Elliptische krommen cryptografie

Quirijn Meijer (4205197)

3 Juli 2015

Inhoud

1 Gebruiken van cryptografie

Cryptografie ontleent zijn belang aan de noodzaak gevoelige over te dragen informatie voldoende te beschermen van onbevoegde partijen, zodanig dat enkel verzender en ontvanger het bericht in kunnen zien. Om dit te bereiken wordt het bericht versleuteld door de verzender, waarna de ontvanger deze kan ontsleutelen. Het algoritme dat hiervoor gebruikt wordt moet zo gekozen worden dat zij garandeert dat eenieder anders dan de bestemde ontvanger het onderschept bericht niet kan ontsleutelen. Er wordt vooralsnog aangenomen dat dit geheel onmogelijk maken een onmogelijkheid in zichzelf is. In realiteit worden om deze reden de algoritmen zo ontworpen dat een bepaalde mate van veiligheid

gegarandeerd kan worden. Voor vele algoritmen houdt deze mate van beveiliging verband met de sleutel die gekozen wordt. Hierdoor kan de beveiliging van een boodschap naar behoeven gekozen worden. Dit betekent echter niet dat een dergelijke sleutel een pijler is voor de veiligheid van ieder willekeurig algoritme. De komende hoofdstukken in dit verslag zullen de theorie benodigd om op veilige wijze een gemeenschappelijke sleutel vast te stellen uit de doeken doen. Voor dit gebeurt volgen nog enkele voorbeelden.

Het dagelijks leven biedt vele voorbeelden van cryptografie. Op het internet wordt bij vrijwel iedere vorm van identificatie het verkeer versleuteld. Specifieke voorbeelden zijn het inloggen op websites, het doen van aankopen via het internet, of internetbankieren. Een voorbeeld buiten het internet om is het opnemen van geld bij een geldautomaat.

De Enigma machine, gebruikt door de Wehrmacht tijdens de tweede wereldoorlog, verdient een eervolle vermelding. Deze machine werd gebruikt om de oorlogsmachine op afstand te coördineren. Het breken van de encryptie van de Enigma met behulp van de Bombe is een klassiek voorbeeld van de samenkomst van cryptografie en de computer.

2 Het trapdoor principe

Een groot deel van de efficiëntie van een cryptografisch systeem berust op de eenvoud waarmee informatie versleuteld en ontsleuteld kan worden wanneer de benodigde parameters bekend zijn, en de moeilijkheid van de ontsleuteling wanneer deze deels of niet bekend zijn. Het doel van het versleutelen van informatie is immers het afschermen van de informatie voor partijen die niet specifiek toegang tot de informatie verleend zijn. Dit mag echter geen hinder vormen voor de vertrouwde partijen. Een cryptografisch systeem moet zodoende beschikken over de eigenschap dat zij eenvoudig toepasbaar is, maar lastig omkeerbaar door kwaadwillenden. Een functie die deze eigenschappen bezit wordt een trapdoorfunctie genoemd. Dit is een metafoor voor een werkelijk valluik: één richting volgen moet eenvoudig zijn, de tegengestelde richting niet. De trapdoor-functie in dit verslag zal het Diffie-Hellman protocol zijn, welke zich baseert op de theorie van elliptische krommen over eindige velden van priem orde. Het achterliggend probleem dat de veiligheid van dit protocol waarborgt is het elliptische kromme discreet logaritmisch probleem.

3 De theorie van elliptische krommen

Een elliptische kromme $\mathrm{K}(\mathbb{R})$ wordt gedefiniëerd door alle paren $(x, y) \in \mathbb{R}^2$ die voldoen aan de volgende vergelijking:

