Дискретний логарифм

Проблема обчислення дискретного логарифма є не лише цікавою, а й вкрай корисною для систем захисту інформації. Ефективний алгоритм знаходження дискретного логарифму значною мірою знизив би безпеку систем ідентифікації користувача та схеми обміну ключей.

Означення. Нехай G – скінченна циклічна група порядка n. Нехай g – генератор G та $b \in$ G. ***Дискретним логарифмом*** числа b за основою g називається таке число x ($0 \le x \le n - 1$), що $g^x = b$ та позначається $x = \log_g b$.

Проблема дискретного логарифму. Нехай p – просте число, g – генератор множини Z_p^*, $y \in Z_p^*$. Знайти таке значення x ($0 \le x \le p - 2$), що $g^x \equiv y \pmod p$. Число x називається ***дискретним логарифмом*** числа y за основою g та модулем p.

Узагальнена проблема дискретного логарифму. Нехай G – скінченна циклічна група порядка n, g – її генератор, $b \in$ G. Необхідно знайти таке число x ($0 \le x \le n - 1$), що $g^x = b$.

Розширенням узагальненої проблеми може стати задача розв'язку рівняння $g^x = b$, коли знято умову циклічності групи G, а також умову того, що g – генератор G (в такому випадку рівняння може і не мати розв'язку).

Приклад. $g = 3$ є генератором Z_7*: $3^1 = 3$, $3^2 = 2$, $3^3 = 6$, $3^4 = 4$, $3^5 = 5$, $3^6 = 1$.
$\log_3 4 = 4 \pmod 6$, тому що розв'язком рівняння $3^x = 4 \pmod 7$ буде x = 4.

Теорема. Нехай a – генератор скінченної циклічної групи G порядка n. Якщо існує алгоритм, який обчислює дискретний логарифм за основою a, то цей алгоритм може також обчислити дискретний логарифм за будь-якою основою b, яка є генератором G.

Доведення. Нехай $k \in$ G, $x = \log_a k$, $y = \log_b k$, $z = \log_a b$. Тоді $a^x = b^y = (a^z)^y$, звідки $x = zy \bmod n$. Підставимо в останню рівність замість змінних логарифмічні вирази:

$$\log_a k = (\log_a b)\,(\log_b k) \bmod n$$

або

$$\log_b k = (\log_a k)\,(\log_a b)^{-1} \bmod n.$$

З останньої рівності випливає справедливість теореми.

Примітивний алгоритм

Для знаходження $\log_g b$ (g – генератор G порядка n, $b \in$ G) будемо обчислювати значення g, g^2, g^3, g^4, ... поки не отримаємо b. Часова оцінка алгоритму – O(n). Якщо n – велике число, то час обчислення логарифму є достатньо великим і тому алгоритм є неефективним.

Алгоритм великого та малого кроку

Першим детермінованим алгоритмом для обчислення дискретного логарифму був алгоритм великого та малого кроку, запропонований Шанком (Shank) [1].

Для обчислення $\log_g b$ в групі Z_n^* необхідно зробити наступні кроки:

1. Обчислити $a = \lceil \sqrt{n} \rceil$;
2. Побудувати список $L_1 = 1, g^a, g^{2a}, ..., g^{a^2}$ (за модулем n);
3. Побудувати список $L_2 = b, bg, bg^2, ..., bg^{a-1}$ (за модулем n);
4. Знайти число z, яке зустрілося в L_1 та L_2.

Тоді $z = bg^k = g^{la}$ для деяких k та l. Звідси $b = g^{la-k} = g^x$; $x = la - k$.

Два питання постає при дослідженні роботи наведеного алгоритму:

1. Чи завжди знайдеться число, яке буде присутнім в обох списках?
2. Як ефективно знайти значення z?

Запишемо $x = sa + t$ для деяких s, t таких що $0 \le s, t < a$. Тоді $b = g^x = g^{sa+t}$. Домножимо рівність на g^{a-t}, отримаємо: $bg^{a-t} = g^{s(a+1)}$. Значення зліва обов'язково зустрінеться в L_2, а справа – в L_1.

