

Diskrétní logaritmus

13. a 14. přednáška z kryptografie

Obsah

- 1 Protokoly Diffieho-Hellmanův a ElGamalův
 - Diskrétní logaritmus
 - Diffieho-Hellmanův a ElGamalův protokol
 - Bezpečnost obou protokolů
- 2 Výpočet diskrétního logaritmu
 - "Baby step-giant step" algoritmus
 - Pohlingův-Hellmanův algoritmus
- 3 Hledání generátoru v \mathbb{Z}_p^*

Diskrétní logaritmus

Definice

Nechť $G = \langle a \rangle$ je cyklická grupa řádu n s generátorem a . Každý prvek $b \in G$ lze zapsat jako $b = a^x$ pro jediné $x \in \mathbb{Z}_n$. Toto x se nazývá *diskrétní logaritmus* o základu a z prvku b v grupě G . Značí se $\text{dlog}_a(b)$.

Příklad

$\mathbb{Z}_9^* = \langle 2 \rangle$, $|\mathbb{Z}_9^*| = 6$.

$b \in \mathbb{Z}_9^*$	1	2	4	5	7	8
$\text{dlog}_2(b)$	0	1	2	5	4	3

Diskrétní logaritmus

Tvrzení

Nechť $G = \langle a \rangle$ je cyklická grupa řádu n . Exponenciální zobrazení $\exp_a : (\mathbb{Z}_n, +) \rightarrow (G, \cdot) : x \mapsto a^x$ je grupový isomorfismus. Diskrétní logaritmus $\text{dlog}_a : (G, \cdot) \rightarrow (\mathbb{Z}_n, +) : g = a^x \mapsto x$ je k němu inverzní zobrazení, tudíž také grupový isomorfismus.

Pro $b, c \in G$, $k \in \mathbb{Z}$ platí vzorce:

- $\text{dlog}_a(b \cdot c) = \text{dlog}_a(b) + \text{dlog}_a(c)$
- $\text{dlog}_a(1) = 0$
- $\text{dlog}_a(b^k) = k \text{dlog}_a(b)$, $\text{dlog}_a(b^{-1}) = -\text{dlog}_a(b)$

Diskrétní logaritmus

Poznámka

Někdy se mluví o diskétním logaritmu obecněji:

Nechť G je Abelova grupa, $a, b \in G$ libovolné prvky.

Jestliže $b \in \langle a \rangle$, pak je $\text{dlog}_a(b)$ definován jako libovolné $x \in \mathbb{Z}$, pro něž je $a^x = b$. Takové x je určeno jednoznačně modulo řád prvku a . Jestliže $b \notin \langle a \rangle$, pak $\text{dlog}_a(b)$ definován není.

Pokud je grupa G cyklická, $r(a) = r$ (a nemusí být generátor), pak $b \in \langle a \rangle$, právě když b řeší rovnici $x^r = 1$.

Aneb $\text{dlog}_a(b)$ je definován, právě když $b^{r(a)} = 1$ v grupě G .

Diskrétní logaritmus

Problém diskrétního logaritmu

Předpokládá se, že pro většinu grup je výpočet diskrétního logaritmu exponenciální nebo subexponenciální problém - např. pro grupy \mathbb{Z}_p^* a jejich podgrupy nebo pro grupy eliptických křivek. (V šifrovacích protokolech si pod grupou G představujte tyto grupy.)

V grupě $(\mathbb{Z}_n, +)$ není těžké spočítat diskrétní logaritmus.

V aditivní grupě je $\text{dlog}_a(b) = x \in \mathbb{Z}_n$ takové, že $b = x \cdot a$ v \mathbb{Z}_n .

To je lineární rovnice a umíme ji řešit rozšířeným Eukleidovým algoritmem v čase $O(\ln(n)^2)$.

Diskrétní logaritmus v kryptografii

- Diffieho-Hellmanův protokol je veřejná domluva na společném tajném klíči pro symetrické šifrování. Jeho bezpečnost se opírá o problém diskrétního logaritmu.
- Autoři: Whitfield Diffie a Martin Hellman, USA, 1976.
- ElGamalovo šifrování je nesymetrické šifrování s veřejným klíčem, které se opírá o stejnou myšlenku jako Diffieho-Hellmanova výměna klíčů.
- Autor: Taher ElGamal, USA (Egypt), 1985.

