Radboud Universiteit Nijmegen

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Quantumcryptografie

Systemen brekende algoritmes

Auteur:
Sandra van Dijk
(4063252)

Begeleider:
Wieb Bosma

Versie 2

27 augustus 2014

Inhoudsopgave

Samenvatting

Quantumcyptografie is de titel van deze scriptie en hierin zal dan ook bekeken worden wat er gebeurt met de klassieke cryptografie als de quantumcomputer zou bestaan. Er worden hierin twee klassieke cryptografie systemen besproken en daarnaast de redenen waarom deze in gevaar zouden komen.

Het eerste systeem is RSA. Een asymmetrisch cryptografie systeem dat zijn veiligheid kan garanderen door de moeilijkheidsgraad van de priemfactorisering van grote gehele getallen. Peter Shor, een Amerikaanse wiskundige, heeft een quantumalgoritme geschreven dat het factoriseren uitvoert binnen polynomiale tijd. Shor's algoritme voor factoriseren kan hiermee meerdere systemen breken die van dit principe afhangen.

Het tweede systeem dat wordt besproken is het Diffie-Hellman sleuteluitwisselingsprotocol. Ook dit is een asymmetrisch cryptografie systeem, maar hierbij wordt de veiligheid door iets anders gegarandeerd. Het maakt gebruik van het feit dat discrete logaritmes moeilijk te berekenen zijn. Shor heeft hiervoor een quantumalgoritme geschreven waarbij deze weer binnen polynomiale tijd het mogelijk maakt om de discrete logaritmes te vinden. Er zal alleen een schets van het algoritme worden gegeven.

Van beide systemen worden de algoritmes besproken en vervolgens de quantumalgoritmes stap voor stap uitgelegd. Daarnaast worden er verschillende stellingen, lemma's en onderwerpen besproken die nodig zijn om de quantumalgoritmes te begrijpen.

1 Klassieke cryptografie

Cryptografie is het versleutelen en ontsleutelen van informatie zodat een ongewenste afluisteraar het bericht niet kan begrijpen. Deze berichten gaan van een zender naar een ontvanger die berichten willen uitwisselen. Traditioneel gezien heten deze hoofdpersonen in de cryptografie Alice en Bob die geheime berichten naar elkaar willen versturen. Als buitenstaander die heel graag wil weten wat zij elkaar sturen, is er Eve. De cryptografie systemen zijn ervoor bedoeld dat zij deze berichten niet kan lezen.

Hieronder staan de algoritmes van twee asymmetrische cryptografie systemen. Het asymmetrische hieraan is dat er gebruik wordt gemaakt van twee sleutels; één voor het versleutelen van het bericht en een tweede voor het ontsleutelen. Beide systemen worden als veilig beschouwd door de moeilijkheidsgraad van priemfactorisering of het vinden van een discrete logaritme.

1.1 RSA

Het veelgebruikte cryptografie systeem RSA is een asymmetrische encryptie-algoritme dat gebruik maakt van publieke en geheime sleutels. Het algoritme werd ontworpen door Ron Rivest, Adi Shamir en Len Adleman in 1977 en wordt veel gebruikt in handel via internet. Er wordt gebruik gemaakt van het feit dat het moeilijk is om erg grote getallen op te breken in priemfactoren. Voor verdere verwijzingen wordt het algoritme eerst zelf bekeken.

Algoritme 1.1

- *Kies twee grote priemgetallen* $p, q \in \mathbb{N}$*, waarbij* $p \neq q$*. Bereken* $N = p \cdot q$*.*
- *Kies een geheel getal* $1 < e < \varphi(N)$ *dat relatief priem is met* $\varphi(N) = (p-1)(q-1)$*. Dit is mogelijk met het uitgebreid Euclidisch algoritme.*
- *Bereken* d *zodanig dat* $e \cdot d \equiv 1 \bmod (p-1)(q-1)$*. Ook dit is mogelijk door het uitgebreid Euclidisch algoritme.*
- *Combinatie* (N, e) *wordt de publieke sleutel, die over onbeveiligde systemen verstuurd kan worden. De combinatie* (N, d) *wordt de geheime sleutel. Deze zal gebruikt worden voor het ontcijferen van de vercijferde berichten.*

Stel Bob wil een geheim bericht naar Alice sturen. Alice wil hiervoor RSA gebruiken en gebruikt eerst het algoritme om aan de juiste sleutels te komen. Bob zal de combinatie (N, e) hebben en kan hiermee zijn bericht versleutelen en doorsturen. Alice kan met de geheime sleutel het bericht ontcijferen.

Dit werkt als volgt:

- Bob wil bericht m versturen en vercijfert deze als $n \equiv m^e \bmod N$.
- Bob verstuurt n naar Alice.
- Alice ontvangt n en ontcijfert deze: $n^d \equiv (m^e)^d \equiv m^{(e \cdot d)} \equiv m \bmod N$.

Omdat e en d inverses zijn modulo $(p-1)(q-1)$, werkt dit algoritme. Want als

$$e \cdot d \equiv 1 \bmod (p-1)(q-1) \quad \text{dan} \quad e \cdot d \equiv 1 \bmod p-1 \quad \text{en} \quad e \cdot d \equiv 1 \bmod q-1.$$

Dan geldt volgens de Kleine stelling van Fermat:

$$n^{e \cdot d} \equiv n \bmod p \quad \text{en} \quad n^{e \cdot d} \equiv n \bmod q.$$

En omdat p en q verschillende priemgetallen zijn, geldt:

$$n^{e \cdot d} \equiv n \bmod p \cdot q.$$

Het algoritme in gebruik, is te zien in onderstaand voorbeeld.

Voorbeeld 1.2 Neem priemgetallen $p = 11$ en $q = 37$. Dat betekent dat $N = p \cdot q = 11 \cdot 37 = 407$ en $(p-1)(q-1) = 10 \cdot 36 = 360$. Neem een e die copriem is ten opzichte van 360, bijvoorbeeld $e = 7$. Dan wordt er een d gezocht, zodat $e \cdot d \equiv 1 \bmod 360$. $d = 103$ voldoet hieraan.

Stel Bob wil het bericht $m = 104$ naar Alice versturen. Deze vercijfert hij door $m^e = 104^7 \equiv 3 \bmod 407$. Dus hij verstuurt het bericht $n = 3$. Alice ontvangt dit en ontcijfert dit als volgt: $n^d = 3^{103} \equiv 104 \bmod 407$. En krijgt hiermee het originele bericht $m = 104$ terug.

1.1.1 Priemfactorisatie

Volgens de hoofdstelling van de rekenkunde is het mogelijk om elk positief geheel getal groter dan 1, te schrijven als een uniek product van priemgetallen. Als men een dergelijk getal schrijft als product van priemgetallen, noemt men dat priemfactorisatie. Voor deze ontbinding is er geen snel algoritme.

In het RSA systeem wordt hier dan ook gebruik van gemaakt. Een eventuele afluisteraar Eve, kan alleen meeluisteren als zij ook de geheime sleutel weet. Stel dat zij N kan factoriseren in de priemgetallen p en q, dan zou zij dezelfde berekeningen kunnen doen als Alice en zo achter de geheime sleutel komen. Dit zorgt ervoor dat alle versleutelde berichten die over het gedeelde netwerk worden verstuurd, ontsleuteld kunnen worden door Eve.

Omdat het te moeilijk is om het openbare getal N te factoriseren in priemgetallen, wordt dit systeem bestempeld als een veilig systeem. Hiervoor wordt er wel aangeraden om erg grote priemgetallen p en q te gebruiken. Als de priemgetallen te dicht bij elkaar liggen, of als N te klein is, is het wel mogelijk om snel achter de priemfactoren te komen.

Stel dat het op een snelle manier mogelijk wordt om een priemfactorisatie te realiseren voor grote gehele getallen, dan is dit systeem niet meer betrouwbaar. In hoofdstuk 3 wordt een quantumalgoritme bekeken die dat mogelijk kan maken. Als een quantumcomputer zou bestaan, zou RSA niet meer veilig gebruikt kunnen worden.

1.2 Diffie-Hellman Sleuteluitwisselingsprotocol

Dit cryptografie systeem werd voor het eerst gepubliceerd door Whitfield Diffie en Martin Hellman in 1976. Het is een asymmetrisch sleuteluitwisselingsprotocol waarbij de veiligheid gebaseerd is op het feit dat het moeilijk is om een discrete logaritme van een getal te berekenen. Voor verdere verwijzingen, volgt hieronder eerst het algoritme zelf.

Algoritme 1.3

- *Alice en Bob kiezen een priemgetal p en een basisgetal g binnen een multiplicatieve groep G van gehele getallen modulo p. Deze p en g zijn openbaar.*
- *Alice kiest een geheime $a \in \mathbb{Z}$ en berekent $A \equiv g^a \bmod p$ en verstuurt deze naar Bob.*
- *Bob kiest een geheime $b \in \mathbb{Z}$ en berekent $B \equiv g^b \bmod p$ en verstuurt deze naar Alice.*
- *Alice ontvangt B en berekent $s \equiv B^a \bmod p$ dit wordt de geheime sleutel.*
- *Bob ontvangt A en berekent $s \equiv A^b \bmod p$, waarbij dit dezelfde geheime sleutel is.*

De geheime sleutel s is hetzelfde omdat

$$s \equiv A^b \equiv (g^a)^b \equiv g^{(a \cdot b)} \equiv g^{(b \cdot a)} \equiv (g^b)^a \equiv B^a \bmod p.$$

Een eventuele Eve, afluisteraar, kan alleen g^a en g^b gezien hebben, naast de openbare p en g. Hieruit is het niet mogelijk om $s \equiv g^{a \cdot b} \bmod p$ te berekenen.

