Kryptologi og krypteringssystemer

Henning E. Andersen
Institut for Matematiske Fag
Aalborg Universitet
Fredrik Bajers Vej 7G
9220 Aalborg Ø
Email: henning@math.auc.dk

Indhold

Om disse arbejdsark

Disse noter omhandler kryptografi og nogle af de tilhørende krypteringssystemer, herunder klassiske (symmetriske) systemer repræsenteret ved mono- og polyalfabetiske systemer og public-key (asymmetriske) systemer repræsenteret ved RSA.

Rent matematisk er hovedvægten i noterne lagt på at give eleverne en forståelse for den talteori, der ligger til grund for RSA systemet. De mere historiske aspekter af kryptografien er udeladt, idet noterne ellers ville blive meget omfattende, men bagerst i noterne findes en litteraturliste og linksamling, hvor man som underviser kan hente inspiration til de mere historiske sider af kryptografien. Specielt kan flere af historierne i *Kodebogen* af Simon Singh anbefales som "krydderi" på undervisningen.

Udover gennemgang af matematikken bag RSA, så indeholder notesættet en del forslag til elevopgaver, der er forholdsvis tæt knyttede til strukturen i gennemgangen af stoffet. Det er således forsøgt at formulere opgaver, der understøtter forklaringerne givet i notesættets teoriafsnit og øger elevernes forståelse af disse.

Noterne er skrevet med henblik på anvendelse i forbindelse med 10 - timers forløbene under standardforsøget i den nye gymnasiereform, men alene deres omfang gør, at underviseren må udvælge hvilke dele af notesættet, der skal indgå i et sådant forløb og i hvilken detaljegrad.

Flere af beviserne i notesættet er matematisk krævende, så disse kan enten udelades eller overlades til en foredragsholder udefra - alt efter underviserens temperament og elevernes matematiske modenhed.

Notesættet er således hovedsageligt tænkt anvendt i mindre forløb kombineret med besøg af en foredragsholder fra et universitet eller lignende. Foredragsholderen kan udover at præsentere eksempelvis RSA i større sammenhænge, gennemgå vanskelige beviser, o.l. ligeledes medvirke i gruppearbejdet med eksempelvis praktisk anvendelse af Vigenére og/eller RSA eller løsning af opgaver fra notesættet.

Ønskes en foredragsholder fra Institut for Matematiske Fag ved Aalborg Universitet, så kan en sådan bestilles via hjemmesiden:

http://www.math.auc.dk/events/highschool.htm.

De matematiske institutter ved landets øvrige universiteter tilbyder muligvis lignende ordninger, der kan tilgås via de respektive universiteters hjemmesider.

Sidst men ikke mindst, så vil jeg gerne sige tak til Elisabeth Husum, Hobro Gymnasium, og Ulla Folkmann, Viborg Handelsskole, for gode råd, opgaveforslag, o.s.v. Det har været en fornøjelse at samarbejde med jer omkring tilblivelsen af disse noter, der ikke havde haft samme kvalitet uden jeres hjælp.

Aalborg d. 12. december 2003

Henning E. Andersen

Herunder gives eksempler på forløb, der kunne tænkes/er blevet gennemført (side n linie m fra oven skrives n^m og side n linie m fra neden skrives n_m).

Forløb 1 (eksperimenterende):
I dette forløb er det ideen slet ikke at komme ind på, hvordan man beregner største fælles divisor, men blot fortælle, at der findes algoritmer til det brug. Forløbet omfatter således siderne: 4 - 13^{12} (Beviserne for sætning 4.3 og 4.5 springes over.), 20 - 21^{16}, 22_{14} - 25_7, 27_4 - 28^{19} og 29 - 30^8. Der suppleres med et passende antal opgaver.

Forløb 2 (mere teoretisk):
Forløbet omfatter siderne: 4^1 - 4_3, 7^3 - 18^{14}, 20 - 21^{16}, 22_{14} - 25_7, 27_4 - 28_{10} og 29 - 30^8. Der suppleres med et passende antal opgaver.

1 Hvad er kryptologi?

Ordet *kryptologi* er en fællesbetegnelse for *kryptografi* og *kryptoanalyse*, der hver især kan beskrives på følgende måde:

Kryptografi handler om at forvandle en klartekst til en ulæselig (krypteret) tekst, der kun kan læses af den person, som teksten er tiltænkt.

Kryptoanalyse handler derimod om at bryde en krypteret tekst, så den kan læses af andre end den person, som teksten var tiltænkt.

En kryptograf og en kryptoanalytiker har altså modsatrettede interesser, når det gælder krypteret tekst. Kryptografen ønsker således at krypteringen er tilstrækkelig stærk til at den krypterede tekst ikke kan læses af uvedkommende, mens krytoanalytikeren netop er interesseret i at kunne læse en krypteret tekst, der er tiltænkt en anden person - altså bryde krypteringen.

I det efterfølgende vil vi hovedsageligt kun beskæftige os med kryptografi og den bagvedliggende matematik, idet dette emne i sig selv er ganske stort.

2 Mono- og polyalfabetiske krypteringssystemer

Tidlige krypteringssystemer bygger hovedsageligt på substitution, hvor krypteringen består i, at de enkelte bogstaver i klarteksten systematisk erstattes med et andet bogstav i alfabetet. Disse substitutioner kan foretages med bogstaver fra ét alfabet (kaldes monoalfabetiske kryperingssystemer) eller med bogstaver fra flere alfabeter (kaldes polyalfabetiske krypteringssystemer). Dekrypteringen består i begge tilfælde af en "omvendt" ombytning af bogstaverne i den krypterede tekst.

Eksempel 2.1

Her vises et eksempel på anvendelse af et monoalfabetisk krypteringssystem. Lad os forestille os, at vi ønsker at kryptere teksten *Snehvide og de syv små dværge* ved hjælp af et monoalfabetisk krypteringssystem.

Først vælges en forskydning - her vælger vi en forskydning på 4 bogstaver, men en hvilken som helst forskydning mellem 1 og 29 kan anvendes. Dernæst skrives det sædvanlige alfabet og det forskudte alfabet over hinanden:

```
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z Æ Ø Å
E F G H I J K L M N O P Q R S T U V W X Y Z Æ Ø Å A B C D
```

Nu krypteres klarteksten (mellemrum ignoreres) ved at substituere bogstaverne 'A' med 'E', 'B' med 'F', o.s.v., som nedenfor:

Klartekst	`SNEHVIDE OG DE SYV SMÅ DVÆRGE`
Krypteret tekst	`WRILZMHI SK HI WÅZ WQD HZBVKI`

Den krypterede tekst (eventuelt uden mellemrum) kan derefter sendes og senere dekrypteres ved hjælp af den omvendte substitution - dette kræver blot at modtageren kender forskydningen. ▲

Monoalfabetiske krypteringssystemer, som det, der blev skitseret i eksempel 2.1 ovenfor, er dog særdeles sårbare, idet hvis man kender perioden, så kan enhver

læse den krypterede tekst. Med en computer tager det således ikke lang tid at afprøve samtlige muligheder for forskydning, når blot man ved hvilket sprog meddelelsen er skrevet på.

Et polyalfabetisk krypteringssystem bygger ligeledes på substitution af bogstaver, men i modsætning til et monoalfabetisk krypteringssystem, så skifter man i et polyalfabetisk krypteringssystem forskydning under krypteringen på en systematisk måde. Dette bevirker, at et polyalfabetisk krypteringssystem ikke har de svagheder, som et monoalfabetisk krypteringssystem har, idet udskiftningen af forskydning undervejs gør det vanskeligere at bryde krypteringen.

Det skal dog nævnes, at et polyalfabetisk krypteringssystem har andre svagheder og derfor ikke er sikkert at anvende i praksis.

Eksempel 2.2

Her vises et eksempel på anvendelse af et polyalfabetisk krypteringssystem. Lad os igen forestille os, at vi ønsker at kryptere teksten *Snehvide og de syv små dværge*, men denne gang ved hjælp af et polyalfabetisk krypteringssystem.

Først vælges et kodeord, der skal angive vores skift af forskydning - her vælger vi kodeordet *mobil*. Dernæst skrives det sædvanlige alfabet og samtlige 29 mulige forskydninger af alfabetet op oven over hinanden i et såkaldt *Vigenére tableau*:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
Æ	Æ	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
Ø	Ø	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ
Å	Å	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	Æ	Ø

Herefter skrives nøglen (med et passende antal gentagelser) og klarteksten (uden mellemrum) over hinanden:

Nøgle (gentaget)	`MOBILMOBILMOBILMOBILMOBI`
Klartekst	`SNEHVIDEOGDESYVSMÅDVÆRGE`

Nu krypteres klarteksten ved at substituere de bogstaver, der forekommer i klarteksten, med bogstaverne i den række i Vigenére tableauet, der er markeret med det tilsvarende bogstav i kodeordet og den søjle, som svarer til bogstavet i klarteksten.

Eksempelvis substitueres det første 'S' med det tilsvarende bogstav i rækken markeret med 'M' og søjlen markeret med 'S' - det vil sige, at det første 'S' byttes ud med 'B'. Dernæst substitueres 'N' med det tilsvarende bogstav i rækken markeret med 'O' og søjlen markeret med 'N' - det vil sige, at 'N' udskiftes med 'Ø'. Fortsættes på denne måde, fås følgende kryptering:

Nøgle (gentaget)	`MOBILMOBILMOBILMOBILMOBI`
Klartekst	`SNEHVIDEOGDESYVSMÅDVÆRGE`
Krypteret tekst	`BØFPDURFWRPSTDDBÆALDJCHM`

Her kan teksten dekrypteres, når blot man kender kodeordet eller de skiftende perioder. ▲

En af svaghederne ved polyalfabetiske krypteringssystemer er, at de kan brydes ved at anvende det faktum, at de enkelte bogstaver og stavelser ikke bliver anvendt lige ofte i sproget.

