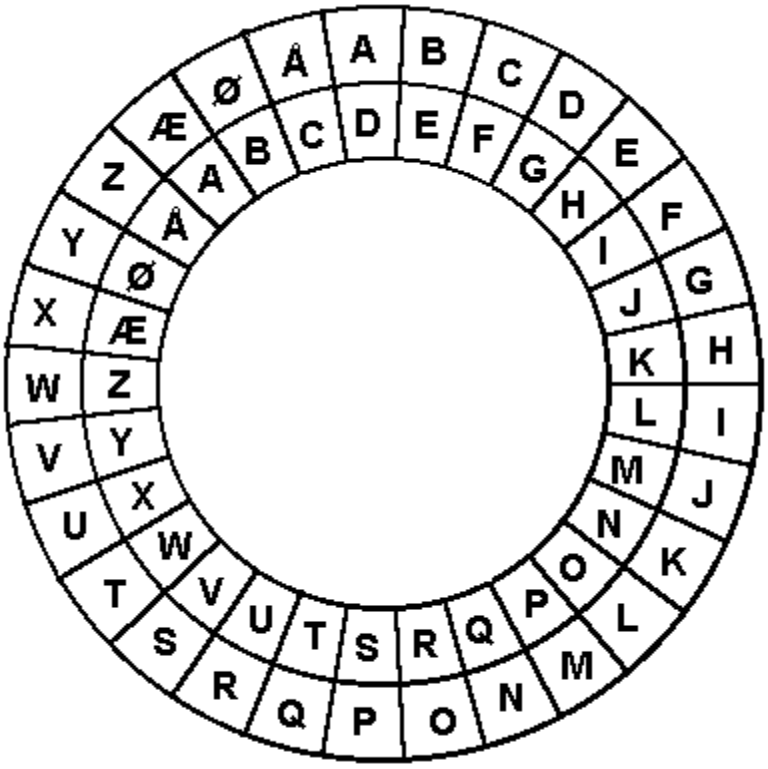
Kryptologi

af

Kenneth Hansen



Indhold

0. Forord

Dette hæfte omhandler kryptologi - videnskaben omhandlende hemmelig skrift og kommunikation.

Hæftet er naturligt opdelt i to dele - de klassiske kryptosystemer er omtalt i kapitlerne 1-8, hvorefter vi bevæger os over i informationssamfundet og opfyldelsen af dettes krav. Dette sker i kapitlerne 9-12.

Et par praktiske bemærkninger:

Der er en mere eller mindre rigtig facitliste til alle opgaverne bagerst i hæftet.

Samtlige klar- og kryptotekster er skrevet med store bogstaver og en speciel skrifttype - her er et eksempel:

ALAS, YE MORTAL. THINE DAYS ARE COUNTED

1. Additiv monoalfabetisk substitution

Den simpleste måde at lave hemmelig skrift på, og vel nok den først anvendte, er at foretage en såkaldt *monoalfabetisk substitution*. Dette går ganske simpelt ud på at erstatte alle forekomster af et bogstav i sin tekst med et andet.
For at gøre tingene simple starter vi med at betragte såkaldte *additive monoalfabetiske substitutioner.*

Cæsar-substitution

Et af de ældste kendte eksempler på anvendelse af hemmelig skrift stammer fra den romerske general og statsmand Julius Cæsar. Under sine militære kampagner bl.a. i Gallien havde han ofte brug for at sende skriftlige beskeder til sine hærførere. Problemet med sådanne beskeder er jo naturligvis, at de kan opsnappes af fjenden, som derved kommer i besiddelse af farlig information. Cæsars løsning på dette problem var at sende sit budskab i kodet tilstand.
Cæsars metode - den såkaldte *Cæsar-substitution* - gik ud på at erstatte hvert bogstav i budskabet med det bogstav, som lå tre pladser længere nede i alfabetet. Således blev bogstavet `A` erstattet med bogstavet `D`.
Budskabet

```
ANGRIB VED DAGGRY
```

blev ligeledes erstattet med

```
DQKULE YHG GDJJUØ
```

Det viser sig hurtigt, at man nemmest kommer over både kodningen og afkodningen ved at lave en tabel over hele alfabetets bogstaver og de bogstaver, de erstattes med:

klartekst	`ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ`
kodetekst	`DEFGHIJKLMNOPQRSTUVWXYZÆØÅABC`

Her ser man også, at ved bogstaverne *Æ*, *Ø* og *Å*, hvor man jo ikke kan finde bogstaver længere nede i alfabetet, starter forfra ved A, B og C.
En anden metode er at bruge et *kodehjul* - se senere.

Opgave 1.1

Forestil dig, at du er rådgiver og vismand for den galliske høvding Vercingetorix. Hans to spejdere, Asterix og Obelix, har netop opsnappet et budskab, og han beder dig finde ud af, hvad der står.
Budskabet er

```
RPQLD JDOOLD HVW GLYVLD SDUWHV WUHV
```

Som altid ved hemmelig skrift og deslige er der en risiko - måden man koder og afkoder på kan falde i fjendens hænder. Utallige erfaringer gennem tiderne har vist, at denne information **altid** falder i fjendens hænder - det er kun et spørgsmål om tid.
En måde at omgå dette problem på er at ændre sin kodningsmetode regelmæssigt. Omvendt er dette upraktisk - man kan jo have hærførere (eller andet personel), som ikke er så fleksible som en selv.

Det viser sig derfor at være smart at opdele ens kodningsmetoder i to dele. Selve *systemet* er det samme, men *nøglen* kan ændres relativt nemt.
F.eks. kunne man i Cæsar-substitutionen beholde den grundlæggende idé med at ændre hvert bogstav til et efterfølgende bogstav i alfabetet, men ændre det antal bogstaver, man går frem: Når vismændene hos de barbariske gallere endeligt har fundet ud af systemet med at gå tre bogstaver frem i alfabetet, så indfører man bare et system med at gå fire bogstaver frem - til gallernes store ærgrelse.

