $\text{GF}(p^h)$, i.e. l'ordre de g est $q-1$, nous cherchons à déterminer $E(m) - hd$ modulo $q-1$:

$$\begin{aligned} E(m) - hd &\equiv \left(\sum_{i=0}^{p-1} m_i c_i \right) - hd \pmod{q-1} \\ &\equiv \left(\sum_{i=0}^{p-1} m_i (d + \log_g(t + \alpha_{\sigma(i)})) \right) - hd \pmod{q-1} \\ &\equiv \left(hd + \sum_{i=0}^{p-1} m_i \log_g(t + \alpha_{\sigma(i)}) \right) - hd \pmod{q-1} \\ &\equiv \sum_{i=0}^{p-1} m_i \log_g(t + \alpha_{\sigma(i)}) \pmod{q-1} \end{aligned}$$

D'où, l'égalité dans $\text{GF}(p^h)$:

$$G(t) = g^{E(m)-hd} = \prod_{i=0}^{p-1} (t + \alpha_{\sigma(i)})^{m_i} = \prod_{m_i=1} (t + \alpha_{\sigma(i)}).$$

L'élément $G(t)$ est exprimé dans la base $(1, t, t^2, \dots, t^{h-1})$ du $\text{GF}(p)$ -espace vectoriel $\text{GF}(p^h)$, où $t = x \pmod{\mu(x)}$. Ainsi :

$$G(x) \equiv \prod_{m_i=1} (x + \alpha_{\sigma(i)}) \pmod{\mu(x)}.$$

Il existe donc un polynôme $\lambda(x) \in \text{GF}(p)[x]$ tel que :

$$G(x) = \lambda(x)\mu(x) + \prod_{m_i=1} (x + \alpha_{\sigma(i)}).$$

En raisonnant sur les degrés et du fait que $\mu(x)$ et $\prod (x + \alpha_{\sigma(i)})^{m_i}$ soient unitaires, nous déduisons que $\lambda(x) = -1$, dont il découle l'égalité de polynômes :

$$G(x) + \mu(x) = \prod_{m_i=1} (x + \alpha_{\sigma(i)}).$$

La factorisation de ce dernier permet de recouvrer le message m .

3.2 Exemple

Prenons $p := 17$ et $h := 6$, ainsi $p^h = 24\,137\,569$.

```
sage: p = 17
sage: h = 6
sage: q = p ** h
```

Le corps fini $\text{GF}(17^6)$ construit par **Sage** est, en l'occurrence,

$$\frac{\mathbb{F}_{17}[x]}{(P(x))}, \text{ où } P(x) := x^6 + 2x^4 + 10x^2 + 3x + 3 \in \mathbb{F}_{17}[x].$$

```
sage: K.<a> = FiniteField(q)
sage: P = a.minimal_polynomial()
```

La numérotation α du sous-corps premier choisie est :

$$(\alpha_0, \dots, \alpha_{16}) = (2, 12, 4, 1, 0, 10, 7, 8, 15, 16, 3, 5, 13, 9, 11, 6, 14)$$

```
sage: alpha = [K(i) for i in range (p)]
sage: shuffle (alpha)
```

Génération des clés

Nous prenons un élément $t \in \text{GF}(17^6)$ de degré algébrique 6 sur $\text{GF}(17)$. Comme plus haut, nous notons $a = x \pmod{P(x)}$, le t sélectionné est alors :

$$t := 9a^5 + 16a^4 + 10a^3 + 3a^2 + 12a + 12,$$

de polynôme minimal :

$$\mu(x) := x^6 + 9x^5 + 8x^4 + 14x^3 + x^2 + 11x + 6 \in \text{GF}(17)[x].$$

```

sage: while True :
sage:     t = K.random_element()
sage:     if t.minimal_polynomial().degree() == h :
sage:         break
sage: mu = t.minimal_polynomial()

```

Ensuite, comme élément primitif de $\text{GF}(17^6)$, nous prenons :

$$g := 2a^5 + 5a^4 + 14a^3 + 2a^2 + 10a + 16.$$

```

sage: while True :
sage:     g = K.random_element()
sage:     if g.multiplicative_order() == q - 1 :
sage:         break

```

L'entier d est pris égal à 1 530 545.

```

sage: d = randint (0, q - 2)

```

Enfin, nous choisissons une permutation σ de l'ensemble $\{0, \dots, 16\}$:

$$\sigma := (0, 10, 8, 5, 1, 6, 14)(2, 3, 9, 16, 13)(4, 12, 7, 15)(11).$$

```

sage: s = Permutations([i for i in range (p)]).random_element()

```

Nous avons fini de fabriquer la clé privée! Reste à construire la clé publique :

c_0	$:=$	21 667 185	c_1	$:=$	3 210 064	c_2	$:=$	6 070 281
c_3	$:=$	3 093 929	c_4	$:=$	19 945 987	c_5	$:=$	294 610
c_6	$:=$	4 230 580	c_7	$:=$	18 951 770	c_8	$:=$	7 364 695
c_9	$:=$	23 348 812	c_{10}	$:=$	7 918 908	c_{11}	$:=$	3 562 855
c_{12}	$:=$	6 735 636	c_{13}	$:=$	13 077 876	c_{14}	$:=$	11 303 489
c_{15}	$:=$	22 106 426	c_{16}	$:=$	18 193 975			

```

sage: c = [mod (d + log (t + alpha[s[i]], g), q - 1)
            for i in range (p)]

```

Chiffrement d'un message

Maintenant, donnons nous un message à chiffrer de longueur 17 et de poids de Hamming 6 :

$$m := [m_0 \cdots m_{16}] = [00100101100100100].$$

```

sage: m = [1 for i in range (h)] + [0 for i in range (p - h)]
sage: shuffle (m)

```

Chiffrons m :

$$e := 23\,410\,132.$$

```

sage: e = mod (sum ([m[i]*c[i] for i in range (p)]), q-1)

```

Déchiffrement du message

Nous souhaitons écrire g^{e-hd} comme un polynôme en t . Or **Sage** nous donne g^{e-hd} comme polynôme en a :

$$g^{e-hd} = a^5 + 11a^3 + 9a^2 + 15a + 1.$$

Pour parvenir à exprimer g comme nous le souhaitons, il faut effectuer un changement de base du $\text{GF}(17)$ -espace vectoriel $\text{GF}(17^6)$: passer de la base $\mathcal{A} := (1, a, a^2, \dots, a^{h-1})$ à la base $\mathcal{T} := (1, t, t^2, \dots, t^{h-1})$. La matrice de passage facile à calculer est celle qui passe de la base \mathcal{T} à la base \mathcal{A} : il suffit d'écrire dans **Sage** les différentes puissances de t , et **Sage** les exprime en fonction des puissances de a . Voici cette matrice :

$$\begin{bmatrix} 1 & 12 & 7 & 11 & 14 & 0 \\ 0 & 12 & 9 & 12 & 6 & 12 \\ 0 & 3 & 8 & 3 & 7 & 12 \\ 0 & 10 & 0 & 15 & 15 & 14 \\ 0 & 16 & 4 & 0 & 13 & 3 \\ 0 & 9 & 15 & 10 & 5 & 6 \end{bmatrix}$$

Celle qui nous intéresse est son inverse, la matrice de passage de la base \mathcal{A} à la base \mathcal{T} :

$$\begin{bmatrix} 1 & 5 & 10 & 12 & 15 & 11 \\ 0 & 8 & 8 & 0 & 11 & 5 \\ 0 & 13 & 13 & 9 & 5 & 1 \\ 0 & 9 & 10 & 3 & 11 & 9 \\ 0 & 12 & 6 & 2 & 16 & 8 \\ 0 & 7 & 16 & 2 & 13 & 11 \end{bmatrix}$$

```
sage: V = K.vector_space()
sage: M = Matrix (GF(p),
                  [V(t ** i) for i in range (h)]).transpose()
sage: Minv = M.inverse()
```

Comme g^{e-hd} vaut dans la base \mathcal{A} le vecteur $(1, 15, 9, 11, 0, 1)$, il est facile d'obtenir l'égalité :

$$g^{e-hd} = 10t^5 + 9t^4 + 12t^3 + 4t^2 + 10t + 3.$$

Notons $G(x) := 10x^5 + 9x^4 + 12x^3 + 4x^2 + 10x + 3 \in \text{GF}(17)[x]$. Ainsi le message est recouvré en factorisant le polynôme :

$$G(x) + \mu(x) = x^6 + 2x^5 + 9x^3 + 5x^2 + 4x + 9,$$

ce que nous faisons :

$$G(x) + \mu(x) = (x + 1)(x + 2)(x + 5)(x + 6)(x + 10)(x + 12) \in \text{GF}(17)[x].$$

Nous pouvons alors déterminer les α_i utilisés et nous savons alors que le bit $m_{\sigma^{-1}(i)}$ est à 1, alors que les autres sont nuls.

```
sage: A.<x> = PolynomialRing (GF(p))
sage: Q = A(list (Minv * V(g ** (e - h * d)))) + A(mu)
sage: beta = [p - Q.roots()[i][0] for i in range (h)]
```

Pour facilité l'explication, notons β_i , pour i tel que $0 \leq i \leq 5$, les éléments de $\text{GF}(17)$ tels que $-\beta_i$ est racine de $G(x) + \mu(x)$, les β_i étant tous distincts. Par exemple, $\beta_0 = 1, \beta_1 = 2, \dots, \beta_5 = 12$. Ainsi, pour tout i tel que $0 \leq i \leq 5$, il faut déterminer l'indice j , où $0 \leq j \leq 16$, tel que $\alpha_j = \beta_i$, et alors nous déduisons qu'il y a un 1 au $\sigma^{-1}(j)$ bit du message.

Par exemple, pour $\beta_0 = 1$: nous cherchons d'abord l'indice j tel que $\alpha_j = 1$, à savoir $j = 3$. Puis nous déterminons $\sigma^{-1}(3) = 2$. Et nous avons bien $m_2 = 1$. Idem, pour $\beta_2 = 5$: $\alpha_{11} = 5$, $\sigma^{-1}(11) = 11$ et nous constatons en effet que $m_{11} = 1$.

```
sage: sInv = [s.index(i) for i in range (p)]
sage: message = [0 for i in range (p)]
sage: for k in beta :
sage:     message[sInv [alpha.index(k)]] = 1
```

3.3 Transformation des messages

Nous avons supposé que tous les messages chiffrés via le cryptosystème étaient de longueur p et de poids h . Cependant, nous pouvons transformer un message binaire quelconque pour qu'il ait cette forme.

Étant donné un message binaire, découpons le en blocs de $\lfloor \log_2 \binom{p}{h} \rfloor$ bits. Chaque bloc est la représentation binaire d'un nombre n tel que $0 \leq n < \binom{p}{h}$. Pour faire correspondre ces nombres à des vecteurs binaires de poids h , nous utilisons l'ordre total induit par l'ordre lexicographique sur les vecteurs et l'ordre sur les entiers. Si n est supérieur à $\binom{p-1}{h-1}$, le premier bit correspondant au vecteur est mis à 1, sinon il est mis à 0. Puis p et h sont actualisés, et nous itérons jusqu'à ce que les p bits soient déterminés :

Algorithme 1 Algorithme de transformation d'un bloc de message en vecteur

Entrée : p, h et un bloc n du message d'origine tel que $0 \leq n < \binom{p}{h}$

Sortie : un vecteur y de taille p et de poids h

```
1: pour  $i$  allant de 1 à  $p$  faire
2:   si  $n \geq \binom{p-i}{h}$  alors
3:      $y_i \leftarrow 1$ 
4:      $n \leftarrow n - \binom{p-i}{h-1}$ 
5:      $h \leftarrow h - 1$ 
6:   sinon
7:      $y_i \leftarrow 0$ 
8:   fin si
9: fin pour
10: retourner  $y$ 
```

La transformation inverse est donnée par l'algorithme suivant :

4 Les attaques avec divulgation partielle de la clé privée

Nous examinons les attaques possibles proposées dans le papier de B. Chor et R. Rivest [2, Section VII, A] et dans celui de S. Vaudenay [10, Section 4]. No-

Algorithme 2 Algorithme de transformation d'un vecteur en bloc de message

Entrée : p , h et un vecteur y de taille p et de poids h **Sortie :** un bloc n du message d'origine

```
1:  $n \leftarrow 0$ 
2: pour  $i$  allant de 1 à  $p$  faire
3:   si  $y_i = 1$  alors
4:      $n \leftarrow n + \binom{p-i}{h}$ 
5:      $h \leftarrow h - 1$ 
6:   fin si
7: fin pour
8: retourner  $n$ 
```

tons aussi que Huber [5] propose une attaque plus alambiquée lorsque l'élément primitif g est connu.

Commençons par une observation de Serge Vaudenay [10, Section 3].

4.1 Symétrie dans la clé privée

Dans le cryptosystème de Chor-Rivest, nous choisissons la clé privée de façon aléatoire puis, à partir de celle-ci, nous calculons la clé publique. Le système repose sur la difficulté de trouver une clé secrète à partir de la clé publique. Cependant, nous remarquons qu'il y a plusieurs clés privées *équivalentes*, à savoir qu'il existe plusieurs jeux de clés privées qui donnent la même clé publique.

Affinons. Nous pouvons remplacer t et g par leur puissance p -ième, la clé publique reste inchangée car :

$$\log_{g^p} (t^p + \alpha_{\sigma(i)}) = \frac{1}{p} \log_g ((t + \alpha_{\sigma(i)})^p) = \log_g (t + \alpha_{\sigma(i)}).$$

Nous pouvons aussi remplacer (t, α_σ) par $(t + u, \alpha_\sigma - u)$, pour tout $u \in \text{GF}(p)$. Et enfin, nous pouvons remplacer (t, d, α_σ) par $(ut, d - \log_g(u), u\alpha_\sigma)$, quel que soit $u \in \text{GF}(p)^\times$. Cela donne donc au moins $hp(p-1)$ clés privées équivalentes. Il s'agit alors de déterminer une de ces clés.

Plus détaillé, peut-être abscons ...

Notons

$$\mathcal{U} := \text{GF}(p^h) \times \text{GF}(p^h) \times \frac{\mathbb{Z}}{(p^h - 2)\mathbb{Z}} \times \mathfrak{S}(\{0, 1, \dots, p-1\}),$$

où $\mathfrak{S}(\{0, 1, \dots, p-1\})$ est l'ensemble des bijections de l'ensemble $\{0, 1, \dots, p-1\}$.

Un élément (g, t, d, σ) de \mathcal{U} est une clé privée possible du cryptosystème si g est primitif et t est algébrique de degré h (il n'y a pas de restriction sur l'entier d et la permutation σ). Notons \mathcal{P} l'ensemble des clés privées possibles du cryptosystème.

Remarque 4.1. Le cardinal de \mathcal{U} est :

$$p^h \times p^h \times (p^h - 2) \times p! = p^{3h} p! - 2p^{2h} p!.$$

Le cardinal de \mathcal{P} est :

$$\phi(p^h - 1) \times \Xi(p, h) \times (p^h - 2) \times p!,$$

où $\Xi(p, h)$ est le nombre d'éléments algébriques de degré h dans le corps $\text{GF}(p^h)$ vu comme extension du corps $\text{GF}(p)$. Si h se factorise en $h = \prod_{1 \leq i \leq l} r_i^{\alpha_i}$, alors

$$\Xi(p, h) = \sum_{\substack{d \in \mathbb{N}^l \\ d \leq \alpha}} (-1)^{|\alpha-d|} \left(p^{\prod_j r_j^{d_j}} \right) \quad (???)$$

Définition 4.2 (Clés équivalentes). Deux clés privées de \mathcal{P} sont dites *équivalentes* si elles fabriquent la même clé publique. En d'autres termes, deux clés privées (g, t, d, σ) et (g', t', d', π) sont équivalentes, si on a l'égalité d'ensembles :

$$\{d + \log_g(t + \alpha_{\sigma(i)}) : 0 \leq i \leq p-1\} = \{d' + \log_{g'}(t' + \alpha_{\pi(i)}) : 0 \leq i \leq p-1\}.$$

Commençons par nous pencher sur une action de groupe sur l'ensemble des permutations $\mathfrak{S}(\{0, 1, \dots, p-1\})$:

Proposition 4.3. *Le groupe $G := \text{GF}(p)^\times \ltimes \text{GF}(p)$, muni de la loi*

$$(u, v) \cdot (u', v') = (uu', v + uv'),$$

l'élément neutre étant $(1, 0)$ et l'inverse de (u, v) étant $(u^{-1}, -vu^{-1})$, agit sur l'ensemble $\mathfrak{S} := \mathfrak{S}(\{0, 1, \dots, p-1\})$ par :

$$\begin{aligned} G \times \mathfrak{S} &\longrightarrow \mathfrak{S} \\ ((u, v), \sigma) &\longmapsto (u, v) \diamond \sigma, \end{aligned}$$

où la permutation $(u, v) \diamond \sigma$ est définie comme suit :

$$\alpha_{((u, v) \diamond \sigma)(i)} = u\alpha_{\sigma(i)} + v.$$

Démonstration. Si $(u, v) \in G$ et si $\sigma \in \mathfrak{S}$, alors $(u, v) \diamond \sigma$ est bien une permutation : en effet pour tous $i \neq j$, on a $\alpha_{((u, v) \diamond \sigma)(i)} \neq \alpha_{((u, v) \diamond \sigma)(j)}$. Montrons que nous avons bien une action de groupe : soit $\sigma \in \mathfrak{S}$, on a facilement que $(1, 0) \diamond \sigma = \sigma$, maintenant si (u, v) et (u', v') sont dans G , alors quel que soit i :

$$\begin{aligned} \alpha_{(u, v) \diamond ((u', v') \diamond \sigma)(i)} &= u\alpha_{(u', v') \diamond \sigma(i)} + v \\ &= u(u'\alpha_{\sigma(i)} + v') + v \\ &= uu'\alpha_{\sigma(i)} + uv' + v \\ &= \alpha_{((u, v) \cdot (u', v')) \diamond \sigma(i)}. \end{aligned}$$

Donc $(u, v) \diamond ((u', v') \diamond \sigma) = ((u, v) \cdot (u', v')) \diamond \sigma$. □

Proposition 4.4. *Soit $\sigma \in \mathfrak{S}$. Pour tous $i \neq j$, il existe un unique $(u, v) \in G$ tel que, si on note $\pi := (u, v) \diamond \sigma$,*

$$\pi(0) = i \quad \text{et} \quad \pi(1) = j.$$

Démonstration. Il faut prendre :

$$u = \frac{\alpha_i - \alpha_j}{\alpha_{\sigma(0)} - \alpha_{\sigma(1)}} \in \text{GF}(p)^\times \quad \text{et} \quad v = \frac{\alpha_j\alpha_{\sigma(0)} - \alpha_i\alpha_{\sigma(1)}}{\alpha_{\sigma(0)} - \alpha_{\sigma(1)}} \in \text{GF}(p).$$

□

Proposition 4.5. *L'action du groupe G partitionne l'ensemble \mathfrak{S} en $(p-2)!$ orbites de cardinal $p^2 - p$.*

Plus haut, nous avons noté trois façons de fabriquer des clés équivalentes à une clé privée donnée. Pour étayer cela, nous allons expliciter les groupes et action de groupes en jeu :

▷ le groupe $\mathfrak{A} = \{\text{id}_{\text{GF}(p^h)}, \text{Frob}, \text{Frob}^2, \dots, \text{Frob}^{h-1}\} = \langle \text{Frob} \rangle$, où Frob est l'automorphisme de Frobenius de $\text{GF}(p^h)/\text{GF}(p)$, dont la loi est la composition des applications, agit sur l'ensemble \mathcal{P} par :

$$\begin{aligned} \mathfrak{A} \times \mathcal{P} &\longrightarrow \mathcal{P} \\ (f, \mathbf{k}) = (f, (g, t, d, \sigma)) &\longmapsto f \odot \mathbf{k} = (f(g), f(t), d, \sigma). \end{aligned}$$

Pour une clé privée $\mathbf{k} \in \mathcal{P}$, l'orbite de \mathbf{k} possède h éléments.