Dit sleuteluitwisselingsprotocol kan gebruikt worden voor een bericht m door het bericht te vermenigvuldigen met de geheime sleutel: $mg^{a \cdot b}$. Als de andere persoon dit bericht ontvangt, moet het natuurlijk ontsleuteld kunnen worden. Bob weet bijvoorbeeld $|G|$ (het aantal elementen van G), b en g^a en heeft voor het ontsleutelen $(g^{a \cdot b})^{-1}$ nodig. Om deze te berekenen wordt gebruik gemaakt van het feit dat geldt $x^{|G|} \equiv 1$ voor alle $x \in G$. In dit geval geldt dat $|G| = p-1$. Het berekenen van de $(g^{a \cdot b})^{-1}$ is mogelijk omdat

$$(g^a)^{|G|-b} = g^{a(|G|-b)} = g^{a|G|-ab} = g^{a|G|} g^{-ab} = (g^{|G|})^a g^{-ab} = 1^a (g^{ab})^{-1} = (g^{ab})^{-1}.$$

Dan kan een versleuteld bericht ontsleuteld worden door $mg^{a \cdot b}(g^{a \cdot b})^{-1} = m$. Voor nog meer duidelijkheid volgt hieronder een voorbeeld van het Diffie-Hellman sleuteluitwisselingsprincipe.

Voorbeeld 1.4 Alice en Bob bepalen als priemgetal $p = 37$, groep $G = \mathbb{Z}/37\mathbb{Z}$ en basis $g = 7$.

- Alice neemt als willekeurige $a = 8$, dan is $A = g^a = 7^8 \equiv 16 \bmod 37$.
- Bob neemt als willekeurige $b = 25$, dan is $B = g^b = 7^{25} \equiv 34 \bmod 37$.

Alice ontvangt B en Bob ontvangt A. Hierdoor kunnen ze beide de geheime sleutel s berekenen:

$$s \equiv 16^{25} \equiv 12 \bmod 37 \quad \text{en} \quad s \equiv 34^8 \equiv 12 \bmod 37.$$

Stel dat Alice het bericht $m = 24$ wilt versturen. Zij vercijfert dit in het bericht $n = ms = 24 \cdot 12 \equiv 29 \bmod 37$. Bob ontvangt het bericht $n = 29$ en gaat deze ontcijferen. Hiervoor gaat hij eerst $(g^{ab})^{-1}$ berekenen:

$$(g^{ab})^{-1} = (g^a)^{|G|-b} = 16^{\phi(p)-25} = 16^{(p-1)-25} = 16^{36-25} \equiv 34 \bmod 37.$$

Het ontcijferde bericht wordt dan: $n(g^{ab})^{-1} = 29 \cdot 34 \equiv 24 \bmod 37 = m$.

1.2.1 Discrete logaritme probleem

De veiligheid van het Diffie-Hellman sleuteluitwisselingsprotocol is gebaseerd op het feit dat het moeilijk is om een discrete logaritme te vinden. Er wordt gebruik gemaakt van het volgende principe: zij p een priemgetal, dan is de multiplicatieve groep $\mathbb{Z}/p\mathbb{Z}$ een cyclische groep. Dat wil zeggen $\exists g \in \mathbb{Z}/p\mathbb{Z}$ zodat het rijtje $1, g, g^2, g^3, \ldots, g^{p-2}$ alle niet-nul residuen afgaat modulo p. Hiermee kan een discrete logaritme worden gedefinieerd.

Definitie 1.5 Stel p priemgetal en g de voortbrenger van de groep $\mathbb{Z}/p\mathbb{Z}$. Het *discrete logaritme* van $x \in \mathbb{Z}/p\mathbb{Z}$ met betrekking tot p en g is het gehele getal r met $0 \leq r < p$ zodanig dat

$$g^r \equiv x \bmod p.$$

Een eventuele afluisteraar Eve krijgt in het sleuteluitwisselingsprotocol de getallen g^a en g^b mee, maar kan hier niet de geheime sleutel $g^{(ab)}$ van maken. Stel dat Eve wel het discrete logaritme, bijvoorbeeld a kan berekenen, dan is het wel mogelijk om de sleutel te bepalen uit g^b en a.

Net als bij RSA hangt dit systeem af van een erg moeilijke berekening. En net als bij het priemfactoriseren is er geen algoritme voor die met voldoende snelheid dit probleem kan oplossen. In hoofdstuk 4 wordt een quantumalgoritme behandeld dat er op een snelle manier voor zorgt dat het discrete algoritme wordt berekend. Als een quantumcomputer zou bestaan, zou het Diffie-Hellman sleuteluitwisselingsprotocol niet meer veilig gebruikt kunnen worden.

2 Quantumberekeningen

De klassieke cryptografie wordt in de praktijk erg vaak gebruikt. Banken en veiligheidssystemen kunnen hierdoor bepaalde garanties geven op de veiligheid van hun systemen. Wetenschappers zijn nu bezig met het ontwerpen van quantumcomputers die op een andere manier algoritmes uitvoeren dan mogelijk is op klassieke computers. Deze berekeningen worden op een heel andere manier tot stand gebracht.

Het priemfactoriseren en het berekenen van een discrete logaritme zijn voorbeelden van berekeningen die door een quantumcomputer misschien wel uitgevoerd kunnen worden. Later zullen er twee algoritmes bekeken worden die laten zien dat dit inderdaad mogelijk wordt gemaakt. Maar eerst is het handig om quantumberekeningen beter te begrijpen. Hiervoor moet eerst nog even benadrukt worden dat quantumcomputers nog niet bestaan in de hoedanigheid dat zulke algoritmes toegepast kunnen worden.

2.1 Qubits

Alle quantumberekeningen worden gedaan met behulp van quantumbits, deze worden qubits genoemd. Deze qubits kunnen het beste vergeleken worden met vectoren. Bekijk de complexe vectorruimte $\mathbb{C}^2$. Deze wordt opgespannen door twee orthonormale vectoren, in dit geval door $|0\rangle$ en $|1\rangle$. Elke vector in deze complexe vectorruimte kan vervolgens geschreven worden als $\alpha|0\rangle + \beta|1\rangle$ waarbij α en β complexe getallen zijn. Een qubit is precies een vector van de vorm $\alpha|0\rangle + \beta|1\rangle$ met α en β complexe getallen, maar met als extra eis dat er moet gelden $|\alpha|^2 + |\beta|^2 = 1$. Dit betekent dat al deze vectoren dus van lengte 1 zijn.

De vectoren $|0\rangle$ en $|1\rangle$ zijn de basistoestanden, een vector van lengte 1 in $\mathbb{C}^2$ heet een superpositie van de basistoestanden en vele qubits samen zijn een register. Als er een register van n qubits is, kan deze in een superpositie verkeren van 2^n basistoestanden. Wat dus eigenlijk een lineaire combinatie is van de 2^n basistoestanden van lengte 1. Deze wordt genoteerd met behulp van alle basistoestanden $|0\rangle$ tot en met $|2^n - 1\rangle$ met een amplitude $\alpha_i \in \mathbb{C}$ van $|i\rangle$ ervoor. Een dergelijk register van n qubits ziet er dan als volgt uit

$$\alpha_0|0\rangle + \alpha_1|1\rangle + \cdots + \alpha_{2^n-1}|2^n - 1\rangle, \quad \text{met} \quad \sum_{j=0}^{2^n-1} |\alpha_j|^2 = 1.$$

Deze $|0\rangle$ tot en met $|2^n - 1\rangle$ vormen dan een orthonormale basis voor een 2^n-dimensionale Hilbertruimte. Waarbij de Hilbertruimte een 2^n-dimensionale vector ruimte is waar het inproduct op geldt. Een *quantumtoestand* is dan een vector van lengte 1 in deze ruimte.

2.2 Metingen en transformaties

De net gedefinieerde qubits kunnen worden gebruikt in twee verschillende operaties: metingen of transformaties. Bij een meting van een qubit van de vorm $\alpha|0\rangle + \beta|1\rangle$ kan er niet die superpositie worden gemeten. De meting kan en zal alleen één van de basistoestanden waarnemen. De basistoestanden kunnen worden waargenomen met een bepaalde kans, waarbij de $|0\rangle$ wordt waargenomen met een kans van $|\alpha|^2$ en $|1\rangle$ wordt waargenomen met een kans van $|\beta|^2$. Zo ook bij de grotere qubits, de kans dat basistoestand $|i\rangle$ wordt waargenomen zal $|\alpha_i|^2$ zijn. Als een quantumtoestand wordt gemeten en basistoestand $|j\rangle$ wordt gemeten, dan verdwijnt de originele superpositie met alle amplitudes.

Daarnaast is het ook mogelijk om maar een deel van een register te meten. Dit zorgt voor het unieke natuurkundige fenomeen uit de quantummechanica: quantumverstrengeling. Hierbij zijn twee objecten, in dit geval qubits zodanig met elkaar verbonden, dat het ene object niet meer volledig beschreven kan worden zonder het andere ook specifiek te noemen. Zoals net beschreven kunnen deze deeltjes zich in meerdere staten bevinden, zolang ze niet gemeten worden. De verstrengeling geeft dan een relatie aan tussen de metingen en de deeltjes.