Her er det vist på tide at indføre nogle **præcise** betegnelser for det, vi hidtil har arbejdet med:

Budskaber sendes altid fra en *afsender* til en *modtager*. Budskabernes indhold skal helst hemmeligholdes for *fjenden*.

Et *budskab* er en sekvens af karakterer. En *karakter* er et element i et *alfabet*. En karakter kunne således være et bogstav, eller et tal, eller et mystisk symbol, eller...

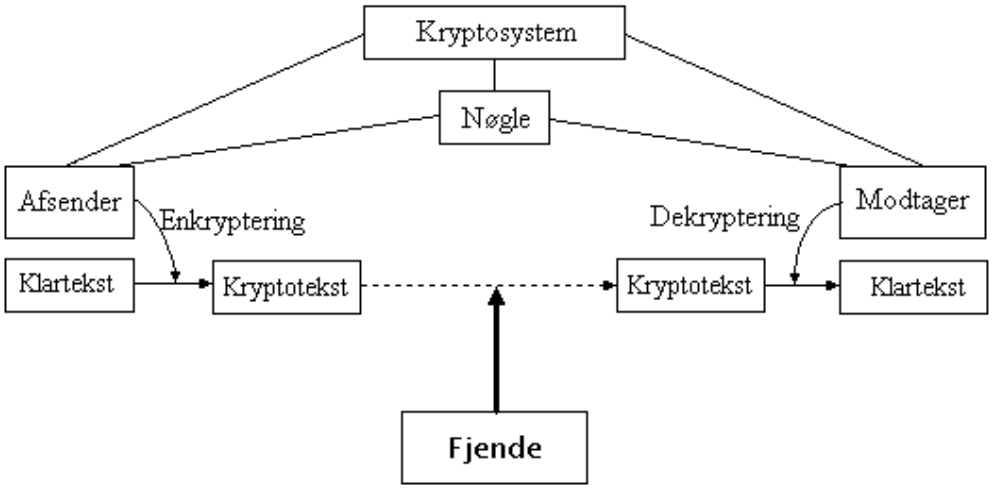
Budskabet er oprindeligt skrevet i *klartekst*, dvs. det er en sekvens af karakterer i *klartekstalfabetet*. Afsenderen *enkrypterer* budskabet, dvs. ændrer det til en sekvens af karakterer i *kryptotekstalfabetet* - en såkaldt *kryptotekst*. Herefter sendes budskabet til modtageren, som *dekrypterer* det.

Både *enkrypteringen* og *dekrypteringen* er processer, hvor man anvender et *kryptosystem* og en *nøgle*.

Selve processen med at enkryptere og dekryptere teksterne kaldes *kryptografi*.

Fjenden, som man antager har fuld tilgængelighed til kryptoteksterne, vil formentligt lave en *kryptoanalyse*, dvs. forsøge at finde klarteksten ud fra kryptoteksten, men også finde kryptosystemet og nøglen.

Mere malerisk kan situationen beskrives på følgende måde:



Ved Cæsar-substitutionen er afsenderen Cæsar selv, modtageren er hans hærfører, fjenden er galleren Vercingetorix.

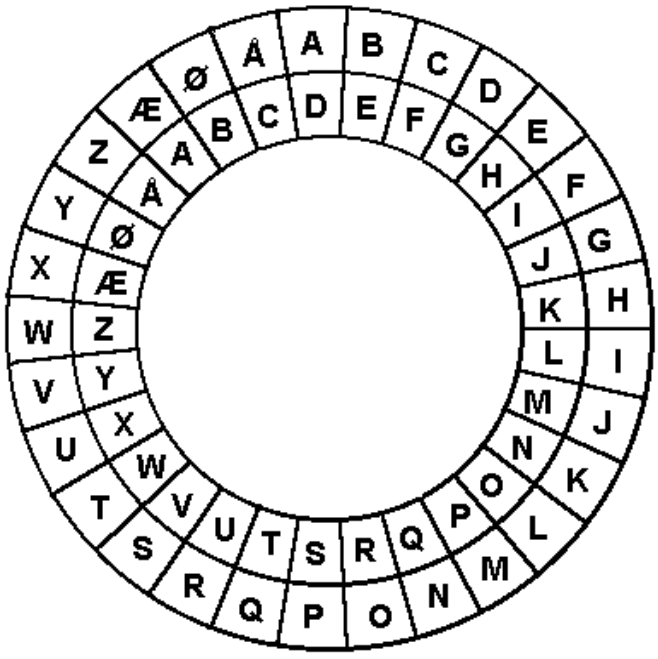
Klartekst- og kryptoalfabetet er begge mængden
{A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,Æ,,Å} .

Kryptosystemet er ideen med at erstatte hvert bogstav med sin *n*'te efterfølger, og nøglen er et tal *n*.

Ovennævnte kryptosystem kaldes *additiv monoalfabetisk substitution*; 'substitution' fordi hvert bogstav erstattes med et andet, 'monoalfabetisk' fordi man kun anvender ét kryptoalfabet, og 'additiv' fordi man erstatter hvert bogstav i klartekstalfabetet med bogstavets 'nummer' **plus** nøgleværdien.

