

Subexponenciální algoritmus pro faktORIZACI

24. a 25. přednáška z kryptografie

Obsah

- 1 Subexponenciální algoritmus pro faktORIZACI**
 - Algoritmus SEF
 - Analýza algoritmu SEF

- 2 FaktORIZACE POMOCÍ KVADRATICKÉHO SÍTA**
 - Algoritmus QSF
 - Analýza algoritmu QSF

Algoritmus SEF

Používaná fakta

Subexponenciální algoritmus pro faktORIZACI SEF používá (podobně jako SEDL) y -hladkost a lineární algebru nad tělesem.

Časová náročnost pro faktORIZACI n bude $O(2^{c\sqrt{\text{len}(n)\text{len}(\text{len}(n))}})$.

Tvrzení

Známe-li netriviální druhou odmocninu z 1 v \mathbb{Z}_n , tj. $c \neq \pm 1$ takové, že $c^2 = 1$ v \mathbb{Z}_n , pak umíme faktORIZOVAT n , konkrétně $\text{gcd}(c \pm 1, n) \neq 1$ je faktor n .

Algoritmus SEF

Subexponenciální algoritmus pro faktORIZACI (SEF)

Vstup: přirozené číslo n , které není prvočíslo, ani mocnina prvočíslo, ani není dělitelné žádným prvočíslem $p \leq y$, kde y je parametr hladkosti (mimo jiné je n liché);

Výstup: netriviální faktor čísla n , anebo hláška "neúspěch";

Algoritmus najde druhou odmocninu z 1 v \mathbb{Z}_n , pokud je tato netriviální, spočte z ní faktor čísla n .

Poznámka

Grupa $\mathbb{Z}_{p^e}^*$ je pro $p > 2$ cyklická, nejsou v ní tedy netriviální druhé odmocniny z 1. Nebude-li n dělitelné žádným prvočíslem $p \leq y$, tak všechna y -hladná čísla v \mathbb{Z}_n budou invertibilní.

Algoritmus SEF

Subexponenciální algoritmus pro faktORIZACI (SEF)

Předpoklady pro n zajistíme předvýpočty, které budou časově méně náročné než vlastní algoritmus.

- n není prvočíslo:
Millerův-Rabinův test $MR(-, \tilde{k})$ vyžaduje čas $O(\tilde{k} \text{len}(n)^3)$.
- n není dokonalá mocnina, $n \neq m^e$ pro $m, e \in \mathbb{N}$:
Pomocí algoritmu na hledání celočíselných odmocnin lze najít m, e v čase $O(\text{len}(n)^3 \text{len}(\text{len}(n)))$.
- n není dělitelné žádným prvočíslem $p_1, \dots, p_k \leq y$,
kde y je parametr hladkosti:
Prosté dělení trvá $O(k \text{len}(n)^2)$, kde $k < y \doteq e^{\sqrt{\ln(n)}}$.

Algoritmus SEF

1. fáze algoritmu SEF

Nechť p_1, \dots, p_k jsou všechna prvočísla do y , je jich tedy k .

Náhodnou volbou nalezneme $(k + 1)$ y -hladkých čtverců ze \mathbb{Z}_n^* .

Provedeme to pro každé $1 \leq i \leq k + 1$ takto:

- zvol náhodně $a_i \in \mathbb{Z}_n^*$
- ověř prostým dělením, zda $a_i^2 = m_i$ v \mathbb{Z}_n je y -hladké, tj. zda $m_i = p_1^{e_{i1}} \cdot \dots \cdot p_k^{e_{ik}}$ v \mathbb{Z} , kde $0 \leq m_i < n$;
pak $a_i^2 = p_1^{e_{i1}} \cdot \dots \cdot p_k^{e_{ik}}$ v \mathbb{Z}_n^*
- dokud ne, tak opakuj náhodnou volbu

Algoritmus SEF

2. fáze algoritmu SEF

Pro $1 \leq i \leq k + 1$ uvažujme k -tice exponentů $\bar{v}_i = (e_{i_1}, \dots, e_{i_k})$ jakožto vektory nad \mathbb{Z}_2 .

\mathbb{Z}_2 je těleso, k -tice $\mathbb{Z}_2^{\times k}$ tvoří vektorový prostor dimenze k , našich $(k + 1)$ vektorů tedy musí být lineárně závislých.

Existují koeficienty $c_1, \dots, c_{k+1} \in \mathbb{Z}_2$, ne všechny nulové, tak, že

$$c_1 \bar{v}_1 + \dots + c_{k+1} \bar{v}_{k+1} = (0, \dots, 0) \in \mathbb{Z}_2^{\times k}.$$

Podíváme-li se na tuto kombinaci nad \mathbb{Z} , pak všechny složky výsledného vektoru jsou sudé:

$$c_1 \bar{v}_1 + \dots + c_{k+1} \bar{v}_{k+1} = (e_1, \dots, e_{k+1}) \in \mathbb{Z}^{\times k}, \quad 2 \mid e_i \text{ pro každé } i.$$

Koeficienty c_1, \dots, c_{k+1} nalezneme pomocí Gaussovy eliminace, která nad tělesem \mathbb{Z}_2 funguje.