$$\mathrm{K}(\mathbb{R}) : y^2 = x^3 + ax + b$$

voor $a, b \in \mathbb{R}$, met als randvoorwaarde dat de discriminant

$$\Delta_{\mathrm{K}} = -16(4a^3 + 27b^2)$$

ongelijk is aan 0. Deze voorwaarde garandeert dat de kromme niet-singulier is. Als conventie wordt vervolgens een afzonderlijk punt O in oneindig toegevoegd. De punten op deze krommen kunnen nu onderling opgeteld worden, met als resultaat een som van de punten die zich wederom op de kromme bevindt. Evenzo kan een scalaire vermenigvuldiging met dezelfde eigenschap gedefiniëerd worden. Een belangrijke observatie is dat wanneer de coëfficiënten a, b en de coördinaten x, y van de punten op de kromme niet langer genomen worden uit $\mathbb{R}$, maar uit een eindig lichaam $\mathbb{F}_p$ van priem orde $p > 3$, deze eigenschappen behouden blijven. De resulterende grafiek van de kromme zal geheel anders van aangezicht zijn, en begrensd in het vlak. Het is nu mogelijk de verzameling punten op $\mathrm{K}(\mathbb{F}_p)$ samen met de operatie van optelling van punten te nemen om een groep $\mathbf{G} = (\mathrm{K}(\mathbb{F}_p), +)$ te vormen. Merk op dat niet iedere kromme voldoet aan de vereisten om bruikbaar te zijn voor cryptografische doeleinden, maar de veelal voorgeschreven krommen dit wel doen. Merk verder op dat $\mathbf{G}$ een cyclische groep is, of isomorf met het product van twee cyclische groepen. De groep is hierdoor abels. Een opsomming van deze groep is als volgt:

- Voor alle punten P, Q op de kromme $\mathrm{K}(\mathbb{F}_p)$ geldt dat de som $P + Q$ op de kromme ligt. De groep $\mathbf{G}$ is gesloten onder de groepsoperatie.

- De groepsoperatie is associatief. Ofwel, voor $P, Q, R \in \mathbf{G}$ geldt $P + (Q + R) = (P + Q) + R$.

- Het toegevoegde punt in oneindig wordt gerelateerd aan het eenheidselement $e_{\mathbf{G}}$.

- Voor iedere $P \in \mathbf{G}$ bestaat een element $P^{-1} \in \mathbf{G}$ zodat $P + P^{-1} = P^{-1} + P = e_{\mathbf{G}}$.

Deze groepseigenschappen zijn niet vanzelfsprekend, maar behoeven een te uitgebreid bewijs voor dit verslag. Het volstaat te concluderen dat een groep bestaat welke correspondeert met de kromme $\mathrm{K}(\mathbb{F}_p)$.

4 Het Diffie-Hellman protocol

Het Diffie-Hellman protocol is een systeem dat de uitwisseling van cryptografische sleutels tussen twee partijen beveiligt. Hier op volgend kan deze sleutel gebruikt worden om de communicatie tussenbeide te beveiligen.

Er wordt aangenomen dat de variabelen (a, b, p) die een elliptische kromme $\mathrm{K}(\mathbb{F}_p)$ definiëren publiek bekend zijn, en dat verzender en ontvanger overeenstemmen deze kromme te gebruiken. Daarnaast is een specifiek genererend element k van de groep $\mathbf{G}$ behorend bij deze kromme bekend bij beide partijen. De communicatie vindt vervolgens plaats zoals voorgeschreven door het volgende stappenplan.

1. De eerste partij kiest een getal $\lambda \in [1, p-1]$ en zendt de tweede partij het punt k^λ toe. De verzender beschikt nu over een sleutel (λ, k^λ) waarin λ de zogeheten privé-sleutel is, en k^λ de publieke sleutel.

2. De tweede partij kiest een getal $\mu \in [1, p-1]$ en stuurt de publieke sleutel k^μ als reactie.

3. De tweede partij berekent het punt $(k^\lambda)^\mu = k^{\lambda\mu}$.

4. De eerste partij berekent het punt $(k^\mu)^\lambda = k^{\mu\lambda}$, welke dankzij de commutativiteit van de groep $\mathbf{G}$ gelijk is aan $k^{\lambda\mu}$. Dit punt zal in het vervolg met C aangeduid worden.

Beide partijen beschikken nu over hetzelfde punt C. Omdat alleen de punten k^λ en k^μ gepubliceerd zijn is het voor een derde partij die een van beide onderschept onmogelijk C te berekenen zonder dat μ danwel λ bij deze partij bekend is. Deze partij kan pogen een van beiden uit de publieke sleutels te winnen, om hiermee C te berekenen. Het volgend hoofdstuk verklaart waarom deze aanpak geen vruchtbare onderneming is.

