4 Faktoriseringsalgoritmer

Vi vil i dette kapitel se på en af de anvendelser, som elliptiske kurver har, nemlig faktorisering af heltal. Vi ved fra aritmetikkens fundamentalsætning, at ethvert positivt heltal større end 1 enten er et primtal eller kan skrives som et entydigt produkt af primtal. Vi ønsker da for et heltal n, at bestemme sådan en primtalsfaktor. Vi vil i dette kapitel introducere to forskellige algoritmer, som kan benyttes til faktorisering. Først vil vi se på Pollards $p-1$ algoritme, som dog ikke benytter sig af elliptiske kurver, men som var inspirationen til den anden algoritme vi vil se på, nemlig Lenstras algoritme der benytter elliptiske kurver.

Motivationen for disse hurtigere metoder er, at en naiv tilgang til faktoriseringsproblemet er meget langsom. Antag at n er et sammensat tal, som vi ønsker at faktorisere. Hvis n faktoriseres som $n = n_1 n_2$ er det klart, at $\min\{n_1, n_2\} \leq \sqrt{n}$. Vi kan da finde en faktor ved at undersøge om først $2 \mid n$, dernæst om $3 \mid n$ osv. Vi vil da finde en faktor senest når vi kommer til $\sqrt{n}$. Dette bliver hurtigt uoverkommeligt når n er stort.

4.1 Pollards $p-1$ metode

Vi skal i dette afsnit beskrive Pollards $p-1$ algoritme.

Før vi beskriver algoritmen skal vi først bruge følgende definition.

Definition 7 (B-potensglat)**.** Lad $B \in \mathbb{Z}^+$. Hvis $n \in \mathbb{Z}^+$ har primtalsfaktoriseringen $n = \prod {p_i}^{e_i}$, da siges n at være B-potensglat hvis ${p_i}^{e_i} \leq B$ for alle i.

Eksempel 4. Da $50 = 2 \cdot 5^2$ følger det, at 50 er 25-potensglat. Bemærk, at den netop ikke er 5-potensglat.

Med disse detaljer på plads er vi nu klar til, at beskrive Pollards $p-1$ algoritme. Antag at det sammensatte tal n, som vi ønsker at faktorisere, har en primfaktor p sådan, at $p-1$ har mange små primtalsfaktorer. Fra

Fermats lille sætning ved vi, at

$$a^{p-1} \equiv 1 \pmod{p},$$

hvilket betyder at $p \mid \gcd(a^{p-1} - 1, n)$. Men da vi ikke kender p (det er jo den faktor vi leder efter)

Algoritme 2 (Pollards $p-1$ algoritme)**.** Lad $n \geq 2$ være et sammensat tal, som er tallet vi ønsker at finde en faktor for.

1. Vælg et tal $k \in \mathbb{Z}^+$ sådan, at k er et produkt af mange små primtal opløftet i små potenser. F.eks. kan k vælges som

   $$k = \mathrm{LCM}[1, 2, \ldots, K],$$

   for $K \in \mathbb{Z}^+$.

2. Vælg et heltal a sådan, at $1 < a < n$.

3. Udregn $\gcd(a, n)$. Hvis $\gcd(a, n) > 1$ har vi fundet en ikke-triviel faktor for n og vi er færdige. Ellers fortsæt til næste trin.

4. Udregn $D = \gcd(a^k - 1, n)$. Hvis $1 < D < n$ er D en ikke-triviel faktor for n og vi er færdige. Hvis $D = 1$ gå da tilbage til trin 1 og vælg et større k. Hvis $D = n$ gå da til trin 2 og vælg et nyt a.

Følgende er et eksempel på anvendelsen af Pollards algoritme, hvor det går godt, altså hvor $p - 1$ har små primfaktorer.

Eksempel 5. Vi vil forsøge at faktorisere

$$N = 30042491.$$

Vi ser at $2^{N-1} = 2^{30042490} \equiv 25171326 \pmod{30042491}$, så N er ikke et primtal. Vi vælger som beskrevet i algoritmen

$$a = 2 \quad \text{og} \quad k = \mathrm{LCM}(1, 2, 3, 4, 5, 6, 7) = 420.$$

Da $420 = 2^2 + 2^5 + 2^7 + 2^8$ skal vi udregne 2^{2^i} for $0 \leq i \leq 8$. Vi springer de første par udregninger over.

i	$2^{2^i} \pmod{N}$
5	28933574
6	27713768
7	10802810
8	16714289

Denne tabel gør det forholdsvist let for os, at bestemme

$$\begin{aligned} 2^{420} &= 2^{2^2+2^5+2^7+2^8} \\ &\equiv 16 \cdot 28933574 \cdot 10802810 \cdot 16714289 \pmod{30042491} \\ &\equiv 27976515 \pmod{30042491} \end{aligned}$$

Ved anvendelse af den euklidiske algoritme finder vi dernæst, at

$$\gcd(2^{420} - 1, N) = \gcd(27976514, 30042491) = 1.$$

Her fejler testen altså og vi er nået frem til, at N ikke har nogle primtalsfaktorer p sådan, at $p - 1$ deler 420.