Algoritmus SEF

2. fáze algoritmu SEF

Uvažujme všech $(k + 1)$ rovností tvaru $a_i^2 = p_1^{e_{i1}} \cdot \dots \cdot p_k^{e_{ik}}$ v \mathbb{Z}_n^* . Pokud každou i -tou rovnost umocníme na příslušné c_i a všechny rovnosti navzájem vynásobíme, získáme rovnost:

$$a^2 = p_1^{e_1} \cdot \dots \cdot p_k^{e_k} \text{ v } \mathbb{Z}_n^*,$$

kde $a = \prod_{i=1}^{k+1} a_i^{c_i}$ a všechna e_i jsou sudá.

Poznámka: $c_i \in \mathbb{Z}_2 = \{0, 1\}$, tudíž jsme vynásobili navzájem jen ty rovnosti, pro něž je $c_i = 1$. Rovnosti, pro něž je $c_i = 0$, umocněním degenerují na rovnost $1 = 1$.

Algoritmus SEF

2. fáze algoritmu SEF

Položme $b = p_1^{\frac{e_1}{2}} \cdot \dots \cdot p_k^{\frac{e_k}{2}} \in \mathbb{Z}_n^*$, přičemž $\frac{e_i}{2} \in \mathbb{N}$.

Z rovnosti:

$$a^2 = b^2 \in \mathbb{Z}_n^*$$

dostáváme rovnost:

$$(ab^{-1})^2 = 1 \in \mathbb{Z}_n^*$$

Našli jsme druhou odmocninu z jedné v \mathbb{Z}_n^* , a to $c = ab^{-1}$.

Pokud je $c \neq \pm 1$, najdeme faktor $\gcd(c - 1, n)$ čísla n .

Pokud je $c = \pm 1$, ohlásíme neúspěch.

Algoritmus SEF

- for $i \leftarrow 1$ to $k + 1$ do
 - repeat
 - choose $a_i \xleftarrow{\mathcal{U}} \mathbb{Z}_n^*$ at random
 - $m_i \leftarrow a_i^2$ in \mathbb{Z}_n
 - test if m_i is y -smooth (trial division)
 - until $m_i = p_1^{e_{i1}} \cdot \dots \cdot p_k^{e_{ik}}$ for some $e_{i1}, \dots, e_{ik} \in \mathbb{Z}$
 - $\bar{v}_i \leftarrow (e_{i1}, \dots, e_{ik})$ in $\mathbb{Z}^{\times k}$ enddo
- apply Gaussian elimination over \mathbb{Z}_2 to find $c_1, \dots, c_{k+1} \in \mathbb{Z}_2$, not all zero, such that $c_1 \bar{v}_1 + \dots + c_{k+1} \bar{v}_{k+1} = (0, \dots, 0)$ in $\mathbb{Z}_2^{\times k}$
- for $j \leftarrow 1$ to k do $e_j \leftarrow \sum_{i=1}^{k+1} c_i e_{ij}$ in \mathbb{Z} enddo
- $a \leftarrow \prod_{i=1}^{k+1} a_i^{c_i}$, $b \leftarrow p_1^{\frac{e_1}{2}} \cdot \dots \cdot p_k^{\frac{e_k}{2}}$, $c \leftarrow ab^{-1}$ in \mathbb{Z}_n
- if $c = \pm 1$ then output "failure"
else output $\gcd(c - 1, n)$ endif

Algoritmus SEF

Příklad

FaktORIZUJTE $n = 77$ a zvolte parametr hladkosti $y = 5$.

(77 není mocnina prvočísla, ani není dělitelné prvočísly $2, 3, 5 \leq y$.)

- 1. fáze - počítáme v \mathbb{Z}_{77}^* , náhodnou volbou získáme rovnosti:
 $R_1: 59^2 = 16 = 2^4$, odtud $\bar{v}_1 = (4, 0, 0)$.
 $R_2: 3^2 = 9 = 3^2$, odtud $\bar{v}_2 = (0, 2, 0)$.
 $R_3: 37^2 = 60 = 2^2 \cdot 3 \cdot 5$, odtud $\bar{v}_3 = (2, 1, 1)$.
 $R_4: 13^2 = 15 = 3 \cdot 5$, odtud $\bar{v}_4 = (0, 1, 1)$.
- 2. fáze - počítáme nad \mathbb{Z}_2 , kde $c_1 \bar{v}_1 + c_2 \bar{v}_2 + c_3 \bar{v}_3 + c_4 \bar{v}_4 = \bar{0}$ dává:
 $c_1(0, 0, 0) + c_2(0, 0, 0) + c_3(0, 1, 1) + c_4(0, 1, 1) = (0, 0, 0)$
Netriviální řešení je např. $c_1 = c_2 = 0, c_3 = c_4 = 1$.

