Algorithmen zur Berechnung diskreter Logarithmen

- Das Problem des 'diskreten Logarithmus' ist die Basis vom Diffie-Hellman und El-Gamal-Verfahren.
- Etablierte Annahme: Dieses Problem ist schwierig zu lösen (Grundlage für Verwendung der obigen Verfahren).
- Inhalt: Übersicht über die bekannten Algorithmen für Logarithmus-Berechnungen.

Beschreibung des Problems

Recap: Diskrete Logarithmen

- **Problemstellung:** Lösung der Gleichung $g^x = a$ **in einer Gruppe**
- **Anders ausgedrückt:** Bestimmung von $\log_g(a)$.

Zusammenhang zu Verschlüsselungsverfahren

Recap: Diffie Hellman und El Gamal Verfahren

- Diffie Hellman:
  - private Schlüssel: a, b
  - öffentlich bekannte Werte: g^a, g^b
  - gemeinsamer geheimer Schlüssel: $g^{ab} \pmod p$
- El Gamal (leicht vereinfacht):
  - private Schlüssel: a, b
  - öffentlich bekannte Werte: g^a, g^b
  - Verschlüsselung von Bob: $c = g^{ab} \pmod p$

Bem: In beiden Fällen sind a, b diskrete Logarithmen.
(Jeder effiziente Algorithmus für diskrete Logarithmen knackt automatisch die beiden obigen Algorithmen.)

Übersicht

Inhalt

Strategie 1: Enumeration

Vorgehen: Die Zahlen $x = 1, 2, 3, \ldots, n$ der Reihe nach durchgehen und prüfen, ob die Gleichung $a^x = g$ erfüllt ist.

- Algorithmus hat exponentielle Laufzeit ($\rightarrow$ nicht praktikabel für grosse n).

Strategie 2: Baby Step – Giant Step Algorithmus

Bem: Dieser Algorithmus hat im Vergleich zur Enumeration:

+ eine schnellere Laufzeit
- grösseren Speicher-Bedarf

Vorüberlegungen I

- Ganzzahl-Division: 50 : 8 = 6, Rest 2. Somit: $50 = 6 \cdot 8 + 2$.
- Ganzzahl-Division: 77 : 8 = 9, Rest 5. Somit: $77 = 9 \cdot 8 + 5$.
- **Allgemein:** Jede ganze Zahl x lässt sich darstellen als $x = q \cdot 8 + r$.
- **Noch allgemeiner:** Für jedes m gilt: Jede ganze Zahl lässt sich darstellen als $x = q \cdot m + r$.

Strategie 2: Baby Step – Giant Step Algorithmus

Recap: Gesucht: Lösung der Gleichung $g^x = a$ (in einer Gruppe G).

Vorüberlegungen II:

- Setze $n :=$ Ordnung der betrachteten (zyklischen) Gruppe G.
- Setze $m := \lceil \sqrt{n} \rceil$
- Ansatz für x: $x = q \cdot m + r$ für entsprechende Werte q,r.
  (Hinweis: q, r sind spezifiert durch die Ganzzahl-Division $x : m$.)
- Einsetzen des Ansatzes gibt: $g^x = g^{qm+r} = g^{qm} \cdot g^r \stackrel{!}{=} a$
- Diese Gleichung lässt sich umformen zu $g^r = a \cdot g^{-qm} = a \cdot (g^{-m})^q$
- Verbleibende Gleichung ist somit: $\underbrace{g^r}_{\text{Baby Steps}} = \underbrace{a \cdot \left(g^{-m}\right)^q}_{\text{Giant Steps}}$

Grundidee des Algorithmus

- Berechnung von g^r für alle r mit $0 \leq r \leq m - 1$ (Baby Steps)
- Berechnung von $a \cdot (g^{-m})^q$ für alle q mit $0 \leq q \leq \frac{n}{m}$ (Giant Steps)
- Suche nach einem Paar (r, q), bei dem obige 2 Werte gleich sind.

Bem: m wurde so gewählt, dass #(Baby Steps) $\approx$ #(Giant Steps) ist.

Eigentlicher Algorithmus

Input: Gruppe G und ein Element g

$n := |G|, \quad m := \lceil \sqrt{n} \rceil$.

for $j \in \{0, 1, 2, \ldots, m-1\}$

Berechne (j, g^j) (in der Gruppe G). // Baby Step

end

$h := g^{-m}$

for $i \in \{0, 1, 2, \ldots, \lceil \frac{n}{m} \rceil\}$

Berechne (i, ah^i) (in der Gruppe G). // Giant Step

Prüfe, ob es (aus den Baby Steps) ein j gibt mit $g^j = ah^i$.

if (Prüfung erfolgreich)

return $x := im + j$ // Giant Step

end

end

Beispiel

Aufgabe: Bestimme den diskreten Logarithmus von 57 zur Basis 3 in der Gruppe $\mathbb{Z}^*_{113}$