Université Versailles-Saint-Quentin MA^2 - Master Algèbre Appliquée

Couplages sur courbes elliptiques définies sur des corps finis

Vanessa Vitse

Mémoire de Master 2

Soutenu le 10/09/2008

Réalisé sous la direction de
Louis Goubin

Table des matières

Introduction **2**

1 Rappels sur les courbes elliptiques **4**
1.1 Equation de Weierstrass d'une courbe elliptique, courbes tordues 4
1.2 Groupe de Picard d'une courbe elliptique 6
1.3 Isogénies, endomorphismes et points de torsion 7
1.3.1 Isogénie, isogénie duale 7
1.3.2 Groupes de torsion 8
1.3.3 Structure de l'anneau des endomorphismes 9
1.4 Courbes elliptiques définies sur des corps finis 9
1.4.1 Conjecture de Weil et borne de Hasse 9
1.4.2 Courbes supersingulières et ordinaires 11
1.4.3 Anneaux d'endomorphismes et corps quadratiques 12

2 Couplage de Weil et de Tate **12**
2.1 Couplage de Weil : définition, propriétés 12
2.2 Couplage de Weil et points de l-torsion 13
2.3 Couplage de Tate 14
2.4 Algorithme de Miller 15
2.4.1 Principe de l'algorithme 15
2.4.2 Raffinements 16
2.4.3 Implémentation et étude de la complexité 18

3 Utilisation des couplages en cryptographie **20**
3.1 Attaque de MOV/Frey-Rück 20
3.2 Hypothèses de sécurité liées aux couplages 20
3.3 Distribution non interactive de clés basée sur l'identité 21
3.4 Un protocole Diffie-Hellman pour trois parties en un tour 23
3.5 Le chiffrement de Boneh-Franklin basé sur l'identité 24

4 Construction de courbes adaptées (*pairing-friendly curves*) **25**
4.1 Utilisation du couplage de Tate 25
4.2 Courbes supersingulières et applications de distorsion 27
4.3 Construction de courbes par la méthode CM 31
4.4 Un exemple de courbes ordinaires : les courbes MNT 31

A Implémentation **35**
A.1 Présentation générale des différents modules 35
A.2 Un exemple pour IBE 37
A.3 Algorithmes pour la recherche de points engendrant la r-torsion 38
A.4 Fonctions de hachage utilisées pour IBE 40

Bibliographie **41**

Introduction

Miller et Koblitz ont introduit en 1985, indépendamment l'un de l'autre, la cryptographie fondée sur les courbes algébriques. Ils proposent de généraliser des protocoles tels que l'échange de clés Diffie-Hellman ou la signature d'El Gamal, dont la sécurité repose sur la difficulté de calculer des logarithmes discrets sur $\mathbf{F}_q^*$, à d'autres groupes, notamment aux groupes des points rationnels d'une courbe elliptique définie sur un corps fini. Car, contrairement au groupe multiplicatif de $\mathbf{F}_q$, seuls des algorithmes exponentiels en la taille de q sont généralement disponibles pour le calcul de logarithmes discrets dans $E(\mathbf{F}_q)$.

La mise en place de tels schémas sur courbe elliptique soulève cependant une difficulté : pour pouvoir s'assurer que les points de la courbe considérée ont un ordre divisible par un grand facteur premier, il est indispensable de savoir calculer la cardinalité du groupe des points rationnnels. C'est pourquoi dans un premier temps n'ont été considérées que des courbes définies sur une extension d'un petit corps (il est facile d'obtenir la cardinalité de $E(\mathbf{F}_{q^k})$ une fois déterminée celle de $E(\mathbf{F}_q)$ par recherche exhaustive des points). Mais cette approche ne fournit pas une grande diversité de courbes et ne permet pas de travailler en grande caractéristique. Une autre idée était alors d'utiliser des courbes particulières dont on sait calculer facilement la cardinalité : les courbes supersingulières. Malheureusement, en 1993, Menezes, Okamoto et Vanstone lancent une attaque sur ce type de courbe [MOV93], rendant leur usage en cryptographie délicat ; l'idée est d'utiliser les propriétés du couplage de Weil (dont l'existence était connue des mathématiciens depuis le milieu du XX$^{\text{ème}}$ siècle) pour ramener le problème du logarithme discret sur $E(\mathbf{F}_q)$ à celui sur $\mathbf{F}_{q^k}$ où $k \leq 6$, pour lequel on connaît des algorithmes sous-exponentiels.

Depuis, de considérables progrès ont été réalisés dans ce domaine et, notamment grâce aux travaux de Schoof, Elkies et Atkin, il existe désormais des algorithmes efficaces pour calculer le nombre de points. D'autres méthodes permettent aussi de construire des courbes avec un nombre de points fixés a priori, rendant possible l'implémentation de schémas cryptographiques basés sur la difficulté de résoudre le problème du logarithme discret sur courbes elliptiques.

Plus récemment, suite à l'apparition des couplages de Weil et Tate avec les attaques MOV et Frey-Rück [FR94], des applications cryptographiques plus constructives sont apparues. En 2000, Joux met à profit ces couplages en expliquant qu'il est possible, avec les propriétés de bilinéarité du couplage de Weil, de faire un échange type Diffie-Hellman entre trois personnes en un tour seulement [Jou04]. Lors de la conférence Crypto 2001, Boneh et Franklin proposent à leur tour un schéma de chiffrement basé sur l'identité utilisant ce couplage [BF03], et répondent ainsi à un problème posé par Shamir en 1984 et resté jusqu'alors sans réponse. La cryptographie basée sur les couplages connaît depuis un véritable engouement, donnant lieu depuis 2007 à une conférence annuelle dédiée.

La plupart de ces nouveaux schémas cryptographiques nécessitent cependant des courbes ayant des propriétés spécifiques que n'ont pas en général les courbes produites aléatoirement. Un intérêt particulier est donc porté à la construction de ces courbes dites *pairing-friendly* ou courbes bien couplées.

L'objectif de ce travail est de présenter un panorama des notions mathématiques nécessaires à la compréhension des couplages et à la construction de courbes bien couplées, dans le but d'une implémentation complète du protocole de chiffrement basé sur l'identité de Boneh-Franklin en langage `C++`.

Plan du mémoire

Ce mémoire se décompose en quatre parties. On présente tout d'abord le contexte mathématique sous-jacent à la cryptographie basée sur courbes elliptiques. On détaille ensuite la construction de deux couplages fondamentaux, le couplage de Weil et le couplage de Tate, ainsi qu'un algorithme dû à Miller [Mil86] permettant un calcul efficace de ces couplages. Un premier module réalisé en `C++` s'appuyant sur les bibliothèques GMP (*GNU Multiprecision Package*) et NTL (*Number Theory Library*) implémente ces couplages.

Des protocoles célèbres, tels que l'échange tripartite en un tour de Joux [Jou04] ou le chiffrement basé sur l'identité de Boneh-Franklin [BF03], ainsi que les hypothèses de sécurité afférentes à ces protocoles sont étudiés dans la troisième partie. Ce contexte cryptographique nécessitant des courbes et couplages bien particuliers, on explique dans la quatrième partie comment il est possible d'en construire en pratique avec la méthode de multiplication complexe. Deux autres modules ont ainsi été implémentés : le premier permet une analyse détaillée d'une courbe dont on connait l'équation de Weierstrass, le second permet la construction de courbes ordinaires spécifiques dites MNT. Une version simple du protocole de chiffrement basé sur l'identité de Boneh-Franklin (`BasicIndent`) a alors pu être programmée en utilisant différentes courbes générées à l'aide de la méthode de multiplication complexe.

Dans l'annexe, on détaille les choix d'implémentation ainsi que le fonctionnement des différents modules. Un listing des programmes suit.

Ce mémoire s'inscrit dans le cadre du stage de fin d'année du Master d'Algèbre Appliquée de l'Université de Versailles-Saint-Quentin qui s'est déroulé de mai à septembre 2008.

Je remercie Louis Goubin d'avoir accepté de m'encadrer pour ce travail, Antoine Joux pour ses précieux conseils en programmation, ainsi que Reynald Lercier qui a eu la gentillesse de mettre à disposition son module de calcul du nombre de points d'une courbe elliptique.

1 Rappels sur les courbes elliptiques

On détaille dans cette partie les outils mathématiques nécessaires à l'étude des couplages sur courbes elliptiques définies sur des corps finis. Pour les résultats énoncés sans démonstration, on renvoie à l'ouvrage de référence [Sil86].

1.1 Equation de Weierstrass d'une courbe elliptique, courbes tordues

Définition 1.1. *Une courbe elliptique E définie sur un corps k est une courbe algébrique plane non singulière dont l'équation est de la forme suivante, appelée* équation de Weierstrass *:*

$$y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6$$

où $a_1, a_2, a_3, a_4, a_6 \in k$.

Une telle courbe possède un unique point à l'infini, noté O.

Cette définition n'est pas celle que l'on trouve dans un contexte algébrique plus général (où une courbe elliptique est une courbe non singulière de genre 1 avec un point distingué). Dans la mesure où toute courbe elliptique admet une équation de Weierstrass, et que c'est sous cette forme qu'on les implémente en pratique, on se contentera dans le cadre du mémoire de cette définition simplifiée.

Si K est une extension algébrique de k, on note $E(K)$ l'ensemble des *points K-rationnels* de E, i.e. l'ensemble des solutions de l'équation de Weierstrass dans K (y compris le point à l'infini).

Proposition 1.2. *Tout isomorphisme entre deux courbes elliptiques est de la forme*

$$\begin{cases} x = u^2x' + r \\ y = u^3y' + u^2sx' + t \end{cases}$$

avec $u, r, s, t \in \overline{k}$, $u \neq 0$. Si $u, r, s, t \in K$, on dit que les deux courbes sont K-isomorphes.

Si $\mathrm{char}(k) \neq 2, 3$, toute courbe elliptique définie sur k est k-isomorphe à une courbe d'*équation de Weierstrass réduite*, i.e. de la forme :

$$y^2 = x^3 + Ax + B.$$

On associe à une telle courbe l'élément

$$j(E) = 1728\frac{4A^3}{4A^3 + 27B^2}$$

appelée *j-invariant*.

Proposition 1.3. *Deux courbes elliptiques sont isomorphes si et seulement si elles ont même j-invariant.*

Soit E une courbe elliptique définie sur k. Si $\tilde{E}$ définie sur k a le même j-invariant que E, alors les deux courbes sont isomorphes, mais cet isomorphisme n'est pas nécessairement défini sur k. Dans ce dernier cas, on dit que $\tilde{E}$ est la *tordue* de E.

De façon plus précise, si $\mathrm{char}(k) \neq 2, 3$ et si E et $\tilde{E}$ sont d'équations réduites :

$$(E) : y^2 = x^3 + ax + b \qquad (\tilde{E}) : y'^2 = x'^3 + \tilde{a}x' + \tilde{b},$$

alors

$$j(E) = j(\tilde{E}) \Rightarrow a^3\tilde{b}^2 = \tilde{a}^3 b^2.$$

Par conséquent, si $a, b, \tilde{a}, \tilde{b}$ sont non nuls (autrement dit si $j(E) \neq 0, 1728$), en posant $v = \dfrac{b\tilde{a}}{a\tilde{b}}$, on a

$$(\tilde{E}) : y'^2 = x'^3 + av^{-2}x' + bv^{-3}, \text{ avec } v \in k \setminus \{0\}$$

On distingue alors deux cas :
– soit v est un carré dans k, alors en prenant $u \in k$ tel que $u^2 = v$, on obtient un k-isomorphisme

$$\varphi : (x', y') \mapsto \begin{cases} x = u^2 x' \\ y = u^3 y' \end{cases}$$

entre les deux courbes E et $\tilde{E}$.
– soit v n'est pas un carré dans k, alors les deux courbes ne sont pas k-isomorphes. Cependant en prenant une racine $u \in K$ de v dans une extension quadratique K de k, on établit un K-isomorphisme entre E et $\tilde{E}$. On dit dans ce cas que $\tilde{E}$ est une *tordue quadratique* de E.

En résumé :

Proposition 1.4. *On suppose $j(E) \neq 0, 1728$.*
Si E et $\tilde{E}$ définies sur k ont même j-invariant, mais ne sont pas k-isomorphes, il existe une extension quadratique K de k telle que les deux courbes soient K-isomorphes.

Exemple. Dans le cas des corps finis, deux courbes elliptiques définies sur $\mathbf{F}_q$ (q puissance d'un nombre premier p) qui ont même j-invariant différents de 0 et 1728 sont $\mathbf{F}_{q^2}$-isomorphes.

Plus précisément, on a le résultat suivant :

Proposition 1.5.
Soient $j_0 \in \mathbf{F}_q$, $j_0 \neq 0, 1728$ et $\mathcal{E}_{j_0} = \{E$ courbe elliptique définie sur $\mathbf{F}_q : j(E) = j_0\}$.

On définit sur $\mathcal{E}_{j_0}$ la relation d'équivalence :

$$E_1 \sim E_2 \Leftrightarrow E_1 \text{ est } \mathbf{F}_q\text{-isomorphe à } E_2$$

Alors

$$\#\left(\mathcal{E}_{j_0}/_{\sim}\right) = 2$$

A isomorphisme près, on a donc une seule tordue pour E que l'on appellera la tordue de E.

Démonstration. On note

$$(E) : y^2 = x^3 + ax + b$$

l'équation réduite de E. Soient E_1 et E_2 deux courbes $\mathbf{F}_{q^2}$-isomorphes à E, d'équations réduites :

$$(E_1) : y_1^2 = x_1^3 + a_1x_1 + b_1 \qquad (E_2) : y_2^2 = x_2^3 + a_2x_2 + b_2.$$

Comme $j_0 \neq 0, 1728$, si E_1 et E_2 ne sont pas $\mathbf{F}_q$-isomorphes à E, alors

$$v_1 = \frac{ba_1}{ab_1} \text{ et } v_2 = \frac{ba_2}{ab_2}$$

sont des non résidus quadratiques. En particulier, on peut trouver $u \in \mathbf{F}_q$ tel que $u^2 = \dfrac{v_2}{v_1}$, et qui définit donc un $\mathbf{F}_q$-isomorphisme

$$\varphi : (x_2, y_2) \mapsto \begin{cases} x_1 = u^2 x_2 \\ y_1 = u^3 y_2 \end{cases}$$

entre les deux courbes E_1 et E_2. □

Remarque 1.6. Dans le cas où $j = 0$ ou 1728, il existe aussi des tordues *quartiques* ou *sextiques*.

1.2 Groupe de Picard d'une courbe elliptique

Soit E une courbe elliptique définie sur un corps k.

L'ensemble des diviseurs de E est le groupe abélien libre engendré par les points de la courbe, on le note

$$Div_{\bar{k}}(E) = \left\{ \sum_{P \in E} n_P(P) : n_P \in \mathbf{Z} \text{ presque tous nuls} \right\}$$

Si $D = \sum_{P \in E} n_P(P) \in Div_{\bar{k}}(E)$, l'ensemble des points $P \in E(\bar{k})$ tels que $n_P \neq 0$ est appelé *support* de D et noté $\text{supp}(D)$.

On dit que le diviseur D est *défini sur* k, si D est laissé invariant par l'action du groupe de Galois $Gal(\bar{k}/k)$ et on note $D \in Div_k(E)$.

Le degré de $D = \sum_{P \in E} n_P(P) \in Div(E)$ est défini par

$$\deg(D) = \sum_{P \in E} n_P \in \mathbf{Z}$$

et l'ensemble des diviseurs de degré 0 est noté $Div^0(E)$.

Si $f \in k(E)$ est une fonction rationnelle définie sur E et P est un point de la courbe, on note $\text{ord}_P(f)$ *la valuation* de la fonction f au point P. On peut alors définir le sous-groupe des *diviseurs principaux* comme l'image de l'application

$$\begin{array}{rccc} \text{div} : & k(E) & \to & Div_k(E) \\ & f & \mapsto & \text{div}\,(f) = \sum_{P \in E} \text{ord}_P(f)(P) \end{array}$$

Tous ces diviseurs sont de degré nul ([Sil86] p.34).

Exemple (Diviseur de l'équation d'une droite). Soient P et Q deux points de la courbe E, et $(f = 0)$ l'équation de la droite affine passant par ces deux points. D'après Bézout, cette droite coupe la courbe en un $3^{\text{ème}}$ point R, avec éventuellement $R = O$ (le point à l'infini). Le diviseur associé à f est donc de la forme :

$$\text{div}\,(f) = (P) + (Q) + (R) - 3(O)$$

Soient K une extension finie de k, $D = \sum_{P \in E} n_P(P) \in Div_K(E)$ et $f \in K(E)$, tels que les supports de $\text{div}\,(f)$ et de D soient disjoints. Il est alors possible de définir

$$f(D) = \prod_{P \in E} f(P)^{n_P} \in K$$

Dans le cas où le diviseur D est principal, on a de plus la propriété suivante :

Propriété 1.7 (Loi de réciprocité de Weil [Gal05])**.**
Soient f, g définie sur E telles que $supp(div\,(f)) \cap supp(div\,(g)) = \emptyset$. Alors

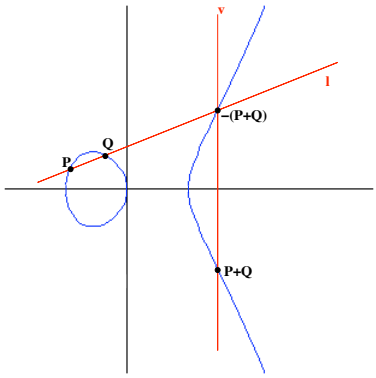
$$f(div\,(g)) = g(div\,(f))$$

Le groupe de Picard (des diviseurs de degré 0) $Pic^0(E)$ est obtenu en quotientant le sous-groupe $Div^0(E)$ par le groupe des diviseurs principaux. Avec des notations évidentes, on peut définir de façon similaire $Pic^0_k(E)$. L'application

$$\begin{array}{ccc} E & \to & Div^0(E) \\ P & \mapsto & (P) - (O) \end{array}$$

induit alors une bijection $\kappa : E \to Pic^0(E)$ ([Sil86] prop. III.3.4).

Par transport de structure, on obtient alors une loi de groupe commutative sur E, notée $+$, d'élément neutre O et qui correspond à la construction géométrique bien connue. Des formules explicites pour le calcul des coordonnées du point $P + Q$ à partir de celles de P et Q sont données dans [Sil86] (algorithme III.2.3 p. 58).



Le lemme suivant ([Sil86] p.67), qui donne une caractérisation des diviseurs principaux, sera utile pour la construction des couplages sur courbe elliptique :

Lemme 1.8. *Soit $D = \sum n_p(P) \in Div_k(E)$ un diviseur de E. Alors*

$$D \text{ est principal } \Leftrightarrow \sum n_P = 0 \text{ et } \sum [n_P]P = O$$

où $[n_P]P$ est obtenu en sommant n_P fois le point P.

1.3 Isogénies, endomorphismes et points de torsion

1.3.1 Isogénie, isogénie duale

Soient E_1, E_2 deux courbes elliptiques définies sur un corps k et O_1, O_2 leurs points à l'infini, une fonction rationnelle $\varphi : E_1 \to E_2$ tel que $\varphi(O_1) = O_2$ est appelé *isogénie*.

Comme tout morphisme de courbes, une isogénie est soit surjective, soit constante. Les isogénies vérifient par ailleurs la propriété remarquable d'être aussi des morphismes de groupes ([Sil86] III.4).

A toute isogénie non constante $\varphi : E_1 \to E_2$, on associe le morphisme de corps injectif

$$\varphi^* : \bar{k}(E_2) \to \bar{k}(E_1), \ f \mapsto f \circ \varphi.$$

On dit que $\varphi : E_1 \to E_2$ est *séparable, inséparable ou purement inséparable* si l'extension de corps correspondante $\bar{k}(E_1) \ / \ \varphi^*\bar{k}(E_2)$ est respectivement séparable, inséparable ou purement inséparable.

On définit le *degré* de φ, noté $\deg \varphi$, comme étant le degré de l'extension correspondante $\bar{k}(E_1) \ / \ \varphi^*\bar{k}(E_2)$; de même pour le *degré de séparabilité* $\deg_s \varphi$ et le *degré d'inséparabilité* $\deg_i \varphi$. En particulier

$$\deg \varphi = \deg_s \varphi \ \deg_i \varphi.$$

Proposition 1.9. *([Sil86] p. 76)*
Soit $\varphi : E_1 \to E_2$ une isogénie non constante.

