Théorie des nombres et cryptographie

David Kohel et Igor Shparlinski
Aix-Marseille Université

20 juin 2014

Théorie des nombres et cryptographie

La théorie des nombres est un domaine des mathématiques qui depuis des siècles a été perçu comme une discipline pure sans aucune interaction avec le monde réel. Dans *A Mathematician's Apology* (1940), Hardy a écrit :

I have never done anything 'useful'. No discovery of mine has made, or is likely to make, directly or indirectly, for good or ill, the least difference to the amenity of the world.

Néanmoins, ce domaine a emergé au milieu du XX^e siècle au coeur de la recherche en technologie de l'information, en particulier, jouant un rôle important dans la théorie des codes et la cryptographie.

Parmi les applications en cryptographie, certaines parmi les plus intéressantes et les plus novatrices résident en cryptographie à clef publique. Après des rappels du contexte général cryptographique, nous nous concentrons ici sur la cryptographie à clef publique.

Théorie des nombres et cryptographie

La théorie des nombres est un domaine des mathématiques qui depuis des siècles a été perçu comme une discipline pure sans aucune interaction avec le monde réel. Dans *A Mathematician's Apology* (1940), Hardy a écrit :

Je n'ai jamais rien fait d'«utile». Aucune de mes découvertes n'a fait, ou n'est susceptible de faire, directement ou indirectement, en bien ou en mal, la moindre différence aux conforts du monde.

Néanmoins, ce domaine a emergé au milieu du XX^e siècle au coeur de la recherche en technologie de l'information, en particulier, jouant un rôle important dans la théorie des codes et la cryptographie.

Parmi les applications en cryptographie, certaines parmi les plus intéressantes et les plus novatrices résident en cryptographie à clef publique. Après des rappels du contexte général cryptographique, nous nous concentrons ici sur la cryptographie à clef publique.

Rappels de la cryptographie

La construction de base de la cryptographie est un *cryptosystème symétrique* — un ensemble d'applications compatibles

$$\mathcal{E} : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C} \text{ et } \mathcal{D} : \mathcal{K} \times \mathcal{C} \longrightarrow \mathcal{M}$$

tel que $\mathcal{D}(K, \mathcal{E}(K, M)) = M$ pour tout $K \in \mathcal{K}$ et $M \in \mathcal{M}$, où on a

$\mathcal{K}$: l'espace des clefs,
$\mathcal{M}$: l'espace des messages ou textes clairs,
$\mathcal{C}$: l'espace des textes chiffrés.

Une seule *clef secrète* K sert à la fois à déterminer un *chiffrement* et un *déchiffrement*

$$E_K = \mathcal{E}(K, \cdot) : \mathcal{M} \to \mathcal{C},$$
$$D_K = \mathcal{D}(K, \cdot) : \mathcal{C} \to \mathcal{M}.$$

Par conséquent, les protagonistes doivent d'abord avoir accès à un canal sécurisé pour établir la clef secrète commune dont ils se serviront pour communiquer plus tard.

Une nouvelle direction en cryptographie

En 1976, Diffie et Hellman ont introduit les fondements de la cryptographie à clef publique, en remplaçant la clef secrète unique par une paire (K, K') de clefs — une *clef publique* pour le chiffrement et une *clef privée* pour le déchiffrement. Dans ce schéma, on a

$$\mathcal{E} : \mathcal{K} \times \mathcal{M} \longrightarrow \mathcal{C} \text{ et } \mathcal{D} : \mathcal{K}' \times \mathcal{C} \longrightarrow \mathcal{M}$$

tel que $\mathcal{D}(K', \mathcal{E}(K, M)) = M$ pour tout $M \in \mathcal{M}$. La connaissance de K' doit être difficile à deduire par la seule connaissance de la clef publique K, et donc tout le monde peut appliquer le chiffrement

$$E_K = \mathcal{E}(K, \cdot) : \mathcal{M} \to \mathcal{C},$$

et seulement le propriétaire de la clef privée K' peut déchiffrer avec

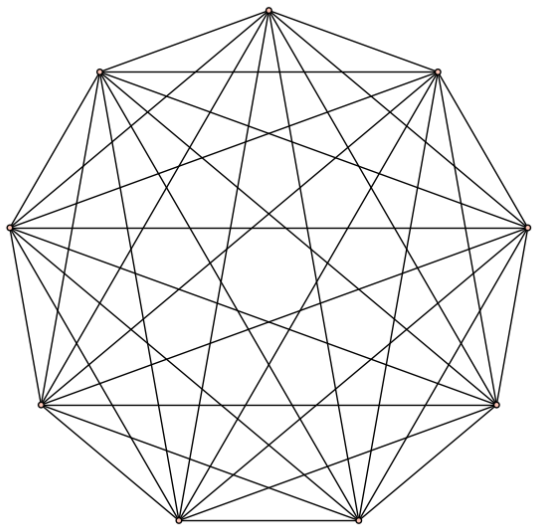
$$D_{K'} = \mathcal{D}(K', \cdot) : \mathcal{C} \to \mathcal{M}.$$

Rôle des cryptosystèmes symétriques

Les cryptosytèmes symétriques sont sûrs et efficaces, mais ont besoin d'un canal sécurisé pour établir une clef secrète commune.

