Algebraen og talteorien bag offentlig nøgle kryptering og signering.

Noter til kursus Fuglsø, okt. 2000
Revideret uddrag af udkast til bog i Gyldendals Aspekt serie
(Kan hentes som pdf-fil på www.imf.au.dk\ matjph)

Johan P. Hansen
e-mail: matjph@imf.au.dk
Matematisk Institut
Århus Univsersitet

Indhold

Indledning

Noternes hovedtema er *matematikken bag offentlig nøgle kryptosystemer*, hvor den bagvedliggende matematik er klassisk, men har fået særdeles aktuel betydning i forbindelse med kryptering og digital signatur (elektronisk underskrift), hvorom der netop er lovgivet i Danmark.

Gauss indførte begrebet *kongruens*, det er en meget simpel ide, hvis vigtighed og nyttighed i talteori imidlertid ikke kan overvurderes. Kapitel 1 til 5 behandler de grundliggende egenskaber ved kongruenser og modulær aritmetik og udgør grundlaget for behandlingen af hovedtemaet.

KAPITEL 1

Største fælles divisor

Lad m og n være hele tal, som bekendt siger vi at m *går op i* n, hvis der findes et helt tal k, så $n = km$. Vi siger også, at m er en *divisor* i n, eller n er et *(helt) multiplum* af m. Vi bruger skrivemåden

$$m|n.$$

For to positive hele tal vil vi konkret bestemme det største hele tal, der er en divisor i begge tal.

DEFINITION 1. Lad a, b være positive hele tal. Det største hele tal, der er en divisor i såvel a som b, kaldes *største fælles divisior* af a og b, hvilket vi skriver

$$\mathrm{sfd(a, b)}.$$

Det er let direkte at se, at $\mathrm{sfd}(30, 21) = 3$ og at $\mathrm{sfd}(625, 81) = 3$. For store tal er det svært umiddelbart at finde den største fælles divisor. *Euklids algoritme* giver imidlertid en overordentlig effektiv metode til at bestemme $\mathrm{sfd(a, b)}$ for store tal a, b.

Euklids algoritme

Afvikling af eksempler.

EKSEMPEL 2. Lad os først belyse algoritmen ved at gennemgå, at $\mathrm{sfd}(30, 21) = 3$. Vi starter med at forsøge at dividere 30 med 21, divisionen går ikke op, vi får resten 9:

$$30 \;=\; 1 \;\cdot\; 21 \;+\; 9 \tag{0.1}$$

Dernæst dividerer vi 21 (det tal vi dividerede med i (0.1)) med 9 (den rest vi fandt i (0.1)):

$$21 \;=\; 2 \;\cdot\; 9 \;+\; 3 \tag{0.2}$$

Endelig dividerer vi 9 (det tal vi dividerede med i (0.2)) med 3 (den rest vi fandt i (0.2)):

$$9 \;=\; 3 \;\cdot\; 3 \;+\; 0$$

og ser, at divisionen går op. Euklids algoritme siger nu, at $\mathrm{sfd}(30, 21) = 3$. Vi kan sætte divisionerne op i et skema, der er gengivet i tabel 1.

$$\begin{aligned} 30 &= 1 \cdot 21 + 9 \\ 21 &= 2 \cdot 9 + \mathbf{3} \\ 9 &= 3 \cdot 3 + 0 \end{aligned}$$

TABEL 1. Euklids algoritme afviklet trinvis til at bestemme, at $\mathrm{sfd}(30, 21) = 3$.

$$\begin{aligned} 123456789 &= 15 \cdot 23456789 + 6172844 \\ 23456789 &= 3 \cdot 6172844 + 4938257 \\ 6172844 &= 1 \cdot 4938257 + 1234587 \\ 4938257 &= 3 \cdot 1234587 + 123496 \\ 1234587 &= 1 \cdot 123496 + 91 \\ 123496 &= 13565 \cdot 91 + 81 \\ 91 &= 1 \cdot 81 + 10 \\ 81 &= 8 \cdot 10 + \mathbf{1} \\ 10 &= 10 \cdot 1 + 0 \end{aligned}$$

TABEL 2. Euklids algoritme afviklet trinvis til at bestemme, at $\mathrm{sfd}(123456789, 23456789) = 1$.

Vi gennemløber endnu engang algoritmen - det skulle gerne overbevise om, at den er effektiv også for store tal.

EKSEMPEL 3. Vi gennemløber Euklids algoritme for $a = 123456789$ og $b = 23456789$ ved først at dividere a med b og derpå succesivt at dividere med resten (opnået i den forudgående division) i det tal man dividerede med (i den forudgående division). På skemaform er beregningerne opstillet i tabel 2. Vi noterer, at resterne bliver mindre og mindre for på et tidspunkt at give 0 og divisionen op. Euklids algoritme siger nu, at

$$\mathrm{sfd}(123456789, 23456789) = 1,$$

den sidste rest forskellig fra 0.

ØVELSE 4. Brug Euklids algoritme til af bestemme $\mathrm{sfd}(2345, 6789)$ og $\mathrm{sfd}(2435, 9786)$.

Analyse af Euklids algoritme. Euklids algoritme gennemføres for a og b ved først at dividere a med b og derpå succesivt at dividere med resten (opnået i den forudgående division) i det tal man dividerede

med (i den forudgående division):

$$\begin{array}{lcllcl} a & = & k_1 & \cdot\ b & + & r_1 \\ b & = & k_2 & \cdot\ r_1 & + & r_2 \\ r_1 & = & k_3 & \cdot\ r_2 & + & r_3 \\ & & & \vdots & & \\ r_{n-3} & = & k_{n-1} & \cdot\ r_{n-2} & + & r_{n-1} \\ r_{n-2} & = & k_n & \cdot\ r_{n-1} & + & \mathbf{r_n} \\ r_{n-1} & = & k_{n+1} & \cdot\ r_n & + & 0 \end{array} \tag{0.3}$$

Resterne bliver løbende mindre for på et tidspunkt at give 0 (divisionen går op). Euklids algoritme siger nu, at $\mathrm{sfd(a,b)} = \mathrm{r_n}$, den sidste rest forskellig fra 0.

Af sidste linie i (0.3) fremgår, at $r_n|r_{n-1}$, af næstsidste line slutter vi dernæst, at $r_n|r_{n-2}$. Fortsætter vi således op igennem linierne får vi, at $r_n|b$ og til sidst, at $r_n|a$, vi konkluderer, at r_n er en fælles divisor i a, b.

Hvorfor er r_n den største fælles divisor? Lad d være en anden divisor i a og b. Af første line i (0.3) følger, at $d|r_1$ og ved hjælp af næste line slutter vi, at $d|r_2$. Fortsætter vi ned igennem linierne får vi til sidst, at $d|r_n$

Vi har nu bevist, at Euklids algoritme faktisk beregner den største fælles divisor. Da resterne bliver mindre og mindre for hvert trin må processen også stoppe efter et endeligt antal trin. Vi samler det op i følgende formulering.

METODE 5. ***(Euklids algoritme)*** *Den største fælles divisor af to positive hele tal a og b bestemmes i endelig mange trin således:*

start: *sæt $r_{-1} = a$ og $r_0 = b$*

trin: *beregn trin for trin kvotienter k_i og rester r_i med $0 \leq r_{i+1} < r_i$*

$$r_{i-1} = k_{i+1} \cdot r_i + r_{i+1}, \quad i = 0, 1, 2, \ldots, n \tag{0.4}$$

slut: *indtil $r_{n+1} = 0$*

så er r_n, den største fælles divisor.

Bezouts identitet

Euklids algoritme kan bruges til på simpel at udtrykke $\mathrm{sfd}(a, b)$ ved a og b.

SÆTNING 6. *Der findes hele tal x og y, så*

$$\mathrm{sfd}(a, b) = ax + by.$$

Tallene x, y er ikke entydigt bestemte, altså der findes flere talsæt x, y, der løser problemet. Der findes en metode, der udnytter Euklids algoritme til at bestemme et talsæt x, y.

