Шанксов и Полард-ро алгоритам - Теореме о колизији

Владимир Божовић

Одсјек за математику
Природно-математички факултет

КРИПТОГРАФИЈА

Комплексност ДЛП

- Рјешавамо оригинални ДЛП, $g^x = h$ у $\mathbb{F}_p$.

Trial-and-error метод

Ако је прост број p изабран између 2^k и 2^{k+1}, тада су g, h, и p највише k битова дужине, тако да је величина улазних података $O(k)$ битова. (Примјетимо да је $O(k)$ исто што и $O(\log_2(p))$. Ако покушавамо ријешити ДЛП примјењујући trial-and-error метод, то захтјева $O(p)$ корака за његово рјешавање. Пошто је $O(p) = O(2^k)$, ово је алгоритам са експоненцијално временом извршења.

Шанксов Baby step-Giant step алгоритам

Шанксов алгоритам је заснован на идеји формирања двије листе и тражењу елемента који се налази у обје.

- Нека G група и нека је $g \in G$ елемент реда $N \geq 2$. Следећи алгоритам рјешава $g^x = h$ дискретни логаритам.
  (1) Нека је $n = 1 + \lfloor \sqrt{N} \rfloor$, што посебно значи $n > \sqrt{N}$.
  (2) Формирамо двије листе,

  Листа 1. $e, g, g^2, \ldots, g^n$,
  Листа 2. $h, hg^{-n}, hg^{-2n}, \ldots, hg^{-n^2}$.

  (3) Наћи заједнички елемент у двије листе, рецимо $g^i = hg^{-jn}$.
  (4) Тада је $x = i + jn$ рјешење једначине (ДЛП) $g^x = h$.

Комплексност Шанксовог алгоритма

Комплексност овог алгоритма се мјери кроз три корака: формирања листа, њиховог сортирања и на крају, проналажења заједничког елемента двије листе.

1. Формирање двије листе "кошта" $2n$ мултипликација, па је то $O(n)$ корака.
2. Сортирање листе према најбржем алгоритму сортирања је $O(n\log(n))$.
3. Испитивање да ли је неки елемент листе 2 унутар листе 1 се може обавити у $O(\log(n))$ корака, а како испитујемо редом све елементе то је комплексност $O(n\log(n))$.

Дакле, сабирањем ове три величине, видимо да је комплексност $O(n\log(n))$.

Теорема о колизији

- Претпоставимо да кутија садржи N бројева. Боб бира n различитих бројева из те кутије и поставља их у листу. Потом, врати све у кутију и прави другу листу (са враћањем) од m бројева.
- Важна и интересантна чињеница је да ако су n и m мало већи од $\sqrt{N}$, онда је веома вјероватно да тако формиране листе имају заједнички елемент.

Теорема о колизији

Теорема

Кутија садржи N лоптица од којих су n црвене, а остале су плаве. Боб извлачи са враћањем m лоптица.

a. *Вјероватноћа да Боб извуче бар једну црвену лоптицу је*

$$P(\textit{најмање једна црвена}) = 1 - \left(1 - \frac{n}{N}\right)^m.$$

б.

$$P(\textit{најмање једна црвена}) \geq 1 - e^{-mn/N}.$$

Теорема о колизији

Proof.

a.

$$
\begin{aligned}
P(\text{најмање једна црвена}) &= 1 - \prod_{i=1}^{m} P(\text{i-та је плава}) \\
&= 1 - \prod_{i=1}^{m} \left(\frac{N-n}{N} \right) \\
&= 1 - \left(1 - \frac{n}{N} \right)^{m}
\end{aligned}
$$

□

Теорема о колизији

Proof.

b. Користимо познату неједнакост

$$e^{-x} \geq 1 - x,\ x \in \mathbb{R},$$

па слиједи

$$1 - \left(1 - \frac{n}{N}\right)^m \geq 1 - \left(e^{-n/N}\right)^m = 1 - e^{-mn/N},$$

чиме је доказана тражена неједнакост. □

Теорема о колизији

- Повежимо ове резултате са постављеним задатком. Претпоставимо имамо кутију са N нумерисаних, плавих куглица. Боб бира n различитих куглица из те кутије, запамти избор, а потом их префарба у црвено и враћа у кутију.
- Извлачи потом другу листу (са враћањем) од m куглица. Јасно је да је вјероватноћа да изабере најмање једну црвену лоптицу иста као вјероватноћа да ће двије формиране листе бројева имати заједнички елемент.

