3 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.

3.1 Pollards $p-1$ algoritme

Lad n være et sammensat tal og lad p være en primfaktor for n. Vi ved fra Fermats lille sætning, at $a^{p-1} \equiv 1 \pmod{p}$ når $\gcd(a,p) = 1$. Hvis vi da kendte $p-1$ kunne vi bestemme p (udover den åbenlyse måde) ved

$$\gcd(a^{p-1} - 1, n) = p.$$

(måske et multiplum af p?), da hvis $x \equiv 1 \pmod{l}$, hvor l er en faktor i n, er $\gcd(x-1, n)$ divisibel med denne faktor l.

Vi kender dog ikke $p-1$ og vi kan derfor ikke foretage denne udregning. Det viser sig dog, at vi kan nøjes med et multiplum af $p-1$, da

$$a^{t(p-1)} - 1 = (a^{p-1})^k - 1 \equiv 1^k - 1 \equiv 0 \pmod{p}.$$

Idéen er da, at vi vælger et heltal

$$k = 2^{e_2} \cdot 3^{e_3} \cdot 5^{e_5} \cdots r^{e_r},$$

hvor $2, 3, \ldots, r$ er primtal og $e_1, e_2, \ldots, e_r$ er små positive heltal. Vi udregner da $\gcd(a^k - 1, n)$. Hvis vi er i det heldige tilfælde, hvor n har en faktor sådan, at $p - 1 \mid k$, da vil $p \mid a^k - 1$ og vi har så, at

$$\gcd(a^k - 1, n) \geq p > 1.$$

Hvis $\gcd(a^k - 1, n) \neq n$ har vi altså fundet en ikke-triviel faktor for n og vi kan dele n i to faktorer og gentage de ovenstående trin. Hvis vi derimod har, at $\gcd(a^k - 1, n) = n$ vælger vi et andet a og forsøger igen, og hvis $\gcd(a^k - 1, n) = 1$ vælger vi et større k.

Dette er tankegangen i Pollards $p - 1$ algoritme og vi opsummerer det i algoritmen:

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

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

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

   for et $K \in \mathbb{Z}^+$ og hvor LCM er det mindste fælles multiplum.

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 2. 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

$$a = 2 \quad \text{og} \quad k = \text{LCM}[1, 2, \ldots, 7] = 420.$$

Da $420 = 2^2 + 2^5 + 2^7 + 2^8$ skal vi udregne 2^{2^i} for $0 \leq i \leq 8$. Dette resulterer i følgende tabel:

i	$2^{2^i} \pmod n$		
1	4	5	28933574
2	16	6	27713768
3	256	7	10802810
4	65536	8	16714289
5	28933574		

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

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

Her fejler testen altså og vi er nået frem til, at N ikke har nogle primtalsfaktorer p sådan, at $p-1$ deler 420. Algoritmen foreskriver da, at vi skal vælge et nyt k. Vi lader

$$k = \mathrm{LCM}[1, 2, \ldots, 11] = 27720.$$

Da $27720 = 2^{14} + 2^{13} + 2^{11} + 2^{10} + 2^6 + 2^3$ skal vi udvide tabellen til at indeholde værdierne for 2^{2^i} for $0 \leq i \leq 14$:

i	$2^{2^i} \pmod n$		
9	19694714	12	26818902
10	3779241	13	8658967
11	11677316	14	3783587

Vi fortsætter på samme måde, som vi gjorde før og bestemmer

$$\begin{aligned} 2^{27720} &= 2^{2^3+2^{2^6}+2^{2^{10}}+2^{2^{11}}+2^{2^{13}}+2^{2^{14}}} \\ &= 256 \cdot 27713768 \cdot 3779241 \cdot 11677316 \cdot 8658967 \cdot 3783587 \\ &= 16458222 \pmod{30042491}. \end{aligned}$$

Vi finder dernæst, at

$$\gcd(2^{27720} - 1 \pmod n, n) = \gcd(16458221, 30042491) = 9241,$$

hvilket betyder at vi har fundet en ikke-triviel faktor for n. Mere præcist har vi fundet faktoriseringen

$$30042491 = 3251 \cdot 9241.$$