Le degré de séparabilité de φ est égal au nombre de points du noyau de φ :

$$\#(\ker \varphi) = \deg_s \varphi$$

Le degré d'inséparabilité est égal au degré de ramification de φ au dessus de chaque point de E_2.

L'ensemble des isogénies d'une courbe E dans elle-même forme un anneau, appelé *anneau des endomorphismes de* E et noté $\mathrm{End}(E)$.

Exemple.

1. Etant donnée la loi de groupe commutative sur E, on peut définir l'endomorphisme *multiplication par* m $[m] \in End(E)$ qui consiste à additionner un point m fois à lui-même. Pour tout $m \in \mathbf{Z}$, $\deg[m] = m^2$.
2. Soient E une courbe elliptique définie sur $\mathbf{F}_q$ et $\sigma \in Gal(\mathbf{F}_q/\mathbf{F}_p)$ tel que $\sigma(x) = x^p$. On note E^σ la courbe obtenue en appliquant σ aux coefficients de E. Le *p-ième morphisme de Frobenius* Φ_p défini par $\Phi_p(x, y) = (x^p, y^p)$ est une isogénie purement inséparable de E dans E^σ, et $\deg \Phi_p = \deg_i \Phi_p = p$.
3. De façon similaire, pour E une courbe elliptique définie sur $\mathbf{F}_q$, on définit le *q-ième morphisme de Frobenius* $\Phi_q \in \mathrm{End}(E)$, tel que $\Phi_q(x, y) = (x^q, y^q)$. Cet endomorphisme est purement inséparable de degré q.

A chaque isogénie non constante $\varphi : E_1 \to E_2$ correspond une unique isogénie $\hat{\varphi} : E_2 \to E_1$, appelée *isogénie duale*, telle que $\hat{\varphi} \circ \varphi = [\deg \varphi] \in \mathrm{End}(E_1)$ et $\varphi \circ \hat{\varphi} = [\deg \varphi] \in \mathrm{End}(E_2)$.

Les propriétés remarquables de ces isogénies sont données dans [Sil86] (section III.6). Elles permettent en particulier de donner une bonne description des points de m-torsion, et pourront s'avérer utiles dans la construction de couplages, dits symétriques ou *self-pairings*.

1.3.2 Groupes de torsion

On note $E[m] = \ker([m])$ le sous-groupe des *points de m-torsion*. De la même façon, on définit pour K extension quelconque du corps k, le sous-groupe $E(K)[m]$ des points *K-rationnels de m-torsion*.

La structure des groupes de torsion est détaillée dans le théorème suivant :

Théorème 1.10 ([Sil86] p. 89)**.**
Soit E une courbe elliptique définie sur un corps k et $m \in \mathbf{Z}^*$.
– *Si $char(k) = 0$ ou $char(k) \wedge m = 1$, alors*

$$E[m] \simeq (\mathbf{Z}/m\mathbf{Z}) \times (\mathbf{Z}/m\mathbf{Z}).$$

– *Si $char(k) = p$, alors*
  – *ou bien $E[p^e] \simeq \{O\}$ pour tout $e \in \mathbf{N}^*$,*
  – *ou bien $E[p^e] \simeq \mathbf{Z}/p^e\mathbf{Z}$ pour tout $e \in \mathbf{N}^*$.*

De ce théorème, on peut déduire facilement la structure de groupe des points $\mathbf{F}_q$-rationnels d'une courbe E définie sur $\mathbf{F}_q$:

Corollaire 1.11. *Soit E une courbe elliptique définie sur $\mathbf{F}_q$, alors*

$$E(\mathbf{F}_q) \simeq \mathbf{Z}/n_1\mathbf{Z} \times \mathbf{Z}/n_2\mathbf{Z}$$

avec $n_1|n_2$ et $n_1|q-1$.

Démonstration. $E(\mathbf{F}_q)$ étant un groupe commutatif, il est de la forme $\mathbf{Z}/n_1\mathbf{Z} \times \ldots \times \mathbf{Z}/n_k\mathbf{Z}$, où $n_1|\ldots|n_k$. S'il existe $m \in \mathbf{Z}^*$ premier à la caractéristique p de $\mathbf{F}_q$, tel que $m|n_1$, on compte alors m^k points de m-torsion, ce qui impose d'après le théorème que $k \leq 2$. De même, si $p|n_1$, avec le théorème, on a nécessairement $k \leq 1$. Dans tous les cas, $E(\mathbf{F}_q)$ est de la forme $\mathbf{Z}/n_1\mathbf{Z} \times \mathbf{Z}/n_2\mathbf{Z}$, où $n_1|n_2$.

Il reste à voir que $n_1|q-1$. On utilise pour cela le fait suivant (démontré dans la section 2.2 à l'aide du couplage de Weil) :

Lemme 1.12 (cf lemme 2.3)**.**
Soit E une courbe elliptique définie sur $\mathbf{F}_q$ ($q = p^d$), $e \geq 1$ un entier quelconque, et m un entier premier à p. Alors :

$$E[m] \subset E(\mathbf{F}_{q^e}) \Rightarrow \mu_m \subset \mathbf{F}_{q^e}^*$$

où μ_m désigne l'ensemble des racines m-ièmes de l'unité de $\overline{\mathbf{F}_q}$.

En particulier, comme $E[n_1] \subset E(\mathbf{F}_q)$, on a $n_1 | q-1$. □

1.3.3 Structure de l'anneau des endomorphismes

Il est intéressant pour la construction de certains couplages dits "self-pairing" de bien connaître la structure de $\mathrm{End}(E)$. Avec les exemples donnés ci-dessous (voir 1.3.1), il est clair que celui-ci contient au moins un sous-anneau isomorphe à $\mathbf{Z}$ (pour tout $m \in \mathbf{Z}$, $[m] \in \mathrm{End}(E)$), et que cette inclusion est stricte pour les corps finis (si E définie sur $\mathbf{F}_q$, $\Phi_q \in \mathrm{End}(E)$).

Le théorème qui suit résume les différentes possibilités pour la structure de $\mathrm{End}(E)$:

Théorème 1.13 ([Sil86] p. 102)**.**
L'anneau des endomorphismes d'une courbe elliptique est soit $\mathbf{Z}$*, soit un ordre dans un corps quadratique imaginaire, soit un ordre dans une algèbre de quaternions.*

Lorsque $\mathrm{End}(E)$ est strictement plus gros que $\mathbf{Z}$, on dit que la courbe E est à *multiplication complexe*. En particulier, si la courbe est définie sur un corps fini, $\mathrm{End}(E)$ contient également l'endomorphisme de Frobenius[1] et donc la courbe est toujours à multiplication complexe.

On détaillera dans la suite la relation étroite qu'il existe entre la structure de l'anneau d'endomorphismes d'une courbe elliptique définie sur un corps fini de caractéristique p, et celle du groupe des points de p-torsion. On commence par donner quelques résultats généraux concernant le calcul de la cardinalité d'une courbe elliptique définie sur corps fini.

1.4 Courbes elliptiques définies sur des corps finis

Dans tout ce qui suit, E désigne une courbe elliptique définie sur $\mathbf{F}_q$, avec $q = p^d$ et p premier, et $\Phi_{p^r} : E \to E^{(p^r)}$ le p^r-ième Frobenius.

1.4.1 Conjecture de Weil et borne de Hasse

Théorème 1.14 (Weil, [Sil86] V.2)**.**
Pour tout $\varphi \in End(E)$*, il existe un unique entier* $t \in \mathbf{Z}$ *appelé* trace de φ *tel que* φ *vérifie dans* $End(E)$ *l'équation* $\varphi^2 - [t] \circ \varphi + [\deg \varphi] = 0$.

En particulier, le q*-ième morphisme de Frobenius vérifie*

$$\Phi_q^2 - [\mathrm{Tr}(\Phi_q)] \circ \Phi_q + [q] = 0$$

Le polynôme $\chi(\Phi_q)(X) = X^2 - \mathrm{Tr}(\Phi_q)X + q$ *est appelé* polynôme caractéristique de Φ_q *et son discriminant est négatif.*

Ce théorème s'obtient comme application des résultats connus sous le nom de *conjecture de Weil* pour les courbes elliptiques.

En remarquant que les points rationnels de la courbe sont exactement ceux laissés fixes par l'action du q-ième Frobenius, on retrouve que

$$\#E(\mathbf{F}_q) = \#\ker([1] - \Phi_q) = \deg_s([1] - \Phi_q) = \deg([1] - \Phi_q)$$

[1] Il est possible que l'endomorphisme de Frobenius soit égal à l'endomorphisme de multiplication par un entier $m \in \mathbf{Z}$, mais dans ce cas, E est supersingulière et $\mathbf{Z} \subsetneq \mathrm{End}(E)$.

car $[1] - \Phi_q$ est séparable ([Sil86] corollaire III.5.5) Par ailleurs, en utilisant le polynôme caractéristique de Φ_q, on trouve que

$$\chi([1] - \Phi_q)(X) = X^2 + (\mathrm{Tr}(\Phi_q) - 2)X + (q + 1 - \mathrm{Tr}(\Phi_q))$$

en particulier

$$\#E(\mathbf{F}_q) = \deg([1] - \Phi_q) = \chi(\Phi_q)(1) = q + 1 - \mathrm{Tr}(\Phi_q). \tag{1}$$

On a une bonne approximation du nombre de points rationnels de la courbe donnée par la borne de Hasse :

Théorème 1.15 (Hasse)**.**

$$|\mathrm{Tr}(\Phi_q)| = |\#E - (q + 1)| \leq 2\sqrt{q}$$

De plus, les cardinalités de E sur les différentes extensions de $\mathbf{F}_q$ sont liées :

Proposition 1.16. *Si α, β sont les racines complexes de $X^2 - \mathrm{Tr}(\Phi_q)X + q$, alors*

$$\#E(\mathbf{F}_{q^k}) = q^k + 1 - (\alpha^k + \beta^k)$$

Remarque 1.17. D'un point de vue pratique, il est intéressant de constater que $tr_n := \mathrm{Tr}(\Phi_{q^n}) = \alpha^n + \beta^n$ vérifie naturellement la relation de récurrence :

$$tr_{n+2} = tr_1\, tr_{n+1} - q\, tr_n$$

et qu'il est donc facile de calculer $\#E(\mathbf{F}_{q^k}) = q^k + 1 - tr_k$ connaissant $tr_1 = q + 1 - (\#E(\mathbf{F}_q))$.

Le résultat qui suit donne le nombre de points rationnels de la tordue $\tilde{E}$ de E :

Théorème 1.18. *Soit $t = q + 1 - \#E(\mathbf{F}_q)$. Alors*

$$\#\tilde{E}(\mathbf{F}_q) = q + 1 + t$$

Démonstration.
Soient v un résidu non quadratique de $\mathbf{F}_q$ et

$$(E) : y^2 = x^3 + ax + b \qquad (\tilde{E}) : y^2 = x^3 + av^{-2}x' + bv^{-3}$$

les équations réduites de E et $\tilde{E}$.

Si on note $g(X) = X^3 + aX + b$ et $h(X) = X^3 + av^{-2}X + bv^{-3}$, alors on a la relation :

$$h(X) = v^{-3}g(vX)$$

ce qui permet de distinguer trois cas pour le calcul simultané des cardinalités de E et $\tilde{E}$:
– Si $g(vx)$ est un résidu quadratique non nul dans $\mathbf{F}_q$, alors $h(x) = v^{-3}g(vx)$ n'en est pas un. En particulier, $Y^2 = g(vx)$ admet deux solutions distinctes y_1, y_2 et $Y^2 = h(x)$ n'admet pas de solution.
– Si $g(vx)$ n'est un résidu quadratique dans $\mathbf{F}_q$, alors $h(x) = v^{-3}g(vx)$ en est un. En particulier, $Y^2 = g(vx)$ n'admet pas de solution et $Y^2 = h(x)$ admet deux solutions distinctes y_1, y_2.
– Si $g(vx) = 0$, alors $h(x) = 0$, en particulier 0 est la seule solution pour les équations $Y^2 = g(vx)$ et $Y^2 = h(x)$.
L'application $x \mapsto vx$ étant une bijection de $\mathbf{F}_q$, le nombre de points rationnels total de E et $\tilde{E}$ obtenu en prenant toutes les valeurs possibles pour $x \in \mathbf{F}_q$ et en rajoutant le point à l'infini appartenant aux deux courbes, est :

$$\#E(\mathbf{F}_q) + \#\tilde{E}(\mathbf{F}_q) = 2q + 2$$

□

Remarque 1.19. On donne ici une autre preuve permettant de voir facilement qu'une courbe elliptique et sa tordue ont même trace au signe près.

Si on note $\tilde{\Phi}_q$ le q-ième Frobenius sur $\tilde{E}$ et $\tilde{\alpha}, \tilde{\beta}$ les racines complexes de $X^2 - \mathrm{Tr}(\tilde{\Phi}_q)X + q$, par $\mathbf{F}_{q^2}$-isomorphisme entre E et $\tilde{E}$, on a :

$$\alpha^2 + \beta^2 = \tilde{\alpha}^2 + \tilde{\beta}^2$$

Comme par ailleurs $\alpha\beta = \tilde{\alpha}\tilde{\beta} = q$, on en déduit que

$$(\alpha + \beta)^2 = (\tilde{\alpha} + \tilde{\beta})^2 \Rightarrow \mathrm{Tr}(\Phi_q) = \pm\,\mathrm{Tr}(\tilde{\Phi}_q).$$

1.4.2 Courbes supersingulières et ordinaires

Suivant la structure de End(E) et de $E[p]$, on distingue deux types de courbes :

Théorème 1.20 ([Sil86] III.3)**.**

1. *On dit que E est* supersingulière *si l'une des conditions équivalentes suivantes est vérifiée :*
   - $E[p^r] = \{O\}$ *pour tout $r \geq 1$ (en particulier, il n'y a pas de points de p-torsion).*
   - $\hat{\Phi}_{p^r}$ *est purement inséparable pour tout $r \geq 1$.*
   - $[p] : E \to E$ *est purement inséparable et $j \in \mathbf{F}_{p^2}$.*
   - *End(E) est un ordre dans une algèbre de quaternions, en particulier End(E) est un $\mathbf{Z}$-module de rang 4.*
2. *On dit que E est* ordinaire *dans tous les autres cas, et on a alors*

$$E[p^r] \simeq \mathbf{Z}/p^r\mathbf{Z}.$$

Il est également possible de caractériser les courbes supersingulières en considérant la trace du q-ième Frobenius $\Phi_q \in End(E)$:

Propriété 1.21.
E est supersingulière si et seulement si $\mathrm{Tr}(\Phi_q) = 0 \mod p$

Démonstration. On rappelle que Φ_q est tel que $\Phi_q^2 - [\mathrm{Tr}(\Phi_q)] \circ \Phi_q + [q]$ est nul sur E. En prenant l'isogénie duale de cet endomorphisme, on trouve que l'endomorphisme $\hat{\Phi}_q^2 - [\mathrm{Tr}(\Phi_q)] \circ \hat{\Phi}_q + [q]$ est également nul sur E, et donc $\mathrm{Tr}(\Phi_q) = \mathrm{Tr}(\hat{\Phi}_q)$.

Par conséquent,

$$\begin{aligned}(\Phi_q + \hat{\Phi}_q)^2 &= \Phi_q^2 + [2q] + \hat{\Phi}_q^2 \\ &= [\mathrm{Tr}(\Phi_q)] \circ \Phi_q - [q] + [2q] + [\mathrm{Tr}(\hat{\Phi}_q)] \circ \hat{\Phi}_q - [q] \\ &= [\mathrm{Tr}(\Phi_q)] \circ (\Phi_q + \hat{\Phi}_q)\end{aligned}$$

On distingue alors 2 cas :
- $\Phi_q + \hat{\Phi}_q$ n'est pas l'endomorphisme nul, en particulier il est surjectif et donc $\Phi_q + \hat{\Phi}_q = [\mathrm{Tr}(\Phi_q)]$. Si ω est la différentielle invariante sur E, on a alors

  $$(\Phi_q + \hat{\Phi}_q)^*\omega = [\mathrm{Tr}(\Phi_q)]^*\omega = (\mathrm{Tr}\,\Phi_q)\omega$$

  et comme Φ_q n'est pas séparable, $(\Phi_q + \hat{\Phi}_q)^*\omega = (\Phi_q)^*\omega + (\hat{\Phi}_q)^*\omega = (\hat{\Phi}_q)^*\omega$. Ainsi, on a l'équivalence

  $$\begin{aligned}\deg_s[p] = p &\Leftrightarrow \deg_s[q] = q \\ &\Leftrightarrow \deg_s \hat{\Phi}_q = q \text{ (Φ_q étant purement inséparable et } \Phi_q \circ \hat{\Phi}_q = [q]) \\ &\Leftrightarrow \hat{\Phi}_q \text{ est séparable} \\ &\Leftrightarrow (\hat{\Phi}_q)^*\omega \neq 0 \\ &\Leftrightarrow (\mathrm{Tr}\,\Phi_q)\omega \neq 0 \\ &\Leftrightarrow \mathrm{Tr}\,\Phi_q \neq 0 \bmod p\end{aligned}$$

- $\Phi_q + \hat{\Phi}_q = 0$, i.e. $\Phi_q = -\hat{\Phi}_q$, alors avec le polynôme caractéristique commun à Φ_q et $\hat{\Phi}_q$ on trouve : $\hat{\Phi}_q^2 + [\mathrm{Tr}(\Phi_q)] \circ \hat{\Phi}_q + [q] = 0 \Rightarrow [2\,\mathrm{Tr}\,\Phi_q] \circ \Phi_q = 0 \Rightarrow [2\,\mathrm{Tr}\,\Phi_q] = 0$, en particulier $\mathrm{Tr}\,\Phi_q = 0$.

□

1.4.3 Anneaux d'endomorphismes et corps quadratiques

Soit E une courbe ordinaire définie sur $\mathbf{F}_q$.

D'après les théorèmes 1.13 et 1.20, il existe un isomorphisme φ entre $\mathrm{End}(E)$ et un ordre dans un corps quadratique imaginaire $K \subset \mathbf{C}$.

Connaissant l'équation caractéristique $X^2 - tX + q = 0$ du Frobenius Φ_q, il est possible de déterminer précisément K :
on a nécessairement $z = \varphi(\Phi_q) \in \mathbf{C}$ qui vérifie $z^2 - tz + q = 0$, i.e. $z = \dfrac{t \pm \sqrt{t^2 - 4q}}{2}$. Si on note

$$4q - t^2 = y^2 D$$

où D est sans facteurs carrés, on en déduit :

$$\mathrm{End}(E) \hookrightarrow \mathbf{Q}(\sqrt{-D})$$

Cette remarque est à la base de la méthode de la multiplication complexe, voir section 4.3.

2 Couplage de Weil et de Tate

On considère une courbe elliptique E définie sur $\mathbf{F}_q$, avec $q = p^r$ (p premier) et n un entier premier à p. On note $\mu_n \subset \overline{\mathbf{F}_q}$ l'ensemble des racines n-ièmes de l'unité.

2.1 Couplage de Weil : définition, propriétés

Soient $P, Q \in E[n]$ deux points de n-torsion, et D_P, D_Q deux diviseurs tels que $D_P \sim (P) - (O)$, $D_Q \sim (Q) - (O)$ et $\mathrm{supp}(D_P) \cap \mathrm{supp}(D_Q) = \emptyset$. On peut toujours construire de tels diviseurs en considérant $D_P = (P) - (O)$ et D_Q de la forme $(Q+S) - (S)$ avec $S \in E$ (en effet, comme $(Q+S) - (Q) - (S) + (O) \sim 0$, on a $(Q+S) - (S) \sim (Q) - (O)$).

Puisque P et Q sont d'ordre n, d'après le lemme 1.8, il existe deux fonctions $f_P, f_Q \in \overline{\mathbf{F}_q}(E)$ définies à constantes près, telles que $\mathrm{div}\,(f_P) = n(P) - n(O)$ et $\mathrm{div}\,(f_Q) = n(Q) - n(O)$.