▷ le groupe $\text{GF}(p)^\times$ muni de la multiplication agit sur l'ensemble \mathcal{P} par :

$$\begin{aligned} \text{GF}(p)^\times \times \mathcal{P} &\longrightarrow \mathcal{P} \\ (u, \mathbf{k}) = (u, (g, t, d, \sigma)) &\longmapsto u \otimes \mathbf{k} = (g, ut, d - \log_g(u), (u, 0) \diamond \sigma). \end{aligned}$$

L'action ici est peut-être moins évidente : soient u et u' dans $\text{GF}(p)^\times$ et soit $\mathbf{k} \in \mathcal{P}$, vérifions que $(u'u) \otimes \mathbf{k} = u' \otimes (u \otimes \mathbf{k})$.

$$\begin{aligned} u' \otimes (u \otimes \mathbf{k}) &= u' \otimes (g, ut, d - \log_g(u), (u, 0) \diamond \sigma) \\ &= (g, u'ut, d - \log_g(u) - \log_g(u'), (u', 0) \diamond ((u, 0) \diamond \sigma)) \\ &= (g, u'ut, d - (\log_g(u) + \log_g(u')), ((u', 0) \cdot (u, 0)) \diamond \sigma) \\ &= (g, (u'u)t, d - \log_g(u'u), (u'u, 0) \diamond \sigma) \\ &= (u'u) \otimes \mathbf{k}. \end{aligned}$$

Et pour tout $\mathbf{k} \in \mathcal{P}$, l'orbite de \mathbf{k} contient $p-1$ éléments.

▷ le groupe $\text{GF}(p)$ muni de l'addition agit sur l'ensemble \mathcal{P} par :

$$\begin{aligned} \text{GF}(p) \times \mathcal{P} &\longrightarrow \mathcal{P} \\ (v, \mathbf{k}) = (v, (g, t, d, \sigma)) &\longmapsto v \oplus \mathbf{k} = (g, t + v, d, (1, v) \diamond \sigma). \end{aligned}$$

Ici encore, vérifions qu'on ait bien une action : soient v et v' dans $\text{GF}(p)$ et soit $\mathbf{k} \in \mathcal{P}$, vérifions que $(v' + v) \oplus \mathbf{k} = v' \oplus (v \oplus \mathbf{k})$.

$$\begin{aligned} v' \oplus (v \oplus \mathbf{k}) &= v' \oplus (g, t + v, d, (1, v) \diamond \sigma) \\ &= (g, t + v + v', d, (1, v') \diamond ((1, v) \diamond \sigma)) \\ &= (g, t + v + v', d, ((1, v') \cdot (1, v)) \diamond \sigma) \\ &= (g, t + v + v', d, (1, v + v') \diamond \sigma) \\ &= (v' + v) \oplus \mathbf{k}. \end{aligned}$$

Pour tout $\mathbf{k} \in \mathcal{P}$, l'orbite de \mathbf{k} contient p éléments.

Théorème 4.6. *Le groupe $\mathfrak{A} \times \text{GF}(p)^\times \times \text{GF}(p)$ agit sur \mathcal{P} par :*

$$\begin{aligned} (\mathfrak{A} \times \text{GF}(p)^\times \times \text{GF}(p)) \times \mathcal{P} &\longrightarrow \mathcal{P} \\ ((f, u, v), (g, t, d, \sigma)) &\longmapsto \left(f(g), uf(t) - v, d - \log_{f(g)}(u), (u, v) \diamond \sigma \right). \end{aligned}$$

Le cardinal de l'orbite d'un élément est $hp(p-1)$. Chaque clé secrète est donc équivalente à au moins $hp(p-1) - 1$ autres clés secrètes.

Démonstration. La loi de groupe est : pour tous (f, u, v) et (f', u', v') dans $\mathfrak{A} \times \text{GF}(p)^\times \ltimes \text{GF}(p)$,

$$(f, u, v)(f', u', v') = (ff', uu', v + uv').$$

L'élément neutre de ce groupe est $(\text{id}_{\text{GF}(p^h)}, 1, 0)$ et l'inverse de (f, u, v) est l'élément $(f^{-1}, u^{-1}, -vu^{-1})$.

Montrons que nous avons bien une action. Soit $(g, t, d, \sigma) \in \mathcal{P}$ et calculons :

$$\begin{aligned} (\text{id}_{\text{GF}(p^h)}, 1, 0) \cdot (g, t, d, \sigma) &= (\text{id}_{\text{GF}(p^h)}(g), \text{id}_{\text{GF}(p^h)}(t), d, (1, 0) \diamond \sigma) \\ &= (g, t, d, \sigma). \end{aligned}$$

Soient $((f, u, v), (f', u', v')) \in (\mathfrak{A} \times \text{GF}(p)^\times \ltimes \text{GF}(p))^2$ et $(g, t, d, \sigma) \in \mathcal{P}$.

$$\begin{aligned} ((f, u, v)(f', u', v')) \cdot (g, t, d, \sigma) &= (ff', uu', v + uv') \cdot (g, t, d, \sigma) \\ &= (ff'(g), uu'(ff'(t)) - v - uv', d - \log_{ff'(g)}(uu'), (uu', v + uv') \diamond \sigma), \end{aligned}$$

D'autre part, $(f, u, v) \cdot ((f', u', v') \cdot (g, t, d, \sigma))$ vaut

$$(f, u, v) \cdot (f'(g), u'f'(t) - v', d - \log_{f'(g)}(u'), (u', v') \diamond \sigma),$$

et en remarquant que $\log_{f'(g)}(u') = \log_{ff'(g)}(u')$, car $u' \in \text{GF}(p)$, on a :

$$\begin{aligned} &= (ff'(g), uf(u'f'(t) - v') - v, d - \log_{ff'(g)}(uu'), (u, v) \diamond ((u', v') \diamond \sigma)) \\ &= (ff'(g), uu'(ff'(t)) - uv' - v, d - \log_{ff'(g)}(uu'), (uu', v + uv') \diamond \sigma). \end{aligned}$$

Montrons que cette action est libre : soient $(f, u, v) \in \mathfrak{A} \times \text{GF}(p)^\times \ltimes \text{GF}(p)$ et $(g, t, d, \sigma) \in \mathcal{P}$ tels que

$$(f, u, v) \cdot (g, t, d, \sigma) = (f(g), uf(t) - v, d - \log_{f(g)}(u), (u, v) \diamond \sigma) = (g, t, d, \sigma).$$

L'égalité $f(g) = g$ et le fait que g soit primitif impliquent que $f = \text{id}_{\text{GF}(p^h)}$. De même, $d - \log_g(u) = d$ donne $\log_g(u) = 0$, d'où $u = 1$. Du reste, nous en déduisons que $v = 0$. Ainsi, si (f, u, v) fixe (g, t, d, σ) , alors $(f, u, v) = (\text{id}_{\text{GF}(p^h)}, 1, 0)$: donc l'action est libre et chaque orbite contient $hp(p-1)$ éléments. \square

Théorème 4.7. Soit $(g, t, d, \sigma) \in \mathcal{P}$ une clé privée. Soit (g', t', d', π) une clé équivalente à (g, t, d, σ) , i.e. il existe $(f, u, v) \in \mathfrak{A} \times \text{GF}(p)^\times \ltimes \text{GF}(p)$ tel que :

$$(g', t', d', \pi) = (f, u, v) \cdot (g, t, d, \sigma).$$

Si $\#\#$ *condition* $\#\#$, alors $t \neq t'$.

Démonstration. Soit $(f, u, v) \in \mathfrak{A} \times \text{GF}(p)^\times \ltimes \text{GF}(p)$ tel que

$$t = uf(t) + v.$$

À savoir, pour un certain $0 \leq j \leq h-1$,

$$t = ut^{p^j} + v.$$

Ou en reformulant, il existe $(a, b) \in \text{GF}(p)^\times \times \text{GF}(p)$ tel que :

$$t^{p^j} + at + b = 0,$$

où en fait $a := -u^{-1}$ et $b := vu^{-1}$. Définissons le polynôme

$$\nu(x) := x^{p^j} + ax + b \in \text{GF}(p)[x].$$

Le polynôme minimal $\mu(x) \in \text{GF}(p)[x]$ de t est de degré h et divise $\nu(x)$. \square

Corollaire 4.8. *Sous ses conditions, le théorème précédent implique que dans l'orbite d'une clé privée (sous l'action que nous considérons), il n'existe pas deux éléments ayant la même seconde coordonnée (le t algébrique de degré h sur $\text{GF}(p^h)$). Ainsi, si nous fixons t dans la clé privée, il n'existe pas d'autre clé équivalente (atteignable par l'action de groupe) ayant le même t . En d'autres termes, fixer t revient à fixer la clé secrète (parmi ses clés équivalentes).*

Proposition 4.9. *Il est faux que nous pouvons choisir t et l'image de 0 et 1 par σ . (À prouver!!!)*

4.2 Attaque due à B. Chor et R. Rivest (g et d connus)

Sachant d , nous pouvons calculer $(b_0, \dots, b_{p-1}) := (c_0 - d, \dots, c_{p-1} - d)$. Soit $t' := g^{b_0}$, de sorte que :

$$t' = g^{\log_g(t + \alpha_{\sigma(0)})} = t + \alpha_{\sigma(0)}.$$

D'où $t - t' \in \text{GF}(p)$, et

$$\{t + \alpha_i : \alpha_i \in \text{GF}(p)\} = \{t' + \alpha_i : \alpha_i \in \text{GF}(p)\}.$$

Donc pour chaque $\alpha_i \in \text{GF}(p)$, il existe un unique $\alpha_{\pi(i)} \in \text{GF}(p)$ vérifiant

$$g^{b_{\pi(i)}} = t' + \alpha_i.$$

En utilisant t' , g , π et d , l'attaquant peut faire le même déchiffrement que le destinataire légitime.

4.3 Attaque due à S. Vaudenay (g connu)

Supposons que $\sigma(0) = i$ et $\sigma(1) = j$, par la symétrie dans la clé privée, nous savons qu'un choix arbitraire de (i, j) convient. Calculons ensuite :

$$g^{c_0 - c_1} = \frac{t + \alpha_i}{t + \alpha_j},$$

et nous pouvons alors déterminer t .

4.4 Attaque due à B. Chor et R. Rivest (t et d connus)

Choisissons un générateur arbitraire g' de $\text{GF}(p^h)^\times$ et calculons

$$b_i := \log_{g'}(t + \alpha_i).$$

En tant qu'ensembles, nous avons

$$\{c_0 - d, \dots, c_{p-1} - d\} = L\{b_0, \dots, b_{p-1}\},$$

où l'égalité est modulo $p^h - 1$. Les nombres L et $p^h - 1$ sont premiers entre eux et L satisfait à l'égalité $g = g'^L$. Une fois que nous avons recouvert L , nous avons fini, car $g = g'^L$ et nous pouvons reconstruire σ , obtenant ainsi toutes les pièces de la clé privée.

Si un des b_i , disons b_0 , est premier avec $p^h - 1$, alors L est l'un des

$$(d - c_j)b_0^{-1} \pmod{p^h - 1},$$

pour un $0 \leq j \leq p-1$. Sinon, l'attaquant peut calculer L modulo chaque facteur puissance de nombre premier et les combiner en utilisant le théorème des restes chinois.

4.5 Attaque due à O. Goldreich (t connu) [version modifiée]

Choisissons un générateur arbitraire g' de $\text{GF}(p^h)^\times$ et calculons

$$b_i := \log_{g'}(t + \alpha_i).$$

Notons $L := \log_{g'}(g)$, de sorte que $g = g'^L$. Remarquons que, si $x \in \text{GF}(p^h)$:

$$L \log_g(x) = \log_{g'}(x).$$

Considérons les ensembles suivants :

$$\begin{aligned} \{b_0 - b_0, b_0 - b_1, \dots, b_0 - b_{p-1}\} &= \left\{ \log_{g'} \left(\frac{t + \alpha_0}{t + \alpha_0} \right), \dots, \log_{g'} \left(\frac{t + \alpha_0}{t + \alpha_{p-1}} \right) \right\}. \\ L\{c_0 - c_0, c_1 - c_0, \dots, c_{p-1} - c_0\} &= L \left\{ \log_g \left(\frac{t + \alpha_{\sigma(0)}}{t + \alpha_{\sigma(0)}} \right), \dots, \log_g \left(\frac{t + \alpha_{\sigma(p-1)}}{t + \alpha_{\sigma(0)}} \right) \right\} \\ &= \left\{ \log_{g'} \left(\frac{t + \alpha_{\sigma(0)}}{t + \alpha_{\sigma(0)}} \right), \dots, \log_{g'} \left(\frac{t + \alpha_{\sigma(p-1)}}{t + \alpha_{\sigma(0)}} \right) \right\}. \end{aligned}$$

Notons i l'antécédent et j l'image de 0 par la permutation σ , i.e. $\sigma(i) = 0$ et $\sigma(0) = j$. Alors :

$$b_0 - b_j = L(c_i - c_0).$$

Il faut ensuite se servir de l'égalité d'ensembles :

$$\{b_0 - b_j, b_1 - b_j, \dots, b_{p-1} - b_j\} = L\{c_0 - c_0, c_1 - c_0, \dots, c_{p-1} - c_0\},$$

pour vérifier que nous avons trouvé le bon L .

Algorithme 3 Algorithme implémentant l'attaque de Goldreich

Entrée : $\text{GF}(p^h)$, (c_0, \dots, c_{p-1}) et t **Sortie :** g et d ou "échec"

```
1: choisir  $g'$  un élément primitif arbitraire de  $\text{GF}(p^h)$ 
2: calculer les  $b_i := \log_{g'}(t + \alpha_i)$ 
3:  $B \leftarrow [b_0 - b_i \text{ pour } i \text{ allant de } 0 \text{ à } p-1]$ 
4:  $C \leftarrow [c_i - c_0 \text{ pour } i \text{ allant de } 0 \text{ à } p-1]$ 
5: pour  $i$  allant de 0 à  $p-1$  faire
6:   pour  $j$  allant de 0 à  $p-1$  faire
7:     si  $C[i] \neq 0$  et  $B[j] \neq 0$  alors
8:        $d \leftarrow \text{pgcd}(C[i], p^h - 1)$ 
9:       si  $d$  divise  $B[j]$  alors
10:         $b \leftarrow B[j]/d$ 
11:         $c \leftarrow C[i]/d$ 
12:         $r \leftarrow (p^h - 1)/d$ 
13:         $L \leftarrow c^{-1}b \pmod{r}$ 
14:        pour  $k$  allant de 0 à  $d-1$  faire
15:           $\mathcal{C} \leftarrow \{L \cdot C[n] \pmod{r} \text{ pour } n \text{ allant de } 0 \text{ à } p-1\}$ 
16:           $\mathcal{B} \leftarrow \{B[n] - B[j] \pmod{r} \text{ pour } n \text{ allant de } 0 \text{ à } p-1\}$ 
17:          si  $\mathcal{C} = \mathcal{B}$  alors  $\{\text{le } L \text{ est bon et } \sigma(i) = 0 \text{ et } \sigma(0) = j\}$ 
18:             $g \leftarrow g'^L$ 
19:             $d \leftarrow c_0 - \log_g(t + \alpha_j) \pmod{p^h - 2}$ 
20:            retourner  $g$  et  $d$ 
21:          fin si
22:           $L \leftarrow L + r$ 
23:        fin pour
24:      fin si
25:    fin si
26:  fin pour
27: fin pour
28: retourner "échec"
```

4.6 Attaque due à A. Odlyzko (σ connue)

Il faut trouver une combinaison linéaire de la forme

$$\sum_{i=1}^{p-1} x_i (c_{\sigma^{-1}(i)} - c_{\sigma^{-1}(0)}) = 0,$$

avec des entiers x_i relativement petits. Cela peut être fait via l'algorithme LLL. Nous pouvons nous attendre à $|x_i| \leq B$, où $B \sim p^{h/(p-1)}$.