Le canal sécurisé prend typiquement la forme d'une réunion face-à-face, mais peut être un courrier fiable qui transporte les clefs entre les protagonistes de la communication.

En plus, pour des communications eventuelles entre n entités, il est necessaire d'établir $n(n-1)/2$ clefs secrètes, avec un renouvellement regulier.

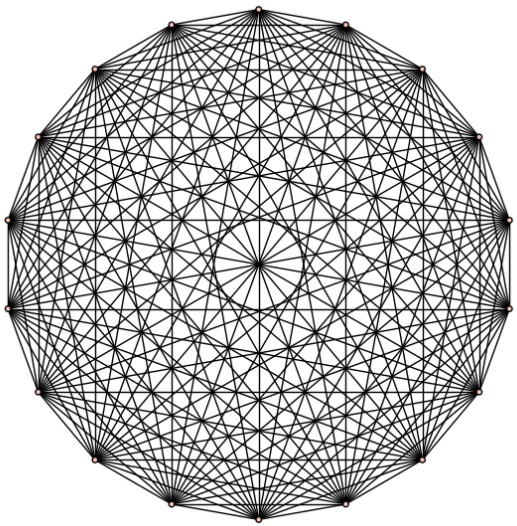


Rôle des cryptosystèmes symétriques

Les cryptosytèmes symétriques sont sûrs et efficaces, mais ont besoin d'un canal sécurisé pour établir une clef secrète commune.

Le canal sécurisé prend typiquement la forme d'une réunion face-à-face, mais peut être un courrier fiable qui transporte les clefs entre les protagonistes de la communication.

En plus, pour des communications eventuelles entre n entités, il est necessaire d'établir $n(n-1)/2$ clefs secrètes, avec un renouvellement regulier.

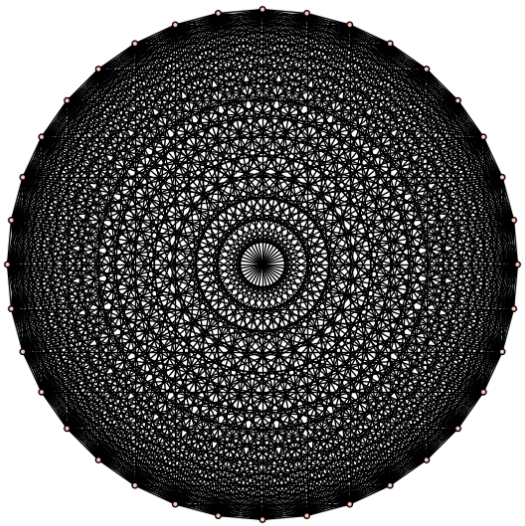


Rôle des cryptosystèmes symétriques

Les cryptosytèmes symétriques sont sûrs et efficaces, mais ont besoin d'un canal sécurisé pour établir une clef secrète commune.

Le canal sécurisé prend typiquement la forme d'une réunion face-à-face, mais peut être un courrier fiable qui transporte les clefs entre les protagonistes de la communication.

En plus, pour des communications eventuelles entre n entités, il est necessaire d'établir $n(n-1)/2$ clefs secrètes, avec un renouvellement regulier.



Rôle des cryptosystèmes à clef publique

Les cryptosystèmes à clef publique ont besoin de clefs de taille beaucoup plus important et sont plus lents que leurs analogues symétriques. Néanmoins, ces cryptosystèmes s'en sert d'un canal complètement public (non-sécurisé), soit pour établir une clef secrète, soit pour un envoyer un message.

Pour communication entre n entités, il suffit d'établir n paires de clefs publiques-privées. Cela resoudre le problème de distribution des clefs secrètes. Néanmoins il introduit un besoin de l'authentification et gestion des clefs.

Grâce à leur souplesse et leur adaptablilité à des nouvelles tâches comme la signature numérique, le vote électronique, la preuve de connaissance, etc., ces cryptosystèmes se trouvent partout.

Protocole de Diffie et Hellman

Avec l'introduction des concepts de cryptographie à clef publique Diffie et Hellman ont introduit le premier schéma permettant à deux protagonistes de créer un secret commun en utilisant seulement un canal de communication ouvert. Leur idée simple et élégante s'appelle actuellement le protocole d'échange de clef de Diffie-Hellman.

