Indhold

Resumé: Elliptiske kurver og gruppestrukturen på dem bliver præsenteret. Der gives bevis for gruppestrukturen ved hjælp af divisorer og Picard-gruppen. Derefter ses på hvordan kurver kan defineres over endelige legemer. Så præsenteres asymmetrisk kryptografi og det diskrete logaritmeproblem. Endelig diskuteres muligheder, fordele og problemer ved at anvende elliptiske kurvegrupper til dette formål.

1 Indledning

Elliptiske kurver, som denne rapport handler om, er sjove fordi de berører forskelligartede matematiske emner. I første omgang ligner det – og navnet lyder som – et geometrisk objekt. Det viser sig så at punkterne på kurven har en interessant indbyrdes struktur, således at emnet får mere at gøre med algebra og gruppeteori. Med gruppestrukturen kommer anvendelsesmulighed indenfor f.eks. talteori og, som vi skal se nogle eksempler på til sidst i rapporten, kryptologi.

I del 2 defineres først kurven som objekt, og den algebraiske struktur, gruppestrukturen, præsenteres. Dette er ikke specielt kompliceret, men det er derimod beviset for at gruppen opfører sig som den skal, nemlig som en abelsk gruppe. Dette handler resten af del 2om.

Jeg har valgt at holde mig til en 'simpel' version af definitionen på kurven. Det gør det nemmere at følge med i hvad der foregår, og derfor forhåbentlig nemmere at læse end f.eks. (Enge, 1999), hvori der løbende tages højde for diverse 'besværlige' tilfælde.

Del 3 præsenterer grundlæggende idéen bag asymmetrisk kryptografi. Derefter lidt om hvordan kryptering kan implementeres ved hjælp af elliptiske kurver, og om hvad fordelen kunne være ved at gøre det. For eksempel kunne det være interessant at finde ud af:

- Hvilke kurver giver en „sikker" kode?
- Hvad er det som gør koden sværere at knække end f.eks. en RSA-indkodning af tilsvarende tyngde (størrelsen af den offentlige nøgle kunne være en sammenligning, eller forholdet i størrelse mellem den indkodede og den 'rå' tekst.)

1.1 Baggrund

Som datalogistuderende med hang til matematik er det svært at forestille sig et mere 'behageligt' emne at skrive opgave om: En fin matematisk definition, som ender med at kunne bruges til noget så konkret som kryptering af data.

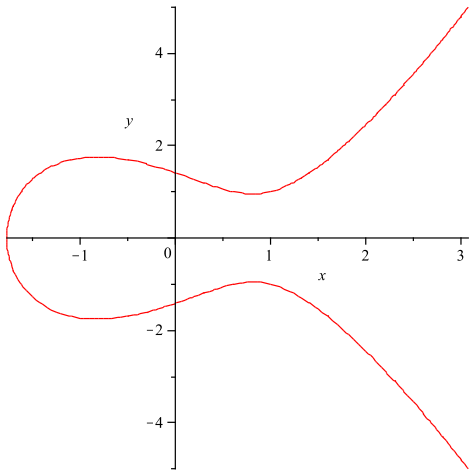
Mit svage punkt har været algebraen; det har været min første oplevelse med gruppeteori, og mange af de algebraiske sætninger der i (Enge, 1999) gør gennemgangen hurtigere, gjorde den langsommere for mig. En mere 'jordnær' gennemgang fandt jeg i (Charlap og Robbins, 1988), som er mit faste holdepunkt gennem hele første del af opgaven. Jeg har skrevet opgaven ud fra den måde jeg selv har forstået stoffet, og håber at den vil være læsbar for andre med en lignende baggrund.

2 Elliptisk kurve

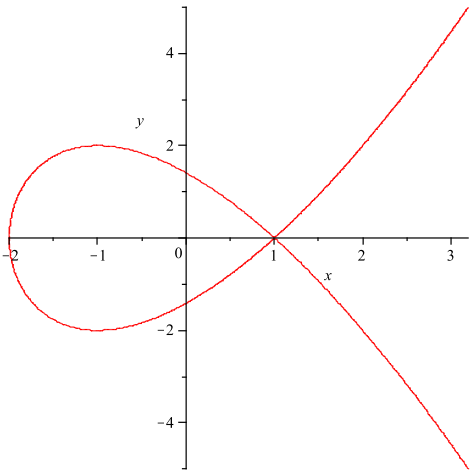
En elliptisk kurve E består af de punkter (x, y) som opfylder ligningen

$$y^2 = x^3 + Ax + B \tag{1}$$

Kurven kan afbildes som på figur 1. En sådan tegning giver på sin vis et falsk billede af kurven — selvom vi kunne gøre figuren uendelig stor i de retninger hvor kurven fortsætter ud af billedet, ville den kun viser den del som ligger i $\mathbb{R}^2$. Gruppestrukturen afhænger af at *alle polynomier har rødder*, så det er nødvendigt at udvide det talsystem som indeholder x, y og koefficienterne A og B, til at være et *algebraisk lukket legeme*. Det nemmeste er at sige at A og B er reelle, og variablene er komplekse tal. På den måde kan vi betragte figur 1 som et reelt billede af kurven, og må acceptere at en del af kurven, nemlig den del som ikke ligger i $\mathbb{R}^2$, er 'usynlig'.



(a) En ikke-singulær kurve med parametrene $A = -2$ og $B = 2$. Polynomiet $x^3 - 2x + 2$ har én reel rod, -2. Denne rod gør at (-2,0) ligger på kurven.



(b) Denne kurve er singulær: Parametrene $A = -3$ og $B = 2$ giver $\Delta = 0$. Kurven har ikke en veldefineret tangent i punktet (1,0)

Figur 1: Afbildning i $\mathbb{R}^2$ af nogle elliptiske kurver.

For at de beregninger vi skal udføre kommer til at fungere, skal kurven være ikke-singulær. Det betyder, som sædvanligt når man taler om en kurve, at kurven har en veldefineret tangent i ethvert punkt. I praksis betyder det også (og er faktisk ækvivalent med) at polynomiet på højre side af (1) har tre forskellige rødder (af hvilke to af dem gerne må være komplekse, bare de er forskellige). Disse benævnes med ω_1, ω_2 og ω_3 – eller bare ω hvis det er ligegyldigt hvilken. For at sikre sig at de er forskellige kan

man prøve at skrive polynomiet som et produkt. Antag at to af rødderne er ens, f.eks. $\omega_2 = \omega_3$. Så er

$$\begin{aligned} x^3 + Ax + B &= (x - \omega_2)^2(x - \omega_1) \\ &= x^3 - (\omega_1 + 2\omega_2)x^2 + (\omega_2^2 + 2\omega_1\omega_2)x - \omega_1\omega_2^2 \end{aligned}$$

Koefficienten til x^2 er jo nul, så $\omega_1 = -2\omega_2$. Ved at sammenligne de andre koefficienter fåes

$$\begin{aligned} A = \omega_2^2 + 2\omega_1\omega_2 = -3\omega_2^2 \qquad &\Longleftrightarrow \qquad A^3 = -27\omega_2^6 \\ B = -\omega_1\omega_2^2 = 2\omega_2^3 \qquad &\Longleftrightarrow \qquad B^2 = 4\omega_2^6 \end{aligned}$$

Til sammen giver de to ligninger længst til højre at

$$4A^3 = -27B^2$$

– denne ligning er opfyldt hvis kurven er singulær. Tallet $\Delta = 4A^3 + 27B^2$ kaldes kurvens diskriminant, og kan altså bruges til at checke parametrene A og B inden man arbejder med kurven (Charlap og Robbins, 1988).

Fra nu af bruges E om kurven givet ved ligning (1), med parametre A og B valgt således at $\Delta \neq 0$.

2.1 Gruppeoperationen på E

Vi skal nu definere en regneoperation som kan bruges på punkter på E. Det skal vise sig at der derved opstår er en *gruppestruktur*, der gør at vi kan bruge kurven E til en masse ting.

Definitionen af gruppestruktur er simpel nok. Følgende definition bruger additiv notation, da det er det vi vil bruge på E.

Definition 2.1 (Abelsk gruppe)**:** En mængde G med en tilhørende operator '+' er en (algebraisk) gruppe hvis den opfylder gruppeaksiomerne:

1. G er lukket: $a, b \in G \Rightarrow a + b \in G$
2. G har et nulelement: $\exists 0_G \in G : \forall a \in G : a + 0_G = 0_G + a = a$
3. Alle elementer kan inverteres: $\forall a \in G : \exists(-a) \in G : a + (-a) = 0_G$
4. Operatoren er associativ: $\forall a, b, c \in G : a + (b + c) = (a + b)c$

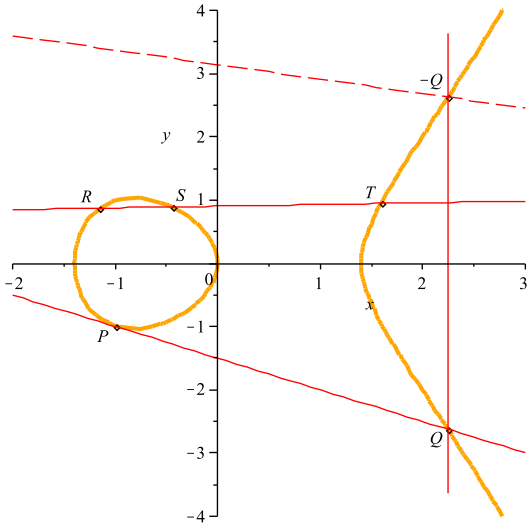
Hvis endnu et aksiom gælder, siges G at være en *abelsk* gruppe.

5. Alle elementer kommuterer $\forall a, b \in G : a + b = b + a$

Operatoren på E defineres i første omgang kan på en geometrisk måde; senere kan vi, ved at regne på punkternes koordinater, se at den samme definition kan bruges generelt (altså også virker på den 'usynlige' del af E). Vi betegner punkter på E med store bogstaver, f.eks. P, Q og R, og bruger notationen $P+Q=R$ når R er det punkt vi kommer hen til ved at 'lægge P og Q sammen' ved hjælp af den nye operator.