EKSEMPEL 7. Først vender vi tilbage til Eksempel 2, hvor Euklids algoritme til bestemmelse af $\text{sfd}(30, 21) = \mathbf{3}$ på skemaform (jævnfør tabel 1) er

$$\begin{array}{rcrcrcr} 30 & = & 1 & \cdot & 21 & + & 9 \\ 21 & = & 2 & \cdot & 9 & + & \mathbf{3} \\ 9 & = & 3 & \cdot & 3 & + & 0 \end{array}$$

Ligninger omskrives til

$$\begin{array}{rcrcrcr} 9 & = & 30 & & -1 & \cdot & 21 \\ \mathbf{3} & = & 21 & - & 2 & \cdot & 9 \end{array}$$

Indsættes udtrykket for 9 fra den forudgående ligning i den sidste ligning fås:

$$\mathbf{3} = 21 - 2 \cdot (30 - 1 \cdot 21) = 30 \cdot (-2) + 21 \cdot 3$$

og vi har den ønskede Bezout identitet med $x = -2$ og $y = 3$

ØVELSE 8. Opstil ved hjælp af Euklids algoritme Bezout identiteter for tilfældene $\text{sfd}(2345, 6789)$ og $\text{sfd}(2435, 9786)$.

Vi opsummerer.

METODE 9. ***(Bezout identitet.)*** *Givet a og b positive hele tal, bestemmelse af et sæt hele tal x, y, så*

$$\text{sfd}(a, b) = ax + by.$$

Næstsidste trin i Euklid's algoritme (0.3) udtrykker $\text{sfd}(a, b)$ *ved de to forudgående rester:*

$$\text{sfd}(a, b) = r_n = r_{n-2} - k_n \cdot r_{n-1} \tag{0.5}$$

Ligning i det forudgående trin i Euklid's algoritme omskrives til:

$$r_{n-1} = r_{n-3} - k_{n-1} \cdot r_{n-2}$$

og anvendes til at eliminere r_{n-1} i (0.5) og vi får

$$\text{sfd}(a, b) = r_{n-2} - k_n \cdot (r_{n-3} - k_{n-1} \cdot r_{n-2}) = -k_n \cdot r_{n-3} + (1 + k_n \cdot k_{n-1}) \cdot r_{n-2}$$

Arbejder vi os således gradvis baglæns igennem ligningerne (0.3) opnås til sidst den ønskede Bezout identitet.

KAPITEL 2

Primtal

Et helt tal $p > 1$ kaldes et *primtal*, hvis de eneste positive divisorer i tallet er de *trivielle divisorer*, nemlig 1 og tallet p selv. Tallet 6 er ikke noget primtal, det har jo de to ikke-trivielle divisorer 2 og 3. Tallene

$$2, 3, 5, 7, 11, 13, 17, 19$$

er derimod primtal. Bemærk lige at tallet 1 ikke er noget primtal.

Et primtal p, der er divisor i $m > 1$ kaldes en *primdivisor* i m.

LEMMA 10. *Ethvert helt tal $m > 1$ har (mindst) en primdivisor p.*

BEVIS. Antag at der fandtes hele tal $m > 1$ uden primdivisorer. Lad m_0 være det mindste af disse. Tallet m_0 kan ikke være et primtal p - så ville det jo have primdivisoren p. Derfor har tallet m_0 en ikke-triviel divisor d, $d \neq 1, d \neq m$. Tallet d er skarpt mindre end m_0 og har derfor en primdivisor p. (Husk på, at m_0 var valgt som det mindste hele tal uden primdivisorer, hvorfor alle tal d med $1 < d < m_0$ har en primdivisor). Primtallet p er en divisor i d, der igen er en divisor i m_0, altså har m_0 en primdivisor. Vores oprindelige antagelse medfører en absurditet - nemlig at m_0 såvel har som ikke har en primdivisor - den er derfor forkert. □

Et primtal har den egenskab, at er den divisor i et produkt, er den divisor i en af faktorerne.

LEMMA 11. *Lad p være et primtal og lad a, b være hele tal. Der gælder, at*

$$p|ab \Rightarrow p|a \vee p|b.$$

BEVIS. Hvis p er en divisor i a, er der intet at vise. Vi kan altså uden videre antage, at p ikke er divisor i a. Så har a og p største fælles divisor lig med 1 og ifølge Sætning 6 (Bezouts identitet) findes der hele tal x og y, så

$$1 = ax + py.$$

Efter multiplikation med b fås

$$b = bax + bpy,$$

hvis højreside klart er divisibel med p. Derfor er b også divisibel med p, hvilket skulle vises. □

Det første hovedresultat i dette kapitel er *aritmetikkens fundamentalsætning*, der siger at ethvert helt tal $n > 1$ kan skrives som et produkt af primtal på en i al væsentlighed entydig måde. Det andet hovedresultat er *Euklids sætning*, der siger at der er uendelig mange primtal.

Aritmetikkens fundamentalsætning

SÆTNING 12. ***(Aritmetikkens fundamentalsætning)*** *Ethvert helt tal $m > 1$ kan skrives som et produkt*

$$m = p_1^{n_1} \cdot p_2^{n_2} \cdot \cdots \cdot p_k^{n_k},$$

hvor $p_i, i = 1, \ldots, k$ er forskellige primtal.

Faktorerne $p_1^{n_1}, p_2^{n_2}, \ldots, p_k^{n_k}$ er entydigt bestemte.

BEVIS. Antag at der findes mindst et helt tal $m > 1$ uden primtalsfaktorisering. Vi vil nu ræsonnere os frem til en absurditet.

Lad m_0 være det mindste tal (større end 1) uden primtalsfaktorisering. Ifølge Lemma 10 har m_0 en primdivisor p og $m_0 = p * k$. Tallet k er ikke 1, hvis det nemlig var tilfældet ville m_0 være et primtal og specielt have en faktorisering i et produkt af primtal. Tallet k er på den anden side mindre end m_0, derfor har det en primtalsfaktorisering, hvilket medfører at også $m_0 = p * k$ har det. Det strider imidlertid mod antagelsen om m_0.

For at vise entydigheden antager vi modsætningsvist, at der findes hele tal med forskellige primtalsfaktoriseringer. Lad m_0 være det mindste sådanne tal (større end 1):

$$m_0 = p_1^{a_1} \cdot p_2^{a_2} \cdot \cdots \cdot p_k^{n_k} = q_1^{b_1} \cdot q_2^{b_2} \cdot \cdots \cdot q_l^{b_l},$$

hvor primtallene er nummererede således at $p_1 < p_2 < \cdots < p_k$ og $q_1 < q_2 < \cdots < q_l$. Ved hjælp af Sætning 11 fås, at p_1 er divisor i q_j og dermed er $p_1 = q_j$ for et j. Tilsvarende er $q_1 = p_i$ for et i. Sammenfattende er $p_1 \leq p_i = q_1 \leq q_j = p_1$, hvorfor $p_1 = q_1$. Vi har derfor de to forskellige faktoriseringer:

$$p_1^{a_1 - 1} \cdot p_2^{a_2} \cdot \cdots \cdot p_k^{n_k} = q_1^{b_1 - 1} \cdot q_2^{b_2} \cdot \cdots \cdot q_l^{b_l},$$

af et tal, der er skrapt mindre end m_0. Det strider mod valget af m_0, som det mindste positive tal med to forskellige primtalsfaktoriseringer. □

Primtalsfaktoriseringens kompleksitet. Beviset for eksistensen af primtalsfaktoriseringen af et helt tal er ikke konstruktivt, det giver ikke nogen metode til at bestemme en primtalsfaktorisering. Det er (antageligt) et meget vanskeligt problem at faktorisere et helt tal i et produkt af primtal indenfor rimelig tid.

For at få en forståelse af problemets kompleksitet forestil dig, at m er et helt tal med 100 cifre, dvs. $m \sim 10^{100}$. Den nærliggende metode til at bestemme en primfaktor ved at prøve sig frem fra neden af, der virker ganske overbevisende for små tal, bliver uhyre tidskrævende for store tal. Lad os for eksempel antage, at du kan afgøre om et helt tal er divisor i m i løbet af 1 million'te del af et sekund (10^{-6} sek.). Ialt vil dine overvejelser af de ialt 10^{100} tal tage:

$$10^{100} \cdot 10^{-6}\text{sek.} = 10^{94}\frac{1}{60 \cdot 60 \cdot 24 \cdot 365} \quad \text{år} \sim 3,2 \cdot 10^{86} \quad \text{år.}$$

Det er mange gange mere end den anslåede alder af universet. Den nærliggende metode til bestemmelse af en primfaktor er på grund af sit kæmpe tidsforbrug ubrugelig til store tal.

Det forhold udnyttes til at lave ubrydelige koder til kryptering og digital signatur - i Kapitel 8 vil du se, hvordan det gøres.