Opgave 2.1

Kryptér sætningen *Vi sender en krypteret meddelelse* ved at benytte en forskydning på 7 og et monoalfabetisk system som i eksempel 2.1.

Opgave 2.2

Hvilken forskydning er der brugt i et monoalfabetisk system til kryptering af ordet *radio*, når det oplyses at den krypterede tekst er *fruzc*?

Opgave 2.3

Vælg en forskydning imellem 1 og 28 (begge inkl.) og kryptér dit navn med den valgte forskydning og et monoalfabetisk krypteringssystem.

Byt nu resultat af krypteringen med din sidemand **uden** at fortælle, hvilken forskydning du har anvendt, og prøv derefter, om du kan finde den forskydning, som din sidemand valgte til at kryptere hans/hendes navn med.

Opgave 2.4

Kryptér sætningen *Vi sender en krypteret meddelelse* ved at benytte ordet *computer* som kodeord og et polyalfabetisk krypteringssystem som i eksempel 2.2. Du kan anvende Vigenére tableau'et i eksemplet ved løsning af opgaven.

Opgave 2.5

Diskutér ulemper ved at anvende et polyalfabetisk krypteringssystem og find eventuelt forslag til metoder til at bryde et polyalfabetisk krypteringssystem.

3 *Public key* - krypteringssystemer

I afsnit 2 så vi eksempler på både mono- og polyalfabetiske krypteringssystemer, der alle havde én ting til fælles: Der anvendes samme nøgle til både kryptering og dekryptering. Den slags systemer kaldes *symmetriske* eller *konventionelle*.

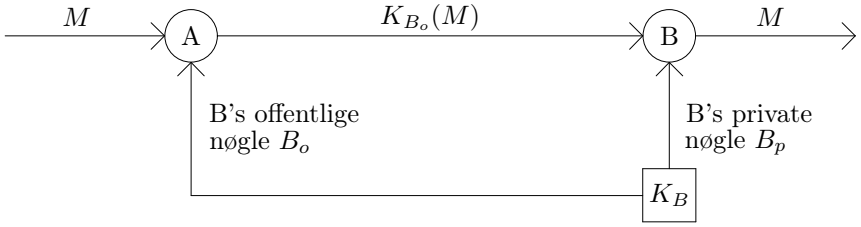
I 1976 opfandt W. Diffie og M. E. Hellman et krypteringssystem, som de kaldte et *Public key* - krypteringssystem. Ideen bygger på eksistensen af matematiske funktioner f, der er hurtige at beregne, mens den inverse funktion f^{-1} er meget svær at finde. Sådanne matematiske funktioner kaldes *trapdoor one-way functions*.

Denne ide var der mange matematikere, der i slutningen af firserne indså kunne bruges til kryptering, idet man uden risiko kan offentliggøre funktionen f, når den inverse funktion f^{-1} er nærmest umulig at finde selvom man kender f.

I et *asymmetrisk* eller *public key* - krypteringssystem genereres der således et nøglesæt bestående af 2 nøgler: en offentlig nøgle, som alle har adgang til, og en privat nøgle, som kun modtageren kender.

Når en sender, A, ønsker at sende en krypteret meddelelse til en modtager, B, så krypterer A meddelelsen M med B's offentlige nøgle B_o og får den krypterede meddelelse $K_{B_o}(M)$, som derefter sendes.

Ved modtagelsen dekrypterer B meddelelsen $K_{B_o}(M)$ med sin private nøgle B_p og får $D_{B_p}(K_{B_o}(M)) = M$. Skematisk ser brugen af et *public key* - krypteringssystem således ud:



For at dette skal fungere i praksis, så skal det være umuligt at finde den private nøgle ud fra den offentlige nøgle - også selvom man har adgang til både klartekst og den krypterede tekst. Med andre ord, så skal den offentlige nøgle "definere" en *trapdoor one-way function*. Og samtidig skal man med den private nøgle nemt kunne dekryptere en meddelelse ved modtagelsen.

I de efterfølgende afsnit skal vi se nærmere på, hvordan denne ide realiseres rent matematisk i *public key* - krypteringssystemet RSA, der bygger på følgende faktum:

> *Ethvert helt tal større end 1 kan entydigt skrives som et produkt af primtal (dvs. entydigt pånær rækkefølgen af primfaktorerne).*

I RSA udnyttes det, at det er forholdsvis overkommeligt at gange tal sammen (også selvom de er store), mens det er nærmest umuligt at finde primfaktorerne, der omtales ovenfor, når tallene bliver bare lidt store.

Resten af disse noter vil således omhandle matematikken bag RSA krypteringssystemet selvom der findes mange andre *trapdoor one-way functions* og *public-key* krypteringssystemer, som vi kunne beskæftige os med.

4 Heltalsdivision og division med rest

Et af de grundlæggende begreber bag *public key* - krypteringssystemer er heltalsdivision og division med rest. Lad os først definere, hvad vi mener med en divisor:

Definition 4.1

Et helt tal n siges at være ***divisibelt*** *med et helt tal d, hvor $d \neq 0$, hvis der eksisterer et helt tal q således at*

$$n = qd$$

Dette skrives $d|n$. Tallet d kaldes ***divisoren*** *og q kaldes* ***kvotienten.***

I virkeligheden er definition 4.1 bare en formel beskrivelse af noget, som vi udmærket kender fra folkeskolens mindste klasser. Lad os se, hvad definitionen egentlig betyder ved hjælp af et eksempel:

Eksempel 4.2

Lad $n = 12$ og $d = 4$ i definition 4.1. Da er 12 divisibelt med 4, idet der eksisterer et helt tal $q = 3$, så

$$n = qd,$$

hvilket i dette tilfælde er det samme som

$$12 = 3 \cdot 4.$$

Det vil sige, at 4 går op i 12, hvilket skrives $4|12$. Tallet 4 er i dette eksempel divisoren og 3 er kvotienten.

Lad nu $n = 17$ og $d = 3$ i definition 4.1. Da er 17 ikke divisibelt med 3, idet vi ikke kan finde et helt tal q, så

$$17 = q \cdot 3.$$

Det vil sige, at 3 ikke går op i 17, hvilket skrives $3 \nmid 17$. ▲

Definition 4.1 beskriver altså, hvad vi mener med, at et helt tal n er divisibelt med et andet helt tal $d \neq 0$ (eller at d går op i n), men hvad nu, hvis d ikke går op i n som i tilfældet med $n = 17$ og $d = 3$ i eksempel 4.2? Så må vi indføre begrebet **rest** ved heltalsdivision:

Sætning 4.3

Lad n og d være hele tal, hvor $d \neq 0$. Da eksisterer der hele tal q og r således, at

$$n = qd + r$$

*Tallet d kaldes **divisoren**, tallet q kaldes **kvotienten** og tallet r kaldes en **rest** ved division af n med d.*

Bevis:

Givet hele tal n og $d \neq 0$, så kan vi altid vælge $q = 1$ og $r = n - d$, hvilket medfører, at

$$qd + r = d + n - d = n$$

som ønsket. Sætningen er dermed bevist. □

Bemærk, at hvis $d|n$ i sætning 4.3, så kan q vælges, så $r = 0$.

Eksempel 4.4

Lad os se på tilfældet $n = 17$ og $d = 3$ som sidst i eksempel 4.2. I følge sætning 4.3 så kan vi finde hele tal q og r, så

$$17 = q \cdot 3 + r.$$

Eksempelvis vil $q = 4$ og $r = 5$ opfylde sætningen, idet

$$4 \cdot 3 + 5 = 12 + 5 = 17,$$

men vi kunne også vælge $q = 6$ og $r = -1$, idet

$$6 \cdot 3 + (-1) = 18 - 1 = 17.$$

Det vil sige, at både 5 og -1 kan være rest ved division af 17 med 3. ▲

Eksempel 4.4 viser, at hverken resten eller kvotienten ved division af et helt tal med et andet er entydigt bestemte. Dette kan sikres ved at stille lidt flere krav, så der gælder følgende resultat:

Sætning 4.5

*Lad n og d være vilkårlige hele tal, hvor $d > 0$. Da eksisterer der **entydigt bestemte** hele tal q og r således, at*

$$n = qd + r,$$

*hvor $0 \leqslant r < d$. Tallet r kaldes **den principale rest** ved division af n med d.*

Bevis:
Givet hele tal n og $d > 0$, så findes der ifølge sætning 4.3 hele tal q og r, så $n = qd + r$, hvilket kan omskrives til $r = n - qd$.

Betragt nu mængden

$$S_n = \Big\{ n + k \cdot d \;\Big|\; k \in \mathbb{Z} \Big\}.$$

Mængden S_n må indeholde et ikke-negativt tal, idet hvis $n < d$, så findes et helt tal k, så $n + kd \geqslant 0$, og hvis $n \geqslant d$, så er $n - d \geqslant 0$.

Lad nu r være det mindste element i mængden S_n, hvor $r \geqslant 0$ og lad q være et helt tal, så $r = n - qd$.

Hvis vi antager at $r \geqslant d$, så er

$$\begin{aligned} n - (q+1)d &= n - qd - d \\ &= r - d \\ &\geqslant 0. \end{aligned}$$

Ydermere er $r - d < r$, da $d > 0$, så vi alt i alt har, at

$$0 \leqslant r - d < r, \text{ hvor } (r-d) \in S_n.$$

Men dette kan ikke passe, da vi valgte r til at være det mindste ikke-negative element i mængden S_n. Derfor må vores antagelse være forkert, således at $0 \leqslant r < d$.

Tilbage er nu at vise, at der kun findes ét helt tal r, hvor $0 \leqslant r < d$, og ét helt tal q (afhængigt af r), så $n = qd + r$.

