Théorie Algorithmique des Nombres

Méthodes sous-exponentielles

David Kohel

16/11/2020

Problème du log discret

Algorithme de Calcul d'indices d'après Odlyzko

Soit donné un groupe $G = \mathbb{F}_p^*$ et on détermine un base de friabilité :

$$B = \{-1, P_1 = 2, P_2 = 3, 5, 7, \dots, P_m\}.$$

On veut déterminer le logarithme discret $k = \log_a(b)$ pour $a, b \in \mathbb{F}_p^*$. (a, b donnés)

1e étape : Déterminer des relations :

$$(e_0, e_1, \dots, e_m) \in \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}^m,$$

càd. des éléments du noyau de :

$$\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}^m \xrightarrow{\pi} \mathbb{F}_p^* = G$$

$$(e_0, e_1, \dots, e_m) \longmapsto (-1)^{e_0} P_1^{e_1} \cdots P_m^{e_m}$$

Remarques

- Cette application est un homomorphisme de groupes abéliens :

  $\mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}^m$ avec addition

  $G = \mathbb{F}_p^*$ avec multiplication

  $\pi(u+v) = \pi(u)\pi(v)$.

- $\mathbb{Z}^{m+1} \longrightarrow \mathbb{Z}/2\mathbb{Z} \times \mathbb{Z}^m$ avec $(2, 0, \dots, 0)$ dans le noyau

Remarque 1e étape est un précalcul, indépendant du générateur a est élément b dont on cherche le log discret.

Dans la suite, on va élargir la base de générateurs à

$$\{-1, 2, 3, \dots, p_m, a, b\}.$$

On obtient donc une application

$$\mathbb{Z}^{m+3} \longrightarrow G = \mathbb{F}_p^*$$

$$(e_0, e_1, \dots, e_m, e_{m+1}, e_{m+2}) \longmapsto (-1)^{e_0} p_1^{e_1} \cdots p_m^{e_m} a^{e_{m+1}} b^{e_{m+2}}.$$

Comme pour la factorisation on cherche à identifier des relations

$$\pi((c_0, c_1, \dots, c_m)) = \alpha \overset{?}{=} \pi((d_0, d_1, \dots, d_m))$$

? = recherche d'une factorisation dans la base $\mathcal{B}$!

Mêmes idées d'un crible : on cherche à construire :

$$\left|(-1)^{c_0} p_1^{c_1} \cdots p_m^{c_m} - lp\right| < \lambda_4 \sqrt{p}, \quad (*)$$

en choissant des exposants (c_i) convenables.

Astuce : On cherche des entiers (c_i) tel que $c_1 \log p_1 + \cdots + c_m \log p_m \sim \log(lp)$.

N.B. $\log p_1, \dots, \log p_m$ fixés — on peut trouver une approximation de $\log(lp)$ par des méthodes d'approximation ou optimisation linéaire.

Avec $\pi((c_0, c_1, \dots, c_m)) \in [\![\lfloor \lambda\sqrt{p} \rfloor, \lfloor \lambda\sqrt{p} \rfloor]\!] \subseteq \mathbb{F}_p^*$ on a une probabilité plus élevée de trouver une factorisation :

$$\pi((c_0, c_1, \dots, c_m)) = \pi((d_0, d_1, \dots, d_m)).$$

On a donc $(e_0, e_1, \dots, e_m) = (c_0 - d_0, \dots, c_m - d_m) \in \ker(\pi)$, car π est un homomorphisme de groupes.

Objectif : Déterminer $\geq m+1$ relations tel que les relations engendre $+\varepsilon$ un sous-groupe de rang $m+1$ dans $\ker(\pi)$.

Remarque : $\pi : \mathbb{Z}^{m+1} \longrightarrow G \left(\cong \mathbb{Z}/(p-1)\mathbb{Z}\right)$

Le noyau est de rang $m+1$, indice fini (car $\mathbb{Z}^{m+1}/\ker\pi \cong G \cong \mathbb{Z}/(p-1)\mathbb{Z}$),

Si A est une matrice tel que les lignes sont des générateurs de $\ker(\pi)$, $A \in \mathbb{M}_{(m+1)\times(m+1)}(\mathbb{Z})$, on a $\det(A) = p-1$.