Euklids sætning

Allerede Euklid viste, at listen over primtal

$$2, 3, 5, 7, 11, 13, 17, 19, \ldots$$

faktisk fortsætter i det uendelige.

SÆTNING 13. ***(Euklids sætning)*** *Der er uendelig mange primtal.*

BEVIS. Antag at der kun er endelig mange primtal, dem kalder vi $p_1, p_2, \ldots, p_k$. Betragt nu tallet

$$m = p_1 \cdot p_2 \cdot \cdots \cdot p_k + 1. \tag{0.6}$$

Ifølge Lemma 10 har m en primdivisor p. Fra antagelsen følger, at primtallet p må være et af tallene $p_1, p_2, \ldots, p_k$. Tallet $p > 1$ er derfor en divisor i både m og $p_1 \cdot p_2 \cdot \cdots \cdot p_k$ og dermed i tallet 1, jvf. (0.6). Det kan ikke passe - antagelsen om at der kun er endelig mange primtal må altså være forkert. □

KAPITEL 3

Kongruenser

I dette kapitel vil vi behandle modulær aritmetik. For et givet helt tal n vil vi erstatte ethvert andet helt tal med dets rest ved division med n. Da det kun er endelig mange muligheder for rester $(0, 1, 2, \ldots, n-1)$ betyder det, at vi erstatter de hele tal $\mathbb{Z}$ med et talsystem $\mathbb{Z}_n$ som kun indeholder n elementer. Vi kan regne (lægge sammen, trække fra og gange) i $\mathbb{Z}_n$ ligesom i $\mathbb{Z}$.

Restklasser

Før vi går i gang med den generelle teori, vil vi se på et eksempel, der gerne skulle afsløre styrken i at regne med rester.

Eksempel 14. Hvilken ugedag er det om 1000 dage? Det kunne du let svare på ved slå op og tælle fremad i en kalender, der rækker 3 år frem i tiden. Der er en meget simplere metode, når vi husker på, at ugedagene gentages hver syvende dag. Dividerer vi 1000 med 7 får vi:

$$1000 = 142 \cdot 7 + 6.$$

De 1000 dage rummer altså 142 hele uger og en rest på 6 dage. Ugedagen om 1000 dage er altså den samme som den om 6 dage eller den samme som om -1 dag, altså igår.

Gauss indførte begrebet kongruenser, der systematiserer regning med rester, i sin bog *Disquisitiones arithmeticae* [G1] fra 1801 om højere aritmetik.

Definition 15. Lad n være et helt tal. Vi siger at de hele tal a, b er *kongruente modulo* n, hvis de har samme rest ved division med n, altså hvis

$$n | a - b.$$

Vi skriver

$$a \equiv b \mod n.$$

For eksempel er $1000 \equiv 6 \mod 7$ og $22 \equiv -2 \mod 4$.

Lad n være valgt. *Restklassen* eller *kongruensklassen* $[a]$ til a består af alle de hele tal b, der er kongruente til a modulo n, altså af alle tal

b, der har samme rest som a ved division med n:

$$[a] = \{b | a \equiv b \mod n\} = \{\ldots, a-2n, a-n, a, a+n, a+2n, \ldots\}.$$

Vi noterer

$$[a] = [b] \Leftrightarrow a \equiv b \mod n$$

Der er ialt netop n forskellige kongruensklasser modulo n:

$$\begin{aligned} [0] &= \{\ldots, 0, n, 2n, \ldots\} \\ [1] &= \{\ldots, 1, 1+n, 1+2n, \ldots\} \\ [2] &= \{\ldots, 2, 2+n, 2+2n, \ldots\} \\ &\vdots \\ [n-1] &= \{\ldots, n-1, (n-1)+n, (n-1)+2n, \ldots\} \end{aligned}$$

En mængde af n hele tal kaldes *et komplet sæt af rester*, hvis der er præcis et tal fra hver af de n kongruensklasser.

Modulær addition og multiplikation

DEFINITION 16. De n kongruensklasser $[0], [1], \ldots, [n-1]$ udgør mængden $\mathbb{Z}_n$. Addition og multiplikation af kongruensklasser defineres således:

$$\begin{aligned} [a] + [b] &= [a+b] \\ [a] \cdot [b] &= [a \cdot b]. \end{aligned}$$

At ovenstående giver en veldefineret addition og multiplikation af restklasser forudsætter at resultaterne er uafhængige af valg af repræsentanter for restklasserne. Det beror på, at

$$a_1 \equiv a_2 \mod n \wedge b_1 \equiv b_2 \mod n \quad \Rightarrow \begin{cases} a_1 + b_1 & \equiv a_2 + b_2 \mod n \\ a_1 \cdot b_1 & \equiv a_2 \cdot b_2 \mod n \end{cases}$$

EKSEMPEL 17. I tilfældet $n = 6$ ser additions- og multiplicationstabellen i $\mathbb{Z}_6$ ud som angivet i tabel 1 (udtrykt ved det komplette sæt af rester $0, 1, 2, 3, 4, 5$). Ser man nærmer på tabellen vil man opdage en særhed, nemlig at to elementer forskellige fra 0 godt kan have produkt 0.

EKSEMPEL 18. Lad os beregne den mindste positive rest af 3^8 mod 13.

Første regner vi i $\mathbb{Z}_{13}$ og får

$$[3^2] = [9] = [-4]$$

Det udnytter vi dernæst til at bestemme,at

$$[3^4] = [3^2 \cdot 3^2] = [3^2] \cdot [3^2] = [-4] \cdot [-4] = [16] = [3]$$

+	0	1	2	3	4	5
0	0	1	2	3	4	5
1	1	2	3	4	5	0
2	2	3	4	5	0	1
3	3	4	5	0	1	2
4	4	5	0	1	2	3
5	5	0	1	2	3	4

·	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

TABEL 1. Additions- og multiplicationstabellen i $\mathbb{Z}_6$ (udtrykt ved det komplette sæt af rester $0, 1, 2, 3, 4, 5$)

og endelig at

$$[3^8] = [3^4 \cdot 3^4] = [3^4] \cdot [3^4] = [3] \cdot [3] = [9].$$

Det søgte resultat er altså 9.

Vi kunne også have formuleret de samme beregningerne således:

$$3^2 = 9 \equiv -4 \mod 13,$$

hvorfor

$$3^4 = 3^2 \cdot 3^2 \equiv (-4) \cdot (-4) = 16 \equiv 3 \mod 13$$

og

$$3^8 = 3^4 \cdot 3^4 \equiv 3 \cdot 3 = 9 \mod 13$$

ØVELSE 19. Bestem den mindste positive rest af

a) $5^{10} \mod 23$.

b) $19 \cdot 14 \mod 23$

c) $5^{10} + 19 \cdot 14 \mod 23$

EKSEMPEL 20. Lad os bruge modulær addition og multiplikation til at se, at polynomiet $f(x) = x^5 - x^2 + x - 3$ ikke har rødder blandt de hele tal. Vælg $n = 4$ og betragt kongruensen

$$f(x) = x^5 - x^2 + x - 3 \equiv 0 \mod 4$$

Der er 4 kongruensklasser modulo 4 med for eksempel $-1, 0, 1, 2$ som et fuldstændigt sæt af rester. Vi får, at

$$\begin{aligned} f(-1) &= -6 \equiv 2 \mod 4 \\ f(0) &= -3 \equiv 1 \mod 4 \\ f(1) &= -2 \equiv 2 \mod 4 \\ f(2) &= 27 \equiv 3 \mod 4 \end{aligned}$$

Kongruensen $f(x) \equiv 0 \mod 4$ har åbenbart ingen løsninger, hvorfor polynomiet $f(x)$ ikke har noget helt tal som rod.

ØVELSE 21. Vis, at ingen af følgende polynomier har hele tal som rødder.

a) $x^3 - x + 1$.
b) $x^3 + x^2 - x + 1$
c) $x^3 + x^2 - x + 3$

Modulær division - løsning af lineære kongruenser.

Vi har netop set, at vi kan addere og multiplicere elementerne i $\mathbb{Z}_n$ med hinanden.

Division af kongruensklassen $[b]$ med kongruensklassen $[a]$ er mere kompliceret. Lad os se på Eksempel 17, hvor vi behandlede $\mathbb{Z}_6$. Multiplicationstabellen var:

$\cdot$	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	1	2	3	4	5
2	0	2	4	0	2	4
3	0	3	0	3	0	3
4	0	4	2	0	4	2
5	0	5	4	3	2	1

Vi noterer ved at se på rækken med 2, at der faktisk er to klasser, nemlig [1] og [4], der multipliceret med [2] giver klassen [2]. På den anden side er der ingen klasser der multipliceret med [2] giver klassen [3].

