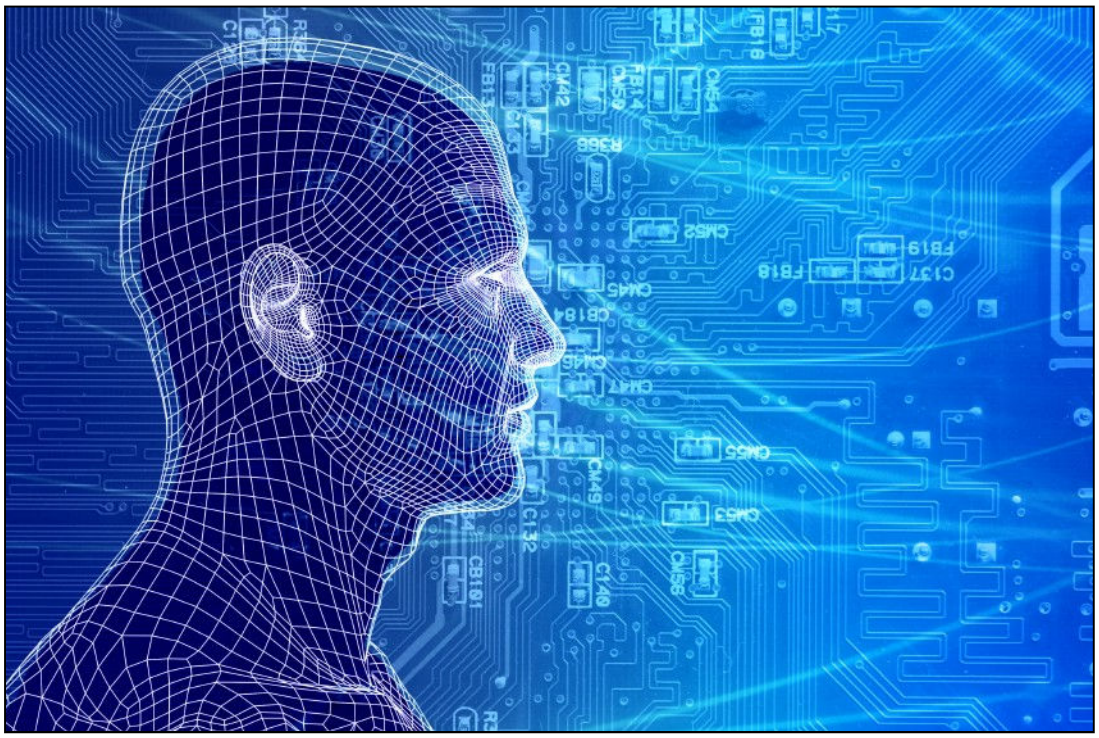
RSA-kryptosystemet



Billeder:
Forside: ©iStock.com/demo10

1. Indledning

I denne lille note skal vi studere det såkaldte RSA kryptosystem, udviklet af *Rivest*, *Shamir* og *Adleman* i 1977. Metodens sikkerhed bygger på den meget efterprøvede antagelse, at det er et meget stort arbejde at finde primtalsfaktorer i et meget stort tal. Matematikken i systemet involverer derfor blandt andet primtalsteori. Hertil kommer regning med rester og fælles divisorer. Alt sammen hører det indenfor området talteori. Vi skal først præsentere udvalgte elementer af talteorien og dernæst forklare, hvordan RSA-systemet er skruet sammen.

2. Regning med rester

Fra folkeskolen ved vi, at et helt tal a går op i et andet helt tal b, såfremt divisionen af b med a giver et helt tal. Situationen kan også defineres således:

Definition 1

Et helt tal $a \neq 0$ *går op i* et helt tal b, såfremt der findes et helt tal q, så $b = q \cdot a$. Vi skriver $a \mid b$. Tallet a betegnes *divisoren* og q *kvotienten*.

Fra folkeskolen ved vi også, at ikke alle divisioner går op. Der kan forekomme en rest. For eksempel vil 53 divideret med 5 give 10 med 3 til rest, eftersom $53 = 10 \cdot 5 + 3$ og fordi resten 3 skal være et tal ≥ 0, som er mindre end divisoren, som er 5. Det er oplagt, at følgende sætning gælder:

Sætning 2

For vilkårlige hele tal $a > 0$ og b gælder, at der findes *entydigt* bestemte hele tal q og r, så $b = q \cdot a + r$, hvor $0 \leq r < a$.

Eksempel 3

Der er en meget smart måde, hvorpå man kan finde kvotienten og resten ved en division. Hvis $b \geq 0$: Foretag divisionen, eventuelt på lommeregneren. Da vil heltalsdelen være lig med kvotienten og hvis man ganger brøkdelen med divisoren a, får man resten. For eksempel er $43/12 = 3{,}58333...$, og ifølge metoden er heltalsdelen 3 lig med kvotienten, mens resten fås ved at gange brøkdelen 0,58333… med 12, hvilket giver $12 \cdot 0{,}58333... = 7$. Bemærk, at da man ikke kan medtage uendeligt mange decimaler, kan det være nødvendigt at afrunde resultatet af multiplikationen. Vi ser, at det stemmer, at $43 = 3 \cdot 12 + 7$. Overvej, hvad du vil gøre, hvis $b < 0$. Se opgaverne 20, 21 og 22.

I det følgende skal vi bevise en vigtig sætning om *divisibilitet*! Den skal sener vise sig nyttig i forbindelse med begrebet fælles divisor.

Sætning 4

For hele tal a, b og c gælder:

a) Hvis $a \neq 0$: $a \mid b \Rightarrow a \mid bc$
b) Hvis $a, c \neq 0$: $ac \mid bc \Leftrightarrow a \mid b$
c) Hvis $a, c \neq 0$: $a \mid b \wedge b \mid c \Rightarrow a \mid c$
d) Hvis $x, y \in Z, c \neq 0$: $c \mid a \wedge c \mid b \Rightarrow c \mid (xa + yb)$

Bevis: Vi skal naturligvis gøre heftig brug af definition 1.

a) Da $a \mid b$ ved vi, at der findes et q, så $b = q \cdot a$. Ganger vi på begge sider med c, får vi $bc = qac = (qc) \cdot a$. Dermed har vi vist, at a går op i bc med kvotient qc.
b) $\Rightarrow$: Da $ac \mid bc$ ved vi, at der findes et q, så $bc = q \cdot ac$. Da $c \neq 0$, kan vi dividere med tallet og får $b = q \cdot a$, som viser, at $a \mid b$ med kvotient q. Modsat vej $\Leftarrow$: Da $a \mid b$ findes et q, så $b = q \cdot a$. Da $c \neq 0$ fås $bc = q \cdot a \cdot c = q \cdot (ac)$, hvilket viser, at $ab \mid ac$ med kvotient q.
c) Da $a \mid b$, findes der et q_1 så $b = q_1 \cdot a$, og da $b \mid c$, findes der et q_2, så $c = q_2 \cdot b$. Dermed haves $c = q_2 \cdot b = q_2 \cdot (q_1 \cdot a) = (q_1 \cdot q_2) \cdot a$. Altså $a \mid c$ med kvotient $q_1 q_2$.
d) Overlades til læseren i opgave 23.

□

Eksempel 5

Da $8 \mid 48$ har vi ifølge a) også, at $8 \mid (5 \cdot 48)$ altså $8 \mid 240$. Ifølge b) haves for eksempel at $3a \mid 3b \Leftrightarrow a \mid b$. Vi har $7 \mid 56$ og $56 \mid 504$, altså haves ifølge c), at $7 \mid 504$. d) siger, at hvis et tal c går op i to tal a og b, så går c også op i enhver linearkombination af a og b.

Vi skal herefter se på nogle sætninger, som gælder for rester. Først en definition:

Definition 6

For hele tal a og n med $n > 0$, indfører vi en hensigtsmæssig notation for *resten* af a ved division med n, nemlig $a \pmod{n}$. Dette udtales: "a modulo n".

Eksempel 7

$38 \pmod{7} = 3$, fordi 38 divideret med 7 giver 5 med 3 til rest. På tilsvarende måde fås $-63 \pmod{5} = 2$, fordi –63 divideret med 5 giver –13 med 2 til rest: $-63 = (-13) \cdot 5 + 2$.

Sætning 8

For hele tal a og b med $b > 0$, gælder:

a) $n \mid a \Leftrightarrow a \pmod{n} = 0$
b) Hvis $0 \leq a < n$, så er $a \pmod{n} = a$
c) $(a \pmod{n}) \pmod{n} = a \pmod{n}$
d) For $k \in Z$: $(a + k \cdot n) \pmod{n} = a \pmod{n}$

Bevis: a) udtaler, at hvis divisionen går op, så er resten 0, hvilket er klart. b) siger, at hvis man dividerer et positivt tal op i et ikke-negativt tal, og divisoren er størst, så er resten lig med det, der divideres op i. Dette er klart, for i så fald vil n gå 0 gange op i a: $a = 0 \cdot n + a$. Nu til c): Den siger, at hvis man regner rester ud modulo n, så er der ingen forskel på at gøre det på a eller $a \pmod n$. Denne påstand er en simpel konsekvens af punkt b), idet $a \pmod n < n$. Endelig d): Den siger, at når man udregner rester modulo n, så ændres resten ikke, hvis man lægger et multiplum af n til før division med n. Antag $a = q \cdot n + r$ med $0 \le r < n$. Så er $a + k \cdot n = (q \cdot n + r) + k \cdot n = (k + q) \cdot n + r$, hvilket viser at resten er uændret. Ikke overraskende ser vi, at den nye kvotient er $k + q$.