On décrit ce protocole entre deux protagonistes A et B. D'abord, ils choisissent un grand nombre premier p ($\approx 2^n$, pour n entre 512 et 1024). Le premier est "public", dans le sens que l'on permet à n'importe qui de découvrir le choix de p. Puis ils utilisent p pour définir un corps fini $\mathbb{F}_p\,(=\mathbb{Z}/p\mathbb{Z})$ de p éléments. Dans le groupe $\mathbb{F}_p^*$ des unités, ils choisissent un élément primitif, c'est-à-dire un générateur g tel que

$$\mathbb{F}_p^* = \{g^0, g^1, \ldots, g^{p-2}\},$$

en utilisant le fait que ce groupe est cyclique.

Protocole de Diffie et Hellman

Après le choix *publique* de $(\mathbb{F}_p^*, g)$, les interlocuteurs A et B choisissent des éléments *privés* k et ℓ dans $\mathbb{Z}/(p-1)\mathbb{Z} = \{0, 1, \ldots, p-2\}$ et calculent g^k et g^ℓ dans $\mathbb{F}_p^*$, respectivement. Après l'échange des valeurs g^k et g^ℓ sur le canal non-sécurisé, chaque protagoniste peut facilement calculer

$$\left(g^k\right)^\ell = \left(g^\ell\right)^k = g^{k\ell},$$

qui servira comme clef secrète commune pour un cryptosystème symétrique au choix.

Clairement, ce protocole ne fournit pas de protection absolue car un adversaire peut toujours trouver k, étant donnés g et $h = g^k$. L'entier k s'appelle le logarithme discret de h en base g, et on écrit $k = \log_g h$. Le problème de trouver k, étant donnés g et h dans $\mathbb{F}_p^*$, est le fameux *problème du logarithme discret*.

Le logarithme discret

Une approche naïve – mais déterministe – du problème du logarithme discret est simplement de calculer la suite

$$1 = g^0, g^1, g^2, \ldots,$$

jusqu'à l'obtention de l'identité $g^i = h$, rendant $i = k\,(= \log_g h)$.

La connaissance de k laisse la clef publique sans sécurité. Néanmoins, pour p grand, cette algorithme est computationnellement infaisable. Ainsi, une valeur de p autour de 2^{120} fournit déjà une protection suffisante. Comme nous l'avons remarqué, des valeurs de p beaucoup plus grandes sont recommandées — cela parce qu'il y a des attaques plus efficaces, en particulier la classe d'algorithmes dits de *calcul d'indices*, dont le *crible de corps des nombres* et le plus efficace de cette classe.

Le comportement de ces algorithmes est basé sur nos connaissances et des heuristiques sur l'arithmétique des entiers.

RSA

Dans le protocole de Diffie et Hellman, aucune information qui porte un message n'a été transmise entre A et B, et donc il doit être completé par un cryptosystème symétrique traditionnel.

Le cryptosystème de RSA, de Rivest, Shamir et Adleman, est le premier schéma permettant l'échange de message sans le besoin d'un cryptosystème symétrique.

Soit $N = pq$, pour nombres premiers p et q. Pour un entier a tel que $\mathrm{pgcd}(a, N) = 1$ on

$$a^{\varphi(N)} \equiv 1 \bmod N$$

par le théorème d'Euler, où φ est l'indicatrice d'Euler (donc en particulier $\varphi(N) = (p-1)(q-1)$).

RSA chiffrement

Si A veut recevoir un message privé, il a besoin d'établir un couple de clefs publique–privée comme suit :

- Choisir deux nombres premiers p et q et prendre $N = pq$.
- Choisir un exposant e avec $\gcd(e, \varphi(N)) = 1$.
- Calculer l'exposant d tel que $ed \equiv 1 \bmod \varphi(N)$.

Puis A publie (N, e) comme sa clef publique, et garde d comme clef privée, en se débarrassant de p, q et $\varphi(N)$ de manière sûre.

Pour envoyer un message m de manière sûre à A, en supposant que m est codé par un élément de $\mathbb{Z}/N\mathbb{Z}$, l'expéditeur B calcule et envoie le *texte chiffré* :

$$c \equiv m^e \bmod N.$$

Maintenant, pour déchiffrer, A calcule

$$c^d \equiv (m^e)^d \equiv m^{ed} \equiv m \bmod N.$$

Cette dernière égalité vient du fait que $ed \equiv 1 \bmod \varphi(N)$.

Le cryptosystème d'ElGamal

Le cryptosystème d'ElGamal est une extension du protocole de Diffie et Hellman qui permit l'échange de messages.

Soit p un nombre premier p, avec grand diviseur premier q de $p-1$, et un générateur g du sous-groupe cyclique G d'order q de $\mathbb{F}_p^*$.

Un destinataire A choisit un élément k au hasard dans l'anneau résiduel $\mathbb{Z}/q\mathbb{Z}$ et calcule $h = g^k$ dans $\mathbb{F}_p^*$. Il publie la *clef publique* (p, q, g, h), et retient k pour sa *clef privée*.