Ser vi derimod på rækken med 5, er det klart at der til enhver klasse [b] findes netop en klasse [x] så $[5] \cdot [x] = [b]$.

PROBLEM 22. *Lad n være fast. For a (ikke er et multiplum af n) og b ønskes bestemt de klasser $[x]$, så*

$$[a] \cdot [x] = [b] \quad i \quad \mathbb{Z}_n.$$

Det svarer til at bestemme de hele tal x, så

$$ax \equiv b \mod n.$$

SÆTNING 23. *Lad a, b, n være hele tal, hvorom det antages, at a ikke er et multiplum af n. Lad $d = \mathrm{sfd(a, n)}$. Om løsninger til kongruensen*

$$[a] \cdot [x] = [b] \quad i \quad \mathbb{Z}_n.$$

$$(ax \equiv b \mod n.)$$

gælder

- *hvis d ikke er divisor i b, så er der ingen løsninger*
- *hvis d er divisor i b, så er der netop d kongruensklasser $[x]$, der løser problemet.*
- *specielt at hvis $d = 1$, så er der præcis 1 kongruensklasse, der løser problemet.*

METODE 24. *Lad a, b, n være hele tal. Lad $d = \mathrm{sfd(a, n)}$. Der findes (mindst) et x, så*

$$ax \equiv b \mod n,$$

hvis og kun hvis d er en divisor i b.

Hvis d er divisor i b bestemmes en løsning $x_0 = u\frac{b}{d}$, idet der først bestemmes løsninger u, v til Bezout identiteten

$$au + nv = d$$

ved hjælp af Euklids algoritme og Bezouts identitet, jvf. Sætning 5 og Sætning 9. Enhver anden løsning er på formen

$$x = x_0 + \frac{n}{d}t, \quad t \in \mathbb{Z}.$$

BEVIS. □

ØVELSE 25. Bestem de klasser $[x]$, så

a) $[10] \cdot [x] = [3]$ *i* $\mathbb{Z}_{12}$,
b) $[10] \cdot [x] = [6]$ *i* $\mathbb{Z}_{12}$,
c) $[12] \cdot [x] = [9]$ *i* $\mathbb{Z}_{15}$.

KOROLLAR 26. ***(Forkortning.)*** *Lad a, n være indbyrdes primiske hele tal. I $\mathbb{Z}_n$ gælder, at*

$$[a] \cdot [x_1] = [a] \cdot [x_2] \Leftrightarrow [x_1] = [x_2]$$

Vi kan altså forkorte med $[a]$ i $\mathbb{Z}_n$. Anderledes formuleret

$$a \cdot x_1 \equiv a \cdot x_2 \mod n \Leftrightarrow x_1 \equiv x_2 \mod n$$

KAPITEL 4

Fermats lille sætning

SÆTNING 27. *Lad p være et primtal. For ethvert helt tal a, som ikke er et multiplum af p, gælder at*

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

eller udtrykt ved restklasser modulo p

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

BEVIS. Et komplet sæt af rester forskellige fra 0 udgøres af

$$1, 2, \ldots, p-1 \tag{0.7}$$

Multiplicerer vi elementerne i (0.7) med a fås et andet komplet sæt af rester forskellige fra 0, jvf. Korollar 26:

$$a \cdot 1, a \cdot 2, \ldots, a \cdot (p-1). \tag{0.8}$$

Rækkefølgen kan være anderledes; men begge lister må indeholde samtlige $p-1$ ikke-nul restklasser. Derfor er produktet af elementerne i (0.7) det samme som produktet af elementerne i (0.8) modulo p:

$$1 \cdot 2 \cdot \cdots \cdot (p-1) \equiv (a \cdot 1) \cdot (a \cdot 2) \cdot \cdots \cdot (a \cdot (p-1)) \mod p \quad \Rightarrow$$

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

Anvender vi endnu engang Korollar 26 til at forkorte $(p-1)!$ væk fås, at

$$1 \equiv a^{p-1} \mod p.$$

□

KOROLLAR 28. *Lad p være et primtal. For alle a gælder at*

$$a^p \equiv a \mod p$$

eller sagt på en anden måde

$$[a^p] = [a] \quad i \quad \mathbb{Z}_p$$

EKSEMPEL 29. Lad os bestemme den mindste positive rest af 2^{68} mod 19. Da 19 er et primtal sikrer Fermats lille sætning, at $a^{18} = 1$ mod 19. Derfor er

$$2^{68} = 2^{18 \cdot 3 + 14} = (2^{18})^3 \cdot 2^{14} \equiv 2^{14} \mod 19.$$

Videre fås, idet $16 \equiv -3 \mod 19$, at

$$2^{14} = 2^{4\cdot 3+2} = (2^4)^3 \cdot 2^2 = 16^3 \cdot 2^2 \equiv$$

$$(-3)^3 \cdot 2^2 \equiv (-27) \cdot 2^2 \equiv 11 \cdot 2^2 \equiv 44 \equiv 6 \mod 19.$$

ØVELSE 30. Find den mindste positive rest af $3^{91} \mod 23$.

Primtalstest

Fermats lille sætning er grundlaget i en *primtalstest* - en nyttigt metode til at afsløre, at et tal n ikke er et primtal. Findes der nemlig et tal a for hvilket ligningen

$$a^n \equiv a \mod n$$

ikke holder, må konklusion være, at n ikke kan være et primtal. Testen giver dog ikke nogen faktorisering af n, omend den måtte have afsløret, at n ikke er et primtal.

Selvom n klarer testen for alle a, er der imidlertid ikke sikkerhed for, at n faktisk er et primtal.

Testen var kendt af kineserne og for 25 århundreder siden påstod de, at tal, der overlever testen for $a = 2$, faktisk er primtal. Det var dog først i 1819, at man fandt tal, der tilfredsstiller

$$2^n \equiv 2 \mod n$$

uden at være primtal. Sådanne tal kaldes *pseudo-primtal*. I forhold til testen virker de som primtal, men er sammensatte tal.

Testen er praktisk gennemførlig, idet vi senere i kapitel 6 skal lære en hurtig metode til i modulær aritmetik at beregne $a^n \mod n$.

EKSEMPEL 31. **(Pseudo-primtal)** Vi vil se, at 341 er et pseudo-primtal. Først noterer vi, at

$$2^{10} = 1024 \equiv 1 \mod 341,$$

hvorfor

$$2^{341} = (2^{10})^{34} \cdot 2 \equiv 2 \mod 341$$

og 341 har klaret testen for $a = 2$.

ØVELSE 32. Vis, at 341 ikke klarer testen for $a = 3$. Vent eventuelt til efter, at du har lært den hurtige metode til modulær eksponentering i kapitel 6.

En anvendelse af Fermats lille sætning i kryptografi

Kryptering anvendes til at sende meddelelser hemmeligt og sikkert.

Mange krypteringsmetoder beror på talteori, en meget simpel metode er at kryptere ved at erstatte et bogstav med dets efterfølger i alfabetet. Matematisk udtrykt erstatter vi alfabetet med hele tal: $A = 0, B = 1, C = 2, \ldots, Å = 28$ og krypterer nu ved at addere 1 mod 29. Tilsvarende koder kan laves ved at addere et andet tal k (dette tal kaldes nøglen), altså bruge afbildningen

$$x \mapsto x + k$$

Den romerske kejser Julius Cæsar anvendte $k = 3$. Dekrypteringen af en meddelelse gøres simpelthen ved at subtraere k mod 29. Disse koder er meget lette at bryde. Prøv blot med alle mulige værdier for k, indtil du får en læselig meddelelse.

Øvelse 33. Et navn gemmer sig bag meddelelse QBÅVS, der er krypteret efter metoden ovenfor. Bestem navnet og værdien af k.

Bedre koder kan bygges på Fermats lille sætning . Vælg først et stort primtal p og et helt tal e, der er primisk med $p - 1$. Disse to tal er nøglen . Krypteringen foregår ved at anvende afbildningen

$$x \mapsto x^e \mod p$$

Dekrypteringen foregår ved først at bestemme f, g, så

$$ef + (p-1)g = 1,$$

hvilket kan lade sig gøre ifølge Sætning 9, da e og $p - 1$ er indbyrdes primiske. Dernæst beregner modtageren under anvendelse af Fermats lille sætning

$$(x^e)^f = x^{ef} = x^{1-(p-1)g} = x(x^{(p-1)})^{-g} = x$$

og har dermed rekonstrueret meddelelsen x.