□

Vi er nu klar til at bevise en meget vigtig sætning, som skal vise sig meget vigtig, når rester af for eksempel meget store potenser skal bestemmes.

Sætning 9

For $a, b \in Z, n \in N$ gælder:

a) $(a+b)(\text{mod } n) = (a \pmod n + b \pmod n)(\text{mod } n)$

b) $(a \cdot b)(\text{mod } n) = (a \pmod n \cdot b \pmod n)(\text{mod } n)$

c) $a^t \pmod n = (a \pmod n)^t \pmod n$

Eksempel 10

Lad os lige se på et eksempel, før vi beviser sætningen. Sætning 9a) siger kort fortalt, at hvis man vil udregne rester modulo n for en sum $a+b$, så kan man gøre dette ved først at udregne resterne af henholdsvis a og b modulo n, addere resterne og derefter udregne resten af resultatet modulo n. Noget helt tilsvarende gælder for multiplikation og potensopløftning. Lad os som eksempel sige, at $a = 2781$, $b = 8204$, $n = 671$ og at vi ønsker at foretage en multiplikation af a og b modulo n. Vi udregner hver af siderne i a):

$$(a \cdot b)(\text{mod } n) = (2781 \cdot 8204)(\text{mod } 671) = 22815324 \pmod{671} = 653$$

$$\begin{aligned}(a \pmod n + b \pmod n)(\text{mod } n) &= (2781 \pmod{671} \cdot 8204 \pmod{671})(\text{mod } 671) \\ &= (97 \cdot 152)(\text{mod } 671) \\ &= 14744 \pmod{671} \\ &= 653\end{aligned}$$

Vi ser som ventet, at vi får samme rest. Fordelen ved den anden udregning er, at man ikke behøver at regne med så store tal! Når man udregner rester at potenser, så er fordelen endnu større, og vi skal i sætning 28 og eksempel 32 se, hvorledes man yderligere kan lette disse beregninger.

□

Bevis for sætning 9:

a) Vi ved, at a og b kan skrives på formen $a = q_1 \cdot n + r_1$ og $b = q_2 \cdot n + r_2$, hvoraf fås $a + b = (q_1 \cdot n + r_1) + (q_2 \cdot n + r_2) = (q_1 + q_2) \cdot n + (r_1 + r_2)$. Ifølge sætning 8d) får vi følgende: $(a+b)(\text{mod } n) = ((q_1 + q_2) \cdot n + (r_1 + r_2))(\text{mod } n) = (r_1 + r_2)(\text{mod } n)$ ved regning med rester modulo n. Og da $a\,(\text{mod } n) = r_1$ og $b\,(\text{mod } n) = r_2$ er det ønskede vist.

b) Overlades til læseren. Se opgave 24.

c) Denne del følger ved gentagen anvendelse af sætning 9b).

□

3. Største fælles divisor

Et tal d siges at være en *fælles divisor* for a og b, såfremt $d \mid a$ og $d \mid b$. Der er kun et endeligt antal tal, som går op i henholdsvis a og b. Det største af de tal, som går op i både a og b, kaldes den *største fælles divisor* for a og b og betegnes $sdf(a,b)$ eller blot (a,b). Begrebet er vigtigt, fordi det indgår i nogle helt centrale sætninger. Hvis specielt $(a,b) = 1$, så har a og b ingen fælles divisorer udover 1, og a og b siges da at være *indbyrdes primiske*. Vi skal se på egenskaber for den største fælles divisor.

Eksempel 11

Vi skal bestemme den største fælles divisor for $a = 28$ og $b = 48$.
Divisorer i 28: $\pm 1, \pm 2, \pm 4, \pm 7, \pm 14, \pm 28$.
Divisorer i 48: $\pm 1, \pm 2, \pm 3, \pm 4, \pm 6, \pm 8, \pm 12, \pm 16, \pm 24, \pm 48$.
Fælles divisorer: $\pm 1, \pm 2, \pm 4$.
Vi ser, at $(a,b) = (28,48) = 4$.

Nu kan det jo være et ret omfattende arbejde at søge efter divisorer i tal, specielt hvis tallene er ret store. Derfor er det et nærliggende spørgsmål, om der findes en genvej til at bestemme den største fælles divisor for to tal? Svaret er heldigvis positivt, hvilket den næste sætning indikerer.

Sætning 12

Hvis r er resten af b ved division med a, så gælder: $(a,b) = (a,r) = (a, b\,(\text{mod } a))$.

Bevis: Det er klart nok at vise, at $d \mid a \wedge d \mid b \Leftrightarrow d \mid a \wedge d \mid r$. Vi viser begge veje. $\Rightarrow$: Vi antager, at venstresiden gælder. Da $b = q \cdot a + r$, har vi $r = b - q \cdot a$. Da d går op i både a og b, må tallet derfor også gå op i r ifølge sætning 4d). Så højresiden er dermed vist. $\Leftarrow$: Vi antager at højresiden gælder. Da $b = q \cdot a + r$ og d går op i både a og r, så går d også op i b, igen ifølge sætning 4d), hvormed venstresiden er vist.

□

Eksempel 13

Bestem (1650,23322).

Løsning: Vi skal benytte sætning 12 et antal gange til at reducere problemet til et simplere. Bestemmelsen af resterne er angivet til højre. Vi ser, at resultatet er 6.

Største fælles divisor	Restberegninger
$(1650, 23322) = (1650, 222)$ $= (96, 222)$ $= (96, 30)$ $= (6, 30)$ $= (6, 0)$ $= 6$	$23322 = 14 \cdot 1650 + 222$ $1650 = 7 \cdot 222 + 96$ $222 = 2 \cdot 96 + 30$ $96 = 3 \cdot 30 + 6$ $30 = 5 \cdot 6 + 0$

Den metode, som her er beskrevet, betegnes *Euklids algoritme*, opkaldt efter græske matematiker *Euklid* (ca. 300 f.Kr.).

□

Det viser sig, at Euklids algoritme samtidigt kan bruges til at skrive den største fælles divisor (a,b) som en linearkombination af a og b. Hvordan det foregår, kan lettest illustreres via eksempel 13 ovenfor, hvor vi indfører betegnelser for beregningerne i højre kolonne. Her svarer a til 1650 og b til 23322. Venstre kolonne nedenfor svarer til højre kolonne ovenfor.

Restberegninger	Linearkombinationer
$b = q_1 \cdot a + r_1$	$r_1 = b - q_1 \cdot a = (-q_1) \cdot a + 1 \cdot b = s_1 \cdot a + t_1 \cdot b$
$a = q_2 \cdot r_1 + r_2$	$r_2 = a - q_2 \cdot r_1 = \ldots = s_2 \cdot a + t_2 \cdot b$
$r_1 = q_3 \cdot r_2 + r_3$	$r_3 = r_1 - q_3 \cdot r_2 = \ldots = s_3 \cdot a + t_3 \cdot b$
$r_2 = q_4 \cdot r_3 + r_4$	$r_4 = r_2 - q_4 \cdot r_3 = \ldots = s_4 \cdot a + t_4 \cdot b$
$r_3 = q_5 \cdot r_4 + 0$	

Idéen i omskrivningerne i højre kolonne er, at man isolerer resten fra divisionen i venstre kolonne. I første række ser vi for eksempel, at r_1 kan skrives som en linearkombination af a og b: $r_1 = s_1 \cdot a + t_1 \cdot b$, hvor $s_1 = -q_1$ og $t_1 = 1$. Hvis dette udtryk for r_1 indsættes på r_1's plads i udtrykket $a - q_2 \cdot r_1$ i 2. række, så får man efter reduktion, at også resten r_2 kan skrives som en linearkombination af a og b, dvs. $s_2 \cdot a + t_2 \cdot b$. Herefter indsættes både udtrykket for r_1 og udtrykket for r_2 i udtrykket $r_1 - q_3 \cdot r_2$ i tredje række, og man får igen en linearkombination af a og b, nemlig $s_3 \cdot a + t_3 \cdot b$, etc. Som vi ved, er den største fælles divisor for a og b i eksemplet ovenfor lig med r_4, og den sidste lig-

ning i højre kolonne fortæller, at den største fælles divisor (r_4) kan skrives som en linearkombination af *a* og *b*, nemlig $s_4 \cdot a + t_4 \cdot b$. Lad os gennemføre processen i tilfældet med eksempel 13:

Rester
$222 = 23322 - 14 \cdot 1650$
$96 = 1650 - 7 \cdot 222 = 1650 - 7 \cdot (23322 - 14 \cdot 1650) = 99 \cdot 1650 - 7 \cdot 23322$
$30 = 222 - 2 \cdot 96 = (23322 - 14 \cdot 1650) - 2 \cdot (99 \cdot 1650 - 7 \cdot 23322)$
$= -212 \cdot 1650 + 15 \cdot 23322$
$6 = 96 - 3 \cdot 30 = (99 \cdot 1650 - 7 \cdot 23322) - 3 \cdot (-212 \cdot 1650 + 15 \cdot 23322)$
$= 735 \cdot 1650 - 52 \cdot 23322$

Den sidste linje viser, hvordan den største fælles divisor kan skrives som en linear kombination af *a* og *b*: $6 = 735 \cdot 1650 - 52 \cdot 23322$. Der gælder faktisk generelt:

Sætning 14

Lad *a* og *b* være hele tal, som er forskellig fra 0. Da findes hele tal *s* og *t*, så $(a,b) = s \cdot a + t \cdot b$.