Algoritmus SEF

Příklad - pokračování

- kompletování výpočtů - počítáme v \mathbb{Z}_{77}^* ,
 $R_1^0 \cdot R_2^0 \cdot R_3^1 \cdot R_4^1 = R_3 \cdot R_4$ dává rovnost:
 $(37 \cdot 13)^2 = 2^2 \cdot 3^2 \cdot 5^2$, tj. $19^2 = 30^2$ v \mathbb{Z}_{77}^* ,
 $c = 19 \cdot 30^{-1} = 34$ je netriviální druhá odmocnina z 1.
Odtud $\gcd(c - 1, n) = \gcd(33, 77) = 11$ je faktor čísla $n = 77$.

Poznámka:

- Netriviální řešení $c_1 = c_3 = c_4 = 0$, $c_2 = 1$ by vedlo k rovnosti
 $R_2: 3^2 = 3^2$ v \mathbb{Z}_{77}^* ,
 $c = 3 \cdot 3^{-1} = 1$ je triviální druhá odmocnina z 1.
Algoritmus by ohlásil neúspěch.

ANALÝZA ALGORTMU SEF

Tvrzení

Pravděpodobnost, že algoritmus SEF ohlásí neúspěch, je nejvýše $\frac{1}{2}$.

Důkaz

Rovnice $x^2 = 1$ má v \mathbb{Z}_n , pro liché $n = \prod_{i=1}^r p_i^{e_i}$, právě 2^r řešení. Lze dokázat, že každé řešení může být nalezeno algoritmem SEF se stejnou pravděpodobností.

Pak $P[\text{neúspěch}] = \frac{2}{2^r} = \frac{1}{2^{r-1}} \leq \frac{1}{2}$ díky předpokladu, že $r \geq 2$.

Analýza algoritmu SEF

Očekávaný čas algoritmu SEF

- 1. fáze SEF: Označme σ pravděpodobnost, že náhodný čtverec ze \mathbb{Z}_n^* je y -hladký. Pak očekávaný počet cyklů pro nalezení jednoho y -hladkého čtverce je $\frac{1}{\sigma}$.
V každém cyklu dělíme všemi k prvočísly do y ($y < n$).
Těchto y -hladkých čtverců potřebujeme najít $(k + 1)$.
 $E(\text{TIME1}) = O\left(\frac{k^2}{\sigma} \ln(n)^2\right)$
- 2. fáze SEF: Gaussova eliminace na matici typu $(k, k + 1)$ vyžaduje zhruba k^3 operací v \mathbb{Z}_2 a její čas bude dominantní pro druhou fázi. $\text{TIME2} = O(k^3 \ln(n)^2)$
- Očekávaný čas pro SEF: $E(\text{TIME}) = O\left(\left(\frac{k^2}{\sigma} + k^3\right) \ln(n)^2\right)$

Analýza algoritmu SEF

Očekávaný čas algoritmu SEF

Odhadneme k a σ pomocí y jako u algoritmu SEDL s následujícím doplněním:

Umíme odhadnout počet y -hladkých čísel do n . Díky předpokladu, že žádné $p_i \leq y$ nedělí n , jsou všechna tato y -hladká čísla v \mathbb{Z}_n^* .

Problém: V SEF hledáme náhodně y -hladké čtverce.

Kolik je ale y -hladkých čísel mezi čtverci v \mathbb{Z}_n^* ?

Odpověď: Zhruba stejně "husto". Lze dokázat, že pravděpodobnost trefy do y -hladkého čtverce v \mathbb{Z}_n je stejná jako pravděpodobnost trefy do y -hladkého čísla v \mathbb{Z}_n .

ANALÝZA ALGORTMU SEF

Očekávaný čas algoritmu SEF

Předpokládejme, že $y = e^{\ln(n)^{\lambda+o(1)}}$, kde $0 < \lambda < 1$.

- $\sigma = \frac{\Psi(y,n)}{|\mathbb{Z}_n^*|} \geq \frac{\Psi(y,n)}{n} \geq e^{(-1+o(1))\frac{\ln(n)}{\ln(y)}} \ln(\ln(n))$
- Dle Čebyševovy věty je $k = \pi(y) = \Theta\left(\frac{y}{\ln(y)}\right)$, odtud $k = e^{(1+o(1))\ln(y)}$.
- $\text{len}(n)^2 = e^{o(1)\ln(y)}$ díky našemu předpokladu pro y .

Analýza algoritmu SEF

Očekávaný čas algoritmu SEF

Dosadíme do $E(\text{TIME}) = O\left(\left(\frac{k^2}{\sigma} + k^3\right) \ln(n)^2\right)$ a získáme odhad:

$$E(\text{TIME}) \leq e^{(1+o(1)) \max\left\{\frac{\ln(n)}{\ln(y)} \ln(\ln(n)) + 2 \ln(y); 3 \ln(y)\right\}}$$

Nyní chceme zvolit parametr hladkosti y tak, aby byl odhad očekávaného času minimální.

