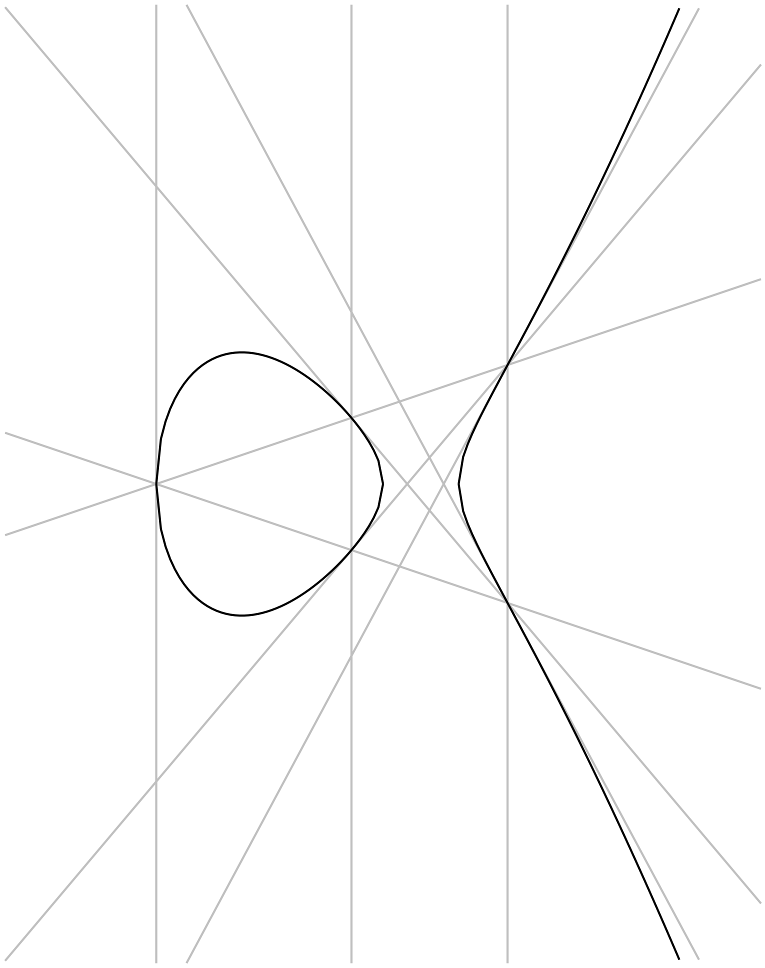
Applications des couplages sur les courbes elliptiques en cryptologie

Mémoire de stage de D.E.A.
27 septembre 2002



Régis Dupont,
Laboratoire d'Informatique de l'École polytechnique
`dupont@lix.polytechnique.fr`

Remerciements

Je tiens à remercier l'ensemble du LIX pour leur accueil. Effectuer mon stage dans ce laboratoire fut un réel plaisir et l'occasion d'apprendre énormément. Je suis particulièrement reconnaissant envers les membres de l'équipe crypto : Andreas Enge et François Morain, qui m'ont guidé tout au long de ce stage, mais aussi Mireille Fouquet, Pierrick Gaudry, Nicolas Gürel et Emmanuel Thomé qui ont tous répondu avec patience à mes nombreuses questions.

Table des matières

Introduction

En 1976, Diffie et Hellman [19] révolutionnaient le monde de la cryptographie en proposant un protocole de partage de clé qui constitue historiquement le premier exemple de cryptosystème à clé publique. La sécurité de leur protocole repose sur la difficulté du problème dit problème de Diffie–Hellman (DH). Il s'agit, dans un groupe cyclique $G = \langle g \rangle$, de calculer g^{ab} à partir de g^a et de g^b. Ce problème est lié au problème du logarithme discret (DL) dans G (étant donné $h \in G$, déterminer un entier x tel que $h = g^x$).

Bien que le cryptosystème à clé publique le plus connu du grand public, RSA [2], repose sur la difficulté de la factorisation des entiers, il existe de nombreux autres systèmes basés sur le problème du logarithme discret, l'un des plus connus étant sans doute le cryptosystème d'ElGamal [22] permettant chiffrement et signature.

Les premiers groupes utilisés dans de tels protocoles étaient les groupes multiplicatifs de corps finis. Aujourd'hui, même si on ne connait toujours pas d'algorithme en temps polynomial pour y résoudre le problème du logarithme discret, il existe pour ce faire des algorithmes sous-exponentiels.

Les courbes elliptiques ont été introduites en cryptographie en 1985 indépendemment par Koblitz [42] et Miller [56], qui proposent d'utiliser les groupes associés dans des cryptosystèmes basés sur le logarithme discret. En effet tous les algorithmes connus à ce jour pour résoudre le problème du logarithme discret sur des courbes elliptiques génériques sont exponentiels. Ceci permet, dans les systèmes utilisant les courbes elliptiques, d'utiliser des clés de taille bien moindre que pour les systèmes basés sur la difficulté de la factorisation ou du logarithme discret dans les corps finis. Cet avantage est sans doute pour beaucoup dans l'essor actuel des cryptosystèmes basés sur les courbes elliptiques.

Les couplages de Weil et de Tate sur les courbes elliptiques sont des outils bien connus des mathématiciens, qui permettent de démontrer un grand nombre de propriétés des courbes elliptiques sur les corps finis. En 1993, Menezes, Okamoto et Vanstone ont montré comment, en utilisant le couplage de Weil, on pouvait réduire le problème du logarithme discret sur une certaine classe de courbes elliptiques au même problème dans une extension de faible degré de leur corps de base. Plus récemment, Frey et Rück [28] ont montré qu'une plus grande classe de courbes étaient susceptibles à une attaque du même type, mais utilisant le couplage de Tate.

En 2000, Sakai, Ohgishi et Kasahara [59] et Joux [39] proposent indépendemment des cryptosystèmes utilisant les couplages sur les courbes elliptiques. Le protocole imaginé par Joux est un analogue du protocole de Diffie–Hellman permettant à trois entités de créer une clé secrète commune. Sakai, Ohgishi et Kasahara étudient l'utilisation des couplages dans des cryptosystèmes basés sur l'identité, notion introduite en 1984 par Shamir [62]. Il s'agit de systèmes à clé publique où la clé publique de chaque participant se déduit directement de son identité, rendant ainsi inutile sa certification par une authorité de confiance. Un certain nombre de protocoles de signature basés sur l'identité étaient déjà connus (le premier ayant été proposé par Shamir [62]), mais aucun protocole de chiffrement ni de distribution de clé basé sur l'identité utilisable n'avait encore été proposé. Sakai, Ohgishi et Kasahara proposent eux un protocole de création de clé commune et un protocole de signature basés sur l'identité, mais leur travail restera longtemps méconnu d'une partie de la communauté des cryptologues. Nous avons d'ailleurs redédouvert l'un de leurs protocoles et en avons, avec Andreas Enge, trouvé une preuve de sécurité [20] avant d'avoir connaissance de leurs travaux.

En 2001 Boneh et Franklin [10], et Cocks [16] proposent indépendemment deux protocoles de chiffrement basés sur l'identité, le premier utilisant le couplage de Weil et le second utilisant des propriétés de résiduosité quadratique. Les couplages de Weil et de Tate et leur utilisation dans les cryptosystèmes attirent alors l'attention de nombreux chercheurs, et de nouveaux protocoles utilisant les couplages ou les systèmes basés sur l'identité sont présentés chaque mois ou presque depuis [3, 9, 11, 13, 14, 32, 36, 37, 48, 49, 58, 65, 68].

Parallèlement, ce regain d'intérêt pour les couplages sur les courbes elliptiques a conduit à la recherche de courbes sur lesquelles les couplages de Weil ou de Tate sont efficacement calculables [7, 21, 30, 57], ainsi qu'à la recherches d'algorithmes efficaces de calcul des couplages [6, 31].

Ce mémoire présente les travaux effectué durant mon stage de DEA, qui ont concerné d'une part l'étude de différents protocoles basés sur l'identité, et d'autre part l'étude des courbes sur lesquelles les couplages de Weil ou de Tate sont efficacement calculables et les méthodes de génération de telles courbes.

Le premier chapitre introduit les couplages de Weil et de Tate sur les courbes elliptiques. Quelques applications des couplages sur les courbes elliptiques (notamment en cryptanalyse) sont décrites dans le chapitre 2. Le chapitre 3 aborde la notion de cryptosystèmes basés sur l'identité et en donne un certain nombre d'exemples utilisant les couplages, ainsi qu'un exemple de preuve de sécurité pour un tel système (que nous avons trouvé durant ce stage). Le chapitre 4 étudie la génération de courbes ordinaires sur lesquelles les couplages de Weil ou de Tate sont efficacement calculables, et présente une méthode originale pour ce faire. Enfin le chapitre 5 décrit une implantation du calcul de couplages que j'ai réalisée durant ce stage. En annexe figurent quelques protocoles basés sur l'identité n'utilisant pas les couplages, ainsi que quelques exemples numériques de courbes elliptiques sur lesquelles les couplages de Weil ou de Tate sont efficacement calculables.

Conventions et notations

Dans tout ce rapport excepté le chapitre 4, $\mathbb{F}_q$ désigne un corps fini à q éléments de caractéristique p avec $q = p^n$, et $\overline{\mathbb{F}}_q$ une clôture algébrique de $\mathbb{F}_q$. $\mathbb{K}$ est un corps intermédiaire quelconque entre $\mathbb{F}_q$ et $\overline{\mathbb{F}}_q$ (noter que l'on peut avoir $\mathbb{K} = \mathbb{F}_q$ ou $\mathbb{K} = \overline{\mathbb{F}}_q$).

E est l'équation (affine) d'une courbe elliptique à coefficients dans $\mathbb{F}_q$ (E est donc un élément de l'anneau $\mathbb{F}_q[x, y]$). Rappelons qu'une telle équation prend la forme :

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

où a_1, a_2, a_3, a_4 et a_6 sont des éléments de $\mathbb{F}_q$.

On notera

$$E(\mathbb{K}) = \{(x, y) \in \mathbb{K}^2 \mid E(x, y) = 0\} \cup \{\infty\}.$$

On rappelle que $E(\mathbb{K})$ est doté d'une structure de groupe abélien, dont l'élément neutre ∞ est communément appelé *point à l'infini.*

Nous n'exposons pas ici l'expression de la loi de groupe sur $E(\mathbb{K})$. Nous renvoyons à [8, 23, 54, 64] pour une introduction détaillée aux courbes elliptiques.

Chapitre 1

Couplages de Weil et de Tate sur les courbes elliptiques

1.1 Courbes elliptiques sur les corps finis

1.1.1 Endomorphisme de Frobenius

On appelle endomorphisme de Frobenius associé à $E(\mathbb{F}_q)$ l'application définie par :

$$\begin{array}{rccc} \pi : & E(\overline{\mathbb{F}}_q) & \to & E(\overline{\mathbb{F}}_q) \\ & (x, y) & \mapsto & (x^q, y^q). \end{array}$$

Le théorème suivant nous donne des informations sur la forme de son polynôme caractéristique :

Théorème 1.1 (Hasse) *Le polynôme caractéristique χ de l'endomorphisme de Frobenius π s'écrit :*

$$\chi(X) = X^2 - tX + q.$$

De plus, la trace t de π vérifie l'inégalité :

$$|t| \leqslant 2\sqrt{q}$$

Démonstration : la preuve originale figure dans [35]. Une preuve classique figure dans [64], théorème V.1.1.

♣

Comme $E(\mathbb{F}_q) = \ker(1 - \pi)$ et que l'endomorphisme $1 - \pi$ est séparable (voir encore [64]) on a $\#E(\mathbb{F}_q) = \chi(1)$, d'où l'inégalité :

$$|q + 1 - \#E(\mathbb{F}_q)| \leqslant 2\sqrt{q}.$$

On en déduit que $|\log q - \log \#E(\mathbb{F}_q)| \leqslant 1$, ce qui permet de rapidement estimer l'ordre de grandeur du cardinal d'une courbe.

Soit $\mathbb{F}_{q^n}$ une extension finie de $\mathbb{F}_q$. On note π_n l'endomorphisme de Frobenius associé à $E(\mathbb{F}_{q^n})$, et χ_n son polynôme caractéristique. Classiquement, le cardinal de $E(\mathbb{F}_{q^n})$ se déduit de celui de $E(\mathbb{F}_q)$ *via* le résultat suivant :

Théorème 1.2 (Weil) *Soient α et β les deux racines de χ dans $\mathbb{C}$. On a l'égalité :*

$$\chi_n(X) = (X - \alpha^n)(X - \beta^n)$$

Démonstration : Ce théorème, qui est un cas particulier d'un résultat plus général, a (dans le cas des courbes elliptiques auquel on se restreint ici) été énoncé et prouvé par Weil dans [67]. On en trouvera une preuve assez élémentaire dans [23].

♣

En pratique, on calcule le polynôme χ_n (et donc le cardinal de $E(\mathbb{F}_{q^n})$) *via* l'égalité suivante :

$$\chi_n(Y) = \mathrm{Res}_X\left(\chi(X), Y - X^n\right).$$

1.1.2 Courbes supersingulières

Théorème 1.3 *Avec les notations introduites dans le paragraphe précédent, les conditions suivantes sont équivalentes :*

1. *l'anneau d'endomorphismes* $\mathrm{End}(E(\overline{\mathbb{F}}_q))$ *est non commutatif (en particulier c'est un ordre dans une algèbre de quaternions),*
2. *$E(\overline{\mathbb{F}}_q)$ n'a pas de points d'ordre p,*
3. *$p \mid t$,*
4. *$p^{\lceil n/2 \rceil} \mid t$,*
5. *il existe un entier k tel que $\pi^k = \pm q^{k/2}$.*

Démonstration : voir [64], théorème V.3.1, ainsi que la preuve du théorème 6 de [30].

♣

Si E vérifie ces propriétés, elle sera dite *supersingulière*. Dans le cas contraire, on dira que E est *ordinaire*.

1.1.3 Points de torsion

Pour tout entier $r \in \mathbb{N}$, on peut définir l'endomorphisme de $E(\overline{\mathbb{F}}_q)$ noté $[r]$ par :

$$\forall P \in E(\overline{\mathbb{F}}_q), [r]P = \overbrace{P + P + \ldots + P}^{r \text{ fois}}.$$

Cette définition se généralise naturellement aux cas où $r \in \mathbb{Z}$ en posant, pour $r < 0$: $[r]P = -[-r]P$. L'anneau $\mathbb{Z}$ s'envoie donc naturellement dans $\mathrm{End}(E(\overline{\mathbb{F}}_q))$.

La proposition II.3.1 de [64] implique que $[r]$ est surjective.

Pour $r \in \mathbb{N}$, on définit l'ensemble des points de r-torsion de $E(\overline{\mathbb{F}}_q)$, noté $E[r]$, comme étant le noyau de $[r]$. Si $\mathbb{K}$ est un corps intermédiaire entre $\mathbb{F}_q$ et $\overline{\mathbb{F}}_q$, on pose :

$$E(\mathbb{K})[r] = E[r] \cap E(\mathbb{K}).$$

La proposition suivante nous renseigne sur la structure du groupe $E[r]$:

Proposition 1.1 *Soit $r \in \mathbb{N}$,*
- *si $p \nmid n$, alors $E[r]$ est isomorphe à $\mathbb{Z}/r\mathbb{Z} \oplus \mathbb{Z}/r\mathbb{Z}$ en tant que groupe abélien,*
- *si r est une puissance de p, alors $E[r] = \{\infty\}$ si E est supersingulière, et $E[r]$ est isomorphe à $\mathbb{Z}/r\mathbb{Z}$ si E est ordinaire.*

On définit le *degré MOV* de $E(\mathbb{F}_q)$ relativement à r[1] comme étant le plus petit entier k tel que $E[r] = E(\mathbb{F}_{q^k})[r]$.

Soit ℓ un diviseur premier de $\#E(\mathbb{F}_q)$, on a le résultat suivant :

Théorème 1.4 *Si $\ell \neq p$ et $\ell \nmid q-1$, alors :*

$$E[\ell] = E(\mathbb{F}_{q^k})[\ell] \Leftrightarrow \ell \mid q^k - 1.$$

Démonstration : ce résultat est dû à Balasubramanian et Koblitz [5].

♣

Ce résultat est particulièrement intéressant en ce qui concerne le calcul du degré MOV, puisque si ℓ vérifie les hypothèses, le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ n'est autre que l'ordre de q modulo ℓ.

Un théorème dû à Menezes, Okamoto et Vanstone [53] donne une borne supérieure au degré MOV d'une courbe supersingulière relativement à tout diviseur premier de son cardinal :

Théorème 1.5 *Si E est supersingulière et que ℓ est un diviseur premier de son cardinal ($\ell \neq p$, $\ell \nmid q-1$), alors le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ est inférieur ou égal à 6.*

Démonstration : Cette preuve n'est pas la preuve originale de Menezes, Okamoto et Vanstone, mais une preuve plus récente due à Galbraith [30], qui a l'avantage de se généraliser aux courbes hyperelliptiques.

Supposons E supersingulière. Le polynôme caractéristique du Frobenius associé à $E(\mathbb{F}_q)$ s'écrit :

$$\chi(X) = X^2 - tX + p^n.$$

Supposons pour l'instant que n est pair, avec $n = 2n'$. Comme la courbe est supersingulière, le théorème 1.3 (point 4) montre que $p^{n'} \mid t$. On considère alors le polynôme

$$\chi'(X) = \chi(p^{n'}X)/p^n.$$

D'après ce qui précède, χ' est à coefficients entiers. Si α et β désignent les racines de χ (qui sont deux complexes conjugués de module $p^{n'}$), alors les racines de χ' sont $\alpha/p^{n'}$ et $\beta/p^{n'}$, qui sont deux nombres algébriques de module 1, donc des racines de l'unité. χ' est donc égal soit à $(X+1)^2$, soit à un polynôme cyclotomique de degré 2. On en déduit qu'il existe un entier $k \leqslant 6$ tel que $\alpha^k = \beta^k = p^{kn'}$. Le polynôme caractéristique du Frobenius sur $E(\mathbb{F}_{q^k})$ s'écrit alors

$$\chi_k(X) = (X - \alpha^k)(X - \beta^k) = (X - p^{kn'})^2.$$

[1] Dans les applications cryptographiques, on travaille généralement dans un grand sous-groupe d'ordre premier de $E(\mathbb{F}_q)$. On parlera alors de degré MOV (sans plus de précisions) pour désigner le degré MOV de $E(\mathbb{F}_q)$ relativement au plus grand facteur premier de $\#E(\mathbb{F}_q)$.

Le cardinal de $E(\mathbb{F}_{q^k})$ est donc égal à $(p^{kn'}-1)^2$. Soit maintenant ℓ un diviseur premier du cardinal de $E(\mathbb{F}_q)$, alors ℓ divise également le cardinal de $E(\mathbb{F}_{q^k})$, et donc $\ell \mid p^{kn'}-1 \mid q^k-1$, ce qui montre que le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ est inférieur ou égal à 6.

Dans le cas où n est impair, on commence par poser $N = 2n$, et on montre comme ci-dessus (en considérant la courbe $E(\mathbb{F}_{q^2})$) qu'il existe un entier $k \leqslant 6$ tel que le cardinal de $E(\mathbb{F}_{q^{2k}})$ soit égal à $(p^{kn}-1)^2 = (q^k-1)^2$. Le même argument que précédemment permet de conclure.

♣

1.1.4 Diviseurs sur une courbe elliptique

On définit le groupe des *diviseurs* de $E(\mathbb{F}_q)$, que l'on note $\mathrm{Div}_{\mathbb{F}_q}(E)$, comme étant le groupe libre engendré par les points de la courbe. Ses éléments sont donc des sommes finies de points de $E(\mathbb{F}_q)$. Un diviseur s'écrit :

$$D = \sum_{P_i \in E(\mathbb{F}_q)} n_i(P_i)$$

où les n_i sont des entiers relatifs presque tous nuls. Les points P_i sont notés entre parenthèses afin de ne pas confondre les diviseurs $(P+Q)$ et $(P)+(Q)$.

On définit le *support* de D comme étant l'ensemble :

$$\mathrm{supp}(D) = \{P_i \mid n_i \neq 0\}.$$

Le *degré* d'un diviseur est défini comme étant la somme de ses coefficients :

$$\deg(D) = \sum_{P_i \in E(\mathbb{F}_q)} n_i.$$

On note $\mathrm{Div}^0_{\mathbb{F}_q}(E)$ le sous-groupe de $\mathrm{Div}_{\mathbb{F}_q}(E)$ formé des diviseurs de degré nul.

Soit K le corps des fonctions de la courbe ($K = \mathbb{F}_q(x,y)/(f(x,y))$, où $f(x,y) = 0$ est une équation affine de la courbe $E(\mathbb{F}_q)$), et soit $P \in E(\mathbb{F}_q)$. L'ensemble des fonctions de K qui sont définies en P forme un anneau de valuation discrète noté $\mathcal{O}_P$ [64]. Soit M_P l'idéal maximal de $\mathcal{O}_P$ (*i.e.* l'idéal formé des fonctions s'annulant en P). La valuation discrète v_P de $\mathcal{O}_P$ est définie par :

$$\forall g \in \mathcal{O}_P, v_P(g) = \max\{i \in \mathbb{N} \mid g \in M_P^i\}.$$

Si g est une fonction non nulle de K, et $P \in E(\mathbb{F}_q)$, on définit l'ordre de f en P noté $\mathrm{ord}_P(g)$, par :
- si g est définie et non nulle en P, $\mathrm{ord}_P(g) = 0$,
- si g est définie en P et $g(P) = 0$, $\mathrm{ord}_P(g) = v_P(g)$,
- si g a un pôle en P, $\mathrm{ord}_P(g) = -v_P(1/g)$.