Vi skal ikke give et formelt bevis, fordi notationen bliver ret tung. Fremgangsmåden beskrevet i eksemplet ovenfor er dog nok til at man kan overbevise sig om, at metoden fungerer generelt. Det kan undertiden være nyttigt at kunne finde de aktuelle koefficienter i linearkombinationen ved hjælp af Euklids algoritme. Sætningen har også teoretisk interesse, idet den ofte indgår som redskab til at bevise andre sætninger. Se for eksempel følgende sætning:

Sætning 15

a) $d \mid a \ \wedge \ d \mid b \ \Rightarrow \ d \mid (a,b)$

b) For et helt tal $m > 0$ gælder: $(m \cdot a, m \cdot b) = m \cdot (a,b)$

c) $(a,n) = 1 \ \wedge \ (b,n) = 1 \ \Rightarrow \ (ab,n) = 1$

d) $c \mid ab \ \wedge \ (c,b) = 1 \ \Rightarrow \ c \mid a$

Bemærkninger:

a) siger, at hvis et tal *d* går op i *a* og *b*, så går tallet også op i den største fælles divisor for *a* og *b*.

b) siger, at hvis man ganger en fælles faktor *m* på *a* og *b*, fås den største fælles divisor for de nye tal ved at gange *m* på den største fælles divisor for de oprindelige tal.

c) siger, at hvis både *a* og *b* er indbyrdes primiske med *n*, så er produktet $a \cdot b$ også indbyrdes primisk med *n*.

d) siger, at hvis et tal c går op i et produkt af to tal, og hvis c er indbyrdes primisk med det ene af tallene i produktet, så går c op i det andet tal i produktet.

Bevis for sætning 15:

a) Påstanden følger direkte af sætning 14 og sætning 4d).

b) Da begge sider i ligheden er positive, kan ligheden vises ved at påvise, at venstresiden går op i højresiden og at højresiden går op i venstresiden.

Vise $(m\cdot a, m\cdot b) \mid m\cdot(a,b)$: Der findes et s og et t, så $(a,b) = s\cdot a + t\cdot b$, ifølge sætning 14. Ganger vi med m, fås $m\cdot(a,b) = m\cdot(s\cdot a + t\cdot b) = s\cdot(m\cdot a) + t\cdot(m\cdot b)$. Vi mangler blot at påvise, at $(m\cdot a, m\cdot b)$ går op i højresiden, men det er klart, fordi $(m\cdot a, m\cdot b)$ per definition går op i både $m\cdot a$ og $m\cdot b$.

Vise $m\cdot(a,b) \mid (m\cdot a, m\cdot b)$: Per definition haves $(a,b) \mid a$ og $(a,b) \mid b$. Ifølge sætning 4b) haves derfor: $m\cdot(a,b) \mid m\cdot a$ og $m\cdot(a,b) \mid m\cdot b$. Altså går $m\cdot(a,b)$ op i både $m\cdot a$ og $m\cdot b$. Derfor må $m\cdot(a,b)$ også gå op i deres største fælles divisor, $(m\cdot a, m\cdot b)$, ifølge sætning 15a). Det var det, vi ville vise.

c) Ifølge sætning 14 findes der hele tal s_1 og t_1, så $s_1\cdot a + t_1\cdot n = 1$ samt hele tal s_2 og t_2, så $s_2\cdot b + t_2\cdot n = 1$. Når vi ganger de to ligninger sammen, får vi følgende ligning: $1 = (s_1\cdot a + t_1\cdot n)(s_2\cdot b + t_2\cdot n) = (s_1 s_2)\cdot ab + (s_1 t_2 a + t_1 s_2 b + t_1 t_2 n)\cdot n$. Der er altså en linearkombination af ab og n, som er lig med 1. En sætning siger da, at den største fælles divisor mellem tallene er lig med 1 (se opgave 32).

d) Vi har $(ab, ac) = a\cdot(b,c) = a\cdot 1 = a$, hvor vi i første lighedstegn har benyttet sætning 15b) og i andet lighedstegn har udnyttet antagelsen $(b,c) = 1$ fra sætningen. Det er klart, at $c \mid ac$. Ifølge sætning 15a): $c \mid ab \wedge c \mid ac \Rightarrow c \mid (ab, ac) \Leftrightarrow c \mid a$, hvor vi i sidste ensbetydende tegn har benyttet ovenstående udledning $(ac, ac) = a$.

□

4. Primtal

Et helt tal større end 1 kaldes for et *primtal*, hvis tallet kun har de trivielle positive divisorer 1 og tallet selv. Et helt tal større end 1, som ikke er et primtal, kaldes for et *sammensat tal*. Listen af primtal starter således:

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, …

Vi skal se, at denne liste er uendelig. Først skal vi imidlertid se på en vigtig sætning, som siger, at hvis et primtal går op i et produkt af to tal, så går primtallet op i mindst det ene af de to tal i produktet.

Sætning 16
Lad p være et primtal. Da gælder: $p \mid ab \Rightarrow p \mid a \vee p \mid b$.

Bevis: Antag p ikke går op i a. Da haves $(a,p)=1$. Ifølge sætning 15d) vil vi derfor have $p \mid b$. Primtallet må altså gå op i mindst en af faktorerne.

□

Det ses ret nemt, at resultatet i sætning 16 kan generaliseres (se opgave 33) til, at hvis p er et primtal, så gælder $p \mid a_1 \cdot a_2 \cdot \ldots \cdot a_n \Rightarrow p \mid a_1 \vee p \mid a_2 \vee \ldots \vee p \mid a_n$. Vi skal nu vise en fundamental sætning, som har været kendt tidligt i historien, men som først blev bevist af tyskeren *Carl Friedrich Gauss* (1777 – 1855).

Sætning 17 (Algebraens fundamentalsætning)
Ethvert positivt tal n kan skrives som et produkt af primtal. Faktoriseringen er entydig, når der ses bort fra rækkefølgen, hvori faktorerne skrives.

Bevis: Lad os starte med eksistens-delen. Hvis n er et primtal er vi færdige. Hvis derimod n er et sammensat tal, kan det skrives som $n = n_1 \cdot n_2$, hvor n_1 og n_2 begge er større end 1, men mindre end n. Processen gentages nu på n_1 og n_2: Hvis n_1 er et primtal, lader vi det stå, ellers faktoriseres det, etc. Processen må på et tidspunkt stoppe, altså den er endelig, for tallene, man faktoriserer bliver mindre og mindre. Til sidst har man kun primtal tilbage. Dermed er n skrevet som et produkt af primtal. *Entydighed*: Vi skal vise, at hvis vi har to primtals-faktoriseringer $p_1 \cdot p_2 \cdot \ldots \cdot p_r$ og $q_1 \cdot q_2 \cdot \ldots \cdot q_s$ af n, så vil de nødvendigvis indeholde de samme primtal med samme hyppighed. Det eneste, som kan være forskellig er rækkefølgen, hvori de er opskrevet. Da $p_1 \mid p_1 \cdot p_2 \cdot \ldots \cdot p_r$, må vi også have $p_1 \mid q_1 \cdot q_2 \cdot \ldots \cdot q_s$. Ifølge sætning 16 eller dens generalisation, må p_1 gå op i et af q'erne; lad os sige, at det er q_1. Da dette er et primtal, må vi have $p_1 = q_1$. Herefter divideres faktoren væk i begge faktoriseringer, således, at vi har $p_2 \cdot \ldots \cdot p_r = q_2 \cdot \ldots \cdot q_s$. Processen gentages: $p_2 \mid q_2 \cdot \ldots \cdot q_s$, etc. Når alle p'erne er forkortet væk, må også alle q'erne være det, dvs. $r = s$.

□

Sætning 18
Hvis ingen af primtallene $\leq \sqrt{n}$ går op i n, så er n et primtal.

Bevis: Hvis et sammensat tal går op i n, så vil det sammensatte tals primtalsfaktorer også gå op i n. Derfor er det tilstrækkeligt at teste for alle primtal. At man kun behøver at teste for alle primtal op til og med $\sqrt{n}$ følger af, at hvis n er sammensat, så vil mindst en af dens primtalsfaktorer være $\leq \sqrt{n}$. Hvis man multiplicerer to eller flere tal, som er $> \sqrt{n}$, får man jo et tal, som er større end n.

□

Eksempel 19

Vi skal undersøge, om tallet $n = 2351$ er et primtal eller ej. $\sqrt{n} = \sqrt{2351} = 48,49$, så vi behøver kun teste for divisorer blandt primtallene op til 48: 2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47. Ingen af disse tal går op i 2351, så vi har altså at gøre med et primtal.