5 Het discreet logaritmisch probleem

Gegeven een punt l uit de groep $\mathbf{G}$ van orde n, en een genererend element $k \in \mathbf{G}$. Het discreet logaritmisch probleem is de naam gegeven aan het probleem ν te vinden zo dat $l = k^\nu$. De naam is vanzelfsprekend afgeleid uit het feit dat voor deze combinatie geldt dat $\nu = \log_k(l)$. Het Diffie-Hellman protocol baseert zich op de aanname dat geen voldoende efficiënte algoritmen bestaan om dit getal ν te vinden. Voorbeelden van algoritmen die een oplossing aan dit probleem kunnen bieden, maar niet efficiënt genoeg zijn, zijn de *brute force*-aanpak en het *baby step, giant step*-algoritme. De classificatie van deze algoritmen zijn respectievelijk $\mathcal{O}(n)$ en $\mathcal{O}(\sqrt{n})$, waaruit blijkt dat voor afdoende grote n de tijd benodigd om een oplossing te vinden aanzienlijk is. Ter illustratie: Curve25519, een veelgebruikte kromme, gebruikt het priemgetal $p = 2^{255} - 19$. Zoals de naam doet vermoeden is een brute force aanval weinig elegant, in de zodanigheid dat voor iedere $x \in [1, n]$ gecontroleerd wordt of x voldoet aan $k^x = l$. Is dit het geval, dan kan voor alle gebruiken geconcludeerd worden dat $x = \nu$ en wordt de bewerking afgebroken. De classificatie verraadde reeds dat in het ergste geval alle mogelijke waarden voor x gecontroleerd worden. Deze onderneming kan met de hedendaags beschikbare computatiekracht jaren in beslag nemen.

6 ElGamal encryptie

Als voorbeeld van een toepassing van het Diffie-Hellman sleuteluitwisselings-protocol wordt in dit hoofdstuk een eenvoudig encryptie systeem geïntroduceerd.

Neem aan dat dit een voortzetting is van het stappenplan zoals beschreven in het hoofdstuk *Het Diffie-Hellman protocol*, en het proces om een gedeelde sleutel C te bepalen reeds is doorlopen. Neem verder aan dat een inverteerbare afbeelding $E : m \to E_m$ die een boodschap m verbindt aan een punt $E_m \in \mathrm{K}(\mathbb{F}_p)$ bij beide partijen bekend is. Om een versleutelde boodschap te delen verstuurt de verzender het punt $E_m + C$, waarna de ontvanger de boodschap uit dit punt kan winnen door het punt C op het ontvangen punt in mindering te brengen en de inverse functie toe te passen. Zo vindt de ontvanger $E^{-1}(E_m + C - C) = E^{-1}(E_m) = m$.

Normalitair wordt in het ElGamal systeem de sleutel tijdens de overdracht vastgesteld. Deze taak is in dit geval tevoren overgenomen door het Diffie-Hellman protocol. Merk op dat de veiligheid van dit systeem rust op de mogelijkheid het punt C te vinden, waardoor ook hier de veiligheid gegarandeerd wordt door het discreet logaritmisch probleem.

7 Een aanval op het discreet logaritme

Laat $\mathbf{G}$ een groep van orde n gegenereerd door een element k zijn, en $D \in \mathbf{G}$. Omdat $\mathbf{G}$ cyclisch is bestaat een $\nu \in \mathbb{N}_{\leq n}$ zo dat $D = k^\nu$. Dit verantwoordt het initiatief om een aanval uit te voeren. Let wel dat ondanks dat dit het bestaan van een waarde voor ν garandeert deze niet uniek vastgelegd is. Wanneer een waarde voor ν gevonden is voldoen alle waarden $\nu + \gamma n$ voor $\gamma \in \mathbb{N}$, omdat $k^{\nu+\gamma n} = k^\nu (k^n)^\gamma = k^\nu O^\gamma = k^\nu O = k^\nu$.