Soit α un élément primitif de $\text{GF}(p^h)$:

$$\begin{aligned} & \sum_{i=1}^{p-1} x_i (c_{\sigma^{-1}(i)} - c_{\sigma^{-1}(0)}) = 0 \\ \Leftrightarrow & \prod_{i=1}^{p-1} \alpha^{x_i (c_{\sigma^{-1}(i)} - c_{\sigma^{-1}(0)})} = 1 \end{aligned}$$

$$\begin{aligned}
&\Leftrightarrow \prod_{i=1}^{p-1} \alpha^{x_i (\log_g(t-\alpha_i) - \log_g(t-\alpha_0))} = 1 \\
&\Leftrightarrow \prod_{i=1}^{p-1} \alpha^{x_i \log_g(t-\alpha_i)} = \prod_{i=1}^{p-1} \alpha^{x_i \log_g(t-\alpha_0)} \\
&\Leftrightarrow \prod_{i=1}^{p-1} (t - \alpha_i)^{x_i} = \prod_{i=1}^{p-1} (t - \alpha_0)^{x_i}
\end{aligned}$$

Ainsi t est racine du polynôme

$$\prod_{i=1}^{p-1} (y - \alpha_i)^{x_i} - \prod_{i=1}^{p-1} (y - \alpha_0)^{x_i} \in \text{GF}(p)[y],$$

dont un algorithme comme celui de Rabin, permet de trouver les racines.

5 L'attaque de Brickell

Dans le papier de B. Chor et R. Rivest [2, Section VII, A.5], Ernest Brickell propose une attaque où rien de la clé privée n'est connu.

Le but de cette attaque est de trouver une équation polynomiale de petit degré satisfaite par g . En utilisant un réseau bien choisi, il est possible de trouver des entiers x_i , la plus part nuls, tels que les deux équations

$$\begin{aligned}
\sum_{i=0}^{p-1} x_i c_i &\equiv 0 \pmod{p^h - 1}, \\
\sum_{i=0}^{p-1} x_i &\equiv 0 \pmod{p^h - 1},
\end{aligned}$$

soient satisfaites. Notons $b_i := \log_g(t + \alpha_{\sigma(i)})$. La deuxième équation garantie :

$$\begin{aligned}
g^{\sum_{i=0}^{p-1} x_i c_i} &= g^{\sum_{i=0}^{p-1} x_i (d+b_i)} \\
&= g^{\sum_{i=0}^{p-1} x_i b_i} g^{d \sum_{i=0}^{p-1} x_i} \\
&= g^{\sum_{i=0}^{p-1} x_i b_i}
\end{aligned}$$

et donc par la première égalité :

$$g^{\sum_{i=0}^{p-1} x_i b_i} = \prod_{i=0}^{p-1} (t + \alpha_{\sigma(i)})^{x_i} = 1.$$

Définissons, pour tout $0 \leq i \leq p-1$, les entiers x_i^+ et x_i^- par :

$$x_i^+ := \begin{cases} x_i & \text{si } x_i \geq 0, \\ 0 & \text{sinon.} \end{cases}$$

$$x_i^- := \begin{cases} -x_i & \text{si } x_i \leq 0, \\ 0 & \text{sinon.} \end{cases}$$

Pour π une permutation de l'ensemble $\{0, \dots, p-1\}$, soit $r_\pi(y) \in \text{GF}(p)[y]$ le polynôme suivant :

$$r_\pi(y) := \prod_{i=0}^{p-1} (y + \alpha_{\pi(i)})^{x_i^+} - \prod_{i=0}^{p-1} (y + \alpha_{\pi(i)})^{x_i^-}.$$

Notons ℓ le nombre de x_i non-nuls. Chaque application $\{i : x_i \neq 0\} \rightarrow \text{GF}(p)$ bijective donne un polynôme $r_\pi(y)$ différent. Seul le “bon” polynôme aura t comme racine. Donc en moyenne il faut essayer $(1/2)(p!/(p-\ell)!)$ applications pour obtenir le bon polynôme. Pour chaque telle application, il faut trouver dans $\text{GF}(p^h)$ les racines de $r_\pi(y)$, puis nous les testons grâce à l'attaque de O. Goldreich exposée en sous-section 4.5.

Proposition 5.1. *Pour effectuer l'attaque de Ernest Brickell, il existe un algorithme de complexité $\mathcal{O}(p^{2\sqrt{h}} h^2 \log_2(p))$ opérations arithmétiques dans $\text{GF}(p)$.*

6 La cryptanalyse de Vaudenay

Nous présentons ici l'attaque de Serge Vaudenay [10]. Elle s'appuie sur la forte friabilité de h , i.e. sur l'existence de nombreux sous-corps de $\text{GF}(p^h)$.

Commençons par donner quelques outils pour l'explication de l'attaque.

Proposition 6.1. *Pour tout facteur r de h , il existe un générateur g_{p^r} du groupe multiplicatif du sous-corps $\text{GF}(p^r)$ de $\text{GF}(p^h)$ et un polynôme $Q(x) \in \text{GF}(p^r)[x]$ de degré h/r et tel que $-t$ en est une racine, et pour tout $0 \leq i \leq p-1$, nous avons :*

$$Q(\alpha_{\sigma(i)}) = (g_{p^r})^{c_i}.$$

Démonstration. Soit

$$Q(x) = (g_{p^r})^d \prod_{i=0}^{h/r-1} (x + t^{p^{ri}}), \quad \text{où } g_{p^r} := \prod_{i=0}^{h/r-1} g^{p^{ri}},$$

g_{p^r} peut être considéré comme la norme de g considéré dans l'extension de corps $\text{GF}(p^r) \subseteq \text{GF}(p^h)$, ainsi $(g_{p^r})^{p^r} = g_{p^r}$ et g_{p^r} est générateur de $\text{GF}(p^r)^\times$ car g est primitif dans $\text{GF}(p^h)$. Nous remarquons que $Q(x^{p^r}) = Q(x)^{p^r}$, ainsi $Q(x) \in \text{GF}(p^r)[x]$, en effet :

$$\begin{aligned} Q(x)^{p^r} &= (g_{p^r})^{dp^r} \prod_{i=0}^{h/r-1} (x + t^{p^{ri}})^{p^r} \\ &= (g_{p^r})^d \prod_{i=0}^{h/r-1} (x^{p^r} + t^{p^{ri}p^r}) \\ &= (g_{p^r})^d \prod_{i=1}^{h/r} (x^{p^r} + t^{p^{ri}}) \\ &= Q(x^{p^r}), \end{aligned}$$

car $t^{p^{r(h/r)}} = t^{p^h} = t = t^{p^0}$. Cela montre aussi que $Q(-t) = 0$. Nous avons aussi que $Q(x)$ est de degré h/r . Et enfin, pour $0 \leq j \leq p-1$, calculons $(g_{p^r})^{c_i}$:

$$\begin{aligned}
(g_{p^r})^{c_j} &= \left(\prod_{i=0}^{h/r-1} g^{p^{ri}} \right)^{c_j} \\
&= \prod_{i=0}^{h/r-1} (g^{c_j})^{p^{ri}} \\
&= \prod_{i=0}^{h/r-1} g^{dp^{ri}} (\alpha_{\sigma(j)} + t)^{p^{ri}} \\
&= (g_{p^r})^d \prod_{i=0}^{h/r-1} (\alpha_{\sigma(j)} + t^{p^{ri}}) \\
&= Q(\alpha_{\sigma(j)})
\end{aligned}$$

Cela achève la démonstration de la proposition. \square

Comme h/r est assez petit, il est peu probable qu'il existe d'autres solutions (g_{p^r}, Q) , et g_{p^r} est donc essentiellement unique.

Notation : Pour r divisant h , notons

$$g_{p^r} := g^{(p^h-1)/(p^r-1)}.$$

6.1 Attaque sachant $g^{(p^h-1)/(p^r-1)}$ et σ

Supposons connus $g_{p^r} = g^{(p^h-1)/(p^r-1)}$, pour r divisant h (comme dans la proposition 6.1), ainsi que la permutation σ . Nous pouvons alors interpoler le polynôme $Q(x)$ de la proposition 6.1, avec $h/r + 1$ pairs $(\alpha_{\sigma(i)}, (g_{p^r})^{c_i})_i$. Cela donne un polynôme de degré h/r dont les racines sont les conjugués de $-t$. Par la symétrie dans la clé privée, nous pouvons sélectionner n'importe quelle racine de $Q(x)$. Nous résolvons l'attaque en calculant g et d grâce à l'attaque de O. Goldreich exposée en sous-section 4.5.

6.2 Attaque sachant $g^{(p^h-1)/(p^r-1)}$

Supposons connu l'élément g_{p^r} , la norme de l'élément primitif g considéré dans l'extension $\text{GF}(p^r) \subseteq \text{GF}(p^h)$, comme dans la proposition 6.1.

Soient $(i_0, \dots, i_{h/r})$ une famille de $h/r + 1$ indices distincts entre 0 et $p-1$. Par la proposition 6.1, nous pouvons interpoler le polynôme $Q(x)$ en les $\alpha_{\sigma(i_j)}$:

$$Q(x) = \sum_{j=0}^{h/r} Q(\alpha_{\sigma(i_j)}) \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{x - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_j)} - \alpha_{\sigma(i_k)}}$$

Algorithme 4 Algorithme implémentant la première attaque de Vaudenay

Entrée : $\text{GF}(p^h)$, (c_0, \dots, c_{p-1}) , σ , r divisant h et $g^{(p^h-1)/(p^r-1)}$

Sortie : $t, t^{p^r}, t^{p^{2r}}, \dots, t^{p^{h-r}}$

- 1: $\rho \leftarrow g^{(p^h-1)/(p^r-1)}$
- 2: $n \leftarrow h/r$
- 3: $Q(x) \in \text{GF}(p^r)[x]$: initialisé $Q(x) \leftarrow 0$
- 4: **pour** i allant de 0 à n **faire**
- 5: calculer l'interpolateur de Lagrange :

$$L(x) \leftarrow \prod_{\substack{0 \leq k \leq n \\ k \neq i}} \frac{x - \alpha_{\sigma(k)}}{\alpha_{\sigma(i)} - \alpha_{\sigma(k)}}$$

- 6: $Q(x) \leftarrow Q(x) + \rho^{c_i} L(x)$
 - 7: **fin pour**
 - 8: **retourner** $[-t \in \text{GF}(p^h)$ pour t racine du polynôme $Q(x)$]
-

$$= \sum_{j=0}^{h/r} (g_{p^r})^{c_{i_j}} \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{x - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_j)} - \alpha_{\sigma(i_k)}},$$

ce qui mène aux égalités : quel que soit i tel que $0 \leq i \leq p-1$,

$$(\dagger) \quad (g_{p^r})^{c_i} = \sum_{j=0}^{h/r} (g_{p^r})^{c_{i_j}} \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_j)} - \alpha_{\sigma(i_k)}}.$$

En fait, nous pouvons même écrire : quel que soit i tel que $0 \leq i \leq p-1$,

$$(\ddagger) \quad (g_{p^r})^{c_i} - (g_{p^r})^{c_{i_0}} = \sum_{j=1}^{h/r} ((g_{p^r})^{c_{i_j}} - (g_{p^r})^{c_{i_0}}) \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_j)} - \alpha_{\sigma(i_k)}}.$$

En effet, soit i parmi $i_0, \dots, i_{h/r}$:

$$(g_{p^r})^{c_i} - (g_{p^r})^{c_{i_0}} = \sum_{j=1}^{h/r} ((g_{p^r})^{c_{i_j}} - (g_{p^r})^{c_{i_0}}) \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_j)} - \alpha_{\sigma(i_k)}},$$

car

$$\sum_{j=0}^{h/r} \left(\prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_j)} - \alpha_{\sigma(i_k)}} \right) = 1.$$

Ainsi, nous avons l'égalité de polynôme :

$$Q(x) - (g_{p^r})^{c_{i_0}} = \sum_{j=1}^{h/r} ((g_{p^r})^{c_{i_j}} - (g_{p^r})^{c_{i_0}}) \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{x - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_j)} - \alpha_{\sigma(i_k)}},$$

car les deux polynômes sont de degré h/r et sont égaux sur un ensemble de $h/r + 1$ éléments : d'où l'égalité lorsque le polynôme est évalué en les $\alpha_{\sigma(i)}$, où $0 \leq i \leq p-1$.

À cause de la symétrie de la clé privée, indiquée en sous-section 4.1, nous pouvons choisir arbitrairement $\sigma(i_1)$ et $\sigma(i_2)$. Un algorithme naïf pour trouver la permutation σ est de chercher exhaustivement les valeurs $\sigma(i_j)$ pour $j = 0$ et $3 \leq j \leq h/r$, jusqu'à ce que (\dagger) donne une permutation consistante.

Algorithme 5 Algorithme pour trouver π sachant g_{p^r}

Entrée : $\text{GF}(p^h)$, (c_0, \dots, c_{p-1}) , r divisant h et g_{p^r}

Sortie : une permutation π d'une clé privée équivalente

```

1: choisir des  $i_0, \dots, i_{h/r}$  deux à deux distincts dans  $\{0, \dots, p-1\}$ 
2: choisir arbitrairement  $\pi(i_1)$  et  $\pi(i_2)$  distincts dans  $\{0, \dots, p-1\}$ 
3: pour tout  $\pi(i_0), \pi(i_3), \dots, \pi(i_{h/r})$  distincts deux à deux faire
4:    $S \leftarrow [\pi(i_0), \pi(i_3), \dots, \pi(i_{h/r})]$ 
5:   pour tout  $\ell \notin S$  faire
6:     calculer le membre de droite de  $(\dagger)$  avec  $\alpha_\ell$  au lieu de  $\alpha_{\pi(i)}$  :


$$\text{res} \leftarrow \sum_{j=0}^{h/r} (g_{p^r})^{c_{i_j}} \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{\alpha_\ell - \alpha_{\pi(i_k)}}{\alpha_{\pi(i_j)} - \alpha_{\pi(i_k)}}$$


7:   si il existe  $i$  tel que  $\text{res} = (g_{p^r})^{c_i}$  et  $\pi(i)$  n'est pas définie alors
8:      $\pi(i) \leftarrow \ell$ 
9:      $S \leftarrow S \cup [\pi(i)]$ 
10:  sinon
11:    continuer la boucle ligne 3
12:  fin si
13: fin pour
14: retourner  $\pi$ 
15: fin pour

```

La complexité de cet algorithme est grossièrement $\mathcal{O}(rp^{h/r})$ opérations dans $\text{GF}(p)$: la boucle à la ligne 5 fait en moyenne $\mathcal{O}(pr/h)$ itérations, chacune avec une complexité $\mathcal{O}(h)$, et il nous avons besoin de $\mathcal{O}(p^{h/r} - 1)$ itérations de cette boucle.

La permutation π donnée par l'algorithme 5 est telle qu'il existe $u \in \text{GF}(p)^\times$ et $v \in \text{GF}(p)$ vérifiant pour tout $0 \leq i \leq p-1$:

$$\alpha_{\pi(i)} = u\alpha_{\sigma(i)} + v,$$

où la permutation σ est celle de la clé secrète d'origine, détenue par le destinataire légitime.

$$Q_\pi(x) = \sum_{j=0}^{h/r} (g_{p^r})^{c_{i_j}} \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{x - \alpha_{\pi(i_k)}}{\alpha_{\pi(i_j)} - \alpha_{\pi(i_k)}},$$

$$\begin{aligned}
&= \sum_{j=0}^{h/r} (g_{p^r})^{c_{i_j}} \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{x - u\alpha_{\sigma(i_k)} - v}{u\alpha_{\sigma(i_j)} - u\alpha_{\sigma(i_k)}}, \\
&= \sum_{j=0}^{h/r} (g_{p^r})^{c_{i_j}} \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{u^{-1}x - u^{-1}v - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_j)} - \alpha_{\sigma(i_k)}}, \\
&= Q_{\sigma}(u^{-1}(x - v)).
\end{aligned}$$

Ainsi, si $-\omega$ est racine du polynôme $Q_{\pi}(x)$, alors $-u^{-1}(\omega + v)$ est racine du polynôme $Q_{\sigma}(x)$, ainsi si on fait tourner l'algorithme 4 avec la permutation π , les valeurs retournées sont :

$$ut - v, \quad ut^{p^r} - v, \quad ut^{p^{2r}} - v, \quad \dots, \quad ut^{p^{h-r}} - v.$$

Il est alors plus simple de déterminer un g pour avoir une clé privée équivalente, que d'utiliser l'attaque de Goldreich, sous-section 4.5.

