Diskrete Logarithmen

Ron-Gerrit Vahle   Hendrik Radke

Universität Potsdam
Institut für Informatik

Seminar Kryptographie SS2005

Teil II

Gliederung

Algorithmen

- Pohlig-Hellman
- Index-Calculus
- Theoretische Grenzen

Endliche Körper

Epilog

Pohlig-Hellman-Algorithmus

- Primfaktorzerlegung von $n = \prod_{i=1}^{k} p_i^{c_i}$
- Berechne $x_i = a \mod p_i^{c_i}$
- Berechnung von a nach Gauß-Algorithmus:

$$a = \sum_{i=1}^{k} x_i M_i y_i, M_i = n/p_i^{c_i}, y_i = M_i^{-1} \mod p_i^{c_i}$$

Pohlig-Hellman: Berechnung von x_i

- Sei $x = x_i, q = p_i, c = c_i$
- $\Rightarrow x = a \mod q^c$
- x ist darstellbar als $x = \sum_{i=0}^{c-1} b_i q^i, b_i < q$
- $\exists s \in \mathbb{N}. a = x + sq^c \Rightarrow a = \sum_{i=0}^{c-1} b_i q^i + sq^c$
- Definiere $\beta_0 := \beta, \beta_j := \beta\alpha^{-(b_0 + b_1 q + \cdots + b_{j-1} q^{j-1})}$
- Es gilt: $\beta_j^{n/q^{j+1}} = \alpha^{b_j n/q}$ (zu zeigen)
- $\gamma = \alpha^{n/q} \Rightarrow \exists b_j \in \mathbb{N}. \gamma^{b_j} = \beta^{n/q^{j+1}}$ ($\mathcal{O}(q)$)

Pohlig-Hellman: Beweise I

zu zeigen: $\beta_j^{n/q^{j+1}} = \alpha^{b_j n/q}$

$$
\begin{aligned}
&\beta_j^{n/q^{j+1}} && (\beta_j = \beta\alpha^{-b_i q^i}, \beta = \alpha^a)\\
&= (\alpha^{a-(b_0+b_1 q+\cdots+b_{j-1}q^{j-1})})^{n/q^{j+1}} && (a = \textstyle\sum_{i=0}^{c-1} b_i q^i + sq^c)\\
&= (\alpha^{b_j q^j+\cdots+b_{c-1}q^{c-1}+sq^c})^{n/q^{j+1}} && (K_j = b_{j+1} + b_{j+2}q + \cdots)\\
&= (\alpha^{b_j q^j + K_j q^{j+1}})^{n/q^{j+1}} && (\text{Potenzgesetze})\\
&= \alpha^{b_j n/q}\alpha^{K_j n} && (\alpha^q \equiv 1 \pmod{n})\\
&= \alpha^{b_j n/q}
\end{aligned}
$$

$\square$

Pohlig-Hellman: Beweise II

Chinesisches Rest-Theorem – Gauß-Algorithmus

- Gegeben: $x_1, \ldots, x_r, z_1, \ldots, z_r \in \mathbb{N}$, z_i paarw. teilerfremd
- Gesucht: a, so daß $\forall i . a \equiv x_i \pmod{z_i}$
- definiere $M_i := \frac{n}{z_i}, \quad y_i = M_i^{-1} \pmod{z_i}$
- Sei $a = \sum_{i=1}^{r} x_i M_i y_i \mod n$
- $x_i M_i y_i \equiv x_i \pmod{z_i}$ $\quad (M_i y_i \equiv 1 \pmod{z_i})$
- $x_i M_i y_i \equiv 0 \pmod{m_j}$, falls $i \neq j$ $\quad (m_j | M_i)$
- $\forall j \in \{1, \ldots, r\} . a \equiv \sum_{i=1}^{r} x_i M_i y_i \equiv x_j \pmod{m_j}$

$\square$

Pohlig-Hellman: Ein Beispiel (I)

- Sei $p = 29, \alpha = 2, \beta = 18 \Rightarrow n = p - 1 = 28 = 2^2 7^1$
- Berechne $x_1 = a \mod 2^2$
  - Setze $q = 2, c = 2$
  - $\gamma = \alpha^{n/q} = 2^{14} \mod 29 = 28$
  - $\beta_0^{n/q^1} = 18^{14} \mod 29 = 28$
  - $\gamma^1 \equiv \beta_0^{n/q} \pmod{29}$ $(\Rightarrow b_0 = 1)$
  - $\beta_1^{n/q^2} = \beta\alpha^{-b_0}{}^7 = 18 * 2^{-1}{}^7 \mod 29 = 28$
  - $\gamma^1 \equiv 28 \pmod{29}$ $(\Rightarrow b_1 = 1)$
  - $x_1 = \sum_{i=0}^{1} b_i q^i = 3$

Pohlig-Hellman: Ein Beispiel (II)

