Learning with Errors

Bachelorscriptie

Rieky Peters
s4475844

Begeleider: Wieb Bosma
Tweede lezer: Maarten Solleveld

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

Inhoudsopgave

Inleiding

In 2018 heeft Oded Regev de Gödelprijs gewonnen, dit is een jaarlijkse prijs voor buitengewone publicaties op het gebied van theoretische informatica. Hij heeft de prijs gewonnen voor zijn publicatie over Learning with Errors [Reg09]. Dit is een probleem waarvan Regev heeft bewezen dat het even moeilijk is als een aantal roosterproblemen. Er is een sterk vermoeden dat deze roosterproblemen zelfs met een kwantumcomputer niet kunnen worden opgelost. Als het vermoeden klopt betekent dit dat het Learning with Errors probleem ook niet met een kwantumcomputer kan worden opgelost, wat het erg interessant maakt voor cryptografie. De huidige problemen waarop cryptografische problemen gebaseerd zijn, zoals priemfactorontbinding en het discrete logprobleem, zouden met een kwantumcomputer binnen polynomiale tijd opgelost kunnen worden. Om voorbereid te zijn op de tijd dat kwantumcomputers bestaan kunnen we dus problemen als Learning with Errors goed gebruiken om cryptografische systemen te maken waarvan wordt gedacht dat die zelfs met een kwantumcomputer niet te kraken zijn.

Het Learning with Errors probleem heeft een aantal parameters. Wat Regev bewezen heeft is dat als Learning with Errors efficiënt kan worden opgelost, bepaalde roosterproblemen met een kwantumcomputer ook efficiënt kunnen worden opgelost. Deze stelling geldt alleen voor bepaalde waarden van de parameters van het Learning with Errors probleem, maar voor deze waarden is het cryptografische systeem dat gebaseerd is op dit probleem wel snel. Later heeft Peikert [Pei09] bewezen dat als Learning with Errors efficiënt kan worden opgelost, bepaalde roosterproblemen ook efficiënt kunnen worden opgelost. Er gelden bij deze stelling andere eisen voor de parameters van het probleem, waardoor het cryptografische systeem dat op dit probleem gebaseerd is langzamer is.

Deze bachelorscriptie gaat over het Learning with Errors probleem, cryptografische systemen die hierop gebaseerd zijn, roosters, roosterproblemen en het bewijs dat Regev heeft gegeven.

1 Learning with Errors

Stel we hebben een onbekende vector $(s_1, s_2, s_3, s_4) = \mathbf{s} \in \mathbb{Z}_{11}^4$ en een aantal vergelijkingen van de vorm

$$\begin{aligned} 1s_1 + 2s_2 + 3s_3 + 4s_4 &\approx 9 \bmod 11 \\ 2s_1 + 5s_2 + 6s_3 + 8s_4 &\approx 6 \bmod 11 \\ 4s_1 + 8s_2 + 2s_3 + 5s_4 &\approx 3 \bmod 11 \\ 7s_1 + 3s_2 + 10s_3 + 7s_4 &\approx 1 \bmod 11 \\ &\vdots \\ 9s_1 + 5s_2 + 9s_3 + 6s_4 &\approx 5 \bmod 11. \end{aligned}$$

Elke vergelijking is goed op een bepaalde kleine onbekende fout na. Het Learning with Errors probleem vraagt nu om $\mathbf{s}$ te achterhalen. Als we niet aan elke vergelijking een kleine fout zouden toevoegen is het makkelijk om $\mathbf{s}$ te vinden met Gauss-eliminatie.
De parameters van het probleem zijn de dimensie n, een modulus q, het aantal vergelijkingen m en een kansverdeling χ op $\mathbb{Z}_q$.

Definitie 1.1. We definiëren $A_{\mathbf{s},\chi}$ als de kansverdeling op $\mathbb{Z}_q^n \times \mathbb{Z}_q$ die verkregen wordt door een vector $\mathbf{a}$ uit de uniforme verdeling op $\mathbb{Z}_q^n$ te trekken, een fout e uit de kansverdeling χ op $\mathbb{Z}_q$ te trekken en het paar $(\mathbf{a}, b = \langle \mathbf{a}, \mathbf{s} \rangle + e \bmod q)$ te bepalen. Hierbij is $\langle \mathrm{a,s} \rangle = \sum_{i=1}^n a_i s_i$ het standaard inproduct dat $\mathbb{Z}_q^n \times \mathbb{Z}_q^n$ op $\mathbb{Z}_q$ afbeeldt.

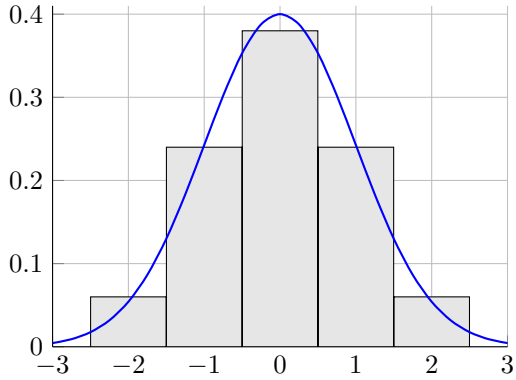
We kunnen het probleem formeel definiëren als volgt: een algoritme lost het Learning with Errors probleem met parameters n, q, m en χ op als voor elke $\mathbf{s} \in \mathbb{Z}_q^n$ en m onafhankelijke trekkingen $(\mathbf{a}_i, b_i)$ uit de kansverdeling $A_{\mathbf{s},\chi}$ met kans exponentieel dicht bij 1 $\mathbf{s}$ wordt achterhaald.
Het is vaak handig om dit in matrixvorm te zetten, waarbij de m vectoren $\mathbf{a}_i \in \mathbb{Z}_q^n$ de kolommen van de matrix $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$ zijn, $\mathbf{b}, \mathbf{e} \in \mathbb{Z}_q^{1 \times m}$ en $\mathbf{s} \in \mathbb{Z}_q^{1 \times n}$.
We kunnen geen Gauss-eliminatie gebruiken om $\mathbf{s}$ te vinden. Stel dat we dit wel doen, dan wordt de fout door het vegen vergroot. Dit wordt duidelijk als we dit bij de eerste vier vergelijkingen in het bovenstaande voorbeeld proberen. We trekken de eerste rij 2 keer van de tweede rij af, 4 keer van de derde en 7 keer van de vierde. Maar we hebben nu ook e_1 respectievelijk 2, 4 en 7 keer van e_2, e_3 en e_4 afgetrokken. Vervolgens moeten we 2 keer de tweede rij, 3 keer de derde rij en 4 keer de vierde rij van de eerste rij aftrekken.

$$\left(\begin{array}{cccc|c} 1 & 2 & 3 & 4 & 9 \\ 2 & 5 & 6 & 8 & 6 \\ 4 & 8 & 2 & 5 & 3 \\ 7 & 3 & 10 & 7 & 1 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 2 & 3 & 4 & 9 \\ 0 & 1 & 0 & 0 & 10 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 \end{array}\right) \Rightarrow \left(\begin{array}{cccc|c} 1 & 0 & 0 & 0 & 6 \\ 0 & 1 & 0 & 0 & 10 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 4 \end{array}\right)$$

We begonnen met een fout $e = (e_1, e_2, e_3, e_4) = (1, -1, 0, 1)$ (want de oplossing is $(7, 2, 4, 10)$) en eindigen met een fout $e' = (e_1', e_2', e_3', e_4') = (e_1 - 2e_2' - 3e_3' - 4e_4', e_2 - 2e_1, e_3 - 4e_1, e_4 - 7e_1) = (-1, -3, -4, -6) \equiv (10, 8, 4, 5)$. We kunnen dus geen enkele informatie afleiden uit de verkregen oplossing $(6, 10, 0, 4)$.

De toegevoegde fout wordt getrokken uit de normale verdeling. Deze verdeling heeft twee parameters: het gemiddelde μ en de variantie σ^2. De dichtheidsfunctie is $f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$. Bij χ is μ gelijk aan 0 en σ gelijk aan αq. We kiezen α doorgaans $1/\mathrm{poly}(n)$, waarbij $\mathrm{poly}(n) = n^c$, voor een constante $c > 0$. Deze verdeling is eigenlijk continu, maar we kunnen hem discretiseren door trekkingen af te ronden naar het dichtstbijzijnde gehele getal modulo q. Dan wordt het een discrete normale verdeling op $\mathbb{Z}_q$. In figuur

1 is een voorbeeld weergegeven van een discrete en een continue normale verdeling met $\mu = 0$ en $\sigma = 1$. Regev geeft een preciezere definitie van deze discrete normale verdeling [Reg09].



Figuur 1: Continue en discrete normale verdeling

Algoritmen

Als we kijken naar algoritmen die het Learning with Errors probleem oplossen willen we ook de complexiteit hiervan bepalen. We gebruiken de grote-$\mathcal{O}$-notatie om aan te geven dat een functie $f : \mathbb{R} \to \mathbb{R}$ asymptotisch van boven wordt begrensd door een andere functie $g : \mathbb{R} \to \mathbb{R}$.

Definitie 1.2. $\mathcal{O}(g(x)) = \{f(x) \mid \exists c \in \mathbb{R}_{>0}, N \in \mathbb{N} \text{ zodanig dat } \forall x > N : |f(x)| \leq c \cdot g(x)\}$.

We schrijven vaak $f(x) = \mathcal{O}(g(x))$ in plaats van $f(x) \in \mathcal{O}(g(x))$. Daarnaast hebben we $f(x) = \Omega(g(x))$ als f van onder wordt begrensd door g en $f(x) = \Theta(g(x))$ als $f(x) = \mathcal{O}(g(x))$ en $f(x) = \Omega(g(x))$. We schrijven $\tilde{\mathcal{O}}(\cdot)$ als de logaritmische factor wordt weggelaten.