De groep is zoals voorheen gedefiniëerd door een elliptische kromme $\mathrm{K}(\mathbb{F}_p)$ en bevat dus hetzelfde aantal punten als zich op de kromme bevinden. Hoewel de parameters (a, b, p) bekend zijn bieden deze geen indicatie voor het aantal punten op de kromme. Wel bestaat een bovengrens, welke volgt uit de stelling van Hasse. Deze stelt dat voor de gegeven elliptische kromme moet gelden dat $|n - (p+1)| \leq 2\sqrt{p}$. Omdat per definitie $n - (p+1) \leq |n - (p+1)|$, moet ook gelden dat $n - (p+1) \leq 2\sqrt{p}$. Hier uit volgt op zijn beurt $|\mathbf{G}| = n \leq 2\sqrt{p} + p + 1$. Om het precies aantal punten te bepalen kan gebruik gemaakt worden van het algoritme van Schoof. Deze zal echter in dit hoofdstuk niet gebruikt worden.

Stel nu $m = \lceil \sqrt{n} \rceil$ indien n bekend is. Zo niet, gebruik dan bovenstaande bovengrens in plaats van n. Uit de stelling van Bachet-Bézout, een gevolg van het uitgebreid algoritme van Euclides, volgt dat voor een positief geheel getal x twee getallen $r, s \in \mathbb{N}_{<m}$ bestaan zo dat $x = rm + s$. De relevantie hiervan wordt later belicht.

Nu deze eigenschappen bekend zijn kan begonnen worden met het formuleren van het *baby step, giant step*-algoritme, de gekozen aanval op het discreet logaritmisch probleem. Dit algoritme verloopt volgens de volgende stappen.

1. Neem het element k^{-m} en stel $i = 1$.
2. Bereken voor ieder getal $j \in [1, m-1]$ het element k^j en sla het gevonden paar (j, k^j) op in een tabel.
3. Bereken $D(k^{-m})^i$ en vergelijk dit element met alle gevonden elementen k^j opgeslagen in de tabel.
4. Wanneer voor een combinatie geldt dat $D(k^{-m})^i$ gelijk is aan k^j wordt de bewerking afgebroken en is $\nu = im + j$ gevonden. Wanneer dit voor geen van de combinaties geldt, stel $i = i + 1$ en herhaal vanaf stap 3.

Voor het teruggekeerde paar (i, j) geldt $D(k^{-m})^i = k^j$, ofwel:

$$D(k^{-m})^i = k^j \Rightarrow Dk^{-im} = k^j \Rightarrow D = k^{im+j}$$

Wat de uitspraak $\nu = im + j$ bevestigt.

Omdat bekend is dat er een paar (i, j) gelijk aan (r, s) moet bestaan wordt de vierde stap op zijn hoogst m keer herhaald. De overige stappen worden in lineaire tijd $c(j)$ uitgevoerd. In het geval dat n bekend is volgt, omdat $\mathcal{O}(\lceil\sqrt{n}\rceil + c) \subset \mathcal{O}(\sqrt{n})$, dat de complexiteit van dit algoritme zoals eerder beweerd $\mathcal{O}(\sqrt{n})$ is. Wanneer n onbekend is en de bovengrens wordt gebruikt volgt uit eenzelfde beredenering dezelfde classificatie. Ondanks de gelijkenis met het *brute force*-algoritme is deze methode dus aanzienlijk sneller, wat te verklaren is door het feit dat het *baby step, giant step*-algoritme het aantal mogelijke waarden voor x die onderzocht worden beperkt. De naam is dan ook ontleend aan de manier waarop dit gebeurt.

Een bijkomstigheid van dit algoritme is dat deze met enkele modificaties gebruikt kan worden om de orde n van $\mathbf{G}$ exact te bepalen. Hoewel dit in dit hoofdstuk niet gedaan wordt is op te merken dat de huidige iteratie in specifieke gevallen ook de mogelijkheid hier toe biedt. Wanneer geldt dat $n < m-1$ zal het element k twee keer in de tabel gegenereerd in de derde stap voorkomen en vice versa. Laat u, v het paar indices met de kleinste afstand tot elkaar zijn zo dat $u < v$ en op posities u en v het element k in de tabel staat. Omdat k een generator van de groep is moet volgen dat $n = v - u$. Wanneer het doel echter enkel het bepalen van n is is deze methode niet de meest efficiënte manier.

