Криптосистеми на еліптичних кривих

Lecture 5: DLP

Грубіян Євген Олександрович

Задача дискретного логарифмування в групі точок еліптичної кривої (ECDLP)

Умова задачі:

Нехай E/K — еліптична крива, визначена над полем K, а $E(K)$ — група K-раціональних точок на E.

Дано точки $P, Q \in E(K)$, причому P є генератором підгрупи, і $Q = [k]P$ для деякого невідомого цілого числа k.

Мета: Знайти дискретний логарифм k (тобто, обчислити k таке, що $Q = [k]P$).

Складність: Вважається важкою задачею в класичній моделі обчислень, що є основою для криптографічних систем на еліптичних кривих.

Наївний алгоритм пошуку дискретного логарифму

Алгоритм:

1. Починаємо з $k = 0$ та обчислюємо послідовно точки:

$$P, [2]P, [3]P, \ldots, [k]P.$$

2. Порівнюємо кожну точку з Q до знаходження збігу:

$$\text{Якщо } [k]P = Q, \text{ то } k \text{ є дискретним логарифмом.}$$

Оцінка часу: $O(n)$, де n — порядок підгрупи, що експоненційно залежить від параметра безпеки криптосистеми.

Алгоритм Шенкса (Baby-Step Giant-Step)

Ідея: Розбити пошук дискретного логарифму на два етапи, використовуючи таблицю для збереження проміжних значень та парадокс днів народжень.

Нехай $n = \#\langle P \rangle$ та вибираємо $m \approx \lceil \sqrt{n} \rceil$.

Кроки:

1. Baby steps: Обчислити та зберегти $[j]P$ для $j = 0, 1, \ldots, m-1$.
2. Giant steps: Обчислити $Q - [im]P$ для $i = 0, 1, \ldots, m-1$ і перевіряти чи співпадає з будь-яким з baby steps.

Якщо знайдено співпадіння: $Q - [im]P = [j]P \Rightarrow Q = [im + j]P$, тобто $k = im + j$. Парадокс днів народжень гарантує що таку колізію з імовірністю близькою до 60% можна знайти за $\sqrt{n}$ кроків
Оцінка часу: $O(\sqrt{n})$ як за обчислювальними операціями, так і за пам'яттю.

Алгоритм Поларда (ρ-метод)

Постановка задачі: Дано генератор P підгрупи E еліптичної кривої, при чому порядок підгрупи $\#\langle P\rangle = p$ - просте число, та точку $Q = [k]P$. Мета: знайти k.

Основна ідея: Створити псевдовипадкову послідовність точок на E, що задається функцією $f : E \to E$ з відповідними коефіцієнтами a_i, b_i так, що

$$R_i = [a_i]P + [b_i]Q,$$

і знайти два різні індекси i та j, для яких $R_i = R_j$. Потім отримати лінійне рівняння для k.

Алгоритм Поларда

Розбиття групи: Розбиваємо E на 3 підмножини S_1, S_2, S_3 (наприклад, за властивістю x-координати).

Визначення функції f та коефіцієнтів: Для кожної точки $R = [a]P + [b]Q$ визначаємо оновлення:

- Якщо $R \in S_1$:

$$f(R) = R + P, \quad a' = a + 1, \quad b' = b.$$

- Якщо $R \in S_2$:

$$f(R) = 2R, \quad a' = 2a, \quad b' = 2b.$$

- Якщо $R \in S_3$:

$$f(R) = R + Q, \quad a' = a, \quad b' = b + 1.$$

Початкові значення задаємо, наприклад, як

$$R_0 = P, \quad a_0 = 1, \quad b_0 = 0.$$

Алгоритм Поларда

Запускаємо алгоритм (застосовуючи метод Флойда або Брента) для послідовності:

$$R_0,\ R_1 = f(R_0),\ R_2 = f(R_1),\ \ldots$$

Знаходимо два індекси $i \neq j$ такі, що:

$$R_i = R_j \quad \Longrightarrow \quad [a_i]P + [b_i]Q = [a_j]P + [b_j]Q.$$

Тоді:

$$[a_i - a_j]P = [b_j - b_i]Q.$$

Оскільки $Q = [k]P$, маємо:

$$[a_i - a_j]P = [b_j - b_i][k]P,$$

тобто, при умові, що $b_j - b_i$ обернений за модулем порядку підгрупи,

$$k \equiv (a_i - a_j)(b_j - b_i)^{-1} \pmod{n}.$$

Метод Флойда (Tortoise and Hare) для виявлення циклу

Мета: Знайти два індекси $i < j$ такі, що $x_i = x_j$ у послідовності, що задається функцією f.

Алгоритм:

1. Встановлюємо два покажчики:
   - «Черепаха» (tortoise): рухається крок за кроком, тобто $x_{i+1} = f(x_i)$.
   - «Заєць» (hare): рухається вдвічі швидше, тобто $x_{2i+1} = f(f(x_i))$.
2. Продовжуємо обчислення поки не отримаємо першу колізію: знайти найменше i таке, що $x_i = x_{2i}$.
3. Після знаходження колізії встановлюємо один покажчик на початок послідовності і рухаємо обидва покажчики з однаковою швидкістю (по одному кроку за раз) для знаходження початку циклу.

Оцінка: Очікувана часова складність алгоритму дорівнює $O(\lambda)$, де λ — довжина циклу, а пам'ять використовується $O(1)$.

Оцінка складності та підсумок алгоритму Pollard Rho

Оцінка часу: Очікувана кількість операцій становить $O(\sqrt{n})$ групових додавань, де n — порядок підгрупи, що використовується. При цьому алгоритм вимагає $O(1)$ пам'яті.

Підсумок:

1. Визначається псевдовипадкова функція f на групі E з оновленням коефіцієнтів a та b.
2. Генерується послідовність точок $R_i = [a_i]P + [b_i]Q$.
3. За допомогою алгоритму пошуку колізій (наприклад, методом Флойда) знаходиться індекс $i \neq j$ для якого $R_i = R_j$.
4. З отриманої колізії розв'язується лінійне рівняння для k.

Алгоритм Сільвера-Поліга-Хеллмана для груп довільного порядку

Нехай G — скінченна абелева група порядку $n = p_1^{e_1} \dots p_s^{e_s}$ згідно основної теореми арифметики, задані точки: $Q = [k]P,\ \mathrm{ord}(P) = n$
Мета: знайти ціле число k (дискретний логарифм) за умови, що $0 \leq k < n$.
У цьому алгоритмі ми розглядаємо окремо випадок, коли порядок точки є степенем простого p^e, а потім узагальнимо алгоритм на випадок порядку точки $n = p_1^{e_1} \dots p_s^{e_s}$ за допомогою китайської теореми про лишки:
Нехай $Q_i = [k_i]P_i$, де $P_i = [n/p_i^{e_i}]P$, $Q_i = [n/p_i^{e_i}]Q$, $k_i = k \mod p_i^{e_i}$.
Зауважимо що $\mathrm{ord}(P_i) | p_i^{e_i}$, тобто ми шукаємо дискретний логарифм в підгрупі $\langle [n/p_i^{e_i}]P \rangle$ порядку $p_i^{e_i}$

Алгоритм Сільвера-Поліга-Хеллмана для групи порядку $p_i^{e_i}$

Ми записуємо невідомий дискретний логарифм у вигляді розкладу за основою p:

$$k_i = a_0 + a_1 p_i + a_2 p_i^2 + \cdots + a_{e_i-1} p_i^{e_i-1}, \quad 0 \le a_j < p_i.$$

Ключова ідея: Обчислювати послідовно коефіцієнти $a_0, a_1, \ldots, a_{e_i-1}$ за допомогою «покрокового підняття» (Hensel lifting):

$$\begin{aligned} Q_i &= [k_i]P_i| \times p_i^{e_i-1} \\ [n/p_i]Q &= [a_0]([n/p_i]P) + [a_1]([n]P) + [a_2]([np_i]P) + \cdots + [a_{e_i-1}]([np_i^{e_i-2}]P) \\ &= [a_0]([n/p_i]P) \end{aligned}$$