Idéen er at betragte skæringspunkter mellem kurven og en ret linje, og definere summen af de punkter som ligger på en ret linje til at være nul. Hvis additionen skal være en gruppeoperation skal 'nul' selvfølgelig også være et punkt på kurven. Til dette formål indføres et 'uendeligt punkt' $\mathcal{O}$, som fungerer som nulelement (additiv identitet) for additionen. Man kan forestille sig at alle rette linjer 'skærer' $\mathcal{O}$.

På figur 2 kan man se nogle sådanne linjer, og ved at kombinere linjernes ligninger med kurvens ligning kan vi finde formler som gør det muligt at beregne et nyt punkt på kurven – altså udføre additionen $P+Q$, når vi kender koordinaterne til P og Q på kurven.



Figur 2: Skæringspunkter mellem kurven med parametrene $A = -2$ og $B = 0$ og nogle rette linjer. $x^3 - 2x$ har tre rødder i $\mathbb{R}$, derfor skærer denne kurve x-aksen tre steder.

Betragt figur 2. Se først på den lodrette linje som går gennem punktet Q. Hvis man sætter Q's koordinater til (x_q, y_q), og sætter $x = x_q$ i ligning (1), får man en andengradsligning i y, så linjen kan kun skære kurven i ét punkt udover Q. Dette punkt, med koordinater $(x_q, -y_q)$, kalder vi $-Q$, så vi får $Q + (-Q) = \mathcal{O}$. Vi finder altså den additivt inverse til et punkt ved at gange y-koordinaten med -1.

Linjen gennem punkterne R, S og T giver at $R + S + T = \mathcal{O}$. Hvis vi kender $R = (x_r, y_r)$ og $S = (x_s, y_s)$, og kan beregne $T = (x_t, y_t)$, kan vi skrive $P + Q = -T$. Vi kan opskrive en ligning for linjen: Hældningen bliver $\lambda = \frac{y_s - y_r}{x_s - x_r}$, og linjens ligning bliver

$$y = \lambda(x - x_r) + y_r \tag{2}$$

Ved at sætte dette udtryk for y ind i kurvens ligning fås et tredjegrads polynomium i x, hvis rødder er skæringspunkternes x-koordinater:

$$f(x) = (\lambda(x - x_r) + y_r)^2 - x^3 - Ax - B \tag{3}$$

Hvis man ganger ud kan man se at koefficienten til x^2 bliver λ^2. Vi kender allerede to rødder i f, nemlig skæringspunkternes x-koordinater x_r og x_s, og vil finde den tredje, x_t. Et polynomium med samme rødder som f er altså

$$\begin{aligned} g(x) &= (x - x_r)(x - x_s)(x - x_t) \\ &= x^3 - (x_r + x_s + x_t)x^2 + (x_r x_s + x_s x_t + x_t x_r)x - x_r x_s x_t \end{aligned}$$

Hvis vi ganger g med -1 får vi et polynomium med samme grad, rødder og højeste koefficient (koefficienten til x^3 i f er jo -1) som f. Altså må $f = -g$, og derfor må koefficienterne til x^2 være ens, altså $\lambda^2 = x_r + x_s + x_t$. Det giver $x_t = \lambda^2 - x_r - x_s$, og vi kan derpå finde y_t ved at sætte x_t ind i linjens ligning (2).

Linjen mellem P og Q tangerer kurven i P. Vi kan finde linjens hældning ved at differentiere (1), og det bliver $\lambda = \frac{3x_p^2+A}{2y_p}$. Hvis vi erstatter R's koordinater med P's (x_p, y_p) i polynomiet f fra (3) og differentierer bliver det

$$f'(x) = 2\lambda(\lambda(x - x_p) + y_p) - 3x^2 - A$$

Med $x = x_p$ og $\lambda = \frac{3x_p^2+A}{2y_p}$ får vi

$$f'(x_p) = 2\left(\frac{3x_p^2 + A}{2y_p}\right) y_p - 3x_p^2 - A = 0$$

Vandret tangent i nulpunktet betyder at x_p er *dobbelt* rod i f, så der er altså ikke andre røringspunkter end P og Q. Dette kan opfattes som om at linjen 'skærer kurven to gange' i P, og derfor defineres $P + P + Q = \mathcal{O}$.

Når vi har λ kan vi ligesom før – uanset hvilken af metoderne vi har brugt til at beregne λ – bruge det til at finde den ukendte rod i f, og derefter finde den tilhørende y-koordinat ved at indsætte i linjens ligning, og kan nu beregne $P + P = -Q$ og $P + Q = -P$.

Definition 2.2 (Additionsformler på E)**:** Lad P_1, P_2 være punkter på E. Det inverse element til P_1 er givet ved

$$-P_1 = (x_1, -y_1)$$

Hvis $P_2 = -P_1$ er

$$P_1 + P_2 = P_1 - P_1 = \mathcal{O}$$

Ellers udføres additionen $P_1 + P_2$ ved at beregne P_3, hvis koordinater fremkommer ved følgende formler:

$$\lambda = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{hvis } x_2 \neq x_1 \\ \frac{3x_1^2+A}{2y_1} & \text{hvis } P_1 = P_2 \end{cases}$$

$$x_3 = \lambda^2 - x_1 - x_2$$

$$y_3 = \lambda(x_3 - x_1) + y_1$$

2.1.1 'sjove' punkter på E

Definitionen af den inverse af et punkt som spejlingen af punktet i x-aksen giver en lidt mere intuitiv forklaring på $\mathcal{O}$ som et 'uendeligt' punkt. På figur 2 forbinder en lodret linje Q med $-Q$. Det er ikke svært at se at hvis et punkt Q' ligger lidt til højre for $-Q$, vil linjen gennem Q og Q' skære kurven i et tredje punkt som ligger langt oppe til højre, og hvis man flytter Q' nærmere til $-Q$, vil $Q + Q'$ nærme sig $\mathcal{O}$, imens punktets koordinater nærmer sig ∞. Vi kalder af og til et punkt på E for et *endeligt* punkt for at understrege at det ikke er $\mathcal{O}$ det drejer sig om.

Tre punkter på kurven har y-koordinat nul, nemlig $(0, \omega_i)$ for $i = 1, 2, 3$. Disse punkter kaldes *andenordens* punkter, da de er de eneste endelige punkter som opfylder $P + P = \mathcal{O}$.

Der er også punkter som har orden 3; det er punkter for hvilke $P + P + P = \mathcal{O}$. Det er de punkter hvor kurven har en vendetangent, så polynomiet f (fra (3) i forrige afsnit) har en *tredobbelt* rod i punktet. Brown (2001) nævner nogle skægge egenskaber ved disse punkter, og viser at der er ni af dem hvis man tæller $\mathcal{O}$ med. Disse 'vendepunkter' kræver dog ikke særbehandling som vi får at se at det er tilfældet for andenordenspunkterne.

2.2 Gruppestruktur

Nu har vi en operator så vi kan lægge to punkter på E sammen, men vi har endnu ikke bevist at aksiomerne i definition 2.1 er opfyldt. Ved et nærmere øjekast på definition 2.2 kan vi dog nemt se at det er tilfældet for de fleste af dem:

- Additionsformlerne er konstrueret så resultatet af additionen ligger på kurven.
- Punkter kommuterer, kan man se på formlerne.
- $\mathcal{O}$ er identitetselement, og det inverse element til (x, y) er $(x, -y)$

Det eneste der mangler for at gøre $(E, +)$ til en abelsk gruppe er altså at bevise at addition er associativ. Beviset for associativitet kan gennemføres ved algebraisk beregning (Stinson (2005) betegner dette bevis som „rather messy“), eller ved hjælp af algebraisk geometri. Jeg vil foretrække det sidste.

Idéen er at definere en ny type objekter, kaldet *divisorer* på E. De defineres ved hjælp af de rationelle funktioner på E, og de udgør en gruppe. Derpå kan man vise at en undergruppe af divisorerne, den såkaldte Picard-gruppe, er isomorf med mængden af punkter på E under addition. Dette viser at $(E, +)$ er en gruppe, herunder at gruppeoperationen er associativ.

En rationel funktion er en kvotient mellem polynomier, så for at kunne arbejde med rationelle funktioner skal man først definere egenskaber ved polynomier på E.

2.2.1 Polynomier på E

Polynomier på E kan nemt defineres ved hjælp af begreberne om ringe og legemer, og man ville derved få en del 'gratis' viden om dem via sætninger fra abstrakt algebra. Det vælger jeg at lade være med – i stedet vil jeg bruge følgende definition, som minder om den fra Charlap og Robbins (1988):

Definition 2.3: Et polynomium på E er et polynomium i x og y, med den yderligere information at (x, y) ligger på E, altså at ligning (1): $y^2 = x^3 + Ax + B$ altid gælder.

En konsekvens af at (x, y) ligger på kurven er at alle potenser af y højere end 1, kan omskrives ved hjælp af ligning (1). Derved kan alle polynomier skrives på standardformen $u(x) + yv(x)$, hvor u og v er polynomier i x alene.

Nu skal graden af polynomium på E defineres. Graden af et 'almindeligt' polynomium, i én variabel, opfylder regnereglerne

$$\deg(pq) = \deg p + \deg q \quad \text{og} \quad \deg(p+q) \leq \max\{\deg p, \deg q\}, \tag{4}$$

hvor uligheden kun optræder når p og q har samme grad *og* højestegradsledene ophæver hinanden fordi de optræder med modsat fortegn i p og q. Samme regler skulle gerne gælde for polynomier på E.