8 Implementatie in Python

In dit hoofdstuk wordt de theorie zoals besproken in de vorige hoofdstukken verenigd, en wordt invulling gegeven aan de operaties op de groep. Vervolgens worden deze vertaald naar hun bijbehorende methoden in Python 3.4. De resulterende uitwerking hiervan kan in de volgende repository gevonden worden:

http://github.com/QuirijnMeijer/EKC

Doorgaans zal in het gehele hoofdstuk aangenomen worden dat over het lichaam $\mathbb{R}$ (wanneer p gelijk is aan 0) gewerkt wordt. Wanneer specifiek met een ander karakteristiek gewerkt wordt is dit vermeld – dit omdat de berekeningen over deze lichamen een aanpassing zijn van de tegenhangers over $\mathbb{R}$.

Elliptische kromme

Negatie

Uit de inherente horizontale symmetrie van iedere elliptische kromme kan afgeleid worden dat de negatie van een punt niets anders is dan de spiegeling van het punt rondom de x-as. De methode levert voor een punt $P = (x, y)$ de spiegeling $-P = (x, -y)$ als Punt-object op. Het is eenvoudig na te gaan dat wanneer (x, y) voldoet aan de vergelijking zoals geïntroduceerd in het derde hoofdstuk, het punt $(x, -y)$ dit ook doet. Er geldt $(y)^2 = (-y)^2$.

Optelling

Om twee punten P, Q, $P \neq Q$ gelegen op een elliptische kromme op te tellen wordt een rechte lijn door de twee punten getrokken, welke dankzij de aard van de elliptische krommen één snijpunt anders dan P, Q zal hebben met de kromme. Dit punt wordt aangeduid met $P * Q$. Op dezelfde wijze volgt nu $P + Q = O * (P * Q) = -(P * Q)$.

Om dit te implementeren in Python beschouwen we dit proces meetkundig [1] om zo de coördinaten van het punt $P+Q$ uitgedrukt in bekende waarden te vinden.

De lijn door de punten $P = (x_1, y_1), Q = (x_2, y_2)$ zoals hierboven beschreven heeft $\alpha = \frac{\Delta y}{\Delta x} = \frac{y_2 - y_1}{x_2 - x_1}$ als hellingscoëfficiënt, en omdat P op de lijn ligt kan het steunpunt β gevonden worden door $y_1 = \alpha x_1 + \beta$ op te lossen. Hieruit volgt $\beta = y_1 - \alpha x_1$. De gezochte vergelijking van de lijn is zo $y = \alpha x + \beta$. Om $P * Q = (x_3, y_3)$ te vinden moet het snijpunt met de elliptische kromme $y^2 = x^3 + ax + b$ bepaald worden. Deze wordt verkregen door $y = \alpha x + \beta$ te substitueren in de vergelijking van de elliptische kromme, $y^2 = (\alpha x + \beta)^2 = x^3 + ax + b$. Uitwerken hiervan levert de volgende vergelijking op:

$$x^3 + (-\alpha^2)x^2 + (a - 2\alpha\beta)x + (b - \beta^2) = 0$$

Omdat bekend is dat x_1, x_2, x_3 door constructie nulpunten zijn van deze vergelijking moet volgen dat:

$$x^3 + (-\alpha^2)x^2 + (a - 2\alpha\beta)x + (b - \beta^2) = (x - x_1)(x - x_2)(x - x_3)$$

Waarvan op zijn beurt het rechterlid gelijk is aan:

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

Door nu de coëfficiënten van x^2 aan beide zijden te vergelijken wordt bevonden dat $-\alpha^2 = -x_1 - x_2 - x_3$. Uit het rearrangeren van deze vergelijking volgt $x_3 = \alpha^2 - x_1 - x_2$. Omdat het punt $P * Q$ op de lijn $y = \alpha x + \beta$ door P, Q ligt moet volgen dat $y_3 = \alpha x_3 + \beta$. De volgende expliciete coördinaten van $P * Q$ in reeds bekende termen zijn nu gevonden:

$$P * Q = (x_3, y_3) = ([\tfrac{y_2 - y_1}{x_2 - x_1}]^2 - x_1 - x_2, [\tfrac{y_2 - y_1}{x_2 - x_1}] \cdot [x_3 - x_1] + y_1)$$

Zoals eerder besproken volgt $P + Q = O * (P * Q) = (x_3, -y_3)$. Wanneer p gelijk is aan 0 zal de methode dit punt terugkeren als Punt-object. Indien de karakteristiek groter dan 3 is worden (x_3, y_3) als breuken geschreven en wordt in iedere stap modulo p gerekend om zo te garanderen dat de coördinaten elementen van $\mathbb{F}_p$ zijn, en het resulterende punt zich bovendien op de elliptische kromme bevindt.