Diffieho-Hellmanova domluva klíče

Protokol

Alice zvolí cyklickou grupu G řádu n a její generátor a .

Dále zvolí $x \in \mathbb{Z}_n$ a spočte prvek $b = a^x$ v grupě G .

Alice pošle Bobovi prvek b a informace o grupě (G, n, a) .

Bob zvolí $y \in \mathbb{Z}_n$ a spočte prvek $c = a^y$ v grupě G .

Bob pošle Alici prvek c .

Alice spočte $s_A = c^x$ a Bob spočte $s_B = b^y$ v grupě G .

Tím oba získají stejný tajný klíč $s = s_A = s_B = a^{xy}$.

Diffieho-Hellmanova domluva klíče

Protokol

Popřípadě třetí strana zvolí cyklickou grupu G řádu n a její generátor a . Zveřejní (G, n, a) v "telefonním seznamu", do nějž Alice vloží pod své jméno veřejnou část klíče $b = a^x$ a Bob vloží pod své jméno veřejnou část klíče $c = a^y$. Zatímco tajnou část klíče x , resp. y nikomu neprozradí.

Šifrování

Domluvený tajný klíč s může Alice s Bobem použít k jakémukoli symetrickému šifrování.

Tato možnost veřejné domluvy na tajném klíči vyřešila problém distribuce klíčů, který symetrické šifrování zatěžoval.

ElGamalovo šifrování

Vytvoření klíčů

Alice zvolí cyklickou grupu G řádu n a její generátor a . Dále zvolí $x \in \mathbb{Z}_n$ a spočte prvek $b = a^x$ v grupě G .

Alice zveřejní prvek b a informace o grupě (G, n, a) v "telefonním seznamu".

Prvek b je Alicin veřejný klíč, x je Alicin tajný klíč.

ElGamalovo šifrování

Šifrování

Bob zvolí $y \in \mathbb{Z}_n$ - tzv. jepičí klíč.

Spočte prvky $c = a^y$, $s = b^y$ (tedy opět $s = a^{xy}$) v grupě G .

Bob zašifruje zprávu $m \in G$: $\bar{m} = m \cdot s$ v grupě G .

Bob pošle Alici dvojici prvků (c, \bar{m}) .

Dešifrování

Alice si spočte $s = c^x$, resp. $s^{-1} = c^{n-x}$ v grupě G .

Alice dešifruje zprávu \bar{m} : $m = \bar{m} \cdot s^{-1}$ v grupě G .

ElGamalovo šifrování

Příklad

Alicin veřejný klíč je $b = 7$ pro $\mathbb{Z}_{37}^* = \langle 2 \rangle$, řád grupy $n = 36$. Alicin soukromý klíč je $x = 32$.

Bob chce zašifrovat zprávu $m = 10$, použije jepičí klíč $y = 8$. Spočte $c = 2^8 = 34$, $s = 7^8 = 16$, $\bar{m} = 10 \cdot 16 = 12$ v \mathbb{Z}_{37}^* . Pošle $(c, \bar{m}) = (34, 12)$.

Alice chce zprávu \bar{m} dešifrovat.

Spočte $s^{-1} = 34^{36-32} = 7$, $m = 12 \cdot 7 = 10$ v \mathbb{Z}_{37}^* .

ElGamalovo šifrování

Pro šifrování i dešifrování u ElGamala potřebujeme:

- Umocňovat v grupě G , což metodou opakovaných čtverců vyžaduje $O(\text{len}(n))$ násobení v grupě G , kde $n = |G|$.
- Pro každou zprávu zvolit jiný jepičí klíč (pomocí generátoru náhodných čísel ze \mathbb{Z}_n).
To je nutné kvůli bezpečnosti: Pokud by Eva dešifrovala jednu zprávu \bar{m} , dopočetla by si klíč $s = \bar{m}m^{-1}$ a dešifrovala by pak všechny zprávy.
- Nevýhoda protokolu - zašifrovaná zpráva bude dvojnásobně dlouhá. Z tohoto důvodu se ElGamalovo šifrování používá mnohem méně než RSA šifrování.

Bezpečnost obou protokolů

Problém diskretního logaritmu

Nechť grupa $G = \langle a \rangle$ je řádu n .

