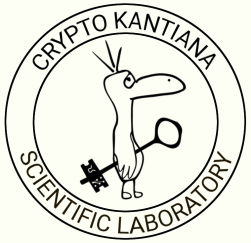
Эллиптические кривые

Лекция 8. Тест на простоту на эллиптических кривых

Семён Новосёлов

БФУ им. И. Канта

2025



IKBFU

Мотивация

Тесты на простоту используются в криптографии для:

- генерации простых чисел, как в криптосистеме RSA
- генерации групп для криптосистем на DLOG (необходимы простые группы)

План

1. Тест на простоту Миллера-Рабина
2. Доказательство простоты с помощью ЭК

Тест на простоту Миллера-Рабина

Малая теорема Ферма

p – простое, $p \nmid a \implies a^{p-1} - 1 \equiv 0 \pmod{p}$.

$$p - 1 = 2^k \cdot q,\ q \text{ – нечётное}$$

$$\Downarrow$$

$$a^{p-1} - 1 = (a^{2^{k-1} \cdot q} - 1) \cdot (a^{2^{k-1} \cdot q} + 1) =$$
$$= (a^q - 1)(a^{2^{k-1} \cdot q} + 1) \cdot \ldots \cdot (a^{2 \cdot q} + 1) \cdot (a^q + 1)$$

- p – простое $\implies$ p делит один из множителей, т.е. выполняется одно из условий:

$$a^q \equiv 1, a^{2^{k-1} \cdot q} \equiv -1, \ldots, a^q \equiv -1 \qquad (1)$$

- p – не простое $\implies$ $\exists a$: все сравнения (1) не выполняются

Идея: для проверки n на простоту выбираем случайное число a, $\gcd(a, n) = 1$ и проверяем (1) по модулю n.

- Число a т.ч. $a^q \not\equiv 1, a^{2^{k-1} \cdot q} \not\equiv -1, \ldots, a^q \not\equiv -1$ называется **свидетелем**, что n – составное.

Алгоритм Миллера-Рабина

Вход: n, a.
Выход: «составное» или «возможно простое»

1. $n - 1 = 2^k q$, q – нечётное
2. $a = a^q \bmod n$
3. **if** $a \equiv 1 \bmod n$:
   return «возможно простое»
4. **for** $i = 0 \ldots k - 1$
5. **if** $a \equiv -1 \bmod n$
   return «возможно простое»
6. $a = a^2 \bmod n$
7. **return** «составное»

Для проверки на простоту:

- алгоритм выполняется K раз для случайных $a \in [2, n-2]$
- время работы: $O(K \log^3 n)$
- вероятность ошибки: 2^{-2K}

Тест Миллера

Алгоритм можно сделать **детерминированным** (предполагая GRH) перебрав все $a \leq 2\ln^2 n$.

- Сложность: $\widetilde{O}(\log^4 n)$.
- Для маленьких n достаточно перебрать только небольшое кол-во a.
- Например, для $n \leq 2^{64}$ достаточно проверить $a = 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31$ и 37.

Тест на простоту на эллиптических кривых

Задача: По данному (большому) числу p определить, является ли p простым числом и, если да, вывести доказательство (**сертификат**) простоты p.

- самый быстрый[1] на сегодняшний день алгоритм предложен Goldwasser-Killan в 1986
- с улучшениями время работы $= \mathrm{poly}\log p$, проверка сертификата простоты: $O(\log^4 p)$

[1] на практике, сложность в худшем случае не известна

- **детерминированные** алгоритмы (Cohen-Lenstra'1984) работают за **квази**-полиномиальное[2] от $\log p$ время $(\log p)^{O(\log\log p)}$

  $\Longrightarrow$ пригодны только для небольших чисел p.

Идея алгоритма Goldwasser-Killan: заменить группу $\mathbb{Z}_n^\times$ как в алгоритме Миллера-Рабина или тесте Поклингтона-Леммера на $E(\mathbb{Z}_n)$.

[2] если не использовать эвристики, как в детерминированной версии Миллера-Рабина

I. Предварительные сведения

Теорема о распределении порядков случайных ЭК

Пусть $p > 5$ – простое, $S \subseteq [p + 1 - \lfloor\sqrt{p}\rfloor, p + 1 + \lfloor\sqrt{p}\rfloor]$ и $A, B \leftarrow \mathbb{F}_p$. Тогда $\exists\, c$ – константа, т.ч.

$$\Pr\left[\#E_{A,B}(\mathbb{F}_p) \in S\right] > \frac{c}{\log p} \cdot \frac{|S| - 2}{2\lfloor\sqrt{p}\rfloor + 1},$$

где $\#E_{A,B}(\mathbb{F}_p)$ – число точек на $E_{A,B} : y = x^3 + Ax + B$.