In het geval dat $P = Q$ is de richtingscoëfficiënt α van de lijn door $P = (x_1, y_1)$ niet goed gedefinieerd. Om dit te verhelpen wordt teruggevallen op de helling in het punt P zoals gegeven door de afgeleide. Om deze te bepalen wordt de vergelijking $y^2 = x^3 + ax + b$ herschreven als $y^2 = f(x)$ en impliciet gedifferentiëerd:

$$\tfrac{d}{dx}y^2 = \tfrac{d}{dx}f(x) \Rightarrow 2y\tfrac{dy}{dx} = f'(x) \Rightarrow \tfrac{dy}{dx} = \tfrac{f'(x)}{2y}$$

Bekend is dat $f(x) = x^3 + ax + b$, dus $f'(x) = 3x^2 + a$. Nu volgt:

$$\alpha = \tfrac{dy}{dx}\big|_{(x_1, y_1)} = \tfrac{3x_1^2 + a}{2y_1}$$

En zo:

$$P * P = (x_3, y_3) = ([\tfrac{3x_1^2 + a}{2y_1}]^2 - 2x_1, [\tfrac{3x_1^2 + a}{2y_1}] \cdot [x_3 - x_1] + y_1)$$

Zoals eerder vermeld moet ook hier rekening gehouden worden met de karakteristiek van het lichaam waarover wordt gewerkt, en op dezelfde manier wordt de negatie $2P = O * (P * P)$ als Punt-object teruggekeerd. De methode die in dit deelhoofdstuk tot stand is gekomen zal in uitvoering zelden direct aangeroepen worden, maar is essentiëel voor de implementering van de scalaire vermenigvuldiging zoals gebruikt in het Diffie-Hellman protocol.

Scalaire vermenigvuldiging

Scalaire vermenigvuldiging van een punt P met een getal $n \in \mathbb{N}$ is gedefiniëerd

als $nP = \sum_{i=1}^{n} P$. Deze methode roept herhaaldelijk de methode gewijd aan optelling aan, en keert het resultaat terug als Punt-object. Omdat de groepsoperatie van $\mathbf{G}$ optelling is wordt nP in deze context ook als P^n geschreven.

Aftrekking

Het aftrekken van een punt Q van een punt P geschiedt door de negatie van het punt Q te nemen, zodat $P - Q$ berekend kan worden als $P + (-Q)$. Het resultaat van deze bewerking is een Punt-object.

Punt

De klasse Punt is bewust eenvoudig gelaten omdat alle taken zijn gedelegeerd naar de klasse behorend bij de elliptische kromme, en dient alleen het doel een punt op de kromme als object te vertegenwoordigen.

Diffie-Hellman

Vanwege de relatieve eenvoud van dit sleuteluitwisselingsprotocol bespant de realisatie van de verantwoordelijke klasse slechts enkele regels. De voornaamste bewerkingen, het genereren van publieke en privé sleutels, zijn afhankelijk van de scalaire vermenigvuldiging zoals hierboven behandeld. Deze functionaliteiten zijn in de klasse hernoemd om de gebruiksvriendelijkheid en overzichtelijkheid te bevorderen.

ElGamal

Alle benodigdheden voor het ElGamal systeem zijn per dit hoofdstuk ingevoegd, en in de bijbehorende klasse is een voorbeeld van een afbeelding E gegeven.

Baby step, giant step

Dit algoritme is voor elliptische krommen over $\mathbb{F}_p$ geïmplementeerd zoals het stappenplan in het hoofdstuk *Een aanval op het discreet logaritme* voor schrijft en benodigt hierdoor geen nadere verklaring.

Paar

De klasse Paar is een ondersteunende klasse die de mogelijkheid biedt de paren (j, k^j) zoals gebruikt in het *baby step, giant step*-algoritme te registreren.

