PROBLÈME DU LOGARITHME DISCRET APPLIQUÉ À LA CRYPTANALYSE SUR COURBES ELLIPTIQUES : ALGORITHME MOV

AUDE LE GLUHER
ENCADRÉE PAR GUÉNAËL RENAULT

22 août 2015

Table des matières

Introduction

Autrefois apanage de l'armée et des services secrets, la cryptographie protège désormais les documents des industriels voire des particuliers. L'un de ses buts, comme son nom l'indique (crypto- signifie cacher et -graphie est un suffixe référant à l'écriture), est de stocker ou d'échanger des informations avec un groupe restreint de personnes tout en empêchant à d'autres d'y avoir accès.

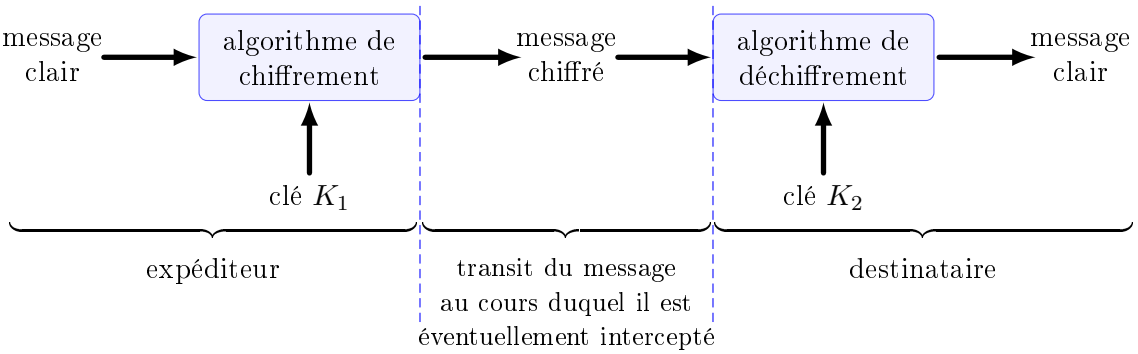
Ce travail se focalisera quant à lui sur la cryptanalyse, discipline dont l'objectif est de retrouver le sens d'un message crypté sans en être le destinataire. Plus particulièrement, délaissant les méthodes physiques ou probabilistes, il se concentrera sur le problème mathématique dit du logarithme discret. Ce dernier est en effet sous-jacent à la cryptanalyse de nombreux cryptosystèmes et en particulier à la cryptanalyse sur courbes elliptiques.

La plus grande partie de ce travail a été l'appréhension de concepts et la compréhension de démonstrations permettant de soutenir le pan théorique des algorithmes étudiés. Certains résultats ne sont pas démontrés explicitement ici mais toutes les preuves ont été étudiées et comprises sauf mention contraire. Une partie implémentation complète ce rapport. Les codes et quelques résultats se trouvent en annexe.

Vocabulaire cryptographique

Dans tout ce travail, un message ou un texte sera qualifié de *clair* avant traitement cryptographique et est dit *chiffré* après traitement cryptographique. Un *chiffre* est un couple formé d'un algorithme - c'est-à-dire un procédé de cryptage général - et d'une *clé*, paramètre spécifiant un algorithme de chiffrement. Ce dernier est toujours supposé connu de tous ; selon le principe de Kerckhoffs : la sécurité d'un système de cryptage ne repose que sur le secret de la clé.

Traditionnellement, l'expéditeur de message chiffré se nomme Alice, le destinataire Bob et l'adversaire qui souhaite déchiffrer le message Eve. Si Alice et Bob souhaitent échanger un message sans que celui-ci soit compréhensible par Eve, ils procèdent de la manière suivante :



Dans la *cryptographie (à clé) symmétrique*, $K_1 = K_2$; dans la *cryptographie (à clé) asymétrique*, $K_1 \neq K_2$.

Remarque. Le gros inconvénient de la cryptographie symétrique est qu'avant d'échanger un secret, Alice et Bob doivent déjà en partager un : ils doivent s'être mis d'accord sur une

clé. C'est pourquoi les chiffres utilisés en pratique, tel RSA, sont asymétriques. Ici, nous ne travaillerons qu'avec des chiffres asymétriques.

1. Le problème du logarithme discret

1.1. **Présentation du DLP.**

Soit $(G, \cdot)$ un groupe cyclique. Soient g un générateur de G et h un élément de G. Le problème du logarithme discret (abrégé par la suite en DLP, pour discrete logarithm problem) consiste à trouver un entier x tel que $\underbrace{g \cdot g ... g \cdot g}_{x \text{ fois}} = h$. Cet entier x est appelé *logarithme de h en base g* et est noté $\log_g(h)$.

Remarques

- La façon dont est posé le problème garantit l'existence de x. C'est généralement sous cette forme qu'on le rencontre en cryptanalyse. Souvent, on trouve même le problème sous cette forme :

  Soit p un nombre premier, g un générateur de $\mathbf{Z}/p\mathbf{Z}^*$ et h un élément de $\mathbf{Z}/p\mathbf{Z}^*$. Trouver un exposant x tel que $g^x \equiv h \bmod p$.
- À l'inverse, on trouve des versions plus larges du DLP, telle :

  Soit $(G, \cdot)$ un groupe. Soient h et g deux éléments de G. Trouver, s'il existe, un entier x tel que $\underbrace{g \cdot g ... g \cdot g}_{x \text{ fois}} = h$.
- Le logarithme discret en base g est défini modulo l'ordre de g.

Exemple 1.1. Dans le groupe $(\mathbf{Z}/p\mathbf{Z}, +)$ où p est un nombre premier.

Soit g un générateur de $\mathbf{Z}/p\mathbf{Z}$, et h un de ses éléments. Trouver x tel que $x \cdot g = h \bmod p$ est simple : il suffit de calculer l'inverse de g modulo p via l'algorithme d'Euclide. Le DLP n'est donc pas toujours un problème difficile.

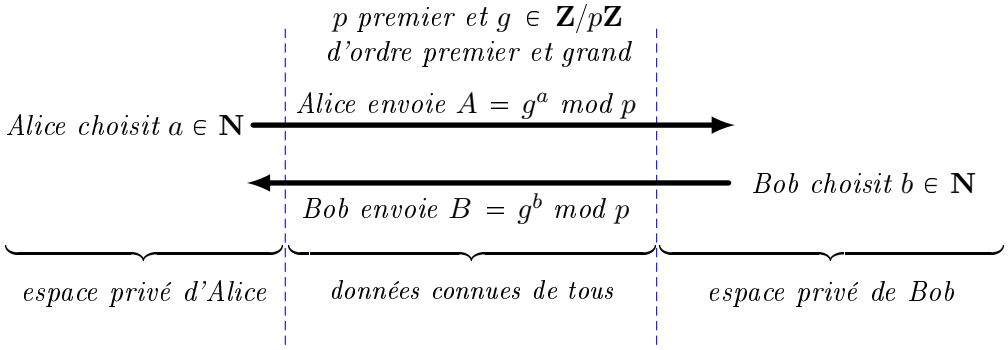
Remarque. Borne haute de complexité pour le DLP

Lorsqu'on calcule un logarithme de h en base g où g est d'ordre fini égal à N, le nombre d'étapes de calcul est majoré par N fois le temps de calcul d'une opération de groupe : on calcule g, $g \cdot g$, $g^2 \cdot g$... Mais, pour des groupes d'ordre très grand, le temps de calcul est prohibitif.

1.2. **Un problème connexe : le DHP.**

Plusieurs solutions ont été trouvées pour répondre au problème de distributions des clés. L'une d'entre elles est l'échange de clés de Diffie-Hellman présenté ci-dessous.

Algorithme 1.1. *Alice et Bob souhaitent échanger une clé sans risque et sans se rencontrer. Voici une solution :*



Alice calcule alors $B^a \equiv g^{ba}$ *mod* p *et Bob calcule* $A^b \equiv g^{ab}$ *mod* p*. La clé commune est* g^{ab} *mod* p*.*

Définition 1.1. Problème de Diffie-Hellman

Pour pouvoir retrouver la clé échangée, il suffit de résoudre le problème suivant : calculer g^{ab} mod p connaissant g, p, g^a et g^b. Ce problème est nommé *problème de Diffie-Hellman* (DHP).

Remarque : Ce problème est plus facile que le DLP. En effet, si l'on sait résoudre le DLP, on peut résoudre l'équation en x suivante : $g^x \equiv g^a$ mod p donc trouver a. Le calcul de $(g^b)^a$ mod p permet alors de trouver la clé.

1.3. Algorithme baby-step giant-step.

Dans cette section, on améliore la complexité de résolution d'un DLP dans un groupe quelconque.

Algorithme 1.2. *Soit* G *un groupe. Soit* g *un élément de* G *d'ordre* $N \geqslant 2$ *et* h *un élément de* G *engendré par* g*. L'algorithme suivant résout le DLP* $g^x = h$ *en* $\mathrm{O}(\sqrt{N}\log(N))$ *étapes :*

(1) *Calculer* $n = \lfloor \sqrt{N} \rfloor + 1$

(2) *Calculer les deux listes* $\begin{cases} e, g, g^2, ..., g^n. \text{ Ce sont les pas de bébé.} \\ h, hg^{-n}, hg^{-2n}, ..., hg^{-n^2}. \text{ Ce sont les pas de géant.} \end{cases}$

(3) *Trouver un élément commun à ces deux listes;* ie *trouver* $(i, j) \in [\![0, n]\!]$ *tel que* $g^i = hg^{-nj}$*.*

(4) *Calculer* $i + nj$*. C'est une solution du DLP.*

Démonstration : Montrons d'abord la correction de cet algorithme.

Pour ce faire, il suffit de montrer qu'il existe toujours un élément commun à la liste des pas de bébé et à celle des pas de géant. Comme h est engendré par g, il existe $x \in \mathbf{N}$ tel que $h = g^x$. Par division euclidienne, il existe $q \in \mathbf{N}$ et $r \in [\![0, n[\![$ tels que $x = nq + r$. Donc :

$$h = g^{nq+r} \tag{1}$$

Comme $x < N$ et $n > \sqrt{N}$, on a $q = \frac{x-r}{n} < \frac{N}{n} < n$. L'équation (1) peut donc en effet se réécrire :

$$g^r = hg^{-nq} \text{ avec } 0 \leqslant r < n \text{ et } 0 \leqslant q < n$$

Montrons maintenant la complexité avancée. Le calcul des deux listes, une fois g^{-n} calculé, requiert $2n$ opérations de groupe. Trions ensuite les deux listes (ce qui peut facilement se faire en $n\log(n)$) pour trouver la correspondance. La complexité de l'algorithme est donc en $\mathrm{O}(n\log(n))$. Comme $n \simeq \sqrt{N}$, on obtient le résultat attendu.