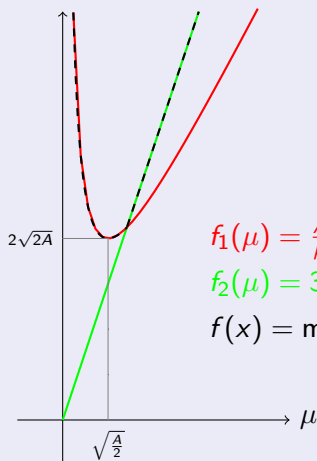
Označme $\mu = \ln(y)$, $A = \ln(n) \ln(\ln(n))$.

Minimum funkce $f(\mu) = \max\left\{\frac{A}{\mu} + 2\mu; 3\mu\right\}$ v exponentu nastane

v bodě $\mu = \sqrt{\frac{A}{2}}$, hodnota minima je $2\sqrt{2A}$ (výpočet viz SEDL).

Analýza algoritmu SEF

Očekávaný čas algoritmu SEF



$$f_1(\mu) = \frac{A}{\mu} + 2\mu,$$

$$f_2(\mu) = 3\mu$$

$$f(x) = \max\{f_1(\mu), f_2(\mu)\}$$

Analýza algoritmu SEF

Očekávaný čas algoritmu SEF

Odhad času bude minimální pro $y = e^{\frac{1}{\sqrt{2}} \sqrt{\ln(n) \ln(\ln(n))}}$.

Při tomto y bude očekávaný čas algoritmu SEF

$$E(\text{TIME}) \leq e^{(2\sqrt{2}+o(1))\sqrt{\ln(n) \ln(\ln(n))}},$$

tedy subexponenciální s konstantou $2\sqrt{2} \doteq 2,828$ v exponentu.

Poznámka

Konstantu v exponentu je možné zmenšit na 2,0, použijeme-li přesnější odhad počtu y -hladných čísel (viz věta 2).

Pro $y = e^{\frac{1}{2} \sqrt{\ln(n) \ln(\ln(n))}}$ bude $E(\text{TIME}) \leq e^{(2+o(1))\sqrt{\ln(n) \ln(\ln(n))}}$.

Kvadratické síto (QSF)

Algoritmus SEF s použitím kvadratického síta

Zrychlení algoritmu SEF získáme, když y -hladké čtverce v 1. fázi algoritmu nebudeme hledat náhodně, ale pomocí kvadratického síta. Konstanta v exponentu klesne na $1,0$.

Budeme potřebovat dva parametry:

- parametr hladkosti y
- parametr síta z

Budeme předpokládat, že pro oba platí:

$$y, z = e^{\ln(n)^{\frac{1}{2} + o(1)}} \doteq e^{\sqrt{\ln(n)}}$$

Kvadratické síto (QSF)

Algoritmus SEF s použitím kvadratického síta

Chceme faktorizovat n , které je liché, není to prvočíslo, ani mocnina prvočísla, není dělitelné žádným prvočíslem $p_i \leq y$, kterých je celkem k .

Kdybychom v 1. fázi algoritmu SEF volili všechna $a_i < \sqrt{n}$, pak bychom z y -hladkých čtverců $a_i^2 < n$ získali v závěru rovnost $a^2 = a^2$ v \mathbb{Z}_n (počítání modulo by se neprojevovalo).

Našli bychom druhou odmocninu z jedné $c = 1$ a algoritmus by ohlásil neúspěch. Aby měl algoritmus šanci na úspěch, musí volit aspoň některá a_i větší než \sqrt{n} .

Kvadratické síto (QSF)

Hledání y -hladkých čtverců

Položme $m = \lfloor \sqrt{n} \rfloor$, tedy $m \in \mathbb{N}$ je takové, že $m^2 \leq n < (m+1)^2$.
Uvažujme celočíselný polynom:

$$F(x) = (x + m)^2 - n$$

Pro $1 \leq s \leq z$ platí (díky předpokladu $z \doteq e^{\sqrt{\ln(n)}}$):

$$1 \leq F(s) \leq z^2 + 2z\sqrt{n} = n^{\frac{1}{2} + o(1)}$$

Aneb $n < (s + m)^2 \leq n + n^{\frac{1}{2} + o(1)}$, $F(s)$ je zbytek po vydělení čísla $(s + m)^2$ číslem n , tedy $F(s)$ je čtverec v \mathbb{Z}_n .

Kvadratické síto (QSF)

Hledání y -hladkých čtverců

Spočteme hodnoty $F(s)$ pro všechna $s = 1, \dots, \lfloor z \rfloor$.