Grâce à la loi de réciprocité de Weil, on montre que le quotient $e_n(P,Q) = \frac{f_P(D_Q)}{f_Q(D_P)}$ est bien défini (autrement dit qu'il est indépendant du choix de $D_Q \sim (Q) - (O)$ et des constantes choisies pour f_P et f_Q) et que c'est une racine n-ième de l'unité :

. Soit $g \in \overline{\mathbf{F}_q}(E)$, en utilisant la loi de réciprocité de Weil, on vérifie :

$$\frac{f_P(D_Q + \mathrm{div}\,(g))}{f_Q(D_P) g^n(D_P)} = \frac{f_P(D_Q) f_P(\mathrm{div}\,(g))}{f_Q(D_P) g(nD_P)} = \frac{f_P(D_Q) f_P(\mathrm{div}\,(g))}{f_Q(D_P) g(\mathrm{div}\,(f_P))} = \frac{f_P(D_Q)}{f_Q(D_P)}$$

. Soit $c \in \overline{\mathbf{F}_q}$:

$$\frac{(cf_P)(D_Q)}{f_Q(D_P)} = \frac{c^{\deg D_Q}\, f_P(D_Q)}{f_Q(D_P)} = \frac{f_P(D_Q)}{f_Q(D_P)}$$

. Avec la loi de réciprocité de Weil, on montre facilement que $e_n(P,Q) \in \mu_n$:

$$\left(\frac{f_P(D_Q)}{f_Q(D_P)}\right)^n = \frac{f_P(nD_Q)}{f_Q(nD_P)} = \frac{f_P(\mathrm{div}\,(f_Q))}{f_Q(\mathrm{div}\,(f_P))} = 1$$

Définition 2.1 (Couplage de Weil)**.**
On définit le couplage de Weil de P et Q par :

$$\begin{array}{rcl} e_n : E[n] \times E[n] & \to & \mu_n \\ (P,Q) & \mapsto & e_n(P,Q) = \frac{f_P(D_Q)}{f_Q(D_P)} \end{array}$$

Ce couplage vérifie les propriétés suivantes ([Sil86] p. 96-98) :

Proposition 2.2.

1. Bilinéarité *: Pour tous points* $P, Q, P_1, P_2, Q_1, Q_2 \in E[n]$

$$e_n(P_1 + P_2, Q) = e_n(P_1, Q)\, e_n(P_2, Q)$$

$$e_n(P, Q_1 + Q_2) = e_n(P, Q_1)\, e_n(P, Q_2)$$

2. Antisymétrie *: Pour tous points* $P, Q \in E[n]$,

$$e_n(Q, P) = e_n(P, Q)^{-1}$$

   En particulier $e_n(P, P) = 1$ *pour tout* $P \in E[n]$.
3. Non dégénérescence *: Pour tout* $P \in E[n]$, $P \neq O$, *il existe* $Q \in E[n]$ *tel que* $e_n(P, Q) \neq 1$.
4. Invariance par action de Galois *: Si* $\sigma \in Gal(\overline{\mathbf{F}_q})$, *alors* $e_n(P^\sigma, Q^\sigma) = \sigma(e_n(P, Q))$.
5. Dualité *: Pour tous points* $P, Q \in E[n]$, *et tout endomorphisme* $\varphi \in End(E)$,

$$e_n(P, \varphi(Q)) = e_n(\hat{\varphi}(P), Q)$$

2.2 Couplage de Weil et points de l-torsion

Soit $l \,|\, \#E(\mathbf{F}_q)$ premier différent de p.

Le couplage de Weil permet de voir que pour obtenir toute la l-torsion, il faut considérer une extension suffisamment grande de $\mathbf{F}_q$:

Lemme 2.3. *Soit* $d \geq 1$ *un entier quelconque, alors :*

$$E[l] \subset E(\mathbf{F}_{q^d}) \Rightarrow \mu_l \subset \mathbf{F}_{q^d}^*$$

Démonstration. Le couplage de Weil étant non dégénéré, il existe deux points de l-torsion $P, Q \in E[l] \subset E(\mathbf{F}_{q^d})$ tels que $e_l(P, Q) \neq 1$. Pour calculer le couplage de Weil de $P, Q \in E(\mathbf{F}_{q^d})$, on peut choisir $D_P, D_Q \in Div^0_{\mathbf{F}_{q^d}}(E)$ et $f_P, f_Q \in \mathbf{F}_{q^d}(E)$, on a alors $e_l(P, Q) \in \mathbf{F}_{q^d}$.

Comme l est premier, $e_l(P, Q)$ est une racine primitive qui engendre μ_l, en particulier $\mu_l \subset \mathbf{F}_{q^d}^*$. □

Il est alors naturel de s'intéresser à la plus petite extension de $\mathbf{F}_q$ contenant μ_n :

Définition 2.4 (Degré d'immersion)**.** *Le* degré d'immersion *est le plus petit entier* k *tel que* $l | q^k - 1$, *autrement dit* $\mathbf{F}_{q^k}$ *est la plus petite extension de* $\mathbf{F}_q$ *contenant* μ_l.

Lorsque le degré de plongement $k > 1$, on a la réciproque du lemme 2.3 ([Gal05] p.192) :

Proposition 2.5 (Balasubramanian-Koblitz)**.**
Si $k > 1$, *alors*

$$E(\mathbf{F}_{q^k})[l] = E[l] \simeq \mathbf{Z}/l\mathbf{Z} \times \mathbf{Z}/l\mathbf{Z}$$

Le couplage de Weil permet également de montrer un autre résultat intéressant sur les endomorphismes de E :

Lemme 2.6. *Si* $\varphi \in End(E)$, *alors la restriction de* φ *à* $E[l] \simeq \mathbf{Z}/l\mathbf{Z} \times \mathbf{Z}/l\mathbf{Z}$ *est une application* $\mathbf{Z}/l\mathbf{Z}$*-linéaire dont le déterminant vérifie :*

$$\deg \varphi = \det \varphi \mod l$$

Démonstration. Soient P, Q deux points engendrant la l-torsion et $Mat_\varphi = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$ la matrice représentant l'endomorphisme $\varphi \in \mathrm{End}(E[l])$ dans la base $\langle P, Q\rangle$. Le couplage de Weil étant antisymétrique et φ et $\hat{\varphi}$ étant adjoints pour le couplage de Weil, on a d'une part :

$$e_l(\varphi(P), \varphi(Q)) = e_l(P,P)^{ab}\, e_l(P,Q)^{ad}\, e_l(Q,P)^{bc}\, e_l(Q,Q)^{cd} = e_l(P,Q)^{ad-bc} = e_l(P,Q)^{\det\varphi}$$

et d'autre part :

$$e_l(\varphi(P), \varphi(Q)) = e_l(\hat{\varphi}\circ\varphi(P), Q) = e_l(P,Q)^{\deg\varphi}$$

Comme $e_l(P,Q) \neq 1$ est une racine primitive, c'est un élément d'ordre l, en particulier

$$\deg\varphi = \det\varphi \mod l.$$

□

2.3 Couplage de Tate

De façon similaire au couplage de Weil, on va définir le couplage de Tate comme une application bilinéaire à valeurs dans l'ensemble des racines n-ièmes de l'unité.

On considère des points $P \in E(\mathbf{F}_{q^k})[n]$ et $Q \in E(\mathbf{F}_{q^k})$. Alors il existe une fonction $f_P \in \mathbf{F}_{q^k}(E)$ telle que $\mathrm{div}\,(f_P) = n(P) - n(O)$ (lemme 1.8) et un diviseur $D_Q \sim (Q) - (O)$, $D_Q \in Div^0_{\mathbf{F}_{q^k}}(E)$ tel que

$$\mathrm{supp}(D_Q) \cap \mathrm{supp}(\mathrm{div}\,(f_P)) = \emptyset.$$

On définit ensuite comme en 1.2, une évaluation de f_P en $cl((Q)-(O)) \in Pic^0_{\mathbf{F}_{q^k}}(E)$ à valeurs dans le groupe multiplicatif $\mathbf{F}^*_{q^k}/(\mathbf{F}^*_{q^k})^n$, en s'assurant que cette définition est bien indépendante du choix du représentant D_Q de $cl((Q)-(O))$:

soit $g \in \mathbf{F}_{q^k}(E)$ et $D = D_Q + \mathrm{div}\,(g)$, en utilisant la loi de réciprocité de Weil, on vérifie :

$$f_P(D) = f_P(D_Q) f_P(\mathrm{div}\,(g)) = f_P(D_Q) g(\mathrm{div}\,(f_P)) = f_P(D_Q)\left(\frac{g(P)}{g(O)}\right)^n \sim f_P(D_Q) \in \mathbf{F}^*_{q^k}/(\mathbf{F}^*_{q^k})^n$$

On a alors le couplage suivant :

$$\begin{array}{ccc} E(\mathbf{F}_{q^k})[n] \times E(\mathbf{F}_{q^k}) & \to & \mathbf{F}^*_{q^k}/(\mathbf{F}^*_{q^k})^n \\ (P,Q) & \mapsto & \langle P,Q\rangle_n = f_P(D_Q) \end{array}$$

où f_P et D_Q sont obtenus à partir de la construction précisée ci-dessus.

Dans la pratique, on se débarrasse du quotient par $(\mathbf{F}^*_{q^k})^n$ en prenant $f_P(D_Q)^{\frac{q^k-1}{n}}$:

Soit $\varphi : \mathbf{F}^*_{q^k} \to \mu_n$ le morphisme de groupes donné par l'élévation à la puissance $\frac{q^k-1}{n}$, on va montrer que $(\mathbf{F}^*_{q^k})^n = \ker\varphi$, en particulier que φ induit un isomorphisme entre les groupes $\mathbf{F}^*_{q^k}/(\mathbf{F}^*_{q^k})^n$ et μ_n.
L'inclusion $(\mathbf{F}^*_{q^k})^n \subset \ker\varphi$ est claire. Réciproquement, comme les éléments du sous-groupe cyclique $\ker\varphi$ ont tous un ordre qui divise $\frac{q^k-1}{n}$, $\#\ker\varphi = \frac{q^k-1}{n}$. Par ailleurs, avec la suite exacte

$$0 \to \mu_n \to \mathbf{F}^*_{q^k} \to (\mathbf{F}^*_{q^k})^n \to 0$$

on déduit facilement que $\#(\mathbf{F}^*_{q^k})^n = \frac{q^k-1}{n}$.

On obtient ainsi un couplage, encore noté $\langle P,Q\rangle_n$, à valeurs dans l'ensemble $\mu_n \subset \mathbf{F}^*_{q^k}$ des racines n-ièmes de l'unité.

Ce couplage a les propriétés suivantes ([Gal05] p. 188) :

Proposition 2.7.

1. Bilinéarité *: Pour tous points* $P, P_1, P_2 \in E(\mathbf{F}_{q^k})[n]$ *et* $Q, Q_1, Q_2 \in E(\mathbf{F}_{q^k})$,

$$\langle P_1 + P_2, Q\rangle_n = \langle P_1, Q\rangle_n \langle P_2, Q\rangle_n$$

$$\langle P, Q_1 + Q_2\rangle_n = \langle P, Q_1\rangle_n \langle P, Q_2\rangle_n$$

2. Non dégénérescence *: Pour tout* $P \in E(\mathbf{F}_{q^k})[n]$, $P \neq O$, *il existe* $Q \in E(\mathbf{F}_{q^k})$ *tel que* $\langle P, Q\rangle_n \neq 1$. *Réciproquement pour tout* $Q \in E(\mathbf{F}_{q^k})$, $Q \notin nE(\mathbf{F}_{q^k})$, *il existe* $P \in E(\mathbf{F}_{q^k})[n]$ *tel que* $\langle P, Q\rangle_n \neq 1$.
3. Invariance par action de Galois *: Si* $\sigma \in Gal(\mathbf{F}_{q^k}/\mathbf{F}_q)$, *alors* $\langle P^\sigma, Q^\sigma\rangle_n = \sigma(\langle P, Q\rangle_n)$.
4. Dualité *: Pour tous points* $P \in E(\mathbf{F}_{q^k})[n]$ *et* $Q \in E(\mathbf{F}_{q^k})$, *et tout endomorphisme* $\varphi \in End(E)$,

$$\langle P, \varphi(Q)\rangle_n = \langle \hat{\varphi}(P), Q\rangle_n$$

Avec la propriété de bilinéarité, on remarque de plus que pour tout point $P \in E(\mathbf{F}_{q^k})[n]$, l'application $\langle P, .\rangle_n$ est dégénérée sur $nE(\mathbf{F}_q^k)$:

si $R \in E(\mathbf{F}_q^k)$, alors $\langle P, nR\rangle_n = \langle nP, R\rangle_n = \langle O, R\rangle_n = 1$.

Le couplage de Tate est ainsi obtenu en restreignant à $E(\mathbf{F}_{q^k})[n] \times E(\mathbf{F}_{q^k})/nE(\mathbf{F}_{q^k})$ le domaine de définition de $\langle ., .\rangle_n$:

Définition 2.8 (Couplage de Tate)**.**

$$\begin{array}{rccccc} E(\mathbf{F}_{q^k})[n] \times E(\mathbf{F}_{q^k})/nE(\mathbf{F}_{q^k}) & \to & \mathbf{F}_{q^k}^*/(\mathbf{F}_{q^k}^*)^n & \xrightarrow{\sim} & \mu_n \subset \mathbf{F}_{q^k}^* \\ (P, Q) & \mapsto & f_p(D_Q) & \mapsto & \langle P, Q\rangle_n = f_P(D_Q)^{\frac{q^k-1}{n}} \end{array}$$

2.4 Algorithme de Miller

L'algorithme de Miller rend possible les applications des couplages en cryptographie, dans la mesure où, lorsque le degré de plongement k n'est pas trop grand, il permet un calcul efficace de ceux-ci.

2.4.1 Principe de l'algorithme

On rappelle que les couplages de Tate et Weil requièrent le calcul de la fonction $f_P \in \mathbf{F}_q(E)$ telle que $\mathrm{div}\,(f_P) = l(P) - l(O)$. Celui-ci peut se faire de façon incrémentale en utilisant la loi de groupe géométrique. Le diviseur $D = (P) + (Q) - 2(O)$ peut en effet se réécrire sous la forme $D = (P + Q) - (O) + \mathrm{div}\,(h)$ avec h qui se calcule aisément en considérant les fonctions affines ℓ et v telles que $\ell = 0$ est l'équation de la droite passant par P et Q et $v = 0$ est celle de la droite passant par $P + Q$ et O :

$\mathrm{div}\,(\ell) = (P) + (Q) + (-(P + Q)) - 3(O)$ et $\mathrm{div}\,(v) = (P + Q) + (-(P + Q)) - 2(O)$, donc on a

$$\begin{aligned} (P) + (Q) - 2(O) &= (P + Q) - (O) + \mathrm{div}\,(\ell) - \mathrm{div}\,(v) \\ &= (P + Q) - (O) + \mathrm{div}\left(\frac{\ell}{v}\right) \end{aligned}$$

On peut ainsi trouver de proche en proche pour tout $i \in \mathbf{Z}$ des fonctions f_i telles que

$$i(P) - i(O) = ([i]P) - (O) + \mathrm{div}\,(f_i) \tag{2}$$

et en particulier pour $i = l$ on retrouve $f_P = f_l$.

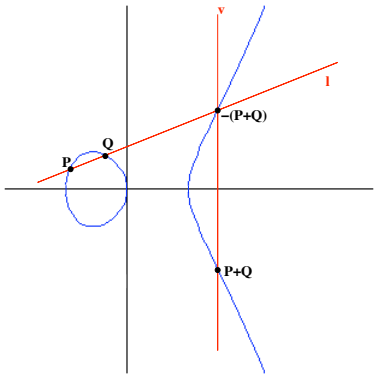


FIG. 1 – Courbe elliptique d'équation réduite $y^2 = x^3 + ax + b$

De façon plus détaillée, connaissant f_i et f_j vérifiant l'équation (2), on construit f_{i+j} en constatant que si ℓ et v sont des fonctions affines telles que $\ell = 0$ est l'équation de la droite passant par $[i]P$ et $[j]P$ et $v = 0$ est celle de la droite passant par $[i+j]P$ et O, alors :

$$\begin{aligned}(i+j)(P) - (i+j)(O) &= ([i]P) + ([j]P) - 2(O) + \operatorname{div}(f_i f_j) \\ &= ([i+j]P) - (O) + \operatorname{div}\left(\frac{\ell}{v}\right) + \operatorname{div}(f_i f_j) \\ &= ([i+j]P) - (O) + \operatorname{div}\left(\frac{f_i f_j \ell}{v}\right)\end{aligned}$$

En particulier, $f_{i+j} = \dfrac{f_i f_j \ell}{v}$ convient.

En initialisant la suite à $f_0 = 1$ et en utilisant une chaîne d'addition pour l, comme par exemple celle utilisée dans l'algorithme *double and add*, on obtient alors l'algorithme de Miller ([Mil86]) :

Alg. 1 Algorithme de Miller

ENTRÉE : $P \in E[l]$, $l = (l_{k-1}...l_0)_2$
SORTIE : f telle que $\operatorname{div}(f) = l(P) - l(O)$

```
f ← 1
T ← P
pour i = k − 1 à 0 faire
        ℓ ← tangente en T
        v ← droite passant par 2T et O
        f ← f² ℓ/v
        T ← 2T
        si l_i = 1 alors
                ℓ ← droite passant par P et T
                v ← droite passant par T + P et O
                f ← f ℓ/v
                T ← T + P
        fin si
fin pour
retourner f
```

2.4.2 Raffinements

Les couplages de Tate et Weil s'obtiennent en évaluant la fonction f_P sur le diviseur D_Q, soit en calculant le quotient $\frac{f_P(Q+S)}{f_P(S)}$ où $S \in E(\mathbf{F}_{q^k})$ est bien choisi. On peut donc réduire la quantité de mémoire requise par l'algorithme en évaluant à chaque étape la fonction f en $Q+S$ et S. On peut également éviter les divisions faites à chaque étape, en les ramenant à une seule division en fin d'algorithme.

Concernant le choix de S, on doit cependant prendre quelques précautions. Soit n_i le i-ème entier apparaissant dans la chaîne d'addition de l, on note f_{n_i}, ℓ_{n_i} et v_{n_i} les fonctions f, ℓ et v obtenues à la i-ème étape de l'algorithme. On a donc $\operatorname{div}(f_{n_i}) = n_i(P) - ([n_i]P) - (n_i - 1)(O)$, ce qui oblige pour l'évaluation à prendre S et $Q+S$ différents de O, P et $[n_i]P$. Un autre problème peut intervenir lors de l'évaluation des fonctions ℓ et v à chaque tour de boucle : par exemple le diviseur de ℓ_{n_i} peut être égal soit à $2(T) + (-[2]T) - 3(O)$, soit à $(P) + (T) + (-(P+T)) - 3(O)$, en particulier $-[n_i]P$ est un zéro de ℓ_{n_i}. Avec un raisonnement similaire pour v, on en déduit que $Q+S$ et S doivent également être différents de $-[n_i]P$.

Au total, il y a donc $O(\log_2(l))$ points à éviter pour le choix de S, ce qui lorsque l est grand, n'est pas très contraignant (par exemple si on choisit S dans le sous-groupe à l points engendré par P, la probabilité de prendre un S qui ne convient pas est de l'ordre de $\frac{c\log_2(l)}{l}$, donc négligeable pour l assez grand). On convient donc que S peut être pris au hasard sur $E(\mathbf{F}_{q^k})$ et même dans $E(\mathbf{F}_q)$ pour simplifier au maximum les calculs.

On peut simplifier le calcul du couplage de Weil à l'aide du résultat suivant :

Théorème 2.9 ([Mil04] Prop. 8)**.** *Soient $P, Q \in E[l]$ tels que $P \neq Q$. Alors :*

$$e_l(P,Q) = (-1)^l \frac{f_P(Q)}{f_Q(P)}$$

où f_P et f_Q sont les fonctions obtenues avec l'algorithme de Miller 2.4.1.

De la même façon, on simplifie le calcul du couplage de Tate avec le théorème suivant :

Théorème 2.10. *Soient $P \in E(\mathbf{F}_{q^k})$ et $Q \in E(\mathbf{F}_{q^k})/lE(\mathbf{F}_{q^k})$ tel que $Q \neq P, O$. Alors*

$$\langle P,Q\rangle_l = f_P(Q)^{\frac{q^k-1}{l}}$$

où f_P est la fonction obtenue avec l'algorithme de Miller 2.4.1.