Chceme pro daný prvek $b \in G$ najít $x \in \mathbb{Z}_n$ tak, že $b = a^x$ v G .

Bezpečnost Diffie-Hellmanova a ElGamalova protokolu se opírá o složitost problému diskretního logaritmu v dané grupě.

Pro podgrupy v \mathbb{Z}_p^* či v grupy eliptických křivek doposud známe:

- exponenciální algoritmy vyžadující $O(\sqrt{n}) = O(2^{\frac{1}{2} \text{len}(n)})$ násobení v G , tyto algoritmy fungují v každé cyklické grupě;
- je-li G podgrupa v \mathbb{Z}_p^* , pak také subexponenciální pravděpodobnostní algoritmus pracující v čase $O(e^{(2\sqrt{2}+o(1))\sqrt{\ln(p)\ln(\ln(p))}})$.

Exponenciální zobrazení je v těchto grupách *jednosměrná funkce*.

ElGamalovo šifrování

Pro vytvoření obou protokolů potřebujeme cyklickou grupu a její generátor. V praxi se volí (viz Shoup, 2005):

- G podgrupa v \mathbb{Z}_p^* ,
kde p je prvočíslo o 1024 bitech - dostatečné vzhledem k subexponenciálním algoritmům na diskretní logaritmus, $|G| = q$ je prvočíslo o 160 bitech - dostatečné vzhledem k exponenciálním algoritmům na diskretní logaritmus.
Zprávy lze volit $m \in \mathbb{Z}_p^*$ (a bude to fungovat) a umocňování (na jepičí klíče) půjde rychleji, neboť exponent je menší než q .
- G je grupa bodů na eliptické křivce řádu $|G|$ o 160 bitech. (V těchto grupách není znám subexponenciální algoritmus na diskretní logaritmus.)
Zprávy je nutno konvertovat do grupy G .

Bezpečnost obou protokolů

Diffieho-Hellmanův problém

Ze znalosti $b = a^x$ a $c = a^y$ spočítat $s = a^{xy}$ v grupě $G = \langle a \rangle$.

Přitom exponenty x, y neznáme.

Zatím se umí Diffieho-Hellmanův problém vyřešit pouze přes diskretní logaritmus, tedy přes vypočítání x a y . Věřící se, že Diffie-Hellmanův problém je exponenciálně složitý.

Diffieho-Hellmanův rozpoznávací problém

Poznat, zda trojice (b, c, d) je tvaru (a^x, a^y, a^{xy}) v grupě $G = \langle a \rangle$.

Zatím se míní, že Diffieho-Hellmanovy trojice (a^x, a^y, a^{xy}) nejsou odlišitelné od náhodných trojic (a^x, a^y, a^z) ani pravděpodobnostními metodami a že i tento problém je exponenciálně složitý.

Bezpečnost obou protokolů

Poznámky

- Pokud je $|G| = n$ dělitelné malými prvočísly, pak se diskretní logaritmus dá spočítat řádově rychleji (exponenciální čas s menším exponentem - viz. Pohlingův-Hellmanův algoritmus).
- Pokud je $|G| = n$ je dělitelné malými prvočísly, pak existuje pravděpodobnostní algoritmus, který rozpozná Diffieho-Hellmanovu trojici v polynomiálním čase $(q + \text{len}(p))^{O(1)}$ s pravděpodobností $\frac{1}{q}$, kde q je nejmenší prvočíslo, které dělí n . Složitost algoritmu je pro G podgrupu v \mathbb{Z}_p^* .
- Proto se volí $|G| = q$ prvočíslo.

Diskretní logaritmus - výpočet

Nechť nadále $G = \langle a \rangle$ je cyklická grupa řádu n s generátorem a . Pro $b \in G$ počítáme $\text{dlog}_a(b)$, tj. $x \in \mathbb{Z}_n$ takové, že $b = a^x$ v G . Následující algoritmy fungují v každé cyklické grupě. Pro časovou složitost budeme určovat jen počet násobení, rychlost násobení je v každé grupě jiná.

"Brute force" algoritmus

Výpočet hrubou silou: Počítáme a^i pro $0 \leq i < n$ tak, že postupně násobíme prvkem a , dokud nevyjde prvek b .

Časová složitost - provádíme nejvýše n násobení v grupě G , tedy potřebný čas je $O(n) = O(2^{\text{len}(n)})$ násobení v G .