Voor het Learning with Errors probleem bestaat een aantal algoritmen. Het meest simpele algoritme blijft net zolang trekkingen doen uit $A_{\mathbf{s},\chi}$ totdat er poly(n) paren $(\mathbf{a}, b)$ zijn met $\mathbf{a} = (1, 0, \ldots, 0)$, ofwel vergelijkingen van de vorm $s_1 \approx \ldots$, waaruit we s_1 kunnen afleiden. Omdat $\mathbf{a}$ uit de uniforme verdeling op $\mathbb{Z}_q^n$ wordt getrokken is de kans dat $\mathbf{a} = (1, 0, \ldots, 0)$ gelijk aan q^{-n}. Om poly(n) vergelijkingen van deze vorm te krijgen moeten we q^npoly(n) trekkingen doen. Dit moeten we voor $s_1, \ldots, s_n$ doen. Dus $nq^n\text{poly}(n) = q^n n^d = 2^{n \log(q) \log(n^d)} = 2^{\mathcal{O}(n \log(n))}$ is het aantal vergelijkingen dat we moeten genereren. Dit is ook de complexiteit van het algoritme. Het beste algoritme voor het Learning with Errors probleem is bedacht door Blum, Kalai en Wasserman [BKW03]. We doen $2^{\mathcal{O}(n)}$ trekkingen uit $A_{\mathbf{s},\chi}$. Vind vervolgens een verzameling S van ongeveer n vergelijkingen zodanig dat $\sum_S \mathbf{a}_i = (1, 0, \ldots, 0)$. Dan geldt $\sum_S b_i = \sum_S (\langle \mathbf{a}_i, \mathbf{s} \rangle + e_i) = \sum_S e_i + \langle \sum_S \mathbf{a}_i, \mathbf{s} \rangle = \sum_S e_i + s_1$. Omdat elke e_i getrokken is uit χ, en χ normaal verdeeld is met $\mu = 0$ geldt dat $\sum_S e_i \approx 0$. Op dezelfde manier vinden we $s_2, \ldots, s_n$. De complexiteit van dit algoritme is gelijk aan het aantal trekkingen: $2^{\mathcal{O}(n)}$.

Varianten van het probleem

Hetgeen hierboven beschreven is staat eigenlijk bekend als het zoekprobleem. Er is namelijk ook een beslissingsprobleem: gegeven m onafhankelijke trekkingen $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ waar elke trekking ofwel uit $A_{\mathbf{s},\chi}$ met een vaste $\mathbf{s} \in \mathbb{Z}_q^n$, ofwel uit de uniforme kansverdeling komt, bepaal welke van de twee het geval is. In hoofdstuk 2 zullen we zien dat cryptografische systemen die Learning with Errors gebruiken gebaseerd zijn op de moeilijkheid van het beslissingsprobleem.

Het zoekprobleem kan worden gereduceerd naar het beslissingsprobleem. Met andere woorden, als we het beslissingsprobleem kunnen oplossen, kunnen we ook binnen polynomiale tijd het zoekprobleem oplossen.

Lemma 1.3. *Zij $n \in \mathbb{Z}_{\geq 1}, q \leq poly(n)$ een priemgetal en χ een kansverdeling op $\mathbb{Z}_q$. Stel we hebben een algoritme W dat voor elke $\mathbf{s}$ trekkingen uit $A_{\mathbf{s},\chi}$ met kans exponentieel dicht bij 1 accepteert en trekkingen uit een uniforme verdeling U op $\mathbb{Z}_q^n \times \mathbb{Z}_q$ met kans exponentieel dicht bij 1 verwerpt. Dan bestaat er een efficiënt algoritme W' dat, gegeven trekkingen uit $A_{\mathbf{s},\chi}$ voor een bepaalde $\mathbf{s}$, deze $\mathbf{s}$ vindt met kans exponentieel dicht bij 1.*

Bewijs. We laten zien hoe W' de eerste coördinaat van $\mathbf{s}$ vindt, de andere gaan op dezelfde manier. Neem een willekeurige $k \in \mathbb{Z}_q$. Gegeven een paar $(\mathbf{a}, b)$ uit $A_{\mathbf{s},\chi}$, bepaal $(\mathbf{a} + (r, 0, \ldots, 0), b + rk)$ waarbij $r \in \mathbb{Z}_q$ willekeurig. Als $k = s_1$ beeldt deze transformatie $A_{\mathbf{s},\chi}$ af op zichzelf. Er geldt namelijk dat $b + rk = \langle \mathbf{a}, \mathbf{s} \rangle + e + rk = \sum_{i=1}^{n} a_i s_i + e + r s_1 = \langle \mathbf{a} + (r, 0, \ldots, 0), \mathbf{s} \rangle + e$ en omdat r en $\mathbf{a}$ willekeurig zijn, is $\mathbf{a} + (r, 0, \ldots, 0)$ ook willekeurig. Dus als $(\mathbf{a}, b)$ uit $A_{\mathbf{s},\chi}$ komt, dan ook $(\mathbf{a} + (r, 0, \ldots, 0), b + rk)$. Als $k \neq s_1$, beeldt de transformatie $A_{\mathbf{s},\chi}$ af op U, want q is priem en dus onderling ondeelbaar met $b + rk$, dus komt $b + rk$ uit een uniforme verdeling op $\mathbb{Z}_q$. Met behulp van W komen we erachter of $k = s_1$ of niet. Er zijn $q \leq \text{poly}(n)$ mogelijkheden voor s_1 dus we kunnen ze allemaal afgaan. □

In [Pei09] wordt dit lemma bewezen voor q een product van verschillende priemgetallen die allemaal begrensd worden door poly(n) en χ een normale verdeling op $\mathbb{Z}_q$.
Het omgekeerde van lemma 1.3, als we het zoekprobleem kunnen oplossen, kunnen we ook het beslissingsprobleem oplossen, geldt ook en is makkelijker: doe trekkingen uit de onbekende kansverdeling en stop deze paren in het algoritme dat het zoekprobleem oplost. Als er geen oplossing wordt gevonden kwamen de trekkingen uit een uniforme verdeling op $\mathbb{Z}_q^n \times \mathbb{Z}_q$. Als er wel een oplossing $\mathbf{s}$ wordt gevonden bepalen we voor elk paar $b_i - \langle \mathbf{a}_i, \mathbf{s} \rangle$. Als deze uitkomsten verdeeld zijn volgens de kansverdeling χ kwamen de trekkingen uit $A_{\mathbf{s},\chi}$, en anders uit de uniforme verdeling.

Lemma 1.4. *Zij $n, q \in \mathbb{Z}_{\geq 1}$ en χ een kansverdeling op $\mathbb{Z}_q$. Stel we hebben een algoritme W dat $A_{\mathbf{s},\chi}$ onderscheidt van de uniforme verdeling U voor een niet te verwaarlozen deel van alle mogelijke $\mathbf{s} \in \mathbb{Z}_q^n$. Dan bestaat er een algoritme W' dat voor alle $\mathbf{s}$ met een kans exponentieel dicht bij 1 accepteert als de trekkingen uit $A_{\mathbf{s},\chi}$ komen en met een kans exponentieel dicht bij 1 verwerpt als de trekkingen uit U komen.*

Met andere woorden, als we het beslissingsprobleem kunnen oplossen voor een deel van alle mogelijke $\mathbf{s}$, dan kunnen we het probleem oplossen voor elke $\mathbf{s}$. Hieronder staat een schets van het bewijs, een exact bewijs is te vinden in [Reg09].

Bewijsschets. Voor elke $\mathbf{t} \in \mathbb{Z}_q^n$ definiëren we de functie $f_{\mathbf{t}} : \mathbb{Z}_q^n \times \mathbb{Z}_q \to \mathbb{Z}_q^n \times \mathbb{Z}_q$ door $f_{\mathbf{t}}(\mathbf{a}, b) = (\mathbf{a}, b + \langle \mathbf{a}, \mathbf{t} \rangle)$. Omdat $b + \langle \mathbf{a}, \mathbf{t} \rangle = \langle \mathbf{a}, \mathbf{s} \rangle + e + \langle \mathbf{a}, \mathbf{t} \rangle = \langle \mathbf{a}, \mathbf{s} + \mathbf{t} \rangle + e$, beeldt de functie $f_{\mathbf{t}}$ $A_{\mathbf{s},\chi}$ af op $A_{\mathbf{s}+\mathbf{t},\chi}$. Daarnaast beeldt $f_{\mathbf{t}}$ de uniforme verdeling U af op zichzelf. We weten dat voor een niet te verwaarlozen deel van alle mogelijke $\mathbf{s}$, de kans dat W het paar $(\mathbf{a}, b)$ accepteert als het uit $A_{\mathbf{s},\chi}$ komt en als het uit U komt significant verschilt. We maken nu het algoritme W' als volgt: laat D een onbekende kansverdeling op $\mathbb{Z}_q^n \times \mathbb{Z}_q$ die ofwel U, ofwel $A_{\mathbf{s},\chi}$ is. We schrijven $f_{\mathbf{t}}(D) = U$ als $D = U$ en $f_{\mathbf{t}}(D) = A_{\mathbf{s}+\mathbf{t},\chi}$ als $D = A_{\mathbf{s},\chi}$. Herhaal het volgende poly(n) keer: kies $\mathbf{t} \in \mathbb{Z}_q^n$ willekeurig, schat vervolgens de kans dat W een paar $(\mathbf{a}, b)$ accepteert als het uit $f_{\mathbf{t}}(D)$ of uit U komt. Doe dit door W poly(n) keer toe te passen op beide mogelijkheden voor D. Als de schattingen significant verschillen accepteren we, anders herhalen we het voor een andere $\mathbf{t} \in \mathbb{Z}_q^n$. Als ze nooit significant verschillen verwerpen we.
We moeten nu nog laten zien dat W' $A_{\mathbf{s},\chi}$ van U onderscheidt voor alle $\mathbf{s}$. Stel dat $D = U$, er geldt dat $f_{\mathbf{t}}(U) = U$ dus de kans dat W' accepteert ligt exponentieel dicht bij 0. Stel nu dat $D = A_{\mathbf{s},\chi}$ voor een bepaalde $\mathbf{s}$. De kans dat W het paar $(\mathbf{a}, b)$ accepteert als het uit $A_{\mathbf{s}+\mathbf{t},\chi}$ komt en als het uit U komt verschilt significant, dus de kans dat W' accepteert ligt exponentieel dicht bij 1. □