Démonstration. On note $\tau_Q : R \mapsto R+Q$ la translation par le point Q. Pour tout point $S \in E(\mathbf{F}_{q^k}^*)$ tel que $S \neq O, P, -Q, P-Q$, on a $\langle P,Q\rangle_l = \left(\frac{f_P \circ \tau_Q(S)}{f_P(S)}\right)^{\frac{q^k-1}{l}}$, et

$$\operatorname{div}\frac{f_P \circ \tau_Q}{f_P} = \operatorname{div}(f_P \circ \tau_Q) - \operatorname{div}(f_P) = l(P-Q) - l(-Q) - l(P) + l(O),$$

Soit v l'équation affine de la droite verticale passant par $P-Q$ et ℓ l'équation affine de la droite passant par P et $-Q$, alors

$$\begin{cases} \operatorname{div} v = (P-Q) + (Q-P) - 2(O) \\ \operatorname{div} \ell = (P) + (-Q) + (Q-P) - 3(O) \end{cases}$$

en particulier

$$\operatorname{div}\frac{f_P \circ \tau_Q}{f_P} = \operatorname{div}\left(\frac{v}{\ell}\right)^l$$

Par conséquent,

$$\frac{f_P \circ \tau_Q(S)}{f_P(S)} = c\left(\frac{v(S)}{\ell(S)}\right)^l \Rightarrow \langle P,Q\rangle_l = c^{\frac{q^k-1}{l}} \tag{3}$$

Afin de déterminer la constante c, on considère l'écriture de $\frac{f_P \circ \tau_Q}{f_P} = c\left(\frac{v}{\ell}\right)^l$ dans l'anneau local E_O des germes de fonctions en O. Soit $z = \frac{x}{y}$ une uniformisante au point O, alors

. $v = x - x_{P-Q} = z^{-2}(1 + zu(z))$ où $u \in E_O$

. $\ell = y - \alpha x - \beta = z^{-3}(1 + zv(z))$ où $v \in E_O$

. $f_P \circ \tau_Q = f_P(Q)(1 + zt(z))$ où $t \in E_O$

. $f_P = z^{-l}(f_0 + zf_1(z))$ où $f_0, f_1 \in E_O$.

Si on construit f_P récursivement avec l'algorithme de Miller, à l'aide d'équations de droites de type $x - \gamma = 0$, $y - \delta x - \lambda = 0$, on a nécessairement f_P normalisée en O, i.e. $f_0 = 1$. On en déduit avec (3) :

$$c = f_P(Q)$$

□

Il est donc possible de simplifier l'algorithme de Miller en évaluant la fonction construite en Q à chaque itération.

Remarque 2.11. En pratique, on préfèrera utiliser le couplage de Tate, qui ne requiert qu'un seule évaluation de la fonction f_P en Q, plutôt que le couplage de Weil qui impose une évaluation de f_Q en P et une division supplémentaires.

2.4.3 Implémentation et étude de la complexité

On détaille (alg. 2.4.3) l'algorithme permettant de calculer $f_P(Q)$. Lors de la dernière étape de l'algorithme, il faut faire attention au fait que ℓ est une droite verticale et v la droite à l'infini. Par ailleurs, à la fin de l'algorithme, on doit avoir $T = O$, ce qui permet de vérifier que l'ordre de P est correct.

Remarque 2.12. Il est clair que pour calculer $\frac{f_P(Q+S)}{f_P(S)}$, il ne faut pas appeler deux fois l'algorithme, mais l'adapter pour évaluer f_{n_i} à chaque étape en $Q + S$ et en S.

Complexité de l'algorithme de Miller amélioré :
Les opérations les plus coûteuses dans l'algorithme sont les divisions dans $\mathbf{F}_q$ pour le calcul de λ et les multiplications dans $\mathbf{F}_{q^k}$ pour le calcul des f_{n_i} à chaque étape, ainsi que la division de f_1 par f_2 dans $\mathbf{F}_{q^k}$ à la fin de l'algorithme.

Pour les algorithmes classiques permettant de multiplier deux éléments d'un corps fini $\mathbf{F}_q$, la complexité est en $O((\log q)^\mu)$, où μ dépend de l'algorithme utilisé (typiquement $\mu = \log 3$ pour Karastuba). Une division dans $\mathbf{F}_q$ est donc de complexité en $O((\log q)^{\mu+1})$. Au final, on a donc une complexité pour Miller en

$$O\left(\log l\left((k\log q)^\mu + (\log q)^{\mu+1}\right) + (k\log q)^{\mu+1}\right) = O\left((\log q)^{\mu+1}(\log l + k^{\mu+1})\right)$$

Cette complexité est polynomiale en q et l, mais exponentielle en k. Malheureusement les courbes ayant un petit degré de plongement sont très rares [BK98], ce qui imposera de travailler avec des courbes particulières, dites *courbes bien couplées* (voir section 4).

3 Utilisation des couplages en cryptographie

Les couplages de Weil et Tate ont des propriétés particulièrement intéressantes pour la cryptographie, dans la mesure où on connaît un algorithme efficace pour les calculer. A titre d'exemple, on présente dans ce qui suit une de leurs toutes premières applications (en 1993) à la cryptanalyse ainsi que des exemples célèbres de protocoles les utilisant.

Alg. 2 Algorithme de Miller amélioré

ENTRÉE : $P = (x_1, y_1) \in E(\mathbf{F}_q)[l]$, $Q = (x_2, y_2) \in E(\mathbf{F}_{q^k})[l]$, $l = (l_{k-1}...l_0)_2$
SORTIE : $f(Q)$ où f telle que $\mathrm{div}\,(f) = l(P) - l(O)$
(**Variables :** $T = (x_3, y_3) \in E(\mathbf{F}_q)$, $f_1, f_2, \lambda \in E(\mathbf{F}_{q^k})$)
$f_1 \leftarrow 1$
$f_2 \leftarrow 1$
$T \leftarrow P$
pour $i = k - 1$ à 1 **faire**
    $\lambda \leftarrow$ coefficient de la tangente à E en T
    $f_1 \leftarrow f_1^2(y_2 - \lambda(x_2 - x_3) - y_3)$
    $f_2 \leftarrow f_2^2(x_2 + 2x_3 - \lambda^2)$
    $T \leftarrow 2T$
    si $l_i = 1$ **alors**
        $\lambda \leftarrow$ coefficient de la droite passant par P et T
        $f_1 \leftarrow f_1(y_2 - \lambda(x_2 - x_1) - y_3)$
        $f_2 \leftarrow f_2(x_2 + x_3 + x_1 - \lambda^2)$
        $T \leftarrow T + P$
    fin si
fin pour
si $l_0 = 1$ **alors**
    $\lambda \leftarrow$ coefficient de la tangente à E en T
    $f_1 \leftarrow f_1^2(y_2 - \lambda(x_2 - x_3) - y_3)$
    $f_2 \leftarrow f_2^2(x_2 + 2x_3 - \lambda^2)$
    $T \leftarrow 2T$
    $f_1 \leftarrow f_1(x_2 - x_1)$
    $T \leftarrow T + P$
sinon
    $f_1 \leftarrow f_1^2(x_2 - x_3)$
    $f_2 \leftarrow f_2^2$
    $T \leftarrow 2T$
fin si
retourner $\frac{f_1}{f_2}$

3.1 Attaque de MOV/Frey-Rück

$(E(\mathbf{F}_q), +)$ étant un groupe commutatif, si $r \neq p$ est un diviseur premier du nombre de points rationnels $\#E(\mathbf{F}_q)$ de la courbe, alors il existe un point $P \in E(\mathbf{F}_q)[r]$ d'ordre r.

Avec les mêmes notations que précédemment, on s'intéresse au *problème du logarithme discret* sur $E(\mathbf{F}_q)[r]$:

$$\text{Etant donnés } P, P' \in E(\mathbf{F}_q)[r] \text{ tels que } [m]P = P', \text{ trouver } m.$$

Pour une courbe elliptique quelconque, on ne connaît que des algorithmes génériques, type *Baby step, Giant step* de complexité en calcul exponentielle (en $O(\sqrt{r})$). Grâce aux propriétés de bilinéarité et de non dégénérescence du couplage de Tate, il est possible de transférer ce problème dans le groupe multiplicatif d'un corps fini (attaque de Frey-Rück [FR94] ou attaque de Menezes, Okamoto et Vanstone [MOV93]). En effet,

$$\langle [m]P, Q\rangle = \langle P, Q\rangle^m \in \mu_r \subset \mathbf{F}_{q^k} \text{ où } k \text{ est le degré d'immersion.}$$

Si on choisit Q tel que $\langle P, Q\rangle \neq 1$, on se ramène au problème du log discret dans $\mathbf{F}_{q'}^*$ ($q' = q^k$). Sur un tel groupe, il existe des algorithmes, basés sur le calcul d'index, de complexité sous-exponentielle en $O(e^{c(\log q')^{\frac{1}{3}}(\log\log q')^{\frac{2}{3}}})$.

En particulier pour les courbes admettant des sous-groupes de r-torsion pour lesquels le degré d'immersion k est proche de 1, l'attaque MOV est plus performante que les algorithmes standards.

Menezes, Okamoto et Vanstone ont également montré que le degré de plongement d'une courbe supersingulière est toujours plus petit que 6 (voir section 4.2) ; il faut donc être particulièrement prudent lorsque l'on utilise ce type de courbe. Pour parer à l'attaque MOV, il sera nécessaire en particulier de prendre q grand.

Par exemple, supposons que l'on souhaite travailler avec une courbe supersingulière définie sur $\mathbf{F}_p$, avec $k = 2$. Pour avoir une sécurité de 80 bits, r doit avoir au moins 160 bits afin de contrer les attaques génériques type *Baby-Step/Giant-Step* ; mais p^k doit aussi comporter au moins 1024 bits pour contrer l'attaque MOV, ce qui impose $|p|_2 \geq 512$. Le cofacteur de r dans $\#E(\mathbf{F}_p)$ est donc très grand, ce qui entraine une perte d'efficacité au niveau mémoire, temps de calcul et bande passante.

On décrit dans la suite les schémas cryptographiques à clé publique fondamentaux basés sur des couplages. Dans chaque cas, on précise les propriétés du couplage utilisées, ainsi que les hypothèses standards à faire pour assurer la sécurité.

3.2 Hypothèses de sécurité liées aux couplages

Dans les schémas cryptographiques que l'on va décrire, on utilise essentiellement deux types de couplages non dégénérés :
- les "self-pairings", de la forme $\hat{e} : G_1 \times G_1 \to G_3$ bilinéaire et non dégénéré, où G_1 et G_3 sont deux groupes cycliques d'ordre r premier.
- les couplages asymétriques, plus simples à construire, et qui sont de la forme $e : G_1 \times G_2 \to G_3$ bilinéaire et non dégénéré, où G_1, G_2 et G_3 sont des groupes cycliques d'ordre r premier.

Les courbes elliptiques (et hyperelliptiques) sont pour l'instant les seuls contextes connus dans lesquels de tels couplages sont calculables de façon efficace. Généralement, on prend $G_1 = \langle P\rangle$ où P est un point rationnel de r-torsion d'une courbe elliptique E définie sur $\mathbf{F}_q$ ($q = p^d$, $l \neq p$ premier), $G_2 = \langle Q\rangle$ où $Q \in E(\mathbf{F}_{q^k})$ est un point de r-torsion non multiple de P, et $G_3 = \mu_r \subset \mathbf{F}_{q^k}$ le groupe des racines r-ièmes de l'unité.

On rappelle que pour assurer la sécurité des cryptosystèmes classiques basés sur le calcul du logarithme discret, on utilise un groupe $G = \langle P\rangle$ noté additivement, dans lesquels l'un des trois problèmes suivants au moins est difficile :
- *DDH (Decisional Diffie-Hellman problem) :* étant donnés $P, [a]P, [b]P$ et $[c]P$, déterminer si $ab = c$.

– *CDH (Computational Diffie-Hellman problem) :* étant donnés $P, [a]P$ et $[b]P$, calculer $[ab]P$.
– *DL (Discrete Log problem) :* étant donnés P et $[a]P$, trouver a.
Ces trois problèmes sont clairement classés par ordre de difficulté croissante.

Mais lorsqu'on travaille avec des groupes admettant un couplage, ces problèmes ne sont plus nécessairement appropriés. Si par exemple, on considère pour simplifier les notations des *self-pairings*, c'est-à-dire des applications bilinéaires symétriques non dégénérés $\hat{e} : G_1 \times G_1 \to G_3$ où $G_1 = \langle P \rangle$ est un groupe cyclique noté additivement, le problème DDH sur G_1 devient facile. Il est par contre pertinent d'introduire les problèmes suivants :

– *DBDH (Decisional Bilinear Diffie-Hellman problem) :* étant donnés $P, [a]P, [b]P$ et $[c]P$ dans G_1 et $\hat{e}(P,P)^d$, déterminer si $d = abc$.
– *BDH (Bilinear Diffie-Hellman problem) :* étant donnés $P, [a]P, [b]P$ et $[c]P$ dans G_1, calculer $\hat{e}(P,P)^{abc}$.
– *Inversion problem :* étant donnés P et $\hat{e}([a]P, P)$, trouver $[a]P$.

Remarque 3.1. De la même façon qu'un algorithme permettant de résoudre DL peut être utilisé pour résoudre CDH et DDH, il est possible de trouver des relations entre les complexités de ces différents problèmes :

- $BDH \propto CDH_{G_1}$
  On suppose qu'on connaît un algorithme permettant de résoudre CDH sur G_1. Avec cet algorithme, on peut, étant donnés $P, [a]P, [b]P$ et $[c]P$, calculer $[ab]P$ puis $\hat{e}([ab]P, [c]P) = \hat{e}(P,P)^{abc}$, ce qui permet de résoudre BDH sur $\langle G_1, G_3, \hat{e} \rangle$.
- $BDH \propto CDH_{G_3}$
  Etant donnés $P, [a]P, [b]P$ et $[c]P$, on peut calculer grâce à la bilinéarité $\hat{e}(P,P)^{bc} = \hat{e}([b]P, [c]P)$ et $\hat{e}(P,P)^a = \hat{e}([a]P, P)$ pour en déduire grâce à l'algorithme de résolution de CDH_{G_3} le couplage $\hat{e}(P,P)^{abc}$.
- $BDH \propto Inv$
  Etant donnés $P, [a]P, [b]P$ et $[c]P$, on peut calculer $\hat{e}([a]P, [b]P) = \hat{e}([ab]P, P)$ et en déduire $[ab]P$ grâce à l'algorithme d'inversion. Il est alors facile de calculer $\hat{e}([ab]P, [c]P) = \hat{e}(P,P)^{abc}$.
- $DDH_{G_3} \propto Inv$
  La loi de groupe sur G_3 étant notée multiplicativement, on cherche étant donnés g, g^a, g^b, g^c à déterminer si $g^c = g^{ab}$. Grâce à Inv, on peut déduire de g^a (resp. g^b, g^c) la valeur de $[aP]$ (resp. $[b]P, [c]P$). Avec les propriétés du couplage, il est alors facile de déterminer si $\hat{e}([ab]P, P) = \hat{e}([a]P, [b]P)$ est égal à $\hat{e}([c]P, P)$.

Par contre la réduction $CDH \propto BDH$ est encore un problème ouvert (voir [BF03],[Jou04]).

Il est par ailleurs possible d'adapter ces problèmes au cas où le couplage serait asymétrique. Par exemple, pour un couplage $e : G_1 \times G_2 \to G_3$, on définit le problème suivant :

Co-BDH (Co-bilinear Diffie-Hellman problem) : étant donnés P (resp.Q) un générateur de G_1 (resp. G_2), $[a]P$, $[b]P$, $[a]Q$ et $[c]Q$ calculer $e(P,Q)^{abc}$.

3.3 Distribution non interactive de clés basée sur l'identité

En 2000, Sakai, Ohgishi et Kasahara ([SOK00]) ont mis au point une version non interactive du protocole d'échange de clés en utilisant des couplages. Dans ce contexte, on se donne :
– un système de paramètres $\{G_1, G_3, \hat{e}\}$, où G_1 et G_3 sont des groupes cycliques d'ordre r, et $\hat{e} : G_1 \times G_1 \to G_3$ est une application bilinéaire, symétrique et non dégénérée.
– une fonction de hachage $H_1 : \{0;1\}^* \to G_1$
Un tiers de confiance ou PKG (*Private Key Generator*) est responsable de la certification de l'identité d'un intervenant et de la maintenance du système de paramètres. Il détient une clé secrète $s \in \mathbf{Z}_r^*$, appelée *master key*, permettant de délivrer à un intervenant une clé secrète basée sur son identité. Pour obtenir un secret commun, Alice et Bob suivent les deux étapes suivantes du protocole :

1. chacun demande au PKG de lui générer un secret S à partir de sa propre identité (on peut prendre par exemple comme identifiant son adresse de courrier électronique) :
   – Alice reçoit $S_A = [s]Q_A$, où $Q_A = H_1(Id_A)$
   – Bob reçoit $S_B = [s]Q_B$, où $Q_B = H_1(Id_B)$

2. chacun peut alors calculer la clé commune K_{AB} sans discussion préalable :
   - Alice calcule $K_{AB} = \hat{e}(S_A, H_1(Id_B)) = \hat{e}(Q_A, Q_B)^s$
   - Bob calcule $K_{AB} = \hat{e}(H_1(Id_A), S_B) = \hat{e}(Q_A, Q_B)^s$

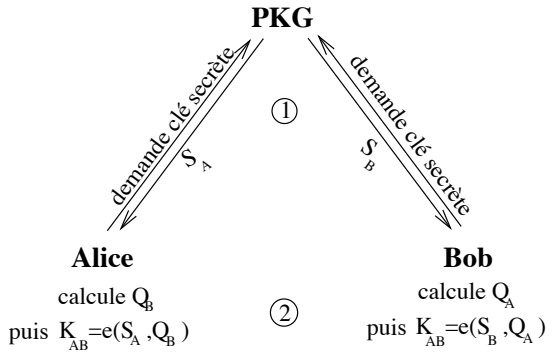


FIG. 2 – Distribution non interactive de clés basée sur l'identité

Remarque :
Les rôles d'Alice et de Bob étant complètement symétriques, une seule fonction de hachage à valeurs dans G_1 est utilisée, ce qui justifie l'utilisation d'un self-pairing.

Sécurité :
Il est clair que si Charlie sait résoudre le problème BDH, alors il est facile pour lui de retrouver la clé secrète K_{AB} d'Alice et Bob. En effet, Charlie connaît l'identité d'Alice et Bob, donc peut calculer Q_A et Q_B, il choisit alors un point Q comme générateur de G_1 (par exemple $Q = H_1(Id_C)$) et demande au PKG de calculer la valeur $[s]Q$. S'il sait résoudre BDH, connaissant $Q, Q_A = [a]Q$, $Q_B = [b]Q$ et $[s]Q$, il peut calculer $\hat{e}(Q, Q)^{abs} = \hat{e}(Q_A, Q_B)^s = K_{AB}$.

Dupont et Enge ([DE06]) ont prouvé pour une version quasi-similaire à celle de [SOK00] qu'être capable de trouver la clé générée par ce protocole est aussi difficile que de résoudre le problème BDH.

Comparaison avec le schéma de distribution de clé ECDH
Une alternative courante pour qu'Alice et Bob puissent s'échanger une clé est d'utiliser le protocole ECDH (Elliptic Curve Diffie-Hellman). Dans le schéma Diffie-Hellman original, on dispose d'un groupe G d'ordre premier r engendré par un point P pour lequel CDH est difficile. Alice choisit un secret $a \in \mathbf{Z}_r^*$ et envoie $q_A = [a]P$ à Bob. De même Bob envoie à Alice $q_B = [b]P$ où $b \in \mathbf{Z}_r^*$ est son secret. Ils peuvent alors calculer leur clé secrète commune $k_{AB} = [a]q_B = [b]q_A = [ab]P$, et un attaquant passif reste impuissant tant que CDH est difficile sur $G = \langle P \rangle$.

Cette version de Diffie-Hellman n'est cependant pas satisfaisante puisqu'Alice et Bob ne s'autentifient pas l'un auprès de l'autre, en particulier une attaque type "man-in-the-middle" est rendue possible. Pour que Bob puisse être sûr que q_A a bien été envoyé par Alice, celle-ci doit tout d'abord posséder un jeu de clés (k_p^A, k_s^A), ainsi qu'un certificat, contenant son identité et sa clé publique k_p^A, qui est signé par une autorité de certification (AC). Elle peut alors signer q_A avec sa clé privée k_s^A et Bob vérifie la signature avec la clef publique k_p^A figurant dans son certificat.