Eksempel 34. Lad $p = 29$ og $e = 5$, som er primisk med $p-1 = 28$. Bogstaverne i navnet PETER omsættes direkte til tallene 15.5.20.5.18 som krypteres til talrækken 23.22.24.22.15 ved afbildningen $x \mapsto x^5$ mod 29.

Dekryptering indebærer at bestemme f - her kan $f = 17$ bruges - og foregår ved at anvende afbildningen $x \mapsto x^{17}$ mod 29.

Koder bygget op på denne måde synes meget sikre. Antag at en lytter har fået kendskab til p, en meddelelse x og den tilhørende krypterede version $y = x^e$ mod p af meddelelsen. For at bryde koden skal

lytteren bestemme e (eller f). Der er imidlertid ingen hurtige metoder til at bestemme e, der tilfredsstiller ligningen

$$y = x^e \mod p,$$

hvilket er metodens styrke. Dette problem kaldes *det diskrete logaritme problem.*

Metodens svaghed er, at sender og modtager forlods i dybeste hemmelighed skal udveksle nøglen, altså tallene p, e.

I kapitel 8 skal vi se på såkaldt *offentlige nøgle kryptering*, der beror på en generalisering af Fermats lille sætning og tillader udveksling af nøglen i fuld offentlighed.

ØVELSE 35. Bestem et e, så $27 \equiv 10^e \mod 29$.

KAPITEL 5

Eulers sætning

Eulers funktion

For et helt tal n definerer vi Eulers funktion $\phi(n)$ som antallet af hele tal $a = 1, 2, \ldots, n$, der er primisk med n, altså som har 1 som største fælles divisor med n. Funktionen ϕ kaldes *Eulers ϕ-funktion*. For $n = 12$ er de 4 tal $1, 5, 7, 11$ primiske med 12, hvorimod $2, 3, 4, 6, 8, 9, 10$ ikke er det, derfor er $\phi(12) = 4$.

ØVELSE 36. Overvej, at tabel 1 er korrekt.

Systematisk undersøgelse af alle tal $1, \ldots, m-1$ for at bestemme $\phi(m)$ er allerede tidskrævende når $m \sim 1000$ og umulig når $m \sim 10^{100}$. Forestil m er et helt tal med 100 cifre. For at bestemme $\phi(m)$ skal du tælle de hele tal mindre end m, der er indbyrdes primisk med m. Lad os antage, at du kan afgøre om et helt tal er primisk med m i løbet af 1 million'te del af et sekund (10^{-6} sek.). Ialt vil dine overvejelser af de ialt 10^{200} tal tage:

$$10^{100} \cdot 10^{-6}\text{sek.} = 10^{94}\frac{1}{60 \cdot 60 \cdot 24 \cdot 365} \quad \text{år} \sim 3,2\,10^{86} \quad \text{år.}$$

Det er flere gange universets alder! Den direkte metode til bestemmelse af $\phi(m)$ er åbenbart UHYRE tidskrævende.

I visse tilfælde er det imidlertid let at bestemme $\phi(m)$. For et primtal p er alle tal $1, 2, \ldots, p-1$ primiske med p, derfor er $\phi(p) = p - 1$. Kendes en faktorisering af $m = pq$ i et produkt af primtal er det også en let at bestemme $\phi(m)$.

PROPOSITION 37. *For et primtal p gælder, at*

$$\phi(p) = p - 1.$$

n	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
$\phi(n)$	1	1	2	2	4	2	6	4	6	4	10	4	12	6	10	8

TABEL 1. Tabel over værdier $\phi(n)$ af Eulers funktion for $n = 1, \ldots, 16$

For 2 forskellige primtal p, q gælder,at

$$\phi(pq) = (p-1)(q-1).$$

BEVIS. Lad $m = pq$, hvor p, q er to forskellige primtal. Tallene mindre end m, der ikke er primiske med m, er netop de tal, der har p eller q som primfaktorer, altså de $q-1$ tal

$$p \cdot 1, p \cdot 2, \ldots, p \cdot (q-1)$$

og de $p-1$ tal

$$1 \cdot q, 2 \cdot q, \ldots, (p-1) \cdot q.$$

Antallet af tal, der er primiske med $m = pq$ er derfor

$$(pq-1) - (q-1) - (p-1) = (p-1)(q-1).$$

□

Vi skal i beviset for sætningen om eksistensen af primitive rødder modulo p gøre brug af en særlig egenskab ved ϕ. Lad os starte med at se på $n = 12$, der har divisorerne $1, 2, 3, 4, 6, 12$. Konsulterer vi tabel 1, ser vi, at

$$\phi(1)+\phi(2)+\phi(3)+\phi(4)+\phi(6)+\phi(12) = 1+1+2+2+2+4 = 12 = n.$$

SÆTNING 38. *Hvis $n \geq 1$, så er*

$$n = \sum_{d>0, d|n} \phi(d)$$

BEVIS. Lad $S = \{1, \ldots, n\}$ og lad for enhver divisor d i n:

$$S_d = \{a \in S \mid \mathrm{sfd}(a, n) = \frac{n}{d}\}$$

Mængden S er den disjunkte forening af S_d'erne, thi for et $a \in S$ er der en entydigt bestemt divisor d i n, så $\mathrm{sfd}(a, n) = \frac{n}{d}$, derfor er

$$n = |S| = \sum_{d>0, d|n} |S_d|$$

og det er nok at vise, at

$$|S_d| = \phi(d)$$

for alle d.

For at vise dette dividere vi alle $a \in S_d$ med den største fælles divisor $\frac{n}{d}$, det giver en afbildning

$$S_d \to \{\overline{a} \mid 1 \leq \overline{a} \leq d \wedge \mathrm{sfd}(\overline{a}, d) = 1\}, \quad a \mapsto \frac{a}{\left(\frac{n}{d}\right)},$$

der let ses at være bijektiv, hvilket viser påstanden.

□

Reduceret sæt af rester

Tidligere indførte vi begrebet et komplet sæt af rester modulo n - det er n tal, hvoraf ingen er indbyrdes ækvivalente modulo n, altså n tal, hvor der er præcis et fra hver restklasse. Et *reduceret sæt af rester modulo* n er $\phi(n)$ tal, hvoraf ingen er indbyrdes ækvivalente modulo n og hver for sig er tallene primiske med n.

Ser vi for eksempel på tilfældet $n = 12$, er tallene

$$0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11$$

et komplet sæt af rester, mens tallene

$$1, 5, 7, 11$$

er et reduceret sæt af rester. Tallene $5, 25, 35, 55$ er et andet reduceret sæt af rester.

Det sidste sæt af reducerede rester er fremkommet ved multiplikation med 5 af det første. Det er et eksempel gå et generelt fænomen.

LEMMA 39. *Lad a være primisk med n. Hvis*

$$r_1, \ldots, r_{\phi(n)}$$

er et reduceret sæt af rester modulo n, så er

$$ar_1, \ldots, ar_{\phi(n)}$$

også et reduceret sæt af rester modulo n.

BEVIS. Anvend Korollar 26. □

Eulers sætning

Euler beviste følgende generalisering af Sætning 27, Fermats lille sætning.

SÆTNING 40. ***(Eulers sætning).*** *Lad a være indbyrdes primisk med n. Der gælder at*

$$a^{\phi(n)} \equiv 1 \mod n$$

eller sagt på en anden måde

$$[a^{\phi(n)}] = [1] \quad i \quad \mathbb{Z}_n$$

BEVIS. Beviset anvender nøjagtig den samme metode og ide som beviset for Fermats sidste sætning. Lad

$$r_1, \ldots, r_{\phi(n)} \tag{0.9}$$

være et reduceret sæt af rester. Multiplicerer vi elementerne i (0.9) med a fås et andet reduceret sæt af rester, jvf. Lemma 39:

$$ar_1, \ldots, ar_{\phi(n)}. \tag{0.10}$$

Derfor er produktet af elementerne i (0.9) det samme som produktet af elementerne i (0.10) modulo n:

$$r_1 \cdot r_2 \cdot \cdots \cdot r_{\phi(n)} \equiv (a \cdot r_1) \cdot (a \cdot r_2) \cdot \cdots \cdot (a \cdot r_{\phi(n)}) \mod n \quad \Rightarrow$$

$$r_1 \cdot r_2 \cdot \cdots \cdot r_{\phi(n)} \equiv a^{\phi(n)} \cdot r_1 \cdot r_2 \cdot \cdots \cdot r_{\phi(n)} \mod n.$$

Da alle r_i er indbyrdes primiske med n, kan de bortforkortes ifølge Korollar 26 og

$$1 \equiv a^{\phi(n)} \mod n.$$

□

ØVELSE 41. Bestem $\phi(14)$ og et sæt af reducerede rester. Vis ved direkte udregning, at $a^{\phi(14)} \equiv 1 \mod 14$ for alle a, der er primiske med 14.