Diskretní logaritmus - výpočet

"Baby step-giant step" algoritmus

Zvolme aproximaci $m \doteq \sqrt{n}$, pak také $m' = \lceil \frac{n}{m} \rceil \doteq \sqrt{n}$. Zapišeme $x = \text{dlog}_a(b) = vm + u$, pak $0 \leq u < m$, $0 \leq v \leq m'$, protože $0 \leq x < n$. Hledáme příslušná u, v .

$$b = a^x = a^{vm+u}, \quad \text{odtud} \quad b(a^{-m})^v = a^u.$$

Baby steps (dětské krůčky): Spočteme a^i pro všechna $0 \leq i < m$ a uložíme je do paměti.

Vhodná implementace - vyhledávací (vyvážený binární) strom T , kde $T(g) = i$, pokud $a^i = g$, a $T(g) = \perp$ pro ostatní prvky $g \in G$. Pak bude třeba $O(\sqrt{n})$ prostoru a vyhledávání bude v čase $O(\text{len}(n))$.

Diskretní logaritmus - výpočet

"Baby step-giant step" algoritmus

Giant steps (obří kroky): Počítáme $b(a^{n-m})^j$ pro $0 \leq j < m'$, dokud nezískáme výsledek stejný jako některý z výsledků v části "baby steps".

Tam, kde vznikly stejné výsledky, je $i = u$ a $j = v$, tudíž získáme $\text{dlog}_a(b) = x = vm + u$.

Časová složitost - provádíme max. $2\sqrt{n}$ násobení v grupě G a \sqrt{n} vyhledávání ve stromě T . (Přitom čas $O(\text{len}(n))$ na vyhledávání je menší než čas na násobení ve většině grup.) Potřebný čas je $O(\sqrt{n}) = O(2^{\frac{1}{2}\text{len}(n)})$ násobení v G , tedy exponenciální s polovičním exponentem než u hrubé síly.

Prostorová složitost je $O(\sqrt{n}) = O(2^{\frac{1}{2}\text{len}(n)})$, tedy také exponenciální, což je větší problém.

Diskrétní logaritmus - výpočet

Příklad

Grupa \mathbb{Z}_{37}^* je cyklická řádu $n = 36$ s generátorem $a = 2$.

Spočteme $\text{dlog}_2(7)$ v grupě \mathbb{Z}_{37}^* .

Položíme $m = \sqrt{36} = 6$, pak $m' = \frac{36}{6} = 6$.

Tedy $x = \text{dlog}_2(7) = 6v + u$, kde $0 \leq u, v \leq 5$.

$$7 = 2^x = 2^{6v+u}, \text{ odtud } 2^u = 7(2^{-6})^v = 7(2^{30})^v = 7 \cdot 11^v$$

Baby steps: $2^0 = 1, 2^1 = 2, 2^2 = 4, 2^3 = 8, 2^4 = 16, 2^5 = 32$

Giant steps: $7 \cdot 11^0 = 7, 7 \cdot 11^1 = 3, 7 \cdot 11^2 = 33, 7 \cdot 11^3 = 30, 7 \cdot 11^4 = 34, 7 \cdot 11^5 = 4$ (postupně násobíme 11-ti)

Rovnost nastala pro $2^2 = 4 = 7 \cdot 11^5$, tedy $u = 2, v = 5$,

$\text{dlog}_2(7) = 6 \cdot 5 + 2 = 32$.

Diskrétní logaritmus - výpočet

Pohlingův-Hellmanův algoritmus

Nechť G je cyklická grupa řádu n s generátorem a , nechť $b \in G$.

Známe-li faktorizaci čísla n , lze urychlit výpočet $\text{dlog}_a(b)$.

- $|G| = n = \prod_{i=1}^k q_i^{e_i}$, pak lze počítat diskretní logaritmy v podgrupách řádů $q_i^{e_i}$ a použít Čínskou větu o zbytcích.

Je-li $a^x = b$ v grupě G , pak $(a^{n_i})^x = b^{n_i}$, kde $n_i = n/q_i^{e_i}$.