1.4. **Algorithme de Pohlig-Hellman et amélioration.**

On cherche toujours à résoudre le DLP $g^x = h$ dans un groupe G quelconque. Ici, on cherche à tirer parti de la connaissance d'une factorisation de l'ordre N de g.

Algorithme 1.3. *Algorithme de Pohlig-Hellman*

Soit G un groupe dans lequel on sait résoudre le DLP pour tout élément d'ordre q^e où e est entier et q est un nombre premier en $\mathrm{O}(C_{q^e})$ étapes. Soit g un élément de G d'ordre $\prod_{i=1}^{t} q_i^{e_i}$ et h un élément de G engendré par h. Alors on l'algorithme suivant résout le DLP $g^x = h$ en $\mathrm{O}(\sum_{i=1}^{t} C_{q_i^{e_i}} + \log(N))$.

(1) *Pour tout $i \in [\![1,t]\!]$, calculer $g_i = g^{\frac{N}{Q_i^{e_i}}}$ et $h_i = h^{\frac{N}{Q_i^{e_i}}}$.*

(2) *Par hypothèse, pour tout $i \in [\![1,t]\!]$, on sait calculer une solution y_i du DLP $g_i^y = h_i$.*

(3) *Grâce au théorème des restes chinois, calculer une solution du système de congruences $\mathcal{S} = \{x \equiv y_i \ mod \ q_i^{e_i}\}$. L'entier x est solution du DLP initial.*

Démonstration : L'expression de la complexité vient de l'hypothèse et de la complexité de l'algorithme donnant les entiers de Bézout.

Montrons que x est effectivement solution du DLP $g^x = h$.

Comme x est solution de $\mathcal{S}$, pour tout $i \in [\![1,t]\!]$, il existe $z_i \in \mathbf{Z}$ tel que $x = y_i + q_i^{e_i} z_i$. Un rapide calcul montre alors que pour tout $i \in [\![1,t]\!]$, $(g^x)^{\frac{N}{q_i^{e_i}}} = h^{\frac{N}{q_i^{e_i}}}$.

Ces égalités se réécrivent : pour tout $i \in [\![1,t]\!]$,

$$\frac{N}{q_i^{e_i}} \times x \equiv \frac{N}{q_i^{e_i}} \times \log_g(h) \bmod N \tag{2}$$

D'autre part, comme le plus grand diviseur commun aux entiers $\left(\frac{N}{q_i^{e_i}}\right)_{i\in[\![1,t]\!]}$ est 1, il existe des entiers $(c_i)_{i\in[\![1,t]\!]}$ tels que $\sum_{i=1}^{t} c_i \frac{N}{q_i^{e_i}} = 1$.

En multipliant les congruences (2) par c_i et en sommant toutes les congruences obtenues on aboutit à :

$$\sum_{i=1}^{t} c_i \frac{N}{q_i^{e_i}} \times x \equiv \sum_{i=1}^{t} c_i \frac{N}{q_i^{e_i}} \times \log_g(h) \bmod N$$

puis à $x \equiv \log_g(h) \bmod N$ comme attendu.

Algorithme 1.4. *Amélioration*

Soit G un groupe dans lequel on sait résoudre le DLP pour tout élément d'ordre premier q en $\mathrm{O}(C_q)$ étapes. Soit g un élément de G d'ordre q^e où q est premier et h un élément de G engendré par h. Alors l'algorithme suivant résout le DLP $g^x = h$ en $\mathrm{O}(eC_q)$.

(1) *On cherche une solution x sous la forme* $\sum_{i=0}^{e-1} x_i q^i$.

(2) *Pour tout* $i \in [\![0, e-1]\!]$, *résoudre le DLP* $\left(g^{q^{e-1}}\right)^{x_i} = \left(hg^{-x_0-x_1q-\ldots-x_{i-1}q^{i-1}}\right)^{q^{e-i-1}}$

Démonstration : L'analyse de la complexité est claire puisque $g^{q^{e-1}}$ est d'ordre premier q. La correction de cet algorithme est tout aussi claire : on calcule successivement x_0, x_1...jusque x_{i-1}.

Remarques

- L'algorithme précédent permet de calculer des DLP pour des éléments d'ordre une puissance d'un nombre premier. L'algorithme de Pohlig-Hellman permet de fusionner les résultats obtenus. En combinant les deux, on gagne en efficacité.
- Au vu des algorithmes précédents, afin que le DLP $g^x = h$ soit difficile, il faut éviter de choisir un générateur g dont l'ordre se décompose en produit de petits entiers premiers.

2. Courbes elliptiques et fonctions rationnelles

Dans cette section, $\mathbf{K}$ désigne un corps algébriquement clos de caractéristique différente de deux ou trois.

2.1. Définition d'une courbe elliptique.

Définition 2.1. On définit une courbe affine plane comme étant le lieu des points d'annulation d'un polynôme en deux variables de $\mathbf{K}[X, Y]$. De même on définit une courbe projective plane comme étant le lieu des points d'annulation d'un polynôme homogène en trois variables de $\mathbf{K}[X, Y, Z]$.

Définition 2.2. Soit $\mathcal{C}$ une courbe affine plane associée au polynôme $P \in \mathbf{K}[X, Y]$. On dit que $\mathcal{C}$ est une courbe lisse si elle admet une unique tangente en tout point.

Définition 2.3. Une courbe elliptique sur $\mathbf{K}$ est l'ensemble des points du plan projectif $\mathbf{P}^2(\mathbf{K}^3)$ dont les coordonnées homogènes $[x : y : z]$ vérifient l'équation

$$y^2z = x^3 + +axz^2 + bz^3$$

où a et b sont des éléments de $\mathbf{K}$ tels que $4a^3 + 27b^2 \neq 0$.

Remarque : Intéressons nous aux points sur la droite à l'infini d'une courbe elliptique. Soit P un tel point et $[x : y : 0]$ ses coordonnées homogènes. Ces dernières doivent vérifier l'équation $x^3 = 0$. Par intégrité de $\mathbf{K}$, on a donc $x = 0$. De plus, comme $[0 : 0 : 0]$ n'est pas un point du plan projectif, on a nécessairement $y \neq 0$. En divisant les coordonnées de P par y, on obtient finalement qu'un seul point d'une courbe elliptique se situe sur la droite à l'infini : celui de coordonnées homogènes $[0 : 1 : 0]$. D'où la "définition" suivante que l'on choisit souvent pour décrire une courbe elliptique.

Définition 2.4. Définition courante d'une courbe elliptique

On nomme courbe elliptique l'ensemble des points d'une courbe plane affine définie par un polynôme de $\mathbf{K}[X,Y]$ du type $Y^2 - X^3 - aX - b$ où a et b vérifient $4a^3 + 27b^2 \neq 0$, auquel on ajoute le point à l'infini, noté par la suite P_∞ et tel que

$$P_\infty = \bigcap_{a \in \mathbf{K}} \{X - a = 0\}$$

Remarques

- La définition de P_∞ signifie plus simplement que ce point est arbitrairement considéré comme le point d'intersection de toutes les droites verticales de $\mathbf{K}^2$.
- Soit P un point de $\mathcal{E}$ différent de P_∞ de coordonnées (x_P, y_P). On dit que P est *ordinaire* si $y_P \neq 0$ et *spécial* sinon. Comme $\mathbf{K}$ est algébriquement clos et que $\mathcal{E}$ est une courbe elliptique, il existe $(\alpha, \beta, \gamma) \in \mathbf{K}^3$ tel que $X^3 + aX + b = (X - \alpha)(X - \beta)(X - \gamma)$. Les seuls points spéciaux sont donc ceux de coordonnées $(\alpha, 0)$, $(\beta, 0)$ et $(\gamma, 0)$. La proposition 2.1 montre que α, β et γ sont en fait deux à deux distincts.
- Plutôt que de donner le polynôme décrivant une courbe elliptique $\mathcal{E}$, on préfère souvent dire qu'un point appartient à $\mathcal{E}$ si et seulement si c'est le point à l'infini ou ses coordonnées $(x, y) \in \mathbf{K}^2$ vérifient l'équation, dite "de Weierstrass" :

$$Y^2 = X^3 + aX + b$$

Proposition 2.1. *Soient a et b deux éléments de $\mathbf{K}$. Soit $\mathcal{C}$ une courbe affine plane définie par le polynôme $f = X^3 + aX + b - Y^2$. Alors, $\mathcal{C}$ est lisse si et seulement si $4a^3 + 27b^2 \neq 0$. En particulier, une courbe elliptique est lisse.*

Démonstration :

On montre d'abord que $P = X^3 + aX + b$ n'a pas de racine multiple si et seulement si $4a^3 + 27b^2 \neq 0$. En effet :

$$\begin{aligned}
& X^3 + aX + b \text{ admet une racine multiple} \\
\Leftrightarrow\ & X^3 + aX + b \text{ et } 3X^2 + a \text{ ont une racine commune} \\
\Leftrightarrow\ & \exists \alpha \in \mathbf{K} \text{ tel que } X - \alpha | X^3 + aX + b \text{ et } X - \alpha | 3X^2 + a \\
\Leftrightarrow\ & \exists \alpha \in \mathbf{K} \text{ tel que } X - \alpha | \text{pgcd}(X^3 + aX + b, 3X^2 + a) \\
\Leftrightarrow\ & (a = 0 \text{ et } \exists \alpha \text{ tel que } X - \alpha | b) \text{ ou } (a \neq 0 \text{ et } \exists \alpha \text{ tel que } X - \alpha | \tfrac{27b^2 + 4a^3}{4a^2}) \\
\Leftrightarrow\ & (a = 0 \text{ et } b = 0) \text{ ou } (a \neq 0 \text{ et } 27b^2 + 4a^3 = 0) \\
\Leftrightarrow\ & 27b^2 + 4a^3 = 0
\end{aligned}$$

Ce qui conclut la preuve par contraposée.

Alors :

$$\begin{aligned}
& \mathcal{C} \text{ est lisse} \\
\Leftrightarrow\ & \forall (x, y) \in \mathcal{C}, \text{grad}(f)(x, y) \neq 0 \\
\Leftrightarrow\ & \forall (x, y) \in \mathcal{C}, (3x^2 + a \neq 0 \text{ ou } -2y \neq 0) \\
\Leftrightarrow\ & \forall (x, y) \in \mathcal{C}, (3x^2 + a \neq 0 \text{ ou } y \neq 0)
\end{aligned}$$

Si $4a^3 + 27b^2 \neq 0$, P n'a pas de racine multiple. Donc, pour tout point (x, y) de $\mathcal{C}$, si $y = 0$ alors $3x^2 + a \neq 0$. Donc $\mathcal{C}$ est lisse vu l'équivalence précédente. Réciproquement, si

pour tout point (x, y) de $\mathcal{C}$ on a $3x^2 + a \neq 0$ ou $y \neq 0$, par l'absurde P ne peut avoir de racine multiple. Donc $4a^3 + 27b^2 \neq 0$.

