Punten tellen op Elliptische Krommen over Eindige Lichamen

Auteur: Britt van Veggel Begeleider: Bernd Souvignier

Faculteit der Natuurwetenschappen, Wiskunde en Informatica
Radboud Universiteit Nijmegen
Mei 2018

Inhoudsopgave

1 Introductie **4**

2 Inleiding Elliptische Krommen **6**

2.1 Definitie en Weierstrasseis 6

2.2 Elliptische krommen in het projectieve vlak 12

2.3 Endomorfismen en de Frobenius afbeelding 13

2.4 Torsie punten en de Division Polynomials 18

2.5 De stelling van Hasse en zijn belangrijke gevolgen 23

3 Manieren om punten te tellen **27**

3.1 Legendre symbool voor het tellen van punten 27

3.1.1 Resultaten prestatietest 29

3.2 Punten tellen voor een uitgebreidingslichaam 30

3.3 Baby Step, Giant Step en de stelling van Mestre 32

3.3.1 Baby Step, Giant Step om de orde van een punt te bepalen 32

3.3.2 Baby Step, Giant Step om de orde van een elliptische kromme te bepalen 33

3.3.3 De stelling van Mestre 34

3.3.4 Resultaten prestatietest 38

3.4 Schoof's algoritme 39

3.4.1 Algoritme 39

3.4.2 Kanttekeningen bij algoritme 40

3.4.3 Resultaten prestatietest 42

3.5 Vergelijken prestatietesten 43

4 Bibliografie **45**

5 Appendix **46**

5.1 Bewijs stelling 2.2 . . . 46

5.1.1 Deel 1, stap 1 . . . 46

5.1.2 Deel 1, stap 2 . . . 46

5.1.3 Deel 2 . . . 46

5.2 Optelfunctie . . . 46

5.3 Division Polynomials . . . 47

5.3.1 Division Polynomials . . . 47

5.3.2 De ϕ_n functie . . . 48

5.3.3 De ω_n functie . . . 49

5.3.4 Veelvoud van een punt berekenen . . . 49

5.4 Legendre symbool voor het tellen van punten . . . 50

5.4.1 CPU tijd functie . . . 51

5.5 Punten tellen voor een uitgebreider lichaam . . . 52

5.6 Baby Step, Giant Step en Mestres Methode . . . 52

5.6.1 BSGS voor de orde van een punt . . . 52

5.6.2 Mestre's algoritme . . . 53

5.6.3 Combineren van BSGS en Mestres stelling om de orde van $E(\mathbb{F}_q)$ te bepalen . . . 54

5.6.4 Controle functie . . . 55

5.6.5 CPU tijd functie . . . 55

5.7 Schoof's algoritme . . . 56

5.7.1 algoritme . . . 56

5.7.2 Controle functie . . . 59

5.7.3 CPU tijd functie . . . 60

1 Introductie

Deze scriptie gaat over algoritmen die het aantal punten op een elliptische kromme over een eindig lichaam $\mathbb{F}_q$ telt. Een elliptische kromme is een verzameling van punten (x, y) die voldoen aan de vergelijking $y^2 = x^3 + Ax + B$ waarbij A en B constanten zijn uit een zeker lichaam. Per lichaam (bijvoorbeeld $\mathbb{R}, \mathbb{C}$ of $\mathbb{F}_q$) is de theorie rondom elliptische krommen behoorlijk anders. In deze scriptie bekijken we vooral de theorie met betrekking tot krommen over eindige lichamen.

In hoofdstuk 2 bespreken we alle theorie die voorafgaat aan de algoritmen die we in hoofdstuk 3 bespreken. We beginnen met de definitie van een elliptische kromme en de opbouw naar een definitie voor een optelling op elliptische krommen. Onder deze optelling voldoen elliptische krommen aan de eisen van een abelse groep. Hierbij is het punt in oneindig, ook genoteerd als ∞, het eenheidselement.

We vervolgen met het bespreken van endomorfismen op elliptische krommen. Hierbij gaan we vooral kijken naar het Frobenius endomorfisme Φ_q. Dit endomorfisme is namelijk erg handig in het bepalen van het aantal punten op een elliptische kromme, zoals we zullen zien in stelling 2.9.

In paragraaf 2.4 gaan we kijken naar torsie-punten en division polynomen. Een torsie-punt is een punt op een kromme met een eindige orde. Omdat elliptische krommen over eindige lichamen abelse groepen zijn, kunnen we aan de hand van de orde van een aantal punten op een kromme, de orde van de gehele kromme inschatten. Een division polynoom ψ_l is een recursief polynoom waarvan de nulpunten de x-coordinaten van de l-torsie punten (de punten met orde l) zijn. Anderzijds kan een veelvoud van een punt P uitgedrukt worden in division polynomen en zo wordt de orde van een punt P bepaald.

In de laatste paragraaf van hoofdstuk 2, paragraaf 2.5, bespreken we de twee belangrijkste stellingen voor de algoritmen in hoofdstuk 3: de stelling van Hasse (stelling 2.20) en stelling 2.21 over het spoor van Frobenius.

De stelling van Hasse toont aan dat er een begrensd domein is waarin de orde van een kromme $E(\mathbb{F}_q)$ valt. Dit domein is afhankelijk van de grootte van q. Dit is vooral een gunstig resultaat voor het Baby Step, Giant Step algoritme, dat in paragraaf 3.3 besproken wordt. De begrenzingen die Hasse aantoont, gebruiken de waarde $a := q + 1 - \#E(\mathbb{F}_q)$. Deze waarde wordt ook wel het *spoor van Frobenius* genoemd en speelt een rol in het karakteristieke polynoom $\Phi_q^2 - a\Phi_q + q$ van het Frobenius endomorfisme. In stelling 2.21 tonen we aan dat $\Phi_q^2 - a\Phi_q + q$ het 0-endomorfisme is, ofwel het endomorfisme dat elk punt P afbeeld op ∞. Hierdoor kunnen we de orde van $E(\mathbb{F}_q)$ ook gaan afleiden.

In hoofdstuk 3 bespreken we de algoritmen om de ordes van krommen $E(\mathbb{F}_q)$ voor zekere q te bepalen. Voor elk algoritme behandelen we de theorie en de te nemen stappen van het algoritme. Daarnaast hebben we voor elk algoritme een script geschreven in de programmeertaal MAGMA. Dit is een programmeertaal gericht op algebra, getaltheorie, algebraïsche meetkunde en algebraïsche combinatoriek. De scripts voor deze algoritmen zijn te vinden in de Appendix, hoofdstuk 5. De algoritmen en functies die we uitgewerkt hebben, zijn meestal al bevat MAGMA. Toch schrijven we de scripts opnieuw om zo beter weer te geven

hoe deze algoritmen werken.

We beginnen met het bespreken van de simpelere algoritmen. Zo beginnen we met het symbool van Legendre om het aantal punten op een kromme te bepalen (sectie 3.1). Hierbij wordt voor elk element x uit $\mathbb{F}_q$ gekeken of het de x-coordinaat van een punt op de kromme E is of niet.

Daaraan voegen we een algoritme toe dat de orde van een kromme over een uitbreiding van een lichaam kan bepalen waarbij de orde van de kromme over het kleinere lichaam al bekend is.

Daarna volgen de Arjen Robben en Robin van Persie van de algoritmen: het Baby Step, Giant Step algoritme en het algoritme van Schoof.

Het Baby Step, Giant Step algoritme berekent de orde van een aantal punten op de kromme, en kijkt of het kleinst gemene veelvoud van deze ordes eenduidig binnen het domein van Hasse valt. Dit algoritme breiden we nog uit met de stelling van Mestre (sub-paragraaf 3.3.3), waardoor het Baby Step, Giant Step algoritme een waterdicht algoritme wordt.

Als laatste bekijken we het algoritme van Schoof. Dit algoritme is zeer uitgebreid en combineert veel theorie uit hoofdstuk 2. Vooral de theorie van de stelling van Hasse (stelling 2.20), het spoor van Frobenius (stelling 2.21), Division Polynomials (sectie 2.4) en de Chinese Reststelling spelen een grote rol.

Tot slot bespreken we de prestaties van deze algoritmen op het gebied van CPU tijd. Aan het eind van elke paragraaf staat een sub-paragraaf met de gemiddelde resultaten en de outliers. In de allerlaatste paragraaf van hoofdstuk 3 vergelijken we de prestaties van de drie belangrijkste algoritmen met elkaar en concluderen voor welke elliptische krommes welk algoritme het meest geschikt is.

2 Inleiding Elliptische Krommen

Als we spreken over het lichaam $\mathbb{F}_q$, dan is q een macht van een priemgetal p.

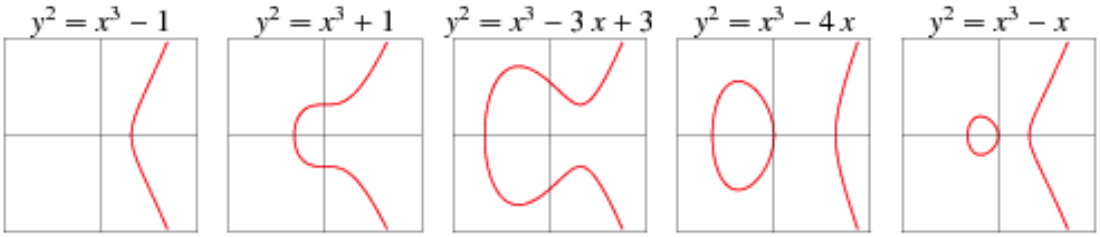
2.1 Definitie en Weierstrasseis

Er zijn meerdere manieren om elliptische krommen te benaderen. Hierdoor zijn er meerdere definities van een elliptische kromme. In paragraaf 2.2 bespreken we de benadering in het projectieve vlak. Allereerst beginnen we met een praktische manier. De definitie die we daarvoor zullen hanteren is als volgt:

Definitie 2.1. Zij $\mathbb{F}$ een lichaam. Een *elliptische kromme gedefinieerd over* $\mathbb{F}$ is een verzameling E van punten $(x, y) \in \mathbb{F}^2$ die voldoen aan de vergelijking $y^2 = x^3 + Ax + B$ waarbij A en B constanten in $\mathbb{F}$ zijn.

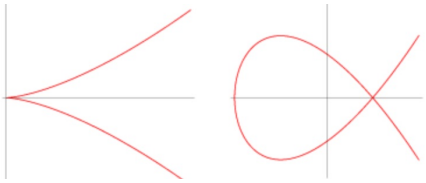
Als E gedefinieerd is over $\mathbb{F}$ noteren we dit als $E(\mathbb{F})$.

Deze definitie is niet meteen makkelijk te visualiseren. In $\mathbb{R}^2$ kan een elliptische kromme er als volgt uit zien:



Voor $\mathbb{C}$ en eindige lichamen is de visualisatie ingewikkelder.

Waar we naartoe willen, is om de optelling op elliptische kromme zo te definiëren dat er een abelse groepsstructuur ontstaat voor $(E, +)$. Hiervoor moeten de kromme en de optelling aan een aantal eisen voldoen. Een kromme over $\mathbb{R}$ moet, bijvoorbeeld, glad zijn om de optelling van een punt met zichzelf mogelijk te laten zijn. Krommen die niet glad zijn, bevatten punten waarin de kromme niet continu differentieerbaar is en zien er als volgt uit:



Definitie 2.2. Een *niet-singuliere kromme* over een van de lichamen $\mathbb{R}$ of $\mathbb{C}$ is een elliptische kromme waarbij alle punten continu differentieerbaar zijn.

Stelling 2.1. *Een kromme is niet-singulier dan en slechts dan als hij aan* $4A^3 + 27B^2 \neq 0$ *voldoet.*

Bewijs. Een elliptische kromme is in de meeste punten continu differentieerbaar. De functie is namelijk te herschrijven tot de functie $f(x, y) = y^2 - x^3 - Ax - B = 0$. Hiervan zijn de partiële afgeleiden $\frac{\partial}{\partial y}2y$ en $\frac{\partial}{\partial x} = -3x^2 - A$.

De enige punten die problemen kunnen opleveren, zijn de punten waar $y = 0$, ofwel, de nulpunten. Deze punten zijn niet continu differentieerbaar als ze een hogere multipliciteit hebben.

Om te kijken wanneer een elliptische kromme een dubbel nulpunt heeft, herschrijven we de algemene derdegraads polynoom:

$$(x - x_1)(x - x_2)(x - x_3) = x^3 - (x_1 + x_2 + x_3)x^2 + (x_1x_2 + x_1x_3 + x_2x_3)x - x_1x_2x_3$$

Als $x_1 = x_3$, dan volgt de vergelijking $x^3 - (2x_1 + x_2)x^2 + (x_1^2 + 2x_1x_2)x - x_1^2x_2$. Dus voor een dubbel nulpunt gelden de volgende vergelijkingen:

$$\begin{cases} 2x_1 + x_2 = 0 & \text{desda } 2x_1 = -x_2 \\ x_1^2 + 2x_1x_2 = A & \text{desda } A = -3x_1^2 \\ -x_1^2x_2 = B & \text{desda } B = 2x_1^3 \end{cases}$$

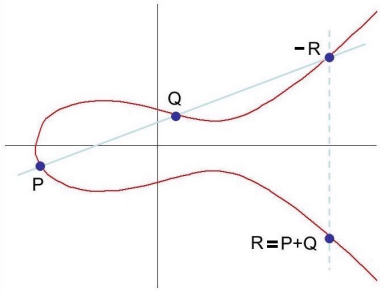
Dit is equivalent met

$$4A^3 + 27B^2 = 4(-3x_1^2)^3 + 27(2x_1^3)^2 = 4 \cdot -27x_1^6 + 27 \cdot 4x_1^6 = 0$$

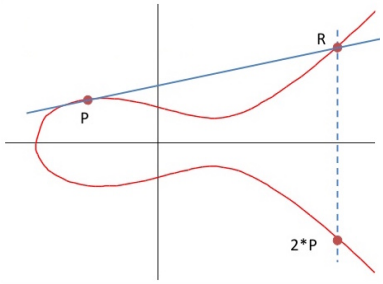
En dus volgt, als een kromme singulier is, is dat equivalent met de eis $4A^3 + 27B^2 = 0$. Dus een kromme voldoet aan $4A^3 + 27B^2 \neq 0$ dan en slechts dan als hij niet-singulier is. □

Het idee van de optelling voor twee punten met $x_1 \neq x_2$ is dat je door twee verschillende punten die niet elkaars spiegelpunt zijn, een lijn trekt, het derde snijpunt van deze lijn met de kromme vindt en deze spiegelt in de x-as. (In sommige gevallen is een van de twee punten het derde snijpunt. Dat levert verder geen problemen op, dus dit terzijde.) Hierdoor geldt dat alle punten die op een lijn liggen samen opgeteld altijd het punt in oneindig opleveren.

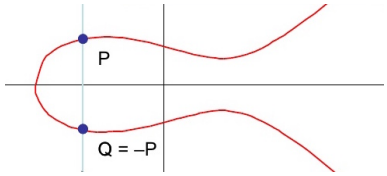
In $\mathbb{R}^2$ ziet dit er als volgt uit:



Als je een punt bij zichzelf optelt, gebruik je de richtingscoëfficiënt van dat punt op de elliptische kromme. Je kan de formule omschrijven naar: $f(x,y) = y^2 - x^3 - Ax - B = 0$. Dan zijn de partiele afgeleiden $\frac{\partial}{\partial y} f(x,y) = 2y \cdot y' = 0$ en $\frac{\partial}{\partial x} f(x,y) = -3x^2 - A = 0$. Als je deze aan elkaar gelijk stelt volgt $y'(x) = \frac{3x^2+A}{2y}$. Dit ziet er zo uit:



Voor een punt $P = (x,y)$ noteren we het punt $(x,-y)$ als $-P$. Het punt $-P$ ligt dus gespiegeld in de x-as ten opzichte van P. De lijn door de punten P en $-P$ staat loodrecht op de x-as. Deze heeft verder geen snijpunt met E. Het punt $P + -P$ definiëren we als (∞, ∞) maar uit gemak noteren we dit punt als ∞. Dit punt noemen we het *punt in oneindig*. De conventie rondom dit punt is dat het zowel bovenaan en onderaan de x-as ligt. Dus $\infty = (\infty, \infty) = (\infty, -\infty)$. Dit zal allemaal duidelijker worden in paragraaf 2.2 over het projectieve vlak. In $\mathbb{R}^2$ ziet het de optelling $P + -P$ er zo uit:



De volgende stelling bewijst dat elke lijn de elliptische kromme E in drie punten snijdt. Deze stelling zorgt ervoor dat de optelling zoals gegeven in definitie 2.3 (dat daarna volgt) wegedefinieerd is.

Stelling 2.2. *Zij $P = (x_1, y_1)$ en $Q = (x_2, y_2)$ een tweetal punten op de kromme E over een lichaam $\mathbb{F}$ met $P \neq \pm Q$ en $y_1 \neq 0$. Zij verder $m_1 = \dfrac{y_2 - y_1}{x_2 - x_1}$ en $m_2 = \dfrac{3x_1^2 + A}{2y_1}$.*

Dan heeft de lijn die P en Q snijdt, drie snijpunten met E. Dat wil zeggen, de vergelijking $x^3 + Ax + B = \big(m_1(x - x_1) + y_1\big)^2$ heeft drie oplossingen.

Verder heeft de raaklijn $l : y = m_2(x - x_1) + y_1$ twee snijpunten met E.

Bewijs. We herschrijven de eerste vergelijking tot $N(x) = x^3 + Ax + B - (m_1(x - x_1) + y_1)^2$. Dit is een vergelijking van graad 3, omdat de eerste term van graad 3 is en alle andere termen geen hogere graad dan 2 behalen. We weten de eerste twee nulpunten van $N(x)$, dus kunnen we de derde afleiden. Daarnaast geldt dat als een derdegraads polynoom twee nulpunten heeft in een lichaam $\mathbb{F}$, dat het derde noodzakelijk ook bevat in $\mathbb{F}$ is. We noteren het derde nulpunt als x_3.

We willen de vergelijking

$$(x - x_1)(x - x_2)(x - x_3) = x^3 + Ax + B - (m_1(x - x_1) + y_1)^2$$

opolossen. We schrijven eerst het linkerlid uit tot:

$$x^3 - (x_1 + x_2 + x_3)x^2 + (x_1x_2 + x_1x_3 + x_2x_3)x - x_1x_2x_3$$

Dan is het rechterlid gelijk aan:

$$x^3 - m_1^2x^2 + (A - 2m_1y_1 + 2m_1^2x_1)x + B + m_1^2x_1^2 + 2m_1y_1x_1 - y_1^2$$

en dus willen we dat er geldt $-m_1^2 = -(x_1+x_2+x_3)$ en daarom volgt $x_3 = m_1^2 - x_1 - x_2 = (\frac{y_2 - y_1}{x_2 - x_1})^2 - x_1 - x_2$.

Nu willen we nog aantonen dat dit daadwerkelijk een punt op de kromme is. We kunnen de vergelijking $y^2 = x^3 + Ax + B$ omschrijven naar de functie $f(x, y) = y^2 - x^3 - Ax - B$, en dan zijn de punten op de elliptische kromme de nulpunten van f. We vullen hiervoor $x_3 = m_1^2 - x_1 - x_2$ en $y_3 \coloneqq m_1(x_1 - x_3) - y_1$ in in f.

Nu willen we aantonen dat $f(x_3, y_3) = 0$. Omdat dit veel en nauwkeurig rekenwerk is, hebben we het programma MAGMA gebruikt. We hebben onderstaande berekeningen gedaan over het lichaam `R<x_1,x_2,y_1,y_2,A,B>:=FunctionField(Rationals(),6)`. In de appendix, sectie 5.1.1, staat welke code precies moet worden ingevuld in MAGMA.

