Jacob Kjærsgaard Hansen
12.12.2008

Abstract

This text will at first introduce the reader to the modular arithmetic. Subsequently, the reader will be led through prime number tests. By using modular arithmetic these can relatively fast conclude whether a number is a prime number or not. This is useful in the RSA-encryption system, which the reader afterwards will be guided through mixed with the historic background. Finally, the text ends up with a view of the quantum theory. This theory is the base of a treat against the RSA-system, but also it is the theory that can ensure the codes in the future. Examples and analogies are ought to make the reader familiar with the subjects that sometimes seem a little irrational. In the end, the RSA-system is compared with its supposed follower, the quantum cryptography. This comparison concludes that the RSA-system is unbreakable in practice nowadays, but a system based on the quantum cryptography is absolutely unbreakable. Otherwise, it will mean that the quantum theory is wrong. In the end, it is concluded too that the quantum cryptography probably will be useful in general practice before the treat against the RSA-system really will matter.

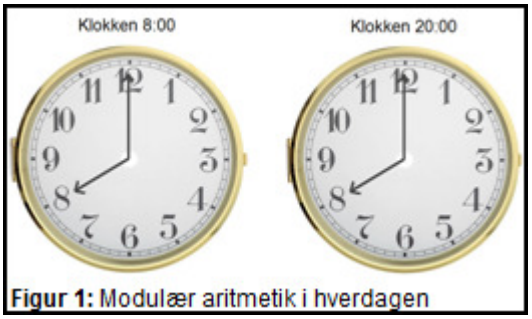
Indholdsfortegnelse

1. Indledning

Behovet for hemmeligholdelse og autentifikation stiger i takt med informationsteknologiens udbredelse. Mange finansielle transaktioner sikres i dag ved hjælp af et krypteringssystem kaldet RSA. Dette system kan også bruges som en slags digital underskrift, hvilket er med til at gøre systemet endnu mere interessant for nutidens samfund, hvor parterne ofte kun kommunikerer digitalt. Systemet er tungt baseret på et matematisk område kaldet modulær aritmetik, hvorfor starten af teksten vil føre læseren igennem dette område og gøre denne fortrolig hermed. Dette vil blive brugt som springbræt videre til blandt andet primtalstest, som ligeledes er en væsentlig forudsætning for at kunne bruge RSA-systemet i praksis. Herefter gennemgås de væsentligste krypteringssystemer, der er blevet brugt gennem tiden, og hvilke styrker og svagheder, disse har. Dette for at synliggøre, hvorfor netop RSA-systemet er så markant anderledes end tidligere krypteringssystemer. Alt dette skal gerne ruste læseren til gennemgangen af selve RSA-systemet, hvor også et konkret eksempel på kryptering og dekryptering med systemet vil være at finde. Afsnittet vil blive afrundet med en vurdering af, hvorfor systemet i praksis er ubrydeligt, og hvilke anvendelsesmuligheder systemet har i nutidens samfund. Fremtiden kan dog true RSA-systemet, og derfor vil teksten afslutningsvis bringe læseren igennem dels truslen mod systemet og dels et nyt system, der forventes at kunne afløse RSA-systemet. Både truslen og redningen bygger begge på kvanteteorien, hvorfor denne skitseres.

En del områder af teksten indeholder aspekter, som kan virke abstrakte og umiddelbart stridige mod almindelig sund fornuft. Der gøres derfor brug af mange eksempler, analogier og enkelte tankeeksperimenter, der skal gøre stoffet og emnerne lettere at forstå – også for de, der ikke allerede er fortrolige med området. Som afrundende bemærkning gøres der opmærksom på, at teksten tager sit udspring i den talteoretiske del af matematikken, hvorfor det er underforstået, at der kun arbejdes med hele tal, medmindre andet er nævnt.

2. Begreber og definitioner



Figur 1: Modulær aritmetik i hverdagen

- "Hvad er klokken?"

- "Klokken er otte," lyder et svar.

- "20:00," svarer en anden.

Mange har formodentlig stået i en situation i stil med ovenstående. Man stiller et spørgsmål angående tiden, og får to umiddelbart forskellige svar. Men man ved egentlig godt, hvad der menes, og begge svar er i bund og grund korrekte: Klokken er otte om aftenen. Figur 1 viser, hvordan man kan referere til det samme klokkeslæt med to forskellige tal.

Der findes inden for matematikken et område kaldet *modulær aritmetik*, som netop beskriver situationer som ovenstående. Modulær aritmetik bruges desuden flittigt inden for kodeteori, hvorfor dette hovedafsnit vil definere og forsøge at klarlægge de nye begreber, som modulær aritmetik medbringer, da de vil blive brugt gennem resten af teksten.

2.1 Kongruens og restklasser

Man skriver, at to hele tal a og b er kongruente modulo n, hvis det hele tal n (der ikke må være 0) går op i forskellen mellem a og b.

Definition 1[1]

$$a \equiv b \pmod{n}$$
$$\Leftrightarrow n|(a-b)$$

Lad os vende tilbage til eksemplet fra indledningen:

$$20 - 8 = 1 \cdot 12 + 0$$

Da resten er 0, siges 12 at gå op i forskellen 20 – 8, og vi skriver derfor, at

$$20 \equiv 8 \pmod{12}$$

Dette læses som, at 20 og 8 kongruente modulo 12. 12 kommer i dette eksempel af, at en urskive kun kan vise 12 forskellige timer, og man derfor "starter forfra" hver 12. time.

[1] *Algebra og talteori*, s. 27

Definition 1 er ensbetydende med, at hvis man dividerer et tal *a* med *n* og får resten *r*, så er *a* kongruent til alle andre hele tal *b*, der har samme rest *r* ved division med *n*. Alle disse tal *b* kaldes *restklassen* til *a*:

Definition 2[2]

$$[a] = \{b | a \equiv b \pmod{n}\} = \{\dots, a-2n, a-n, a, a+n, a+2n, \dots\}$$

Vi vil igen se på eksemplet fra før og vise, at 8 og 20 tilhører samme restklasse:

$$8 = 0 \cdot 12 + 8$$
$$20 = 1 \cdot 12 + 8$$

Her ses det mere klart, at både 8 og 20 har samme rest, når de deles med 12. Da resten er 8 tilhører de begge restklassen til 8.

Vi ved også fra eksemplet efter Definition 1, at 8 og 20 er kongruente modulo 12, og da Definition 2 siger, at restklassen til *a* består af alle de tal *b*, for hvilke det gælder, at *a* og *b* er kongruente modulo *n*, ses det også her, at både 8 og 20 tilhører restklassen til 8.

Den sidste måde at vise dette på, er ved at skrive hvilke tal, restklassen indeholder ud fra Definition 2. Her ses det, at både 8 og 20 er elementer i mængden:

$$[8] = \{\dots, -16, -4, 8, 20, 32, \dots\}$$

Ovenstående tre eksempler viser alle på hver sin måde, at 8 og 20 tilhører samme restklasse.

To forskellige tal kan altså godt tilhøre samme restklasse. Men hvor mange restklasser findes der så? Hvis vi dividerer et hvilket som helst tal med *n*, kan vi højst få *n* forskellige rester (inklusiv 0). Dermed findes der præcis *n* restklasser, og de vil se således ud:

$$[0] = \{\dots, -2n, -n, 0, n, 2n, \dots\}$$
$$[1] = \{\dots, 1-2n, 1-n, 1, 1+n, 1+2n, \dots\}$$
$$\dots$$
$$[n-1] = \{\dots, (n-1)-2n, (n-1)-n, (n-1), (n-1)+n, (n-1)+2n, \dots\}$$

Et komplet sæt af restklasser med præcis en repræsentant fra hver restklasse kaldes en rest-klassering. Et eksempel på dette er

$$\mathbb{Z}_{12} = \{[0], [1], [2], [3], [4], [5], [6], [7], [8], [9], [10], [11]\}$$

[2] *Algebra og talteori*, s. 27

Her ses det tydeligt, at alle restklasserne er repræsenteret. Restklasserne behøver dog ikke at være repræsenteret ved deres hovedrester, som følgende eksempel viser

$$\mathbb{Z}_{12} = \{[12], [25], [14], [-9], [28], [17], [18], [-5], [20], [-3], [22], [35]\}$$

I dette eksempel er det ikke nær så tydeligt at se, at det er en restklassering. Men denne restklassering er endda den samme som før, fordi alle resterne modulo 12 også her er repræsenteret netop én gang. Som det kan ses, er eksempelvis restklassen til 8 nu repræsenteret med restklassen til 20 i stedet, og dette lader sig gøre, fordi vi tidligere fandt ud af, at 8 og 20 tilhører samme restklasse.