$|\mathrm{ord}_P(g)|$ correspond en fait à la multiplicité de P en tant que zéro (si $\mathrm{ord}_P(g) > 0$) ou que pôle (si $\mathrm{ord}_P(g) < 0$) de g.

La proposition suivante nous permet d'associer à toute fonction $g \in K$ un diviseur :

Proposition 1.2 *Pour toute fonction g de K, le diviseur*

$$\sum_{P_i \in E(\mathbb{F}_q)} \mathrm{ord}_{P_i}(g)(P_i)$$

est un diviseur de degré nul de $E(\mathbb{F}_q)$, que l'on note $\mathrm{div}(g)$.

Démonstration : voir [64], proposition II.3.1.

♣

Un diviseur D pouvant s'écrire sous la forme $D = \mathrm{div}(g)$, pour $g \in K$, est dit *principal.* L'ensemble des diviseurs principaux de $E(\mathbb{F}_q)$ forme un sous-groupe de $\mathrm{Div}^0_{\mathbb{F}_q}(E)$ noté $\mathrm{Prin}_{\mathbb{F}_q}(E)$. Pour toutes fonctions g et h de K, on a les égalités :

$$\mathrm{div}(gh) = \mathrm{div}(g) + \mathrm{div}(h)$$

$$\mathrm{div}(1/g) = -\mathrm{div}(g)$$

Le théorème qui suit établit un lien entre les diviseurs et la structure de groupe de $E(\mathbb{F}_q)$:

Théorème 1.6 $E(\mathbb{F}_q)$ *est isomorphe, en tant que groupe abélien, au groupe quotient*

$$\mathrm{Div}^0_{\mathbb{F}_q}(E)/\mathrm{Prin}_{\mathbb{F}_q}(E).$$

Démonstration : voir [64], proposition III.3.4.

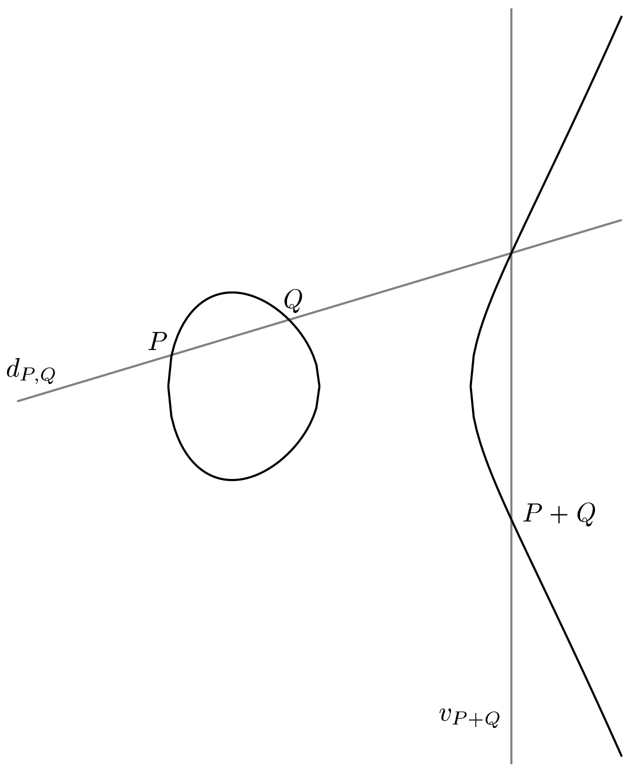
♣

Soient P et Q deux points de $E(\mathbb{F}_q)$. On a alors les égalités :

$$\mathrm{div}(v_P) = (P) + (-P) - 2(\infty) \tag{1.1}$$

$$\mathrm{div}(d_{P,Q}) = (P) + (Q) + (-P-Q) - 3(\infty), \tag{1.2}$$

où $d_{P,Q}$ est une équation de la droite passant par les points P et Q (de la tangente à $E(\mathbb{F}_q)$ en P si $P = Q$), et v_P est une équation de la droite passant par les points P et $-P$ (qui n'est autre que la verticale passant par P). Ceci est illustré sur la figure suivante :



Étant donnés un diviseur $D = \sum_{P_i \in E(\mathbb{F}_q)} n_i(P_i)$ et une fonction $g \in K$, on définit l'évaluation de g en D par :

$$g(D) = g\left(\sum_{P_i \in E(\mathbb{F}_q)} n_i(P_i)\right) = \prod_{P_i \in E(\mathbb{F}_q)} g(P_i)^{n_i}.$$

1.2 Définition et propriétés du couplage de Weil

Le nombre r désigne dans cette partie un entier positif premier avec p. On rappelle que dans ce cas, $E[r]$ est isomorphe à $\mathbb{Z}/r\mathbb{Z} \times \mathbb{Z}/r\mathbb{Z}$.

1.2.1 Définition

Soit $T \in E[r]$, et soit $T' \in E(\overline{\mathbb{F}}_q)$ tel que $[r]T' = T$ (un tel T' existe de par la surjectivité de $[r]$). On considère le diviseur D défini par :

$$D = \sum_{R \in E[r]} (T' + R) - (R).$$

Comme $\#E[r] = r^2$ et que $[r^2]T' = \infty$, on déduit du théorème 1.6 le fait que le diviseur D est principal, et donc qu'il existe une fonction g du corps de fonctions de $E(\overline{\mathbb{F}}_q)$ telle que $\mathrm{div}(g) = D$.

Soit $S \in E[r]$, on définit le couplage de Weil e_r de S et T par :

$$e_r(S, T) = \frac{g(X + S)}{g(X)}$$

où X est un point quelconque de $E(\overline{\mathbb{F}}_q)$ tel que les valeurs $g(X + S)$ et $g(X)$ soient définies et non nulles.

1.2.2 Propriétés

Proposition 1.3 *Le couplage de Weil e_r est une application*

$$e_r : E[r] \times E[r] \mapsto \mu_{\mathbf{r}}$$

où $\mu_{\mathbf{r}}$ désigne le sous-groupe de $\overline{\mathbb{F}}_q$ constitué des racines $r^{\text{èmes}}$ de l'unité. De plus, e_r possède les propriétés suivantes :

1. *bilinéarité :*
$$\begin{aligned} \forall S, S_1, S_2, T, T_1, T_2 \in E[r] \quad e_r(S_1 + S_2, T) &= e_r(S_1, T)e_r(S_2, T) \\ e_r(S, T_1 + T_2) &= e_r(S, T_1)e_r(S, T_2), \end{aligned}$$
2. *identité :*
$$\forall S \in E[r],\ e_r(S, S) = 1,$$
3. *e_r est alternée :*
$$\forall S, T \in E[r],\ e_r(S, T) = e_r(T, S)^{-1},$$
4. *non dégénérescence :*
$$\begin{aligned} (\forall S \in E[r],\ e_r(S, T) = 1) &\Leftrightarrow T = \infty \\ (\forall T \in E[r],\ e_r(S, T) = 1) &\Leftrightarrow S = \infty. \end{aligned}$$

Démonstration :

On commence par montrer la linéarité à gauche : soient T et g comme dans la définition de e_r, et soient S_1 et S_2 deux points de $E[r]$. Soit alors $X \in E(\overline{\mathbb{F}}_q)$ un point tel que les valeurs $g(X+S_1+S_2)$, $g(X+S_1)$ et $g(X)$ soient toutes trois définies et non nulles. On a :

$$\begin{array}{rcl} e_r(S_1+S_2,T) & = & \frac{g(X+S_1+S_2)}{g(X)} \\ & = & \frac{g(X+S_1+S_2)}{g(X+S_1)}\frac{g(X+S_1)}{g(X)} \\ & = & e_r(S_2,T)e_r(S_1,T) \end{array}$$

ce qui prouve la linéarité à gauche. On en déduit directement que, si S et T sont des points de r-torsion, alors :

$$\begin{array}{rcl} e_r(S,T)^r & = & e_r([r]S,T) \\ & = & e_r(\infty,T) \\ & = & 1 \end{array}$$

ce qui prouve que e_r est bien à valeurs dans les racines $r^{\text{èmes}}$ de l'unité.

Le reste de la démonstration nécessite l'introduction d'un certain nombre de notions plus avancées qui ne nous seront pas utiles par la suite. Nous renvoyons donc à [64], paragraphe III.8 ou à [23], paragraphe 3.7 pour une preuve intégrale.

♣

Il est important de noter que l'introduction du couplage de Weil (et la preuve de ses propriétés) ne nécessite que très peu des résultats que nous avons énoncés précédemment. Le couplage de Weil est en fait un outil puissant dans la preuve des propriétés des groupes associés aux courbes elliptiques sur des corps finis, en particulier pour déterminer la structure du groupe associé, ou pour prouver le théorème de Hasse [23].

1.2.3 Définition alternative

Il existe une seconde définition du couplage de Weil e_r, plus symétrique, que l'on introduit ici car l'algorithme de calcul du couplage introduit en 1.5.1 est basé sur cette seconde définition.

Soient P et Q deux points de $E[r]$. Soient A et B deux diviseurs de degré nul tels que A (resp. B) soit équivalent à $(P)-(\infty)$ (resp. à $(Q)-(\infty)$) dans $\mathrm{Div}^0_{\overline{\mathbb{F}}_q}(E)/\mathrm{Prin}_{\overline{\mathbb{F}}_q}(E)$ (on prendra par exemple $A=(P+S)-(S)$ et $B=(Q+T)-(T)$ avec S et T des points quelconques de $E(\overline{\mathbb{F}}_q)$). On suppose de plus que A et B sont à supports disjoints.

Comme P et Q sont des points de r-torsion, il est clair que $rA \in \mathrm{Prin}_{\overline{\mathbb{F}}_q}(E)$ (et de même pour B). Il existe donc deux fonctions f_A et f_B telles que $rA = \mathrm{div}(f_A)$ et $rB = \mathrm{div}(f_B)$. On a alors l'égalité :

$$e_r(P,Q) = \frac{f_A(B)}{f_B(A)}.$$

Démonstration : la preuve de l'équivalence de cette définition du couplage de Weil avec celle donnée précédemment se trouve dans [64], exercice 3.16.

♣

1.3 Définition et propriétés du couplage de Tate

Dans cette partie encore, r désigne un entier premier avec p, et k désigne l'ordre de q modulo r, qui n'est pas nécessairement égal au degré MOV de $E(\mathbb{F}_q)$ relativement à r quand $r \nmid q-1$. On supposera de plus que r divise $\#E(\mathbb{F}_{q^k})$, de sorte que $E(\mathbb{F}_{q^k})$ contienne au moins r points de r-torsion (mais pas nécessairement les r^2 points de $E[r]$).

On rappelle qu'alors, le groupe $\mathbb{F}_{q^k}^\times$ contient les racines $r^{\text{èmes}}$ de l'unité.

1.3.1 Définition

Le couplage de Tate (associé à r) est une application :

$$\tilde{e}_r : E(\mathbb{F}_{q^k})[r] \times E(\mathbb{F}_{q^k})/[r]E(\mathbb{F}_{q^k}) \to \mathbb{F}_{q^k}^\times / \left(\mathbb{F}_{q^k}^\times\right)^r .$$

Soit $P \in E(\mathbb{F}_{q^k})[r]$, et soit $Q \in E(\mathbb{F}_{q^k})$. Il existe une fonction $\mathbb{F}_{q^k}$-rationnelle f_P telle que :

$$\mathrm{div}(f_P) = r(P) - r(\infty).$$

Il existe un diviseur D équivalent à $(Q)-(\infty)$ (par exemple $D = (Q+T)-(T)$ avec T un point quelconque de $E(\mathbb{F}_{q^k})$) dont le support est disjoint du support de f_P. On définit alors le couplage de Tate entre P et Q par :

$$t_r(P,Q) = f_P(D).$$

Il est souvent préférable d'utiliser une définition légèrement différente :

$$t_r(P,Q) = f_P(D)^{\frac{q^k-1}{r}},$$

qui permet d'avoir un couplage à valeurs dans les racines $r^{\text{èmes}}$ de l'unité. Dans la suite, on utilisera cette seconde définition du couplage de Tate.

1.3.2 Propriétés

Proposition 1.4 *Le couplage de Tate t_r est une application*

1. *bien définie,*
2. *bilinéaire,*
3. *non dégénérée, i.e. pour tout point $P \in E(\mathbb{F}_{q^k})[r] \setminus \{\infty\}$, il existe un point $Q \in E(\mathbb{F}_{q^k})$ tel que $t_r(P,Q) \neq 1$.*

Démonstration : voir [28].

♣

1.4 Couplages de Weil et de Tate modifiés

1.4.1 Couplage de Weil modifié

Supposons la courbe $E(\mathbb{F}_q)$ supersingulière, et notons encore k son degré MOV relativement à un diviseur premier ℓ de son cardinal. Il existe alors parfois un morphisme

$$\Psi : E(\mathbb{F}_{q^k}) \to E(\mathbb{F}_{q^k})$$

non $\mathbb{F}_q$-rationnel. Ainsi, si P désigne un générateur de $E(\mathbb{F}_q)[\ell]$, P et $\Psi(P)$ engendrent $E[\ell]$.

On définit alors le couplage de Weil modifié (implicitement associé à Ψ) :

$$\hat{e}_\ell : \begin{array}{ccc} E[\ell] \times E[\ell] & \rightarrow & \mathbb{F}_{q^k}^\times \\ (A, B) & \mapsto & e_\ell(A, \Psi(B)) \end{array}$$

qui vérifie bien sûr toutes les propriétés du couplage de Weil ordinaire, mais pour lequel $\hat{e}_\ell(P,P) \neq 1$.

Des morphismes tels que Ψ sont parfois aussi appelés *distortions.* Des exemples numériques de courbes supersingulières et distortions associées sont donnés en annexe.

On notera que de tels morphismes ne peuvent exister dans le cas où $E(\mathbb{F}_q)$ est une courbe ordinaire. En effet, d'après le théorème 1.3 (point 1), l'anneau d'endomorphismes de $E(\mathbb{F}_q)$ est alors commutatif. En particulier, si un tel morphisme Ψ existait, il commuterait avec le Frobenius π, ce qui est impossible (les points de $E(\mathbb{F}_q)$ étant justement les points fixes de π).

1.4.2 Couplage de Tate modifié

Soit ℓ un diviseur premier de $\#E(\mathbb{F}_q)$, et soit k l'ordre de q modulo ℓ. On a vu que l'on peut alors définir le couplage de Tate

$$t_\ell : E(\mathbb{F}_{q^k})[\ell] \times E(\mathbb{F}_q)/\ell E(\mathbb{F}_q) \rightarrow \mu_\ell.$$

La proposition qui suit va nous permettre de montrer que, sous certaines conditions, t_ℓ peut être vu comme une application

$$t_\ell : E(\mathbb{F}_{q^k})[\ell] \times E(\mathbb{F}_{q^k})[\ell] \rightarrow \mu_\ell.$$

Proposition 1.5 *Soit ℓ un diviseur premier de $\#E(\mathbb{F}_q)$. Supposons de plus que $E(\mathbb{F}_q)$ ne contient pas de point non nul de ℓ^2-torsion. Alors l'application $\Psi : E(\mathbb{F}_q)[\ell] \rightarrow E(\mathbb{F}_q)/\ell E(\mathbb{F}_q)$ qui à un point $P \in E(\mathbb{F}_q)[\ell]$ associe sa classe dans $E(\mathbb{F}_q)/\ell E(\mathbb{F}_q)$ est une bijection.*

Démonstration : Les deux groupes ayant même cardinal, il suffit de montrer l'injectivité de Ψ. Soient P et Q deux point de $E(\mathbb{F}_q)[\ell]$ tels que $\Psi(P) = \Psi(Q)$. Il existe donc un point $R \in E(\mathbb{F}_q)$ tel que $P = Q + [\ell]R$. P et Q étant des points de ℓ-torsion, on a donc $[\ell^2]R = \infty$. Si P et Q étaient distincts, $[\ell]R$ serait non nul, et R serait donc un point d'ordre ℓ^2 de $E(\mathbb{F}_q)$, ce qui ne peut être. On en déduit l'injectivité et donc la surjectivité de Ψ.

♣

Supposons que $E(\mathbb{F}_{q^k})$ ne contienne pas de point non nul de ℓ^2-torsion. On peut donc lui appliquer la proposition précédente, et ainsi considérer que t_ℓ agit sur des couples de points de ℓ-torsion.

Remarquons maintenant que, contrairement au couplage de Weil, le couplage de Tate n'est pas alterné, on peut donc avoir $t_\ell(P,P) \neq 1$. Cependant, on montre facilement que si $P \in E(\mathbb{F}_q)$, alors $t_\ell(P,P)$ ne peut être distinct de 1 que si $k = 1$ (en effet, $t_\ell(P,P) \in \mu_\ell \cap \mathbb{F}_q^\times$). Cette condition n'est cependant pas suffisante. Cependant, Galbraith [30] a montré que si $P \in E(\mathbb{F}_q)[\ell]$, et que Ψ est un endomorphisme de $E(\mathbb{F}_{q^k})$ tel que $\Psi(P) \notin E(\mathbb{F}_q)$, alors $t_\ell(P, \Psi(P)) \neq 1$. Ce résultat permet donc, sur les courbes supersingulières pour lesquelles une telle distortion existe, de définir un couplage de Tate modifié $\hat{t}_\ell$ tout comme on a défini un couplage de Weil modifié.

1.5 Algorithmes de calcul de couplages

1.5.1 Calcul du couplage de Weil

Soit ℓ un diviseur premier de $\#E(\mathbb{F}_q)$ distinct de p et ne divisant pas $q-1$. Soit k le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ. On rappelle qu'alors $E[\ell] = E(\mathbb{F}_{q^k})[\ell]$, et que k n'est autre que l'ordre de q modulo ℓ. En particulier, $\mathbb{F}_{q^k}^\times$ contient le groupe des racines $\ell^{\text{èmes}}$ de l'unité.

L'algorithme que nous introduisons ici repose sur la définition du couplage de Weil donnée en 1.2.3. Soient P et Q deux points de $E[\ell]$, et soient S et T deux points de $E(\mathbb{F}_{q^k})$ tels que $A = (P+S) - (S)$ et $B = (Q+T) - (T)$ soient à supports disjoints. On a donc :

$$\begin{aligned} e_\ell(P,Q) &= \tfrac{f_A(B)}{f_B(A)} \\ &= \tfrac{f_A(Q+T) f_B(S)}{f_A(T) f_B(P+S)} \end{aligned}$$

où les fonctions f_A et f_B sont telles que $\mathrm{div}(f_A) = \ell A$ et $\mathrm{div}(f_B) = \ell B$.

On notera que si les supports n'étaient pas disjoints, on aurait $f_A(B) = f_B(A) = 0$, et le quotient serait indéfini.

Le problème revient donc, étant donné le diviseur A, à calculer cette fonction f_A (et de même pour B). Pour ce faire, nous allons utiliser (1.1) et (1.2). Ces égalités nous permettent tout d'abord d'écrire :

$$A = (P) - (\infty) + \mathrm{div}\left(\frac{v_{P+S}}{d_{P,S}}\right). \tag{1.3}$$

On en déduit aussi, pour tous points U et V de $E(\overline{\mathbb{F}}_q)$, l'égalité :

$$(U) - (\infty) + (V) - (\infty) = (U+V) - (\infty) + \mathrm{div}\left(\frac{d_{U,V}}{v_{U+V}}\right), \tag{1.4}$$

et cette dernière égalité donne directement :

$$(U) - (\infty) - ((V) - (\infty)) = (U-V) - (\infty) + \mathrm{div}\left(\frac{v_U}{d_{V,U-V}}\right). \tag{1.5}$$

Pour calculer f_A, il suffit donc de calculer ℓA en partant de l'expression de A donnée par 1.3, en utilisant une chaîne d'addition-soustraction quelconque pour ℓ et en appliquant (1.4) ou (1.5) à chaque étape. On obtient de cette façon f_A sous la forme d'une fraction rationnelle. On notera que f_A est ainsi définie à une constante multiplicative près, mais cette constante disparait lors du calcul de $e_\ell(P,Q)$ puisque l'on a un quotient $f_A(Q+T)/f_A(T)$.

Cette méthode, si elle permet effectivement de calculer des couplages, présente néanmoins un inconvénient lorsque ℓ est grand. En effet, le numérateur et le dénominateur de la fraction rationnelle trouvée pour f_A sont de degré $O(\ell)$.

Dans la pratique, un algorithme assez voisin mais ne présentant pas cet inconvénient, dû à Miller [55], est utilisé. Cet algorithme est basé sur une remarque simple : le calcul du couplage ne nécessite pas de connaître f_A sous la forme d'une fraction rationnelle. Il suffit en fait d'évaluer f_A en deux points ($Q+T$ et T). Or dans la méthode de calcul de f_A décrite ci-dessus, on peut remarquer que la fraction rationnelle est obtenue par multiplications successives de fractions rationnelles intermédiaires. En remplaçant ces fractions rationnelles intermédiaires par leur évalutation en $Q+T$ et en T, on obtient donc directement l'évaluation de f_A en $Q+T$ et T. On peut d'ailleurs ainsi calculer directement l'évaluation du quotient $f_A(Q+T)/f_A(T)$, ce qui est préférable. Bien entendu, cette méthode impose que chacune des fractions rationnelles intermédiaires ne doit avoir ni zéro, ni pôle en T ni en $Q+T$, car on aurait alors un résultat indéfini. Le support de tout diviseur rencontré dans la chaîne d'addition-soustraction utilisée pour calculer ℓA ne doit donc contenir ni T, ni $Q+T$.

Cet algorithme est décrit plus en détail dans le chapitre 5. En particulier, sa complexité y est étudiée. On peut toutefois noter que tous les calculs se font *a priori* dans le corps $\mathbb{F}_{q^k}$, et donc la rapidité du calcul du couplage dépend directement du temps pris par l'arithmétique dans $\mathbb{F}_{q^k}$.