Het Learning with Errors probleem heeft ook een normale vorm, hierbij worden de coördinaten van $\mathbf{s}$ niet uniform getrokken uit $\mathbb{Z}_q$, maar uit de kansverdeling χ. Voor cryptografische constructies is dit efficiënter, hierover meer in hoofdstuk 2. Applebaum et al. [App+09] heeft het volgende lemma bewezen:

Lemma 1.5. *De zoek- en beslissingsvariant van de normale vorm van het probleem zijn minstens even moeilijk als respectievelijk de zoek- en beslissingsvariant waarbij* $\mathbf{s}$ *uniform verdeeld is.*

Bewijs. We moeten laten zien dat er een efficiënte transformatie T bestaat die $A_{\mathbf{s},\chi}$ op $A_{\mathbf{x},\chi}$ afbeeldt en de uniforme verdeling U op $\mathbb{Z}_q^n \times \mathbb{Z}_q$ op zichzelf afbeeldt, voor een willekeurige $\mathbf{s} \in \mathbb{Z}_q^n$ en $\mathbf{x}$ uit de kansverdeling χ^n. De transformatie T beschikt over een kansverdeling D op $\mathbb{Z}_q^n \times \mathbb{Z}_q$, waarbij D ofwel $A_{\mathbf{s},\chi}$ is, ofwel U. Trek paren $(\mathbf{a}, b)$ uit D en kijken steeds of het paar lineair onafhankelijk is van alle voorgaande paren. Als dit het geval is bewaren we het paar, anders gooien we hem weg. Zetten we de verkregen n $\mathbf{a}$'s als kolommen in een matrix dan krijgen we een inverteerbare matrix $\bar{\mathbf{A}} \in \mathbb{Z}_q^{n \times n}$, de b's vormen een vector $\bar{\mathbf{b}} \in \mathbb{Z}_q^n$. Vervolgens bepalen we voor een gegeven paar $(\mathbf{a}, b) \in \mathbb{Z}_q^n \times \mathbb{Z}_q$ uit D het paar $(\mathbf{a}', b') \in \mathbb{Z}_q^n \times \mathbb{Z}_q$, waar

$$\mathbf{a}' = -\bar{\mathbf{A}}^{-1}\mathbf{a},\ b' = b + \langle \mathbf{a}', \bar{\mathbf{b}} \rangle.$$

Omdat $\bar{\mathbf{A}}$ inverteerbaar is en $\mathbf{a}$ getrokken is uit de uniforme verdeling op $\mathbb{Z}_q^n$, komt $\mathbf{a}'$ ook uit de uniforme verdeling op $\mathbb{Z}_q^n$. We bekijken nu de twee gevallen voor D. Als $D = U$, dan komt $(\mathbf{a}', b')$ ook uit de kansverdeling U omdat b, en dus ook b', uit de uniforme verdeling komen. Dus de transformatie beeldt U af op zichzelf. Als $D = A_{\mathbf{s},\chi}$, dan $b = \langle \mathbf{a}, \mathbf{s} \rangle + e$, waarbij e uit de kansverdeling χ komt, en $\bar{\mathbf{b}} = \bar{\mathbf{A}}^t\mathbf{s} + \mathbf{x}$, waarbij $\mathbf{x}$ uit de kansverdeling χ^n komt. We hebben nu dus

$$b' = \langle \mathbf{a}, \mathbf{s} \rangle + e + \langle \mathbf{a}', \bar{\mathbf{b}} \rangle = \langle \mathbf{a}, \mathbf{s} \rangle + e - \langle \bar{\mathbf{A}}^{-1}\mathbf{a}, \bar{\mathbf{A}}^t\mathbf{s} \rangle + \langle \mathbf{a}', \mathbf{x} \rangle = \langle \mathbf{a}', \mathbf{x} \rangle + e.$$

Dus $(\mathbf{a}', b')$ komt uit $A_{\mathbf{x},\chi}$. Dus de transformatie beeldt $A_{\mathbf{s},\chi}$ af op $A_{\mathbf{x},\chi}$. □

2 Cryptografische toepassing van Learning with Errors

In dit hoofdstuk wordt eerst kort uitgelegd hoe asymmetrische cryptografie werkt, vervolgens bekijken we een cryptosysteem dat Learning with Errors gebruikt en daarna wordt er gekeken naar efficiëntere varianten van dit systeem.

Asymmetrische cryptografie

Stel Alice en Bob willen met elkaar communiceren zonder dat anderen weten waar ze het over hebben. Er is een buitenstaander genaamd Eve die erachter wil komen wat Alice en Bob tegen elkaar zeggen. Als Bob een bericht wil sturen naar Alice moet dit versleuteld worden zodat anderen het niet kunnen lezen. Alice heeft twee sleutels, een publieke sleutel die bekend is voor iedereen en een privé-sleutel die alleen Alice zelf weet. Bob versleutelt zijn bericht met behulp van de publieke sleutel, het hangt van het systeem af op elke manier deze sleutel wordt toegepast op het bericht. Als Alice het versleutelde bericht ontvangt gebruikt ze de privé-sleutel om het te ontcijferen. Asymmetrische cryptografie is altijd gebaseerd op een probleem dat moeilijk is, tenzij je beschikt over bepaalde informatie. De rede hiervoor is dat alleen Alice, die als enige deze informatie heeft, het originele bericht kan achterhalen. Een voorbeeld van zo'n probleem is priemfactorontbinding: gegeven twee priemgetallen p, q is het makkelijk om $n = pq$ te bepalen, maar gegeven n is het moeilijk om p en q te vinden, tenzij je p weet. In dit geval is p dus de informatie waar alleen Alice over beschikt, Bob weet alleen n. Hier is het bekende cryptografische systeem RSA op gebaseerd. Het cryptosysteem dat hieronder wordt beschreven is gebaseerd op het Learning with Errors probleem.

Een cryptosysteem gebaseerd op Learning with Errors

Het systeem [Reg09] wordt geparametriseerd door de dimensie n, modulus q, kansverdeling χ op $\mathbb{Z}_q$ en het aantal paren m. De privé-sleutel is een $\mathbf{s} \in \mathbb{Z}_q^n$. We doen $m \approx (n+1)\log q$ trekkingen uit de kansverdeling $A_{\mathbf{s},\chi}$ en zetten deze als kolommen in een matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^t \end{bmatrix} \in \mathbb{Z}_q^{(n+1)\times m},$$

waar $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \bmod q$.
Voor de $\mathbf{e}$ getrokken uit χ^m willen we dat $\|\mathbf{e}\| \leq B = \frac{q}{4m}$ met grote kans, de reden hiervoor wordt duidelijk als we laten zien hoe de decryptie werkt.
Voor encryptie van een bit $\mu \in \{0,1\}$ is de publieke sleutel en een $\mathbf{x}$ getrokken uit de uniforme kansverdeling op $\{0,1\}^m$ nodig. De versleutelde tekst is

$$\mathbf{c} = \mathbf{P}\mathbf{x} + (\mathbf{0}, \mu \cdot \lfloor \tfrac{q}{2} \rceil) \in \mathbb{Z}_q^{n+1}.$$

Voor decryptie van de versleutelde tekst $\mathbf{c}$ kunnen we de privé-sleutel $\mathbf{s}$ gebruiken. Nu kunnen we μ achterhalen:

$$\mu = \begin{cases} 0 & \text{als } \|(-\mathbf{s}, 1)^t \cdot \mathbf{c}\| \leq \frac{q}{4} \\ 1 & \text{anders.} \end{cases}$$

Dat dit klopt volgt uit de volgende berekening:

$$\begin{aligned}
\|(-\mathbf{s},1)^t \cdot \mathbf{c}\| &= \|(-\mathbf{s},1)^t \cdot (\mathbf{P}\mathbf{x} + (\mathbf{0}, \mu \cdot \lfloor \tfrac{q}{2} \rceil))\| \\
&= \|(-\mathbf{s},1)^t \cdot \mathbf{P}\mathbf{x} + \mu \cdot \lfloor \tfrac{q}{2} \rceil\| \\
&= \| - \mathbf{s}^t\mathbf{A}\mathbf{x} + \mathbf{b}^t\mathbf{x} + \mu \cdot \lfloor \tfrac{q}{2} \rceil\| \\
&= \|\mathbf{e}^t\mathbf{x} + \mu \cdot \lfloor \tfrac{q}{2} \rceil\| \\
&\leq \|\mathbf{e}^t\mathbf{x}\| + \|\mu \cdot \lfloor \tfrac{q}{2} \rceil\| \\
&\leq m \cdot B + \|\mu \cdot \lfloor \tfrac{q}{2} \rceil\| \\
&= \tfrac{q}{4} + \|\mu \cdot \lfloor \tfrac{q}{2} \rceil\|.
\end{aligned}$$