Pour *chiffrer* un message m de $\mathbb{F}_p$, l'expéditeur B choisit un ℓ au hasard dans $\mathbb{Z}/q\mathbb{Z}$, calcule la paire

$$(r, s) = (g^\ell, mh^\ell) \in \mathbb{F}_p^* \times \mathbb{F}_p,$$

qu'il envoie à A.

Pour *déchiffrer* le texte chiffré (r, s), A calcule

$$r^k = g^{k\ell} = h^\ell \text{ et puis } s(r^k)^{-1} = s(h^\ell)^{-1} = m.$$

Le cryptosystème d'ElGamal

Donc pour *chiffrer* un message m de $\mathbb{F}_p$, on calcule la paire

$$(r, s) = (g^\ell, mh^\ell) \in \mathbb{F}_p^* \times \mathbb{F}_p,$$

pour un ℓ au hasard dans $\mathbb{Z}/q\mathbb{Z}$, et le destinataire A *déchiffre* (r, s) par

$$m = s(r^k)^{-1} \,[= s(g^{k\ell})^{-1} = s(h^\ell)^{-1}]$$

avec sa clef secrète k.

Remarques :

1. Le quantités g^k et g^ℓ sont exactement les valeurs échangés dans le protocole de Diffie et Hellman, et la securité se repose sur le même problème sous-jacent.

2. Le choix aléatoire de ℓ associe des textes chiffrés multiples à un message donné.

3. L'avantage majeur de ce cryptosystème sur RSA est qu'il se généralise facilement à un groupe abélien arbitraire, en particulier, aux courbes elliptiques.

Cryptosystèmes à sac à dos

L'idée de baser un cryptosystème sur le problème du *sac à dos* a été introduite par Merkle et Hellman en 1978. D'abord, on rappelle ce problème :

Étant donnés n objets de tailles $a_1, \ldots, a_n$ et un "sac à dos" de capacité A, trouver une sélection $I \subseteq \{1, \ldots, n\}$ d'objets qui tiennent juste dans le sac à dos, c'est-à-dire

$$\sum_{i \in I} a_i = A,$$

ou démontrer qu'aucun tel choix n'existe.

Ce problème se décrit de manière équivalente comme le problème de trouver un vecteur entier $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$, avec x_i dans $\{0, 1\}$, tel que

$$(a_1, \ldots, a_n) \cdot (x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i = A.$$

Cryptosystèmes à base de réseaux

Si le vecteur $(x_1, \ldots, x_n)$ in $\mathbb{Z}^n$, avec x_i dans $\{0, 1\}$, satisfait

$$(a_1, \ldots, a_n) \cdot (x_1, \ldots, x_n) = \sum_{i=1}^{n} a_i x_i = A,$$

en mettant $a_{n+1} = -A$ et $x_{n+1} = 1$, on peut exprimer cette dernière équation comme

$$(a_1, \ldots, a_{n+1}) \cdot (x_1, \ldots, x_{n+1}) = \sum_{i=1}^{n+1} a_i x_i = 0.$$

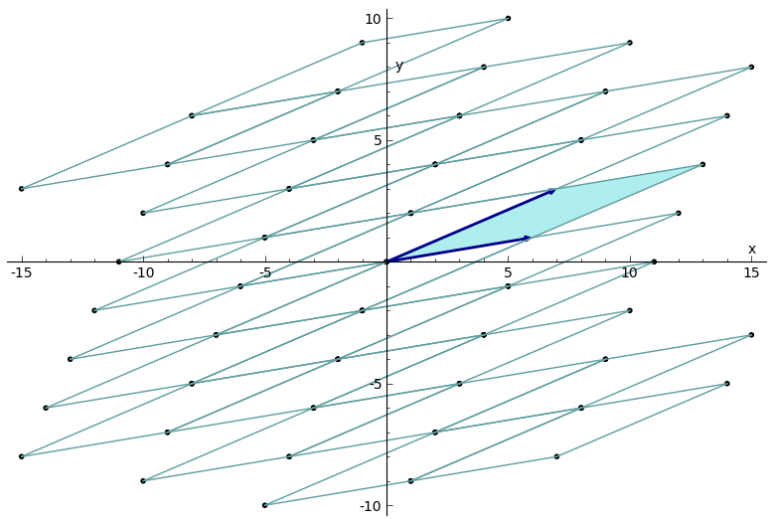
On reconnait le problème du sac à dos comme celui de trouver un vecteur court (de $\mathbb{Z}^{n+1}$), dans le complément orthogonal $L \subset \mathbb{Z}^{n+1}$ de $(a_1, \ldots, a_{n+1})$ engendré sur $\mathbb{R}$ par

$$\{v_1 = (-a_2, a_1, 0, \ldots, 0), \ldots, v_n = (0, \ldots, 0, -a_{n+1}, a_n)\}.$$

Ainsi on place le cryptosystème “sac à dos” dans le contexte plus général des cryptosystèmes à base de réseaux.