- Berechne $x_2 = a \mod 7^1$
  - Setze $q = 7, c = 1$
  - $\gamma = \alpha^{n/q} = 2^4 \mod 29 = 16$
  - $\beta_0^{n/q^1} = 18^4 \mod 29 = 25$
  - $\gamma^4 \equiv 25 \pmod{29}$ $(\Rightarrow b_0 = 4)$
  - $x_2 = \sum_{i=0}^{0} b_i q^i = 4$
- $a = \sum_{i=1}^{2} x_i M_i y_i, M_i = n/p_i^{c_i}, y_i = M_i^{-1} \mod p_i^{c_i}$
- $x_1 = 3, x_2 = 4, M_1 = 7, M_2 = 4, y_1 = 3, y_2 = 2$
- $a \equiv 11 \pmod{28}$
- Probe: $2^11 \mod 29 = 18$

Pohlig-Hellman: Komplexität

- Berechnung der Primfaktorzerlegung von n — $(???)$
- Berechnung von $x_i, 1 \leq i \leq r$ — $\mathcal{O}(r)$
  - c_i Faktoren b_i — $\mathcal{O}(c_i)$
    - Berechnung der b_i — $\mathcal{O}(\sqrt{p_i})$
- Berechnung von a nach Gauß:
  - Summe aus r Elementen — $\mathcal{O}(r)$
    - Ermitteln der Inversen y_i — $\mathcal{O}(\log p_i)$
- Größe von r vernachlässigbar
- Gesamtkomplexität $(c = \max\{c_i\}, q = \mathrm{ax}\{p_i\})$: — $\mathcal{O}(c\sqrt{q})$

Pohlig-Hellman: Grenzen

- Primfaktorisierung von $n = p_1^{c_1} \cdots p_k^{c_k}$ ist exponentiell
- nur effektiv, falls alle p_i schnell gefunden werden
- gilt für s-glatte Zahlen, d.h. alle $p_i < s$, s hinreichend klein

Index-Calculus-Algorithmus (I)

1. Bestimme eine Faktorbasis $\mathcal{B} = p_1, \ldots, p_r \subseteq G$ mit r „kleinen“ Primzahlen, die eine hinreichend große Menge von $x \in G$ durch Linearkombination darstellen können
2. Berechne $\log_\alpha p_i, \forall p_i \in \mathcal{B}$ (Vorberechnung)
3. wähle eine zufällige Zahl $s \in \mathbb{N}$
4. versuche Faktorisierung mit $\mathcal{B}$: $\beta\alpha^s = \prod_{i=1}^{r} p_i^{c_i}$
5. falls (4) fehlschlägt, wiederhole (3)
6. sonst gilt: $\beta\alpha^s \equiv \prod_{i=1}^{r} p_i^{c_i} \pmod{p}$
   $\Rightarrow \log_\alpha \beta + s \equiv \sum_{i=1}^{r} c_i \log_\alpha p_i \pmod{p-1}$

Index-Calculus-Algorithmus (II)

Vorberechnung der Logarithmen von $\mathcal{B}$

1. Wahl einer zufälligen Zahl $k, 0 \leq k \leq n-1$
2. Schreibe α^k als Produkt aus Elementen von $\mathcal{B}$: $\alpha^k = \prod_{i=1}^{r} p_i^{c_i}$
   $\Rightarrow k \equiv \sum_{i=1}^{r} c_i \log_\alpha p_i \pmod{n}$
3. Wiederhole Schritte 1-2, bis $r+t$ Gleichungen gefunden sind, t ist „klein“, z.B. 10
4. Lösen des lin. Gleichungssystems mit $r+t$ Gleichungen und r Unbekannten $\log_\alpha p_i$

Index Calculus – ein Beispiel (I)

Vorberechnung

- Sei $p = 10007, \alpha = 5$
- Benutze Faktorbasis $\mathcal{B} = \{2, 3, 5, 7\}$
- Bestimme $\log_\alpha p_i$
  - $\log_5 5 = 1$
  - $5^{4063} \mod 10007 = 42 = 2 * 3 * 7$
  - $5^{5136} \mod 10007 = 54 = 2 * 3^3$
  - $5^{9865} \mod 10007 = 189 = 3^3 * 7$
  - $4063 \equiv \log_5 2 + \log_5 3 + \log_5 7 \pmod{10006}$
  - $5136 \equiv \log_5 2 + 3 \log_5 3 \pmod{10006}$
  - $9865 \equiv 3 \log_5 3 + \log_5 7 \pmod{10006}$
  - Lösen des lin. Gleichungssystems:
  - $\log_5 2 = 6578, \log_5 3 = 6190, \log_5 7 = 1301$

Index Calculus – ein Beispiel (II)

- Sei $\beta = 9451$
- Wähle „zufälliges" $k = 7736$
- $9451 * 5^{7736} \mod 10007 = 8400 = 2^4 * 3^1 * 5^2 * 7^1$
- $\log_\alpha \beta = (4 \log_5 2 + 1 \log_5 3 + 2 \log_5 5 + 1 \log_5 7) \mod 10006$
- $= 6057$
- Probe: $5^{6057} \mod 10007 = 9451$