In eerste instantie, wordt $f(x_3, y)$ berekend met y_1 en y_2 als losse variabelen. Hieruit volgt een grote breuk met zes verschillende variabelen en meer dan 20 termen in de teller (uitkomst bijgevoegd onder sectie 5.1.1). Om te bewijzen dat $f(x_3, y)$ gelijk is aan 0, hoeven we enkel aan te tonen dat de teller gelijk is aan 0. Dus in de teller vervangen we nu alle y_1^2 met $x_1^3 + Ax_1 + B$ en alle y_2^2 met $x_2^3 + Ax_2 + B$. Wat er dus ingevuld dient te worden in MAGMA, staat in sectie 5.1.2. Dit geeft `0` als uitkomst.

Alle termen vallen tegen elkaar weg, en we kunnen aannemen dat het punt $x_3 = m_1^2 - x_1 - x_2$ het y-coordinaat $m_1(x_1 - x_3) - y_1$ oplevert. Zodoende heeft de lijn door P en Q drie snijpunten met de elliptische kromme.

Voor de raaklijn $l : y = m_2(x - x_1) + y_1$ met $m_2 \coloneqq \dfrac{3x_1^2 + A}{2y_1}$ aan het punt P op de kromme, willen we aantonen dat deze raaklijn de kromme ook snijd in het punt $(x_3, y_3) \coloneqq (m_2^2 - 2x_1, m_2(m_2^2 - 3x_1) + y_1)$. Daarom willen we laten zien dat:

$$y_3^2 = x_3^3 + Ax_3 + B$$

Deze berekeningen doen we ook aan de hand van MAGMA. Hiermee willen we aantonen dat $g(x, y) = y^2 - x^3 + Ax + B$, met $x = x_3$ en $y = y_3$ gegeven zoals hierboven, gelijk is aan 0. We vullen de code in sectie 5.1.3 in. Dit geeft de vergelijking `y_1^2-A*x_1-B-x_1^3` uit. Dat is gelijk aan 0 per aanname. En dus is het punt (x_3, y_3) een tweede nulpunt van de vergelijking $(m_2(x - x_1) + y_1)^2 = x^3 + Ax + B$. □

Definitie 2.3. Zij E een elliptische kromme, gedefinieerd door $y^2 = x^3 + Ax + B$. Zij $P_1 = (x_1, y_1)$ en $P_2 = (x_2, y_2)$ punten op deze kromme, beiden ongelijk aan het punt in oneindig. Dan word de optelling $P_1 + P_2 = P_3 = (x_3, y_3)$ op E gedefinieerd door de volgende regels:

i Als $x_1 \neq x_2$ dan

$$x_3 = m_1^2 - x_1 - x_2 \text{ en } y_3 = m_1(x_1 - x_3) - y_1 \text{ waar } m_1 = \frac{y_2 - y_1}{x_2 - x_1}$$

ii Als $x_1 = x_2$, maar $y_1 \neq y_2$ dan geldt $P_1 + P_2 = \infty$.

iii Als $P_1 = P_2$ en $y_1 \neq 0$ dan

$$x_3 = m_2^2 - 2x_1 \text{ en } y_3 = m_2(x_1 - x_3) - y_1 \text{ waar } m_2 = \frac{3x_1^2 + A}{2y_1}$$

iv Als $P_1 = P_2$ en $y_1 = 0$ dan geldt $P_1 + P_2 = \infty$.

Tot slot geld $P + \infty = P$ voor alle punten P op E.

Dit alles samen geeft ons de volgende notatie voor de verzameling van een elliptische kromme over een lichaam $\mathbb{F}$:

$$E(\mathbb{F}) = \{\infty\} \cup \{(x, y) \in \mathbb{F} \times \mathbb{F} \mid y^2 = x^3 + Ax + B\} \quad .$$

Een MAGMA-script voor de optelling kan gevonden worden in de appendix onder sectie 5.2. Het opvallendste dat hier gebeurt, is de toevoeging van de Division Polynomials. Deze bespreken we uitgebreid in sectie 2.4. We zullen daar uitleggen waarom we deze hebben toegevoegd.

Voorbeeld 2.1. Om deze definitie iets meer vorm te geven kijken we naar een elliptische kromme over $\mathbb{F}_5$. Stel $E : y^2 = x^3 + 2x + 4$. Dan kun je hierop de punten $(0, 2)$ en $(4, 1)$ vinden. Dan volgt

$$(0,2)+(4,1) = \Big(\Big(\tfrac{1-2}{4-0}\Big)^2 - 0 - 4, \quad \tfrac{1-2}{4-0}(0 - \Big(\tfrac{1-2}{4-0}\Big)^2 + 0 + 4) - 2\Big)$$

In $\mathbb{F}_5$ is de inverse van 4 gelijk aan 4, dus dit is gelijk aan $((-1\cdot 4)^2+1, -4(-4^2+4)-2) = (2,1)$. Zodoende hebben we ons punt gevonden.

Als we nu $(0,2)+(0,2)$ willen berekenen, doen we dat als volgt:

$$(0,2)+(0,2) = \Big(\Big(\frac{3\cdot 0+2}{2\cdot 2}\Big)^2 - 2\cdot 0, \quad \frac{3\cdot 0+2}{2\cdot 2}(0 - \Big(\frac{3\cdot 0+2}{2\cdot 2}\Big)^2 - 2\cdot 0) - 2\Big)$$

$\equiv ((2\cdot 4)^2 - 2\cdot 0, \quad 2\cdot 4(0-(2\cdot 4)^2 + 2\cdot 0) - 2) \equiv (4,1)$.

Stelling 2.3. *Gegeven de optelling gedefinieerd zoals in definitie 2.3, volgt dat $(E,+)$ een additieve abelse groep is met ∞ als het nulelement.*

Bewijs. We willen hiervoor dat de volgende vier eigenschappen gelden:

a De optelling is commutatief, dat wil zeggen $P_1+P_2 = P_2+P_1$ geldt voor alle $P_1, P_2 \in E$

b ∞ is het eenheidselement, dat wil zeggen $P+\infty = P$ geldt voor alle $P \in E$

c Voor elk punt P op E bestaat er een inverse, dat wil zeggen er bestaat een punt P' op E waarvoor geldt $P+P' = \infty$. Dit punt zal $-P$ zijn

d De optelling is associatief, dat wil zeggen $(P_1+P_2)+P_3 = P_1+(P_2+P_3)$ geldt voor alle $P_1, P_2, P_3 \in E$

Bewijs van a. Voor de gevallen ii, iii en iv uit definitie 2.3 is dit triviaal. Rest enkel i te bewijzen. Het x-coordinaat van P_1+P_2 is $(\frac{y_2-y_1}{x_2-x_1})^2 - x_1 - x_2 = (\frac{y_1-y_2}{x_1-x_2})^2 - x_2 - x_1$ en dit is dan weer gelijk aan het x-coordinaat van P_2+P_1. Uit eenzelfde x-coordinaat, volgt een overeenkomend y-coordinaat. Dus de optelling is commutatief.

Bewijs van b. Per definitie geldt $P+\infty = P$ voor alle $P \in E$. Dus ook b geldt.

Bewijs van c. Eigenschap c volgt uit ii van definitie 5.2. Voor het punt $P = (x,y)$ is de inverse $P' = (x,-y)$, want $P+P' = (x,y)+(x,-y) = \infty$. Zoals we hierboven hebben aangetoond is ∞ het eenheidselement. Dus er bestaat een inverse.

Bewijs van d. Dit bewijs volgt logischerwijs door de formules uit definitie 5.2 in te vullen en uit te schrijven. Het wordt niet uitgewerkt in deze scriptie omdat het een vrij technisch en uitgebreid bewijs is waar meer dan 24 gevallen onderscheiden moeten worden (bijvoorbeeld gevallen zoals $P_1 = P_2+P_3$ en $P_2+P_3 = \infty$ etc.).

Een alternatief bewijs voor de associativiteit is gegeven in sectie 2.4 in het boek van Washington [8]. Hierbij wordt de associativiteit bewezen via het projectieve vlak. □

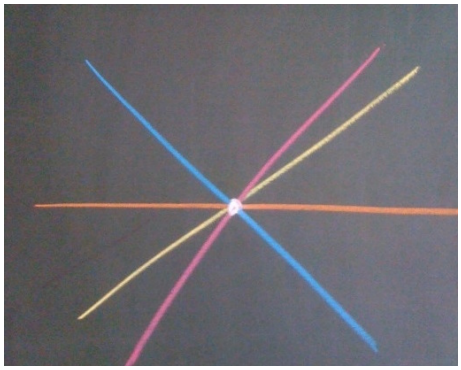
2.2 Elliptische krommen in het projectieve vlak

De vergelijking $y^2 = x^3 + Ax + B$ die we in deze scriptie hanteren heet de *Weierstrass formule*. Dit is een toegankelijke manier op naar elliptische krommen te kijken. De abstractere methode om elliptische krommen te benaderen is via het projectieve vlak.

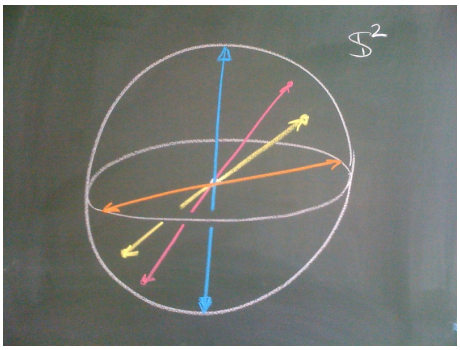
Definitie 2.4. De *projectieve ruimte van dimensie n (over een lichaam K)* is de verzameling van alle $(n+1)$-tupels $(x_0, \ldots, x_n)$ waarbij minstens een x_i niet nul is, modulo de equivalentierelatie $\sim_{\mathbb{P}}$. Deze equivalentierelatie is gedefinieert als $(x_0, \ldots, x_n) \sim_{\mathbb{P}} (y_0, \ldots, y_n) \iff \exists \lambda \in K$ zodanig dat $x_i = \lambda y_i \forall i$. Deze verzameling wordt genoteerd als $\mathbb{P}^n$ of $\mathbb{P}^n(K)$.

Opmerking. Alle equivalente tupels zijn veelvouden van elkaar. Daarom zijn alle tupels met $z \neq 0$ bevat in de equivalentieklasse $[(\frac{x}{z}, \frac{y}{z}, 1)]_{\mathbb{P}}$. Zodoende blijven alleen de eerste twee componenten over om in te variëren. Op de tupels $(x, y, 0)$ komen we na de definitie van een homogene polynoom terug. Dit zijn de zogeheten punten bij oneindig.

Zo'n verzameling heeft als punten alle lijnen in K^{n+1} die ontstaan vanuit de oorsprong én het nulpunt (het punt met $z = 0$). Alle punten die op dezelfde lijn liggen behoren namelijk tot een klasse. In $\mathbb{P}^1(\mathbb{R})$ zien elementen er zo uit:



We kunnen ook om de oorsprong een bol maken en kijken naar de punten waar de lijnen deze bol snijden. Dan kunnen we de equivalentierelatie $\sim$ definieren, waarbij punt A en punt B equivalent zijn als ze op dezelfde lijn liggen (zoals bijvoorbeeld de twee snijdpunten aan de uiteinden van de gele lijn in onderstaand plaatje). Dan zijn de snijpunten van de lijnen met de bol modulo de equivalentie relatie $\sim$ isomorf met de punten in $\mathbb{P}^n$. In $\mathbb{S}^2/_{\sim} \cong \mathbb{P}^2$ ziet dat er als volgt uit:



Dit is het geval in $\mathbb{P}^2(\mathbb{R})$. Voor eindige lichamen werkt dit anders. Hierbij heeft elke lijn dezelfde hoeveelheid punten en gaan door elk punt dezelfde hoeveelheid lijnen.

Om de verzameling $\mathbb{P}^n(K)$ te behouden, willen we dat voor een functie $f : \mathbb{P}^n(K) \to \mathbb{P}^n(K)$ met nulpunt (x, y, z) het punt $(\lambda x, \lambda y, \lambda z)$ ook een nulpunt is. Functies binnen het projectieve vlak zien er dus als volgt uit:

Definitie 2.5. Een polynoom is *homogeen van graad* n als de som van de machten van elke term gelijk zijn aan n. Voor een term van de vorm $ax^iy^jz^k$ met $a \in K$ geldt dus $i + j + k = n$.

De projectieve variant van een elliptische kromme $E : y^2 = x^3 + Ax + B$ is de homogene verzameling $y^2z = x^3 + Axz^2 + Bz^3$ van graad 3. Hieruit volgt dat uit $z = 0$ volgt $0 = x^3$, en dus $x = 0$. Omdat de tupel niet nul is, volgt $y \neq 0$. Dus volgt dat alle tupels met $z = 0$ en $x = 0$ equivalent zijn met $(0, 1, 0)$ en dus is het punt in oneindig equivalent met de tupel $(0, 1, 0)$. We stelden al eerder dat het punt in oneindig zowel boven- als onderaan de kromme ligt. Dit komt overeen met het feit $(0, 1, 0) \sim_{\mathbb{P}} (0, -1, 0)$. Hieruit mogen we concluderen dat een elliptische kromme over K een injectie naar $\mathbb{P}^2(K)$ heeft.

In $\mathbb{F}_2$ is de Weierstrass formule niet bruikbaar. Voor dit geval wordt verwezen naar hoofdstuk V van Silverman's boek [7] en paragraaf 2.8 in het boek van Washington [8].

2.3 Endomorfismen en de Frobenius afbeelding

Zij K een lichaam, dan noteren we de algebraische afsluiting van een lichaam K als $\overline{K}$.

Definitie 2.6. Een *endomorfisme op* E is een homomorfisme $\alpha\colon E(\overline{K}) \to E(\overline{K})$ gegeven door rationale functies van x en y.

In andere woorden, een endomorfisme α voldoet aan de eis dat voor alle punten P_1 en P_2 geldt $\alpha(P_1+P_2) = \alpha(P_1) + \alpha(P_2)$ en er zijn rationele functies $R_1(x, y)$ en $R_2(x, y)$ zodat $\alpha(x, y) = (R_1(x, y), R_2(x, y))$ voor

alle $(x, y) \in E(\overline{K})$. Ook geldt $\alpha(\infty) = \infty$. Het endomorfisme dat elk punt op ∞ afbeeldt, noteren we als [0] of 0.

Verder bekijken we endomorfismen altijd voor elliptische krommen over een algebraisch afgesloten lichaam omdat je zo de inversen kan uitdrukken in n-de-machtswortels die wel bestaan in de algebraische afsluiting. Daarnaast is het essentieel voor het *Frobenius endomorfisme*, omdat dit endomorfisme anders de identiteitsafbeelding is. Maar wat het Frobenius endomorfisme is en waarom bovenstaand geldt, leggen we later in het hoofdstuk uit.

Voorbeeld 2.2. Neem $\alpha(P) = 2P$ op de standaardkromme $E : y^2 = x^3 + Ax + B$. Dan volgt

$$R_1(x, y) = \Big(\frac{3x^2 + A}{2y}\Big)^2 - 2x \text{ en}$$

$$R_2(x, y) = \Big(\frac{3x^2 + A}{2y}\Big)(3x - \Big(\frac{3x^2 + A}{2y}\Big)^2) - y$$

α is dus een endomorfisme omdat R_1 en R_2 rationale functies zijn.

Stelling 2.4. *Elk endomorfisme α is te herschrijven tot de vorm $\alpha(x, y) = (r_1(x), r_2(x)y)$ met r_1 en r_2 rationale functies van x.*

Bewijs. Zij $\alpha(x, y) = (R_1(x, y), R_2(x, y))$ met R_1 en R_2 rationale functies. Dan kunnen we beide functies R_1 en R_2 herschrijven tot iets van de vorm

$$R_i(x, y) = \frac{p_1(x) + p_2(x)y}{p_3(x) + p_4(x)y}$$

want elke y^{2i} kun je vervangen door $y^{2i} = (x^3 + Ax + B)^i$. We kunnen dan de noemer vermenigvuldigen met $p_3(x) - p_4(x)y$ en y^2 vervangen door $x^3 + Ax + B$. Hierdoor stuiten we op iets van de vorm

$$R(x, y) = \frac{q_1(x) + q_2(x)y}{q_3(x)}$$

Omdat α een homomorfisme is volgt $\alpha(x, -y) = \alpha(-(x, y)) = -\alpha(x, y)$ en dus willen we dat geldt $R_1(x, -y) = R_1(x, y)$ en $R_2(x, -y) = -R_2(x, y)$. Daarom geldt $q_2(x) = 0$ en dus is R_1 van de vorm $\frac{q_1(x)}{q_3(x)}$ te zijn, en R_2 van de vorm $\frac{q_2(x)y}{q_3(x)}$. □

Definitie 2.7. Definieer de *graad* van een niet-nul endomorfisme $\alpha = (r_1(x, y), r_2(x, y))$ met $r_1(x, y) = \frac{p(x)}{q(x)}$ als $max\{\deg p(x), \deg q(x)\}$.

Definitie 2.8. Een endomorfisme α is *separabel* als de afgeleide $r_1'(x)$ niet nul is.

Definitie 2.9. Zij E een elliptische kromme gedefinieerd over $\mathbb{F}_q$. De *Frobenius afbeelding* op E is de afbeelding

$$\Phi_q(x, y) = (x^q, y^q)$$

Stelling 2.5. *Het Frobenius endomorfisme Φ_q is een niet-separabel endomorfisme van graad q op de kromme E gedefinieerd over $\mathbb{F}_q$.*

Bewijs. Dat de graad q is, is duidelijk. Om aan te tonen dat Φ_q een endomorfisme is, moeten we bewijzen dat $\Phi_q\colon E(\overline{\mathbb{F}_q}) \to E(\overline{\mathbb{F}_q})$ een homomorfisme is. De functies x^q en y^q zijn namelijk duidelijk rationaal. Neem de punten (x_1, y_1) en (x_2, y_2) uit $E(\overline{\mathbb{F}_q})$ die samen optellen tot $(x_3, y_3) \in E(\overline{\mathbb{F}_q})$. We gaan de 4 verschillende gevallen onderscheiden.

i Als $x_1 \neq x_2$, dan is $(x_3, y_3) = (m^2 - x_1 - x_2, m(x_1 - x_3) - y_1)$ met $m = \frac{y_2 - y_1}{x_2 - x_1}$. Dan volgt $(x_3^q, y_3^q) = (m'^2 - x_1^q - x_2^q, m'(x_1^q - x_3^q)) - y_1^q)$ met $m' = \frac{y_2^q - y_1^q}{x_2^q - x_1^q}$. En dit is gelijk aan $(x_1^q, y_1^q) + (x_2^q, y_2^q) = \Phi_q(x_1, y_1) + \Phi_q(x_2, y_2)$.

ii Als $x_1 = x_2$ en $y_1 \neq y_2$, dan $\Phi_q(x_3, y_3) = \Phi_q(\infty) = \infty$. Uit $x_1 = x_2$ volgt dat $x_1^q = x_2^q$ en $y_1 \neq y_2 \Leftrightarrow y_1 - y_2 \neq 0 \Leftrightarrow y_1^q - y_2^q \neq 0 \Leftrightarrow y_1^q \neq y_2^q$. En dus volgt dat $\Phi_q(x_1, y_1) + \Phi_q(x_2, y_2) = \infty$.

iii Als $x_1 = x_2$ en $y_1 = y_2 \neq 0$ volgt $x_1^q = x_2^q$ en $y_1^q = y_2^q \neq 0$ en dus

$$\Phi_q((x_1, y_1) + (x_2, y_2)) = \Phi_q(x_3, y_3)$$

$$= \Phi_q(\Big(\frac{3x_1^2 + A}{2y_1}\Big)^2 - 2x_1, \frac{3x_1^2 + A}{2y_1}(x_3 - x_1) - y_1)$$

$$= (\Big(\Big(\frac{3x_1^2 + A}{2y_1}\Big)^2\Big)^q - 2^q x_1^q, \Big(\frac{3x_1^2 + A}{2y_1}\Big)^q (x_3^q - x_1^q) - y_1^q)$$

$$= (\Big(\frac{3^q (x_1^q)^2 + A^q}{2^q y_1^q}\Big)^2 - 2^q x_1^q, \Big(\frac{3^q (x_1^q)^2 + A^q}{2^q y_1^q}\Big)^q (x_3^q - x_1^q) - y_1^q)$$

Er geldt $2^q = 2$, $3^q = 3$ en $A^q = A$, dus

$$= (\Big(\frac{3(x_1^q)^2 + A}{2y_1^q}\Big)^2 - 2x_1^q, \Big(\frac{3(x_1^q)^2 + A}{2y_1^q}\Big)^q (x_3^q - x_1^q) - y_1^q)$$

$$= \Phi_q(x_1, y_1) + \Phi_q(x_2, y_2)$$

iv Als $x_1 = x_2$ en $y_1 = y_2 = 0$ volgt $x_1^q = x_2^q$ en $y_1^q = y_2^q = 0$, dus $\Phi_q((x_1, y_1) + (x_2, y_2)) = \Phi_q(\infty) = \infty$ en volgt $\Phi_q(x_1, y_1) + \Phi_q(x_2, y_2) = (x_1^q, y_1^q) + (x_2^q, y_2^q) = \infty$.

v $\Phi_q((x, y) + \infty) = \Phi_q(x, y) = \Phi_q(x, y) + \infty = \Phi_q(x, y) + \Phi_q(\infty)$.