Hvis vi nu antager, at der findes to talpar q_1, r_1 og q_2, r_2 , så $n = q_1 d + r_1$ og $n = q_2 d + r_2$, hvor $0 \leqslant r_1 < d$ og $0 \leqslant r_2 < d$. Vi kan så skrive følgende:

$$\begin{aligned} q_1 d + r_1 &= q_2 d + r_2 \\ \Updownarrow & \\ (q_1 - q_2)d &= r_2 - r_1. \end{aligned}$$

Da $0 \leqslant r_1 < d$ og $0 \leqslant r_2 < d$, så må der gælde, at $-d < (r_2 - r_1) < d$. Dette betyder at $r_2 - r_1 = 0$, idet $d|(r_2 - r_1)$. Men så gælder der, at $r_2 = r_1$ og at

$$(q_1 - q_2)d = 0,$$

hvilket medfører, at $q_1 - q_2 = 0$, da $d > 0$.

Derfor er $q_1 = q_2$ og $r_2 = r_1$ således at, hvis q og r opfylder sætningen, så er q og r entydigt bestemt som ønsket. □

Lad os se på et eksempel på brugen af sætning 4.5:

Eksempel 4.6

Lad $n = 17$ og $d = 3$ i sætning 4.5. Hvis vi vælger $q = 5$ og $r = 2$, så har vi, at

$$17 = 5 \cdot 3 + 2, \text{ hvor } 0 \leqslant 2 < 3.$$

Tallet $r = 2$ er således den principale rest ved heltalsdivision af $n = 17$ med $d = 3$.

Hvis n er et negativt tal, så kan vi stadigvæk finde den principale rest r ved heltalsdivision med et helt tal $d > 0$ ifølge sætning 4.5.

Lad nu $n = -12$ og $d = 5$. Så kan vi vælge $q = -3$ og $r = 3$, så

$$-12 = (-3) \cdot 5 + 3, \text{ hvor } 0 \leqslant 3 < 5.$$

Tallet $r = 3$ er derfor den principale rest ved heltalsdivision af $n = -12$ med $d = 5$. ▲

Som det vil fremgå af afsnit 6 i disse noter kan det - at finde den principale rest r ved division af et helt tal n med et andet helt tal d - skrives:

$$n(mod\ d) = r.$$

Således fås for taleksemplerne i eksempel 4.6 at den principale rest af 17 divideret med 3 er lig med 2 kan skrives: $17(mod\ 3) = 2$, og tilsvarende kan den principale rest 3, som fås ved division af -12 med 5 skrives: $-12(mod\ 5) = 3$.

Opgave 4.1

Udfyld om muligt nedenstående tabel:

n	$=$	q	$\cdot$	d
32				8
27				4
32				9
46				11
31				9
-28				-7
24				-6
-51				3
51				-3
42				-4

Angiv divisor og kvotient i de tilfælde, hvor n er divisibel med d.

Opgave 4.2

Udfyld nedenstående tabel, så du får forskellige rester r:

n	$=$	q	$\cdot$	d	$+$	r
27				4		
27				4		
27		6		4		3
27				4		
27				4		
27				4		
27		10		4		-13

Opgave 4.3

Hvad er den principale rest i opgave 4.2?

Opgave 4.4

Angiv den principale rest og den tilhørende entydigt bestemte kvotient ved division af n med d for følgende værdier af n og d.

n	d	Kvotient	Principal rest
33	9		
27	4		
-32	9		
46	11		
31	9		
-7	5		
24	6		
-51	3		
53	3		
42	4		

5 Euklid's algoritme

I dette afsnit præsenteres Euklid's algoritme, der givet to hele tal finder det største hele tal, som går op i begge de givne heltal. Algoritmen er opkaldt efter den græske matematiker Euklid (325-265 f. Kr.), idet han inkluderede en beskrivelse af algoritmen i sit berømte værk "Elementerne".

Lad os starte med at definere, hvad vi mener med fælles divisorer, og indføre lidt notation:

Definition 5.1

*Lad a og b være hele tal, ikke begge 0. Hvis der for et helt tal c gælder, at $c|a$ og $c|b$, da kaldes c en **fælles divisor** for a og b.*

*Mængden af fælles divisorer for a og b har et største element d, der kaldes den **største fælles divisor** for a og b. Dette skrives $gcd(a, b) = d$.*

*Hvis $gcd(a, b) = 1$ da siges a og b at være **indbyrdes primiske**.*

Lad os se på et par eksempler, der involverer fælles divisorer:

Eksempel 5.2

Lad os se nærmere på tallene 12 og 32. Tallet 12 har divisorerne:

$$1, 2, 3, 4, 6 \text{ og } 12,$$

mens tallet 32 har divisorerne:

$$1, 2, 4, 8, 16 \text{ og } 32.$$

De fælles divisorer for tallene 12 og 32 er således:

$$1, 2 \text{ og } 4,$$

hvoraf 4 er det største. Det vil sige, at den største fælles divisor for 12 og 32 er 4, hvilket skrives $gcd(12, 32) = 4$.

Havde vi derimod valgt tallene 18 og 25, der henholdsvis har divisorerne

$$1, 2, 3, 6, 9 \text{ og } 18,$$

og

$$1, 5 \text{ og } 25,$$

så havde vi fået, at $gcd(18, 25) = 1$. Det vil sige, at tallene 18 og 25 ifølge definition 5.1 er indbyrdes primiske. ▲

Med hensyn til den største fælles divisor gælder der følgende resultat, som senere hen skal vise sig at være meget nyttigt:

Sætning 5.3

Lad a og b være hele tal, ikke begge 0. Da eksisterer der hele tal s og t således, at

$$gcd(a, b) = sa + tb.$$

Bevis:

Lad a og b være hele tal, ikke begge 0, og lad $gcd(a, b) = d$. Da $d|a$ og $d|b$, så findes der hele tal q_1 og q_2, således at $a = q_1 d$ og $b = q_2 d$.

Betragt nu mængden af linearkombinbationer af a og b, dvs. mængden:

$$S = \Big\{ sa + tb \;\Big|\; s, t \in \mathbb{Z} \Big\}.$$

Vi vælger nu det mindste positive element i S og kalder dette element x. Det vil sige, at der findes hele tal s_0 og t_0, så $x = s_0 a + t_0 b$.

Først vil vi vise, at x er divisor i både a og b. Ifølge sætning 4.5 eksisterer der entydigt bestemte hele tal q og r, således at $a = qx + r$, hvor $0 \leqslant r < x$. Nu kan vi ved hjælp af $x = s_0 a + t_0 b$ skrive følgende:

$$\begin{aligned} r &= a - qx \\ &= a - q(s_0 a + t_0 b) \\ &= (1 - qs_0)a + (-qt_0)b, \end{aligned}$$

hvilket viser at $r \in S$. Men da $0 \leqslant r < x$ og x var valgt som det mindste positive element i S, så må der gælde, at $r = 0$ og at $x|a$. Beviset for at $x|b$ er helt tilsvarende.

Vi ved altså, at x er divisor i både a og b. Hvis vi nu kan vise, at $d|x$, så må der gælde, at $d = x$ og dermed at $d = s_0 a + t_0 b$, da $d = gcd(a, b)$ er den største fælles divisor for a og b.

Men da $a = q_1 d$ og $b = q_2 d$, så fås:

$$\begin{aligned} x &= s_0 a + t_0 b \\ &= s_0 q_1 d + t_0 q_2 d \\ &= (s_0 q_1 + t_0 q_2) d, \end{aligned}$$

hvilket viser, at $d|x$, så $d = x$ som ønsket. Sætningen er dermed bevist. □

Sætning 5.3 viser altså, at den største fælles divisor mellem to hele tal a og b kan skrives som en linearkombination af a og b. Lad os se på et eksempel:

Eksempel 5.4

Hvis vi ser på tallene 12 og 32 igen, så fandt vi i eksempel 5.2 ud af, at $gcd(12, 32) = 4$. Ifølge sætning 5.3 så eksisterer der hele tal s og t således at

$$4 = s \cdot 12 + t \cdot 32.$$

Med $s = 3$ og $t = -1$ er sætningen opfyldt, idet $4 = 3 \cdot 12 + (-1) \cdot 32$. På samme måde får vi at $gcd(18, 25) = 1 = 7 \cdot 18 + (-5) \cdot 25$, hvor $s = 7$ og $t = -5$ opfylder sætning 5.3. ▲

Med hensyn til fælles divisorer i a og b kan vi nu vise følgende resultat ved hjælp af sætning 5.3:

Sætning 5.5

Lad a og b være hele tal, ikke begge 0, og lad $d = gcd(a, b)$. Hvis $c|a$ og $c|b$, så gælder der at $c|d$.

Bevis:

Da $c|a$ og $c|b$, så findes der hele tal q_1 og q_2, således at $a = q_1 \cdot c$ og $b = q_2 \cdot c$. Desuden da a og b ikke begge er 0, så eksisterer der ifølge sætning 5.3 hele tal s og t, så $gcd(a, b) = d = s \cdot a + t \cdot b$.

Vi kan nu skrive:

$$d = s \cdot a + t \cdot b = s \cdot q_1 \cdot c + t \cdot q_2 \cdot c = (s \cdot q_1 + t \cdot q_2) \cdot c,$$

hvilket viser at $c|d$. □

I eksempel 5.4 er tallene så små, at man forholdsvis nemt kan finde tallene s og t, der opfylder sætning 5.3, men hvad gør man med større tal? Dette problem skal vi senere se en løsning på, men først må vi introducere Euklid's algoritme. Følgende sætning ligger til grund for algoritmen:

Sætning 5.6

Lad a og b være hele tal, ikke begge 0. Da gælder der, at

$$gcd(a, b) = gcd(b, r),$$

hvor r er den principale rest ved division af a med b

Før vi giver os til at bevise sætning 5.6, så vil vi vise følgende lille hjælpesætning:

Lemma 5.7

Lad a, b og c være hele tal, hvor $c \neq 0$. Hvis $c|a$ og $c|b$, så gælder der, at $c|(p_1 \cdot a + p_2 \cdot b)$, hvor p_1 og p_2 er hele tal.