Теорема о колизији

Примјер

Кутија садржи N нумерисаних куглица. Боб бира n различитих куглица из те кутије, запамти избор, а потом их враћа у кутију. Потом извлачи другу листу од n куглица. Колико мора бити велико n да би вјероватноћа да ће двије формиране листе бројева имати заједнички елемент (непразан пресјек) била већа од 50 %? А за 99.99 %?

$$P(\text{непразан пресјек}) \geq 1 - e^{-n^2/N} = 0.5 \implies n \approx 0.83\sqrt{N}.$$

$$P(\text{непразан пресјек}) \geq 1 - e^{-n^2/N} = 0.9999 \implies n \approx 3.035\sqrt{N}.$$

Импликације теореме о колизији

- Нека је функција $f : S \to T$, таква да је $|T| < |S| = N$. Рођендански парадокс нам каже да је потребно направити листу од отприлике $\sqrt{N}$ елемената $f(z)$, прије него што се уочи "колизија".
- Овај закључак је нарочито важан за оцјену сигурности криптографских хеш функција.

Пробабилистичка варијација Шанксовог алгоритма

Теорема

Нека је G група и $h \in G$ елемент реда N. Уз претпоставку да ДЛОГ проблем

$$h^x = b$$

има рјешење, онда се рјешење може наћи у $\mathcal{O}(n \log(n))$ корака, гдје је $n = \sqrt{N}$.

Ако посматрамо $x = y - z$, онда је $h^y = b \cdot h^z$. Дакле, идеја је да се направе двије листе

$$h^{y_1}, h^{y_2}, \ldots, h^{y_n} \text{ и } b \cdot h^{z_1}, b \cdot h^{z_2}, \ldots, b \cdot h^{z_n},$$

гдје су експоненти бирани на случајан начин.

Пробабилистичка варијација Шанксовог алгоритма

- Примјетимо да обје листе припадају групи $H =< h >$, која има N елемената. На основу претходно дате анализе, вјероватноћа да ове листе

$$h^{y_1}, h^{y_2}, \ldots, h^{y_n} \text{ и } b \cdot h^{z_1}, b \cdot h^{z_2}, \ldots, b \cdot h^{z_n}$$

имају непразан пресјек је приближно $1 - e^{-n^2/N}$.

- Стога, ако је $n \approx 3\sqrt{N}$, онда је вјероватноћа непразног пресјека приближно 99.98%

- Проблем са претходним приступом је у томе што ови алгоритми који се заснивају на колизији (Шанксов или пробабилистички) захтјевају озбиљне меморијске капацитете.
- Насупрот томе, Џон Полард је предложио веома лијеп алгоритам који малтене не захтјева никакву меморију по цијени од мало више рачунања него раније.
- Оригинално, овај алгоритам се користио за тражење просте факторизације броја, али се сасвим лијепо прилагођава тражењу ДЛОГ-а.