Om in te zien dat dit een veilig systeem is moeten we laten zien dat voor een gegeven publieke sleutel er geen verschil te zien is in de versleutelde tekst als μ een 0 of een 1 is. We bekijken twee alternatieve manieren om $\mathbf{P}$ en $\mathbf{c}$ te maken die verschillen van de normale manier hierboven. De eerste manier is dat we $\mathbf{P}$ uniform trekken uit $\mathbb{Z}_q^{(n+1)\times m}$ in plaats van uit $A_{\mathbf{s},\chi}$, noem deze publieke sleutel $\mathbf{P}'$. De versleutelde tekst $\mathbf{c}$ wordt op de normale manier gemaakt. Het paar $(\mathbf{P}', \mathbf{c})$ is niet te onderscheiden van $(\mathbf{P}, \mathbf{c})$ vanwege de moeilijkheid van het beslissingsprobleem van Learning with Errors.
Bij de tweede manier is $\mathbf{P}$ zoals bij de eerste manier, maar nu is $\mathbf{c}$ ook uniform en onafhankelijk van $\mathbf{P}$ getrokken uit $\mathbb{Z}_q^{n+1}$, noem het verkregen paar $(\mathbf{P}', \mathbf{c}')$. $(\mathbf{P}', \mathbf{c})$ en $(\mathbf{P}', \mathbf{c}')$ zijn niet van elkaar te onderscheiden vanwege het Leftover Hash Lemma [Hås+99], die stelt dat $(\mathbf{P}', \mathbf{u} = \mathbf{P}'\mathbf{x})$ voor uniforme en onafhankelijke $\mathbf{P}' \in \mathbb{Z}_q^{(n+1)\times m}$ en $\mathbf{x} \in \{0,1\}^m$ statistisch ononderscheidbaar is van $(\mathbf{P}', \mathbf{u})$, waarbij $\mathbf{u} \in \mathbb{Z}_q^{n+1}$. Als we $(\mathbf{0}, \mu \cdot \lfloor \frac{q}{2} \rceil)$ toevoegen aan $\mathbf{u}$ blijft deze uniform. Het Leftover Hash Lemma mogen we toepassen omdat $m \approx (n+1)\log q$.
Omdat de bovenstaande manieren ononderscheidbaar zijn voor elke bit μ en de laatste helemaal niet van μ afhangt zijn versleutelde teksten in de normale manier voor $\mu = 0, 1$ ook niet van elkaar te onderscheiden.

Dit cryptosysteem gebruikt publieke sleutels die $\tilde{\mathcal{O}}(n^2)$ bits zijn, de privé-sleutel en een versleutelde tekst van $\mu \in \{0,1\}$ zijn $\tilde{\mathcal{O}}(n)$ bits. Als we de normale vorm van het Learning with Errors probleem gebruiken kunnen we dit verkleinen. Trek de n coördinaten van $\mathbf{s}$ uit de kansverdeling χ op $\mathbb{Z}_q$. Zij $\mathbf{P} \in \mathbb{Z}_q^{(n+1)\times(m-n)}$ zoals hierboven, met $\mathbf{A} \in \mathbb{Z}_q^{n\times(m-n)}$ en $\mathbf{b}^t \in \mathbb{Z}_q^{m-n}$. Om een bit $\mu \in \{0,1\}$ te versleutelen nemen we een $\mathbf{x}$ uit de uniforme verdeling op $\{0,1\}^{m+1}$ en bepalen we de versleutelde tekst:

$$\mathbf{c} = [\mathbf{I}_{n+1} \mid \mathbf{P}] \cdot \mathbf{x} + (\mathbf{0}, \mu \cdot \lfloor \tfrac{q}{2} \rceil) \in \mathbb{Z}_q^{n+1}.$$

Om μ uit de versleutelde tekst te achterhalen berekenen we

$$\begin{aligned}
(-\mathbf{s},1)^t \cdot \mathbf{c} &= (-\mathbf{s},1)^t \cdot [\mathbf{I}_{n+1} \mid \mathbf{P}] \cdot \mathbf{x} + \mu \cdot \lfloor \tfrac{q}{2} \rceil \\
&= (-\mathbf{s}, 1, -\mathbf{s}^t\mathbf{A} + \mathbf{b}^t)^t \cdot \mathbf{x} + \mu \cdot \lfloor \tfrac{q}{2} \rceil \\
&= (-\mathbf{s}, 1, \mathbf{e})^t \cdot \mathbf{x} + \mu \cdot \lfloor \tfrac{q}{2} \rceil \\
&\approx \mu \cdot \lfloor \tfrac{q}{2} \rceil \pmod q
\end{aligned}$$

en kijk of dit dichter bij 0 of bij $\lfloor \frac{q}{2} \rceil \bmod q$ ligt. Omdat $\mathbf{s}$ en $\mathbf{e}$ uit χ komen is $(-\mathbf{s}, 1, \mathbf{e})^t$ een korte vector, als we dit vermenigvuldigen met $\mathbf{x}$ is dit getal met grote kans ongeveer 0.
Het bewijs van de veiligheid van dit systeem is nagenoeg gelijk aan het bewijs van het voorgaande cryptosysteem. Het enige verschil is dat het nu gebaseerd is op de moeilijkheid van de normale vorm van het beslissingsprobleem.

In een compacter cryptosysteem van Lindner en Peikert [LP10] zijn de sleutels en versleutelde teksten een factor log q kleiner. De privé-sleutel is $\mathbf{s} \in \mathbb{Z}_q^n$ met coördinaten getrokken uit de kansverdeling χ. Zij $\mathbf{A} \in \mathbb{Z}_q^{n \times n}$ uniform, de publieke sleutel is

$$\mathbf{P} = \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^t \end{bmatrix} \in \mathbb{Z}_q^{(n+1) \times n},$$

waar $\mathbf{b}^t = \mathbf{s}^t\mathbf{A} + \mathbf{e}^t \bmod q$.
Om een bit $\mu \in \{0, 1\}$ te versleutelen kiezen we $\mathbf{x} \in \mathbb{Z}_q^n$ en $\mathbf{y} \in \mathbb{Z}_q^{n+1}$ met coördinaten getrokken uit de kansverdeling χ, de versleutelde tekst wordt

$$\mathbf{c} = \mathbf{P}\mathbf{x} + \mathbf{y} + (\mathbf{0}, \mu \cdot \lfloor \tfrac{q}{2} \rceil) \in \mathbb{Z}_q^{n+1}.$$

Voor de decryptie van de versleutelde tekst gebruiken we $\mathbf{s}$ en bepalen we

$$\begin{aligned}(-\mathbf{s}, 1)^t \cdot \mathbf{c} &= (-\mathbf{s}, 1)^t \cdot \mathbf{P}\mathbf{x} + (-\mathbf{s}, 1)^t \cdot \mathbf{y} + \mu \cdot \lfloor \tfrac{q}{2} \rceil \\ &= -\mathbf{s}^t\mathbf{A}\mathbf{x} + \mathbf{b}^t \cdot \mathbf{x} + (-\mathbf{s}, 1)^t \cdot \mathbf{y} + \mu \cdot \lfloor \tfrac{q}{2} \rceil \\ &= \mathbf{e}^t\mathbf{x} + (-\mathbf{s}, 1)^t \cdot \mathbf{y} + \mu \cdot \lfloor \tfrac{q}{2} \rceil \\ &\approx \mu \cdot \lfloor \tfrac{q}{2} \rceil \pmod q\end{aligned}$$

en kijk of dit dichter bij 0 of $\lfloor \frac{q}{2} \rceil \bmod q$ ligt. Omdat $\mathbf{s}, \mathbf{e}, \mathbf{x}$ en $\mathbf{y}$ uit χ komen is $\mathbf{e}^t\mathbf{x} + (-\mathbf{s}, 1)^t \cdot \mathbf{y}$ met grote kans ongeveer 0.

We bewijzen nu dat dit een veilig systeem is op een vergelijkbare manier als hierboven. Stel dat we $\mathbf{P}$ uniform trekken uit $\mathbb{Z}_q^{(n+1) \times n}$ in plaats van uit $A_{\mathbf{s},\chi}$, noem deze publieke sleutel $\mathbf{P}'$. $\mathbf{P}$ en $\mathbf{P}'$ zijn niet van elkaar te onderscheiden vanwege de moeilijkheid van de normale vorm van het beslissingsprobleem. Als de versleutelde tekst $\mathbf{c}$ op de manier zoals hierboven beschreven is wordt gemaakt dan zijn $(\mathbf{P}, \mathbf{c})$ en $(\mathbf{P}', \mathbf{c})$ ook niet van elkaar te onderscheiden. Stel nu dat $\mathbf{c}$ uniform wordt getrokken uit $\mathbb{Z}_q^{n+1}$, noem deze versleutelde tekst $\mathbf{c}'$. Als we de term $\mu \cdot \lfloor \frac{q}{2} \rceil$ even weglaten, zien we dat $(\mathbf{P}', \mathbf{c})$ precies $n + 1$ trekkingen uit $A_{\mathbf{s},\chi}$ zijn, en die zijn niet te onderscheiden van uniform vanwege de moeilijkheid van de normale vorm van het beslissingprobleem. Dus zijn $(\mathbf{P}', \mathbf{c})$ en $(\mathbf{P}', \mathbf{c}')$ niet van elkaar te onderscheiden. Hieruit volgt dat we niet uit $\mathbf{c}$ kunnen afleiden of het bit μ gelijk is aan 0 of 1.

Stel we hebben het volgende experiment, met parameter $b \in \{0, 1\}$: Maak een publieke sleutel en een privé-sleutel, versleutel een bit b en geef de versleutelde tekst c aan Eve. Vervolgens moet Eve bepalen of $b = 0$ of 1.
Een cryptosysteem heet semantisch veilig als het onmogelijk is voor Eve om te achterhalen of $b = 0$ of 1. Dit is het geval voor bovenstaande cryptosystemen. Als een van deze cryptosystemen in de echte wereld zou worden toegepast, is semantische veiligheid niet voldoende. Een beter begrip van veiligheid is actieve veiligheid. Stel Eve heeft een decryptie-algoritme met de privé-sleutel en mag het algoritme toepassen op elke versleutelde tekst behalve c. Een systeem is actief veilig als het voor Eve onmogelijk is om te achterhalen of $b = 0$ of 1. Peikert en Waters [PW11] hebben een cryptografisch systeem bedacht dat gebaseerd is op de moeilijkheid van Learning with Errors dat actief veilig is.