Verder zijn er meerdere lineaire transformaties mogelijk. Deze transformaties worden beschreven door unitaire matrices. Een matrix U is unitair als de inverse U^{-1} gelijk is aan de geconjugeerde getransponeerde matrix $U^\dagger$. Deze matrices behouden lengtes en aangezien de qubits gezien kunnen worden als vectoren met lengte 1, is dit een nodige eis. Daarnaast zijn unitaire matrices inverteerbaar, waardoor de operaties omkeerbaar zijn. Uiteraard met uitzondering van de metingen, aangezien vanuit een gemeten basistoestand niet de superpositie terug te krijgen is. Hieronder volgen wat voorbeelden van zulke transformaties.

2.2.1 Hadamard transformatie

Een belangrijk en bekend voorbeeld van een transformatie is de *Hadamard transformatie*, bedacht door de Franse wiskundige Jacques Hadamard. Als matrix wordt deze transformatie gegeven als:

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}.$$

Als H wordt toegepast op de basistoestanden $|0\rangle$ en $|1\rangle$, die worden gerepresenteerd door vectoren $\begin{pmatrix} 1 \\ 0 \end{pmatrix}$ resprensievelijk $\begin{pmatrix} 0 \\ 1 \end{pmatrix}$ en vervolgens worden gemeten, is er gelijke kans op beide basistoestanden:

$$H|0\rangle = \frac{1}{\sqrt{2}}|0\rangle + \frac{1}{\sqrt{2}}|1\rangle$$
$$H|1\rangle = \frac{1}{\sqrt{2}}|0\rangle - \frac{1}{\sqrt{2}}|1\rangle.$$

Veralgemeniseerd geeft dit:

$$H^{\otimes n}: \quad \mathbb{C}^N \to \quad \mathbb{C}^N \tag{2.1}$$
$$|i\rangle \quad \mapsto \quad \frac{1}{\sqrt{2^n}} \sum_{j\in\{0,1\}^n} (-1)^{i\odot j}|j\rangle. \tag{2.2}$$

Met $i, j \in \{0,1\}^n$, $i \odot j = \sum_{k=1}^{n} i_k j_k$ het inproduct van een register met n qubits en $N = 2n$.

2.2.2 Query

Een andere vaak gebruikte transformatie in de quantumberekeningen heet een *query*. Neem een functie $f : \{0,1\}^n \to \{0,1\}^m$, dan kan er een transformatie O gegeven worden door:

$$O_f : |i\rangle|0\rangle \mapsto |i\rangle|f(i)\rangle.$$

Daarbij bestaat register $|i\rangle$ uit n qubits en registers $|0\rangle$ en $|f(i)\rangle$ uit m qubits. Vervolgens kan deze afbeelding O_f worden uitgebreid naar een transformatie op $\mathbb{C}^{n+m}$:

$$O_f : |i\rangle|k\rangle \mapsto |i\rangle|f(i) \oplus k\rangle.$$

Hierbij is $|f(i) \oplus k\rangle$ de componentsgewijze optelling modulo 2. Bekijk nu het rijtje $(x_1, \ldots, x_N) \in \{0,1\}^N$, waarbij $N = 2^n$. Dan kan de functie $f(i) = x_i$ bekeken worden die de volgende unitaire transformatie geeft:

$$O : |i, b\rangle \mapsto |i, x_i \oplus b\rangle (= |i, f(i) \oplus b\rangle), \quad \text{met} \quad b \in \{0,1\}.$$

Dan kan een transformatie van de volgende vorm gemaakt worden door te kijken naar het beeld van $|i, \alpha\rangle$ onder O. Hierbij is $\alpha = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle)$.

$$O_{\pm} : |i\rangle \mapsto (-1)^{x_i}|i\rangle.$$

2.3 Quantum-Fouriertransformaties

Bovenstaande transformaties, queries en metingen worden gebruikt in de algoritmes die later worden besproken. Echter is één van de belangrijkste gebruikte tranformaties de quantum-Fouriertransformatie (QFT). Deze zal eerst besproken worden voordat de algoritmes bekeken kunnen worden.

De quantum-Fouriertransformatie gedraagt zich als een discrete Fouriertransformatie. Deze transformatie wordt daarom hieronder eerst uitgelegd.

2.3.1 Discrete Fouriertransformatie

Bekijk eerst de definitie van een discrete Fouriertransformatie:

Definitie 2.1 Zij $f = (f_0, f_1, \ldots, f_{N-1})$ een vector in $\mathbb{C}^N$. Dan is de discrete Fouriertransformatie een functie:

$$\begin{aligned} \mathcal{F} :& \mathbb{C}^N \to \mathbb{C}^N \\ & f \quad \mapsto \tilde{f} \end{aligned}$$

gedefinieerd door:

$$\tilde{f}_j := \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \zeta^{-jk} f_k,$$

met $\zeta = \exp\left(\frac{2\pi i}{N}\right)$ een primitieve N-de eenheidswortel. (Oftewel, een complex getal dat na een n-de machtsverheffing 1 oplevert). De discrete Fouriertransformatie (DFT) is inverteerbaar. De matrix representatie van $\mathcal{F}$ op de standaard basis is een Vandermonde matrix:

$$\mathcal{F} \to \frac{1}{\sqrt{N}} \begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \zeta^{-1} & \zeta^{-2} & \cdots & \zeta^{-(N-1)} \\ 1 & \zeta^{-2} & \zeta^{-4} & \cdots & \zeta^{-2(N-1)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \zeta^{-(N-1)} & \zeta^{-2(N-1)} & \cdots & \zeta^{-(N-1)^2} \end{pmatrix}.$$

2.3.2 QFT

De quantum-Fouriertransformaties zijn lineaire transformaties toegespast op qubits. Het is een quantumparallel van de discrete Fouriertransformatie. Hieronder zal eerst de definitie besproken worden.

Definitie 2.2 Zij $\{|0\rangle, |1\rangle, \ldots, |N-1\rangle\}$ een orthonormale basis voor een quantumsysteem en zij $|\phi\rangle = \sum_{j=0}^{N-1} |j\rangle$ een quantumtoestand. Dan is een quantum-Fouriertransformatie F_N een functie gedefineerd door:

$$|\phi\rangle = \sum_{j=0}^{N-1} |j\rangle \mapsto \sum_{j=0}^{N-1} \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \zeta^{-jk} |k\rangle.$$

Een basistoestand wordt dan getransformeerd als

$$|j\rangle \mapsto \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} \zeta^{-jk} |k\rangle.$$

Waardoor de representatie voor F_N wordt

$$F_N = \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} \zeta^{-jk} |k\rangle\langle j|,$$

met $\langle j|$ als notatie voor een dubbele rij vector. Er geldt: $\bar{\zeta} = \zeta^{-1}$, hierdoor is de geadjugeerde:

$$F_N^\dagger = \frac{1}{\sqrt{N}} \sum_{j,k=0}^{N-1} \zeta^{jk} |k\rangle\langle j|.$$

Deze quantum-Fouriertransformatie is unitair. Bekijk:

$$\begin{aligned}
F_N F_N^\dagger &= \frac{1}{N} \sum_{j,k=0}^{N-1} \zeta^{-jk} |k\rangle\langle j| \sum_{r,s=0}^{N-1} \zeta^{rs} |r\rangle\langle s| \\
&= \frac{1}{N} \sum_{j,k,r,s=0}^{N-1} \zeta^{rs-jk} |k\rangle\langle j|r\rangle\langle s| \\
&= \frac{1}{N} \sum_{j,k,r,s=0} 6N - 1\zeta^{rs-jk} \delta_{jr} |k\rangle\langle s \\
&= \frac{1}{N} \sum_{k,s=0}^{N-1} \left(\sum_{r=0}^{N-1} \zeta^{r(s-k)} \right) |k\rangle\langle s| \\
&= \sum_{k,s=0}^{N-1} \delta_{ks} |k\rangle\langle s| \\
&= \sum_{k=0}^{N-1} |k\rangle\langle k| = I.
\end{aligned}$$

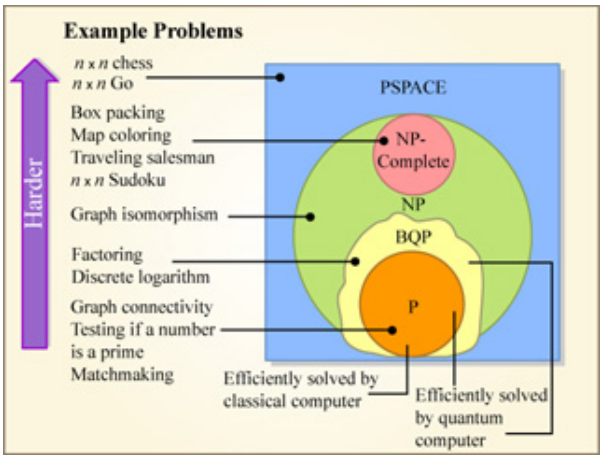
Hierbij is gebruik gemaakt van het feit dat $\sum_{r=0}^{N-1} \exp\left(\frac{2\pi i r(s-k)}{N}\right) = N\delta_{sk}$.

Toepassingen van quantum-Fouriertransformaties zijn te vinden in de algoritmes die later behandeld worden.

2.3.3 Complexiteit

Het toevoegen van de quantumberekeningen binnen de wiskunde, zorgt voor een nieuwe complexiteitsklasse. De klasse **BQP**, in het Engels de *Bounded-error Quantum Polynomial time*. Deze klasse bestaat uit talen die herkend kunnen worden met een kans van minstens 2/3 door het uniforme polynomiale quantumcircuit. Het is hiermee de quantumanaloog voor de complexiteitsklasse **BPP**, de *Bounded-error Probabilistic Polynomial time*.