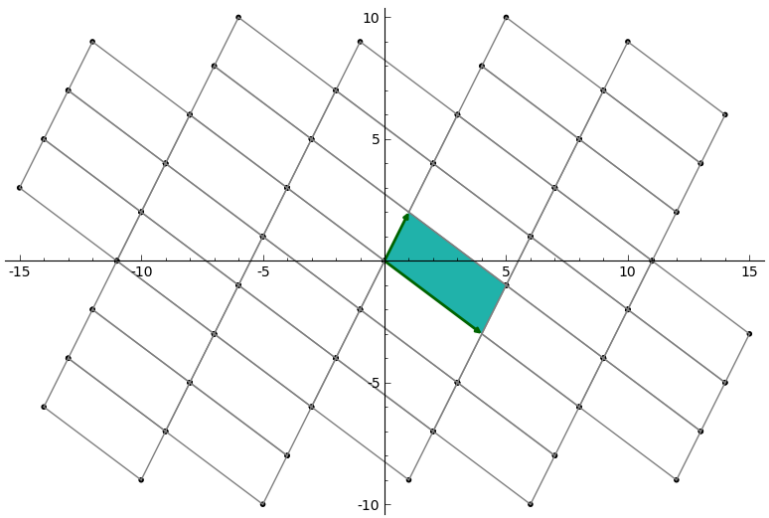
Cryptosystèmes à base de réseaux

L'idée très simple de Merkle et Hellman permet un chiffrement et déchiffrement très rapide. Malheureusement, ce schéma et toutes les extensions semblables sont cassées par l'application du fameux algorithme LLL de Lenstra, Lenstra et Lovàsz, qui est très efficace pour trouver un vecteur court $(x_1, \ldots, x_{n+1})$ dans un réseau L.



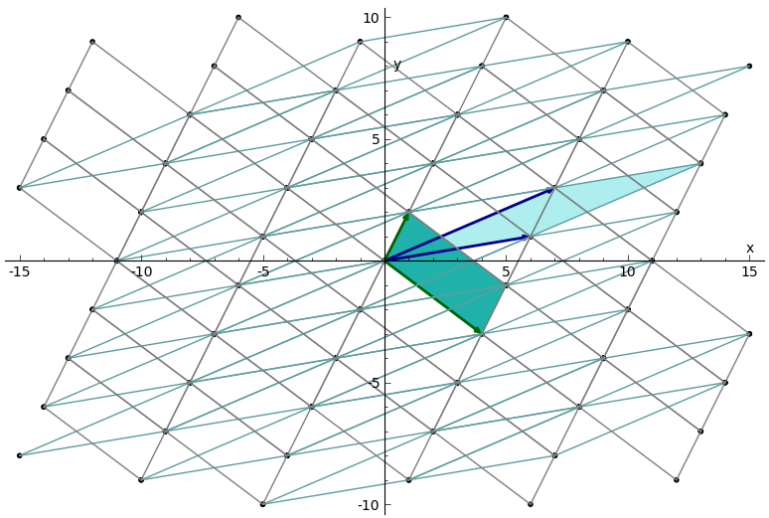
Cryptosystèmes à base de réseaux

L'idée très simple de Merkle et Hellman permet un chiffrement et déchiffrement très rapide. Malheureusement, ce schéma et toutes les extensions semblables sont cassées par l'application du fameux algorithme LLL de Lenstra, Lenstra et Lovàsz, qui est très efficace pour trouver un vecteur court $(x_1, \ldots, x_{n+1})$ dans un réseau L.



Cryptosystèmes à base de réseaux

L'idée très simple de Merkle et Hellman permet un chiffrement et déchiffrement très rapide. Malheureusement, ce schéma et toutes les extensions semblables sont cassées par l'application du fameux algorithme LLL de Lenstra, Lenstra et Lovàsz, qui est très efficace pour trouver un vecteur court $(x_1, \ldots, x_{n+1})$ dans un réseau L.



Cryptosystèmes à base de réseaux

À la suite de plusieurs tentatives, sans réussite, de contruire un cryptosystème fiable autour de problèmes difficiles dans un réseau, cette direction a été abandonnée pendant un certain temps.

En 1997, des percées théoriques de Ajtai et Dwork, et de Goldreich, Goldwasser et Halevi ont relancé l'interêt pour la cryptographie à base de réseaux. En même temps le cryptosystème NTRU de Hoffstein, Pipher and Silverman s'est établi, après une série de modifications et de réglages du schéma original, comme un cryptosystème efficace et sûr.

Plus récemment une nouvelle direction prometteuse en cryptographie à base de réseaux, appellée *chiffrement homomorphique*, est en pleine croissance.

Cryptographie à base de courbes elliptiques

Les protocoles de Diffie-Hellman et ElGamal peuvent être adaptés à n'importe quel groupe abélien, et non seulement à des sous-groupes de $\mathbb{F}_p^*$. Ainsi, en 1986, Koblitz et Miller ont proposé l'utilisation du groupe des points d'une courbe elliptique sur un corps fini.