Kodehjulet

Er man en professionel anvender af additiv monoalfabetisk substitution, så er det en god idé at lave et *kodehjul* - se figuren nedenfor.
Dette består at to runde papskiver, hvor den ene har en diameter ca. 1 cm større end den anden. Periferien på de to skiver er inddelt i 29 felter, og i hvert felt er der skrevet er bogstav. Bogstaverne står naturligvis i alfabetisk rækkefølge.
Gennem de to skivers fælles centrum er det sat en clips, således at skiverne kan drejes uafhængigt af hinanden.
Ved anvendelse af kodehjulet indstiller man først den lille skive, således at klartekst-bogstavet `A` står ud for det bogstav, `A` krypteres til. På figuren er dette åbenbart `D`.
Enkryptering foregår nu ved at erstatte et bogstav fra den ydre ring med til tilsvarende bogstav fra den indre ring, og dekryptering er naturligvis at erstatte et bogstav fra den indre ring med et bogstav fra den ydre ring.



Kryptoanalyse af additiv monoalfabetisk substitution

Desværre er det ekstremt nemt at lave en kryptoanalyse af additiv monoalfabetisk substitution - selv uden kendskab til nøglen.

Forestil dig, at du skal kryptoanalysere budskabet

XKZÅ DVY CVQCDR ZNÆQ

Dette gøres på følgende måde:

Opskriv kryptoteksten, og under hvert bogstav heri opskriv alfabetet nedad startende med bogstavet i kryptoteksten - se næste side.

På et eller andet tidspunkt vil klarteksten dukke op.

XKZÅ DVY CVQCDR ZNÆQ
YLÆA EWZ DWRDES ÆOØR
ZMØB FXÆ EXSEFT ØPÅS
ÆNÅC GYØ FYTFGU ÅQAT
ØOAD HZÅ GZUGHV ARBU
ÅPBE IÆA HÆVHIW BSCV
AQCF JØB IØWIJX CTDW
BRDG KÅC JÅXJKY DUEX
CSEH LAD KAYKLZ EVFY
DTFI MBE LBZLMÆ FWGZ
EUGJ NCF MCÆMNØ GXHÆ
FVHK ODG NDØNOÅ HYIØ
GWIL PEH OEÅOPA IZJÅ
HXJM QFI PFAPQB JÆKA
IYKN RGJ QGBQRC KØLB
JZLO SHK RHCRSD LÅMC
KÆMP TIL SIDSTE MAND
LØNQ UJM TJETUF NBOE
MÅOR VKN UKFUVG OCPF
NAPS WLO VLGVWH PDQG
OBQT XMP WMHWXI QERH
PCRU YNQ XNIXYJ RFSI
QDSV ZOR YOJYZK SGTJ
RETW ÆPS ZPKZÆL THUK
SFUX ØQT ÆQLÆØM UIVL
TGVY ÅRU ØRMØÅN VJWM
UHWZ ASV ÅSNÅAO WKXN
VIXÆ BTW ATOABP XLYO
WJYØ CUX BUPBCQ YMZP

Et kryptosystem, der kan knækkes så let, er vist ikke meget værd. Vi vil derfor prøve at finde på nogle mere komplicerede og sofistikerede systemer.

Opgaver

1.2 Nedenstående tekst er krypteret med en additiv monoalfabetisk substitution. Hvad står der?

```
IPWNRMJMEG CP QHMTCPC CLB GLRCEPØJPCELGLE
```

2. Modulær aritmetik

Vi skal nu se nærmere på, hvorfor Cæsar-substitutionen kaldes en *additiv* monoalfabetisk substitution:

Alfabetet ved Cæsar-substitutionen er jo

{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}

men man kunne lige så godt bruge tal - A erstattes med 0, B med 1, ... Å med 28.
Dette har den fordel, at substitution med nøgleværdien *k* kan beskrives matematisk ved

$$E_k(a) = a + k \qquad \text{og} \qquad D_k(b) = b - k$$

Her er E_k *enkrypteringsfunktionen*, D_k dekrypteringsfunktionen, *a* er en vilkårlig klartekstskarakter og *b* en vilkårlig kryptotekstkarakter.

Vælger vi f.eks. den klassiske Cæsar-substitution med $k = 3$, så ses, at

$$E_3(2) = 2 + 3 = 5,$$

hvilket oversættes til, at bogstavet C krypteres til E.
Desværre opstår der nogle problemer, f.eks.

$$E_3(27) = 27 + 3 = 30$$

Men Ø krypteres ikke til bogstav 30 (hvad det så end er), men til B, bogstav 1...

Dette, og andre problemer, kan løses ved at indføre den *modulære aritmetik*, som er en del af *talteorien*.
I det følgende vil vi kun beskæftige os med hele tal. Vi starter med et af de grundlæggende begreber, *divisibilitet*.

Definition 1

Lad *a* og *b* være hele tal. Vi siger, at *a går op i b*, symbolsk $a|b$, hvis der findes et helt tal *k*, således at $b = k \cdot a$.
Alternativt siger vi, at *a* er en *divisor* i *b*.

Eksempel

$2|28$, idet $28 = 14 \cdot 2$ (altså, $k = 14$)

Til gengæld har vi, at 5 ikke går op i 43. Ganske vist kan vi skrive $43 = 8{,}6 \cdot 5$, men $k = 8{,}6$ er ikke et helt tal!

Syge eksempler

1 går op i alle tal, idet: $b = b \cdot 1$ - vi kan altså vælge *k* som tallet selv.
Alle tal går op i 0. Vi kan nemlig vælge *k* som 0 og få $0 = 0 \cdot a$.
Det eneste tal, som 0 går op i, er 0 selv. Vi har nemlig, uanset værdien af *k*, at $k \cdot 0 = 0$.

Som navnet antyder, hænger divisibilitet tæt sammen med division. Vi husker fra folkeskolen, at man kan dividere to hele tal med hinanden på to forskellige måder.