Spočteme $x_i = \text{dlog}_{a^{n_i}}(b^{n_i})$ v podgrupě $H_i = \langle a^{n_i} \rangle$ řádu $q_i^{e_i}$

pro každé $1 \leq i \leq k$ a získáme tak zbytkovou soustavu k rovnic

$x \equiv x_i \pmod{q_i^{e_i}}$, jejímž řešením je $x = \text{dlog}_a(b)$.

Diskrétní logaritmus - výpočet

Poznámky

- Pollardova ρ -metoda na výpočet diskretního logaritmu je pravděpodobnostní algoritmus pracující v očekávaném čase $O(\sqrt{n}) = O(2^{\frac{1}{2} \text{len}(n)})$ násobením v G . Jeho výhodou je polynomiální paměťová náročnost. Publikoval John Pollard v roce 1978.
- Index calculus je subexponenciální algoritmus na diskretní logaritmus fungující pro podgrupy G prvočíselného řádu q v grupě \mathbb{Z}_p^* . Pracuje v čase $O(e^{(2\sqrt{2}+o(1))\sqrt{\ln(p)\ln(\ln(p))}})$. Představíme jej v samostatné přednášce pod názvem SEDL.

Diskrétní logaritmus - výpočet

Pohlingův-Hellmanův algoritmus

- $|G| = q^e$ mocnina prvočísla, tak výpočet diskretního logaritmu lze převést rekursivně na výpočet e diskretních logaritmů v podgrupě řádu q . Potřebný čas je $O(eq^{\frac{1}{2}} + e \text{len}(q))$ násobením v G .

Je-li $a^x = b$ v grupě G , pak $x = x_{e-1}q^{e-1} + \dots + x_1q + x_0 < q^e$.

Číslice $0 \leq x_i < q$ budeme počítat postupně od x_0 jako diskretní logaritmy v podgrupě $H = \langle a^{(q^{e-1})} \rangle$ řádu q .

Umocníme rovnici $a^x = b$ na q^{e-1} , protože $r(a) = q^e$, dostaneme $(a^{(q^{e-1})})^{x_0} = b^{(q^{e-1})}$ a spočteme x_0 . Pak umocníme rovnici na q^{e-2} a dopočteme x_1 . Pokračujeme až po x_{e-1} a máme q -ární rozvoj pro x .

- $|G| = q$ prvočísla, tak použijeme klasický "Baby step - giant step" algoritmus. Potřebný čas je $O(q^{\frac{1}{2}})$ násobením v G .

Diskrétní logaritmus - výpočet

Pohlingův-Hellmanův algoritmus

- Bud' q je největší prvočíslo ve faktorizaci $n = |G|$, čas potřebný na výpočet diskretního logaritmu v grupě G je dán časem jeho výpočtu v podgrupě řádu q .
Časová složitost je tedy zhruba $O(q^{\frac{1}{2}})$ násobení v G .
- Algoritmus publikovali Stephen Pohling a Martin Hellman v roce 1978 jako Pohlingův-Hellmanův útok na Diffieho-Hellmanovu domluvu klíče. Je-li (velké) $n = |G|$ součinem malých prvočísel, pak není protokol bezpečný.

Diskrétní logaritmus - výpočet

Příklad

Grupa $G = \mathbb{Z}_{37}^*$ je cyklická řádu $n = 36$ s generátorem $a = 2$. Spočteme $x = \text{dlog}_2(7)$ v grupě \mathbb{Z}_{37}^* .

$$n = 36 = 2^2 \cdot 3^2 = 4 \cdot 9.$$

Nechť $x \in \mathbb{Z}_{36}$ má zbytky $\theta(x) = (x', x'') \in \mathbb{Z}_4 \times \mathbb{Z}_9$.

Rovnost $7 = 2^x$ v grupě G implikuje rovnosti:

- $7^9 = (2^9)^x = (2^9)^{x'}$ v podgrupě H' řádu 4 s generátorem 2^9 ,
 $2^9 = 31$, $7^9 = 1$, tedy $x' = \text{dlog}_{31}(1) = 0$
- $7^4 = (2^4)^x = (2^4)^{x''}$ v podgrupě H'' řádu 9 s generátorem 2^4 ,
 $2^4 = 16$, $7^4 = 33$, tedy $x'' = \text{dlog}_{16}(33) = 5$ (použijeme Baby step - giant step algoritmus pro grupu H'').

Z Čínské věty o zbytcích dopočteme $x = \theta^{-1}(0, 5) = 32$.