◁ **Док-во:** Lenstra'1987 ▷

Неформальная интерпретация теоремы:

- число точек $E_{A,B}$ ведёт себя как случайное число из интервала $[p + 1 - \lfloor\sqrt{p}\rfloor, p + 1 + \lfloor\sqrt{p}\rfloor]$

Лемма

Пусть $n \in \mathbb{Z},\ 2, 3 \nmid n; p > 3$ – простой делитель n и $4A^3 + 27B^2 \not\equiv 0 \bmod p$.

- Для любого $x \in \mathbb{Z}/n\mathbb{Z}$ пусть $x_p := x \bmod p$.
- Для любой точки $L = (x, y) \in E_{A,B}(\mathbb{Z}/n\mathbb{Z})$ пусть $L_p = (x_p, y_p) \in E_{A,B}(\mathbb{F}_p)$.

Тогда $\forall L, M \in E_{A,B}(\mathbb{Z}/n\mathbb{Z})$, если $L + M$ определено, то $(L + M)_p = L_p + M_p$.

Теорема (Критерий простоты)

Положим:

- $n \in \mathbb{Z}$, $A, B \in \mathbb{Z}/n\mathbb{Z}$
- $2, 3 \nmid n$, $\gcd(4A^3 + 27B^2, n) = 1$
- $L \in E_{A,B}(\mathbb{Z}/n\mathbb{Z})$, $L \neq \infty$

Тогда:

$\exists$ простое $q > (n^{1/4} + 1)^2$, т.ч. $qL = \infty \implies n$ – простое.

◁ От противного: пусть n – составное $\Rightarrow \exists p > 3$, т.ч. $p \mid n$, p – простое. Т.к. p – простое, то $p \leq \sqrt{n}$.

- Заметим: $\gcd(4A^3 + 27B^2, p) \neq 0 \bmod p$.
  Иначе: противоречие с $\gcd(4A^3 + 27B^2, n) = 1$.
- Тогда по Лемме имеем $L_p \in E_{A,B}(\mathbb{F}_p)$ и
  $q \cdot L_p = (qL)_p = \infty_p = \infty \Rightarrow \operatorname{ord}(L_p) \mid q \Rightarrow \operatorname{ord}(L_p) = q$, т.к. q – простое.
- Имеем:
  $(n^{1/4} + 1)^2 < q = \operatorname{ord}(L_p) \leq \#E_{A,B}(\mathbb{F}_p) < (\sqrt{p} + 1)^2$.
- $\Longrightarrow \quad p > \sqrt{n}$.
- Это противоречие, значит, n – простое. ▷

II. Алгоритм: тест на простоту

Вкратце:

- доказательство простоты p сводим к доказательству простоты $q \leq \frac{p}{2} + o(p)$
- рекурсивно применяем алгоритм для $p = q$, пока не получим достаточно малое значение q – такое, что **детерминированные** тесты будут эффективны.

По критерию простоты задача сводится к построению для p, кривой $E_{A,B}$ над $\mathbb{F}_p$ с точкой L порядка $q \approx p/2$.

Алгоритм 1. Gen_curve

Вход: p

Выход: A, B, q

1. $A, B \xleftarrow{\$} \mathbb{F}_p$, т.ч. $(4A^3 + 27B^2, p) = 1$ и $\#E(\mathbb{F}_p) \equiv 0 \pmod 2$
2. $q = \#E_{A,B}(\mathbb{F}_p)/2$
   if $2 \mid q$ или $3 \mid q$
   перейти к шагу 1
3. Запустить вероятностный алгоритм проверки q на простоту (Миллера—Рабина) на $O(\log p)$ шагов (т.е. чтобы вероятность ошибки была $\sim 2^{-\log p}$).

Алгоритм 2. Find_point

Вход: p, q, A, B.
Выход: точка $L \in E(\mathbb{F}_p)$ порядка q.