KAPITEL 6

Potenser modulo m

Hvordan udregnes $a^k \mod m$, når a, k og m er store. For små tal er det ikke noget problem; men hvordan med

$$1456^{1498675849257698} \mod 345123?$$

Multiplikation af 1456 med sig selv 1498675849257698 gange giver et astronomisk stort tal, som end ikke computere kan håndtere direkte. Vi skal lære en metode, der går under navnet *gentagen kvadrering*, som gør det til en leg at foretage udregningen, selv på en lommeregner og sågar i hånden.

En effektiv metode til at udregne

$$a^k \mod m,$$

når a, k og m er store, er en forudsætning for at vi senere i kapitel 8 kan konstruere og anvende sikre krypteringsværktøjer.

Gentagen kvadrering

Lad os illustrere metoden på nogle eksempler.

EKSEMPEL 42. Beregning af

$$3^7 \mod 5.$$

Først bestemmer vi

$$\begin{aligned}
3^1 &\equiv 3 && && && && \mod 5\\
3^2 &\equiv (3^1)^2 \equiv 9 \equiv 4 && && && \mod 5\\
3^4 &\equiv (3^2)^2 \equiv 4^2 \equiv 16 \equiv 1 && && \mod 5
\end{aligned}$$

Opskriver vi $7 = 4 + 2 + 1$, får vi ved at anvende potensregnereglerne, at

$$3^7 = 3^4 \cdot 3^2 \cdot 3^1 \equiv 1 \cdot 4 \cdot 3 \equiv 2 \mod 5.$$

Altså har vi, at

$$3^7 \equiv 2 \mod 5.$$

Resultatet af det følgende eksempel skal vi bruge i kapitel 7 om roduddragning modulo m.

EKSEMPEL 43. Beregning af

$$13^{53} \mod 77.$$

Først bestemmer vi

$$\begin{array}{lllllllll}
13^1 & \equiv & 13 & & & & & & \mod 5 \\
13^2 & \equiv & (13^1)^2 & \equiv & 169 & \equiv & 15 & & \mod 77 \\
13^4 & \equiv & (13^2)^2 & \equiv & 15^2 & \equiv & 225 & \equiv 71 & \mod 77 \\
13^8 & \equiv & (13^4)^2 & \equiv & 71^2 & \equiv & 5041 & \equiv 36 & \mod 77 \\
13^{16} & \equiv & (13^8)^2 & \equiv & 36^2 & \equiv & 1296 & \equiv 64 & \mod 77 \\
13^32 & \equiv & (13^{16})^2 & \equiv & 64^2 & \equiv & 4096 & \equiv 15 & \mod 77
\end{array}$$

Opskriver vi $53 = 32 + 16 + 4 + 1$, får vi ved at anvende potensregnereglerne, at

$$13^{53} = 13^{32} \cdot 3^{16} \cdot 13^4 \cdot 3^1 \equiv 15 \cdot 64 \cdot 71 \cdot 13 \equiv 41 \mod 77.$$

Altså har vi, at

$$13^{53} \equiv 41 \mod 77. \tag{0.11}$$

Lad os nu til sidst se på et stort eksempel.

EKSEMPEL 44. Beregningen af

$$5^{711} \mod 954.$$

Første laver vi hjælpeberegningerne. Vi starter i første række med 5^1 mod 954, næste række er $5^2 \mod 954$, tilsvarende fås de efterfølgende rækker ved *gentagen kvadrering* af resultatet af den foregående række efterfulgt af bestemmelse af resten ved division med 954.

$$\begin{array}{lllllllll}
5^1 & \equiv & 5 & & & & & & \mod 954 \\
5^2 & \equiv & (5^1)^2 & \equiv & 25 & & & & \mod 954 \\
5^4 & \equiv & (5^2)^2 & \equiv & 25^2 & \equiv & 625 & & \mod 954 \\
5^8 & \equiv & (5^4)2 & \equiv & 625^2 & \equiv & 390625 & \equiv 439 & \mod 954 \\
5^{16} & \equiv & (5^8)^2 & \equiv & 439^2 & \equiv & 192721 & \equiv 13 & \mod 954 \\
5^{32} & \equiv & (5^{16})^2 & \equiv & 13^2 & \equiv & 169 & & \mod 954 \\
5^{64} & \equiv & (5^{32})^2 & \equiv & 169^2 & \equiv & 28561 & \equiv 895 & \mod 954 \\
5^{128} & \equiv & (5^{64})^2 & \equiv & 895^2 & \equiv & 801025 & \equiv 619 & \mod 954 \\
5^{256} & \equiv & (5^{128})^2 & \equiv & 619^2 & \equiv & 383161 & \equiv 607 & \mod 954 \\
5^{512} & \equiv & (5^{256})^2 & \equiv & 607^2 & \equiv & 368449 & \equiv 205 & \mod 954
\end{array}$$

Dernæst opskriver vi $711 = 512 + 128 + 64 + 4 + 2 + 1$ og får ved at anvend potensregnereglerne, at

$$5^{711} = 5^{512+128+64+4+2+1} = 5^{512} \cdot 5^{128} \cdot 5^{64} \cdot 5^4 \cdot 5^2 \cdot 5^1$$

Udnytter vi nu resultaterne fra tidligere kan højre siden udregnes, idet vi konsekvent regner $\mod 954$:

$$
\begin{array}{cccccccccccc}
5^{512} & \cdot & 5^{128} & \cdot & 5^{64} & \cdot & 5^4 & \cdot & 5^2 & \cdot & 5 & \equiv \\
205 & \cdot & 619 & \cdot & 895 & \cdot & 625 & \cdot & 25 & \cdot & 5 & \equiv \\
 & 126895 & \cdot & 895 & \cdot & 625 & \cdot & 25 & \cdot & 5 & \equiv & \\
 & 13 & \cdot & 895 & \cdot & 625 & \cdot & 25 & \cdot & 5 & \equiv & \\
 & & 11635 & \cdot & 625 & \cdot & 25 & \cdot & 5 & \equiv & & \\
 & & 187 & \cdot & 625 & \cdot & 25 & \cdot & 5 & \equiv & & \\
 & & & 116875 & \cdot & 25 & \cdot & 5 & \equiv & & & \\
 & & & 487 & \cdot & 25 & \cdot & 5 & \equiv & & & \\
 & & & & 12175 & \cdot & 5 & = & & & & \\
 & & & & 725 & \cdot & 5 & = & & & & \\
 & & & & & 3635 & \equiv & & & & & \\
 & & & & & 773 & & & & & &
\end{array}
$$

Konklusionen er, at

$$5^{711} \equiv 773 \mod 954.$$

De tre eksempler skulle gerne overbevise dig om, at du i *gentagen kvadrering* har en effektive metode til potensudregning modulo m.

KAPITEL 7

Roduddragning modulo m

I kapitel 6, så vi at gentagen kvadrering var en effektiv metode til at fortage potensopløftning modulo m.

Vi vil nu se på den modsatte operation, nemlig roduddragning modulo m. Vi vil erfare, at generelt er roduddragning en UHYRE langsommelig proces; men kendskab til $\phi(m)$ ændrer situationen radikalt, processen bliver let og hurtig.

PROBLEM 45. *Lad k, b, m være givne hele tal, bestem en løsning til ligningen*

$$x^k \equiv b \mod m, \tag{0.12}$$

bestem for eksempel en løsning til ligningen

$$x^{17} \equiv 13 \mod 77. \tag{0.13}$$

Den mest naive løsningsmetode er at prøve sig frem ved at indsætte tallene $x = 1, 2, 3, \ldots, m$ et ad gangen i (0.12). Hvis m er stor vil det tage meget lang tid. Det er faktisk dette forhold, der er sikkerheden i det krypteringssystem, vi skal behandle i kapitel 8. Roduddragning modulo m tager i almindelighed lang tid (i modsætning til potensopløftning).