Notation : Soit $\mathcal{E}$ une courbe elliptique. En cryptographie, on ne s'intéresse qu'aux points de $\mathcal{E}$ ayant des coordonnées appartenant à un corps fini $\mathbf{F}_p$ où p est premier. On désigne l'ensemble de ces points par $\mathcal{E}(\mathbf{F}_p)$.

2.2. **Fonctions polynomiales sur une courbe elliptique.**

Dans cette section et les sections suivantes de la deuxième partie, $\mathcal{E}$ est une courbe elliptique sur $\mathbf{K}$ décrite par l'équation $Y^2 = X^3 + aX + b$.

Définition 2.5. Anneau de coordonnées

L'anneau de coordonnées de la courbe elliptique $\mathcal{E}$ est l'anneau quotient $\frac{\mathbf{K}[X,Y]}{(Y^2-X^3-aX-b)}$. On le note $\mathbf{K}[\mathcal{E}]$ et tout élément de cet anneau est nommé fonction polynomiale sur $\mathcal{E}$.

Remarque : Tout élément G de $\mathbf{K}[\mathcal{E}]$ peut s'écrire sous la forme dite réduite $G(X, Y) = P(X) + YQ(X)$ avec $(P, Q) \in \mathbf{K}[X]$ en remplaçant successivement Y^2 par $X^3 + aX + b$. (Cette égalité remplace en fait un symbole d'équivalence). Cette écriture est d'ailleurs unique.

Proposition 2.2. *Soit $G \in \mathbf{K}[\mathcal{E}]\backslash\{0\}$. Alors G admet un nombre fini de zéros.*

Démonstration :

Introduisons tout d'abord quelques notions. La remarque précédente assure l'existence de $(P, Q) \in \mathbf{K}[X]$ tel que $G = P+YQ$. On appelle conjugué de G l'objet suivant : $\overline{G} = P-YQ$. On appelle alors norme de G, notée $n(G)$ le polynôme suivant :

$$n(G) = G\overline{G} = P(X)^2 - Y^2Q(X)^2 = P(X)^2 - (X^3 + aX + b)^2Q(X)^2$$

La norme de G est donc un élément de $\mathbf{K}[X]$.

Soit dès lors (α, β) un zéro de G. On a donc $P(\alpha) + \beta Q(\alpha) = 0$. Alors,

$$\begin{aligned} n(G)(\alpha) &= P(\alpha)^2 - (\alpha^3 + a\alpha + b)^2Q(\alpha)^2 \\ &= \beta^2Q(\alpha)^2 - \beta^2Q(\alpha)^2 \\ &= 0 \end{aligned}$$

Donc α est racine de $n(G)$. On en déduit que toute abscisse d'un zéro de G est un zéro de $n(G)$ qui en a un nombre fini. D'où le résultat.

Définition 2.6. Soit $G \in \mathbf{K}[\mathcal{E}]$. On appelle degré de G, noté $\deg_{\mathcal{E}}(G)$, le degré au sens habituel de la norme de G. Autrement dit, $\deg_{\mathcal{E}}(G) = \deg(n(G))$.

Par exemple, pour tout $k \in \mathbf{N}$, $\begin{cases} \deg_{\mathcal{E}}(X^k) = 2k \\ \deg_{\mathcal{E}}(Y^k) = 3k \end{cases}$

2.3. **Fonctions rationnelles sur $\mathcal{E}$.**

Lemme 2.1. *Le polynôme $P = Y^2 - X^3 - aX - b$ décrivant la courbe elliptique $\mathcal{E}$ est irréductible dans $\mathbf{K}[X, Y]$.*

Démonstration :

Comme $\mathbf{F}[X, Y]$ est isomorphe à $\mathbf{F}[X][Y]$, on peut considérer que P est un élément de $\mathbf{F}[X][Y]$. Comme P décrit une courbe elliptique, il existe trois éléments de $\mathbf{F}$ α, β et γ deux à deux distincts tels que $X^3 + aX + b = (X - \alpha)(X - \beta)(X - \gamma)$. On a :

- $(X-\alpha)|-(X^3+aX+b)$
- $(X-\alpha)^2 \nmid -(X^3+aX+b)$
- $(X-\alpha) \nmid 1$
- $(X-\alpha)$ est premier car irréductible dans l'anneau factoriel $\mathbf{F}[X]$.

Le critère d'Eisenstein est le fait que P est primitif assurent que ce polynôme est irréductible dans $\mathbf{F}[X][Y]$.

On déduit de ce résultat que $\mathbf{K}[\mathcal{E}]$ est un anneau intègre, on peut donc parler du corps des fractions de $\mathbf{K}[\mathcal{E}]$, que l'on note $\mathbf{K}(\mathcal{E})$. Un élément de $\mathbf{K}(\mathcal{E})$ est appelé fraction rationnelle sur $\mathcal{E}$.

Définition 2.7. Point régulier et pôle

Soit P un point de $\mathcal{E}$ différent de P_∞. Soit $R \in \mathbf{K}(\mathcal{E})$. Le point P est régulier pour R s'il existe un représentant $\frac{G}{H}$ de R tel que $H(P) \neq 0$. Sinon, on dit que P est un pôle de R.

Définition 2.8. Valeur d'une fonction rationnelle en un point de $\mathcal{E}$.

Soit $R \in \mathbf{K}(\mathcal{E})$ et $P \in \mathcal{E}$. La valeur de R en P est :

- $R(P)$ si P est régulier
- $+\infty$ si P est un pôle
- Si $P = P_\infty$ et $\frac{G}{H}$ est un représentant de R, notons a (resp. b) le coefficient de plus haut degré de G (resp. H). Cette valeur vaut $\begin{cases} 0 & \text{si } \deg_{\mathcal{E}}(G) < \deg_{\mathcal{E}}(H) \\ +\infty & \text{si } \deg_{\mathcal{E}}(G) > \deg_{\mathcal{E}}(H) \\ \frac{a}{b} & \text{si } \deg_{\mathcal{E}}(G) = \deg_{\mathcal{E}}(H) \end{cases}$

Proposition 2.3. *Toute fonction rationnelle sur $\mathcal{E}$ admet un nombre fini de zéros et de pôles*

Démonstration : On reprend en les adaptant les arguments de 2.2.

Remarque : Une fonction rationnelle n'ayant ni zéros ni pôles est constante.

2.4. **Uniformisantes.**

Définition 2.9. Uniformisante

Soit P un point de $\mathcal{E}$. Une uniformisante en P est une fonction rationnelle u telle que :

$$\begin{cases} u(P) = 0 \\ \forall g \in \mathbf{K}(\mathcal{E})\backslash 0, \exists d \in \mathbf{N} \text{ et } r \in \mathbf{K}(\mathcal{E}) \text{ dont } P \text{ n'est ni zéro ni pôle tel que } g = u^d r \end{cases}$$

Proposition 2.4. *L'entier d dont il est question dans la précédente définition est indépendant de l'uniformisante.*

Démonstration : Soit $P \in \mathcal{E}$ et u et v deux uniformisantes en P. Alors, il existe $(e, f) \in \mathbf{N}$ et $(r, s) \in \mathbf{K}(\mathcal{E})$ tels que P n'est ni zéro ni pôle de r ou de s tels que $u = v^e r$ et $v = u^f s$.

On a donc $u = u^{ef} s^e r$ puis $1 = u^{ef-1} s^e r$. On évalue cette dernière égalité en P. Comme $u(P) = 0$, on a nécessairement $ef - 1 = 0$. Comme e et f sont des entiers naturels, $e = f = 1$.

Soit désormais $g \in \mathbf{K}(\mathcal{E})$ dont l'ordre en P relativement à u est d. On a $g = u^d t$ où P n'est ni zéro ni pôle de t. Par le point précédent, $g = (vr)^d t = v^d r^d t$ Comme P n'est ne zéro ni pôle de $r^d t$, l'ordre de g en P relativement à v est aussi d.

Définition 2.10. Ordre d'une fonction en un point

Soit P un point de $\mathcal{E}$ et g un élément de $\mathbf{K}(\mathcal{E})$. Soit u une uniformisante en P. Alors il existe $d \in \mathbf{Z}$ et $r \in \mathbf{K}(\mathcal{E})$ dont P n'est ni zéro ni pôle tels que $g = u^d r$. L'entier d est appelé *ordre de g en P* et est noté $\mathrm{ord}_P(g)$.

Remarques

- La proposition 2.4 assure la bonne définition de l'ordre.
- Si P n'est ni un zéro ni un pôle, pour toute fonction rationnelle f, $\mathrm{ord}_P(f) = 0$.

Proposition 2.5. *Tout point P d'une courbe elliptique admet une uniformisante. Plus précisément :*

- *Si $P = (x_P, y_P)$ est un point ordinaire, $X - x_P$ est une uniformisante en P.*
- *Si P est un point spécial, Y est une uniformisante en P.*
- *Si $P = P_\infty$, $\frac{X}{Y}$ est une uniformisante en P.*

Démonstration : Étudiée et comprise.

3. Un outil pratique : les diviseurs

Dans cette section, $\mathbf{K}$ désigne un corps algébriquement clos de caractéristique différente de deux ou trois et $\mathcal{E}$ une courbe elliptique sur $\mathbf{K}$ décrite par l'équation $Y^2 = X^3 + aX + b$. Cette partie requiert la connaissance du fonctionnement de la loi + sur les points de $\mathcal{E}$; loi décrite aux paragraphes 4.1.1 et 4.1.2.

3.1. Définitions.

Définition 3.1. Diviseur

Un diviseur de $\mathcal{E}$ est une application de $\mathcal{E}$ dans $\mathbf{Z}$ prenant un nombre fini de valeurs non nulles. On note $\mathrm{Div}(\mathcal{E})$ l'ensemble des diviseurs de $\mathcal{E}$.

On définit une loi de composition interne, notée +, sur $\mathrm{Div}(\mathcal{E})$ qui en fait clairement un groupe commutatif. Si D et D' sont deux diviseurs de $\mathcal{E}$, on a

$$D + D' : \left\{ \begin{array}{ccc} \mathcal{E} & \longrightarrow & \mathbf{Z} \\ P & \longmapsto & D(P) + D(P') \end{array} \right.$$

Notation : Soit $D : \left\{ \begin{array}{ccc} \mathcal{E} & \longrightarrow & \mathbf{Z} \\ P & \longmapsto & a_P \end{array} \right.$ un diviseur de $\mathcal{E}$.