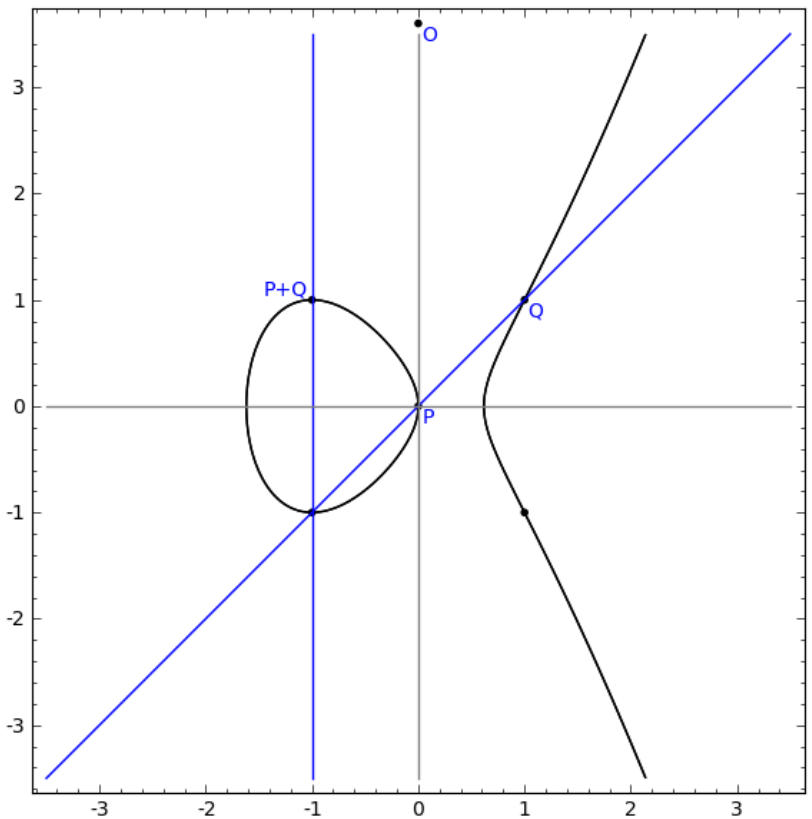
On rappelle qu'une courbe elliptique E sur un corps K (de caractéristique autre que 2 et 3) peut être donnée par une courbe plane

$$y^2 = x^3 + ax + b,$$

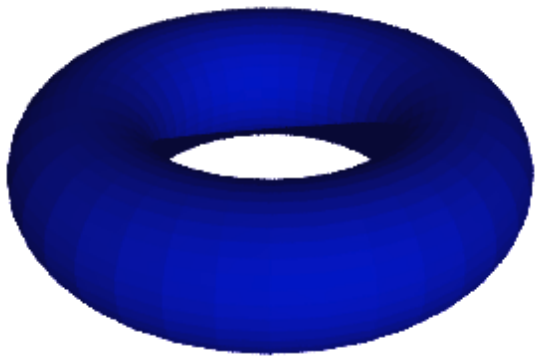
au lieu de la forme générale $y^2 + (a_1x + a_3)y = x^3 + a_2x^2 + a_4x + a_6$.

L'ensemble de points $E(K)$ sur K admet un groupe, avec un point distingué O à l'infini ($O = (0 : 1 : 0) \in \mathbb{P}^2$), défini par la régle que trois points colinéaires somme à O.

Cryptographie à base de courbes elliptiques



$\cong_{\mathbb{C}}$



$$E : y^2 = x^3 + x^2 - x$$

Cryptographie à base de courbes elliptiques

Pour mettre en oeuvre les protocoles de Diffie-Hellman et ElGamal, il est necessaire de déterminer l'ordre du groupe des points.

En 1985, Schoof a publié un algorithme, avec complexité polynômiale, pour déterminer cet ordre. Il a utilisé la représentation galoisienne

$$\rho : \mathrm{Gal}(\overline{\mathbb{F}}_p/\mathbb{F}_p) \longrightarrow \mathrm{Aut}(E[\ell]) \cong \mathrm{GL}_2(\mathbb{F}_\ell),$$

sur les points de ℓ-torsion ($\ell \neq p$ un nombre premier)

$$E[\ell] = \{\, P \in E(\overline{\mathbb{F}}_p) \mid \ell P = O \,\} \cong \mathbb{F}_\ell^2.$$

Si $|E(\mathbb{F}_p)| = p - t + 1$, alors $\mathrm{Tr}(\rho(\sigma)) = t \bmod \ell$.

Cela a permis Koblitz et Miller d'introduire la cryptographie à base des courbes elliptiques l'année suivante, en 1986.

Nombres premiers et nombres friables

Puisque la théorie des nombres joue un rôle majeur dans la plupart des constructions en cryptographie à clef publique, il est naturel que les objets de base en théorie des nombres, les nombres premiers, soient d'intérêt primordial en cryptographie.