Graden af et polynomium ændrer sig selvfølgelig ikke ved omskrivningen til standardform, så y^2 må have samme grad som $x^3 + Ax + B$. Derfor vedtages:

Definition 2.4 (Grad af polynomier på E)**:** For et polynomium p på E betegnes graden ved $\deg_E(p(x, y))$.Når x og y er polynomier på E, defineres deres grad som:

$$\deg_E x = 2 \quad \text{og} \quad \deg_E y = 3.$$

Mere generelt bliver $\deg_E(v(x)) = 2 \deg v$, hvor v er et polynomium i én variabel. For et polynomium på standardform bliver det til

$$\deg_E p(x, y) = \deg_E(u(x) + yv(x)) = \max\{2 \deg u, 3 + 2 \deg v\}$$

Bemærk lighedstegn tilsidst: Her kan ikke være ulighed, fordi $u(x)$ og $yv(x)$ kan ikke have samme grad da graden af $u(v)$ er lige og graden af $yv(x)$ er ulige.

For at se at definitionen giver opfylder regnereglerne i (4), bruger vi følgende

Definition 2.5 (Konjugering og norm af polynomier)**:** For et polynomium $p(x, y) = u(x) + yv(x)$ defineres:

$$\begin{aligned} \overline{p}(x, y) &= u(x) - yv(x) \\ \mathrm{N}(p(x, y)) &= p(x, y)\bar{p}(x, y) = u(x)^2 - \left(x^3 + Ax + B\right) v(x)^2 \end{aligned}$$

Disse definitioner har nogle egenskaber som vi kan får brug for:

- Vi så før at når et punkt $P = (x_p, y_p)$ ligger på kurven, gør $-P = (x_p, -y_p)$ også. Når p er et polynomium på E bliver altså $p(P) = \bar{p}(-P)$

- Normen $\mathrm{N}(p(x, y))$ har den egenskab at den er uafhængig af y. Endvidere er graden af $\mathrm{N}(p(x, y))$, betragtet som et polynomium i *én* variabel, lig med graden af $p(x, y)$, betragtet som et polynomium på E, hvilket kan ses af definition 2.4 og de almindelige regneregler i (4).

Nu ses det let at (4) gælder for vilkårlige polynomier p og q:

$$\deg_E(pq) = \deg(\mathrm{N}(pq)) = \deg(pq \cdot \overline{pq}) = \deg(\mathrm{N}(p)) + \deg(\mathrm{N}(q)) = \deg_E p + \deg_E q$$

og, med $p(x, y) = u_p(x) + yv_p(x)$ og $q(x, y) = u_q(x) + yv_q(x)$:

$$\begin{aligned}
\deg_E(p + q) &= \deg_E((u_p + u_q)(x) + y(v_p + v_q)(x)) \\
&= \max\{2\deg(u_p + u_q), 3 + 2\deg(v_p + v_q)\} \\
&\leq \max\{2\deg u_p, 3 + 2\deg v_p, 2\deg u_q, 3 + 2\deg v_q\} \\
&= \max\{\deg_E p, \deg_E q\}
\end{aligned}$$

Endelig kan vi se at hvis $p(x, y) = 0$ for alle $(x, y) \in E$, må både u og v være nulpolynomiet; for hvis p altid er nul, må $\mathrm{N}(p) = p\bar{p}$ også være det. Altså må der gælde

$$u^2 = (x^3 + Ax + B)v^2,$$

hvilket kun kan lade sig gøre hvis $u = v = 0$, da højresiden har ulige grad, og venstresiden lige (som polynomier i x).

Heraf følger også at hvis $u_1(x) + yv_1(x) = u_2(x) + yv_2(x)$ på hele E, så er $u_1 = u_2$ og $v_1 = v_2$. Vi har altså at standardformen $p = u + yv$ er entydig.

2.2.2 Rationelle funktioner

En rationel funktion f på E er en funktion som kan skrives $f = p/q$, hvor p og q er polynomier på E. Bemærk at denne repræsentation af f ikke er entydig; tværtimod kan på samme tid $f = r/s$ – når blot at $sp = rq$.

Vi siger at værdien af en funktion f er *veldefineret i et punkt* P hvis f kan skrives $f = p/q$ på en måde så nævneren $q(P) \neq 0$. Hvis $p(P) = 0$ er selvfølgelig $f(P) = 0$; i dette tilfælde er $1/f$ så ikke defineret i P, og vi siger at f har en *pol* i P.

I det følgende betyder „funktion“, hvis ikke andet fremgår, altid „rationel funktion på E som ikke er identisk nul“.

Definition 2.6 (Værdien af en funktion i $\mathcal{O}$)**:** defineres på samme måde som grænseværdien af en almindelig rationel funktion af én variabel x, for x gående mod ∞. Hvis $f = p/q$ afhænger $f(\mathcal{O})$ af $\deg_E p$ og $\deg_E q$:

$$f(\mathcal{O}) = \begin{cases} 0 & \text{hvis} \quad \deg_E p < \deg_E q \\ a/b & \text{hvis} \quad \deg_E p = \deg_E q \\ \text{udefineret} & \text{hvis} \quad \deg_E p > \deg_E q, \end{cases}$$

hvor a og b er højestegradskoefficienterne i henholdsvis p og q.

I det tilfælde at f er udefineret i $\mathcal{O}$ siger vi også at f har en pol i $\mathcal{O}$. Da et polynomium er en rationel funktion hvor nævneren er konstant, og altså har grad nul (f.eks. $p = p/1$), har alle polynomier en pol i $\mathcal{O}$.

Den næste sætning hjælper os til at beskrive poler.

Sætning 2.7: *I ethvert punkt P på kurven E findes der en funktion g med $g(P) = 0$, som opfylder at enhver rationel funktion r på E kan skrives som*

$$f = g^d s$$

hvor s er en funktion som er veldefineret men ikke nul i P, og d er et helt tal, som er entydigt bestemt ved f og punktet P.

Bevis. Lad $P = (x_p, y_p)$, $f = p/q$, og $p(x, y) = u(x) + yv(x)$.

Først bemærkes at hvis f er (veldefineret og) forskellig fra nul i P, er den eneste mulighed at sætte $d = 0$ (da $g(P) = 0$ vil $g^d s$ blive nul i P hvis $d \neq 0$). Hvis f har en pol i P, er $1/f$ nul i P, og hvis $1/f = g^d s$ får man $f = g^{-d}(1/s)$, hvor $1/s$ er veldefineret fordi $s \neq 0$. I beviset antages derfor at $f(P) = 0$ — nærmere betegnet at $p(P) = 0$ og $q(P) \neq 0$.

Alt efter hvilket P det drejer sig om deles beviset op i tre dele, nemlig $\mathcal{O}$, andenordens punkter, og alle de 'almindelige' punkter. Først de almindelige punkter:

I dette tilfælde sættes $g = x - x_p$. Hvis $\bar{p}(P) \neq 0$ kan vi forlænge brøken p/q med $\bar{p}$, og får

$$f = \frac{\mathrm{N}(p)}{q\bar{p}},$$

som stadig er veldefineret i P, og hvor tælleren er et polynomium i x. Eftersom $p(P) = 0$ må $p(P)\bar{p}(P) = 0$. Så er x_p rod i $\mathrm{N}(p)$, så derfor kan tælleren skrives $N(p) = g^d p'$, hvor p' er et polynomium i x med $p'(x_P) \neq 0$ – og så er vi færdige idet

$$f = g^d \frac{p'}{q\bar{p}},$$

hvor både p', q og $\bar{p}$ er forskellige fra nul i P.

Hvis $\bar{p}(P) = 0$ er $u(x_p)+y_p v(x_p) = u(x_p)-y_p v(x_p) = 0$. Da P ikke er et andenordens punkt er $y_p \neq 0$, og derfor må x_p være rod i både u og v. Altså kan vi skrive $p(x, y) = (x - x_p)(u_1(x) + yu_1(x))$. Kald $p_1 = u_1(x) + yu_1(x)$. Hvis $p_1(P) \neq 0$ kan vi sætte $d = 1$, og i modsat fald kan vi blive ved med at faktorere g ud k gange, indtil enten $p_k(P) \neq 0$ – så er vi færdige, og sætter $d = k$, eller fordi $\bar{p}_k(P) \neq 0$ – så vi er tilbage i situationen fra før, med

$$\frac{p_k}{q} = g^d s \quad \text{og derfor} \quad f = g^k \frac{p_k}{q} = g^{d+k} s.$$

Graden af u_k og v_k falder hver gang (som polynomium i x har g grad 1, så $u_{k-1} = gu_k \Leftrightarrow \deg u_k = \deg u_{k-1} - 1$), så det virker for et endeligt k, da graden af u er endelig.

Hvis P har orden 2 er x_p en af de tre rødder i højresiden af kurvens ligning; lad $P = (\omega_1, 0)$. Da $p(P) = u(\omega_1) + 0 \cdot v(\omega_1) = 0$, må altså $u(\omega_1) = 0$, så u kan skrives som $(x - \omega_1)u_1$. Da de tre rødder er forskellige er $(x - \omega_2)(x - \omega_3)$ forskellig fra nul i P, og vi kan igen forlænge p/q:

$$\begin{aligned} f = \frac{u + yv}{q} &= \frac{(x - \omega_1)(x - \omega_2)(x - \omega_3)u_1 + y(x - \omega_2)(x - \omega_3)v}{(x - \omega_2)(x - \omega_3)q} \\ &= \frac{y^2 u_1 + y(x - \omega_2)(x - \omega_3)v}{(x - \omega_2)(x - \omega_3)q} \\ &= y\left(\frac{(x - \omega_2)(x - \omega_3)v + yu_1}{(x - \omega_2)(x - \omega_3)q}\right) \end{aligned}$$

Hvis den sidste parentes giver nul i P kan man fortsætte med at trække faktor y ud – denne gang ser vi at $v(\omega_1) = 0$ så $v = (x - \omega_1)v_1$. Igen kan stopper processen efter endeligt mange gange, da $(x - \omega_1)$ kun kan gå endeligt mange gange op i u og v.

Det vil sige at for andenordens P kan vi bruge $g = y$.