Bevis:
Lad a, b og c være hele tal, hvor $c \neq 0$, $c|a$ og $c|b$. Lad desuden p_1 og p_2 være vilkårlige hele tal. Da $c|a$ og $c|b$, så eksisterer der hele tal q_1 og q_2, således at $a = q_1 \cdot c$ og $b = q_2 \cdot c$ ved brug af definition 4.1. Regner vi nu på summen

$$\begin{aligned} p_1 \cdot a + p_2 \cdot b &= p_1 \cdot q_1 \cdot c + p_2 \cdot q_2 \cdot c \\ &= (p_1 \cdot q_1 + p_2 \cdot q_2)c, \end{aligned}$$

så ses det, at $c|(p_1 \cdot a + p_2 \cdot b)$ som ønsket. Lemmaet er dermed bevist. □

Ved hjælp af lemma 5.7 og sætning 5.5 kan vi nu bevise sætning 5.6:

Bevis (for Sætning 5.6):
Lad a og b være hele tal, ikke begge 0. Vi kan således antage, at $b \neq 0$, idet hvis dette ikke er tilfældet, så byttes a og b i beviset.

Der eksisterer således ifølge sætning 4.5 hele tal q og r, så $a = q \cdot b + r$, hvor r er den principale rest ved division af a med b, dvs. $0 \leqslant r < b$.

Lad $d = gcd(a, b)$. Da $a = q \cdot b + r$ kan omskrives til $r = a + (-q) \cdot b$ og $d|a$ og $d|b$, så gælder der således ifølge lemma 5.7, at $d|r$.

Lad nu $d_1 = gcd(b, r)$, så gælder der ved brug af sætning 5.5, at $d|d_1$, da $d|b$ og $d|r$. Hvis vi ligeledes kan vise, at $d_1|d$, så må der gælde, at $d_1 = d$.

Men da $d_1 = gcd(b, r)$ og $d_1|b$ og $d_1|r$, så gælder der ifølge lemma 5.7, at $d_1|(q \cdot b + r)$, hvilket betyder at $d_1|a$. Da $d_1|a$ og $d_1|b$, så gælder der ved brug af sætning 5.5, at $d_1|gcd(a, b)$ og dermed at $d_1|d$.

Sætningen er således bevist. □

Lad os se på et eksempel på brugen af sætning 5.6:

Eksempel 5.8

I eksempel 5.2 fandt vi ud af, at $gcd(32, 12) = 4$. Da 32 kan skrives som

$$32 = 2 \cdot 12 + 8,$$

hvor 8 er den principale rest ved division af 32 med 12, så gælder der ifølge sætning 5.6, at $4 = gcd(12, 8)$. At dette er rigtigt kan ses ved at betragte divisorerne i 12 og 8 og finde den største fælles divisor.

Tallet 12 har divisorerne $1, 2, 3, 4, 6$ og 12, mens tallet 8 har divisorerne $1, 2, 4$ og 8. Heraf ses, at $gcd(12, 8)$ er netop $4 = gcd(32, 12)$. ▲

Resultatet i sætning 5.6 giver os muligheden for, at finde den største fælles divisor for to hele tal a og b ved at lave gentagne divisioner med rest, hvor vi hele tiden bruger den nye principale rest som divisor på følgende måde:

$$\begin{aligned} a &= q_1 \cdot b + r_1 \\ b &= q_2 \cdot r_1 + r_2 \\ r_1 &= q_3 \cdot r_2 + r_3 \\ &\vdots \\ r_{n-2} &= q_n \cdot r_{n-1} + r_n \end{aligned}$$

Da vi hele tiden dividerer med et mindre tal, så vil de principale rester ved de enkelte divisioner også blive mindre og mindre, indtil vi får resten $r_n = 0$. Det vil sige, at

$$r_1 > r_2 > r_3 > \cdots > r_{n-1} > r_n = 0.$$

Da der ifølge sætning 5.6 gælder, at

$$gcd(a, b) = gcd(b, r_1) = gcd(r_1, r_2) = \cdots = gcd(r_{n-2}, r_{n-1}) = gcd(r_{n-1}, r_n),$$

hvor $r_n = 0$, og da $gcd(r_{n-1}, 0) = r_{n-1}$, så må der gælde, at $gcd(a, b) = r_{n-1}$.

Den efterfølgende algoritme, der bygger på resultatet i sætning 5.6 og ideen ovenfor, kaldes Euklid's algoritme.

```
1  input a,b;
2  x=a;
3  y=b;
4  hvis y=0 saa
5     returner x;
6  ellers
7     r = den principale rest ved division af x med y;
8     x=y;
9     y=r;
10 gaa til linie 4;
```

Listing 1: Euklid's algoritme

Eksempel 5.9

Hvis vi for eksempel ønsker at finde $gcd(72, 46)$, så kan Euklid's algoritme i listing 1 anvendes. En "skrivebordskørsel" af algoritmen ser således ud:

Iteration	x	y	r
1	72	46	26
2	46	26	20
3	26	20	6
4	20	6	2
5	6	2	0
6	2	0	

Efter sjette iteration standser algoritmen og returnerer $x = 2$, der er den største fælles divisor mellem 72 og 46. ▲

Ikke nok med, at Euklid's algoritme kan finde den største fælles divisor: Den kan udvides, så den ydermere finder tallene s og t omtalt i sætning 5.3.

```
1  input a,b;
2  (m,s,t)=(a,1,0);
3  (n,u,v)=(b,0,1);
4  q=nedre heltalsdel af m/n;
5  r=m-nq;
6  hvis r=0 saa
7     returner (n,u,v);
8  ellers
9     (m',s',t')=(n,u,v);
10    (n,u,v)=(r,s-qu,t-qv);
11    (m,s,t)=(m',s',t');
12 gaa til linie 4;
```

Listing 2: Euklid's udvidede algoritme

Eksempel 5.10

Hvis vi for eksempel ønsker at finde $gcd(33, 27)$ og at finde hele tal s og t, så $gcd(33, 27) = s * 33 + t * 27$, så kan Euklid's udvidede algoritme i listing 2 anvendes.

Det algoritmen pricipielt gør er dels at finde den største fælles divisor:

Iteration	a	b	r	Ligning
1	33	27	6	$33 = 1 * 27 + 6$
2	27	6	3	$27 = 4 * 6 + 3$
3	6	3	0	$6 = 2 * 3 + 0$

Her standser algoritmen, idet resten r efter 3 iterationer er blevet 0. Det vil med andre ord sige, at $gcd(33, 27) = 3$. Derefter finder algoritmen s og t. Hvis vi ser på ligningen i 2. iteration, så kan den omskrives til:

$$\begin{aligned} 27 &= 4 * 6 + 3 \\ \Updownarrow & \\ 3 &= 1 * 27 - 4 * 6 \end{aligned} \tag{1}$$

Desuden kan ligningen i 1. iteration omskrives til:

$$\begin{aligned} 33 &= 1 * 27 + 6 \\ \Updownarrow & \\ 6 &= 1 * 33 - 1 * 27 \end{aligned} \tag{2}$$

Ved at substituere højresiden af (2) ind i (1), fås:

$$\begin{aligned} 3 &= 1 * 27 - 4 * 6 \\ &= 1 * 27 - 4 * (1 * 33 - 1 * 27) \\ &= 1 * 27 - 4 * 33 + 4 * 27 \\ &= (-4) * 33 + 5 * 27 \end{aligned}$$

der er på formen $gcd(33, 27) = s*33+t*27$, hvor $s = -4, t = 5$ og $gcd(33, 27) = 3$.

I virkeligheden laver algoritmen i listing 2 ikke beregningerne efter hinanden. I stedet for "samler" algoritmen værdierne af s og t op undervejs i beregningen af den største fælles divisor, hvilket en nærmere gennemgang af algoritmen vil afsløre. ▲

Vi har desuden brug for følgende to resultater vedrørende den største fælles divisor:

Sætning 5.11

Lad a og b være hele tal, ikke begge 0, og lad $m > 0$ være et helt tal. Da gælder der, at

$$gcd(m \cdot a, m \cdot b) = m \cdot gcd(a, b).$$

Vi vil ikke bevise sætning 5.11 (beviset kan eksempelvis ses i Landrock & Nissen), men vi kan anvende sætningen til at vise følgende resultat:

Sætning 5.12

Lad a, b, c være hele tal, hvorom der gælder, at $b > 0$ og $c > 0$. Hvis $c|ab$ og $gcd(c, a) = 1$, så gælder der at $c|b$.

Bevis:

Da $b > 0$ og $c > 0$, så er $gcd(ab, bc) = b \cdot gcd(a, c)$ ifølge sætning 5.11. Og da $gcd(a, c) = 1$, så fås at $gcd(ab, bc) = b \cdot gcd(a, c) = b$.

Desuden, da det antages, at $c|ab$, og $c|bc$ er trivielt opfyldt, så fås ved brug af sætning 5.5 at $c|gcd(ab, bc)$, hvilket er det samme som $c|b$. Sætningen er dermed bevist. □

Opgave 5.1

Hvad er $gcd(12, 32)$?

Opgave 5.2

Er 16 og 33 indbyrdes primiske? Er 21 og 50?

Opgave 5.3

Find alle de tal $0 < n < 24$ hvor n og 24 er indbyrdes primiske.

Opgave 5.4

Da $8|24$ og $8|32$ så gælder der ifølge lemma 5.7, at $8|(3 \cdot 24 + (-4) \cdot 32)$. Kontrollér at dette er rigtigt.

Opgave 5.5

Find ved hjælp af Euklid's algoritme $gcd(119, 85)$.

Opgave 5.6

Find ved hjælp af Euklid's udvidede algoritme $gcd(32, 18)$ og tallene s og t således, at $32s + 18t = gcd(32, 18)$.