Aangezien deze klasse net als de quantumberekeningen en algoritmes behoorlijk nieuw is, zijn er nog veel onduidelijkheden over. Het algoritme van Shor in het volgende hoofdstuk, lost met quantumberekeningen een probleem op die voorheen in de **NP**-klasse hoorde. De priemfactorisering is met klassieke methodes en algoritmes niet op te lossen binnen polynomiale tijd. Maar met het quantumalgoritme is dit wel mogelijk en behoort hierdoor in een andere klasse dan waar hij eerst in stond.



Figuur 2.1: Verwachte relatie **BQP** met andere complexiteitsklassen
Bron: Aaronson, Scott. 6.845 Quantum Complexity Theory, Fall 2010.
(MIT OpenCourseWare: Massachusetts Institute of Technology),
http://ocw.mit.edu/courses/electrical-engineering-and-computer-science/6-845-quantum-complexity-theory-fall-2010
(Accessed 13 Aug, 2014). License: Creative Commons BY-NC-SA

Het onopgeloste $\mathbf{P} \neq \mathbf{NP}$ probleem wordt hiermee nog een stukje uitgebreid. Of misschien wel makkelijker gemaakt. Er ontstaat een verschuiving in de verschillende klassen, zodat ze niet netjes op elkaar aansluiten. In figuur 2.1 is een representatie te zien van de verwachte relatie die **BQP** heeft met overige complexiteitsklassen. Om verder in te gaan op de complexiteit is hier niet mogelijk, aangezien dit een volledig onderwerp op zichzelf is.

3 Shor's algoritme

Zoals in de vorige hoofdstukken is uitgelegd, zijn sommige cryptografie systemen te breken met quantumcomputers. Het cryptografie systeem RSA wordt gezien als veilig omdat men ervan uit gaat dat een vermenigvuldiging van grote priemgetallen moeilijk te factoriseren is. Maar stel dat het factoriseren met een quantumcomputer binnen polynomiale tijd te realiseren is, zal het systeem dan gebroken worden?

De Amerikaanse wiskundige Peter Shor heeft in 1994 een quantumalgoritme gemaakt voor het factoriseren van gehele getallen. Het probleem dat het algoritme oplost is als volgt: "Gegeven een geheel getal N, vind zijn priemfactoren".

Het factorisatieprobleem wordt hier op een klassieke manier omgezet naar een ander probleem. Dit probleem is het vinden van de periode van een willekeurig getal, dat relatief priem is ten opzichte van het originele gehele getal. Het omzetten van dit probleem en het oplossen van het periodeprobleem is beide mogelijk binnen polynomiale tijd en is daarom een goede manier om bijvoorbeeld RSA te kraken.

3.1 Periode zoeken

Voordat het eerste deel van het algoritme van Shor wordt bekeken, is het handig om het begrip "orde" duidelijk te hebben.

Definitie 3.1 De *orde* van een willekeurige $0 < x \leq n$ is het kleinste natuurlijke getal $0 < r \leq n$ zodat $x^r \equiv 1 \bmod n$.

In definitie 1.5 van het discrete logaritme wordt een voorbrenger van een multiplicatieve groep uitgelegd. Deze voortbrenger heeft dus de maximale orde van een groep. Voor het algoritme van Shor, zijn er getallen nodig die een even orde r hebben en waarbij geldt dat $x^{r/2} + 1$ en $x^{r/2} - 1$ geen veelvouden van n zijn. Om te laten zien dat een dergelijke orde bij een willekeurige x gevonden kan worden met een relatief grote kans, zijn eerst de volgende lemma en stelling nodig.

Lemma 3.2 *Als n een oneven geheel getal is, dan heeft minstens de helft van de elementen van $\mathbb{Z}/n\mathbb{Z}$ een even orde.*

Bewijs:
Stel een oneven r is de orde van x. Dan geldt:

$$(-x)^r = (-1)^r x^r = (-1)^r \equiv -1 \bmod (n).$$

Dit betekent dat $-x$ orde $2r$ heeft en dus even is. Oftewel, voor elke $x \in \mathbb{Z}/n\mathbb{Z}$ met oneven orde is er een $-x$ met even orde. ■

Stelling 3.3 *Zij n een oneven geheel getal en zij $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ de priemfactorisatie van n. Als $x \in \mathbb{Z}/n\mathbb{Z}$ een willekeurig element is, dan is de kans dat x een even orde r heeft èn $x^{r/2} \not\equiv -1 \bmod n$ minstens $1 - \frac{1}{2}^{k-1}$*

Bewijs:
Het kiezen van een willekeurige $x \in \mathbb{Z}/n\mathbb{Z}$ is volgens de Chinese Reststelling gelijk aan het willekeurig kiezen van $x_i \in \mathbb{Z}/p_i^{e_i}\mathbb{Z}$ voor elke p_i. Zij nu r de orde van x en r_i de orde van x_i. Hierbij zal $x^{r/2}$ nooit 1 modulo n zijn. Het handigste is om te laten zien dat de kans van òf r oneven òf $x^{r/2} \equiv -1 \bmod n$ hoogstens $(1/2)^{k-1}$ is.

Merk ook op dat geldt: $r = \text{kgv}(r_1, r_2, \ldots, r_k)$ met kgv het kleinste gemeenschappelijke veelvoud. Want: als $x^r \equiv 1 \bmod n$, dan geldt ook $x^r \equiv 1 \bmod p_i^{e_i}$.

Stel r is oneven. Dit kan alleen als alle r_i's ook oneven zijn. Volgens lemma 3.2 is r_i oneven met kans minstens $1/2$, dus r is oneven met kans hoogstens $(1/2)^k$.

Stel r is even, dan kan het nog steeds zijn dat $x^{r/2} \equiv \pm 1 \bmod n$. Met de Chinese reststelling geldt dan ook dat $x^{r/2} \equiv \pm 1 \bmod p_i^{e_i}$ voor alle p_i. Deze situatie moet omzeild zien te worden. Want als er geldt dat $x^{r/2} \equiv 1 \bmod n$, dan is r niet de orde. En als $\equiv -1 \mod n$, dan geeft het algoritme geen goede factorisatie.

De kans dat men een dergelijke x kiest, zodat beide gevallen voorkomen is: $2 \cdot 2^{-k} = 2^{-k+1}$. Het combineren van de kansen, zorgt ervoor dat men een goede x kiest met kans minstens

$$(1 - 2^{-k})(1 - 2^{-k+1}) \leq 1 - 3 \cdot 2^{-k} \leq 1 - \frac{1}{2}^{k-1}.$$

■

3.2 Eerste deel Shor's algoritme

Stel geheel getal n moet gefactoriseerd worden. Kies een $x \in 2, \ldots, n-1$ willekeurig zodat x copriem is tot n. Bekijk de volgende rij:

$$1 = x^0 \bmod n, x^1 \bmod n, x^2 \bmod n, \ldots, \text{etc.}$$

Dan bestaat er een $r \in \mathbb{N}$ met $0 < r \leq n$ zodanig dat $x^r \equiv 1 \bmod n$. Deze r is dus de periode van x binnen de ring $\mathbb{Z}/n\mathbb{Z}$. Volgens stelling 3.3 en lemma 3.2

is r met kans $\geq \frac{1}{4}$ even én zijn $x^{r/2}+1$ en $x^{r/2}-1$ geen veelvouden van n. Er geldt als r even is:

$$\begin{aligned} x^r &\equiv 1 \bmod n \Leftrightarrow \\ (x^{r/2})^2 &\equiv 1 \bmod n \Leftrightarrow \\ (x^{r/2}+1)(x^{r/2}-1) &\equiv 0 \bmod n \Leftrightarrow \\ (x^{r/2}+1)(x^{r/2}-1) &= kn \quad \text{voor een zekere} \quad k. \end{aligned}$$

Uit het feit dat $x > 0$, volgt dat $x^{r/2}-1 > 0$ en $x^{r/2}+1 > 0$. Hieruit kan geconcludeerd worden dat $k > 0$. Met het algoritme van Euclides kunnen de $\text{ggd}(x^{r/2}-1, n)$ en $\text{ggd}(x^{r/2}+1, n)$ berekend worden (met ggd de grootste gemene deler). Deze getallen zijn vervolgens twee niet triviale delers van n en daarmee factoren van n. Voorbeeld 3.4 zal laten zien dat dit daadwerkelijk werkt met kleine getallen zonder hulp van een quantumcomputer.

Voorbeeld 3.4 Zij $n = 35$. Kies $x = 13$, aangezien deze relatief priem is met $n = 35$. Bekijk vervolgens het rijtje $1 = x^0 \bmod n, x^1 \bmod n, x^2 \bmod n, \ldots, etc$ tot er een periode r is gevonden.

$$13^1 \equiv 13 \bmod 35, 13^2 \equiv 29 \bmod 35, 13^3 \equiv 27 \bmod 35, 13^4 \equiv 1 \bmod 35.$$

Oftewel de orde van 13 is $r = 4$. Deze voldoet aan de eisen dat r even is en $13^{4/2}+1 = 170$ en $13^{4/2}-1 = 168$ zijn geen veelvouden van $n = 35$. Berekenen van de grootste gemene delers geeft: $\text{ggd}(13^{4/2}+1, 35) = \text{ggd}(170, 35) = 5$ en $\text{ggd}(13^{4/2}-1, 35) = \text{ggd}(168, 35) = 7$. En 5 en 7 zijn inderdaad de priemfactoren van $n = 35$.