Bovenstaande cryptosystemen zijn semantisch veilig als het beslissingsprobleem van Learning with Errors moeilijk is. Dat dit een moeilijk probleem is zullen we zien in hoofdstuk 4, maar daarvoor moeten we eerst roosters bekijken.

3 Roosters

We beginnen met de definitie van een rooster.

Definitie 3.1. Een n-dimensionaal rooster $\mathcal{L}$ is een discrete additieve ondergroep van $\mathbb{R}^n$.

Een verzameling is een additieve ondergroep als $\mathbf{0} \in \mathcal{L}$ en als voor elke $\mathbf{x}, \mathbf{y} \in \mathcal{L}$ ook $-\mathbf{x}, \mathbf{x}+\mathbf{y} \in \mathcal{L}$. Discreet betekent hier dat er voor elke $\mathbf{x} \in \mathcal{L}$ een $r > 0$ bestaat zodanig dat $B_r(\mathbf{x}) \cap \mathcal{L} = \{\mathbf{x}\}$, waarbij $B_r(\mathbf{x})$ de (open) bal met straal r rond $\mathbf{x}$. Ofwel, elke $\mathbf{x} \in \mathcal{L}$ heeft een omgeving waarin $\mathbf{x}$ het enige roosterpunt is. Hier bekijken we alleen $\mathcal{L} \subset \mathbb{Z}^n$.

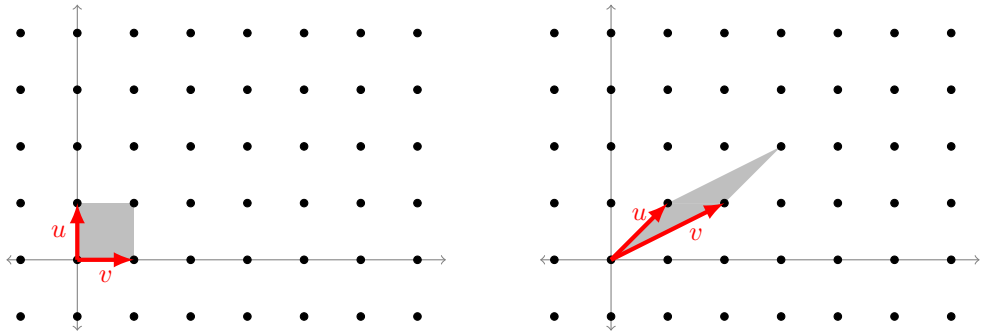
Voorbeeld. Het triviale rooster is $\{\mathbf{0}\}$, dit is het enige eindige rooster. $\mathcal{L} = \mathbb{Z}^n$ met $n \in \mathbb{N}_{>0}$ is een n-dimensionaal rooster. De even getallen, $2\mathbb{Z}$, vormen een 1-dimensionaal rooster, algemener geldt voor elk rooster $\mathcal{L}$ dat $c\mathcal{L}$ voor elke $c \in \mathbb{R}$ ook een rooster is.

Definitie 3.2. Een basis van een rooster $\mathcal{L}$ is een verzameling lineair onafhankelijke vectoren $\mathbf{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_n\} \subset \mathbb{R}^n$ zodanig dat $\mathcal{L} = \mathcal{L}(\mathbf{B}) = \{\sum_{i=1}^n x_i \mathbf{b}_i \mid x_i \in \mathbb{Z}\}$.
Als we $\mathbf{B} \in \mathbb{Z}^{n \times n}$ zien als inverteerbare matrix met $\mathbf{b}_1, \ldots, \mathbf{b}_n$ als kolommen, dan $\mathcal{L} = \mathbf{B} \cdot \mathbb{Z}^n = \{\mathbf{Bx} \mid \mathbf{x} \in \mathbb{Z}^n\}$.

Elk rooster heeft een basis, maar deze is in het algemeen niet uniek. Zonder deze eigenschap zou roostergebaseerde cryptografie niet mogelijk zijn. $\mathbf{B}_1 = \{(1,0),(0,1)\}$ is een basis van $\mathbb{Z}^2$, maar ook $\mathbf{B}_2 = \{(2,1),(1,1)\}$ is een basis. Dit volgt uit het volgende lemma:

Lemma 3.3. *Bases $\mathbf{B}_1, \mathbf{B}_2$ genereren hetzelfde rooster dan en slechts dan als er een $\mathbf{U} \in \mathbb{Z}^{n \times n}$ bestaat zodanig dat $det(\mathbf{U}) = \pm 1$ en $\mathbf{B}_1 = \mathbf{B}_2\mathbf{U}$.*

Deze $\mathbf{U}$ heet ook wel een unimodulaire matrix. Kiezen we nu voor $\mathbf{U} = \{(1,-1),(-1,2)\}$ dan geldt dus dat $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2)$.



Figuur 2: Twee verschillende bases genereren hetzelfde rooster

Bewijs. Stel $\mathbf{B}_1, \mathbf{B}_2$ genereren hetzelfde rooster, ofwel $\mathbf{B}_1 \cdot \mathbb{Z}^n = \mathbf{B}_2 \cdot \mathbb{Z}^n$. Dit betekent dat er $\mathbf{U}, \mathbf{V} \in \mathbb{Z}^{n \times n}$ bestaan zodanig dat $\mathbf{B}_1 = \mathbf{B}_2\mathbf{U}$ en $\mathbf{B}_2 = \mathbf{B}_1\mathbf{V} = \mathbf{B}_2\mathbf{UV}$. Uit $\mathbf{B}_2(\mathbf{I} - \mathbf{UV}) = 0$ volgt dat $\mathbf{UV} = \mathbf{I}$, want $\mathbf{B}_2$ is inverteerbaar. Er volgt nu dat $\det(\mathbf{U})\det(\mathbf{V}) = \det(\mathbf{UV}) = \det(\mathbf{I}) = 1$. Omdat $\mathbf{U}, \mathbf{V} \in \mathbb{Z}^{n \times n}$ geldt dat $\det(\mathbf{U}), \det(\mathbf{V}) \in \mathbb{Z}$. Dus $\det(\mathbf{U}) = \det(\mathbf{V}) = \pm 1$. Dus er bestaat een unimodulaire matrix $\mathbf{U}$ zodat $\mathbf{B}_1 = \mathbf{B}_2\mathbf{U}$.
Stel nu dat $\mathbf{B}_1 = \mathbf{B}_2\mathbf{U}$ voor een unimodulaire matrix $\mathbf{U}$. Omdat $\det(\mathbf{U}) \neq 0$ bestaat $\mathbf{U}^{-1}$. We laten eerst zien dat $\mathbf{U}^{-1} \in \mathbb{Z}^{n \times n}$. Er geldt dat $\mathbf{U}^{-1} = \frac{1}{\det(\mathbf{U})}\text{adj}(\mathbf{U})$. Per definitie van de geadjungeerde matrix bevat $\text{adj}(\mathbf{U})$ alleen maar gehele getallen als $\mathbf{U} \in \mathbb{Z}^{n \times n}$, en omdat $\frac{1}{\det(\mathbf{U})} = \pm 1$ geldt dat $\mathbf{U}^{-1} \in \mathbb{Z}^{n \times n}$.

We kunnen nu het bewijs afmaken. We weten dat $\mathbf{B}_1 = \mathbf{B}_2\mathbf{U}$, dus $\mathbf{B}_2 = \mathbf{B}_1\mathbf{U}^{-1}$, waarbij $\mathbf{U}, \mathbf{U}^{-1} \in \mathbb{Z}^{n\times n}$. Er geldt $\mathcal{L}(\mathbf{B}_1) = \{\mathbf{B}_1\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\} = \{\mathbf{B}_2\mathbf{U}\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\} \subset \{\mathbf{B}_2\mathbf{y} \mid \mathbf{y} \in \mathbb{Z}^n\} = \mathcal{L}(\mathbf{B}_2)$ en $\mathcal{L}(\mathbf{B}_2) = \{\mathbf{B}_2\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\} = \{\mathbf{B}_1\mathbf{U}^{-1}\mathbf{x} \mid \mathbf{x} \in \mathbb{Z}^n\} \subset \{\mathbf{B}_1\mathbf{y} \mid \mathbf{y} \in \mathbb{Z}^n\} = \mathcal{L}(\mathbf{B}_1)$. Dus $\mathcal{L}(\mathbf{B}_1) = \mathcal{L}(\mathbf{B}_2)$. □

Definitie 3.4. De rang k van een rooster $\mathcal{L} \subset \mathbb{R}^n$ is de dimensie van $\text{span}(\mathcal{L})$. Als $k = n$ heeft het rooster een volledige rang.

Hier bekijken we alleen roosters met volledige rang.

Definitie 3.5. Het fundamenteel parallellepipedum van een rooster $\mathcal{L}$ met basis $\mathbf{B}$ is $\mathcal{P}(\mathbf{B}) := \mathbf{B} \cdot [-\frac{1}{2}, \frac{1}{2})^n = \{\sum_{i=1}^n x_i\mathbf{b}_i \mid x_i \in [-\frac{1}{2}, \frac{1}{2})\}$.

$\mathcal{P}(\mathbf{B})$ hangt niet alleen van het rooster af, maar ook van de basis $\mathbf{B}$. In figuur 2 zijn de grijze oppervlakken de fundamentele parallellepidedums van $\mathbf{B}_1$ en $\mathbf{B}_2$. Er geldt dat translaties van parallellepipedums door elementen van het rooster heel $\mathbb{R}^n$ overdekken zonder overlap:

Lemma 3.6. $\mathbb{R}^n = \bigcup_{\mathbf{v}\in\mathcal{L}} (\mathbf{v} + \mathcal{P}(\mathbf{B}))$.