Opgave 5.7

Implementér Euklid's udvidede algoritme fra listing 2 på din lommeregner. Bemærk at de fleste lommeregnere har en indbygget funktion (hedder ofte **floor**), der kan finde den nedre heltalsdel af en brøk. Spørg din lærer, hvis du er i tvivl om, hvad funktionen hedder på *din* lommeregner.

Opgave 5.8

Brug din implementation af Euklid's udvidede algoritme fra opgave 5.7 til at finde $gcd(a, b)$ og tallene s og t, hvor $gcd(a, b) = s \cdot a + t \cdot b$, for følgende værdier af a og b:

a	b	s	t	$gcd(a, b)$	$s \cdot a + t \cdot b$
217	33				
412	29				
72	42				
87	33				
39	169				
4123	425				
327	432				
43	0				
8642	6824				

Udregn derefter værdien af $s \cdot a + t \cdot b$ for alle de værdier af s og t, som du har fundet, og sammenlign resultatet med den tilsvarende værdi af $gcd(a, b)$. Har din implementation af Euklid's udvidede algoritme regnet rigtigt?

6 Modulær aritmetik

Bortset fra Euklid's udvidede algoritme, så har vi brug for at se på det, der hedder modulær aritmetik, før vi kan kaste os over *public key* - kryptering. Først en definition:

Definition 6.1

Lad $a, b \in \mathbb{Z}$ og $n \in \mathbb{N}$. Da siges a at være ***kongruent med*** *b* ***modulo*** *n, hvis $n|(a-b)$. Dette skrives $a \equiv b(mod\ n)$.*

Definition 6.1 betyder, at a og b er kongruente modulo n, hvis a og b har samme principale rest ved division med n, men det betyder *ikke*, at b behøver at være den principale rest ved division af a med n. Lad os se på et par eksempler:

Eksempel 6.2

Eksempelvis gælder der, at $11 \equiv 5(mod\ 3)$, idet 3 går op i $11 - 5 = 6$, da $11 - 5 = 6 = 2 \cdot 3$. Bemærk ligeledes, at vi kan skrive 11 som $11 = 3 \cdot 3 + 2$ og 5 som $5 = 1 \cdot 3 + 2$. Det vil med andre ord sige, at 11 og 5 har samme principale rest ved division med 3, nemlig 2.

Faktisk findes der en hel mængde af tal, som 11 er kongruent med modulo 3, nemlig mængden:

$$S = \Big\{ k \cdot 3 + 2 \;\Big|\; k \in \mathbb{Z} \Big\}.$$

At dette er sandt er nemt at se, idet, hvis $x \in S$, så gælder der, at

$$11 - x = 11 - (k \cdot 3 + 2) = 11 - k \cdot 3 - 2 = 9 - k \cdot 3 = (3 - k) \cdot 3,$$

hvor k er et helt tal. Dette viser, at 3 går op i alle tal på formen $11 - x$, hvor $x \in S$. Sagt på en anden måde, så består mængden S af alle de hele tal, der har 2 som principal rest ved division med 3 og dermed af alle de hele tal, som 11 er kongruent med modulo 3.

Bemærk at mængden S også indeholder negative tal (sæt eksempelvis $k = -2$), så vi kan sagtens have, at b nævnt i definition 6.1 er et negativt tal. Eksempelvis tilhører -4 mængden S så $11 \equiv -4(mod\ 3)$, idet $11 - (-4) = 15 = 5 \cdot 3$. ▲

Der gælder følgende regneregler for regneudtryk modulo n, hvor der med notationen $a(mod\ n)$ menes den principale rest af a ved division med n:

$$a + b \equiv \Big(a(mod\ n) + b(mod\ n)\Big)(mod\ n) \tag{3}$$

$$ab \equiv \Big(a(mod\ n) \cdot b(mod\ n)\Big)(mod\ n) \tag{4}$$

$$a^{s \cdot t} \equiv \Big(a^s(mod\ n)\Big)^t(mod\ n) \tag{5}$$

Regnereglerne i (3)-(5) er meget nyttige, idet de muliggør udregninger, som de fleste lommeregnere har svært ved at klare. Lad os se på et eksempel:

Eksempel 6.3

Lad os udregne $7^{144}(mod\ 5)$. Da $144 = 2 \cdot 72 = 2 \cdot 2 \cdot 36$, så fås ved gentagen brug af regnereglen i (5), at:

$$\begin{aligned}
7^{144}(mod\ 5) &= 7^{(2\cdot 2\cdot 36)}(mod\ 5) \\
&= \Big(7^2(mod\ 5)\Big)^{(2\cdot 36)}(mod\ 5) \\
&= \Big(49(mod\ 5)\Big)^{(2\cdot 36)}(mod\ 5) \\
&= \Big(4(mod\ 5)\Big)^{(2\cdot 36)}(mod\ 5) \\
&= 4^{(2\cdot 36)}(mod\ 5) \\
&= \Big(4^2(mod\ 5)\Big)^{36}(mod\ 5) \\
&= \Big(16(mod\ 5)\Big)^{36}(mod\ 5) \\
&= \Big(1(mod\ 5)\Big)^{36}(mod\ 5) \\
&= 1^{36}(mod\ 5) \\
&= 1(mod\ 5) = 1
\end{aligned}$$

Det vil sige, at $7^{144}(mod\ 5) = 1$. Bemærk således at brugen af regnereglerne i (3) - (5) her muliggør en udregning, som de fleste lommeregnere må give op overfor, idet 7^{144} er et tal med over 120 cifre. ▲

Det følgende resultat kæder kongruens modulo n sammen med indbyrdes primiske tal, idet der gælder følgende sætning:

Sætning 6.4

Lad m og n være hele tal, hvor $gcd(m,n) = 1$. Der gælder da, at $a \equiv b(mod\ m)$ og $a \equiv b(mod\ n)$ hvis og kun hvis $a \equiv b(mod\ mn)$.

Bevis:

Vi skal først vise, at hvis $a \equiv b(mod\ m)$ og $a \equiv b(mod\ n)$, så gælder der, at $a \equiv b(mod\ mn)$.

Hvis $a \equiv b(mod\ m)$ og $a \equiv b(mod\ n)$, så gælder der ifølge definition 6.1, at $m|(a-b)$ og $n|(a-b)$. Det vil sige, at der eksisterer q_1, så m går op i $a-b = nq_1$. Men da $gcd(m,n) = 1$, så eksisterer der ved brug af sætning 5.12 et helt tal q_2, så $mq_2 = q_1$. Det vil sige, at vi alt i alt har, at $a - b = nq_1 = nmq_2$, således at $mn|(a-b)$ og dermed at $a \equiv b(mod\ mn)$.

Vi skal dernæst vise, at hvis $a \equiv b(mod\ mn)$, så gælder der, at $a \equiv b(mod\ m)$ og $a \equiv b(mod\ n)$.

Hvis $a \equiv b(mod\ mn)$, så gælder der ved brug af definition 6.1, at $mn|(a-b)$ eller hvad der er det samme: at der eksisterer et helt tal q, så $a - b = nmq$. Men

dette viser jo netop, at $n|(a-b)$ og at $m|(a-b)$, således at $a \equiv b(mod\ m)$ og $a \equiv b(mod\ n)$ ifølge definition 6.1. Sætningen er dermed bevist. □

Lad os se på et eksempel på anvendelsen af sætning 6.4.

Eksempel 6.5

Lad $m = 4$ og $n = 5$, således at $gcd(m,n) = gcd(4,5) = 1$ og $m \cdot n = 4 \cdot 5 = 20$. Lad desuden $a = 63$ og $b = 3$ således at $a - b = 63 - 3 = 60$. Det vil sige, at $a \equiv b(mod\ mn)$, hvilket er det samme som $63 \equiv 3(mod\ 20)$, da $20|(63-3) = 60$.

Ifølge sætning 6.4 skal der så gælde, at $a \equiv b(mod\ m)$ og $a \equiv b(mod\ n)$, hvilket er det samme som $63 \equiv 3(mod\ 4)$ og $63 \equiv 3(mod\ 5)$. At dette er rigtigt kan ses af at

$$a - b = 63 - 3 = 60 = 15 \cdot 4 = 15 \cdot m$$

og at

$$a - b = 63 - 3 = 60 = 12 \cdot 5 = 12 \cdot n.$$

Omvendt så har vi ligeledes ifølge sætning 6.4 at, hvis $a \equiv b(mod\ m)$, $a \equiv b(mod\ n)$ og $gcd(m,n) = 1$, så gælder der at $a \equiv b(mod\ mn)$.

Eksempelvis er $19 \equiv -1(mod\ 2)$, idet $19 - (-1) = 20 = 10 \cdot 2$, og $19 \equiv -1(mod\ 5)$, idet $19 - (-1) = 20 = 4 \cdot 5$. Desuden er $gcd(2,5) = 1$, da både 2 og 5 er primtal. Det vil sige, at ifølge sætning 6.4, så gælder at $19 \equiv -1(mod\ 2 \cdot 5)$, hvilket er rigtigt, da $19 - (-1) = 20 = 2 \cdot 10$. ▲

I det følgende defineres en funktion, der kaldes Euler's ϕ-funktion (det græske bogstav ϕ udtales '*fi*'):

Definition 6.6

Lad $n \in \mathbb{N}$ og lad Z_n^* betegne mængden $Z_n^* = \{1, 2, 3, \ldots, n-1\}$. Definer nu mængden

$$A_n = \{a \in Z_n^* \mid gcd(a,n) = 1\}.$$

Da defineres $\phi(n)$ som antallet af elementer i A_n.

Eksempel 6.7

Lad os se på et eksempel på definition 6.6 med $n = 20$. Da bliver

$$Z_{20}^* = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19\}$$

og

$$A_{20} = \{1, 3, 7, 9, 11, 13, 17, 19\},$$

hvorved $\phi(20) = 8$. ▲

Der gælder følgende sætning vedrørende alle hele tal større end 1:

Sætning 6.8

Ethvert helt tal større end 1 *kan entydigt skrives som et produkt af primtal (dvs. entydigt pånær rækkefølgen af primfaktorerne).*

Vi vil ikke bevise sætning 6.8 her, men bare nøjes med at se på et eksempel.