L'exemple le plus célèbre d'une telle question ouverte est la célèbre *Hypothèse de Riemann*, qui a des implications directes sur la distribution des nombres premiers.

Trouver des grands nombres premiers d'un nombre de bits donné est une tâche très courante dans la construction de plusieurs cryptosystèmes.

Nombres premiers

On pense généralement que si p_n est le n-ième nombre premier, alors

$$p_{n+1} - p_n \leq c(\log p_n)^2$$

pour une constante absolue $c > 0$, qui est une version moins stricte de la fameuse conjecture de Cramér, mais le meilleur résultat connu, dû à Baker, Harman et Pintz, n'affirme que $p_{n+1} - p_n \leq c p_n^{0.525}$.

D'un autre côté, on peut simplement choisir des entiers aléatoires dans un intervalle de la forme $[2^{k-1}, 2^k - 1]$, de la forme

$$p = 1 b_{k-2} \ldots b_2 b_1 1,$$

et tester leur éventuelle primalité avec un test probabiliste.

Avec une probabilité écrasante, ceci produit un nombre premier de k bits en un temps polynômial.

Nombres friables

Un autre ensemble d'entiers qui a une grande importance en cryptographie est l'ensemble des nombres dit *y-friables*. Ce sont les entiers n tels que tous les diviseurs premiers $p \mid n$ satisfont $p \leq y$.

En particulier, le design et l'analyse de plusieurs algorithmes cryptographiques sont basés sur notre capacité à contrôler la fonction $\psi(x, y)$ qui compte le nombre d'entiers $n \leq x$ qui sont y-friables.

Si on met $u = \log(x)/\log(y)$, il existe une fonction speciale $\rho(u)$ tel que

$$\psi(x, y) \sim x\rho(u)$$

tel que $\rho(u) \sim u^{-u}$. En particulier pour a fixé, $\psi(x, x^{1/a}) \sim x\rho(a)$.

Calcul d'indices

On suppose fixés un nombre premier p et des éléments g et $h = g^k$ de $G = \langle g \rangle \subseteq \mathbb{F}_p^*$, avec G d'ordre q. Les petits premiers $p_1, p_2, \ldots, p_s$ sont des générateurs distingués de $\mathbb{F}_p^*$ (provenant de $\mathbb{N}^*$).

On suppose tout d'abord que l'on connaît les logarithmes discrets $\log_g p_1, \ldots, \log_g p_s$, sont connus et on poursuit la stratégie suivante.

Étape 1 :

Choisir un entier aléatoire m et calculer $v \equiv hg^m \equiv g^{k+m} \bmod p$, en remarquant que $\log_g v = k + m$ (où $k = \log_g h$).

Étape 2 : Si on trouve une factorisation $v = p_1^{\alpha_1} \ldots p_s^{\alpha_s}$, alors

$$\log_g v = \alpha_1 \log_g p_1 + \ldots + \alpha_s \log_g p_s \bmod q.$$

et on donne comme résultat

$$k = \alpha_1 \log_g p_1 + \ldots + \alpha_s \log_g p_s - m \bmod q.$$

Si non, on retourne à l'**Étape 1**.

Calcul d'indices

Pour obtenir les logarithmes discrets des générateurs $\{p_1, \ldots, p_s\}$ on applique la même strategie pour déterminer assez des relations linéaires

$$\alpha_1 \log_g p_1 + \ldots + \alpha_s \log_g p_s - m \equiv 0 \bmod q,$$

tel que la matrice des relations détermine tous les $\log_g p_i$.

En choisissant une borne y sur p_s à maintenir un équilibre entre la phase de collection de relations et la phase d'algèbre linéaire (environ y^3), cela nous donne un algorithme ayant une complexité finale *sous-exponentielle*, d'environ

$$\exp\left(c\sqrt{(\log p)(\log\log p)}\right)$$

pour une constante c.

Courbes elliptiques

L'analogie entre les groupes $\mathbb{F}_p^*$ et $E(\mathbb{F}_p)$ pour une courbe elliptique E sur $\mathbb{F}_p$ va plus loin. Le groupe multiplicatif $\mathbb{F}_p^*$ s'identifie aux points sur $\mathbb{F}_p$ de la courbe affine $\mathbb{G}_m$ d'équation $uv = 1$, i.e.

