Осень 2019

Лекция №6 — 25.10.19

Лектор: Елена Киршанова *Оформил Филипп Максимов*

1 Алгоритм подсчёт $\mathbb{F}_q$-рациональных точек эллиптической кривой

Из предыдущей лекции:
$E(\mathbb{F}_q) : y^2 = x^3 + Ax + B$ — эллиптическая кривая.

$\#E(\mathbb{F}_q) = |E(\mathbb{F}_q)|$ — число q-рациональных точек кривой (или порядок кривой).

Теорема Хассе даёт верхнюю и нижнюю границы оценки для $|E(\mathbb{F}_q)|$:

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

Асимптотика: $\#E(\mathbb{F}_q) = \mathcal{O}(q)$.

Мы доказали, что $\#E(\mathbb{F}_q) = q + 1 + \sum_{x \in \mathbb{F}_q} \left(\frac{x^3 + Ax + B}{\mathbb{F}_q} \right)$, где $\left(\frac{\cdot}{\mathbb{F}_q} \right)$ — кв. вычет в $\mathbb{F}_q$ (если q — простое $\Rightarrow$ символ Лежандра)
Время вычисления квадратичного вычета в таком случае: $\mathcal{O}(q \log q) = \widetilde{\mathcal{O}}(q)$ — экспоненциальное от числа бит q.

1.1 Алгоритм подсчёта точек Baby Step — Giant Step = Алгоритм нахождения порядка точки $P \in E(\mathbb{F}_q)$

Идея: Пусть $N = \#E(\mathbb{F}_q)$ — неизвестно.

По теореме Лагранжа: $N \cdot P = \infty \ \ \forall P$

Так как $q + 1 - 2\sqrt{q} \leq N \leq q + 1 + 2\sqrt{q}$ (т.е. N лежит в интервале р-ра $4\sqrt{q}$, мы можем проверить все N из этого интервала: $N \cdot P \stackrel{?}{=} \infty$).

Наивный метод (брутфорс): $\mathcal{O}(\sqrt{q})$
Алгоритм BS-GS – стандартный алгоритм нахождения коллизий / цикла функции– ускоряет до $\mathcal{O}(q^{\frac{1}{4}})$

Для начала опишем алгоритм нахождения порядка точки $P \in E$.

1.1.1 Алгоритм BS—GS для нахождения порядока точки P

Вход: $P \in E(\mathbb{F}_q)$
Выход: $k = \mathrm{ord}(P)$

1. $Q = (q+1)P$

2. (BS) Выберем минимальное $m \in \mathbb{Z}$: $m > q^{\frac{1}{4}}$
Вычислим и сохраним (в список L) все $\pm j \cdot P, j \in \{0, \dots, m\}$. Сортируем L.

3. (GS) Вычисляем точки $Q + k(2mP)$ для всех $k \in \{-m \dots m\}$, пока не найдём в списке L точку $\pm jP$, что

$$Q + k(2mP) = \pm jP.$$

$Q + k(2mP) = \pm jP \iff (q + 1 + 2mk \mp j)P = \infty.$

4. Положим $M = q + 1 + 2mK \mp j$ (порядок P — делитель M)

5. Факторизуем $M = p_1^{e_1}..p_r^{e_r}$

6. Для $i = 1..r$:

Если $\frac{M}{p_i}P = \infty$:

$M \leftarrow \frac{M}{p_i}$

Вернуться к Шагу 5 (неоптимально, но корректно).

Если $\frac{M}{p_i}p_i \neq \infty \ \forall i$:

Вернуть M.

Иначе

Вернуть ‘fail’.

1.1.2 Корректность

1. Найдём ли мы коллизию на шаге 3?

Лемма 1 (Разбиение элемента x на $2m$ старших бит и $(\log x - 2m)$ младших)**.** *Пусть* $x \in \mathbb{Z} : |x| \leq 2m^2$. *Тогда* $\exists\ x_0, x_1 \in \mathbb{Z}$, *т.ч.* $-m < x_0 \leq m, -m < x_1 \leq m$

Доказательство. Пусть

$$x_0 = x \bmod 2m, \text{ где mod } B \in \left(-\frac{B}{2}, \frac{B}{2}\right]$$

$$x_1 = \frac{x - x_0}{2m}$$

Тогда $|x_1| \leq \dfrac{2m^2 + m}{2m} < m + 1.$ □

2. Почему шаг 6 возвращает порядок P?