Kender vi $\phi(m)$, findes der imidlertid en meget effektiv metode til roduddragning, den skal vi behandle nu.

Roduddragning modulo m, når vi kender $\phi(m)$

Vi vil sideløbende behandle den generelle situation (0.12) og eksemplet (0.13), hvor $\phi(77) = \phi(7)\phi(11) = (7-1)(11-1) = 60$, jvf. Proposition 37.

Indledningsvis bestemmer vi positive hele tal u, v, der er løsninger til

$$ku - \phi(m)v = 1.$$

Det kan vi gøre ved metoden behandlet i Sætning 9, dog under den forudsætning, at k og $\phi(m)$ er indbyrdes primiske. I (0.13) er forudsætningen opfyldt, og vi finder at $u = 53, v = 15$.

$$17 \cdot 53 - 60 \cdot 15 = 1.$$

For alle hele tal x, der er primiske med m, har vi derfor, at

$$(x^k)^u = x^{ku} = x^{1+\phi(m)v} = x(x^{\phi(m)})^v = x.$$

ifølge Sætning 5 (Euler's sætning). Specielt gælder, at

$$(x^{17})^{53} = x^{17\cdot 53} = x^{1+60\cdot 15} = x(x^{60})^{15} = x.$$

Opløfter vi nu venstre og højre side af (0.12) (hhv. 0.13) i potensen u (hhv. 53), fås:

$$x \equiv (x^k)^u \equiv b^u \mod m,$$
$$x \equiv (x^{17})^{53} \equiv 13^{53} \equiv 41 \mod 77.$$

jvf. 0.11 i Eksempel 43 og vi har uddraget den k'te rod.

METODE 46. ***(k'te rod modulo m).***

Lad k, b, m være givne hele tal med k og $\phi(m)$ indbyrdes primiske. Følgende skridt bestemmer en løsning x til:

$$x^k \equiv b \mod m,$$

(1) *Bestem $\phi(m)$. jvf. kapitel 4.*

(2) *Bestem positive hele tal u, v, der er løsninger til*

$$ku - \phi(m)v = 1,$$

jævnfør Sætning 9.

(3) *Udregn $b^u \mod m$ ved gentagen kvadrering, jvf. kapitel 6.*

Af de tre trin i metoden volder kun det første problemer.

KAPITEL 8

Ubrydelige koder - kryptering og digital signatur

Underskrifter og brevhemmelighed

I dette kapitel findes en introduktion, der kan give en almen forståelse for, hvad digitale signaturer og kryptering er, og hvad der kræves for, at de kan anvendes i praksis.

Underskrift. Til daglig anvendes den personlige underskrift til at give en meddelelse troværdighed. Breve underskrives, bilag attesteres, aftaler underskrives af begge parter osv. Den der underskriver forpligtiger sig selv eller andre til meddelelsens indhold. I lovgivningen findes tusindvis af udtrykkelige krav om skriftlighed og underskrift. Ofte er meddelelsen udformet på et særlig trykt papir, som er med til at give meddelelsen sin troværdighed.

Andre meddelelser, f.eks. en faktura, accepteres uden, at den er forsynet med en underskrift, fordi den er udformet på en kendt måde og i øvrigt kan kontrolleres.

I en række sammenhænge giver den almindelige underskrift ikke tilstrækkelig sikkerhed, og der er derfor etableret yderligere sikkerhed. Der kan f.eks. være tale om vitterlighedsvidner, hvor andre personer tillige underskriver meddelelser . Dernæst kan der være behov for at kunne dokumentere, at et brev er sendt og modtaget. Her kan f.eks. anvendes anbefalede breve.

Elektronisk underskrift - digital signatur. Ved overgang til elektronisk kommunikation er det nødvendig at anvende sikkerhedsfunktioner, som giver meddelelsen troværdighed. En elektronisk signatur er data, som tilføjet en elektronisk meddelelse giver vished om, hvem der har udfærdiget meddelelsen.

Når man åbner en butik på Internettet, kan kunderne komme fra hele landet - ja, fra hele verden. Det er derfor nødvendigt med sikkerhed for, at den, der bestiller varer, findes og vil betale. Offentlige myndigheder, der vil give borgerne mulighed for at udfylde blanketter og give oplysninger til en sagsbehandling fra myndighedens hjemmeside, må have sikkerhed for, hvem oplysningerne kommer fra. Virksomheder, som e-handler, skal kunne underskrive sine ordrer og modtage en

faktura, der direkte kan behandles af IT-systemerne. Når en organisation modtager elektroniske meddelelser, som behandles direkte af dens IT-systemer uden menneskelig mellemkomst, er en sikker elektronisk signatur en forudsætning for, at dette kan ske på betryggende vis.

Offentlige myndigheder vil generelt åbne for elektronisk kommunikation med borgere og virksomheder via hjemmesider og med e-post. I alle de tilfælde, hvor lovgivning stiller krav om skriftlighed og underskrift, er det nødvendigt med en sikker elektronisk signatur, som kan sidestilles med underskriften på papirer.

Brevhemmelighed - kryptering. En persons kommunikation med andre bør naturligvis ikke kunne læses af uvedkommende. I de fleste tilfælde sikres denne brevhemmelighed ganske enkelt ved, at brevet sendes i en kuvert. Kuverten sammen med tilliden til Post Danmark sikrer brevhemmeligheden. Ønskes en større sikkerhed for brevhemmeligheden kan kuverten med meddelelsen sendes med en betroet kurer eller lignende.

I den elektroniske kommunikation er det nødvendigt med en sikkerhedsfunktion, som tilsvarende kan sikre brevhemmeligheden. Ved anvendelse af kryptering af meddelelsen kan det sikres, at den kun kan læses af rette vedkommende. Med kryptering sker der en kodning eller forvanskning af meddelelsens indhold (klarteksten), så det man kan se, er det rene volapyk. Denne forvanskning sker dog på en kontrolleret måde således, at indholdet (klarteksten) senere kan genetableres. Man siger, at meddelelsen først enkrypteres og senere - når den skal læses igen - dekrypteres.

Organisationer, der sender følsomme personoplysninger, er f.eks. forpligtiget til at sikre, at oplysningerne ikke kommer til uvedkommendes kendskab. Registertilsynet har udtalt, at ved elektronisk kommunikation skal dette sikres med en tilstrækkelig sikker kryptering.

Digital signatur, hvordan?

Den digitale signatur anvender en krypteringsteknik med to nøgler, som har den egenskab, at man kan enkryptere en meddelelse (dvs. ændre den til volapyk) med den ene nøgle og bringe meddelelsen tilbage til sin oprindelige tekst (klarteksten) med den anden nøgle, se **??**.

Udgangspunktet for en digital signatur er en digital meddelelse, som skal signeres. Af praktiske grunde beregnes på en helt bestemt måde et stort tal ud fra meddelelsen. Det er dette tal, som krypteres med den ene nøgle og tilføjes meddelelsen. Dette volapyk udgør den digitale signatur. Når modtageren senere skal kontrollere underskriften, vil

hun først beregne det samme tal ud fra meddelelsen. Derefter vil programmet dekryptere den digitale signatur - dvs. det store tal vil blive gendannet. Sammenlignes de to tal, og er de ens, kan det konstateres, at den digitale signatur er udført med den anden nøgle.

Men hvorfor er det en signatur? Den første nøgle - som anvendes til at danne den digitale signatur - er indehaverens *private nøgle* . Privat skal forstås helt bogstaveligt, idet det udelukkende er indehaverens selv, som må have adgang til den. Den anden nøgle derimod er en fuldstændig *offentlig nøgle*. Alle, som er interesseret, kan få adgang til den. Når modtageren derfor beregner en digital signatur med den private nøgle, er signaturen unik - det er kun den ene private nøgle, som kan danne lige netop denne digitale signatur. Samtidig er det kun den tilhørende offentlige nøgle, som kan verificere signaturen ved at genskabe det tal, der er beregnet ud fra meddelelsen.

Den digitale signatur er således intetsigende data, som er tilføjet den meddelelse, der er underskrevet. Man kan ikke ud af selve den digitale signatur se, hvem der har underskrevet meddelelsen. I tilknytning til den digitale signatur vil der derfor være en henvisning til den offentlige nøgle, enten ved at den er sendt med eller til en computer på Internettet, hvor den kan hentes. Den offentlige nøgle opbevares i et såkaldt *certifikat*, der udover nøglen også indeholder nøgleindehaverens navn. Når man verificerer en underskrift, har man således meddelelsens indhold og navn mv. i certifikatet med den offentlige nøgle til at sikre sig, at den digitale signatur er foretaget af rette vedkommende.