Bewijs. We schrijven $\lfloor a \rceil$ voor de afronding van a naar het dichtstbijzijnde gehele getal, waarbij een half wordt afgerond naar boven. Dus $-\frac{1}{2} \leq a - \lfloor a \rceil < \frac{1}{2}$.
Neem een willekeurige $p \in \mathbb{R}^n$. Er geldt

$$p = \sum_{i=1}^n x_i\mathbf{b}_i = \sum_{i=1}^n \lfloor x_i \rceil \mathbf{b}_i + \sum_{i=1}^n (x_i - \lfloor x_i \rceil)\mathbf{b}_i.$$

Omdat $\sum_{i=1}^n \lfloor x_i \rceil \mathbf{b}_i \in \mathcal{L}$ en $\sum_{i=1}^n (x_i - \lfloor x_i \rceil)\mathbf{b}_i \in \mathcal{P}(\mathbf{B})$ geldt $\mathbb{R}^n = \bigcup_{\mathbf{v}\in\mathcal{L}} (\mathbf{v} + \mathcal{P}(\mathbf{B}))$.
We moeten nu nog laten zien dat er geen overlap is. Stel voor $\mathbf{v} \neq \mathbf{w} \in \mathcal{L}$ geldt $(\mathbf{v}+\mathcal{P}(\mathbf{B}))\cap(\mathbf{w}+\mathcal{P}(\mathbf{B})) \neq \emptyset$. Dan $\mathbf{v} + \alpha = \mathbf{w} + \beta$ voor $\alpha, \beta \in \mathcal{P}(\mathbf{B})$, dus $\mathbf{v} - \mathbf{w} = \beta - \alpha$. Omdat $\mathbf{v} - \mathbf{w} \in \mathcal{L}$ is $\mathbf{v} - \mathbf{w} = \sum_{i=1}^n z_i\mathbf{b}_i$, waarbij $z_i \in \mathbb{Z}$ voor elke i. Daarnaast hebben we $\beta - \alpha = \sum_{i=1}^n \beta_i\mathbf{b}_i - \sum_{i=1}^n \alpha_i\mathbf{b}_i = \sum_{i=1}^n (\beta_i - \alpha_i)\mathbf{b}_i$, waarbij voor elke i geldt $\alpha_i, \beta_i \in [-\frac{1}{2}, \frac{1}{2})$, ofwel $\beta_i - \alpha_i \in (-1, 1)$. Maar voor elke i geldt $\beta_i - \alpha_i = z_i$, dus $z_i = 0$. Hieruit volgt dat $\mathbf{v} = \mathbf{w}$. □

Definitie 3.7. De determinant van een rooster $\mathcal{L}$ is $\det(\mathcal{L}) := \text{vol}(\mathcal{P}(\mathbf{B})) = |\det(\mathbf{B})|$.

Definitie 3.8. De duale van een rooster $\mathcal{L}$ is $\mathcal{L}^* := \{\mathbf{v} \in \mathbb{R}^n \mid \langle \mathbf{v}, \mathbf{w} \rangle \in \mathbb{Z}\ \forall \mathbf{w} \in \mathcal{L}\}$ waarbij $\langle ., . \rangle$ het standaardinproduct is.

Voorbeeld. Er geldt dat $(\mathbb{Z}^n)^* = \mathbb{Z}^n$. Voor elke $c \in \mathbb{R} \setminus \{0\}$ en voor elk rooster $\mathcal{L}$ geldt $(c\mathcal{L})^* = \frac{1}{c}\mathcal{L}^*$.

Lemma 3.9. *Laat $\mathcal{L}(\mathbf{B})$ een rooster, dan is $\mathcal{L}^*$ ook een rooster.*

Bewijs. We moeten laten zien dat $\mathcal{L}^*$ een additieve ondergroep van $\mathbb{R}^n$ is en dat het discreet is. Dat het een additieve ondergroep is volgt direct uit de lineariteit van het inproduct en het feit dat $\mathbf{0} \in \mathcal{L}^*$. Dat $\mathcal{L}^*$ discreet is, is minder duidelijk. We willen bewijzen dat er een constante $c > 0$ bestaat zodanig dat $\forall \mathbf{v}, \mathbf{w} \in \mathcal{L}^*$ geldt $\mathbf{v} = \mathbf{w}$ of $\|\mathbf{v} - \mathbf{w}\| \geq c$. Laat $\mathbf{v}, \mathbf{w} \in \mathcal{L}^*$ en stel dat $\langle \mathbf{v}, \mathbf{b}_i \rangle = \langle \mathbf{w}, \mathbf{b}_i \rangle$ voor $i = 1, \ldots, n$, waar $\mathbf{b}_i \in \mathbf{B}$. Dan $\forall \mathbf{x} \in \mathbb{R}^n$ $x = \alpha_1\mathbf{b}_1 + \cdots + \alpha_n\mathbf{b}_n$ voor zekere $\alpha_i \in \mathbb{R}$ omdat de rang volledig is. Dus

$\forall \mathbf{x} \in \mathbb{R}^n$

$$\begin{aligned}\langle \mathbf{v}-\mathbf{w}, \mathbf{x}\rangle &= \langle \mathbf{v}, \mathbf{x}\rangle - \langle \mathbf{w}, \mathbf{x}\rangle \\ &= \langle \mathbf{v}, \sum_{i=1}^{n} \alpha_i \mathbf{b}_i\rangle - \langle \mathbf{w}, \sum_{i=1}^{n} \alpha_i \mathbf{b}_i\rangle \\ &= \sum_{i=1}^{n} \alpha_i \langle \mathbf{v}, \mathbf{b}_i\rangle - \sum_{i=1}^{n} \alpha_i \langle \mathbf{w}, \mathbf{b}_i\rangle \\ &= \sum_{i=1}^{n} \alpha_i (\langle \mathbf{v}, \mathbf{b}_i\rangle - \langle \mathbf{w}, \mathbf{b}_i\rangle) \\ &= 0\end{aligned}$$

dus $\mathbf{v}-\mathbf{w}=0$ ofwel $\mathbf{v}=\mathbf{w}$. Dus stel dat er een $i \in \{1,\ldots,n\}$ is zodanig dat $\langle \mathbf{v}, \mathbf{b}_i\rangle \neq \langle \mathbf{w}, \mathbf{b}_i\rangle$, dan $0 \neq \langle \mathbf{v}-\mathbf{w}, \mathbf{b}_i\rangle \in \mathbb{Z}$. We definiëren $P_i = \{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{x}, \mathbf{b}_i\rangle = 0\}$, dit is een lineaire deelruimte van $\mathbb{R}^n$ en voor elke $k \in \mathbb{Z}$ met $k \neq 0$ geldt dat $\{\mathbf{x} \in \mathbb{R}^n \mid \langle \mathbf{x}, \mathbf{b}_i\rangle = k\} = \mathbf{z}_{i,k} + P_i$ waar $\mathbf{z}_{i,k} = \frac{k}{\|\mathbf{b}_i\|^2}\mathbf{b}_i$. Immers, $\langle \frac{k}{\|\mathbf{b}_i\|^2}\mathbf{b}_i, \mathbf{b}_i\rangle = \frac{k}{\|\mathbf{b}_i\|^2}\langle \mathbf{b}_i, \mathbf{b}_i\rangle = k$ en uit de driehoeksongelijkheid volgt dat voor elke $\mathbf{x} \in \mathbf{z}_{i,k} + P_i$ geldt dat $\|\mathbf{x}\| \geq \|\mathbf{z}_{i,k}\|$ en $\langle \mathbf{x}, \mathbf{b}_i\rangle = \langle \mathbf{z}_{i,k}, \mathbf{b}_i\rangle = k$. Dus als $\mathbf{x} \in \mathcal{L}^*$ en $\langle \mathbf{x}, \mathbf{b}_i\rangle = k \neq 0$ dan is $\|\mathbf{x}\| \geq \|\mathbf{z}_{i,k}\|$ en $\|\mathbf{z}_{i,k}\| = \sqrt{\langle \mathbf{z}_{i,k}, \mathbf{z}_{i,k}\rangle} = |\frac{k}{\|\mathbf{b}_i\|}| \geq \frac{1}{\|\mathbf{b}_i\|}$. Laat nu $c = \min_{1\leq i\leq n} \frac{1}{\|\mathbf{b}_i\|}$. Nu geldt voor elke $\mathbf{x} \in \mathcal{L}^*$ dat $\|\mathbf{x}\| \geq c$. Dus in het bijzonder geldt dit voor $\mathbf{v}-\mathbf{w}$. Dus $\mathcal{L}^*$ is discreet. □

Er geldt $\det(\mathcal{L}^*) = \frac{1}{\det(\mathcal{L})}$ omdat $\mathcal{L}^* = (\mathcal{L}^T)^{-1}$.

Definitie 3.10. De kortste afstand van een rooster $\mathcal{L}$ is $\lambda_1(\mathcal{L}) := \min_{\mathbf{v}\in\mathcal{L}\setminus\mathbf{0}} \|\mathbf{v}\|$. Het i-de opvolgende minimum is $\lambda_i(\mathcal{L}) := \min\{r \mid \mathcal{L} \text{ bevat } i \text{ lineair onafhankelijke vectoren van lengte } \leq r\}$.

Dan geldt $\lambda_1(\mathcal{L}) \leq \lambda_2(\mathcal{L}) \leq \cdots \leq \lambda_n(\mathcal{L})$.