Оскільки $\mathrm{ord}(P)|n$. Зазначимо що $\mathrm{ord}([n/p_i]P) = p_i$, тобто шукаємо a_0 за алгоритмами ρ-Поларда або BSGS

Алгоритм Сільвера-Поліга-Хеллмана для групи порядку $p_i^{e_i}$

Помножимо наш вираз для дискретного логарифму на $p_i^{e_i-2}$:

$$
\begin{aligned}
Q_i &= [k_i]P_i \times p_i^{e_i-2} \\
[n/p_i]Q &= [a_0]([n/p_i^2]P) + [a_1]([n/p_i]P) + [a_2]([n]P) + \cdots + [a_{e_i-1}]([np_i^{e_i-3}]P \\
&= [a_0]([n/p_i]P) + [a_1]([n/p_i]P) \\
[n/p_i]Q - [a_0]([n/p_i]P) &= [a_1]([n/p_i]P)
\end{aligned}
$$

Де $\operatorname{ord}([n/p_i]P) = p_i$, тобто знову отримали задачу дискретного логарифму в підгрупі простого порядку. Знаходимо a_1 за допомогою відомих методів.
Ітеративно знаходимо всі наступні a_j, $\quad j = 2..e_i - 1$ шляхом множення виразу $Q_i = [k_i]P_i$ на $p_i^{e_i-j-1}$ та подальшого вираження a_j як дискретного логарифму в групі порядку p_i із врахуванням всіх знайдених $a_0, \ldots, a_{j-1}$.

Відновлення дискретного логарифму в групі G

Якщо порядок групи n має наступний вираз

$$n = p_1^{e1} \cdots p_s^{e_s},$$

то для кожного простого p_i за алгоритмом, описаним вище, знаходять:

$$k \equiv k_i \pmod{p_i^{e_i}}.$$

Потім, використовуючи Китайську теорему лишків, отримують єдиний розв'язок k за модулем n.

Складність: $O(\sum_{i=1..s}(e_i \sqrt{p_i}))$ за умови використання ρ-алгоритму в групах простого порядку.

Квантовий алгоритм Шора

Основна ідея: Використання квантового алгоритму для ефективного розв'язання дискретного логарифму в абелевих групах, зокрема на еліптичних кривих, за допомогою перетворення Фур'є.

Ключові кроки:

1. Створення суперпозиції всіх можливих експонент.
2. Побудова функції $f(a, b) = [a]P + [b]Q$ та обчислення її квантового стану.
3. Застосування квантового перетворення Фур'є для отримання інформації про періодичність функції.
4. Класичне постпроцесування з отриманих вимірювань для відновлення дискретного логарифму k.

Оцінка часу: Квантовий алгоритм Шора працює з поліноміальною кількістю квантових вентилів, точніше $O(\mathrm{poly}(\log n))$, що є експоненційним прискоренням порівняно з класичними алгоритмами.

Підсумки

Всі розглянуті алгоритми в класичній моделі обчислень мають експоненційну складність, проте в квантовій моделі поліноміальну.

- Наївний алгоритм для групи порядку n: $O(n)$ — експоненційно повільний.
- BSGS для групи порядку n: $O(\sqrt{n})$ як за часом, так і за пам'яттю.
- Алгоритм Поларда для групи простого порядку p: $O(\sqrt{p})$ за часом з малою константою та $O(1)$ за пам'яттю.
- Алгоритм Сільвера-Поліга-Хелмана для групи порядку $n = p_1^{e1} \cdots p_s^{e_s}$: $O(\sum_{i=1..s}(e_i \sqrt{p_i}))$ за часом та $O(s)$ за пам'яттю за умови використання ρ-алгоритму в групах простого порядку.
- Квантовий алгоритм Шора: $O(\mathsf{poly}(\log n))$ — поліноміальна складність квантової схеми, але потребує практичного квантового комп'ютера.