Апстрактна формулација Полард-ро метода

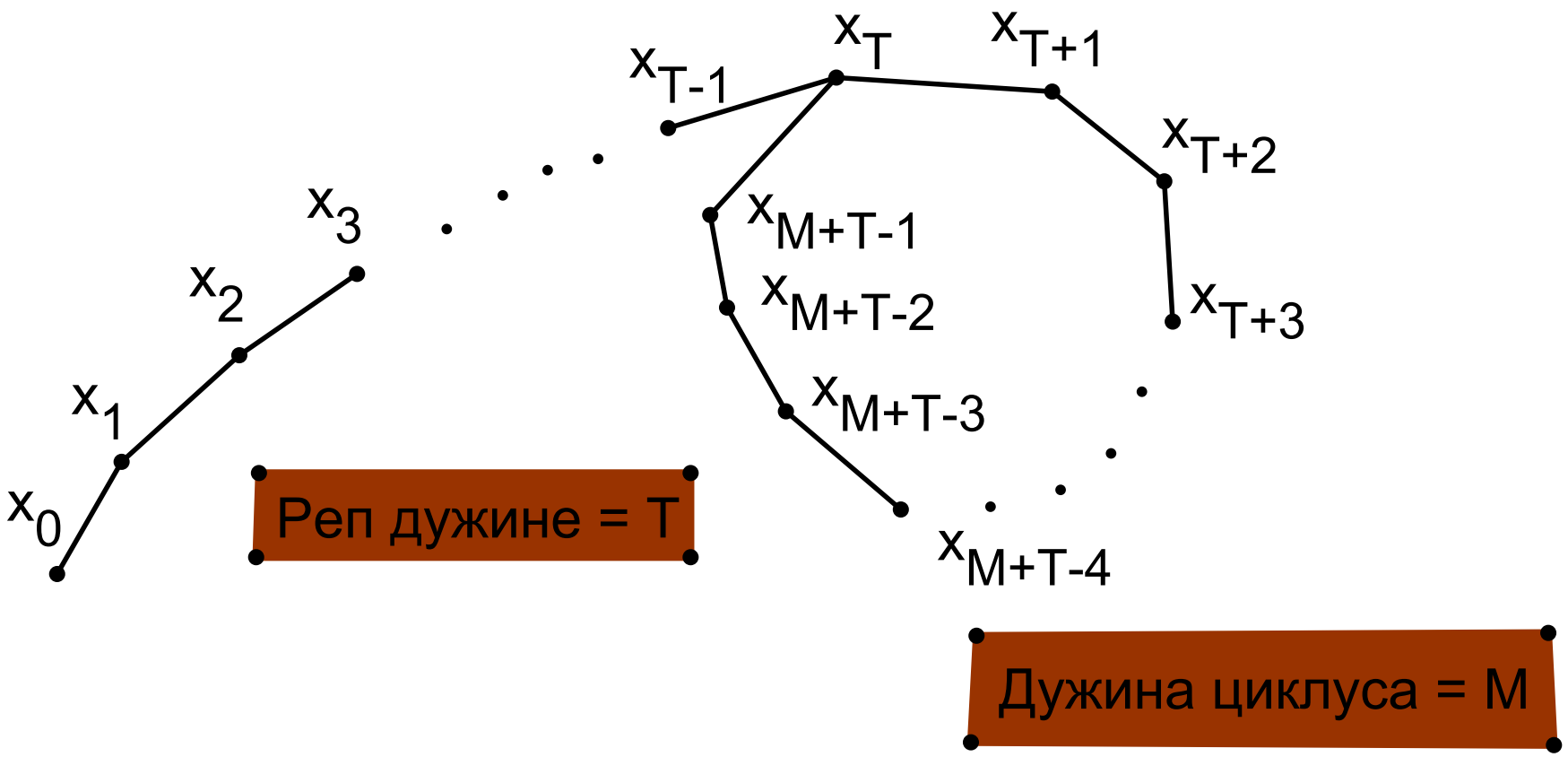
Почињемо са уопштеним, апстрактним уводом за Полард-ро метод. Нека је S коначан скуп и нека је $f : S \to S$ функција. За неко почетно $x \in S$, формирамо низ

$$x_0 = x,\ x_1 = f(x_0),\ x_2 = f(x_1), \ldots$$

Другим ријечима, чланови низа су дефинисани на следећи начин

$$x_i = (\underbrace{f \circ f \circ f \circ f \cdots f}_{i \text{ итерација } f})(x).$$

Претходни низ (x_i) називамо *орбита* од x од функције f и означавамо га са $O_f^+(x)$.