Pour des applications cryptographiques, nous le verrons, q est en général de l'ordre de quelques centaines de bits, et le paramètre k ne doit donc pas être trop grand si l'on veut pouvoir calculer des couplages efficacement.

1.5.2 Calcul du couplage de Tate

Soit ℓ un diviseur de $\#E(\mathbb{F}_q)$ distinct de p, et soit k l'ordre de q modulo ℓ. Le corps $\mathbb{F}_{q^k}^\times$ contient donc le groupe des racines $\ell^{\text{èmes}}$ de l'unité, mais l'on n'a pas *a priori* l'égalité $E[\ell] = E(\mathbb{F}_{q^k})[\ell]$ (la condition $\ell \nmid q-1$ implique l'égalité).

Soit $P \in E(\mathbb{F}_{q^k})[\ell]$ et soit $Q \in E(\mathbb{F}_{q^k})$. On a vu que le couplage de Tate entre P et Q est donné par :

$$t_\ell(P,Q) = \left(\frac{f_P(Q+T)}{f_P(T)}\right)^{\frac{q^k-1}{\ell}},$$

où T est un point quelconque de $E(\mathbb{F}_{q^k})$ tel que $Q+T \neq P$ et $T \neq P$ (et $\mathrm{div}(f_P) = \ell\left((P)-(\infty)\right)$).

Le couplage de Tate peut donc se calculer avec les mêmes idées que celles exposées plus haut pour le calcul du couplage de Weil. On notera que le calcul est *a priori* plus simple, puisque modulo l'exponentiation finale dans le corps $\mathbb{F}_{q^k}$, il suffit d'évaluer $f_P(Q+T)/f_P(T)$ alors que le calcul d'un couplage de Weil nécessite environ le double de calculs (évaluation de $f_A(P+S)/f_A(S)$ et de $f_B(Q+T)/f_B(T)$). Le calcul du couplage de Tate sera étudié plus en détail au chapitre 5.

Chapitre 2

Premières applications des couplages en cryptologie

2.1 Attaques sur le logarithme discret

2.1.1 Complexité des attaques contre le logarithme discret

Nous avons vu en introduction que l'avantage des groupes associés aux courbes elliptiques par rapport aux groupes multiplicatifs de corps finis est que les attaques contre le logarithme discret y sont plus dures.

Plus précisément, il n'existe en général pas d'attaques contre le logarithme discret sur des courbes elliptiques plus efficaces que les attaques génériques contre le logarithme discret dans un groupe commutatif quelconque. Or ces attaques (méthodes "ρ" de Pollard, "pas de bébé - pas de géant" de Shanks, algorithme de Pohlig–Hellman) sont toutes exponentielles en la taille plus grand facteur premier du cardinal du groupe considéré : elles permettent de résoudre le problème du logarithme discret dans un groupe commutatif G de cardinal n premier en un temps $O(\sqrt{n})$.

En ce qui concerne le logarithme discret dans des corps finis, par contre, il existe des attaques spécifiques tirant parti de la structure de corps sous-jacente. Ces attaques sont connues sous le nom d'attaques par *calcul d'index*, et ont une complexité sous-exponentielle. En particulier, si p est un nombre premier, le problème du logarithme discret peut être résolu dans un sous-groupe de $\mathbb{F}_p^\times$ en un temps

$$O\left(e^{c(\log p)^{1/3}(\log\log p)^{2/3}}\right),$$

pour une certaine constante c.

En toute généralité, dans un corps fini $\mathbb{F}_q$ (non nécessairement premier), le problème du logarithme discret peut être résolu en un temps

$$O\left(e^{c'(\log q)^{1/2}(\log\log q)^{1/2}}\right),$$

pour une certaine constante c' [1, 60].

Aujourd'hui, le problème du logarithme discret dans un groupe de taille 160 bits associé à une courbe elliptique est généralement considéré comme étant de difficulté équivalente au problème du logarithme discret dans un corps fini de 1024 bits. Le choix de l'utilisation des courbes elliptiques permet donc de réduire la taille des clés pour obtenir une sécurité réputée équivalente.

Nous allons voir dans ce qui suit que le choix d'une courbe elliptique destinée à être utilisée dans un cryptosystème dont la sécurité est basée sur le problème du logarithme discret doit se faire avec circonspection. Il existe en effet certaines classes de courbes elliptiques pour lesquelles on connait des attaques sous-exponentielles contre le logarithme discret (on notera que ces attaques ne se réduisent pas à celles décrites dans ce rapport).

2.1.2 L'attaque de Menezes, Okamoto et Vanstone

Dans [53], Menezes, Okamoto et Vanstone montrent qu'en utilisant le couplage de Weil, le problème du logarithme discret sur une courbe elliptique peut se réduire au problème du logarithme discret dans un corps fini (plus précisément dans une extension du corps de base de la courbe).

Soit ℓ un diviseur premier de $\#E(\mathbb{F}_q)$ distinct de p et ne divisant pas $q-1$. Soit k le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ. Soit P un générateur du groupe $E(\mathbb{F}_q)[\ell]$ et soit $R \in E(\mathbb{F}_q)[\ell]$. On cherche à résoudre le problème de logarithme discret associé, *i.e.* on cherche un entier x tel que $R = [x]P$.

Soit $Q \in E(\mathbb{F}_{q^k})[\ell]$ un point de ℓ-torsion indépendant de P (donc tel que le groupe $\langle P, Q\rangle$ engendré par P et Q soit égal à $E(\mathbb{F}_{q^k})[\ell] = E[\ell]$). $e_\ell(P,Q)$ est donc une racine primitive $\ell^{\text{ème}}$ de l'unité. Comme $e_\ell(R,Q) \in \langle e_\ell(P,Q)\rangle$ on peut déterminer, en résolvant un problème de logarithme discret dans $\langle e_\ell(P,Q)\rangle \subset \mathbb{F}_{q^k}^\times$, un entier y tel que :

$$e_\ell(R,Q) = e_\ell(P,Q)^y$$

La bilinéarité de e_ℓ implique alors que l'on a $R = [y]P$, et donc on a résolu le problème du logarithme discret dans $E(\mathbb{F}_q)[\ell]$.

L'intérêt est évidemment que la résolution du problème du logarithme discret est sous-exponentielle (en $k \log q$) dans $\mathbb{F}_{q^k}^\times$, alors qu'elle est exponentielle (en $\log \ell$) dans $E(\mathbb{F}_q)[\ell]$. Dans le cas où ℓ est de taille 160 bits par exemple, avec les meilleurs algorithmes connus aujourd'hui, il sera plus simple de résoudre le logarithme discret dans le corps fini $\mathbb{F}_{q^k}^\times$ pour $k \leqslant 6$ (car alors la taille du corps fini est inférieure à 1024 bits).

Pour que cette réduction soit efficace, il faut donc que l'entier k soit assez petit, non seulement pour que le problème du logarithme discret soit plus simple dans le corps fini $\mathbb{F}_{q^k}^\times$ que dans le groupe $E(\mathbb{F}_q)[\ell]$, mais aussi pour que le couplage e_ℓ soit efficacement calculable.

Les courbes supersingulières (pour lesquelles $k \leqslant 6$) sont donc toutes sujettes à cette attaque, mais aussi les courbes ordinaires dont le degré MOV est suffisamment petit pour que le couplage de Weil soit efficacement calculable (voir 5.2.2 pour plus de précisions). Il convient donc, avant d'utiliser une courbe elliptique dans un cryptosystème basé sur le logarithme discret, de vérifier que son degré MOV (relativement au cardinal du groupe dans lequel on va travailler) est suffisamment grand.

2.1.3 L'attaque de Frey et Rück

Dans [28], Frey et Rück proposent une attaque tout à fait similaire à l'attaque MOV, mais utilisant le couplage de Tate. Comme expliqué en 5.2.2, étant donnés une courbe $E(\mathbb{F}_q)$ et un diviseur premier ℓ de son cardinal, le couplage de Tate t_ℓ est efficacement calculable dans les cas où l'ordre k de q modulo ℓ est relativement petit. Le seul cas où k ne correspond pas au degré MOV de la courbe est le cas où $\ell \mid q-1$ (et donc $k=1$). Dans ce cas, on peut bien sûr calculer le couplage de Tate t_ℓ, alors que le couplage de Weil e_ℓ n'est pas défini.

Toute courbe vulnérable vis-à-vis de l'attaque MOV est donc aussi vulnérable vis-à-vis de l'attaque de Frey et Rück, mais cette dernière attaque est cependant légèrement plus puissante. Elle permet par exemple d'attaquer des courbes de trace $t = 2$, puisqu'alors leur cardinal vaut $q-1$.

2.2 Séparation de DH et DDH sur certaines courbes

2.2.1 Problèmes DH et DDH

Nous l'avons vu en introduction, un certain nombre de cryptosystèmes ont leur sécurité basée sur la difficulté du problème de Diffie–Hellman (DH) dans une certaine classe de groupes. Il existe un problème assez proche du problème DH sur lequel on peut aussi baser des cryptosystèmes : le *problème décisionnel de Diffie–Hellman* (DDH). Il s'agit, étant donnés quatre éléments (P, Q, R, S) tels que Q, R et S sont dans $\langle P \rangle$, de déterminer si (P, Q, R, S) est un quadruplet de Diffie–Hellman, *i.e.* de déterminer s'il existe deux entiers x et y tels que $Q = P^x$, $R = P^y$ et $S = P^{xy}$.

On notera que si P est un point d'ordre ℓ (*i.e.* un générateur de $E(\mathbb{F}_q)[\ell]$), et qu'un couplage de Weil modifié $\hat{e}_\ell$ est efficacement calculable (il faut pour cela que $E(\mathbb{F}_q)$ soit supersingulière), alors le problème DDH est facile dans $\langle P \rangle$: étant donné un quadruplet (P, Q, R, S), il suffit de tester si $\hat{e}_\ell(P, S) = \hat{e}_\ell(Q, R)$ pour décider si (P, Q, R, S) est un quadruplet de Diffie–Hellman. En effet, comme Q, R et S sont dans $\langle P \rangle$, il existe q, r et s dans $[0, \ell - 1]$ tels que $Q = [q]P$, $R = [r]P$ et $S = [s]P$. On a alors, par bilinéarité du couplage de Weil modifié :

$$\hat{e}_\ell(P, S) = \hat{e}_\ell(P, P)^s,$$

et

$$\hat{e}_\ell(Q, R) = \hat{e}_\ell(P, P)^{qr}.$$

Les deux quantités sont donc égales si et seulement si $s \equiv qr\ [\ell]$, c'est-à-dire si le quadruplet est bien un quadruplet de Diffie–Hellman.

2.2.2 Où DH et DL deviennent équivalents

Dans [50], Maurer et Wolf montrent que dans un groupe cyclique G quelconque, le problème de Diffie–Hellman et le problème du logarithme discret sont équivalents modulo certaines conditions sur le cardinal de G (plus précisément, la réduction proposée par les auteurs entre DL et DH est possible si le cardinal de G admet un unique facteur de grande taille, les autres facteurs étant de taille bornée).

2.2.3 Séparation de DH et DDH

Joux et Nguyen [41] montrent comment construire des courbes elliptiques supersingulières pour lesquelles la réduction de Maurer et Wolf s'applique, et donc pour lesquelles les problèmes DH et DL sont polynomialement équivalents (et *a priori* durs), mais pour lesquelles on peut définir des couplages modifiés (*via* une distortion), ce qui fait que pour ces courbes, le problème DDH est facile. Ils exhibent plusieurs exemples de courbes présentant ces caractéristiques. On notera que de tels groupes sont aussi appelés groupes Gap–Diffie–Hellman [11] en raison de cet écart entre les difficultés des problèmes DH et DDH.

Ce résultat tend donc à montrer que les problèmes DH et DDH ne sont sans doute pas équivalents dans un groupe cyclique quelconque.

2.3 Protocole de Diffie-Hellman tripartite

En 1999, Joux [39] fut l'un des premiers à proposer un cryptosystème utilisant les couplages sur les courbes elliptiques. Il s'agit d'un protocole de création de clé commune tripartite, dont le principe est assez proche du protocole de Diffie-Hellman. L'intérêt pratique de ce protocole est qu'il ne nécessite de la part de chacun des trois participants que la diffusion aux deux autres d'un unique message. La description de ce protocole entre trois participants A, B et C est la suivante :

- **Initialisation :** une courbe elliptique $E(\mathbb{F}_q)$, dont le cardinal est divisible par un (grand) entier premier ℓ est choisie. Le degré MOV k de $E(\mathbb{F}_q)$ relativement à ℓ doit être assez petit pour que le couplage de Weil associé puisse être calculé efficacement (voir 5.2.2 pour plus de précisions). La courbe $E(\mathbb{F}_q)$ est rendue publique, ainsi que deux points $P \in E(\mathbb{F}_q)[\ell]$ et $Q \in E(\mathbb{F}_{q^k})[\ell]$ engendrant $E[\ell]$.
- **Création de clé commune entre A, B et C :** A choisit un entier aléatoire $a \in [0, \ell - 1]$, puis calcule aP et aQ et les envoie à B et C. Ces derniers font de même avec des entiers b et c. Chacun peut alors calculer la quantité $e_\ell(P,Q)^{abc}$: A en calculant $e_\ell(bP, cQ)^a$, B en calculant $e_\ell(cP, aQ)^b$ et C en calculant $e_\ell(aP, bQ)^c$ (l'égalité de ces quantités découle directement de la bilinéarité du couplage).

La sécurité de ce protocole dépend directement de la difficulté du problème de Weil–Diffie–Hellman, un cas particulier du problème GBDH que nous définirons en 3.3.

On notera que l'on peut utiliser, à la place du couplage de Weil, le couplage de Tate ou encore un couplage de Weil ou de Tate modifié (comme définis en 1.4). L'utilisation de couplages modifiés permet d'ailleurs de réduire la bande passante nécessaire à l'exécution de ce protocole.

Chapitre 3

Cryptosystèmes basés sur l'identité utilisant les couplages

3.1 Principe

Le concept de cryptosystème basé sur l'identité a été introduit par Shamir en 1984 [62]. Il s'agit, dans un cryptosystème à clé publique, de pouvoir remplacer l'espace des clés publiques par l'espace des chaînes quelconques de caractères (tout en ayant un moyen de calculer la clé privée correspondant à toute clé publique possible). On peut alors concevoir des systèmes à clé publique où la clé publique de chaque utilisateur est un élément directement lié à son identité (par exemple son adresse mail ou son adresse IP).

Toutefois, une limitation fondamentale des systèmes basés sur l'identité apparaît très vite. En effet, il est impossible que chaque utilisateur puisse lui-même calculer la clé privée correspondant à son identité (sinon, il pourrait de la même façon calculer la clé privée de tout autre utilisateur à partir de l'identité de celui-ci). Il y a donc dans un tel système nécessité d'une autorité de confiance, le *générateur de clés privées* (PKG en anglais, terme que nous utiliserons par la suite), dont le rôle est de répondre aux requêtes des utilisateurs en calculant la clé privée correspondant à leur identité et en la leur transmettant (il va sans dire que cette transmission doit être protégée, mais nous n'étudierons pas cette étape dans ce qui suit).

En général, comme nous le verrons, le PKG dispose d'une donnée secrète dite *clé maître* qui lui permet de calculer la clé privée correspondant à toute chaine de caractères.

On notera que dans [62], Shamir propose uniquement un protocole de signature basé sur l'identité. Il faudra attendre une quinzaine d'années pour voir apparaître des protocoles de chiffrement basés sur l'identité. La première solution proposée, due à Cocks [16], repose sur des propriétés de résiduosité quadratique. Son inconvénient majeur est que la taille du chiffré est beaucoup plus grande que celle du clair. Nous allons voir ici que les couplages de Weil et de Tate apportent une réponse satisfaisante au problème, comme cela a été découvert indépendemment par Sakai, Ohgishi et Kasahara [59] (leurs travaux étant longtemps restés confidentiels) et par Boneh et Franklin [10], dont l'intérêt pour l'utilisation des couplages dans les cryptosystèmes semble avoir été provoqué par le protocole tripartite de création de clé de Joux (voir 2.3).

3.2 Applications

Le premier intérêt d'un cryptosystème basé sur l'identité est que, contrairement à un système à clé publique classique, il ne nécessite plus la certification des clés publiques par une autorité de confiance, puisque la clé publique correspond justement à l'identité de son propriétaire. Nous verrons même en 3.7 que, comme il est possible d'organiser une infrastructure de certification de clés de façon hiérarchique, il est aussi possible de hiérarchiser un cryptosystème basé sur l'identité (au niveau de la génération des clés privées).

Les systèmes basés sur l'identité permettent aussi de gérer facilement le problème de la révocation des clés. On peut par exemple imaginer un système où les clés sont renouvelées tous les mois. Dans un tel système, la clé publique d'Alice serait de la forme :

`email d'Alice || mois || année`.

Ainsi, chaque clé ne serait valide que pour un mois, et tous les mois Alice devrait demander une nouvelle clé privée au PKG (qui déciderait ou non de la lui délivrer).

Une autre application intéressante, décrite par Boneh et Franklin [10], est la délégation de clé privée. Supposons que Bob veuille pouvoir déchiffrer ses mails sur son portable, préservant l'intégrité de sa clé privée en cas de vol de son portable. En utilisant un système à clé publique classique, le problème semble insoluble, car Bob doit mettre sa clé privée sur son portable pour déchiffrer ses mails. Si maintenant Bob dispose d'un algorithme de chiffrement basé sur l'identité, il peut prendre la place du PKG et choisir une clé maître qu'il gardera secrète. De cette façon, les correspondants de Bob peuvent, pour lui envoyer des mails, utiliser une clé correspondant à la date d'envoi. Bob peut ainsi, à l'aide de la clé maître qu'il possède, calculer des clé(s) de déchiffrement correspondant à un certain nombre de jours, et les transférer sur son portable. De cette façon, le vol de son portable ne compromettra que les mails reçus durant une certaine période. On peut bien sûr imaginer un système assez similaire où Bob ne délègue pas ses clés à son portable, mais à un tiers (par exemple une personne qui doit le remplacer durant un certain laps de temps).

On peut enfin noter que l'on peut assez facilement mettre en œuvre des techniques de partage de secret (par exemple des schémas à seuil, comme le protocole de Shamir [61]) pour répartir une clé maître entre plusieurs PKG, qui n'ont alors plus individuellement la possibilité de calculer les clés privées des utilisateurs (chaque PKG ne pouvant calculer qu'un fragment de clé).

3.3 Couplages et problème GBDH

Tous les protocoles présentés dans cette partie utilisent un couplage

$$e : G \times \hat{G} \to V,$$

où $(G, +)$, $(\hat{G}, +)$ et $(V, \times)$ sont trois groupes d'ordre premier ℓ. Les éléments non neutres de ces groupes seront notés G^* (resp. $\hat{G}^*$ et V^*).

On supposera que e est bilinéaire, non dégénéré et efficacement calculable.

3.3.1 Problème GBDH

La sécurité des protocoles que nous allons décrire est directement liée à la difficulté du problème dit *problème bilinéaire généralisé de Diffie–Hellman* (GBDH). Ce problème se pose comme suit : étant donné un tuple (P, aP, cP, Q, bQ, cQ) avec $P \in G$, $Q \in \hat{G}$ et $a, b, c \in \mathbb{Z}$, calculer $e(P, Q)^{abc}$.

En 3.6.4, nous montrons un exemple de réduction du cassage d'un protocole basé sur l'identité utilisant les couplages au problème GBDH.

On notera que la difficulté du problème GBDH implique la difficulté du problème DL dans les groupes G, $\hat{G}$ et V (la réciproque n'est pas nécessairement vraie).

Dans la pratique, les seuls couplages que l'on connaisse aujourd'hui qui remplissent les conditions sus-citées sont les couplages de Weil et de Tate sur des courbes algébriques. Nous limitons ici notre présentation au cas des courbes elliptiques.

3.3.2 Couplages de Weil et de Tate

Soit ℓ un diviseur premier de $\#E(\mathbb{F}_q)$ distinct de p et ne divisant pas $q-1$, et soit k le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ. On supposera k suffisamment petit pour que les couplages de Tate et de Weil associés à ℓ soient efficacement calculables (voir 5.2.2 pour plus de détails).

La première possibilité d'utilisation du couplage de Weil est, avec les notations introduites plus haut, de choisir deux générateurs P et Q de $E[\ell]$ (P étant un générateur de $E(\mathbb{F}_q)[\ell]$). On pose alors $G = \langle P \rangle$, $\hat{G} = \langle Q \rangle$, $V = \mathbb{F}_{q^k}^*$ et bien sûr $e = e_\ell$.

Si $E(\mathbb{F}_{q^k})$ ne contient pas de points non nuls de ℓ^2-torsion, on a vu en 1.4 que l'on pouvait de la même façon utiliser le couplage de Tate t_ℓ.

De plus, lorsque l'on peut définir des couplages de Weil ou de Tate modifiés, on peut poser $G = \hat{G} = \langle P \rangle = E(\mathbb{F}_q)[\ell]$ (et $e = \hat{e}_\ell$ ou $e = \hat{t}_\ell$).

L'intérêt de l'utilisation de couplages modifiés dans les protocoles est qu'on pourra alors se restreindre à ne manipuler que des points de $E(\mathbb{F}_q)$, qui peuvent être représentés en utilisant environ $\log q$ bits (en utilisant la technique dite de compression de point présentée par exemple dans [8]), alors que les points de $E(\mathbb{F}_{q^k})$ nécessitent $k \log q$ bits. Suivant les protocoles concernés, on obtiendra donc une réduction de la taille des clés, de la taille des textes chiffrés,...

