4 Faktoriseringsalgoritmer

I dette kapitel ønsker vi at se på faktoriseringsalgoritmer. Det viser sig nemlig, at en af de anvendelser som elliptiske kurver besidder, er indenfor faktoriseringen af heltal. Faktoriseringsproblemet, altså hvordan man bestemmer en faktor for et tal n er yderst relevant, da alle heltal kan faktoriseres:

Sætning 5 (Aritmetikkens fundamentalsætning)**.** *Et heltal $n > 1$ kan faktoriseres entydigt som et produkt af primtal, så hvis*

$$p_1 \cdot p_2 \cdot \ldots \cdot p_k = q_1 \cdot q_2 \cdot \ldots \cdot q,$$

hvor p_i og q_j er primtal for $1 \leq i \leq k$ og $1 \leq j \leq l$ er $k = l$ og $p_i = q_i$ for alle $i = 1, 2, \ldots, k$ (efter eventuelle ombytninger). Desuden er faktorerne $p_1^{n_1}, p_2^{n_2}, \ldots, p_k^{n_k}$ entydigt bestemte.

For et bevis af sætningen se f.eks. [1]. Det vigtige at bemærke er, at beviset ikke er konstruktivt og dermed ikke giver os en måde, hvorpå vi kan finde disse faktorer. Men hvordan kan vi så finde disse faktorer, som vi nu ved findes? Hvis vi har et sammensat tal n, som vi ønsker at faktorisere kunne vi angribe problemet med en naiv tilgang. Vi antager for nemhedens skyld at $n = pq$, hvilket gør det klart at $\min\{p, q\} \leq \sqrt{n}$. Vi kan altså finde en faktor ved at undersøge om først $2 \mid n$, dernæst om $3 \mid n$ osv. indtil at vi finder en faktor, hvilket vil ske senest når vi når til $\sqrt{n}$. Denne løsning er fin for tilstrækkeligt små tal, men det bliver hurtigt uoverkommeligt for store tal (eksempler på hvor lang tid det tager?).

Sikkerheden i moderne kryptosystemer hviler på dette faktum, at det tager lang tid at faktorisere et heltal. Derfor er det interessant at undersøge om man gøre det hurtigere end den med den naive tilgang. Vi skal se på to af sådanne algoritmer, nemlig Pollards $p-1$ algoritme og Lenstras algoritme, som benytter elliptiske kurver til at finde en faktor.

4.1 Pollards $p - 1$ algoritme

Vi starter med at se på Pollards $p - 1$ algoritme, da Lenstras algoritme er stærkt inspireret af denne og delvist kan ses som en analog til den, hvilket gør det naturligt at betragte den først. Pollards $p - 1$ algoritme blev først præsenteret i [3] i 1970'erne af J. M. Pollard. Algoritmen hviler på Fermats lille sætning:

Sætning 6 (Fermats lille sætning)**.** *Lad p være et primtal som ikke går op i a. Da gælder der, at*

$$a^{p-1} = 1 \pmod{p}.$$

Et bevis findes i appendikset.

Vi kan da se på, hvordan Pollards $p-1$ algoritme virker. 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

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

   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 4. 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

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

Da $420 = 2^2 + 2^5 + 2^7 + 2^8$ skal vi udregne 2^{2^i} for $0 \le i \le 8$. 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 $0 \le i \le 14$:

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

$$30042491 = 3251 \cdot 9241.$$

4.2 Lenstras elliptiske kurve algoritme

I [2] Lenstra præsenterede i [LENSTRA] en algoritme, som er stærkt inspireret af Pollards $p-1$ algoritme. Fordelen ved Lenstras algoritme er, at den ikke er låst fast til én enkelt gruppe af orden $p-1$. Hvis algoritmen ikke finder en faktor er vi i stand til at skifte gruppen vi arbejder med i håbet om, at det så vil lykkedes med denne.

Problemet med Pollards $p-1$ algoritme opstår, hvis tallet vi ønsker at faktorisere har primfaktorer som ikke er B-glat for store B. Vi ser på tallet

$$n = 1688955439703788849,$$

som er konstrueret ved at gange to tilfældige (og meget hurtigt glemte) primtal p sammen, som begge har den egenskab at $p-1$ ikke er B-glat for $B = 10^8$. Pollards $p-1$ algoritme ville være ineffektiv for et sådan tal, men Lenstras algoritme viser sig at have en løsning på sådan et tal.

Algoritme 2 (Lenstras algoritme)**.** Lad $n \geq 2$ være et sammensat tal, som vi ønsker at finde en faktor for.

1. Vælg $x, y, A \in [1, n]$. Lad da $B = y^2 - x^3 - Ax \pmod{n}$ for da har vi den elliptiske kurve

$$E : y^2 = x^3 + Ax + B,$$

   hvorpå punktet $P = (x, y)$ er placeret.

2. Tjek at $D = \gcd(4A^3 + 27B^2, n) = 1$. Hvis $D = n$ går vi tilbage til (1) og vælger et nyt b. Hvis $1 < D < n$ har vi fundet en faktor af n og vi er færdige.

3. Vælg et positivt heltal k som et produkt af mange små primtal, lad eksempelvis

$$k = \mathrm{LCM}[1, 2, 3, ..., K],$$

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

4. Forsøg at bestemme $kP = P + P + \ldots + P$. Hvis udregningen kan lade sig gøre går vi tilbage til (1) og vælger en ny kurve, eller går til (3) og vælger et større k.

Det der kan gå galt er, at ikke alle elementer i $\mathbb{Z}/n\mathbb{Z}$ har en invers, da n ikke er et primtal.