En effet, pour $0 \leq j < h/r$, à savoir on choisit n'importe quel élément retourné par l'algorithme 4, choisissons un générateur arbitraire g' de $\text{GF}(p^h)^{\times}$ et calculons

$$b_i := \log_{g'} \left(ut^{p^{j_r}} - v + \alpha_{\pi(i)} \right)$$

Notons $L := \log_{g'}(g^{p^{j_r}})$, de sorte que $g^{p^{j_r}} = g'^L$. Alors pour tout $0 \leq i \leq p-1$:

$$\begin{aligned}
b_i - b_0 &= \log_{g'} \left(\frac{ut^{p^{j_r}} - v + \alpha_{\pi(i)}}{ut^{p^{j_r}} - v + \alpha_{\pi(0)}} \right) \\
&= \log_{g'} \left(\frac{ut^{p^{j_r}} - v + u\alpha_{\sigma(i)} + v}{ut^{p^{j_r}} - v + u\alpha_{\sigma(0)} + v} \right) \\
&= \log_{g'} \left(\frac{ut^{p^{j_r}} + u\alpha_{\sigma(i)}}{ut^{p^{j_r}} + u\alpha_{\sigma(0)}} \right) \\
&= L \log_{g^{p^{j_r}}} \left(\frac{t^{p^{j_r}} + \alpha_{\sigma(i)}}{t^{p^{j_r}} + \alpha_{\sigma(0)}} \right) \\
&= L(c_i - c_0)
\end{aligned}$$

Bien sûr, si en entrée de l'algorithme 6, le " t " donné est $ut^{p^{j_r}} - v$, il retourne le conjugué $g^{p^{j_r}}$ du g d'origine et l'entier $d - \log_{g^{p^{j_r}}}(u)$.

Quand r est suffisamment grand, il existe un meilleur algorithme. En fait, si $r \geq h/r$, la famille $((g_{p^r})^{c_{i_j}} - (g_{p^r})^{c_{i_0}})_{1 \leq j \leq h/r}$ est libre (voir plus loin). Cela signifie que les coefficients dans (\dagger) sont les seuls coefficients dans $\text{GF}(p)$ de l'écriture de $(g_{p^r})^{c_i} - (g_{p^r})^{c_{i_0}}$, pour $0 \leq i \leq p-1$, comme combinaison linéaire des vecteurs :

$$(g_{p^r})^{c_{i_0}} - (g_{p^r})^{c_{i_0}}, \quad (g_{p^r})^{c_{i_1}} - (g_{p^r})^{c_{i_0}}, \quad \dots, \quad (g_{p^r})^{c_{i_{h/r}}} - (g_{p^r})^{c_{i_0}}.$$

Algorithme 6 Algorithme implémentant l'attaque de Goldreich simplifiée

Entrée : $\text{GF}(p^h)$, (c_0, \dots, c_{p-1}) , t et π

Sortie : g et d ou "échec"

```

1: choisir  $g'$  un élément primitif arbitraire de  $\text{GF}(p^h)$ 
2: calculer les  $b_i := \log_{g'}(t + \alpha_{\pi(i)})$ 
3:  $B \leftarrow [b_i - b_0 \text{ pour } i \text{ allant de } 0 \text{ à } p-1]$ 
4:  $C \leftarrow [c_i - c_0 \text{ pour } i \text{ allant de } 0 \text{ à } p-1]$ 
5: pour  $i$  allant de 0 à  $p-1$  faire
6:   si  $\text{pgcd}(C[i], p^h - 1) = 1$  alors
7:      $L \leftarrow C[i]^{-1} B[i] \pmod{p^h - 1}$ 
8:      $\mathcal{C} \leftarrow \{L \cdot C[n] \pmod{p^h - 1} \text{ pour } n \text{ allant de } 0 \text{ à } p-1\}$ 
9:      $\mathcal{B} \leftarrow \{B[n] \pmod{p^h - 1} \text{ pour } n \text{ allant de } 0 \text{ à } p-1\}$ 
10:    si  $\mathcal{C} = \mathcal{B}$  alors
11:       $g \leftarrow g'^L$ 
12:       $d \leftarrow c_0 - \log_g(t + \alpha_{\pi(0)}) \pmod{p^h - 2}$ 
13:      retourner  $g$  et  $d$ 
14:    fin si
15:  fin si
16: fin pour
17: retourner "échec"
  
```

Notons a_j^i le coefficient de $(g_{p^r})^{c_{i_j}} - (g_{p^r})^{c_{i_0}}$ pour $(g_{p^r})^{c_i} - (g_{p^r})^{c_{i_0}}$. Par (\dagger) , nous avons :

$$\begin{aligned}
 \frac{a_2^i}{a_1^i} &= \left(\prod_{\substack{0 \leq k \leq h/r \\ k \neq 2}} \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_2)} - \alpha_{\sigma(i_k)}} \right) \left(\prod_{\substack{0 \leq k' \leq h/r \\ k' \neq 1}} \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_{k'})}}{\alpha_{\sigma(i_1)} - \alpha_{\sigma(i_{k'})}} \right)^{-1} \\
 &= \left(\prod_{\substack{0 \leq k \leq h/r \\ k \neq 2}} \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_2)} - \alpha_{\sigma(i_k)}} \right) \left(\prod_{\substack{0 \leq k' \leq h/r \\ k' \neq 1}} \frac{\alpha_{\sigma(i_1)} - \alpha_{\sigma(i_{k'})}}{\alpha_{\sigma(i)} - \alpha_{\sigma(i_{k'})}} \right) \\
 &= \left(\prod_{\substack{0 \leq k, k' \leq h/r \\ k \neq 2, k' \neq 1}} \frac{\alpha_{\sigma(i_1)} - \alpha_{\sigma(i_{k'})}}{\alpha_{\sigma(i_2)} - \alpha_{\sigma(i_k)}} \right) \left(\prod_{\substack{0 \leq k, k' \leq h/r \\ k \neq 2, k' \neq 1}} \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i)} - \alpha_{\sigma(i_{k'})}} \right) \\
 &= \left(\prod_{\substack{0 \leq k, k' \leq h/r \\ k \neq 2, k' \neq 1}} \frac{\alpha_{\sigma(i_1)} - \alpha_{\sigma(i_{k'})}}{\alpha_{\sigma(i_2)} - \alpha_{\sigma(i_k)}} \right) \left(\frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_1)}}{\alpha_{\sigma(i)} - \alpha_{\sigma(i_2)}} \right)
 \end{aligned}$$

Ainsi, il existe u dans $\text{GF}(p)$, indépendant de i , tel que

$$(\star) \quad \frac{a_2^i}{a_1^i} = u \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_1)}}{\alpha_{\sigma(i)} - \alpha_{\sigma(i_2)}}.$$

En passant toutes les valeurs de u en revue, nous pouvons obtenir $\sigma(i)$ de l'équation (\star) . Cette remarque donne naissance à l'algorithme suivant :

Algorithme 7 Algorithme pour trouver π sachant g_{p^r} lorsque $r \geq \sqrt{h}$

Entrée : $\text{GF}(p^h)$, (c_0, \dots, c_{p-1}) , r divisant h tel que $r \geq \sqrt{h}$ et g_{p^r}

Sortie : une permutation π d'une clé privée équivalente

- 1: choisir des $i_0, \dots, i_{h/r}$ deux à deux distincts dans $\{0, \dots, p-1\}$
 - 2: pré-calculer la matrice de changement de bases de la base "classique" vers la "base" $((g_{p^r})^{c_{i_j}} - (g_{p^r})^{c_{i_0}})_{0 \leq j \leq h/r}$
 - 3: choisir arbitrairement $\pi(i_1)$ et $\pi(i_2)$ distincts dans $\{0, \dots, p-1\}$
 - 4: **pour tout** u dans $\text{GF}(p)$ **faire**
 - 5: **pour tout** i tel que $0 \leq i \leq p-1$ et $i \neq i_0, i_1, i_2$ **faire**
 - 6: écrire $(g_{p^r})^{c_i} - (g_{p^r})^{c_{i_0}}$ dans la "base" $((g_{p^r})^{c_{i_j}} - (g_{p^r})^{c_{i_0}})_{1 \leq j \leq h/r}$
 - 7: récupérer les coefficients a_1^i et a_2^i
 - 8: trouver la valeur de $\pi(i)$ grâce à (\star)
 - 9: **fin pour**
 - 10: **si** π est constante **alors**
 - 11: compléter $\pi(i_0)$
 - 12: **retourner** π
 - 13: **fin si**
 - 14: **fin pour**
-

$$\frac{a_2^i}{a_1^i} = u \frac{\alpha_{\pi(i)} - \alpha_{\pi(i_1)}}{\alpha_{\pi(i)} - \alpha_{\pi(i_2)}} \Leftrightarrow \alpha_{\pi(i)} = \frac{a_2^i \alpha_{\pi(i_2)} - u a_1^i \alpha_{\pi(i_1)}}{a_2^i - u a_1^i}.$$

6.3 Test pour trouver $g^{(p^h-1)/(p^r-1)}$

Les équations : quel que soit i tel que $0 \leq i \leq p-1$,

$$(\dagger) \quad (g_{p^r})^{c_i} - (g_{p^r})^{c_{i_0}} = \sum_{j=1}^{h/r} ((g_{p^r})^{c_{i_j}} - (g_{p^r})^{c_{i_0}}) \prod_{\substack{0 \leq k \leq h/r \\ k \neq j}} \frac{\alpha_{\sigma(i)} - \alpha_{\sigma(i_k)}}{\alpha_{\sigma(i_j)} - \alpha_{\sigma(i_k)}},$$

signifient que tous les $(g_{p^r})^{c_i}$ sont en fait dans un même sous-espace affine de dimension h/r du $\text{GF}(p)$ -espace vectoriel $\text{GF}(p^r)$.

Donc si on suppose que $h/r + 1 \leq r$, à savoir $r \geq \sqrt{h+1/4} + 1/2$, on peut donner un test facile pour g_{p^r} .

Proposition 6.2. *S'il existe un facteur r de h de sorte que $r \geq \sqrt{h+1/4} + 1/2$, et si*

$$g_{p^r} := g^{(p^h-1)/(p^r-1)} = g^{1+p^r+p^{2r}+\dots+p^{h-r}},$$

alors tous les $g_{p^r}^{c_i}$ sont sur le même sous-espace affine de dimension h/r de $\text{GF}(p^r)$ lorsque celui-ci est considéré comme un espace affine de dimension r sur $\text{GF}(p)$.

Démonstration. To do!! □

L'existence d'un tel r peut être considérée comme un mauvais prérequis, cependant vu que les paramètres du cryptosystème de Chor-Rivest doivent être choisis de sorte que le problème du logarithme discret soit facile, nous savons

déjà que h à plusieurs facteurs, et il est donc fort probable que cette hypothèse sur r soit satisfaite. En fait, les h sans tels facteurs sont les nombres premiers et les carrés de nombres premiers. Le vrai problème est que r ne doit pas être trop grand.

Nous pouvons donc écrire un algorithme qui vérifie si un candidat pour g_{p^r} est bon : l'algorithme vérifie simplement si les $g_{p^r}^{c_i}$ sont affinement liés. Cet algorithme a pour complexité moyenne $\mathcal{O}(h^3/r)$ opérations dans $\text{GF}(p)$. Comme il y a $\phi(p^r - 1)/r$ candidats, nous pouvons chercher exhaustivement un g_{p^r} avec une complexité de $\mathcal{O}(h^3 p^r / r^2)$.

Algorithme 8 Algorithme pour trouver g_{p^r} lorsque $r \geq \sqrt{h+1/4} + 1/2$

Entrée : $\text{GF}(p^h)$, (c_0, \dots, c_{p-1}) et r divisant h tel que $r \geq \sqrt{h+1/4} + 1/2$

Sortie : un élément g_{p^r}

```

1: choisir des  $i_0, \dots, i_{h/r}$  deux à deux distincts dans  $\{0, \dots, p-1\}$ 
2: pour tout  $\zeta \in \text{GF}(p^h)$  générateur de  $\text{GF}(p^r)^\times$  faire
3:   calculer l'équation du sous-espace affine  $V$  engendré par  $(\zeta^{c_{i_0}}, \dots, \zeta^{c_{i_{h/r}}})$ 
4:   pour tout  $i$  différent de  $i_0, \dots, i_{h/r}$  faire
5:     si  $\zeta^{c_i}$  ne satisfait pas à l'équation du sous-espace affine  $V$  alors
6:       continuer la boucle ligne 2
7:   fin si
8: fin pour
9: retourner  $\zeta$ 
10: fin pour

```

6.4 Utilisation des c_i

Nous allons améliorer l'attaque précédente en utilisant la connaissance de tous les c_i . Nous allons nous servir du fait suivant :

Proposition 6.3. *Soit $Q(x)$ un polynôme de $\text{GF}(p^r)[x]$ de degré d et soit e un entier tel que $1 \leq e < (p-1)/d$. Nous avons*

$$\sum_{a \in \text{GF}(p)} Q(a)^e = 0.$$

Le lemme qui suit va permettre de démontrer cette proposition.

Lemme 6.4. *Soit k un entier tel que $1 \leq k < p-1$. Alors*

$$\sum_{a \in \text{GF}(p)} a^k = 0.$$

Ainsi, si $P(x) \in \text{GF}(p^r)[x]$, pour $r \geq 1$, est de degré inférieur à $p-1$, alors :

$$\sum_{a \in \text{GF}(p)} P(a) = 0.$$

Démonstration du lemme. Tout d'abord remarquons que $0 \in \text{GF}(p)$ ne contribue pas à la somme, et ensuite, notons g un élément primitif du corps $\text{GF}(p)$.

Alors, pour $1 \leq k < p-1$:

$$\sum_{a \in \text{GF}(p)} a^k = \sum_{a \in \text{GF}(p)^\times} a^k = \sum_{j=0}^{p-2} g^{jk} = \frac{g^{k(p-1)} - 1}{g^k - 1} = \frac{1^k - 1}{g^k - 1} = 0.$$

Soit $P(x) \in \text{GF}(p^r)[x]$ de degré inférieur à $p-1$:

$$P(x) = b_{p-2}x^{p-2} + \cdots + b_1x + b_0, \quad \text{où } b_i \in \text{GF}(p^r).$$

Ainsi, on obtient bien :

$$\begin{aligned} \sum_{a \in \text{GF}(p)} P(a) &= \sum_{a \in \text{GF}(p)} (b_{p-2}a^{p-2} + \cdots + b_1a + b_0) \\ &= \sum_{k=1}^{p-2} \left(b_k \sum_{a \in \text{GF}(p)} a^k \right) + pb_0 \\ &= 0. \end{aligned}$$

□

Démonstration de la proposition. Comme $Q(x) \in \text{GF}(p^r)[x]$ est de degré d et que $1 \leq e < (p-1)/d$, le polynôme $Q(x)^e$ est de degré inférieur à $p-1$. Ainsi le lemme précédent permet de conclure :

$$\sum_{a \in \text{GF}(p)} Q(a)^e = 0.$$

□

Proposition 6.5. *Pour tout $1 \leq e < (p-1)r/h$, nous avons :*

$$\sum_{i=0}^{p-1} (g_{p^r})^{ec_i} = 0.$$

Démonstration. Considérons le polynôme $Q(x) \in \text{GF}(p^r)[x]$ de la proposition 6.1 : le polynôme $Q(x)$ est de degré h/r , à coefficients dans $\text{GF}(p^r)$ et tel que, pour tout $0 \leq i \leq p-1$:

$$Q(\alpha_{\sigma(i)}) = (g_{p^r})^{c_i}.$$

Ainsi par la proposition précédente, pour tout $1 \leq e < (p-1)r/h$:

$$\sum_{a \in \text{GF}(p)} Q(a)^e = \sum_{i=0}^{p-1} Q(\alpha_{\sigma(i)})^e = 0,$$

d'où l'égalité souhaitée :

$$\sum_{i=0}^{p-1} (g_{p^r})^{ec_i} = 0.$$

□

Cela donne une façon plus simple de sélectionner tous les candidats pour g_{p^r} – que la recherche exhaustive. Son principal avantage est qu’il fonctionne dans n’importe quel sous-corps. Par exemple, on peut considérer $r = 1$ et trouver les seuls g_p tels que pour tout $1 \leq e < (p-1)/h$, nous avons :

$$\sum_{i=0}^{p-1} (g_p)^{ec_i} = 0.$$

La complexité moyenne pour vérifier un candidat est $\mathcal{O}(p)$ GF(p)-opérations : il est peu probable qu’un mauvais candidat ne sera pas détecté par le $e = 1$. Ainsi, nous pouvons recouvrer g_p en $\mathcal{O}(p^2)$ opérations binaires.

Malheureusement, le g_{p^r} ne peut pas être utilisé efficacement quand r est trop petit. Cependant, on peut toujours utiliser g_{p^r} des petits sous-corps pour le calculer dans des corps plus grands. Notre but est de calculer g_{p^r} avec r assez grand (tel que $r^2 \geq h$). Nous considérons le problème du calcul de g_{p^r} lorsque r_1, \dots, r_k sont les facteurs de r et avec la connaissance de $g_{p^{r_i}}$. Comme $g_{p^{r_i}} = (g_{p^r})^{1+p^{r_i}+p^{2r_i}+\dots+p^{r-r_i}}$, nous avons

$$(\dagger) \quad \log(g_{p^r}) = \frac{\log(g_{p^{r_i}})}{1 + p^{r_i} + p^{2r_i} + p^{3r_i} + \dots + p^{r-r_i}} \pmod{p^{r_i} - 1},$$

où la base des logarithmes est n’importe quel élément primitif γ fixé dans le corps GF(p^r). La connaissance de tous les $g_{p^{r_i}}$ donne donc la connaissance de $\log(g_{p^r})$ modulo

$$\ell := \text{ppcm} \{p^{r_1} - 1, p^{r_2} - 1, \dots, p^{r_k} - 1\}.$$

Nous avons donc besoin seulement de $(p^r - 1)/\ell$ essais pour recouvrer g_{p^r} .

Chaque boucle de la ligne 8 demande en moyenne $\mathcal{O}(pr^2)$ opérations dans GF(p).