Eksempel 6.9

Hvis vi ser på tallet 48, så kan det skrives på følgende måde:

$$48 = 2 \cdot 24 = 2 \cdot 2 \cdot 12 = 2 \cdot 2 \cdot 2 \cdot 6 = 2 \cdot 2 \cdot 2 \cdot 2 \cdot 3 = 2^4 \cdot 3,$$

hvor 2 og 3 er primtal. Vi har altså skrevet 48 som et produkt af primfaktorerne 2 og 3.

I dette eksempel er primtalsfaktoriseringen forholdsvis overkommelig, men hvis tallet, der ønskes opløst i primfaktorer, er stort, så bliver det meget sværere at gøre. Lad os tage tallet 6732 og opløse det i primfaktorer:

$$\begin{aligned} 6732 &= 2 \cdot 3366 \\ &= 2 \cdot 2 \cdot 1683 \\ &= 2 \cdot 2 \cdot 3 \cdot 561 \\ &= 2 \cdot 2 \cdot 3 \cdot 3 \cdot 187 \\ &= 2 \cdot 2 \cdot 3 \cdot 3 \cdot 11 \cdot 17 \\ &= 2^2 \cdot 3^2 \cdot 11 \cdot 17, \end{aligned}$$

hvor $2, 3, 11$ og 17 alle er primtal.

Men jo større tallet bliver og jo større primfaktorerne bliver, jo sværere bliver det at gøre. Hvad er eksempelvis primtalsfaktoriseringen af tallet 803147? (svaret står i opgave 6.2) ▲

Primtalsfaktoriseringen af et helt tal $n > 1$ er interessant i krypteringssammenhæng, idet primtalsfaktorisering som antydet i eksempel 6.9 er et vanskeligt problem, når n er et stort tal og primfaktorerne i n er store tal.

Primtalsfaktoriseringen af et helt tal $n > 1$ kan ligeledes kædes sammen med Euler's ϕ-funktion, $\phi(n)$, fra definition 6.6, idet der gælder følgende sætning:

Sætning 6.10

Lad $n \in \mathbb{N}$ og lad $p_1, p_2, \ldots, p_s$ være de forskellige primfaktorer i n. Da gælder der at

$$\phi(n) = n\left(1 - \frac{1}{p_1}\right)\left(1 - \frac{1}{p_2}\right)\cdots\left(1 - \frac{1}{p_s}\right).$$

Vi vil heller ikke bevise sætning 6.10 men igen nøjes med at se på et eksempel på brugen af sætningen.

Eksempel 6.11

Hvis vi ser på tallet 48 igen, så fandt vi i eksempel 6.9 ud af, at 48 kunne primtalsfaktoriseres som $48 = 2^4 \cdot 3$. Det vil sige, at ifølge sætning 6.10, så gælder der, at

$$\phi(48) = 48\left(1-\frac{1}{2}\right)\left(1-\frac{1}{3}\right) = 48 \cdot \frac{1}{2} \cdot \frac{2}{3} = 48 \cdot \frac{1}{3} = 16.$$

Der findes altså 16 tal i mængden $Z_{48}^* = \{1, 2, 3, \ldots, 48\}$, der er indbyrdes primiske med 48. Disse 16 tal er elementerne i mængden

$$A_{48} = \{1, 5, 7, 11, 13, 17, 19, 23, 25, 29, 31, 35, 37, 41, 43, 47\}.$$

Bemærk at brugen af sætning 6.10 indebærer, at man må kende primtalsfaktoriseringen af tallet n, der generelt er et svært problem, når n er et stort tal. ▲

Sætning 6.10 har desuden den konsekvens, at hvis p og q er primtal, så er

$$\begin{aligned}\phi(pq) &= pq\left(1-\frac{1}{p}\right)\left(1-\frac{1}{q}\right) = q\left(p-\frac{p}{p}\right)\left(1-\frac{1}{q}\right) = (p-1)\left(q-\frac{q}{q}\right)\\ &= (p-1)(q-1)\end{aligned}$$

og

$$\phi(p) = p\left(1-\frac{1}{p}\right) = \left(p-\frac{p}{p}\right) = p-1.$$

Den efterfølgende sætning kaldes Euler's sætning:

Sætning 6.12

Lad $a, n \in \mathbb{N}$. Hvis $gcd(a, n) = 1$, da gælder der at $a^{\phi(n)} \equiv 1 (mod\ n)$.

Beviset for sætning 6.12 kan eksempelvis ses i Landrock & Nissen. Sætning 6.12 giver os endnu en regneregel vedrørende modulo n:

$$a^r \equiv a^{r(mod\ \phi(n))} (mod\ n) \tag{6}$$

Regnereglen i (6) er ligeledes meget nyttig.

Eksempel 6.13

Lad os udregne $7^{144} (mod\ 5)$ fra eksempel 6.3 igen, men denne gang vil vi benytte regnereglen i (6) i vores udregning.

Da 5 er et primtal, så er $\phi(5) = 5 - 1 = 4$ ifølge bemærkningerne umiddelbart efter eksempel 6.11. Vi kan således udnytte regnereglen i (6) ved at gøre følgende:

$$\begin{aligned} 7^{144}(mod\ 5) &= 7^{144(mod\ 4)}(mod\ 5) \\ &= 7^0(mod\ 5) \\ &= 1(mod\ 5) = 1 \end{aligned}$$

idet $4 * 36 = 144$, således at $144(mod\ 4) = 0$, og $7^0 = 1$. ▲

Nu mangler vi kun at indføre et enkelt nyt begreb, før vi kan se på RSA. Vi definerer således følgende:

Definition 6.14

Lad $n \in \mathbb{N}$. Da siges $a \in Z_n$ at have et ***inverst element modulo n****, hvis der eksisterer $b \in Z_n$ således at $ab \equiv 1(mod\ n)$. a's inverse element modulo n skrives ofte som $a^{-1}(mod\ n)$.*

Eksempel 6.15

Lad os se på et eksempel på definition 6.14. Hvis vi har $a = 5$ og $n = 7$, så er $5 \cdot 3 \equiv 1(mod\ 7)$. Dermed er 3 og 5 hinandens inverse modulo 7. ▲

Hvis man kender $\phi(n)$, så kan a's inverse element modulo n ved brug af sætning 6.12 findes som $a^{-1} = a^{\phi(n)-1}(mod\ n)$, idet

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

Dette er dog ikke praktisk anvendeligt, idet det indebærer at n må primtalsfaktoriseres for at finde $\phi(n)$. Den følgende sætning fortæller os, hvornår vi kan finde et inverst element til a modulo n.

Sætning 6.16

Hvis $gcd(a, n) = 1$, så har a et inverst element modulo n.

Bevis:
Da $gcd(a, n) = 1$ så findes der ifølge sætning 5.3 to tal s og t således at $as + tn = 1$, der kan omskrives til $as - 1 = -tn$. Det vil sige at $n|(as - 1)$, hvilket er det samme som $as \equiv 1(mod\ n)$. Dermed er s det inverse element til a modulo n. □

Beviset for sætning 6.16 antyder en genvej til at finde et invers element til a modulo n, idet der findes hele tal s og t således at $1 = gcd(a, n) = sa + tn$, hvor s er det søgte inverse element. Men tallene s og t kan jo netop findes ved hjælp af Euklid's udvidede algoritme som vist i eksempel 5.10. Vi kan altså med andre ord finde inverse elementer modulo n ved hjælp af Euklid's udvidede algoritme i stedet for direkte at anvende sætning 6.12. Lad os se på et eksempel:

Eksempel 6.17

Lad os finde det inverse element til 3 modulo 17. Først anvender vi Euklids udvidede algoritme i listing 2 på samme måde, som vi gjorde i eksempel 5.10.

Først finder vi den største fælles divisor for 3 og 17:

Iteration	a	b	r	Ligning
1	17	3	2	$17 = 5 * 3 + 2$
2	3	2	1	$3 = 1 * 2 + 1$
3	2	1	0	$2 = 2 * 1 + 0$

Her standser algoritmen, idet resten r efter 3 iterationer er blevet 0. Det vil med andre ord sige, at $gcd(17, 3) = 1$.

Derefter finder vi s og t. Hvis vi ser på ligningen i 2. iteration, så kan den omskrives til:

$$\begin{aligned} 3 &= 1 * 2 + 1 \\ \Updownarrow & \\ 1 &= 1 * 3 - 1 * 2 \end{aligned} \tag{7}$$

Desuden kan ligningen i 1. iteration omskrives til:

$$\begin{aligned} 17 &= 5 * 3 + 2 \\ \Updownarrow & \\ 2 &= 1 * 17 - 5 * 3 \end{aligned} \tag{8}$$

Ved at substituere højresiden af (8) ind i (7), fås:

$$\begin{aligned} 1 &= 1 * 3 - 1 * 2 \\ &= 1 * 3 - 1 * (1 * 17 - 5 * 3) \\ &= 1 * 3 - 1 * 17 + 5 * 3 \\ &= 6 * 3 - 1 * 17 \end{aligned}$$

der er på formen $gcd(17, 3) = s * 3 + t * 17$, hvor $s = 6, t = -1$ og $gcd(17, 3) = 1$. Vi har således fundet det inverse element til 3 modulo 17 som $s = 6$. At dette er sandt ses af, at $s \cdot 3 \equiv 1 (mod\ 17)$ med $s = 6$ er opfyldt, idet $6 * 3 - 1 = 18 - 1 = 17$ og 17 dermed går op i $6 * 3 - 1$. ▲

Opgave 6.1

Vi har at $11 \equiv 5 (mod\ 3)$. Udfyld følgende

$11 \equiv \_\_\_\_(mod\ 3)$	$\_\_\_\_ \equiv 5(mod\ 3)$	$\_\_\_\_ \equiv \_\_\_\_(mod\ 4)$
$11 \equiv \_\_\_\_(mod\ 3)$	$\_\_\_\_ \equiv 5(mod\ 3)$	$\_\_\_\_ \equiv \_\_\_\_(mod\ 4)$
$11 \equiv \_\_\_\_(mod\ 3)$	$\_\_\_\_ \equiv 5(mod\ 3)$	$\_\_\_\_ \equiv \_\_\_\_(mod\ 4)$

Opgave 6.2

Kotrollér at primtalsfaktoriseringen af tallet 803147 i eksempel 6.9 faktisk er $803147 = 1039 \cdot 773$.