On dénote D par la somme formelle $\sum_{P \in \mathcal{E}} a_P(P)$.

Définition 3.2. Somme et degré d'un diviseur

Soit $D = \sum_{P \in \mathcal{E}} a_P(P)$ un diviseur de $\mathcal{E}$.

Le degré de D est l'entier $\sum_{P \in \mathcal{E}} a_P$, noté $\deg(D)$.

La fonction $\deg : \mathcal{E} \to \mathbf{Z}$ est un morphisme de groupes dont le noyau est noté $\mathrm{Div}^0(\mathcal{E})$.

La somme de D est le point de $\mathcal{E}$ $\sum_{P \in \mathcal{E}} a_P P$, notée $\mathrm{som}(D)$.

Remarque : Tous ces objets sont bien définis puisque toutes les sommes sont finies par définition d'un diviseur.

3.2. **Diviseur d'une fonction.**

Définition 3.3. Soit f une fonction rationnelle non nulle. Par définition le diviseur de f est $\mathrm{Div}(f) = \sum_{P \in \mathcal{E}} \mathrm{ord}_P(f)(P)$.

Remarque : Cet objet est bien un diviseur au sens précédent car f a un nombre fini de zéros et de pôles par la proposition 2.3.

Proposition 3.1. *Pour toutes fonctions rationnelles r et s,* $\mathrm{Div}(rs) = \mathrm{Div}(r) + \mathrm{Div}(s)$ *et* $\mathrm{Div}(\frac{r}{s}) = \mathrm{Div}(r) - \mathrm{Div}(s)$.

Démonstration : Claire

Proposition 3.2. *Deux fonctions non nulles ont même diviseur si et seulement si elles sont proportionnelles.*

Démonstration : Si f et g sont proportionnelles elles ont mêmes zéros et mêmes pôles donc ont le même diviseur. Réciproquement, si elles ont même diviseur alors $\mathrm{Div}(\frac{f}{g}) = \mathrm{Div}(f) - \mathrm{Div}(g) = 0$. La fonction $\frac{f}{g}$ n'a donc ni zéro ni pôle : elle est constante.

Définition 3.4. Diviseur principal
Un diviseur D est dit principal s'il existe une fonction rationnelle f telle que $D = \mathrm{Div}(f)$. On note $\mathrm{Princ}(\mathcal{E})$ l'ensemble des diviseurs principaux.

Définition 3.5. Relation d'équivalence sur $\mathrm{Div}(\mathcal{E})$
On introduit une relation d'équivalence $\sim$ sur $\mathrm{Div}(\mathcal{E})$. Soit D_1 et D_2 deux diviseurs.

$$D_1 \sim D_2 \Leftrightarrow D_1 - D_2 \text{ est un diviseur principal}$$

Théorème 3.1. *Caractérisation des diviseurs principaux*
Un diviseur est principal si et seulement si son degré est nul et sa somme vaut P_∞.

Démonstration : Admise. Elle repose sur des propriétés de l'ordre. Plusieurs parties de la preuve ont été travaillées.

Lemme 3.1. *Soit P_1 et P_2 deux points de $\mathcal{E}$. Alors il existe $g \in \mathbf{K}(\mathcal{E})$ tel que* $(P_1) + (P_2) = (P_1 + P_2) + (P_\infty) + \mathrm{Div}(g)$.

Démonstration : Procédons par cas.
Si P_1 ou P_2 est le point à l'infini, la fonction 1 convient puisque $\mathrm{Div}(g) = 0$.
Si $P_1 = -P_2$, notons x_1 l'abscisse commune de P_1 et P_2. On a alors $\mathrm{Div}(X - x_1) = (P_1) + (P_2) - 2(P_\infty)$. Donc $(P_1) + (P_2) = \underbrace{2(P_\infty)}_{=(P_1+P_2)} + \mathrm{Div}(g)$ avec $g = X - x_1$.
Si $P_1 \neq P_2$, considérons la droite $\mathcal{D}$ passant par P_1 et P_2 décrite par le polynôme $\alpha X + \beta Y + \gamma$. Notons $P_3 = (x_3, y_3)$ le troisième point d'intersection de $\mathcal{E}$ et $\mathcal{D}$.
Comme $P_1 \neq P_2$, β est différent de 0 donc $\deg_{\mathcal{E}}(\alpha X + \beta Y + \gamma) = 3$ et donc P_∞ est d'ordre 3.

On a donc $\mathrm{Div}(\alpha X+\beta Y+\gamma) = (P_1)+(P_2)+(P_3)-3(P_\infty)$. D'autre part, $\mathrm{Div}(X-x_3) = (P_3)+(-P_3)-2(P_\infty)$. Donc :

$$\mathrm{Div}\left(\frac{\alpha X+\beta Y+\gamma}{X-x_3}\right) = (P_1)+(P_2)-(-P_3)-(P_\infty)$$

Comme $P_1+P_2=-P_3$, en prenant $g=\frac{\alpha X+\beta Y+\gamma}{X-x_3}$, on obtient le résultat attendu.

Si $P_1=P_2$, on reprend le raisonnement précédent avec la tangente à $\mathcal{E}$ en P_1.

Corollaire 3.1. *Soit D un diviseur de degré nul. Alors il existe $P\in\mathbf{K}(\mathcal{E})$ tel que $D\sim(P)-(P_\infty)$.*

Démonstration : Par le lemme 3.1, en regroupant les termes de D ayant des coefficients de même signe, il est possible de remplacer deux points par leur somme quitte à rajouter un multiple de P_∞ ou le diviseur d'une fonction. On aboutit donc soit à $D\sim(P)-(Q)+n(P_\infty)$ soit à $D\sim(P)+nP_\infty$ soit à $D\sim-(P)+nP_\infty$ avec $n\in\mathbf{Z}$.

Dans le premier cas, comme $\deg(D)=0$, $n=0$ aussi.
Comme $\mathrm{Div}(X-x_Q)=(Q)+(-Q)-2P_\infty$, on a :

$$\begin{aligned} D &\sim D+\mathrm{Div}(X-x_Q)\\ &\sim (P)+(-Q)-2(P_\infty)\\ &\sim (P-Q)-(P_\infty) \text{ par le lemme 3.1}\end{aligned}$$

Dans le deuxième cas, comme $\deg(D)=0$, on a $n=-1$ d'où immédiatement le résultat.

Dans le dernier cas, comme $\deg(D)=0$, on a $n=1$.
Comme $\mathrm{Div}(X-x_P)=(P)+(-P)-2P_\infty$, on a :

$$\begin{aligned} D &\sim D+\mathrm{Div}(X-x_P)\\ &\sim (-P)-(P_\infty)\end{aligned}$$

4. Cryptographie elliptique

Dans cette section, $\mathbf{K}$ désigne un corps algébriquement clos de caractéristique différente de deux ou trois et $\mathcal{E}$ une courbe elliptique sur $\mathbf{K}$ décrite par l'équation $Y^2=X^3+aX+b$.

4.1. Loi de groupe sur une courbe elliptique.

On définit une loi interne sur $\mathcal{E}$, notée + qui fait de $(\mathcal{E},+)$ un groupe commutatif. Les illustrations s'appuient sur la courbe d'équation $Y^2=X^3-3X+3$.

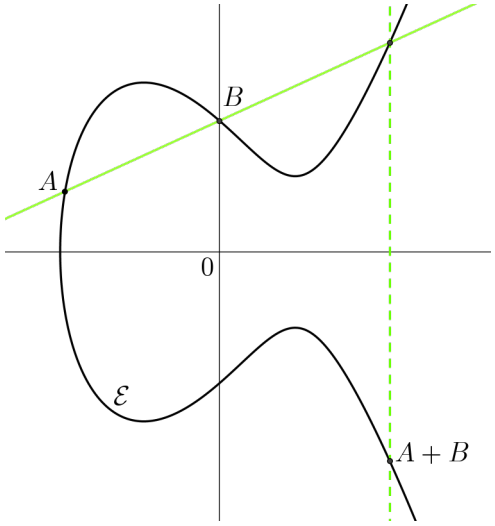
4.1.1. *Considérations géométriques.*

Soit A et B deux points de $\mathcal{E}$. On note $\mathcal{D}$ la droite passant par A et B, en considérant que cette droite est la tangente à $\mathcal{E}$ en A si $A=B$. On peut toujours considérer cette droite :

- si $A\neq B$, il existe une unique droite projective passant par A et B
- si $A=B$, la tangente à $\mathcal{E}$ en A existe bien car $\mathcal{E}$ est lisse par définition

Comme **K** est algébriquement clos (au besoin, on se place sur la clôture algébrique de **K**) on peut utiliser le théorème de Bézout. Ce dernier assure alors que $\mathcal{E}$ et $\mathcal{D}$, de degrés respectifs 3 et 1, s'intersectent en exactement 3 points.

Deux d'entre eux sont A et B. On définit la somme de A et B comme étant le symétrique du troisième par rapport à l'axe des abscisses. La figure ci-contre illustre le propos.



Remarque : L'illustration n'est pas réaliste - puisque $\mathcal{E}$ est un objet du plan projectif - mais elle permet de comprendre géométriquement la situation.

4.1.2. *Algorithme d'addition de deux points.*

Soit A et B deux points de $\mathcal{E}$. On cherche à calculer les coordonnées de $A+B=C$. Bien qu'il faille se placer dans le plan projectif pour assurer l'existence de C on calcule ici les coordonnées affines de C plutôt que ses coordonnées projectives : il suffit de traiter le [1] cas particulier du point à l'infini à part. Les coordonnées de C sont :

Si $A = P_\infty$ alors $C = B$.
Si $B = P_\infty$ alors $C = A$.

Sinon, on peut noter (x_A, y_A) les coordonnées de A et (x_B, y_B) celles de B.
Si $x_A = x_B$ et $y_A = -y_B$ (A et B sont opposés), alors $C = P_\infty$.
Sinon, on peut calculer le coefficient directeur λ de la droite joignant A et B puis les coordonnées (x_C, y_C) de C. On obtient :

$$\lambda = \begin{cases} \frac{y_B - y_A}{x_B - x_A} & \text{si } A \neq B \\ \frac{3x_A^2 + a}{2y_A} & \text{si } A = B \end{cases}$$

$$\text{Puis } \begin{cases} x_C & \lambda^2 - x_A - x_B \\ y_C & \lambda(x_A - x_C) - y_A \end{cases}$$

Remarque : Soit $P \in \mathcal{E}$ et $n \in \mathbf{N}$. Pour chiffrer un message via une courbe elliptique, il est nécessaire, comme on le verra dans la partie 4.2, de calculer rapidement nP. L'algorithme consistant à calculer P, $2P$, $3P$... nécessite de faire n additions dans $(\mathcal{E}, +)$ ce qui n'est pas efficace. D'où l'algorithme suivant.