Sécurité du problème GBDH associé

Le problème GBDH associé au couplage de Weil (resp. de Tate) est aussi parfois appelé problème de Weil–Diffie–Hellman (WDH) (resp. problème de Tate–Diffie–Hellman ou TDH) [10]. Il est intéressant de noter que les seules attaques connues à ce jour contre ces problèmes sont les attaques contre le logarithme discret dans l'un des trois groupes G, $\hat{G}$ ou V.

La sécurité des protocoles que nous allons présenter est donc directement reliée à la difficulté du logarithme discret dans le groupe parmi G, $\hat{G}$ et V *le plus vulnérable* aux attaques contre ce même logarithme discret.

Avec les paramètres de sécurité actuels, il faudra donc que ℓ soit de l'ordre de 160 à 200 bits, et que q^k vaille au moins 1024 bits. Étant donné que la difficulté du logarithme discret dans $E(\mathbb{F}_q)[\ell]$ dépend de ℓ et non de q, il est donc préférable que les tailles des entiers ℓ et q soient voisines.

De plus, nous avons déjà vu que, ℓ étant d'assez grande taille, k doit être relativement petit pour que e_ℓ soit effectivement calculable. Et c'est cette condition qui est problématique, car si l'on fixe la taille de ℓ, il est *a priori* difficile de construire une courbe $E(\mathbb{F}_q)$ ayant un faible degré MOV. Nous avons toutefois vu une exception notable en 1.5 : si $E(\mathbb{F}_q)$ est supersingulière, alors k vaut au plus 6. Nous verrons dans le chapitre 4 comment obtenir des courbes elliptiques ordinaires utilisables dans les protocoles basés sur les couplages.

3.4 Chiffrement basé sur l'identité

Nous présentons ici le protocole introduit par Boneh et Franklin dans [10] (en généralisant légèrement la présentation). Ce protocole chiffre des données par blocs de n bits, et se décompose naturellement en quatre algorithmes :

- **Génération des paramètres :** le PKG génère un quadruplet $(G, \hat{G}, V, e)$ comme précisé en 3.3 (en particulier ces trois groupes sont tous d'ordre ℓ premier). Le problème GBDH associé doit être dur. Le PKG génère aussi deux fonctions de hachage $\hat{H} : \{0,1\}^* \to \hat{G}$ et $h : V \to \{0,1\}^n$. Enfin il choisit un générateur P de G, et un entier aléatoire $s \in [1, \ell-1]$. Il calcule $P_{\text{pub}} = sP$, et publie tous ces paramètres, excepté s, qui est la clé maître.
- **Extraction de clé privée :** lorsque l'utilisateur dont l'identité est ID lui demande sa clé privée, le PKG calcule la clé privée correspondante $d_{ID} = s\hat{H}(ID)$ et la lui transmet.
- **Chiffrement :** pour chiffrer le message M (de n bits) avec la clé ID, on choisit un entier $r \in [1, \ell-1]$ aléatoire, et on envoie le couple :
$$C = (rP, M \oplus h(v_{ID,r}))$$
où
$$v_{ID,r} = e(P_{\text{pub}}, r\hat{H}(ID)).$$
- **Déchiffrement :** pour déchiffrer le couple $C = (A, B)$ (chiffré avec la clé publique ID), le possesseur de la clé privée d_{ID} calcule :
$$M = B \oplus h(e(A, d_{ID})).$$

Montrons la consistence de ce protocole. Durant la phase de chiffrement un message M est xoré avec $h(v_{ID,r})$ pour obtenir B, et au déchiffrement on xore B avec $h(e(A, d_{ID}))$. Il faut donc montrer que $v_{ID,r} = e(A, d_{ID})$. On a :

$$\begin{aligned} v_{ID,r} &= e(P_{\text{pub}}, r\hat{H}(ID)) \\ &= e(sP, r\hat{H}(ID)) \\ &= e(rP, s\hat{H}(ID)) \text{ (par bilinéarité de } e) \\ &= e(A, d_{ID}), \end{aligned}$$

ce qui montre que le chiffré (A, B) se déchiffre bien en M.

Boneh et Franklin définissent différentes notions de sécurité pour les systèmes de chiffrement basés sur l'identité. Ils prouvent que le protocole ci-dessus réalise bien une fonction de chiffrement à sens unique, mais n'est pas sémantiquement sûr. On peut le transformer pour lui donner cette propriété en utilisant le schéma de Fujisaki et Okamoto [29]. Le protocole modifié est alors décrit par les quatre algorithmes suivants :

- **Génération des paramètres :** l'algorithme est le même que précédemment, mais le PKG génère de plus deux fonctions de hachage $h' : \{0,1\}^n \times \{0,1\}^n \to [1, \ell]$ et $h'' : \{0,1\}^n \to \{0,1\}^n$ qui sont publiques.
- **Extraction de clé privée :** même algorithme que précédemment.
- **Chiffrement :** pour chiffrer le message M (de n bits) avec la clé publique ID, on choisit une chaîne de caractères aléatoire $\sigma \in \{0,1\}^n$, on pose
$$r = h'(\sigma, M),$$
et le chiffré est alors :
$$(rP, \sigma \oplus h(v_{ID,r}), M \oplus h''(\sigma)),$$

avec

$$v_{ID,r} = e(P_{\text{pub}}, r\hat{H}(ID)).$$

- **Déchiffrement :** pour déchiffrer le triplet (A, B, C) (chiffré avec la clé publique ID), le possesseur de la clé privée d_{ID} calcule :

$$B \oplus h(e(A, d_{ID})) = \sigma,$$

$$C \oplus h''(\sigma) = M.$$

Il pose alors $r = h'(\sigma, M)$, et vérifie que $A = rP$. Si c'est le cas, le message déchiffré est M, sinon le triplet (A, B, C) est considéré comme non valide.

La consistence de ce second protocole est encore une conséquence directe de la bilinéarité de e.

3.4.1 Variante avec authentification

On peut réaliser à peu de frais une variante du système de chiffrement que nous avons décrit ci-dessus permettant l'authentification de l'expéditeur. Nous donnons encore une description des quatre algorithmes composant ce protocole :

- **Génération des paramètres :** les paramètres sont les mêmes que pour le second protocole de Boneh et Franklin décrit ci-dessus, avec de plus une fonction de hachage $H : \{0,1\}^* \to G$.
- **Extraction de clé privée :** on distingue dans ce protocole deux types de clés privées : les clés privées de déchiffrement, qui s'obtiennent de la même façon que dans le protocole précédent, et les clés privées d'authentification. Lorsque l'utilisateur dont l'identité correspond à ID demande sa clé privée d'authentification, le PKG calcule $a_{ID} = sH(ID)$ et le transmet à l'utilisateur.
- **Chiffrement :** supposons que l'utilisateur d'identité ID_A souhaite chiffrer un message M (de n bits) à destination de l'utilisateur dont l'identité est ID. S'il ne souhaite pas s'identifier, il procède comme dans le protocole précédent. Sinon, il lui suffit de remplacer P par $H(ID_A)$ et P_{pub} par a_{ID_A}. Il doit bien sûr ajouter son identité en clair au message chiffré.
- **Déchiffrement :** pour déchiffrer un message envoyé par l'utilisateur dont l'identité est ID_A à celui dont l'identité est ID, en utilisant d_{ID}, on procède comme précédemment en remplaçant P par $H(ID_A)$ si le message a été chiffré avec authentification.

Nous avons trouvé cette variante avec Andreas Enge en avril 2002 mais n'avons pas jugé utile de la publier. Indépendemment, Lynn [48] puis Malone-Lee [49] ont proposé le même protocole quelques semaines plus tard. Nous renvoyons à leurs articles pour la preuve de la propriété d'authentification. La sécurité sémantique du chiffrement se déduit directement de celle du protocole précédent.

3.4.2 Signatures courtes

Moni Naor a montré que tout système de chiffrement basé sur l'identité pouvait être transformé en un protocole de signature à clé publique. La transformation est la suivante : la clé privée d'un signataire correspond à une clé maître dans le protocole de chiffrement. La signature d'un message M correspond à la clé privée pour $ID = M$ dans le protocole de chiffrement. Ainsi, pour vérifier que la signature d'un message M est valide, il suffit de générer un message aléatoire M', de le chiffrer avec M comme clé (à l'aide du protocole de chiffrement basé sur l'identité), puis de le déchiffrer en utilisant la signature comme clé de déchiffrement. La signature est acceptée comme valide si l'on retombe bien sur M'.

Dans le cas particulier du protocole de Boneh et Franklin, on peut simplifier le protocole de signature proposé ci-dessus. C'est le travail qu'ont réalisé Boneh, Lynn et Shacham [11], et que nous décrivons ici brièvement. Ce protocole se décompose en trois algorithmes :

- **Génération de clés :** le signataire génère un quadruplet $(G, \hat{G}, V, e)$ comme précisé en 3.3 (en particulier ces trois groupes sont tous d'ordre ℓ premier). Le problème GBDH associé doit être dur. Il génère aussi une fonction de hachage $\hat{H} : \{0,1\}^* \to \hat{G}$, et choisit enfin un générateur P de G, et un entier aléatoire $s \in [1, \ell - 1]$. Il calcule $P_{\text{pub}} = sP$, et publie tous ces paramètres, excepté s qui est sa clé privée.
- **Signature :** la signature du message $M \in \{0,1\}^*$ avec la clé privée s est le point $\sigma = s\hat{H}(M) \in \hat{G}$.
- **Vérification :** étant donnés une clé publique P_{pub}, un message M et une signature σ, on vérifie que la signature a bien été produite par le possesseur de la clé privée correspondant à P_{pub} en s'assurant que l'on a bien l'égalité

$$e(P, \sigma) = e(P_{\text{pub}}, \hat{H}(M)).$$

Boneh, Lynn et Shacham montrent que la résistance à la contrefaçon existentielle et aux attaques à messages clairs choisis de ce protocole se réduisent au problème GBDH sous-jacent. Nous renvoyons à leur article pour cette démonstration.

L'aspect intéressant de ce protocole est que s'il est implanté en utilisant un couplage sur une courbe elliptique $E(\mathbb{F}_q)$, alors la taille des signatures produites peut être ramenée à la taille de q (en utilisant la technique de compression de point). Si l'on impose un paramètre ℓ de taille proche de celle de q, par exemple 160 bits, on obtient des signatures elles aussi sur 160 bits, ce qui rend ce protocole particulièrement adapté aux besoins de signatures courtes (les signatures produites par les protocoles classiques utilisant les courbes elliptiques produisent des signatures dont la taille est double).

3.5 Signature basée sur l'identité

Nous présentons l'un des deux protocoles de signature proposés par Hess [36], qui en donne une preuve de sécurité. Le second protocole de signature proposé par Hess (dont il n'avait pas prouvé la sécurité) a été cassé de façon élémentaire par Cheon [15]. On notera qu'il existe d'autres protocoles de signature basés sur l'identité utilisant les couplages [14, 58, 59], mais ces derniers sont présentés sans preuve de sécurité.

Le protocole de Hess se décompose encore en quatre algorithmes :

- **Génération des paramètres :** le PKG génère un quadruplet $(G, \hat{G}, V, e)$ comme précisé en 3.3 (en particulier ces trois groupes sont tous d'ordre ℓ premier). Le problème GBDH associé doit être dûr. Le PKG génère aussi deux fonctions de hachage $\hat{H} : \{0,1\}^* \to \hat{G}$ et $h : \{0,1\}^* \times V \to [1, \ell - 1]$. Enfin il choisit un générateur P de G, et un entier aléatoire $s \in [1, \ell - 1]$. Il calcule $P_{\text{pub}} = sP$, et publie tous ces paramètres, excepté s, qui est la clé maître.
- **Extraction de clé privée :** lorsque l'utilisateur dont l'identité est ID lui demande sa clé privée, le PKG calcule la clé privée correspondante $d_{ID} = s\hat{H}(ID)$ et la lui transmet.
- **Signature :** pour signer le message M, le possesseur de la clé privée d_{ID} :
  - choisit un élément $Q_1 \in G^*$ aléatoire, ainsi qu'un entier $k \in [1, l - 1]$ lui aussi aléatoire,
  - calcule $r = e(P, Q_1)^k$,
  - calcule $v = h(M, r)$,

– calcule $u = vd_{ID} + kQ_1$.
La signature est alors le couple $(u, v) \in \hat{G} \times [1, l-1]$.
– **Vérification :** pour vérifier la signature (u, v) associée au message M (émise par l'usager dont l'identité est ID), on commence par calculer :

$$r = e(P, u)e(P_{\text{pub}}, -H(ID))^v,$$

puis il accepte la signature si et seulement si $v = h(M, r)$.

Là encore, la consistance de ce protocole est une conséquence directe de la bilinéarité du couplage e. Hess montre que la sécurité de ce protocole se réduit au problème GBDH. Nous renvoyons à son article pour la démonstration.

On notera que lorsque les groupes G et $\hat{G}$ sont égaux, la vérification n'est autre qu'une instance du problme DDH associé.

3.6 Distribution de clé non interactive

3.6.1 Définition, intérêts

Un protocole de distribution de clé non interactif est un moyen de créer une clé secrète commune entre deux protagonistes (traditionnellement prénommés Alice et Bob). Cependant, au contraire du protocole de Diffie–Hellman, ce type de protocole ne nécessite pas de communication entre Alice et Bob.

Une façon naturelle de construire un tel protocole est de le baser sur l'identité, et c'est le cas pour le protocole que nous proposons, qui nécessite donc l'intervention d'un PKG.

Le fonctionnement est le suivant : lorsqu'elle désire créer une clé secrète commune avec Bob, Alice commence par demander au PKG la clé privée correspondant à son identité. Il lui suffit alors de cette clé privée et de l'identité de Bob pour créer une clé secrète commune avec ce dernier (il peut calculer la même clé en utilisant sa propre clé privée et l'identité d'Alice). Bien sûr, aucun autre usager du système ne doit pouvoir calculer la clé secrète que partagent Alice et Bob.

L'intérêt principal d'un tel protocole est que si le système compte n utilisateurs, il suffit que chacun demande sa clé privée au PKG (soit n communications) pour pouvoir ensuite créer une clé secrète avec tout autre utilisateur du système, alors que pour atteindre le même objectif, l'utilisation d'un protocole comme celui de Diffie–Hellman (où plutôt une version authentifiée comme MQV [45]) requérerait de l'ordre de n^2 communications.

L'autre grand avantage d'un tel protocole par rapport à un protocole "classique" est que si Alice veut envoyer un mail chiffré à Bob, elle peut le faire sans contacter Bob au préalable. Si Bob reçoit un mail chiffré de la part d'Alice et qu'il ne possède pas encore de clé privée, il est toujours temps pour lui de demander cette dernière au PKG pour déchiffrer ensuite à l'aide de celle-ci le mail d'Alice.

3.6.2 Protocoles existants

En 1992, Maurer et Yacobi ont proposé le premier protocole non interactif de distribution de clés [51]. Leur système est basé sur le problème du logarithme discret dans $(\mathbb{Z}/n\mathbb{Z})^\times$ avec n composé. Toutefois, Lee et Lim [46] ont rapidement montré que ce système est vulnérable. De même, le système modifié proposé dans [52] peut être cassé par deux participants en collusion qui, avec une grande probabilité, peuvent recouvrer la clé maître du PKG, c'est-à-dire la factorisation de n [44]. De plus, comme la complexité du calcul d'une clé privée par le PKG et la complexité du cassage du système sont assez proches (toutes deux sont sous-exponentielles), un système de ce type soumet le PKG à de lourds calculs et ne permet qu'une faible marge de sécurité dans le choix des paramètres.

Hünlein, Jacobson et Weber ont plus récemment proposé un autre protocole [38], qui est basé sur les ordres non maximaux dans des corps quadratiques imaginaires. Là encore, la complexité de la génération des clés privées par le PKG est sous-exponentielle, ce qui suppose que le PKG dispose d'une capacité de calcul énorme. De plus, la génération de clés privées et le cassage du système ont la même complexité (elles sont toutes deux en $L(1/2)$), la marge de sécurité est donc extrèmement étroite.

Enfin, dans son mémoire [43], Kügler introduit un système de distribution de clés non interactif qui est aussi basé sur le problème du logarithme discret dans $(\mathbb{Z}/n\mathbb{Z})^\times$ avec n composé, mais dans lequel le PKG peut calculer les clés privées en temps polynomial. Cependant, il ne prouve pas la sécurité de son protocole.

3.6.3 Un protocole basé sur les couplages

Nous pensions que le protocole que nous présentons ici était original, avant de réaliser qu'un tel système est décrit dans [59], article rédigé en japonais et présenté à une conférence dont les proceedings ne sont pas publiés ! Nous proposons ici une preuve de la sécurité de ce protocole, qui n'apparaît pas dans [59]. Ce travail a été réalisé en collaboration avec Andreas Enge [20].

Ce protocole se découpe naturellement en trois algorithmes :

- **Génération de paramètres :** le PKG génère un quadruplet $(G, \hat{G}, V, e)$ comme précisé en 3.3 (en particulier ces trois groupes sont tous d'ordre ℓ premier). Le problème GBDH associé doit être dûr. Le PKG génère aussi deux fonctions de hachage $H : \{0,1\}^* \to G$ et $\hat{H} : \{0,1\}^* \to \hat{G}$. Enfin il choisit deux éléments P et Q, générateurs respectifs de G et $\hat{G}$, ainsi qu'un entier aléatoire $s \in [1, \ell - 1]$. Il calcule $P_{\text{pub}} = sP$, et publie tous ces paramètres, excepté s, qui est la clé maître.
- **Extraction de clé privée :** lorsque l'utilisateur A dont l'identité est ID_A lui demande sa clé privée, le PKG calcule la clé privée correspondante $(S_A, \hat{S}_A) = (sH(ID_A), s\hat{H}(ID_A))$ et la lui transmet.
- **Création d'une clé secrète commune :** lorsque A souhaite calculer une clé secrète partagée avec B, il commence par calculer $(P_B, Q_B) = (H(ID_B), \hat{H}(ID_B))$, puis il calcule :
$$(e(S_A, Q_B), e(P_B, \hat{S}_A)).$$
B peut calculer le même couple en commençant par calculer $(P_A, Q_A) = (H(ID_A), \hat{H}(ID_A))$, puis
$$(e(P_S, \hat{S}_B), e(S_B, Q_A)).$$
Ces deux couples sont égaux de par la bilinéarité de e. Il suffit ensuite à A et B d'utiliser une fonction de hachage (publique) pour obtenir à partir de ce couple une clé secrète commune.

3.6.4 Sécurité

Modèle d'attaque

Dans le protocole décrit ci-dessus, le seul trafic susceptible d'être intercepté est la distribution de clés privées aux utilisateurs par le PKG. Il semble donc licite de considérer que ce protocole est sûr si un attaquant possédant un nombre arbitraire de clés privées d'utilisateurs ne peut calculer le secret partagé par deux autres utilisateurs dont il ne connait pas les clés privées. En particulier, une collusion d'utilisateurs partageant leurs clés ne doit pas ainsi avoir d'avantage dans l'attaque contre le secret d'un couple d'autres utilisateurs.

Plus précisément, on dira qu'un attaquant $\mathcal{A}$ (t,ϵ)-*casse* le protocole s'il tourne en temps au plus t et qu'il a un avantage supérieur ou égal à ϵ dans le jeu suivant.

- **Génération de paramètres :** un challenger publie des paramètres généraux $(G, \hat{G}, V, \ell, e, H, \hat{H})$ et choisit une clé maître $s \in [1, \ell - 1]$ qu'il garde secrète.
- **Requêtes d'extraction :** $\mathcal{A}$ envoie un certain nombre de requêtes d'extraction de clés $ID_1, ID_2, \ldots, ID_n$ au challenger qui, à la reception de la requête ID_i, calcule le couple $(sH(ID_i), s\hat{H}(ID_i))$ et l'envoie à $\mathcal{A}$.
- **Essai :** lorsque $\mathcal{A}$ estime qu'il a accumulé assez d'information, il choisit deux identités ID_A et ID_B (différentes de tous les ID_i), et exhibe un quadruplet $(ID_A, ID_B, \alpha, \beta)$.

L'avantage de l'attaquant $\mathcal{A}$ dans ce jeu est défini par :

$$\mathrm{Avant}(\mathcal{A}) = p_{\mathcal{A},1} + p_{\mathcal{A},2},$$

avec

$$p_{\mathcal{A},1} = \mathrm{Prob}\left(e(H(ID_A), \hat{H}(ID_B))^s = \alpha\right)$$

et

$$p_{\mathcal{A},2} = \mathrm{Prob}\left(e(H(ID_B), \hat{H}(ID_A))^s = \beta\right).$$

Preuve de sécurité

Nous montrons ici que le problème GBDH et la sécurité du protocole de distribution de clé non interactive décrit en 3.6.3 sont polynomialement équivalents.

On dit qu'un algorithme probabiliste $\mathcal{A}$ (t,ϵ)-*résout* le problème GBDH associé aux paramètres $(G, \hat{G}, V, \ell, e)$ si $\mathcal{A}$ tourne en temps au plus t et résout correctement le problème GBDH avec probabilié au moins ϵ, c'est-à-dire si

$$\mathrm{Prob}\left(\mathcal{A}(P, aP, cP, Q, bQ, cQ) = e(P,Q)^{abc}\right) \geqslant \epsilon,$$

la probabilité étant prise sur les distributions uniformes et indépendantes de $P \in G$, $Q \in \hat{G}$, $a, b, c \in [1, \ell - 1]$ et sur les choix aléatoires de $\mathcal{A}$.