Eksempel 20

Vi skal primtalsfaktorisere tallet $n = 4319090279$. Vi tester for de mindre primtal: Det mindste primtal, som går op i n er 7 og vi har $4319090279 = 7 \cdot 617012897$. Vi søger videre efter primtalsfaktorer i $n_1 = 617012897$, og starter med at teste for 7, 11, 13, … og opdager, at 19 går op: $617012897 = 19 \cdot 32474363$. Vi arbejder videre med faktoren $n_2 = 32474363$, og søger efter primtalsfaktorer fra 19 og opefter. Allerede med 19 er der gevinst, $32474363 = 19 \cdot 1709177$. Hvad angår $n_3 = 1709177$, må vi arbejde lidt mere. Først ved 727 finder vi en divisor: $1709177 = 727 \cdot 2351$. Tallet 2351 er et primtal, som vi fandt ud af i eksempel 19. Hermed har vi fundet den ønskede primfaktoropløsning af n: $439090279 = 7 \cdot 19^2 \cdot 727 \cdot 2351$.

Med eksempel 20 så vi, at der er en hel del arbejde forbundet med at faktorisere et tal. Problemet viser sig at blive markant større, jo større n er. Og hvis primtalsfaktorerne er store, bliver det rigtigt svært, så kan vi ikke "trævle" problemet op som i eksempel 20 ovenfor. Selvom utallige matematikere igennem århundreder har ledt efter hurtige genveje til at faktorisere tal, så er det ikke lykkedes. Antagelsen er derfor at en sådan hurtig metode ikke eksisterer. At det (formentlig) er et stort arbejde at faktorisere et tal, er netop kernen i RSA-krypteringsteknikken.

Sætning 21
Der findes uendeligt mange primtal.

Bevis: Antag modsætningsvist, at der kun er endeligt mange primtal $p_1, p_2, \ldots, p_r$. Betragt da tallet $n = p_1 \cdot p_2 \cdot \ldots \cdot p_r + 1$. Ifølge algebraens fundamentalsætning kan n faktoriseres i primtal. Men primtalsfaktorerne i n kan umuligt være $p_1, p_2, \ldots, p_r$, fordi man får 1 til rest ved division med disse op i n. Der er altså mindst ét primtal udover de r primtal $p_1, p_2, \ldots, p_r$. Dette er en modstrid. Altså er antagelsen om, at der er endeligt mange primtal, forkert.

□

Sætning 21 er på en måde godt nyt for os, for det betyder, at der ikke er nogen grænser for, hvor store primtal, der kan genereres. De store primtal får vi brug for, for at kunne foretage RSA-kryptering. Når tal er meget store, kræver det et kæmpe arbejde, før man kan være helt sikker på, at tallet er et primtal. Vi skal dog stille os tilfreds med mindre: Der findes metoder, hvormed man kan afgøre med *næsten* sikkerhed, at et givet tal er et primtal. Mere om det senere.

5. Eulers φ-funktion

I dette afsnit skal vi se på den såkaldte *Eulers* φ-*funktion*, som indgår i flere sætninger, som har med regning med rester at gøre. Før begrebet introduceres, skal vi se på en næsten indlysende sætning, der siger, at to tal x og y har samme rest modulo n hvis og kun hvis n går op i differensen af tallene:

Sætning 22

$x \pmod n = y \pmod n \quad \Leftrightarrow \quad n \mid (x-y)$

Bevis: $\Rightarrow$: Las os betegne den fælles rest modulo n med r. Dermed har vi $x = q_1 \cdot n + r$ og $y = q_2 \cdot n + r$. Det giver differensen $x - y = (q_1 \cdot n + r) - (q_2 \cdot n + r) = (q_1 - q_2) \cdot n$, som viser, at $n \mid (x-y)$. $\Leftarrow$: Vi antager, at $n \mid (x-y)$, dvs. der findes et q, så $x - y = q \cdot n$. Det giver $x = y + q \cdot n$. Ifølge sætning 8d) haves $x \pmod n = y \pmod n$.

□

Nu til en sætning, der siger, at hvis $a \cdot x$ og $a \cdot y$ giver samme rest ved division med n, og a og n er indbyrdes primiske, så giver x og y også samme rest ved division med n.

Sætning 23

$ax \pmod n = ay \pmod n \quad \wedge \quad (a,n) = 1 \quad \Rightarrow \quad x \pmod n = y \pmod n$

Bevis: Første oplysning på venstre side giver os at $n \mid (ax - ay)$ eller $n \mid a \cdot (x-y)$, ifølge sætning 22. Da a og n er indbyrdes primiske, fås ifølge sætning 15d), at $n \mid (x-y)$, hvilket ifølge sætning 22 betyder, at $x \pmod n = y \pmod n$.

□

Definition 24

Mængden af rester modulo n betegnes ofte med $Z_n = \{0, 1, 2, \ldots, n-1\}$. Det viser sig interessant at studere den delmængde af rester, som er forskellige fra 0 og som er indbyrdes primiske med n: $Z_n^* = \{r \in Z_n \mid r \neq 0 \wedge (r,n) = 1\}$. Antallet af elementer i mængden Z_n^* betegnes $\varphi(n)$. Funktionen φ, defineret for alle $n \in N$, kaldes *Eulers* φ-*funktion*.

Sætning 25

Lad $n \in N$ og lad $p_1, p_2, \ldots, p_r$ være de *forskellige* primfaktorer i n. Da er

$$\varphi(n) = n \cdot \left(1 - \tfrac{1}{p_1}\right) \cdot \left(1 - \tfrac{1}{p_2}\right) \cdot \ldots \cdot \left(1 - \tfrac{1}{p_r}\right)$$

Specielt: Hvis p og q er to *forskellige* primtal, gælder $\varphi(pq) = (p-1)(q-1)$.
Specielt: Hvis p er et primtal gælder $\varphi(p) = p - 1$ og $\varphi(p^2) = p(p-1)$.

Bevis: Beviset for den generelle formel er for svær til at blive præsenteret her. Specialtilfældene er derimod noget nemmere og er henlagt til opgave 50.

□

Eksempel 26

Lad os betragte tilfældet $n=15$. Regning med rester modulo 15 giver følgende:

$$Z_{15} = \{0,1,2,3,4,5,6,7,8,9,10,11,12,13,14\}$$

Vi søger nu de rester, som er indbyrdes primiske med $n=15$. Da 15 har primfaktorerne 3 og 5, skal vi altså udvælge de tal, hvori hverken 3 eller 5 går op. For eksempel må vi afvise 10, fordi 5 er divisor. Alt i alt fås følgende:

$$Z_{15}^* = \{1,2,4,7,8,11,13,14\}$$

Da der er 8 elementer heri, er $\varphi(15)=8$. Det stemmer også med sætning 25, hvor specialtilfældet $n=3\cdot 5$ giver $\varphi(3\cdot 5)=(3-1)\cdot(5-1)=2\cdot 4=8$.

Nu skal vi foretage nogle beregninger, som leder frem til den næste, meget centrale sætning. Vi skal prøve at gange et fast tal, 26, som er indbyrdes primisk med $n=15$, på resterne i Z_{15}^*, udregne resterne modulo 15 og se, hvad der sker:

$$\begin{aligned}
26\cdot 1 \pmod{15} &= 26 \pmod{15} = 11\\
26\cdot 2 \pmod{15} &= 52 \pmod{15} = 7\\
26\cdot 4 \pmod{15} &= 104 \pmod{15} = 14\\
26\cdot 7 \pmod{15} &= 182 \pmod{15} = 2\\
26\cdot 8 \pmod{15} &= 208 \pmod{15} = 13\\
26\cdot 11 \pmod{15} &= 286 \pmod{15} = 1\\
26\cdot 13 \pmod{15} &= 338 \pmod{15} = 8\\
26\cdot 14 \pmod{15} &= 364 \pmod{15} = 4
\end{aligned}$$

Udregningerne resulterer naturligvis i rester i Z_{15}, men hvad der er overraskede er, at resterne faktisk holder sig indenfor delmængden Z_{15}^*, ja faktisk opnår man resterne heri i en permuteret rækkefølge: hver rest forekommer nøjagtig én gang. I beviset for den følgende sætning skal vi se, at dette ingenlunde er et tilfælde.

□

Sætning 27

Hvis $(a,n)=1$ gælder, at $a^{\varphi(n)} \pmod{n} = 1$

Bevis: Lad $Z_n^* = \{r_1, r_2, \ldots, r_{\varphi(n)}\}$. Vi er nu interesseret i at se på resterne $ar_i \pmod{n}$, hvor $i=1,2,\ldots,\varphi(n)$. Først vil vi vise, at de alle er forskellige og derefter, at de alle ligger i Z_n^*, hvormed vi vil have vist, at man får alle resterne i Z_n^* netop én gang.

Antag, at $ar_i \pmod n = ar_j \pmod n$. Ifølge sætning 23 haves $r_i \pmod n = r_j \pmod n$, da $(a,n)=1$. Da $r_i, r_j < n$ reduceres ligningen til $r_i = r_j$ ifølge sætning 8b). Eller sagt på en anden måde: $r_i \neq r_j \Rightarrow ar_i \pmod n \neq ar_j \pmod n$. Elementerne er altså forskellige!