Index Calculus – Grenzen und Komplexität

- Kardinalität r der Faktorbasis wichtig:
  - zu klein $\Rightarrow$ zu wenig $x \in G$ faktorisierbar
  - zu groß $\Rightarrow$ Berechnung wird komplizierter
- Ermitteln der Komplexität durch heuristische Analyse
- Vorberechnung: $\mathcal{O}(e^{(1+\Omega(1))\sqrt{\ln p \ln \ln p}})$
- Lösung für geg. β: $\mathcal{O}(e^{(1/2+\Omega(1))\sqrt{\ln p \ln \ln p}})$

Theoretische Grenzen (I)

- allgemeiner Algorithmus muss auf beliebigen Gruppen funktionieren
- alle zyklischen Gruppen $(G, \cdot)$ der Ordnung n sind isomorph
- DLP in $(\mathbb{Z}_n, +)$ ist trivial
- Idee: Finde bijektive Abbildung $\varphi : G \to \mathbb{Z}_n$, so dass $\varphi(x \cdot y) = (\varphi(x) + \varphi(y)) \mod n$

Theoretische Grenzen (II)

- Sei $\sigma : \mathbb{Z}_n \to G$ eine injektive Abbildung
- Gegeben: $\sigma(1), \sigma(a)$
- Gesucht: $a = \sigma^{-1}(\sigma(a))$
- Orakel berechnet für m disjunkte Paare (c_i, d_i) Linearkombination $\sigma_i = \sigma(c_i * 1 + d_i * a) \mod n$
- Kollision: $\sigma_i = \sigma_j, i \neq j$
- (c_i, d_i) disjunkt $\Rightarrow c_i \neq c_j, d_i \neq d_j$
- $a = (c_i - c_j)(d_j - d_i)^{-1} \mod n$

Theoretische Grenzen (III)

Wie wahrscheinlich ist eine Berechnung von a?

- m disjunkte Paare $(c_i, d_i) \in M \Rightarrow$ max. $\binom{m}{2} =: g$ Kollisionen
- Wahrscheinlichkeit für $a \in \mathrm{Kollision}(M)$: g/n
- sonst: errate a aus Differenzmenge $G \setminus M$
- Wahrscheinlichkeit für richtiges Raten: $\frac{n-g}{n} * \frac{1}{n-g}$
- $\frac{g}{n} + \frac{1}{n-g} * \frac{n-g}{n} = \frac{g+1}{n}$
- $\frac{(\binom{m}{2}+1)}{n} \geq 1 \Rightarrow m^2 + m \geq 2n \Rightarrow \Omega(\sqrt{n})$

Endliche Körper

- Statt $\mathbb{Z}_p$: Körper $\mathbb{F}_{p^n}$ über Polynomen
- Sei $\mathbb{Z}_p[x]$ Menge der Polynome $\sum_{i=0}^{n} a_i x^i$, $a_i \in \mathbb{Z}_p$, $n \in \mathbb{N}$, p prim
- Sei $\mathbb{Z}_p[x]/f(x) = \{g \in \mathbb{Z}_p[x] | \mathrm{grad}(g) < n, n = \mathrm{grad}(f)\}$
- $g(x) \equiv h(x) \pmod{f(x)} \Leftrightarrow f(x) | (g(x) - h(x))$ („modulo“)
- $f \in \mathbb{Z}_p[x]$ irreduzibel $\Leftrightarrow \forall f_1, f_2 \in \mathbb{Z}_p[x]. f_1 f_2 \neq f$ („prim“)
- Anpassung von Index-Calculus möglich
- Komplexität von IC: $\mathcal{O}(e^{(1.405+\Omega(1))\sqrt[3]{2n \ln n}})$ Vorberechnung
  $\mathcal{O}(e^{(1.098+\Omega(1))\sqrt[3]{2n \ln n}})$ Problem

Zusammenfassung

- $\text{DLP} := \{(\alpha, \beta, a) | \alpha, \beta \in G, a \in \mathbb{N}, \alpha^a = \beta\}$
- Diskrete Logarithmen werden zur public-Key-Verschlüsselung (ElGamal) eingesetzt
- übliche Gruppen: $\mathbb{Z}_p^*, \mathbb{F}_{p^n}$, elliptische Kurven *(nächste Woche)*
- für allgem. Gruppen sicher: Komplexität $\sqrt{n}$
- Aber:
  - min. ein „großer" Primfaktor nötig (Pohlig-Hellman)
  - Index-Calculus *kann* Resultat in subexponentieller Zeit ermitteln

Quellen

A Menezes, P. van Oorschot, S. Vanstone.
Handbook of Applied Cryptography.
CRC, 1996.

D. Stinson.
Cryptography – Theory and Practice.
Chapman&Hall/CRC, 2002.

Samuel S. Wagstaff.
Cryptanalysis of number theoretic Ciphers.
Chapman&Hall/CRC, 2003.

Vielen Dank für Ihre Aufmerksamkeit

Fragen?