

## 32 Дискретное логарифмирование

### 32.1 Метод больших-малых шагов

Пусть  $G$  — циклическая группа, порожденная элементом  $g$ , т. е.  $G = \langle g \rangle$ . Будем считать, что  $G$  — группа порядка  $q$ , т. е.  $\text{ord } g = q$ .

Нас будут интересовать методы решения уравнения

$$g^x = y, \quad y \in G,$$

относительно  $x \in \{0, 1, \dots, q-1\}$ , т. е. методы определения дискретного логарифма (индекса)  $\log_g y$ .

**Пример 32.1.** В схеме ЭльГамала  $G$  — подгруппа  $\mathbb{F}_p^*$ ,  $\langle p, g, b \rangle$  — открытый ключ,  $\langle a \rangle$  — личный ключ и задача дискретного логарифмирования превращается в задачу определения личного ключа по открытому.  $\square$

Пусть  $m = \lceil \sqrt{q} \rceil$ . Метод больших-малых шагов состоит в отыскании совпадения элемента последовательности

$$1, g, g^2, \dots, g^{m-1},$$

с элементом последовательности

$$y, yg^{-m}, yg^{-2m}, \dots, yg^{-(m-1)m}.$$

Если найдено совпадение  $g^j = yg^{-im}$ , то  $g^{im+j} = y$  и  $\log_g y = (im + j) \bmod q$ .

---

#### АЛГОРИТМ БОЛЬШИХ — МАЛЫХ ШАГОВ

---

*Вход:* (описание  $G, g, y$ ).

*Выход:*  $\log_g y$ .

*Шаги алгоритма:*

1. Построить массив пар  $(j, g^j)$ ,  $j = 0, 1, \dots, m-1$ . Отсортировать пары по второму элементу.
2. Установить  $c \leftarrow y$ .
3. Для  $i = 0, \dots, m-1$  выполнить
  - а) искать совпадение  $c \stackrel{?}{=} g^j$  в массиве;
  - б) если найдено совпадение  $c = g^j$ , то вернуть  $im + j$ ;
  - в)  $c \leftarrow c \cdot g^{-m}$ .

*Сложность.* Требуется выполнить  $O(\sqrt{q})$  групповых операций и  $O(\sqrt{q} \log q)$  сравнений элементов группы при сортировке на шаге 1. Если одна групповая операция является более трудоемкой, чем  $\log q$  сравнений, то сложность алгоритма —  $O(\sqrt{q})$  групповых операций.

---

### 32.2 $\rho$ -метод

Мы уже знакомы с  $\rho$ -методом факторизации. Рассмотрим теперь  $\rho$ -метод логарифмирования, который был предложен Поллардом в 1978 году.

**Идея алгоритма.** Проведем следующие построения:

1. Разобьем  $G$  на подмножества  $G_1, G_2$  и  $G_3$  примерно равной мощности.
2. Построим функцию  $\varphi: G \rightarrow G$ ,

$$\varphi(z) = \begin{cases} yz, & z \in G_1, \\ z^2, & z \in G_2, \\ gz, & z \in G_3. \end{cases}$$

3. Выберем  $z_0 = 1$  (1 — единица  $G$ ) и построим последовательность  $z_t = \varphi(z_{t-1})$ ,  $t = 1, 2, \dots$

Все элементы последовательности имеют вид  $z_t = g^{u_t} y^{v_t}$ . Если мы нашли совпадение  $z_t = z_\tau$ ,  $t < \tau$ , то

$$g^{u_t} y^{v_t} = g^{u_\tau} y^{v_\tau} \Rightarrow y^{v_t - v_\tau} = g^{u_\tau - u_t} \Rightarrow (v_t - v_\tau) \log_g y \equiv (u_\tau - u_t) \pmod{q}.$$

Таким образом, если число  $(v_t - v_\tau)$  обратимо по модулю  $q$ , то

$$\log_g y = (u_\tau - u_t)(v_t - v_\tau)^{-1} \pmod{q}.$$

**Шаги алгоритма.** Для вычисления дискретного логарифма требуется определить элементы последовательности  $(z_t)$  и найти коллизию  $z_t = z_\tau$  (можно воспользоваться алгоритмом Брента). Кроме этого, требуется знать числа  $u_t, v_t$ . Числа можно определять по следующим правилам:

$$(u_t, v_t) = \begin{cases} (u_{t-1}, v_{t-1} + 1) \pmod{q}, & z_{t-1} \in G_1, \\ (2u_{t-1}, 2v_{t-1}) \pmod{q}, & z_{t-1} \in G_2, \\ (u_{t-1} + 1, v_{t-1}) \pmod{q}, & z_{t-1} \in G_3. \end{cases}$$

**Сложность алгоритма.** Для определения дискретного логарифма требуется вычислить  $O(\sqrt{q})$  элементов последовательности  $(z_t)$ , т. е. выполнить  $O(\sqrt{q})$  групповых операций (сложения при определении последовательностей  $(u_t), (v_t)$  менее трудоемки, чем групповые операции).