Nous pouvons maintenant définir l’algorithme global de l’attaque.

Définition 6.6. Soit G un graphe orienté acyclique (*DAG for directed acyclic graph*) étiqueté enraciné dont la racine est étiquetée par un corps fini GF(p^r) et tel que pour chaque arête $u \rightarrow v$ de G , l’étiquette $l(u)$ de u est un sous-corps maximal (au sens de l’inclusion) de l’étiquette $L(v)$ de v et une extension de GF(p). On dit que G est un DAG p -factorisant de GF(p^r).

Définition 6.7. À une extension GF(p^r) et G son DAG p -factorisant, on associe :

$$C(G) := \sum_v \frac{\#L(v) - 1}{\text{ppcm}\{\#L(w) - 1 \mid v \leftarrow w\}},$$

où nous prenons pour convention : $\text{ppcm}(\emptyset) = 1$.

Nous pouvons définir un algorithme pour calculer g_{p^r} avec pour complexité $\mathcal{O}(pr^2 C(G))$, où G est le DAG p -factorisant de GF(p^r). Donc, nous pouvons casser le cryptosystème de Chor-Rivest avec comme paramètre h ni un nombre premier, ni un carré de nombre premier, avec une complexité de

$$\mathcal{O} \left(\min_{\substack{r \text{ divisant } h \\ r^2 \geq h}} \min_{\substack{G \text{ est un DAG } p\text{-factorisant} \\ \text{de GF}(p^r)}} pr^2 C(G) \right)$$

Algorithme 9 Algorithme pour récupérer g_{p^r} à partir des $g_{p^{r_i}}$

Entrée : $\text{GF}(p^h)$, (c_0, \dots, c_{p-1}) , r divisant h , $\{r_i\}_{1 \leq i \leq k}$ diviseurs de r et $g_{p^{r_i}}$

Sortie : l'ensemble des g_{p^r} possibles

- 1: choisir γ un élément primitif de $\text{GF}(p^r)$
- 2: **pour tout** i allant de 1 à k **faire**
- 3: résoudre l'équation (\dagger) :

$$x_i = \frac{\log(g_{p^{r_i}})}{1 + p^{r_i} + p^{2r_i} + p^{3r_i} + \dots + p^{r-r_i}} \pmod{p^{r_i} - 1}$$

- 4: **fin pour**
- 5: $\ell \leftarrow \text{ppcm}\{p^{r_1} - 1, p^{r_2} - 1, \dots, p^{r_k} - 1\}$
- 6: résoudre $x \equiv x_i \pmod{p^{r_i} - 1}$ où x est unique modulo ℓ
- 7: $\beta \leftarrow \gamma^x$
- 8: **pour** y allant de 0 à $(p^r - 1)/\ell - 1$ **faire**
- 9: **pour** e allant de 1 à $(p - 1)r/h - 1$ **faire**
- 10: calculer la somme :

$$\text{res} \leftarrow \sum_{i=0}^{p-1} \beta^{ec_i} \gamma^{ec_i \ell y}$$

- 11: **si** $\text{res} \neq 0$ **alors**
 - 12: continuer la boucle ligne 8
 - 13: **fin si**
 - 14: **fin pour**
 - 15: ajouter $\beta \gamma^{\ell y}$ à la liste des g_{p^r} possibles
 - 16: **fin pour**
-

7 Le calcul de logarithmes dans un groupe

Les résultats proposés ici sont essentiellement tous issus du manuel de Steven Galbraith [4, section 13 et 14, pp. 246 à 297].

Définition 7.1. Soit G un groupe noté multiplicativement. Le *problème du logarithme discret* est : étant donné $g \in G$ et $h \in \langle g \rangle$, trouver a , tel que $h = g^a$.

Nous notons alors $a = \log_g(h)$, et g est appelé la base du logarithme a .

Exemple 7.2. Considérons le problème du logarithme discret dans le groupe additif des entiers modulo p , où p est un nombre premier. C'est-à-dire qu'étant donné $g \neq 0$ et h dans $\mathbb{Z}/p\mathbb{Z}$, nous cherchons $a \in \mathbb{Z}$ tel que $ag = h$. Comme $\mathbb{Z}/p\mathbb{Z}$ est un corps, il suffit de prendre $a \equiv g^{-1}h \pmod{p}$. Nous pouvons déterminer g^{-1} grâce à une identité de Bézout. Ainsi, résoudre ce cas de logarithme discret se fait en temps polynomial.

7.1 Recherche exhaustive

Soient G un groupe, $g \in G$ et $h \in \langle g \rangle$. La façon la plus simple de résoudre ce problème du logarithme discret est de calculer à la suite les g^a pour $0 \leq a < r$, où r est l'ordre de g , et de comparer la valeur obtenue avec h . Cela s'effectue en au plus $r - 2$ opérations dans le groupe et r comparaisons.

Algorithme 10 Attaque général

Entrée : $\text{GF}(p^h)$ et (c_0, \dots, c_{p-1}) **Sortie :** une clé secrète équivalente

- 1: pour le plus petit facteur r de h tel que $r \geq \sqrt{h + 1/4} + 1/2$, trouver G un DAG p -factorisant minimisant $C(G)$
 - 2: **pour** u sommet de G , tel que pour tout $u \leftarrow u_i$, u_i a été visité, **faire**
 - 3: l'algorithme 9 avec $\text{GF}(p^r) = L(u)$ et $\text{GF}(p^{r_i}) = L(u_i)$
 - 4: **fin pour**
 - 5: appliquer l'algorithme 7 à g_{p^r}
-

Algorithme 11 Algorithme naïf : recherche exhaustive

Entrée : g et $h \in \langle g \rangle$ **Sortie :** a tel que $g^a = h$ et $1 \leq a \leq r$

- 1: $a \leftarrow 1$
 - 2: $t \leftarrow g$
 - 3: **tant que** $t \neq h$ **faire**
 - 4: $a \leftarrow a + 1$
 - 5: $t \leftarrow tg$
 - 6: **fin tant que**
 - 7: **retourner** a
-

7.2 Méthode de Pohlig et Hellman

Soient G un groupe, $g \in G$ et $h \in \langle g \rangle$. Supposons que g est d'ordre n . Écrivons

$$n = \prod_{i=1}^r p_i^{\alpha_i},$$

où les p_i sont premiers et les α_i sont tels que p_i ne divise pas $n/p_i^{\alpha_i}$. L'idée de la méthode de Stephen C. Pohlig et Martin E. Hellman [6] est de calculer a modulo les puissances de premier $p_i^{\alpha_i}$ et de recouvrer la solution en utilisant le théorème des restes chinois.

Pour cela, nous utilisons l'homomorphisme de groupes suivant, qui permet de réduire la recherche du logarithme discret aux sous-groupes d'ordre une puissance de nombre premier.

Lemme 7.3. *Soient G un groupe cyclique d'ordre n , g un générateur de G , i.e. $G = \langle g \rangle$, et q un diviseur de n . L'application $\varphi_q : x \mapsto x^{n/q}$ est un homomorphisme de groupes de G dans l'unique sous-groupe de G d'ordre q . Ainsi, si $h = g^a$, alors*

$$\varphi_q(h) = \varphi_q(g)^{a \bmod q}.$$

Démonstration. Soit H un sous-groupe de G d'ordre q . Montrons que nous avons l'égalité $H = \langle g^{n/q} \rangle$. Le groupe engendré par $g^{n/q}$ est d'ordre q car g est d'ordre n , i.e. pour tout $1 \leq k \leq q-1$, l'élément $(g^{n/q})^k$ est distinct de l'élément neutre. Considérons, pour α un entier, g^α dans H et distinct de l'élément neutre. Ainsi, $(g^\alpha)^q$ est l'élément neutre du groupe. Cela implique que l'ordre de g divise l'entier αq , ce qui est équivalent à n/q divise α . Donc, tout élément de H est une puissance de $g^{n/q}$. Nous concluons grâce à la cardinalité des sous-groupes : l'unique sous-groupe de G de cardinal q est $H = \langle g^{n/q} \rangle$.

Montrons que φ_q est un homomorphisme de G dans H . Tout élément de G à la puissance n/q est d'ordre divisant q . Ainsi, l'image de G par φ_q est incluse dans H . Et pour tous α et β entiers :

$$\varphi_q(g^\alpha g^\beta) = (g^\alpha g^\beta)^{n/q} = (g^\alpha)^{n/q} (g^\beta)^{n/q} = \varphi_q(g^\alpha) \varphi_q(g^\beta).$$

Donc φ_q est un homomorphisme du groupe G dans le groupe H .

Soit a un entier. Notons $a = qb + r$ la division euclidienne de a par q , en d'autres termes $r \equiv a \pmod{q}$. En sachant que φ_q est un homomorphisme et que $\varphi_q(g)$ est d'ordre q , nous pouvons écrire :

$$\varphi_q(g^a) = \varphi_q(g)^a = \varphi_q(g)^{qb+r} = (\varphi_q(g)^q)^b \varphi_q(g)^r = \varphi_q(g)^r = \varphi_q(g)^{a \bmod q}.$$

□

Dans les sous-groupes cycliques d'ordre une puissance de nombre premier, voici comment nous pouvons procéder : supposons que g_0 est d'ordre p^α et que $h_0 := g_0^a$, pour un entier. Nous pouvons écrire $a = a_0 + a_1p + \dots + a_{\alpha-1}p^{\alpha-1}$, où les a_i sont tels que $0 \leq a_i < p$. Soit $g_1 := g_0^{p^{\alpha-1}}$. Élever à la puissance $p^{\alpha-1}$ donne :

$$h_0^{p^{\alpha-1}} = (g_0^a)^{p^{\alpha-1}} = (g_0^{p^{\alpha-1}})^{a_0 + a_1p + \dots + a_{\alpha-1}p^{\alpha-1}} = \prod_{i=0}^{\alpha-1} g^{a_i p^{\alpha-1+i}} = g_1^{a_0}.$$

Nous pouvons ensuite déterminer a_0 , en cherchant de manière exhaustive ou avec une autre méthode, voir sous-section 7.3.

Pour avoir a_1 , définissons $h_1 := h_0 g_0^{-a_0}$ de sorte que

$$h_1 = g_0^{a_1 p + \dots + a_{\alpha-1} p^{\alpha-1}}.$$

Puis, en élevant à la puissance $p^{\alpha-2}$:

$$h_1^{p^{\alpha-2}} = g_1^{a_1}.$$

Pour obtenir a_2 , nous posons $h_2 := h_1 g_0^{-a_1 p}$, et ainsi de suite pour les autres valeurs. Cela donne la solution modulo p^α .

Une fois que a est connu pour tout p^α divisant n , nous pouvons reconstituer a à l'aide du théorème des restes chinois.

Exemple 7.4. Soit $p := 61$, $g := 7$ et $h := 25$. Mettons en œuvre l'algorithme Pohlig-Hellman pour trouver a tel que $g^a \equiv h \pmod{p}$. Notons que $\text{GF}(61)^\times$ est d'ordre $2^2 \cdot 3 \cdot 5$.

Trouvons d'abord a modulo 4. Comme $(p-1)/4 = 15$, nous posons :

$$g_0 := g^{15} \equiv 11 \pmod{61} \quad \text{et} \quad h_0 := h^{15} \equiv 1 \pmod{61}.$$

Pour obtenir de l'information modulo 2, nous calculons :

$$g_1 := g_0^2 \equiv -1 \pmod{61} \quad \text{et} \quad h_0^2 \equiv 1 \pmod{61}.$$

Il s'en suit que $a \equiv 0 \pmod{2}$. Comme a est nul modulo 2, nous résolvons pour b modulo 4, l'équation modulaire $h_0 \equiv g_1^b \pmod{61}$, ce qui implique que b est pair et ainsi $a \equiv 0 \pmod{4}$.

Algorithme 12 Algorithme de Pohlig-Hellman avec recherche exhaustive

Entrée : g d'ordre n , $h \in \langle g \rangle$ et $(p_i, \alpha_i)_{1 \leq i \leq r}$ tel que $n = \prod p_i^{\alpha_i}$ et p_i premier ne divisant pas $n/p_i^{\alpha_i}$

Sortie : a tel que $g^a = h$

```
1: pour  $i$  allant de 1 à  $r$  faire
2:    $a_i \leftarrow 0$ 
3:   pour  $j$  allant de 1 à  $\alpha_i$  faire
4:      $g_0 \leftarrow g^{n/p_i^j}$ 
5:      $h_0 \leftarrow g_0^{-a_i} h^{n/p_i^j}$ 
6:     si  $h_0 \neq 1$  alors
7:        $g_0 \leftarrow g^{n/p_i}$ 
8:        $b \leftarrow 1$ 
9:        $t \leftarrow g_0$ 
10:      tant que  $h_0 \neq t$  faire
11:         $b \leftarrow b + 1$ 
12:         $t \leftarrow t g_0$ 
13:      fin tant que
14:       $a_i \leftarrow a_i + b p_i^{j-1}$ 
15:    fin si
16:  fin pour
17: fin pour
18: calculer  $a \equiv a_i \pmod{p_i^{\alpha_i}}$ , pour  $1 \leq i \leq r$ , grâce à l'isomorphisme du
   théorème des restes chinois
19: retourner  $a$ 
```

Ensuite, calculons a modulo 3. Comme $(p-1)/3 = 20$, nous redéfinissons :

$$g_0 := g^{20} \equiv 47 \pmod{61} \quad \text{et} \quad h_0 := h^{20} \equiv 13 \pmod{61}.$$

Réolvons pour a modulo 3, l'équation modulaire $h_0 \equiv g_0^a \pmod{61}$, ce qui donne $a \equiv 2 \pmod{3}$.

Enfin, déterminons a modulo 5. Comme $(p-1)/5 = 12$, nous redéfinissons :

$$g_0 := g^{12} \equiv 34 \pmod{61} \quad \text{et} \quad h_0 := h^{12} \equiv 13 \pmod{61}.$$

Dont nous obtenons $a \equiv 1 \pmod{5}$.

Le théorème des restes chinois donne :

$$a \equiv 2 \cdot 4 \cdot 5 \cdot (4^{-1} \cdot 5^{-1} \pmod{3}) + 4 \cdot 3 \cdot (4^{-1} \cdot 3^{-1} \pmod{5}) \pmod{60},$$

ce qui fait $a = 2 \cdot 4 \cdot 5 \cdot 2 - 4 \cdot 3 \cdot 2 \equiv 56 \pmod{60}$, i.e. $7^{56} \equiv 25 \pmod{61}$.

Définition 7.5 (Entier B -friable). Un entier est dit B -friable si tous ses diviseurs premiers sont inférieurs ou égaux à B .

Théorème 7.6. Soit $g \in G$ d'ordre n . Soit $B \in \mathbb{N}$ tel que n est B -friable. Alors l'algorithme de Pohlig-Hellman résout le problème du logarithme discret en $\mathcal{O}(\log(n)^2 + B \log(n))$ opérations dans le groupe.

À cause de cette méthode, les nombres B -friables pour un petit B n'opposent pas de difficulté majeure à la recherche de logarithmes discrets. Donc,

en général, en cryptographie, il est préférable d'utiliser des éléments d'ordre un grand nombre premier.

Dans leur papier, S. Pohlig et M. Hellman proposent d'utiliser des entiers premiers de la forme $2p+1$ avec p premier : dans la littérature, $2p+1$ est appelé nombre premier sûr et p est appelé nombre premier de Sophie Germain.

7.3 Algorithme pas de bébé, pas de géant

Cet algorithme, souvent attribué à Shanks, est une incarnation du principe de *time/memory tradeoff*. Supposons g d'ordre un nombre premier r et $h = g^a$ pour un certain $0 \leq a < r$. Soit $m := \lceil \sqrt{r} \rceil$. Il existe alors deux entiers a_0 et a_1 tels que $a = a_0 + a_1 m$ et $0 \leq a_0, a_1 < m$. Il s'en suit :

$$g^{a_0} = h(g^{-m})^{a_1},$$

et cette observation mène à l'algorithme 13.

Cependant, l'utilisation importante de l'espace mémoire pour les valeurs calculées par l'algorithme baby-step-giant-step devient rapidement prohibitif, car il requiert $\mathcal{O}(\sqrt{r} \log(r))$ bits de stockage (si les éléments du groupe sont représentables en mémoire en $\mathcal{O}(\log(r))$ bits).

Algorithme 13 Algorithme baby-step-giant-step

Entrée : g d'ordre r , $h \in \langle g \rangle$

Sortie : a tel que $g^a = h$

```

1:  $m \leftarrow \lceil \sqrt{r} \rceil$ 
2:  $L$  est une liste vide
3:  $x \leftarrow 1$ 
4: pour  $i$  allant de 0 à  $m$  faire {Calculs des baby-steps}
5:    $L \leftarrow L + [x, i]$ 
6:    $x \leftarrow xg$ 
7: fin pour
8:  $u \leftarrow g^{-m}$ 
9:  $y \leftarrow h$ 
10:  $j \leftarrow 0$ 
11: tant que  $[y, *] \notin L$  faire {Calculs des giant-steps}
12:    $y \leftarrow yu$ 
13:    $j \leftarrow j + 1$ 
14: fin tant que
15: trouver  $[x, i] \in L$  tel que  $x = y$ 
16: retourner  $i + mj$ 
```

Exemple 7.7.

Théorème 7.8. Soit $g \in G$ d'ordre r . Si les éléments du groupe sont représentables en mémoire en $\mathcal{O}(\log(r))$ bits et si les opérations dans le groupe se font en $\mathcal{O}(\log(r)^2)$ opérations binaires, alors l'algorithme baby-step-giant-step résout le problème du logarithme discret en $\mathcal{O}(\sqrt{r} \log(r)^2)$ opérations binaires.