2.2 Eulers phi-funktion, primiske restklasser og ordner

I 1700-tallets Schweiz levede en matematiker ved navn Leonard Euler[3]. Ligesom så mange andre matematikere gennem tiden var han også fascineret af de såkaldte *primtal*. Et primtal er et tal, som kun har 1 og tallet selv som hele positive divisorer. Alle andre tal større end 1 kaldes sammensatte, fordi de entydig kan skrives som produktet af primtal. Et af Eulers bidrag til matematikken er *Eulers phi-funktion*. Denne funktion fortæller, hvor mange primiske restklasser en given restklassering indeholder. Hvis to tal eller restklasser er indbyrdes primiske, betyder det, at den største fælles divisor mellem disse to tal eller restklasser er 1. For tal skrives dette kort sådan: Når sfd(a,b) = 1 er a og b indbyrdes primiske. Efter samme princip skriver man det for restklasser: Når sfd([a],[b]) = 1 er [a] og [b] indbyrdes primiske.

Lad os videreføre eksemplet og undersøge restklasseringen $\mathbb{Z}_{12}$: 12 har divisorerne 2 og 3, da $2 \cdot 2 \cdot 3 = 12$. De primiske restklasser må derfor ikke have disse divisorer. De eneste restklasser i denne restklassering, der opfylder dette krav er $[1], [5], [7]$ og $[11]$. Dette noteres som de primiske restklasser til $\mathbb{Z}_{12}$ ved at tilføje en stjerne: $\mathbb{Z}_{12}^* = \{[1], [5], [7], [11]\}$.

Eulers phi-funktion $\varphi(n)$ fortæller som sagt, hvor mange elementer $\mathbb{Z}_n^*$ indeholder. Dette kaldes samtidig restklasseringens gruppeorden. I eksemplet med restklasseringen $\mathbb{Z}_{12}$ fås $\varphi(12) = 4$, da der er fire elementer i $\mathbb{Z}_{12}^*$, og restklasseringen $\mathbb{Z}_{12}$ har dermed gruppeordenen 4. Hvis p er et primtal, vil alle elementerne i restklasseringen $\mathbb{Z}_p$ bortset fra [0] være indbyrdes primiske med p. Derved vil $\mathbb{Z}_p^*$ indeholde alle elementerne fra [1] til [p – 1], og derfor vil $\varphi(p) = p - 1$, hvilket desuden betyder, at en restklassering for et primtal altid vil have gruppeordenen $p - 1$.

[3] *Algebra og talteori*, s. 46 og *Primtal – Matematikkens gådefulde tal fra A-Ø*, s. 77

Nu har vi kigget på restklasseringen og defineret en gruppeorden for denne. Men de enkelte primiske elementer har også en orden. En sådan orden kaldes en elementorden.

Definition 3

Elementordenen er den mindste potens en restklasse skal opløftes til for at give resten 1.

Af definitionen kan det ses, at man kun kan finde elementordner for primiske restklasser, da ingen andre restklasser i restklasseringen kan give resten 1 ved potensopløftning, fordi de har divisorer tilfælles med n. Lad os finde elementordnerne til de primiske restklasser fra eksemplet:

$$[1]^1 = [1] \text{ og derfor er } \text{ord}([1]) = 1$$
$$[5]^2 = [5^2] = [25] = [1] \text{ og derfor er } \text{ord}([5]) = 2$$
$$[7]^2 = [7^2] = [49] = [1] \text{ og derfor er } \text{ord}([7]) = 2$$
$$[11]^2 = [11^2] = [121] = [1] \text{ og derfor er } \text{ord}([11]) = 2$$

Som det kan ses, er eksempelvis elementordenen til [1] = 1 og elementordenen til [7] = 2. Som bemærkning kan det tilføjes, at *Lagranges sætning* siger, at en restklasses elementorden altid vil gå op i restklasseringens gruppeorden. Det kan lige noteres, at det stemmer her, da 1 og 2 begge går op i 4, men ellers vil denne tekst ikke beskæftige sig yderligere med Lagranges sætning.

Disse primiske elementer har også en anden egenskab. De har et inverst element i $\mathbb{Z}_n^*$ og dette vil vi se nærmere på nu.

Definition 4

Et primisk element ganget sit inverse element giver resten 1.

I eksemplet fra tidligere er der fire primiske restklasser, og det huskes, at $n = 12$. De primiske restklassers inverse elementer er følgende:

$$[1] \cdot [1] = [1 \cdot 1] = [1] \Leftrightarrow [1]^{-1} = [1]$$
$$[5] \cdot [5] = [5 \cdot 5] = [25] = [1] \Leftrightarrow [5]^{-1} = [5]$$
$$[7] \cdot [7] = [7 \cdot 7] = [49] = [1] \Leftrightarrow [7]^{-1} = [7]$$
$$[11] \cdot [11] = [11 \cdot 11] = [121] = [1] \Leftrightarrow [11]^{-1} = [11]$$

Som det ses, er alle de primiske restklasser i dette eksempel deres egen inverse. Det skal dog bemærkes, at dette ikke altid er tilfældet.

3. Primtalstest

Når vi nu har fået begreberne på plads, kan vi begynde at arbejde med dem og udnytte noget af den modulære aritmetiks styrke inden for kodningsteori og primtalstestning. Når man koder meddelelser ved hjælp af RSA-krypteringssystemet, som der vendes tilbage til senere, har man brug for to enorme primtal. Den umiddelbare måde til at teste om et tal p er et primtal eller ej, er ved blot at prøve at dele tallet p med alle de tal a, der er mindre eller lig med $\sqrt{p}$ og se, om man finder nogle divisorer til p. Hvis man ikke gør det, må p være et primtal.

Sætning 1

Ethvert heltal p er et primtal, hvis det ingen positive divisorer a og b har, der er mindre eller lig med $\sqrt{p}$, hvor $a, b \neq 1, p$.

Bevis

Det antages, at p er et sammensat tal. p kan derfor som minimum kun være produkt af to positive tal a og b, som ikke er hverken 1 eller tallet p selv

$$p = a \cdot b$$

Dette medfører, a og b ikke begge kan være større end $\sqrt{p}$. Hvis tallet er sammensat, er den ene divisor derfor mindre eller lig med $\sqrt{p}$. Hvis ikke man finder nogen divisorer, der er mindre eller lig med $\sqrt{p}$ må det derfor betyde, at p er et primtal. ■

Det kan bemærkes, at hvis man kender alle primtallene op til og med $\sqrt{p}$ behøver man endda kun at tjekke, om disse tal er divisorer. Alle andre tal i denne mængde vil jo være et produkt af disse primtal, og dermed vil man i sidste ende komme frem til de samme primfaktorer til p.

Denne metode fungerer udmærket ved undersøgelse af små tal, og man finder samtidig en faktorisering til et eventuelt sammensat tal, hvilket vi senere hen skal se det fornuftige ved. Men ligeså god metoden er til at teste små tal, ligeså ringe er den til at teste store tal. Jo større tallet p bliver, jo flere tal skal man forsøge sig med, og når p bliver tilstrækkeligt stort, tager det utrolig lang tid at teste og faktorisere det. Det er denne egenskab, man udnytter i RSA-systemet, som der vendes tilbage til senere.

Ovenstående er den umiddelbare måde at undersøge om et tal er et primtal eller ej. Men vi skal nu se nærmere på nogle metoder, der bygger på modulær aritmetik, og samtidig vil disses styrker og svagheder blive diskuteret.

3.1 Fermats lille sætning

Franskmanden Pierre de Fermat levede i 1600-tallet[4], og han betragtes i dag som den første moderne talteoretiker[5]. Han huskes blandt andet for *Fermats lille sætning*, som i dag ofte bruges i forbindelse med primtalsundersøgelser og dermed også i kodeteori. I dette afsnit vil vi se nærmere på denne sætning, bevise den samt diskutere dens styrker og svagheder.