Il est intéressant de comparer les avantages et inconvénients de ces deux protocoles :

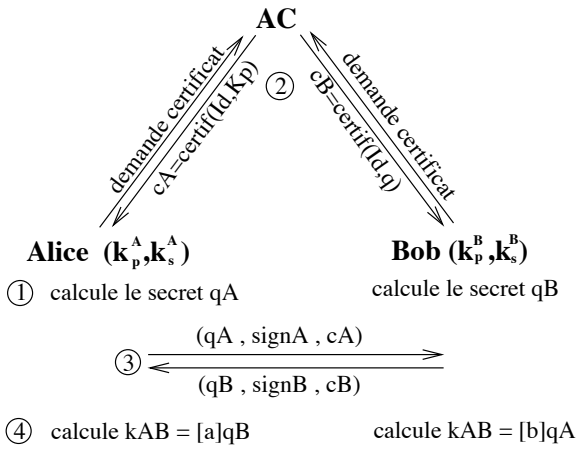


FIG. 3 – Distribution de clé ECDH avec signature et certificat.

Protocole de [SOK00]	**Protocole ECDH**
- *identity-based :* l'identité d'un intervenant est utilisée comme clé publique de celui-ci	- *non identity-based :* un certificat est nécessaire pour associer une clé publique à un intervenant
- Alice peut calculer la clé K_{AB} sans aucune intervention de Bob	- Alice doit attendre de recevoir la valeur q_B calculée par Bob pour pouvoir calculer la clé commune k_{AB}
- demande au PKG d'un secret	- demande à l'AC d'un certificat pour la clé publique servant à signer
- Confiance totale accordée au PKG : celui-ci peut en effet faire un séquestre de clés (*key escrow*)	- Pas de sequestre de clé possible

3.4 Un protocole Diffie-Hellman pour trois parties en un tour

Une idée naïve de protocole Diffie-Hellman pour trois parties permettrait à Alice, Bob et Charlie d'échanger un secret après 2 tours : on se donne un groupe G d'ordre premier r engendré par P, et chaque intervenant procède de la façon suivante

	Alice	**Bob**	**Charlie**
secret	a	b	c
1er tour	envoie $[a]P$ à Bob	envoie $[b]P$ à Charlie	envoie $[c]P$ à Alice
2ème tour	envoie $[a]([c]P)$ à Bob	envoie $[b]([a]P)$ à Charlie	envoie $[c]([b]P)$ à Alice
calcul de K_{ABC}	$[a]([cb]P)$	$[b]([ac]P)$	$[c]([ba]P)$

En 2000, Joux propose une version de ce protocole en un tour utilisant des couplages ([Jou04]) : on dispose d'un système de paramètres $(G_1, G_3, \hat{e}, P)$ où G_1 est un groupe cyclique d'ordre premier r engendré par P et $\hat{e} : G_1 \times G_1 \to G_3$ est un self-pairing. On modifie le protocole précédent de la façon suivante :

	Alice	**Bob**	**Charlie**
secret	a	b	c
1er tour	broadcast $[a]P$	broadcast $[b]P$	broadcast $[c]P$
calcul de K_{ABC}	$\hat{e}([b]P,[c]P)^a$	$\hat{e}([a]P,[c]P)^b$	$\hat{e}([a]P,[b]P)^c$

Ici encore, la sécurité est basée sur BDH.

Comme c'est le cas dans l'article original de [Jou04], il est possible de faire un échange entre trois parties sans utiliser de self-pairing : on choisit un système de paramètres (G_1, G_2, G_3, e, P, Q) où G_1, G_2 sont des groupes

cycliques d'ordre premier r, ainsi qu'un couplage bilinéaire $e : G_1 \times G_2 \to G_3$ non dégénéré. Le protocole à suivre est :

	Alice	**Bob**	**Charlie**
secret	a	b	c
1er tour	broadcast $[a]P, [a]Q$	broadcast $[b]P, [b]Q$	broadcast $[c]P, [c]Q$
calcul de K_{ABC}	$e([b]P, [c]Q)^a$	$e([a]P, [c]Q)^b$	$e([a]P, [b]Q)^c$

L'avantage de ce schéma est que l'on peut utiliser directement le couplage de Tate (ou Weil) sur n'importe quel type de courbe. Cependant il présente l'inconvénient d'utiliser deux fois plus de bande passante, puisque chaque intervenant ayant pour secret s doit transmettre $[s]P$ et $[s]Q$. Sa sécurité s'appuie sur l'hypothèse que Co-BDH est difficile.

3.5 Le chiffrement de Boneh-Franklin basé sur l'identité

Le chiffrement proposé par Boneh-Franklin en 2001 est considéré comme l'application la plus importante des couplages en cryptographie, puisqu'il répond à un problème de chiffrement basé sur l'identité posé par Shamir en 1986 et resté jusqu'alors sans réponse [Sha85].

On présente la version la plus simple (`BasicIdent`) de ce chiffrement, afin de mettre en valeur les idées principales de Boneh-Franklin. Un schéma plus complet et prouvé plus sûr (`FullIdent`) est également donné dans [BF03].

On se place dans le même type d'infrastructure que pour [SOK00] : le PKG publie un système de paramètres $\langle G_1, G_3, \hat{e}\rangle$, un générateur P de G_1 ainsi que le point $P_{pub} = [s]P$ obtenu à partir de la clé maître (*master key*) $s \in \mathbf{Z}_r^*$. En plus de $H_1 : \{0;1\}^* \to G_1$, on met à disposition des utilisateurs une $2^{\text{ème}}$ fonction de hachage $H_2 : G_3 \to \{0;1\}^n$, où n est le nombre de bits des messages transmis.

De façon générale, un schéma de chiffrement basé sur l'identité (*IBE*) est défini par la donnée de 4 algorithmes `Setup`, `Extract` (qui fournit à un intervenant une clé privée basée sur son identité), `Encrypt` et `Decrypt`. On donne le détail de ces algorithmes pour le schéma de Boneh-Franklin :

- `Setup` prend en entrée un paramètre de sécurité à partir duquel sont générés les paramètres du système décrits précédemment :
$$\langle G_1, G_3, \hat{e}, P, P_{pub}, H_1, H_2\rangle$$
L'espace des messages est $\mathcal{M} = \{0;1\}^n$ et l'espace des chiffrés est $\mathcal{C} = G_1 \times \{0;1\}^n$.
- `Extract` prend en entrée l'identité Id d'un utilisateur et fournit à celui-ci la clé privée correspondante $S_{Id} = [s]H_1(Id)$
- `Encrypt` permet de chiffrer un message $M \in \mathcal{M}$ destiné à un utilisateur à partir de son identité Id en 3 étapes :
  1. Calculer : $Q_{Id} = H_1(Id) \in G_1$
  2. Tirer un nombre aléatoire : $t \in \mathbf{Z}_r^*$
  3. Calculer le chiffré C de M : $C = \langle [t]P, M \oplus H_2(\hat{e}(Q_{Id}, P_{pub})^t)\rangle$
- `Decrypt` permet de déchiffrer $C = \langle C_1, C_2\rangle$ adressé à l'utilisateur Id grâce à sa clé privée S_{Id} en calculant
$$M' = C_2 \oplus H_2(\hat{e}(S_{Id}, C_1))$$

Pour vérifier la consistance de ce schéma, il suffit de remarquer que

$$\hat{e}(Q_{Id}, P_{pub})^t = \hat{e}(Q_{Id}, P)^{st} = \hat{e}([s]Q_{Id}, [t]P) = \hat{e}(S_{Id}, C_1)$$

Sécurité :
Comme dans [SOK00], on peut réduire la sécurité de ce schéma à BDH : si Bob adresse un message chiffré $C = \langle C_1, C_2\rangle$ à Alice, Charlie a accès à $P, P_{pub} = [s]P, C_1 = [t]P$ et $Q_A = H_1(Id_A) = [a]P$ (la clé publique d'Alice) et donc peut calculer, avec un algorithme résolvant BDH, $\hat{e}(P, P)^{sat} = \hat{e}(P_{pub}, Q_A)^t$, ce qui lui permet

de retrouver le message clair. Réciproquement, Boneh et Franklin montrent que dans le modèle de l'oracle aléatoire, le chiffrement décrit (`BasicIdent`) est sémantiquement sûr contre les attaques à texte clair choisi en autorisant des requêtes de clés privées (IND-ID-CPA) sous l'hypothèse que BDH est difficile.

Remarque :
On constate qu'il est tout à fait possible de remplacer le self-pairing $\hat{e}$ par un couplage asymétrique $e : G_1 \times G_2 \to G_3$:
– on modifie `Setup` en prenant $P, P_{pub} \in G_2$,
– dans `Extract`, on choisit des clés $S_{Id} = [s]Q_{Id} \in G_1$,
– `Encrypt` et `Decrypt` fonctionnent de la même façon en remplaçant $\hat{e}$ par e.
En particulier, on peut utiliser pour ce schéma une variété de courbes elliptiques plus étendue.

4 Construction de courbes adaptées (*pairing-friendly curves*)

Soit E une courbe elliptique définie sur $\mathbf{F}_q$ ($q = p^d$), r un entier premier différent de p divisant $\#E(\mathbf{F}_q)$ et k le degré de plongement associé. On note $G_1 = \langle P \rangle$ où $P \in E(\mathbf{F}_q)[r]$, $G_2 = \langle Q \rangle$ où $Q \in E(\mathbf{F}_{q^k})[r]$ et $G_3 = \mu_r \subset \mathbf{F}_{q^k}$.

Comme on l'a vu en section 3.2, si $e : G_1 \times G_2 \to G_3$ est un couplage, les groupes G_1, G_2 et G_3 doivent vérifier des hypothèses de sécurité, imposant certaines conditions sur les paramètres q, r et k. Le degré de plongement k ne doit également pas être trop grand en pratique, pour que les temps de calculs restent raisonnables.

Lorsqu'on peut trouver une courbe E pour laquelle tous ces critères sont vérifiés, on dit que la courbe est adaptée ou *pairing-friendly*.

4.1 Utilisation du couplage de Tate

On rappelle que le couplage de Tate est non dégénéré sur

$$\begin{array}{rcl} E(\mathbf{F}_{q^k})[r] \times E(\mathbf{F}_{q^k})/rE(\mathbf{F}_{q^k}) & \to & \mu_r \subset \mathbf{F}_{q^k}^* \\ (P, Q) & \mapsto & \langle P, Q \rangle \end{array}$$

en particulier si $P \in E(\mathbf{F}_q)[r]$, on peut toujours trouver $Q \in E(\mathbf{F}_{q^k})/rE(\mathbf{F}_{q^k})$ tel que $\langle P, Q \rangle \neq 1$.

Si l'on souhaite utiliser ce couplage dans le contexte cryptographique présenté en section 3, il est pertinent de déterminer à quelle condition on a un isomorphisme entre les groupes $E(\mathbf{F}_{q^k})[r]$ et $E(\mathbf{F}_{q^k})/rE(\mathbf{F}_{q^k})$.

La suite exacte

$$0 \to E(\mathbf{F}_{q^k})[r] \to E(\mathbf{F}_{q^k}) \to rE(\mathbf{F}_{q^k}) \to 0$$

permet déjà de dire que

$$rE(\mathbf{F}_{q^k}) \simeq E(\mathbf{F}_{q^k})/E(\mathbf{F}_{q^k})[r],$$

avec en particulier $\#E(\mathbf{F}_{q^k})[r] = \#\left(E(\mathbf{F}_{q^k})/rE(\mathbf{F}_{q^k})\right)$.

Il suffit donc de voir à quelle condition le morphisme naturel de groupes $\varphi : E(\mathbf{F}_{q^k})[r] \to E(\mathbf{F}_{q^k})/rE(\mathbf{F}_{q^k})$ est injectif. Si $R \in \ker(\varphi)$, alors il existe $S \in E(\mathbf{F}_{q^k})$ tel que $R = [r]S$, en particulier $S \in E(\mathbf{F}_{q^k})[r^2]$. Montrer que φ est injectif revient donc à montrer que les points rationnels de r^2-torsion sont nécessairement des points rationnels de r-torsion. On a donc l'équivalence :

$$E(\mathbf{F}_{q^k})[r] \simeq (\mathbf{F}_{q^k})/rE(\mathbf{F}_{q^k}) \Leftrightarrow E(\mathbf{F}_{q^k})[r^2] = E(\mathbf{F}_{q^k})[r].$$

Pour pouvoir utiliser le couplage de Tate pour les applications cryptographiques, on devra donc s'assurer que l'on est bien sous l'hypothèse suivante :

$$E(\mathbf{F}_{q^k})[r^2] = E(\mathbf{F}_{q^k})[r] \tag{4}$$

Remarque 4.1. Dans le cas où cette hypothèse ne serait pas vérifiée, on peut toujours utiliser le couplage de Weil, qui a l'avantage de toujours être non dégénéré sur $E[r] \times E[r]$, mais au prix d'une perte d'efficacité dans les temps de calculs (cf remarque 2.11 de la section 2.4.2).

La proposition suivante donne un critère simple pour déterminer quand le couplage de Tate peut être utilisé :

Proposition 4.2. *Si $r^2 || \#E(\mathbf{F}_{q^k})$ (i.e. $r^2 | \#E(\mathbf{F}_{q^k})$ et $r^3 \nmid \#E(\mathbf{F}_{q^k})$) et $k > 1$, alors l'hypothèse* (4) *est vérifiée. En particulier, le couplage de Tate est non dégénéré sur $E[r] \times E[r]$.*

Démonstration. D'après la proposition 2.5 (Balasubramanian et Koblitz) : comme $r^3 \nmid \#E(\mathbf{F}_{q^k})$, on a nécessairement $E(\mathbf{F}_{q^k})[r^2] \simeq \mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r\mathbf{Z}$ donc $E(\mathbf{F}_{q^k})[r^2] = E(\mathbf{F}_{q^k})[r]$. □

On détermine dans la suite sur quel groupe G_1 il est possible de prendre Tate comme self-pairing.

Lorsque $k > 1$, on ne pourra pas l'utiliser tel quel sur $E(\mathbf{F}_q)[r]$:

Proposition 4.3. *Soient E une courbe elliptique définie sur $\mathbf{F}_q$, G_1 un sous-groupe de $E(\mathbf{F}_q)$ engendré par un point $P \in E(\mathbf{F}_q)[r]$ de r-torsion (r premier et $r \neq p$) et k le degré de plongement associé. On suppose $k > 1$.*

Si $R \in E(\mathbf{F}_{q^d})$ où $d|k$ et $d < k$, alors $\langle P, R \rangle_r = 1$.

En particulier, $\langle P, P \rangle_r = 1$ et le couplage de Tate restreint à $E(\mathbf{F}_q)[r] \times E(\mathbf{F}_q)[r]$ est dégénéré.

Démonstration. f_P étant une fonction définie sur $\mathbf{F}_q$ et D_R étant un diviseur défini sur $\mathbf{F}_{q^d}$, $f_P(D_R)$ est un élément de $\mathbf{F}_{q^d}^*$ qui est nécessairement trivial dans le quotient $\mathbf{F}_{q^k}^*/(\mathbf{F}_{q^k}^*)^r$. En effet, comme r est premier et $d < k$, $\mathbf{F}_{q^d}^*$ ne contient aucune racine primitive r-ième de l'unité. Le morphisme de groupes $\pi : x \in \mathbf{F}_{q^d}^* \mapsto x^r \in \mathbf{F}_{q^d}^*$ est donc injectif, en particulier tout élément de $\mathbf{F}_{q^d}$ est une puissance r-ième d'un élément de $\mathbf{F}_{q^k}$.

Ainsi en prenant $d = 1$, on a que $\langle P, P \rangle_r = 1$ et par bilinéarité le couplage de Tate est dégénéré sur $G_1 = \langle P \rangle$. Par ailleurs, avec le lemme 2.3, comme $k > 1$, $E(\mathbf{F}_q)[r]$ ne peut contenir toute la r-torsion $E[r]$. Autrement dit $E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z}$, en particulier $G_1 = E(\mathbf{F}_q)[r]$ et le couplage est dégénéré sur les points rationnels de r-torsion. □

Dans le cas où le degré de plongement vaut 1, la non-dégénerescence de Tate vu comme self-pairing sur $G_1 = E(\mathbf{F}_q)[r]$ n'est pas toujours assurée, sauf dans le cas suivant (on distinguera dans la preuve tous les autres cas de figure possibles) :

Proposition 4.4. *Soient E une courbe elliptique définie sur $\mathbf{F}_q$, G_1 un sous-groupe de $E(\mathbf{F}_q)$ engendré par un point $P \in E(\mathbf{F}_q)[r]$ de r-torsion (r premier et $r \neq p$). On suppose que le degré de plongement associé k vaut 1.*

Si $r^2 \nmid \#E(\mathbf{F}_q)$, alors l'hypothèse (4) *est vérifiée et $\langle P, P \rangle \neq 1$. En particulier le couplage de Tate vu comme self-pairing sur $E(\mathbf{F}_q)[r]$ est non dégénéré.*

Démonstration. On cherche à déterminer dans quels cas on a $E(\mathbf{F}_q)[r^2] = E(\mathbf{F}_q)[r]$.

$E(\mathbf{F}_q)[r^2]$ étant à la fois un sous-groupe de $E(\mathbf{F}_q)$ et de $E[r^2] \simeq \mathbf{Z}/r^2\mathbf{Z} \times \mathbf{Z}/r^2\mathbf{Z}$, nécessairement

$$\#E(\mathbf{F}_q)[r^2] \,|\, \operatorname{pgcd}\left(\#E(\mathbf{F}_q), r^4\right).$$

On détaille ici tous les cas de figure possibles :

– **Cas 1 :** $r^2 \nmid \#E(\mathbf{F}_q)$, i.e. $\#E(\mathbf{F}_q)[r^2] = r$
Comme $E(\mathbf{F}_q)[r]$ est un sous-groupe d'ordre au moins r de $E(\mathbf{F}_q)[r^2]$, $E(\mathbf{F}_q)[r^2] = E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z}$. Le couplage de Tate est alors non dégénéré sur $E(\mathbf{F}_q)[r] \times E(\mathbf{F}_q)[r]$. Le sous-groupe $E(\mathbf{F}_q)[r]$ étant cyclique, on en déduit que le couplage de Tate est non dégénéré sur $G_1 \times G_1$.

- **Cas 2 :** $r^2|\#E(\mathbf{F}_q)$ et $r^3 \nmid \#E(\mathbf{F}_q)$, i.e. $\#E(\mathbf{F}_q)[r^2]|r^2$
  On distingue à nouveau plusieurs cas suivant la structure de $E(\mathbf{F}_q)[r]$ et de $E(\mathbf{F}_q)[r^2]$ vus comme sous-groupes de $E(\mathbf{F}_q) \simeq \mathbf{Z}/n_1\mathbf{Z} \times \mathbf{Z}/n_2\mathbf{Z}$ (où $n_1|n_2$) :
  - si $r^2|n_2$, alors $E(\mathbf{F}_q)$ admet un sous-groupe isomorphe à $\mathbf{Z}/r^2\mathbf{Z}$ de points de r^2-torsion. Par conséquent $E(\mathbf{F}_q)[r^2] \simeq \mathbf{Z}/r^2\mathbf{Z}$, qui admet un unique sous-groupe d'ordre r, ce qui impose $E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z} = rE(\mathbf{F}_q)[r^2]$. En particulier le couplage de Tate est dégénéré sur $G_1 \times G_1$.
  - si $r|n_1$, alors $E(\mathbf{F}_q)$ admet un sous-groupe isomorphe à $\mathbf{Z}/r\mathbf{Z}\times\mathbf{Z}/r\mathbf{Z}$ de points de r-torsion. Par conséquent $E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r\mathbf{Z} \simeq E[r]$ et comme $E(\mathbf{F}_q)[r^2]$ admet au plus r^2 éléments, $E(\mathbf{F}_q)[r] = E(\mathbf{F}_q)[r^2]$. Dans ce cas le couplage de Tate est non dégénéré sur $E(\mathbf{F}_q)[r] \times E(\mathbf{F}_q)[r]$, en particulier il existe un point rationnel Q de r-torsion tel que $\langle P, Q\rangle \neq 1$. Il est à noter cependant, que comme $E(\mathbf{F}_q)[r]$ n'est pas cyclique, Q n'est pas nécessairement dans le groupe engendré par P, et que donc on ne peut conclure quant à la non-dégénérescence du couplage sur G_1.
- **Cas 3 :** $r^3|\#E(\mathbf{F}_q)$ et $r^4 \nmid \#E(\mathbf{F}_q)$, i.e. $\#E(\mathbf{F}_q)[r^2]|r^3$
  Ce cas se traite de façon similaire au cas 2, en distinguant deux sous-cas :
  - si $r|n_1$ et $r^2|n_2$, alors $E(\mathbf{F}_q)$ admet un sous-groupe isomorphe à $\mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r^2\mathbf{Z}$ de points de r^2-torsion. En particulier $E(\mathbf{F}_q)[r^2] \simeq \mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r^2\mathbf{Z}$ et donc le sous-groupe des points rationnels de r-torsion est $E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r\mathbf{Z}$. Le couplage de Tate est alors dégénéré sur G_1.
  - si $r^3|n_2$, alors $E(\mathbf{F}_q)[r^2] \simeq \mathbf{Z}/r^2\mathbf{Z}$ et ce qui ramène au cas 2a.
- **Cas 4 :** $r^4|\#E(\mathbf{F}_q)$, i.e $\#E(\mathbf{F}_q)[r^2]|r^4$
  On distingue trois sous-cas :
  - soit $r \nmid n_1$ et$r^4|n_2$, alors on est ramené au cas 2a.
  - soit $r|n_1$ et$r^3|n_2$, alors on est ramené au cas 3a.
  - soit $r^2|n_1$ et $r^2|n_2$, alors $E(\mathbf{F}_q)[r^2] \simeq \mathbf{Z}/r^2\mathbf{Z}\times\mathbf{Z}/r^2\mathbf{Z}$, ce qui implique $E(\mathbf{F}_q)[r] = rE(\mathbf{F}_q)[r^2]$, en particulier le couplage de Tate est dégénéré sur $G_1 \times G_1$.