Proposition 3.1 *Si le problème GBDH associé à des paramètres $(G, \hat{G}, V, \ell, e)$ peut être (t,ϵ)-résolu, alors le protocole de distribution de clé utilisant les paramètres $(G, \hat{G}, V, \ell, e, H, \hat{H})$ peut être $(t+\delta,\epsilon)$-cassé, où δ désigne le temps nécessaire pour compléter deux requêtes d'extraction et pour évaluer H et $\hat{H}$ chacune en un point.*

Démonstration : Un attaquant du protocole peut extraire deux paires (P, sP) et (Q, sQ) pour $P \in G$ et $Q \in \hat{G}$ aléatoires. Il lui suffit donc de choisir aléatoirement deux identités ID_A et ID_B et de calculer $R = H(ID_A) = aP$ et $S = \hat{H}(ID_B) = bQ$ où a et b lui sont inconnus (mais sont aléatoires). La solution à l'instance (P, R, sP, Q, S, sQ) du problème GBDH fournit à l'attaquant le secret partagé entre A et B.

♣

Théorème 3.1 *Supposons les fonctions de hachage H et $\hat{H}$ données par des oracles aléatoires. Supposons qu'il existe un adversaire $\mathcal{A}$ qui (t, ϵ)-casse le protocole de distribution de clé avec les paramètres $(G, \hat{G}, V, \ell, e, H, \hat{H})$. Supposons enfin que l'on connaisse une borne supérieure q_E au nombre de requêtes d'extraction émises par $\mathcal{A}$. Alors il existe un algorithme $\mathcal{B}$ qui $(t', \epsilon/(2e^2(1+q_E)^2))$-casse le problème GBDH associé aux paramètres $(G, \hat{G}, V, \ell, e)$, où e est le nombre d'Euler,*

$$t' = Kt(t_1 + t_2 + \log q_E) + t_3,$$

K est une petite constante et

t_1	*est le temps nécessaire à une multiplication scalaire dans G ou $\hat{G}$, ou à une exponentiation dans V,*
t_2	*est le temps nécessaire à la génération d'un bit aléatoire,*
$\log q_E$	*est le temps nécessaire pour rechercher une entrée dans une liste triée ayant au plus q_E éléments,*
t_3	*est le temps requis par l'algorithme d'Euclide étendu pour des arguments appartenant à l'intervalle* $[1, \ell - 1]$.

On peut remarquer qu'en général, t' sera t fois un polynôme en $\log \ell$, et $\log \ell \leqslant t$ puisque $\mathcal{A}$ renvoie un élément de V, donc en fait t' sera polynomial en t.

Supposer que l'on connait une borne supérieure explicite q_E du nombre de requêtes d'extraction émises par $\mathcal{A}$ (ou *a forteriori* une borne explicite du temps de calcul $t \geqslant q_E$ de $\mathcal{A}$) montre bien les limites de ce théorème. Ce type de supposition est cependant couramment fait dans la littérature [10, 11].

Démonstration : L'algorithme prend en entrée une instance aléatoire et uniformément distribuée $(P, P_a, P_c, Q, Q_b, Q_c) = (P, aP, cP, Q, bQ, cQ)$ du problème GBDH. Pour en trouver la solution $e(P, Q)^{abc}$ avec l'assistance de $\mathcal{A}$, $\mathcal{B}$ va contrôler les fonctions de hachage H et $\hat{H}$. Lorsque par exemple $\mathcal{A}$ lui demandera une évaluation de H, $\mathcal{B}$ génèrera un élément aléatoire du groupe G, obtenu en multipliant P ou P_a par un entier aléatoire. Ainsi, $\mathcal{B}$ se conforme au modèle de l'oracle aléatoire (pour $\mathcal{A}$, la fonction de hachage paraît bien aléatoire), tout en conservant une information additionnelle (les logarithmes discrets dans la base P ou P_a des valeurs de H). Évidemment, comme $\mathcal{B}$ ne connait pas a, il ne peut contrôler que les logarithmes que dans l'une des deux bases. Pour pouvoir répondre aux requêtes d'extraction, $\mathcal{B}$ doit pouvoir rattacher aux différents multiples de P et P_a les identités correspondantes. Pour avoir accès à la solution du problème GBDH, il doit rattacher à un multiple de P_a l'identité apparaissant dans l'essai de $\mathcal{A}$. Ceci représente un certain dilemme pour $\mathcal{B}$, puisque $\mathcal{A}$ peut lui demander d'évaluer le haché d'une identité *avant* d'émettre une requête d'extraction pour cette même identité ou de publier son essai contenant cette même identité. Pour résoudre ce problème, $\mathcal{B}$ choisit au hasard des multiples de P ou de P_a, et s'interrompt dès qu'il s'aperçoit que son choix n'était pas le bon. Les probabilités pour $\mathcal{B}$ de choisir P ou P_a doivent dépendre de q_E, car sinon la probabilité de succès de $\mathcal{B}$ deviendrait exponentiellement petite quand q_E tend vers l'infini. La fréquence avec laquelle $\mathcal{B}$ doit renvoyer des multiples de P augmente avec le nombre de requêtes d'extraction émises par $\mathcal{A}$. C'est la raison pour laquelle $\mathcal{B}$ doit connaître une borne supérieure q_E du nombre de ces requêtes, et de plus sa probabilité de succès décroit quand le nombre de requêtes d'extraction de $\mathcal{A}$ augmente.

Décrivons maintenant de façon plus formelle les différentes routines que $\mathcal{B}$ doit mettre en œuvre :

Requêtes d'évaluation de H : $\mathcal{B}$ conserve une liste (initialement vide) L de quadruplets $(X, R, h, u) \in \{0,1\}^* \times G \times [1, \ell-1] \times \{0,1\}$, triée selon X. Lorsque $\mathcal{A}$ envoie une requête d'évaluation de H en une chaîne de caractères X, $\mathcal{B}$ commence par tester si L contient un quadruplet de la forme (X, R, h, u). Si ce n'est pas le cas, $\mathcal{B}$

- choisit un élément aléatoire $h \in [1, \ell - 1]$ avec distribution uniforme,
- choisit $u \in \{0, 1\}$ avec $\text{Prob}(u = 0) = \delta$, où δ est un paramètre qui reste à déterminer,
- si $u = 0$, il pose $R = hP$, sinon il pose $R = hP_a$,
- il ajoute (X, R, h, u) à la liste L.

Finalement, $\mathcal{B}$ envoie R à $\mathcal{A}$.

Requêtes d'évaluation de $\hat{H}$: elle sont traitées d'une façon similaire aux requêtes d'évaluation de H ($\mathcal{B}$ utilise une seconde liste $\hat{L}$, et renvoie un multiple de Q avec probabilité δ et un multiple de Q_b avec probabilité $1 - \delta$).

Requêtes d'extraction : pour répondre à une requête venant de $\mathcal{A}$ concernant la chaîne de caractères ID, l'algorithme $\mathcal{B}$:

- à l'aide des routines répondant aux requêtes d'évaluation de H et de $\hat{H}$, fait en sorte que L contienne un quadruplet de la forme (ID, R, h, u), et que $\hat{L}$ contienne un quadruplet de la forme $(ID, S, \hat{h}, \hat{u})$,
- teste si $u = 1$ ou si $\hat{u} = 1$, auquel cas il s'interrompt,
- calcule le couple $(hP_c, \hat{h}Q_c)$ et le renvoie à $\mathcal{A}$.

Essai : lorsqu'il reçoit l'essai $(ID_A, ID_B, \alpha, \beta)$ de la part de $\mathcal{A}$, l'algorithme $\mathcal{B}$

- utilise les routines répondant aux requêtes d'évaluation de H et de $\hat{H}$ pour s'assurer que L contient bien des couples de la forme (ID_i, R_i, h_i, u_i) et $\hat{L}$ des couples de la forme $(ID_i, S_i, \hat{H}_i, \hat{u}_i)$, pour $i = A, B$,
- choisit aléatoirement $t \in \{0, 1\}$,
- si $t = 0$, teste si $u_A = 1$ et $\hat{u}_B = 1$ (sinon s'interrompt), et alors renvoie $\alpha^{1/(h_A \hat{h}_B)}$ comme essai,
- si $t = 1$, teste si $u_B = 1$ et $\hat{h}_A = 1$ (sinon s'interrompt), et alors renvoie $\beta^{1/(h_B \hat{h}_A)}$ comme essai.

Supposons maintenant que $\mathcal{B}$ ne s'interrompe pas et renvoie γ comme essai. Avec probabilité $1/2$, on a $t = 0$, et donc $u_A = 1$, $\hat{u}_B = 1$, $H(ID_A) = h_A P_a = a h_A P$, $\hat{H}(ID_B) = \hat{h}_B Q_b = b \hat{h}_B Q$ et $\gamma = \alpha^{1/(h_A \hat{h}_B)}$. Indépendemment, avec probabilité $p_{\mathcal{A},1}$, on a $\alpha = e(H(ID_A), \hat{H}(ID_B))^c$. Donc l'événement suivant survient avec probabilité $p_{\mathcal{A},1}/2$:

$$
\begin{aligned}
\gamma &= \alpha^{1/(h_A \hat{h}_B)} = e\Big(H(ID_A), \hat{H}(ID_B)\Big)^{c/(h_A \hat{h}_B)} \\
&= e\Big(a\, h_A\, P, b\, \hat{h}_B\, Q\Big)^{c/(h_A \hat{h}_B)} = e(P, Q)^{abc},
\end{aligned}
$$

où la dernière égalité est une conséquence de la bilinéarité du couplage.

Une analyse similaire pour $t = 1$ montre que l'essai de $\mathcal{B}$ est correct (*i.e.* est la solution du problème GBDH) avec une probabilité totale $(p_{\mathcal{A},1} + p_{\mathcal{A},2})/2 \geqslant \epsilon/2$ si $\mathcal{B}$ ne s'interrompt pas.

Calculons maintenant la probabilité que $\mathcal{B}$ s'interrompe. On sait que $\mathcal{A}$ émet au plus q_E requêtes d'extraction. La probabilité de non-interruption lors de chaque requête étant δ^2, et la probabilité de non-interruption durant la phase d'essai étant de $(1 - \delta)^2$, la probabilité totale pour que $\mathcal{B}$ ne s'interrompe pas vaut au plus $\delta^{2q_E}(1 - \delta)^2$. En minimisant cette fonction, on trouve pour δ la valeur optimale $\delta = q_E/(1 + q_E)$, et une probabilité totale de non-interruption d'au moins $1/(e(1 + q_E))^2$. La probabilité pour que $\mathcal{B}$ renvoie la solution correcte pour le problème GBDH est donc d'au moins $\epsilon/(2e^2(1 + q_E)^2)$.

L'analyse de la complexité temporelle de $\mathcal{B}$ est assez simple et directe, excepté pour l'extraction d'une racine dans V. On notera qu'étant donnés un élément α dans un groupe (multiplicatif) arbitraire d'ordre ℓ et un entier $r \in [1, \ell - 1]$, la racine $\alpha^{1/r}$ peut être obtenue en calculant $s = r^{-1} \bmod \ell$ *via* l'algorithme d'Euclide étendu, puis en élevant α à la puissance s.

♣

Ceci montre donc que le problème GBDH et notre protocole sont polynomiallement équivalents, en quantifiant assez précisément de quelle façon les temps de calcul et les probabilités de réussite sont transformés. La difficulté du problème GBDH implique donc la sécurité de notre protocole.

3.7 Systèmes hiérarchiques

À Eurocrypt 2002, Horwitz et Lynn [37] ont introduit la notion de systèmes hiérarchiques basés sur l'identité. Cette notion est à rapprocher de celle de hiérarchie des authorités de certification pour les cryptosystèmes à clé publique ordinaire. Dans un système basé sur l'identité n'utilisant qu'un unique PKG, la charge de travail de ce dernier peut devenir énorme, puisqu'il lui revient de répondre, *via* des canaux sécurisés, aux requêtes d'extraction de clé privée de l'ensemble des utilisateurs du système. Dans un système hiérarchique, un PKG dit *primaire*, disposant d'une clé maître primaire, peut déléguer la génération de clé à des PKG secondaires, en leur transmettant des clés maîtres secondaires. Bien sûr, ce type de hiérarchie peut avoir un nombre quelconque de niveaux et non seulement deux. Il est alors souhaitable que la divulgation d'une clé maître à un niveau quelconque de la hiérarchie ne compromette que les clés privées engendrées par cette clé maître et ses descendantes dans la hiérarchie.

Dans [37], Horwitz et Lynn proposent un système de chiffrement hiérarchique basé sur l'identité inspiré du système de Boneh et Franklin. L'un des inconvénients de leur système est que la hauteur de la hiérarchie est fixée à deux, et qu'une collusion d'utilisateurs d'un PKG secondaire peut déterminer la clé maître secondaire détenue par ce dernier. Récemment, Gentry et Silverberg [32] ont proposé un système hiérarchique ne souffrant pas de cet inconvénient.

Chapitre 4

Construction de courbes ordinaires de faible degré MOV

4.1 Intérêt

Soit ℓ un diviseur premier de $\#E(\mathbb{F}_q)$ distinct de p et ne divisant pas $q-1$, et soit k le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ. Supposons que l'on veuille utiliser cette courbe dans un cryptosystème dont la sécurité repose sur le problème GBDH associé (en utilisant le couplage de Weil ou de Tate). Les seules attaques connues à ce jour contre de tels systèmes sont les attaques directement contre le logarithme discret, soit dans $E[\ell]$ (attaque A), soit dans $\mathbb{F}_{q^k}^*$ (attaque B). Nous avons vu en 2.1.1 qu'avec les meilleurs algorithmes connus aujourd'hui pour résoudre le problème du logarithme discret, l'attaque A nécessite un temps exponentiel en le nombre de bits de ℓ, alors que l'attaque B peut se réaliser en un temps sous-exponentiel en le nombre de bits de q^k.

Supposons maintenant que l'entier ℓ est voisin de $\#Eq$ [1]. D'après le théorème de Hasse (1.1), ℓ est alors voisin de q. Dans ces conditions, l'attaque A est exponentielle en $\log q$ alors que l'attaque B est sous-exponentielle en $k \log q$. Il apparaît donc clairement que l'entier k est un paramètre de sécurité dans ce système.

Pour fixer les idées, imaginons que q est un entier de 200 bits. Si $E(\mathbb{F}_q)$ est supersingulière, alors $k \leqslant 6$, et donc $k \log q \leqslant 1200$. Dans ces conditions, et étant données les performances connues des programmes implantant les attaques A et B contre le logarithme discret, l'attaque B est réputée être nettement moins coûteuse que l'attaque A (pour une comparaison entre les efficacités respectives des attaques A et B, nous renvoyons à [47]). Supposons maintenant que l'on puisse utiliser une courbe (nécessairement ordinaire) pour laquelle $k = 10$, toujours définie sur un corps de taille $\log q = 200$ bits. Le coût de l'attaque A est inchangé, alors que l'attaque B devient beaucoup plus coûteuse que pour $k = 6$, car la taille de $\mathbb{F}_{q^k}^*$ est alors de 2000 bits (on considère d'ailleurs souvent que, pour ces valeurs de q et k, les attaques A et B ont un coût sensiblement équivalent).

L'utilisation de courbes ordinaires peut donc permettre non seulement de se prémunir d'attaques (encore hypothétiques) qui seraient propres aux courbe supersingulières, mais aussi d'utiliser des courbes ayant un degré MOV k plus grand que 6, entraînant ainsi une meilleure résistance aux attaques connues et permettant de mieux ajuster le rapport entre la taille de la clé ℓ et la sécurité offerte.

[1] Par voisin, on entend signifier que $|\log \ell - \log \#Eq| = O(1)$.

4.2 Problématique

On souhaite donc construire des courbes elliptiques ordinaires $E(\mathbb{F}_q)$, dont le cardinal a un facteur premier ℓ de taille cryptographique (typiquement 200 bits), et dont le degré MOV k relativement à ℓ soit assez petit. Pour les raisons vues plus haut, il serait par ailleurs préférable de pouvoir construire de telles courbes avec $k \geq 6$.

Le problème est que l'entier k est (dans les cas où ℓ ne divise pas $q-1$) l'ordre de q modulo ℓ. Nous verrons en 5.2.2 que la taille de k est généralement proche de celle de ℓ, et donc beaucoup plus grande que désiré.

Nous verrons en 4.3 que, si a est un nombre premier, on peut espérer pouvoir construire une courbe elliptique $E(\mathbb{F}_q)$ de trace t (et donc de cardinal $q+1-t$) dans les cas où $4q-t^2$ peut s'écrire sous la forme Df^2, avec D relativement petit (plus de précisions sont données en 4.3). Il suffit alors de trouver quatre entiers q, t, ℓ et k tels que :

- q est premier,
- t est dans l'intervalle de Hasse associé à q, soit $|t| \leqslant 2\sqrt{q}$,
- ℓ est un diviseur premier de $q+1-t$ (sa taille doit être suffisante pour une utilisation cryptographique),
- $\ell \mid \Phi_k(q)$ (cette condition assure que k est bien l'ordre de q modulo ℓ).

La difficulté du problème vient de ces relations multiples entre les paramètres.

4.3 Génération de courbes par multiplication complexe

Nous exposons ici rapidement les idées permettant d'utiliser la multiplication complexe afin de construire des courbes elliptiques ordinaires de cardinal donné sur un corps fini premier. Nous ne définissons pas les notions utilisées, mais renvoyons à l'ouvrage de Cox [18] pour une exposition détaillée de la théorie de la multiplication complexe. Dans ce qui suit, l'entier q est supposé premier (de sorte que $q{=}p$).

Étant donné un entier t, il existe toujours une courbe elliptique $E(\mathbb{F}_q)$ de cardinal $q+1-t$. L'endomorphisme de Frobenius π (agissant sur $E(\overline{\mathbb{F}}_q)$) engendre alors, avec les endomorphismes de la forme $[r]$ ($r \in \mathbb{Z}$), un ordre $\mathcal{O}$ dans un corps quadratique imaginaire[2]. Le discriminant de cet ordre est donné par $\Delta = t^2 - 4q$. Si on écrit $\Delta = -f^2D$, où $-D$ est le discriminant fondamental et f le conducteur de l'ordre $\mathcal{O}$, alors on peut construire une courbe $E(\mathbb{F}_q)$ ayant le cardinal voulu comme une courbe ayant multiplication complexe par l'ordre maximal (de discriminant $-D$). On commence par calculer H, le polynôme de classes de Hilbert de $\mathbb{Q}(\sqrt{-D})$. On réduit alors H modulo q, puis on détermine une racine du polynôme ainsi obtenu (qui admet en fait une racine dans $\mathbb{F}_q$). On peut alors rapidement déterminer l'équation d'une courbe ayant le cardinal désiré d'après cette racine.

Dans la pratique, la mise en œuvre de cet algorithme est fortement limitée par la phase de calcul du polynôme H, dont le degré, correspondant au nombre de classes $h(D)$ de $\mathbb{Q}(\sqrt{-D})$, est asymptotiquement $\sqrt{D}$. Un grand nombre de variantes de cette méthode existent et permettent d'en améliorer les performances [4, 25, 26, 27], mais toutes restent impraticables dès que l'entier D est trop élevé. Actuellement, un programme écrit par Andreas Enge utilisant ces améliorations permet de générer par multiplication complexe des courbes pour lesquelles le nombre de classes associé est de 5000 en une quarantaine de minutes sur un Pentium III cadencé à 700 MHz.

[2]Sauf dans le cas où $t = 0$, qui correspond à une courbe supersingulière et qui ne nous intéresse pas ici.

4.4 La méthode de Miyaji, Nakabayashi et Takano

4.4.1 Méthode originale

Dans [57], Miyaji, Nakabayashi et Takano donnent des conditions nécessaires et suffisantes sur la trace d'une courbe $E(\mathbb{F}_q)$ de cardinal ℓ pour que la condition $\ell \mid q^k - 1$ soit vérifiée, dans les cas où $k = 3, 4$ ou 6. Le résultat qu'ils obtiennent est le suivant :

Théorème 4.1 *Soit $E(\mathbb{F}_q)$ une courbe elliptique de cardinal $\ell = q + 1 - t$ premier. Alors :*
- $\ell \mid q^3 - 1$ *si et seulement si*
  1. *il existe un entier r tel que $q = 12r^2 - 1$ et $t = -1 \pm 6r$, ou*
  2. *il existe un entier s pair tel que $q = p^r$ et $t = \pm\sqrt{q}$ (i.e. la courbe $E(\mathbb{F}_q)$ est supersingulière).*
- $\ell \mid q^4 - 1$ *si et seulement si*
  1. *il existe un entier r tel que $q = r^2 + r + 1$ et $t = -r$ ou $r + 1$, ou*
  2. *il existe un entier s impair tel que $q = 2^s$ et $t = \pm\sqrt{2q}$ (i.e. la courbe $E(\mathbb{F}_q)$ est supersingulière).*
- $\ell \mid q^6 - 1$ *si et seulement si*
  1. *il existe un entier r tel que $q = 4r^2 + 1$ et $t = 1 \pm 2r$, ou*
  2. *il existe un entier s impair tel que $q = 3^s$ et $t = \pm\sqrt{3q}$ (i.e. la courbe $E(\mathbb{F}_q)$ est supersingulière).*

La méthode employée pour démontrer ce théorème repose sur le fait que pour ces valeurs de k, on a $\varphi(k) = 2$, et c'est bien la raison qui fait que l'on ne peut directement obtenir de conditions du même type pour des valeurs de k supérieures à 6.

Toutefois ces conditions permettent, en utilisant la technique de la multiplication complexe exposée plus haut, de générer des courbes elliptiques ordinaires dont le degré MOV vaut 3, 4 ou 6.

4.4.2 Tentative d'extension de Barreto, Lynn et Scott

Une généralisation de la méthode de Miyaji, Nakabayashi et Takano aux cas où $\varphi(k) = 4$ est proposée dans [7]. Les auteurs ne se restreignent plus au cas où $\ell = \#E(\mathbb{F}_q)$, et recherchent des conditions suffisantes pour que $\ell \mid q^k - 1$. L'étape critique de cette généralisation est la résolution d'une équation diophantienne quartique elliptique (dans le cas où $\varphi(k) = 2$, on tombe sur l'équation de Pell). Aucune solution algorithmique satisfaisante n'étant connue pour ce problème, les auteurs ne sont pas parvenus à calculer des exemples de taille cryptographique en utilisant cette généralisation.

