MI-PB-8

Diskrétní logaritmus – Diffie-Hellman, ElGamal, algoritmy Babystep-giantstep, Pollardova rho metoda, Pohlig-Hellman, Index calculus.

Diskrétní logaritmus:
G grupa, $g, h \in G$. Pokud existuje $x \in \mathbb{Z}$ t.ž. $g^x = h$, pak x je logaritmus h o základu g: $x = \log_g h$.

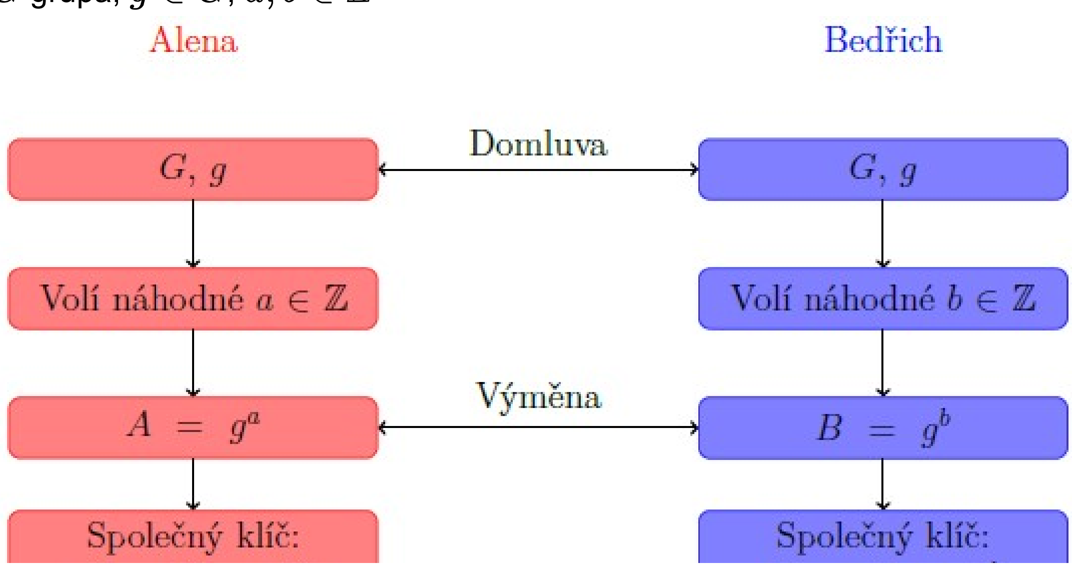
Problém diskrétního logaritmu: problém hledání x t.ž. $g^x = h$.

Útok hrubou silou: G grupa, $g, h \in G$, $\mathrm{ord}(g) = N$. Pokud existuje diskrétní logaritmus $\log_g h$, lze jej nalézt v $O(N)$ krocích.
Obtížnost řešení $x = \log_g h$:

- nezávisí na g
- závisí za G:
  - $g, h \in \mathbb{Z}_{p-1}^{+} \Rightarrow x \cdot g \equiv h \pmod{p-1} \Rightarrow$ Euklidův algoritmus
  - $g, h \in \mathbb{Z}_{p}^{\times} \Rightarrow g^x \equiv h \pmod{p} \Rightarrow$ nutná hrubá síla