□

Les courbes qui vérifient les hypothèses de la proposition 4.4 disposent d'un self-pairing particulièrement simple, et donc sont intéressantes d'un point de vue cryptographique. Malheureusement, ces courbes sont assez difficiles à obtenir (voir exemple en fin de section 4.4).

Dans la suite, on supposera toujours que l'hypothèse (4) est vérifiée.

4.2 Courbes supersingulières et applications de distorsion

Comme décrit dans la section 3.1, le degré de plongement associé aux points de r-torsion d'une courbe supersingulière E est toujours petit, en particulier les courbes supersingulières sont de bons candidats pour les courbes bien couplées (pairing-friendly).

De façon plus précise, on a le résultat suivant, dû à Menezes, Okamoto et Vanstone, dont on donne une ébauche de preuve lorsque $p > 3$:

Proposition 4.5 ([CFA+06] p. 124)**.**
Soit E une courbe elliptique supersingulière définie sur $\mathbf{F}_q$ ($q = p^d$), admettant un point d'ordre r premier différent de p. Le degré de plongement k associé à r vérifie :
- *si $p = 2$, alors $k \leq 4$*
- *si $p = 3$, alors $k \leq 6$*
- *si $p \geq 5$, alors $k \leq 3$; si de plus $d = 1$, alors $k \leq 2$*

et ces bornes sont toujours atteintes.

Démonstration. Pour simplifier, on prendra $p > 3$ et $j(E) \in \mathbf{F}_p$ (on sait par le théorème 1.20 que l'on a toujours $j(E) \in \mathbf{F}_{p^2}$). On va montrer qu'alors nécessairement $k = 1$ ou $k = 2$.

Etant donné que $j(E) \in \mathbf{F}_p$, il est possible de trouver une courbe E_0 définie sur $\mathbf{F}_p$ telle que $j(E_0) = j(E)$ ([Sil86] p. 50). Par conséquent, E et E_0 sont $\mathbf{F}_{q^2}$-isomorphes via un changement de coordonnées de Weierstrass.

En particulier, les q^2-ièmes Frobenius définis sur E et E_0 ont même polynôme caractéristique. Comme E_0 est définie sur $\mathbf{F}_p$, on peut également calculer le q^2-ième Frobenius en fonction du p-ième Frobenius : $\Phi_q^2 = \Phi_p^{2d}$. On considère alors les polynômes caractéristiques $\chi_{\Phi_p}(X) = (X-\alpha)(X-\beta)$ de Φ_p et $\chi_{\Phi_q^2}(X) = (X-\alpha^{2d})(X-\beta^{2d})$ de Φ_q^2 définis sur E_0.

E étant supersingulière, le polynôme caractéristique de Φ_q défini sur E est de la forme $\chi_{\Phi_q}(X) = (X-a)(X-b)$ avec $\mathrm{Tr}(\Phi_q) = a+b = 0 \mod p$ et $ab = p^d$, en particulier $\mathrm{Tr}(\Phi_{q^2}) = a^2+b^2 = 0 \mod p$. De $\alpha^{2d}+\beta^{2d} = a^2+b^2 = 0 \mod p$ et de $\alpha\beta = p = 0 \mod p$, on déduit $\mathrm{Tr}(\Phi_p) = \alpha+\beta = 0 \mod p$. Avec la borne de Hasse, on a $|\mathrm{Tr}(\Phi_p)| \leq 2\sqrt{p}$ où $p > 3$, donc $\mathrm{Tr}(\Phi_p) = 0$. Ainsi, $\alpha = -\beta = \pm i\sqrt{p}$ et $\#E(\mathbf{F}_{q^2}) = \chi_{\Phi_q^2}(1) = (1-(-1)^d p^d)^2 = (1-(-1)^d q)^2$.

Donc [r premier et $r|\#E(\mathbf{F}_{q^2})] \Rightarrow r|(1-(-1)^d q) \Rightarrow k = 1$ ou 2 suivant la parité de d. □

On a la classification suivante des courbes supersingulières en fonction du degré de plongement :

Théorème 4.6 ([Wat69], [SX95])**.**
Soit E une courbe supersingulière définie sur $\mathbf{F}_q$ de trace t. Alors on est dans l'un des 5 cas suivants :
- *soit $q = p^{2b}$ et $t = \pm 2\sqrt{q}$, alors le degré de plongement k vaut 1,*
- *soit $q = p^a$ avec (a impair) ou ($p \not\equiv 1 \bmod 4$ et a pair) et $t = 0$, alors $k = 2$,*
- *soit $q = p^{2b}$ avec $p \not\equiv 1 \bmod 3$ et $t = \pm\sqrt{q}$, alors $k = 3$,*
- *soit $q = 2^{2b+1}$ et $t = \pm\sqrt{2q}$, alors $k = 4$,*
- *soit $q = 3^{2b+1}$ et $t = \pm\sqrt{3q}$, alors $k = 6$.*

Les courbes supersingulières ont donc l'avantage d'avoir un petit degré de plongement associé, ce qui rend possible le calcul de couplage. On montre dans ce qui suit qu'elles ont également l'avantage d'être naturellement munies de self-pairing.

Les couplages de Tate et de Weil utilisés tels quels n'étant pas de bons candidats pour les self-pairings (étant généralement dégénérés sur $E(\mathbf{F}_q)[r] \times E(\mathbf{F}_q)[r]$), on utilisera la technique due à Verheul [Ver04] qui consiste à introduire des endomorphismes particuliers appelés applications de distorsion :

Définition 4.7 (Applications de distorsion)**.**
Soit $P \in E[r]$ un point de r-torsion. Une application de distorsion relativement au groupe $\langle P\rangle$, est un endomorphisme $\varphi \in End(E)$ tel que pour tout $Q \in \langle P\rangle \setminus \{O\}$, $\varphi(Q) \notin \langle P\rangle$.

Si l'on peut trouver un tel endomorphisme sur E, alors il est facile de définir un self-pairing e sur $E(\mathbf{F}_q)[r] \times E(\mathbf{F}_q)[r]$ à partir du couplage de Weil ou de Tate :

Proposition 4.8. *On note indifféremment e le couplage de Weil ou de Tate. Si $\varphi \in End(E)$ est une application de distorsion, alors pour $k > 1$:*

$$\begin{array}{rcl} E(\mathbf{F}_q)[r] \times E(\mathbf{F}_q)[r] & \to & \mu_r \subset \mathbf{F}_{q^k}^* \\ (P,Q) & \mapsto & \hat{e}(P,Q) = e(P,\varphi(Q)) \end{array}$$

est un self-pairing.

Démonstration. Soit $P \in E(\mathbf{F}_q)$ un point d'ordre premier r. Comme φ est un endomorphisme de E, $([r]\circ\varphi)(P) = \varphi([r]P) = O$, en particulier $\varphi(P) \neq O$ est d'ordre r dans $E(\mathbf{F}_{q^k})$. On a donc trouvé une base $\{P, \varphi(P)\}$ de $E[r] = E(\mathbf{F}_{q^k})[r]$, et comme $e(P,P) = 1$ (prop. 4.3), par non-dégénérescence de Weil/Tate on a bien $\hat{e}(P,P) \neq 1$. □

On montre que cette construction n'est en fait possible que sur les courbes supersingulières :

Théorème 4.9. *Soit E une courbe elliptique et $P \in E(\mathbf{F}_q)[r]$ un point de r-torsion. On suppose le degré de plongement $k > 1$.*

S'il existe une application de distorsion φ relativement au groupe $\langle P\rangle$, alors E est supersingulière.

Démonstration. D'après le lemme 2.3, $\langle P\rangle = E(\mathbf{F}_q)[r]$. Par conséquent $\varphi(P) \in (E(\mathbf{F}_{q^k})[r]) \setminus E(\mathbf{F}_q)$, donc on a $\Phi_q(\varphi(P)) \neq \varphi(P) = \varphi(\Phi_q(P))$. $\mathrm{End}(E)$ est alors non commutatif, et la courbe est supersingulière (théorème 1.20). □

Théorème 4.10 (Existence d'applications de distorsion [Ver04])**.**
On note ψ l'application naturelle qui à un endomorphisme $\varphi \in End(E)$ associe sa restriction $\varphi_r \in End(E[r]) \simeq \mathcal{M}_2(\mathbf{Z}/r\mathbf{Z})$ définie sur l'ensemble des points de r-torsion. Si E est une courbe supersingulière, alors ψ est surjective. En particulier, il existe toujours une application de distorsion sur une courbe supersingulière.

Démonstration. On montre que $\ker\psi = [r]\mathrm{End}(E)$: soit $f \in \ker(\psi)$, alors $E[r] \subset \ker(f)$, i.e. $\ker([r]) \subset \ker(f)$. Comme $[r]$ est séparable, le théorème de factorisation des isogénies (voir [Sil86] cor. III.4.11) assure l'existence de $g \in \mathrm{End}(E)$ tel que $f = [r] \circ g$.

On rappelle également que si E est supersingulière, alors $\mathrm{End}(E)$ est un $\mathbf{Z}$-module de rang 4 (cf th. 1.20), par conséquent on a :

$$\mathrm{Im}\psi \simeq \mathrm{End}(E)/([r]\mathrm{End}(E)) \simeq (\mathbf{Z}/rZ)^4$$

et donc ψ est surjective. □

Voici quelques exemples classiques de courbes supersingulières pour lesquelles on connaît des applications de distorsion.

Exemple ([Gal05]).
- $k = 2$:
  $E : y^2 = x^3 + a$, courbe définie sur $\mathbf{F}_p$ où $p = 2 \bmod 3$
  cardinalité : $\#E(\mathbf{F}_p) = p + 1$
  application de distorsion : $(x, y) \mapsto (\zeta x, y)$, ζ tel que $\zeta^3 = 1$
- $k = 2$:
  $E : y^2 = x^3 + x$, courbe définie sur $\mathbf{F}_p$ où $p = 3 \bmod 4$
  cardinalité : $\#E(\mathbf{F}_p) = p + 1$
  application de distorsion : $(x, y) \mapsto (-x, iy)$, $i^2 = -1$
- $k = 3$:
  $E : y^2 = x^3 + a$, courbe définie sur $\mathbf{F}_{p^2}$ où $p = 5 \bmod 6$ et $a \in \mathbf{F}_{p^2} \setminus \mathbf{F}_p$ est un carré, mais pas un cube
  cardinalité : $\#E(\mathbf{F}_{p^2}) = p^2 - p + 1$ application de distorsion : $(x, y) \mapsto (x^p/(\gamma a^{(p-2)/3}), y^p/a^{(p-1)/2})$, $\gamma \in \mathbf{F}_{p^6}$ tel que $\gamma^3 = a$
- $k = 4$:
  $E : y^2 + y = x^3 + x + a$, courbe définie sur $\mathbf{F}_2$
  cardinalité : $\#E(\mathbf{F}_{2^{2b+1}}) = 2^{2b+1} \pm 2^{b+1} + 1$
  application de distorsion : $(x, y) \mapsto (u^2x + s^2, y + u^2sx + s)$, $u \in \mathbf{F}_{2^2}$ et $s \in \mathbf{F}_{2^4}$ tels que $u^2 + u + 1 = 0$ et $s^2 + (u + 1)s + 1 = 0$
- $k = 6$:
  $E : y^2 = x^3 - x \pm 1$, courbe définie sur $\mathbf{F}_3$
  cardinalité : $\#E(\mathbf{F}_{3^{2b+1}}) = 3^{2b+1} \pm 3^{b+1} + 1$
  application de distorsion : $(x, y) \mapsto (\alpha - x, iy)$, $i \in \mathbf{F}_{3^2}$ et $\alpha \in \mathbf{F}_{3^3}$ tels que $i^2 = -1$ et $\alpha^3 - \alpha \mp 1 = 0$

On renvoie à [GR04] pour plus de détails sur la construction d'applications de distorsion.

A titre d'application, on démontre avec les applications de distorsion, l'antisymétrie du couplage de Tate sur les courbes supersingulières. Ceci justifie que Tate tel quel n'est pas un bon candidat pour la construction de self-pairing sur ces courbes, puisqu'il n'existe pas de sous-groupe cyclique de $E[r]$ sur lequel Tate ne soit pas dégénéré.

Proposition 4.11 (Antisymétrie du couplage de Tate sur les courbes supersingulières)**.**
Soit E supersingulière définie sur $\mathbf{F}_q$, $r \mid \#E(\mathbf{F}_q)$ et $k > 1$ le degré de plongement correspondant. Alors pour tous points $P, Q \in E(\mathbf{F}_{q^k})[r]$,

$$\langle P, Q\rangle = \langle Q, P\rangle^{-1}$$

Démonstration. Avec la proposition 2.5, on sait que

$$E(\mathbf{F}_{q^k})[r] \simeq \mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r\mathbf{Z},$$

on étudie alors les valeurs propres et vecteurs propres de l'endomorphisme de Frobenius agissant sur cet espace vectoriel :

Lemme 4.12 (Couplage de Tate et valeurs propres du Frobenius)**.**
Soient $\lambda \in \mathbf{Z}/r\mathbf{Z}$ une valeur propre de l'endomorphisme de Frobenius vu comme application $\mathbf{Z}/r\mathbf{Z}$-linéaire de $E[r]$ et $P \in E[r]$ un vecteur propre associé à λ. Alors

$$\langle P, P\rangle = 1.$$

Démonstration. On a d'une part

$$\langle \varphi(P), \varphi(P)\rangle = \langle P, P\rangle^{\lambda^2}$$

et d'autre part

$$\langle \varphi(P), \varphi(P)\rangle = \langle \hat{\varphi} \circ \varphi(P), P\rangle = \langle P, P\rangle^{q}$$

Ainsi si $\langle P, P\rangle \neq 1$, alors $r|(\lambda^2 - q)$. Mais ceci est impossible, étant donné que les racines modulo r du polynôme caractéristique du Frobenius

$$\chi(\varphi)(X) = X^2 - \mathrm{Tr}(\varphi)X + q$$

sont 1 et q, et que l'on a supposé $l \wedge q = 1$ et $k > 1$. □

Soit P et Q deux points engendrant la r-torsion, on choisira P rationnel et $Q \in E(\mathbf{F}_{q^k})[r]$ vecteur propre de l'endomorphisme de Frobenius, associé à la valeur propre q. En particulier $\langle P, P\rangle = \langle Q, Q\rangle = 1$. Comme E est supersingulière, il existe ([Ver04] p. 289) une application de distorsion $\varphi \in End(E)$ telle que la matrice de φ soit de la forme :

$$Mat_{\varphi} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad \text{avec } c, b \neq 0 \bmod r$$

On a alors d'une part

$$\langle \varphi(P), \varphi(Q)\rangle = \langle \hat{\varphi} \circ \varphi(P), Q\rangle = \langle P, Q\rangle^{\deg \varphi}$$

et d'autre part

$$\langle \varphi(P), \varphi(Q)\rangle = \langle P, P\rangle^{ab} \langle P, Q\rangle^{ad} \langle Q, P\rangle^{bc} \langle Q, Q\rangle^{cd} = \langle P, Q\rangle^{ad} \langle Q, P\rangle^{bc}$$

Donc

$$\langle P, Q\rangle^{\deg \varphi - ad} = \langle Q, P\rangle^{bc}$$

Comme $\deg \varphi = \det \varphi \mod r$ (cf lemme 2.6), on en déduit que

$$(\langle P, Q\rangle \langle Q, P\rangle)^{bc} = 1$$

Si $\langle P, Q\rangle \langle Q, P\rangle \in \mu_r$ est non trivial, il est d'ordre r, en particulier $bc = 0 \mod r$ ce qui est exclus.

Finalement, $(\langle P, Q\rangle \langle Q, P\rangle) = 1$ et le couplage de Tate est antisymétrique. □

Pour obtenir un degré de plongement supérieur à 6, il est nécessaire de travailler avec des courbes ordinaires munies de couplages asymétriques. Malheureusement, comme ces courbes ont en général un degré de plongement bien trop gros (de l'ordre de q), une recherche de courbe aléatoire ne fournira pas de courbes pairing-friendly. On peut cependant construire grâce à la méthode de multipliction complexe (CM) plusieurs familles de courbes ordinaires (courbes MNT, de Brezing-Weng, de Freeman...) avec un degré de plongement prescris.

On présente dans ce qui suit les éléments clefs de la méthode CM, ainsi que la construction de courbes MNT pour un degré de plongement $k = 6$.

4.3 Construction de courbes par la méthode CM

Atkin et Morain dans [AM93] adaptent au cas des corps finis la méthode de multiplication complexe (CM) utilisée pour les courbes elliptiques définies sur $\mathbf{C}$.

Etant donnés q un nombre premier et un entier t dans la borne de Hasse, cette méthode permet de trouver une courbe E définie sur $\mathbf{F}_q$ ayant $q+1-t$ points.
L'idée est de retrouver à partir du polynôme caractéristique de l'endomorphisme de Frobenius sur $E(\mathbf{F}_q)$

$$X^2 - tX + q = 0 \tag{5}$$

des informations sur l'anneau $\mathrm{End}(E)$ des endomorphismes de la courbe. On considère le discriminant de l'équation (5) que l'on écrit sous la forme

$$4q - t^2 = Dy^2 \tag{6}$$

où $-D$ est un entier appelé *discriminant fondamental* tel que $-D \neq 1$ et
– soit $D = 3 \bmod 4$ et sans facteurs carrés,
– soit $D = 4m$ avec $m = 1$ ou $2 \bmod 4$ sans facteurs carrés.
La méthode d'Atkin et Morain consiste à construire E telle que $\mathrm{End}(E)$ soit l'anneau des entiers de $\mathbf{Q}(\sqrt{-D})$. On introduit à cet effet le polynôme de classe de Hilbert $H_D(X) \in \mathbf{Z}[X]$ dont les racines dans $\mathbf{F}_q$ sont toutes des j-invariants de courbes ayant l'anneau d'endomorphismes souhaité. L'algorithme qui permet de calculer ce polynôme consiste à reconstruire les coefficients de H_D à partir de ses racines complexes, que l'on obtient via des formules à base de séries convergentes ([Coh93] p. 415).

Pour retrouver la courbe à partir de son j-invariant, on utilise le résultat suivant :

Proposition 4.13 ([BSS00] Lem. VIII.3.)**.**
Tout élément de $\mathbf{F}_q$ est le j-invariant d'une courbe elliptique définie sur $\mathbf{F}_q$. En particulier, si $j \neq 0, 1728$, alors on peut prendre la courbe d'équation

$$y^2 = x^3 + 3kc^2x + 2kc^3$$

où $k = \dfrac{j}{1728 - j}$ et $c \in \mathbf{F}_q$ quelconque.