8 La méthode rho de Pollard

Après une introduction succincte, nous expliquons le fonctionnement de la méthode rho de Pollard, qui a été conçue en 1978 par John Michael Pollard [7]. Puis, nous étudierons les améliorations apportées par Edlyn Teske [9] en 2001 et Shi Bai et Richard Pierce Brent [1] en 2008.

Il est encouragé de consulter [4, section 14, pp. 264].

8.1 Discussion liminaire : partitionner, marcher, chercher

Introduction : La méthode rho permet, avec une forte probabilité, de résoudre le problème du logarithme discret – définition 7.1 – dans un groupe abélien. Elle se base sur des marches aléatoires : elle définit donc un algorithme non-déterministe, i.e. sur une même entrée, chaque exécution de l'algorithme peut avoir une issue distincte. Soit $G := \langle g \rangle$ et supposons g d'ordre un nombre premier r – le cas ardu par Pohlig-Hellman – et $h = g^a$ pour un certain $0 \leq a < r$.

Notation : Si W est un ensemble et w est un élément de W choisi aléatoirement selon la distribution uniforme, alors nous notons $w \in_a W$.

Idée : Observons que si nous trouvons a_i, b_i, a_j et b_j dans $\mathbb{Z}/r\mathbb{Z}$ tels que :

$$(\star) \quad g^{a_i} h^{b_i} = g^{a_j} h^{b_j}$$

et $b_i \not\equiv b_j \pmod{r}$, alors nous pouvons résoudre le problème du logarithme discret comme suit :

$$h = g^{(a_i - a_j)(b_j - b_i)^{-1} \bmod r}.$$

Marche pseudo-aléatoire : Pollard [7] s'est inspiré de la théorie des marches aléatoires et a donc pensé à construire une suite pseudo-aléatoire $x_i = g^{a_i} h^{b_i}$ d'éléments de G par itération d'une fonction convenable $f : G \rightarrow G$, appelée *fonction d'itération*. En d'autres termes, il choisit une valeur de départ x_1 et définit le reste de la suite par $x_{i+1} = f(x_i)$, pour $i \geq 1$. Une telle suite x_1, x_2, \dots est appelée une *marche pseudo-aléatoire déterministe*.

Comme G est fini, cette suite est *ultimement périodique* : il existe deux entiers $\lambda \geq 1$ et $\mu \geq 0$ tels que les éléments $x_1, x_2, \dots, x_{\lambda+\mu-1}$ soient distincts deux à deux et $x_k = x_{\lambda+k}$ pour tout $k \geq \mu$. Nous appelons μ la *pré-période de la suite* et λ la *période de la suite*. Sous l'hypothèse que $x_1 \in_a G$ et que f est une fonction aléatoire, la valeur attendue de λ et μ est proche de

$$\sqrt{\frac{\pi r}{8}} \approx 0.627\sqrt{r}.$$

Collision : Une pair (x_i, x_j) est appelée *collision* si $x_i = x_j$ et $1 \leq i < j$, comme dans (\star) . Si les éléments dans la marche sont choisis uniformément et indépendants dans G , alors par le paradoxe des anniversaires (théorème F.1), la valeur attendue pour j est

$$\sqrt{\frac{\pi r}{2}} \approx 1.253\sqrt{r}.$$

Rho : Comme l'image obtenue en “dessinant” les termes de $(x_i)_{i \geq 1}$, en commençant en bas et finissant par le cycle en haut, forme la lettre grecque ρ , une méthode utilisant une telle suite s'appelle une *méthode rho*.

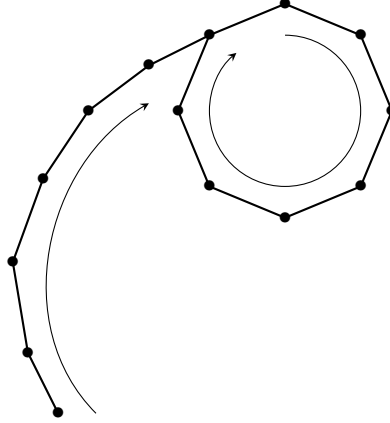


FIGURE 1 – Illustration de la forme “rho”

Partition : Pour simuler une fonction aléatoire – la fonction d’itération – de G dans lui-même, il est pratique de commencer par partitionner G en n ensembles¹ (d’à peu près la même taille), de sorte que $G = \mathcal{S}_0 \cup \mathcal{S}_1 \cup \dots \cup \mathcal{S}_{n-1}$. Les ensembles \mathcal{S}_i sont définis en utilisant une *fonction de sélection*

$$s : G \rightarrow \{0, \dots, n-1\},$$

par $\mathcal{S}_i = \{g \in G : s(g) = i\}$.

Résumé : Les trois ingrédients principaux d’une bonne méthode rho sont : une partition de G grâce à une fonction de sélection, une suite $(x_i)_{i \geq 1}$ à l’aspect aléatoire construite grâce à une fonction d’itération et à la partition précédente – la fonction d’itération a autant de règles que le cardinal de la partition – et un algorithme pour chercher la collision dans la suite $(x_i)_{i \geq 1}$.

8.2 Méthode rho de Pollard originale dans $(\mathbb{Z}/r\mathbb{Z})^\times$

Dans [7], Pollard décrit sa méthode pour $G = (\mathbb{Z}/r\mathbb{Z})^\times$, où r est un nombre premier, et g un élément primitif du corps $\mathbb{Z}/r\mathbb{Z}$.

Remarque 8.1. Ici, r est premier mais le groupe G est d’ordre $r-1$.

Soit $h \in (\mathbb{Z}/r\mathbb{Z})^\times$ dont nous cherchons le logarithme en base g . La fonction d’itération $f : (\mathbb{Z}/r\mathbb{Z})^\times \rightarrow (\mathbb{Z}/r\mathbb{Z})^\times$ utilisée par Pollard est :

$$f(x) = \begin{cases} gx & \text{si } 0 < x \leq p/3, \\ x^2 & \text{si } p/3 < x \leq 2p/3, \\ hx & \text{si } 2p/3 < x < p. \end{cases}$$

1. En pratique, les valeurs typiques de n sont 32, 256 ou 2048.

La suite est alors définie par $x_1 = 1$ et $x_{i+1} = f(x_i)$ pour $i \geq 1$. Pour garder trace de la décomposition $x_i = g^{a_i} h^{b_i}$, nous calculons les suites $(a_i)_{i \geq 1}$ et $(b_i)_{i \geq 1}$ comme suit :

$$a_1 = 0 \quad \text{et} \quad a_{i+1} = \begin{cases} a_i + 1 \pmod{r-1} & \text{si } 0 < x \leq p/3, \\ 2a_i \pmod{r-1} & \text{si } p/3 < x \leq 2p/3, \\ a_i \pmod{r-1} & \text{si } 2p/3 < x < p. \end{cases}$$

$$b_1 = 0 \quad \text{et} \quad b_{i+1} = \begin{cases} b_i \pmod{r-1} & \text{si } 0 < x \leq p/3, \\ 2b_i \pmod{r-1} & \text{si } p/3 < x \leq 2p/3, \\ b_i + 1 \pmod{r-1} & \text{si } 2p/3 < x < p. \end{cases}$$

Nous rassemblons cela sous la notation :

$$(x_{i+1}, a_{i+1}, b_{i+1}) = \text{walk}(x_i, a_i, b_i).$$

Remarque 8.2. Il faut garder à l'esprit que les valeurs a_{i+1} et b_{i+1} sont uniquement déterminées par la valeur x_i , et que la fonction **walk** dépend seulement de la fonction d'itération f .

Maintenant, il faut rechercher une collision dans la suite $(x_i)_{i \geq 1}$. Pollard utilise l'*algorithme de recherche de cycle Floyd*, aussi appelé l'*algorithme du lièvre et de la tortue*. Cet algorithme compare x_i et x_{2i} , et donne une collision (x_i, x_{2i}) , pour $i \geq 1$. Le plus petit i tel que $x_i = x_{2i}$ est appelé l'*épacte*.

Notons μ la pré-période de la suite et λ la période de la suite.

Proposition 8.3. *En gardant les notations précédentes, $x_{2i} = x_i$ si, et seulement si, λ divise i et $i \geq \mu$. En outre, il existe un $\mu \leq i < \mu + \lambda$ tel que $x_{2i} = x_i$.*

Démonstration. Si $x_i = x_j$, avec $1 \leq i < j$, alors λ divise $j - i$, ce qui prouve la première assertion du lemme. La seconde vient du fait qu'il existe un multiple de λ entre μ et $\mu + \lambda$. \square

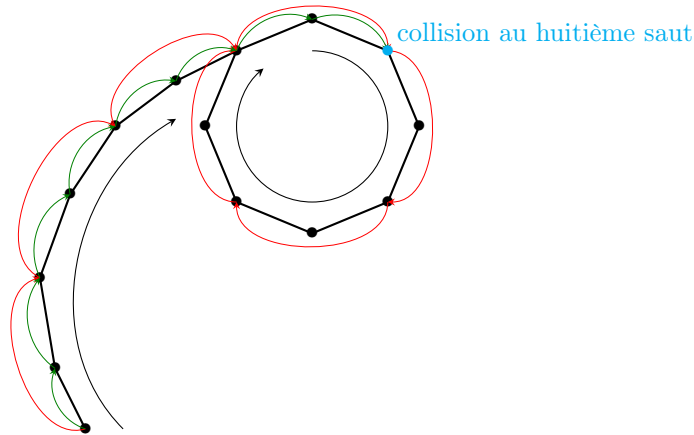


FIGURE 2 – Illustration de l'algorithme du lièvre (rouge) et de la tortue (verte)

Algorithme 14 Algorithme rho de Pollard avec recherche de cycle de Floyd

Entrée : r un nombre premier, $g \in \mathbb{Z}/r\mathbb{Z}$ primitif et $h \in (\mathbb{Z}/r\mathbb{Z})^\times$

Sortie : a tel que $g^a = h$ avec $1 \leq a < r$, ou “échec”

```
1:  $x_1 \leftarrow 1$ 
2:  $a_1 \leftarrow 0$ 
3:  $b_1 \leftarrow 0$ 
4:  $(x_2, a_2, b_2) \leftarrow \text{walk}(\text{walk}(x_1, a_1, b_1))$ 
5:  $(x_1, a_1, b_1) \leftarrow \text{walk}(x_1, a_1, b_1)$ 
6: tant que  $x_1 \neq x_2$  faire
7:    $(x_1, a_1, b_1) \leftarrow \text{walk}(x_1, a_1, b_1)$ 
8:    $(x_2, a_2, b_2) \leftarrow \text{walk}(\text{walk}(x_2, a_2, b_2))$ 
9: fin tant que
10:  $d \leftarrow \text{pgcd}(b_1 - b_2, r - 1)$ 
11: si  $d$  divise  $(a_2 - a_1)$  alors
12:   résoudre l'équation diophantienne  $u(b_1 - b_2)/d + v(r - 1)/d = (a_2 - a_1)/d$ 
13:    $\alpha \leftarrow g^{u(a_2 - a_1)/d} \pmod{r}$ 
14:    $\vartheta \leftarrow g^{(r-1)/d} \pmod{r}$ 
15:   pour tout  $j$  allant de 0 à  $d - 1$  faire
16:     si  $\alpha \vartheta^j = h \pmod{r}$  alors
17:       retourner  $u(a_2 - a_1)/d + j(r - 1)/d \pmod{r - 1}$ 
18:     fin si
19:   fin pour
20: sinon
21:   retourner “échec”
22: fin si
```

Les lignes 10 à 20 de l'algorithme 14 méritent de plus amples explications. Quand une collision (x_i, x_{2i}) est atteinte, nous avons l'égalité :

$$(*) \quad h^n = g^m, \quad \text{avec} \quad \begin{cases} m & \equiv a_{2i} - a_i \pmod{r-1}, \\ n & \equiv b_i - b_{2i} \pmod{r-1}. \end{cases}$$

Or $h^n = (g^a)^n$, ce qui donne

$$an \equiv m \pmod{r-1}.$$

Si n est inversible modulo $r - 1$, alors nous obtenons directement le logarithme de h en base g : $a \equiv mn^{-1} \pmod{r-1}$. En revanche, si tel n'est pas le cas, alors nous pouvons le recouvrer si, et seulement si, $d := \text{pgcd}(n, r - 1)$ divise m . En effet, cela revient à résoudre l'équation diophantienne :

$$(\dagger) \quad nx + (r - 1)y = m.$$

Nous notons u et v les entiers de l'identité de Bézout :

$$(\ddagger) \quad un + v(r - 1) = d.$$

Les entiers x , tels que qu'il existe $y \in \mathbb{Z}$ et (x, y) est solution de (\dagger) , forment l'ensemble :

$$u \frac{m}{d} + \frac{r-1}{d} \mathbb{Z}.$$

Ainsi, il existe un unique k tel que $0 \leq k \leq d-1$ et

$$a = u \frac{m}{d} + k \frac{r-1}{d}.$$

Posons

$$\alpha := g^{um/d} \quad \text{et} \quad \vartheta := g^{(r-1)/d}.$$

L'élément ϑ est une racine d -ième de l'unité dans $\mathbb{Z}/r\mathbb{Z}$ et $h = \alpha\vartheta^k$ dans $\mathbb{Z}/r\mathbb{Z}$. Nous pouvons alors chercher k de manière exhaustive, sachant que nous avons au préalable calculé α et ϑ .

La façon de faire de Pollard [7] est la suivante : après avoir calculer la relation de Bézout (\ddagger) , nous élevons $(*)$ à la puissance u , donnant ainsi :

$$(h^n)^u = h^d = g^{um},$$

où um est de la forme $d\lambda$, car d divise m , par $an \equiv m \pmod{r-1}$ et (\ddagger) . Ainsi,

$$h = g^\lambda \vartheta^i,$$

où $\vartheta := g^{(r-1)/d}$ est une racine d -ième de l'unité, et i , tel $0 \leq i \leq d-1$, reste à déterminer.

8.3 Méthode rho de Pollard originale généralisée

Définition 8.4 (rho-marche). Une *rho-marche* est définie comme suit. Pré-calculer $g_j = g^{u_j} h^{v_j}$ pour $0 \leq j \leq n_s - 1$, où $0 \leq u_j, v_j < r$ sont choisis de façon uniformément aléatoire. Poser $x_1 = g$. La *rho-marche originale* est

$$x_{i+1} = f(x_i) = \begin{cases} x_i^2 & \text{si } S(x_i) = 0, \\ x_i g_j & \text{si } S(x_i) = j, \quad 1 \leq j \leq n_s - 1. \end{cases}$$

La *rho-marche additive* est

$$x_{i+1} = f(x_i) = x_i g_{S(x_i)}.$$

Remarque 8.5. Une fois la fonction de sélection S et les valeurs de u_j et v_j choisies, la marche est déterministe.

Il est nécessaire de garder trace de la décomposition

$$x_i = g^{a_i} h^{b_i}.$$

Pour la rho-marche originale, les valeurs $a_i, b_i \in \mathbb{Z}/r\mathbb{Z}$ sont obtenues en posant $a_1 = 1, b_1 = 0$ et mettant à jour :

$$a_{i+1} = \begin{cases} 2a_i & \pmod{r} & \text{si } S(x_i) = 0, \\ a_i + u_{S(x_i)} & \pmod{r} & \text{si } S(x_i) > 0, \end{cases}$$

$$b_{i+1} = \begin{cases} 2b_i & \pmod{r} & \text{si } S(x_i) = 0, \\ b_i + v_{S(x_i)} & \pmod{r} & \text{si } S(x_i) > 0. \end{cases}$$

Pour la rho-marche additive, les valeurs $a_i, b_i \in \mathbb{Z}/r\mathbb{Z}$ sont obtenues par :

$$a_1 = 1 \text{ et } b_1 = 0,$$

$$a_{i+1} = a_i + u_{S(x_i)} \pmod{r},$$

$$b_{i+1} = b_i + v_{S(x_i)} \pmod{r}.$$

Nous rassemblons cela sous la notation :

$$(x_{i+1}, a_{i+1}, b_{i+1}) = \text{walk}(x_i, a_i, b_i).$$

9 L'algorithme $L(1/4 + o(1))$ de Joux

Soit p un nombre premier et n un entier. Notons :

$$L_Q(\alpha, c) := \exp \left((c + o(1)) (\log Q)^\alpha (\log \log Q)^{1-\alpha} \right),$$

où $Q := p^n$, $0 \leq \alpha \leq 1$ et $c \geq 0$.

9.1 Travaux précédents le logarithme discret en petite caractéristique

10 La réduction de réseau Lenstra-Lenstra-Lovász

11 L'attaque Lagarias-Odlyzko sur les sacs à dos à densité faible

A Le b.a.-ba de la théorie des groupes

Définition A.1 (Groupe).

B L'anatomie des corps

Nous souhaitons nous rafraîchir la mémoire sur la théorie des corps. Malheureusement, les rudiments sur les anneaux ne sont pas rappelés, il est vivement conseillé de consulter le manuel de Patrice Tauvel [8], cette partie l'utilise d'ailleurs extensivement, de même que le cours d'algèbre de Michel Demazure [3] pour la partie sur les corps finis.

Commençons par donner quelques définitions :

Définition B.1 (Corps). Soit $(K, +_K, \cdot_K)$, où K est un ensemble, $+_K : K \times K \rightarrow K$ et $\cdot_K : K \times K \rightarrow K$ sont deux lois de composition interne. Pour faire de $(K, +, \cdot)$ un *corps*, ces deux lois doivent satisfaire aux conditions suivantes :

1. $(K, +_K)$ est un groupe abélien, dont l'élément neutre est noté 0_K .
2. $(K \setminus \{0_K\}, \cdot_K)$ est un groupe abélien, dont l'élément neutre est noté 1_K .
3. La loi \cdot_K se distribue sur la loi $+_K$.

