Algorithmus von Cippola

Algorithmus von Cippola

EINGABE: $p \in \mathbb{P}$, $a \bmod p$ mit $(\frac{a}{p}) = 1$

1. REPEAT
   1. Wähle $b \in \{1, \ldots, p-1\}$ zufällig. Setze $D := b^2 - a$.

   UNTIL $(\frac{D}{p}) = (-1)$.
2. Berechne $x := (b + \sqrt{D})^{\frac{p+1}{2}}$ in $\mathbb{F}_p[\sqrt{D}]$.

AUSGABE: $x \bmod p$ mit $x^2 \equiv a \bmod p$

Laufzeit: erwartete Laufzeit $\mathcal{O}(\log^3 p)$.

Bsp: : Wir berechnen die Quadratwurzel von $a = 2$ in $\mathbb{F}_7$.

- Für $b = 1$ gilt $(\frac{D}{p}) = (\frac{-1}{7}) = (-1)$. Es folgt

$$(b + \sqrt{D})^{\frac{p+1}{2}} = (1 + \sqrt{-1})^4 = (2\sqrt{-1})^2 = -4 \equiv 3 \bmod 7.$$

- Wir prüfen $3^2 = 9 \equiv 2 \bmod 7$.

Williams $p+1$ Methode

Idee von Williams $(p+1)$-Methode:

- Sei $n = pr$ mit $1 < p < n$, p prim, $p \nmid r$.
- Sei $D \in \mathbb{N}$ mit $\mathrm{ggT}(D, n) = 1$. Falls $(\frac{D}{p}) = (-1)$, dann gilt für
$$G_p = \{\omega \in (\mathbb{F}_p[\sqrt{D}])^* \mid N(\omega) = 1\}, \text{ dass } |G_p| = p+1.$$
- Sei $p+1$ b-glatt, d.h. $p+1 = \prod_{p \in B} p^{e_B}$.
- Sei k ein Vielfaches von $\prod_{p \in B} p^{e_B}$. Dann gilt
$$\omega^k = x + y\sqrt{D} \equiv 1 \bmod p \text{ für alle } \omega \in (\mathbb{Z}/n\mathbb{Z})[\sqrt{D}]^* \text{ mit } N(\omega) = 1.$$
- Falls zusätzlich $x \not\equiv 1 \bmod r$ folgt $p \leq \mathrm{ggT}(x-1, n) < n$.

Williams $p+1$ Methode

Algorithmus Williams $p+1$-Methode

EINGABE: $n = pr$ zusammengesetzt, p prim, Schranke C mit $p \leq C$.

1. Wähle b geeignet. Sei $B = \{p_1, \ldots, p_s\}$.
2. Wähle $a \in_R \{1, \ldots, n-1\}$. Falls $\mathrm{ggT}(a, n) > 1$, Ausgabe des ggT.
3. Setze $D := a^2 - 1$ und $\omega := a + \sqrt{D}$.
4. Für $i = 1 \ldots s$
   1. Wähle e_i maximal mit $p_i^{e_i} < C$. Berechne $\omega := \omega^{p_i^{e_i}}$ in $(\mathbb{Z}/n\mathbb{Z})[\sqrt{D}]$.
5. Sei $\omega = x + y\sqrt{D}$. Falls $\mathrm{ggT}(x-1, N) \notin \{1, N\}$, Ausgabe des ggT.

Korrektheit: In Schritt 3 wählen wir ein $\omega \in (\mathbb{Z}/n\mathbb{Z})[\sqrt{D}]^*$ mit

$$N(\omega) = a^2 - D = a^2 - (a^2 - 1) = 1.$$

- Mit Ws $\approx \frac{1}{2}$ gilt $(\frac{D}{p}) = (-1)$. Falls $(\frac{D}{p}) = 1$, ist $(\mathbb{Z}/p\mathbb{Z})[\sqrt{D}]^* = U_p$.
- In diesem Fall ist Williams Methode genau die $(p-1)$-Methode.
- Die sonstige Korrektheit folgt analog zur $(p-1)$-Methode.

Laufzeit: $\mathcal{O}(s \log^3 n)$ analog zur $(p-1)$-Methode.

Elliptische Kurven Faktorisierung

Idee der Elliptischen Kurven Faktorisierung (Lenstra 1993):

- Rechne auf einer elliptischen Kurve mit den Punkten
  $E(n) := \{(x,y) \in (\mathbb{Z}/n\mathbb{Z})^2 \mid y^2 = x^3 + ax + b \text{ mit } a,b \in \mathbb{Z}/n\mathbb{Z}\} \cup \mathcal{O}$.
- Für primes n besitzen die Punkte $E(n)$ eine Gruppenstruktur.
- Für $n = pr$ gilt $E(n) \cong E(p) \times E(r)$.
- Für zufällige $a,b \in \mathbb{Z}/n\mathbb{Z}$ ist $|E(p)|$ fast uniform verteilt in
  $$[p+1-2\sqrt{p}, p+1+2\sqrt{p}].$$
- Wir wählen solange a,b, bis $|E(p)|$ in kleine Primfaktoren zerfällt.
- D.h. im Gegensatz zu Pollards und Williams Methode können wir die Glattheit der Gruppenordnung über die Wahl von a,b steuern.
- Die Laufzeit der Elliptischen Kurven Faktorisierung ist
  $$L_p[\tfrac{1}{2}, \sqrt{2}] = e^{\sqrt{2 \ln p \ln \ln p}}.$$

Faktorisieren auf Quantenrechnern

Idee von Shors Faktorisierungsalgorithmus (1994):