Hierdoor is het factorisatieprobleem dus omgezet in een periode-bepaling probleem. Het tweede deel van het algoritme van Shor geeft een oplossing voor dit probleem dat werkt binnen quantumpolynomiale tijd.

3.3 Kettingbreuken

Voor een klein onderdeel van Shor's algoritme zijn kettingbreuken nodig. Om hier bekend mee te raken, volgt eerst de definitie.

Definitie 3.5 Een eindige kettingbreuk, met notatie $[a_0; a_1, a_2, ..., a_n]$, is een breuk die zich blijft herhalen. Deze ziet er als volgt uit:

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{\ddots \cfrac{1}{a_n}}}},$$

waarbij alle $a_i \in \mathbb{Z}$, met $a_i \geq 1$ als $i \geq 1$. Elk rationaal getal heeft nu 2 kettingbreukontwikkelingen; één ontwikkeling die op een 1 eindigt en een ontwikkeling die dat niet doet. Soms eist men uniciteit door $a_n \geq 2$ te verlangen.

Een $z = p/q \in \mathbb{Q}$ kan geschreven worden als een kettingbreuk $[a_0; a_1, a_2, ..., a_n]$. Alle rationale getallen daarvoor; $[a_0], [a_0; a_1], ..., [a_0; a_1, a_2, \ldots, a_n]$ worden de convergenten van p/q genoemd.

Een kettingbreuk is oneindig als het bijbehoorende getal x irrationaal is. Voor de verduidelijking volgt hieronder een voorbeeld.

Voorbeeld 3.6 Stel breuk 804/137 wordt omgezet naar een kettingbreuk. Er wordt gekeken naar de gehele getallen en de resten die de breuk geeft. Er volgt een reeks van vergelijkingen die iets weghebben van het algoritme van Euclides:

$$\begin{aligned}
\frac{804}{137} &= 5 + \frac{119}{137} \\
\frac{137}{119} &= 1 + \frac{18}{119} \\
\frac{119}{8} &= 14 + \frac{7}{8} \\
\frac{8}{7} &= 1 + \frac{1}{7} \\
\frac{7}{1} &= 7 + \frac{0}{1}.
\end{aligned}$$

Gezien de resten steeds een breuk zijn en het omgekeerde weer geschreven kan worden als gehele getallen met rest-breuken, kan deze breuk geschreven worden als de volgende kettingbreuk:

$$\frac{804}{137} = 5 + \cfrac{1}{1 + \cfrac{1}{14 + \cfrac{1}{1 + \frac{1}{7}}}}.$$

Gebruikmakend van de makkelijkere en kortere notatie geeft dit:

$$\frac{804}{137} = [5; 1, 14, 1, 7].$$

Voor het algoritme van Shor zal nodig zijn om breuken binnen een bepaalde afstand van elkaar te bepalen. Dit kan met behulp van het ontwikkelen van een kettingbreuk. Dat er ook daadwerkelijk een breuk gevonden kan worden binnen een bepaalde afstand, zal duidelijk worden aan de hand van stelling 3.7 en lemma 3.8. Later zullen deze gebruikt worden in het algoritme.

Stelling 3.7 *Zij $x \in \mathbb{R}$. Als $p/q \in \mathbb{Q}$ voldoet aan:*

$$|x - \frac{p}{q}| < \frac{1}{2q^2}.$$

Dan is $p/q = p_k/q_k$ voor een convergent p_k/q_k van x.

Bewijs:
Ontwikkel p/q in een eindige kettingbreuk van oneven lengte n. Hierbij geldt: $p/q = p_n/q_n$ en:

$$\frac{p_n}{q_n} - x = \frac{\delta}{q_n^2} \quad \text{met} \quad \delta < \frac{1}{2}.$$

Vervolgens bestaat er een $y > 0$ zodanig dat:

$$x = \frac{yp_n + p_{n-1}}{yq_n + q_{n-1}}.$$

Dan geldt:

$$\frac{\delta}{q_n^2} = \frac{p_n}{q_n} - x = \frac{p_n q_{n-1} - p_{n-1} q_n}{q_n(yq_n + q_{n-1})} = \frac{(-1)^{n+1}}{q_n(yq_n + q_{n-1})}.$$

Dus:

$$\delta = \frac{q_n}{yq_n + qq_{n-1}} \quad \text{en} \quad y = \frac{1}{\delta} - \frac{q_{n-1}}{q_n} > 1.$$

Uit onderstaand lemma 3.8 volgt dat p_{n-1}/q_{n-1} en $p_n/q_n = p/q$ opvolgende convergenten van x zijn. ■

Lemma 3.8 *Als*

$$x = \frac{py + r}{qy + s}$$

met $y \in \mathbb{R}$ en $p, q, r, s \in \mathbb{Z}$ zodat $y > 1$, $q > s > 0$ en $ps - qr = \pm 1$. Dan is er een $n \geq 0$ zodat er geldt:

$$y = x_{n+1} \quad \frac{p}{q} = \frac{p_n}{q_n} \quad \textit{en} \quad \frac{r}{s} = \frac{p_{n-1}}{q_{n-1}},$$

met $x = [a_0; a_1, \dots]$, $x_i = [a_i; a_{i+1}, \dots, a_n]$ en $p_i/q_i = [a_0; a_1, \dots, a_i]$ voor $i \geq 0$.

Bewijs:
Ontwikkel p/q in een kettingbreuk: $p/q = [A_0; A_1, \dots, A_n] = v_n/w_n$ en laat $v_{n-1}/w_{n-1} = [A_0; A_1, \dots, A_{n-1}]$. Deze kettingbreuk is zo aangepast dat voor de lengte n geldt:

$$(-1)^{n+1} = v_n w_{n-1} - v_{n-1} w_n = ps - qr = \pm 1.$$

Dan is: $v_n w_{n-1} - v_{n-1} w_n = v_n s - w_n r$ waaruit volgt: $v_n(w_{n-1} - s) = w_n(v_{n-1} - r)$ Omdat v_n, w_n onderling ondeelbaar zijn en $v_n > v_{n-1}$, geldt dat $s = w_{n-1}$ en $r = v_{n-1}$ Maar de kettingbreukontwikkeling van

$$\frac{v_n y + v_{n-1}}{w_n y + w_{n-1}} = [A_0; A_1, ..., A_n, y].$$

Dus is $[A_0; A_1, \dots, A_n]$ het beginstuk van de kettingbreuk voor x en is y de staart, waardoor geldt:

$$[A_0; A_1, \dots, A_n] = [a_0; a_1, \dots, a_n] \quad \text{en} \quad y = [A_{n+1}; A_{n+2}, \dots] = [a_{n+1}; a_{n+2}, \dots].$$

■

3.4 Tweede deel Shor's algoritme

Definieer de volgende functie: $f : \mathbb{N} \to [n]$, met $[n]$ een ring met n elementen, met de eigenschap: er bestaat een $r \in [n]$ met:

$$f(a) = f(b) \Leftrightarrow a \equiv b \bmod r.$$

Hiermee geldt $f(a) \equiv x^a \bmod n$. Waarbij het doel is om r te vinden.

Definieer ook de volgende quantum-Fouriertransformatie:
Voor een willekeurige $0 < q < n$, zij $\mathbb{Z}_q := \{0, \ldots, q-1\}$. Voor alle $a \in \mathbb{Z}_q$, definieer functie:

$$\begin{aligned} \chi_a : \quad & \mathbb{Z}_q \to \mathbb{C} \\ & b \mapsto e^{2\pi i \frac{ab}{q}}, \end{aligned}$$

die de standaardbasis $\{|a\rangle | a \in \mathbb{Z}_q\}$ omzet in de Fourierbasis $\{|\chi_a\rangle | a \in \mathbb{Z}_q\}$ met

$$|\chi_a\rangle = \frac{1}{\sqrt{q}} \sum_{b \in \mathbb{Z}_q} \chi_a(b)|b\rangle.$$

De quantum-Fouriertransformatie wordt dan:

$$QFT : |a\rangle \to |\chi_a\rangle.$$

Het uitvoeren van de QFT op een m-bit basistoestand $|a\rangle = |a_1 \ldots a_m\rangle$ geeft een "unentangled"toestand:

$$|\chi_z\rangle = \frac{1}{\sqrt{2^m}}(|0\rangle + 2^{2\pi i(0.a_m)}|1\rangle)(|0\rangle + e^{2\pi i(0.a_{m-1}a_m)}|1\rangle) \cdots (|0\rangle + e^{2\pi i(0.a_1 \ldots a_m)}|1\rangle).$$

Deze quantum-Fouriertransformatie zal gebruikt worden in het vervolg van het algoritme. Dit vervolg wordt vervolgens opgedeeld in twee gevallen.

