

Kryptologie

Johannes Köbler



Institut für Informatik
Humboldt-Universität zu Berlin

WS 2022/23

Berechnung des diskreten Logarithmus

Zur Erinnerung:

- Sei $(G, \circ, 1)$ eine Gruppe und sei $\alpha \in G$
- Weiter bezeichne $[\alpha] = \{\alpha^i \mid i = 0, \dots, n-1\}$ die von α in G erzeugte Untergruppe, wobei $n = \text{ord}_G(\alpha) = \min\{e \geq 1 \mid \alpha^e = 1\}$ die Ordnung von α ist
- Dann heißt die eindeutig bestimmte Zahl $e \in \{0, \dots, n-1\}$ mit $\beta = \alpha^e$ **diskreter Logarithmus $\log_{G,\alpha}(\beta)$ von β zur Basis α in G**

Das diskrete Logarithmusproblem (DLP):

Gegeben: (Beschreibung einer) Gruppe G , ein Element $\alpha \in G$ und die Ordnung $n = \text{ord}_G(\alpha)$ von α in G sowie ein Element $\beta \in [\alpha]$

Gesucht: Der diskrete Logarithmus $e = \log_{\alpha}(\beta)$ von β zur Basis α

- In vielen Gruppen ist die Funktion $e \mapsto \alpha^e$ effizient mittels wiederholtem Quadrieren und Multiplizieren berechenbar
- Bei geeigneter Wahl von G und α ist jedoch kein effizienter Algorithmus zur Berechnung der Umkehrfunktion, also von $\beta \mapsto \log_\alpha(\beta)$ bekannt, d.h. $e \mapsto \alpha^e$ ist ein Kandidat für eine Einwegfunktion

Beispiel

- Sei $G = \mathbb{Z}_p^*$, p prim, und sei α ein Erzeuger von $\mathbb{Z}_p^*$
- Dann ist $[\alpha] = \mathbb{Z}_p^*$ und α hat die Ordnung $n = p - 1$
- Ist p hinreichend groß und enthält $p - 1$ mindestens einen großen Primfaktor, so sind keine effizienten Algorithmen zur Berechnung von $\log_\alpha(\beta)$ in $\mathbb{Z}_p^*$ bekannt

Naive Berechnung des diskreten Logarithmus

```
1  $\gamma := 1$ 
2 for  $i := 0$  to  $n - 1$  do
3   if  $\gamma = \beta$  then output( $i$ )
4    $\gamma := \alpha\gamma$ 
```

- Dieser Algorithmus läuft in Zeit $\mathcal{O}(n)$ und benötigt nur logarithmischen Speicherplatz (wobei wir annehmen, dass elementare Gruppenoperationen in konstanter Zeit ausführbar sind)
- Falls wir im Vorfeld eine Tabelle mit den Logarithmen aller möglichen Werte für β erstellen, können wir danach für jedes β den diskreten Logarithmus durch Table-Lookup in konstanter Zeit bestimmen

DLP-Berechnung mittels Precomputation

Precomputation: Speichere die Exponenten $e = 0, \dots, n - 1$ unter der Adresse α^e in einer Tabelle T

Computation: Gib bei Eingabe β den Wert $T[\beta]$ aus

Die Precomputation erfordert Zeit $\mathcal{O}(n)$ und Platz $\mathcal{O}(n \log n)$

Der Algorithmus von Shanks

- Der folgende Algorithmus von Shanks (auch **baby-step giant-step Alg.** genannt) berechnet ebenfalls im Vorfeld eine Tabelle von DLP-Werten
- Allerdings nur für die Potenzen α^{jm} , $j = 0, \dots, m-1$ und $m = \lceil \sqrt{n} \rceil$
- Dadurch erhöht sich zwar die Laufzeit zur Bestimmung des diskreten Logarithmus für β von $O(1)$ auf $O(\sqrt{n})$
- Dafür wird der Speicherplatz von $O(n \log n)$ auf $O(\sqrt{n} \log n)$ reduziert

Algorithmus Shanks($G, n, \alpha, \beta, m = \lceil \sqrt{n} \rceil$)

Precomputation: Sortiere die Paare (α^{im}, i) , $0 \leq i \leq m-1$, nach der ersten Komponente in eine Tabelle $T1$

Computation: Sortiere die Paare $(\beta\alpha^{-j}, j)$, $0 \leq j \leq m-1$, nach der ersten Komponente in eine Tabelle $T2$ und ermittle durch parallele sequentielle Suche Paare (γ, i) in $T1$ und (γ, j) in $T2$ mit der gleichen ersten Komponente

output $im + j$ // es gilt $\beta\alpha^{-j} = \gamma = \alpha^{im}$