Sidste tilfælde $P = \mathcal{O}$. Her sættes $g = x/y$ og $d = \deg_E q - \deg_E p$. Da $f(\mathcal{O}) = 0$ er d positiv, ifølge definition 2.6. Nu bliver $\deg_E y^d p = 3d + \deg_E p$ og $\deg_E x^d q = 2d + \deg_E q$. Så er

$$\deg_E x^d q = 2d + \deg_E q = 2d + d + \deg_E p = \deg_E y^d p,$$

hvilket ifølge definition 2.6 betyder at $x^d q / y^d p = (x/y)^d (p/q)$ er endelig i $\mathcal{O}$. Så kan vi skrive f på den ønskede form:

$$f = g^d \left(\frac{x}{y}\right)^d f.$$

Entydighed af k: Hvis to funktioner g og $\tilde{g}$ opfylder betingelsen for sætningen i punktet P kan man skrive

$$g = \tilde{g}^m s \quad \text{og} \quad \tilde{g} = g^n \tilde{s}$$

Tilsammen giver det $g = (g^n \tilde{s})^m s = g^{mn} \tilde{s}^m s$. Hvis man dividerer med g får man $g^{mn-1} \tilde{s}^m s = 1$. Hvis man nu antager $mn \neq 1$ og indsætter punktet P for man $1 = 0$ (da $g(P) = 0$), en modstrid. Derfor må $m = n = 1$. Derfor kan enhver funktion f skrives

$$f = g^d t = \tilde{g}^d s^d t \qquad \square$$

Definition 2.8: Entydigheden af d gør at *orden af en funktion* i et punkt P er veldefineret: Lad f være en funktion og P et punkt på E. Så er

$$\mathrm{ord}_P(f) = \text{tallet } d \text{ fra sætning 2.7}$$

Hvis d er positiv siger vi at f har et nulpunkt af orden d i P; hvis d er negativ har f en pol af orden $|d|$ i P.

Beviset for sætning 2.7 giver en måde at beregne $\mathrm{ord}_{\mathcal{O}}(f)$ direkte, når bare vi kender en repræsentation af f, hvor graden af tæller og nævner er kendte. Specielt for et polynomium p er $\mathrm{ord}_{\mathcal{O}}(p) = -\deg_E p$. Vi kan også beregne ordenen af et produkt: Hvis man skriver to funktioner f_1 og f_2 på den form som benyttes i sætningen, og bagefter ganger dem sammen, ser man at

$$\mathrm{ord}_P(f_1 f_2) = \mathrm{ord}_P(f_1) + ord_P(f_2). \tag{5}$$

Eksempel 2.9: Betragt $\ell = x - x_0$ som polynomium på E. ℓ kan have enten et eller to nulpunkter på E — et hvis $x_0 \in \{\omega_1, \omega_3, \omega_2\}$ og to for alle andre x_0. Hvis $x = \omega_1$ (wlog) kan ℓ skrives

$$\ell = y^2 \frac{1}{(x - \omega_2)(x - \omega_3)},$$

så $\mathrm{ord}_{(\omega_1,0)}(\ell) = 2$. For $x_0 \neq \omega$ er $\ell = (x - x_0)^1 \cdot 1$, dvs. at $\mathrm{ord}_P(\ell) = 1$ når P er et af de to punkter med x-koordinat x_0. ℓ har altså et nulpunkt af orden 2 eller to nulpunkter af orden 1.

Uanset værdien af x_0 gælder $\deg_E \ell = 2$, så $\mathrm{ord}_{\mathcal{O}}(\ell) = -2$; ℓ har altså en pol af orden 2 i $\mathcal{O}$.

Lemma 2.10: *Lad p være et polynomium på E, og $\mathcal{R}$ betegne mængden af rødder i p: $\mathcal{R} = \{P \in E | p(P) = 0\}$. Så gælder*

$$\sum_{P \in \mathcal{R}} \mathrm{ord}_P(p) = \deg_E p$$

Bevis. Vi ved at $\mathrm{N}(p) = p\bar{p}$ er et polynomium i x af grad $\nu = \deg_E p$, og kan altså skrives som et produkt af ν faktorer – ikke nødvendigvis forskellige – af formen $(x - x_i)$. Vi så at hver sådan faktor har to nulpunkter af orden 1 eller et enkelt nulpunkt af orden 2. Den samlede orden af nulpunkter for $p\bar{p}$ er altså 2ν.

Men $\bar{p}$ har nøjagtig lige så mange nulpunkter som p (da $p(Q) = \bar{p}(-Q), Q \in E$), og den samlede orden af nulpunkterne er også ens for de to ($\bar{p}(x, -y) = p(x, y)$ medfører at $\mathrm{ord}_{(x,-y)}(\bar{p}) = \mathrm{ord}_{(x,y)}(p)$). Derfor må den samlede orden af nulpunkterne være ν både for p og $\bar{p}$. □

Dette resultat kan udvides til at gælde rationelle funktioner:

Sætning 2.11: *Når f er en (rationel) funktion på E så gælder*

$$\sum_{P \in E} \mathrm{ord}_P(f) = 0$$

Bevis. Hvis f er et polynomium så vi at der gælder $\mathrm{ord}_{\mathcal{O}}(f) = -\deg_E f$. Så følger sætningen direkte af lemma 2.10, da orden jo er nul i alle punkter som *ikke* er nulpunkter.

For $f = p/q$, bemærk at da $\mathrm{ord}_P(1/q) = -\mathrm{ord}_P(q)$, og så giver ligning (5)

$$\sum_{P \in E} \mathrm{ord}_P(p/q) = \sum_{P \in E} \mathrm{ord}_P(p) - \sum_{P \in E} \mathrm{ord}_P(q) = 0. \qquad \square$$

Vi skal bruge to lemmaer mere. De fortæller hvilke rationelle funktioner der i virkeligheden er polynomier, og begge er – sammen med deres beviser – taget næsten uændret fra Charlap og Robbins (1988).

Lemma 2.12: *Ethvert ikke-konstant polynomium på E har mindst to enkelte nulpunkter eller et dobbelt nulpunkt.*

Bevis. Hvis polynomiet ikke er konstant må det indeholde x eller y, og derfor have grad mindst to. Påstanden følger nu direkte fra lemma 2.10. □

Lemma 2.13: *En funktion f som ikke har andre poler end $\mathcal{O}$, er et polynomium.*

Den tilsvarende påstand for rationelle funktioner i én variabel – at en rationel funktion som ikke har nogen poler, er et polynomium – benyttes uden bevis.

Bevis. Bemærk at $f = p/q$ kan skrives som $f(x, y) = a(x) + yb(x)$, hvor a og b er rationelle funktioner i x, ved at gange i tæller og nævner med $\bar{q}$. Lad $\bar{f}$ betegne den konjugerede $a(x) - yb(x)$.

Hvis f er fri for poler (i endelige punkter) må det samme gælde $\bar{f}$, og derfor også $f + \bar{f} = 2a(x)$. a kan altså ikke have en pol, og må derfor være et polynomium. Dette

medfører så at $f - a = yb$ ikke har poler, og dermed at $(yb)^2 = (x^3 + Ax + B)b^2$ ikke har det. Hvis b har en pol i et punkt x_0 har b^2 en *dobbelt* pol i x_0. Den eneste måde for $(yb)^2$ at undgå at få en pol i x_0 vil så være at $(x^3 + Ax + B)$ har en dobbelt *rod* i x_0 – og dette kan kun ske hvis E, imod vores tidligere antagelse, er singulær. Altså må $b(x)$ være fri for poler, og altså er også b et polynomium. Da både a og b således er polynomier i x, er $f = a + yb$ et polynomium på E. □

2.2.3 Divisorer og Picard-gruppen

Definition 2.14: For en mængde C betegner den *fri abelske gruppe med basis* C mængden af endelige linære kombinationer med heltallige koefficienter af elementer i C

$$\left\{ \sum_{P \in C} n_P P \,\middle|\, n_P \in \mathbb{Z}, n_P \neq 0 \text{ for endeligt mange } P \right\}$$

med addition defineret på samme måde som for 'almindelige' linære kombinationer:

$$\sum_{P \in C} n_P P + \sum_{P \in C} m_P P = \sum_{P \in C} (n_P + m_P) P.$$

Den fri abelske gruppe med basis E kalder vi for *divisorerne på* E, og bruger betegnelsen $\mathcal{D}(E)$. For at kunne se forskel på addition af punkter (som defineret i 2.2) og addition mellem divisorer bruges notationen $\langle P \rangle$ til at betegne en divisor som har koefficient 1 til punktet P, og koefficient nul til alle andre punkter. Nulelementet i $\mathcal{D}$ er divisoren hvor alle koefficienter er nul, og betegnes $0_{\mathcal{D}}$.

Definition 2.15: For en divisor $D \in \mathcal{D}(E)$ betegner *graden af* D størrelsen

$$\deg D = \sum_{P \in E} n_P$$

og normen af D defineres

$$|D| = \sum_{P \neq \mathcal{O}} |n_P|.$$

Hver gang vi har en funktion f kan vi nu danne en divisor, som vi kalder δf:

$$\delta f = \sum_{P \in E} \operatorname{ord}_P(f) \langle P \rangle$$

Divisorer som er fremkommet på denne måde kaldes *funktionelle* (eng. *principal divisors*[1]). Sætning 2.11 siger at funktionelle divisorer har grad nul. Hvis vi kalder mængden af divisorer med grad nul for $\mathcal{D}_0$ og mængden af funktionelle divisorer for $\mathcal{F}$ har

[1] 'funktionelle divisorer' er mit eget bud på en oversættelse af det engelske udtryk. Jeg synes 'principal' lyder underligt på dansk, og den direkte oversættelse, 'hoved-divisorer', endnu værre. Desuden kommer de funktionelle divisorer netop fra funktioner, så jeg synes det giver mening at kalde dem det.

vi altså $\mathcal{F} \subset \mathcal{D}_0$. Det er klart at $\mathcal{D}_0$ er en undergruppe af $\mathcal{D}$, og det samme gælder for $\mathcal{F}$, hvilket følger af regnereglen (5) for ord_P af et produkt:

$$\delta fg = \sum_{P \in E} \mathrm{ord}_P(fg)\langle P \rangle = \sum_{P \in E} \left(\mathrm{ord}_P(f) + \mathrm{ord}_P(g)\right) \langle P \rangle = \delta f + \delta g \tag{6}$$

Definition 2.16: Den nemmeste måde at definere Picard-gruppen Pic_0 på, er som gruppen af kongruensklasser:

$$\mathrm{Pic}_0 = \mathcal{D}_0/\mathcal{F}$$

Det giver med det samme anledning til en ækvivalensrelation på $\mathcal{D}_0$, nemlig kongruens modulo $\mathcal{F}$: Relationen $\sim$ defineres ved

$$D_1 \sim D_2 \quad \Longleftrightarrow \quad D_1 - D_2 \in \mathcal{F} \quad \Longleftrightarrow \quad D_1 - D_2 \text{ er funktionel}$$

Definitionen skal forstås på den måde at et element i Pic_0 er en delmængde af $\mathcal{D}_0$ i hvilken alle elementer er ækvivalente mht. relationen $\sim$. Nulelementet i Pic_0 er $\mathcal{F}$, og for enhver funktionel divisor D gælder $D \sim 0_{\mathcal{D}}$. At Pic_0 er en gruppe følger, med grundlæggende gruppeteori, af at $\mathcal{F} \subset \mathcal{D}_0$.