Pour alléger les notations, les “ K ” en indice seront omis – mais cette écriture sera utilisée pour ôter le doute en cas d'ambiguïté. De même, nous dirons “Soit K un corps”, au lieu de “Soit $(K, +, \cdot)$ un corps” – les lois seront sous-entendues. La loi $+$ est appelée *addition*, et la loi \cdot est appelée *multiplication*, et comme à l'accoutumée, le produit $a \cdot b$ sera noté ab .

Exemple B.2. Les complexes \mathbb{C} , les réels \mathbb{R} et les rationnels \mathbb{Q} , munis des opérations usuels, sont des corps. Pour tout p nombre premier, l'ensemble des classes d'équivalence $\mathbb{Z}/p\mathbb{Z}$, munis des opérations induites par celles de \mathbb{Z} , est un corps.

Remarque B.3. La définition implique que, quel que soit $a \in K$, $0 \cdot a = 0$ et $-a = (-1) \cdot a$. De plus, pour tous a et b dans K , si $ab = 0$ alors a ou b est nul.

Exemple B.4. $\mathbb{Z}/n\mathbb{Z}$ n'est pas un corps, dès que n est composé. En effet, soit d un diviseur de n distinct de 1 et n . Alors la classe d'équivalence de d dans $\mathbb{Z}/n\mathbb{Z}$ est un diviseur de zéro, donc $\mathbb{Z}/n\mathbb{Z}$ n'est pas intègre.

Définition B.5 (Groupe multiplicatif). Soit K un corps. $(K \setminus \{0\}, \cdot)$ est un groupe appelé *groupe multiplicatif du corps K* , et il est noté K^\times .

Anneau de polynôme sur un corps

Théorème B.6. Soit K un corps. L'anneau $K[X]$ est principal. En outre, si \mathfrak{a} est un idéal non nul de $K[X]$, il existe un unique polynôme unitaire $U(X)$ de $K[X]$ tel que

$$\mathfrak{a} = (U(X)) := U(X)K[X].$$

Démonstration. Nous avons $\{0\} = (0)$. Soient \mathfrak{a} un idéal non nul de $K[X]$ et $P(X) \in \mathfrak{a}$, de degré minimal parmi les éléments non-nuls de \mathfrak{a} . Nous avons déjà $(P(X)) \subseteq \mathfrak{a}$. Montrons l'autre inclusion : soit $S(X) \in \mathfrak{a}$. La division euclidienne de $S(X)$ par $P(X)$ donne l'écriture $S(X) = Q(X)P(X) + R(X)$, où $Q(X)$ et $R(X)$ sont dans $K[X]$ et $\deg(R(X)) < \deg(P(X))$ ou $R(X) = 0$. Ainsi :

$$R(x) = S(X) - P(X)Q(X) \in \mathfrak{a}.$$

Il s'en suit, par le choix de $P(X)$, que $R(X) = 0$. Cela montre que $K[X]$ est principal. Dans ce qui précède, nous pouvons supposer $P(X)$ unitaire. Si $S(X)$ est tel que $(S(X)) = (P(X))$, alors $S(X)$ et $P(X)$ sont deux polynômes associés. Donc si $Q(X)$ est unitaire, nous avons l'égalité $S(X) = Q(X)$. \square

Proposition B.7. *L'anneau $A[X]$ est un anneau principal si, et seulement si, l'anneau A est un corps.*

Théorème B.8. *Soit K un corps. Soit $P(X) \in K[X]$ de degré $d \geq 1$ et notons \mathfrak{a} l'idéal de $K[X]$ engendré par $P(X)$. Notons $\varphi : K[X] \rightarrow \mathcal{A} := K[X]/\mathfrak{a}$ la surjection canonique, et posons $x = \varphi(X)$.*

1. $(1_{\mathcal{A}}, x, \dots, x^{d-1})$ est une base du K -espace vectoriel \mathcal{A} .
2. Les conditions suivantes sont équivalentes :
 - (a) L'anneau \mathcal{A} est intègre.
 - (b) L'anneau \mathcal{A} est un corps.
 - (c) Le polynôme $P(X)$ est irréductible.

Démonstration. Nous prouvons les points dans l'ordre.

1. Si $A(X) \in K[X]$, nous pouvons écrire $A(X) = P(X)Q(X) + R(X)$, avec $\deg(R(X)) \leq d-1$. Ainsi $\varphi(A(X)) = \varphi(R(X))$ et $(1_{\mathcal{A}}, x, \dots, x^{d-1})$ est une famille génératrice de \mathcal{A} . S'il existe $\lambda_0, \lambda_1, \dots, \lambda_{d-1} \in K$ tel que

$$\lambda_0 1_{\mathcal{A}} + \lambda_1 x + \dots + \lambda_{d-1} x^{d-1} = 0,$$

alors le polynôme $\lambda_0 + \lambda_1 X + \dots + \lambda_{d-1} X^{d-1}$ est dans \mathfrak{a} et est de degré strictement inférieur à d : cela implique que tous les λ_i soient nuls.

2. L'implication $(b) \Rightarrow (a)$ est claire.
Montrons $(a) \Rightarrow (c)$. Par l'absurde : écrivons $P(X) = Q(X)R(X)$, avec $\deg(Q) < d$, $\deg(R) < d$. Alors $\varphi(Q)$ et $\varphi(R)$ sont non-nuls, bien que $\varphi(Q)\varphi(R) = \varphi(P)$ le soit.
Montrons $(c) \Rightarrow (b)$. Comme $d \geq 1$, on a $\mathfrak{a} \neq K[X]$. Soit \mathfrak{b} un idéal de $K[X]$ contenant strictement \mathfrak{a} et soit $Q(X) \in \mathfrak{b} \setminus \mathfrak{a}$. Si P est irréductible, il est étranger à Q et, grâce à une identité de Bézout, nous obtenons :

$$K[X] = (P(X)) + (Q(X)) \subseteq \mathfrak{b}.$$

D'où $\mathfrak{b} = K[X]$. Par conséquent, \mathfrak{a} est un idéal maximal de $K[X]$, et \mathcal{A} est un corps. \square

B.1 Homomorphismes et caractéristique

Définition B.9 (Homomorphisme d'anneaux). Soient A et B deux anneaux. L'application $f : A \rightarrow B$ est un homomorphisme d'anneaux si :

1. $\forall (a, b) \in A^2 : f(a + b) = f(a) + f(b)$,
2. $\forall (a, b) \in A^2 : f(ab) = f(a)f(b)$,
3. $f(1_A) = 1_B$.

Notation : Soit A un anneau. Pour tous $a \in A$ et $n \in \mathbb{Z}$, notons

$$na := \begin{cases} a + a + \cdots + a \text{ (} n \text{ termes)} & \text{si } n > 0, \\ 0 & \text{si } n = 0, \\ (-a) + (-a) + \cdots + (-a) \text{ (} -n \text{ termes)} & \text{si } n < 0. \end{cases}$$

Proposition B.10. Soit A un anneau. L'application $\phi_A : \mathbb{Z} \rightarrow A$ définie par $\phi_A(n) := n1_A$, est un homomorphisme d'anneaux dont le noyau est $c\mathbb{Z}$ avec $c \in \mathbb{N}$.

Démonstration. ϕ_A est bien un homomorphisme d'anneaux et le noyau ϕ_A est un idéal de \mathbb{Z} , donc de la forme $c\mathbb{Z}$ avec $c \in \mathbb{N}$. \square

Définition B.11 (Caractéristique). Soit A un anneau. La *caractéristique* de l'anneau A , noté $\text{car}(A)$, est l'entier $c \in \mathbb{N}$ tel que $\ker(\phi_A) = c\mathbb{Z}$.

Exemple B.12. $\text{car}(\mathbb{Z}) = \text{car}(\mathbb{Q}) = \text{car}(\mathbb{R}) = \text{car}(\mathbb{C}) = 0$.
Pour n entier, $\text{car}(\mathbb{Z}/n\mathbb{Z}) = n$.

Proposition B.13. Si A est un anneau intègre, alors sa caractéristique nulle ou est un nombre premier.

Démonstration. Si $\text{car}(A) = pq$, avec p et q des entiers non-nuls distincts de 1, alors les éléments $p1_A$ et $q1_A$ sont non-nuls de produit nul. Ainsi A n'est pas intègre. \square

Exemple B.14. La réciproque est fautive : $\mathbb{R} \times \mathbb{R}$ est un anneau de caractéristique nulle alors qu'il n'est pas intègre.

B.2 Endomorphisme de Frobenius

Rappel : Soit A un anneau. Si a et b sont deux éléments de A qui commutent l'un à l'autre, i.e. $ab = ba$, alors pour $n \in \mathbb{N}$, la *formule du binôme de Newton* donne :

$$(a + b)^n = \sum_{k=0}^n \binom{n}{k} a^k b^{n-k},$$

où $\binom{n}{k}$ est le nombre de combinaison de k objets dans un ensemble à n objets :

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}.$$

Proposition B.15. Soit A un anneau commutatif de caractéristique un nombre premier p . L'application

$$f : \begin{array}{ccc} A & \longrightarrow & A \\ x & \longmapsto & x^p \end{array}$$

est un endomorphisme de l'anneau A .

Démonstration. Soit $0 < k < p$. La formule

$$k \binom{p}{k} = p \binom{p-1}{k-1},$$

et le lemme d'Euclide impliquent que p divise $\binom{p}{k}$. Nous pouvons maintenant démontrer la proposition. Soit $(a, b) \in A^2$. Comme A est commutatif, nous pouvons appliquer la formule du binôme de Newton :

$$f(a + b) = (a + b)^p = \sum_{k=0}^p \binom{p}{k} a^k b^{p-k} = a^p + b^p = f(a) + f(b).$$

De plus, $f(ab) = (ab)^p = a^p b^p = f(a)f(b)$, car A est commutatif, et $f(1) = 1$. Donc f est bien un endomorphisme de l'anneau A . \square

Exemple B.16. La proposition n'est plus vraie dès que la caractéristique de l'anneau est un nombre composé : plaçons nous dans $\mathbb{Z}/4\mathbb{Z}$, nous avons d'une part $(1 + 1)^4 = 2^4 = 0$ et d'autre part $1^4 + 1^4 = 1 + 1 = 2$. Alors que clairement $0 \neq 2$ dans $\mathbb{Z}/4\mathbb{Z}$.

Définition B.17 (Endomorphisme de Frobenius). Soit A un anneau commutatif de caractéristique un nombre premier p . L'application $x \mapsto x^p$ est appelé l'*endomorphisme de Frobenius* de l'anneau A . Il est noté Frob_A .

B.3 Sous-corps premiers

Définition B.18 (Sous-corps). Soit L un corps. Une partie $K \subseteq L$ est un *sous-corps* de L si K est un corps pour les opérations héritées de L .

Exemple B.19. \mathbb{Q} est un sous-corps de \mathbb{R} et \mathbb{R} est un sous-corps de \mathbb{C} .

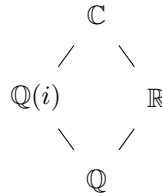


FIGURE 3 – Inclusions de corps

Définition B.20 (Corps premier). K est un *corps premier* si K est un corps ne contenant aucun sous-corps autre que lui-même.

Proposition B.21. Le corps \mathbb{Q} est un corps premier et, pour tout nombre premier p , le corps $\mathbb{Z}/p\mathbb{Z}$ est un corps premier.

Définition B.22 (Sous-corps premier). Soit K un corps. Le sous-corps premier de K est l'intersection de tous ses sous-corps.

Proposition B.23. Soient K un corps et P le sous-corps premier de K . Si $\text{car}(K) = 0$ alors P est isomorphe à \mathbb{Q} , et si $\text{car}(K) \neq 0$ alors $p = \text{car}(K)$ est un nombre premier et P est isomorphe à $\mathbb{Z}/p\mathbb{Z}$.

B.4 Extensions de corps

Définition B.24 (Homomorphisme de corps). Un *homomorphisme de corps* est un homomorphisme d'anneaux (voir définition B.9).

Proposition B.25. *Un homomorphisme de corps est injectif.*

Démonstration. Soient K_1 et K_2 deux corps et $f : K_1 \rightarrow K_2$ un homomorphisme de corps. Rappelons que les seuls idéaux d'un corps sont l'idéal nul et le corps tout entier. Le noyau $\ker(f)$ est un idéal de K_1 , et par la remarque précédente, de deux choses l'une : soit $\ker(f) = K_1$, soit $\ker(f) = \{0_{K_1}\}$. Or comme f est un homomorphisme d'anneaux, $f(1_{K_1}) = 1_{K_2} \neq 0_{K_2}$, cela implique donc $\ker(f) \neq K_1$. D'où $\ker(f) = \{0_{K_1}\}$ et f est injective. \square

Définition B.26 (Extension de corps). Soit K un corps. Une *extension* du corps K est un couple (L, ι) , où L est un corps et ι est un homomorphisme de corps.

Soit (L, ι) une extension du corps K . Comme ι est injective, nous pouvons identifier K à son image $\iota(K)$ dans L . Dans la pratique, sauf risque de confusion, nous faisons systématiquement cette identification : nous disons alors que L est une extension de K si K est un sous-corps de L , et nous le notons alors $K \subseteq L$. Il est alors évident que K et L ont la même caractéristique.

Exemple B.27. Considérons le polynôme $x^3 - 3 \in \mathbb{Q}[x]$. Dans \mathbb{C} , il admet α , αj et αj^2 pour racines, où $\alpha := \sqrt[3]{3}$ et $j := e^{2i\pi/3}$. Les extensions sont alors :

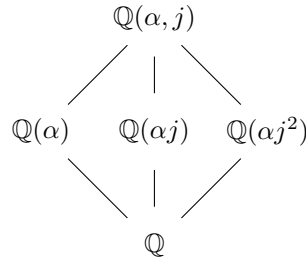


FIGURE 4 – Sous-corps de $\mathbb{Q}(\alpha, j) = \mathbb{Q}(\alpha, \alpha j, \alpha j^2)$

Proposition B.28. *Soit $K \subseteq L$ une extension de corps. L est un K -espace vectoriel. Une base du K -espace vectoriel L est appelée une base de L sur K .*

Définition B.29 (Degré d'une extension). Soit $K \subseteq L$ une extension de corps. Le *degré de l'extension* $K \subseteq L$ est le cardinal, fini ou non, d'une base de L sur K . Le degré est noté $[L : K] := \dim_K(L)$. Une extension de degré fini (resp. 2, 3) est dite finie (resp. quadratique, cubique).

Exemple B.30. L'extension de corps $\mathbb{R} \subseteq \mathbb{C}$ est une extension quadratique.

Définition B.31. Soient $K \subseteq L$ une extension de corps et S une partie de L . Notons $K[S]$ le sous-anneau de L engendré par K et S , et $K(S)$ le sous-corps de L engendré par K et S .

Si $S = \{x_1, \dots, x_n\}$, nous notons $K[x_1, \dots, x_n]$ pour $K[S]$ et $K(x_1, \dots, x_n)$ pour $K(S)$.

Définition B.32 (Extension de type fini). L'extension $K \subseteq L$ est *type fini* s'il existe une partie finie S de L telle que $L = K(S)$.

Définition B.33 (Extension simple et élément primitif). L'extension $K \subseteq L$ est *simple* s'il existe une partie S de L réduite à un élément telle que $L = K(S)$. Tout élément x de L vérifiant $L = K(x)$ est appelé un *élément primitif* de l'extension.

Proposition B.34. Soient $K \subseteq L$ une extension de corps et S, T deux parties de L , et \mathcal{F} l'ensemble des parties finies de S . Alors :

$$K(S \cup T) = K(S)(T) = K(T)(S), \quad K(S) = \bigcup_{U \in \mathcal{F}} K(U).$$

Définition B.35 (Sous-extension de corps). Une *sous-extension* d'une extension de corps $K \subseteq M$ est un corps L tel que $K \subseteq L \subseteq M$. En d'autres termes, c'est un sous-corps de M contenant K .

Théorème B.36 (Multiplicativité du degré). Soient $K \subseteq L \subseteq M$ des extensions de corps.

1. Soient $(e_i)_{i \in I}$ une base de L sur K et $(f_j)_{j \in J}$ une base de M sur L . Alors la famille $(e_i f_j)_{(i,j) \in I \times J}$ est une base de M sur K .
2. L'extension $K \subseteq M$ est finie si, et seulement si, les extensions $K \subseteq L$ et $L \subseteq M$ le sont. Si tel est le cas :

$$[M : K] = [M : L][L : K].$$

Démonstration. Il suffit de montrer la première assertion. Soient $(\lambda_{ij})_{(i,j) \in I \times J}$ une famille presque nulle d'élément de K . Alors

$$\sum_{i,j} \lambda_{ij} e_i f_j = 0 \Rightarrow \sum_j \left(\sum_i \lambda_{ij} e_i \right) f_j = 0.$$

Pour tout $(i, j) \in I \times J$, les éléments $\lambda_{ij} e_i$ sont dans L . Du fait que la famille $(f_j)_{j \in J}$ est libre sur L , nous déduisons :

$$\sum_i \lambda_{ij} e_i = 0,$$

quel que soit $j \in J$. Comme $(e_i)_{i \in I}$ est une base de L sur K , les λ_{ij} sont nuls pour tout $(i, j) \in I \times J$. Le reste se démontre aisément. \square

B.5 Extensions algébriques

Un idéal \mathfrak{a} d'un anneau commutatif A est dit *premier* s'il vérifie les conditions suivantes équivalentes :

1. A/\mathfrak{a} est un anneau intègre.
2. $\mathfrak{a} \neq A$ et, si $(a, b) \in A^2$ vérifie $ab \in \mathfrak{a}$, alors soit $a \in \mathfrak{a}$, soit $b \in \mathfrak{a}$.