Dans le même article, Barreto, Lynn et Scott réussissent, par un paramétrage judicieux, à calculer des courbes convenables dans le cas où l'entier k est de la forme $2^i 3^j p^s$. On notera que ces courbes ont multiplication complexe par l'ordre quadratique de discriminant fondamental -3, ce qui est susceptible de les rendre vulnérable à certaines classes d'attaques (encore hypothétiques).

4.5 La méthode de Barreto, Lynn et Scott

Dans [7], Barreto, Lynn et Scott proposent aussi une méthode assez proche dans son esprit de celle présentée ci-dessus, mais basée sur un paramétrage différent. C'est cette méthode, qui n'est plus tout à fait une généralisation des travaux de Miyaji, Nakabayashi et Takano, que nous présentons ici.

4.5.1 Le paramétrage utilisé

On commence par fixer un degré MOV k désiré. Le but est ici encore de rechercher un entier premier q et une courbe $E(\mathbb{F}_q)$ dont le cardinal est divisible par un grand entier ℓ, le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ devant être égal à k.

On pose encore $m = \#E(\mathbb{F}_q) = q + 1 - t$, et on introduit un entier u non nul. On va en fait chercher m sous la forme $m = h\Phi_k(u)$, ℓ étant un facteur de $\Phi_k(u)$. On impose aussi à la trace t d'être de la forme $t = u^d + 1$, avec $d \in [1, \varphi(k)/2]$ (cette dernière condition impose à m d'être dans l'intervalle de Hasse). On supposera de plus d premier avec k.

On a donc $q \equiv t - 1$ $[\ell]$. Pour que k soit l'ordre de q modulo ℓ, il faut que $\ell \mid \Phi_k(q)$, donc que $\ell \mid \Phi_k(u^d)$. Cette relation est toujours vérifiée puisque l'on a imposé $\ell \mid \Phi_k(u)$, et que d et k sont premiers entre eux.

Comme $t = u - 1$ et que $m = q + 1 - t = h\Phi_k(u)$, cette courbe est à multiplication complexe par l'ordre de discriminant $\Delta = t^2 - 4q$, soit

$$\Delta = f^2 D = 4h\Phi_k(u) - (u^d - 1)^2. \tag{4.1}$$

La construction sera possible pourvu que le discriminant fondamental D reste de taille limitée.

4.5.2 Recherche de paramètres convenables

Supposons que les entiers D et u soient fixés, et que $t \nmid D$. On recherche alors une solution en h (et f) à l'équation 4.1. En posant $A = 4\Phi_k(u)$ et $B = (u^d - 1)^2$, cette équation peut se récrire $f^2 D = Ah - B$.

Pour résoudre cette équation, on commence par déterminer le plus petit h_0 positif tel que $D \mid Ah_0 - B$. Il existe donc un z_0 tel que $Ah_0 - B = Dz_0$. Trois cas peuvent se présenter :

- si B est un multiple de D, alors $h_0 = 0$ et z_0 se calcule facilement ;
- si A et B sont inversibles modulo D, alors $h_0 = BA^{-1}$ modulo D, et $z_0 = (Ah_0 - B)/D$;
- si B est inversible modulo D et que A ne l'est pas, alors il n'existe pas de solutions à l'équation 4.1 pour ce choix de paramètres D et u.

Cette discussion montre que h_0 est toujours inférieur à D. On pose alors $h_i = h_0 + iD$ et $z_i = z_0 + iA$, de sorte que $Dz_i = Ah_i - B$. Les solutions que l'on recherche correspondent aux cas où z_i est un carré ($z_i = f^2$). z_0 doit donc être un résidu quadratique modulo A. Si tel est le cas on peut poser $i_0 = (f_0^2 - z_0)/A$, où f_0 est une racine carrée de z_0 modulo A. On peut alors rechercher les solutions sous la forme $i_\alpha = i_0 + \alpha A$, soit $z_{i_\alpha} = f_0^2 + \alpha A^2$.

On notera que, si la taille de ℓ est fixée, rechercher la plus petite solution h_0 en h revient à chercher à minimiser la taille de q, et donc à maximiser le ratio entre la taille de q et la taille de ℓ (ce qui est en général souhaitable).

4.5.3 Algorithme

Si l'on fixe $d = 1$, la technique de recherche de paramètres décrite ci-dessus peut se transcrire directement en la procédure suivante :

procédure SMALLK (k, L)

1. choisir un entier u aléatoire de taille $L - \varphi(k)$;
2. $\ell \leftarrow \Phi_k(u)$, $t \leftarrow u + 1$, $A \leftarrow 4\ell$, $B \leftarrow (u-1)^2$;
3. **si** ℓ n'est pas premier **ou** A est un carré **alors**
   3.1. **renvoyer** SMALLK(k, L) ;
4. déterminer le plus petit h_0 positif tel que $Ah_0 - B = z_0 D$;
5. **si** z_0 non résidu quadratique modulo ℓ **alors**
   5.1. **renvoyer** SMALLK(k, L) ;
6. calculer une racine carrée f de z_0 modulo ℓ ;
7. **si** $4 \nmid f^2 - z_0$ **alors**
   7.1. **renvoyer** SMALLK(k, L) ;
8. $i_0 \leftarrow (f^2 - z_0)/A$, $h = h_0 + i_0 D$, $m = h\ell$, $q = m + t - 1$;
9. **si** q n'est pas premier **alors**
   9.1. **renvoyer** SMALLK(k, L) ;
   sinon
   9.2. construire la courbe $E(\mathbb{F}_q)$ de cardinal m par multiplication complexe ;

Remarques :

- L'argument L de cette procédure correspond à la taille minimale en bits du paramètre ℓ que l'on souhaite obtenir.
- L'étape 4. s'effectue rapidement en utilisant les arguments introduits en 4.5.2.
- Le test en 7. sert à vérifier que V est bien une racine carrée de z_0 modulo A.

La complexité de cette procédure n'est pas analysée dans [7]. Des exemples de courbes de tailles cryptographiques générées par cette procédure sont inclus dans l'article.

4.6 Une méthode originale

Au cours de ce stage nous avons, avec Andreas Enge et François Morain, développé une méthode originale permettant de construire des courbes elliptiques ordinaires ayant un degré MOV spécifié [21].

4.6.1 Approche adoptée

Notre technique est la suivante : supposons que l'entier k (qui sera le degré MOV) est fixé. Toute puissance q d'un nombre premier peut s'écrire de façon unique sous la forme :

$$q = n^2 + a \text{ avec } n \geqslant 1 \text{ et } 0 \leqslant a \leqslant n$$

ou

$$q = n^2 + n + a \text{ avec } n \geqslant 1 \text{ et } 1 \leqslant a \leqslant n.$$

Nous allons construire une courbe $E(\mathbb{F}_q)$ par la technique de la multiplication complexe, en imposant la relation $|t| = \lfloor 2\sqrt{q} \rfloor$ pour la trace, soit

$$t = \pm 2n \text{ pour } q = n^2 + a$$

et

$$t = \pm(2n + 1) \text{ pour } q = n^2 + n + a.$$

k	$R_k(Y)$
2	$4Y + 4$
3	$16Y^2 + 12Y + 9$
4	$16Y^2 + 4$
5	$256Y^4 + 320Y^3 + 160Y^2 + 25$
6	$16Y^2 - 4Y + 1$
7	$4096Y^6 + 7168Y^5 + 5376Y^4 + 2240Y^3 + 784Y^2 - 196Y + 49$
8	$256Y^4 + 256Y^3 + 128Y^2 - 32Y + 4$
9	$4096Y^6 + 6144Y^5 + 2304Y^4 + 192Y^3 + 576Y^2 - 108Y + 9$
10	$256Y^4 + 64Y^3 + 96Y^2 - 16Y + 1$
11	$1048576Y^{10} + 2883584Y^9 + 3604480Y^8 + 2703360Y^7 + 1351680Y^6 + 473088Y^5 + 123904Y^4$ $+17424Y^2 - 2420Y + 121$

TAB. 4.1 – Valeurs du résultant R_k

Nous nous limitons ici au cas où $q = n^2 + a$ et $t = 2n$, les autres cas se traitant de façon tout à fait similaire. Le cardinal de la courbe $E(\mathbb{F}_q)$ s'écrit alors

$$m = q + 1 - t = (n-1)^2 + a,$$

et l'entier m doit être divisible par un entier premier ℓ. Imposer le fait que le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ soit égal à k revient à imposer la relation $\ell \mid \Phi_k(q)$ soit, puisque $\ell \mid q + 1 - t$:

$$\Phi_k(t-1) \equiv 0 \bmod \ell,$$

où Φ_k est le $k^{\text{ème}}$ polynôme cyclotomique. En combinant ces équations, il vient que les entiers n, a et ℓ sont reliés par

$$\begin{cases} \Phi_k(2n-1) & \equiv 0 \bmod \ell \\ (n-1)^2 + a & \equiv 0 \bmod \ell \end{cases} \tag{4.2}$$

Pour éliminer la variable n, considérons les polynômes $P_k(X) = \Phi_k(2X - 1)$ et $Q(X, a) = (X-1)^2 + Y$, ainsi que leur résultant

$$R_k(Y) = \mathrm{Res}_X(P_k(X), Q(X, Y)).$$

La proposition suivante établit certaines propriétés des polynômes R_k, dont les premières valeurs sont données dans le Tableau 4.1.

Proposition 4.1 *Le polynôme $R_k(Y) \in \mathbb{Z}[Y]$ est irréductible et son terme dominant est $4^{\varphi(k)}Y^{\varphi(k)}$.*

Démonstration : Supposons $k \geqslant 2$ (pour $k = 2$, l'assertion est triviale). En écrivant le résultant d'un polynôme f de coefficient dominant c avec un polynôme g sous la forme

$$\mathrm{Res}(f, g) = c^{\deg g} \prod_{\alpha \text{ root of } f} g(\alpha),$$

(voir par exemple [66]), il vient l'égalité

$$R_k(Y) = \left(2^{\varphi(k)}\right)^2 \prod \left(Y + \left(\frac{\zeta^i - 1}{2}\right)^2\right),$$

où ζ est une racine primitive $k^{\text{ème}}$ de l'unité et où le produit porte sur les entiers $i \in [1, k-1]$ premiers avec k. En particulier, le terme dominant de $R_k(Y)$ est bien $4^{\varphi(k)} Y^{\varphi(k)}$.

Soit $\alpha = \left(\frac{\zeta-1}{2}\right)^2$ une racine de $R_k(Y)$. Alors soit α est encore un générateur de $\mathbb{Q}(\zeta)$, auquel cas $R_k(Y)$ est irréductible, soit $\mathbb{Q}(\alpha)$ est un sous-corps d'indice 2 de $\mathbb{Q}(\zeta)$. Dans ce dernier cas α est de degré $\varphi(k)/2$ sur $\mathbb{Q}$, et il existe donc un polynôme unitaire $S \in \mathbb{Q}[X]$ de degré $\varphi(k)/2$ tel que $S(4\alpha) = S\left((\zeta-1)^2\right) = 0$. Comme $S\left((X-1)^2\right)$ est unitaire et de degré $\varphi(k)$, on a donc

$$\Phi_k(X) = S\left((X-1)^2\right).$$

Mais le coefficient de $X^{\varphi(k)-1}$ dans $S\left((X-1)^2\right)$ vaut $-\varphi(k)$, alors que dans $\Phi_k(X)$ il est égal à l'opposé de la somme des racines primitives $k^{\text{èmes}}$ de l'unité. L'irréductibilité de $R_k(Y)$ découle de cette contradiction.

♣

Pour obtenir une solution du système 4.2, nous fixons des valeurs de a. Le discriminant $\Delta = t^2 - 4q = -4a = -f^2 D$ est alors lui aussi fixé, et on prendra soin de choisir a tel que D ne soit pas trop grand (il faudra pouvoir construire le corps de classes de Hilbert associé). On essaie ensuite de factoriser l'entier $R_k(a)$ et d'obtenir un facteur premier ℓ suffisamment grand. En cas de succès, on calcule $\operatorname{pgcd}(P_k(X), Q(X,a))$ modulo ℓ pour obtenir l'entier n. On teste alors si $n^2 + a$ est premier (obtenir une puissance d'un nombre premier semble irréaliste), auquel cas on construit $E(\mathbb{F}_q)$ par multiplication complexe (avec le discriminant fondamental $-D$).

On notera que cette méthode produit des courbes pour lesquelles la taille de q est environ le double de la taille de ℓ. Cet inconvénient (on préfère toujours, notamment pour des raisons d'efficacité dans l'implantation, que ℓ soit proche de q en taille) n'est dans la plupart des applications pas rédhibitoire, excepté bien sûr dans le protocole de signatures courtes proposé dans [11]. Il est intéressant de constater que les courbes générées par la méthode décrite en 4.5 présentent le même inconvénient.

4.6.2 Algorithme

Notre méthode se transcrit naturellement en une procédure prenant en entrée l'entier k ainsi qu'un paramètre de sécurité L qui correspond à la taille minimale du sous-groupe de la courbe elliptique que l'on désire obtenir. Ce programme produit donc une courbe elliptique $E(\mathbb{F}_q)$ (définie sur un corps premier) dont le cardinal est divisible par un entier premier $\ell \geqslant L$, et dont le degré MOV relativement à ℓ est k.

procédure SMALLKBIS (k, L)
  pour a **de** 1 **à** $a_{\max}$ **faire**
    1. factoriser $R_k(a)$;
    2. **si** $R_k(a)$ a un facteur premier $\ell \geqslant L$ **alors**
      2.1 déterminer une racine n de $\operatorname{pgcd}(P_k(X), Q(X,a))$ modulo ℓ ;
      2.2 **pour** s **de** 0 **à** $s_{\max}$ **faire**
        si $a \leqslant n + s\ell$ **alors**
          – calculer $q = (n+s\ell)^2 + a$;
          – **si** q est premier **alors** construire $E(\mathbb{F}_q)$ par multiplication complexe ;

Remarques :

- L'entier n peut être remplacé par tout nombre qui lui est congru modulo ℓ ; c'est pourquoi l'on introduit le paramètre s en 2.2 (la valeur $s_{\max}$ sera choisie relativement petite).
- Pour que la technique de génération par multiplication complexe puisse s'appliquer, il suffit que $4a$ s'écrive f^2D avec f quelconque et D de la taille désirée. Une amélioration consiste donc à boucler sur (D, f) plutôt que sur a, ce qui permet de contrôler la taille de D.
- La factorisation de $R_k(a)$ peut s'effectuer par une phase de crible permettant de factoriser plusieurs valeurs simultanément. En pratique, nous introduisons une borne B et ne retenons que les valeurs de $R_k(a)$ composées de petits facteurs premiers inférieurs à B et d'un grand cofacteur premier.
- Plutôt que de commencer par $a = 1$, il est préférable de calculer la plus petite valeur de a telle que $R_k(a) \geqslant L$, ce qui est aisé puisque $R_k(a) \sim (4a)^{\varphi(k)}$. Pour des raisons d'efficacité, il est préférable de garder $R_k(a)$ proche de L. Ceci devient impossible dès que $\varphi(k)$ devient trop grand puisque les valeurs de $R_k(a)$ deviennent beaucoup plus grandes que L, et il est alors quasiment impossible de trouver un facteur de $R_k(a)$ dont la taille soit proche de celle de L.

4.6.3 Performances

Procédons à une rapide analyse heuristique de cet algorithme, en supposant pour simplifier que l'on ne retient que les entiers $R_k(a)$ premiers, et que la probabilité pour que cet événement survienne est $O(1/\log L)$. L'entier n est de taille environ L^2, et q sera donc lui aussi premier avec probabilité $O(1/\log L)$. La probabilité de succès est donc en $O(1/\log^2 L)$.

Pour avoir une idée plus précise des performances de notre méthode, nous avons implanté la recherche des paramètres q, ℓ, n et a en MAGMA [17]. Les temps nécessaires à la génération de ces paramètres pour des tailles cryptographiques (L de 160 à 200 bits) vont de 1.5 secondes pour $k = 12$ à environ 30 secondes pour $k = 50$ sur un Pentium III cadencé à 450 MHz. La construction des courbes par multiplication complexe a été menée à bien avec un programme développé par Andreas Enge, écrit en C++ et basé sur les librairies GMP [33], MPFR [34], MPC [24] et NTL [63]. Des exemples numériques ainsi que les temps de calcul correspondants sont donnés en annexe.

Chapitre 5

Implantation

5.1 Objectifs

J'ai décidé d'implanter les algorithmes de calcul de couplages (de Tate et de Weil) sur des courbes elliptiques définies sur des corps premiers en grande caractéristique (avec en vue des applications cryptographiques). Mon choix s'est porté sur une implantation en langage C, en utilisant la librairie GMP [33] pour l'arithmétique sur les grands entiers.

Mon objectif était principalement de voir quelles performances étaient possibles. En effet, la plupart des résultats publiés concernent des calculs de couplages en petite caractéristique (souvent 3, qui seul permet d'atteindre le degré MOV maximal de 6 sur les courbes supersingulières), et dans de tels cas certaines améliorations sont possibles [6, 31]. Ce fut aussi une bonne occasion de mettre en œuvre un certain nombre d'algorithmes dont j'avais souvent entendu parler sans jamais les implanter, et donc d'acquérir quelque expérience en matière de choix d'algorithmes arithmétiques.

D'autre part, cette implantation me semblait un bon complément au côté assez théorique du reste du stage.

5.2 Algorithmes utilisés

Comme précisé plus haut, l'objectif était de calculer des couplages sur des courbes $E(\mathbb{F}_p)$, avec p premier. J'ai donc travaillé sur de telles courbes, dont le cardinal était divisible par un (grand) entier premier ℓ (typiquement, ℓ était de l'ordre de 200 bits). Les courbes que j'ai utilisées sont du type des exemples de taille cryptographique donnés en B.2, donc la taille de p était typiquement de 400 bits. J'ai travaillé avec des degrés MOV k allant de 6 à 20. On notera ici T_p le nombre de bits de p.

Les comparaisons de performances des différentes opérations ont été effectuées sur un Athlon 1600 XP.

5.2.1 Calculs dans les corps finis

Les opérations dans le corps de base $\mathbb{F}_p$ ont été implantées en utilisant directement les fonctions de manipulation des grands entiers de GMP.

Le corps $\mathbb{F}_{p^k}$ a été représenté comme le quotient de $\mathbb{F}_p$ par un polynôme de degré k, de poids minimal, et dont tous les coefficients non-nuls valaient 1, de façon à optimiser la réduction modulo ce polynôme. Les éléments de $\mathbb{F}_{p^k}$ sont alors des polynômes de degré au plus $k-1$. L'addition dans $\mathbb{F}_{p^k}$ se ramène à k additions dans $\mathbb{F}_p$, qui se font comme de simples additions d'entiers, en soustrayant éventuellement p ensuite si le résultat obtenu dépasse p. Les soustractions se font de manière similaire.

L'un des problèmes a été d'implanter la multiplication dans $\mathbb{F}_{p^k}$ le plus efficacement possible. Après plusieurs essais (multiplication naïve de polynômes, stockage d'un polynôme dans un unique grand entier GMP puis utilisation de la multiplication de GMP, algorithme Toom-Cook avec découpage des polynômes en trois parties), j'ai retenu l'algorithme de Karatsuba (branchant sur une multiplication naïve suivant les degrés). Il faut noter que les temps de calcul de ces algorithmes ne sont pas directement ceux auxquels on pourrait s'attendre en analysant le nombre de multiplications de grands entiers qu'ils effectuent, car la réduction modulo p d'un entier de $2T_p$ bits est elle aussi coûteuse (un peu plus de deux fois plus coûteuse que la multiplication de deux entiers de taille T_p). On peut donc, en ne plaçant les réductions modulo p qu'aux endroits où elles sont vraiment nécessaires (soit avant les multiplications) gagner un temps non négligeable. En particulier, lorsque l'on veut réduire un polynôme de degré $2(k-1)$ dont les coefficients sont de taille $2T_p$, il est préférable d'effectuer tout d'abord la réduction modulo le polynôme engendrant le corps (qui est très efficace de par la forme de ce polynôme : elle se réduit à $(d-1)T_p$ soustractions d'entiers de taille $2T_p$, où d est le poids du polynôme engendrant le corps) avant de réduire modulo p les k coefficients du polynôme ainsi obtenu.

Enfin, l'inversion dans $\mathbb{F}_{p^k}$ se fait par un calcul de pgcd étendu avec le polynôme engendrant le corps. J'ai utilisé l'algorithme d'Euclide naïf, mais certaines améliorations sont sans doute possibles (bien que le degré k des polynômes en jeu soit assez petit).