Відсортуємо отримані списки L_1 та L_2 за час $O(a * \log a)$. За лінійний час проглядаємо списки зліва направо порівнюючи їх голови: якщо вони рівні, то значення z знайдене, якщо ні – то видалити менше число і продовжити перевірку.

Приклад. Розв'язати рівняння: $2^x \equiv 11 \pmod{13}$.

$a = \lceil \sqrt{13} \rceil = 4$;

L_1: $1, 2^4 \equiv 3, 2^8 \equiv 9, 2^{12} \equiv 1, 2^{16} \equiv 3$;

L_2: $11, 11 * 2 \equiv 9, 11 * 2^2 \equiv 5, 11 * 2^3 \equiv 10$;

Число 9 зустрілося в обох списках. $11 * 2 \equiv 2^8$, $11 \equiv 2^7$, звідки $x = 7$.

Відповідь: $x = 7$.

Інший підхід до реалізації алгоритму великого та малого кроку можна отримати якщо рівність $b = g^{sa+t}$ ($a = \lceil \sqrt{n} \rceil$, $0 \le s, t < a$) переписати у вигляді $b * (g^{-a})^s = g^t$. Обчислимо g^{-a} та складемо таблицю значень g^t, $0 \le t < a$. Далі починаємо знаходити значення $b * (g^{-a})^s$, $s = 0, 1, \ldots$ перевіряючи їх наявність у таблиці g^t. Як тільки знаходяться такі s та t, алгоритм зупиняється.

Приклад. Обчислити $\log_2 3$ в групі Z_{19}^*.

$3 = 2^x = 2^{sa+1}$, $3 * (2^{-a})^s = 2^t$. Складемо таблицю 2^t, $0 \le t < \lceil \sqrt{19} \rceil = 5$:

t	0	1	2	3	4
2^t	1	2	4	8	16

$2^{-1} \equiv 10 \pmod{19}$, оскільки $2 * 10 \equiv 1 \pmod{19}$.
Тоді $3 * (2^{-5})^s \pmod{19} \equiv 3 * (10^5)^s \pmod{19} \equiv 3 * 3^s \pmod{19}$
Обчислюємо $3 * 3^s$, $s = 0, 1, \ldots$:

s	0	1	2
$3 * 3^s$	3	9	8

Значення 8, яке отримали при s = 2, присутнє в таблиці 2^t, $0 \le t < 5$.
Звідси $3 * (2^{-5})^2 = 2^3$ або $3 = (2^5)^2 * 2^3 = 2^{5*2+3} = 2^{13}$.
Відповідь: $3 = 2^{13}$, тобто $\log_2 3 = 13$.

Алгоритм Полард - ро

Нехай G – циклічна група з модулем n (n – просте). Розіб'ємо елементи групи G на три підмножини S_1, S_2 та S_3, які мають приблизно однакову потужність. При цьому необхідне виконання умови: $1 \notin S_2$. Визначимо послідовність елементів x_i наступним чином:

$$x_0 = 1,\ x_{i+1} = \begin{cases} b \cdot x_i, x_i \in S_1 \\ x_i^2, x_i \in S_2 \\ a \cdot x_i, x_i \in S_3 \end{cases},\ i \ge 0 \quad (1)$$

Ця послідовність у свою чергу утворить дві послідовності c_i та d_i , що задовольняють умові

$$x_i = a^{c_i} b^{d_i}$$

та визначаються наступним чином:

$$c_0 = 0,\ c_{i+1} = \begin{cases} c_i, x_i \in S_1 \\ (2 \cdot c_i) \bmod (n-1), x_i \in S_2 \\ (c_i + 1) \bmod (n-1), x_i \in S_3 \end{cases},\ i \ge 0 \quad (2)$$

та

$$d_0 = 0,\ d_{i+1} = \begin{cases} (d_i + 1) \bmod (n-1), x_i \in S_1 \\ (2 \cdot d_i) \bmod (n-1), x_i \in S_2 \\ d_i, x_i \in S_3 \end{cases},\ i \ge 0 \quad (3)$$