Diskrétní logaritmus - výpočet

Příklad

$x'' = \text{dlog}_{16}(33) = 5$ v podgrupě H'' řádu $9 = 3^2$ s generátorem 16 můžeme spočítat rekurzivně:

Označme $x'' = 3x_1 + x_0$, kde $0 \leq x_0, x_1 < 3$.

- Rovnici $33 = 16^{x''} = 16^{3x_1 + x_0}$ umocníme nejdříve na třetí,
 $33^3 = 16^{9x_1 + 3x_0} = 16^{3x_0}$, v exponentu se počítá modulo 9.
 $16^3 = 26$, $33^3 = 10$, tedy $x_0 = \text{dlog}_{26}(10) = 2$.
- Dosadíme do původní rovnice: $33 = 16^{3x_1 + 2}$ a upravíme:
 $16^{3x_1} = 33 \cdot 16^{-2} = 26$, tedy $x_1 = \text{dlog}_{26}(26) = 1$.

Našli jsme $x'' = 3 \cdot 2 + 1 = 5$.

Hledání generátoru v \mathbb{Z}_p^*

Víme, že prvek a je generátor cyklické grupy G řádu n právě, když $a^d \neq 1$ pro každého vlastního dělitele d čísla n . Přitom stačí testovat jen maximální dělitele čísla n .

Aneb, abychom mohli ověřit, zda je prvek generátorem, musíme znát faktorizaci čísla n (řádu grupy).

V následující části se omezíme na cyklické grupy \mathbb{Z}_p^* , algoritmy by analogicky fungovaly pro každou cyklickou grupu.

Časová složitost je počítána vzhledem k násobení v \mathbb{Z}_p , kde vynásobení dvou čísel odhadujeme časem $O(\text{len}(p)^2)$.

Hledání generátoru v \mathbb{Z}_p^*

Algoritmus číslo 1

Nechť p je prvočíslo a $p - 1 = \prod_{i=1}^k q_i^{e_i}$ je faktorizace pro $|\mathbb{Z}_p^*|$.
Veškeré výpočty jsou prováděny v \mathbb{Z}_p .

- repeat
 - vyber náhodně $a \in \mathbb{Z}_p^*$
 - $GEN \leftarrow true$, $i \leftarrow 1$ [prvek a může být generátorem]
 - repeat
 - if $a^{\frac{p-1}{q_i}} = 1$ then $GEN \leftarrow false$ endif
 - $i \leftarrow i + 1$
 - until *not* GEN or $i > k$
- until GEN
- output a

Hledání generátoru v \mathbb{Z}_p^*

Korektnost algoritmu č.1

- Algoritmus zastaví, protože \mathbb{Z}_p^* je cyklická grupa.
- Prvek a na výstupu bude generátorem grupy \mathbb{Z}_p^* díky následujícímu tvrzení.

Tvrzení

Prvek a je generátor cyklické grupy G řádu n právě, když $a^{\frac{n}{p}} \neq 1$ pro každé prvočíslo p , kde $p \mid n$.

Hledání generátoru v \mathbb{Z}_p^*

Časová náročnost algoritmu č.1

- Cyklická grupa řádu n má celkem $\varphi(n)$ možností, jak zvolit generátor. Spočteme pravděpodobnost, že náhodně zvolený prvek ze \mathbb{Z}_p^* je generátor:

$$P[a \text{ generátor}] = \frac{\varphi(p-1)}{p-1} = \prod_{i=1}^k \frac{q_i - 1}{q_i} \geq \prod_{i=2}^{k+1} \frac{i-1}{i} = \frac{1}{k+1}$$

- Průměrný počet náhodných výběrů tedy bude nejvýše $k + 1$. (Časem toto odvodíme precizněji, zavedeme náhodnou veličinu L = počet vnějších repeat-until cyklů a střední hodnota této náhodné veličiny bude $E(L) \leq k + 1$.)

Hledání generátoru v \mathbb{Z}_p^*

Časová náročnost algoritmu č.1

- V každém cyklu počítáme nejvýše k mocnin s exponentem menším než p metodou opakovaných čtverců, tedy budeme potřebovat čas $O(k \text{len}(p)^3)$.
- Celkový očekávaný čas je $O(k^2 \text{len}(p)^3)$, kde k je počet různých prvočísel ve faktorizaci $p - 1$. Očekávaný čas je také $O(\text{len}(p)^5)$, neboť $k < \text{len}(p)$.