Den første måde giver normalt en brøk. Hvis man f.eks. dividerer 7 med 3, så får man som resultat

$$\frac{7}{3} = 2\frac{1}{3} = 2{,}33333333...$$

Denne metode er ikke særligt hensigtsmæssig, når man arbejder med hele tal, så vi bruger derfor den anden.

Dividerer man 7 med 3, så får man 2 og 1 til rest. Dette kaldes *heltalsdivision*. Man plejer at opskrive dette i en *divisionsligning*:

$$7 = 2 \cdot 3 + 1.$$

Generelt definerer man

Definition 2

Lad *a* og *b* være hele tal. Så defineres

a DIV b = kvotienten ved heltalsdivisionen $b : a$

a MOD b = resten ved heltalsdivisionen $b : a$

Eksempler

$387 \text{ DIV } 17 = 22 \qquad 387 \text{ MOD } 17 = 13$

På lommeregneren kan dette beregnes ved at lave den almindelige division 387:17. Resultatet bliver 22,764..., og heltalsdelen, 22, er 387 DIV 17.
Heltalsdelen 22 trækkes fra, og de tilbageblevne 0,764... ganges med 17 og giver resten 13.

$-8 \text{ DIV } 17 = -1 \qquad -8 \text{ MOD } 17 = 9.$

Hvorfor nu det? Jo, divisionsligningen er
$-8 = (-1) \cdot 17 + 9$
hvoraf de to resultater kan aflæses.
DIV og MOD kan altså også bruges med negative tal.

Man ser, at operatorerne DIV og MOD løser vores problem omkring Cæsar-substitutionen. Ændrer vi vores ligninger til

$$E_k(a) = (a + k) \text{ MOD } 29 \qquad \text{og} \qquad D_k(b) = (b - k) \text{ MOD } 29$$

så virker de! Tallet 29 anvendes, fordi der er 29 bogstaver i vores alfabet.

I praksis er operatorerne MOD og DIV for kluntede - man anvender i stedet for restklasser eller kongruenser. Det er her, at den modulære aritmetik for alvor begynder!

Definition 3

Lad n være et fast helt tal, og a og b to vilkårlige hele tal.

Vi siger, at *a er kongruent til b modulo n*, hvis

$$n \mid (a-b)$$

Dette skrives normalt som

$$a \equiv b \pmod{n}$$

Definition 4

Lad n være et fast helt tal, og a et vilkårligt helt tal.

Mængden

$$\{x \in \mathbf{Z} \mid a \equiv x \pmod{n}\}$$

kaldes en *restklasse modulo n*.

Eksempel

Lad os se lidt på restklasserne modulo 2. Der er to, nemlig

{...,–8,–6,–4,–2,0,2,4,6,8,10,12,...}

{...,–9,–7,–5,–3,–1,1,3,5,7,9,11,...}

bedre kendt som de *lige* og de *ulige* tal.

Restklasserne modulo 3 er

{...,-9,-6,-3,0,3,6,9,12,...}
{...,-8,-5,-2,1,4,7,10,13,...}
{...,-7,-4,-1,2,5,8,11,14,...}

I den første restklasse giver alle tallene resten 0 ved division med 3, i den næste giver alle tallene resten 1 ved division med 3, og i den sidste fås resten 2.

Helt generelt gælder der

Sætning 5

Der findes n restklasser modulo n.

Alle elementer i en restklasse giver den samme rest ved division med n.

Man vælger derfor at forveksle en restklasse med denne rest - den *principale rest*.

Definition 6

Lad n være et positivt helt tal. $\mathbf{Z}_n$ - *mængden af restklasser modulo n*, defineres som mængden

$$\mathbf{Z}_n = \{0,1,2,...,n-2,n-1\}$$

I virkeligheden opfatter vi alfabetet i Cæsar-substitutionen som $\mathbf{Z}_{29}$.

Kan vi nu regne i disse restklassemængder som normalt? Ja og nej - nedenstående sætning viser, at addition, subtraktion og multiplikation kan foretages som normalt. Division er derimod meget mere kompliceret, og vi vender tilbage til dette senere.

Sætning 7

Lad n være et fast tal, og lad a og b være hele tal opfyldende

$$a \equiv b \pmod{n}.$$

Lad c være et vilkårligt helt tal. Så gælder

1) $a + c \equiv b + c \pmod{n}$

2) $a - c \equiv b - c \pmod{n}$

3) $a \cdot c \equiv b \cdot c \pmod{n}$

Bevis:

Betingelsen $a \equiv b \pmod{n}$ betyder jo, at $n \mid (a-b)$, hvilket igen betyder, at der findes et helt tal k, således at $a - b = k \cdot n$

1) Idet $(a+c)-(b+c) = a+c-b-c = a-b = kn$ ses, at

$$n \mid ((a+c)-(b+c))$$

som omformuleres til

$$a + c \equiv b + c \pmod{n}$$

2) Dette bevis forløber ganske som 1), den vigtige udregning er

$$(a-c)-(b-c) = a-c-b+c = a-b = kn$$

3) Vi har $ca - cb = c(a-b) = ckn = (ck) \cdot n$, hvilket viser, at $n \mid (ca-cb)$.

□

Denne sætning viser, at man må selv vælge, om man beregner den principale rest modulo n før, under eller efter en regneoperation.

Regnede opgaver

Beregn: $(10203 \cdot 26173 + 193818373) \text{ MOD } 7$

Svar: Her er det lettest at beregne 7-resten før man udfører multiplikationen:

$$(10203 \cdot 26173 + 193818373) \text{ MOD } 7 =$$
$$(10203 \text{ MOD } 7) \cdot (26173 \text{ MOD } 7) + (193818373 \text{ MOD } 7) =$$
$$4 \cdot 0 + 0 = 0$$

Beregn: $2^{100000} \text{ MOD } 5$

Svar: Her kan man under ingen omstændigheder bruge sin lommeregner - prøv selv!