Soient $K \subseteq L$ une extension de corps et $x \in L$. L'application

$$\begin{aligned} \phi_x : K[X] &\longrightarrow L \\ P &\longmapsto P(x), \end{aligned}$$

est un homomorphisme de K -algèbres d'image $K[x]$. Prosaiquement, $K[X]$ et L sont vus comme des K -espaces vectoriels munis d'une multiplication interne, ou encore, comme deux anneaux stables par multiplication par un élément du corps K .

Notons \mathfrak{a}_x le noyau de ϕ_x . De deux choses l'une :

- ▷ $\mathfrak{a} = \{0\}$: l'élément x est dit *transcendant* sur K . L'application ϕ_x induit un isomorphisme de $K[X]$ sur $K[x]$ qui se prolonge en un isomorphisme de $K(X)$ sur $K(x)$. Le degré de l'extension $K \subseteq K(x)$ est infini.
- ▷ $\mathfrak{a} \neq \{0\}$: l'élément x est dit *algébrique* sur K . Comme L est intègre, l'idéal \mathfrak{a}_x est un idéal premier de $K[X]$. D'après le théorème B.6, il existe un unique polynôme unitaire irréductible $\mu_{x,K}(X)$ de $K[X]$ tel que $\mathfrak{a}_x = (\mu_{x,K}(X))$. Nous disons que le polynôme $\mu_{x,K}$ est le *polynôme minimal* de x sur K . Son degré, noté $\deg_K(x)$, est appelé le *degré de x sur K* . Le polynôme $\mu_{x,K}$ sera noté μ_x .

Par passage au quotient, ϕ_x induit un isomorphisme de K -algèbres :

$$\psi_x : \frac{K[X]}{(\mu_x(X))} \longrightarrow K[x].$$

Compte tenu du théorème B.8, nous avons $K(x) = K[x]$, et $\psi_x(K)$ est isomorphe à K . Nous pouvons identifier K et $\psi_x(K)$, et considérer $K(x)$ comme une extension de K . Si nous posons $n = \deg_K(x)$, toujours d'après le théorème B.8, $\{1, x, \dots, x^{n-1}\}$ est une base du K -espace vectoriel $K(x)$. Ainsi :

$$[K(x) : K] = \deg_K(x).$$

Définition B.37 (Extension algébrique). Une extension $K \subseteq L$ est dite *algébrique* si tout élément de L est algébrique sur K . Si c'est le cas, nous pouvons aussi dire que L est algébrique sur K .

Définition B.38 (Extension transcendante). Une extension $K \subseteq L$ est dite *transcendante* si elle n'est pas algébrique.

Proposition B.39. Soit $K \subseteq L$ une extension de corps.

1. Si l'extension est finie, elle est algébrique. En outre, pour tout $x \in L$, l'entier $\deg_K(x)$ divise $[L : K]$.
2. Dire que $x \in L$ est algébrique sur K signifie que l'extension $K \subseteq K(x)$ est de degré fini.

Démonstration. Dans l'ordre :

1. Soit $x \in L$. Si l'extension $K \subseteq L$ est finie, la famille $(x^n)_{n \in \mathbb{N}}$ est liée sur K . Il existe donc $P(X) \in K[X] \setminus \{0\}$ tel que $P(x) = 0$. Ainsi, x est algébrique sur K . On a alors $[L : K] = [L : K(x)][K(x) : K]$, d'où découle le second point, puisque $[K(x) : K] = \deg_K(x)$.
2. Si x est algébrique sur K , l'extension $K \subseteq K(x)$ est finie (c'est le second "▷" au dessus). La réciproque est un cas particulier du premier point.

□

Remarque B.40. Une extension algébrique n'est pas nécessairement de degré finie. En revanche si $K \subseteq L$ est une extension de corps, toute K -sous-algèbre de L est un corps.

Proposition B.41. Soient $K \subseteq L$ une extension et S une partie de L . On suppose que tout élément de S est algébrique sur K . Alors,

1. l'extension $K \subseteq K(S)$ est algébrique,
2. si S est finie, l'extension $K \subseteq K(S)$ est finie.

Démonstration. D'après la proposition B.34, il suffit de prouver le second point. C'est clair si $S = \emptyset$. Supposons $S \neq \emptyset$ et l'assertion établie pour les parties de K de cardinal strictement inférieur à celui de S . Soient $x \in S$ et $T = S \setminus \{x\}$. Nous avons $K(S) = K(T)(x)$. D'après l'hypothèse de récurrence, $[K(T) : K]$ est fini. D'autre part, x étant algébrique sur K l'est sur $K(T)$. Il en résulte que $[K(S) : K(T)]$ est fini. Cela conclut, car $[K(S) : K] = [K(S) : K(T)][K(T) : K]$. \square

Théorème B.42. Soit $L \subseteq L$ une extension. Les conditions suivantes sont équivalentes :

1. L'extension $K \subseteq L$ est finie.
2. Il existe $x_1, \dots, x_n \in L$ algébriques sur K tels que $L = K[x_1, \dots, x_n]$.
3. Il existe $x_1, \dots, x_n \in L$ algébriques sur K tels que $L = K(x_1, \dots, x_n)$.

Démonstration. Montrons (1) \Rightarrow (2). Si l'extension est finie, il existe $n \in \mathbb{N}$ et x_1, \dots, x_n dans L tels que $L = Kx_1 + \dots + Kx_n$. D'où $L = K[x_1, \dots, x_n]$. D'autre part, les x_i sont algébriques sur K .

Montrons (2) \Rightarrow (3). Si x_r , pour $2 \leq r \leq n$, est algébrique sur K , il l'est sur $K(x_1, \dots, x_{r-1})$, et nous avons donc $K(x_1, \dots, x_r) = K(x_1, \dots, x_{r-1})[x_r]$. D'où $K(x_1, \dots, x_n) = K[x_1, \dots, x_n]$.

L'implication (3) \Rightarrow (1) est plutôt claire. \square

B.6 Extensions de décomposition

Groupe de Galois

Nous notons $\text{Aut}(K)$ l'ensemble des automorphismes du corps K , c'est un sous-groupe du groupe des bijections de K , la loi considérée étant la composition des applications.

Si G est un sous-groupe de $\text{Aut}(K)$, nous notons K^G l'ensemble des $x \in L$ tels que $\sigma(x) = x$ pour tout $\sigma \in G$. Il est immédiat de vérifier que K^G est un sous-corps de K , appelé le *corps des invariants* de G dans K .

Définition B.43 (K -homomorphisme). Soient $K \subseteq L$ et $K \subseteq M$ deux extensions d'un même corps K . Nous appelons K -homomorphisme de L dans M tout homomorphisme σ du corps L dans le corps M tel que $\sigma(x) = x$ pour tout $x \in K$. L'ensemble des K -homomorphismes de L dans M est noté $\text{Hom}_K(L, M)$.

Apportons quelques précisions. Soient $K \subseteq L$ et $K \subseteq M$ comme dans la définition.

1. Dire que $\sigma \in \text{Hom}_K(L, M)$ signifie que σ est un homomorphisme de la K -algèbre L dans la K -algèbre M .

2. Un élément de $\text{Hom}_K(L, M)$ est injectif. S'il est surjectif (donc bijectif), il est dit un *K-isomorphisme* de L sur M . L'ensemble des *K-isomorphismes* de L dans M est noté $\text{Isom}_K(L, M)$.
3. Un élément de $\text{Hom}_K(L, L)$ est appelé un *K-endomorphisme* de L , et un élément de $\text{Isom}_K(L, L)$ est appelé un *K-automorphisme* de L . Il est clair que $\text{Isom}_K(L, L)$ est un sous-groupe de $\text{Aut}(L)$, où la loi est la composition des applications.
4. Si l'extension $K \subseteq L$ est finie, nous avons $\text{Hom}_K(L, L) = \text{Isom}_K(L, L)$. Ce n'est plus nécessairement le cas si $[L : K]$ est infini. Par exemple, si $L = K(X)$ (corps des fractions à une indéterminée sur K), l'application $L \rightarrow L, F(X) \mapsto F(X^2)$ est un élément de $\text{Hom}_K(L, L) \setminus \text{Isom}_K(L, L)$.

Définition B.44 (Groupe de Galois). Soit $K \subseteq L$ un extension. Le groupe $\text{Isom}_K(L, L)$ est appelé le *groupe de Galois* de L sur K , et est noté $\text{Gal}(L/K)$.

Proposition B.45. Soient K un corps et P son sous-corps premiers. Les groupes $\text{Aut}(K)$ et $\text{Gal}(K/P)$ sont égaux.

Soient $K \subseteq L$ une extension de corps, \mathcal{A} une K -algèbre, et $\mathcal{L}_K(\mathcal{A}, L)$ l'ensemble des application K -linéaire de \mathcal{A} dans L . On vérifie facilement que l'on munit $\mathcal{L}_K(\mathcal{A}, L)$ d'une structure de L -espace vectoriel en posant :

$$(f + g)(x) = f(x) + g(x) \quad \text{et} \quad (\lambda f)(x) = \lambda f(x),$$

pour $f, g \in \mathcal{L}_K(\mathcal{A}, L)$, $\lambda \in L$, et $x \in \mathcal{A}$.

Théorème B.46. Soient $K \subseteq L$ une extension de corps, \mathcal{A} une K -algèbre, et \mathcal{H} l'ensemble des homomorphismes de K -algèbres de \mathcal{A} dans L . Alors \mathcal{H} est une partie libre du L -espace vectoriel $\mathcal{L}_K(\mathcal{A}, L)$.

Démonstration. Soit (u_1, \dots, u_n) une famille d'éléments deux à deux distincts de \mathcal{H} . Pour tout i , on a $u_i(1_{\mathcal{A}}) = 1_K$, donc $u_i \neq 0$.

Montrons que la famille précédente est libre sur L . C'est clair si $n = 1$, d'après ce qui précède. Supposons $n \geq 2$, et raisonnons par récurrence sur l'entier n . Soient $\lambda_1, \dots, \lambda_n \in L$ tels que

$$\lambda_1 u_1 + \dots + \lambda_n u_n = 0.$$

Si $x, y \in \mathcal{A}$, on a :

$$0 = \sum_{i=1}^n \lambda_i u_i(xy) - u_n(x) \sum_{i=1}^n \lambda_i u_i(y) = \sum_{i=1}^{n-1} \lambda_i (u_i(x) - u_n(x)) u_i(y).$$

On en déduit que, pour tout $x \in \mathcal{A}$,

$$\sum_{i=1}^{n-1} \lambda_i (u_i(x) - u_n(x)) u_i = 0$$

D'après l'hypothèse de récurrence, on obtient $\lambda_i (u_i(x) - u_n(x)) = 0$ pour tout $1 \leq i \leq n-1$ et pour tout $x \in \mathcal{A}$. Les u_i étant distincts, il vient $\lambda_i = 0$ si i est distinct de n . Alors $\lambda_n u_n = 0$, puis $\lambda_n = 0$, car $u_n \neq 0$. \square

B.7 Corps finis

Ne pas hésiter à se référer au Cours d'Algèbre de Michel Demazure [3].

Proposition B.47. *Soit K un corps de caractéristique $p > 0$ et notons Frob l'endomorphisme de Frobenius de K . Si K est fini alors Frob est un automorphisme de K , et si $K = \mathbb{F}_p$ alors $\text{Frob} = \text{id}_{\mathbb{F}_p}$.*

Démonstration. Par la proposition B.13, p est un nombre premier et la proposition B.15 dit que Frob est un endomorphisme de K . Comme tout morphisme de corps est injectif, si K est fini alors Frob est un automorphisme de K . Si $K = \mathbb{F}_p$, le petit théorème de Fermat donne $x^{p-1} = 1$ pour tout $x \in \mathbb{F}_p^\times$, donc $x^p = x$ quel que soit $x \in \mathbb{F}_p$. \square

C La théorie de Kummer

D La théorie d'Artin-Schreier

E L'étude ad hoc du polynôme $\nu(x)$

Soit $h \geq 1$ et soit t un élément algébrique de degré h dans $\text{GF}(p^h)/\text{GF}(p)$. Notons $\mu(x) \in \text{GF}(p)[x]$ le polynôme minimal de t . Étudions le polynôme $\nu(x)$ défini par :

$$\nu(x) := x^{p^j} + ax + b \in \text{GF}(p)[x],$$

où $a \in \text{GF}(p)^\times$, $b \in \text{GF}(p)$ et $0 \leq j \leq h-1$.

Supposons que t est racine de $\nu(x)$, i.e. $\mu(x)$ divise $\nu(x)$, et que $h \leq p$. Cela implique immédiatement que $j \geq 1$.

Si $b = 0$:

Alors $\nu(t) = t(t^{p^j-1} + a) = 0$, à savoir $t^{p^j-1} + a = 0$, et t est une racine $p^j - 1$ de $-a$.

F Le paradoxe des anniversaires

Le “paradoxe des anniversaires”, comme expliqué dans [4, sous-section 14.1], vient de l'application suivante : dans un ensemble aléatoire de 23 personnes ou plus, la probabilité que deux individus partagent le même anniversaire est supérieure à 0,5.

Théorème F.1. *Soit S un ensemble à n éléments. Si les éléments sont tirés de façon uniformément aléatoire de S , alors le nombre attendu d'éléments à tirer avant d'avoir un tirage identique à un précédent est moins de $\sqrt{\pi n/2} + 2$.*

Démonstration. Soit X la variable aléatoire du nombre d'éléments choisis dans S (uniformément aléatoire) avant qu'un élément soit sélectionné deux fois. Après que ℓ éléments distincts aient été choisis, la probabilité que le suivant soit aussi

distinct des précédents est $(1 - \ell/n)$. D'où, la probabilité $\mathbb{P}(X > \ell)$ est donnée par :

$$\mathbb{P}_{n,\ell} = 1 \left(1 - \frac{1}{n}\right) \left(1 - \frac{2}{n}\right) \cdots \left(1 - \frac{\ell-1}{n}\right).$$

Remarquons que $\mathbb{P}_{n,\ell} = 0$ quand $\ell \geq n$. Maintenant, servons nous du fait que $1 - x \leq e^{-x}$ pour $x \geq 0$:

$$\mathbb{P}_{n,\ell} \leq e^{-1/n} e^{-2/n} \cdots e^{-(\ell-1)/n} = e^{-\sum_{j=0}^{\ell-1} j/n} = e^{-(1/2)(\ell-1)\ell/n} \leq e^{-(\ell-1)^2/(2n)}$$

Par définition, l'espérance mathématique de X est :

$$\begin{aligned} \mathbb{E}(X) &= \sum_{\ell=1}^{\infty} \ell \mathbb{P}(X = \ell) \\ &= \sum_{\ell=1}^{\infty} \ell (\mathbb{P}(X > \ell-1) - \mathbb{P}(X > \ell)) \\ &= \sum_{\ell=0}^{\infty} (\ell+1 - \ell) \mathbb{P}(X > \ell) \\ &= \sum_{\ell=0}^{\infty} \mathbb{P}(X > \ell) \\ &\leq 1 + \sum_{\ell=0}^{\infty} e^{-(\ell-1)^2/(2n)} \end{aligned}$$

Nous estimons cette dernière série par l'intégrale :

$$1 + \int_0^{\infty} e^{-x^2/(2n)} dx.$$

Comme $x \mapsto e^{-x^2/(2n)}$ est décroissante et prend ses valeurs dans $[0, 1]$, la différence entre la valeur de la série et la valeur de l'intégrale est au plus 1. Le changement de variable $u := x/\sqrt{2n}$ donne :

$$1 + \int_0^{\infty} e^{-x^2/(2n)} dx = 1 + \sqrt{2n} \int_0^{\infty} e^{-u^2} du = 1 + \sqrt{2n} \frac{\sqrt{\pi}}{2}.$$

D'où la valeur :

$$\mathbb{E}(X) \leq \sqrt{\frac{\pi n}{2}} + 2.$$

□

Références

- [1] S. Bai and R. P. Brent. **On the Efficiency of Pollard's Rho Method for Discrete Logarithms**. In *Conference in Research and Practice in Information Technology*. Departement of Computer Science & Center for Mathematics and its Applications, Australian National University, 2008.
- [2] B. Chor and R. L. Rivest. **A Knapsack-Type Public Key Cryptosystem Based on Arithmetic in Finite Fields**. *IEEE Transactions on Information Theory*, 1988.
- [3] M. Demazure. *Cours d'Algèbre*. Nouvelle bibliothèque mathématique. Cassini, 2008.
- [4] S. D. Galbraith. *Mathematics of Public Key Cryptography*. Cambridge University Press, 2012.
- [5] K. Huber. **Specialised Attack on Chor-Rivest Public Key Cryptosystem**. *Journal of Cryptology*, 2000.
- [6] S. C. Pohlig and M. E. Hellman. **An Improved Algorithm for Computing Logarithms over $\text{GF}(p)$ and Its Cryptographic Significance**. *IEEE Transactions on Information Theory*, 1978.
- [7] J. M. Pollard. **Monte Carlo Methods for Index Computation (mod p)**. *Mathematics of Computation*, 1978.
- [8] P. Tauvel. *Corps commutatifs et théorie de Galois*. Mathématiques en devenir. Calvage & Mounet, 2008.
- [9] E. Teske. **On Random Walks for Pollard's Rho Method**. *Mathematics of Computation*, 2001.
- [10] S. Vaudenay. **Cryptanalysis of the Chor-Rivest Cryptosystem**. *Journal of Cryptology*, 2000.