Лемма 2. *Пусть G — аддитивная группа, $g \in G$. Положим $M > 0$, что $Mg = 0$, $M = p_1^{e_1} ... p_r^{e_r}$, где p_i — различные простые. Тогда если $\left(\frac{M}{p_i}\right) g \neq 0 \ \forall i \in \{1..r\}$, то M — порядок g.*

Доказательство. Пусть k — порядок g. Тогда $k \mid M$. Положим $k \neq M$ (от противного).

Пусть p_i — простое делящее $\frac{M}{k}$.

Тогда $(p_i \cdot k) \mid M$, или $k \mid \left(\frac{M}{p_i}\right) \Rightarrow \left(\frac{M}{p_i}\right) g = 0$ (т.е. мы нашли $p_i : \left(\frac{M}{p_i}\right) g = 0$), что противоречит утверждению теоремы $\Rightarrow k = M$.

□

1.1.3 Анализ сложности алгоритма

Шаг 1 (Быстрое сложение)

$\mathcal{O}(\log q)$ операций «+» на кривой, каждый «+»: $\operatorname{polylg} q \Rightarrow \mathcal{O}(\operatorname{polylg} q)$

Шаг 2

$\widetilde{\mathcal{O}}(m) = \widetilde{\mathcal{O}}(q^{1/4})$ — времени
$\mathcal{O}(q^{1/4})$ — память.

Шаг 3

$\widetilde{\mathcal{O}}(2m) = \widetilde{\mathcal{O}}(q^{1/4})$ — ожидаемое количество переборов k.

Шаг 4

Элементарные операции в $\mathbb{F}_q$

Шаг 5

$L\left[\frac{1}{3}, \frac{64}{9}^{\frac{1}{3}}\right] = \exp\left(\left[\frac{64}{9}^{\frac{1}{3}} + \mathcal{O}(1)\right] (\log n)^{\frac{1}{3}} (\log\log q)^{\frac{2}{3}}\right)$

Шаг 6

$\widetilde{\mathcal{O}}(\sqrt{\log M} \cdot \operatorname{poly}\log q) = \widetilde{\mathcal{O}}(\sqrt{\log M}) = \widetilde{\mathcal{O}}(\operatorname{poly}\log q)$
макс. $r \approx \sqrt{\log M}$

Итого: самый затратный шаг — №3, $\widetilde{\mathcal{O}}(q^{\frac{1}{4}})$

1.1.4 Замечания к алгоритму

1. Для оптимизации памяти (вычислений) на шаге 3 достаточно хранить x-координату.

2. Классическим алгоритмом cycle finding (Поллард-ρ) можно реализовать алгоритм, используя только $\operatorname{poly}\log q$ памяти

1.2 Алгоритм вычисления $\#(\mathbb{F}_q)$

Выбрать N – максимальное число случайных точек $P \in E(\mathbb{F}_q)$

Положить $L = 1$ Для $i = 1 \dots N$: 1. Выбрать $P \in E(\mathbb{F}_q)$ (случайно выбираем x-координату, посчитать $y : y^2 = x^3 + Ax + B$)

2. Найти $k_p = \operatorname{ord} P$

3. $L = \operatorname{lcm}(L, k_p)$

4. Если $L >= 4\sqrt{q}$ and $L < q + 1 - 2\sqrt{q}$:

   Вычислить $m = (q + 1 + \lceil 2\sqrt{q} \rfloor // L$

   Вернуть $L \cdot m$

Иначе

   Вернуть L

Вернуть 'fail'

Всевозможные порядки, которые могут быть получены на шаге 3:

$$\operatorname{ord} p_i \in \{ \prod_{0 \le \ell_i \le e_i} p_i^{e_i} \}(\text{все делители } \#E(\mathbb{F}_q)), \text{ где } \#E(\mathbb{F}_q) = \prod p_i^{e_i}$$

Всего $\prod_{i=1}^{r} e_i$ всевозможных порядков, примерно $\widetilde{\mathcal{O}}(\log \#E(\mathbb{F}_q))$ повторов на шаге 3.

2 Алгоритм Схофа (Schoof's Algorithm)

R. Schoof "Counting points on elliptic curves over finite fields"

$E : y^2 = x^3 + Ax + B$

Первый полиномиальный алгоритм от $\log q$(!), q — простое. Основная идея: вычислить $\#E(\mathbb{F}_q) \bmod \{2, 3 ... p\}$ — простые числа и затем восстановить $\#E(\mathbb{F}_q)$ по CRT.