5.2.2 Calcul du couplage de Tate

L'algorithme décrit en 1.5.2 se transcrit directement en la procédure suivante, permettant de calculer le couplage de Tate entre un point $P \in E(\mathbb{F}_{q^k})[\ell]$ et un point $Q \in E(\mathbb{F}_{q^k})$:

procédure TatePairing $(P, Q, E(\mathbb{F}_q), \ell)$
1. générer un point aléatoire $T \in E(\mathbb{F}_q)$ et calculer $Q + T$;
2. initialisation : $a \leftarrow 1$, $b \leftarrow 1$, $R \leftarrow P$;
3. **pour** i **de** $L-2$ **à** 0 **faire**
   - 3.1. $S \leftarrow [2]R$;
   - 3.2. $a \leftarrow a^2 \cdot d_{R,R}(Q+T) \cdot d_S(T)$;
   - 3.3. $b \leftarrow b^2 \cdot d_{R,R}(T) \cdot d_S(Q+T)$;
   - 3.4. $R \leftarrow S$;
   - 3.5. **si** $\ell_i = 1$ **alors**
     - 3.5.1. $S \leftarrow R + P$;
     - 3.5.2. $a \leftarrow a \cdot d_{R,P}(Q+T) \cdot d_S(T)$;
     - 3.5.3. $b \leftarrow b \cdot d_{R,P}(T) \cdot d_S(Q+T)$;
     - 3.5.4. $R \leftarrow S$;
4. **si** $a = 0$ **ou** $b = 0$ **alors**
   - 4.1. **renvoyer** TatePairing$(P, Q, E(\mathbb{F}_q), \ell)$;

   sinon
   - 4.2. **renvoyer** a/b ;

On notera que la chaîne d'addition retenue ici correspond à la technique classique d'exponentiation rapide avec lecture de l'exposant ℓ (dont l'écriture binaire est $\ell_{L-1} \dots \ell_1 \ell_0$) de la gauche vers la droite.

Le test 4. permet de tester *a posteriori* que T et $Q+T$ ne sont pas présents dans la chaîne d'addition utilisée pour exponentier P. Si T n'est pas de ℓ-torsion, $Q+T$ ne l'est pas non plus et la procédure termine. Pour les courbes que j'utilisais, la probabilité de rappeler la procédure était négligeable (cette procédure dépend de la structure du groupe $E(\mathbb{F}_q)$, et pour q de 400 bits et ℓ de 200 bits, cette probabilité est inférieure à 2^{-200}).

Nous n'avons pas encore évoqué la représentation des points de la courbe $E(\mathbb{F}_{q^k})$. En fait, toutes les représentations (coordonnées affines, projectives, jacobiennes, ...) sont possibles, il suffit d'adapter les formules permettant le calcul de $d_{A,B}(C)$ et de $d_A(C)$. Le choix doit se faire en fonction du rapport entre le coût d'une inversion et celui d'une multiplication dans $\mathbb{F}_{q^k}$. Ce rapport doit être assez élevé pour justifier l'abandon des coordonnées affines, ce qui n'est pas le cas avec la taille typique de mes paramètres.

En coordonnées affines, les formules les plus simples pour évaluer $d_{A,B}(C)$ et $d_A(C)$ sont :

$$d_{A,B}(C) = (y_B - y_A)(x_B - x_C) - (x_B - x_A)(y_B - y_C),$$

et

$$d_A(C) = (3x_A^2 + \alpha)(x_A - x_C) - 2y_A(y_A - y_C),$$

pour une courbe $E(\mathbb{F}_{q^k})$ dont l'équation est de la forme

$$y^2 = x^3 + \alpha x + \beta.$$

On notera que ces formules peuvent se calculer avec respectivement deux et trois multiplications (on néglige le coût des additions et soustractions). Notons M (resp. I) le coût d'une multiplication (resp. d'une inversion) dans $\mathbb{F}_{q^k}$. On rappelle que le coût d'une addition de points de $E(\mathbb{F}_{q^k})$ est alors de $I + 3M$, et celui d'un doublage de point $I + 4M$. Les étapes 3.1. à 3.4. de notre procédure ont donc un coût de $I + 20M$, et les étapes 3.5.1. à 3.5.4. un coût de $I + 17M$. Le coût global de la procédure se déduit alors directement de la forme de ℓ.

Enfin, une procédure de calcul de couplages de Weil peut facilement être dérivée de ce programme, la complexité étant alors double.

Validité de la procédure

Soient $P \in E(\mathbb{F}_{q^k})[\ell]$ et $Q \in E(\mathbb{F}_{q^k})$. Supposons que le point T généré par la procédure ne soit pas de ℓ-torsion. Définissons alors, pour tout entier $k \in [0, \ell]$, la fonction f_k par :

$$k\left((P) - (\infty)\right) = ([k]P) - (\infty) + \operatorname{div}(f_k)$$

(on a donc $f_1 = 1$).

L'invariant de la boucle principale est alors le suivant : pour tout entier $i \in [0, L-2]$, les valeurs de a et b à la fin de la boucle 3. sont telles que

$$\frac{a}{b} = \frac{f_{\hat{\ell}_i}(Q+T)}{f_{\hat{\ell}_i}(T)},$$

où l'entier $\hat{\ell}_i$ est défini par

$$\hat{\ell}_i = \sum_{j=i}^{L-1} \ell_j 2^{j-i}.$$

Ceci découle directement des propriétés des diviseurs vues en 1.1.4, et montre donc la validité de la procédure.

Influence de la valeur de k

Ce qui précède montre que la procédure proposée, comme tous les algorithmes connus pour l'évaluation de couplages de Weil ou de Tate, a une complexité temporelle de la forme $O(\log \ell \cdot M(q,k))$, où $M(q,k)$ est la complexité d'une multiplication dans $\mathbb{F}_{q^k}$.

Comme, pour les applications cryptographiques, la taille de q est généralement de plusieurs centaines de bits, $M(q,k)$ croîtra rapidement en fonction de k (la croissance étant au moins en $k \log k$).

Or k est le degré MOV de $E(\mathbb{F}_q)$ relativement à ℓ, qui correspond à l'ordre de q modulo ℓ et a donc *a priori* de grande chances d'avoir une taille proche de celle de ℓ, auquel cas la valeur de $M(q,k)$ serait bien évidemment trop élevée pour espérer pouvoir calculer des couplages.

Plus précisément, Bamasubramanian et Koblitz ont montré le théorème suivant :

Théorème 5.1 *Soit $M \in \mathbb{N}$, soit (p,E) une paire aléatoire constituée d'un entier premier $p \in [M/2, M]$ et d'une courbe elliptique E définie sur $\mathbb{F}_p$ ayant un nombre premier ℓ de points. Alors la probabilité pour que le degré MOV k de E relativement à ℓ vérifie $k \leqslant (\log p)^2$ est inférieure à*

$$c\frac{(\log M)^9(\log\log M)^2}{M}$$

pour une certaine constante c effectivement calculable.

Démonstration : voir [5], théorème 2.

♣

5.2.3 Améliorations

Comme nous l'avons mentionné, lorsque la courbe $E(\mathbb{F}_q)$ est supersingulière, le degré MOV maximal $k=6$ ne peut être atteint qu'en caractéristique 3. De ce fait, certains articles récents [6, 31] décrivent les optimisations possibles s'appliquant au calcul de couplages de Tate en caractéristique 3. Bon nombre de ces optimisations sont cependant spécifiques à cette caractéristique.

Les optimisations que nous donnons ici peuvent s'appliquer quelle que soit la caractéristique.

Optimiser la chaîne d'addition-soustraction

Toutes les optimisations connues pour l'exponentiation de point sont applicables au calcul du couplage de Tate. Cependant, la complexité peut être modifiée. En effet, toute chaîne d'addition-soustraction va additionner, soustraire et doubler des diviseurs de la forme

$$D_k = ([k]P) - (\infty) + \operatorname{div}(f_k).$$

Les phases de doublages auront toujours la même complexité, mais dans le cas d'une addition (ou d'une soustraction) concernant D_k et $D_{k'}$ (étapes 3.5.1. à 3.5.4. de la procédure), si k et k' sont distincts de 1, alors f_k et $f_{k'}$ seront *a priori* aussi différentes de 1, et le surcoût (par rapport au cas où $k=1$ ou $k'=1$) sera de deux multiplications, pour arriver à un coût total de $I+19M$.

On notera que si l'on travaille toujours sur la même courbe, il est intéressant de rechercher une chaîne d'addition-soustraction pour ℓ qui soit la plus courte possible (même si le problème est NP-complet en toute généralité).

La méthode (générique) que j'utilise est le fenêtrage de l'exposant en base 16, connue aussi sous le nom de *sliding window*, et exposée en détail dans [8].

Cas particulier où $P \in E(\mathbb{F}_q)[\ell]$

Rappelons que, pour que le résultat du couplage soit défini de façon unique, il convient d'élever le résultat produit par notre procédure à la puissance $(q^k-1)/\ell$. On se place de plus dans le cas où $q-1 \mid (q^k-1)/\ell$ (il est facile de voir que les courbes ordinaires de faible degré MOV produites par notre méthode vérifient toujours cette condition).

Supposons maintenant que le point P de ℓ-torsion ait ses coordonnées dans le corps $\mathbb{F}_q$. À tout moment dans l'exécution de la procédure, les points R et S auront aussi leurs coordonnées dans $\mathbb{F}_q$, et donc les valeurs $d_S(T)$ et $d_{R,R}(T)$ seront toujours dans $\mathbb{F}_q$. Élevées à la puissance $(q^k-1)/\ell$, ces valeurs donnent donc 1, ce qui fait que l'on peut les omettre. Cela revient à noter que dans ce cas,

$$t_\ell(P,Q) = f_P(Q+T)^{\frac{q^k-1}{\ell}}.$$

De plus, les opérations arithmétiques nécessaires à l'exponentiation de P se déroulent toutes dans $\mathbb{F}_q$ et non dans $\mathbb{F}_{q^k}$: leur coût est donc bien moindre.

Enfin, lorsque l'on doit calculer plusieurs couplages successifs entre un même point P de ℓ-torsion et plusieurs points $Q_1,\ldots,Q_n$ (ce qui est souvent le cas avec les cryptosystèmes basés sur les couplages), il est possible de précalculer entièrement les multiples de P apparaissant dans la chaîne d'addition-soustraction que l'on utilise. Après cette phase de précalcul, chaque étape de doublage (resp. d'addition) de diviseurs aura un coût de $9M$ (resp. de $7M$), ce qui devient particulièrement intéressant.

5.3 Résultats obtenus

Les deux premières courbes utilisées sont des courbes générées par la méthode décrite en 4.6, ce qui explique le rapport de deux entre les tailles de q et de ℓ. La troisième courbe est une courbe engendrée par la méthode de Miyaji, Nakabayashi et Takano exposée en 4.4.1 avec $k=3$, et la quatrième courbe est une courbe supersingulière avec $k=2$. On notera que dans les quatre cas, le nombre premier q est sans structure particulière : aucune optimisation utilisant la forme de q n'a été utilisée.

Le temps de l'exponentiation finale n'est pas compté dans les temps de calcul de couplages de Tate (on notera que cette exponentiation n'est nécessaire que lorsque l'on désire comparer entre elles des valeurs de couplages, et il est alors préférable de quotienter les valeurs à comparer pour ne procéder qu'à une unique exponentiation).

ℓ	161 bits	212 bits	160 bits	160 bits
q	324 bits	428 bits	160 bits	512 bits
k	7	10	3	2
Multiplication dans $\mathbb{F}_{q^k}$	31 μs	88 μs	6 μs	9 μs
Calcul de $t_\ell(P,Q)$, $P,Q \in E(\mathbb{F}_{q^k})$	180 ms	620 ms	32 ms	51 ms
Calcul de $e_\ell(P,Q)$, $P,Q \in E(\mathbb{F}_{q^k})$	360 ms	1.2 s	64 ms	102 ms
Calcul de $t_\ell(P,Q)$, $P \in E(\mathbb{F}_q)$, $Q \in E(\mathbb{F}_{q^k})$	37 ms	110 ms	8.9 ms	21 ms
Exponentiation finale (exposant $(q^k-1)/\ell$)	100 ms	540 ms	2.9 ms	12 ms

(Temps obtenus sur un Athlon 1600 XP)

On notera que le temps de l'exponentiation finale est loin d'être négligeable relativement aux temps de calculs de couplages (en particulier lorsque le point P est dans $\mathbb{F}_q$).

A titre de comparaison, le calcul d'un couplage de Weil entre deux points de $E(\mathbb{F}_{q^k})$ avec l'implantation fournie par MAGMA nécessite environ 12 secondes sur la première courbe et 44 secondes sur la seconde. Les autres implantations de couplages de Tate dont les performances sont publiées (principalement [6]) calculent ces couplages sur des courbes supersingulières seulement, en caractéristique 2, 3 ou bien en grande caractéristique p (de taille 512 bits par exemple). Les degrés MOV correspondants sont alors plus petits (4, 6 et 2 respectivement), et un certain nombre d'optimisations sont possibles (comme le choix d'un nombre premier p de forme spéciale) pour accélérer les multiplications.

Conclusion et perspectives

Nous avons vu que les couplages sur les courbes elliptiques permettent la mise en œuvre de cryptosystèmes basés sur l'identité, et en particulier de systèmes de chiffrement et de création de clé commune basés sur l'identité, pour lesquels les seules autres solutions connues sont peu pratiques. De plus, on peut en général prouver que la sécurité de ces protocoles est directement reliée à un problème dit problème GBDH. Cependant, ce dernier problème (en particulier les instances utilisant les couplages sur les courbes elliptiques) a été relativement peu étudié. Joux [40] a synthétisé les relations connues entre les différents problèmes liés au problème GBDH, mais on ne sait toujours pas par exemple s'il existe des attaques spécifiques contre ce problème.

On notera que les courbes elliptiques ordinaires sur lesquelles les couplages sont efficacement calculables, qui présentent un certain nombre d'avantages par rapport aux courbes supersingulières, sont relativement difficiles à trouver. De plus, les méthodes du type de celle que nous avons trouvée pour générer de telles courbes présentent certains inconvénients. En particulier, la taille des courbes engendrées est en général deux fois plus grande que la taille du sous-groupe utilisable, et ces courbes sont définies sur des corps premiers. Il serait intéressant de savoir si l'on peut s'affranchir de ces limitations.

Par ailleurs les couplages de Weil et de Tate sont aussi définis sur les jacobiennes de courbes hyperelliptiques. S'il a déjà été démontré que, comme pour les courbes elliptiques, les couplages sont efficacement calculables sur les courbes hyperelliptiques supersingulières, il reste à voir si l'on ne peut générer des courbes non supersingulières utilisables dans des systèmes utilisant les couplages.

Enfin on peut noter que les cryptosystèmes présentés dans ce mémoire peuvent utiliser tout couplage pour lequel le problème GBDH est difficile. Il serait donc intéressant de trouver d'autres groupes pour lesquels de tels couplages peuvent être définis, ou encore des groupes pour lesquels on puisse définir une application multilinéaire présentant des propriétés similaires (ce qu'on récemment évoqué Boneh et Silverberg dans [12]).

Annexe A

Cryptosystèmes basés sur l'identité n'utilisant pas de couplages

A.1 Le protocole de signature de Shamir

Il s'agit du protocole initialement proposé par Shamir dans [62]. Comme pour RSA, la sécurité de ce protocole est basé sur la difficulté du problème de factorisation : le PKG dispose du produit $N = pq$ de deux nombres premiers dont il est le seul à connaître la factorisation. Étant donnés un entier e premier avec $\varphi(N)$ et un entier i quelconque, le PKG peut donc facilement déterminer un entier g tel que $g^e \equiv i$ $[N]$. Sans la connaissance de la factorisation de N, ce même problème devient difficile : c'est ce fait qu'exploite le protocole de Shamir.

Ce protocole se décompose en quatre algorithmes :

- **Génération des paramètres :** le PKG génère un entier RSA N, ainsi qu'un entier e premier avec $\varphi(N)$ et deux fonctions de hachage $H_1 : [0, N-1] \times \{0,1\}^* \to [0, N-2]$ et $H_2 : \{0,1\}^* \to [0, N-1]$. Il rend tous ces paramètres publics (sauf bien sûr la factorisation de N qui constitue la clé maître).
- **Extraction de clé privée :** lorsque l'utilisateur dont l'identité est ID lui demande sa clé privée, le PKG calcule la cé privée correspondante g_{ID} en résolvant l'équation $g_{ID}^e \equiv H_2(ID)$ $[N]$ et la lui transmet.
- **Signature :** pour signer le message M, le possesseur de la clé privée g_{ID} :
  - génère un entier $r \in [0, N-1]$ aléatoire,
  - calcule $t \equiv r^e$ $[N]$,
  - calcule $s \equiv g_{ID} \cdot r^{H_1(t,M)}$ $[N]$.

  La signature est alors le couple $(s,t) \in [0, N-1]^2$.
- **Vérification :** pour vérifier la validité de la signature (s,t) associée au message M (émise par l'usager dont l'identité est ID), il suffit de vérifier que l'on a bien l'égalité

$$s^e \equiv H_2(ID) \cdot t^{H_1(t,M)} \ [N].$$

La consistence de ce protocole se vérifie rapidement : on a

$$s \equiv g_{ID} \cdot r^{H_1(t,M)} \ [N],$$

et donc

$$s^e \equiv g_{ID}^e \cdot (r^e)^{H_1(t,M)} \ [N].$$

Or $g_{ID}^e \equiv H_2(ID)\ [N]$, et $r^e \equiv t\ [N]$, donc l'égalité

$$s^e \equiv H_2(ID) \cdot t^{H_1(t,M)}\ [N].$$

est bien vérifiée.

On notera que, tout comme pour RSA, la sécurité de ce protocole n'est sans doute pas équivalente au problème de factorisation.

A.2 Le protocole de distribution de clé de Maurer et Yacobi

Il s'agit du protocole de distribution de clé non interactif décrit dans [51], dont la sécurité repose sur la difficulté de la factorisation d'un entier $M = p_1 p_2 \dots p_n$ produit de plusieurs facteurs premiers. Ce protocole se compose des algorithmes suivants :

- **Génération des paramètres :** le PKG publie un entier M produit d'entiers premiers distincts $p_1, \dots, p_n$ dont il garde la factorisation secrète (cette factorisation constituant la clé maître). Les p_i doivent être tels que la factorisation de M soit un problème difficile, alors qu'il doit être possible pour le PKG de résoudre des problèmes de logarithme discret dans $\mathbb{F}_{p_i}^\times$. Le PKG détermine alors un générateur g d'un grand sous-groupe cyclique G de $(\mathbb{Z}/M\mathbb{Z})^*$ (de façon à ce que le problème du logarithme discret dans G ne puisse être résolu sans la factorisation de M), ainsi qu'une fonction de hachage $H : \{0,1\}^* \to G$. Ces paramètres sont rendus publics.
- **Extraction de clé privée :** la clé privée correspondant à une identité ID est l'entier $d_{ID} \in [0, \#G - 1]$ tel que $g^{d_{ID}} \equiv H(ID)\ [M]$. La connaissance de la factorisation de M permet au PKG de calculer cette clé privée en utilisant un algorithme de type Pohlig–Hellman. On notera toutefois que ceci se fait en temps sous-exponentiel, mais non polynomial.
- **Création de clé commune :** les utilisateurs ID_A et ID_B partagent alors le secret $g^{f(ID_A) \cdot f(ID_B)}$, et le calcul de ce dernier nécessite soit d_{ID_A}, soit d_{ID_B}.

Ce protocole souffre d'un inconvénient majeur : la phase d'extraction de clé privée requiert une charge importante de travail (en fait, le cassage du protocole et l'extraction de clé privée ont tous deux une complexité sous-exponentielle). De plus, plusieurs types d'attaques sont possibles sur ce type de systèmes (nous renvoyons à [46, 44] pour plus de détails).

On notera que Hühnlein, Jacobson et Weber [38] ont proposé un protocole du même type, mais utilisant des ordres non maximaux dans des corps quadratiques imaginaires. Là encore, la génération de clés privées ne peut se faire en temps polynomial.

A.3 Le protocole de chiffrement de Cocks

Ce protocole, décrit dans [16], est basé sur la difficulté d'extraire une racine carrée modulo un entier N dont on ne connait pas la factorisation en produit de deux nombres premiers p et q.

Notons $\left(\frac{a}{b}\right)$ le symbole de Jacobi (dont on rappelle qu'il peut être évalué efficacement). Lors d'une phase de génération des paramètres du système, le PKG engendre un entier $N = pq$ dont les facteurs premiers p et q sont tous deux congrus à 3 modulo 4. L'entier N est rendu public, ainsi qu'une fonction de hachage $f : \{0,1\}^* \to [0, N-1]$ telle que

$$\forall x \in \mathrm{Im} f, \left(\frac{x}{N}\right) = +1$$

(une telle fonction se construit facilement à partir d'une fonction de hachage classique). La factorisation de N, que le PKG garde secrète, constitue la clé maître.

Si ID désigne l'identité d'un utilisateur, alors $\left(\frac{f(ID)}{N}\right) = +1$, et donc $\left(\frac{f(ID)}{p}\right) = \left(\frac{f(ID)}{q}\right)$. Comme $p \equiv q \equiv 3$ [4], alors soit $f(ID)$, soit $-f(ID)$ est un carré modulo N, et la factorisation de N permet au PKG d'en calculer une racine. On suppose ici que $f(ID)$ est un carré modulo N (l'autre cas se traiterait de manière similaire). La clé privée correspondant à ID calculée par le PKG est un entier r_{ID} tel que $r_{ID}^2 \equiv f(ID)$ $[N]$.

Le protocole de Cocks ne permet de chiffrer des messages que bit par bit : supposons que l'on veuille chiffrer le bit x, codé sous la forme ± 1, avec la clé ID. Il suffit pour cela d'engendrer un entier $t \in [0, N-1]$ aléatoire tel que $\left(\frac{t}{N}\right) = x$. Le chiffré correspondant est alors $s \equiv (t + f(ID)/t)$ $[N]$.

Pour déchiffrer un tel message s, le possesseur de r_{ID} calcule $s + 2r_{ID}$ modulo N. Un calcul rapide montre que

$$s + 2r_{ID} \equiv t \cdot (1 + r_{ID}/t)^2 \ [N],$$

et donc que $\left(\frac{s+2r_{ID}}{N}\right) = \left(\frac{t}{N}\right) = x$.

L'utilité pratique d'un tel système est bien sûr limitée par la taille du chiffré par rapport au clair (le rapport est égal à la taille de N en bits).