Pokud je nějaké $F(s)$ y -hladké číslo, pak jsme našli y -hladký čtverec v \mathbb{Z}_n^* .

$$\text{Když } F(s) = (s + m)^2 - n = p_1^{e_1} \cdot \dots \cdot p_k^{e_k} \text{ v } \mathbb{Z},$$
$$\text{pak } (s + m)^2 = p_1^{e_1} \cdot \dots \cdot p_k^{e_k} \text{ v } \mathbb{Z}_n^*.$$

Faktorizace tohoto čtverce odpovídá jeho zbytku modulo n a díky předpokladu, že $p_i \nmid n$ pro každé i , je čtverec invertibilní.

Zbývá otázka, jak volit z , abychom mohli mezi hodnotami $F(s)$ najít dostatečný počet y -hladkých čtverců.

Kvadratické síto (QSF)

Hledání y -hladkých čtverců

Hustota y -hladkých čísel kolem \sqrt{n} je větší než kolem n .
Pravděpodobnost, že některá z hodnot $F(s)$ je y -hladká je větší než pravděpodobnost, že náhodný čtverec ze \mathbb{Z}_n^* je y -hladký.

Nechť $\tilde{\sigma}$ je pravděpodobnost, že náhodné číslo do \sqrt{n} je y -hladké,
 σ je pravděpodobnost, že náhodné číslo do n je y -hladké.

$$\begin{aligned}\tilde{\sigma} &= \frac{\Psi(y, \sqrt{n})}{\sqrt{n}} = e^{(-1+o(1)) \frac{\ln(\sqrt{n})}{\ln(y)}} \ln\left(\frac{\ln(\sqrt{n})}{\ln(y)}\right) = \\ &= e^{(-\frac{1}{4}+o(1)) \frac{\ln(n)}{\ln(y)}} \ln(\ln(n)) > \sigma = e^{(-1+o(1)) \frac{\ln(n)}{\ln(y)}} \ln(\ln(n))\end{aligned}$$

Použili jsme přesnější odhad pro $\Psi(y, \sqrt{n})$ a předpoklad $y \doteq e^{\sqrt{\ln(n)}}$. Už toto zaručí zrychlení 1. fáze algoritmu.

Kvadratické síto (QSF)

Určení parametru z

Parametr z stanovíme tak, abychom měli šanci najít mezi hodnotami $F(1), \dots, F(\lfloor z \rfloor)$ celkem $k + 1$ y -hladkých čtverců.

Je-li $\tilde{\sigma}$ pravděpodobnost, že náhodné číslo do \sqrt{n} je y -hladké, ($\tilde{\sigma}$ také odhaduje pravděpodobnost, že náhodný čtverec kolem \sqrt{n} je y -hladký), pak k nalezení jednoho y -hladkého čtverce potřebujeme prozkoumat průměrně $\frac{1}{\tilde{\sigma}}$ čísel tvaru $F(s)$.

Položíme tedy $z = \frac{k}{\tilde{\sigma}}$.

Pokud bychom nenašli dostatek y -hladkých čtverců, tak parametr z zdvojnásobíme a budeme pokračovat v hledání.

Kvadratické síto (QSF)

Určení parametru z

Poznámka: V našem odhadu parametru z je "podfuk". My přece nevolíme čísla náhodně!!!

Ve skutečnosti nevíme, kolik je y -hladkých čísel mezi hodnotami našeho polynomu $F(x)$ (neexistuje rigorózní důkaz). Algoritmus faktORIZACE POMOCÍ KVADRATICKÉHO SÍTA tedy může ohlásit neúspěch už během 1. fáze, protože nenajde dostatek y -hladkých čtverců. Zkušenost přesto ukazuje, že algoritmus QSF funguje a dokonce v očekávaném čase (heuristické ověření).

Kvadratické síto (QSF)

Sítovací procedura

Další zrychlení 1. fáze algoritmu zajistí sítovací procedura. Ověřování, zda je $F(s)$ y -hladké, nebudeme dělat pro každé $F(s)$ zvlášť, ale uděláme je pro všechna $F(1), \dots, F(\lfloor z \rfloor)$ najednou.

Vytvoříme pole V délky $\lfloor z \rfloor$, které inicializujeme:

$$V[s] \leftarrow F(s) \text{ pro všechna } s = 1, \dots, \lfloor z \rfloor$$

(Aneb bude zde subexponenciální prostorová složitost.)

Pokud vydělíme každé $V[s]$ všemi prvočísly $p_1, \dots, p_k \leq y$ tolikrát, kolikrát to půjde beze zbytku, tak nám y -hladká čísla propadnou sítem:

$F(s)$ je y -hladké, právě když po dělení vš. $p_i \leq y$ je $V[s] = 1$.

Kvadratické síto (QSF)

Sítovací procedura

Urychlení spočívá v tom, že libovolným prvočíslem $p \leq y$ budeme dělit jen ta $F(s)$ (resp. $V[s]$), která jimi dělitelná jsou:

$p \mid F(s)$, právě když $F(s) = 0$ v \mathbb{Z}_p , což nastane,
právě když s (resp. $[s]_p \in \mathbb{Z}_p$) je kořenem $F(x)$,
který chápeme jako polynom nad \mathbb{Z}_p .