Diffieho-Hellmanova výměna klíčů

G grupa, $g \in G$, $a, b \in \mathbb{Z}$



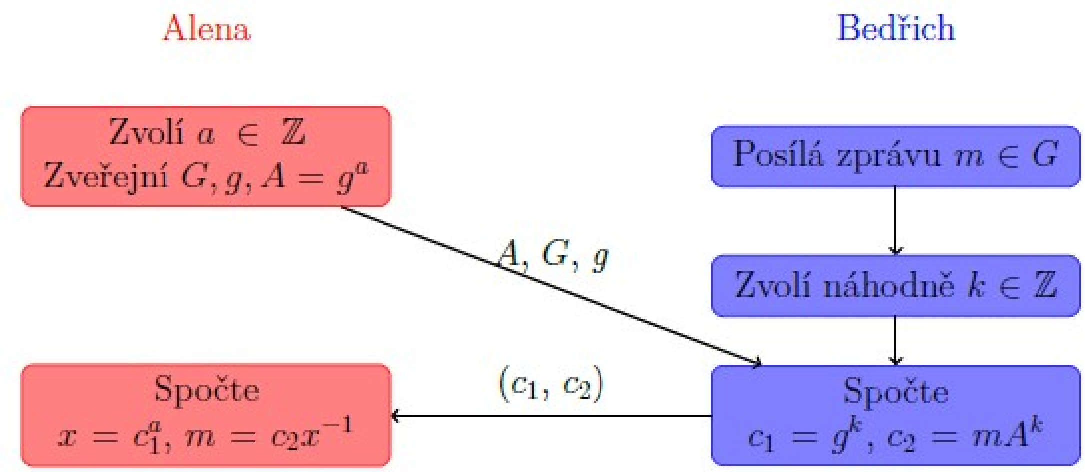
Útočník může odposlechnout $g \in G, g^a, g^b$
Diffieho-Hellmanův problém: Problém hledání g^{ab} při znalosti g^a a g^b, kde G grupa, $g \in G$, $a, b \in \mathbb{Z}$.

Útočník může řešit PDL a najít tak $\log_g g^a = a$: **Umí PDL** $\Rightarrow$ **umí DHP** v dalších $O(\log N)$ krocích (násobení).

Pokud umí DHP, neví se, jestli umí i PDL $\Rightarrow$ předpokládá se, že **PDL je těžší než DHP.**

Šifrovací systém ElGamal

G grupa, $g \in G$



Klíč $k \in \mathbb{Z}$ je **efemerní** -- slouží k odeslání pouze 1 zprávy.

Pokud zprávy $m, m' \in G$ odeslány se stejným klíčem $k = k'$:

- $c_1 = c_1'$
- $c_2 = mA^k \Rightarrow m^{-1}c_2 = A^k$
  $c_2' = m'A^k \Rightarrow m'^{-1}c_2' = A^k$
- $m^{-1}c_2 = m'^{-1}c_2'$
  $c_2 c_2'^{-1} = mm'^{-1}$
  $c_2 c_2'^{-1} m' = m$
- $\Rightarrow$ pokud znám obsah jedné zprávy, odhalím i druhou

Bezpečnost: Založena na DHP: potřeba prolomit g^{ak} z $c_1 = g^k, A = g^a$

Babystep-giantstep

G grupa, $g, h \in G$, $\mathrm{ord}(g) = N$. Pokud $\log_g h$ existuje, BSGS ho řeší v $O(\sqrt{N})$ krocích.

Algoritmus:

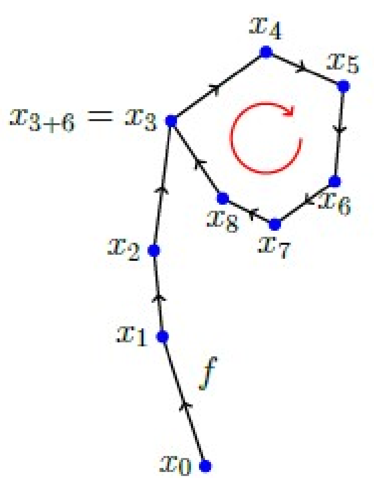
- $n = \lceil \sqrt{N} \rceil$
- Napočítat seznam $e, g, g^2, g^3, \ldots, g^{n-1}$
- Napočítat seznam $h, h \cdot g^{-n}, h \cdot g^{-2n}, \ldots, h \cdot g^{-(n-1)n}$
- Najít společný prvek obou seznamů: $g^i = h \cdot g^{-jn}$, kde $i, j \in \{0, \ldots, n-1\}$
- $\Rightarrow \log_g h = i + jn$

Důkaz:
$g^i g^{jn} = h \Leftrightarrow g^{i+jn} = h$
x se zapíše jako $x = r + qn$, kde $r < n$ a také $q = \frac{x-r}{n} < \frac{N}{n} < n$
$g^x = h$ lze zapsat jako $g^{r+qn} g = h$
Potom $g^r = h \cdot g^{-qn}$: prvek seznamu 1 $=$ prvek seznamu 2

Pollardova ρ-metoda

S končená mmnožina s N prvky, $f : S \to S$ zobrazení. Zvolme $x_0 \in S$ počáteční bod posloupnosti definované jako $x_i = (\underbrace{f \circ f \circ \ldots \circ f}_{i\text{-krát}})(x_0)$.