Sætning 2[6]

Lad p være et primtal. For ethvert helt tal a, der er primisk med p, gælder det, at

$$a^{p-1} \equiv 1 \pmod{p}$$

eller udtrykt med restklasser modulo p

$$[a]^{p-1} = [1] \text{ i } \mathbb{Z}_p$$

Bevis

Vi starter med den primiske restklassering til p, da det i sætningen siges, at a og p skal være indbyrdes primiske, og p derfor ikke må gå op i a:

$$\mathbb{Z}_p^* = \{[1], [2], \dots, [p-1]\}$$

Hvis hver repræsentant i $\mathbb{Z}_p^*$ ganges med det hele tal a, fås den samme primiske restklassering blot med andre restrepræsentanter:

$$\mathbb{Z}_p^* = \{[a \cdot 1], [a \cdot 2], \dots, [a \cdot (p-1)]\}$$

Nu har vi to udtryk for $\mathbb{Z}_p^*$ og produkterne af de to restklasseringe må være kongruente modulo p, da p må gå op i forskellen mellem produkterne:

$$1 \cdot 2 \cdots (p-1) \equiv (a \cdot 1) \cdot (a \cdot 2) \cdots \big(a \cdot (p-1)\big) \pmod{p}$$

[4] *Algebra og talteori*, s. 37

[5] *Primtal – Matematikkens gådefulde tal fra A-Ø*, s. 101

[6] *Algebra og talteori*, s. 36

Venstresiden kan nu samles ved hjælp af udtrykket fakultet. På højre side sættes a uden for parenteserne, og det a er der $p-1$ af, fordi a blev ganget på hvert element i den oprindelige restklassering. Det, der står tilbage på højresiden, kan nu ligesom på venstresiden samles i fakultet:

$$(p-1)! \equiv a^{p-1} \cdot (p-1)! \pmod{p}$$

De to fakulteter af $p-1$ kan nu reduceres væk ved at gange med de inverse, og tilbage står den beviste Fermats lille sætning:

$$1 \equiv a^{p-1} \pmod{p} \blacksquare$$

En primtalstest baseret på Fermats lille sætning, vil ifølge Sætning 2 lade alle primtal passere, da den gælder for alle primtal. En sådan test er samtidig langt hurtigere at gennemføre end den første test vi så på, hvor alle tal mindre eller lig $\sqrt{p}$ skulle afprøves. Men der er et problem ved Fermats lille sætning. Den gælder også for nogle sammensatte tal. Det vil sige, at hvis et tal ikke opfylder sætningen, kan man med sikkerhed sige, at tallet er sammensat. Hvis tallet derimod opfylder sætningen og derved passerer en test baseret på sætningen, kan man kun sige, at der er stor sandsynlighed for, at tallet er et primtal. Da der findes flere tal, der opfylder kravene til a, kan man vælge et nyt a og gennemføre testen igen. På den måde kan sandsynligheden for, at et sammensat tal slipper igennem mindskes.

De sammensatte tal, der opfylder Fermats lille sætning og slipper igennem testen, kaldes pseudoprimtal. Man kan som sagt reducere sandsynligheden for pseudoprimtal ved at gennemføre testen med forskellige værdier af a. Der findes dog sammensatte tal, som vil passere testen for alle værdier af a. Disse tal kaldes absolutte pseudoprimtal eller Carmichael-tal. Disse tal er dog meget sjældne: Der findes kun 105.212 Carmichael-tal[7] under en million milliarder (10^{15}). I Figur 2 er de 10 mindste vist. Sandsynligheden er derfor meget lille for, at et sammensat tal vil bestå en test baseret på Fermats lille sætning for alle a – men den er der, og det vil være en svaghed ved en test baseret på Fermats lille sætning.

561	6.601
1.105	8.911
1.729	10.585
2.465	15.841
2.821	29.341

Figur 2: De 10 mindste Carmichael-tal

[7] *Primtal – Matematikkens gådefulde tal fra A-Ø*, s. 38

3.2 Wilsons sætning

Wilsons sætning er opkaldt efter John Wilson, der levede i slutningen af 1700-tallet[8]. Ligesom Fermats lille sætning benytter denne sætning sig også af modulær aritmetik:

Sætning 3[9]

Hvis og kun hvis *p* er et primtal, gælder det, at

$$(p-1)! \equiv -1 \pmod{p}$$

Bevis

Sætningen er todelt. Først skal det vises, at sætningen gælder for primtal, og dernæst at den aldrig gælder for sammensatte tal.

Del 1 vil blive vist med et direkte bevis: Hvis *p* er et primtal, vil den primiske restklassering $\mathbb{Z}_p^*$ indeholde alle elementerne fra $[1]$ til $[p-1]$. I henhold til sætningen ganges disse elementer sammen, og ifølge Definition 4 vil to inverse elementer give produktet 1. Vi vil derfor undersøge hvilke og hvor mange elementer, der er sin egen inverse. De elementer, der er sin egen inverse vil nemlig ikke have nogen invers "partner" at blive ganget med, hvorfor disse elementer er interessante for resultatet.

Hvis *a* er sin egen inverse er $[a]^2 = [a^2] = [1]$ ifølge Definition 4. Dette betyder, at

$$a^2 \equiv 1 \pmod{p}$$

$$\Leftrightarrow p|(a^2-1)$$

Dette følger af Definition 1. a^2-1 kan omskrives med kvadratsætninger til $(a+1)(a-1)$. Da *p* er et primtal og går op i produktet a^2-1, må *p* enten gå op i $(a+1)$ eller $(a-1)$:

$$p|(a+1) \vee p|(a-1)$$

$$\Leftrightarrow a \equiv -1 \pmod{p} \vee a \equiv 1 \pmod{p}$$

$$\Leftrightarrow [a] = [-1] \vee [a] = [1]$$

Af dette ses, at de eneste elementer i $\mathbb{Z}_p^*$, der er sin egen inverse er $[-1]$ og $[1]$. Elementet $[-1]$ er det samme som $[p-1]$.

[8] *Primtal – Matematikkens gådefulde tal fra A-Ø*, s. 264

[9] *Algebra og talteori*, s. 33 og *Talteori*, s. 84

Når alle andre elementer end $[-1]$ og $[1]$ i $\mathbb{Z}_p^*$ ganges sammen vil de give $[1]$, fordi de parvis vil være hinandens inverse. Som det ses nedenfor, vil det derfor i alt give resten – 1:

$$[1] \cdot [1] \cdot [1] \cdot [1] \cdots [-1] = [-1]$$

$$\Leftrightarrow (p-1)! \equiv -1 \pmod{p}$$

Dermed er det bevist, at Wilsons sætning gælder for primtal. Nu skal vi se på, hvorfor den ikke gælder for sammensatte tal:

Del 2 vil blive vist med et indirekte bevis: Det antages, at *p* er et sammensat tal bestående af minimum to positive faktorer: $p = a \cdot b$, hvor *a* og *b* ikke er 1 eller tallet *p* selv. Så vil *a* forekomme blandt faktorerne i $(p-1)!$:

$$(p-1)! = 1 \cdot 2 \cdots a \cdots (p-1) \equiv -1 \pmod{p}$$

a kan sættes uden for den parentes, de andre tal nu bliver omsluttet af, og der ganges med – 1:

$$-\big(1 \cdot 2 \cdots (p-1)\big) \cdot a \equiv 1 \pmod{p}$$

Dette betyder ifølge Definition 4, at *a* må være et inverst element, da det ifølge ovenstående udsagn giver resten 1, når det ganges med et andet tal. Hvis *a* er et inverst element, må $a \in \mathbb{Z}_p^*$. Men dette betyder samtidig, at *a* er indbyrdes primisk med *p*. Men *p* er jo netop et produkt af *a* og *b*, og vi er dermed kommet frem til en modstrid, som viser, at Wilsons sætning ikke gælder for sammensatte tal. På baggrund af bevisets to dele kan det konkluderes, at Wilsons sætning kun gælder for primtal. ■

Hvis en primtalstest baseres på Wilsons sætning, vil man være helt sikker på om tallet, der testes, er et primtal eller ej. Dette er selvfølgelig en stor styrke. Men Wilsons sætning gør desværre brug af fakultetstegnet, og dermed skal *p* ikke være særlig stor før, det tager lang tid at gennemløbe en test baseret på denne sætning. Dog kan man forestille sig, at en kombineret Fermat-Wilson-test, vil være lidt mere effektiv. Med Fermats lille sætning kan populært sagt skille fårene fra bukkene, og nøjes med at lade de tal, der måske er pseudoprimtal, gennemløbe en test baseret på Wilsons sætning. Dermed vil man hurtigt få skilt mange sammensatte tal fra ved hjælp af Fermats lille sætning, og man vil blive helt klar over, om tallet er et primtal, eller om det blot er et pseudoprimtal ved hjælp af Wilsons sætning. Dette vil dog stadig tage lang tid, og derfor er talteoretikerne til stadighed på jagt efter en bedre metode til primtalstestning.