I stedet prøver man på følgende:

$$2^{100000} = 2^{10 \cdot 10000} = (2^{10})^{10000} = (1024)^{10000} = (-1)^{10000} = 1$$

Opgaver

2.1 Beregn

a) $635 + 2718 \pmod{19}$ b) $99 - 261 \pmod{23}$

c) $272 - 1982 \pmod{12}$ d) $372 \cdot 281 \pmod{4}$

e) $829 \cdot 729 \pmod{911}$ f) $5^{12} + 12^5 \pmod{3}$

g) $8^{1000} \pmod{9}$ h) $8^{999} \pmod{9}$

2.2 Grunden til, at man ikke generelt kan dividere indenfor modulær aritmetik, er at forkortelsesreglen

$$ac \equiv bc \pmod{n} \Rightarrow a \equiv b \pmod{n}$$

ikke altid gælder.

Giv et eksempel, hvor reglen ikke gælder. (Hjælp: Sæt $n = 6$)

3. Monoalfabetisk substitution

Som tidligere set lider den additive monoalfabetiske substitution af en fatal svaghed - den kan brydes næsten automatisk!

Dette opdagede man hurtigt, og i renæssancens Italien gik man over til mere avancerede kryptosystemer. Men allerede på dette tidspunkt kendte man til den generelle metode til kryptoanalyse af monoalfabetiske systemer - den statistiske metode. Mere herom senere.

Generelt ønsker man at kunne producere en oversættelse fra klartekstalfabetet til kryptoalfabetet, som ikke var additiv. Dette blev traditionelt gjort på to måder:

Monoalfabetisk substitution med kodeord

Nøglen er her et kodeord - typisk et ord, som er nemt at huske. Lad os vælge kodeordet `ELFENBENSKYSTEN`.
For at producere oversættelsen fra klartekstalfabetet til kryptoalfabetet opskrives nøgleordet, men bogstaver slettes, hvis de allerede optræder tidligere i ordet:

```
ELFNBSKYT
```

Herefter opskrives resten af alfabetet i sædvanlig rækkefølge

```
ELFNBSKYTACDGHIJMOPQRUVWXZÆØÅ
```

Endelig opskrives kryptoalfabetet nedenunder, men i omvendt rækkefølge:

```
ELFNBSKYTACDGHIJMOPQRUVWXZÆØÅ
ÅØÆZYXWVUTSRQPONMLKJIHGFEDCBA
```

Kryptoalfabetet opskrives i omvendt rækkefølge, idet man ellers ville få de sidste bogstaver i alfabetet, her Z, Æ Ø og Å, oversat til sig selv.
Denne metode har den fordel, at den er men at huske, og en hemmelig agent behøver ikke have nogle afslørende dokumenter på sig. Endvidere er det meget nemt at skifte kodeord.

'Mystisk' kryptoalfabet

Forfatteren lærte i sin barndom den såkaldte *frimurerkode*. Selve oversættelsen fra klartekstalfabet til kryptoalfabet var i nedenstående figur:

Hvert bogstav i klarteksten skulle erstattes med de linier og prikker, som omgav bogstavet på figuren. Eksempelvis

ØRNEN ER LANDET

Naturligvis skal man ikke lade sig forvirre af de underlige symboler - man erstatter bare hvert enkelt symbol med et bogstav, og straks er man tilbage i en almindelig monoalfabetisk substitution.
Faktisk holdt man op med at benytte disse mystiske kryptoalfabeter ved telegrafens fremkomst i forrige århundrede. Det er nemlig svært at formidle underlige symboler vha. en telegraf eller en radio eller en telefonlinie.

Kryptoanalyse af monoalfabetiske substitutioner

Det er relativt nemt at bryde monoalfabetiske substitutioner. Nøglen til succes ligger i at konstatere, at alle bogstaver ikke optræder lige hyppigt i det danske sprog. F.eks. optræder bogstavet 'e' meget hyppigt, mens 'q' eller 'x' er meget sjældne gæster.
Hyppighedsfordelingen for bogstaverne i det danske sprog er

A	0,057	G	0,043	M	0,028	S	0,053	Y	0,010
B	0,015	H	0,012	N	0,073	T	0,078	Z	0,000
C	0,003	I	0,057	O	0,051	U	0,023	Æ	0,010
D	0,055	J	0,004	P	0,024	V	0,032	Ø	0,007
E	0,163	K	0,033	Q	0,000	W	0,001	Å	0,008
F	0,025	L	0,049	R	0,087	X	0,000		

List mere overskueligt bliver det, hvis man grupperer bogstaverne efter (omtrentlig) hyppighed:

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

Dette skal forstås på den måde, at 'e' er det hyppigste bogstav; herefter kommer 'r', 'n' og 't', som er omtrent lige hyppige, osv.
Endvidere kan man optælle *digrammer* og *trigrammer*, dvs. kombinationer af to eller tre bogstaver. Det viser sig, at de hyppigste digrammer er

ER DE EN ET GE RE TE TI

og at de hyppigste trigrammer er

DET DEN ERE DER

Måden, hvorpå man kryptoanalyserer en monoalfabetisk substitueret kryptotekst er ved at lave en optælling af forekomsterne af de forskellige karakterer i kryptoalfabetet.

Med det samme kan man antage, at den hyppigste forekommende karakter svarer til klartekstkarakteren 'e'. Det er lidt sværere at tildele karakterer til 'r', 'n' og 't', men med lidt prøven sig frem kan det lade sig gøre.
Efter en del pusleri har man fået klarteksten frem.