1. Il n'y a en effet qu'un seul point à l'infini sur $\mathcal{E}$ comme vu dans la partie 2.1

Algorithme 4.1. *Exponentiation rapide*

Soit $P \in \mathcal{E}$ et $n \in \mathbf{N}$. L'algorithme double-and-add *permet de calculer nP et reprend l'algorithme d'exponentiation rapide classique avec la loi $+$ définie précédemment :*

$$double\_and\_add(n,P) = \begin{cases} Si\ n = 0\ renvoyer\ P_\infty \\ Si\ n \equiv 0\ mod\ 2\ calculer\ Q = \frac{n}{2}P\ puis\ renvoyer\ Q+Q \\ Si\ n \equiv 1\ mod\ 2\ calculer\ Q = \frac{n}{2}P\ où\ la\ division\ précédente \\ \ est\ euclidienne\ puis\ renvoyer\ Q+Q+P \end{cases}$$

Complexité : La complexité de cet algorithme est en $\Theta(\log_2(n))$ puisqu'on fait une addition par chiffre 1 dans l'écriture binaire de n. On peut même améliorer cette complexité. En effet, calculer l'opposé d'un point est très facile (l'opposé de P_∞ est P_∞ et celui de (x_P, y_P) est $(x_P, -y_P)$). On peut donc décider d'écrire n sous forme non adjacente, écriture qui compte en moyenne plus de zéros que l'écriture binaire, puis de procéder par succession d'additions ou soustractions et de doublements.

4.1.3. *Cette loi munit $\mathcal{E}$ d'une structure de groupe commutatif.*

Théorème 4.1. *$(\mathcal{E}, +)$ est un groupe commutatif.*

Démontration : On démontre ici tout les axiomes sauf l'associativité.

La loi est interne par le théorème de Bézout.

P_∞ est un neutre pour $+$.

Soit $P \in \mathcal{E}$. Si $P = P_\infty$ alors P est son propre opposé. Sinon, on peut écrire les coordonnées de P sous la forme (x_P, y_P) et le point $\tilde{P}$ de coordonnées $(x_P, -y_P)$ est l'opposé de P. En effet la droite $(P\tilde{P})$ est verticale donc intersecte $\mathcal{E}$ en P_∞ dont l'opposé est aussi P_∞. Donc $P + \tilde{P} = P_\infty$.

Enfin, la commutativité est claire puisque la droite passant par deux points A et B de $\mathcal{E}$ est la même que celle passant par B et A.

La suite de cette section est consacrée à la preuve de l'associativité de la loi.

Lemme 4.1. *Soit $A = \begin{pmatrix} a_1 & b_1 \\ a_2 & b_2 \\ a_3 & b_3 \\ a_4 & b_4 \end{pmatrix}$ une matrice de $\mathbf{M}_{4,2}(\mathbf{K})$ dont les lignes sont deux à deux indépendantes. Alors, pour tous polynômes P et Q premiers entre eux de $\mathbf{K}[X]$ tels qu'il existe quatre polynômes U_1, U_2, U_3 et U_4 de $\mathbf{K}[X]$ tel que :*

$$\begin{cases} a_1P + b_1Q = U_1^2 \\ a_2P + b_2Q = U_2^2 \\ a_3P + b_3Q = U_3^2 \\ a_4P + b_4Q = U_4^2 \end{cases}$$

on a : P et Q sont constants.

Démonstration : Procédons par l'absurde.

On choisit dès lors P et Q vérifiant les hypothèses et dont l'un des deux n'est pas constant ; il est loisible de supposer $M = \max(\deg(P), \deg(Q)) > 0$ minimal.

Le but est de construire une matrice B vérifiant les mêmes hypothèses que A et telle que les polynômes U_1 et U_2 donnés par hypothèse jouent le rôle de P et Q avec la matrice B. Les deux égalités $a_1P + b_1Q = U_1^2$ et $a_2P + b_2Q = U_2^2$ assurant que $\max(\deg(U_1), \deg(U_2)) \leqslant \frac{1}{2}M$, la minimalité de M sera contredite.

- Remarquons que pour tout couple (i, j) de $[\![1, 4]\!]$ tel que $i \neq j$, U_i et U_j n'ont pas de racine commune. Sinon il existerait $s \in \mathbf{K}$ tel que $U_i(x) = U_j(x) = 0$. Alors, les égalités $a_iP(x) + b_iQ(x) = 0$ et $a_jP(x) + b_jQ(x) = 0$ assurent l'existence d'une racine commune à P et Q ce qui contredit $P \wedge Q = 1$.
- Comme la famille $\{(a_1, b_1), (a_2, b_2)\}$ est libre dans $\mathbf{K}^2$, c'en est une base. Donc il existe $(\tilde{\alpha}, \tilde{\beta}, \tilde{\gamma}, \tilde{\delta}) \in \mathbf{K}^4$ tel que

$$\begin{cases} (a_3, b_3) = \tilde{\alpha}(a_1, b_1) + \tilde{\beta}(a_2, b_2) \\ (a_4, b_4) = \tilde{\gamma}(a_1, b_1) + \tilde{\delta}(a_2, b_2) \end{cases}$$

  Comme $\mathbf{K}$ est algébriquement clos, il existe $(\alpha, \beta) \in \mathbf{K}^2$ tel que $\alpha^2 = \tilde{\alpha}$ et $\beta = -\tilde{\beta}$. Un rapide calcule montre alors que $U_3^2 = (\alpha U_1 + \beta U_2)(\alpha U_1 - \beta U_2)$. De même, il existe $(\gamma, \delta) \in \mathbf{K}^2$ tel que $U_4 = (\gamma U_1 + \delta U_2)(\gamma U_1 - \delta U_2)$.

  Le premier point garantit que α, β, γ et δ sont différents de 0.
- Notons $B = \begin{pmatrix} \alpha & \beta \\ \alpha & -\beta \\ \gamma & \delta \\ \gamma & -\delta \end{pmatrix}$. L'hypothèse de non constance de P ou Q ainsi que l'absence de racine commune à U_3 et U_4 permet de montrer que les lignes de B sont deux à deux indépendantes.
- Les polynômes $\alpha U_1 + \beta U_2$ et $\alpha U_1 - \beta U_2$ n'ont pas de racine commune ; sinon, U_1 et U_2 ont une racine commune ce qui contredit le premier point. Cette observation et le fait que $U_3^2 = (\alpha U_1 + \beta U_2)(\alpha U_1 - \beta U_2)$ assurent que $\alpha U_1 + \beta U_2$ et $\alpha U_1 - \beta U_2$ sont des carrés. De même, $(\gamma U_1 + \delta U_2)$ et $(\gamma U_1 - \delta U_2)$ sont des carrés.
- Conclusion : la matrice B et les polynômes U_1 et U_2 vérifient les hypothèses du lemme 4.1 et invalident la minimalité de M.

Lemme 4.2. *On rappelle que $\mathcal{E}$ est une courbe elliptique décrite par l'équation $Y^2 = X^3 + aX + b$. Soit T une indéterminée. Alors il n'existe pas de fonctions rationnelles R et S de $\mathbf{K}[T]$ non constantes et telles que $S(T)^2 = R(T)^3 + aR(T) + b$.*

Démonstration : Supposons par l'absurde qu'il existe $(P_1, P_2) \in \mathbf{K}[X]$ (resp. (Q_1, Q_2)) n'ayant pas de racine commune et tel que $R(T) = \frac{P_1(T)}{P_2(T)}$ est non constant (resp. $S(T) = \frac{Q_1(T)}{Q_2(T)}$ est non constant). On aurait alors

$$Q_1(T)^2 P_2(T)^3 = (P_1(T)^3 + aP_1(T)P_2(T)^2 + bP_2(T)^3)Q_2(T)^2 \tag{3}$$

Comme Q_1 et Q_2 n'ont pas de racine commune et comme $\mathbf{K}$ est algébriquement clos, $Q_1^2 \wedge Q_2^2 = 1$. Comme $Q_2^2 | Q_1^2 P_2^3$, par le théorème de Gauss, $Q_2^2 | P_2^3$. On montre de même que $P_2^3 | Q_2^2$. Les polynômes P_2^3 et Q_2^2 sont donc associés. On peut les supposer égaux et simplifier l'égalité (3) en

$$Q_1^2 = P_1^3 + aP_1P_2^2 + bP_2^3 \tag{4}$$

Comme $\mathbf{K}$ est algébriquement clos et que $\mathcal{E}$ est une courbe elliptique, il existe $(e_1, e_2, e_3) \in \mathbf{K}^3$ deux à deux distincts tel que $X^3 + aX + b = (X - e_1)(X - e_2)(X - e_3)$. Cette égalité associée à l'égalité (4) donne

$$Q_1^2 = (P_1 - e_1P_2)(P_1 - e_2P_2)(P_1 - e_3P_2) \tag{5}$$

Or, on montre facilement par l'absurde que pour tout $(i, j) \in [\![1, 3]\!]$ tel que $i \neq j$, $P_1 - e_iP_2$ et $P_1 - e_jP_2$ n'ont pas de racine commune. Cette observation associée au fait que $\prod_{i=1}^{3}(P_1 - e_iP_2)$ est le carré du polynôme Q_1 par (5) assure que pour tout $i \in [\![1, 3]\!]$, $P_1 - e_iP_2$ est le carré d'un polynôme.

Considérons dès lors la matrice $A = \begin{pmatrix} 1 & -e_1 \\ 1 & -e_2 \\ 1 & -e_3 \\ 0 & 1 \end{pmatrix}$ dont les lignes sont deux à deux indépendantes. Cette matrice et les polynômes P_1 et P_2 entrent dans les hypothèses du lemme 4.1. On en déduit que P_1 et P_2 sont constants puis que R l'est aussi ce qui est une contradiction.

Lemme 4.3. *Soit P et Q deux points de $\mathcal{E}$. Supposons qu'il existe $h \in \mathbf{K}(\mathcal{E})$ telle que $(P) - (Q) = \mathrm{Div}(h)$. Alors $P = Q$.*

Démonstration : Par l'absurde, $P \neq Q$.