Per definition opfylder alle resterne i Z_n^*, at $(r_i,n)=1$. Da vi desuden har antaget, at $(a,n)=1$, så fås af sætning 15c), at $(ar_i,n)=1$. Ifølge sætning 12 har vi første lighedstegn i $(ar_i \pmod n), n) = (ar_i, n) = 1$. Da $ar_i \pmod n < n$ og $ar_i \pmod n$ og n er indbyrdes primiske, har vi vist, at $ar_i \pmod n \in Z_n^*$.

Vi har nu vist, at $ar_1 \pmod n, ar_2 \pmod n, \ldots, ar_{\varphi(n)} \pmod n$ er de samme som resterne $r_1, r_2, \ldots, r_{\varphi(n)}$, eventuelt blot i en anden rækkefølge. Hvis vi ganger førstnævnte sammen så får vi altså det samme som ved at gange de sidstnævnte rester sammen:

$$\begin{aligned}\left[r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(n)}\right](\text{mod } n) &= \left[ar_1 \pmod n \cdot ar_2 \pmod n \cdot \ldots \cdot ar_{\varphi(n)} \pmod n\right](\text{mod } n) \\ &= \left[ar_1 \cdot ar_2 \cdot \ldots \cdot ar_{\varphi(n)}\right](\text{mod } n) \\ &= \left[a^{\varphi(n)} \cdot r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(n)}\right](\text{mod } n)\end{aligned}$$

hvor vi i andet lighedstegn har benyttet sætning 9b). r_i'erne er per definition af Z_n^* indbyrdes primiske med n. Derfor er produktet af r_i'erne også indbyrdes primisk med n, ifølge sætning 15c). Da således $r_1, r_2, \ldots, r_{\varphi(n)}$ og n er indbyrdes primiske, kan sætning 23 benyttes til at "forkorte" $r_1, r_2, \ldots, r_{\varphi(n)}$ bort på begge sider:

$$1 \pmod n = a^{\varphi(n)} \pmod n \quad \Leftrightarrow \quad a^{\varphi(n)} \pmod n = 1$$

□

Sætning 28
For alle $a \in Z$ og $n,t \in N$ med $(a,n)=1$ gælder: $a^t \pmod n = a^{t \,(\text{mod } \varphi(n))} \pmod n$

Bevis: Skriv t på formen $t = q \cdot \varphi(n) + r$, hvor $0 \le r < \varphi(n)$. Vi har så $r = t \pmod{\varphi(n)}$:

$$\begin{aligned}a^t \pmod n &= a^{q \cdot \varphi(n) + r} \pmod n \\ &= (a^{\varphi(n)})^q \cdot a^r \pmod n \\ &= \left[\left((a^{\varphi(n)} \pmod n)^q \pmod n\right) \cdot \left(a^r \pmod n\right)\right](\text{mod } n) \\ &= \left[\left(1^q \pmod n\right) \cdot \left(a^r \pmod n\right)\right](\text{mod } n) \\ &= \left(a^r \pmod n\right)\end{aligned}$$

hvor vi i tredje lighedstegn har benyttet både sætning 9b) og 9c). I fjerde lighedstegn er benyttet sætning 27. Dette beviser sætningen.

□

Eksempel 29

Sætning 28 skal vise sig uhyre vigtig i vores arbejde med at beregne rester modulo *n*. I sætning 9 så vi, at vi kunne tage rester modulo *n* i summer, produkter og i roden af potenser, før vi tog de endelige rester modulo *n*. Sætning 28 siger, at vi også kan gøre det i eksponenten i en potens. Blot skal vi ikke der regne modulo *n*, men modulo $\varphi(n)$. Lad os se på et eksempel:

Vi skal udregne $37^{52} \pmod{17}$. Tallet 37^{52} er et tal med 81 cifre. Det ville tage en evighed at udregne tallet og derefter tage resten modulo 17. Heldigvis har vi både sætning 9 samt sætning 28 til rådighed.

$$\begin{aligned} 37^{52} \pmod{17} &= (37 \pmod{17})^{52} \pmod{17} \\ &= 3^{52} \pmod{17} \\ &= 3^{52 \pmod{16}} \pmod{17} \\ &= 3^4 \pmod{17} \\ &= 81 \pmod{17} \\ &= 13 \end{aligned}$$

I første lighedstegn er benyttet sætning 9c). I tredje lighedstegn har vi benyttet sætning 28: Betingelserne hertil er opfyldt, idet $(a,n) = (3,17) = 1$. Da $n = 17$ er et primtal, er det endvidere simpelt at udregne $\varphi(n)$ ifølge sætning 25: $\varphi(17) = 17 - 1 = 16$.

□

Sætning 30 (Inverst element)
Lad $a \in Z_n$. Da gælder:

$(a,n) = 1$

$\Updownarrow$

Der findes en entydig løsning $x \in Z_n$ til ligningen $ax \pmod{n} = 1$

Bevis: $\Downarrow$: Vi antager, at $(a,n) = 1$. Ifølge sætning 14 findes en linearkombination af *a* og *n*, som er lig med 1: $sa + tn = 1$. Omskrivningen $sa = 1 - tn$ viser, at $as \pmod{n} = 1$, da man får 1 til rest ved division med *n*. Ifølge sætning 9b) kan man "sætte modulo *n*" ind på hver faktor i produktet: $(a \pmod{n} \cdot s \pmod{n}) \pmod{n} = 1$. Da $a \in Z_n$ er $a < n$, dvs. $a \pmod{n} = a$. Dermed er $(a \cdot s \pmod{n}) \pmod{n} = 1$, som viser, at $x = s \pmod{n}$ er en løsning til $ax \pmod{n} = 1$. Eksistensen er altså vist. Entydighed af løsning: Antag, at der er to løsninger $x_1, x_2 \in Z_n$ til ligningen $ax \pmod{n} = 1$. Resterne modulo *n* er altså begge 1, dvs. $ax_1 \pmod{n} = ax_2 \pmod{n}$. Da haves $x_1 \pmod{n} = x_2 \pmod{n}$, ifølge sætning 23, og eftersom x_1 og x_2 begge er mindre end *n* fås endeligt $x_1 = x_2$. De to løsninger til ligningen er altså ens, hvilket beviser entydighed. Vi mangler at vise sætningen den anden vej.

$\Uparrow$: Vi antager, at der er en løsning x til $ax \pmod{n} = 1$, hvilket betyder, at der er et helt tal q, så $ax = q \cdot n + 1$. Men da har vi en linearkombination af a og n som er lig med 1: $x \cdot a + (-q) \cdot n = 1$. Ifølge en sætning, se opgave 32, giver det, at $(a, n) = 1$, som ønsket.

□

Eksempel 31

Lad os sige, at vi vil finde en løsning x til ligningen $35 \cdot x \pmod{5512} = 1$. Ifølge sætning 30 er der en løsning, da $(35, 5512) = 1$. Gennemgår man beviset for sætningen, afsløres også hvordan løsningen findes: Ifølge sætning 14 findes der en linearkombination af 35 og 5512, som giver 1: $s \cdot 35 + t \cdot 5512 = 1$. Rent teknisk findes s og t ved at benytte Euklids algoritme og regne baglæns, som skitseret i teksten efter eksempel 13. Gør man det, får man $s = 315$ og $t = -2$, dvs. $315 \cdot 35 + (-2) \cdot 5512 = 1$. Ifølge beviset for sætning 30 er løsningen til problemet derfor $x = s \pmod{5512} = 315 \pmod{5512} = 315$. Løsningen $x = 315$ kaldes for det *inverse element* til 35 modulo 5512.

Vi skal se endnu et eksempel på, hvordan man reducerer restberegninger. Sætning 9c) og sætning 28 er ofte brugbare ved restberegninger af potenser a^t modulo et tal n. Men hvis n er større end a og der enten gælder $(a, n) \neq 1$ eller $\varphi(n) > t$, så er sætningerne ikke brugbare. Alligevel kan man udføre små kunster for at reducere beregningsarbejdet, som vi skal se i det følgende eksempel.

Eksempel 32

Lad os sige, at vi skal udregne $2021^{35} \pmod{5671}$. Det er et ikke lille arbejde at udregne potensen 2021^{35}. Heldigvis kan beregningen stykkes op i mindre stumper, som følgende omskrivning viser, idet 2021 kaldes for m: $m^{35} = ((((m^2)^2)^2)^2 \cdot m)^2 \cdot m$. Når vi skal udregne resten af m^{35} modulo $n = 5671$, så benyttes sætning 9 gentagne gange til at konkludere, at vi kan "sætte modulus indenfor". I skemaet nedenfor starter vi indefra den inderste parentes og arbejder os udefter, idet vi hele tiden regner modulo n. Vi ser, at svaret er: $2021^{35} \pmod{5671} = 2799$.