Et ofte benyttet trick er at konstatere, at de eneste ord på dansk med 1 bogstav er 'i', 'ø' og 'å'. Med mindre der er tale om en geografitekst så er det meget sandsynligt, at den enkeltstående kryptokarakter er 'i'.

Eksempel

Vi vil kryptoanalysere teksten

```
PDT JZT FPJWDT FEPZT F TZSØRFEPHHØZÆSZR
HØSA JZP HSÆAZJZ ZAZØRTFHØZ ASJPFPW KLTZ
JZP HSÆÆZ MNT EW ZMRZT TZSØRFEPZP
```

En bogstavoptælling giver

A	D	E	F	H	J	K	L	M
4	2	5	7	6	6	1	1	2
N	P	R	S	T	W	Z	Æ	Ø
1	10	5	7	10	3	17	4	6

Med det samme ser vi, at 'Z' må svare til 'e', og ud fra ordet 'F' må bogstavet 'F' svare til 'i'.

```
PDT JZT FPJWDT FEPZT F TZSØRFEPHHØZÆSZR
     e  i      i  e  i  e   i     e  e
HØSA JZP HSÆAZJZ ZAZØRTFHØZ ASJPFPW KLTZ
      e      e e e e   i  e     i      e
JZP HSÆÆZ MNT EW ZMRZT TZSØRFEPZP
 e      e        e  e   e   i ie
```

'T' og 'P' må være enten 'n', 'r' eller 't'. uanset hvad, så optræder kombinationerne 'JZT' og 'JZP' adskillige gange - dette tyder på, at 'J' er 'd'.
Endvidere optræder kombinationen 'PFPW' i ord 11. Da 'F' er 'i', prøver vi at sætte 'P' lig 'n' og 'W' lig 'g'.

```
PDT JZT FPJWDT FEPZT F TZSØRFEPHHØZÆSZR
n   de  indg   i ne  i  e   i n   e  e
HØSA JZP HSÆAZJZ ZAZØRTFHØZ ASJPFPW KLTZ
     den     ede e e   i  e   dning    e
JZP HSÆÆZ MNT EW ZMRZT TZSØRFEPZP
den    e        e  e   e   i nen
```

Lad os prøve at sætte 'T' lig 'r':

```
PDT JZT FPJWDT FEPZT F TZSØRFEPHHØZÆSZR
n r der indg r i ner i re   i n   e  e
HØSA JZP HSÆAZJZ ZAZØRTFHØZ ASJPFPW KLTZ
     den     ede e e  ri  e   dning   re
JZP HSÆÆZ MNT EW ZMRZT TZSØRFEPZP
den     e   r    e  er re   i nen
```

Dette giver pote - det første ord kan kun være 'når', og det tredie 'indgår'. 'D' er altså 'å'.

```
PDT JZT FPJWDT FEPZT F TZSØRFEPHHØZÆSZR
når der indgår i ner i re   i n   e  e
HØSA JZP HSÆAZJZ ZAZØRTFHØZ ASJPFPW KLTZ
     den     ede e e  ri  e   dning   re
JZP HSÆÆZ MNT EW ZMRZT TZSØRFEPZP
den     e   r    e  er re   i nen
```

Man er nu godt på vej til at have gennemført kryptoanalysen.

Opgaver

3.1 Færdiggør kryptoanalysen ovenfor. (Teksten har noget med kemi at gøre!)

3.2 Kryptoanalysér følgende:

```
WHG WAQSDOHXO S WHGXDHSB CB IQG DCP ÆOH S PGHLQ
ODQKOHSPOXLOUUO HOQIULGLOH, ÆOH NSQLO, GL WSQQSCXOX
SDDO DIX HOQIULOHOH S LC DOHXOH POX CBQJ S
XOILHCXOH. GXLGUUOL GW XOILHCXOH MUO QOXOHO MOQLOPL
LSU LC OUUOH LHO, S BOXXOPQXSL LC OX FGUN
```

Bogstavfordelingen er

A	B	C	D	F	G	H	I	J	K	L
1	4	10	8	1	10	22	6	1	1	19
M	N	O	P	Q	S	U	W	X	Æ	
2	2	37	6	13	14	11	5	18	2	

3.3 Angiv mindst 3 forbedringer til et monoalfabetisk kryptosystem, så det bliver sværere at gennemføre en kryptoanalyse.

4. Algebraiske kryptosystemer

Vi skal nu se, hvorledes man kan lave monoalfabetiske substitutioner vha. modulær aritmetik. Det skal bemærkes, at disse systemer aldrig har spillet nogen praktisk rolle, idet man udmærket viste, hvorledes en monoalfabetisk substitution kunne kryptoanalyseres lang tid før, man opfandt modulær aritmetik. Men disse systemer har alligevel en vis interesse, idet de viser vejen frem til de systemer, man anvender i dag. (Og endelig er der mulighed for at liste endnu en dosis talteori ind...)

Lad os se på, hvorledes man kan kryptere telefonnumre. Vi har her alfabetet

$$\mathbf{Z}_{10} = \{0,1,2,3,4,5,6,7,8,9\}$$

nemlig de ti mulige cifre.

Man kunne f.eks. lave en additiv substitution:

$$E(a) = a + 4 \qquad \text{og} \qquad D(a) = a - 4$$

hvor vi naturligvis regner modulo 10.

Men fra tidligere ved vi, at additive substitutioner let kan brydes. Vi laver derfor et mere avanceret system, f.eks.

$$E(a) = 3a + 4.$$

Hvad er den tilhørende dekrypteringsfunktion?