Kvadratický polynom $F(x) = (x + m)^2 - n$ může mít v tělese \mathbb{Z}_p nejvýše dva kořeny, označme je s_1, s_2 .

Prvočíslem p jsou dělitelná právě ta $F(s)$, pro něž $s = s_j + lp$, kde $s_j \in \{s_1, s_2\}$ a $l \in \mathbb{N}$ libovolné takové, aby vyšlo $s \leq [z]$. (Zapamatujeme si též "kolikrát" lze $F(s)$ dělit tímto p , kvůli prvočíselnému rozkladu.)

Kvadratické síto (QSF)

Kořeny kvadrátu nad \mathbb{Z}_p

Chceme najít kořeny polynomu $F(x) = (x + m)^2 - n$ v tělese \mathbb{Z}_p .

- Nad \mathbb{Z}_2 je $F(x) = x^2 + m^2 - n$, kde n je liché (předpoklad).
Pro m sudé je $F(x) = x^2 - 1$ a má dvojnásobný kořen $1 \in \mathbb{Z}_2$.
Pro m liché je $F(x) = x^2$ a má dvojnásobný kořen $0 \in \mathbb{Z}_2$.
- Nad \mathbb{Z}_p pro $p > 2$ je $F(x) = 0$, právě když $(x + m)^2 = n$ v \mathbb{Z}_p .
Není-li n čtverec v \mathbb{Z}_p^* , pak $F(x)$ nemá kořen v \mathbb{Z}_p .
Je-li $n = (\pm d)^2$ čtverec v \mathbb{Z}_p^* , pak $F(x)$ má dva kořeny v \mathbb{Z}_p ,
a to $-m \pm d$. (Přitom $n \in \mathbb{Z}_p^*$ díky předpokladu $p \nmid n$.)

Kvadratické síto (QSF)

Kořeny kvadrátu nad \mathbb{Z}_p

Pro hledání kořenů v \mathbb{Z}_p , kde $p > 2$ je prvočíslo, potřebujeme:

- 1 Umět poznat čtverce v \mathbb{Z}_p^* :

Eulerovo kritérium: $a \in \mathbb{Z}_p^*$ je čtverec, jen když $a^{\frac{p-1}{2}} = 1$ v \mathbb{Z}_p .

- 2 Umět počítat druhé odmocniny v \mathbb{Z}_p^* :

- Je-li $p \equiv 3 \pmod{4}$, pak čtverec $a \in \mathbb{Z}_p^*$ má druhé odmocniny $\pm b = \pm a^{\frac{p+1}{4}}$ v \mathbb{Z}_p^* .

- Pro libovolné p existuje algoritmus na hledání druhých odmocnin v \mathbb{Z}_p^* , který pracuje v čase

$O(\text{len}(p)^3 + h \text{len}(h) \text{len}(p)^2) \subseteq O(\text{len}(p)^3 \text{len}(\text{len}(p)))$,
kde $p - 1 = 2^h \tilde{m}$, \tilde{m} je liché.

Sítovací procedura

Let p_1, \dots, p_k are all primes upto y .

- for $s \leftarrow 1$ to $\lfloor z \rfloor$ do $V[s] \leftarrow F(s)$ enddo
- for $i \leftarrow 1$ to k do
 - find roots of $F(x)$ in \mathbb{Z}_{p_i} (there are at most two of them)
 - for every root s_j do
 - $s \leftarrow s_j$
 - while $s \leq \lfloor z \rfloor$ do
 - $e \leftarrow 0$
 - repeat $V[s] \leftarrow \frac{V[s]}{p_i}$, $e \leftarrow e + 1$
 - until $p_i \nmid V[s]$
 - put in list of divisors $D[s]$ for s prime power p_i^e
 - $s \leftarrow s + p_i$ enddo
 - enddo, enddo
- $F(s)$ is y -smooth iff $V[s] = 1$

Kvadratické síto (QSF)

Čas běhu síťovací procedury

Předpokládáme, že $y, z = e^{\ln(n)^{\frac{1}{2}+o(1)}} \doteq e^{\sqrt{\ln(n)}}$.

Odtud $\text{len}(y) \doteq \text{len}(n)^{\frac{1}{2}}$.

- Inicializace pole V trvá $O(z \text{len}(n)^2)$.
- Výpočet kořenů polynomů $F(x)$ nad všemi \mathbb{Z}_{p_i} (všech prvočísel $p_i \leq y$ je k , tedy $k < y \doteq z$) trvá zhruba $O(k \text{len}(y)^4) = O(k \text{len}(n)^2)$.
- Vlastní síťování trvá $O(\sum_{p \leq y} \frac{z}{p} \text{len}(p) \text{len}(n)^2)$, což je zhruba $O(z \text{len}(n)^3)$. To je čas dominantní nad předchozími časy.