$2021^2 \pmod{5671} = 1321$	m^2
$1321^2 \pmod{5671} = 4044$	$(m^2)^2$
$4044^2 \pmod{5671} = 4443$	$((m^2)^2)^2$
$4443^2 \pmod{5671} = 5169$	$(((m^2)^2)^2)^2$
$5169 \cdot 2021 \pmod{5671} = 567$	$(((m^2)^2)^2)^2 \cdot m$
$567^2 \pmod{5671} = 3913$	$((((m^2)^2)^2)^2 \cdot m)^2$
$(3913 \cdot 2021) \pmod{5671} = 2799$	$((((m^2)^2)^2)^2 \cdot m)^2 \cdot m$

Bemærkning 33

Omskrivningen $m^{35} = ((((m^2)^2)^2)^2 \cdot m)^2 \cdot m$ er faktisk ikke helt "tilfældig". Hvis man i stedet skriver udtrykket som $m^{35} = (((((m^1)^2 \cdot m^0)^2 \cdot m^0)^2 \cdot m^0)^2 \cdot m^1)^2 \cdot m^1$, så ser man, at rækken af eksponenter af m, regnet fra venstre er 100011, netop det binære tal for 35!

6. RSA-kryptosystemet

Det kryptosystem, som vi her skal omtale, kaldes for *RSA-kryptosystemet*, opkaldt efter *Rivest*, *Shamir* og *Adleman*, som udviklede systemet i 1977. Sikkerheden i systemet hænger kort sagt på den antagelse, at det tager meget lang tid at faktorisere et sammensat tal, som er produktet af to store primtal. I det følgende vil det blive forudsat, at læseren er bekendt med begrebet *public key kryptosystem*.

Beregningen af nøgler i RSA-systemet foregår som følger:

1. Vælg to store forskellige primtal p og q, hver på over 100 cifre, og sæt $n = p \cdot q$.
2. Udregn $\varphi(n) = (p-1)(q-1)$.
3. Vælg et helt tal e, så $0 < e < \varphi(n)$ og $(e, \varphi(n)) = 1$.
4. Beregn den entydige løsning d til $ed \pmod{\varphi(n)} = 1$ (jf. sætning 30).

Den offentlige nøgle: n og e.
Den hemmelige nøgle: d

Klarteksten, som er den ukrypterede tekst, inddeles i blokke af passende længde, således at hver blok repræsenterer et tal m, som er mindre end n.

Kryptering	**Dekryptering**
Sker ved at opløfte m til e'te potens og finde resten modulo n. Resultatet kalder vi c: $$m \rightarrow c = m^e \pmod{n}$$	Sker ved at opløfte det krypterede tal c i d'te potens og tage resten modulo n: $$c \rightarrow c^d \pmod{n}$$

Pointen er naturligvis, at hvis man først krypterer og dernæst dekrypterer, så kommer man tilbage til udgangspunktet:

$$c^d \pmod{n} = (m^e \pmod{n})^d \pmod{n} = (m^e)^d \pmod{n} = m^{ed} \pmod{n} = m$$

hvor vi i andet lighedstegn har benyttet sætning 9c). Vi mangler at vise lighedstegn nummer 4, som vil godtgøre, at vi virkelig kommer tilbage til udgangspunktet. Beviset herfor følger senere i sætning 35.

Det er værd at lægge mærke til, at de eneste tal, som bruges i krypteringen og dekrypteringen er e, n og d. Tallene p, q og $\varphi(n)$ har tjent deres formål i konstruktionen af førstnævnte tal.

Vi er nu klar til at se på et eksempel!

Eksempel 34

I dette eksempel holder vi størrelsen af primtallene nede, så regnearbejdet bliver overkommeligt.

1. Vælg primtallene $p = 53$ og $q = 107$, hvormed $n = 53 \cdot 107 = 5671$.
2. Beregn værdien af Eulers φ-funktion i 5671: $\varphi(n) = (p-1)(q-1) = 52 \cdot 106 = 5512$.
3. Vi skal vælge et positivt tal *e*, som er mindre end $\varphi(n)$ og indbyrdes primisk med $\varphi(n)$. Her er der utallige muligheder. Vi vælger $e = 35$.
4. Som den hemmelige nøgle *d* vælges løsningen til $ed \pmod{\varphi(n)} = 1$, som i dette tilfælde er $35 \cdot d \pmod{5512} = 1$. Du kan se, hvordan man finder løsningen i eksempel 31. Resultatet af beregningen er $d = 315$.

Så den offentlige nøgle er $n = 5671$ og $e = 35$, mens den hemmelige nøgle er $d = 315$.

Lad os sige, at *klarteksten* er: TUR_TIL_LONDON. Hvert bogstav skal oversættes til et tal. Man kan eventuelt gør det via den såkaldte *ASCII tabel*, som indeholder 256 forskellige tegn, nummereret fra 000 til 255. Imidlertid vil vi forsimple det lidt her og indføre vores egen lille oversættelsestabel kun indeholdende de 29 danske bogstaver, tallene fra 0 til 9 samt et mellemrumstegn _. Det er vigtigt, at alle tegn får tildelt tal med lige mange cifre, her 2. Så undertiden må der benyttes foranstillede nuller.

_	**A**	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**
00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
O	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**	**Æ**	**Ø**	**Å**
15	16	17	18	19	20	21	22	23	24	25	26	27	28	29
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**					
30	31	32	33	34	35	36	37	38	39					

Herefter inddeles klarteksten i blokke. Den maksimale blokstørrelse, vi kan benytte, er 4, idet alle de tal, de enkelte blokke repræsenterer, skal være mindre end *n*.

TU	R_	TI	L_	LO	ND	ON
2021	1800	2009	1200	1215	1404	1514

Kryptering

Krypteringen sker ifølge forrige side via $m \rightarrow m^{35} \pmod{5671}$, dvs. vi tager hver blok ovenfor, opløfter tallet til potensen 35 og udregner rester modulo 5671. Eksempel 32 viser, hvordan beregningerne kan gennemføres, illustreret på den første blok. På næste side er resultaterne af beregningerne på de 7 blokke anført.

$$2021^{35} \pmod{5671} = 2799$$
$$1800^{35} \pmod{5671} = 459$$
$$2009^{35} \pmod{5671} = 3944$$
$$1200^{35} \pmod{5671} = 896$$
$$1215^{35} \pmod{5671} = 1796$$
$$1404^{35} \pmod{5671} = 368$$
$$1514^{35} \pmod{5671} = 1507$$

Den krypterede kode er dermed: 2799 459 3944 896 1796 368 1507.

Dekryptering

Dekrypteringen sker via $c \to c^{315} \pmod{5671}$. Det er den samme type beregninger som ovenfor, og vi får følgende resultater:

$$2799^{315} \pmod{5671} = 2021$$
$$459^{315} \pmod{5671} = 1800$$
$$3944^{315} \pmod{5671} = 2009$$
$$896^{315} \pmod{5671} = 1200$$
$$1796^{315} \pmod{5671} = 1215$$
$$368^{315} \pmod{5671} = 1404$$
$$1507^{315} \pmod{5671} = 1514$$

Den dekrypterede kode er dermed: 2021 1800 2009 1200 1215 1404 1514.
Ved hjælp af den hemmelige nøgle $d = 315$ er vi altså kommet tilbage til de oprindelige talkombinationer, som via tabellen kan oversættes til teksten TUR_TIL_LONDON.

□

Vi er nu klar til at se et bevis for, at systemet virker, dvs. at man kommer tilbage til de oprindelige talblokke, når man benytter den hemmelige nøgle.

Sætning 35
Med de i RSA-systemet angivne størrelser gælder: $m^{ed} \pmod{n} = m$

Bevis: Vi splitter op i to tilfælde: Når $(m,n) = 1$ og $(m,n) \neq 1$. Antag først, at $(m,n) = 1$. Da fås ifølge sætning 28: $m^{ed} \pmod{n} = m^{ed \pmod{\varphi(n)}} \pmod{n} = m^1 \pmod{n} = m$, hvor vi i andet lighedstegn har benyttet egenskab 4 for RSA: $ed \pmod{\varphi(n)} = 1$. Sidste lighedstegn fås da $m < n$. Lad os til slut se på tilfældet $(m,n) \neq 1$, som er noget mere kompliceret. Da $n = p \cdot q$ er et produkt af to primtal, så må enten p eller q være en divisor i m, for

ellers kan m og n ikke have nogen fælles divisor større end 1. Vi kan ikke både have, at p og q går op i m, for så vil også $n = p \cdot q$ gå op i m, i modstrid med, at $m < n$. Vi kan uden indskrænkning antage, at p går op i m, og at q ikke går op i m. Sidstnævnte betyder, at $(q,m) = 1$, da q er et primtal. Det er nok at vise, at $m^{ed} \pmod p = m \pmod p$ og $m^{ed} \pmod q = m \pmod q$, for så haves $m^{ed} \pmod n = m \pmod n = m$, ifølge Lemma 36 nedenfor. Den første restberegning er nem: Da $p \mid m$ haves

$$m^{ed} \pmod p \;=\; (m \pmod p)^{ed} \pmod p \;=\; 0^{ed} \pmod p \;=\; 0 \;=\; m \pmod p$$

hvor sætning 9c) er benyttet i det andet lighedstegn. Den anden restberegning er lidt sværere: Da $(q,m) = 1$, kan *Fermats lille sætning* (se opgave 55) anvendes til at slutte, at $m^{q-1} \pmod q = 1$. Da $ed \pmod{\varphi(n)} = 1$, findes et helt tal k, så $ed = k \cdot \varphi(n) + 1$. Det giver alt i alt, at

$$m^{ed} \;=\; m^{k\cdot\varphi(n)+1} \;=\; m \cdot m^{k\cdot\varphi(n)} \;=\; m \cdot m^{k\cdot(p-1)(q-1)} \;=\; m \cdot (m^{q-1})^{k\cdot(p-1)}$$

og tages rester modulo q og anvendes sætning 9, fås:

$$\begin{aligned} m^{ed} \pmod q \;&=\; \Big[\, m \pmod q \cdot (m^{q-1} \pmod q)^{k\cdot(p-1)} \pmod q \,\Big] \pmod q \\ &=\; \Big[\, m \pmod q \cdot 1^{k\cdot(p-1)} \pmod q \,\Big] \pmod q \\ &=\; m \pmod q \end{aligned}$$

Hvormed det ønskede er vist.