Potom platí, že pro nějaké $T + L \in \mathbb{N}$ nastave rovnost $x_{2i} = x_i$ pro $1 \leq i < T + L$.
($T + L$ -- počet prvků posloupnosti x_i, T -- tail, L -- loop)



Kolize $x_{2i} = x_i$: "dvojskok" na obrázku ($x_{2i} = y_i = (f \circ f)(y_{i-1})$)

- Při různých volbách zobrazení f a bodu x_0 je střední hodnota veličiny $T + L$ je $E(T + L) \approx 3,545\sqrt{N}$, takže kolizi lze najít v $O(\sqrt{N})$ krocích.

Aplikace v PDL: G grupa, $g^x = h$

- $S = G, x_0 = e$
- $G = S = S_1 \cup S_2 \cup S_3$, kde $S_i \cap S_j = \emptyset$ pro $i \neq j, e \notin S_2$
- Konkrétní volba f:

$$x_{i+1} = \begin{cases} g \cdot x_i & x_i \in S_1 \\ x_i^2 & x_i \in S_2 \\ h \cdot x_i & x_i \in S_3 \end{cases}$$

- $x_i = (\underbrace{f \circ ... \circ f}_{i\text{-krát}})(x_0) = g^{\alpha_i} h^{\beta_i}$
- Pro exponenty platí: $\alpha_0 = \beta_0 = 0$

$$\alpha_{i+1} = \begin{cases} \alpha_i + 1 & x_i \in S_1 \\ 2\alpha_i & x_i \in S_2 \\ \alpha_i & x_i \in S_3 \end{cases}$$

$$\beta_{i+1} = \begin{cases} \beta_i & x_i \in S_1 \\ 2\beta_i & x_i \in S_2 \\ \beta_i + 1 & x_i \in S_3 \end{cases}$$

- $y_i = x_{2i} = g^{\gamma_i} h^{\delta_i}$
- Kolize: $g^{\alpha_i} h^{\beta_i} = g^{\gamma_i} h^{\delta_i}$
  $g^{\alpha_i - \gamma_i} = h^{-\beta_i + \delta_i} = (g^x)^{-\beta_i + \delta_i} = g^{x(-\beta_i + \delta_i)}$
- Potom:

$$x(-\beta_i + \delta_i) \equiv (\alpha_i - \gamma_i) \pmod N$$

Ekvivalence výše nemusí mít řešení. Lepší je před Pollard-ρ pustit Pohlig-Hellmanna a řešit v prvočíselném řádu.

Během výpočtu je v paměti pouze $S_1, S_2, S_3, \underbrace{\alpha_i, \beta_i, \gamma_i, \delta_i}_{\in \mathbb{Z}_N}, \underbrace{x_i, y_i}_{\in G}$.

Pohlig-Hellmannův algoritmus

Efektivní řešení PDL $g^x = h$ na grupách, jejichž řád je složené číslo, které lze faktorizovat na malá prvočísla.