Kvadratické síto (QSF)

Čas běhu síťovací procedury

Podívejme se podrobněji na odhad času pro síťování.

- Pro každé prvočíslo $p_i \leq y$ najdeme nejvýše dva kořeny s_1, s_2 a každý kořen dá $\frac{z}{p_i}$ hodnot $F(s)$ dělitelných p_i .
Přitom jedno $F(s)$ lze dělit číslem p_i nejvýše $\log_{p_i}(F(s))$ –krát, což je $O(\text{len}(n))$ dělení. Pro p_i potřebujeme čas $O(\frac{z}{p_i} \text{len}(p_i) \text{len}(n)^2) = O(\frac{z}{p_i} \text{len}(n)^{2,5})$.
- Sečteme časy pro všechna p_i :
 $O(\sum_{i=1}^k \frac{z}{p_i} \text{len}(n)^{2,5}) = O(z \text{len}(n)^3)$.
Odhadli jsme sumu integrálem:

$$\sum_{p_i \leq y} \frac{1}{p_i} \leq \int_1^y \frac{1}{\tilde{y}} d\tilde{y} = [\ln \tilde{y}]_1^y = \ln y \in O(\text{len}(n)^{\frac{1}{2}})$$

Algoritmus QSF - 1'st stage

- $m \leftarrow \lfloor \sqrt{n} \rfloor$, $F(x) = (x + m)^2 - n$
- repeat
 - use the sieving procedure with parameter z
(it creates fields V and D of length $\lfloor z \rfloor$)
 - $z \leftarrow 2z$
- until $V[s] = 1$ for at least $k + 1$ different values of s
- for the first $k + 1$ values of s such that $V[s] = 1$ do
 - $a_i \leftarrow s + m$
 - find in $D[s]$ the factorization of $a_i^2 = p_1^{e_{i1}} \cdot \dots \cdot p_k^{e_{ik}}$ in \mathbb{Z}_n
 - $\bar{v}_i \leftarrow (e_{i1}, \dots, e_{ik})$ in $\mathbb{Z}^{\times k}$ enddo

2. fáze je stejná jako v SEF - Gaussova eliminace nad \mathbb{Z}_2 a dopočítání druhé odmocniny z 1. Pokud je tato různá od ± 1 , najdeme faktor n .

Analýza algoritmu QSF

Očekávaný čas algoritmu QSF

- 1. fáze QSF: $E(\text{TIME1}) = O(z \text{len}(n)^3) = O(\frac{k}{\tilde{\sigma}} \text{len}(n)^3)$
- 2. fáze QSF: $\text{TIME2} = O(k^3 \text{len}(n)^2)$
- Očekávaný čas pro QSF: $E(\text{TIME}) = O((\frac{k}{\tilde{\sigma}} + k^3) \text{len}(n)^3)$

Dosadíme-li za k a $\tilde{\sigma}$ (podobně jako v SEF), získáme odhad:

$$E(\text{TIME}) \leq e^{(1+o(1)) \max\{\frac{1}{4} \frac{\ln(n)}{\ln(y)} \ln(\ln(n)) + \ln(y); 3 \ln(y)\}}$$

Analýza algoritmu QSF

Volba parametru y

Chceme zvolit parametr hladkosti y tak, aby byl odhad minimální.

Označme $\mu = \ln(y)$, $A = \ln(n) \ln(\ln(n))$.

Chceme najít minimum pro funkci $f(\mu) = \max\{\frac{1}{4}\frac{A}{\mu} + \mu; 3\mu\}$.

Funkce $f_1(\mu) = \frac{1}{4}\frac{A}{\mu} + \mu$ nabývá minima v bodě $\mu = \frac{\sqrt{A}}{2}$,
hodnota minima je \sqrt{A} .

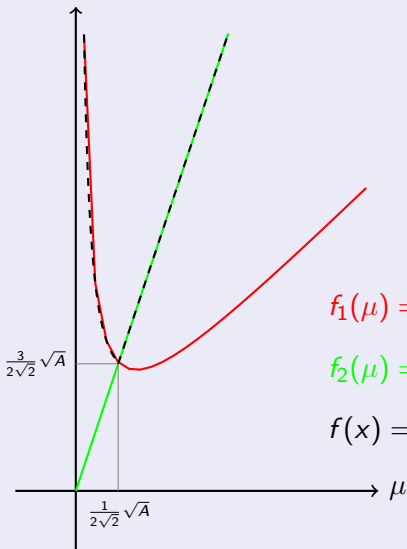
Funkce $f_2(\mu) = 3\mu$ nabývá v tomto bodě hodnoty $\frac{3}{2}\sqrt{A} > \sqrt{A}$.