Opgave 6.3

Primtalsfaktoriser 200 og 333.

Opgave 6.4

Find et 4-cifret tal ved at multiplicere flere primtal (gerne store primtal!). Byt derefter med sidemanden. Tag tid og se hvem der hurtigst kan primtalsfaktorisere den andens tal.

Opgave 6.5

Find $\phi(26)$ og $\phi(13)$.

Opgave 6.6

Lad $n = 30$ og find Z_{30}^*, A_{30} og $\phi(30)$.

Opgave 6.7

Udregn $4^{148}(mod\ 7)$ og $5^{256}(mod\ 24)$ ved hjælp af regnereglerne (3) - (6).

Opgave 6.8

Find selv på tal, så sætning 6.4 eftervises.

Opgave 6.9

Find $a \in \mathbb{N}$ så $9a \equiv 1(mod\ 16)$. Du kan vælge enten at bruge Euklid's udvidede algoritme eller Euler's sætning (sætning 6.12).

Opgave 6.10

Har 8 et inverst element modulo 12? Begrund dit svar.

7 RSA (Rivest, Shamir & Adleman)

I 1978 opfandt R. L. Rivest, A. Shamir og L. M. Adleman et krypteringssystem, der bygger på overbevisningen om, at primtalsfaktorisering generelt er et vanskeligt problem (et problem i klassen NP[1]).

[1]Problemerne i klassen NP har dels det til fælles, at den bedste metode man kender til at finde en optimal løsning er at afprøve samtlige muligheder (eksponentielt mange) og dels, at man ikke ved, hvorvidt der findes en algoritme, der kan løse problemerne i polynomiel tid.

Systemet er opbygget på følgende måde:

- Vælg 2 forskellige store primtal p og q, hvor p og q indeholder 200 eller flere cifre, og lad $n = pq$.
- Beregn $\phi(n) = (p-1)(q-1)$.
- Vælg e så $gcd(e, \phi(n)) = 1$.
- Kryptering, $K(m)$, er da givet ved $K(m) = m^e (mod\ n)$, hvor $0 \leqslant m < n$.
- Beregn d så $ed \equiv 1 (mod\ \phi(n))$.
- Dekryptering $D(y)$ er da givet ved $D(y) = y^d (mod\ n)$.

Talparret (e, n) udgør således den offentlige nøgle, mens talparret (d, n) udgør den private nøgle. Sikkerheden ligger således i, at givet talparret (e, n) (den offentlige nøgle), så er det meget svært at finde d, idet det kræver at man kender $\phi(n)$ og dermed primtalsfaktoriseringen af n. Hvis primtallene p og q vælges tilstrækkeligt store, så er det en uoverkommelig opgave at finde dem ved at afprøve alle muligheder.

Det kan vises, at med ovenstående valg af e, d og n, så gælder følgende sætning:

Sætning 7.1

Lad $n = pq$, hvor p og q er forskellige primtal. Lad desuden K og D være defineret som ovenfor og lad $0 \leqslant m < n$. Da gælder der, at

$$D(K(m)) = K(D(m)) = m.$$

Bevis:

At $[m^e (mod\ n)]^d (mod\ n) \equiv [m^d (mod\ n)]^e (mod\ n) \equiv m^{ed} (mod\ n)$ fås umiddelbart af regneregel (5) og de almindelige potensregneregler. Tilbage er kun at vise, at $m^{ed} (mod\ n) \equiv m (mod\ n) = m$, når $0 \leqslant m < n$. Vi må se på to tilfælde:

Antag først at $gcd(m, n) = 1$. Da e og d er valgt således, at $ed \equiv 1 (mod\ \phi(n))$, så gælder der, at $\phi(n) | (ed - 1)$. Ved brug af sætning 6.12 fås således at

$$m^{ed} (mod\ n) \equiv m^{ed (mod\ \phi(n))} (mod\ n) \equiv m (mod\ n) = m,$$

hvis $m < n$.

Antag nu, at $gcd(m, n) \neq 1$. Der må da gælde, enten at $p|m$ eller at $q|m$, da $n = pq$, hvor p og q begge er primtal. Antag således at $p|m$. Det vil sige, at der eksisterer $k \in \mathbb{N}$ således, at $m = kp$ og $gcd(k, q) = 1$. Dette medfører, at $m^{ed} \equiv 0 (mod\ p)$ og $m^{ed} \equiv m (mod\ p)$.

Da $\phi(n) = (p-1)(q-1)$ og $\phi(n) | (ed - 1)$, så gælder der, at $(q-1) | (ed - 1)$. Da $\phi(q) = q - 1$ og $gcd(m, q) = 1$, så fås ved brug af sætning 6.12, at

$$m^{ed} (mod\ q) \equiv m^{ed (mod\ \phi(q))} (mod\ q) \equiv m (mod\ q).$$

Det vil sige, at $m^{ed} (mod\ p) \equiv m (mod\ p)$ og $m^{ed} (mod\ q) \equiv m (mod\ q)$, hvilket ifølge sætning 6.4 er ensbetydende med, at $m^{ed} \equiv m (mod\ n) = m$, hvor $n = pq$, hvis $m < n$ som ønsket. Udregningerne er tilsvarende hvis $q|m$. □

Vi vælger i det efterfølgende at nummerere bogstaverne i alfabetet fra 1 til 29, dvs.

A=01	B=02	C=03	D=04	E=05	F=06	G=07	H=08
I=09	J=10	K=11	L=12	M=13	N=14	O=15	P=16
Q=17	R=18	S=19	T=20	U=21	V=22	W=23	X=24
Y=25	Z=26	Æ=27	Ø=28	Å=29			

Lad os se på et eksempel på brugen af RSA-krypteringssystemet:

Eksempel 7.2

Lad os vælge $p = 41$ og $q = 47$ således at $n = 41 * 47 = 1927$ og $\phi(1927) = (41-1)(47-1) = 40 * 46 = 1840$.

Vi vælger nu $e = 33$, idet $gcd(33, 1840) = 1$. Da $1840 = 2^4 * 5 * 23$, så fås at $\phi(1840) = 1840(1-1/2)(1-1/5)(1-1/23) = 704$. Det vil sige, at $d = e^{703} = 33^{703}(mod\ 1840) = 1617$ og at $33 * 1617 \equiv 1(mod\ 1840)$.

Talparret $(33, 1927)$ udgør således den offentlige nøgle, mens talparret $(1617, 1927)$ udgør den private nøgle. Bemærk at tallene $p = 41, q = 47$ og $\phi(1927)$ har tjent deres formål og kan kasseres.

Hvis vi nu ønsker at kryptere teksten *skoleelev*, så gøres det på følgende måde:

Først skrives teksten som tal:

Tekst	S	K	O	L	E	E	L	E	V
Tekst som tal	19	11	15	12	05	05	12	05	22

Dernæst skrives teksten som talblokke, hvor hver blok m ikke må repræsentere et tal større end $n = 1927$ - vi vælger derfor blokke m med tre cifre og udregner $m^e(mod\ n)$:

Talblokke m	191	115	120	505	120	522
$m^{33}(mod\ 1927)$	1380	1578	161	932	161	458

Udregningerne ovenfor kan forenkles ved hjælp af regnereglerne (3)-(6) og de almindelige potensregneregler. Eksempelvis kan $191^{33}(mod\ 1927)$ udregnes på følgende måde:

$$
\begin{aligned}
& 191^{33}(mod\ 1927) \equiv [191^3(mod\ 1927)]^{11}(mod\ 1927) \\
\equiv\ & [6967871(mod\ 1927)]^{11}(mod\ 1927) \\
\equiv\ & 1766^{11}(mod\ 1927) \\
\equiv\ & 1766 * 1766^{10}(mod\ 1927) \\
\equiv\ & (1766(mod\ 1927) * 1766^{10}(mod\ 1927))(mod\ 1927) \\
\equiv\ & (1766(mod\ 1927) * (1766^2(mod\ 1927))^5(mod\ 1927))(mod\ 1927) \\
\equiv\ & (1766(mod\ 1927) * 870^5(mod\ 1927))(mod\ 1927) \\
\equiv\ & (1766(mod\ 1927) * (870(mod\ 1927) * 870^4(mod\ 1927))(mod\ 1927))(mod\ 1927) \\
\equiv\ & (1766(mod\ 1927) * (870(mod\ 1927) * (870^2(mod\ 1927))^2(mod\ 1927)))(mod\ 1927) \\
\equiv\ & (1766(mod\ 1927) * (870(mod\ 1927) * 1516^2(mod\ 1927))(mod\ 1927))(mod\ 1927) \\
\equiv\ & (1766(mod\ 1927) * (870 * 1272(mod\ 1927)))(mod\ 1927) \\
\equiv\ & (1766(mod\ 1927) * 542(mod\ 1927))(mod\ 1927) \\
\equiv\ & 1766 * 542(mod\ 1927) = 1380
\end{aligned}
$$

Vores krypterede meddelelse består således af numrene 1380, 1578, 161, 932, 161 og 458. For at dekryptere disse, må vi udregne $y^{1617}(mod\ 1927)$, hvor y er numrene ovenfor. Det vil sige

Talblokke y	1380	1578	161	932	161	458
$y^{1617}(mod\ 1927)$	191	115	120	505	120	522

Talrækken $191, 115, 120, 505, 120, 522$ inddeles nu igen i blokke af 2 cifre, dvs. $19, 11, 15, 12, 05, 05, 12, 05, 22$, hvorefter de enkelte tal oversættes tilbage til bogstaver ved hjælp af tabellen umiddelbart før dette eksempel. ▲

Opgave 7.1

Vælg 2 forskellige tal p og q blandt de følgende primtal og udregn derefter $n = pq$ (Husk at vælge p og q således at $n > 29$).

$$\{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37\}$$

Opgave 7.2

Udregn $\phi(n)$ hvor n er tallet fra opgave 7.1.