3.4.1 Periode vinden als r een deler is van q

Begin met twee registers met lengtes $[\log q]$ en $[\log n]$: $|0\rangle|0\rangle$. Gebruik de quantum-Fouriertransformatie op het eerste register. Als resultaat geeft dit een uniforme superpositie, waarbij het tweede register nog steeds alleen uit nullen bestaat:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|0\rangle.$$

Pas vervolgens de functie $f(a)$ toe op de tweede register, nadat de quantum-Fouriertransformatie is uitgevoerd:

$$\frac{1}{\sqrt{q}} \sum_{a=0}^{q-1} |a\rangle|f(a)\rangle.$$

Een observatie op het tweede register geeft een waarde $f(s)$, met $s < r$ per definitie. Want $f(a) = f(s) \Leftrightarrow a \equiv s \bmod r$ met r een deler van q. Deze a zal

van de vorm zijn: $a = jr + s$ met $0 \leq j < q/r$, want dan is $f(a) = f(s)$. Het eerste register zal vervallen naar de superpositie met waarden $|s\rangle, |r+s\rangle, |2r+s\rangle, \ldots, |q-r+s\rangle$. Het tweede register zal vervallen naar de klassieke toestand $|f(s)\rangle$, waarna deze genegeerd wordt. Er ontstaat de superpositie:

$$\sqrt{\frac{r}{q}} \sum_{f=0}^{q-1} |jr+s\rangle.$$

Opnieuw wordt de quantum-Fouriertransformatie toegepast, waarbij de volgende superpositie ontstaat:

$$\sqrt{\frac{r}{q}} \sum_{j=0}^{q/r-1} \sum_{b=0}^{q-1} e^{2\pi i \frac{(jr+s)b}{q}} |b\rangle = \sqrt{\frac{r}{q}} \sum_{b=0}^{q-1} x^{2\pi i \frac{sb}{q}} \Big(\sum_{j=0}^{q/r-1} x^{2\pi i \frac{jrb}{q}} \Big) |b\rangle.$$

Herschrijven met behulp van $\sum_{j=0}^{n-1} a^j = (1-a^n)/(1-a)$ voor $a \neq 1$ geeft:

$$\sum_{j=0}^{q/r-1} e^{2\pi i \frac{jrb}{q}} = \sum_{j=0}^{q/r-1} (e^{2\pi i \frac{r}{q}})^j = \begin{cases} q/r & \text{als} \quad e^{2\pi i \frac{rb}{q}} = 1 \\ \frac{1-e^{2\pi i b}}{1-e^{2\pi i \frac{rb}{q}}} & \text{als} \quad e^{2\pi i \frac{rb}{q}} \neq 1 \end{cases}$$

Er geldt: $e^{2\pi i \frac{rb}{q}} = 1$, dan en slechts dan als rb/q is een rationaal getal, dan en slechts dan als b is een veelvoud van q/r en dan geldt:

$$\frac{b}{q} = \frac{c}{r},$$

met b en q bekend, c en r onbekend met $0 \leq c < r$. De kans dat c copriem is met r, is relatief groot. Bekijk hiervoor lemma 3.9. Vervolgens is r te vinden door de breuk b/q zo klein mogelijk te schrijven, mits c inderdaad copriem is met r.

Lemma 3.9 *De kans dat twee willekeurige getallen relatief priem zijn, is ongeveer* $6/\pi^2$.

Beredeneren van de kans geeft de volgende benadering. Gegeven twee willekeurig gekozen getallen A en B. Er geldt dat A en B relatief priem zijn dan en slechts dan als geen enkel priemgetal deelt A of deelt B en de twee getallen zijn onderling ondeelbaar.

De kans dat getal A deelbaar is door priemgetal p is intuïtief $1/p$. Stel dat beide getallen, A en B, deelbaar zijn door p, dan is deze kans gelijk aan $1/p^2$. Oftewel, de kans dat minstens één van de getallen ondeelbaar is door p, is $1 - 1/p^2$. De kans dat de twee getallen relatief priem zijn wordt dan benaderd door het product over alle priemgetallen:

$$\prod_{p \text{ priem}} (1 - \frac{1}{p^2}) = \left(\prod_{p \text{ priem}} \frac{1}{1-p^{-2}} \right)^{-1} = \frac{1}{\zeta(2)} = \frac{6}{\pi^2},$$

waarbij $\zeta(s)$ de Riemann-zeta functie aanduidt. Leonhard Euler heeft daarbij bewezen dat geldt:

$$\prod_{p \text{ priem}} \frac{1}{1-p^{-s}} = \sum_{n \geq 1} \frac{1}{n^s} = \zeta(s).$$

Algemener geldt dat k willekeurig gekozen, gehele getallen, relatief priem zijn met kans $1/\zeta(k)$. ■

3.4.2 Periode vinden als r geen deler is van q

Met grote kans, zal bovenstaand algoritme een b geven zodat:

$$\left|\frac{b}{q} - \frac{c}{r}\right| \leq \frac{1}{2q}.$$

Met wederom b, q bekend en c, r onbekend. De twee breuken hebben beide een noemer $\leq n$ en volgens lemma 3.10 zullen dus maximaal $1/n^2 > 1/q$ van elkaar verwijderd liggen.

Lemma 3.10 *Twee breuken x/y en x'/y', beide met de noemer kleiner dan N, zullen maximaal $1/N^2$ van elkaar vandaan liggen.*

Bewijs:
Zij $z = x/y$ en $z' = x'/y'$, met $y, y' \leq N$ en $z \neq z'$. Dan is $|xy' - x'y| \geq 1$. Waaruit volgt dat $|z - z'| = |(xy' - x'y)/yy'| \geq 1/N^2$. ■

Hierdoor zal c/r de enige breuk zijn met noemer $\leq n$ op afstand $\leq 1/2q$ van b/q. Dankzij stelling 3.7 is bekend dat een breuk op deze afstand in de kettingbreuk ontwikkeling van b/q moet liggen. Deze kettingbreuk ontwikkeling zal de breuk c/r geven. Aangezien de kans weer $6/\pi^2$ is dat c copriem is tot r volgens lemma 3.9, geeft het kleiner schrijven van de breuk de gezochte r.

3.5 Periode vinden al genoeg om berichten in RSA te lezen

Om één instantie van RSA te kraken moet het bericht m mod N gevonden worden. Hierbij zijn m^e, e en N verzonden over het publieke netwerk en dus al bekend. Oftewel, het is nodig om een methode te vinden waarbij het mogelijk is om de e-de machtswortel te bepalen in de multiplicatieve groep van de gehele getallen modulo N.

De oplossing die het algoritme van Shor gebruikt, zoekt juist de factorisatie van N en probeert zo algemeen het cryptografische systeem te breken. Stel dat $N = pq$ dan kan $\phi(N) = (p-1)(q-1)$ gemakkelijk berekend worden. Waarbij het mogelijk is om een d te vinden zodat $ed \equiv 1 \bmod \phi(N)$ zodat men het originele bericht $m^{ed} \equiv m \bmod N$ kan vinden.

Maar voor één bericht in RSA is het al voldoende om een d te vinden die voldoet aan $ed \equiv 1 \bmod r$ waarbij r staat voor de orde van m modulo N. Zodat dus $ed = rk + 1$ voor een zekere k. Dan geldt er:

$$(m^e)^d \equiv m^{(ed)} \equiv m^{rk+1} \equiv m \bmod N.$$

Aangezien e relatief priem is ten opzichte van $\phi(N)$, is de orde van m gelijk aan de orde van m^e. Als m^e bekend is, kan de orde r berekend worden met behulp

van het algoritme van Shor. Vervolgens kan d berekend worden, waarbij deze voldoet aan $ed \equiv 1 \bmod r$ door gebruik te maken van het uitgebreide algoritme van Euclides.

Hierdoor is het niet nodig om het algoritme van Shor vaker te laten lopen over willekeurig gekozen a's om zo de priemfactoren te achterhalen. Het algoritme kan gebruikt worden om de orde van m^e te vinden en deze op te lossen voor m, zonder dat de priemfactoren van N gevonden worden. Dit geldt alleen voor specifieke berichten m en dus niet voor het hele systeem.

4 Discrete logaritme probleem

Voor alle priemgetallen p is de multiplicatieve groep van $\mathbb{Z}/p\mathbb{Z}$ een cyclische groep. Dat wil zeggen $\exists g \in \mathbb{Z}/p\mathbb{Z}$ zodat het rijtje $1, g, g^2, g^3, \ldots, g^{p-2}$ alle niet-nul residuen afgaat modulo p. De orde van g is dus $p-1$. De definitie van een discrete logaritme, zoals eerder gegeven, wordt dan:

Definitie 4.1 Stel p priemgetal en g de voortbrenger van de groep $\mathbb{Z}/p\mathbb{Z}$. Het discrete logaritme van $x \in \mathbb{Z}/p\mathbb{Z}$ met betrekking tot p en g is het gehele getal r met $0 \leq r < p$ zodanig dat

$$g^r \equiv x \bmod p.$$

In 1993 heeft de Amerikaanse wiskundige Daniel M. Gordon hier een klassiek algoritme[1] voor geschreven, echter is deze te langzaam voor gebruik. Een jaar later, in 1994, is er door Shor een quantumalgoritme geschreven dat veel sneller is dan het klassieke algoritme. Binnen polynomiale tijd kan een discrete logaritme worden berekend door gebruik te maken van een quantumcomputer. Dit algoritme maakt gebruik van twee modulaire exponenties en twee quantum-Fouriertransformaties. De beschrijving van het algoritme hieronder is slechts een schets. Voor het uitgebreide algoritme zal de publicatie van Shor[2] het beste zijn.