Her skal vi løse ligningen

$$b = 3a + 4$$

modulo 10. Dette kan umiddelbart ikke gøres, for vi kan jo ikke nødvendigvis dividere indenfor $\mathbf{Z}_{10}$. Men følgende trick løser problemet:

$$b = 3a + 4$$
$$\Updownarrow$$
$b - 4 = 3a$ (subtraktion med 4 på begge sider)
$$\Updownarrow$$
$7(b-4) = 7 \cdot 3a$ (multiplikation med 7 på begge sider)
$$\Updownarrow$$
$$7b - 28 = 21a$$
$$\Updownarrow$$
$7b + 2 = a$ (idet $21 \equiv 1$ og $-28 \equiv 2 \pmod{10}$.

Dekrypteringsfunktionen er altså

$$D(b) = 7a + 2$$

Dette virker imidlertid ikke altid, f.eks. kan vi betragte

$$E(a) = 2a + 3$$

Denne enkrypteringsfunktion giver oversættelsestabellen:

```
0123456789
3579135791
```

og det ses, at det er umuligt at dekryptere noget som helst - optræder kryptotekstkarakteren '7', så er det umuligt at vide, om den skal oversættes til '2' eller '7'.

Det viser sig, at kun for ganske bestemte elementer x i $\mathbf{Z}_n$ kan vi omvende enkrypteringen $E(a) = xa$. Vi vil nedenfor udvikle det nødvendige matematiske apparatur:

Definition 8

Lad *a* og *b* være to hele, positive tal. Det *største fælles divisor mellem a og b*, skrevet (a,b), defineres som det største hele tal, som går op i både *a* og *b*.

Eksempel

Divisorerne i 24 er $D_{24} = \{1,2,3,4,6,8,12,24\}$

Divisorerne i 25 er $D_{25} = \{1,5,25\}$

Divisorerne i 20 er $D_{20} = \{1,2,4,5,10,20\}$

Heraf ses, at

$(24,25) = 1 \quad (24,20) = 4 \quad \text{og} \quad (20,25) = 5$

Endvidere gælder der faktisk, at

$$(24,24) = 24.$$

Ovenstående metode, hvor man finder største fælles divisorer ved at opskrive samtlige divisorer i de to tal og finde de fælles divisorer, er faktisk temmeligt bøvlet. En smartere metode er at anvende *Euklids algoritme*:

Sætning 9

Lad *a* og *b* være hele positive tal, $a \geq b$. Så gælder, at

$$(a,b) = (a \text{ MOD } b, b)$$

Bevis:

Vi sætter $q = a \text{ DIV } b$, $r = a \text{ MOD } b$ og konstaterer, at der gælder

$$a = qb + r \quad \text{eller} \quad r = a - qb$$

Selve beviset er i to dele: Først bevises, at enhver fælles divisor i *a* og *b* også er en fælles divisor i *r* og *b*. Dernæst bevises, at enhver fælles divisor i *r* og *b* er en fælles divisor i *a* og

b. Da mængden af fælles divisorer er ens for de to talpar, så må deres største fælles divisor også være den samme.

Så antag, at $x|a$ og $x|b$. Dette betyder, at der findes hele tal *k* og *l*, opfyldende

$$a = kx \qquad \text{og} \qquad b = lx .$$

Vi ved allerede, at $x|b$, og skal blot vise, at $x|r$. Men dette er nemt:

$$r = a - qb = kx - qlx = (k - ql)x .$$

Antag omvendt, at $y|r$ og $y|b$. Dette betyder, at der findes hele tal *m* og *n* opfyldende

$$r = my \qquad \text{og} \qquad b = ny .$$

Vi ved allerede, at $y|b$, og skal blot vise, at $y|a$. Men dette er nemt:

$$a = qb + r = qny + my = (qn + m)y .$$

□

Eksempel

Euklids algoritme bruges især til at finde største fælles divisor mellem store tal, og da helst på en computer. Men vi giver alligevel et eksempel:

$(165,465) =$	
$(165,135) =$	(465 MOD 165 = 135)
$(30,135) =$	(165 MOD 135 = 30)
$(30,15) =$	(135 MOD 30 = 15)
$(0,15) = 15$	(30 MOD 15 = 0)

Vi nåede til vejs ende!

Endvidere gælder der følgende sætning:

Sætning 10

Lad *a* og *b* være hele, positive tal. Så findes der hele tal *x* og *y*, således at

$$(a,b) = ax + by$$

Teknisk set siger vi, at (a,b) kan skrives som en *heltallig linearkombination* af *a* og *b*. Generelt vil et af tallene *x* eller *y* være negativt.

Bevis:

Lad $d = (a,b)$, og lad *L* være mængden:

$$L = \{xa + yb \mid x, y \in \mathbf{Z}\}$$

L er altså mængden af alle mulige heltallige linearkombinationer af *a* og *b*.
Lad endelig *e* betegne det mindste *positive* element i *L*, og skriver *e* på formen

$$e = x_0 a + y_0 b$$

Beviset går ud på at konstatere, at $d = e$, og at *d* derfor må ligge i *L* og altså er en heltallig linearkombination af *a* og *b*.

Vi viser først, at *e* er en fælles divisor i både *a* og *b*. Dette er et modstridsbevis: Vi antager, af *e* ikke går op i *a* og kan herudfra konkludere, at der findes et positivt tal i *L*, som er mindre end *e*. Dette er jo noget sludder, for *e* er jo det mindste positive element i *L*.

Antag derfor, at *e* ikke går op i *a*. Sætter vi $q = a \text{ DIV } e$ og $r = a \text{ MOD } e$, så kan vi skrive $a = qe + r$. Endvidere er $0 < r < e$ - var $r = 0$, så ville *e* gå op i *a*. Pointen er nu, at *r* er et positivt tal mindre end *e*, og at *r* faktisk ligger i *L*:

$$r = a - qe = a - q(x_0 a + y_0 b) = (1 - qx_0)a + (-qy_0)b.$$

Et tilsvarende modstridsbevis for *b* viser, at *e* går op i både *a* og *b*.