Hledání generátoru v \mathbb{Z}_p^*

Algoritmus číslo 2

Nechť p je prvočíslo a $p - 1 = \prod_{i=1}^k q_i^{e_i}$ je faktorizace pro $|\mathbb{Z}_p^*|$.
Veškeré výpočty jsou prováděny v \mathbb{Z}_p .

- for $i \leftarrow 1$ to k do
 - repeat
 - vyber náhodně $b \in \mathbb{Z}_p^*$
 - $b_i \leftarrow b^{\frac{p-1}{q_i}}$
 - until $b_i \neq 1$
 - $a_i \leftarrow b^{\frac{p-1}{q_i}}$
- $a \leftarrow \prod_{i=1}^k a_i$
- output a

Hledání generátoru v \mathbb{Z}_p^*

Korektnost algoritmu č.2

- Algoritmus zastaví, protože \mathbb{Z}_p^* je cyklická grupa, tudíž obsahuje prvky všech řádů, které dělí řád grupy.
- Prvek a_i má řád $q_i^{e_i}$ (viz následující tvrzení) a díky nesoudělnosti řádů pro různá a_i bude výstupní prvek a mít řád $\prod_{i=1}^k q_i^{e_i} = p - 1$ a bude generátorem grupy \mathbb{Z}_p^* .

Tvrzení

Buď q prvočíslo a $e \geq 1$ přirozené číslo.

Nechť prvek c Abelovy grupy splňuje $c^{(q^e)} = 1$ a $c^{(q^{e-1})} \neq 1$, pak řád prvku c je q^e .

Hledání generátoru v \mathbb{Z}_p^*

Časová náročnost algoritmu č.2

- Prvek $b_i = b^{\frac{p-1}{q_i}} \neq 1$ má řád q_i .
Pravděpodobnost, že pro náhodně zvolený prvek $b \in \mathbb{Z}_p^*$ je jeho $\frac{p-1}{q_i}$ -tá mocnina různá od 1, je

$$P[b_i \neq 1] = \frac{q_i - 1}{q_i} \geq \frac{1}{2}.$$

- Každý z k repeat-until cyklů se opakuje průměrně dvakrát, počítá se vždy jedna mocnina metodou opakovaných čtvrců.
- Celkový očekávaný čas je $O(2k \ln(p)^3)$, kde k je počet různých prvočísel ve faktorizaci $p - 1$.
Očekávaný čas je také $O(\ln(p)^4)$, neboť $k < \ln(p)$.

Hledání prvku řádu q v \mathbb{Z}_p^*

Algoritmus

Nechť p je prvočíslo a q je prvočíslo takové, že $q \mid p - 1$.
Veškeré výpočty jsou prováděny v \mathbb{Z}_p .

- repeat
 - vyber náhodně $b \in \mathbb{Z}_p^*$
 - $c \leftarrow b^{\frac{p-1}{q}}$
- until $c \neq 1$
- output c

Korektnost a časová složitost

Algoritmus je korektní, očekávaný čas běhu je $O(\ln(p)^3)$.
Přitom každý prvek řádu q může být nalezen se stejnou pravděpodobností.

Hledání prvku řádu q v \mathbb{Z}_p^*

Příklad

Pro $p = 317$ je $p - 1 = 2^2 \cdot 79$.

Hledáme prvek řádu 79 v \mathbb{Z}_{317}^* .

Zvolíme např. $b = 2$. Protože $2^4 = 16 \neq 1$, tak $c = 16$ má řád 79.

Máme podgrupu $G = \langle 16 \rangle$ řádu 79 v grupě \mathbb{Z}_{317}^*
a můžeme ji použít pro ElGamal šifrování.

Informace o grupě v "telefonním seznamu":

$(p, n, a) = (317, 79, 16)$

- násobí se modulo $p = 317$, zprávy $0 < m < 317$
- v exponentu se počítá modulo $n = 79$, jepičí klíče $0 < y < 79$
- generátorem grupy je prvek $a = 16$

Diskrétní logaritmus

Literatura

- Shoup: A Computational Introduction to Number Theory and Algebra. Kapitola 11.
<http://shoup.net/ntb/>