- Remarquons d'abord que pour tout $c \in \mathbf{K}$, $h - c$ a un unique pôle simple en Q puisque Q est pôle simple de H par hypothèse.
- Soit $f \in \mathbf{K}(\mathcal{E})$.

  Si $\mathrm{ord}_Q(f) = 0$, observons $g = \prod_{R \in \mathcal{E}} (h(X, Y) - h(R))^{\mathrm{ord}_R(f)}$. Cette fonction a les mêmes zéros que f. De plus, les pôles de g sont ceux des $h - h(R)$ où R décrit $\mathcal{E}$. Le premier point assure que le seul pôle de g est Q. Enfin, ce pôle est de même ordre que dans f puisque $\mathrm{ord}_Q(g) = \sum_{R \in \mathcal{E}} \mathrm{ord}_R(f) = \deg(\mathrm{Div}(f)) = 0$ par le théorème 3.1. On en déduit que f et g ont même diviseur donc qu'elles sont proportionnelles par la proposition 3.2. Donc f est une fraction rationnelle en h.

  Si Q est zéro ou pôle de f, on applique le raisonnement précédent à $f \times h^{\mathrm{ord}_Q(f)}$ qui n'a ni zéro ni pôle en Q. On conclut à nouveau que f est une fraction rationnelle en h.
- On a montré : $\forall f \in \mathcal{E}$, f est une fraction rationnelle en h. En particulier, X et Y sont des fonctions rationnelles en h. C'est impossible par le lemme 4.2.

Théorème 4.2. *La loi + introduite sur $\mathcal{E}$ est associative.*

Démonstration : L'objectif est de montrer que $\mathcal{E}$ est en bijection avec le groupe $\frac{\mathrm{Div}^0(\mathcal{E})}{\mathrm{Princ}(\mathcal{E})}$. Pour tout $D \in \mathrm{Div}^0(\mathcal{E})$, on note $[D]$ la classe de D modulo les diviseurs principaux.

On montre que $\varphi : \left\{ \begin{array}{ccc} \mathcal{E} & \longrightarrow & \frac{\mathrm{Div}^0(\mathcal{E})}{\mathrm{Princ}(\mathcal{E})} \\ P & \longmapsto & [(P) - (P_\infty)] \end{array} \right.$ est une bijection.

Prouvons donc que : $\forall D \in \mathrm{Div}^0(\mathcal{E})$, $\exists !\ P \in \mathcal{E}$ tel que $D - (P) + (P_\infty)$ est principal.

Existence : Soit $D \in \mathrm{Div}^0(\mathcal{E})$. Comme $\deg(D) = 0$, le corollaire 3.1 assure qu'il existe $P \in \mathcal{E}$ tel que $D \sim (P) - (P_\infty)$.

Unicité : Soit P et Q deux points de $\mathcal{E}$ tels que $(P)-(P_\infty) \sim (Q)-(P_\infty)$. Alors $(P)-(Q)$ est un diviseur principal et le lemme 4.3 assure que $P = Q$.

On montre à présent que $\forall (P, Q) \in \mathcal{E}, \varphi(P + Q) = \varphi(P) + \varphi(Q)$.
Si $P = Q = P_\infty$, $\varphi(P_\infty) = [(P_\infty) - (P_\infty)] = 0 = \varphi(P_\infty) + \varphi(P_\infty)$.
Si $P = -Q$, $\varphi(P + Q) = \varphi(P_\infty) = 0$. D'autre part,

$$\begin{aligned} \varphi(P) + \varphi(Q) &= [(P) - (P_\infty)] + [(-P) - (P_\infty)] \\ &= [(P) + (-P) - 2(P_\infty)] \\ &= [(P - P) + (P_\infty) + \mathrm{Div}(g) - 2(P_\infty)] \text{ où } g \in \mathbf{K}(\mathcal{E}) \\ &= [\mathrm{Div}(g)] \\ &= 0 \end{aligned}$$

Sinon, notons L le polynôme décrivant la droite $\mathcal{D}$ passant par P et Q (éventuellement la tangente à $\mathcal{E}$ en P si $P = Q$), R le troisième point d'intersection entre $\mathcal{D}$ et $\mathcal{E}$, et V le polynôme décrivant la droite verticale passant par R. On a :

$$\left\{ \begin{array}{l} \mathrm{Div}(L) = (P) + (Q) + (R) - 3(P_\infty) \\ \mathrm{Div}(V) = (R) + (-R) - 2(P_\infty) \end{array} \right.$$

Comme $\mathrm{Div}\left(\frac{L}{V}\right) = (P)+(Q)-(P+Q)-(P_\infty)$ est principal, $(P)+(Q)-(P+Q)-(P_\infty) \sim 0$. Donc il existe $g \in \mathbf{K}(\mathcal{E})$ tel que $(P) + (Q) - (P + Q) - (P_\infty) = \mathrm{Div}(g)$. On réécrit cette égalité sous la forme : $(P) - (P_\infty) + (Q) - (P_\infty) = (P + Q) - (P_\infty) + \mathrm{Div}(g)$. En passant aux classes d'équivalence modulo $\mathrm{Princ}(\mathcal{E})$, on obtient :

$$[(P) - (P_\infty)] + [(Q) - (P_\infty)] = [(P + Q) - (P_\infty)]$$

c'est à dire $\varphi(P) + \varphi(Q) = \varphi(P + Q)$.

Le quotient $\frac{\mathrm{Div}^0(\mathcal{E})}{\mathrm{Princ}(\mathcal{E})}$ est un groupe : l'addition y est donc associative. Comme l'opération + sur $\mathcal{E}$ lui est isomorphe, + est également associative. Cela conclut la preuve du fait que $(\mathcal{E}, +)$ est un groupe.

4.2. **Comment crypter un message avec une courbe elliptique.**

Soit p un nombre premier. Alors $(\mathcal{E}(\mathbf{F}_p), +)$ reste un groupe (fini). On peut utiliser ce groupe pour chiffrer et déchiffrer des messages via le cryptosystème suivant dit cryptosystème à clé publique de ElGamal sur courbe elliptique.

- Données publiques : un entier premier p, une courbe elliptique $\mathcal{E}$ sur $\mathbf{F}_p$ et un point P de $\mathcal{E}(\mathbf{F}_p)$.
- Création de la clé d'Alice : Alice choisit un entier n_A, qui sera sa clé privée. Puis elle calcule et diffuse le point $Q_A = n_A P$ qui sera sa clé publique.
- Chiffrement d'un message : Connaissant Q_A, Bob souhaite envoyer un message clair $M \in \mathcal{E}(\mathbf{F}_p)$ (le passage d'un texte en français à une séquence d'élément de $\mathcal{E}(\mathbf{F}_p)$, bien que non trivial, n'est qu'une étape de traduction qui n'a pas été abordée dans ce travail). Pour ce faire, il choisit une clé éphémère $k \in \mathbf{N}$. Puis il calcule $C_1 = kP$ et $C_2 = M + kQ_A$. Il envoie enfin le message chiffré (C_1, C_2).
- Déchiffrement du message : Alice calcule $C_2 - n_A C_1 = M + kn_A P - n_A kP = M$ et retrouve le message clair.

Une façon pour Eve de déchiffrer un message est de retrouver la clé privée d'Alice connaissant P et Q_A. Il lui faut alors résoudre l'équation en x suivante : $xP = Q_A$. C'est un calcul de logarithme discret.

5. L'algorithme MOV

Dans cette section, $\mathbf{K}$ désigne un corps de caractéristique différente de deux ou trois. On en considère la clôture algébrique au besoin. $\mathcal{E}$ une courbe elliptique sur $\mathbf{K}$ décrite par l'équation $Y^2 = X^3 + aX + b$.

5.1. Points de n-torsion.

Définition 5.1. Groupe de n-torsion
Soit n un élément de $\mathbf{N}$. Le groupe de n torsion de $\mathcal{E}$, noté $\mathcal{E}[n]$, est l'ensemble de points $\{P \in \mathcal{E} | nP = P_\infty\}$.

Proposition 5.1. *Soit $n \in \mathbf{N}$. Alors $\mathcal{E}[n]$ est isomorphe à $\mathbf{Z}/n\mathbf{Z} \times \mathbf{Z}/n\mathbf{Z}$. En particulier, le cardinal de $\mathcal{E}[n]$ est n^2.*

Démonstration : Admise.

5.2. Couplage de Weil.

5.2.1. *Première définition.*

Cette première définition de le couplage de Weil permet d'en montrer facilement les propriétés.

Notation : Soit $n \in \mathbf{N}$. On note $[n] : \left\{ \begin{array}{ccc} \mathcal{E} & \longleftrightarrow & \mathcal{E} \\ P & \longmapsto & nP \end{array} \right.$

Lemme 5.1. *Soient $n \in \mathbf{N}$ et $T \in \mathcal{E}[n]$ tel qu'il existe T_0 tel que $nT_0 = T$. Les diviseurs $n(T) - n(T_0)$ et $D_T = \sum_{P|nP=T} (P) - \sum_{P \in \mathcal{E}[n]} (P)$ sont principaux.*

Démonstration :
Comme le diviseur $n(T) - n(T_0)$ a pour degré 0 et somme P_∞, il est principal par le théorème 3.1. Donc il existe $f_T \in \mathbf{K}(\mathcal{E})$ tel que $\mathrm{Div}(f_T) = n(T) - n(T_0)$.

Pour D_T, remarquons que $\{P \in \mathcal{E} | nP = T\} = \{P \in \mathcal{E} | \exists R \in \mathcal{E}[n] \text{ tel que } P = R + T_0\}$. On en déduit que $\deg(D_T) = 0$. De plus,

$$\begin{aligned} \text{som}(D_T) &= \sum_{P \in \mathcal{E}[n]} (P + T_0) - \sum_{P \in \mathcal{E}[n]} P \\ &= \sum_{P \in \mathcal{E}[n]} T_0 \\ &= n^2 T_0 \text{ par la proposition 5.1} \\ &= nT = P_\infty \end{aligned}$$

Donc D_T est aussi principal et il existe $g_T \in \mathbf{K}(\mathcal{E})$ tel que $D_T = \text{Div}(g_T)$.

Proposition 5.2. *On reprend les notations du lemme précédent.*
Alors $\text{Div}(f_T \circ [n]) = \text{Div}(g_T^n)$.