Sammenhængen i (6) mellem multiplikation af funktioner og addition af divisorer gør at vi kan bruge vores viden om polynomier til at vise følgende:

Sætning 2.17: *For ethvert element $D \in \mathcal{D}_0(E)$ findes der et entydigt punkt $P \in E$ så*

$$D \sim \langle P \rangle - \langle \mathcal{O} \rangle$$

I afsnit 2.1 så vi på forskellige konfigurationer af linjer og deres skæringspunkter med kurven. Nu vil vi betragte en linje som et polynomium ℓ på E, af grad 2 eller 3. Nulpunkterne for ℓ er præcis skæringspunkterne. Vi ved altså at vi, ved hjælp af linjer, kan lave følgende funktionelle divisorer:

(i) $\langle P \rangle + \langle Q \rangle + \langle R \rangle - 3\langle \mathcal{O} \rangle$ for en linje der skærer E i tre punkter
(ii) $2\langle P \rangle + \langle Q \rangle - 3\langle \mathcal{O} \rangle$ for en tangent i P, som skærer kurven i et andet punkt Q.
(iii) $2\langle P \rangle - 2\langle \mathcal{O} \rangle$ for en lodret linje gennem et andenordens punkt P.
(iv) $\langle P \rangle + \langle -P \rangle - 2\langle \mathcal{O} \rangle$ for en lodret linje gennem P og $-P$.
(v) $3\langle P \rangle - 3\langle \mathcal{O} \rangle$, for tangenten i et 'vendepunkt' som de der nævntes i afsnit 2.1.1.

Teknikken til at bevise sætning 2.17 er at vise at man for enhver divisor med norm større end 1 kan finde en ækvivalent divisor hvis norm er mindre. Dette er et lemma i sig selv:

Lemma 2.18 (Linær reduktion)**:** *For enhver divisor $D_1 \in \mathcal{D}_0$ med $|D_1| > 1$ findes der en divisor $D_2 \in \mathcal{D}_0$ som opfylder $D_2 \sim D_1$ og med $|D_2| < |D_1|$.*

Bevis. Lad $D_1 = \sum_P n_P\langle P\rangle$, og antag $|D_1| > 1$. Vi vil finde $L \in \mathcal{D}_0$ som er divisor for en ret linje, så den har en af de fem former beskrevet ovenfor, og som kan reducere D_1, dvs. enten $|D_1 + L| < |D_1|$ eller $|D_1 - L| < |D_1|$. Der er forskellige tilfælde:

Enten er der et punkt P med $|n_P| \geq 2$. I dette tilfælde vælges divisoren til tangenten i P. Hvis P er et andenordens punkt bliver $L = 2\langle P\rangle - 2\langle\mathcal{O}\rangle$, og vi kan reducere normen ved at trække L fra (lægge til hvis n_P er negativ). Hvis ikke P er af orden 2, sættes $L = 2\langle P\rangle + \langle Q\rangle - 3\langle\mathcal{O}\rangle$, og vi kan reducere $|n_P|$ med to mod at gøre $|n_Q|$ (højest) én større. Alt i alt reduceres normen med mindst én.

Hvis der ikke findes et $|n_P| \geq 2$, må der være mindst to punkter P og Q med koefficienter ± 1, da normen af D_1 jo er større end 1. Hvis P og Q kan vælges så n_P har samme fortegn som n_Q, kan vi reducere med $L = \langle P\rangle + \langle Q\rangle + \langle R\rangle - 3\langle\mathcal{O}\rangle$. Derved bliver $n_P = n_Q = 0$, og $|n_R|$ forøges med højest én, og igen har vi reduceret normen.

Hvis endelig $D_1 = \langle P\rangle - \langle Q\rangle$ (koefficienten til $\mathcal{O}$ er nul her, da $D_1 \in \mathcal{D}_0$) må vi først lægge $\langle Q\rangle + \langle -Q\rangle - 2\langle\mathcal{O}\rangle$, divisoren for en lodret linje gennem Q, til D_1, og så bruge lemmaet på $\langle P\rangle + \langle -Q\rangle - 2\langle\mathcal{O}\rangle$, som er ækvivalent med og har samme norm som D_1. □

Bevis for 2.17. Bemærk først at hvis D selv er funktionel er $D \sim 0_{\mathcal{D}} = \langle\mathcal{O}\rangle - \langle\mathcal{O}\rangle$. Hvis D ikke er funktionel, kan vi benytte lemmaet et endeligt antal gange og få den ønskede form.

For at være sikre på at denne repræsentation er entydig, kan vi prøve at antage $\langle P\rangle - \langle\mathcal{O}\rangle \sim \langle Q\rangle - \langle\mathcal{O}\rangle$. Så skal $\langle P\rangle - \langle Q\rangle$ være funktionel. Ved at først at trække divisoren for en lodret linje gennem P fra, og derefter lægge divisoren for linjen gennem $-P$ og Q til, ser vi at så skal $\langle R\rangle - \langle\mathcal{O}\rangle$ være funktionel. Men hvis $\delta f = \langle R\rangle - \langle\mathcal{O}\rangle$ for en rationel funktion f, så har den for det første ingen poler i endelige punkter, og er derfor et polynomium (lemma 2.13) som dernæst ses kun har en rod, hvilket er i modstrid med lemma 2.12. Konklusionen er at P er entydigt identificeret af D. □

Det fantastiske ved sætning 2.17 er at det fortæller at der er en bijektion mellem E og Pic_0. Elementerne i Pic_0 er jo netop ækvivalensklasser ved $\sim$, og sætningen viser at enhver af disse ækvivalensklasser kan identificeres med et entydigt punkt på E. Nu skal vi bare overbevise os om at billedet af addition i Pic_0 er den addition vi definerede i afsnit 2.1. Hvis det er sandt har vi bevist at E og Pic_0 er isomorfe, hvilket til overflod medfører at $(E, +)$ er en gruppe. Vi skal altså bevise

Sætning 2.19: *Når P, Q og R er punkter på en elliptisk kurve, gælder*

$$(\langle P\rangle - \langle\mathcal{O}\rangle) + (\langle Q\rangle - \langle\mathcal{O}\rangle) \sim \langle R\rangle - \langle\mathcal{O}\rangle \quad \Longleftrightarrow \quad P + Q = R$$

Bevis. Divisoren længst til venstre kan skrives $\langle P\rangle + \langle Q\rangle - 2\langle\mathcal{O}\rangle$. Kald denne divisor for for D. Se på linjen gennem P og Q (eller tangenten i P, hvis $P = Q$). Hvis vi antager

$P+Q=R$, ved vi at $-R$ ligger på en linje hvis divisor er $L=\langle P\rangle+\langle Q\rangle+\langle -R\rangle-3\langle\mathcal{O}\rangle$. Nu bliver

$$D-L=-\langle -R\rangle+\langle\mathcal{O}\rangle\sim D.$$

Nu kan vi tage den den lodrette linje gennem R. Den har divisor $L'=\langle R\rangle+\langle -R\rangle-2\langle\mathcal{O}\rangle$, så ved at lægge den til får vi det ønskede resultat

$$D-L+L'=\langle R\rangle-\langle\mathcal{O}\rangle\sim D.$$

Entydigheden i sætning 2.17 giver så at relationen $D\sim\langle R\rangle-\langle\mathcal{O}\rangle$ kun er sand hvis $P+Q=R$. □

Beviset virker næsten for simpelt, men det fungerer faktisk i alle tilfælde, hvis man tillader at P, Q og R ikke behøves at være forskellige. Med $Q=-P$ fås $R=\mathcal{O}$ og $D\sim 0_{\mathcal{D}}$.

2.3 Endelige kurver

Indtil nu har vi behandlet kurven E som et geometrisk objekt i $\mathbb{R}^2$ – dog med den krølle at vi har forlangt, og brugt i beviserne, at E er ikke-singulær *som kurve i* $\mathbb{C}^2$.

Nu vil vi prøve at udskifte $\mathbb{R}$ i definitionen med et *endeligt legeme*. Jeg vil ikke give en fuldstændig definition på hvad et endeligt legeme er, men blot gøre opmærksom på de egenskaber vi skal bruge:

- Definitionen af et legeme minder om definitionen af en gruppe, men med to forskellige operationer, som regel skrives '+' og '×'. Aksiomerne for et legeme beskriver at begge operationer skal være invertible, og at de skal opføre sig 'som de plejer'. Resultatet er at alt hvad man kan med polynomier i $\mathbb{R}$ (gange dem sammen, tælle rødder etc.) kan man også gøre i et legeme.

- Teorien om legemer siger at der for ethvert legeme k findes en *entydig algebraisk afslutning* K. Det kræver nogen overvejelse at indse dette, og jeg vil ikke komme nærmere ind på det. Se f.eks. (Hungerford, 1996, afsnit 10). K er altså et legeme, $k\subset K$, og ethvert polynomium defineret over K har rødder i K.