**Пример 32.2.** На сегодняшний день все серьезные достижения по решению задачи дискретного логарифмирования в группах точек эллиптических кривых (см. следующие лекции) получены с помощью  $\rho$ -метода. Компания Certicom в 1997 году объявила конкурсные ECDLP различной степени сложности. На сегодняшний день решено 10 задач из списка Certicom. Рекордное достижение — логарифмирование в группе, порядок которой является числом из 109 двоичных разрядов. В октябре 2009 года был начат эксперимент по дискретному логарифмированию в группе точек эллиптической кривой над полем  $\mathbb{F}_{2^{131}}$ . Порядок целевой группы является числом из 130 двоичных разрядов. Эксперимент продолжается. С его промежуточными результатами можно ознакомиться в Интернет по адресу <http://ecc-challenge.info>.  $\square$

### 32.3 Метод Поллига — Хеллмана

Пусть  $q = q_1 q_2$ ,  $q_i > 1$ . Введем в рассмотрение элементы  $g_1 = g^{q_2}$ ,  $g_2 = g^{q_1}$  и пусть  $G_1 = \langle g_1 \rangle$ ,  $G_2 = \langle g_2 \rangle$ . Имеем:  $\text{ord } g_i = q_i$  и  $|G_i| = q_i$ .

Будем искать решение уравнения  $g^x = y$  в виде

$$x = x_2 q_1 + x_1, \quad 0 \leq x_2 < q_2, \quad 0 \leq x_1 < q_1.$$

Если  $g^{x_2 q_1 + x_1} = y$ , то

$$(g^{x_2 q_1 + x_1})^{q_2} = y^{q_2} \Rightarrow g^{x_1 q_2} = b^{q_2} \Rightarrow g_1^{x_1} = b^{q_2}.$$

Поэтому можно поступить следующим образом:

1. Найти  $x_1 = \log_{g_1} y^{q_2}$  в группе  $G_1$ .
2. Найти  $x_2 = \log_{g_2} y g^{-x_1}$  в группе  $G_2$ .
3. Определить  $x = x_2 q_1 + x_1$ .

Таким образом, для дискретного логарифмирования в  $G$  требуется выполнить логарифмирование в группах  $G_1$  и  $G_2$  меньшего порядка и использовать несложные дополнительные вычисления.

Если порядок  $G_i$  не является простым числом, то мы снова можем заменить логарифмирование в  $G_i$  на логарифмирование в меньших группах и так далее. При достижении групп простого порядка можно использовать метод больших-малых шагов.

Если  $|G| = \prod_{i=1}^s q_i^{e_i}$ , то для определения  $\log_g y$  потребуется выполнить

$$O\left(\sum_{i=1}^s \alpha_i \sqrt{q_i}\right)$$

групповых операций.

**Пример 32.3.** В первоначальном варианте своей схемы ЭльГамаль предлагал использовать в качестве  $g$  примитивный элемент  $\mathbb{F}_p^*$ , т. е. элемент порядка  $p - 1$ . Если  $p - 1$  не имеет больших простых делителей, то с помощью метода Поллига — Хеллмана можно достаточно эффективно находить личный ключ  $x = \log_g y$ . Поэтому требование наличия у  $p - 1$  большого простого делителя  $q$  является весьма важным. Кроме этого, вместо примитивного элемента  $g$  можно использовать элемент порядка  $q$ , поскольку сложность логарифмирования при таком переходе уменьшается незначительно.  $\square$