□

Lemma 36

Lad p og q være to forskellige primtal. Da gælder:

$$x \pmod p = a \pmod p \;\wedge\; x \pmod q = a \pmod q \quad\Leftrightarrow\quad x \pmod{pq} = a \pmod{pq}$$

Bevis: Sætningen siger, at hvis to tal har samme rest modulo et primtal og de to tal også har samme rest modulo et andet primtal, så har de to tal også samme rest modulo produktet af de to primtal. Sætningen følger umiddelbart af sætning 22:

$$\begin{aligned} & x \pmod p = a \pmod p \;\wedge\; x \pmod q = a \pmod q \\ \Updownarrow\;& \\ & p \mid (x-a) \;\wedge\; q \mid (x-a) \\ \Updownarrow\;& \\ & pq \mid (x-a) \\ \Updownarrow\;& \\ & x \pmod{pq} \;=\; a \pmod{pq} \end{aligned}$$

Den anden biimplikation fås, idet p er en primfaktor i $(x-a)$ og q er en anden primfaktor i $(x-a)$ hvis og kun hvis $pq \mid (x-a)$. Det viser det ønskede.

□

Generering af primtal

I RSA-systemet skal der genereres to store primtal, som faktorer i n. Det er faktisk et stort arbejde at afgøre om et givet stort tal med sikkerhed er et primtal. Imidlertid findes der tests, som med meget stor sikkerhed, om end ikke 100%, kan afgøre, om et tal er et primtal. En meget effektiv test er den såkaldte *Rabin-Miller test*. Med blot få tests, vil kun uhyre få tal passere primtalstesten, selvom de faktisk er sammensatte tal. Det vil føre for vidt at komme ind på detaljer her!

Litteratur

Peter Landrock, Knud Nissen. *Kryptologi – fra viden til videnskab*. Forlaget ABACUS, 1997.

Opgaver

Opgave 20

Afgør om følgende divisioner går op, og angiv i så fald kvotienten. Hvis divisionen ikke går op, angiv da både kvotient og rest. Skriv i alle tilfælde resultatet af divisionen på formen $b = q \cdot a + r$.

a) 120 divideret med 20.
b) 182 divideret med 7.
c) 67 divideret med 8.
d) 104 divideret med 21

Opgave 21

Benyt metoden i eksempel 3 til at bestemme kvotient og rest ved nedenstående divisioner. Opskriv desuden resultatet på formen $b = q \cdot a + r$.

a) 4382 divideret med 27.
b) 178642596 divideret med 5061.
c) –7524 divideret med 62 (her skal du tilrette metoden en smule, da det, der divideres op i, er negativt. Overvej!).

Opgave 22

Prøv i tilfældet $b \geq 0$ at bevise, at metoden angivet i eksempel 3 altid vil fungere.

Opgave 23

Bevis sætning 4d). *Hjælp*: Skriv ned, hvad du ved, og hvad du skal vise.

Opgave 24

Bevis sætning 9b) ved at imitere idéerne fra beviset for sætning 9a).

Opgave 25

Benyt sætning 9 til at udregne følgende rester med Texas 89.

a) $(651+1721)(\mathrm{mod}\ 23)$
b) $(7183 \cdot 89512)(\mathrm{mod}\ 127)$
c) $(87 \cdot 973401)(\mathrm{mod}\ 766)$
d) $87^7\ (\mathrm{mod}\ 14)$
e) $285^6\ (\mathrm{mod}\ 65)$
f) $(5410542+670127)(\mathrm{mod}\ 7521)$
g) $(8702-6211)(\mathrm{mod}\ 45)$
h) $11111^{11}\ (\mathrm{mod}\ 11)$
i) $(61 \cdot 910)(\mathrm{mod}\ 51)$
j) $(8721675 \cdot 921344)(\mathrm{mod}\ 8551)$

Opgave 26 (sværere)

Udregn $(618238178981990 \cdot 779892001331475)(\text{mod } 348812)$ ved hjælp af Texas 89.

Hjælp: Du må dele op i blokke, fordi Texas 89 ikke kan regne med så mange cifre.

Opgave 30

Bestem følgende største fælles divisorer, eventuelt ved benyttelse af sætning 12.

a) $(123,18)$
b) $(8126,54)$
c) $(123876,27962)$
d) $(73831,30073)$
e) $(5661,1950)$

Opgave 31

Benyt Euklids algoritme til at bestemme den største fælles divisor $(6699,364)$ og bestem samtidigt den linearkombination af 6699 og 364, som er lig med største fælles divisor, jf. sætning 14 og teksten før den.

Opgave 32

Lad $a,b \in Z$ og antag, at der findes hele tal s og t, så $s \cdot a + t \cdot b = 1$. Vis, at så er $(a,b) = 1$. *Hjælp*: Antag, at den største fælles divisor *ikke* er lig med 1 og udnyt sætn. 4.

Opgave 33

Benyt sætning 16 til at vise, at hvis p er et primtal, så gælder følgende implikation: $p \mid a_1 \cdot a_2 \cdot \ldots \cdot a_n \Rightarrow p \mid a_1 \vee p \mid a_2 \vee \ldots \vee p \mid a_n$.

Opgave 34

Benyt metoden fra teksten efter eksempel 13 til at bestemme en linearkombination af a og b, som er lig med største fælles divisor mellem de to tal: $s \cdot a + t \cdot b = (a,b)$:

a) $a = 14,\ b = 38$ b) $a = 68,\ b = 863$

Opgave 40

Anvend sætning 18 til at afgøre, om følgende tal er primtal:

a) 151 b) 2089 c) 457589 d) 3299 e) 7147 f) 25343
g) 3861653 h) 2209 i) 419

Opgave 41

Nedenstående tal er sammensatte tal. Foretag en primfaktoropløsning af tallene.

a) 851 b) 14105 c) 56356 d) 869847 e) 126149 f) 6511319

Opgave 50

Bevis de to specialtilfælde fra sætning 25:

a) Hvis n er et produkt af to *forskellige* primtal p og q er $\varphi(pq) = (p-1)(q-1)$.
b) Hvis n er et primtal p er $\varphi(p) = p-1$.
c) Hvis n er kvadratet på et primtal p er $\varphi(p^2) = p \cdot (1-p)$.

Hjælp: Udnyt, at hvis to tal *ikke* er indbyrdes primiske, så er det fordi de har mindst en fælles primfaktor. Overvej hvilke tal, der kan være et problem og må sorteres fra Z_n.

Opgave 51

a) Bestem Z_7^* b) Bestem Z_{10}^* c) Bestem Z_{12}^* d) Bestem Z_{30}^*

Opgave 52

Benyt sætning 25 til at udregne:

a) $\varphi(5)$ b) $\varphi(21)$ c) $\varphi(62)$ d) $\varphi(49)$ e) $\varphi(407)$

Opgave 53

Benyt sætning 9 og/eller sætning 28 til at udregne følgende rester:

a) $17^{23} \pmod 6$ b) $11^{25} \pmod{21}$ c) $8^{103} \pmod{97}$ d) $5^{14} \pmod 3$
e) $51^{22} \pmod{22}$ f) $104^{297} \pmod{49}$

Opgave 54

Benyt fremgangsmåden fra eksempel 31 til at udregne $6372^{22} \pmod{7249}$. Du må gerne benytte grafregneren til at udregne et tal modulo et andet tal. $a \pmod n$ indtastes som mod(a,n).

Opgave 55 (Fermats lille sætning)

Bevis *Fermats lille sætning*, som siger, at hvis p er et primtal og $(a, p) = 1$, så gælder: $a^{p-1} \pmod p = 1$. *Hjælp*: Benyt sætningerne 25 og 27.

Opgave 56 (Inverst element)

Benyt metoden anført i eksempel 31 til at finde løsningerne til:

a) $52 \cdot x \pmod{317} = 1$ b) $11 \cdot x \pmod{59} = 1$

Kommentar: Lad $a \in Z_n$ og $(a, n) = 1$. Løsningen $x \in Z_n$ til ligningen $a \cdot x \pmod{n} = 1$ kaldes da for det *inverse element til a modulo n*.