Med algoritmen på plads er vi nu i stand til at beregne et eksempel, hvor variablerne bliver valgt sådan at det går godt i første omgang:

Eksempel 5. Lad nu

$$n = 753161713$$

være det tal, som vi ønsker at faktorisere. Da $2^{n-1} = 437782651 \pmod{n}$ er n ikke et primtal. Vi vælger da $x = 0$, $y = 1$ og $A = 164$. Vi har dermed, at $B = 1^2 - 0^3 - 164 \cdot 0 = 1$ og den elliptiske kurve vi vil arbejde over bliver

$$E : y^2 = x^3 + 164x + 1,$$

hvorpå punktet $P = (0, 1)$ er placeret. Vi ser, at

$$\begin{aligned} D &= \gcd(4 \cdot 164^3 + 27 \pmod{753161713}, 753161713) \\ &= \gcd(17643803, 753161713) = 1, \end{aligned}$$

så vi fortsætter derfor med algoritmen. Vi lader

$$k = \mathrm{LCM}[1, 2, \ldots, 10] = 2520.$$

Da $2520 = 2^{11} + 2^8 + 2^7 + 2^6 + 2^4 + 2^3$ skal vi beregne $2^iP \pmod{753161713}$ for $0 \leq i \leq 11$. Dette gøres med additionsformlen og vi opsummerer vores resultater i tabellen nedenfor:

i	$2^iP \pmod{753161713}$	i	$2^iP \pmod{753161713}$
0	$(0, 1)$	6	$(743238772, 703386057)$
1	$(6724, 752610344)$	7	$(309161840, 219780637)$
2	$(293427237, 450490340)$	8	$(116974611, 722899047)$
3	$(468952095, 385687511)$	9	$(329743899, 182819134)$
4	$(288125200, 446796094)$	10	$(163952469, 456288424)$
5	$(106753239, 115973502)$	11	$(15710788, 301760412)$

Vi kan nu addere disse punkter igen vha. additionsformlerne, hvor vi stadigvæk regner modulo n:

$$(2^3 + 2^4)P = (606730980, 447512524).$$

Algoritmen giver os en faktor netop når additionen bryder sammen, hvilket kan ske da $\mathbb{Z}/n\mathbb{Z}$ ikke er et legeme. Dette problem viser sig i dette eksempel allerede ved den næste addition, hvor vi forsøger at udregne

$$\begin{aligned} (2^3 + 2^4 + 2^6)P = &(743238772, 703386057) \\ &+(606730980, 447512524) \pmod{n}. \end{aligned}$$

For at denne addition skal kunne lade sig gøre, skal differensen af deres x-koordinater have en invers modulo n. Dette er kun tilfældet, hvis $\gcd(x_2 - x_1, n) = 1$ (se appendiks, sætning k). Men vi ser, at

$$\gcd(606730980 - 743238772, 753161713) = 19259,$$

så der findes altså ikke en invers, men vi har i stedet fundet en faktor i n. Dermed har vi faktoriseringen

$$753161713 = 19259 \cdot 39107.$$

Nu kan det virke til, at det var spild da vi lavede hele tabellen, men i beregningerne af $2^i P \pmod{753161713}$ ville vi også kunne have løbet ind i et element, som ikke havde en invers og som dermed kunne give os en faktor.

A Appendiks

Her samler vi nogle af de (hovedsagligt) mindre resultater, som benyttes igennem kapitlerne. De præsenteres her kort og henvises til i opgaven, når de er blevet anvendt.

Proposition 3. *Et element $a \in \mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$ har ikke en invers i $\mathbb{Z}_n$ hvis* $\gcd(a, n) > 1$.

Bevis. Antag for modstrid, at $d = \gcd(a, n) > 1$, men at der samtidigt eksisterer en invers c til a modulo n. Da $d = \gcd(a, n)$ findes et heltal e, som ikke er nul, sådan at $de = n$. Da $d > 1$ har vi også, at $|e| < |n|$ så e er ikke nul modulo n. Da d deler a har vi, at $n = de$ deler ae så $ae = 0 \pmod n$. Vi har altså, at

$$e = e \cdot 1 = eac = 0 \cdot c = 0 \pmod n,$$

hvilket er i modstrid med at e ikke kunne være 0 modulo n. Altså har a ikke en invers når $\gcd(a, n) > 1$. □

Vi giver nu beviset for sætning 6:

Bevis for Fermats lille sætning. Vi ser først på de $p - 1$ positive multipla af a

$$a, 2a, \ldots, (p-1)a. \tag{A.1}$$

Hvis $ra = sa \pmod p$ har vi, at $r = s \pmod p$, så elementerne listet i (A.1) er forskellige og ikke-nul. De må altså være kongruente til $1, 2, \ldots, p-1$ men ikke nødvendigvis i den opskrevne rækkefølge. Ganger vi elementerne sammen må de to kongruenser være de samme, altså er

$$a \cdot 2a \cdots (p-1)a = 1 \cdot 2 \cdots (p-1) \pmod p,$$

hvilket giver os, at

$$a^{p-1}(p-1)! = (p-1)! \pmod p \Rightarrow a^{p-1} = 1 \pmod p.$$

□

Bibliografi

[1] Johan P. Hansen. *Algebra og talteori.*

[2] H. W. Lenstra Jr. „Factoring Integers with Elliptic Curves". I: *The Annals of Mathematics, Second Series,* 126 (03 nov. 1987), s. 649–673.

[3] J. M. Pollard. „Theorems on factorization and primality testing". I: *Mathematical Proceedings of the Cambridge Philosophical Society* 76 (03 okt. 1974), s. 521–528.