Si E et $\tilde{E}$ ont le même j-invariant $j \neq 0, 1728$, alors soit $\tilde{E}$ est isomorphe à E sur $\mathbf{F}_q$, soit $\tilde{E}$ est une tordue quadratique *de E et sa trace est l'opposé de celle de E.*

Le problème est que cette méthode ne peut fonctionner que pour D relativement petit. La taille des coefficients et le degré de ce polynôme sont en effet en $O(\sqrt{D})$, ce qui nécessite de prendre en pratique $D < 2^{25}$, et un choix adapté pour les valeurs de q et t.

Exemple. On cherche une courbe sur $\mathbf{F}_{17}$ de trace $t = 3$ et donc ayant $q + 1 - t = 15$ points rationnels. Alors $4q - t^2 = 59 = D$ et

$$H_{-59}(X) = X^3 + 30197678080X^2 - 140811576541184X + 374643194001883136$$

Les racines dans $\mathbf{F}_{17}$ de $H_{-59}(X) \equiv X^3 - 5X^2 - 5X + 5 \bmod 17$ sont $2, 7$ et 13.

Pour $j = 2$, on trouve la courbe d'équation $y^2 = x^3 + 12x + 8$ qui est de cardinalité $15 = q + 1 - t$.

Pour $j = 7$, on trouve la courbe d'équation $y^2 = x^3 + x + 12$ qui est de cardinalité $15 = q + 1 - t$.

Pour $j = 13$, on trouve la courbe d'équation $y^2 = x^3 + 6x + 4$ qui est de cardinalité $15 = q + 1 - t$.

4.4 Un exemple de courbes ordinaires : les courbes MNT

La stratégie de Miyaji, Nakabayashi et Takano [MNT01] consiste à paramétrer quadratiquement q et t, en s'inspirant du résultat suivant :

Théorème 4.14. *Soit E une courbe elliptique ordinaire définie sur $\mathbf{F}_q$ dont le nombre de points rationnels $q+1-t$ est premier et de degré de plongement $k=6$. Alors il existe un entier l tel que*

$$q = 4l^2 + 1 \quad et \quad t = 1 \pm 2l$$

En réinjectant ce paramétrage dans (6), on trouve :

$$\begin{aligned} & 4(4l^2+1) - (1 \pm 2l)^2 = Dy^2 \\ \Leftrightarrow \quad & 12l^2 \mp 4l + 3 = Dy^2 \\ \Leftrightarrow \quad & (6l \mp 1)^2 + 8 = 3Dy^2 \end{aligned}$$

ce qui ramène à la résolution d'une équation diophantienne de la forme

$$x^2 - 3Dy^2 = -8 \tag{7}$$

où $x = 6l \mp 1$. Ce type d'équation diophantienne est appelé *équation de Pell généralisée.*

Pour construire une courbe ordinaire avec $k=6$, on choisit un discriminant fondamental $-D$, et on cherche parmi les couples (x,y) solutions de l'équation (7), ceux qui vérifient $x \equiv \pm 1 \bmod 6$ et tels que $q = 1 + 4\left(\frac{x\pm1}{6}\right)^2$ soit un grand nombre premier. On vérifie ensuite que $q+1-t$ où $t = 1 \pm 2\frac{x\pm1}{6}$ possède un grand facteur premier et un degré de plongement $k=6$. Si aucune solution ne remplit ces critères, on passe au discriminant fondamental suivant.

Remarque 4.15. On peut restreindre les valeurs possibles pour D en remarquant si on a une solution de l'équation (7), alors -8 (et donc -2) est carré modulo $3D$. Donc si p est facteur premier de D, p vérifie nécessairement $p \equiv 1$ ou $3 \bmod 8$. Par ailleurs, D doit être impair pour que $x \equiv \pm 1 \bmod 6$.

Pour résoudre une équation de Pell généralisée, la technique consiste d'abord à chercher une solution minimale (x_0, y_0) de l'équation de Pell

$$x^2 - 3Dy^2 = 1 \tag{8}$$

Une telle solution vérifie

$$\left| \frac{x_0}{y_0} - \sqrt{3D} \right| < \frac{1}{2y_0^2}$$

et peut donc être obtenue en détectant, dans le développement en fractions continues de $\sqrt{3D}$, la première réduite $\dfrac{x_0}{y_0}$ où x_0 et y_0 vérifient l'équation (8) (voir [ZÓ0] prop. 1.28).

On détaille l'algorithme permettant de calculer les réduites du développement en fractions continues de $\sqrt{n}$:

Lemme 4.16. *Soient a_k le k-ième coefficient intervenant dans le développement en fractions continues de $\sqrt{n}$:*

$$\sqrt{n} = a_0 + \frac{1}{a_1 + \frac{1}{a_2 + \ldots}} = [a_0, a_1, a_2, \ldots]$$

On définit r_k par

$$\sqrt{n} = a_0 + \frac{1}{a_1 + \frac{1}{\ldots a_{k-1} + \frac{1}{r_k}}}$$

autrement dit,

$$r_k = [a_k, a_{k+1}, \ldots].$$

Alors il existe des suites d'entiers (b_k) *et* (c_k) *telles que*

$$\begin{cases} r_k = \dfrac{b_k + \sqrt{n}}{c_k} \\ a_k = \left\lfloor \dfrac{a_0 + b_k}{c_k} \right\rfloor \\ b_{k+1} = a_k c_k - b_k \\ c_{k+1} = 2a_k b_k - a_k^2 c_k + c_{k-1} \\ a_0 = \lfloor \sqrt{n} \rfloor \\ b_0 = 0 \\ c_{-1} = n,\ c_0 = 1,\ c_1 = n - a_0^2 \end{cases}$$

Démonstration. Par définition, $r_0 = \sqrt{n}$, donc $b_0 = 0$ et $c_0 = 1$. Puis, $r_1 = \dfrac{1}{r_0 - a_0} = \dfrac{\sqrt{n} + a_0}{n - a_0^2}$, dont on déduit $b_1 = a_0$ et $c_1 = n - a_0^2$. On vérifie facilement par récurrence les relations sur r_k, b_k, c_k. Par ailleurs,

$$a_k = \lfloor r_k \rfloor = \left\lfloor \frac{b_k + a_0 + \sqrt{n} - a_0}{c_k} \right\rfloor = \left\lfloor \frac{b_k + a_0}{c_k} \right\rfloor$$

puisque $|\sqrt{n} - a_0| < 1$. □

Il est alors facile de calculer par récurrence la k-ième réduite $\frac{x_k}{y_k} = [a_0, \ldots, a_k]$ ([Zó0] p. 17) :

$$\begin{cases} x_k = a_k x_{k-1} + x_{k-2} \\ y_k = a_k y_{k-1} + y_{k-2} \\ x_{-1} = 1,\ x_0 = a_0,\ x_1 = a_0 a_1 + 1 \\ y_{-1} = 0,\ y_0 = 1,\ y_1 = a_1 \end{cases}$$

On en déduit l'algorithme suivant :

Alg. 3 Calcul d'une solution minimale (x, y) de l'équation de Pell $x^2 - ny^2 = 1$

ENTRÉE : n
SORTIE : x, y
  $a_0 \leftarrow \lfloor \sqrt{n} \rfloor$, $a \leftarrow a_0$, $b \leftarrow 0$, $\tilde{c} \leftarrow n$, $c \leftarrow 1$, $\tilde{x} \leftarrow 1$, $x \leftarrow a_0$, $\tilde{y} \leftarrow 0$, $y \leftarrow 1$
  répéter
    $b' \leftarrow ac - b$
    $c' \leftarrow 2ab - a^2 c + \tilde{c}$
    $a \leftarrow \lfloor \frac{a_0 + b'}{c'} \rfloor$
    $x' \leftarrow ax + \tilde{x}$
    $y' \leftarrow ay + \tilde{y}$
    $b \leftarrow b'$
    $\tilde{c} \leftarrow c$
    $c \leftarrow c'$
    $\tilde{x} \leftarrow x$
    $x \leftarrow x'$
    $\tilde{y} \leftarrow y$
    $y \leftarrow y'$
  jusqu'à ce que $x^2 - ny^2 = 1$

Remarque 4.17. Il est classique que le développement en fractions continues de $\sqrt{n}$ est périodique. Si on note k la période, une solution pour Pell est trouvée au bout de k itérations si k est pair ou $2k$ itérations si k est impair. En particulier, l'algorithme s'arrête.

On détaille ensuite comment résoudre l'équation de Pell généralisée (7) :

Lemme 4.18 (Résolution de l'équation généralisée de Pell)**.**
Si $n > N^2$, alors les solutions de l'équation généralisée de Pell

$$x^2 - ny^2 = N \tag{9}$$

s'il en existe, s'obtiennent comme des réduites du développement en fractions continues de $\sqrt{n}$.

Démonstration. On a

$$\begin{aligned} x^2 - ny^2 = N &\Leftrightarrow (x - \sqrt{n}y)(x + \sqrt{n}y) = N \\ &\Leftrightarrow \left(\frac{x}{y} - \sqrt{n}\right)\left(\frac{x}{y} + \sqrt{n}\right) = \frac{N}{y^2} \\ &\Rightarrow \left|\frac{x}{y} - \sqrt{n}\right| < \frac{1}{2y^2} \end{aligned}$$

en particulier $\frac{x}{y}$ est une réduite du développement en fractions continues de $\sqrt{n}$. □

Remarque 4.19. Il est à noter également [Mat00] que si l'équation de Pell généralisée (7) admet une solution (x_p, y_p), alors cette solution sera détectée lors de la recherche de la solution minimale (x_0, y_0) de l'équation de Pell (8).

On obtient alors une infinité de solutions pour (7) en considérant les éléments de la forme

$$(x_p + \sqrt{3D}y_p)(x_0 + \sqrt{3D}y_0)^k \quad \text{avec } k \in \mathbf{Z}$$

Exemple.
Pour $D = 43$, on trouve un développement en fractions continues de $\sqrt{3D}$ égal à $[11; \overline{2; 1; 3; 1; 6; 1; 3; 1; 2; 22}]$ et donc la solution de l'équation

$$x^2 - 129y^2 = 1$$

est obtenue en écrivant

$$\frac{x}{y} = [11; 2; 1; 3; 1; 6; 1; 3; 1; 2] = \frac{16855}{1484}.$$

Au passage, la réduite $[11] = \frac{11}{1}$ donne une solution particulière pour l'équation de Pell généralisée :

$$11^2 - 129 \times 1^2 = -8$$

On construit ensuite une famille de solutions de la forme

$$x + y\sqrt{129} = (11 + \sqrt{129})(16855 + 1484\sqrt{129})^n, \ n \in \mathbf{Z}.$$

Pour chaque solution, on teste si le $q = 1 + 4\left(\frac{x \pm 1}{6}\right)^2$ correspondant est premier et si la cardinalité correspondante ($= q + 1 - t$, où $t = 1 \pm 2\frac{x \pm 1}{6}$) possède un grand facteur premier :

- $n = 0 : (x, y) = (11, 1)$, $l = 2$, $q = 17$ est premier, $t = 5$ donc $q + 1 - t = 13$
- $n = 1 : (x, y) = (376841, 33179)$, $l = 62807$, $q = 15778876997$ n'est pas premier.
- $n = -1 : (x, y) = (-6031, 531)$, $l = -1005$, $q = 4040101$ n'est pas premier.
- $n = 2 : (x, y) = (12703310099, 1118464089)$, $l = 2117218350$, $q = 17930454166306890001$ n'est pas premier.
- $n = -2 : (x, y) = (-203305021, 17900009)$, $l = -33884170$, $q = 4592547906355601$ est premier, $t = -67768339$ donc $q + 1 - t = 4592547974123941 = 13 \times 2347 \times 150521057131$.

La solution trouvée pour $n = -2$ donne une cardinalité ayant un facteur premier de 37 bits et un degré de plongement $k = 6$.

Le polynôme de classe de Hilbert pour $-D = -43$ est de degré 1 :

$$H_{-43}(X) = X + 884736000$$

et admet 4592547021619601 comme racine modulo 4592547906355601.

Une courbe de j-invariant 4592547021619601 sur $\mathbf{F}_{4592547906355601}$ est donnée par l'équation :

$$\tilde{E} \ : \ y^2 = x^3 + 2564278200474279x + 1709518800316186$$

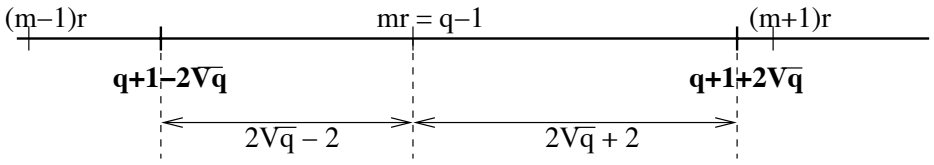
On constate que le nombre de points de cette courbe n'est pas égal à $q + 1 - t$. Il est donc égal à $q + 1 + t$, et on prend donc pour la courbe cherchée E une tordue de $\tilde{E}$. Par exemple :

$$E \ : \ y^2 = x^3 + 1763476217229032x + 3447467182151685$$

Il est à noter cependant que la méthode de multiplication complexe présentée ici ne permet pas de résoudre tous les problème de construction de courbes :

Exemple (Courbes de trace 2 et de degré de plongement $k = 1$).
Lorsque r est un diviseur de $\#E(\mathbf{F}_q)$ assez grand, ou de façon plus précise $r \geq 2\sqrt{q} + 2$, les courbes elliptiques admettant des points de r-torsion et un degré de plongement égal à 1, ont nécessairement leur q-ième morphisme de Frobenius de trace égale à 2.

En effet, si on note $I = [q + 1 - 2\sqrt{q}; q + 1 + 2\sqrt{q}]$ l'intervalle des valeurs possibles pour $\#E(\mathbf{F}_q)$ donné par la borne de Hasse, on a $q - 1 \in I$ qui est le seul multiple possible de r dans cet intervalle :



En particulier, $q - 1 = \#E(\mathbf{F}_q) = q + 1 - \mathrm{Tr}\,\Phi_q$ et $\mathrm{Tr}\,\Phi_q = 2$.

La construction d'une courbe de trace 2 vérifiant les propriétés de la proposition 4.4 dans le cas où le degré de plongement vaut 1 pose a priori problème.

Si on utilise la méthode de multiplication complexe, on doit choisir un discriminant fondamental D sans facteur carré et petit, puis trouver $q = p^d$ et y tels que $4q - (\mathrm{Tr}\,\Phi_q)^2 = y^2 D$. Dans ce cas précis, comme $\mathrm{Tr}(\Phi_q) = 2$, on a $y^2 D = 4(q - 1)$, en particulier si r est un grand facteur premier de $q - 1$, nécessairement $r|y$, donc $r^2|q - 1$. D'autres méthodes doivent donc être envisagées pour la construction de telles courbes.

A Implémentation

A.1 Présentation générale des différents modules

On présente dans cette partie les différents modules qui ont été implémentés en langage `C++`.

Trois librairies sont utilisées :
– GMP (*GNU Multiprecision Package*) [G$^+$] : elle permet de faire de l'arithmétique sur de grands entiers, et est utilisée implicitement par NTL pour améliorer ses performances.

– NTL (*Number Theory Library*) [Sho] sur laquelle s'appuient tous nos programmes : elle permet la gestion des grands nombres, fournit une implémentation des corps finis et de l'arithmétique modulaire, permet le calcul de racines de polynômes ou la recherche de polynôme irréductibles sur les corps finis, donne des tests de primalité, intègre un générateur de nombres aléatoires et de nombres premiers, fournit un calcul multiprécision dans les réels...
– crypto++ [Dai] : elle est utilisée dans les programmes `encrypt, decrypt` et `extract` pour le calcul de la fonction de hachage SHA-512.

La factorisation d'entiers et le calcul du nombre de points rationnels d'une courbe elliptique ont posé problème lors de l'implémentation. La première solution envisagée pour résoudre le problème de factorisation était d'utiliser le projet GMP-ECM (Elliptic Curve Method for Integer Factorization) mis à disposition par l'INRIA, mais ceci s'est avéré trop délicat à utiliser. L'utilisation de PARI/GP [C$^+$] par un appel externe s'est finalement révélée très efficace pour résoudre ce problème. Concernant le calcul de la cardinalité d'une courbe, on utilise le module de Reynald Lercier [Ler] basé sur `ZEN` [CL], qui a l'avantage d'implémenter différents algorithmes tels que SEA, AGM... et qui est particulièrement efficace moyennant une vingtaine d'heures de précalculs (sur un MacBook Processeur Intel Core Duo à 2GHz).

On résume ici rapidement les fonctionnalités des différents modules implémentés :
– `ellipCurve` : ce module permet l'implémentation des courbes définies sur $\mathbf{F}_q$ où $q = p^d$ et $p > 3$, et de leurs points $\mathbf{F}_{q^k}$-rationnels. On retrouve les opérations habituelles sur courbes elliptiques : calcul du discriminant et du j-invariant, addition de points, multiplication par un entier, tirage de points aléatoires sur la courbe, algorithme de Miller pour le calcul des couplages de Weil et Tate. Le tableau suivant résume les différentes classes NTL appelées par notre programme :

	$d = 1$	$d > 1$	
	(p grand)	p petit	p grand
$E : y^2 = x^3 + ax + b$ définie sur $\mathbf{F}_q$ où a, b sont représentés dans les classes :	`ZZ_p`	`zz_pE`	`ZZ_pE`
$\mathbf{F}_{q^k} = \mathbf{F}_q[X]/(mod)$ où mod est représenté dans les classes :	`ZZ_pX`	`zz_pEX`	`ZZ_pEX`

– `mathTools` : ce module regroupe toutes les fonctions mathématiques utilisées dans les programmes, à savoir la factorisation d'entiers par appel à `GP`, la résolution d'équations de Pell, le calcul du polynôme de Hilbert qui nécessite en particulier l'implémentation des complexes en multiprécision et le calcul du j-invariant d'une courbe complexe.

– `analysis` et `curvAnalysis` : ce dernier est un exécutable qui propose à l'utilisateur de saisir un corps fini $\mathbf{F}_{p^d}$, puis l'équation de Weierstrass réduite d'une courbe elliptique. Le programme `analysis` analyse ensuite les propriétés de cette courbe : discriminant, j-invariant, cardinalité par appel à `ZEN`, caractère ordinaire ou supersingulier, factorisation du nombre de points, degré de plongement, base de la r-torsion, et calcul du couplage de Tate sur cette base.

– `complexMult` : ce module implémente la méthode CM de construction de courbes, ainsi que la stratégie MNT de recherche de courbes ordinaires ayant un degré de plongement $k = 6$.

– IBE : comme décrit en section 3.5, le chiffrement basé sur l'identité repose sur quatre programmes :
  – `setup` : cet exécutable demande à l'utilisateur de choisir un type de courbe (supersingulière avec $k = 2$ ou MNT avec $k = 6$) et un degré de sécurité. Il configure ensuite le système IBE en construisant la courbe adaptée. Les paramètres publics sont stockés dans le fichier `.ibe_config` et la clé secrète dans le fichier `.master_key`.
  – `extract` : cet exécutable fournit à partir du haché de l'identité de l'utilisateur sa clé privée et la stocke dans le fichier `.private_key`.
  – `encrypt` : cet exécutable demande à l'utilisateur de saisir un destinataire et un message, puis lui retourne le message chiffré correspondant (en base 64).
  – `decrypt` : cet exécutable décrypte un message chiffré en utilisant la clé privée de l'utilisateur.

La figure 4 résume les dépendances qui existent entre les différents programmes.