Plan de la démonstration : On montre que les zéros et pôles de $f_T \circ [n]$ et g_T^n sont les mêmes. Comme elle sont définies à une constante multiplicative près, on peut même supposer que $f_T \circ [n] = g_T^n$

Définition 5.2. Couplage de Weil
Soit $P \in \mathcal{E}$. On nomme couplage (ou appariement) de Weil l'application suivante :

$$e_n : \begin{cases} \mathcal{E}[n] \times \mathcal{E}[n] & \longrightarrow & \mathbf{U}_n = \{x \in \overline{\mathbf{K}} | x^n = 1\} \\ (S, T) & \longmapsto & \frac{g_T(P+S)}{g_T(P)} \end{cases} \quad \text{où } P \text{ est un élément de } \mathcal{E}.$$

Démonstration : On montre que e_n est bien définie.
Pour ce faire, on montre que $h : \begin{cases} \mathcal{E} & \longrightarrow & \mathbf{U}_n \\ P & \longmapsto & \frac{g_T(P+S)}{g_T(P)} \end{cases}$ est constante.
Cela requiert des notions sur les morphismes sur $\mathcal{E}$ et les isogénies que nous n'avons pas le temps de développer ici.

Le fait que e_n est à valeurs dans $\mathbf{U}_n$ est en revanche clair puisque, pour tout $S \in \mathcal{E}[n]$, pour tout $P \in \mathcal{E}$, on a :

$$g_T(P + S)^n = f_T(nP + nS) = f_T(nP) = g_T(P)^n$$

Proposition 5.3. *Propriétés de* e_n.

(1) *Bilinéarité :* $\forall (R, S, T) \in \mathcal{E}[n]$, $\begin{cases} e_n(R + S, T) = e_n(R, T) e_n(S, T) \\ e_n(R, T + S) = e_n(R, T) e_n(R, S) \end{cases}$

(2) e_n *est non dégénérée :*
Soit $S \in \mathcal{E}[n]$. *Si pour tout* $T \in \mathcal{E}[n]$ *on a* $e_n(S, T) = 1$ *alors* $S = P_\infty$.
De même, soit $T \in \mathcal{E}[n]$. *Si pour tout* $S \in \mathcal{E}[n]$ *on a* $e_n(S, T) = 1$ *alors* $T = P_\infty$.

(3) e_n *est normale :* $\forall T \in \mathcal{E}[n], e_n(T, T) = 1$.

(4) e_n *est antisymétrique :* $\forall (S, T) \in \mathcal{E}[n], e_n(S, T) = \frac{1}{e_n(T,S)}$.

Démonstration : La définition qu'on a donnée de le couplage de Weil est adaptée pour faciliter ces démonstrations. Là encore des notions sur les isogénies sont nécessaires.

5.2.2. *Seconde définition.*

Cette seconde définition de le couplage de Weil permet de calculer effectivement cette application.

Définition 5.3. Seconde définition du couplage de Weil

Soient $n \in \mathbf{N}$ et S et T deux éléments distincts de $\mathcal{E}[n]$. On note f_T (resp f_S) une fonction de $\mathbf{K}(\mathcal{E})$ dont le diviseur est $n(T) - n(P_\infty)$ (resp $n(S) - n(P_\infty)$). Ces fonctions ont même degré et sont définies à une constante multiplicative près : on peut donc les choisir de telle sorte que $\frac{f_T}{f_S}(P_\infty) = 1$ (on dit alors que f_T et f_S sont normalisées).

Si f_T et f_S sont normalisées alors $e_n(S,T) = (-1)^n \frac{f_T(S)}{f_S(T)}$.

Remarque : Pour que cette définition fasse sens, il faut montrer que cette définition coïncide avec la définition de 5.2. La démonstration a été étudiée et comprise.

Algorithme 5.1. *Calcul de le couplage de Weil*

Pour calculer $e_n(T,S)$ il suffit, selon la dernière définition, de savoir calculer $f_S(T)$ (et $f_T(S)$ par le même procédé) où S et T sont des éléments de $\mathcal{E}[n]$.

Pour tout $i \in [\![1,n]\!]$, on note D_i le diviseur $i(S) - (iS) - (i-1)(P_\infty)$. Il est principal donc est le diviseur d'une fonction f_I de $\mathbf{K}(\mathcal{E})$. Pour $i = n$ en particulier $D_n = n(S) - (nS) - (n-1)(P_\infty) = n(S) - n(P_\infty) = \mathrm{Div}(f_S)$. On peut donc supposer $f_n = f_S$.

L'idée est de calculer par récurrence les $f_i(T)$ pour obtenir $f_S(T)$.

Lemme 5.2. *Soit $S \in \mathcal{E}[n]$ et $(i,j) \in \mathbf{N}^*$.*

- *Si $(i+j)S \neq P_\infty$ alors $f_{i+j} = f_i f_j \frac{l}{d}$ où l est le polynôme décrivant la droite passant par iS et jS et d est celui décrivant la droite verticale qui passe par $(i+j)S$.*
- *Si $(i+j)S = P_\infty$ alors $f_{i+j} = f_i f_j d$ où d est le polynôme décrivant la droite verticale passant par iS et jS.*

Démonstration :

- Si $(i+j)S \neq P_\infty$, il suffit de montrer que $\mathrm{Div}(f_{i+j}) = \mathrm{Div}(f_i) + \mathrm{Div}(f_j) + Div(l) - \mathrm{Div}(d)$. Or, par un rapide calcul n'utilisant que la définition des f_i, on trouve que ces deux quantités sont égales à $(i+j)(S) - ((i+j)S) - (i+j-1)(P_\infty)$ ce qui conclut.
- Si $(i+j)S = P_\infty$, il suffit de montrer que $\mathrm{Div}(f_{i+j}) = \mathrm{Div}(f_i) + \mathrm{Div}(f_j) + \mathrm{Div}(d)$. Un calcul similaire montre que ces deux quantités sont égales à $(i+j)(S) - (i+j)(P_\infty)$.

On commence donc par calculer $f_1 = 1$. Puis on obtient f_n en construisant n par une chaîne d'additions ; c'est-à-dire qu'on construit une suite $i_0 = 1 < i_1 < ... < i_l = n$ telle que $\forall j \geqslant 1, \exists a, b < j$ tel que $i_j = i_a + i_b$.

Remarques

- Pour que le calcul aboutisse, T ne doit pas être un zéro d'une droite verticale passant par un point kS (annulation du dénominateur). Mais, s'il existe k tel que T est sur la droite verticale passant par kS alors $T = \pm kS$. Dans ce cas, $e_n(S,T) = e_n(S, \pm kS) = e_n(S,S)^{\pm k} = 1$.

- Pour obtenir des fonctions normalisées, il suffit de rendre les f_i unitaires. Pour ce faire, il suffit d'écrire les droites verticales sous la forme $X - x_P$ et les autres sous la forme $Y + aX + b$.

5.2.3. *Une application du couplage de Weil : l'algorithme MOV.*

Lemme 5.3. *Soit $n \in \mathbf{N}$. Soient P et S deux points de $\mathcal{E}[n]$. Alors, (P, S) est une base de $\mathcal{E}[n]$ si et seulement si $\rho = e_n(P, S)$ est une racine n-ième primitive de l'unité.*

Démonstration :
Supposons que (P, S) est une base de $\mathcal{E}[n]$. Soit d tel que $\rho^d = 1$. On a déjà $d \leqslant n$. Il suffit donc de montrer que $n|d$ pour conclure que ρ engendre $\mathbf{U}_n$.

Soit $R \in \mathcal{E}[n]$. Par la proposition 5.1, il existe $(a, b) \in \mathbf{Z}$ tel que $R = aP + bS$. Donc

$$e_n(R, dS) = e_n(aP + bS, dS) = e_n(P, S)^{da} e_n(S, S)^{db} = 1$$

Comme e_n est non dégénérée, cela implique que $dS = P_\infty$ donc que l'ordre de S (à savoir n puisque S engendre une groupe isomorphe à $\mathbf{Z}/n\mathbf{Z}$) divise d.

Réciproquement, si ρ est une racine primitive, on montre que P et S sont d'ordre n.

Proposition 5.4. *Soient p un nombre premier et S un élément de $\mathcal{E}[n]$. On note k l'entier tel que $\overline{\mathbf{F}_p} = \mathbf{F}_{q_k}$ et $\mathbf{U}_n = \{x \in \mathbf{F}_{q^k} | x^n = 1\}$. Alors, $h_S : \begin{cases} \mathcal{E}[n] & \longrightarrow & \mathbf{U}_n \\ R & \longmapsto & e_n(R, S) \end{cases}$ est un isomorphisme de groupes.*

Démonstration : La bilinéarité de e_n implique que h_S est un morphisme. Le lemme 5.3 garantit la bijection

Le but est d'utiliser cette bijection de façon à ramener le calcul d'un logarithme discret dans $\mathcal{E}$ à un calcul de logarithme discret dans $\mathbf{F}_{q^k}$, corps dans lequel on a des algorithmes efficaces.

Algorithme 5.2. *Algorithme MOV (Menezes, Okamoto et Vanstone)*
Entrée : un point P d'ordre n dans $\mathcal{E}(\mathbf{F_q}$ et un point Q appartenant au sous groupe de $\mathcal{E}[n]$ engendré par P. Sortie : Un entier l tel que $lP = Q$.

(1) *On détermine un entier k tel que $\mathcal{E}[n] \subset \mathcal{E}(\mathbf{F}_{q^k})$*
(2) *On choisit $S \in \mathcal{E}(\mathbf{F}_{q^k})$ tel que (P, S) est une base de e_n (par exemple, on choisit S au hasard dans $\mathcal{E}[n]$ jusqu'à ce que (P, S) soit une base de $\mathcal{E}[n]$).*
(3) *On calcule $\alpha = e_n(P, S)$ et $\beta = e_n(Q, S)$.*
(4) *Dans $\mathbf{F}^*_{q^k}$, on calcule un logarithme l de β en base α.*

Démonstration : On montre que l est bien un logarithme de Q en base P. On sait qu'il existe $j \in \mathbf{N}$ tel que $Q = jP$ et que l'algorithme MOV trouve un entier l tel que $e_n(P, S)^l = e_n(Q, S)$. Donc :

$$e_n(P, S)^l = e_n(Q, S) = e_n(jP, S) = e_n(P, S)^j$$

Donc $e_n(P, S)^{l-j} = 1$. Or, $e_n(P, S)$ est une racine primitive de l'unité (puisque (P, S) est une base de $\mathcal{E}[n]$). Donc $l \equiv j \bmod n$ ce qui conclut.

Remarques

- Ni la complexité de cet algorithme, ni le calcul effectif de k dans la première étape n'ont été étudiés.
- Plus l'entier k est petit, plus le calcul de logarithme discret dans $\mathbf{F}_{q^k}^*$ est facile. Pour que le DLP sur une courbe elliptique soit compliqué, il faut donc éviter les courbes pour lesquelles k est petit.