En dus is Φ_q een homomorfisme gegeven door een rationale functie, en dus een endomorfisme.

Tot slot kunnen we concluderen dat Φ_q niet-separabel is omdat $r_1(x) = x^q$ en dus $r_1'(x) = qx^{q-1} \equiv 0$. $\square$

Stelling 2.6. *Zij $\alpha \neq 0$ een separabel endomorfisme over een elliptische kromme E. Dan geldt $deg(\alpha) = \#Ker(\alpha)$ waarbij $Ker(\alpha)$ de kern is van het homomorfisme van $\alpha\colon E(\overline{K}) \to E(\overline{K})$.*

Bewijs. We schrijven $\alpha(x, y) = (r_1(x), yr_2(x))$ waarbij $r_1(x) = \frac{p(x)}{q(x)}$. Allereerst nemen we aan dat α separabel is. Dus $r_1' \neq 0$, dus weten we dat $pq' - p'q \neq 0$.

Definieer nu $\Gamma \coloneqq \{x \in \overline{K} \mid ((pq' - p'q)q)(x) = 0\} \subsetneqq \overline{K}$. We zullen later aantonen dat $\Gamma \sim Ker(\alpha)$. Deze Γ is een eindige verzameling, omdat $pq' - p'q$ niet het nulpolynoom is. Er zijn dus maar eindig veel punten x met $(pq' - p'q)(x) = 0$.

Neem een $(a, b) \in E(\overline{K})$ die voldoet aan:

- $a \neq 0$, $b \neq 0$ en $(a, b) \neq \infty$
- $deg(p(x) - aq(x)) = max\{deg(p), deg(q)\} = deg(\alpha)$
- $a \notin r_1(\Gamma)$
- $(a, b) \in \alpha(E(\overline{K}))$

Omdat Γ eindig is, volgt dat $\alpha(\Gamma)$ eindig is. De rationale afbeelding $r_1(x)$ neemt oneindig veel verschillende waarden aan als we de x laten lopen over $\overline{K}$. Omdat voor elke x er een punt $(x, y) \in E(\overline{K})$ bestaat, volgt dat $\alpha(E(\overline{K}))$ een eindige verzameling is. En dus bestaat er zo een (a, b).

We willen nu aantonen dat er precies $deg(\alpha)$ punten $(x_1, y_1) \in E(\overline{K})$ zijn waarvoor geldt $\alpha(x_1, y_1) = (a, b)$. Als dit waar is, geldt namelijk dat elke klasse $(x_1, y_1) \cdot Ker(\alpha)$ precies $deg(\alpha)$ punten heeft, en dat $Ker(\alpha)$ dus $deg(\alpha)$ punten heeft. (Voor achterliggende theorie zie Appendix B in het boek van Washington [8]).

Omdat $(a, b) \in \alpha(E(\overline{K}))$, volgt dat er een punt (x_1, y_1) te vinden is zodat

$$\frac{p(x_1)}{q(x_1)} = a \quad \text{en} \quad y_1 r_2(x_1) = b.$$

En uit $(a, b) \neq 0$ volgt dat $q(x_1) \neq 0$ en dat $r_2(x_1)$ bestaat. Uit $y_1 r_2(x_1) = b$ kunnen we afleiden dat $y_1 = \frac{b}{r_2(x_1)}$, en dat y_1 dus een functie van x_1 is. Daarom hoeven dus enkel alle x_1 geteld te worden die op a afbeelden. Verder weten we dat $p(x) - aq(x)$ precies $deg(\alpha)$ nulpunten heeft, waarbij de punten met hogere multipliciteit meegeteld zijn.

Dan rest te bewijzen dat er geen dubbele nulpunten zijn. Stel dat x_0 een hogere multipliciteit heeft. Dan geldt $p(x_0) - aq(x_0) = 0$ en $p'(x_0) - aq'(x_0) = 0$. Dan $p(x_0) = aq(x_0)$ en dus $a = p(x_0)/q(x_0) = r_1(x_0)$. Evenzo, $a = p'(x_0)/q'(x_0)$.

Uit $p(x_0)/q(x_0) = p'(x_0)/q'(x_0)$ volgt $p'(x_0)q(x_0) - p(x_0)q'(x_0) = 0$. Dus x_0 is een nulpunt van $pq' - p'q$ en dus zeker van $(pq' - p'q)q$. Maar dan $a = r_1(x_0) \in r_1(\Gamma)$, een tegenspraak.

En dus kunnen er geen dubbele nulpunten bestaan, en zijn er dus $deg(\alpha)$ verschillende nulpunten van $p(x) - aq(x)$.

Hieruit kunnen we concluderen dat er dus $deg(\alpha)$ punten $x_1 \in \overline{K}$ zijn waarvoor $\frac{p(x_1)}{q(x_1)} = a$ en $y_1 r_2(x_1) = b$ gelden. Dus bestaat de klasse $(a, b) \cdot Ker(\alpha)$ uit $deg(\alpha)$ punten, en omdat $|(a, b) \cdot Ker(\alpha)| = \#Ker(\alpha)$ volgt dat $deg(\alpha) = \#Ker(\alpha)$. □

Opmerking. Voor de niet-separabele endomorfismen α volgt dat $deg(\alpha) > \#Ker(\alpha)$, omdat er wel dubbele nulpunten mogen zijn.

Stelling 2.7. *Voor een elliptische kromme E over $\mathbb{F}_q$ volgt $Ker(\Phi_q - 1) = E(\mathbb{F}_q)$. In het bijzonder geldt $Ker(\Phi_q^n - 1) = E(\mathbb{F}_{q^n})$.*

Bewijs. We weten dat $\Phi_q(x) = x$ geldt, dan en slechts dan als $x \in \mathbb{F}_q$. Hetzelfde geldt voor y. Dus als $(x, y) \in E(\mathbb{F}_q)$ dan volgt $x, y \in \mathbb{F}_q$ en dus gelden $\Phi_q(x) = x$ en $\Phi_q(y) = y$

$$\Leftrightarrow \Phi_q(x, y) = (x, y)$$

$$\Leftrightarrow (\Phi_q - 1)(x, y) = 0$$

$$\Leftrightarrow (x, y) \in Ker(\Phi_q - 1)$$

En dus geld dat de punten in $E(\mathbb{F}_q)$ dezelfde punten zijn als de punten in $Ker(\Phi_q - 1)$. De tweede bewering volgt omdat Φ_{q^n} is gelijk aan Φ_q^n want

$$\begin{aligned}
\Phi_q^n(x, y) &= \Phi_q \circ \ldots \circ \Phi_q(x, y) && n \text{ keer} \\
&= \Phi_q \circ \ldots \circ \Phi_q(x^q, y^q) && n - 1 \text{ keer} \\
&= ((\ldots(x^q)^q \ldots)^q, (\ldots(y^q)^q \ldots)^q) && n \text{ keer} \\
&= (x^{q^n}, y^{q^n}) \\
&= \Phi_{q^n}(x, y)
\end{aligned}$$

□

Stelling 2.8. *Zij E een elliptische kromme gedefinieerd over $\mathbb{F}_q$ en zij $r, s \in \mathbb{N}_{>0}$ Dan is het endomorfisme $r\Phi_q + s$ separabel dan en slechts dan als $p \nmid s$.*

Bewijs. We noteren de vermenigvuldiging van een punt (x, y) met een zekere r als

$$r(x, y) = (R_r(x), yS_r(x)).$$

Dan volgt $(\Phi_q r)(x, y) = (R_r^q(x), y^q S_r^q(x)) = (R_r^q(x), y(x^3 + Ax + B)^{\frac{q-1}{2}} S_r^q(x))$. Hierdoor gelden de verhoudingen $\dfrac{R'_{r\Phi_q}}{S_{r\Phi_q}} = \dfrac{qR_r^{q-1} \cdot R'_r}{S_{r\Phi_q}} = 0$ en $\dfrac{R'_s}{S_s} = s$ (de tweede volgt uit propositie 2.28 in Washington [8]). Dan volgt uit lemma 2.26 van Washington:

$$\frac{R'_{r\Phi_q+s}}{S_{r\Phi_q+s}} = \frac{R'_{r\Phi_q}}{S_{r\Phi_q}} + \frac{R'_s}{S_s} = s$$

En dit is ongelijk nul dan en slechts dan als $p \nmid s$. □

Stelling 2.9. *Zij E elliptische kromme over $\mathbb{F}_q$ en n een natuurlijk getal ongelijk nul. Dan is $\Phi_q^n - 1$ een separabel endomorfisme, dus $\#E(\mathbb{F}_{q^n}) = deg(\Phi_q^n - 1)$.*

Bewijs. Endomorfismen op abelse groepen zijn bij elkaar opgeteld ook endomorfismen. Daarnaast geldt dat $r_1(x) = x^q - x$ dus $r'_1(x) = qx^{q-1} - 1 \equiv -1 \neq 0$. Dus $\Phi_q - 1$ is een separabel endomorfisme.

Verder volgt uit stelling 2.6 dat $deg(\Phi_q^n - 1) = \#Ker(\Phi_q^n - 1)$. En $Ker(\Phi_q^n - 1) \cong E(\mathbb{F}_{q^n})$ want voor elke $x \in \mathbb{F}_{q^n}$ geldt geldt $\Phi_{q^n}(x) = x$ dan en slechts dan als $\Phi_q^n(x) = x$.

Dit is equivalent met $(\Phi_q^n - 1)(x) = 0 \Leftrightarrow x \in Ker(\Phi_q^n - 1)$. Dus $\#E(\mathbb{F}_{q^n}) = \#Ker(\Phi_q^n - 1)$. □

2.4 Torsie punten en de Division Polynomials

Definitie 2.10. De *verzameling van n-torsie punten* $E[n]$ is de verzameling van punten P uit $E(\overline{K})$ met orde n. In andere woorden

$$E[n] = \{P \in E(\overline{K}) \mid nP = \infty\}.$$

Stelling 2.10. *Zij E een elliptische kromme over het lichaam K. Als de karakteristiek van K ongelijk aan 2 is, dan geldt*

$$E[2] \cong \mathbb{Z}_2 \oplus \mathbb{Z}_2.$$

Bewijs. Als de karakteristiek van K niet 2 is, zijn de 2-torsie punten makkelijk te vinden. Dit zijn de punten waarvan de raaklijn loodrecht op de x-as staat, en de raaklijn dus naar oneindig gaat. We weten dat de afgeleide van de vergelijking gelijk is aan $\dfrac{3x^2 + A}{2y}$. Dit gaat naar oneindig waar $y = 0$. En dat geldt alleen voor de nulpunten van $y^2 = x^3 + Ax + B$.

Zij $y^2 = (x - e_1)(x - e_2)(x - e_3)$ met $e_i \in \overline{K}$. Voor deze punten $(e_i, 0)$ geldt $2P = \infty$ per definitie van de optelling. Hierdoor ontstaat de 2-torsie verzameling $E[2] = \{(e_1, 0), (e_2, 0), (e_3, 0), \infty\}$.

Er geldt ook dat $(e_i, 0) + (e_j, 0) = (e_k, 0)$ met $i \neq j \neq k \neq i$ en $i, j, k \in \{1, 2, 3\}$ omdat ze alle drie op de x-as liggen, en dus het overgebleven snijpunt met de lijn door $(e_i, 0) + (e_j, 0)$ zijn. Deze verzameling is daarom isomorf met $\mathbb{Z}_2 \oplus \mathbb{Z}_2$. □

Opmerking. Elliptische krommen van karakteristiek 2 worden niet behandeld in deze scriptie, maar als de karakteristiek van K wel 2 is, dan geldt $E[2] \cong \mathbb{Z}_2$ of $E[2] \cong 0$. Voor het bewijs hiervan verwijs ik naar propositie 3.1 in het boek van Washington [8].

Definitie 2.11. De *division polynomen* $\psi_m \in \mathbb{Z}[x, y, A, B]$ zijn recursieve polynomen, met

$$\begin{aligned}
\psi_0 &= 0 \\
\psi_1 &= 1 \\
\psi_2 &= 2y \\
\psi_3 &= 3x^4 + 6Ax^2 + 12Bx - A^2 \\
\psi_4 &= \psi_3 x^4 + 6Ax^2 + 12Bx - A^2 \\
\psi_{2m+1} &= \psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3 && \textit{met} \quad m \geq 2 \\
\psi_{2m} &= \frac{\psi_m}{2y}(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2) && \textit{met} \quad m \geq 3
\end{aligned}$$

De komende drie stellingen worden veel gebruikt in het algoritme van Schoof, dat we in paragraaf 3.4 bespreken. De MAGMA scripts voor deze polynomen zijn te vinden in sectie 5.3 van de appendix.

Stelling 2.11. *Voor oneven n geldt $\psi_n \in \mathbb{Z}[x, y^2, A, B]$ en voor even n geldt $\psi_n \in 2y\mathbb{Z}[x, y^2, A, B]$.*

Bewijs. De gevallen waar $n \leq 4$ zijn triviaal. Het geval $n > 4$ wordt per inductie bewezen. Neem als inductiehypothese aan dat de stelling geldt voor alle $n < 2m$ waarbij $m > 2$. Dan geldt $2m > m + 2$, dus volgt dat alle polynomen in de definitie van ψ_{2m} en ψ_{2m+1} voldoen aan de inductiehypothese.

Voor ψ_{2m} geldt dan het volgende. Als m even is, dan zijn $\psi_m, \psi_{m+2}, \psi_{m-2}$ bevat in $2y\mathbb{Z}[x, y^2, A, B]$. Voor oneven m volgt dat ψ_{m-1} en ψ_{m+1} bevat zijn in $2y\mathbb{Z}[x, y^2, A, B]$. Dus volgt in beide gevallen dat de vergelijking uitkomt op $\frac{(2y)^2\psi_m}{2y}(\ldots) = 2y\psi_m(\ldots) \in 2y\mathbb{Z}[x, y^2, A, B]$.

In het geval van ψ_{2m+1} volgt dat voor zowel m even als oneven, een van de twee termen $\psi_{m+2}\psi_m^3$ of $\psi_{m-1}\psi_{m+1}^3$ een factor $2y \cdot (2y)^3 = 16y^4 = 16(x^3 + Ax + B)^2 \in \mathbb{Z}[x, y^2, A, B]$ heeft, en de andere term al geheel bevat was in $\mathbb{Z}[x, y^2, A, B]$. Dus volgt $\psi_{2m+1} \in \mathbb{Z}[x, y^2, A, B]$. □

Voor de division polynomen hebben we scripts in MAGMA geschreven. Deze zijn te vinden in de appendix bij sectie 5.3.1. Voor de implementatie van de even polynomen hebben we gebruik gemaakt van stelling 2.11. Het script is namelijk zo geschreven dat elke ψ_n met even n een factor y mist. Eigenlijk is elke ψ_n met even n gelijk aan $\psi_n = y\overline{\psi}_n$ met $\overline{\psi}_n := \frac{\psi_n}{y} \in \mathbb{Z}[x, y^2, A, B]$.

Daarnaast vervangen we alle y^2 met $x^3 + Ax + B$. Hierdoor is $\psi_n \in \mathbb{Z}[x, A, B]$, in plaats van $\mathbb{Z}[x, y^2, A, B]$, en krijg je polynomen met enkel x als variabel.

Definitie 2.12. Definieer de polynomen

$$\begin{aligned} \phi_m &= x\psi_m^2 - \psi_{m+1}\psi_{m-1} \\ \omega_m &= \frac{1}{4y}(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2) \end{aligned}$$

Stelling 2.12. *$\phi_n \in \mathbb{Z}[x, y^2, A, B]$ voor alle n. Voor n oneven volgt $\omega_n \in y\mathbb{Z}[x, y^2, A, B]$ en voor n even geldt $\omega_n \in \mathbb{Z}[x, y^2, A, B]$.*

Bewijs. Naar aanleiding van stelling 2.11, herschrijven we ψ_m met m even als $2y\tilde{\psi}_m$ waarbij $\tilde{\psi}_m \in \mathbb{Z}[x, y^2, A, B]$.

We beginnen met $\phi_m = x\psi_m^2 - \psi_{m+1}\psi_{m-1}$. Voor m even volgt dat

$$\phi_m = x \cdot (2y)^2\tilde{\psi}_m^2 - \psi_{m+1}\psi_{m-1} = 4x(x^3 + Ax + B)\tilde{\psi}_m^2 - \psi_{m+1}\psi_{m-1}$$

en voor m oneven volgt dan

$$\phi_m = x\psi_m^2 - (2y)\tilde{\psi}_{m+1} \cdot (2y)\tilde{\psi}_{m-1} = x\psi_m^2 - 4(x^3 + Ax + B)\tilde{\psi}_{m+1}\tilde{\psi}_{m-1}.$$

Beide gevallen zijn dus bevat in $\mathbb{Z}[x, y^2, A, B]$.

Voor ω_m weten we $\omega_m = \frac{1}{4y}(\psi_{m+2}\psi_{m-1}^2 - \psi_{m-2}\psi_{m+1}^2)$. Dan volgt voor even m:

$$\omega_m = \frac{1}{4y}(2y \cdot \tilde{\psi}_{m+2}\psi_{m-1}^2 - 2y \cdot \tilde{\psi}_{m-2}\psi_{m+1}^2) = \frac{1}{2}(\tilde{\psi}_{m+2}\psi_{m-1}^2 - \tilde{\psi}_{m-2}\psi_{m+1}^2)$$

en dit is bevat in $\mathbb{Z}[x, y^2, A, B]$. En voor m oneven volgt:

$$\frac{1}{4y}(\psi_{m+2} \cdot (2y)^2\tilde{\psi}_{m-1}^2 - \psi_{m-2} \cdot (2y)^2\tilde{\psi}_{m+1}^2) = y(\psi_{m+2}\tilde{\psi}_{m-1}^2 - \psi_{m-2}\tilde{\psi}_{m+1}^2)$$

en dit is bevat in $y\mathbb{Z}[x, y^2, A, B]$. □

Stelling 2.13. *Zij $P = (x, y)$ een punt op de elliptische kromme E die voldoet aan $y^2 = x^3 + Ax + B$ over een lichaam met karakteristiek ongelijk aan 2, en n een natuurlijk getal. Dan*

$$nP = \Big(\frac{\phi_n(x)}{\psi_n^2(x)}, \frac{\omega_n(x, y)}{\psi_n^3(x, y)}\Big)$$

Het bewijs van deze stelling gaan we niet behandelen. Dit bewijs stoelt namelijk op de theorie voor elliptische krommen over $\mathbb{C}$ en gebruikt daarbij de *Weierstrass $\wp$-functie*. Hiervoor verwijzen we naar sectie 9.5 van Washington [8]. De MAGMA scripts voor het berekenen van de x en y coördinaten van veelvouden van punten zijn te vinden in sectie 5.3.4.

Opmerking. We noteren het endomorfisme dat een punt n keer bij zichzelf opteld ook wel als $[n]$.