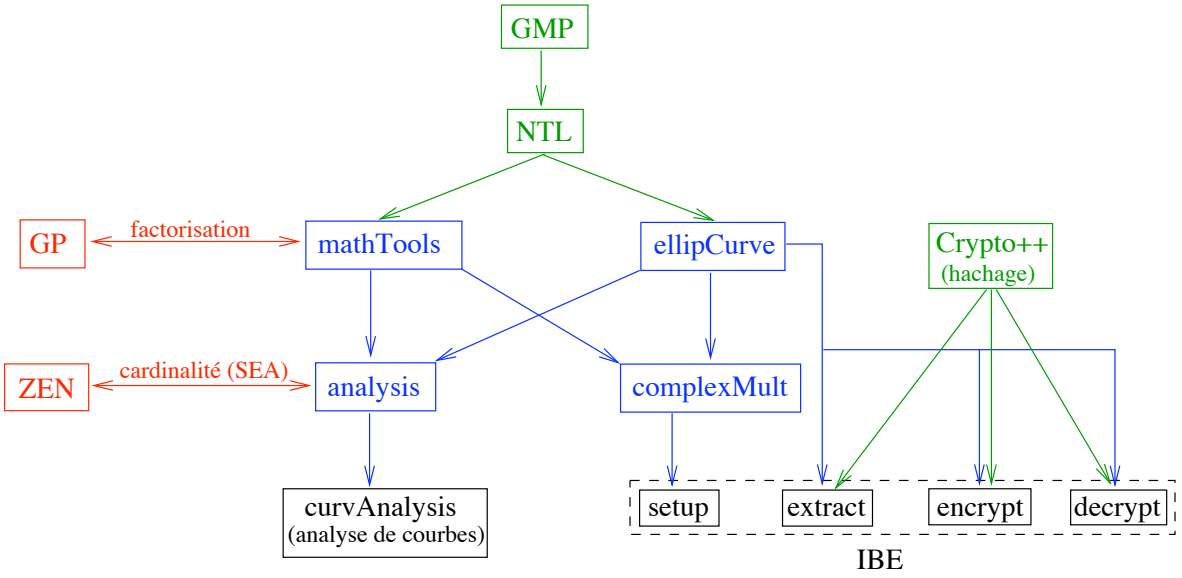


FIG. 4 – Dépendance des fichiers (en bleu et noir, les programmes réalisés, en noir les exécutables, en vert les librairies utilisées, en rouge les modules extérieurs utilisés)

A.2 Un exemple pour IBE

Voici un exemple correspondant à l'appel des 4 algorithmes constituant les 4 phases d'IBE.

On commence par lancer `setup` qui affiche la courbe supersingulière trouvée :

```
nina:~/Documents/MAA/STAGE/programs vanessavitse$ setup
__Choice of curve__
_ for a supersingular curve (k=2) enter 1
_ for an MNT curve (k=6) enter 2
Your choice : 1

__Security level__
Good security is reached with 80 bits
Your choice (in bitschar‘ less than 150)  : 50

Base field : F_107874350821290134184135795319183757671672932154515877835597
Order of the G1 subgroup : r = 68083551102365835333030312491411
Curve equation : y^2 = x^3 + 0x + 1
Irreducible polynomial defining F_(p^2) : [1 1 1]
Generating secret key...
Public keys :
P = [[811729700613253007454181515137911232314081837784210724444458]:
[931328722489420921287461349873834011602477856379653364369 57]:[1]]
Ppub = [[251092342002649374790504082737334898343063673091717427382 64]:
[106159844625109821362200662642139936376887314311663834035367]:[1]]
Size of message encryption blocks : n = 64
Writing data in .ibe_config file...
Writing secret key...
```

Le fichier `.ibe_config` contient maintenant tous les paramètres publics du système ; dans l'ordre : le degré de plongement (ici $k = 2$), le nombre p d'éléments du corps fini, les coefficients a et b de l'équation réduite de la courbe (ici $y^2 = x^3 + 1$), la cardinalité de G_1, la taille des blocs pour le chiffrement en octets, le polynôme

définissant l'extension F_{p^k} (ici $1+X+X^2$), l'abscisse et l'ordonnée de P et de $P_{pub} = [s]P$. Le fichier `.master_key` qui n'est censé être lu que par le PKG contient la valeur de s.

```
nina:~/Documents/MAA/STAGE/programs vanessavitse$ more .ibe_config
2 10787435082129013418413579531918375767167293215451587783559 7 0 1
6808355110236585353330312491411 64 [1 1 1]
[81172970061325300745418151513791123231408183778421072444458]
[93132872248942092128746134987383401160247785637965336436957]
[25109234200264937479050408273733489834306367309171742738264]
[10615984462510982136220066264213993637688731431166383403536 7]

nina:~/Documents/MAA/STAGE/programs vanessavitse$ more .master_key
58038078300777679823841215 6151
```

L'appel à `extract` inscrit dans le fichier `.private_key` la clé privée de l'utilisateur, i.e. l'abscisse et l'ordonnée de $S_{id} = [s]H_1(id)$.

```
nina:~/Documents/MAA/STAGE/programs vanessavitse$ extract
Enter your identity : vanessa

nina:~/Documents/MAA/STAGE/programs vanessavitse$ more .private_key
[43049734599590681689470908432788298032623249120370802994544]
[80055385917485984006474460513993146192431815378982281735039]
```

On fait enfin appel à `encrypt` et `decrypt`. Le message "This is the plaintext" est adressé à "vanessa". Le message chiffré $\langle C_1, C_2\rangle = \langle [t]P, M \oplus H_2(\hat{e}(P_{pub}, Q_{id})^t\rangle$ où $Q_{id} = H_1(id)$ est alors affiché : on a d'abord l'abscisse et l'ordonnée de C_1, puis C_2, le tout en base 64.

```
nina:~/Documents/MAA/STAGE/programs vanessavitse$ encrypt
Enter the identity of the person addressed : vanessa
Enter your message : This is the plaintext
Your message : This is the plaintext
Encrypted message :
7kjrCxfOW6jya9DnXJQ+VO7LjZoRAKNODwAA X/KsOCrgGiIFwDlAV77hWqBte9u+vSKoDwAA
u82S19/JF81AWVNFIlxtnBgcwYfv41EKwyfZQlz81w8WLXXl4gLzE8fwN/xxmS8dfVxlF/7RAH+tMTqBie5EuQ==

nina:~/Documents/MAA/STAGE/programs vanessavitse$ decrypt
Enter the encrypted message : 7kjrCxfOW6jya9DnXJQ+VO7LjZoRAKNODwAA
X/KsOCrgGiIFwDlAV77hWqBte9u+vSKoDwAA
u82S19/JF81AWVNFIlxtnBgcwYfv41EKwyfZQlz81w8WLXXl4gLzE8fwN/xxmS8dfVxlF/7RAH+tMTqBie5EuQ==
Decrypted message : This is the plaintext
```

A.3 Algorithmes pour la recherche de points engendrant la r-torsion

Dans cette partie, on explique comment trouver en pratique, suivant les valeurs du degré de plongement, des points qui engendrent la r-torsion.

1. *On suppose $k > 1$ et $r^2 \,||\, \#E(\mathbf{F}_{q^k})$.*
   On doit trouver $P \in E(\mathbf{F}_q)[r] \setminus \{O\}$ et $Q \in E(\mathbf{F}_{q^k})[r] \setminus E(\mathbf{F}_q)$.
   Pour trouver P, une méthode efficace consiste à choisir un point rationnel P_r au hasard sur la courbe (simple extraction de racine carrée dans $\mathbf{F}_q$), et à calculer $P = [\#E(\mathbf{F}_q)/r]\, P_r$. Le point P ainsi trouvé est clairement de r-torsion ; s'il est égal à O, on recommence avec un autre point P_r aléatoire. On peut déterminer la probabilité que $P \neq O$, ce qui revient à étudier le noyau de l'application $[\#E(\mathbf{F}_q)/r]$: $E(\mathbf{F}_q) \to E(\mathbf{F}_q)$.
   Comme $k > 1$, on a nécessairement $E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z}$ (cf lemme 2.3), et donc
   $$E(\mathbf{F}_q) \simeq \mathbf{Z}/n_1\mathbf{Z} \times \mathbf{Z}/n_2\mathbf{Z} \quad \text{avec} \quad r \wedge n_1 = 1.$$

On en déduit alors facilement que $\#\ker\left[\#E(\mathbf{F}_q)/r\right] = \dfrac{\#E(\mathbf{F}_q)}{r}$, et donc la probabilité de choisir P_r tel que $\left[\#E(\mathbf{F}_q)/r\right] P_r = O$ est égale à $1/r$, en particulier est négligeable.

Alg. 4 Algorithme probabiliste pour générer un point rationnel de r-torsion

ENTRÉE : E courbe elliptique définie sur $\mathbf{F}_q$ ($q = p^d$), r premier et k degré de plongement tel que $k > 1$ et $r^2 \,||\, \#E(\mathbf{F}_{q^k})$
SORTIE : $P \in E(\mathbf{F}_q)[r]$, $P \neq O$
  Calculer $\#E(\mathbf{F}_q)$
  $P \leftarrow O$
  tant que $P = O$ **faire**
    Tirer un point rationnel P_r au hasard sur la courbe
    $P \leftarrow \left[\#E(\mathbf{F}_q)/r\right] P_r$
  fin tant que
  retourner P

Pour trouver $Q \in E(\mathbf{F}_{q^k})[r] \setminus E(\mathbf{F}_q)[r]$, on peut de la même façon choisir un point Q_r au hasard dans $E(\mathbf{F}_{q^k})$ et calculer $\left[\#E(\mathbf{F}_{q^k})/r\right] Q_r$. Cependant, comme $E(\mathbf{F}_{q^k})[r] \simeq \mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r\mathbf{Z}$ (cf prop. 2.5) et $r^2 \,||\, \#E(\mathbf{F}_{q^k})$, on a

$$E(\mathbf{F}_{q^k}) \simeq \mathbf{Z}/n_1\mathbf{Z} \times \mathbf{Z}/n_2\mathbf{Z} \quad \text{avec} \quad r \,||\, n_1 \text{ et } r \,||\, n_2.$$

En particulier $\left[\#E(\mathbf{F}_{q^k})/r\right]$ est nulle sur $E(\mathbf{F}_{q^k})$. Par contre, si on considère l'application $\left[\#E(\mathbf{F}_{q^k})/r^2\right]$, on trouve avec un raisonnement similaire que

$$\#\ker\left[\#E(\mathbf{F}_{q^k})/r^2\right] = \frac{\#E(\mathbf{F}_{q^k})}{r^2}$$

et donc la probabilité que $\left[\#E(\mathbf{F}_{q^k})/r^2\right] Q_r = O$ est égale à $1/r^2$, soit une probabilité négligeable. Il reste à voir avec quelle probabilité on a $Q \in E(\mathbf{F}_{q^k}) \setminus E(\mathbf{F}_q)$. Etant donné que $E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z}$ et que $E(\mathbf{F}_{q^k})[r] \simeq \mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r\mathbf{Z}$, ceci est vérifié avec une probabilité égale à $1 - 1/r$.

Alg. 5 Algorithme probabiliste pour générer un point de r-torsion dans $E(\mathbf{F}_{q^k}) \setminus E(\mathbf{F}_q)$

ENTRÉE : E courbe elliptique définie sur $\mathbf{F}_q$ ($q = p^d$), r premier, k degré de plongement tel que $k > 1$ et $r^2 \,||\, \#E(\mathbf{F}_{q^k})$ et $\#E(\mathbf{F}_q)$
SORTIE : $Q \in E(\mathbf{F}_{q^k})[r]$, $Q \neq O$
  Calculer $\#E(\mathbf{F}_{q^k})$ à l'aide de la formule de récurrence sur les traces (voir rq 1.17)
  $Q \leftarrow O$
  tant que $Q \in E(\mathbf{F}_q)$ **faire**
    Tirer un point rationnel Q_r au hasard dans $E(\mathbf{F}_{q^k})$
    $Q \leftarrow \left[\#E(\mathbf{F}_{q^k})/r^2\right] Q_r$
  fin tant que
  retourner Q

2. *On suppose $k = 1$ et $r \,||\, \#E(\mathbf{F}_q)$.*

   On a alors nécessairement $E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z}$, on peut donc appliquer l'algorithme 4 pour trouver P qui engendre $E(\mathbf{F}_q)[r]$. On a vu qu'alors le couplage de Tate est non dégénéré sur $G_1 = \langle P \rangle$.

3. *On suppose $k = 1$, $r^2 \,||\, \#E(\mathbf{F}_q)$ et $E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r\mathbf{Z}$.*

   Pour les mêmes raisons que celles invoquées dans l'algorithme 5, l'application $[\#E(\mathbf{F}_q)/r]$ est nulle sur $E(\mathbf{F}_q)$. Avec un raisonnement similaire que celui mené dans le cas précédent, on peut trouver avec une forte probabilité ($= 1 - 1/r$) des points P et Q qui engendrent $E(\mathbf{F}_q)[r]$ grâce à l'algorithme 6.
   Remarque A.1. Pour vérifier que P et Q sont indépendants, on peut tester que leur couplage de Weil est différent de 1.

Alg. 6 Algorithme probabiliste pour générer deux points de r-torsion indépendants dans $E(\mathbf{F}_q)$

ENTRÉE : E courbe elliptique définie sur $\mathbf{F}_q$ ($q = p^d$), r premier tel que $r^2 \,||\#E(\mathbf{F}_q)$, $E(\mathbf{F}_q)[r] \simeq \mathbf{Z}/r\mathbf{Z} \times \mathbf{Z}/r\mathbf{Z}$, et de degré de plongement $k = 1$.
SORTIE : $P, Q \in E(\mathbf{F}_q)[r]$ indépendants
$P \leftarrow O$, $Q \leftarrow O$
tant que $P = O$ **faire**
Tirer un point rationnel P_r au hasard dans $E(\mathbf{F}_q)$
$P \leftarrow \left[\#E(\mathbf{F}_q)/r^2\right] P_r$
fin tant que
tant que P et Q colinéaires **faire**
Tirer un point rationnel Q_r au hasard dans $E(\mathbf{F}_q)$
$Q \leftarrow \left[\#E(\mathbf{F}_q)/r^2\right] Q_r$
fin tant que
retourner (P, Q)

A.4 Fonctions de hachage utilisées pour IBE

Pour le protocole `BasicIdent` présenté en section 3.5, on a besoin de deux fonctions de hachage $H_1 : \{0;1\}^* \to G_1$, et $H_2 : \mu_r \subset \mathbf{F}_{p^k}^* \to \{0;1\}^n$. Trouver de telles fonctions cryptographiquement sûres est un problème difficile. On présente ici les solutions retenues pour la programmation d'IBE, et qui n'ont pas la prétention d'être prouvées sûres.

- Pour H_1, on distingue deux cas :
  - cas supersingulier : la courbe est d'équation $y^2 = x^3 + 1$ sur $\mathbf{F}_p$ avec $p \equiv 2 \bmod 3$. On utilise SHA-512 fournie par le module `crypto++` [Dai], pour hacher dans $\{0;1\}^{512} \simeq \{0; \ldots; 2^{512} - 1\}$, puis on considère le résultat modulo p comme l'ordonnée d'un point de la courbe. On utilise ensuite la bijectivité de $x \mapsto x^3$ sur $\mathbf{F}_p$ pour trouver l'unique abscisse correspondante. On obtient ainsi un point R que l'on multiplie par $\#E(\mathbf{F}_p)/r = (p+1)/r$ pour trouver un point rationel de r-torsion.
  - cas ordinaire : on hache comme précédemment avec SHA-512 dans $\{0; \ldots; 2^{512} - 1\}$, puis on considère le résultat modulo p comme l'abscisse x d'un point de la courbe. On incrémente x jusqu'à ce que $x^3 + ax + b$ soit un résidu quadratique. On extrait la racine carrée pour trouver un point R que l'on multiplie par $\#E(\mathbf{F}_p)/r = (p + 1 - t)/r$ pour trouver un point rationel de r-torsion.
- Pour H_2, on se contente simplement de considérer la représentation sur la sortie écran des éléments de $\mathbf{F}_p^k$, i.e. une chaîne de caractères, puis on utilise SHA-512 pour la hacher dans $\{0;1\}^{512}$.

Références

[AM93] A. O. L. Atkin and F. Morain. Elliptic curves and primality proving. *Math. Comp.*, 61(203) :29–68, 1993.

[BF03] Dan Boneh and Matthew Franklin. Identity-based encryption from the Weil pairing. *SIAM J. Comput.*, 32(3) :586–615 (electronic), 2003.

[BK98] R. Balasubramanian and Neal Koblitz. The improbability that an elliptic curve has subexponential discrete log problem under the Menezes-Okamoto-Vanstone algorithm. *J. Cryptology*, 11(2) :141–145, 1998.

[BSS00] I. F. Blake, G. Seroussi, and N. P. Smart. *Elliptic curves in cryptography*, volume 265 of *London Mathematical Society Lecture Note Series*. Cambridge University Press, Cambridge, 2000. Reprint of the 1999 original.

[C$^+$] Henri Cohen et al. PARI/GP. http ://pari.math.u-bordeaux.fr.

[CFA$^+$06] Henri Cohen, Gerhard Frey, Roberto Avanzi, Christophe Doche, Tanja Lange, Kim Nguyen, and Frederik Vercauteren, editors. *Handbook of elliptic and hyperelliptic curve cryptography*. Discrete Mathematics and its Applications (Boca Raton). Chapman & Hall/CRC, Boca Raton, FL, 2006.

[CL] Florent Chabaud and Reynald Lercier. Zen, a toolbox for fast computation in finite extension over finite rings. http ://zenfact.sourceforge.net.

[Coh93] Henri Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.

[Dai] Wei Dai. Crypto++ library. http ://www.cryptopp.com.

[DE06] Régis Dupont and Andreas Enge. Provably secure non-interactive key distribution based on pairings. *Discrete Appl. Math.*, 154(2) :270–276, 2006.

[FR94] Gerhard Frey and Hans-Georg Rück. A remark concerning m-divisibility and the discrete logarithm in the divisor class group of curves. *Math. Comp.*, 62(206) :865–874, 1994.

[G$^+$] Torbjörn Granlund et al. GMP : GNU multiple precision arithmetic library. http ://gmplib.org.

[Gal05] Steven D. Galbraith. Pairings. In *Advances in elliptic curve cryptography*, volume 317 of *London Math. Soc. Lecture Note Ser.*, pages 183–213. Cambridge Univ. Press, Cambridge, 2005.

[GR04] Steven D. Galbraith and Victor Rotger. Easy decision Diffie-Hellman groups. *LMS J. Comput. Math.*, 7 :201–218 (electronic), 2004.

[Jou04] Antoine Joux. A one round protocol for tripartite Diffie-Hellman. *J. Cryptology*, 17(4) :263–276, 2004.

[Ler] Reynald Lercier. Package sea-1.0. pour ZEN. Communication personnelle.

[Mat00] Keith Matthews. The Diophantine equation $x^2 - Dy^2 = N$, $D > 0$. *Expo. Math.*, 18(4) :323–331, 2000.

[Mil86] Victor S. Miller. Short programs for functions on curves. In *IBM Thomas J. Watson Research Center*, 1986. http ://crypto.stanford.edu/miller/miller.ps.

[Mil04] Victor S. Miller. The Weil pairing, and its efficient calculation. *J. Cryptology*, 17(4) :235–261, 2004.

[MNT01] Atsuko Miyaji, Masaki Nakabayashi, and Shunzo Takano. New explicit conditions of elliptic curve traces for FR-reductions. *IEICE Trans. Fundamentals*, E84(5) :1234–1243, 2001.

[MOV93] Alfred J. Menezes, Tatsuaki Okamoto, and Scott A. Vanstone. Reducing elliptic curve logarithms to logarithms in a finite field. *IEEE Trans. Inform. Theory*, 39(5) :1639–1646, 1993.

[Sha85] Adi Shamir. Identity-based cryptosystems and signature schemes. In *Advances in cryptology (Santa Barbara, Calif., 1984)*, volume 196 of *Lecture Notes in Comput. Sci.*, pages 47–53. Springer, Berlin, 1985.

[Sho] Victor Shoup. NTL : A library for doing number theory. http ://www.shoup.net/ntl.

[Sil86] Joseph H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1986.

[SOK00] Ryuichi Sakai, Kiyoshi Ohgishi, and Masao Kasahara. Cryptosystems based on pairing. In *Proc. of SCIS 2000*, 2000.

[SX95] Henning Stichtenoth and Chao Ping Xing. On the structure of the divisor class group of a class of curves over finite fields. *Arch. Math. (Basel)*, 65(2) :141–150, 1995.

[Ver04] Eric R. Verheul. Evidence that XTR is more secure than supersingular elliptic curve cryptosystems. *J. Cryptology*, 17(4) :277–296, 2004.

[Wat69] William C. Waterhouse. Abelian varieties over finite fields. *Ann. Sci. École Norm. Sup. (4)*, 2 :521–560, 1969.

[Zó0] Gilles Zémor. *Cours de cryptographie*, volume 6 of *Enseignement des Mathématiques*. Cassini, Paris, 2000.