$$\mathbb{G}_m(\mathbb{F}_p) = \{(u, v) \in \mathbb{F}_p^2 \ : \ uv = 1\}.$$

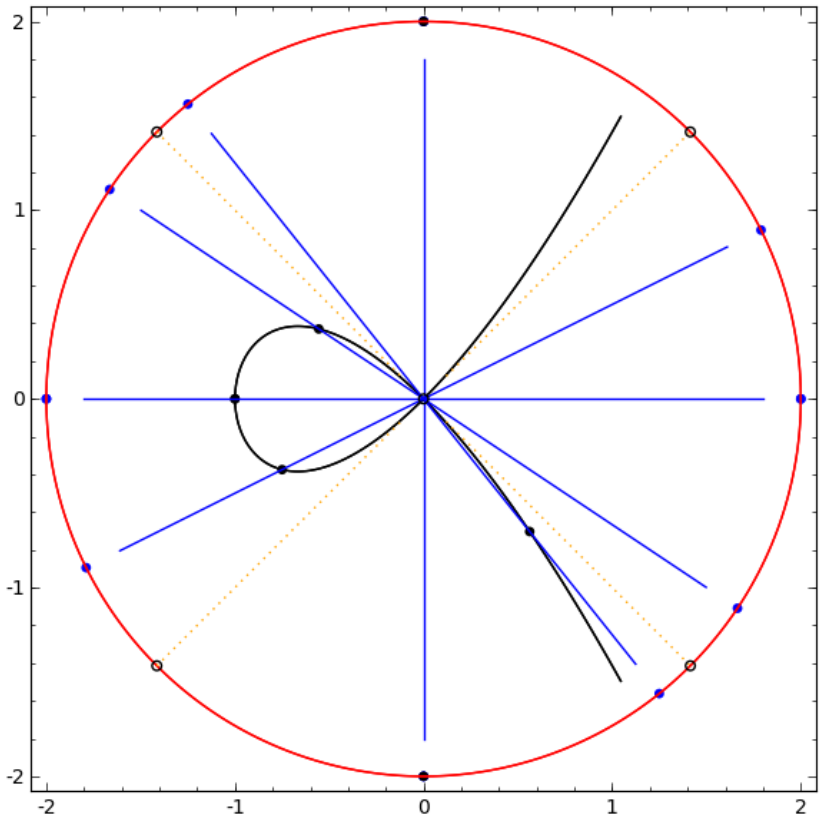
Une courbe elliptique est une cubique projective non singulière. Or dans la famille des cubiques $y^2 = f(x)$, il y a aussi des courbes singulières, comme

$$C : y^2 = x^3 + x^2.$$

Par restriction au lieu non singulier $C_{ns} = C \backslash \{(0, 0)\}$, la structure de groupe usuelle devient la *variété de groupe* qui est isomorphe à $\mathbb{G}_m$. L'isomorphisme est donné par la formule explicite

$$(x, y) \longmapsto \left(\frac{y - x}{y + x}, \frac{y + x}{y - x} \right).$$

Courbe elliptique singulière



$$\cong \mathbb{G}_m = \mathbb{A}^1 \setminus \{0\} = \mathbb{P}^1 \setminus \{0, \infty\}$$

$$(x, y) \longmapsto \frac{y - x}{y + x}.$$

Courbes elliptiques

La grande distinction entre les groupes $\mathbb{F}_p^*$ et groupes de courbes elliptiques est la manque d'un système ample de générateurs, qui est la talon d'Achille d'ElGamal et de RSA.

En effet, par le théorème de Mordell, pour toute courbe elliptique $E/\mathbb{Q}$, le groupe $E(\mathbb{Q})$ admet un système fini de générateurs, donc

$$E(\mathbb{Q}) \cong E(\mathbb{Q})_{\text{tors}} \times \mathbb{Z}^r,$$

où $|E(\mathbb{Q})_{\text{tors}}| \leq 14$ par un théorème de Mazur.

Par conséquent, il n'y a pas de méthode sous-exponentielle connu pour le problème du logarithme discret sur les courbes elliptiques sur $\mathbb{F}_p$.

Perspectives du monde classique

On peut se demander quelles sont les chances de survie du sujet au delà de la première moitié du XXI^e siècle. Si on croît au progrès constant des ordinateurs et des algorithmes, les systèmes à base de courbes elliptiques supplanteront les systèmes classiques, RSA et ElGamal.

Pour battre des ordinateurs et des algorithmes toujours plus puissants, il suffit d'allonger les clés. Chaque allongement linéaire améliore la sécurité d'un facteur sous-exponentiel, dans le cas de RSA, et d'un facteur exponentiel, pour les systèmes à courbes elliptiques.

Le cryptographe garde toujours l'avantage, mais, pour le même niveau de sécurité, les courbes elliptiques sont plus efficaces, et cela de plus en plus.

Perspectives du monde quantique

L'arrivée des ordinateurs quantiques change la donne, en introduisant un nouveau modèle de computation.

Shor a trouvé des algorithmes en temps polynômiaux pour la factorisation des entiers et le problème du logarithme discret sur un ordinateur quantique hypothétique.

Dans le monde post-quantique, RSA ainsi que la cryptographie à courbes elliptiques s'effondreront, alors que la cryptographie à base de réseaux (ainsi que certains systèmes utilisant des codes correcteurs d'erreurs, entre autres) ont une chance de survie.

Attendons de voir si un ordinateur quantique fiable sera réalisé de notre vivant, et quelles en seront les conséquences pratiques.

Questions ?