- Et legemes *karakteristik* er det antal gange man skal lægge dets *multiplikative identitetselement* (1-elementet) sammen med sig selv for at få det *additive identitetselement* (nulelementet).

Et endeligt legeme er simpelthen en endelig mængde for hvilke disse betingelser gælder. Det grundlæggende eksempel på et endeligt legeme er $\mathbb{Z}_p$, de hele tal modulo et primtal p.

Hvis vi ser på et endeligt legeme k og dets algebraiske afslutning K, kan vi definere en elliptisk kurve over k på samme måde som i (1), ved simpelthen at udskifte $\mathbb{R}$ med

k. Det vil sige lade $A, B \in k$ og $x, y \in K$. De definitioner vi har bliver ved med at virke (med undtagelse af legemer af karakteristik 2 og 3, som har nogle 'uheldige' egenskaber som vi skal se kort på, for derefter at glemme dem igen).

Entydigheden af K gør at polynomier med koefficienter i k har entydige rødder i K. Det vil sige at ω_1, ω_2 og ω_3 fortsat er forskellige når k er et legeme og $\Delta \neq 0$.

Man kan spørge hvordan den del af definitionen som omhandler tangenlinjen kan blive ved med at virke, når man definerer E over et endeligt legeme. Argumentet for definitionen – at polynomiet for tangentlinjens skæring har en dobbelt rod i punktet – benytter jo differentiation, og det er ikke ret oplagt hvad dette vil sige i et endeligt legeme. Svaret på dette spørgsmål ligger imidlertid lige for, hvis man lader den sædvanlige definition af den afledede til et polynomium være rent algebraisk: Hvis polynomiet f – for eksempel polynomiet for skæringspunkterne, men nu defineret over k – har en rod i et punkt x_0, kan det skrives som

$$f(x) = (x - x_0)^m g(x),$$

hvor g er et polynomium som ikke har rod i x_0, og $m \geq 1$ er et helt tal. Nu kan vi tage den afledede:

$$f'(x) = m(x - x_0)^{m-1} g(x) + (x - x_0)^m g'(x) \tag{7}$$

Da $m \geq 1$ er andet led $(x-x_0)^m g'(x)$ er altid nul for $x = x_0$. Til gengæld kan $mg(x)$ ikke være nul i x_0, så hvis x_0 er rod i f' må det være fordi at $(x-x_0)^{m-1}$ bliver nul for $x = x_0$, og det vil sige at $m > 1$ (da $(x - x_0)^0 = 1$ uanset hvilket x_0). Det vil sige at f har en (mindst) dobbelt rod i x_0, hvilket er præcis den konklusion vi var ude efter før. Produktreglen, som benyttes i (7), følger direkte af den algebraiske definition af den afledede for polynomier. (Hungerford, 1996, afsnit 10.5, specielt øvelse 8).

Alle de resterende argumenter om polynomier, rødder og rationelle funktioner i afsnit 2.2 gælder uændrede i et ethvert legeme, og det samme gør konklusionen, nemlig at $(E, +)$ er en abelsk gruppe når E er defineret over k.

2.3.1 Karakteristik 2 og 3

For at indse hvordan det kan være at definitionen bryder sammen i et legeme med karakteristik 2, bemærk at i et sådant legeme er $1+1 = 0$. Division med 2 er således det samme som division med nul, og additionsformlerne på side 7 bliver ubrugelige. Faktisk viser det sig at selve definitionen af kurven, inklusive ligning (1) og diskriminanten Δ, bliver anderledes i legemer af karakteristik 2 eller 3. At $1 + 1 = 0$ er ikke det samme som at der ikke *findes* andet end 1 og 0 – man kan konstruere et såkaldt Galois-legemer $\mathbb{F}_{2^m}$ med 2^m elementer som har karakteristik 2, og som er praktiske at bruge i nogen sammenhænge. I (Enge, 1999) kan man se versioner af definitionerne og de relevante beviser, som gælder for legemer af karakteristik 2 og 3.

For nu vil jeg gå ud fra at $k = \mathbb{Z}_p$ hvor $p > 3$ er et primtal, og så ikke tænke mere på det.

2.3.2 k-rationelle punkter

Punkterne på kurven defineret over k er ligger i $K \times K$, da deres koordinaterne er rødder i et polynomium over k. De punkter på $E(K)$ som har koordinater i k – som er endelig – kaldes for *k-rationelle*. Mængden af k-rationelle punkter betegnes $E(k)$. Så er det vigtigt at $E(k)$ også er en gruppe, altså lukket under addition:

Sætning 2.20: *Lad k være et legeme, og E en elliptisk kurve defineret over k. Så er $(E(k), +)$ en gruppe. Dvs. at*

$$P, Q \in E(k) \Rightarrow P + Q \in E(k)$$

Bevis. Lad $P = (x_1, y_1)$ og $Q = (x_2, y_2)$. Når $P, Q \in E(k)$, så er x_1, y_1, x_2 og y_2 elementer i k. Da k er et legeme ligger x_3 og y_3 (beregnet med additionsformlerne fra definition 2.2) også i k, og dermed $P + Q$ i $E(k)$. □

Vi har faktisk allerede set at denne sætning virker, nemlig da vi så på punkter på en elliptisk kurve defineret over $\mathbb{R}$: Hver gang vi har to punkter på $E(\mathbb{R})$ kan vi finde et tredje, som også ligger på $E(\mathbb{R})$, men for at kunne definere kurven og overbevise os om gruppestrukturen, må vi have hele $E(\mathbb{C})$ med.

I et endeligt legeme k kan vi altså definere en elliptisk kurve $E(k)$, og får derved en endelig gruppe $(E(k), +)$. I næste afsnit skal vi se et eksempel på hvad endelige grupper kan bruges til.

3 Anvendelse: Kryptologi

Anvendelse af tal- og gruppeteori til krypteringsalgoritmer er et meget populært eksempel på konkret udnyttelse af disse grene af matematikken – og af teoretisk datalogi for den sags skyld. Og med god grund: Krypteret kommunikation via internettet er dagligdag for de fleste, og det er sjovt at tænke på at en del af denne kommunikation, som for eksempel kan være overførsel af data for en dankortbetaling *on-line*, består af punkter på elliptiske kurver!

Asymmetrisk kryptering er en betegnelse for et system hvor man har en *offentlig nøgle* og en *privat nøgle*. Hvis afsenderen af en hemmelig meddelelse kender modtagerens offentlige nøgle, kan hun ved hjælp af en kendt algoritme indkode meddelelsen på en måde så modtageren, ved hjælp af sin private nøgle, nemt kan afkode den igen – men det er 'svært' for en tredje person at afkode den. Asymmetrien gør at man undgår det klassiske problem i kryptografi; nemlig at sørge for at afsender og modtager, men ingen andre, kender koden.

Vi skal se på hvordan sådan et system kan sættes op ved hjælp af en algebraisk gruppe, og på hvad det i denne sammenhæng vil sige at noget er 'svært'.

Til sidst skal vi se på hvordan man kan bruge gruppen for en elliptisk kurve, og hvilke problemer og fordele der kommer ud af det.

For at få terminologien på plads: Et *kryptografisk system* eller *kryptosystem* består af funktioner til at kryptere og dekryptere meddelelser, sammen med en (matematisk) beskrivelse af de mængder meddelelser og nøgler kan tages fra. Der skelnes i kryptologi mellem *kryptografi*, det at anvende kryptografiske systemer, og *kryptoanalyse*, som betegner forsøg på at bryde koden, f.eks. ved at dekryptere en meddelelse uden at have den private nøgle.

Traditionen foreskriver at afsenderen af krypterede meddelelser hedder Alice og modtageren Bob, så det skal de også hedde her[2]. Der er også en tredje person, Onde Åge, som opsnapper de krypterede meddelelser og forsøger at bryde koden.

3.1 ElGamal systemet

ElGamals system består af en krypteringsdel og en signaturdel. Formålet med det sidste er, i stedet for at hemmeligholde indholdet af en meddelelse, at fastslå afsenderens identitet. Jeg har valgt kun at se på krypteringsdelen; de to dele er variationer over samme tema, så jeg kunne lige så let have valgt den anden.

Systemet er baseret på en *cyklisk gruppe* G. At gruppen er cyklisk betyder at der findes et element $\alpha \in G$ som har orden $|G|$. Et sådant element kaldes en *frembringer* for gruppen, og man siger at man kan *generere* alle andre gruppeelementer ud fra α:

[2] Enge (1999) kalder dem for Kevin og Laura, men det må være for at sende en skjult hilsen til nogen han kender.

Hvis gruppen har orden n kan man finde en rækkefølge for elementerne ved at regne

$$\alpha, 2\alpha, \ldots, (n-1)\alpha, n\alpha = 0$$

– bemærk at jeg har valgt at fortsætte med additiv notation. $j\alpha$ betyder „α lagt sammen med sig selv j gange“ (j er et heltal, ikke et element i G). Antallet af elementer i G kaldes gruppens *orden*.

Definition 3.1 (ElGamal Kryptosystem)**:** Der er fastlagt en cyklisk gruppe $(G, +)$, som har orden n og frembringeren α.
Bob vælger et helt tal $a \in \mathbb{Z}_n$ som sin private nøgle og så generere elementet $\beta = a\alpha$, den offentlige nøgle.
Alice skal, for at kryptere en meddelelse $m \in G$ til Bob, vælge et tilfældigt tal (en midlertidig nøgle) $t \in \mathbb{Z}_n$ og beregne $t\beta$. Krypteringsfunktionen er så givet ved

$$e_\beta(m, t) = (t\alpha, m + t\beta)$$

Bob kan ud fra den krypterede version af meddelelsen beregne $at\alpha = t\beta$, finde den inverse i G, og derefter m – formelt bliver dekrypteringsfunktionen

$$d_a(c, d) = d - ac$$

Ud over beskaffenheden af G skal der Bob og Alice selvfølgelig også være enige om hvordan repræsentationen af en meddelelse som et element i G kommer i stand, og hvordan man konverterer den anden vej. For $G = \mathbb{Z}_p^*$ kan dette gøres ved at splitte den binære repræsentation af en meddelelse – f.eks. som ascii tekst – op i 'bidder' af passende størrelse, og derefter fortolke hver 'bid' som et heltal, og vælge det tilsvarende element i G.