Begge ovenstående sætninger er mere end 200 år gamle. Men i 2002 skete der noget interessant. De tre indiske matematikere Agrawal, Kayal og Saxena fandt en metode til at teste primtal i polynomiel tid, hvilket er hurtigere end eksempelvis en metode baseret på Wilsons sætning. Denne nye test kaldes AKS-algoritmen. Sætningen bag denne algoritme lyder:

Sætning 4 (AKS)[10]

Lad a være indbyrdes primisk med p. Så er p et primtal, hvis og kun hvis

$$(X - a)^p \equiv (X^p - a^p) \pmod{p}$$

i polynomiumsringen $\mathbb{Z}[X]$.

Denne sætning vil ikke blive bevist, men det ridses kort op, hvilke muligheder denne sætning giver. En test baseret på denne sætning gør det muligt at primtalsteste med lige så stor sikkerhed, som Wilsons sætning giver – denne test gør det bare hurtigere. Det vil dog stadig tage en del tid ved store værdier for p, og derfor er der sikkert stadig mange, der søger med lys og lygte efter den perfekte primtalstest, som både er sikker og hurtig – og måske endda også giver en faktorisering af et eventuelt sammensat tal. Men indtil da vil RSA-krypteringssystemet stadig være sikkert, og hvorfor ses der nærmere på i næste afsnit.

[10] *Primtalsfaktorisering – nogle nye resultater og anvendelser*, s. 10

4. RSA-krypteringssystemet

RSA-krypteringssystemet gør op med mange af fortidens store problemer ved kryptering. For at synliggøre RSA-systemets virkelige styrker vil dette afsnit derfor starte med en gennemgang af kampen om koderne op gennem historien og kodesystemernes styrker og svagheder.

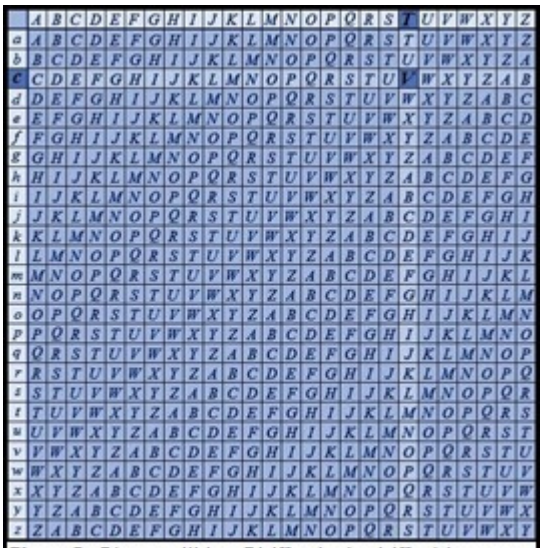
4.1 Krypteringssystemernes historie

Gennem tiden har der altid været en kamp mellem kodemagerne og kodebryderne. Lige fra oldtidens krige frem til nutidens tusindvis af daglige finansielle transaktioner har man haft brug for hemmeligholdelse af oplysninger. Det er blevet påstået om mange af krypteringssystemerne, at de er ubrydelige, selvom dette sjældent har været korrekt. Mange af de såkaldte ubrydelige koder er blevet brudt, fordi snedige kryptanalytikere, alligevel har fundet gentagelser og systematik i det ellers umiddelbare volapyk. Gang på gang er kodemagerne gået i tænkeboks og har forsøgt sig med nye og stærkere systemer. Omkring Romerrigets storhedstid benyttede Cæsar[11] sig af koder, hvor alle bogstaverne i en tekst blev erstattet med det bogstav, som kom tre pladser senere i alfabetet. Skrevet matematisk vil bogstavet x bliver erstattet med y i henhold til $y \equiv x + 3 \pmod{29}$.

Denne form for kode kan brydes forholdsvist simpelt med hyppighedsanalyse af bogstaverne eller med et såkaldt *brute force-angreb*. I et brute force-angreb afprøves samtlige mulige nøgler, og denne metode vil derfor med sikkerhed kunne dekryptere en tekst (såfremt teksten ikke er ren nonsens). De moderne krypteringssystemer er dog sikret mod sådanne angreb, fordi der er så mange mulige nøgler, at det vil tage flere gange universets levetid at gennemgå dem alle – selv med nutidens supercomputere. Men et krypteringssystem à la Cæsars er imidlertid sårbart over for et sådan brute force-angreb. Følgende eksempel vil vise hvorfor: Hvis det antages, at alle bogstaverne i en tekst er blevet erstattet med bogstaver x pladser fremme i et alfabet på 29 bogstaver, kan bogstaverne højst erstattes med bogstaver 28 pladser fremme (den sidste mulighed er jo startbogstavet igen). Dette giver kun 28 mulige nøgler. At gennemgå 28 forskellige nøgler er ikke uoverkommeligt, og statistisk set vil man allerede ved det 14. forsøg have fundet den rigtige nøgle. At antallet af mulige nøgler er så lille skyldes, at alle bogstaverne er forskudt det samme antal pladser frem i alfabetet. Netop dette problem blev taget op af kodemagerne, og de begyndte at udvikle nye kodesystemer. Dette resulterede

[11] *Drømmen om den sikre kode er gået i opfyldelse*, s. 23

blandt andet i systemet *Le Chiffre Indechiffrable*, som ses i Figur 3. I dette system bliver en tekst krypteret med en anden tekst. Dette resulterer i, at alle bogstaverne ikke bliver flyttet lige mange pladser i alfabetet. Dermed findes der 29 mulige nøgler per bogstav i teksten, og dette gør systemet mere modstandsdygtigt over for brute force-angreb. Som navnet antyder, mente man altså, at man nu havde fundet det ubrydelige kodesystem.



Figur 3: Skema til Le Chiffre Indechiffrable

Men ligesom tidligere i historien fandt snedige kodebrydere og kryptanalytikere gentagelser i måden dette system blev brugt på. I bund og grund er Le Chiffre Indechiffrable det samme som Cæsars system – erstatningen af bogstaver er blot individuel. På grund af problemer med nøgledistribuering valgte man derfor ofte et kodeord som for eksempel ”Cæsar” og erstattede så bogstaverne med bogstaverne 3, 27, 19, 1 og 18 pladser længere fremme i alfabetet. Sjældent var beskederne på kun fem bogstaver, så man gentog kodeordet igen og igen, indtil hele teksten var krypteret. Det var denne gentagelse kombineret med, at kodeordet var meningsfuldt, som kodebryderne opdagede. Men hvad gør man så? Man begyndte at indse, at såkaldte engangsblokke kunne være farbar vej. Engangsblokke gør også brug af Le Chiffre Indeciffrable, men for det første er engangsblokkene længere end teksten, der skal kodes, og derved undgås gentagelser. For det andet er kodeordet optimalt set fuldstændigt tilfældigt. Dette gør, at eventuelle kodebrydere intet har at hænge sig i. Hvis kodeordet, som nok mere korrekt kan betegnes som rækken af kodebogstaver, er fuldstændig tilfældig er der nemlig per definition ingen systematik at gribe fat om. Det er bevist, at engangsblokke er ubrydelige, og nedenfor vil vi se på et lille tankeeksperiment, der viser hvorfor:

Vi står i kodebrydernes sted, og har opsnappet fjendes kommando, der er blevet kodet med en engangsblok. Den kodede tekst er ”BJWRID”. Vi indleder et brute force-angreb ved at forsøge med forskellige nøgler og kommer frem til, at nøglen ”BQDRGT” giver klarteksten ”attack”. Det lader altså umiddelbart til, at fjenden vil angribe. Men vi lader brute force-angrebet stå på lidt endnu og finder frem til, at nøglen ”YFRNVA” giver klarteksten ”defend”. Men hvad nu? Vil fjenden angribe eller forsvare sig? Ingen af nøglerne viser tegn på systematik eller gentagelse, så vi er på herrens mark, når vi står i kodebryderens sted. Som

sagt vil et brute force-angreb helt sikkert give den rigtige nøgle og dermed den rigtige klartekst. Men angrebet vil også give alle andre ord på seks bogstaver, og sikkerheden ligger dermed i, at kodebryderne ikke har mulighed for at afgøre, hvilken nøgle og hvilken klartekst, der er den korrekte. I dette eksempel vil kodebryderen få 29^6 forskellige løsninger. Selvfølgelig vil kun en brøkdel heraf være meningsfulde ord, men som det ses i eksemplet er blot to forskellige løsninger nok til at sikre hemmeligholdelsen. Kodebryderen har nok regnet ud, at fjenden enten vil forsvare sig eller angribe, så han har dermed ikke opnået ny viden.