Annexe B

Exemples de courbes

B.1 Courbes supersingulières et distortions associées

Caractéristique 3

- Lorsque s est congru à ± 1 ou ± 5 modulo 12, la courbe elliptique définie sur $\mathbb{F}_{3^s}$ par l'équation $y^2 = x^3 + 2x + 1$ a pour cardinal $3^s + 1 + \sqrt{3^{s+1}}$ si $s \equiv \pm 1$ modulo 12 et $3^s + 1 - \sqrt{3^{s+1}}$ si $s \equiv \pm 5$ modulo 12. Une telle courbe a un degré MOV égal à 6. Si u désigne une solution de l'équation $u^2 = -1$ et r une solution de l'équation $r^3 + 2r + 2 = 0$, alors l'application qui à (x, y) associe $(-x + r, uy)$ définit une distortion pour cette courbe.
- Lorsque s est congru à ± 1 ou ± 5 modulo 12, la courbe elliptique définie sur $\mathbb{F}_{3^s}$ par l'équation $y^2 = x^3 + 2x - 1$ a pour cardinal $3^s + 1 - \sqrt{3^{s+1}}$ si $s \equiv \pm 1$ modulo 12 et $3^s + 1 + \sqrt{3^{s+1}}$ si $s \equiv \pm 5$ modulo 12. Une telle courbe a un degré MOV égal à 6. Si u désigne une solution de l'équation $u^2 = -1$ et r une solution de l'équation $r^3 + 2r - 2 = 0$, alors l'application qui à (x, y) associe $(-x + r, uy)$ définit une distortion pour cette courbe.

L'intérêt particulier de ces courbes est qu'elles atteignent le degré MOV maximal possible pour des courbes supersingulières, soit 6. On notera que cela n'est possible qu'en caractéristique 3.

Caractéristique 2

Les courbes définies par l'équation $y^2 + y = x^3 + x + b$ ($b \in \{0, 1\}$) sur un corps $\mathbb{F}_{2^s}$ (s premier) ont pour cardinal $2^s + 1 \pm \sqrt{2^{s+1}}$ (suivant la valeur de s). Leur degré MOV vaut 4, et l'application qui à (x, y) associe $(\alpha^2 x + \beta^2, y + \alpha^2 \beta x + \gamma)$, où α, β et γ sont définis par

$$\begin{gathered}
\alpha^2 + \alpha + 1 = 0, \\
\beta^4 + \beta + \alpha + 1 = 0, \\
\gamma^2 + \gamma + \beta^6 + \beta^2 = 0,
\end{gathered}$$

est une distortion pour ces courbes.

B.2 Courbes ordinaires de faible degré MOV

Les courbes données ont été contruites en utilisant la méthode développée durant ce stage. Les notations des paramètres reprennent celles introduites dans 4.6. L'entier h correspond au nombre de classes associé au discriminant fondamental D (on rappelle que h est alors le degré du polynôme de classes de Hilbert H_D). Les entiers A et B définissent l'équation $y^2 = x^3 + Ax + B$ de la courbe.

r désigne le temps de calcul nécessaire à la construction de la courbe à partir des paramètres a et p en utilisant la technique de la multiplication complexe, sur un Pentium III cadencé à 700 MHz.

Exemples de petite taille :

$$
\begin{aligned}
k &= 5 \\
a &= 26103 \\
D &= 26103 \\
h &= 88 \\
p &= n^2 + a \\
&= 10316095101096156580609884521822230897927 \\
\ell &= 118856368237249643641 \\
A &= 6361774565981298467679675481620482961778 \\
B &= 7679881411019584505323078495021065607161 \\
r &= 1.2 \text{ sec}
\end{aligned}
$$

$$
\begin{aligned}
k &= 7 \\
a &= 1068 \\
D &= 267 \\
h &= 2 \\
p &= n^2 + a \\
&= 22280215019917539692076037201942564656877 \\
\ell &= 209942810985515700149 \\
A &= 20081485727637137786281947313744519173193 \\
B &= 19348575963543670484350584017678504011965 \\
r &= 0.5 \text{ sec}
\end{aligned}
$$

Exemples de taille cryptographique :

$$
\begin{aligned}
k &= 10 \\
a &= 163841^2 \cdot 381535 \\
D &= 381535 \\
h &= 304 \\
p &= n^2 + a \\
&= 38414730593991071701031266252149562435558492305867302065543191924031267584\backslash \\
&\quad 24784619950343423791044836076585229766559410700100854819 \text{ (428 bits)} \\
\ell &= 4686879083953795487935291153103592178053824492905821016357311641 \text{ (212 bits)} \\
A &= 3614578796541747106204758437452623506218014739109496255047150073038238\backslash \\
&\quad 74440660375308333064155960208871834107728173994725817706209 \\
B &= 9779653359898889715032179580552084314015037548925981335085475716478582\backslash \\
&\quad 3429429213794100661750235442419193580537672582267656086793 \\
r &= 57 \text{ sec}
\end{aligned}
$$

$$
\begin{aligned}
k &= 11 \\
a &= 3432987 \\
D &= 13731947 \\
h &= 675 \\
p &= n^2 + n + a \\
&= 1085821608657960459200424901105246469500036293041071392729642052706715552\backslash \\
&\quad 5209414077340531489889487980320059886340361265142418889395568109 \text{ (452 bits)} \\
t &= +2n + 1 \\
\ell &= 31868518802410275890234469142066082346142304768132007825950373986651 \text{ (225 bits)} \\
A &= 1559295546932200357119739705088716590408695933963361975762035466055625821\backslash \\
&\quad 563020387825392942383755862763911883552315027999018090902306395 \\
B &= 9317871453629336870829152280819931917211836532224958585880327639452271445\backslash \\
&\quad 556969901994211583104666470565255768963327547426970047466787266 \\
r &= 190 \text{ sec}
\end{aligned}
$$

Exemple de grande taille :

$$\begin{array}{rcl}
k & = & 50 \\
a & = & 1382037198759 \\
D & = & 100031 \\
h & = & 360 \\
p & = & n^2 + a \\
 & = & 20842920653141790940385053839262236837004280380662095237577453898532012669\backslash \\
 & & 12278767168768885649030339687884195002492197984521031047569738948314071100\backslash \\
 & & 64773342975230329862156271117105741739036924127527019193851292166824743040\backslash \\
 & & 17589899857634542271026193590188892808214449620075170944719236203955726821\backslash \\
 & & 03091008498680792490913071883312366613892911615075996269740260732750552134\backslash \\
 & & 81113489724548452173601824251662512835208268883544848406302169193525823153\backslash \\
 & & 208277049189474278273411115309203458121169283108544784074064572363 \text{ (1698 bits)} \\
t & = & +2n \\
\ell & = & 14210994604898071775164903075969042517171313244513507262727016999091734563\backslash \\
 & & 97101469611511374559527389145563061943309339665717829765588496498158653799\backslash \\
 & & 516207322655133273742182777107000404838689258946218747177224943815520597387\backslash \\
 & & 891445377017922059001106955159901 \text{ (849 bits)} \\
A & = & 15156186447228839987528411535043949413637399436346140307909666013896538378\backslash \\
 & & 336381991546850978315929395699119111596361095171494737713670738673420839041\backslash \\
 & & 98380464187218562787156173322534497420510729291725211357401488923884852083\backslash \\
 & & 92637690832825732907154833795541123795821108492581148142533859414554476338\backslash \\
 & & 06424510203094405400342839682820962332173559508836191138073638242725941450\backslash \\
 & & 55370937038111692041774266870745665982011383137805845832451594037089761358\backslash \\
 & & 579795589755763993889692815279314697423641993748584778749607071723 \\
B & = & 63129681608772593564478461538837746601803456613534569188279187528407093916\backslash \\
 & & 99984207604008733427703596346264187934866615723122962531811589323517379888\backslash \\
 & & 79917235994711971414373836853088354013230229706156023473011237872966407517\backslash \\
 & & 84569878720112823621889826672622365222851782433914168935656550834354839033\backslash \\
 & & 98386746050053455398484049882197053669694716017309240042713438351342205108\backslash \\
 & & 64189229011166212732978063265525460858763315947112288390673459204357997093\backslash \\
 & & 00876086881369139670649676832950624484076469592416515616766380722 \\
r & = & 1500 \text{ sec}
\end{array}$$

Ce dernier exemple illustre ce qui survient lorsque $\varphi(k)$ est trop élevé : $R_k(a)$ croît tellement vite qu'il devient rapidement impossible d'en déterminer un facteur dont on puisse contrôler la taille. Alors que l'on désirait ici ℓ proche de 200 bits, on doit se contenter d'une taille qui est plus de quatre fois supérieure.

Bibliographie

[1] L. M. Adleman and J. DeMarrais. A subexponential algorithm for discrete logaritmhs over all finite fields. *Math. Comp.*, 61(203):1–15, July 1993.

[2] L. M. Adleman, R. L. Rivest, and A. Shamir. A method for obtaining digital signatures and public-key cryptosystems. *Comm. ACM*, 21(2):120–126, 1978.

[3] S. Al-Riyami and K. Paterson. Authenticated three party key agreement protocols from pairings, 2002. Available at `http://eprint.iacr.org/2002/035/`.

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

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

[6] P. Barreto, H. Kim, B. Lynn, and M. Scott. Efficient algorithms for pairing-based cryptosystems, 2002. Available at `http://eprint.iacr.org/2002/008`.

[7] P. Barreto, B. Lynn, and M. Scott. Constructing elliptic curves with prescribed embedding degree, 2002. Available at `http://eprint.iacr.org/2002/088/`.

[8] I. Blake, G. Seroussi, and N. Smart. *Elliptic curves in cryptography*, volume 265 of *London Math. Soc. Lecture Note Ser.* Cambridge University Press, 1999.

[9] A. Boldyreva. Efficient threshold signature, multisignature and blind signature schemes based on the Gap–Diffie–Hellman group signature scheme, 2002. Available at `http://eprint.iacr.org/2002/118/`.

[10] D. Boneh and M. Franklin. Identity-based encryption from the Weil pairing. In J. Kilian, editor, *Advances in Cryptology – CRYPTO 2001*, volume 2139 of *Lecture Notes in Comput. Sci.*, pages 213–229. Springer-Verlag, 2001.

[11] D. Boneh, B. Lynn, and H. Shacham. Short signatures from the Weil pairing. In C. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 514–532. Springer-Verlag, 2001.

[12] D. Boneh and A. Silverberg. Applications of multilinear forms to cryptography, 2002. Available at `http://eprint.iacr.org/2002/080/`.

[13] C. Castellucia. How to convert any ID-based signature schemes, 2002. Available at `http://eprint.iacr.org/2002/116/`.

[14] J. Cha and J. Cheon. An identity-based signature form Gap–Diffie–Hellman groups, 2002. Available at `http://eprint.iacr.org/2002/018/`.

[15] J. Cheon. A universal forgery of Hess's second ID-based signature against the known-message attack, 2002. Available at `http://eprint.iacr.org/2002/028/`.

[16] C. Cocks. An identity based encryption scheme based on quadratic residues. In B. Honary, editor, *Cryptography and Coding*, volume 2260 of *Lecture Notes in Comput. Sci.*, pages 360–363. Springer-Verlag, 2001. 8th IMA International Conference, Cirencester, UK, 2001. Proceedings.

[17] Computational Algebra Group of the University of Sydney. MAGMA version 2.9, 2001. `http://magma.maths.usyd.edu.au/magma/`.

[18] D. A. Cox. *Primes of the form* $x^2 + ny^2$. John Wiley & Sons, 1989.

[19] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Trans. Inform. Theory*, IT–22(6), November 1976.

[20] R. Dupont and A. Enge. Practical and secure non-interactive key distribution based on pairings, 2002. Available at `http://eprint.iacr.org/2002/136/`.

[21] R. Dupont, A. Enge, and F. Morain. Building curves with arbitrary small mov degree over finite prime fields, 2002. Available at `http://eprint.iacr.org/2002/094/`.

[22] T. ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. *IEEE Trans. Inform. Theory*, IT–31(4):469–472, July 1985.

[23] A. Enge. *Elliptic Curves and Their Applications to Cryptography — An Introduction.* Kluwer Academic Publishers, 1999.

[24] A. Enge and P. Zimmermann. MPC – Multiprecision Complex arithmetic library version 0.1, 2002. Available at `http://www.loria.fr/~zimmerma/free/`.

[25] Andreas Enge and François Morain. Further investigations of the generalised Weber functions. In preparation, 2001.

[26] Andreas Enge and François Morain. Comparing invariants for class fields of imaginary quadratic fields. In Claus Fieker and David R. Kohel, editors, *Algorithmic Number Theory — ANTS-V*, volume 2369 of *Lecture Notes in Computer Science*, pages 252–266, Berlin, 2002. Springer-Verlag.

[27] Andreas Enge and Reinhard Schertz. Constructing elliptic curves from modular curves of positive genus. Preprint, 2001.

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

[29] E. Fujisaki and T. Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In M. Wiener, editor, *Advances in Cryptology – CRYPTO'99*, volume 1666 of *Lecture Notes in Comput. Sci.*, pages 537–554. Springer-Verlag, 1999. 19th Annual International Cryptology Conference, Santa Barbara, California, USA, August 1999. Proceedings.

[30] S. Galbraith. Supersingular curves in cryptography. In C. Boyd, editor, *Advances in Cryptology – ASIACRYPT 2001*, volume 2248 of *Lecture Notes in Comput. Sci.*, pages 495–513. Springer-Verlag, 2002. 7th International Conference on the Theory and Applications of Cryptology and Information Security, Gold Coast, Australia, December 9-13, 2001. Proceedings.

[31] S. Galbraith, K. Harrison, and D. Soldera. Implementing the Tate pairing. In *Algorithmic Number Theory Symposium – ANTS V*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 495–513, 2002.

[32] C. Gentry and A. Silverberg. Hierarchical ID-based cryptography, 2002. Available at `http://eprint.iacr.org/2002/056/`.

[33] T. Granlund *et al.* GMP – GNU Multiprecision library version 4.1, 2002. Available at `http://www.swox.com/gmp/`.

[34] G. Hanrot, G. Lefèvre, P. Zimmermann, et al. MPFR – Multiprecision Floating point library with exact Rounding version 2.0.1, 2001. Available at `http://www.mpfr.org/`.

[35] Helmut Hasse. Abstrakte Begründung der komplexen Multiplikation und Riemannsche Vermutung in Funktionenkörpern. *Abhandlungen aus dem mathematischen Seminar der hamburgischen Universität*, 10:325–348, 1934.

[36] F. Hess. Exponent group signature schemes and efficient identity based signature schemes based on pairings. Cryptology ePrint Archive, Report 2002/012, available at `http://eprint.iacr.org/2002/012/`.

[37] J. Horwitz and B. Lynn. Towards hierarchical identity-based encryption. In L. Knudsen, editor, *Advances in Cryptology – EUROCRYPT 2002*, volume 2332 of *Lecture Notes in Comput. Sci.*, pages 466–481. Springer-Verlag, 2002. International Conference on the Theory and Applications of Cryptographic Techniques, Amsterdam, The Netherlands, April 28 – May 2, 2002. Proceedings.

[38] D. Hühnlein, M. J. Jacobson Jr., and D. Weber. Towards practical non-interactive public-key cryptosystems using non-maximal imaginary quadratic orders. In D. R. Stinson and S. Tavares, editors, *Selected Areas in Cryptography 2000*, volume 2012 of *Lecture Notes in Comput. Sci.*, pages 275–287. Springer-Verlag, 2000. 7th Annual International Workshop, SAC 2000, Waterloo, Ontario, Canada, August 14-15, 2000. Proceedings.

[39] A. Joux. A one round protocol for tripartite Diffie-Hellman. In W. Bosma, editor, *ANTS-IV*, volume 1838 of *Lecture Notes in Comput. Sci.*, pages 358–394. Springer-Verlag, 2000. 4th International Symposium, ANTS-IV, Leiden, The Netherlands, July 2-7, 2000. Proceedings.

[40] A. Joux. The Weil and Tate pairings as building blocks for public key cryptosystems. In C. Fieker and D. Kohel, editors, *ANTS-V*, volume 2369 of *Lecture Notes in Comput. Sci.*, pages 20–32. Springer-Verlag, 2002. 5th International Symposium, ANTS-V, Sydney, Australia, July7-12, 2002. Proceedings.

[41] A. Joux and K. Nguyen. Separating Decision Diffie–Hellman from Diffie–Hellman in cryptographic groups, 2001. Available at `http://eprint.iacr.org/2001/003/`.

[42] N. Koblitz. Elliptic curve cryptosystems. *Math. Comp.*, 48(177):203–209, January 1987.

[43] D. Kügler. Eine Aufwandsanalyse für identitätsbasierte Kryptosyteme. Master's thesis, Technische Universität Darmstadt, Deutschland, 1998. Available at `ftp://ftp.informatik.tu-darmstadt.de/pub/TI/reports/kuegler.IDCS.diplom.ps.gz`.

[44] D. Kügler and M. Maurer. A note on the weakness of the Maurer–Yacobi squaring method. Technical Report TI-15/99, Fachbereich Informatik, Technische Universität Darmstadt, 1999. Available at `ftp://ftp.informatik.tu-darmstadt.de/pub/TI/TR/-TI-99-15.weaksquaring.ps.gz`.

[45] L. Law, A. Menezes, M. Qu, J. Solinas, and S. Vanstone. An efficient protocol for authenticated key agreement. Technical Report CORR 98–05, Department of C& O, University of Waterloo, March 1998.

[46] P. J. Lee and C. H. Lim. Modified Maurer-Yacobi's scheme and its applications. In J. Seberry and Y. Zheng, editors, *Advances in Cryptology – AUSCRYPT'92*, volume 718 of *Lecture Notes in Comput. Sci.*, pages 308–323. Springer-Verlag, 1992. Workshop on the Theory and Application of Cryptographic Techniques, Gold Coast, Queensland, Australia, December 13-16, 1992. Proceedings.

[47] A. K. Lenstra and E. R. Verheul. Selecting cryptographic key sizes. *J. of Cryptology*, 14(4):255–293, September 2001.

[48] B. Lynn. Authenticated identity-based encryption, 2002. Available at `http://eprint.iacr.org/2002/072/`.

[49] J. Malone-Lee. Identity-based signcryption, 2002. Available at `http://eprint.iacr.org/2002/098/`.

[50] U. Maurer and S. Wolf. The relationship between breaking the diffie–hellman protocol and computing discrete logarithms. *SIAM J. Comput.*, 28(5):1689–1721, 1999.

[51] U. Maurer and Y. Yacobi. Non-interactive public-key cryptography. In D. Davies, editor, *Advances in Cryptology – EUROCRYPT '91*, volume 547 of *Lecture Notes in Comput. Sci.*, pages 498–507. Springer–Verlag, 1992. Proceedings of the Workshop on the Theory and Application of Cryptographic Techniques, Brighton, United Kingdom, April 8–11, 1991.

[52] U. Maurer and Y. Yacobi. A non-interactive public-key distribution system. *Des. Codes Cryptogr.*, 9(3):305–316, 1996.

[53] A. Menezes, T. Okamoto, and S. A. Vanstone. Reducing elliptic curves logarithms to logarithms in a finite field. *IEEE Trans. Inform. Theory*, IT–39(5):1639–1646, September 1993.

[54] A. J. Menezes. *Elliptic curve public key cryptosystems.* Kluwer Academic Publishers, 1993.

[55] V. Miller. Short programs for functions on curves. Draft, 1986.

[56] V. Miller. Use of elliptic curves in cryptography. In A. M. Odlyzko, editor, *Advances in Cryptology – CRYPTO '86*, volume 263 of *Lecture Notes in Comput. Sci.*, pages 417–426. Springer-Verlag, 1987. Proceedings, Santa Barbara (USA), August 11–15, 1986.

[57] A. Miyaji, M. Nakabayashi, and S. Takano. New explicit conditions for elliptic curve traces for FR-reduction. *IEICE Trans. Fundamentals*, E84–A(5):1234–1243, 2001.

[58] K. Paterson. ID-based signatures form pairings on elliptic curves, 2002. Available at `http://eprint.iacr.org/2002/004/`.

[59] R. Sakai, K. Ohgishi, and M. Kasahara. Cryptosystems based on pairing. SCIS 2000, The 2000 Symposium on Cryptography and Information Security, Okinawa, Japan, January 26–28.

[60] I. Semaev. Computation of discrete logarithms in an arbitrary finite field. *Discrete Math. Appl.*, 5(2):107–116, 1995.

[61] A. Shamir. How to share a secret. *Comm. ACM*, 22:612–613, 1979.

[62] A. Shamir. Identity-based cryptosystems and signature schemes. In G. Goos and J. Hartmanis, editors, *Advances in Cryptology – CRYPTO'84*, volume 196 of *Lecture Notes in Comput. Sci.*, pages 47–53. Springer-Verlag, 1985. 4th Annual International Cryptography Conference, Santa Barbara, Ca, USA, 19-22 August 1984. Proceedings.

[63] V. Shoup. NTL – Number Theory Library 5.2, 2001. Available at `http://shoup.net/ntl/`.

[64] J. H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Grad. Texts in Math.* Springer, 1986.

[65] N. Smart. An identity-based authenticated key agreement protocol based on the Weil pairing, 2001. Available at `http://eprint.iacr.org/2001/111/`.

[66] J. von zur Gathen and J. Gerhard. *Modern Computer Albebra.* Cambridge University Press, 1999.

[67] A. Weil. Number of solutions of equations in finite fields. *Bull. AMS*, 55:497–508, 1949.

[68] F. Zhang, S. Liu, and K. Kim. ID-based one round authenticated tripartite key agreement protocol with pairings, 2002. Available at `http://eprint.iacr.org/2002/122/`.

Illustration de couverture : un groupe cyclique de points de 6-torsion à coordonnées réelles sur la courbe elliptique
$$y^2 = 2x^3 - x^2 - 2x + 1.$$