1. Вычисление $\#E(\mathbb{F}_q) \bmod 2 : \#E(\mathbb{F}_q)$ — чётно
$\Leftrightarrow E(\mathbb{F}_q)$ содержит точку $\neq \infty$ порядка 2.
Точки порядка 2 имеют y-координату $= 0 \Leftrightarrow x^3 + Ax + B = 0$ в $\mathbb{F}_q$

Как определить, есть ли у $x^3 + Ax + B$ корни в $\mathbb{F}_q$?
Все элементы $\mathbb{F}_q$ — корни $x^q - x$. Т.е. $x^3 + Ax + B = 0$ в $\mathbb{F}_q \Leftrightarrow \gcd(x^q - x, x^3 + Ax + B) \neq 1$ в $\mathbb{F}_p[x]$

$\exists$ эффективный алгоритм (вычисление x^q проводится в $\mathbb{F}_p[x]/(x^3+Ax+B)$ ускоренным алгоритмом возведения в степень.

Сложность: $\mathcal{O}(\log^3 q)$

2. Обобщим на другие $\ell \in \{2..p\}, \ell \neq 2$

Так как. $|\#E(\mathbb{F}_q)-q-1| < 2\sqrt{q}$, для получения $\#E(\mathbb{F}_q)$ достаточно рассмотреть простые ℓ, что

$$\prod_{\ell} \ell > 4\sqrt{q}$$

По теореме о распределении простых чисел ($p_n \sim n \log n$), нам будет достаточно взять $\mathcal{O}(\log q)$ простых ℓ, каждый размером $\mathcal{O}(\log q)$

2.1. Как и в случае $\ell = 2$, рассмотрим группу точек ℓ-кручения

$$E[\ell] = \{P \in E(\overline{\mathbb{F}_q}) : \ell \cdot P = \mathcal{O}\} \simeq \mathbb{Z}/\ell\mathbb{Z} \times \mathbb{Z}/\ell\mathbb{Z},$$

где $\psi_q(x) \in \mathbb{F}_q[x] - \ell$ — многочлены деления ($\in \mathbb{F}_q[x]$ т.к. мы берём простые (нечётные) ℓ). Они могут быть эффективно получены в явном виде с помощью рекуррентных соотношений.

2.2. Эндоморфизм Фробениуса $\phi_q : E \to E, (x, y) \mapsto (x^q, y^q)$ удовлетворяет соотношению (см. Лекцию №5)

$$\phi_q^2 - a\phi + q = 0, \text{ где } a \text{ — след Фробениуса } (a = q + 1 - \#E(\mathbb{F}_q)) \tag{1}$$

$$(\text{т.е. } (x^{q^2}, y^{q^2}) - [a](x^q, y^q) + [q] = \infty)$$

Соотношение (1) справедливо и $\bmod \ell$, т.е.
$\phi_q^2 - a'\phi + q' = 0$ для некоторых $a' \in \{0 \bmod \ell - 1\}$,
что $a \equiv a' \bmod \ell, q' \in \{0 \dots \ell - 1\}, q = q' \bmod \ell$

Важно: соотношение (1) может быть задано с помощью многочленов $\Rightarrow$ для кандидатов a' $\exists$ эффективная проверка:

$$q' \cdot P = \left(\frac{\phi_{q'}(x)}{\psi_{q'}^2(x)}, \frac{\omega_{q'}(x,y)}{\psi_n^3(x,y)} \right)$$

$$(x^{q^2}, y^{q^2}) + q'(X, Y) \equiv a'(x^q, y^q) \bmod \psi_\ell(x) \text{ и } \bmod (Y^2 - X^3 - AX - B)$$

2.1 Сложность алгоритма

Для фиксированного $\ell \in \{2 \dots p\}$ (повт. для каждого ℓ, всего различных $\mathcal{O}(\log q)$):

- Подсчёт $x^{q^2}, x^q, \dots \bmod \psi_\ell(x)$
  $\mathcal{O}(\log q \cdot (\ell^2 \log q)^2)$ = возведение в степень $\times$ количество бит в многочлене

- Умножение точки $(x^q, y^q)a' \leq \ell$ раз для каждого кандидата a : $\mathcal{O}(\ell \cdot (\ell^2 \log q)^2)$

Всего: $\mathcal{O}(\log q \cdot (\mathcal{O}(\log q(\ell^2 \log q)^2) + \mathcal{O}(\ell(\ell^2 \log q)^2))) = \{\ell = \log q\} = \mathcal{O}(\log^8 q)$