Алгоритм буде працювати циклічно шукаючи таке знчення i, для якого $x_i = x_{2i}$. Для таких значень будуть мати місце рівність $a^{c_i} b^{d_i} = a^{c_{2i}} b^{d_{2i}}$ або $b^{d_i - d_{2i}} = a^{c_{2i} - c_i}$. Логарифмуючи останню рівність за основою a, матимемо:

$$(d_i - d_{2i}) * \log_a b \equiv (c_{2i} - c_i) \bmod (n-1) \quad (3)$$

Якщо $d_i \ne d_{2i} \pmod n$, то це рівняння може бути ефективно розв'язано для обчислення $\log_a b$.

Алгоритм

Вхід: генератор a циклічної групи G з модулем n та елемент $b \in G$.
Вихід: дискретний логарифм $x = \log_a b$.

1. $x_0 \leftarrow 1$, $c_0 \leftarrow 0$, $d_0 \leftarrow 0$.
2. for $i = 1, 2, \ldots$ do
   2.1. За значеннями x_{i-1}, c_{i-1}, d_{i-1} та x_{2i-2}, c_{2i-2}, d_{2i-2} обчислити значення x_i, c_i, d_i та x_{2i}, c_{2i}, d_{2i} використовуючи формули (1), (2), (3).
   2.2. if ($x_i = x_{2i}$) then
   $r \leftarrow (d_i - d_{2i}) \bmod (n-1)$;
   if ($r = 0$) then return (FALSE); // розв'язку не знайдено
   $x \leftarrow r^{-1} (c_{2i} - c_i) \bmod (n-1)$.
   return (x).

Якщо алгоритм завершується невдачею (повертає FALSE), то можна запустити його вибравши інші початкові значення c_0, d_0 з інтервалу [1; n - 1] та поклавши $x_0 = a^{c_0} b^{d_0}$.

Якщо НСД($d_i - d_{2i}$, $n-1$)=d>1, то існує розв'язок x_0 порівняння (3) за модулем $(n-1)/d$. Тоді шуканий логарифм $x = x_0 + m(n-1)/d$, де m може набувати значень 0, 1, 2, … d-1.

Приклад. Обчислити $\log_2 9$ в групі Z_{19}^*.

Нехай $S_k = \{x \in Z_n^* : x \equiv k \pmod 3\}, k = 1,2,3$.

Побудуємо наступну таблицю значень послідовностей x_i, c_i, d_i:

i	x_i	c_i	d_i	x_{2i}	c_{2i}	d_{2i}
1	9	0	1	18	1	1
2	18	1	1	4	4	2
3	17	2	1	4	8	6
4	4	4	2	4	16	14
5	17	4	3	4	14	12
6	4	8	6	4	10	8

На 4 кроці отримали $x_4 = x_8$. Підставивши їх значення, отримаємо:

$$2^4 * 9^2 = 2^{16} * 9^{14} \text{ або } 2^{4-16} = 9^{14-2}, 2^{-12} = 9^{12}$$

Логарифмуємо рівність: $12 * \log_2 9 = -12 \pmod{18}$, оскільки $|Z_{19}^*| = 18$.

Враховуючи що $-12 \pmod{18} \equiv 6$, перепишемо рівність у вигляді

$12 * \log_2 9 = 6 \pmod{18}$. НСД(12,18)=6<>1, тому шукаємо іншу пару.

На 6 кроці отримали $x_6 = x_{12}$. Підставивши їх значення, отримаємо:

$$2^8 * 9^6 = 2^{10} * 9^8 \text{ або } 2^{-2} = 9^2,$$

Логарифмуємо рівність: $2 * \log_2 9 = -2 \pmod{18}$, оскільки $|Z_{19}^*| = 18$.

Перепишемо рівність у вигляді $2 * \log_2 9 = 16 \pmod{18}$ або $\log_2 9 = 8 \pmod 9$.

Тобто якщо рівність можна скоротити так, що не прийдеться шукати обернене за модулем, то це можна робити.

Відповідь: $\log_2 9 = 8$.