- Wir wählen ein zufälliges $a \in U_n$ und berechnen $\operatorname{ord}(a)$.
- Falls $\operatorname{ord}(a)$ ungerade, so wählen wir ein neues a.
- Falls $\operatorname{ord}(a)$ gerade, gilt $a^{\operatorname{ord}(a)} \equiv 1 \bmod n$ und $a^{\frac{\operatorname{ord}(a)}{2}} \not\equiv 1 \bmod n$.
- Sei zusätzlich $a^{\frac{\operatorname{ord}(a)}{2}} \not\equiv -1 \bmod n$, dies geschieht mit $\mathrm{Ws} \geq \frac{1}{2}$.
- Dann liefert $\operatorname{ggT}(a^{\frac{\operatorname{ord}(a)}{2}} \pm 1, n)$ nicht-triviale Teiler von n.
- Auf Quantenrechnern kann sehr effizient die diskrete Fouriertransformation (DFT) ausgerechnet werden.
- Die DFT eignet sich zur Periodenbestimmung von Funktionen.
- Als Funktion wählen wir die Exponentierfunktion

$$\exp : \mathbb{Z} \to U_n \text{ mit } i \mapsto a^i.$$

- Wegen $\exp(i + \operatorname{ord}(a)\mathbb{Z}) = \exp(i)$ besitzt $\exp(\cdot)$ Periode $\operatorname{ord}(a)$.
- Laufzeit von Shors Algorithmus auf Quantenrechnern: $\mathcal{O}(\log^3 n)$.

Liften von Lösungen quadratischer Gleichungen

Motivation:

- Quadratisches Sieb: Wir benötigen Lösungen von $X^2 \equiv n \bmod p^k$.
- Für $k = 1$ berechne Lösungen mittels Tonelli-Shanks/Cippola.
- Liefern die Lösungen für $k = 1$ auch die Lösungen für $k > 1$?

Satz Liften von Lösungen quadratischer Gleichungen

Sei $p \in \mathbb{P} \setminus \{2\}$, $(\frac{a}{p}) = 1$ und $k \in \mathbb{N}$. Sei x_k Lösung für $x_k^2 \equiv a \bmod p^k$, d.h. $x_k^2 - a = c_k' p^k$. Dann wird $x_{k+1}^2 \equiv a \bmod p^{k+1}$ gelöst von

$$x_{k+1} := x_k + c_k p^k \text{ mit } c_k \equiv -\frac{c_k'}{2x_k} \bmod p.$$

Beweis:

- Falls $x_{k+1}^2 \equiv a \bmod p^{k+1}$, gilt $x_{k+1}^2 \equiv a \bmod p^\ell$ für alle $\ell \leq k+1$.
- Dies liefert den Ansatz $x_{k+1} \equiv x_k \bmod p^k$ bzw. $x_{k+1} = x_k + c_k p^k$.
- Wir suchen nun c_k.
- Da x_{k+1} modulo p^{k+1} definiert ist, bestimmen wir c_k modulo p.

Liften von Lösungen quadratischer Gleichungen

Beweis: (Fortsetzung)

- Mit Hilfe des Ansatzes $x_{k+1} = x_k + c_k p^k$ erhalten wir

$$\begin{aligned} 0 \equiv x_{k+1}^2 - a &= x_k^2 + 2x_k c_k p^k + (c_k p^k)^2 - a \\ &\equiv x_k^2 - a + 2x_k c_k p^k \bmod p^{k+1}. \end{aligned}$$

- Wegen $x_k^2 - a = c_k' p^k$ folgt

$$0 \equiv x_k^2 - a + 2x_k c_k p^k = (c_k' + 2x_k c_k) p^k \bmod p^{k+1}.$$

- Teilen durch p^k und Auflösen nach c_k liefert $c_k \equiv -\frac{c_k'}{2x_k} \bmod p$.

Anmerkung:

- Wir definieren $c_0 := x_1$. Dann gilt

$$\begin{aligned} x_k &= c_{k-1} p^{k-1} + x_{k-1} = c_{k-1} p^{k-1} + c_{k-2} p^{k-2} + x_{k-2} \\ &= c_{k-1} p^{k-1} + c_{k-2} p^{k-2} + \ldots + c_1 p^1 + x_1 = \sum_{i=0}^{k-1} c_i p^i. \end{aligned}$$

- D.h. x_k lässt sich mittels der $c_{k-1} \ldots c_0$ zur Basis p darstellen.

Liften von Lösungen quadratischer Gleichungen

Bsp: : Wir berechnen die Lösungen von $x_k^2 \equiv 2 \bmod 7^k$ für $k \leq 5$.

- Die Lösung $x_1 \equiv 3 \bmod 7$ finden wir mittels Cippola-Algorithmus.
- Wir wenden danach unsere Formel zum Liften an.

k	x_k	7^k	c'_k	c_k
1	3	7	1	$-\frac{1}{6} = 1$
2	10	49	2	$-\frac{1}{3} = 2$
3	108	343	34	$-\frac{6}{2\cdot 3} = 6$
4	2166	2401	23	$-\frac{2}{2\cdot 3} = 2$
5	4567	–	–	–

- Wir erhalten $x_5 = 2 \cdot 7^4 + 6 \cdot 7^3 + 2 \cdot 7^2 + 1 \cdot 7 + 3$.
- Wir würden gerne $x_\infty = \lim_{k\to\infty} x_k = \sum_{i=0}^{\infty} c_i 7^i$ berechnen.
- Damit hätten wir eine Lösung für alle Gleichungen $X^2 \equiv 2 \bmod 7^k$.
- Im Allgemeinen wird ein solcher Grenzwert aber nicht existieren.