3.2 Den Diskrete Logaritme

Hvordan kan det være at definition 3.1 virker? Hvis vi tager notationen for pålydende, dvs. betragter + som almindelig addition af tal, for eksempel i $\mathbb{Z}_p$, er det ingen kunst at bryde koden. Hvis man kender G, α og Bobs offentlige nøgle β, kan man jo bare beregne $a = \beta/\alpha$. Pointen er at denne beregning er 'snyd' idet man bruger gruppeelementet α til at dividere med; man udnytter at G i virkeligheden er et legeme (α har en multiplikativ invers).

Det er altså ikke en hvilken som helst gruppe der kan bruges. Problemet at løse ligningen $\beta = a\alpha$ for et helt tal a, når α og β er gruppeelementer, kaldes det *diskrete logaritmeproblem med basis* α. Betegnelsen logaritme kommer fra grupper med multiplikativ notation, hvor den tilsvarende ligning ville blive skrevet $\beta = \alpha^a$, og løsningen skrives $\log_\alpha \beta = a$.

Et eksempel på en gruppe hvor det diskrete logaritmeproblem er vanskeligt at løse er ${\mathbb{Z}_p}^*$, den multiplikative gruppe i $\mathbb{Z}_p \setminus \{0\}$ for et stort primtal p. Den er cyklisk og har $p-1$ elementer. Vi bruger dette eksempel, og den multiplikative notation, i resten af dette og det næste afsnit.

Man skal lægge mærke til at det på ingen måde er umuligt at løse det diskrete logaritmeproblem. Tværtimod: Man kan nemt lave en algoritme der beregner dem alle sammen fra en ende af:

```
NaivDiskretLogaritme(α, β)
1  i ← 1
2  θ ← α
3  while „true“
4        do
5           if θ = β
6              then return i
7           i ← i + 1
8           θ ← θα
```

Hvis det altså passer at β er en potens af α, terminerer denne algoritme efter $\log_\alpha \beta$ iterationer. Vi behøver altså ikke vide andet om G end at den er cyklisk (og derfor endelig), så kan vi med sikkerhed løse det diskrete logaritmeproblem i tid $\mathrm{O}(n)$, hvor n er ordenen af G.

I betragtning af den hastighed hvormed aritmetiske beregninger kan udføres på computer, betyder dette at vi, for at kunne kryptere noget som helst, skal have fat i grupper med et astronomisk antal elementer. Man kunne tage et tal som $n > 2^{100}$: Hvis man udfører en milliard (cirka 2^{30}) gruppeoperationer per sekund skal man bruge 2^{70} sekunder, det skulle blive omkring 600 milliarder år, på de n operationer. En algoritme som terminerer senest efter 600 milliarder år er ikke af stor praktisk værdi.

Men hvis gruppen er så stor, hvordan kan man så udføre de beregninger som er nødvendige for at bruge krypteringsalgoritmen? Svaret på dette er ret simpelt når man betragter den binære repræsentation af et helt tal. Lad os vende tilbage til eksemplet med ${\mathbb{Z}_p}^*$. Med multiplikativ notation skal man, for at lave den offentlige nøgle, beregne $\beta = \alpha^a$. Hvis p er et primtal af størrelsesorden 2^{100}, og $0 < a < p$, er den binære repræsentation af a omkring 100 bit lang. Hvis vi skriver den binære repræsentation som $a_{100}a_{99}\cdots a_0$ kan man få værdien af a som

$$a = \sum_{i=0}^{100} a_i \cdot 2^i = (\cdots((a_{100} \cdot 2 + a_{99}) \cdot 2 + a_{98}) \cdot 2 + \cdots) \cdot 2 + a_0$$

ved at bruge Horners regel. Samme teknik kan bruges til at beregne α^a:

```
SquareAndMultiply(α, a)
1  θ ← 1_G
2  for i ← bitlængden af a downto 0
3        do
4           θ ← θ^2
5           if a_i ≠ 0
6              then θ ← θ · α
7  return θ
```

SquareAndMultiply itererer over bitlængden af a, altså $\log_2(a)$ iterationer, med højst to gruppeoperationer per iteration – i eksemplet med $p \approx 2^{100}$ altså højst 100 operationer. Det er dette forhold der gør kryptering muligt: Ved at bruge én ekstra bit i repræsentationen af p får Alice og Bob kun to ekstra regneoperationer hver, mens den onde Åge, som prøver at aflure deres kommunikation, skal bruge *dobbelt* så meget regnekraft per ekstra bit. På datalogsprog siger man at arbejdet ved kryptering vokser *linært* med bitlængden af n, mens arbejdet ved at løse den diskrete logaritme i gruppen vokser *eksponentielt* med bitlængden.

3.3 „Rigtige“ Algoritmer

Der findes selvfølgelig mere udspekulerede måder at beregne den diskrete logaritme på end den som er implementeret i NaivDiskretLogaritme. Vi skal se et grundlæggende eksempel herpå, nemlig Shanks' *Baby Steps/Giant Steps* algoritme. Den er ret simpel, men den nedbringer alligevel antallet af beregninger betragteligt. Algoritmen benytter en oplysning mere end NaivDiskretLogaritme, nemlig n, ordenen af den cykliske gruppe.

SHANKS(α, n, β)

```
1  m ← ⌈√n⌉
2  L_1 ← en tom liste med plads til m elementer.
3  for j ← 0 to m − 1
4        do
5           L_{1,j} ← (α^{mj}, j)
6  sortér elementerne i L_1
7  L_2 ← en tom liste med plads til m elementer.
8  for i ← 0 to m − 1
9        do
10          L_{2,i} ← (βα^{−i}, i)
11 sortér elementerne i L_2
12 find (y, j) ∈ L_1 og (y, i) ∈ L_2, et element fra hver liste
       med matchende første koordinater.
13 return log_α β = (mj + i)  mod n
```

Først vil vi se efter om det resultat algoritmen afleverer er korrekt. Hvis det i linje 12 lykkes at finde to elementer $(y, j) \in L_1$ og $(y, i) \in L_2$, så må

$$y = \alpha^{mj} = \beta\alpha^{-i} \quad \Rightarrow \quad \beta = \alpha^{mj+i}$$

som ønsket. Men er vi sikre på at et sådant elementpar findes? Da β er frembragt af α ved vi at $0 \leq \log_\alpha \beta < n$. Da vi har sat $m = \lceil\sqrt{n}\rceil$ må det være muligt at skrive $\log_\alpha \beta = jm + i$ med $0 \leq i, j < m$ — og da algoritmen gennemløber alle i og j mellem 0 og $m - 1$, må den nødvendigvis nå frem til en løsning.

Lad os nu se på hvor meget arbejde algoritmen udfører. Løkken i linje 3-5 beregner et gruppeelement α^{mj} hver gang. Dette kan gøres ved at beregne $\mu = \alpha^m$ én gang, og derefter i hver iteration regne $\alpha^{mj} = \mu^j = \mu \cdot \mu^{j-1}$. Det er altså bare en enkelt gruppeoperation per iteration. Det samme gælder for løkken i linje 8-10, her er det bare α^{-1} som skal ganges på hver gang. Sortering af en liste med m elementer – linje 6 og 11 – tager $O(m \log m)$ med en effektiv sorteringsalgoritme som for eksempel *Quicksort*, forudsat at listen er opbevaret på en ordentlig måde. I linje 12 løbes de to lister igennem for at finde 'to ens', men da listerne på dette tidspunkt er sorteret efter førstekoordinater, tager det kun tid $O(m)$. Ingen af skridtene tager altså længere end $O(m \log m)$.

Vi kan konkludere at det er muligt at beregne den diskrete logaritme i en cyklisk gruppe af orden n med $O(\sqrt{n})$ beregninger, hvilket er mærkbart mindre end de $O(n)$ vi skulle bruge med NAIVDISKRETLOGARITME. I eksemplet hvor $n \approx 2^{100}$ bliver $\sqrt{n} \approx 2^{50}$, og hvis vi igen mener at kunne udføre 2^{30} beregninger per sekund – hvilket i øvrigt nok er lidt optimistisk – kan vi klare 2^{50} beregninger på et par uger. Den asymptotiske køretid på $O(\sqrt{n})$ gemmer på en konstant vi ikke umiddelbart kender, men som vi

nok må erkende er noget større end 1 (i SHANKS udføres der jo i hvert fald $2\sqrt{n}$ gruppeoperationer, plus sorteringerne). Men selv da skulle det være muligt at finde en løsning i god tid inden universet ophører med at eksistere. Den nye øvre grænse er imidlertid ikke nok til at forrykke magtbalancen; Alice og Bob kan stadig få den mængde arbejde, Åge skal udføre for at bryde koden, til at vokse eksponentielt: De skal blot vælge et 200 bit langt primtal i stedet for det gamle på 100 bit, så er Åge tilbage i samme situation som før han opdagede den smartere algoritme.

Der er algoritmer som er mere effektive end SHANKS. Hukommelsesforbruget – SHANKS bruger jo $O(\sqrt{n})$ plads til at opbevare listerne L_1 og L_2 – kan nedbringes forholdsvis let, og nogle algoritmer kan også løse logaritmeproblemet hurtigere hvis det er muligt at faktorere n. Der er i midlertid ikke nogen som har en asymptotisk køretid bedre end $O(\sqrt{p})$, hvor p er den største primfaktor i n. Denne nedre grænse vises i Stinson (2005) for *enhver* algoritme af den generelle (eng. *generic*) type, som er dem der kun benytter gruppeoperationer og -inversioner til at nå frem til løsningen.