$\langle g \rangle = G$ cyklická grupa, $\#G = N = p \cdot q$
$g^{\frac{N}{p}} = g^q \Rightarrow \# \langle g^q \rangle = p \Rightarrow (g^q)^p = e$

1. část algoritmu: Rozdělení problému na několik menších PDL v grupách s řády odpovídajícími prvočíselnému rozkladu N

- **Předpoklady:**
  $g, \tilde{g}, h, \tilde{h} \in G, \operatorname{ord}(\tilde{g}) = q^l$.
  Známe faktorizaci $\operatorname{ord}(g) = N = q_1^{l_1} q_2^{l_2} ... q_k^{l_k}$.
- Pro $i \in \{1, ..., k\}$:
  - $g_i = g^{N/q_i^{l_i}}$
  - $h_i = h^{N/q_i^{l_i}}$
- Vyřešit menší PDL pro každé y_i, kde $g_i^{y_i} = h_i$
- Pomocí CRT vyřešit soustavu kongruencí $x \equiv y_i \pmod{q_i^{l_i}}$

Složitost 1. kroku: Umíme řešit PDL $\tilde{g}^{\tilde{x}} = \tilde{h}$ v čase $O(S(q^l))$. Potom PDL $g^x = h$ umíme řešit v čase $O(\sum_{i=1}^{k} S(q_i^{l_i}) + \log N)$

2. část algoritmu: Vezme malé PDL z první části a rozdělí je na ještě menší PDL, které odpovídají prvním mocninám prvočísel v rozkladu N (Pokud rozklad N obsahuje člen p^i, pak první část umožňuje řešit malý PDL v grupě řádu p^i. Druhá část umožňuje řešit i-krát v grupě řádu p.)

- **Předpoklad:** z první části zbyly pouze podgrupy řádu q^l
- Zapsat neznámé x jako $x = x_0 + x_1 q + x_2 q^2 + ... + x_{l-1} q^{l-1}$ pro $0 \leq x_i < q$
- Postupně hledat $x_0, ..., x_{l-1}$, kde pro x_i platí

$$(g^{q^{l-1}})^{x_i} = (h g^{-x_0 - ... - x_{i-1} q^{i-1}})^{q^{l-i-1}}$$

Třetí krok plyne z toho, že např. při hledání x_0:

$$h^{q^{l-1}} = (g^x)^{q^{l-1}} = g^{(x_0 + x_1 q + x_2 q^2 + ... + x_{l-1} q^{l-1}) \cdot q^{l-1}} = g^{x_0 q^{l-1}} \cdot \underbrace{g^{(x_1 + x_2 q + ... + x_{l-1} q^{l-2}) \cdot q^l}}_{\text{neutr. prvek (mocnění na násobek řádu grupy)}}$$

Prvek x_0 se získá řešením PDL $x_0 = \log_{g^{q^{l-1}}} h^{q^{l-1}}$

Pro nalezení prvku x_1 platí:

$$h^{q^{l-2}} = (g^x)^{q^{l-2}} = g^{(x_0+x_1q+x_2q^2+\ldots+x_{l-1}q^{l-1})\cdot q^{l-2}} = g^{x_0q^{l-2}} \cdot g^{x_1q^{l-1}} \cdot \underbrace{g^{(x_2+\ldots x_{l-1}q^{l-3})\cdot q^l}}_{\text{neutr. prvek}}$$

Prvek x_1 se získá řešením PDL $(g^{q^{l-1}})^{x_1} = (hg^{-x_0})^{q^{l-2}}$

Složitost 2. kroku: Umíme řešit PDL $\tilde{g}^{\tilde{x}} = \tilde{h}$, kde $\operatorname{ord}(\tilde{g}) = q$ v čase $O(S(q))$. Potom PDL $g^x = h$, kde $\operatorname{ord}(g) = q^l$, umíme řešit v čase $O(l \cdot (S(q) + \log q))$

Důsledek pro volbu grupy:
Při PDL je jedno, na jaké grupě se počítá. Vždy lze použít tento algoritmus, aby se PDL řešil na "hezkých" grupách (grupách, kde všude existují inverze)

Index calculus

Algoritmus řešící PDL, ale pouze na specifických grupách (typicky $G = GF(p^n)^\times$)