4.1 Eerste deel algoritme

Begin met drie registers en vindt een q die een 2-macht is en dicht bij p ligt, oftewel $p < q < 2p$. De eerste twee registers worden in de quantumcomputer gestopt en veranderen in een uniforme superpositie tussen de basisposities $|a\rangle$ en $|b\rangle$ $(\mathrm{mod}(p-1))$. In het derde register wordt de berekening $g^a x^{-b} \bmod p$ uitgevoerd. Dit geeft de volgende toestand:

$$\frac{1}{p-1} \sum_{a=0}^{p-2} \sum_{b=0}^{p-2} |a, b, g^a x^{-b} (\mathrm{mod} p)\rangle.$$

Vervolgens wordt er gebruik gemaakt van de quantum-Fouriertransformatie A_q. Deze transformatie neemt een $0 \leq a < q$ en maakt van de basispositie $|a\rangle$, de

[1] *Discrete logarithms in gf(p) using the number field sieve*, Daniel M. Gordon, 1993

[2] *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, Peter W. Shor, 1994

toestand

$$\frac{1}{q^{1/2}} \sum_{c=0}^{q-1} |c\rangle \exp(2\pi iac/q). \tag{4.1}$$

In deze transformatie wordt gebruik gemaakt van een unitaire matrix waarvan de invoer van (a, c) gelijk is aan $\frac{1}{\sqrt{q}} \exp(2\pi iac/q)$.

In dit geval zorgt deze quantum-Fouriertransformatie ervoor dat $|a\rangle \rightarrow |c\rangle$ en $|b\rangle \rightarrow |d\rangle$ met amplitude

$$\frac{1}{q} \exp(2\pi i(ac + bd)/q).$$

De toestand $|a, b\rangle$ wordt onder deze quantum-Fouriertransformatie:

$$\frac{1}{q} \sum_{a,b=0}^{p-2} \sum_{c,d=0}^{q-1} \exp\left(\frac{2\pi i}{q}(ac + bd)\right) |c, d\rangle.$$

Het toepassen van de quantum-Fouriertransformatie op de registers waar mee begonnen was, geeft toestand:

$$\frac{1}{(p-1)q} \sum_{a,b=0}^{p-2} \sum_{c,d=0}^{q-1} \exp\left(\frac{2\pi i}{q}(ac + bd)\right) |c, d, g^a x^{-b} (\mathrm{mod} p)\rangle.$$

Deze toestand kan een toestand zijn van de vorm $|c, d, y\rangle$ waarbij $y \equiv g^k \bmod p$. De kans dat deze waarneming daadwerkelijk plaatsvindt is:

$$\left| \frac{1}{(p-1)q} \sum_{\substack{a,b \\ a-rb \equiv k}} \exp\left(\frac{2\pi i}{q}(ac + bd)\right) \right|^2, \tag{4.2}$$

waarbij de som genomen wordt over a, b zodanig dat $a - rb \equiv k \bmod p - 1$. Hierbij wordt er nu met twee moduli gerekend, namelijk met $p - 1$ en q. Dit wordt opletten, maar zal geen serieus probleem vormen. Er geldt vervolgens:

$$a = br + k - (p-1)\lfloor \frac{br + k}{p-1} \rfloor,$$

zodat a gesubstitueerd kan worden in vergelijking 4.2, waarmee de kans oftewel de amplitude van $|c, d, g^k (\mathrm{mod} p)\rangle$ wordt:

$$\frac{1}{(p-1)q} \sum_{b=0}^{p-2} \exp\left(\frac{2\pi i}{q}\left(brc + kc + bd - c(p-1)\left\lfloor \frac{br + k}{p-1} \right\rfloor\right)\right). \tag{4.3}$$

4.2 Kans

De absolute waarde gekwadrateerd van bovenstaande vergelijking is de kans op het observeren van de toestand $|c, d, g^k (\bmod p)\rangle$. Nu is het dus belangrijk om te weten hoe vaak deze gezochte quantumtoestand voorkomt. Het analyseren van

deze kans 4.3 begint bij het weghalen van de factor $\exp \frac{2\pi ikc}{q}$. Deze factor kan genegeerd worden aangezien dit de kans niet veranderd. De andere e-machten worden gescheiden in twee delen en b wordt hieruit gefactoriseerd. Hieruit volgt:

$$\frac{1}{(p-1)q} \sum_{b=0}^{p-2} \exp\left(\frac{2\pi i}{q} bT\right) \exp\left(\frac{2\pi i}{q} V\right). \tag{4.4}$$

Met

$$T = rc + d - \frac{r}{p-1}\{c(p-1)\}_q \tag{4.5}$$

en

$$V = \left(\frac{br}{p-1} - \left\lfloor \frac{br+k}{p-1} \right\rfloor\right) \{c(p-1)\}_q. \tag{4.6}$$

Hierbij staat $\{z\}_q$ voor het residu, dat congruent is aan z mod q en in het bereik ligt van $-q/2 < \{z\}_q \leq q/2$.

Alle mogelijke uitkomsten van het algoritme gaan gelabeld worden als "goed" of "slecht". Als het algoritme voldoende goede uitkomsten geeft, zal er een grote kans zijn om de gezochte r af te leiden. Daarom zullen er eerst voorwaarden worden gesteld om te definieren wat een goede uitkomst is.

4.3 Voorwaarden

Als er geldt:

$$|\{T\}_q| = |rc + d - \frac{r}{p-1}\{c(p-1)\}_q - jq| \leq \frac{1}{2}, \tag{4.7}$$

met j het gehele getal dat het dichtste bij T/q ligt, dan varieert de b in de eerste exponentiële term in som 4.4 tussen 0 en $p-2$. Deze term zal dan variëren over hoogstens de helft van de eenheidscirkel.

Als er geldt:

$$|\{c(p-1)\}_q| \leq \frac{q}{12}, \tag{4.8}$$

dan is $|V|$ altijd hoogstens $q/12$. Hierdoor zal de tweede exponentiële term in som 4.4 nooit verder dan $\exp(\pi i/6)$ afwijken van 1.

Als voorwaarden 4.7 en 4.8 gelden, dan wordt een output bestempeld als "goed". Daarbij geven ze allebei een bijdrage aan corresponderende termen die samen de amplitude voor de toestand vormen. Dit geeft voor voorwaarde 4.7 genoeg outputs voor c's zodat het algoritme een geschikte r kan vinden.

Er zullen meer voorwaardes gesteld worden voor een goede output. Hiervoor zal er een ondergrens bepaald worden voor de kans op een goede output. Oftewel een output waarvoor voorwaarden 4.7 en 4.8 gelden. Hiervoor zal b variëren

tussen 0 en $p-2$, waardoor de eerste term $\exp(\frac{2\pi ibT}{q})$ een bereik heeft van 0 tot $2\pi iW$, waarbij

$$W = \frac{p-2}{q}\left(rc + d - \frac{r}{p-1}\{c(p-1)\}_q - jq\right). \tag{4.9}$$

Met j hetzelfde als in voorwaarde 4.7 (het gehele getal dat het dichtste bij T/q ligt). Hierdoor zal de eerste exponentiële term van som 4.4 in de richting $\exp(\pi iW)$ minstens gelijk zijn aan

$$\cos(2\pi|\frac{W}{2} - \frac{Wb}{(p-2)}|). \tag{4.10}$$

Bij voorwaarde 4.8 varieert de fase maximaal $\pi i/6$ door de tweede exponentiele term uit som 4.4: $\exp(\frac{2\pi iV}{q})$. In de richting $\exp(\pi iW)$, zal deze term geminimaliseerd worden zodat deze minstens gelijk is aan

$$\cos(2\pi|\frac{W}{2} - \frac{Wb}{(p-2)}| + \frac{\pi}{6}). \tag{4.11}$$

Dit zorgt ervoor dat de absolute waarde van de amplitude 4.4 wordt:

$$\frac{1}{(p-1)q}\sum_{b=0}^{p-2}\cos\left(2\pi|\frac{W}{2} - \frac{Wb}{(p-2)}| + \frac{\pi}{6}\right). \tag{4.12}$$

Deze som kan vervangen worden door een integraal. Dit geeft de volgende amplitude:

$$\frac{2}{q}\int_0^{1/2}\cos\Big(\frac{\pi}{6} + 2\pi|W|u\Big)du + O\left(\frac{W}{pq}\right). \tag{4.13}$$

Hierbij is $O(\frac{W}{pq})$ een fout-term die ontstaat uit voorwaarde 4.7 en als $|W| \leq 1/2$. Term W varieert tussen $-1/2$ en $1/2$, waarbij het integraal 4.13 minimaal is als $|W| = 1/2$. Dus de kans op het eindigen met een toestand $|c, d, y\rangle$ die voldoet aan de voorwaarden 4.7 en 4.8 is minstens

$$\left(\frac{1}{q}\frac{2}{\pi}\int_{\pi/6}^{2\pi/3}\cos u du\right)^2 = \frac{(\sqrt{3}-1)^2}{\pi^2 q^2} \approx \frac{0,0543\ldots}{q^2} \approx \frac{1}{20q^2}. \tag{4.14}$$

4.4 Tellen van paren

Het is mogelijk om het aantal paren goede output te tellen, oftewel de paren (c, d) die voldoen aan de voorwaarden 4.7 en 4.8. Het aantal paren (c, d) dat voldoet aan voorwaarde 4.7 zijn precies het aantal mogelijke c's. Want voor elke c is er maar één mogelijke d zodat aan voorwaarde 4.7 wordt voldaan.

Als de ggd$(p-1, q)$ niet groot is, zal het aantal c's die aan voorwaarde 4.8 voldoen ongeveer gelijk zijn aan $q/6$, met ggd de grootste gemeenschappelijke deler. Als de ggd$(p-1, q)$ wel groot is, zal dit aantal minstens $q/12$ zijn. Gecombineerd zijn er minstens $q/12$ mogelijke paren die voldoen aan beide voorwaarden.