Conclusion

L'objectif de ce stage était d'une part de comprendre d'un point de vue théorique l'algorithme MOV et d'autre part d'implémenter ledit algorithme. La première partie de cet objectif est remplie, la seconde en partie seulement. Seuls les algorithmes permettant le calcul de logarithmes discrets dans des groupes du type $(\mathbf{F}_{q^k}^*, \times)$ ont été implémentés. Cela est en partie dû à certaines digressions, non présentées ici, lors de l'étude mathématique de l'algorithme MOV autour des courbes vulnérables à cette attaque. La suite logique de ce travail serait sans doute de considérer plus en détail ces courbes. En effet, leur relative vulnérabilité à l'algorithme MOV est souvent contrebalancée par la connaissance de données permettant d'accélérer significativement les calculs lors du chiffrage et du déchiffrage. La tâche suivante serait ensuite de déterminer si possible une famille de courbes alliant sécurité et efficacité.

Références

[1] Phillipe Guillot, *Courbes elliptiques, une présentation élémentaire pour la cryptographie*, Lavoisier, 2010

[2] Michael Hägler, Courbes elliptiques et cryptographie [en ligne], 2006. Disponible sur : http ://math.univ-bpclermont.fr/∼rebolledo/page-fichiers/projetMichael.pdf

[3] D. Husemöller, *Elliptic curves (2^{nd} edition)*, volume 111 of *Graduate Texts in Mathematics*, Springer-Verlag, 2004

[4] Jeffrey Hoffstein, Jill Pipher et Joseph H. Silverman, *An introduction to mathematical cryptography*, Springer, 2008, Chapitres 1, 2 et 5

[5] Guillaume Lafon, Les courbes elliptiques pour les nuls [en ligne], 2003. Disponible sur : http ://www.normalesup.org/∼ glafon/maths/courbes_elliptiques.pdf

[6] Vanessa Vitse, Couplages sur courbes elliptiques définies sur des corps finis [en ligne], université Versailles-Saint-Quentin. Disponible sur : https ://www-fourier.ujf-grenoble.fr/ viva/research/articles/thesis.pdf

[7] Lawrence C. Washington, *Elliptic curves : number theory and cryptograhpy (2^{nd} edition)*, Chapman & Hall/CRC, 2008

[8] Gilles Zémor, *Cours de cryptographie*, Cassini, 2000

Annexe : Implémentation des algorithmes de la partie 1

Prérequis : Algorithme d'Euclide et théorème chinois.

```
#Entree = Deux entiers naturels non nuls.
#Sortie = Le plus grand diviseur commun a ces deux entiers.
def pgcd(a,b):
    aa = max(a,b)
    bb = min(a,b)
    while aa % bb != 0:
        aa,bb = bb , aa % bb
    return bb

#Entree = Deux entiers naturels eventuellement nuls, a et b.
#Sortie = Le couple d'entiers de Bezout (u,v) (dans cet ordre)
#tel que au+bv = pgcd(a,b).
def bezout(a,b):
    r_0 = max(a,b)
    r_1 = min(a,b)
    u_0 = 1
    u_1 = 0
    v_0 = 0
    v_1 = 1
    while r_1 != 0:
        q = r_0/r_1
        r_0, r_1 = r_1, r_0 - q*r_1
        u_0, u_1 = u_1, u_0 - q*u_1
        v_0, v_1 = v_1, v_0 - q*v_1
    if (a == 0 or b == 0):
        u_0, v_0 = 1,1
    elif a % b == 0:
        u_0 = 1
        v_0 = -a/b +1
    elif b % a ==0:
        u_0 = -b/a +1
        v_0 = 1
    if a >= b:
        return u_0, v_0
    else:
        return v_0, u_0
```

```
#Entree = Un entier a et un nombre premier p.
#Sortie = Un representant de la classe de l'inverse de la classe
#de a modulo p.
def inv(a,p):
    aa = a % p
    u, v = bezout(aa,p)
    return u

#Entree = Une liste l et un entier n.
#Sortie = La liste l privee des n premiers elements.
def enleve(l,n):
    return l[n:]

#Entree = Deux couples d'entiers tels que m et n sont premiers entre eux.
#Sortie = Une solution x de systeme : x = a mod n ; x = b mod m.
def th_chinois_2((a,n), (b,m)):
    u, v = bezout(n, m)
    return (b*n*u + a*m*v) % (n*m)

#Entree = Une liste de couples (a_i, m_i) telle que
#les m_i sont deux a deux premiers entre eux.
#Sortie = La solution x du systeme de congruences x = a_i mod m_i
#qui se trouve entre 0 et le produit des m_i.
def th_chinois(l):
    while len(l) != 2:
        M = l[0][1] * l[1][1]
        c = th_chinois_2(l[0], l[1]) % M
        l = enleve(l,2)
        l = [(c,M)] + l
    return th_chinois_2(l[0],l[1])
```

Baby-step, giant-step.

Deux versions de cet algorithme ont été implémentées. La seconde utilise un dictionnaire.

```
#Entree = (g: un entier), (A: un entier naturel), (N: un entier naturel)
#Sortie = Le resultat de g^A modulo N par exponentiation rapide.
def sqm(g,A,n):
    if A == 0:
        return 1
    elif A % 2 == 0:
        b = sqm(g, A//2, n)
        return (b*b) % n
    else:
        b = sqm(g, A//2, n)
        return (b*b*g) % n

#Entree = Deux listes l1 et l2.
#Sortie = Les indices dans l1 et l2 d'un element commun aux deux.
def collision2(l1, l2):
    s1, s2 = set(l1), set(l2)
    l = list(s1 & s2)
    c = l[0]
    return l1.index(c), l2.index(c)

#Entree = Un nombre premier p et deux entiers non nuls modulo p,
#g et h tels qu'il existe une solution a l'equation g^x = h mod p.
#Sortie = Une solution x de l'equation g^x = h mod p
#Methode = Baby step - giant step.
def bs_gs(g,h,p):
    n = int(math.sqrt(p-1)) +1
    bs = [0]*(n+1)
    gs = [0]*(n+1)
    bs[0] = 1
    gs[0] = h
    inverse = sqm(inv(g,p),n,p)
    for k in range(1,n+1):
        bs[k] = (bs[k-1]*g) %p
        gs[k] = (gs[k-1]*inverse) %p
    i , j = collision2(bs, gs)
    return(i + n*j)
```

```
#Entree = Un nombre premier p et deux entiers non nuls modulo p,
#g et h tels qu'il existe une solution a l'equation g^x = h mod p.
#Sortie = Une solution x de l'equation g^x = h mod p
#Methode = Baby step - giant step et dictionnaire
#(ce qui permet d'accelerer la recherche d'une correspondance).
def bs_gs2(g,h,p):
    n = int(math.sqrt(p-1)) +1
    bs = {1:0}
    gg = 1
    hh = h
    inverse = sqm(inv(g,p),n,p)
    x1 = -1
    x2 = -1
    j = 0
    for i in range(1,n+1):
        gg = gg*g % p
        bs[gg] = i
    while (j <= n) and (x1 == -1):
        if hh in bs:
            x1 = bs[hh]
            x2 = j
        else:
            hh = hh*inverse % p
            j += 1
    return (x1 + n*x2)
```

Résultats.

Avec $g = 12569$, $h = 1254$ et $p = 1255211$:

- L'algorithme naïf donne la réponse (1086464) en 0.27 secondes
- " `bs_gs`" donne la réponse en 0.0 secondes
- " `bs_gs2`" donne la réponse en 0.0 secondes

Avec $g = 1255$, $h = 901609336397$ et $p = 1307674368043$:

- L'algorithme naïf donne la réponse en 266 secondes
- " `bs_gs`" donne la réponse en 1.8 secondes
- " `bs_gs2`" donne la réponse en 0.7 secondes

Algorithme de Pohlig-Hellman.

Première version.

```
#Entree = Un entier relatif x et un entier naturel n.
#Sortie = x^n par exponentiation rapide.
def power(x,n):
    if n == 0:
        return 1
    elif n % 2 == 0:
        b = power(x, n//2)
        return b*b
    else:
        b = power(x, n//2)
        return b*b*x

#Entree = Un entier premier p ; deux entiers non nuls modulo p :
#g et h tq l'equation g^x = h mod p admet une solution et
#N = ordre de g dans Z/pZ*
#Sortie = un entier x tel que g^x = h mod p
#Methode = Pohlig Hellman en se ramenant a resoudre le DLP pour
#des elements d'ordre une PUISSANCE d'un nombre premier.
def ph(g,h,p,N):
    facteurs = fact(N)
    l = []
    for cle in facteurs.keys():
        m_i = power(cle, facteurs[cle])
        a_i = bs_gs2(sqm(g, N//m_i, p), sqm(h, N//m_i, p), p)
        l = l + [(a_i,m_i)]
    if len(l) >= 2:
        return th_chinois(l)
    else:
        return l[0][0] #Si il n'y a qu'un facteur premier dans N
```

Version améliorée.

```
#Entree = Un entier premier p, deux nombres modulo p :
#g et h et deux entiers q et e tq q^e est l'ordre de g
#et il existe une solution a g^x = h mod p.
#Sortie = Un entier x tq g^x = h mod p
def ph_1(g,h,q,e,p):
    x = 0
    puiss = power(q, e-1)
    puissi = puiss
    gg = sqm(g, puiss, p)
    hh = sqm(h, puiss, p)
    inverse = inv(g,p)
    for i in range(e):
        hh = sqm(h *sqm(inverse, x, p), puissi, p)
        x_i = bs_gs2(gg, hh, p)
        s_i =  (x_i)*power(q,i)
        x = x + s_i
        puissi = puissi//q
    return x


#Entree = Un entier premier p, deux entiers non nuls modulo p :
#g et h tq l'equation g^x = h mod p admet une solution et
#N = ordre de g dans Z/pZ*
#Sortie = Un entier x tq g^x = h mod p.
#Methode = Pohlig Hellman en se ramenant a resoudre le DLP pour
#des elements d'ordre premier.
def ph2(g,h,p,N):
    facteurs = fact(N)
    l = []
    for cle in facteurs.keys():
        e_i = facteurs[cle]
        m_i = power(cle, e_i)
        a_i = ph_1(g, h, cle, e_i, p)
        l = l + [(a_i,m_i)]
    if len(l) >= 2:
        return th_chinois(l)
    else:
        return l[0][0] #Si il n'y a qu'un facteur dans N
```

Résultats. Contrairement à la théorie, sur les valeurs testées, ces deux algorithmes sont plus lents que " `bs_gs`" et " `bs_gs2`". Deux causes possibles : des valeurs tests pas assez grandes ou plus vraisemblablement une implémentation induisant de trop nombreux calculs inutiles.