Індексний алгоритм

Алгоритм, базований на обчисленні індексів, є найпотужним при обчисленні дискретного логарифму. Необхідно побудувати відносно невелику підмножину S елементів групи G, яка називається ***множниковою основою***. Ця підмножина повинна обиратися таким чином, щоб як можна більша частина елементів G могла бути представлена у вигляді добутку її елементів. При обчисленні значення $\log_a b$ (a – генератор G, $b \in$ G) спочатку обчислюються значення логарифмів елементів з S (які заносяться в тимчасову базу даних), а потім на їх основі обчислюється логарифм числа b.

Алгоритм

Вхід: генератор a циклічної групи G порядка n та елемент $b \in$ G.

Вихід: дискретний логарифм $x = \log_a b$.

1. Побудувати множину S – множникову основу. Нехай S = $\{p_1, p_2, \ldots, p_t\}$. В якості значень p_i можна обрати, наприклад, i - те просте число.

2. Побудувати систему лінійних рівнянь, розв'язком якої будуть значення $\log_a p_i$. Для цього виконаємо наступні кроки:

2.1. Обрати деяке ціле k, $0 \le k \le n$ - 1 та обчислити a^k .

2.2. Спробувати представити значення a^k у вигляді добутку чисел з S:

$$a^k = \prod_{i=1}^{t} p_i^{c_i} \text{ , } c_i \ge 0$$

Якщо така рівність знайдена, то записати рівняння:

$$k = \sum_{i=1}^{t} c_i \log_a p_i \pmod n$$

2.3. Повторювати кроки 2.1. та 2.2. поки не отримаємо $t + c$ лінійних рівнянь. Невелике ціле число c ($1 \le c \le 10$) обирається таким чином, щоб складена система рівнянь мала єдиний розв'язок з великою ймовірністю (якщо скласти лише t рівнянь з t невідомими, то з великою ймовірністю два з цих рівнянь будуть залежними і тоді система буде мати більше одного розв'язку).

3. Розв'язати утворену систему рівнянь, отримати значення $\log_a p_i$, $1 \le i \le t$.

4. Обчислення $\log_a b$.

4.1. Обрати деяке ціле k, $0 \le k \le n$ - 1 та обчислити $b * a^k$.

4.2. Спробувати представити значення $b * a^k$ у вигляді добутку чисел з S:

$$b * a^k = \prod_{i=1}^{t} p_i^{d_i} \text{ , } d_i \ge 0$$

Якщо такого представлення знайти не вдається, виконати знову 4.1. Інакше прологарифмірувавши останню рівність, отримаємо:

$$x = \log_a b = (\sum_{i=1}^{t} d_i \log_a p_i - k) \bmod n$$

Приклад. Обчислити $\log_2 12$ в групі Z_{19}^*.

1. Нехай S = {2, 3, 5} – множникова основа.

2. Будуємо систему рівнянь для знаходження значень $\log_2 p_i$, де $p_i \in S$. Оскільки множина S містить 3 елементи, то достатньо отримати 3 лінійно незалежні рівняння.

$k = 5$: $2^5 \pmod{19} \equiv 13$ – не представимо у вигляді добутку чисел з S.

$k = 7$: $2^7 \pmod{19} \equiv 14$ – не представимо у вигляді добутку чисел з S.

$k = 2$: $2^2 \pmod{19} \equiv 4 = 2^2$. Перше рівняння: $2 = 2\log_2 2$.

$k = 10$: $2^{10} \pmod{19} \equiv 17$ – не представимо у вигляді добутку чисел з S.

$k = 15$: $2^{15} \pmod{19} \equiv 12 = 2^2 * 3$. Друге рівняння: $15 = 2\log_2 2 + \log_2 3$.

$k = 11$: $2^{11} \pmod{19} \equiv 15 = 3 * 5$. Третє рівняння: $11 = \log_2 3 + \log_2 5$.

3. Система рівнянь за модулем 18 (порядок Z_{19}^* дорівнює 18) має вигляд:

$$\begin{cases} 2 = 2\log_2 2 \pmod{18} \\ 15 = 2\log_2 2 + \log_2 3 \pmod{18} \\ 11 = \log_2 3 + \log_2 5 \pmod{18} \end{cases}$$

Її розв'язком буде:

$$\log_2 2 = 1,\ \log_2 3 = 13,\ \log_2 5 = 16$$

4. Обчислення $\log_2 12$.