Opgave 7.3

Vælg e så $gcd(e, \phi(n)) = 1$, hvor $\phi(n)$ er fra opgave 7.2.

Opgave 7.4

Beregn d så $ed \equiv 1(mod\ \phi(n))$, hvor e er fra opgave 7.3 og $\phi(n)$ fra opgave 7.2. Du kan med fordel anvende Euklid's udvidede algoritme.

Opgave 7.5

Oversæt ordet *kat* til en talrække ved hjælp af skemaet umiddelbart før eksempel 7.2. Inddel derefter tallene i lige store blokke, så hver blok indeholder 1 ciffer mindre end dit tal n fra opgave 7.1 (Husk foranstillede nuller).

Opgave 7.6

Kryptér derefter ordet *kat* ved at udregne $m^e(mod\ n)$, hvor m er de enkelte blokke fra opgave 7.5. Udnyt her regnereglerne (3) - (6).

Opgave 7.7

Dekryptér derefter resultatet fra opgave 7.6 ved at udregne $y^d(mod\ n)$, hvor y er blokkene du fik som resultat i opgave 7.6. Udnyt igen regnereglerne (3) - (6).

Opgave 7.8

I opgaverne 7.1 til 7.4 har du genereret en offentlig og en privat nøgle, der henholdsvis består af talparrene (e, n) og (d, n).

Byt offentlig nøgle med din sidemand og kryptér derefter et (kort) ord efter eget valg med sidemandens offentlige nøgle.

Byt nu rækken af krypterede blokke med din sidemand og dekryptér det ord, som din sidemand har krypteret med din offentlige nøgle.

Opgave 7.9

Af formlen for Eulers ϕ-funktion i sætning 6.12 fremgår det, at $\phi(15) = 8$.

I skemaet nedenfor repræsenterer tallet m muligheder for meddelelser i et meget enkelt RSA-system bygget op af primtallene 3 og 5.

Udfyld skemaet og forklar ud fra skemaet, hvilke elementer, der har inverse og hvilke der ikke har det.

m	$m^{\phi(n)-1}(mod\ n)$ $= m^7(mod\ 15)$	$m^{\phi(n)}(mod\ n)$ $= m^8(mod\ 15)$	$m^{\phi(n)+1}(mod\ n)$ $= m^9(mod\ 15)$
0			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			

8 Digital signatur

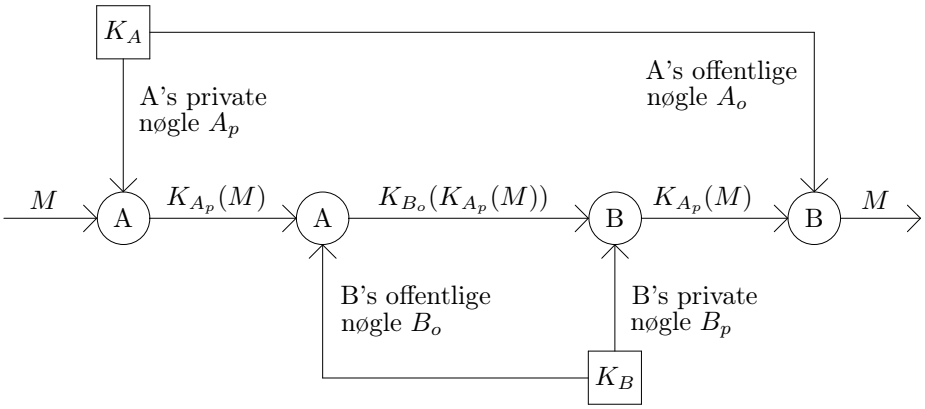
Hvis en afsender A ønsker at signere eller "underskrive" sin meddelelse på en måde, så modtageren B kan være helt sikker på, at meddelelsen kommer fra A, så kan et *public key* - krypteringssystem ligeledes ofte give mulighed herfor.

Dette gøres på følgende måde:

- A krypterer først meddelelsen M med sin egen private nøgle A_p (dette kan *kun* A gøre) og producerer således meddelelsen $K_{A_p}(M)$.
- A krypterer derefter meddelelsen $K_{A_p}(M)$ med B's offentlige nøgle B_o (som alle har adgang til) og sender derefter den krypterede meddelelse $K_{B_o}(K_{A_p}(M))$ til B.
- Ved modtagelsen dekrypterer B først meddelelsen $K_{B_o}(K_{A_p}(M))$ med sin private nøgle B_p og får således meddelelsen $D_{B_p}(K_{B_o}(K_{A_p}(M))) = K_{A_p}(M)$.

- B dekrypterer derefter meddelelsen $K_{A_p}(M)$ med A's offentlige nøgle A_o (som alle ligeledes har adgang til) og får $D_{A_o}(K_{A_p}(M)) = M$.

Hvis meddelelsen M herefter er læsbar, så er B fuldstændig sikker på, at meddelelsen virkelig kommer fra A, idet det kun er A, der kan have krypteret meddelelsen med A's private nøgle. Skematisk ser brugen af digital signatur således ud:



Bemærk: For at digital signatur skal fungere, så er det enten strengt nødvendigt, at rækkefølgen hvori kryptering og dekryptering foretages er ligegyldig, dvs. at:

$$D_{n_1}(K_{n_2}(M)) = K_{n_2}(D_{n_1}(M)) = M,$$

hvor n_1 og n_2 udgør et sammenhørende nøglepar (bestående af en privat og en offentlig nøgle).

Eller også skal være ligegyldigt om man anvender den offentlige nøgle først og den private nøgle bagefter eller omvendt - når bare de to nøgler tilsammen udgør et nøglepar, dvs. at:

$$D_{n_1}(K_{n_2}(M)) = D_{n_2}(K_{n_1}(M)) = M,$$

hvor n_1 og n_2 igen er et sammenhørende nøglepar.

I den skematiske gengivelse ovenfor er den sidste regel anvendt, idet A først krypterer meddelelsen med sin private nøgle - dvs. A konstruerer $K_{A_p}(M)$, hvorefter B slutter af med at dekryptere med A's offentlige nøgle - dvs. B konstruerer $D_{A_o}(K_{A_p}(M)) = M$, der ifølge den sidste regel er det samme som $D_{A_p}(K_{A_o}(M)) = M$.

Opgave 8.1

Gentag øvelsen fra opgave 7.8 men anvend denne gang digital signatur - det vil sige: kryptér først dit valgte ord med din private nøgle og derefter med din sidemands offentlige nøgle.

Ved dekrypteringen må du således først dekryptere med din private nøgle og derefter dekryptere med din sidemands offentlige nøgle.

Opgave 8.2

Tegn den skematiske gengivelse af anvendelsen af digital signatur, hvor reglen

$$D_{n_1}(K_{n_2}(M)) = K_{n_2}(D_{n_1}(M)) = M,$$

anvendes i stedet for.

9 Liste over anvendt litteratur

- Peter Landrock & Knud Nissen, *Kryptologi*, 1. udgave, Forlaget ABACUS, 1990.
- Ramanujachary Kumanduri & Cristina Romero, *Number Theory with computer applications*, Prentice-Hall Inc., 1998.
- Simon Singh, *Kodebogen - Videnskaben om hemmelige budskaber fra oldtidens Ægypten til kvantekryptering*, 1. udgave, Gyldendal, 1999.
- Keith O. Geddes, Stephen R. Czabor & George Labahn, *Algorithms for Computer Algebra*, Kluwer Academic Publishers, 1992.
- A. J. Menezes, P. C. van Oorschot & S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 2001.
- J. Buchmann, *Introduction to Cryptography*, Undergraduate Texts in Mathematics, Springer-Verlag, 2001.

10 Liste over hjemmesider, der omhandler kryptering m.m.

- Applet til demonstration af Euclid's algoritme (udvidet)
  http://users.erols.com/eweidaw/applets/EuclidExtension.html
- Interaktiv hjemmeside om kryptering og talteori
  http://www.math.umn.edu/~garrett/crypto/Interactive.html
- Cryptography and Network Security
  http://williamstallings.com/Security2e.html
- Cryptography for Middle and High School Teachers
  http://www.math.arizona.edu/~jsmith/crypt.html
- Hjemmeside for *Handbook of Applied Cryptography*
  http://www.cacr.math.uwaterloo.ca/hac/
- John Savard's hjemmeside
  http://fn2.freenet.edmonton.ab.ca/~jsavard/crypto.htm
- Lucian Illie
  http://www.csd.uwo.ca/courses/CS434b/

- The Numeroscope
  http://www.woodrow.org/teachers/math/numeroscope/
- Paj's Home: Cryptography
  http://pajhome.org.uk/crypt/
- Rich Holowczak's RSA Demo Applet
  http://cisnet.baruch.cuny.edu/holowczak/classes/9444/rsademo/
- Ronald L. Rivest: Cryptography and Security
  http://theory.lcs.mit.edu/~rivest/crypto-security.htm
- Andrew Hodges: Alan Turing — a short biography
  http://www.turing.org.uk/turing/
- Peter Landrock: Sikker matematik. Kronik i Jyllandsposten 12/10-2000
  som en del af kronikserie i forbindelse med verdensmatematikåret 2000
  http://www.mip.sdu.dk/mat2000/Kronikserie/kronik006.html