Faktorisierungsalgorithmen

Moritz Kerger

05.12.2023

1 Faktorisierung

Als Faktorisierung wird der Vorgang bezeichnet, eine Zahl in ihre Faktoren zu zerlegen. Die Primzahlfaktorisierung einer Zahl N ist immer eindeutig.

$$N = \prod_{p \in \mathbb{P}} p^i, \qquad i \in \mathbb{N}$$

In der Theorie kann mit der Faktorisierung einer sehr großen Zahl N die kryptographische Sicherheit des RSA-Verfahrens gebrochen werden. Dies liegt daran, dass mithilfe des PublicKeys auf den PrivateKey zurück geschlossen werden kann. In der Praxis sind die Algorithmen für das Faktorisieren von modernen RSA-Schlüsseln zu langsam. Nach aktuellem Wissensstand ist das Faktorisierungsproblem nur in **exponentieller** Laufzeit zu lösen. Die asymptotische Laufzeit beträgt also für einen Schlüssel der Länge b

$$O(2^{f(b)})$$

Das Probedivision Verfahren beispielsweise prüft bis $\sqrt{N}$ und hat daher eine Laufzeit von $O(2^{\frac{b}{2}})$. Sind die Primzahlen bei der Generierung des RSA-Schlüssels gut gewählt, ist es mit aktuellen Faktorisierungsalgorithmen nicht möglich, auf den PrivateKey zu schließen. Ein paar Beispiele für große Zahlen, die im Rahmen der RSA-Factoring Challenge faktorisiert wurden sind:

- RSA-330 - 1991 - Mehrere Tage Rechenaufwand
- RSA-640 - 2005 - 5 Monate auf 80 2.2 GHz AMD Opteron CPUs
- RSA-829 - 2020 - 2700 CPU-Jahre auf 2.1 GHz Intel Xeon Gold 6130 CPUs
- RSA-2048 - ?

2 Probedivision

Die Probedivision ist eine Brute-Force Methode, bei der wir die zu faktorisierende Zahl N sukzessive durch jede Zahl kleiner als $\sqrt{N}$ teilen. Da hier viele unnötige Divisionen anfallen, können wir das Verfahren auf mehrere Arten verbessern:

- Teile nur durch jede zweite Zahl ab 3.
- Teile nur durch Zahlen der Form (6n+1) oder (6n-1) ab 3.
- Teile nur durch Primzahlen.

Prüfen wir nur Primfaktoren, so erreichen wir eine Laufzeit von

$$O\left(2^{\frac{n}{2}} \left(\frac{n}{2} \ln 2\right)^{-1}\right)$$

Das Generieren der Primzahltabelle lohnt sich dann, wenn der Algorithmus häufig angewendet wird. Grundsätzlich eignet sich die Probedivision dann, wenn ein Faktor der zusammengesetzten Zahl N besonders klein ist.

3 Faktorisierung nach Lehmann

Die Faktorisierungsmethode von Lehmann nutzt zuerst die Probedivision bis $\sqrt[3]{N}$. So kann geprüft werden, ob es sich um zwei oder mehrere Faktoren handelt. Existieren Faktoren zwischen 0 und $\sqrt[3]{N}$, so handelt es sich um eine Zahl, die aus 3 oder mehr Faktoren besteht. Gibt es Faktoren zwischen $\sqrt[3]{N}$ und $\sqrt{N}$, so muss es sich um zwei Primfaktoren handeln. Wenn in diesem Intervall auch keine Faktoren existieren, dann ist N selbst eine Primzahl. Die asymptotische Laufzeit der Lehmann Faktorisierung ist [1]

$$O\left(2^{\frac{b}{3}}\right)$$

4 Fermat Faktorisierung

Die Fermat Faktorisierung macht sich die Identität der 3. Binomischen Formel zunutze, um Primfaktorpaare in einer Umgebung von a zu finden.

$$N = a^2 - b^2 = (a+b)(a-b) = p \cdot q$$

Man kann also a als den Mittelpunkt zwischen den Primzahlen und b als den Radius betrachten. Initial wird a auf $\left\lceil \sqrt{N} \right\rceil + 1$, also eine Zahl über die Wurzel, bzw. die "Mitte" gesetzt. Durch die Wahl von $b = \sqrt{a^2 - N}$ wissen wir nun, dass wenn b eine ganze Zahl ist, dies der Radius um a sein muss. Ist b keine ganze Zahl, so vergrößern wir a um 1. Wie sich an dieser Vorgehensweise schon erkennen lässt, liefert diese Methode besonders für kleine b, also a die nah an der Wurzel liegen, schnell ein Ergebnis [2]. Wurden die Primzahlen für den RSA-Schlüssel zum Beispiel so gewählt, dass die ersten 500 bits der 1024 bit langen Faktoren gleich sind, so findet dieser Algorithmus bereits nach wenigen Zyklen eine Faktorisierung.

5 Pollard Rho Faktorisierung

Die Pollard Rho Faktorisierung erhält ihren Namen durch die Form des Graphen, wenn die Zahlen in einem Ablaufdiagramm dargestellt werden. Weil die Funktion $g^i(x_0)$ mit $g(x) = x - c \mod N$ ab einem bestimmten Punkt zyklisch wird, bildet sich ein ρ.

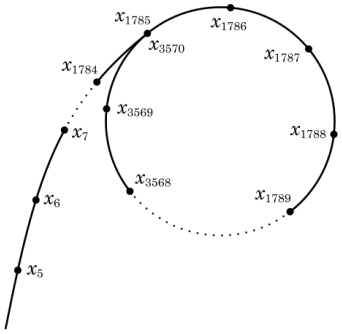

Figure 1: Pollard Rho Zyklus [3]

Diese Art der Faktorisierung eignet sich besonders für zusammengesetzte Zahlen, bei denen der erste Faktor deutlich kleiner ist, als der Zweite. Die Laufzeit ist proportional zu $\sqrt{p}$, wobei p der kleinere Faktor ist. Ein gutes Beispiel ist die achte Fermat-Zahl

$$F_8 = 1238926361552897 \cdot 93461639715357977769163558199606896584051237541638188580280321$$

References

[1] B. R. S. Lehman. Factoring large integers. *Mathematics of Computation*, 28:637–646, 1974.

[2] C. Pomerance and P. Erdös. A tale of two sieves. 1998.

[3] 忍者猫. Pollard rho cycle, 2021. File: `Pollard rho cycle.svg`.

https://github.com/mokeg67/Proseminar