$k = 3$: $12 * 2^3 \pmod{19} \equiv 1$ – не представимо у вигляді добутку чисел з S.

$k = 7$: $12 * 2^7 \pmod{19} \equiv 16 = 2^4$.

$\log_2 12 + 7 \equiv 4\log_2 2 \pmod{18}$, $\log_2 12 \equiv (4\log_2 2 - 7) \pmod{18} = 15$.

Відповідь: $\log_2 12 = 15$.

<u>Алгоритм Поліга – Хелмана</u>

Алгоритм Поліга – Хелмана ефективно розв'язує задачу дискретного логарифма в групі G порядка n, якщо число n має лише малі прості дільники.

Нехай $g, h \in G$, $|G| = p^s$, p – просте. Тоді значення $x = \log_g h$ можна подати у вигляді:

$$x = x_0 + x_1 p + x_2 p^2 + \ldots + x_{s-1} p^{s-1}$$

Піднесемо рівняння $h = g^x$ до степеня p^{s-1}:

$$h^{p^{s-1}} = \left(g^{p^{s-1}}\right)^x = \left(g^{p^{s-1}}\right)^{x_0 + x_1 p + x_2 p^2 + \ldots + x_{s-1} p^{s-1}} =$$

$$\left(g^{p^{s-1}}\right)^{x_0} * \left(g^{p^s}\right)^{x_1} * \left(g^{p^s}\right)^{p x_2} * \ldots * \left(g^{p^s}\right)^{p^{s-2} x_{s-1}} = \left(g^{p^{s-1}}\right)^{x_0},$$

оскільки $g^{p^s} = 1$ (g – генератор групи, p^s – її порядок).

Таким чином з рівності $h^{p^{s-1}} = \left(g^{p^{s-1}}\right)^{x_0}$ знаходимо x_0.

Далі маючи значення $x_0, x_1, \ldots, x_{i-1}$ можна обчислити x_i з рівняння

$$\left(h \cdot g^{-\sum_{j=0}^{i-1} x_j p^j}\right)^{p^{s-(i+1)}} = \left(g^{p^{s-1}}\right)^{x_i}$$

Приклад. Обчислити $\log_3 7$ в Z_{17}^*.

Необхідно розв'язати рівняння $3^x = 7$ в групі, порядок якої дорівнює $16 = 2^4$.

Представимо x у двійковій системі числення: $x = x_0 + 2x_1 + 4x_2 + 8x_3$.

1. Обчислення x_0.

Піднесемо рівняння $3^x = 7$ до степеня $2^3 = 8$:

$3^{8(x_0+2x_1+4x_2+8_3)} = 7^8$, $3^{8x_0+16x_1+32x_2+64_3} = -1$,

$3^{8x_0} * (3^{16})^{x_1} * (3^{16})^{2x_2} * (3^{16})^{4x_3} = -1$.

Оскільки $3^{16} \pmod{17} \equiv 1$, то останнє рівняння прийме вигляд $3^{8x_0} = -1$.

Враховуючи що $3^8 \pmod{17} \equiv -1$, маємо: $(-1)^{x_0} = -1$, $x_0 = 1$.

2. Обчислення x_1.

Домножимо рівність $3^{x_0+2x_1+4x_2+8_3} = 7$ на $3^{-x_0} = 3^{-1} \pmod{17} = 6$, отримаємо:

$3^{2x_1+4x_2+8_3} = 7 * 6$ або $3^{2x_1+4x_2+8_3} = 8$.

Піднесемо рівняння до степеня 4: $3^{8x_1+16x_2+32_3} = 8^4$, $3^{8x_1} = -1$, $x_1 = 1$.

3. Обчислення x_2.

1. D. Shanks. Class number, a theory of factorization and genera. In Proc. Symposium Pure Mathematics, vol.20, pp.415-440. American Mathematical Society, 1970.