2e: Réduire les générateurs de $\ker(\pi)$ à des générateurs

$$(a_{i0}, \dots, a_{im}),\ 0 \leqslant i \leqslant m$$

tel que $A = (a_{ij})$

est une matrices triangulaire supérieure:

$$\begin{bmatrix} n_1 & & * \\ & \ddots & \\ 0 & & n_t \end{bmatrix} = A \text{ avec } \det A = n_1 \cdots n_t = p-1.$$

Si $\langle \{(e_0, e_1, \dots, e_m)\} \rangle \subsetneq \ker(\pi)$, alors on va trouver (indice r)

$$\det A = r(p-1).$$

2e étape: réduction de cette matrice

Pour p donné, on peut faire les étapes 1 & 2 comme précalcul.

3e étape. Pour a, b donné, on cherche encore des relations :

$$(-1)^{e_0} p_1^{e_1} \cdots p_m^{e_m} a^{e_{m+1}} b^{e_{m+2}} = 1 \in G,$$

ce qui permet de calculer une relation entre a et b, en éliminant $(-1), p_1, \ldots, p_m$:

$$a^r b^s = 1$$

et si le pgcd de s et p−1 est 1, on aura

$$a^{-rs^{-1}} = b,$$

donc $k = -rs^{-1} \bmod (p-1)$ est le logarithme discret.

Remarque. Au contraire du cas du problème de factorisation, il est nécessaire de travailler

avec une matrice $A \in \mathbb{M}_{n\times n}(\mathbb{Z}) = \mathbb{M}_n(\mathbb{Z})$ au lieu d'une matrice dans $\mathbb{M}_n(\mathbb{F}_2)$. La partie algèbre linéaire est donc plus chère. Neanmoins, par des considérations semblables on trouve une complexité $L_p(1/2, c)$.

Précisement: $L_p(1/2, \sqrt{2})$.

Rappel (notation):

$$L_n(\lambda, c) := \exp\left(c (\log n)^{\lambda} (\log\log n)^{1-\lambda}\right).$$

dit sous-exponentiel (en $\log n$) si $0 < \lambda < 1$.

Courbes elliptiques: Courbes dont les points forme un groupe:

$E: y^2 = x^3 + ax + b$, $a, b \in \mathbb{F}_p$.

Il existe des algorithmes/formules pour addition

$P = (x_1, y_1) + Q = (x_2, y_2) = (x_3, y_3) \in E$.

Il n'y a pas d'analogue de l'algorithme de calcul d'indices, car il n'y a pas générateurs "petits" ou notion de factorisation. Tous les points (x,y) se ressemblent. Pour cette raison on ne connaît que des algorithmes exponentiels pour le log discret dans E. Par conséquent, pour la même sécurité, on choisit

RSA : $\mathbb{Z}/n\mathbb{Z}$, $n=p,q$. ou

ElGamal : $\mathbb{F}_p^*$

des premiers p,q de 1024 à 2048 bits

En comparaison on peut choisir

$E/\mathbb{F}_p$ (Analogue de ElGamal)

avec 160 à 240 bits.

Pour la même sécurité.

Avec un ordinateur quantique, la factorisation est en temps polynôme (Schor) mais aussi les logs discrets (y inclus sur des courbes elliptiques.

Il y a donc beaucoup de recherche sur des nouveaux cryptosystèmes qui sont eventuellement sûr dans ce monde post-quantique (où il existe des ordinateurs quantiques). Candidats :

à la base de : structures math. : graphes & geom. alg.

- des isogénies des courbes elliptiques (= homomorphisme géométrique)

semblables

- réseaux (problèmes de recherche des vecteurs proches) = analogue d'un espace vectoriel normé avec coefficients des entiers
- la théorie des codes cor. d'erreurs

- schema à la base des polynômes en plusieurs variables.