Lemma 3.11. *Voor elk n-dimensionaal rooster $\mathcal{L}$ geldt $1 \leq \lambda_1(\mathcal{L}) \cdot \lambda_n(\mathcal{L}^*) \leq n$.*

Bewijs. We bewijzen hier alleen de eerste ongelijkheid. Neem een $\mathbf{v} \in \mathcal{L}$ zodanig dat $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$. Neem n willekeurige lineair onafhankelijke vectoren $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathcal{L}^*$. Omdat ze niet allemaal orthogonaal kunnen zijn aan $\mathbf{v}$ bestaat er een i zodanig dat $\langle \mathbf{x}_i, \mathbf{v}\rangle \neq 0$. Per definitie van $\mathcal{L}^*$ geldt dat $\langle \mathbf{x}_i, \mathbf{v}\rangle \in \mathbb{Z}$ en dus $|\langle \mathbf{x}_i, \mathbf{v}\rangle| \geq 1$. Met de ongelijkheid van Cauchy-Schwarz volgt $1 \leq |\langle \mathbf{x}_i, \mathbf{v}\rangle| \leq \|\mathbf{x}_i\| \cdot \|\mathbf{v}\| \leq \lambda_1(\mathcal{L}) \cdot \lambda_n(\mathcal{L}^*)$. De tweede ongelijkheid wordt bewezen door Banaszczyk [Ban93] met inductie naar n. □

We kunnen op een rooster $\mathcal{L}$ een discrete normale verdeling $D_{\mathcal{L},r}$ maken, waarbij $\mu = \mathbf{0}$ en $\sigma = r$. De kans op $\mathbf{x} \in \mathcal{L}$ bij een trekking uit deze kansverdeling is

$$D_{\mathcal{L},r} = \frac{\rho_r(\mathbf{x})}{\rho_r(\mathcal{L})}, \text{ waarbij } \rho_r(\mathbf{x}) = e^{-\pi\|\mathbf{x}/r\|^2} \text{ en } \rho_r(\mathcal{L}) = \sum_{\mathbf{x}\in\mathcal{L}} \rho_r(\mathbf{x}).$$

Micciancio en Regev [MR07] hebben een parameter voor een rooster gedefinieerd die de maximale σ geeft zodanig dat $D_{\mathcal{L},r}$ zich gedraagt als een continue normale verdeling.

Definitie 3.12. Voor een n-dimensionaal rooster $\mathcal{L}$ en $\epsilon \in \mathbb{R}_{>0}$ definiëren we de parameter $\eta_\epsilon(\mathcal{L})$ als de kleinste r zodanig dat $\rho_{1/r}(\mathcal{L}^* \setminus \{\mathbf{0}\}) \leq \epsilon$.

Hierbij is ϵ doorgaans een te verwaarlozen functie op n. Micciancio en Regev bewezen dat voor $\epsilon = 2^{-n}$ voor elk n-dimensionaal rooster $\mathcal{L}$ geldt dat $\eta_\epsilon(\mathcal{L}) \leq \sqrt{n}/\lambda_1(\mathcal{L}^*)$. Er is ook een ondergrens voor deze parameter:

Lemma 3.13. *Voor elke n-dimensionaal rooster $\mathcal{L}$ en $\epsilon \in \mathbb{R}_{>0}$ geldt*

$$\eta_\epsilon(\mathcal{L}) \geq \sqrt{\frac{\ln 1/\epsilon}{\pi}} \cdot \frac{1}{\lambda_1(\mathcal{L}^*)} \geq \sqrt{\frac{\ln 1/\epsilon}{\pi}} \cdot \frac{\lambda_n(\mathcal{L})}{n}.$$

Bewijs. Neem een $\mathbf{v} \in \mathcal{L}^*$ zodanig dat $\|\mathbf{v}\| = \lambda_1(\mathcal{L}^*)$ en laat $r = \eta_\epsilon(\mathcal{L})$. Dan

$$\epsilon \geq \rho_{1/r}(\mathcal{L}^* \setminus \mathbf{0}) = \sum_{\mathbf{x} \in \mathcal{L}^* \setminus \{\mathbf{0}\}} \rho_{1/r}(\mathbf{x}) \geq \rho_{1/r}(\mathbf{v}) = e^{-\pi \|r\mathbf{v}\|^2}$$

Als we dit oplossen naar r krijgen we de eerste ongelijkheid. De tweede ongelijkheid volgt uit lemma 3.11. □

Roosterproblemen

Algebraïsche vragen op roosters, zoals nagaan of een vector in het rooster zit, zijn makkelijk. Dat wil zeggen dat ze binnen polynomiale tijd kunnen worden opgelost. Geometrische vragen, waarvan er hieronder een aantal worden besproken, zijn moeilijk. Dit is wat roosters zo interessant maakt. Het kortste vectorprobleem is gedefinieerd als volgt:

Definitie 3.14. Shortest Vector Problem (SVP): Gegeven een willekeurige basis $\mathbf{B}$ van een rooster $\mathcal{L}$, vind $\mathbf{v} \in \mathcal{L}$ zodanig dat $\|\mathbf{v}\| = \lambda_1(\mathcal{L})$.

Het meest efficiënte algoritme is exponentieel, de tijdscomplexiteit is $2^{0.3774n}$ en de ruimtecomplexiteit is $2^{0.2925n}$ [BGJ13].
Naast dit probleem bestaat er ook een benaderingsprobleem. Dit probleem wordt geparametriseerd door een $\gamma \geq 1$ die afhangt van de dimensie n, dus $\gamma = \gamma(n)$.

Definitie 3.15. Approximate Shortest Vector Problem (SVP_γ): Gegeven een willekeurige basis $\mathbf{B}$ van een n-dimensionaal rooster $\mathcal{L}$, vind $\mathbf{v} \in \mathcal{L}$ zodanig dat $\|\mathbf{v}\| \leq \gamma(n) \cdot \lambda_1(\mathcal{L})$.

Voor $\gamma(n) = 2^{\Omega(n)}$ kan het LLL-algoritme [LLL82] gebruikt worden om het probleem efficiënt op te lossen. Voor kleinere γ kan het BKZ-algoritme [CN11] worden gebruikt maar de complexiteit blijft exponentieel omdat het gebruik maakt van het algoritme om het kortste vectorprobleem op te lossen.
Van SVP_γ bestaat ook een beslissingsvariant:

Definitie 3.16. Decisional Approximate Shortest Vector Problem (GapSVP_γ): Gegeven een willekeurige basis $\mathbf{B}$ van een n-dimensionaal rooster $\mathcal{L}$ en een $d \in \mathbb{R}_{>0}$, bepaal of $\lambda_1(\mathcal{L}) \leq d$ of $\lambda_1(\mathcal{L}) \geq \gamma(n) \cdot d$.

Een ander bekend roosterprobleem is het kortste onafhankelijke vectorprobleem:

Definitie 3.17. Approximate Shortest Independent Vector Problem (SIVP_γ): Gegeven een willekeurige basis $\mathbf{B}$ van een n-dimensionaal rooster $\mathcal{L}$, geef n lineair onafhankelijke $\{\mathbf{s}_1 \dots \mathbf{s}_n\} \subset \mathcal{L}$ zodanig dat $\|\mathbf{s}_i\| \leq \gamma(n) \cdot \lambda_n(\mathcal{L})$ voor alle i.

GapSVP_γ is NP-moeilijk voor kleine γ en makkelijk voor $\gamma = 2^{\mathcal{O}(n)}$ [LLL82]. Hetzelfde geldt voor het onderstaande probleem SIVP_γ. Voor $\gamma = \text{poly}(n)$ bestaat er een vermoeden dat er geen efficiënt algoritme is om deze problemen op te lossen. Ook bestaat er een vermoeden dat er voor deze γ geen efficiënt kwantumalgoritme is om GapSVP_γ en SIVP_γ op te lossen. Het enige bewijs dat dit vermoeden steunt is dat er geen kwantumalgoritmen bekend zijn voor roosterproblemen die essentieel beter zijn dan gewone algoritmen. Dit is een van de grootste open problemen in de wereld van kwantumalgoritmen.
Tot slot worden de volgende problemen genoemd omdat ze nodig zijn voor het volgende hoofdstuk.

Een generalisatie van het SIVP_γ is het Generalized Independent Vector Problem $\text{GIVP}_\gamma^\varphi$, hierbij wordt in definitie 3.17 $\lambda_n(\mathcal{L})$ vervangen door $\varphi(\mathcal{L})$, een willekeurige functie die een rooster $\mathcal{L}$ afbeeldt op een reëel getal. Deze functie komt ook voor in het volgende probleem:

Definitie 3.18. Discrete Gaussian Sampling Problem (DGS_φ): Gegeven een willekeurige basis $\mathbf{B}$ van een n-dimensionaal rooster $\mathcal{L}$ en een $r > \varphi(\mathcal{L})$, geef een trekking uit $D_{\mathcal{L},r}$.

Tot slot hebben we nog het volgende roosterprobleem dat voor een willekeurige $t \in \mathbb{R}^n$ de dichtstbijzijnde vector in het rooster vindt:

Definitie 3.19. Closest Vector Problem ($\text{CVP}_{\mathcal{L},d}$): Gegeven een willekeurige basis $\mathbf{B}$ van een n-dimensionaal rooster $\mathcal{L}$ en een punt $t \in \mathbb{R}^n$ waarvoor geldt dat $\min_{\mathbf{x}\in\mathcal{L}} \|\mathbf{t} - \mathbf{x}\| < d = \lambda_1(\mathcal{L})/2$, vind de unieke $\mathbf{v} \in \mathcal{L}$ zodanig dat $\|\mathbf{t} - \mathbf{v}\| < d$.