Engangsblokke er altså ubrydelige, men i praksis bruges de sjældent, da de har en stor svaghed – nemlig nøgledistribueringen. De kommunikerende parter skal begge være i besiddelse af et sæt identiske engangsblokke, og hver kodeblok må, som navnet antyder, kun benyttes én gang. Dette skyldes, at hvis den samme kode bruges to gange, vil en kodebryder have mulighed for at finde en gentagelse og et system i krypteringen, og dermed vil det som tidligere nævnt være muligt at bryde koden. Kort fortalt vil han for eksempel med et brute force-angreb se, at den nøgle, der giver en meningsfuld klartekst i begge tilfælde, må være den korrekte. Nøgledistribueringen er engangsblokkenes akilleshæl.

Alle de krypteringssystemer, vi har set på indtil nu, har brugt såkaldt symmetrisk kryptering, hvilket vil sige, at man bruger den samme nøgle til kryptering som til dekryptering. Dette sætter et krav om, at de to parter er nødt til at have en fælles hemmelighed. I en moderne verden, hvor vi kan kommunikere med personer vi aldrig før har mødt, er det dog svært at imødekomme et sådan krav om fælles hemmelighed med de systemer, vi indtil nu har set på. Der findes dog et symmetrisk krypteringssystem kaldet Diffie-Hellman-Merkle-systemet, der tillader, at parterne i al offentlighed kan finde frem til fælles nøgle. Systemet kræver dog, at parterne kommunikerer i realtid, hvilket ofte er upraktisk. Dette system vil derfor ikke blive omtalt yderligere i denne tekst. I stedet vil blikket blive rettet mod tre andre herrer – Ronald Rivest, Adi Shamir og Leonard Adleman. Disse tre mænd tænkte, at det måtte være muligt at lave et krypteringssystem, hvor man kunne bruge én nøgle til at kryptere med og en anden til at dekryptere med – såkaldt asymmetrisk kryptering. Dermed ville problemet med nøgledistribuering være løst. Umiddelbart lyder dette forrykt, men følgende analogi viser tankerne bag systemet:

De to personer Alice og Bob vil gerne udveksle informationer. Men der er et par problemer: For det første er der den onde Eve, som forsøger at opsnappe denne kommunikation, og

for det andet har Alice og Bob ingen fælles hemmelighed, som de kan bruge som kode. Til gengæld har de et sindrigt nøglesystem. Alice har nemlig en kasse stående uden for sit hus fyldt med hængelåse, som man "klikker sammen" for at låse. Alle kan frit komme forbi og hente disse hængelåse. Senere hen vil vi kalde disse hængelåse for Alices *offentlige nøgle*, men i eksemplet kan de betragtes som fysiske hængelåse. Alice er den eneste person i verden, der har en nøgle til disse hængelåse, og denne nøgle vil vi senere hen kalde for Alices *private nøgle*. Bob vil gerne sende en besked til Alice. Han har desværre ikke selv mulighed for at hente en hængelås hos Alice, så han ringer til Alice på en ubeskyttet telefonlinje og beder hende sende ham en hængelås. Det er uden betydning, at den onde Eve overhører denne samtale, da Eve ligeså godt selv kunne have hentet en hængelås hos Alice. Når Bob er kommet i besiddelse af Alices hængelås, finder han en kasse frem, fylder den med de hemmelige beskeder til Alice, sætter hængelåsen på og klikker den sammen. Bob sender nu kassen af sted til Alice. Eve forsøger selvfølgelig at åbne kassen undervejs, men da Alice er den eneste person i verden med nøglen til hængelåsen, kan Eve ikke åbne den – selv ikke Bob kan åbne sin kasse igen. Når kassen kommer frem til Alice, tager hun sin nøgle og åbner kassen, og dermed har Bob og Alice udvekslet oplysninger med sikkerhed for, at Eve undervejs ikke har læst med.

Det er muligvis et tankeeksperiment som ovenstående, der har fået de tre herrer til at gå ind i jagten på et asymmetrisk krypteringssystem. Efter en del år lykkedes det dem at beskrive og løse ovennævnte situation matematisk, og RSA-krypteringssystemet var født.

4.2 Matematikken bag RSA[12]

Lad os se nærmere på matematikken og principperne bag RSA-systemet. Alice, Bob og Eve vil igen være hovedpersoner i teksten for at lette overskueligheden. Scenariet er ligesom i historien ovenfor, at Bob gerne vil sende en besked til Alice uden, at Eve læser med.

Alice vælger to enorme primtal, som kaldes p og q. Når disse multipliceres fås N:

$$p \cdot q = N$$

Det er et krav, at N er større end teksten, der skal krypteres, da der ellers kan opstå tvetydighed. Hvis dette ikke er muligt, kan teksten alternativt splittes op i blokke, der er mindre end N. Når N er valgt, vælges et andet tal k, som skal være mindre end og indbyrdes primisk med produktet $(p-1)\cdot(q-1)$:

[12] *Diskrete Matematik*, s. 250 og *Kodebogen* appendiks J

$$k < (p-1)\cdot(q-1) \ \wedge \ \mathrm{sfd}\big(k,(p-1)\cdot(q-1)\big) = 1$$

Disse to tal *N* og *k* skal alle have adgang til for at kunne kryptere en besked til Alice, ligesom alle havde adgang til hængelåsene foran Alices hus. Denne del af systemet er altså offentlig, og tilsammen kaldes de to tal derfor den offentlige nøgle. Den tekst, der skal krypteres, skal først omformes til en talværdi, og dette gøres ud fra en almindelig standard på samme måde, som når tekst gemmes elektronisk på en computer. Klarteksten, som kaldes *T*, skal Bob herefter omforme til kodeteksten *C* i henhold til følgende:

$$C \equiv T^k \pmod{N}$$

Bob kan nu trygt sende *C* af sted til Alice velvidende, at Eve ikke kan forstå indholdet af *C*. Selvom Eve sidder med *C*, *k* og *N*, lader det sig ikke let gøre at finde *T*. Potensopløftning er nemlig en envejsfunktion i modulær aritmetik, hvilket vil sige, at det er let at udføre beregningen, men svært at tilbageføre beregningen igen. En envejsfunktion kan sammenlignes med, at det er let at blande to forskellige farver maling og derved få en tredje farve, men det er umuligt at skille de to startfarver fra hinanden igen ud fra den tredje farve.

Inden for modulær aritmetik er det dog heldigvis ikke helt umuligt at skille farverne igen og komme frem til *T*. Kravet er blot, at man er i besiddelse af specielle informationer, så man kan lave den private nøgle. Den information har Alice, og derfor kan hun som den eneste dekryptere *C* og finde frem til klarteksten *T*. Først må hun dog finde den såkaldte dekrypteringsnøgle *d*, for hvilken det skal gælde, at:

$$k \cdot d \equiv 1 \ \big(\mathrm{mod}\ (p-1)\cdot(q-1)\big)$$

Det vil sige, at Alice skal finde en dekrypteringsnøgle, der giver resten 1, når den ganges med *k* og derpå divideres med $(p-1)\cdot(q-1)$. Umiddelbart er dette ikke altid helt let ved store værdier, men ved hjælp af en algoritme kaldet *Euklids algoritme* kan *d* findes forholdsvist hurtigt og enkelt. Når Alice har fundet sin dekrypteringsnøgle *d*, kan hun gå i gang med at dekryptere *C* i henhold til følgende:

$$T \equiv C^d \pmod{N}$$

Resultatet *T* er selvfølgelig et tal, men Alice og Bob kan sagtens på en almindelig ubeskyttet telefonlinje aftale efter hvilken standard, de omformer bogstaver til tal og omvendt uden, at det påvirker systemets sikkerhed. Alice kan nu læse Bobs hemmelige besked og kommunikationen er fuldendt uden, at Eve læser med.