x_0
x_1
x_2
x_3
x_{T-1}
x_T
x_{T+1}
x_{T+2}
x_{T+3}
x_{M+T-1}
x_{M+T-2}
x_{M+T-3}
x_{M+T-4}
Реп дужине = T
Дужина циклуса = M

Апстрактна формулација Полард-ро метода

Полард је интелигентно увео у "игру" и други низ, који ће, користећи рођендански парадокс, произвести потребну колизију. Увео је низ дефинисан на следећи начин

$$y_0 = x_0 \text{ и } y_{i+1} = f(f(y_i)), \text{ за } i = 0, 1, 2, \ldots$$

Није тешко уочити да је

$$y_i = x_{2i}.$$

Апстрактна формулација Полард-ро метода

Питање је, колико дуго треба да се деси колизија $x_{2i} = x_i$? Закључујемо, да за $j > i$, имамо

$$x_j = x_i \text{ ако и само ако } i \geq T, \text{ и } j \equiv i \pmod{M}.$$

Слиједи

$$x_{2i} = x_i \text{ ако и само ако } i \geq T \text{ и } 2i \equiv i \pmod{M}.$$

Последњи услов је еквивалентан са $M \mid i$. Како је један од бројева $T, T+1, \ldots, T+M-1$ дјељив са M, онда слиједи да је

$$x_{2i} = x_i \text{ за неко } 1 \leq i \leq T+M.$$

Полардова теорема

Нека је S коначан скуп са N елемената, а $f : S \to S$ пресликавање. Нека је x тачка скупа S.

1. За дату орбиту $O_f^+ = \{x_0, x_1, x_2, \ldots\}$ тачке x, дужине репа T и обима циклуса M, вриједи

$$x_{2i} = x_i \text{ за неко } 1 \leq i \leq T + M.$$

2. Ако је f довољно "случајна", онда је очекивана вриједност $T + M$ једнака

$$E(T + M) \approx 1.2533 \cdot \sqrt{N}$$

Подсјећање: Рјешавање линеарних конгруенција

Конгруенција

$$ax \equiv b \pmod{m}$$

има рјешења ако и само ако $d \mid b$ гдје је **нзд**$(a, m) = d$.
Рјешења добијамо у три корака:

1. Скрати се претходна конгруенција са d, па имамо

$$a'x' \equiv b' \pmod{m'}, \text{ одакле је } \textbf{нзд}(a', m') = 1.$$

2. Нека је x_0 јединствено рјешење претходне конгруенције (Еуклидов алгоритам).
3. Сва рјешења почетне конгруенције су дата са

$$x_0 + k\frac{m}{d}, \quad k = 0, 1, \ldots, d-1.$$

ДЛОГ Полард-ро методом

Нека је дат ДЛОГ

$$g^t = a \quad \text{у} \quad \mathbb{Z}_p.$$

Основна идеја је да се нађе нека "колизија" између $g^i a^j$ и $g^k a^\ell$ за неке експоненте i, j, k, ℓ. Онда се из једнакости

$$g^{i-k} = a^{\ell-j}$$

коначно рјешење добија из одговарајућих рјешења једначине

$$i - k \equiv t(\ell - j) \pmod{p-1}.$$

ДЛОГ Полард-ро методом

Полард предлаже функцију

$$f(x) = \begin{cases} gx & \text{if } 0 \leq x < p/3 \\ x^2 & \text{if } p/3 \leq x < 2p/3 \\ ax & \text{if } 2p/3 \leq x < p. \end{cases}$$

Стартна тачка је $x_0 = 1$, што ће рећи да је

$$x_i = (\underbrace{f \circ f \circ f \circ f \cdots f}_{i \text{ итерација } f})(1) = g^{\alpha_i} a^{\beta_i}.$$

Сходно реченом, јасно је да је $\alpha_0 = \beta_0 = 0$.