4 De moeilijkheid van Learning with Errors

In dit hoofdstuk wordt een schets gegeven van het bewijs dat Learning with Errors moeilijk is. Het bewijs is gevonden door Regev [Reg09]. Het idee is dat we laten zien dat als er een efficiënt algoritme bestaat dat Learning with Errors oplost, er dan een efficiënt kwantumalgoritme bestaat dat GapSVP_γ en SIVP_γ oplost. Een efficiënt kwantumalgoritme is een algoritme dat door een kwantumcomputer binnen polynomiale tijd kan worden uitgevoerd. Maar omdat het erg onwaarschijnlijk is dat er een kwantumalgoritme bestaat voor GapSVP_γ en SIVP_γ dat efficiënt is, volgt hieruit dat Learning with Errors waarschijnlijk geen efficiënt algoritme heeft en dus moeilijk is. Als onderdeel van het bewijs heeft Regev laten zien dat als we DGS_φ kunnen oplossen, we ook GapSVP_γ en SIVP_γ kunnen oplossen. Dus om te bewijzen dat Learning with Errors moeilijk is, kunnen we het volgende bewijzen:

Stelling 4.1. *Laat $n, q \in \mathbb{Z}, \alpha \in (0,1)$ zodanig dat $\alpha q > 2\sqrt{n}$ en ϵ een verwaarloosbare functie op n zijn. Als er een efficiënt algoritme bestaat dat Learning with Errors oplost, dan bestaat er een efficiënt kwantumalgoritme dat $DGS_{\sqrt{2n}\cdot\eta_\epsilon(L)/\alpha}$ oplost.*

De bewijzen van de lemma's die in het bewijs van stelling 4.1 worden gebruikt zijn terug te vinden in [Reg09].

Bewijsschets. Gegeven een n-dimensionaal rooster L en een $r > \sqrt{2n}\cdot\eta_\epsilon(L)/\alpha$, willen we dat het algoritme een trekking uit de kansverdeling $D_{L,r}$ teruggeeft. Laat $r_i = r \cdot (\alpha q/\sqrt{n})^i$. Het algoritme begint met n^c trekkingen te doen uit $D_{L,r_{3n}}$, waar $c > 0$ een constante is. Dit kan efficiënt vanwege het onderstaande lemma, dat we mogen toepassen omdat uit lemma 3.13 volgt dat

$$r_{3n} = r\cdot(\tfrac{\alpha q}{\sqrt{n}})^{3n} > 2^{3n} r > 2^{3n}\cdot\sqrt{2n}\cdot\tfrac{\eta_\epsilon(L)}{\alpha} > 2^{3n}\cdot\sqrt{2n}\cdot\tfrac{\lambda_n(L)}{n\alpha} > 2^{3n}\cdot\tfrac{1}{\alpha\sqrt{n}}\cdot\lambda_n(L) > 2^{3n}\cdot\tfrac{1}{2^n}\cdot\lambda_n(L) = 2^{2n}\lambda_n(L).$$

Lemma 4.2. *Er bestaat een efficiënt algoritme dat, gegeven een n-dimensionaal rooster L en $r > 2^{2n}\lambda_n(L)$, een trekking uit de kansverdeling doet die binnen statistische afstand $2^{-\Omega(n)}$ van $D_{L,r}$ ligt.*

De statistische afstand tussen twee kansverdelingen X en Y op D, met dichtheidsfuncties f_X en f_Y, is als volgt gedefinieerd:

$$\Delta(X,Y) := \max_{A\subset D} |f_X(A) - f_Y(A)|.$$

Nu komen we bij de kern van het algoritme, de iteratieve stap. Deze geven we als lemma:

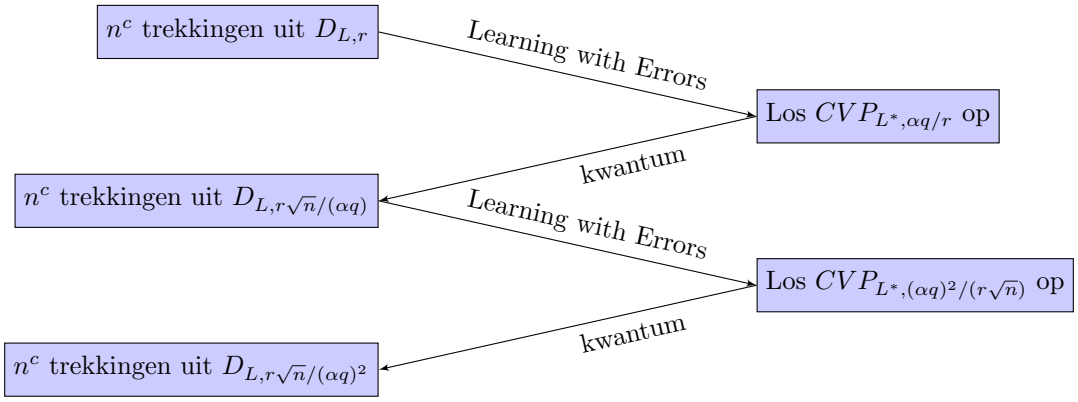
Lemma 4.3. *Laat $n, q \in \mathbb{Z}, \alpha \in (0,1)$ zodanig dat $\alpha q > 2\sqrt{n}$ en laat ϵ een verwaarloosbare functie op n zijn. Als er een efficiënt algoritme bestaat dat Learning with Errors oplost, dan is er een constante $c > 0$ en bestaat er een efficiënt kwantumalgoritme dat, gegeven een n-dimensionaal rooster L, $r > \sqrt{2}q\eta_\epsilon(L)$ en n^c trekkingen uit $D_{L,r}$, een trekking uit $D_{L,r\sqrt{n}/(\alpha q)}$ genereert.*

Vervolgens gebruiken we de n^c trekkingen uit D_{L,r_i} om n^c trekkingen uit $D_{L,r_{i-1}}$ te genereren, voor $i = 3n, 3n-1, \ldots, 1$. Hiervoor wordt de iteratieve stap gebruikt. We mogen lemma 4.3 toepassen omdat voor elke $i \geq 1$ geldt $r_i \geq r_1 = r\cdot\alpha q/\sqrt{n} > \sqrt{2n}\cdot\eta_\epsilon(L)/\alpha\cdot\alpha q/\sqrt{n} = \sqrt{2}q\eta_\epsilon(L)$. Uiteindelijk hebben we n^c trekkingen uit $D_{L,r_0} = D_{L,r}$ en hiervan kunnen we er een pakken.

We kunnen de iteratieve stap nog iets gedetailleerder beschrijven. Het algoritme dat we zoeken bestaat uit twee stappen, weergegeven in figuur 3. De eerste stap is het volgende lemma:

Lemma 4.4. *Laat $n, q \in \mathbb{Z}, \alpha \in (0,1)$ zodanig dat $\alpha q > 2\sqrt{n}$ en laat ϵ een verwaarloosbare functie op n zijn. Als er een efficiënt algoritme bestaat dat Learning with Errors oplost, dan is er een constante $c > 0$ en bestaat er een efficiënt algoritme dat, gegeven een n-dimensionaal rooster L, $r > \sqrt{2}q\eta_\epsilon(L)$ en n^c trekkingen uit $D_{L,r}$, $CVP_{L^*,\alpha q/(\sqrt{2}r)}$ oplost.*

In het bewijs van dit lemma wordt het algoritme dat Learning with Errors oplost gebruikt. In de tweede stap gebruiken we het algoritme uit lemma 4.4 om trekkingen uit $D_{L,r\sqrt{n}/(\alpha q)}$ te genereren. Dit deel is kwantum. Voor meer details van het bewijs wordt verwezen naar [Reg09].



Figuur 3: De eerste stappen van de iteratieve stap

□

Peikert [Pei09] heeft bewezen dat de kwantumreductie ook klassiek kan. De klassieke reductie werkt alleen voor GapSVP en niet voor SIVP. Dit gaat wel ten koste van de efficiëntie, omdat er nu moet gelden dat $q \geq 2^{n/2}$, terwijl de kwantumreductie geldt voor $q \geq 2\sqrt{n}/\alpha$. Een grotere modulus betekent dat trekkingen uit $A_{\mathbf{s},\chi}$ groter zijn in het aantal bits, daardoor worden de sleutels in de cryptografische systemen groter en dit gaat ten koste van de efficiëntie van het systeem.

Referenties

[App+09] B. Applebaum e.a. *Fast cryptographic primitives and circularsecure encryption based on hard learning problems*. In CRYPTO, pages 595–618. 2009.

[Ban93] W. Banaszczyk. *New bounds in some transference theorems in the geometry of numbers*. Mathematische Annalen, 296(4):625–635. 1993.

[BGJ13] A. Becker, N. Gama en A. Joux. *Solving shortest and closest vector problems: The decomposition approach*. Cryptology ePrint Archive, Report 2013/685. `https://eprint.iacr.org/2013/685`. 2013.

[BKW03] A. Blum, A. Kalai en H. Wasserman. *Noise-tolerant learning, the parity problem, and the statistical query model*. Journal of the ACM, 50(4):506–519. 2003.

[CN11] Y. Chen en P. Q. Nguyen. *BKZ 2.0: Better Lattice Security Estimates*. Advances in Cryptology – ASIACRYPT 2011. Springer, Berlin, Heidelberg: 1–20. 2011.

[Hås+99] J. Håstad e.a. *A pseudorandom generator from any one-way function*. SIAM Journal on Computing, 28(4):1364–1396. 1999.

[LLL82] A. K. Lenstra, H.W. Lenstra Jr. en L. Lovász. *Factoring polynomials with rational coefficients*. Mathematische Annalen, 261(4):515–534. 1982.

[LP10] R. Lindner en C. Peikert. *Better Key Sizes (and Attacks) for LWE-Based Encryption*. Cryptology ePrint Archive, Report 2010/613. `https://eprint.iacr.org/2010/613`. 2010.

[MR07] D. Micciancio en O. Regev. *Worst-case to average-case reductions based on Gaussian measures*. SIAM Journal on Computing, 37(1):267-302. 2007.

[Pei09] C. Peikert. *Public-Key Cryptosystems from the Worst-Case Shortest Vector Problem*. In Proc. 41st ACM Symp. on Theory of Computing (STOC). 2009.

[PW11] C. Peikert en B. Waters. *Lossy trapdoor functions and their applications*. SIAM Journal on Computing, 40(6):1803–1844. Preliminary version in STOC 2008. 2011.

[Reg09] O. Regev. *On lattices, learning with errors, random linear codes, and cryptography*. Journal of the ACM, 56(6):1–40. Preliminary version in STOC 2005. 2009.