4.3 RSA-systemet i funktion

Ovenstående var en generel gennemgang af matematikken bag RSA-systemet. Lad os kigge på et eksempel, hvor vi vil sende en hilsen til Alice. For overskuelighedens skyld repræsenteres en hilsen med H, hvorfor teksten "H" skal krypteres, sendes til Alice og dekrypteres. Først omformer vi "H" til et tal ved hjælp af standarden ASCII:

$$\mathrm{H} \overset{\mathrm{ASCII}}{\longleftrightarrow} 48$$

Dernæst ringer vi til Alice og får at vide, at hendes offentlige nøgle består af $N = 323$ og $k = 17$. Vi kan nu gå i gang med at kryptere 48 i overensstemmelse med den før omtalte formel:

$$C \equiv 48^{17} \pmod{323}$$

Ved hjælp af computerprogrammet PARI kan det udregnes, at $C = 116$. Dette tal sendes af sted til Alice, som i mellemtiden har fundet frem til sin dekrypteringsnøgle. Da det er *hendes* offentlige nøgle, vi har brugt, kender hun primfaktorerne til 323, som er $p = 17$ og $q = 19$. Hun udregner nu $(17-1)\cdot(19-1) = 288$. Alice ved, at $17 \cdot d \equiv 1 \pmod{288}$. Hun kan derfor lægge 1 til 288 ganget med et helt tal *n* og dividere med 17: $\frac{288 \cdot n+1}{17}$. Det skal lige understeges, at denne metode *ikke* er Euklids algoritme. Euklids algoritme kan derimod være en genvej ved store tal, hvis ikke man vil afprøve utallige værdier for *n*. Hvis hun starter med $n = 1$, vil hun erfare, at løsningen derved giver et helt tal: $\frac{288 \cdot 1+1}{17} = 17$. Dekrypteringsnøglen *d* må derfor være lig 17. Alice kan nu dekryptere 116 ved hjælp af dekrypteringsnøglen:

$$T \equiv 116^{17} \pmod{323}$$

Igen benyttes PARI, og det udregnes, at $T = 48$. Med standarden ASCII omformer Alice 48 til "H", og Alice har dermed modtaget vores hilsen.

Men har Eve også set, at vi har sendt Alice en hilsen? Det er desværre nok en mulighed, da værdierne af *p* og *q* er meget små. Der skal ikke bruges mange forsøg på at faktorisere 323, og derved vil Eve stå i samme position som Alice. Men hvis nu *N* er på cirka 200 decimale cifre, så vil det tage ufattelig lang tid at opløse *N* i dets primfaktorer. Et sådan *N* ses her:

N = 39.043.656.902.719.242.085.753.614.048.425.830.343.803.692.745.353.074.063.178.015.830.810.832.527.415.469.899.218.792.334.291.586.
011.256.933.572.212.129.149.458.113.106.613.029.433.355.901.071.455.561.893.050.713.583.940.079.202.555.882.879.576.439.138.584.097.653

Dette *N* er skabt ved at multiplicere følgende to primtal *p* og *q* på hver 100 decimale cifre:

p = 4.957.838.141.206.920.636.707.162.388.612.521.660.657.194.646.840.332.191.564.354.922.875.171.507.828.382.105.609.952.936.182.740.401

q = 7.875.137.467.318.482.518.983.800.647.033.248.499.545.138.903.304.667.966.111.639.866.214.284.923.131.873.407.037.436.779.044.296.453

At gå fra p og q til N tager en brøkdel af et sekund med en computer, men at udlede disse to primtal fra N er ikke muligt inden for et acceptabelt tidsrum. Det er blandt andet dette, der er styrken i RSA-krypteringssystemet. Beskeden T kan dog også læses, hvis man ud fra det opsnappede C og den offentlige nøgle k og N kan finde T i ligningen $C \equiv T^k \pmod{N}$. Da dette er en envejsfunktion, findes der dog ingen kendte genveje til at løse problemet, og i praksis er man nødt til at forsøge med alle tænkelige værdier af T. Det er klart, at dette ligesom primfaktoropløsningen er umådeligt tidskrævende, så derfor er det i praksis ikke muligt at bryde RSA-systemet, hvis blot der benyttes tilstrækkelig store værdier.

Ovenfor er det vist, hvorledes RSA-systemet beskytter kommunikation. I dag bruges RSA-systemet blandt andet i de nye betalingskort med chip og generelt ved finansielle transaktioner. RSA-systemet er egentlig en tosidet sag, fordi det også kan bruges til autentifikation. I Danmark er *Digital Signatur* et eksempel på dette. Kort sagt vendes RSA-systemet om. Alice skal modtage en besked fra Bob. Bob vælger så at kryptere beskeden med sin private nøgle. Når Alice modtager beskeden, skal hun derfor bruge Bobs offentlige nøgle til at dekryptere beskeden med. Da Bobs offentlige nøgle er den eneste nøgle, der kan bruges til dekrypteringen, må det være Bob, der har afsendt beskeden. Eve kan umiddelbart også læse med her, men hun kan ikke udgive sig for at være Bob, da hun ikke kender hans private nøgle. Digital Signatur kan derfor bruges ligesom en underskrift, og ifølge dansk lovgivning er Digital Signatur ligeså bindende, som en underskrift er.

De to dele af RSA-systemet kan kombineres, så Bob først ”underskriver” beskeden med sin private nøgle og dernæst kryptere den med Alices offentlige nøgle. Beskeden kan derved siges at være pakket ind i to kasser. Alice er den eneste, der kan åbne den yderste kasse. Inden i den finder hun så en kasse, som alle kan åbne, men som kun kan være lukket af Bob, fordi hun skal bruge hans offentlige nøgle for at åbne den. Derinde ligger så den hemmelige besked, som med sikkerhed er fra Bob og umuligt kan være læst af andre, da den jo har været beskyttet af Alices offentlige nøgle. Det er disse egenskaber, der gør RSA-systemet så brugbart i nutidens samfund.

5. Et kvantespring ind i fremtiden[13]

Vi har set på, hvorfor en tekst krypteret med en kraftig RSA-nøgle kan ikke dekrypteres inden for et acceptabelt tidsrum med nutidens computere. Men som så mange gange før i historien lader kodebryderne sig ikke slå ud. De er begyndt at ty til nye våben i kampen om koderne. Fysikeren David Deutsch var en af de første til at beskrive et sådan nyt våben – kvantecomputeren. I stedet for at fungere på makroskopisk niveau, som normale computere gør, skal kvantecomputeren i stedet operere på mikroskopisk niveau, hvor kvantemekanikkens love virkelig fremviser sin forunderlige verden.

5.1. Kvantecomputeren

En kvantecomputer fungerer på baggrund af de kvantemekaniske love, så lad os først tage et kig på disse. Kvantemekanikken er nok et af de mest bizarre områder inden for fysikken, og nogle af tankeeksperimenterne inden for dette område tangerer det absurde. Men ikke desto mindre er det for eksempel kun de kvantemekaniske love, der gør det muligt at lave beregninger på kernereaktionerne i et atomkraftværk. Det er også kun disse love, der gør det muligt at designe de laserer, man finder i optiske drev som eksempelvis cd-drevet i computeren. Så selvom det følgende kan lyde utroligt, må der være noget om snakken.

Kvantemekanikken tillader, at elementærpartikler som fotoner kan befinde sig i en tilstand kaldet *superposition*. Denne tilstand kan forklares med et tankeeksperiment kaldet "Schrödingers kat": Katten kan være i to tilstande – enten død eller levende. Denne kat puttes ind i en kasse sammen med kapsel cyankalium, som vil dræbe katten, såfremt katten træder på denne kapsel. Ved forsøgets start kan det ved simpel iagttagelse konstateres, at katten er levende. Men så snart, der er kommet låg på kassen, og katten og kapslen med cyankalium er ude af syne, ryger iagttageren ind i uvished. Har katten trådt på kapslen eller ej? Da det ikke er muligt at afgøre om katten derved er død eller levende, siger kvantemekanikken er den både er død og levende – samtidig! Katten befinder sig altså i to tilstande på én gang, og man siger, at katten er i superposition. Så snart kassen åbnes igen, er katten synlig, og dermed tvinges den ind i én bestemt tilstand, og superpositionen ophæves.