9 Gebruik en toelichting

In de implementatie zijn voor het Diffie-Hellman protocol, het ElGamal encryptie systeem en het *baby step, giant step*-algoritme demonstratie scripts opgenomen. Deze etaleren ieder de mogelijkheden van hun implementatie, en wanneer ze samen gevoegd of achtereen uitgevoerd worden is een systeem gerealiseerd dat een uitwisseling van boodschappen van begin tot eind bestrijkt. In dit hoofdstuk wordt het gebruik van de bijbehorende klassen toegelicht.

Diffie-Hellman

Een instantie van de Diffie-Hellman klasse wordt geïnitialiseerd met een elliptische kromme $K = \mathrm{K}(\mathbb{F}_p)$ en een genererend element k. Vervolgens kunnen op basis van de opgegeven parameters door de methoden publiekeSleutel en priveSleutel voor beide partijen sleutelparen van publieke- en privé-sleutels gegenereerd worden. Na uitwisseling van de publieke sleutels bezitten beide partijen over de gedeelde geheime sleutel C.

ElGamal

De ElGamal klasse heeft als parameters (K, k, C), waarin C een gedeelde sleutel is. Wanneer de verzameling parameters gelijk is aan de verzameling zoals gebruikt en gevonden door het Diffie-Hellman protocol is dit een directe voortzetting. De klasse bevat een injectieve afbeelding E (codeerBoodschap) die letters en leestekens uit het alfabet verbindt aan punten op de elliptische kromme K, en een inverse afbeelding E^{-1} (decodeerBoodschap). De uitvoer van de methode codeerBoodschap kan met de methoden formateerCode en printCode getransformeerd worden naar respectievelijk een boodschap die geïnterpreteerd kan worden door de huidige implementatie (zodat deze direct overgedragen kan worden als array) en een lijst punten die de versleutelde karakters van de boodschap representeren. Zoals te zien is door het alfabet en de uitvoer van printCode te vergelijken zijn deze punten gewonnen door toepassing van het ElGamal systeem. De afbeelding E die gebruikt is in de implementatie verbindt verschillende karakters aan machten van k. Dit is slechts een voorbeeld, en deze afbeelding kan door een willekeurige andere afbeelding overschreven worden met de methode verstelAfbeelding. In het gebruik van deze klasse in combinatie met dit voorbeeld in het bijzonder is het belangrijk dat het element k meer unieke elementen moet kunnen genereren dan dat zich karakters in het gebruikt alfabet bevinden. Zoniet, is de afbeelding E niet injectief en is E^{-1} niet goed gedefiniëerd.

BabyStepGiantStep

De BabyStepGiantStep klasse benodigt de parameters (K, k, m), waarin m de orde van de groep $\mathbf{G}$ gegenereerd door $\mathrm{K}(\mathbb{F}_p)$ danwel een bovengrens hiervoor is. Wanneer deze niet bekend zijn kan deze parameter gelijk gesteld worden aan 0, in welk geval de klasse een bovengrens berekent.

Om vervolgens een discreet logaritme te berekenen moet het doel D ingesteld worden door gebruik te maken van de methode verstelDoel. Wanneer dit gedaan is kan een oplossing gevonden worden met de methode vindMacht.

De klasse kan geherïnitialiseerd worden door de methode reset. Wanneer deze methode aangeroepen wordt worden alle wijzigingen ongedaan gemaakt.

Overige klassen

De resterende klassen dienen als fundament voor de drie hierboven beschreven hoofdklassen. Ondanks dit kunnen ze ook op zichzelf gebruikt worden om bijvoorbeeld een elliptische kromme en een verzameling bijbehorende punten te representeren. Wanneer een elliptische kromme K als instantie gecreëerd is en op zijn minst twee punten P, Q op K gedefiniëerd zijn kunnen de standaard Python-operatoren zoals $+$, $-$ en $**$ voor optellen, aftrekken en machtsverheffing gebruikt worden.

De methode klok uit de klasse Ondersteuning verdient een specifieke vermelding. Deze methode neemt een andere methode, parameters incluis, als parameter en berekent de tijd benodigd voor het uitvoeren van de opgegeven methode. Het resultaat van de methode is een array met de teruggekeerde waarde van de opgegeven methode in de eerste positie, en in de tweede positie de benodigde tijd om tot dit resultaat te komen.

Bronnen

1. Tate, Silverman (1994). Rational Points on Elliptic Curves (gecorrigeerd). New York City, New York: Springer-Verlag