Gevolg 2.14. *Voor een punt $P = (x, y)$ geldt, $\psi_n(x, y) = 0 \Leftrightarrow P \in E[n]$.*

Bewijs. Als $\psi_n(x, y)$ gelijk is aan 0, volgt uit stelling 2.13 dat de teller van het x-coördinaat van het n-de veelvoud van (x, y) naar oneindig gaat. En dus is het punt $[n](x, y) = \infty$. □

Zoals we bij definitie 2.3 benoemd hebben, hebben we de Division Polynomials toegevoegd aan de optelling in MAGMA. Dit maakt de optelling namelijk sneller, omdat er minder geheugen nodig is voor de kortere polynomen. Hierbij dient uiteraard wel te gelden dat je binnen een bepaald lichaam $E[l]$ werkt. Maar dit is dan ook precies waar we onze optelling voor gaan gebruiken.

Gevolg 2.15. *Zij E een elliptische kromme. Het endomorfisme $\alpha(P) = nP$ met $n \in \mathbb{N}$ en $P \in E$ heeft graad n^2.*

Bewijs. We bewijzen eerst dat

$$\psi_n = \begin{cases} nx^{\frac{n^2-1}{2}} + \dots & \text{als } n \text{ oneven is} \\ y(nx^{\frac{n^2-4}{2}} + \dots) & \text{als } n \text{ even is} \end{cases}$$

Dit bewijzen we per inductie. Voor $n = 3$ en $n = 4$ volgt dit immers overduidelijk, want

$$\psi_3 = 3x^4 + \ldots = 3x^{\frac{3^2-1}{2}} + \ldots \text{ en } \psi_4 = 4y(x^6 + \ldots) = y(4x^{\frac{4^2-4}{2}} + \ldots).$$

Dus kunnen we onze inductiehypothese aannemen.

De inductiestap moeten we in vier delen bewijzen, namelijk de volgende vier gevallen:

$$\begin{cases} 2m+1 & \begin{cases} m \text{ is even} \\ m \text{ is oneven} \end{cases} \\ 2m & \begin{cases} m \text{ is even} \\ m \text{ is oneven} \end{cases} \end{cases}$$

Hier gaan we vluchtig doorheen omdat het simpele bewijzen zijn. Dus voor ψ_{2m+1} met m is even, weten we

$$\begin{aligned}
\psi_{2m+1} &= \psi_{m+2}\psi_m^3 - \psi_{m-1}\psi_{m+1}^3 \\
&= y^4(((m+2)x^{\frac{(m+2)^2-4}{2}} + \ldots)(m^3x^{3\frac{m^3-4}{2}} + \ldots)) \\
&\quad - ((m-1)x^{\frac{(m-1)^2-1}{2}} + \ldots)((m+1)^3x^{3\frac{(m+1)^2-1}{2}} + \ldots) \\
&= (x^3 + Ax + B)^2((m+2)m^3x^{\frac{(m+2)^2-4}{2}+3\frac{m^3-4}{2}}) \\
&\quad - (m-1)(m+1)^3x^{\frac{(m-1)^2-1}{2}+3\frac{(m+1)^2-1}{2}} + \ldots \\
&= (x^6 + \ldots)(m^4 + 2m^3)(x^{\frac{(2m+1)^2-13}{2}}) \\
&\quad - (m^4 + 2m^3 - 2m - 1)x^{\frac{4m^2+4m+1-1}{2}} + \ldots \\
&= (m^4 + 2m^3 - m^4 - 2m^3 + 2m + 1)x^{\frac{(2m+1)^2-1}{2}} + \ldots \\
&= (2m+1)x^{\frac{(2m+1)^2-1}{2}} + \ldots
\end{aligned}$$

En dus volgt voor m even dat de kopterm voor ψ_{2m+1} gelijk is aan $(2m+1)x^{\frac{(2m+1)^2-1}{2}}$. Op dezelfde manier volgen de andere gevallen. Omdat dit vooral uitschrijfwerk is, laten we dit over aan de lezer. Nu kunnen we concluderen dat voor $\phi_n = x\psi_n^2 - \psi_{n+1}\psi_{n-1}$ met even n de eerste term gelijk is aan:

$$\begin{aligned}
xy^2(nx^{\frac{n^2-4}{2}} + \ldots)^2 - ((n+1)x^{\frac{(n+1)^2-1}{2}} + \ldots)((n-1)x^{\frac{(n-1)^2-1}{2}} + \ldots) \\
= x^4n^2x^{n^2-4} - (n^2-1)x^{n^2} + \ldots \\
= x^{n^2} + \ldots
\end{aligned}$$

En voor oneven n volgt evenzo:

$$\begin{aligned} x(nx^{\frac{n^2-1}{2}} + \ldots)^2 - y((n+1)x^{\frac{(n+1)^2-4}{2}} + \ldots)y((n-1)x^{\frac{(n-1)^2-4}{2}} + \ldots) \\ = xn^2x^{n^2-1} - (n^2-1)(x^3 + \ldots)x^{n^2-3} + \ldots \\ = x^{n^2} + \ldots \end{aligned}$$

En dan geldt voor

$$\psi_n^2 = \begin{cases} y^2(nx^{\frac{n^2-4}{2}} + \ldots)^2 = n^2x^{n^2-1} + \ldots & \text{voor } n \text{ is even} \\ (nx^{\frac{n^2-1}{2}} + \ldots)^2 = n^2x^{n^2-1} + \ldots & \text{voor } n \text{ is oneven} \end{cases}$$

En dus volgt dat de graad van het endomorfisme α gelijk is aan $max\{deg(\phi_n), deg(\psi_n^2)\} = n^2$, op voorwaarde dat ϕ_n en ψ_n geen gemeenschappelijke nulpunten hebben. Voor het bewijs hiervan, zie Corollary 3.7 in Washington [8]. □

Zoals we al eerder bij de definitie 2.3 hebben uitgelegd, hebben we Division Polynomials toegevoegd aan onze MAGMA-implementatie voor de optelling. Voor de optelling worden polynomen berekend, en door de veelvouden van ϕ_n eruit te halen, kan er een inverse berekend worden.

Tot slot bewijzen we het volgende resultaat. Dit is een uitbereiding van stelling 2.10. Hiervoor gebruiken we echter de resultaten van de division polynomen.

Stelling 2.16. *Zij E een elliptische kromme over het lichaam K en n een natuurlijk getal. Als $Kar(K) \nmid n$ en ongelijk 0 is, dan geldt*

$$E[n] \cong \mathbb{Z}_n \oplus \mathbb{Z}_n$$

Bewijs. We weten dat $[n]$ van de vorm $[n](x,y) = (r_1(x), yr_2(x))$ is. Uit stelling 2.13 kunnen we afleiden dat $r_1(x) = \frac{x^{n^2}+\ldots}{n^2x^{n^2-1}+\ldots}$. De teller van de afgeleide wordt dan van de vorm $n^2x^{2n^2-2} + \ldots \neq 0$. Omdat $Kar(K) \nmid n$ geldt $r_1'(x) \neq 0$ dus is $[n]$ separabel. Ook kunnen we uit bovenstaande vergelijking afleiden dat $n^2 = deg([n])$. Uit stelling 2.6 volgt dan $n^2 = \#Ker([n]) = \#E[n]$.

De structuur stelling voor eindige abelse groepen (te vinden in Washington [8], appendix B) stelt dat $E[n]$ isomorf is met

$$\mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \ldots \oplus \mathbb{Z}_{n_k}$$

voor natuurlijke getallen $n_1, n_2, \ldots, n_k$ met $n_i \mid n_{i+1}$ voor alle i. Stel nu dat $l \mid n_1$ voor zekere l. Dan volgt $l \mid n_i$ voor alle i. Dat betekent dat $E[l] \subset E[n]$ orde l^k heeft. Omdat we net hebben aangetoond dat een torsie groep $E[l]$ orde l^2 moet hebben, volgt $k = 2$. Dus volgt dat $E[n] \cong \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2}$, en dus $n_2 \mid n$. Omdat $n^2 = \#E[n] = n_1n_2$ volgt $n_1 = n_2 = n$ en dus geldt

$$E[n] \cong \mathbb{Z}_n \oplus \mathbb{Z}_n.$$

□

2.5 De stelling van Hasse en zijn belangrijke gevolgen

Stelling 2.17. *Voor een elliptische kromme E over een eindig lichaam $\mathbb{F}_q$ geldt*

$$E(\mathbb{F}_q) \cong \mathbb{Z}_n \text{ of } E(\mathbb{F}_q) \cong \mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2}$$

voor een natuurlijke $n \geq 1$, of $n_1, n_2 \geq 1$ met $n_1 \mid n_2$.

Bewijs. $E(\mathbb{F}_q)$ is een abelse groep (stelling 2.3), en dus isomorf aan een directe som van cyclische groepen $\mathbb{Z}_{n_1} \oplus \mathbb{Z}_{n_2} \oplus \mathbb{Z}_{n_3} \oplus \ldots \oplus \mathbb{Z}_{n_r}$ met $n_i \mid n_{i+1}$ voor alle $i \geq 1$. Voor elke i is er een groep $\mathbb{Z}_{n_i}$ met n_1 elementen van een orde die n_1 deelt. Dat geeft n_1^r elementen met een orde die n_1 deelt. Uit stelling 2.16 volgt dat er maximaal n_1^2 van zulke punten bestaan. Dus volgt $r \leq 2$. □

Stelling 2.19 is cruciaal voor de stelling van Hasse, die we na stelling 2.19 gaan behandelen. Voor deze stelling, en een aantal stellingen daaropvolgend, zullen we endomorfismen over een elliptische kromme *E vertalen* naar lineaire transformaties in de vorm van matrices over $E[n]$. Dit werkt als volgt: een endomorfisme α beschrijft een bepaalde actie over een basis $\{T_1, T_2\}$ voor $E[n]$, voor een zekere $n \in \mathbb{N}$. Zij nu $\alpha(T_1) = sT_1 + uT_2$ en $\alpha(T_2) = tT_1 + vT_2$. Dan volgt de transformatie matrix

$$\alpha_n = \begin{bmatrix} s & t \\ u & v \end{bmatrix}.$$

met $s, t, u, v \in \mathbb{Z}_n$, die deze actie beschrijft. En dit is, zeer beknopt, hoe het *vertalen* van endomorfismen naar matrices over $E[n]$ werkt.

Stelling 2.18. *Zij α een endomorfisme op een elliptische kromme E over een lichaam K en n een natuurlijk getal dat niet deelbaar is door de karakteristiek van K. Dan geldt $det(\alpha_n) \equiv deg(\alpha) \pmod{n}$.*

Dit gaan we niet bewijzen. Het bewijs is te vinden in propositie 3.15 in Washington [8]. Dit bewijs berust zich op de *Weil Pairing*, die beschreven wordt in hoofdstuk 3.3.

Stelling 2.19. *Zij $r, s \in \mathbb{Z}$ met $ggd(s, q) = 1$ en E een elliptische kromme over $\mathbb{F}_q$. Dan volgt $deg(r\Phi_q - s) = r^2q + s^2 - rsa$ waarbij $a = q + 1 - \#E(\mathbb{F}_q)$.*

Bewijs. We tonen eerst aan dat voor endomorfismen α en β op de kromme E en gehele getallen a en b geldt

$$deg(a\alpha + b\beta) = a^2 deg(\alpha) + b^2 deg(\beta) + ab(deg(\alpha + \beta) - deg(\alpha) - deg(\beta)).$$

Zij α_n en β_n die matrix representaties van α respectievelijk β over een basis van $E[n]$ waarbij $n \in \mathbb{N}$ en $q \nmid n$. Dan doet $a\alpha_n + b\beta_n$ het zelfde als $a\alpha + b\beta$ maar dan over $E[n]$. Uit simpel uitschrijven volgt

$$det(a\alpha_n + b\beta_n) = a^2 det(\alpha_n) + b^2 det(\beta_n) + ab(det(\alpha_n + \beta_n) - det(\alpha_n) - det(\beta_n))$$

voor alle matrices α_n en β_n. Dus volgt

$$deg(a\alpha + b\beta) \equiv a^2 deg(\alpha) + b^2 deg(\beta) + ab(deg(\alpha + \beta) - deg(\alpha) - deg(\beta)) \pmod{n}.$$

En omdat dit voor elke $n \in \mathbb{N}$ geldt, moet bovenstaande vergelijking gelden.

We weten dat $deg(\Phi_q) = q$ en $deg(-1) = 1$. Dan volgt uit bovenstaande vergelijking $deg(r\Phi_q - s) =$

$$r^2 deg(\Phi_q) + s^2 deg(-1) + rs(deg(\Phi_q - 1) - deg(\Phi_q) - deg(-1))$$

$$= qr^2 + s^2 + rs(deg(\Phi_q - 1) - q + 1)$$

En met stelling 2.9 kunnen we dan concluderen dat dit gelijk is aan $qr^2 + s^2 + rs(\#E(\mathbb{F}_q) - q + 1) = qr^2 + s^2 + rsa$. □

De volgende stelling is een van de fundamentele stellingen in deze scriptie: de stelling van Hasse. Hierop bouwen veel verschillende algorithmes zich voort die gericht zijn op het tellen van punten op elliptische krommen. Deze stelling stelt dat er voor het vinden van het aantal punten op een elliptische kromme over een eindig lichaam een afgebakend domein bestaat dat afhankelijk is van de grootte van het eindige lichaam $\mathbb{F}_q$.

Stelling 2.20 (Stelling van Hasse)**.** *Voor een elliptische kromme E over het eindige lichaam $\mathbb{F}_q$ geldt dat de orde van $E(\mathbb{F}_q)$ voldoet aan*

$$|q + 1 - \#E(\mathbb{F}_q)| \leq 2\sqrt{q}$$

Bewijs. Definieer

$$a \coloneqq q + 1 - \#E(\mathbb{F}_q) = q + 1 - deg(\Phi_q - 1),$$

dan willen we laten zien dat $|a| \leq 2\sqrt{q}$.

We weten $deg(r\Phi_q - s) \geq 0$. Uit stelling 2.9 volgt dat $deg(r\Phi_q - s) = qr^2 + s^2 - rsa$ dus

$$qr^2 + s^2 - rsa \geq 0.$$

Door beide kanten door s^2 te delen volgt $q\left(\frac{r}{s}\right)^2 - a\left(\frac{r}{s}\right) + 1 \geq 0$ voor alle r, s met $ggd(s, q) = 1$. De verzameling van rationale getallen $\frac{r}{s}$ met $ggd(s, q) = 1$ is dicht in $\mathbb{R}$. Dus volgt

$$qx^2 - ax + 1 \geq 0$$

voor alle $x \in \mathbb{R}$. Er is dus maximaal één snijpunt met de x-as, dus heeft $qx^2 - ax + 1$ maximaal één nulpunt in $\mathbb{R}$. De discriminant $a^2 - 4q$ is dus kleiner of gelijk aan 0. En dus volgt $|a| \leq 2\sqrt{q}$. □

Welnu volgt een zeer fundamentele stelling voor het volgende hoofdstuk. Hierin wordt gesproken over de al voorheen benoemde waarde $a \coloneqq q + 1 - \#E(\mathbb{F}_q)$, deze noemen we het *spoor van Frobenius*.

Stelling 2.21. *Zij E een elliptische kromme over $\mathbb{F}_q$. Definieer $a \coloneqq q+1-\#E(\mathbb{F}_q)$, dan volgt dat*

$$\Phi_q^2 - a\Phi_q + q = 0$$

als endomorfismen op E. Ook geldt dat a het enige natuurlijke getal is waarvoor deze relatie geldt. Tot slot geldt dat dit unieke getal a voor alle m met $ggd(m,q)=1$ voldoet aan

$$a \equiv Spoor((\Phi_q)_m) \pmod{m}.$$

Opmerking. In andere woorden, voor $(x,y) \in E(\overline{\mathbb{F}}_q)$ geldt $(x^{q^2}, y^{q^2}) - a(x^q, y^q) + q(x,y) = \infty$.

Bewijs. Uit stelling 2.6 volgt dat als $\Phi_q^2 - a\Phi_q + q$ een niet-nul endomorfisme is, dat de kern dan eindig is. Dus wij zullen bewijzen dat de kern oneindig groot is, dat het endomorfisme $\Phi_q^2 - a\Phi_q + q$ gelijk is aan nul.

Zij $m \in \mathbb{N}_{\leq 1}$ met $ggd(m,q)=1$. Dan induceert Φ_q een matrix $(\Phi_q)_m$ die de transformatie van Φ_q naleeft op $E[m]$. Stel

$$(\Phi_q)_m = \begin{bmatrix} s & t \\ u & v \end{bmatrix}.$$

Dan geldt volgens stelling 2.9 dat $\Phi_q - 1$ separabel is, en volgt uit stelling 2.6 en stelling 2.18

$$\#Ker(\Phi_q - 1) = deg(\Phi_q - 1) \equiv det((\Phi_q)_m - I) \equiv sv - ut + 1 - (s+v).$$

Verder volgt uit stelling 2.18 $sv - tu = det((\Phi_q)_m) \equiv deg(\Phi_q) \equiv q \pmod{m}$ Dus

$$q + 1 - (s+v) \equiv sv - ut + 1 - (s+v) \equiv det((\Phi_q)_m - I) \equiv \#Ker(\Phi_q - 1) = q + 1 - a \pmod{m}$$

en dus volgt de equivalentie $a \equiv s + v \equiv Spoor((\Phi_q)_m)$.

Dan volgt uit de Cayley-Hamilton stelling [3] $(\Phi_q)_m^2 - a(\Phi_q)_m + qI = 0 \pmod{m}$. Daaruit volgt dat de m-torsiepunten in de kern van $\Phi_q)_m^2 - a(\Phi_q)_m + qI$ liggen. En omdat er oneindig veel mogelijkheden voor m zijn, volgt dat de kern van $\Phi_q^2 - a\Phi_q + q$ oneindig is. Dus geldt

$$\Phi_q^2 - a\Phi_q + q = 0.$$

Tot slot bewijzen we de uniciteit van deze $a \coloneqq q+1-\#E(\mathbb{F}_q)$. Stel er is een a_1 met $a_1 \neq a$ die voldoet aan $\Phi_q^2 - a_1\Phi_q + q = 0$. Dan geldt

$$(a - a_1)\Phi_q = (\Phi_q^2 - a\Phi_q + q) - (\Phi_q^2 - a_1\Phi_q + q) = 0.$$

Verder geldt dat $\Phi_q \colon E(\overline{\mathbb{F}_q}) \to E(\overline{\mathbb{F}_q})$ surjectief is. Dit volgt, heel simpel, omdat je voor elke $z \in \overline{\mathbb{F}}_q$ de q-de-machtswortel kunt nemen en dus voor elk punt $(w,z) \in E(\overline{\mathbb{F}_q})$ de inverse $(\sqrt[q]{w}, \sqrt[q]{z}) \in E(\overline{\mathbb{F}_q})$ kunt nemen.

Dus mogen we concluderen dat $(a - a_1)$ de elementen van $E(\overline{\mathbb{F}_q})$ naar 0 laat gaan. Dus, in het bijzonder, laat $(a - a_1)$ $E[m]$ naar 0 gaan voor alle $m \geq 1$. Aangezien $E[m] \neq \infty$ voor alle m met $ggd(m, q) = 1$, volgt $(a - a_1) \equiv 0 \pmod{m}$ voor alle m met $ggd(m, q) = 1$. Dus volgt $(a - a_1) = 0$ en dus is deze a uniek. □

3 Manieren om punten te tellen

De verdieping in deze scriptie ligt op het tellen van punten op elliptische krommen. Er zijn verschillende manieren om dit te doen. Er is een algoritme dat in alle gevallen opgaat, maar de rekentijd van dit algoritme is erg hoog. En voor krommen over *kleinere* eindige lichamen zijn er simpelere manieren om de punten te berekenen. Hierdoor zijn er verschillende algoritmes ontstaan die op verschillende elliptische krommen betrekking hebben.

We gaan naar de belangrijkste algoritmes kijken. We bouwen dit op van simpele methodes over kleinere lichamen naar de uiterste gevallen van zeer grote eindige lichamen waarbij de algoritmes bestaan uit uitgebreide stappen.