En algoritme som er specialiseret med henblik på at løse logaritmen i $\mathbb{Z}_p^*$ er *Index Calculus*-algoritmen. At den er specialiseret betyder at den udnytter at gruppeelementerne er heltal, og derfor ved man mere om dem end at de opfylder gruppeaksiomerne. Ved hjælp af denne algoritme kan det vises at den diskrete logaritme i denne gruppe *ikke* er af eksponentiel kompleksitet, og der findes varianter af algoritmen som viser det samme om andre grupper af typen F^*, multiplikative grupper i $F \setminus \{0\}$ for et legeme F (Stinson, 2005, afsnit 6.2 og Enge, 1999, afsnit 4.4).

Det interessante ved at bruge grupper for elliptiske kurver i kryptografi er at der ikke er nogen tilsvarende specialiserede algoritmer som virker i disse grupper (hvis man går uden om nogle specialtilfælde). Resultatet er størrelsen på gruppen bliver mindre – og det samme gør både størrelsen af nøglerne og den mængde arbejde det kræver at kryptere og dekryptere.

3.4 Kryptosystem over en elliptisk kurve

For at bruge gruppen af en elliptisk kurve over et endeligt legeme, $(E(k), +)$, til at implementere system som det i definition 3.1 skal man løse et par praktiske problemer.

Først og fremmest må vi være sikre på at gruppens orden, betegnet $\#E$, vokser eksponentielt med bitlængden af $|k|$, ordenen af k. Dette garanteres af Hasses Sætning, som vi ikke skal vise, men som siger at

$$|k| + 1 - 2\sqrt{|k|} \leq \#E \leq |k| + 1 + 2\sqrt{|k|},$$

altså at der er omtrent lige så mange punkter på $E(k)$ som der er elementer i k.

For at kunne bruge gruppen til kryptering skal den være cyklisk *og* vi skal bruge en frembringer α og kende gruppens orden. For nogle specialtilfælde af elliptiske kurver, nemlig de såkaldte *supersingulære* og *trace one* kurver, er det nemt at beregne $\#E$.

Uheldigvis er der også udviklet algoritmer som kan løse logaritmeproblemet effektivt i disse grupper. For at opnå sikker kryptering skal man derfor bruger kurver som ikke er en af disse to typer. Man kan så til gengæld ikke beregne $\#E$ direkte, men må benytte en algoritme der „tæller punkterne“ og derved finder $\#E$. Sidste afsnit i (Enge, 1999) handler om denne type algoritmer.

Som nævnt afhænger kompleksiteten af logaritmeproblemet af den største primfaktor p i gruppens orden, så man må finde kurve hvor $n = \#E$ han en stor primfaktor. Finder man sådan en, får man til gengæld også en cyklisk undergruppe af orden p, som kan bruges til kryptering. I en rapport fra 2000 anbefaler *Standards for Efficient Cryptography* en procedure til at udvælge og validere parametrene (kurveparametrene A og B samt parametre som specificerer legemet k). Samtidig anbefaler de at man vælger parametre fra en liste med 'sikre' paramteresæt, som de stiller til rådighed (listen udgør bind to af deres rapport) (SEC, 2000).

I ElGamal krypteringen indkodede Alice sin meddelelse i et eller flere elementer i G. Det er ikke muligt i $E(k)$, for selvom vi kender n, har vi ikke nogen rækkefølge for punkterne i E. Så i stedet for at lade meddelelsen m være et punkt på kurven og beregne $m+t\beta$ som i definition 3.1, kan man lade m være et element i det underliggende legeme k. Så beregnes punktet $(x_0, y_0) = t\beta$ på kurven, og derefter udføres operationen $m \times x_0$ i k, og resultatet sendes som den krypterede meddelelse.

Bob modtager et punkt på kurven og et element $d \in k$, og kan, ved hjælp af sin private nøgle, beregne et nyt punkt på kurven (x_1, y_1), og så beregne $m = d \times x_1^{-1}$ hvor x_1^{-1} er den multiplikativt inverse til x_1 i legemet k.

Hvis man inkorporerer disse justeringer, og skriver punkter på E med store bogstaver som i tidligere afsnit, fås følgende, som er næsten det samme som præsenteres i (Stinson, 2005):

Definition 3.2 (Elliptisk Kurve Kryptosystem)**:** Lad E være en elliptisk kurve defineret over et endeligt legeme k, og lad E have en cyklisk undergruppe af (primisk) orden p med frembringer P.

Bob vælger et tal $a \in \mathbb{Z}_p$ som sin hemmelige nøgle, og beregner derpå $Q = aP$ som sin offentlige nøgle.

Alice skal, for at kryptere en meddelelse $m \in k$ til Bob, vælge et tilfældigt tal (temporær nøgle) $t \in \mathbb{Z}_p$. Så skal hun beregne punktet $(x_0, y_0) = tQ$. Nu kan hun sende den krypterede meddelelse

$$e_Q(m, t) = (tP, m \times x_0)$$

Bob kan nu beregne punktet $(x_1, y_1) = a(tP) = t(aP) = tQ = (x_0, y_0)$ ved hjælp af sin private nøgle, og så finde den inverse til $x_1 = x_0$ i k. I alt bliver

$$d_a(C, d) = d \times x_1^{-1} \quad \text{hvor} \quad (x_1, y_1) = aC$$

Stinson nævner et par små tricks til at gøre denne form for kryptering endnu mere effektiv. For det første kan man, hvis der er en effektiv måde at finde kvadretroden i k, nøjes med at sende x-koordinaten for hver punkt sammen med *fortegnet* for y-koordinaten, så man overfører ét element i k plus én bit i stedet for to elementer i k. Dette kan selvfølgelig gøre overførselshastigheden mindre for store meddelelser. Bogen beskriver dette i detaljer for $k = \mathbb{Z}_p{}^*$.

I en additiv gruppe vil algoritmen SquareAndMultiply blive til DoubleAndAdd. I gruppen for en elliptisk kurve er det meget nemt at finde den inverse (husk at den inverse til et punkt (x_0, y_0) bare er $(x_0, -y_0)$). Derfor kan det altid betale sig at bytte tre eller flere additioner væk for en addition og en subtraktion. Hvis man ser på en binær repræsentation af et tal hvor der optræder tre eller flere 1-bits ved siden af hinanden, kan man skrive det samme tal som binær *med fortegn*, og den ny repræsentation får flere 0-bits en den gamle. Man gennemløber bitmønstret fra højre mod venstre idet man udskifter sekvenser af 1-bits $01\cdots11$ med sekvenser af formen $10\cdots0-1$. For eksempel tallet 55 får 3 bits forskellige fra nul, mod 5 i den almindelige binære repræsentation:

$$\begin{array}{cccccc} 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 & 0 & -1 \\ 1 & 0 & 0 & -1 & 0 & 0 & -1 \end{array} \qquad \begin{array}{r} 32+16+4+2+1=55 \\ 32+16+8-1=55 \\ 64-8-1=55 \end{array}$$

Denne omskrivning udnytter Stinson i en algoritme DoubleAndAdd-OrSubtract, som i gennemsnit sparer ca. 11% af gruppeoperationerne ved beregning af mP for et tal m og et punkt P på kurven.

3.5 sikkerheden

Det ekstra arbejde der kræves til punkttælling, validering af parametre etc., retfærdiggøres som sagt af den mængde arbejde man sparer ved kryptering og dekryptering takket være de kortere nøgler. Tabellen er taget fra en rapport fra (SEC, 2000). Her sammenlignes nøglestørrelser svarende til forskellige sikkerhedsniveauer for symmetrisk kryptering, elliptisk kurve-kryptografi (ECC) og DSA/RSA.

Sikkerhedsniveau (bits)	Symmetrisk metode (nøglestørrelse, bits)	ECC-metode (bitlængde af p)	DSA/RSA (bitlængde, modulus)
56	56	112	512
80	80	160	1024
112	112	224	2048
128	128	256	3072
192	192	384	7680
256	256	512	15360

DSA/RSA er klassikeren indenfor asymmetrisk kryptering. Sikkerheden i RSA bygger på faktorering af heltal – specielt produktet af to store primtal p or q. Dette er ækvivalent med det diskrete logaritmeproblem i gruppen $\mathbb{Z}_{pq}{}^*$. „Symmetrisk metode“ står for en ideel implentation af 'klassisk' kryptografi, hvor krypteringsnøglen skal være fælles for afsender og modtager, men hemmelig for alle andre. Sikkerhedsniveauet, den første kolonne, er et udtryk for hvor mange beregninger modstanderen forventes at måtte udføre hvis koden skal brydes. Sikkerhed på 56 bits betyder 2^{56} beregninger.

Man ser at den symmetriske metode kan holde et sikkerhedsniveau svarende til nøglestørrelsen. ECC-metoden skal hele tiden bruge dobbelt så mange bits, mens nøglestørrelsen for DSA/RSA ser ud til at vokse hurtigere jo højere sikkerhedsniveaue der kræves.

Da udviklingen i hardware indtil nu har fået det ønskede sikkerhedsniveau i praktiske anvendelser til at vokse – det er ikke mange år siden at 80 bits sikkerhed var nok til at lave f.eks. homebanking over internettet – betyder alt dette at fordelen ved elliptiske kurver i forholde til RSA vil blive større og større, med mindre der sker et gennembrud indenfor algoritmer til diskret logaritme — for eksempel at nogen finder en måde at tilpasse *Index Calculus* algoritmen til en arbitrær gruppe.

Litteratur

Sec 1: Elliptic curve cryptography. Technical report, Standards for Efficient Cryptography (SEC), September 2000. URL http://www.secg.org/index.php?action=secg,docs_secg.

Ezra Brown. Magic squares, finite planes, and points of inflection on elliptic curves. *The College Mathematics Journal*, 32(4):260–267, September 2001. URL http://www.math.vt.edu/people/brown/doc/magic.pdf.

Leonard S. Charlap og David P. Robbins. An elementary introduction to elliptic curves, 1988.

Andreas Enge. *Elliptic Curves and Their Applications to Cryptography*. Kluwer Academic, Boston, 1999. ISBN 0792385896.

Thomas W. Hungerford. *Abstract Algebra*. Brooks Cole, Pacific Grove, 1996. ISBN 0030105595.

Douglas R. Stinson. *Cryptography: Theory and Practice*. Chapman & Hall/CRC, 2005.