Opgave 60 (RSA med grafregneren)

I denne opgave skal du ved hjælp af grafregneren prøve at kryptere og dekryptere efter fremgangsmåden i RSA, anvist i eksempel 34. Du skal bruge $n = p \cdot q = 47 \cdot 137 = 6439$.

a) Vis, at $\varphi(n) = 6256$.
b) Vis, at $e = 21$ kan benyttes, dvs. at $(e, \varphi(n)) = 1$.
c) Vis, at den hemmelige nøgle er $d = 3277$. Denne del er lidt svær: Du skal finde løsningen d til ligningen $e \cdot d \pmod{\varphi(n)} = 1$, her $21 \cdot d \pmod{6256} = 1$. Det gøres ved at bruge Euklids algoritme og regne baglæns for at finde en linearkombination af 21 og 6256, som er lig med 1: $s \cdot 21 + t \cdot 6256 = 1$. Teknikken er beskrevet efter eksempel 13. Når s og t er fundet, fås den hemmelige nøgle som $d = s \pmod{6256}$.
d) Lad os sige, at vi skal kryptere ordet TI. Benyt oversættelsestabellen fra eksempel 34 til at finde det firecifrede tal m, som ordet svarer til.
e) Krypter ved at benytte $m \rightarrow m^e \pmod{n}$, dvs. her $m \rightarrow m^{21} \pmod{6439}$. Restberegningen skal reduceres ved at benytte metoden vist i eksempel 32. Du kan benytte omskrivningen $m^{21} = ((((m^2)^2 \cdot m)^2)^2 \cdot m$. Resultatet af restberegningen er den krypterede tekst c. Du skal gerne få $c = 4167$.

Kommentar: Hvis man skal dekryptere koden ovenfor, så skal $c \rightarrow c^d \pmod{n}$, som her svarer til $c \rightarrow c^{3277} \pmod{6439}$, benyttes. Du skal slippe for det her, da regningerne er ret omfattende. Har du adgang til programmet *Derive*, så kan du foretage alle de beregningstunge elementer i RSA-systemet lynhurtigt. Se opgave 58.

Opgave 61 (RSA med programmet *Derive*)

I det følgende skal du gennemføre kryptering og dekryptering ved hjælp af RSA-kryptosystemet. Du skal benytte programmet *Derive* til at foretage de mange beregningstunge skridt, så du kan koncentrere dig om at forstå de enkelte trin, som indgår i anvendelsen af RSA-kryptosystemet.

Programmet *Derive* er ret simpelt at bruge. Man indtaster en kommando i indtastningslinjen forneden. Resultatet af kommandoen vises i historikområdet ovenfor indtastningslinjen, når man klikker på ikonen, som viser et flueben med et lighedstegn under (eller bruge genvejskombinationen Ctrl+Enter). Man kan hente elementer fra historik-

området ned ved at markere dem og trykke på Enter-knappen. Nedenfor er angivet syntaksen for de kommandoer, som er relevante i forbindelse med talteori og RSA.

- next_prime(n) beregner det første primtal større end tallet *n*.
- previous_prime(n) beregner det største primtal, der er mindre end tallet *n*.
- divisors(n) giver en liste over alle divisorer i tallet *n*.
- euler_phi(n) udregner $\varphi(n)$.
- factor(n) primfaktoriserer tallet *n*.
- mod(a,n) udregner $a \pmod{n}$.
- power_mod(a,m,n) udregner $a^m \pmod{n}$.
- inverse_mod(a,n) udregner løsningen x til ligningen $a \cdot x \pmod{n} = 1$, under forudsætning af at $(a,n) = 1$. Løsningen *x* kaldes da for det inverse element til *a* modulo *n*.
- gcd(a,b) beregner den største fælles divisor mellem *a* og *b*.
- extended_gcd(a,b) beregner tre tal: den største fælles divisor *d* mellem *a* og *b* samt koefficienterne *s* og *t* i linearkombinationen af a og *b*, som er lig med *d* (jf. sætning 14): $s \cdot a + t \cdot b = d$. Resultatet skrives på formen: $\left[d,\left[s,t\right]\right]$.

a) Du skal generere et primtal på ca. 17 cifre. Det gøres nemmest ved at vælge et vilkårligt 17-cifret tal *t* og skrive det på *t*'s plads i kommandoen p:= next_prime(t). Afslut med Ctrl+Enter, eller klik på ikonen med fluebenet med et lighedstegn under). Da vises værdien af det genererede primtal! Det smarte ved at skrive p:= foran den egentlige kommando er, at så tildeles værdien til variablen *p*, således, at man fra nu af blot kan skrive *p* i stedet for tallet selv. Generer på samme måde et primtal *q* på ca. 17 cifre.

b) Udregn dernæst $n = p \cdot q$ gennem kommandoen n:=p*q. Benyt igen Ctrl+Enter.

c) Beregn dernæst $\varphi(n)$ og tildel det til en variabel φ ved at skrive phi:=(p-1)*(q-1). Afslut med Ctrl+Enter.

d) Nu skal vælges et tal *e*, så $0 < e < \varphi(n)$ og så $(e,\varphi(n)) = 1$. Man plejer at vælge et ret lille tal, for at optimere krypteringshastigheden. Hvis man vælger et primtal, er der ekstra stor sandsynlighed for, at det som påkrævet er indbyrdes primisk med $\varphi(n)$. Derfor kan det være fornuftigt at skrive e:=next_prime(t), hvor der skrives et lille, fx trecifret tal, på t's plads. Afslut med Ctrl+Enter. Du skal imidlertid være sikker på, at din genererede værdi for *e* er indbyrdes primisk med $\varphi(n)$. Det kan gøres med kommandoen gcd(e,phi), efterfulgt af Ctrl+Enter. Hvis resultatet er 1, er dit valgt godt, ellers må du vælge om.

e) Den hemmelige nøgle *d* skal nu genereres. Vi skal finde *d, så* $0 < d < \varphi(n)$ og så $ed \pmod{\varphi(n)} = 1$. Løsningen *d* kaldes det *inverse element til e modulo* $\varphi(n)$. Da $(e,\varphi(n)) = 1$, findes der ifølge sætning 14 en linearkombination af *e* og $\varphi(n)$, som er lig med 1: $s \cdot e + t \cdot \varphi(n) = 1$. *d* kan da bestemmes ved $d = s \pmod{\varphi(n)}$. I *Derive* kan vi finde *s* og *t* ved kommandoen extended_gcd(e,phi). Skriv det efterfulgt af Ctrl+Enter. Herefter markerer du den del af resultatet, som svarer til *s* (se ovenfor), kopierer den over i udklipsholderen og skriver så kommandoen d:=mod(s,phi) med indholdet af udklipsholderen anbragt på *s*'s plads. Afslut med Ctrl+Enter.

f) Opskriv den offentlige og den hemmelige nøgle, gerne i et tekstfelt via værktøjet *Insert Text*. Dette er bare forklarende tekst, som ikke indgår i beregninger.

g) Før vi kan kryptere klarteksten skatten_er_i_hulen, skal du oversætte tegnene til tal efter tabellen i eksempel 34, og anbringe dem i en lang række uden mellemrum. Del herefter rækken op i to dele og tildel herefter i *Derive* den venstre halvdel til m1 og den højre halvdel til m2. Man deler op i blokke for at sikre sig, at de tal m, der krypteres, er mindre end n.

h) Vi er nu klar til at kryptere. Det skal gøres med $m \rightarrow m^e \pmod{n}$. Kommandoen power_mod er lige den, vi behøver. Skriv c1:=power_mod(m1,e,n), efterfulgt af Ctrl+Enter. Tilsvarende gøres med m2.

i) Nu skal du prøve at dekryptere den kode, du lige har genereret. Det sker naturligvis med den hemmelige nøgle: dekryp1:=power_mod(c1,d,n). Tilsvarende med c2. Kontroller, at du er kommet tilbage til udgangspunktet, dvs. at dekryp1 = m1 og dekryp2 = m2.

j) Nu afhænger RSA-systemets sikkerhed som bekendt af, at det er meget svært for en udenforstående at faktorisere n i de to (ukendte) primfaktorer p og q. Kommandoen factor(n) kan bruges til at teste dette. Prøv det! NB! Størrelsen af primtallene er med vilje valgt så store, så det er på kanten til at en PC i dag kan klare faktoriseringen. Så alt afhængig af de tal, du har valgt, og hastigheden af den computer du har, kan det være, at faktoriseringen kan klares i løbet af et par minutter eller ej. Havde vi benyttet bare lidt større primtal, ville det have været en håbløs opgave for en almindelig PC.

Opgave 62 (RSA med programmet *Derive*)

Alice har sendt en meddelelse til sin kæreste Bob. Den er krypteret med Bob's offentlige nøgle, som alle og enhver i princippet kan finde frem til:

$$n = 467709662065084282650528569047994333785 9 \text{ og } e = 6761$$

Meddelelsen er delt op i to blokke og lyder:

$$5904870343956547253626814656501705088 2$$
$$3474257596068090511069733720631488290341$$

Lad os sige, at du er Bob og ønsker at dekryptere den kodede meddelelse med din egen hemmelige nøgle, som er $d = 178478173070243669457530540704783305484 1$. Foretag dekrypteringen med $c \rightarrow c^d \pmod{n}$ for eksempel i programmet *Derive*. Oversæt herefter hvert par af tal til bogstaver efter tabellen i eksempel 34. Hvad siger meddelelsen fra Alice?

NB! Alle ved, at blokkene har længde 25 før kryptering, så hvis du får et tal med færre cifre, skal der sættes 0'er foran!