Om het exacte algoritme duidelijk te maken, zijn van alle algoritmes scripts uitgewerkt in MAGMA. Dit is een programmeertaal, te gebruiken in een terminal, die vooral voor algebraïsche structuren ontworpen is. Op `http://magma.maths.usyd.edu.au/magma/handbook/` is een handboek te vinden, maar ook tijdens het gebruik van deze taal kan makkelijk achterhaald worden hoe een command werkt.

Er zijn ook prestatietesten gedaan voor de MAGMA code van een aantal algoritmes. Deze zijn gericht op de CPU tijd. Hiervoor is de functie `Timecheck(n,m)` gebruikt, die lichtelijk verschilt per algoritme. Hierbij is `m` de macht m in het domein $[2^{m-1}, 2^m]$ waartussen het priemgetal q willekeurig gekozen wordt en is `n` het aantal willekeurige elliptische krommen dat getest wordt. Voor `n` is standaard 10 gekozen. Hierdoor worden er per `m` 10 willekeurige krommen getests met waarden A en B voor de vergelijking $y^2 = x^3+Ax+B$ binnen $\mathbb{F}_q$. De resultaten worden gegeven in seconden. De resultaten worden weergegeven in een tabel waarbij gekeken wordt naar het gemiddelde van de 10 tests en het aantal seconden van de snelste en langzaamste kromme. De tabellen worden aangevuld met een conclusie over de complexiteit van het algoritme.

De scripts staan in de Appendix, en de achterliggende algoritmes en opvallende aspecten van het implementeren worden per paragraaf besproken.

3.1 Legendre symbool voor het tellen van punten

Het Legendre algoritme, ook wel eens het Lang-Trotter algoritme genoemd, is een heel vanzelfsprekend algoritme waarbij je voor elk element x in het lichaam $\mathbb{F}_q$ bepaalt of het een x-coördinaat van een punt op de elliptische kromme is. Hierbij wordt gekeken of het invullen van x in $x^3 + Ax + B$ een kwadraat y^2 binnen $\mathbb{F}_q$ oplevert. Zo ja, geven deze waardes x en y de punten (x, y) en $(x, -y)$ uit.

Eerst zal het Legendre symbool gedefinieerd worden, dat gebruikt wordt om punten te tellen. Vervolgens zal er een formule opgesteld worden om het aantal punten bij elkaar op te tellen.

Definitie 3.1. Zij K een lichaam. Dan is het Legendre symbool $\left(\frac{x}{p}\right)$ voor een oneven priem $p \in K$ en

$x \in K$ gedefinieerd als

$$\left(\frac{x}{p}\right) = \begin{cases} +1 & \text{als } t^2 \equiv x \pmod{p} \text{ een oplossing met } t \not\equiv 0 \text{ heeft} \\ -1 & \text{als } t^2 \equiv x \pmod{p} \text{ geen oplossing voor t heeft} \\ 0 & \text{als } x \equiv 0 \pmod{p} \end{cases}$$

Opmerking. Dit willen we ook nog wel eens opschrijven gebruikmakend van het eindige lichaam $\mathbb{F}_q$ met oneven q, waar we in werken. In dat geval geldt voor $x \in \mathbb{F}_q$:

$$\left(\frac{x}{\mathbb{F}_q}\right) = \begin{cases} +1 & \text{als } t^2 = x \text{ een oplossing met } t \in \mathbb{F}_q^\times \text{ heeft} \\ -1 & \text{als } t^2 = x \text{ geen oplossing voor } t \in \mathbb{F}_q \text{ heeft} \\ 0 & \text{als } x = 0 \end{cases}$$

Stelling 3.1. *Zij E een elliptische kromme gedefinieerd door $y^2 = x^3 + Ax + B$ in $\mathbb{F}_q$, dan geldt*

$$\#E(\mathbb{F}_q) = q + 1 + \sum_{x \in \mathbb{F}_q} \left(\frac{x^3 + Ax + B}{\mathbb{F}_q}\right)$$

Bewijs. Voor elke x-coördinaat x_0 waarvoor $y_0^2 = x_0^3 + Ax_0 + B$ geldt, met $y_0 \neq 0$ bestaan er twee punten in $E(\mathbb{F}_q)$. Als deze y_0 niet bestaat (en er dus geen $y \in \mathbb{F}_q$ is zodanig dat $y^2 = x_0^3 + Ax_0 + B$) dan zijn er geen punten. Het punt waarvoor $y_0 = 0$ komt echter maar één keer voor. Zodoende geldt voor elke x_0 dat er $1 + \left(\frac{x_0^3 + Ax_0 + B}{\mathbb{F}_q}\right)$ punten op E bestaan. Als je dit sommeert over alle x_0, geeft dit $\sum_{x \in \mathbb{F}_q} \left(1 + \left(\frac{x_0^3 + Ax_0 + B}{\mathbb{F}_q}\right)\right)$ punten. Voeg hieraan $+1$ toe voor ∞ en dit geeft

$$\begin{aligned} \#E(\mathbb{F}_q) &= 1 + \sum_{x \in \mathbb{F}_q} \left(1 + \left(\frac{x_0^3 + Ax_0 + B}{\mathbb{F}_q}\right)\right) \\ &= 1 + q + \sum_{x \in \mathbb{F}_q} \left(\frac{x^3 + Ax + B}{\mathbb{F}_q}\right) \end{aligned}$$

□

Gevolg 3.2. *Zij $x^3 + Ax + B$ een polynoom zonder dubbele nulpunten, waarbij $A, B \in \mathbb{F}_q$ en q oneven. Dan*

$$\left|\sum_{x \in \mathbb{F}_q} \left(\frac{x^3 + Ax + B}{\mathbb{F}_q}\right)\right| \leq 2\sqrt{q}.$$

Bewijs. Dit is een combinatie van stelling 3.1 en Hasse's stelling, stelling 2.20. □

Welnu volgt het algoritme. De MAGMA code is een simpel scriptje waar twee commands voor geschreven zijn. Het command om de punten te tellen is aan te roepen met `Legendre(A,B,q)` waarbij `A` en `B` de

coordinaten in de elliptische kromme $y^2 = x^3 + Ax + B$ zijn, die gegeven is over het eindige lichaam $\mathbb{F}_q$, waarnaar de `q` refereert. Deze functie gebruikt het Legendre symbool, waarvoor een aparte functie geschreven is, aan te roepen via het command `Legendresymbol(x)`. Het script staat in sectie 5.4.

1 Stel je counter op 0
2 Neem een element uit $\mathbb{F}_q$
3 Bereken $\left(\frac{x^3+Ax+B}{\mathbb{F}_q}\right)$
4 Tel $\left(\frac{x^3+Ax+B}{\mathbb{F}_q}\right)$ op bij counter
5 Herhaal stap 3 en 4 voor elk element uit $\mathbb{F}_q$
6 En dan volgt het antwoord voor $\#E(\mathbb{F}_q)$ door de counter bij $q+1$ op te tellen

Voorbeeld 3.1. Neem $E : y^2 = x^3 + 2x + 1$ over $\mathbb{F}_5 \cong \mathbb{Z}/5\mathbb{Z}$. Counter staat nu op 0.
We beginnen met $1 \in \mathbb{F}_5$. Dit geeft $1^3 + 21 + 1 = 4$. Dit is een kwadraat in $\mathbb{F}_5$. Dus Counter = 1.
$2 \in \mathbb{F}_5$ geeft $2^3 + 2 \cdot 2 + 1 \equiv 3$. Dit is geen kwadraat in $\mathbb{F}_5$. Dus Counter = 0.
$3 \in \mathbb{F}_5$ geeft $3^3 + 2 \cdot 3 + 1 \equiv 4$. Dit is een kwadraat in $\mathbb{F}_5$. Dus Counter = 1.
$4 \in \mathbb{F}_5$ geeft $4^3 + 2 \cdot 4 + 1 = 3$. Dit is geen kwadraat in $\mathbb{F}_5$. Dus Counter = 0.
$5 \equiv 0 \in \mathbb{F}_5$ geeft $0^3 + 2 \cdot 0 + 1 = 1$, dus Counter = 1.
Dus volgt $\#E(\mathbb{F}_5) = 5 + 1 + 1 = 7$.

3.1.1 Resultaten prestatietest

Welnu volgt een overzichtje van de resultaten van de prestatietesten. Het script voor het `Timecheck(n,m)` algoritme is te vinden in sectie 5.4.1. Zoals eerder benoemd, is `m` de macht m van het domein $[2^{m-1}, 2^m]$ waartussen het priemgetal q willekeurig gekozen wordt en is voor `n` standaard 10 gekozen. Hierdoor worden er per `m` 10 willekeurige krommen getest met waarden A en B binnen $\mathbb{F}_q$.

`m`	gemiddelde	max	min
10	0	0	0
11	0,003	0,01	0
12	0,004	0,01	0
13	0,01	0,02	0
14	0,018	0,02	0,01
15	0,04	0,05	0,03
16	0,077	0,1	0,05
17	0,152	0,2	0,12
18	0,319	0,42	0,22
19	0,617	0,77	0,46
20	1,249	1,51	0,87

21	2,393	3,3	1,84
22	4,909	6,77	3,78
23	9,894	13,35	7,29
24	20,613	27,05	12,63
25	35,283	47,25	24,45
26	72,323	88,63	56,22
27	132,901	165,98	108,35
28	270,947	337,35	184,28

Tijdens het compilen van `Timecheck(10,29)` bleef MAGMA telkens hangen vanwegen een geheugenprobleem. MAGMA maakt het lichaam $\mathbb{F}_q$ (met q een priemgetal binnen het interval $[2^{28} : 2^{29}]$) geheel aan bij het aanroepen van de for-loop in regel 14 van 5.4, en dat kost te veel geheugen. Hierdoor kan een inschatting van de CPU tijd voor priemen q groter dan 2^{29} niet gedaan worden. Het algoritme zou dus versnelt kunnen worden door de for-loop van 1 tot en met q te laten lopen.

De complexiteit van het Legendre algoritme groeit lineair, omdat je de for-loop over x in $\mathbb{F}_q$ loopt en dus van lengte q is en alle andere commands zijn constant. Daarom $\mathcal{O}(q) = \mathcal{O}(2^m)$.

3.2 Punten tellen voor een uitgebreidingslichaam

Het volgende algoritme is een makkelijke aanvulling op het algoritme met het Legendre symbool.

Stel je weet het aantal punten van een elliptische kromme E over $\mathbb{F}_q$. Dan kun je met het volgende algoritme het aantal punten op E berekenen over het uitbreidingslichaam $\mathbb{F}_{q^n}$, voor een zekere $n \in \mathbb{N}_{\geq 1}$.

Stelling 3.3. *Zij $\#E(\mathbb{F}_q) = q + 1 - a$. Herschrijf het polynoom $x^2 - ax + q = (x - \alpha)(x - \beta)$ waarbij $\alpha, \beta \in \overline{\mathbb{Q}}$. Dan geldt voor alle $n \geq 1$*

$$\#E(\mathbb{F}_{q^n}) = q^n + 1 - (\alpha^n + \beta^n)$$

Bewijs. We tonen allereerst aan dat $\alpha^n + \beta^n$ een geheel getal is, voor alle natuurlijke getallen n.

Definieer $s_n = \alpha^n + \beta^n$. Dan geldt $s_0 = 2$, $s_1 = a$ en $s_{n+1} = as_n - qs_{n-1}$. Voor 0 spreekt het voor zich. Voor 1 geldt dan per definitie $\alpha + \beta = a$. Voor $n + 1$ volgt de vergelijking door $\alpha^2 - a\alpha + q = 0$ te vermenigvuldigen met α^{n-1} om zo uit te komen op $\alpha^{n+1} = a\alpha^n - q\alpha^{n-1}$. Ditzelfde geldt voor β en dus volgt $s_{n+1} = (\alpha^{n+1} + \beta^{n+1}) = a(\alpha^n + \beta^n) - q(\alpha^{n-1} + \beta^{n-1})$. Per inductie volgt dan dat s_n voor alle n een geheel getal is.

We bekijken nu het polynoom $f(x) = (x^n - \alpha^n)(x^n - \beta^n) = x^{2n} - (\alpha^n + \beta^n)x^n + q^n$. Dan is het originele polynoom $x^2 - ax + q = (x - \alpha)(x - \beta)$ een deler van $f(x)$ en het quotientenpolynoom $g(x)$ heeft gehele coëfficiënten. Dan volgt $(\Phi_{q^n})^2 - (\alpha^n + \beta^n)\Phi_{q^n} + q^n = (\Phi_q^n)^2 - (\alpha^n + \beta^n)\Phi_q^n + q^n = f(\Phi_q) = g(\Phi_q)(\Phi_q^2 - a\Phi_q + q) = 0$ als endomorfisme op E. Dit volgt uit stelling 2.21. Uit diezelfde stelling volgt dat deze a uniek is en wel $a = q^n + 1 - \#E(\mathbb{F}_{q^n})$. En dus volgt $\alpha^n + \beta^n = q^n + 1 - \#E(\mathbb{F}_{q^n})$. □

Het MAGMA script geschreven voor dit algoritme kan worden aangeroepen met het command `HogereMacht(cardE, q,n)` waarbij `cardE` de cardinaliteit is van $E(\mathbb{F}_q)$, `q` de cardinaliteit van $\mathbb{F}_q$ en `n` de macht waarmee je het lichaam wil uitbreiden. Deze is te vinden in sectie 5.5.

1 Bereken het spoor van Frobenius ($a = q + 1 - \#E(\mathbb{F}_q)$)
2 Splits het polynoom $x^2 - ax + q$ over het uitbreidingslichaam $\mathbb{Q}(\sqrt{a^2 - 4q})$
3 Verhef de gevonden nulpunten α en β apart tot de n-de macht
4 Geef de waarde $q^n + 1 - (\alpha^n + \beta^n)$ uit

Een belangrijk aspect van dit algoritme is stap 2, waar je het polynoom $x^2 - ax + q$ over $\mathbb{Q}(\sqrt{a^2 - 4q})$ splitst. Hiermee vermijdt je het niet kunnen splitsen van een polynoom omdat het irreducibel in $\mathbb{Q}$ is, maar creeër je ook niet een te grote uitbreiding van $\mathbb{Q}$.

Voorbeeld 3.2. Neem de elliptische kromme $y^2 = x^3 + 2x + 4$ over $\mathbb{F}_5$. Deze elliptische kromme heeft orde 7. Nu willen we deze kromme uitbreiden naar $\mathbb{F}_{5^4}$. We beginnen met het berekenen van het spoor van Frobenius: $a = 5 + 1 - 7 = -1$.

Hieruit is het polynoom $x^2 + x + 5 = (x + \frac{1}{2} - \frac{i\sqrt{19}}{2})(x + \frac{1}{2} + \frac{i\sqrt{19}}{2})$ af te leiden. Dit bevestigt

$$\#E(\mathbb{F}_5) = 5 + 1 - (-\frac{1}{2} + \frac{i\sqrt{19}}{2} - \frac{1}{2} - \frac{i\sqrt{19}}{2}) = 5 + 1 + 1 = 7$$

Dan volgt voor 5^4

$$\begin{aligned}\#E(\mathbb{F}_{5^4}) &= 5^4 + 1\Big((-\frac{1}{2} + \frac{i\sqrt{19}}{2})^4 + (-\frac{1}{2} - \frac{i\sqrt{19}}{2})^4\Big) \\ &= 625 + 1 - \Big(\frac{31 + 9i\sqrt{19}}{2} + \frac{31 - 9i\sqrt{19}}{2}\Big) \\ &= 625 + 1 - 31 = 595\end{aligned}$$

Opmerking. Om bovenstaand voorbeeld in `HogereMacht` te berekenen, neem `cardE:=7`, `q:=5` en `n:=4`.

3.3 Baby Step, Giant Step en de stelling van Mestre

Het Baby Step, Giant Step algoritme is een algoritme om de orde van een kromme E te vinden. Het idee is om de ordes van een willekeurige punten P op de kromme te vinden, tot een kleinst gemeen veelvoud van die ordes binnen het domein van Hasse valt (zoals genoemd in stelling 2.20).

3.3.1 Baby Step, Giant Step om de orde van een punt te bepalen

Om de orde van een punt P te bepalen, vermenigvuldigd men P op twee manieren: via de Baby Steps en via de Giant Steps. De Baby Steps zijn de minimale, opeenvolgende vermenigvuldigingen van P (dus $1P$, $2P$, $3P$ enzovoort). De Giant Steps zijn grote veelvouden van P waarbij rekening gehouden wordt met de grenzen van het domein van Hasse. Hierbij test je alle mogelijke waarden binnen het domein dat volgt uit de stelling van Hasse. De gevonden punten worden opgeslagen en vervolgens vergeleken met elkaar om zo een overeenkomend punt te vinden.

Als dit lukt, kun je de vermenigvuldigingsfactoren van elkaar aftrekken om zo een veelvoud van de orde van het punt P te vinden. Door het punt P vervolgens met de delers van de gevonden orde te vermenigvuldigen, kun je de precieze orde van P bepalen.

De enige vraag die je jezelf kan stellen, is "waarom is er een match"? Dit volgt heel simpel uit de volgende stelling:

Stelling 3.4. *Zij E een elliptische kromme over het lichaam $\mathbb{F}_q$ en m een natuurlijk getal $m > q^{\frac{1}{4}}$. Voor het willekeurige punt P en zijn veelvoud $Q = (q+1)P$ zijn er natuurlijke getallen k en j te vinden, zodanig dat $Q + k(2mP) = \pm jP$ waarbij $j \in \{0, \ldots, m\}$ en $k \in \{-m, \ldots, m\}$.*

Bewijs. Hiervoor bewijzen we eerst het volgende lemma:

> Zij a een natuurlijk getal met $|a| \leq 2m^2$. Dan zijn er natuurlijke getallen a_0 en a_1 met $-m < a_0 \leq m$ en $-m \leq a_1 \leq m$ zodanig dat $a = a_0 + 2ma_1$.

Als $a_0 \equiv a \pmod{2m}$, met $-m < a_0 \leq m$ en $a_1 = \frac{a-a_0}{2m}$. Dan $|a_1| \leq \frac{2m^2+m}{2m} = \frac{2m+1}{2} < \frac{2m+2}{2} = m + 1$.

Laat nu $a = a_0+2ma_1$ zijn, en zij $k = -a_1$. Dan volgt $Q+k(2mP) = (q+1-2ma_1)P = (q+1-a+a_0)P = NP + a_0P = a_0P = \pm jP$. □

De stappen van het algoritme zijn als volgt. Neem hierbij in acht dat E een elliptische kromme over $\mathbb{F}_q$ is met coefficienten A en B, zoals altijd.

1 Neem een natuurlijk getal m met $m > q^{1/4}$.

2 Bereken de punten jP met $j = 1, 2..., m$ en sla ze op. Geef j uit indien $jP = \infty$ en stop. Zo niet, ga dan verder met stap 4.
3 Bereken het punt $Q = (q + 1)P$.
4 Bereken de punten $Q + k(2mP)$ voor $k = -m, -(m - 1), .., m$ en vergelijk ze met alle opgeslagen punten $\pm jP$.
5 Als er sprake is van een overeenkomst $Q + k(2mP) = \pm jP$ volgt $(q + 1 + 2mk \mp j)P = \infty$. Sla $(q + 1 + 2mk \mp j)$ op als M.
6 Factoriseer M. Laat $p_1, p_2, .., p_r$ de priemdelers van M zijn.
7 Bereken $(\frac{M}{p_i})P$ voor $i = 1, .., r$. Als $(\frac{M}{p_i})P = \infty$ voor een zekere i, sla M op als $\frac{M}{p_i}$ en herhaal stap 7. Als $(\frac{M}{p_i})P \neq \infty$ voor alle i, dan is M de orde van P.

Voorbeeld 3.3. We bekijken de kromme $E(\mathbb{F}_{601})$ die voldoet aan $y^2 = x^3 + 37x + 42$. We berekenen de orde voor het punt $P = (517, 507)$, waaruit $Q = (601 + 1)P = (539, 191)$ volgt. En voor m kiezen we $m = 5$ want $\sqrt[4]{601} \approx 4.95$.