Man mener, at det samme gør sig gældende for fotoner, og det er dette fænomen, man vil udnytte i en kvantecomputer. Fotoner har et såkaldt *spin*, som kan sammenlignes med en

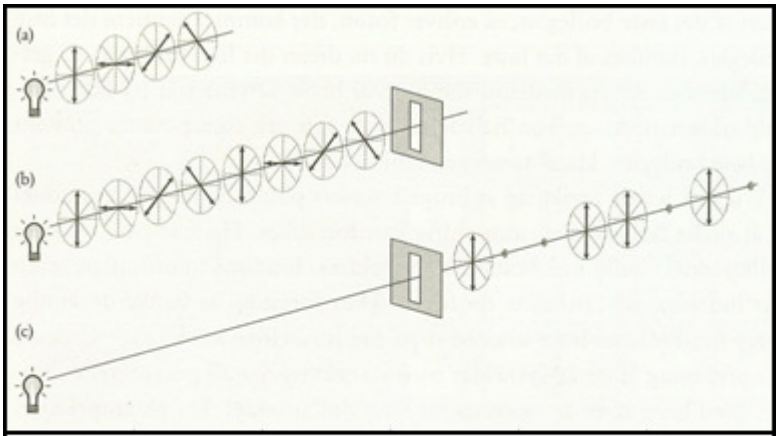
[13] *Kodebogen*, kap. 8

bold, der roterer. Ligesom bolde kan rotere med eller mod uret, kan fotoner have forskellige spin. Tanken er så, at indsætte eksempelvis syv fotoner i en kvantecomputer. Tilsammen kan deres spin kombineres på $2^7 = 128$ forskellige måder. I en traditionel computer vil man være nødt til at afprøve alle kombinationer en ad gangen for at finde løsningen på for eksempel en envejsfunktion eller en primfaktoropløsning. Men teorien siger, at man i kvantecomputeren kan lade alle syv fotoner indtræde i superposition. Dermed repræsenterer de syv fotoner alle de mulige kombinationer af spin samtidig, og løsningen kan derved findes på blot et enkelt trin, hvorimod den traditionelle computer skal bruge 128 trin for at gennemgå alle kombinationer. Kvantecomputeren vil derfor være velegnet til brute force-angreb, hvor den kan gennemgå alle tænkelige nøgler på engang, eller til brydning af RSA-systemet, hvor primfaktorerne til den offentlige nøgle vil kunne findes lynhurtigt.

Umiddelbart ser det altså ud til at kodebryderne vil have vundet kampen om koderne, så snart kvantecomputeren er blevet opfundet. Men lige så vel som kodebryderne ikke giver op uden videre, er kodemagerne også ved at gøre sit næste våben klar.

5.2 Kvantekryptografi

Kodemagerne har også set potentialet i kvantemekanikkens love. Et kvantekryptografisk system kan i fremtiden, når RSA- og andre krypteringssystemer er blevet brudt, stadig sikre privatlivets fred, virksomhederne mod industrispionage og nationerne mod andre nationers snagende efterretningsvæsner. Kvantekryptografien udnytter, at fotoner kan have en bestemt polarisation. Når lyset bevæger sig, bevæger alle fotonerne sig i samme retning. Men hver enkelt foton kan "vibrere" i en hvilken som helst retning. Det er denne "vibration", der kaldes fotonens polarisation. Almindeligt sollys indeholder fotoner med alle slags polarisation. Ved hjælp af polariserende glas, kan man dog ensrette fotonernes polarisation. Man kan i nogle



Figur 4: (a) Lyskilden udsender fotoner med alle slags polarisationer. (b) Fotonerne sendes mod et lodretpolariserende gitter. (c) De lodretpolariserede fotoner slap uhindret igennem gitteret, og de vandretpolariserede blev blokeret. Halvdelen af de diagonalt polariserede fotoner slap imidlertid også igennem og er nu lodretpolariserede.

tilfælde betragte fotonernes polarisation som tændstikker. Kun de tændstikker, der står lodret kan komme igennem et lodret gitter. Derimod vil alle de vandretliggende tændstikker blive standset. Der henvises eventuelt til Figur 4[14]. Fotoner kan dog være polariserede i alle retninger, og tændstikeksemplet kan ikke bruges til at forklare følgende. Det forholder sig nemlig således, at fotoner med en diagonal polarisation i forhold til gitteret vil passere halvdelen af gangene og blive blokeret halvdelen af gangene. De fotoner, der kommer igennem gitteret vil derefter have samme polarisationsretning som gitteret. At dette må være sandt kan vises med følgende eksperiment, som jeg selv har udført, og som alle med Polaroid-solbriller er i stand til at udføre. Der bør bruges Polaroid-solbriller, da disse solbriller med sikkerhed har polariserende glas, fordi virksomheden bag udnytter princippet til at reducere genskær.

Til eksperimentet skal der bruges tre par Polaroid-solbriller[15]. Det ene par tages på som normalt, og man kigger mod en stærk lyskilde som for eksempel en glødepære eller solen (kig dog ikke direkte imod solen). Det andet par holdes dernæst på tværs af det første par, så man kigger igennem de to solbrilleglas. Da de to solbrilleglas begge indeholder polariserende gitre, som nu ligger på tværs af hinanden, vil man erfare, at alt lyset bliver blokeret i overensstemmelse med den føromtalte teori. Det interessante fremkommer imidlertid, når man så tager det tredje par solbriller og holder det diagonalt imellem de to andre. Nu kan man pludselig se igennem alle tre lag solbriller! Fænomenet kan forklares ved, at 50 % af det lys, der passerer det første par solbriller, kommer igennem det diagonale glas. 50 % af dette lys kommer så igennem det sidste glas, og dermed kommer i alt 25 % af lyset, som passerede det første par solbriller ud til modtageren bag det sidste glas. Derimod tillader kvantemekanikken ikke, at polarisationen ændres fra lodret til vandret på én gang, og derfor kommer der intet lys igennem i den første del af forsøget, hvor der kun bliver brugt to brilleglas.

Resultatet af forsøget må betyde, at det er muligt at manipulere med fotonernes polarisation, og at man risikerer at ændre polarisationen, når fotonerne undersøges. Da fotonerne ikke kan deles, har man samtidig kun én chance for at undersøge deres polarisation. Dette udnyttes i det kvantekryptografiske system, som vi vil se nærmere på nu.

Alice og Bob lever nu i en tid, hvor de traditionelle krypteringssystemer er blevet brudt, men de ønsker stadig at kunne kommunikere uden, at Eve lytter med. Alice har to skemaer til rådighed: Et retlinjet og et diagonalt. Det retlinjede skema (+) bruger hun til at sende lodret-

[14] Billedet er fra *Kodebogen*, s. 347. Billedteksten har jeg selv tilføjet.
[15] Alternativ kan man bruge to par Polaroid-solbriller og kigge imod en LCD-skærm, da denne udsender polariseret lys.

og vandretpolariserede fotoner. Det diagonale skema (×) bruger hun på samme måde til at sende diagonalt polariseret fotoner. Med hvert skema kan Alice sende fotoner med to forskellige polarisationer. Hun bliver enig med Bob over en ubeskyttet telefonlinje om, at de vandretpolariserede fotoner repræsenterer 0, og at de lodretpolariserede repræsenterer 1. Det samme gør hun for det diagonale skema. Hun begynder derpå at sende en tilfældig række af nuller og ettaller samtidig med, at hun tilfældigt polariserer fotonerne ved hjælp af de to skemaer. Denne sekvens modtager Bob, og han skal afgøre, hvornår han vil bruge det ene detektorskema til at undersøge fotonernes polarisation, og hvornår han vil bruge det andet. Da han ikke aner, hvilket skema Alice har brugt, vil han selvfølgelig gætte forkert i nogen tilfælde og rigtigt i andre tilfælde. Hvis han har brugt det samme skema som Alice, kan han være sikker på at have aflæst fotonen korrekt. I de tilfælde de ikke har brugt det samme skema, kan Bob imidlertid ikke være sikker på, at han har noteret sig den korrekte værdi, som fotonen repræsenterer. Efter modtagelsen af alle fotonerne ringer Bob derfor til Alice på en ubeskyttet telefonlinje og fortæller hende, hvornår han brugte hvilke detektorskemaer. Alice fortæller nu Bob, hvornår han gættede rigtigt og dermed brugte det rigtige skema, og hvornår han ikke gjorde. Hun fortæller ham dog ikke, hvilken polarisation fotonerne havde. De tal Bob har noteret med det forkerte skema smides nu væk af både Alice og Bob, og begge står tilbage med en kortere – men ens – sekvens af tilfældige tal, som kan bruges som engangsblok.