Protože je funkce $f_2(\mu)$ rostoucí, bude bod minima funkce $f(\mu) = \max\{f_1(\mu); f_2(\mu)\}$ před bodem minima funkce $f_1(\mu)$, bude to bod, v němž se grafy těchto funkcí protínají:

$$f_1(\mu) = f_2(\mu) \text{ pro } \mu = \frac{1}{2\sqrt{2}}\sqrt{A}$$

Hodnota minima funkce $f(\mu)$ je $\frac{3}{2\sqrt{2}}\sqrt{A}$.

Volba parametru y



$$f_1(\mu) = \frac{1}{4} \frac{A}{\mu} + \mu,$$

$$f_2(\mu) = 3\mu$$

$$f(x) = \max\{f_1(\mu), f_2(\mu)\}$$

Analýza algoritmu QSF

Očekávaný čas algoritmu QSF

Volíme parametr hladkosti: $y = e^{\frac{1}{2\sqrt{2}}\sqrt{A}} = e^{\frac{1}{2\sqrt{2}}\sqrt{\ln(n)\ln(\ln(n))}}$

Potom bude parametr síta: $z = \frac{k}{\sigma} = e^{(\frac{3}{2\sqrt{2}}+o(1))\sqrt{\ln(n)\ln(\ln(n))}}$

(Všimněme si, že y i z splňují předpoklady našeho výpočtu.)

Při těchto y a z bude očekávaný čas algoritmu QSF

$$E(\text{TIME}) \leq e^{(\frac{3}{2\sqrt{2}}+o(1))\sqrt{\ln(n)\ln(\ln(n))}},$$

tedy subexponenciální s konstantou $\frac{3}{2\sqrt{2}} \doteq 1,061$ v exponentu.

Analýza algoritmu QSF

Očekávaný čas algoritmu QSF

Při určování parametru y nás vlastně zdržela Gaussova eliminace. Matice, kterou eliminujeme, je řídká - obsahuje exponenty prvočísel ve faktorizaci nalezených y -hladkých čtverců.

Použijeme-li speciální algoritmy pro řešení soustavy k rovnic o $(k + 1)$ neznámých s řídkou maticí, půjde to v čase $O(k^{2+o(1)})$.

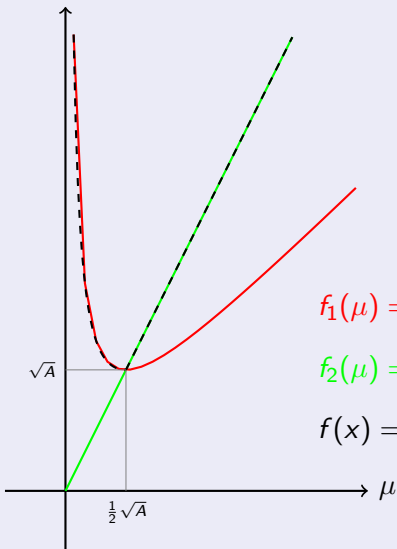
Potom bude funkce $f_2(\mu) = 2\mu$ a nalezený bod minima $\mu = \frac{\sqrt{A}}{2}$ funkce $f_1(\mu)$ je zároveň bodem minima funkce $f(\mu)$.

Hodnota minima bude \sqrt{A} .

V tomto případě pro parametr hladkosti $y = e^{\frac{1}{2}\sqrt{\ln(n)\ln(\ln(n))}}$ bude $z = e^{(1+o(1))\sqrt{\ln(n)\ln(\ln(n))}}$ a očekávaný čas algoritmu QSF

$$E(\text{TIME}) \leq e^{(1+o(1))\sqrt{\ln(n)\ln(\ln(n))}}.$$

Volba parametru y



$$f_1(\mu) = \frac{1}{4} \frac{A}{\mu} + \mu,$$

$$f_2(\mu) = 2\mu$$

$$f(x) = \max\{f_1(\mu), f_2(\mu)\}$$

Dodatek

Další subexponenciální algoritmy na faktorizaci

- Algoritmus používající síto nad číselným tělesem pracuje v očekávaném čase

$$E(\text{TIME}) \leq e^{(c+o(1)) \ln(n)^{\frac{1}{3}} \ln(\ln(n))^{\frac{2}{3}}},$$

kde zatím známá nejmenší konstanta $c = 1,902$ (heuristicky ověřeno).

- Faktorizace přes eliptické křivky má očekávaný čas

$$E(\text{TIME}) \leq e^{(\sqrt{2}+o(1))\sqrt{\ln(p)\ln(\ln(p))}} \ln(n)^{O(1)},$$

kde p je nejmenší prvočíslo, které dělí n (heuristicky ověřeno). Tento algoritmus má, na rozdíl od ostatních, polynomiální prostorovou náročnost.

Algoritmus SEF a QSF

Literatura

- Shoup: A Computational Introduction to Number Theory and Algebra. Kapitola 15.
- Lineární prostory nad tělesem najdete tamtéž v kapitole 13.
<http://shoup.net/ntb/>