Dan berekenen we als eerste de veelvouden van P voor $i = 0, ..., 5$. Dit geeft de punten:

$1P$	$2P$	$3P$	$4P$	$5P$
$(517, 507)$	$(178, 446)$	$(531, 345)$	$(570, 278)$	$(234, 509)$

Nu berekenen we veelvouden van P opgeteld bij onze gevonden Q. Dit geeft

$Q - 5(2 \cdot 5P)$	$Q - 4(2 \cdot 5P)$
$(307, 219)$	$(570, 278)$

We hoeven niet meer verder te rekenen, en weten nu dat de orde van P een deler is van $(601+1)+-4 \cdot 2 \cdot 5 - 4 = 558 = 2 \cdot 3 \cdot 3 \cdot 31$. We testen welke factorisaties mogelijk zijn: $\frac{558}{2}P = 279P = \infty$. Zo kunnen we nog even doorgaan: $\frac{279}{3}P = 93P = \infty$, $\frac{93}{3}P = 31P = \infty$. En omdat 31 priem is, kunnen we niet verder factoriseren en volgt dat de orde van $P = (517, 507)$ gelijk is aan 31.

3.3.2 Baby Step, Giant Step om de orde van een elliptische kromme te bepalen

Als je van een aantal punten op E de orde weet, kun je kijken naar het kleinste gemene veelvoud van de ordes. Als die binnen het domein beschreven in de stelling van Hasse (stelling 2.20) valt, heb je de orde van E gevonden.

Welnu volgt een stapsgewijze uitleg over hoe je met Baby Step, Giant Step de orde van $E(\mathbb{F}_q)$ berekent.

1 Bereken de boven- en ondergrenzen $q + 1 \pm 2\sqrt{q}$ en zet $kgv := 1$.
2 Kies een willekeurig punt P.

3 Bereken de orde van P met Baby Step, Giant Step.
4 Vervang kgv door het kleinste gemene veelvoud van kgv en de orde van P.
5 Deel de boven- en ondergrenzen door kgv en rond de bovengrens naar beneden en de ondergrens naar boven af.
6 i Als deze waarden aan elkaar gelijk zijn, is de orde van je elliptische kromme gelijk aan de orde van P vermenigvuldigd met de overeenkomende factor. In dit geval ben je nu klaar.
ii Indien dit niet zo is, herhaal stap 2 t/m 6.

Opmerking. Herhaal stap **6ii** niet te vaak! Er zijn gevallen waar dit oneindig lang door kan gaan. In ons algoritme is mee te geven hoe vaak je maximaal de orde van een willekeurig punt berekent, in onze checkfunctie doen we dit maximaal 5 keer. Om stap **6ii** niet oneindig lang door te laten gaan, voegen we de stelling van Mestre (sectie 3.3.3) toe. Hier zal uit worden gelegd in welke gevallen deze van toepassing is.

We zetten nu voorbeeld 3.3 voort om de orde van de elliptische kromme $E(\mathbb{F}_{601})$ die voldoet aan $y^2 = x^3 + 37x + 42$ te bepalen.

Voorbeeld 3.4. Zoals we in het voorgaande voorbeeld hebben gezien, is de orde van het willekeurige punt $P = (517, 507)$ gelijk aan 31.

We willen eerst kijken of dit punt binnen het domein van Hasse valt. De ondergrens van het domein is 553 want $601 + 1 - 2\sqrt{601} \approx 552.97$ en de bovengrens is 651 omdat $601 + 1 - 2\sqrt{601} \approx 651.03$. Nu delen we deze grenzen door de gevonden orde van het punt P en als deze overeenkomen, hebben we de orde van $E(\mathbb{F}_{601})$ gevonden. De ondergrens vertaalt naar 18 want $553/31 \approx 17.84$ en de bovengrens geeft 21 want $651/31 = 21$. Dit betekend dat er meerdere veelvouden van 31 binnen het Hasse domein vallen, en dat we dus nog geen eenduidige orde hebben gevonden.

We kiezen een nieuw willeurig punt P, neem $P = (163, 2)$. Dan volgt uit Baby Step, Giant Step dat de orde van dit punt 279 is. De kgv met de orde van het vorige punt wordt 279.

Dan delen we nu de domeingrenzen door dit getal. De ondergrens wordt 2 want $553/279 \approx 1.98$ en dit komt overeen met de bovenbegrenzing, die ook 2 is (want $651/279 \approx 2.33$).

Hieruit volgt dan dat $2 \cdot 279 = 558$ de orde van $E(\mathbb{F}_{601})$ is.

3.3.3 De stelling van Mestre

Een kleine toevoeging aan het Baby Step, Giant Step algoritme voor het vinden van de orde van een elliptische kromme, is de stelling van Mestre. Er zijn namelijk gevallen waar de kromme E isomorf is aan de directe som $\mathbb{Z}_m \oplus \mathbb{Z}_M$ met $m \mid M$ en $M < 4\sqrt{q}$. Hierdoor komt de hoogst mogelijke orde van een punt P minstens twee keer voor in het Hasse-domein van E. En dus is de orde van E op de hierboven beschreven manier van het Baby Step, Giant Step algoritme niet te bepalen.

Voor deze gevallen stelt Mestre dat dit voor een gerelateerde kromme, de kwadratische twist, wel kan.

Zoals we in paragraaf 3.2 zagen, kan er voor elk element $x_0 \in \mathbb{F}_q$ gekeken worden of er een punt (x_0, y_0) op E bestaat. Zodra $x_0^3 + Ax_0 + B$ geen kwadraat in $\mathbb{F}_q$ is, is er geen punt op E te vinden. Door $x^3 + Ax + B$ te vermenigvuldigen met een niet-kwadraat g in $\mathbb{F}_q$ (ook wel, een willekeurig element uit de verzameling $\mathbb{F}_q^\times := \{g \in \mathbb{F}_q \mid \forall x \in \mathbb{F}_q : g \neq x^2\}$), ontstaat er wel een kwadraat, namelijk $g(x^3 + Ax + B)$. Dit betekent dat de elementen x uit $\mathbb{F}_q$ die niet corresponderen met een punt op E, wel voldoen aan de vergelijking $gy^2 = x^3 + Ax + B$.

De punten die voldoen aan deze vergelijking, zijn onafhankelijk van de keuze van g. Dit bewijzen we in gevolg 3.6.

De officiële definitie van een kwadratische twist wordt gehanteerd in Weierstrass vorm. We bewijzen in gevolg 3.5 dat deze definitie equivalent is aan $gy^2 = x^3 + Ax + B$.

Definitie 3.2. Zij E een elliptische kromme over het lichaam $\mathbb{F}_q$ die voldoet aan de vergelijking $y^2 = x^3 + Ax + B$. Voor een niet-kwadraat $g \in \mathbb{F}_q^\times$, is de *kwadratische twist van* E de elliptische kromme E' die voldoet aan de vergelijking $y^2 = x^3 + Ag^2x + Bg^3$.

Gevolg 3.5. *Definitie 3.2 is equivalent met de voorheen besproken punten (x, y) die voldoen aan de vergelijking $gy^2 = x^3 + Ax + B$ met dezelfde $g \in \mathbb{F}_q^\times$.*

Bewijs. Elk punt (x_0, y_0) op $gy^2 = x^3 + Ax + B$ correspondeert met het punt $(x_1, y_1) := (gx_0, g^2x_0)$ dat aan $y^2 = x^3 + Ag^2x + Bg^3$ voldoet. Er geldt namelijk $y_1^2 = (g^2y_0)^2 = g^4y_0^2 = g^3(x_0^3 + Ax_0 + B) = (gx_0)^3 + Ag^2(gx_0) + Bg^3 = x_1^3 + Ag^2x_1 + Bg^3$. □

Het idee van de kwadratische twist stoelt zich op de twee-delige partitie van $\mathbb{F}_q$ in kwadraten en niet-kwadraten. Dit werkt alleen voor de niet-kwadraten, dus de elementen uit $\mathbb{F}_q^\times$. Voor de kwadraten geldt namelijk dat de vergelijking $y^2 = x^3 + Ak^2x + Bk^3$ (waarbij k een kwadraat in $\mathbb{F}_q$ is) een vervormde variant van de kromme E met vergelijking $y^2 = x^3 + Ax + B$ is. Alle punten (x, y) in E worden namelijk van de vorm (kx, kly) waarbij $l^2 = k$ in $\mathbb{F}_q$.

Definitie 3.3. De elliptisiche krommen E en $\tilde{E}$ zijn *isomorf* als er een bijectief homomorfisme $\gamma \colon E \to \tilde{E}$ bestaat.

Gevolg 3.6. *Alle kwadratische twists van een elliptische kromme E zijn isomorf.*

Bewijs. Kies twee willekeurige niet-kwadraten $g, h \in \mathbb{F}_q^\times$. Stel E'_g is de kwadratische twist van E die voldoet aan de vergelijking $y^2 = x^3 + Ag^2x + Bg^3$ en E'_h is de kwadratische twist van E die voldoet aan $y^2 = x^3 + Ah^2x + Bh^3$. Dan willen we laten zien dat er een bijectief endomorfisme $\gamma : E'_g \to E'_h$ bestaat, zodat voor elk punt $(x_g, y_g) \in E'_g$, het punt $\gamma((x_g, y_g))$ bevat is in E'_h.

Neem nu het homomorfisme γ waarbij

$$\begin{cases} (x, y) \mapsto (hg^{-1}x, h^{3/2}g^{-3/2}y) \\ \infty \mapsto \infty \end{cases}$$

Dan willen we hier de volgende 4 eigenschappen over bewijzen:

1. $\gamma\colon E'_g \to E'_h$
2. $h^{3/2}g^{-3/2}$ is bevat in $\mathbb{F}_q$
3. $\gamma(P+Q) = \gamma(P) + \gamma(Q)$ voor alle $P, Q \in E'_g$
4. Er bestaat een inverse voor γ

Welnu zullen we deze opsomming bewijzen. We nemen hierbij standaard aan dat $k^2h = g$ voor een zekere $k \in \mathbb{F}_q$. Dit volgt omdat g en h niet-kwadraten zijn in $\mathbb{F}_q$.

1. We tonen als eerste aan dat $h^{3/2}g^{-3/2} = hg^{-1}k^{-1}$.

$$\begin{aligned} h^{3/2}g^{-3/2} &= h^{3/2}g^{-1/2}g^{-1} \\ &= h^{3/2}(k^2h)^{-1/2}g^{-1} \\ &= h^{3/2}h^{-1/2}g^{-1}k^{-1} \qquad \text{(commutatief want } \mathbb{F}_q \text{ is een lichaam)} \\ &= hg^{-1}k^{-1} \end{aligned}$$

Nu geldt dat voor een punt $(x, y) \in E(\mathbb{F}_q)$ dat na de transformatie $\gamma(x, y) = (hg^{-1}x, h^{3/2}g^{-3/2}y)$, het volgende equivalent is:

$$\begin{aligned} (h^{3/2}g^{-3/2}y)^2 &= (hg^{-1}x)^3 + Ah^2(hg^{-1}x) + Bh^3 \\ (hg^{-1}k^{-1}y)^2 &= (hg^{-1}x)^3 + Ah^2(hg^{-1}x) + Bh^3 \\ h^2g^{-2}k^{-2}y^2 &= h^3g^{-3}x^3 + Ah^3g^{-1}x + Bh^3 \\ g^{-2}k^{-2}y^2 &= hg^{-3}x^3 + Ahg^{-1}x + Bh \\ k^{-2}y^2 &= hg^{-1}x^3 + Ahgx + Bhg^2 \\ y^2 &= k^2hg^{-1}x^3 + Ak^2hgx + Bk^2hg^2 \\ y^2 &= gg^{-1}x^3 + Ag^2x + Bg^3 \\ y^2 &= x^3 + Ag^2x + Bg^3 \end{aligned}$$

En daarom geldt $\gamma(x, y) \in E'_h$.

2. We kunnen hiervoor ook aantonen dat $h^{1/2}g^{-1/2}$ bevat is in $\mathbb{F}_q$. Dan volgt uit $h, g \in \mathbb{F}_q^\times$ dat er een $k \in \mathbb{F}_q$ bestaat zodanig dat $k^2h = g$. Dus geldt $g^{1/2}h^{-1/2} = (k^2h)^{1/2} \cdot h^{-1/2} = kh^{1/2}h^{-1/2} = k \in \mathbb{F}_q$.

3. Voor het behoud van de optelling, moeten we onderscheid maken tussen de vier verschillende gevallen bij optelling. Neem aan dat $P = (x_1, y_1)$ en $Q = (x_2, y_2)$.

i Voor $x_1 \neq x_2$ volgt $\gamma(P+Q) = \gamma((m_1^2 - x_1 - x_2, m_1(x_1 - x_3) - y_1))$ waarbij $m_1 = \frac{y_2 - y_1}{x_2 - x_1}$ en x_3 de

x-coördinaat van onze uitkomst is. Dit is gelijk aan

$$(hg^{-1}(m_1^2 - x_1 - x_2), h^{3/2}g^{-3/2}(m_1(x_1 - x_3) - y_1)).$$

Voor $\gamma(P) + \gamma(Q)$ volgt $(hg^{-1}x_1, h^{3/2}g^{-3/2}y_1) + (hg^{-1}x_2, h^{3/2}g^{-3/2}y_2)$. Ten eerste, er geldt $m_1' = \frac{h^{3/2}g^{-3/2}y_2 - h^{3/2}g^{-3/2}y_1}{hg^{-1}x_2 - hg^{-1}x_1} = h^{1/2}g^{-1/2}m_1$. Dan volgt (met x_3' zijnde de x-coördinaat van de som):

$$\gamma(P) + \gamma(Q) = (m_1'^2 - hg^{-1}x_1 - hg^{-1}x_2, m_1'(hg^{-1}x_1 - x_3') - h^{3/2}g^{-3/2}y_1)$$

$$((h^{1/2}g^{-1/2})^2 m_1^2 - hg^{-1}x_1 - hg^{-1}x_2, h^{1/2}g^{-1/2}m_1(hg^{-1}x_1 - x_3') - h^{3/2}g^{-3/2}y_1)$$

$$(hg^{-1}(m_1^2 - x_1 - x_2), h^{3/2}g^{-3/2}(m_1(x_1 - x_3) - y_1)) = \gamma(P + Q).$$

ii Als $x_1 = x_2$ en $y_1 \neq y_2$ dan volgt $P+Q = \infty$ dus $\gamma(P+Q) = \infty$. Daarnaast volgt $hg^{-1}x_1 = hg^{-1}x_2$ en $h^{3/2}g^{-3/2}y_1 \neq h^{3/2}g^{-3/2}y_2$. Dus $\gamma(P) = \gamma(Q) = \infty$.

iii Als $P = Q$ en $y_1 \neq 0$ volgt $\gamma(P + Q) = \gamma((m_2^2 - 2x_1, m_2(x_1 - x_4) - y_1))$ waarbij $m_2 = \frac{3x_1^2 + Ag^2}{2y_1}$ en x_4 het x-coördinaat van het punt is. Door γ uit te werken, verkrijgen we het punt

$$(hg^{-1}(m_2^2 - 2x_1), h^{3/2}g^{-3/2}(m_2(x_1 - x_4) - y_1)).$$

Anderszijds volgt uit $P = Q$ en $y_1 \neq 0$ dat $\gamma(P) = \gamma(Q)$ en $h^{3/2}g^{-3/2}y_1 \neq 0$. Dan volgt $\gamma(P) + \gamma(Q) = (m_2'^2 - 2hg^{-1}x_1, m_2'(hg^{-1}x_1 - x_4') - h^{3/2}g^{-3/2}y_1)$ waarbij x_4' de x-coördinaat van het punt is en $m_2' = \frac{3(hg^{-1}x_1)^2 + Ah^2}{2h^{3/2}g^{-3/2}y_1} = \frac{3h^2g^{-2}x_1^2 + Ah^2g^{-2}g^2}{2h^{3/2}g^{-3/2}y_1} = g^{1/2}h^{-1/2}m_2$. Dus volgt

$$\gamma(P) + \gamma(Q) = (hg^{-1}m_2^2 - 2hg^{-1}x_1, h^{1/2}g^{-1/2}m_2(hg^{-1}x_1 - hg^{-1}x_4) - h^{3/2}g^{-3/2}y_1)$$

$$= (hg^{-1}(m_2^2 - 2x_1), h^{3/2}g^{-3/2}(m_2(x_1 - x_4) - y_1)) = \gamma(P + Q).$$

iv Als $P = Q$ en $y_1 = 0$ volgt $P + Q = \infty$ en dus $\gamma(P + Q) = \infty$. Verder volgt uit $\gamma(P) = \gamma(Q)$ en $h^{3/2}g^{-3/2}y_1 = 0$ dus volgt $\gamma(P) + \gamma(Q) = \infty$.

Zodoende blijft de optelling behouden onder γ.

4. De inverse voor γ is γ^{-1} waarbij $(x, y) \mapsto (gh^{-1}x, g^{3/2}h^{-3/2}y)$. Zodoende is γ bijectief.

En dus volgt dat twee willekeurige kwadratische twists isomorf zijn aan elkaar. □

Stelling 3.7. *Voor een elliptische kromme E en zijn kwadratische twist E' geldt $\#E(\mathbb{F}_q) + \#E'(\mathbb{F}_q) = 2(q + 1)$.*

Bewijs. Voor elke $x_0 \in \mathbb{F}_q$ geldt dat $x_0^3 + Ax_0 + B$ wel of niet een kwadraat is.

Als $x_0^3 + Ax_0 + B$ wel een kwadraat is, dan zijn er twee punten op E met x_0 als x-coördinaat.

Als $x_0^3 + Ax_0 + B$ geen kwadraat is, dan is $g(x_0^3 + Ax_0 + B)$ met $g \in \mathbb{F}_q^\times$ een kwadraat in $\mathbb{F}_q$. Dus geeft x_0 twee punten met x_0 als x-coördinaat op E'.

Als $x_0^3 + Ax_0 + B = 0$ volgt dat $x_0^3 + Ag^2x_0 + Bg^3 = 0$. Dus is het een punt op E en op E'.

Dit geeft een totaal van $2 \cdot \#\mathbb{F}_q = 2q$ punten. Hieraan voegen we de ∞-punten van E en E' aan toe en dan volgt $\#E(\mathbb{F}_q) + \#E'(\mathbb{F}_q) = 2(q+1)$. □

Stelling 3.8 (Stelling van Mestre)**.** *Zij p een priemgetal groter dan 229, en E een elliptische kromme op $\mathbb{F}_q$. Dan heeft óf E of zijn kwadratische twist E' een punt P waarvan de orde maar één veelvoud in het interval $(p+1-2\sqrt{p}, p+1+2\sqrt{p})$ heeft.*

Het bewijs hiervan gaan we niet behandelen. Het originele bewijs hiervoor kan gevonden worden in het paper van Schoof [6] in hoofdstuk 3, dat voortkomt uit sectie 7.4.3 van het boek van Cohen [1]. Dit bewijs is slechts voor $p > 457$. In Washington [8] staat het bewijs van de uitbreiding voor het geval waar p strikt groter dan 229 is.

3.3.4 Resultaten prestatietest

Welnu volgt een overzichtje van de resultaten van de prestatietesten. Het script voor het `Timecheck(n,m)` algoritme is te vinden in sectie 5.6.5. Zoals eerder benoemd, is `m` de macht m van het domein $[2^{m-1}, 2^m]$ waartussen het priemgetal q willekeurig gekozen wordt en is voor `n` standaard 10 gekozen. Hierdoor worden er per `m` 10 willekeurige krommen getest met waarden A en B binnen $\mathbb{F}_q$.