ДЛОГ Полард-ро методом

Пратећи функцију f можемо рачунати одговарајуће експоненте у $g^{\alpha_i} a^{\beta_i}$ на следећи начин

$$\alpha_{i+1} = \begin{cases} \alpha_i + 1 & \text{if } 0 \leq x < p/3 \\ 2\alpha_i & \text{if } p/3 \leq x < 2p/3 \\ \alpha_i & \text{if } 2p/3 \leq x < p, \end{cases}$$

$$\beta_{i+1} = \begin{cases} \beta_i & \text{if } 0 \leq x < p/3 \\ 2\beta_i & \text{if } p/3 \leq x < 2p/3 \\ \beta_i + 1 & \text{if } 2p/3 \leq x < p. \end{cases}$$

ДЛОГ Полард-ро методом

Слично, формирамо и други низ на следећи начин

$$y_0 = 1 \text{ и } y_{i+1} = f(f(y_i)),$$

одакле је

$$y_i = x_{2i} = g^{\gamma_i} \cdot a^{\delta_i}.$$

Експоненти γ_i, δ_i се рачунају на основу већ дате рекурентне формуле, само што се иста примјењује сваки пут дупло. Користећи назначену процедуру, ми у једном тренутку налазимо тражену колизију и добијамо

$$g^{\alpha_i} a^{\beta_i} = g^{\gamma_i} a^{\delta_i}.$$

ДЛОГ Полард-ро методом

Ако је

$$u \equiv \alpha_i - \gamma_i \pmod{p-1} \text{ и } v \equiv \delta_i - \beta_i \pmod{p-1},$$

онда

$$g^u = a^v, \text{ односно } v \cdot t \equiv u \pmod{p-1}.$$

Рјешења претходне конгруенције (у којој је непознато t, а познато све остало) ће дати рјешење почетног ДЛОГ-а:

$$g^t = a.$$

Полард-ро примјер

Покушавамо да ријешимо ДЛОГ

$$19^t \equiv 24717 \pmod{48611}.$$

$$f(x) = \begin{cases} gx & \text{if } 0 \leq x < p/3 \\ x^2 & \text{if } p/3 \leq x < 2p/3 \\ ax & \text{if } 2p/3 \leq x < p. \end{cases}$$

$$x_0 = 1,\ y_0 = 1,\ x_i = f(x_{i-1}),\ y_i = f(f(y_{i-1})).$$

$x_1 = f(1) = 19$, $y_1 = f(f(1)) = f(19) = 361$. Истовремено, пратимо експоненте

$$\alpha_1 = 1,\ \beta = 0, \gamma_1 = 2, \delta = 0.$$

Полард-ро примјер

i	x_i	$y_i = x_{2i}$	α_i	β_i	γ_i	δ_i
0	1	1	0	0	0	0
1	19	361	1	0	2	0
2	361	33099	2	0	4	0
3	6859	13523	3	0	4	2
4	33099	20703	4	0	6	2
5	33464	14974	4	1	13	4
6	13523	18931	4	2	14	5
7	13882	30726	5	2	56	20
8	20703	1000	6	2	113	40
9	11022	14714	12	4	228	80
			$\vdots$			
542	21034	46993	13669	2519	27258	30257
543	20445	37138	27338	5038	27259	30258
544	40647	33210	6066	10076	5908	11908
545	28362	21034	6066	10077	5909	11909
546	36827	40647	12132	20154	23636	47636
547	11984	36827	12132	20155	47272	46664
548	33252	33252	12133	20155	47273	46665

Полард-ро примјер

$$\alpha_{548} = 12133,\ \beta_{548} = 20155,\ \gamma_{548} = 47273,\ \delta_{548} = 46665.$$

$$19^{12133} \cdot 24717^{20155} = 19^{47273} \cdot 24717^{46665}.$$

$$19^{13470} = 24717^{26510}.$$

$$26510x \equiv 13470 \pmod{48610}$$

Како је **нзд** $26510, 48610 = 10$ и 10 дијели 13470, онда постоје рјешења ове конгруенције

3842, 8703, 13564, 18425, 23286, 28147, 33008, 37869, 42730, 47591.

Пробањем редом рјешења, добијамо да је 37869 тражено рјешење.