### 32.4 $\lambda$ -метод

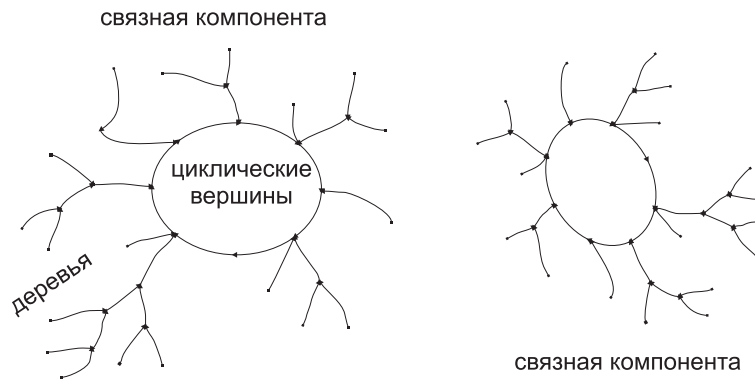
$\lambda$ -метод также был предложен Поллардом. Альтернативное название метода — метод *кенгуру*.

**Идея.** Пусть  $S \subset \{0, 1, \dots, q - 1\}$  и  $h: G \rightarrow S$  — некоторая хэш-функция. Определим преобразование:

$$\varphi: G \rightarrow G, \quad z \mapsto zg^{h(z)}.$$

Поставим в соответствие  $\varphi$  *граф*, вершинами которого являются всевозможные элементы  $G$ : из каждой вершины  $z \in G$  выходит ровно одна дуга, которая заканчивается в  $\varphi(z)$ .

Известно, что граф любого преобразования представляет собой набор *связных компонент*. В свою очередь, каждая связная компонента представляет собой набор циклических вершин, к которым крепятся *деревья*.



Как и  $\rho$ -метод,  $\lambda$ -метод ориентирован на поиск коллизий в графе преобразования  $\varphi$ . В отличие от  $\rho$ -метода, поиск коллизий ведется не только в циклических вершинах графа, но и в вершинах деревьев.

#### Шаги алгоритма.

1. Выбрать  $u_0 \in S$ , вычислить  $z_0 = g^{u_0}$  и построить последовательность

$$z_t = \varphi(z_{t-1}) = z_{t-1}g^{u_t}, \quad u_t = h(z_{t-1}), \quad t = 1, \dots, T.$$

Запомнить  $z_T$  и сумму  $(u_0 + \dots + u_T) \bmod q$ .

2. Выбрать  $z_0^* = y = g^{v_0}$ , где  $v_0$  — неизвестный дискретный логарифм. Построить последовательность

$$z_t^* = \varphi(z_{t-1}^*) = z_{t-1}^*g^{v_t}, \quad v_t = h(z_{t-1}^*), \quad t = 1, 2, \dots,$$

и после определения очередного  $z_t^*$  проверять  $z_t^* \stackrel{?}{=} z_T$ .

3. Если совпадение найдено, то

$$v_0 + v_1 + \dots + v_t \equiv u_0 + u_1 + \dots + u_T \pmod{q} \Rightarrow v_0 = (u_0 + u_1 + \dots + u_T - v_1 - \dots - v_t) \bmod q.$$

Название алгоритма объясняется следующим образом: последовательность  $z_t$  объявляется траекторией *прирученного* кенгуру (мы знаем величины прыжков  $u_0, u_1, \dots$ ), а последовательность  $z_t^*$  считается траекторией *дикого* кенгуру (мы не знаем величину первого прыжка  $v_0$ ).

**Сложность.** Для определения дискретного логарифма также требуется выполнить  $O(\sqrt{q})$  групповых операций.

**Распараллеливание.** Пусть для поиска дискретного логарифма используется не одно вычислительное устройство (машина Тьюринга) а  $m$  устройств. Во сколько раз мы можем уменьшить время поиска?

Оказывается, что для  $\lambda$ -метода время можно уменьшить в  $m$  раз, а для  $\rho$ -метода — только в  $\sqrt{m}$  раз. Суть распараллеливания  $\lambda$ -метода состоит в следующем:

1. Выбрать в  $G$  подмножество *различимых* элементов  $G^*$  такое, что принадлежность  $g \stackrel{?}{\in} G^*$  можно проверить очень быстро (напр., элементы  $G$  кодируются бинарными строками, тогда элементы  $G^*$  — это строки, которые начинаются с определенного префикса).
2. На отдельных машинах вести расчет траекторий ручных и диких кенгуру. Вести общий массив различных элементов, которые достигли кенгуру.
3. Анализировать пересечение траекторий кенгуру в различных точках и, при возможности, определять дискретный логарифм.