`m`	gemiddelde	max	min
10 t/m 19	$\leq$ 0,002	$\leq$ 0,01	0
21	0,005	0,01	0
23	0,006	0,01	0
25	0,008	0,01	0
27	0,018	0,04	0
29	0,028	0,05	0,01
31	0,085	0,15	0,02
33	0,166	0,25	0,06
35	0,219	0,45	0,03
37	0,541	0,99	0,27
39	0,899	2,39	0,07
41	2,229	3,98	0,97
43	3,624	7,82	1,28
45	7,963	13,3	2,42
47	14,341	28,05	3,94

49	33,639	62,02	11,45
50	63,387	116,57	23,32

De complexiteit van Baby Step, Giant Step ligt officieel op $\mathcal{O}(\sqrt[4]{q})$, maar ons algoritme ligt op $\mathcal{O}(\sqrt{q})$ omdat we niet zo efficient zoeken op overeenkomende punten. En daarnaast is de complexiteit van `BabyGiant(E,P,q)` gelijk aan $\sqrt{q}$, omdat er twee for-loops in elkaar zijn van lengte $2m$ en m.

3.4 Schoof's algoritme

Het algoritme van Schoof is een uitgebreid algoritme dat wint in snelheid voor de krommen over eindige lichamen met grote q. Het maakt vooral gebruik van de stelling van Hasse (stelling 2.20), het spoor van Frobenius (stelling 2.21), Division Polynomials (sectie 2.4) en de Chinese Reststelling.

Het idee van René Schoof is simpel. We weten dat $\#E(\mathbb{F}_q) = q + 1 - a$ waarbij $|a| \leq 2\sqrt{q}$. We kunnen a berekenen aan de hand van stelling 2.21. Deze stelt dat $\Phi_q^2 - a\Phi_q + q = 0$. Dus voor een punt (x, y) op een kromme E geldt $(x^{q^2}, y^{q^2}) + q(x, y) = a(x^q, y^q)$.

Als we dan $a \pmod l$ weten voor priemgetallen l uit een verzameling $S := \{2, 3, .., L\}$ waarbij $\prod_{l \in S} l > 4\sqrt{q}$, kunnen we $a \pmod{\prod_{l \in S} l}$ berekenen. Zodoende vinden we een unieke a zodat $q + 1 - a$ met zekerheid binnen het domein $(q + 1 - 2\sqrt{2}, q + 1 - 2\sqrt{2})$ valt. Dit interval heeft namelijk lengte $4\sqrt{q}$ en volgens de stelling van Hasse (stelling 2.20) bestaat er altijd zo'n a.

Om a modulo een zekere priem l te berekenen, berekenen we $(x^{q^2}, y^{q^2}) + q_l(x, y) = a(x^q, y^q)$ voor een willekeurig $E[l]$-torsie punt (x, y). We weten dat geldt $(x, y) \in E[l]$ dan en slechts dan als $\psi_l(x) = 0$. Dus kunnen we de ψ_l-vouden uit alle berekeningen met x en y halen. Verder gebruiken we $q_l \equiv q \pmod l$ met $|q_l| < \frac{l}{2}$ in plaats van q, omdat $[l](x, y) = \infty$, en een kleiner getal sneller rekent.

Allereerst, leggen we het algoritme stapsgewijs uit (zoals hij ook wordt uitgevoerd in het MAGMA script in sectie 5.7.1). Daarna volgen wat opmerkingen die verklaren waarom het algoritme is geïmplementeerd zoals het is.

3.4.1 Algoritme

1 Bereken de verzameling S van oneven priemgetallen, zodat $\prod_{l \in S - \{2\}} l > 2\sqrt{q}$.
2 Bereken $a \pmod 2$ door $ggd(x^3 + Ax + B, x^q - x)$ te berekenen.

- Als deze $ggd = 1$, dan volgt dat $a \pmod 2 \equiv 1$
- Als deze $ggd \neq 1$, dan volgt dat $a \pmod 2 \equiv 0$

Herhaal stap 3 tot en met stap 5 voor alle $l \in S_{\geq 3}$

3 **a** Bereken $q_l \equiv q \pmod{l}$ met $|q| < \frac{l}{2}$

b Neem aan dat (x, y) een punt in $E[l]$ is. Bereken het x-coördinaat x^{q^2} van (x^{q^2}, y^{q^2}) modulo ψ_l en het x-coördinaat $q_l x$ van $q_l(x, y)$ modulo ψ_l.

c Bereken $ggd(x^{q^2} - q_l x, \psi_l)$.

- Als deze ggd ongelijk is aan 1 of zichzelf, vervang ψ_l dan met deze ggd
- Als ggd wel gelijk is aan 1, laat ψ_l zoals hij is

d Als x^{q^2} en $q_l x$ niet overeen komen, ga naar stap 4. Als x^{q^2} wel overeenkomt met $q_l x$, ga naar stap 5.

4 Bereken het x-coördinaat van $(x^{q^2}, y^{q^2}) + q_l(x, y)$ modulo ψ_l volgens de forumule in definitie 2.3 (we refereren vanaf nu naar dit coördinaat als x') en bereken het bijbehorende y-coördinaat (waarnaar we vanaf nu refereren als y'). Berekenen voor $j = 1, 2, .., \frac{l-1}{2}$ vervolgens het x-coördinaat x_j van $j(x, y)$.

i Als $x' - x_j^q \not\equiv 0 \pmod{\psi_l}$, herhaal deze stap voor $j + 1$.

ii Als $x' - x_j^q \equiv 0 \pmod{\psi_l}$, bereken dan het y-coordinaat y_j van $j(x, y)$ en daarna $\frac{y' - y_j^q}{y}$.

- Als $\frac{y' - y_j^q}{y} \equiv 0 \pmod{\psi_l}$, dan volgt dat $a \equiv j \pmod{l}$
- Als $\frac{y' - y_j^q}{y} \not\equiv 0 \pmod{\psi_l}$ volgt $a \equiv -j \pmod{l}$

Als je alle $l \in S$ hebt gehad, ga naar stap 6.

5 Als x^{q^2} en $q_l x$ gelijk aan elkaar zijn, dan kunnen 2 gevallen voorkomen:

A Als er een $w \in \mathbb{F}_q$ bestaat zodanig dat $w^2 \equiv q \pmod{l}$, bereken dan $ggd(teller(x^q - x_w), \psi_l)$.

- Als $ggd(teller(x^q - x_w), \psi_l) \neq 1$, bereken $ggd(teller(\frac{y^q - y_w}{y}, \psi_l)$
  - * Als $ggd(teller(\frac{y^q - y_w}{y}, \psi_l) = 1$, dan volgt dat $a \equiv -2w \pmod{l}$
  - * Als $ggd(teller(\frac{y^q - y_w}{y}, \psi_l) \neq 1$, volgt $a \equiv 2w \pmod{l}$
- Als $ggd(teller(x^q - x_w), \psi_l) = 1$ dan geldt $a \equiv -2w \pmod{l}$.

B Als voor alle $w \in \mathbb{F}_q$ geldt dat $w^2 \not\equiv q \pmod{l}$, dan volgt $a \equiv 0 \pmod{l}$.

Als je alle $l \in S$ hebt gehad, ga naar stap 6.

6 Als $a \pmod{l}$ berekend is voor elke $l \in S$, bereken $a \pmod{\prod l}$ met de Chinese Reststelling. Zorg hierbij dat $|a| \leq 2\sqrt{q}$. Dan volgt dat de kardinaliteit $\#E(\mathbb{F}_q) = q + 1 - a$.

3.4.2 Kanttekeningen bij algoritme

stap 1 We rekenen 2 automatisch bij de verzameling S, dus we zijn sneller klaar door $\prod_{l \in S} l > 4\sqrt{q}$ te vervangen met $\prod_{l \in S - \{2\}} l > 2\sqrt{q}$.

stap 2 Om $a \pmod{2}$ te bepalen, gebruiken we een simpel trucje. Voor alle nulpunten w van $x^3 + Ax + B$ waarbij $w \in \mathbb{F}_q$ geldt dat $(w, 0) \in E[2]$. Daarnaast weten we dat alle nulpunten van het polynoom $x^q - x$ precies de punten uit $\mathbb{F}_q$ zijn. Dus als deze twee polynomen een overeenkomend nulpunt w hebben, dan geldt $(w, 0) \in E[2]$ en dus $2 \mid \#E(\mathbb{F}_q)$.

We berekenen dit met $ggd(x^q - x, x^3 + Ax + B)$. Omdat x^q heel groot kan zijn, berekenen we $x_q \equiv (x^q - x) \pmod{(x^3 + Ax + B)}$. Zodoende is $ggd(x^q - x, x^3 + Ax + B) = ggd(x_q, x^3 + Ax + B)$ veel makkelijker te berekenen. Als deze ggd ongelijk aan 1 is, geldt dus dat $\#E(\mathbb{F}_q)$ even is, en dus $a \equiv 0 \pmod 2$. Als deze ggd wel gelijk is aan 1, volgt $a \equiv 1 \pmod 2$.

stap 3c We berekenen het verschil tussen $x^{q^2} - q_l x$. Dit is namelijk de noemer van de optelling van twee verschillende coördinaten. Als deze geen gemeenschappelijke factor met ψ_l hebben (en de ggd dus 1 is), dan kun je een inverse vinden voor de noemer en is de optelling dus zinvol. Als ze wél een gemeenschappelijke factor hebben, betekent dat dat je een deler van ψ_l hebt gevonden die bevat is in $\mathbb{F}_q[x]$.

Deze deler van ψ_l mag je gebruiken omdat $\psi_l = \prod_i (x - x_i)$ waarbij x_i een x-coördinaat is van een l-torsiepunt. We bekijken dus een aantal l-torsiepunten over een kleinere uitbreiding van $\mathbb{F}_q$. Hierdoor is de rekentijd sneller.

Wat dit betekent voor de $(x^{q^2}, y^{q^2}) + q_l(x, y)$ is dat het x-coördinaat modulo ψ_l gelijk is aan 0. Hierdoor geldt $(x^{q^2}, y^{q^2}) = \pm q_l(x, y)$, en dus zal het altijd naar stap **??** gaan.

stap 3d We maken onderscheid tussen de gevallen $(x^{q^2}, y^{q^2}) = \pm q(x, y)$ en $(x^{q^2}, y^{q^2}) \neq \pm q(x, y)$. Dit leggen we allemaal iets beter uit bij de verklaring voor stap 5.

stap 4 Voor het geval dat $(x^{q^2}, y^{q^2}) \neq \pm q(x, y)$ kunnen we deze punten bij elkaar optellen en alle $k(x^q, y^q)$ afgaan tot we een k gevonden hebben waarvoor $(x^{q^2}, y^{q^2}) + q(x, y) = k(x^q, y^q)$. Deze $k \equiv a \pmod l$.

stap 5 In het geval dat $(x^{q^2}, y^{q^2}) = \pm q(x, y)$ maken we onderscheid tussen twee gevallen:

i $(x^{q^2}, y^{q^2}) = q(x, y)$. In dit geval volgt

$$a\Phi_q(x, y) = \Phi_q^2(x, y) + q(x, y) = 2q(x, y).$$

En dus

$$(2q)^2(x, y) = a^2\Phi_q^2(x, y) = a^2 q(x, y).$$

Dus volgt dat q en kwadraat modulo l is. Zij $w^2 = q \pmod l$, dan volgt $(\Phi_q + w)(\Phi_q - w)(x, y) = (\Phi_q^2 - q)(x, y) = \infty$. Als $(x, y) =: P \in E[l]$, volgt $(\Phi_q - w)P = \infty$ of er is een $P' = (\Phi_q - w)P$ en voor deze P' geldt $(\Phi_q + w)P = \infty$. Hoe dan ook, bestaat er een $P \in E[l]$ met $\pm wP = \Phi_q P$.

- ⋆ Als $\Phi_q P = wP$ volgt $\infty = (\Phi_q^2 - a\Phi_q + q)P = (q - aw + q)P \Rightarrow aw \equiv 2q \equiv 2w^2 \pmod l \Rightarrow a \equiv 2w \pmod l$.
- ⋆ Als $\Phi_q P = -wP$ volgt $\infty = (\Phi_q^2 + a\Phi_q + q)P = (q - aw + q)P \Rightarrow -aw \equiv 2q \equiv 2w^2 \pmod l \Rightarrow a \equiv -2w \pmod l$.

ii $(x^{q^2}, y^{q^2}) = -q(x, y)$. Dan volgt $(x^{q^2}, y^{q^2}) + q(x, y) = \infty$, dus $a \equiv 0 \pmod{l}$.

Bovenstaande gevallen kunnen niet optreden als voor alle $w \in \mathbb{F}_q$ geldt dat $w^2 \not\equiv q \pmod{l}$, omdat er geen w bestaat zodanig dat $w^2 \equiv q \pmod{l}$). Daaruit volgt dat $(x^{q^2}, y^{q^2}) = -q_l(x, y)$ en dus dat $(x^{q^2}, y^{q^2}) + q_l(x, y) = \infty$.

Zoals we in het geval **i** hebben gezien, volgt uit $(x^{q^2}, y^{q^2}) = q(x, y)$ dat er een w bestaat zodanig dat $w^2 \equiv q \pmod{l}$). Daaruit volgt $\Phi_q P = \pm wP$ waardoor geldt $x^q = x_w$. Dus $(x^{q^2}, y^{q^2}) = -q(x, y)$ kan alleen gelden als $x^q \neq x_w$. Dus $ggd(deler(x^q - x_w), \psi_l) = 1$ impliceert dat $(x^{q^2}, y^{q^2}) = -q(x, y)$.

Dit algoritme werkte Schoof uit in zijn artikels *Elliptic curves over finite fields and the computation of square roots mod p* in 1985 [5] en *Counting points on elliptic curves over finite fields* in 1995 [6]. In latere jaren zijn Arthur Atkin en Noam Elkies bezig geweest met computationele verbeteringen aan te brengen aan het algoritme van Schoof. We verwijzen naar het paper van Elkies [2], het paper van Izu, Kogure, Noro en Yokoyama [4] en hoofdstuk 6 in het paper van Schoof [6] voor de theorie.

3.4.3 Resultaten prestatietest

Welnu volgt een overzichtje van de resultaten van de prestatietesten. Het script voor het `Timecheck(n,m)` algoritme is te vinden in sectie 5.7.3. Zoals eerder benoemd, is `m` de macht m van het domein $[2^{m-1}, 2^m]$ waartussen het priemgetal q willekeurig gekozen wordt en is voor `n` standaard 10 gekozen. Hierdoor worden er per `m` 10 willekeurige krommen getest met waarden A en B binnen $\mathbb{F}_q$.

`m`	gemiddelde	max	min
23	0,012	0,02	0,01
27	0,072	0,08	0,06
31	0,189	0,26	0,13
35	0,33	0,41	0,26
39	0,351	0,43	0,26
43	0,355	0,62	0,22
47	0,568	0,7	0,42
51	0,675	0,9	0,49
55	1,294	1,59	0,91
59	1,482	2,01	0,92
63	2,822	3,24	1,51
67	4.864	7.640	3.520
71	4.726	6.830	2.820
75	9.268	12.250	5.900

79	7.517	9.920	4.190
83	14.931	18.480	9.900
87	15.041	21.490	7.980
91	17.561	25.080	9.840
95	28.789	34.000	22.960
99	32.469	43.450	23.120
103	33.198	42.580	25.590
107	47.768	74.570	28.330

De complexiteit van het alrogitme van Schoof is $\mathcal{O}(\log_8(q))$. In ons geval zou dit dus $\mathcal{O}(\log_8(2^m))$ zijn.

3.5 Vergelijken prestatietesten

Om de resultaten en complexiteiten van de drie geteste algoritmes te vergelijken, leggen we ze naast elkaar. Hier kun je zien welke algoritmes beter scoren op wat voor getallen. We bekijken alleen de gemiddelden per interval $[2^{m-1}, 2^m]$ waarbinnen de priemgetallen q voor $\mathbb{F}_q$ vallen.

`m`	Legendre	Baby Step, Giant Step	Schoof
10	0	0	0
12	0,004	0	0,001
14	0,018	0	0,003
16	0,077	0	0,004
18	0,319	0,002	0,003
20	1,249	0,003	0,008
22	4,909	0,004	0,009
24	20,613	0,006	0,026
26	72,323	0,014	0,044
28	270,947	0,022	0,069
30	-	0,045	0,089
32	-	0,138	0,201
34	-	0,255	0,226
36	-	0,492	0,322
38	-	0,986	0,363

40	-	2,036	0,38
42	-	3,648	0,288
44	-	7,556	0,509
46	-	8,488	0,545
48	-	27,681	0,626
50	-	63,387	0,632

Voor een q kleiner dan 2^{15} is de methode van Legendre een goede methode. Het is makkelijk te implementeren en werkt voor kleine q dus vlug.

Voor een q tussen 2^{15} en 2^{35} is Baby Step, Giant Step een geschikt algoritme. Het is een uitgebreider algoritme, maar de resultaten voor grotere q zijn schrikbarend sneller. Dit algoritme is in feite ook voor kleinere q sneller dan Legendre, maar veel meer werk om te implementeren.

Voor q vanaf 2^{35} is, net zoals in de tabel te zien is, Schoof de grote winnaar. Dit algoritme blijft onder de 1 sec. per kromme tot $q \geq 2^{51}$.

4 Bibliografie

Referenties

[1] Henri Cohen. *A course in computational algebraic number theory*, volume 138 of *Graduate Texts in Mathematics*. Springer-Verlag, Berlin, 1993.

[2] Noam Elkies, *Explicit isogenies*, preprint, 1991.

[3] Stephen H. Friedberg, Arnold J. Insel en Lawrence E. Spence, *Linear Algebra: Pearson New International Edition*, Fourth Edition. Pearson Education Limited, Edinburgh Gate, Harlow, 2014.

[4] T. Izu, J. Kogure, M. Noro, and K. Yokoyama, *Efficient Implementation of Schoof's Algorithm*. New York: Springer-Verlag, pp. 66-79, 1998.

[5] René Schoof. *Elliptic curves over finite fields and the computation of square roots mod p*. Math. Comp., 44(170):483-494, 1985.

[6] René Schoof. *Counting points on elliptic curves over finite fields*. J. Théorine des Nombres de Bordeaux, 7: 219-254, 1995.

[7] Joseph. H. Silverman. *The arithmetic of elliptic curves*, volume 106 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1986.