Hlavní myšlenka: Převést PDL na soustavu lineárních rovnic

Předvýpočet: závisí pouze na grupě G, ne na konkrétním problému

- Zvolit faktorovou bázi $S = \{p_1, ..., p_t\} \subset G$
- Náhodně vybrat $l \in \mathbb{N}, l < \#G$
- Spočítat g^l
- Otestovat, zda $g^l = \prod_{i=1}^t p_i^{c_i}$ *(test, jestli g^l jde rozložit na prvky zvolené báze)*
    - Pokud ano, $l \equiv \sum_{i=1}^t c_i \log_g p_i \pmod{\#G}$ *(soustava rovnic, je jich potřeba najít t lin. nezávislých)*
- Řešit $l_j \equiv \sum_{i=1}^t c_{ij} \log_g p_i \pmod{\#G}$ pro neznámé $\log_g p_i$

Řešení PDL:

- Náhodně vybrat $k \in N, k < \#G$
- Otestovat, zda $hg^{-k} = \prod_{i=1}^t p_i^{d_i}$
    - Pokud ano, vztah zlogaritmovat na $\log_g h - k = \sum_{i=1}^t d_i \log_g p_i \pmod{\#G}$
- Potom:

$$\log_g h \equiv \sum_{i=1}^{t} d_i \log_g p_i + k \pmod{\#G}$$

Kde se dosadí $\log_g p_i$ získané z předvýpočtu.

Volba báze:

- Pro $G = \mathbb{Z}_p^\times$ je $S = \{p : p \text{ je prvočíslo}, p < B\}$
- Pro $G = GF(p^n)$ je $S = \{f : f \text{ je ireducibilní}, \deg(f) < B\}$

B**-hladká čísla:** čísla, která nemají v prvočíselném rozkladu faktor větší než B
B**-hladký polynom:** v rozkladu na ireducibilní polynomy nemá žádný faktor stupně vyššího než B

Funkce $L_q[\alpha, c]$: funkce definovaná jako $L_q[\alpha, c] = \exp(c(\ln q)^\alpha (\ln \ln q)^{1-\alpha})$, kde $c > 0, 0 < \alpha < 1$

- běžně používaná pro odhad složitosti subexponenciálních algoritmů (algoritmů $O(e^{f(k)})$, kde $f(k) = o(k)$ -- hodně malý exponent)
- α uvádá "míru exponenciality"
  - $O(L_q[1, c]) = O(e^{c \ln q}) = O(q^c)$ -- plně expoenciální v délce vstupu $\ln q$
  - $O(L_q[0, c]) = O(e^{c \ln \ln q}) = O((\ln q)^c)$ -- polynomiální v délce vstupu $\ln q$

Odhad složitosti Index calculu:
V $GF(2^n)^\times$

- Počet ireducibilních polynomů stupně k nad $\mathbb{Z}_p$: $\approx \frac{2^k}{k}$
- Velikost faktorové báze S (ireducibilní polynomy stupně 0 až m nad $\mathbb{Z}_2$): $|S| \approx \frac{2^{m+1}}{m}$
- Šance na úspěšnou faktorizaci g^l:
  $P_{fakt} = \frac{1}{2^n} \sum_{k=0}^{m-1} N(k, m) \approx \left(\frac{m}{n}\right)^{1+O(1)\frac{n}{m}}$, kde $N(k, m)$ je počet m-hladkých polynomů stupně k
- Střední doba trvání nalezení faktorizace: $\frac{1}{P_{fakt}} = \left(\frac{n}{m}^{1+O(1)\frac{n}{m}}\right)$
- Nutno nalézt o něco víc než $|S|$ kongruencí tvaru $l_j \equiv \sum_{i=1}^{t} c_{ij} \log_g p_i \pmod{\#G}$
- Soustavu kongruencí lze řešit v $O(|S|^3)$
- **Celková složitost:**

$$|S| \frac{1}{P_{fakt}} + |S|^3 \approx \frac{2^{m+1}}{m} \left(\frac{n}{m}\right)^{1+O(1)\frac{n}{m}} + \frac{2^{3m+3}}{m}$$

- Výraz je minimální pro $m = c\sqrt{n \ln n}$, asymptotická složitost je potom

$$O(\exp((c + o(1))\sqrt{n \ln n}))$$

Což odpovídá

$$L_{2^n}[\frac{1}{2}, c]$$