1. $x \xleftarrow{\$} \mathbb{F}_p$ т.ч. $x^3 + Ax + B$ – квадрат в $\mathbb{F}_p$
2. $y \xleftarrow{\$} \left\{\pm\sqrt{x^3 + Ax + B}\right\}, L := (x, y)$
3. **if** $q \cdot L \neq \infty$:
   перейти к шагу 1.
4. **return** L

Алгоритм 3. Prove_prime

Вход: p, LB – число бит в числе такое, что детерминированные алгоритмы простоты эффективны для этого числа.
Выход: сертификат простоты

1. $i = 0, p_0 = p$
2. **while** $p_i > 2^{LB}$:
   - 2.1 $(A_i, B_i), p_{i+1} \leftarrow \text{Gen\_curve}(p_i)$
   - 2.2 $L_i \leftarrow \text{Find\_point}(p_i, p_{i+1}, A, B)$
   - 2.3 $i := i + 1$
   - 2.4 **if** $i \geq (\log p)^{\log\log p}$ или $2 \mid p_i$ или $3 \mid p_i$ перейти к шагу 1
3. проверить p_i на простоту детерминированным алгоритмом
   if не доказано, что p_i – простое: перейти к шагу 1
4. **return** $C = ((A_0, B_0), L_0, p_1, ..., (A_{i-1}, B_{i-1}), L_{i-1}, p_{i-1})$

Корректность

- p – простое. Тогда выход C — сертификат: «свидетельство» простоты p. На шагах 2.1, 2.2 мы получаем кривую E_{A_i,B_i} и точку L_i порядка p_{i+1}, удовлетворяющие условиям Критерия простоты.
- p — составное. Тогда получим делители p на шаге 3 (или раньше) алгоритма Find_point(), аналогично алгоритму факторизации.

Сложность

Алг. 1. Gen_curve

Самый затратный шаг – вычисление $\#E_{A,B}(\mathbb{F}_p)$

$\Longrightarrow$ алгоритм Схоофа-Элкиса-Аткина: $\widetilde{O}(\log^4 p)$.

Алг. 2. Find_point

Самые затратные шаги:

Шаг 1: $x \xleftarrow{\$} \mathbb{F}_p$ – кв. вычет с вероятностью $O(1)$.

Шаг 4: быстрое умножение на q:

$\widetilde{O}(\log q \cdot \log p) = \widetilde{O}(\log^2 p)$ битовых операций.

Алг. 3. Prove_prime

На шаге 2, p_i уменьшается на 2 $\Longrightarrow$ $O(\log p)$ итераций.

Доминирующий шаг: подсчёт точек в Gen_curve

$\Longrightarrow$ общее время работы: $\widetilde{O}(\log^5 p)$

Количество кривых $E_{A,B}$, не удовлетворяющих условиям шага 1 в Gen_curve() $= O(\log^3 p)$ (эвристика)

Проверка сертификата. Алгоритм 4. Check_prime

Вход: $p_0, C = ((A_0, B_0), L_0, p_1, ..., (A_{i-1}, B_{i-1}), L_{i-1}, p_{i-1})$

Выход: {Reject, Accept}

1. **for** $j = 0 \ldots i - 1$:
   - **(a)** assert $(2 \nmid p_j)$
   - **(b)** assert $(3 \nmid p_j)$
   - **(c)** assert $(gcd(4A_j^3 + 27B_j^2, p_j) = 1)$
   - **(d)** assert $(p_{j+1} > (p_j^{1/4} + 1)^2)$
   - **(e)** assert $L_j \neq \infty$
   - **(f)** assert $p_{j+1} L_j = \infty$
2. **return** Accept

Корректность

- Check_prime() возвращает Accept $\Rightarrow p_i$ – простое $\Rightarrow$ p_{i-1} – простое по Критерию простоты ($\Rightarrow \ldots \Rightarrow p_0$ – простое)
- Условия (a),(b) проверяются на шаге 2.4. алгоритма 3. Prove_prime
- (c) – шаг 1 в Алг.1. Gen_curve
- (d) – Теорема Хассе-Вейля: $\#E_{A,B}(\mathbb{F}_{p_j}) \geq (\sqrt{p_j} - 1)^2 \Rightarrow$

$$p_{j+1} = \frac{\#E(\mathbb{F}_{p_j})}{2} \geq \frac{(\sqrt{p_j} - 1)^2}{2} > (p_j^{1/4} + 1)^2 \quad \forall p_j > 37$$

(для малых p_j проверка на простоту тривиальна)

- (e), (f) проверяются в Find_point, шаг 3.

Время работы

- Проверка каждого p_j : $O(\log^3 p)$ — шаг (f) самый затратный.
- Всего: $O(\log p)$ различных p_j в сертификате $C \Rightarrow O(\log^4 p)$.

Литература

- H. Cohen и H. W. Lenstra. Primality testing and Jacobi sums. 1984.
- S. Goldwasser и J. Kilian. Primality testing using elliptic curves. 1999.
- H. W. Lenstra Jr. Factoring integers with elliptic curves. 1987.
- L. C. Washington. Elliptic curves: number theory and cryptography. 2008.

Контакты

snovoselov@kantiana.ru