[8] Lawrence C. Washington. *Elliptic Curves, Number Theory and Cryptography*, the Second Edition. Taylor & Francis Group, LLC, Boca Raton, 2008

5 Appendix

5.1 Bewijs stelling 2.2

5.1.1 Deel 1, stap 1

```
1 R <A,B,y_1,x_1,y_2,x_2> := FunctionField(Rationals(), 6);
2 m_1 := (y_2-y_1)/(x_2-x_1);
3 x_3 := m_1^2-x_1-x_2;
4 y_3 := m_1*(x_3-x_1) + y_1;
5 y_3^2 - x_3^3 -A*x_3-B;
```

Hieruit volgt:

```
(-A*y_1^2*x_1+A*y_1^2*x_2+2*A*y_1*x_1*y_2-2*A*y_1*y_2*x_2+A*x_1^4-2*A*x_1^3*x_2-A*x_1*y_
2^2+2*A*x_1*x_2^3+A*y_2^2*x_2-A*x_2^4-B*x_1^3+3*B*x_1^2*x_2-3*B*x_1*x_2^2+B*x_2^3+y_1^4-2*y_
1^3*y_2-2*y_1^2*x_1^3+3*y_1^2*x_1*x_2^2-y_1^2*x_2^3+2*y_1*x_1^3*y_2+2*y_1*y_2^3-2*y_1*y_
2*x_2^3+x_1^6-3*x_1^4*x_2^2+x_1^3*y_2^2-3*x_1^2*y_2^2*x_2+3*x_1^2*x_2^4-y_2^4+2*y_2^2*x_
2^3-x_2^6)/(x_1^3-3*x_1^2*x_2+3*x_1*x_2^2-x_2^3)
```

5.1.2 Deel 1, stap 2

```
(-A*(x_1^3+A*x_1+B)*x_1+A*(x_1^3+A*x_1+B)*x_2+2*A*y_1*x_1*y_2-2*A*y_1*y_2*x_2+A*x_1^4-2*A*x_
1^3*x_2-A*x_1*(x_2^3+A*x_2+B)+2*A*x_1*x_2^3+A*(x_2^3+A*x_2+B)*x_2-A*x_2^4-B*x_1^3+3*B*x_
1^2*x_2-3*B*x_1*x_2^2+B*x_2^3+(x_1^3+A*x_1+B)^2-2*y_1*(x_1^3+A*x_1+B)*y_2-2*(x_1^3+A*x_
1+B)*x_1^3+3*(x_1^3+A*x_1+B)*x_1*x_2^2-(x_1^3+A*x_1+B)*x_2^3+2*y_1*x_1^3*y_2+2*y_1*y_2*(x_
2^3+A*x_2+B)-2*y_1*y_2*x_2^3+x_1^6-3*x_1^4*x_2^2+x_1^3*(x_2^3+A*x_2+B)-3*x_1^2*(x_2^3+A*x_
2+B)*x_2+3*x_1^2*x_2^4-(x_2^3+A*x_2+B)^2+2*(x_2^3+A*x_2+B)*x_2^3-x_2^6)
```

5.1.3 Deel 2

```
1 R <A,B,y_1,x_1> := FunctionField(Rationals(), 4);
2 m_2 := (3*x_1^2 + A)/(2*y_1);
3 x_3 := m_2^2-2*x_1;
4 y_3 := m_2*(m_2^2-3*x_1) + y_1;
5 y_3^2 - x_3^3 -A*x_3-B;
```

5.2 Optelfunctie

```
1 GroupLaw := function(E, e1, e2, phil)
```

```
2     x1 := e1[1];
3     y1 := e1[2];
4     x2 := e2[1];
5     y2 := e2[2];
6     C := Coefficients(E);
7     A := C[4];
8     B := C[5];
9     x := Parent(x1).1;
10    f := x^3 + A*x + B;
11    if (x1 ne x2) then
12       phil1 := Modinv(x2 - x1, phil);
13       m := (y2 - y1)*phil1;
14       x3 := f*m^2 - x1 - x2;
15       y3 := m*(x1 - x3) -y1;
16       z3 := 1;
17    elif ((x1 eq x2) and (y1 ne y2)) then
18       x3 := 0;
19       y3 := 1;
20       z3 := 0;
21    elif (e1 eq e2 and y1 ne 0) then
22       m := (3*x1*x1 + A)/(2*y1);
23       x3 := m^2 - 2*x1;
24       y3 := m *(x1 - x3) -y1;
25       z3 := 1;
26    elif ((e1 eq e2) and (y1 eq 0)) then
27       x3 := 0;
28       y3 := 1;
29       z3 := 0;
30    end if;
31    u := [x3 mod phil, y3, z3] ;
32    return u;
33 end function;
```

5.3 Division Polynomials

5.3.1 Division Polynomials

```
1  DivPoly := function(E, l)
2     x := PolynomialRing(BaseField(E)).1;
3     C := Coefficients(E);
4     A := C[4];
```

```
5     B := C[5];
6     f := x^3 + A*x +B;
7     phi := [];
8
9     if l eq 0 then
10       return 0;
11    end if;
12    //even altijd verm met y
13    phi[1] := x^0;
14    phi[2] := 2;
15    phi[3] := 3*x^4 + 6*A*x^2 + 12*B*x -A^2 ;
16    phi[4] := 4*(x^6 + 5*A*x^4 + 20*B*x^3 - 5*A*A*x^2 - 4*A*B*x -8*B^2 -A^3) ;
17
18    for i := 5 to l do
19
20       if (i mod 2) eq 1 then
21          m := (i-1) div 2 ;
22          if (m mod 2) eq 1 then
23             phi[i] := phi[m+2]*(phi[m])^3 - f^2 *phi[m-1]*(phi[m+1])^3 ;
24          elif (m mod 2) eq 0 then
25             phi[i] := f^2 *phi[m+2]*(phi[m])^3 - phi[m-1]*(phi[m+1])^3 ;
26          end if;
27       elif (i mod 2) eq 0 then // y^-1 weggelaten
28          m := i div 2 ;
29          phi[i] := (phi[m]*(phi[m+2]*(phi[m-1])^2 - phi[m-2]*(phi[m+1])^2)) / 2 ;
30          // /2 omdat het een polynoomring is en er dus echt gedeeld kan worden
31       end if;
32    end for;
33    return phi;
34 end function;
```

5.3.2 De ϕ_n functie

```
1  PhiFunct := function(E, i, phi, e);
2     x := e[1];
3     y := e[2];
4     if (i mod 2) eq 0 then
5        return phi[i] *y;
6     elif (i mod 2) eq 1 then
7        return phi[i];
8     end if;
```

```
 9 end function;
```

```
 1 PhiPoly := function(E, i, phi)
 2     x := PolynomialRing(BaseField(E)).1;
 3     C := Coefficients(E);
 4     A := C[4];
 5     B := C[5];
 6     f := x^3 + A*x +B;
 7     if i mod 2 eq 0 then
 8         ans := x*f*(phi[i])^2 -(phi[i+1]*phi[i-1]);
 9     elif i mod 2 eq 1 then
10         ans := x*(phi[i])^2 -f*(phi[i+1]*phi[i-1]);
11     end if;
12     return ans;
13 end function;
```

5.3.3 De ω_n functie

```
 1 OmegaPoly := function(E, l, phi)
 2     x := PolynomialRing(BaseField(E)).1;
 3     C := Coefficients(E);
 4     A := C[4];
 5     B := C[5];
 6     f := x^3 + A*x +B;
 7     ans := (phi[l+2] *phi[l-1]^2 - phi[l-2]*phi[l+1]^2) /4;
 8     return ans;
 9 end function;
```

5.3.4 Veelvoud van een punt berekenen

```
 1 XofnP := function(E, n, phi, l)
 2     x := PolynomialRing(BaseField(E)).1;
 3     n := Abs(n);
 4     C := Coefficients(E);
 5     A := C[4];
 6     B := C[5];
 7     f := x^3 + A*x +B;
 8     if n eq 0 then
 9         ans := 0;
10     elif n eq 1 then
11         ans := x;
```

```
12      elif n mod 2 eq 1 then
13          phin2 := Modinv((phi[n])^2, phi[l]);
14          ans := PhiPoly(E, n, phi)*phin2 ;
15      elif n mod 2 eq 0 then
16          phin2 := Modinv(f*(phi[n])^2, phi[l]);
17          ans := PhiPoly(E, n, phi)*phin2 ;
18      end if;
19      return ans;
20  end function;
```

```
1   YofnP := function(E, n, phi, l)
2       x := PolynomialRing(BaseField(E)).1;
3       C := Coefficients(E);
4       A := C[4];
5       B := C[5];
6       f := x^3 + A*x +B;
7       m := Abs(n);
8       if m eq 1 then
9           ans := 1;
10      elif m eq 2 then
11          phin3 := Modinv(f^2 *phi[m]^3, phi[l]);
12          ans := (phi[m+2]*(phi[m-1])^2)*phin3 /4;
13      elif n mod 2 eq 0 then
14          phin3 := Modinv(f^2*((phi[m])^3), phi[l]);
15          ans := OmegaPoly(E, m, phi)*phin3;
16      elif n mod 2 eq 1 then
17          phin3 := Modinv(phi[m]^3, phi[l]);
18          ans := OmegaPoly(E, m, phi)*phin3;
19      end if;
20      if (n ge 0) then
21          return ans;
22      else
23          return -ans;
24      end if;
25  end function;
```

5.4 Legendre symbool voor het tellen van punten

```
1   Legendresymbol := function(x)
2       if x eq 0 then
3           return 0;
```

```
4     elif IsSquare(x) then
5        return 1;
6     else
7        return -1;
8     end if;
9  end function;
10
11 Legendre := function(A, B, q)
12    F := GF(q);
13    counter := 0;
14    for x in GF(q) do
15       counter := counter + Legendresymbol(F!(x^3 + A*x +B));
16    end for;
17    return q + 1 + counter;
18 end function;
```

5.4.1 CPU tijd functie

```
1  Timecheck := function(n,m)
2     z := [];
3     for i := 1 to n do
4        q := NextPrime(Random([2^(m-1)..2^m]));
5        R := GF(q);
6        A := Random(R);
7        B := Random(R);
8        while ((4*A^3 +27*B^2) eq 0) or (-16*(4*A^3 + 27*B^2) eq 0) do
9           A := Random(R);
10          B := Random(R);
11       end while;
12       t0 := Cputime();
13       v := Legendre(A,B,q);
14       z[i] := Cputime(t0);
15    end for;
16    return z;
17 end function;
18
19 for i := 10 to 35 do
20    print i;
21    z := Timecheck(10, i);
22    print(z);
23    print(&+z/#z); print(Maximum(z)) ; print(Minimum(z));
```

```
24 end for;
```

5.5 Punten tellen voor een uitgebreider lichaam

```
1  HogereMacht := function(cardE, q, n)
2     a := q + 1 - cardE;
3     R<x> := PolynomialRing(QuadraticField(a^2 - 4*q));
4     f := x^2 -a*x + q;
5     alfa := Roots(f)[1][1];
6     if Roots(f)[1][2] ne 2 then
7        beta := Roots(f)[2][1];
8     else
9        beta := Roots(f)[1][1];
10    end if;
11    return q^n +1 - (alfa^n +beta^n);
12 end function;
```

5.6 Baby Step, Giant Step en Mestres Methode

5.6.1 BSGS voor de orde van een punt

```
1  BabyGiant := function(E, P, q)
2     Q := (q+1)*P;
3     m := Round(Root(q, 4) + 1/2);
4     a := [];
5     b := [];
6
7     for j := 0 to m do
8        a[j+1] := j*P;
9        if a[j+1] eq E![0,1,0] and j ne 0 then
10          return j ;
11       end if;
12    end for;
13
14    for k := -m to m do
15       b[k+m+1] := Q +k*(2*m*P);
16       for j := 0 to m do
17          if b[k+m+1] eq a[j+1] then
18             M := q+1+2*m*k-j;
19             break k ;
```

```
20            elif  b[k+m+1] eq -a[j+1] then
21               M := q+1+2*m*k+j;
22               break k ;
23            end if;
24         end for;
25      end for;
26
27      Z := PrimeDivisors(M);
28      for p in Z do
29         while (M mod p eq 0) and ((M div p)*P eq E ! [0, 1, 0]) do
30            M := M div p;
31         end while;
32      end for;
33      return M;
34   end function;
```

5.6.2 Mestre's algoritme

```
1   Mestre := function(E, q, max)
2      Et := QuadraticTwist(E);
3      upb := q+1 + 2*Sqrt(q);
4         lowb := q+1 - 2*Sqrt(q);
5         P := Random(Et);
6         while P eq Et![0, 1, 0] do
7            P := Random(Et);
8         end while;
9         ord := BabyGiant(Et, P, q);
10        kgv := ord ;
11        counter := 1 ;
12        lower := Round(lowb/kgv + 1/2);
13        upper := Round(upb/kgv - 1/2);
14
15        while (lower ne upper and counter le max) do
16           P := Random(Et);
17           while P eq Et![0, 1, 0] do
18              P := Random(Et);
19           end while;
20           counter := counter +1 ;
21           ord := BabyGiant(Et, P, q);
22           kgv := LCM(kgv, ord) ;
23           lower := Round(lowb/kgv + 1/2);
```

```
24            upper := Round(upb/kgv - 1/2);
25        end while;
26     if lower eq upper then
27        return lower*kgv;
28     else
29        return 2*(q+1);
30     end if;
31  end function;
```

5.6.3 Combineren van BSGS en Mestres stelling om de orde van $E(\mathbb{F}_q)$ te bepalen

```
1  BabyOrder := function(A, B, q, max);
2     E := EllipticCurve([GF(q)!A, GF(q)!B]);
3     upb := q+1 + 2*Sqrt(q);
4     lowb := q+1 - 2*Sqrt(q);
5     P := Random(E);
6     while P eq E![0, 1, 0] do
7        P := Random(E);
8     end while;
9
10    ord := BabyGiant(E, P, q);
11    kgv := ord ;
12    counter := 1 ;
13    lower := Round(lowb/kgv + 1/2);
14    upper := Round(upb/kgv - 1/2);
15
16    while (lower ne upper and counter le max) do
17       P := Random(E);
18       while P eq E![0, 1, 0] do
19          P := Random(E);
20       end while;
21       counter := counter +1 ;
22       ord := BabyGiant(E, P, q);
23       kgv := LCM(kgv, ord) ;
24       lower := Round(lowb/kgv + 1/2);
25       upper := Round(upb/kgv - 1/2);
26    end while;
27    if (lower eq upper) then
28       return lower*kgv;
29    else
30       mesord := 2*(q+1) - Mestre(E, q, 2*max);
```

```
31        if mesord eq 0 then
32           return $$(A, B, q, 4*max);
33        else
34           return mesord;
35        end if;
36     end if;
37  end function;
```

5.6.4 Controle functie

```
1  Checkcheckbaby := function(n, m);
2     for i := 1 to n do
3        print i;
4        q := RandomPrime(m);
5        while 229 ge q do
6           q := RandomPrime(m);
7        end while;
8        R := GF(q);
9        A := Random(R);
10       B := Random(R);
11       while ((4*A^3 +27*B^2) eq 0) or (-16*(4*A^3 + 27*B^2) eq 0) do
12          A := Random(R);
13          B := Random(R);
14       end while;
15       E := EllipticCurve([R!A, R!B]);
16       print q;
17       print Coefficients(E);
18       bool := BabyOrder(A, B, q, 5) eq Order(E);
19       if not bool then
20          print i;
21          print bool;
22          print Order(E);
23          print BabyOrder(A,B,q, 5);
24          break;
25       end if;
26    end for;
27    return true;
28 end function;
```

5.6.5 CPU tijd functie

```
1  Timecheck := function(n, m);
2     z := [];
3     for i := 1 to n do
4        q := NextPrime(Random([2^(m-1)..2^m]));
5        R := GF(q);
6        A := Random(R);
7        B := Random(R);
8        while ((4*A^3 +27*B^2) eq 0) or (-16*(4*A^3 + 27*B^2) eq 0) do
9           A := Random(R);
10          B := Random(R);
11       end while;
12       t0 := Cputime();
13       v := BabyOrder(A, B, q, 5);
14       z[i] := Cputime(t0);
15    end for;
16    return z;
17 end function;
18
19 for i := 10 to 50 do
20    print i;
21    z := Timecheck(10, i);
22    print(z);
23    print(&+z/#z); print(Maximum(z)) ; print(Minimum(z));
24 end for;
```

5.7 Schoof's algoritme

5.7.1 algoritme

```
1  Schoof := function(A, B, q);
2     E := EllipticCurve([GF(q)!A, GF(q)!B]);
3     S := [ 2, 3 ];
4     prod := 2*3;
5
6  //stap 1
7     while 4*Sqrt(q) ge prod do
8       i := NextPrime(Maximum(S));
9       prod *:= i;
10      S := S cat [ i ];
11    end while;
12    a := [ q+1 : i in [1..#S] ];  //vector w/ a mod(l), undefined a = q+1
```

```
13
14
15 //stap 2
16    R<x> := PolynomialRing(GF(q));
17    p := (Modexp(x, q,x^3 + A*x + B)  - x);
18    if GCD(p, x^3 + A*x + B) eq 1 then
19      a2 := 1 mod(2);
20    else
21      a2 := 0 mod(2);
22    end if;
23    a[1] := a2;
24
25 //stap 3
26    phi := DivPoly(E, Maximum(S));
27    for i in [2..#S] do
28       l := S[i];
29       ql := q mod(l);
30      // a. make |ql| < l/2
31       if Abs(ql) ge l/2 then
32          if ql ge 0 then
33             ql := ql -l;
34          else
35             ql := ql +l;
36          end if;
37       end if;
38       if ql eq 0 then
39          print q;
40          print l;
41          print ql;
42       end if;
43
44      // b. compute x-coordinate of x'
45       f := x^3 +A*x +B;
46       qlx := XofnP(E, ql, phi, l) mod phi[l];
47       qly := YofnP(E, ql, phi, l) mod phi[l];
48       qlP := [qlx, qly, 1];
49       xkwad := Modexp(x, q^2, phi[l]);
50       ykwad := Modexp(f, ((q^2)-1) div 2, phi[l]);
51       kwadP := [xkwad, ykwad, 1];
52       theta := GCD(xkwad - qlx, phi[l]);
53       bool3 := (Degree(theta) gt 0) and (Degree(theta) lt Degree(phi[l]));
```

```
54        if bool3 then
55           tempphil := phi[l];
56           phi[l] := theta;
57        end if;
58        if (qlx mod(phi[l]) eq xkwad mod(phi[l])) then
59           if IsSquare(GF(l)!q) then
60              w := Modsqrt(q, l);
61              xw := XofnP(E, w, phi, l);
62              xqmod := Modexp(x, q, phi[l]);
63              if GCD(xqmod - xw, phi[l]) eq 1 then
64                 a[i] := 0;
65              else
66                 yw := YofnP(E, w, phi, l);
67                 yqmod := Modexp(f, (q-1) div 2, phi[l]);
68                 if GCD((yqmod - yw), phi[l]) ne 1 then
69                    a[i] := 2*w;
70                 else
71                    a[i] := -2*w;
72                 end if;
73              end if;
74           else
75              a[i] := 0;
76           end if;
77        else //if qlx =/= xkwad
78           P := GroupLaw(E, qlP, kwadP, phi[l]); //(x',y') zoals in stap b
79           P[1] := P[1] mod phi[l];
80           P[2] := P[2] mod phi[l];
81         // c. compute y' and yj
82           for j := 1 to ((l-1)div 2) do
83              xj := XofnP(E, j, phi, l) mod phi[l];
84              bool := ( ((R!P[1] - Modexp(xj, q, phi[l])) mod(phi[l])) eq 0);
85              if bool then
86                 yj := YofnP(E, j, phi, l) mod phi[l];
87                 yjq := Modexp(f,(q-1)div 2, phi[l]) * Modexp(yj,q, phi[l]) mod phi[l];
88                 bool2 := (((P[2] - yjq) mod phi[l]) eq 0);
89                 if bool2 then
90                    a[i] := j;
91                    break j;
92                 else
93                    a[i] := -j;
94                    break j;
```

```
95                  end if;
96               end if;
97            end for;
98         end if;
99         if bool3 then
100           phi[l] := tempphil;
101        end if;
102     end for;
103     EltE := ChineseRemainderTheorem(a, S);
104     if EltE ge prod div(2) then
105        EltE := EltE - prod;
106     end if;
107     return q +1 - EltE ;
108  end function;
```

5.7.2 Controle functie

```
1  Checkcheck := function(n, m);
2     for i := 1 to n do
3        print i;
4        q := RandomPrime(m);
5        while 5 ge q do
6           q := RandomPrime(m);
7        end while;
8        R := GF(q);
9        A := Random(R);
10       B := Random(R);
11       while ((4*A^3 +27*B^2) eq 0) or (-16*(4*A^3 + 27*B^2) eq 0) do
12          A := Random(R);
13          B := Random(R);
14       end while;
15       E := EllipticCurve([R!A, R!B]);
16       bool := Schoof(A, B, q) ne Order(E);
17       if bool then
18          print bool;
19       end if;
20    end for;
21    return 0;
22 end function;
```

5.7.3 CPU tijd functie

```
1  Timecheck := function(n, m);
2     z := [];
3     for i := 1 to n do
4        q := NextPrime(Random([2^(m-1)..2^m]));
5        R := GF(q);
6        A := Random(R);
7        B := Random(R);
8        while ((4*A^3 +27*B^2) eq 0) or (-16*(4*A^3 + 27*B^2) eq 0) do
9           A := Random(R);
10          B := Random(R);
11       end while;
12       t0 := Cputime();
13       v := Schoof(A,B,q);
14       z[i] := Cputime(t0);
15    end for;
16    return z;
17 end function;
18
19 for i := 10 to 65 do
20    print i;
21    z := Timecheck(10, i);
22    print(z);
23    print(&+z/#z); print(Maximum(z)) ; print(Minimum(z));
24 end for;
```