Voor y zijn er precies $p-1$ mogelijkheden, namelijk alle elementen in $\mathbb{Z}/p\mathbb{Z}$. Dit geeft ongeveer $pq/12$ mogelijke staten voor $|c, d, y\rangle$. Gecombineerd met de beredeneerde ondergrens $1/20q^2$ voor het observeren van de goede toestand, zorgt voor een kans van minstens $(pq/12)(1/20q^2) = p/240q$ op het vinden van een goed paar. Daarnaast heeft iedere goede c een kans van minstens $(p-1)/20q^2 \geq 1/40q$ om geobserveerd te worden.Het feit dat q zo gekozen is dat $q < 2p$, kan gesubstitueerd worden in deze kans. Dit zorgt voor een kans van $1/480$ op het vinden van een goede uitkomst.

De gezochte r kan uit de paren (c, d) worden gehaald waarvoor geldt:

$$-\frac{1}{2q} \leq \frac{d}{q} + r\left(\frac{c(p-1) - \{c(p-1)\}_q}{(p-1)q}\right) \leq \frac{1}{2q} \bmod 1. \tag{4.15}$$

Deze vergelijking ontstaat door vergelijking 4.7 te delen door q. Aangezien q een deler is van $c(p-1) - \{c(p-1)\}_q$ zal r een breuk moeten zijn met noemer $(p-1)$. Dus als d/q wordt afgerond tot het dichtstbijzijnde veelvoud van $1/(q-1)$ en vervolgens wordt gedeeld door een geheel getal $\bmod (p-1)$ om een r te vinden, wordt een goede afschatting van c de volgende:

$$c' = \frac{c(p-1) - \{c(p-1)\}_q}{q}. \tag{4.16}$$

Om te laten zien dat deze quatum berekening alleen een polynomiaal aantal keren herhaald hoeft te worden, zal de effectiviteit nog besproken moeten worden. Hierbij is het grootste probleem dat c''s die niet copriem zijn ten opzichte van $p-1$, niet gebruikt kunnen worden om te delen.

4.5 Effectiviteit

Bij dit algoritme is het onbekend of alle mogelijke waarden van c' als output worden gegeven met gelijke waarschijnlijkheid. Dit is alleen bekend bij 1/12e deel. Dit maakt het volgende deel lastiger dan het vergelijkbare deel in het priemfactorisatie algoritme.

Als van r alle resten modulo alle priemfactoren van $p-1$ bekend waren, kan de Chinese reststelling toegepast worden om r te vinden. Er kan alleen bewezen worden dat deze resten gevonden kunnen worden voor priemgetallen groter dan 18, maar met extra werk is het toch nog mogelijk om r tevoorschijn te halen.

Elke goede paar (c, d) ontstaat dus met een kans van minstens $1/(20q^2)$ en dat minstens een twaalfde van de mogelijke c's in een goed paar zitten. Uit vergelijking 4.16 volgt dat deze c's liggen van c/q tot $c'/(p-1)$ door af te ronden naar de dichtstbijzijnde gehele veelvoud van $1/(p-1)$. Verder zijn de goede c's precies degene zodat c/q dichtbij $c'/(p-1)$ ligt. Oftwel, elke goede c correspondeert met precies één c'.

Voor elke priemmacht $p_i^{\alpha_i}$ die een deler is van $p-1$, zal met grote waarschijnlijkheid een willekeurige c' geen p_i bevatten. Dit zal later worden aangetoond. Daarnaast worden grote constanten geaccepteerd, want dan negeert het algoritme alle priemmachten onder de 18 . Als r modulo alle priemmachten bekend

is boven de 18, dan kunnen alle mogelijke resten voor priemgetallen onder de 18 geprobeerd worden. Dit zorgt voor een verhoging van de looptijd van het algoritme met een constante factor.

Omdat minstens 1/12e van de c's in een goed paar (c, d) zit, is minstens een twaalfde van de c''s goed. Dus voor een priemmacht $p_i^{\alpha_i}$, zal een willekeurige c' deelbaar zijn door $p_i^{\alpha_i}$ met kans hoogstens $12/p_i^{\alpha_i}$. Als er t goede c''s zijn, is de kans op een priemmacht boven de 18 die alle c''s deelt, hoogstens:

$$\sum_{18<p_i^{\alpha_i}|(p-1)} \left(\frac{12}{p_i^{\alpha_i}}\right)^t.$$

Deze som covergeert voor $t = 2$ en wordt bij een verhoging van t met 1, verlaagd met een factor 2/3. Dus voor een constante grotere t is deze lager dan 1/2.

Elke goede c' is verkregen met een kans kleiner dan $1/(40q)$ voor elk experiment. Er zijn $q/12$ goede c''s, dus na $480t$ experimenten, zijn er naar waarschijnlijkheid t goede c''s gekozen met gelijke kans uit alle goede c''s. Dus er zal een verzameling van c''s gevonden worden zodat alle priemmachten $p_i^{\alpha_i} > 20$ die $p - 1$ delen, relatief priem zijn tot minstens één van de c''s.

Om een algoritme te verkrijgen dat werkt binnen polynomiale tijd, zal het nodig zijn om alle mogelijke verzamelingen met t aantal c''s te proberen. Al is het hier handiger om een algoritme te hebben die de verzameling met c''s zoekt met de meeste gemene factoren. Deze verzameling zal de resten van r geven voor alle priemgetallen groter dan 18.

Voor elk priemgetal p_i kleiner dan 18, zijn er maximaal 18 mogelijkheden voor de rest modulo $p_i^{\alpha_i}$. Er kunnen dus alle mogelijke resten geprobeerd worden modulo priemmachten lager dan 18. Voor elke mogelijkheid kan er de bijbehorende r berekend worden met de Chinese reststelling, waarna gekeken kan worden of het de gezochte discrete logaritme is.

4.6 Efficiëntie

Als dit algoritme daadwerkelijk gebruikt gaat worden, zijn er nog genoeg manieren om het veel efficiënter te maken dan hier beschreven staat. De schatting voor het aantal goede c''s is waarschijnlijk veel te laag, zeker aangezien zwakkere voorwaarde dan 4.7 en 4.8 zullen voldoen. Hierdoor zal het aantal keren draaien van het experiment verminderd kunnen worden. Ook is het onwaarschijnlijk dat de verdeling van slechte waardes van c' een relatie hebben met priemgetallen onder de 18. Als het niet zo is, is het onnodig om kleine priemgetallen apart te behandelen.

Daarnaast gebruikt dit algoritme geen eigenschappen van $\mathbb{Z}_p$, dus kan hetzelfde algoritme gebruikt worden voor het vinden van discrete logaritmes over lichamen als $\mathbb{Z}_{p^\alpha}$, zolang deze ook een cyclische multiplicatieve groep bevat. Hiervoor zijn de orde van de generator nodig, vermenigvuldiging en het nemen van inverses moet mogelijk zijn binnen polynomiale tijd in deze lichamen. Daarbij is de orde

van de generator te vinden met behulp van Shor's algoritme zoals te lezen is in hoofdstuk 3. Het discrete logaritme algoritme is door Boneh en Lipton in 1995 gegeneraliseerd zodat het algoritme gebruikt kan worden om het discrete logaritme te bepalen als de groep abels is, maar niet cyclisch[3].

[3] *Quantum cryptoanalysis of hidden linear functions*, D. Boneh, R.J. Lipton, 1995.

Slot

De behandelde algoritmes maken het dus mogelijk om van grote gehele getallen de priemfactoren te vinden en om discrete logaritmes te achterhalen. Dit alles binnen polynomiale tijd waardoor ze bruikbaar kunnen zijn in de praktijk. Echter zal het dan eerst nodig zijn om de quantumcomputer beter te ontwikkelen dat deze de berekeningen ook daadwerkelijk aankan. Voor geïnteresseerden is het aan te raden zich verder te verdiepen in de complexiteit van het quantumgedeelte van deze scriptie.

De quantumalgoritmes zijn tot dan toe nog steeds interessant om te bekijken en te zien waarom het hierin wel lukt maar niet in de klassieke algoritmes op klassieke computers. Het verschil werd mij ooit uitgelegd met een bijzonder voorbeeld:

Stel een kamer voor met tien deuren. Er wordt verteld dat er achter één van deze deuren een geit staat. Deze geit wil je graag hebben. Een klassieke computer zal bij deur 1 beginnen, deze open maken en vervolgens een conclusie trekken. Als achter de deur geen geit staat, gaat deze computer verder naar deur 2. Dit gaat zo door totdat de geit uiteindelijk gevonden is. Stel dat het achter deur 8 staat, zal deze computer 8 keer een deur moeten openen en een conclusie trekken voordat hij weet waar de geit staat.

Een quantumcomputer kan alle deuren tegelijk openen, door middel van de superpositie die de computer kan aannemen. Hierbij kan hij gelijk concluderen dat de geit achter deur 8 zit en bij een meting dat als output leveren. Deze computer zal binnen 1 stap kunnen vertellen waar de geit staat.

De snelheid en effectiviteit van een dergelijke computer wordt hiermee duidelijk. Ze zijn direct en doelmatig en blijven daarom fascinerend. De mogelijkheden die ze eventueel in de toekomst kunnen hebben zou de cryptografie veranderen.

Bibliografie

[1] P. Shore *Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer*, 1996

[2] R. de Wolf, *Quantum Computing and Communication Complexity*, Universiteit van Amsterdam 2010, pp.

[3] F. Xi Lin, *Shor's Algorithm and the Quantum Fourier Transform*, McGill University

[4] R. Cleve, A. Ekert, C. Macchiavello and M. Mosca, *Quantum algorithms revisited*, The Royal Society, 1998

[5] D. Bacon, *Quantum Computing Shor's Algorithm*, University of Washington