Eve vil i starten stå i samme situation som Bob, hvor hun undertiden vil vælge det forkerte detektorskema. Men hvor Bob og Alice kan blive enige om at smide Bobs fejlresultater væk fra den endelige nøgle, hænger Eve på sine fejl, og hun vil ikke have kendskab til den fulde engangsblok. Sikkerheden ligger dermed i, at Eve kun har en chance for at undersøge fotonernes polarisationer, men denne chance vil hun indimellem misse. Men ikke nok med, at Eve ikke kan opsnappe nøglen, hun vil også afsløre sin tilstedeværelse på linjen. I situationer som følgende vil Eve nemlig afsløre sig selv: Alice sender en lodretpolariseret foton af sted, og Eve vælger at undersøge den med et diagonalt skema. Som vi før har set, kan fotonen enten blokeres eller tvinges til have en diagonal polarisation. Lad os sige, at denne passerer med den nye polarisation. Når denne foton kommer frem til Bob, vælger han tilfældigvis at undersøge den med det retlinjede skema. Den værdi denne foton repræsenterer vil indgå i den endelige nøgle, da Alice og Bob valgte det samme skema, men på grund af Eve kan Bob have noteret sig en forkert værdi, fordi polarisationen kan være drejet 90 grader ligesom det skete med 25 % af fotonerne i forsøget med solbrillerne. For at tjekke om Eve forsøger at lytte med

og dermed risikerer at forandre fotonernes polarisation ringer Bob til Alice på en ubeskyttet telefonlinje og de sammenholder nogle få værdier af nøglen (under 1 %)[16]. Hvis nogle af værdierne er forskellige, må det betyde, at Eve er på spil, og Alice og Bob må begynde forfra på en ny linje. Men hvis alle værdierne er ens, er sandsynligheden utrolig lille for, at Eve har lyttet med. De sammenholdte værdier fjernes fra engangsblokken, og denne kan nu bruges til kryptering af meddelelsen.

Som tidligere vist er engangsblokke absolut ubrydelige, og det er nu vist, hvordan Alice og Bob ved hjælp af de kvantemekaniske love over afstand kan blive enige om en fuldstændig tilfældig og hemmelig engangsblok. Dette vil gøre det kvantekryptografiske system absolut ubrydeligt. De to parter vil endda være i stand til at afsløre en eventuel tilstedeværelse af Eve. Et kodesystem er mange gange i historien blevet kaldt ubrydeligt, men hvis man skal bryde det kvantekryptografiske system, er man nødt til at bryde de kvantemekaniske love. Det vil simpelthen betyde, at forskerne har misforstået, hvordan universet hænger sammen på det mest elementære niveau, og kvanteteorien vil i så fald være forkert. Sammenlignet med RSA-systemet kan det kvantekryptografiske system også i fremtiden forventes at være ubrydeligt, hvorimod RSA-systemet kun kan siges at være praktisk ubrydeligt i dag, men at det ikke kan udelukkes, at der i fremtiden bliver fundet genveje til enten primtalsfaktorisering, løsning af envejsfunktioner eller opfundet maskiner i stil med kvantecomputeren.

Men hvis dette kvantekryptografiske system er ubrydeligt, hvorfor bruges det så endnu ikke overalt? I slutningen af 1980'erne lykkedes det Charles Bennett, opfinderen af systemet, og den studerende John Smolin at få to computere til at blive enige om en engangsblok gennem et kvantekryptografisk system. De to computere var dog placeret under en meter fra hinanden. Man har siden forsøgt at få systemet til at virke over længere og mere brugbare afstande. I 1995 lykkedes det et hold forskere at få to computere til at kommunikere kvantekryptografisk gennem en 23 km lang lysleder. Andre forskere har formået at gøre dette på en afstand af 1 km gennem luft. Problemet med transmission gennem luft er, at fotonerne vekselvirker med luftens molekyler, og dermed kan deres polarisation blive ændret. Det lader dog alligevel til, at kodemagerne er kommet længere med sit kvantekryptografiske system, end kodebryderne er kommet med sine kvantecomputere. Så måske vil kvantekryptografien nå at erstatte RSA-systemet, inden kvantecomputerne når at blive en reel trussel mod koderne.

[16] *Kodebogen*, s. 360

6. Konklusion

Den modulære aritmetiks nye begreber og definitioner er blevet gennemgået, forklaret og diskuteret, så de har kunnet fungere som baggrund for gennemgangen af to kendte sætninger, der kan bruges som primtalstest. Disse to sætninger er tillige blevet bevist, og fordele og ulemper ved brugen af dem som primtalstest er også blevet diskuteret. Begge sætninger er mere end 200 år gamle, og afsnittet afsluttes derfor med en kort beskrivelse af en ny sætning, som er fra år 2002. Alle tre sætninger kan danne grundlag for test, der kan bruges, når de essentielle primtal skal findes til RSA-krypteringssystemet. Det er blevet vist, hvorfor dette system i praksis er ubrydeligt, hvordan det fungerer og hvilke matematiske krav, der stilles til brugen af systemet. Desuden må det konkluderes, at systemet kun er ubrydeligt, hvis N'et i den offentlige nøgle er produktet af to enorme primtal.

Da RSA-systemet er bygget op omkring det faktum, at det i dag er meget tidskrævende at faktorisere enorme tal, kan det bryde sammen, hvis man en dag får lavet tilstrækkeligt hurtige computere. Et bud på en sådan computer kan være kvantecomputeren, hvorfor idéen bag denne er blevet gennemgået ved hjælp af en redegørelse af de dele af kvanteteorien, der tillader en sådan computer. Kvantemekanikkens verden har dog også inspireret kodemagerne, som er i gang med at udvikle et kvantekryptografisk system. Dette system er tillige blevet forklaret, analyseret og vurderet til at være absolut ubrydeligt, da systemet tillader to parter at blive enige om en fælles tilfældig og hemmelig engangsblok over afstand, og endda lader de to parter opdage en eventuelt snagende tredjepart. Det er tidligere blevet vist, hvorfor lige netop engangsblokke er absolut ubrydelige, og dermed bliver også det kvantekryptografiske system absolut ubrydeligt.

Fordelen ved dette frem for RSA-systemets ubrydelighed i praksis er, at det ikke er umuligt at fremtiden vil bringe nye metoder til hurtig primtalsfaktorisering og løsning af envejsfunktioner, og allerede nu truer kvantecomputeren i kulissen med at bryde de krypteringssystemer, vi alle omgiver os med og benytter dagligt. Det vurderes dog sluttelig, at det kvantekryptografiske system formodentlig vil være klar til at afløse RSA-systemet inden kvantecomputeren for alvor bliver en trussel mod nutidens krypteringssystemer.

Kildeoversigt

Aigner, Martin: *Diskrete Matematik*, Vieweg, Wiesbaden 1999

Carstensen, Jens: *Talteori*, Forlaget Systime, Herning 1993

Hansen, Johan P.: *Primtalsfaktorisering – nogle nye resultater og anvendelser*, http://home.imf.au.dk/matjph/haderslev.pdf, udskrift 10.12.2008

Hansen, Johan P. og Henrik G. Spalk: *Algebra og talteori*, Nordisk Forlag, København 2002

Matthiesen, Jens E. (red.): "Drømmen om den sikre kode er gået i opfyldelse", *Illustreret Videnskab*, nr. 12/2002 s. 22, Mariehamn 2002

Singh, Simon: *Kodebogen. Videnskaben om hemmelige budskaber fra oldtidens Ægypten til kvantekryptering*, (*The Code Book. The Science of Secrecy from Ancient Egypt to Quantum Cryptography*, 1999), ovs. Jan Teuber, Nordisk Forlag, København 2001

Wells, David: *Primtal – Matematikkens gådefulde tal fra A-Ø* (*Prime Numbers – The Most Mysterious Figures in Math*, 2005), ovs. Poul G. Hiorth, Nyt Teknisk Forlag, København 2006