Ud over at den digitale signatur giver vished for, hvem der har underskrevet den digitale meddelelse, giver den også sikkerhed for, at meddelelsen ikke er ændret, efter den blev underskrevet. Den digitale signatur verificeres, som beskrevet oven for ved at sammenligne den talværdi, som beregnes ud fra den meddelelse, som man har modtaget, med den talværdi der blev beregnet, da den digitale signatur i sin tid blev dannet. Er meddelelsen blevet ændret, vil disse to talværdier ikke være ens - signaturen ikke kan verificeres. Meddelelsen kan dermed ikke anvendes.

L 229 (som vedtaget): Forslag til lov om elektroniske signaturer.

Folketinget vedtog ved 3. behandling den 18. maj 2000 *Forslag til Lov om elektroniske signaturer*

§ 1. Lovens formål er at fremme en sikker og effektiv anvendelse af elektronisk kommunikation gennem fastsættelse af krav til visse elektroniske signaturer og til nøglecentre, der udsteder certifikater til elektroniske signaturer.

§ 2. Loven finder anvendelse på nøglecentre etableret i Danmark, der udsteder kvalificerede certifikater til offentligheden, jf. dog § 12.

Stk. 2. Loven finder desuden anvendelse på efterprøvelse af, at signaturgenereringssystemer overholder de opstillede krav til sikre signaturgenereringssystemer.

IT-sikkerhedsrådet. IT-sikkerhedsrådet skal tilbyde regeringen og forskningsministeriet den højeste faglige rådgivning inden for IT-sikkerhedsområdet samt fremme en kvalificeret offentlig debat om IT-sikkerhed. IT-sikkerhedsrådet skal bidrage til at formulere en overordnet dansk sikkerhedspolitik for anvendelsen af IT og telekommunikation.

Rådet skal pege på de menneskelige og samfundsmæssige risici og interesser, som den moderne informationsteknologi giver anledning til. Rådet skal også komme med anbefalinger om, hvordan disse risici kan imødegås, og hvordan modstridende interesser på IT-sikkerhedsområdet kan opvejes.

Desuden skal IT-sikkerhedsrådet formulere en plan for, hvordan man bedst beskytter data mod brud på fortrolighed, f.eks. gennem brug af elektronisk signatur og kryptering.

Du kan læse mere på adressen http://www.IT-sikkerhedsraadet.dk.

KAPITEL 9

Matematikken bag ubrydelige koder, digital signatur og kryptering

Tekst til tal og tal til tekst

Før vi kan anvende matematik til kryptering må vi omforme teksten til tal. Det gør vi ved først at fastlægge en tabel, der omsætter bogstaver til tal:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
11	12	13	14	15	16	17	18	19	20	21	22	23	24	25

P	Q	R	S	T	U	V	W	Y	Z	Æ	Ø	Å	_	.
26	27	28	29	30	31	32	33	34	35	36	37	38	39	40

Teksten:

SOLEN_ER_SÅ_RØD

bliver altså til tallet:

29252215243915283929383928371 4

Offentlig nøgle kryptosystem

Til kryptosystemet skal der konstrueres 2 nøgler en offentlig og en privat. Det kan gøres således.

Forudsætninger. Kryptosystemet lægger til grund og hviler på 3 forudsætninger:

- vi let kan udregne
$$a^k \mod m \tag{0.14}$$
(jvf. kapitel 6) selv for store værdier af a, k, m.
- at vi kan uddrage k'te rødder (jvf. kapitel 7), altså løse kongruensen
$$x^k \equiv b \mod m \tag{0.15}$$
med hensyn til x, **såfremt vi kender tallet** $\phi(m)$.
- at det er UHYRE tidskrævende at beregne $\phi(m)$ ud fra m, jvf. kapitel 5.

Vigtige valg og konstruktion af nøgler. Vælg 2 (store) primtal p, q, for eksempel

$$p = 977, q = 1039 \tag{0.16}$$

så er $m = pq = 977 \cdot 1039 = 1015103$ og $\phi(m) = (p-1)(q-1) = 1013088$. Vælg $k = 257$ (indbyrdes primisk med $\phi(m) = 1013088$). Nøglerne er nu konstruerede.

- Offentlige nøgle: k, m
- Private (hemmelige) nøgle: $\phi(m)$

Kryptering. Teksten:

SOLEN_ER_SÅ_RØD

blev til tallet:

29252215243915283929383928 3714

som afsenderen bryder op i 5 tal, hver med 6 cifre (Tallet $m = 1015103$ har 7 cifre):

$$\underbrace{292522}_{a_1}\underbrace{152439}_{a_2}\underbrace{152839}_{a_3}\underbrace{293839}_{a_4}\underbrace{283714}_{a_5}$$

Afsenderen beregner ved hjælp af den offentlige nøgle (k, m) tallene

$$b_k := a_i^k \mod m \text{ for } i = 1 \ldots 5$$

:

$$\begin{aligned}
b_1 &:= 292522^{257} \equiv 50100 \mod 1015103 \\
b_2 &:= 152439^{257} \equiv 580973 \mod 1015103 \\
b_3 &:= 152839^{257} \equiv 998296 \mod 1015103 \\
b_4 &:= 293839^{257} \equiv 966609 \mod 1015103 \\
b_5 &:= 283714^{257} \equiv 267198 \mod 1015103
\end{aligned}$$

Vi sender nu disse 5 tal - det er kryptoteksten.

Dekryptering. Modtageren ønsker at bestemme den k'te rod af de modtagne b_i for at rekonstruere a_i. Det gøres i henhold til kapitel 7 således: Modtageren bruger sin hemmelige nøgle $\phi(m)$ til at bestemme positive hele tal u, v, der er løsninger til

$$ku - \phi(m)v = 1.$$

I nærværende eksempel, hvor $k = 257$ og $\phi(m) = 1013088$ er $u = 169505$ og $v = 43$ løsninger.

Den k'te rod af de modtagne b_i opnås nu ved potensopløftning til u modulo m:

$$b_i^u \equiv ((a_i)^k)^u \equiv a_i \mod m.$$

I eksemplet får vi:

$$\begin{aligned}
b_1^u \mod m &= 50100^{169505} \equiv 292522 \mod 1015103\\
b_2^u \mod m &= 580973^{169505} \equiv 152439 \mod 1015103\\
b_3^u \mod m &= 998296^{169505} \equiv 152839 \mod 1015103\\
b_4^u \mod m &= 966609^{169505} \equiv 293839 \mod 1015103\\
b_5^u \mod m &= 267198^{169505} \equiv 283714 \mod 1015103
\end{aligned}$$

og vi har rekonstrueret den sendte talrække

$$292522152439152839293839283714,$$

som let oversættes ved hjælp af tabellen til den oprindelige tekst

$$\text{SOLEN\_ER\_SÅ\_RØD.}$$

Sikkerheden. Er systemet sikkert? En opsnappet meddelelse, der er indkodet med m, k, som jo er offentlige, kan afkodes, hvis $\phi(m)$ kan beregnes. Når $m = pq$ er et produkt af to primtal p og q så er

$$\phi(m) = (p-1)(q-1) = pq - (p+q) + 1 = m - (p+q) + 1 \quad (0.17)$$

Da m allerede kendes, svarer det til at bestemme $p+q$. Kan $p+q$ imidlertid bestemmes, så kan også p og q bestemmes, de vil nemlig være rødder i 2.gradsligningen:

$$X^2 - (p+q)X + m = 0. \quad (0.18)$$

Dekryptering som ovenfor beskrevet svarer altså til at bestemme faktorerne p og q i m. DET TAGER LANG TID.

Ideen til at lave offentlig nøgle kryptering skyldes Diffie og Hellman (1976). Det system, der er behandlet ovenfor skyldes Rivest, Shamir og Adleman (1977) og kaldes RSA-systemet . Det er kernen i mange de kryptosystemer, der anvendes idag.

Gratis offentligt nøgle kryperingssystem. PGP (Pretty Good Privacy) er et frit tilgægeligt krypteringsystem til digital signatur. PGP er tilgængelig for UNIX, MacOS, Windows og mange andre platforme

Du kan læse mere om og hente PGP ned fra adresserne http://www.PGPi.com og http://www.pgp.dk/