Vi viser nu, at *d* går op i *e*. Idet *d* er en fælles divisor i *a* og *b*, så findes der hele tal *k* og *l*, således at $a = kd$ og $b = ld$. Men

$$e = x_0 a + y_0 b = x_0 kd + y_0 ld = (x_0 k + y_0 l)d$$

Vi har altså, at *d* går op i *e*, hvilket kun kan lade sig gøre, hvis $d \le e$. Omvendt er *e* en fælles divisor i både *a* og *b*, og er derfor mindre end den største fælles divisor *d*, altså $e \le d$. Alt dette kan kun lade sig gøre, hvis $d = e$

□

Som vi skal se om lidt, så har man i praksis brug for at kunne skrive (a,b) som en linearkombination af *a* og *b*. Normalt er det for små tal rimeligt let at finde *x* og *y*, men for store tal og af hensyn til computerprogrammørerne vises her, hvorledes *x* og *y* kan graves frem ved at anvende Euklids algoritme bagfra. Vi tager udgangspunkt i eksemplet tidligere, hvor vi så, at $(165,465) = 15$.

Eksempel

$(165,465) = (165,135) =$	(fordi $465 = 2 \cdot 165 + 135$)
$(30,135) =$	(fordi $165 = 1 \cdot 135 + 30$)
$(30,15) =$	(fordi $135 = 4 \cdot 30 + 15$)
$(0,15) = 15$	(fordi $30 = 2 \cdot 15 + 0$)

Metoden er at skrive tallet 15 på stadigt mere komplicerede måder vha. divisionsligningerne til højre. Sidst på hver linie reduceres der, udelukkende for overskuelighedens skyld:

$$15 = 15 + 0 = 15 - (2 \cdot 15 - 30) = -15 + 30 =$$
$$-(135 - 4 \cdot 30) + 30 = -135 + 5 \cdot 30 =$$
$$-135 + 5 \cdot (165 - 135) = -6 \cdot 135 + 5 \cdot 165 =$$
$$-6 \cdot (465 - 2 \cdot 165) + 5 \cdot 165 = -6 \cdot 465 + 17 \cdot 165$$

Altså

$$15 = -6 \cdot 465 + 17 \cdot 165.$$

Vi kan nu vende tilbage til det oprindelige problem: Under hvilke omstændigheder kan vi omvende funktionen $E(a) = xa$, modulo *n*.

For $n = 10$ og $x = 3$ kunne vi jo, nemlig ved at gange med 7. Dette skyldes, at $3 \cdot 7 = 21 \equiv 1 \pmod{10}$, og vi siger, at 7 er et *multiplikativt inverst element til* 3 *modulo* 10.

Sætning 11

$a \in \mathbf{Z}_n$ har et multiplikativt inverst element modulo *n* hvis og kun hvis $(a,n) = 1$.
(I så fald kaldes *a* og *n* for *indbyrdes primiske*).

Bevis:

Ifølge sætning 10 kan vi skrive

$$xa + yn = 1$$

og regner vi modulo *n* ses, at

$$xa \equiv 1 \pmod{n}$$

I dette tilfælde er *x* altså den multiplikativt inverse til *a*.

□

Denne sætning viser også, hvorledes man vha. Euklids algoritme kan finde multiplikativt inverse elementer. I praksis kan man dog komme langt ved at sjusse sig frem:

Eksempel

Vi regner modulo 12.
For det første ser man vha. Euklids algoritme (eller sund fornuft), at 5 og 12 er indbyrdes primiske. (Faktisk er 5 et primtal, dvs. et tal uden andre divisorer en 1 og tallet selv, og 5 går ikke op i 12).

Hvad er den multiplikativt inverse til 5 modulo 12?

Vi skal altså finde et tal *x*, således at $5x \equiv 1 \pmod{12}$. VI prøver os frem ved at sætte $5x$ lig forskellige tal af formen $12n + 1$.

1. mulighed:	$5x = 1 \Rightarrow x = 0{,}2$	Nej, *x* skal jo være et helt tal.
2. mulighed:	$5x = 13 \Rightarrow x = 2{,}6$	Næææh...
3. mulighed:	$5x = 25 \Rightarrow x = 5$	Bingo!

Den multiplikativt inverse til 5 modulo 12 er (underligt nok) tallet 5 selv.

Opgaver

4.1 Hvorfor valgte forfatteren mon $\mathbb{Z}_{10}$ og ikke, som man skulle forvente, $\mathbb{Z}_{29}$, til at illustrere, at ikke alle algebraiske kryptosystemer er brugbare?

4.2 Bestem vha. Euklids algoritme:

a)	(27,4)	b)	(132,273)	c)	(4607,47)
d)	(998, 424)	e)	(1386,840)	f)	(20328,260876)

4.3 Bestem vha. Euklids algoritme (51,34).
Opskriv derefter, igen vha. Euklids algoritme, (51,34) som en linearkombination af 51 og 34.

4.4 Find samtlige tal i $\mathbb{Z}_{12}$, som er indbyrdes primiske med 12, og bestem deres multiplikativt inverse.

4.5 Find samtlige tal i $\mathbb{Z}_{7}$, som er indbyrdes primiske med 7, og bestem deres multiplikativt inverse.

4.6 Lad p være et primtal. Hvor mange, og hvilke, elementer i $\mathbb{Z}_{p}$ er indbyrdes primiske med p?

5. Polyalfabetisk substitution

Vi vil nu beskrive den *polyalfabetiske substitution*, som er en måde hvorpå man imødegår det statistiske angreb på den monoalfabetiske substitution.

I grove træk går polyalfabetisk substitution ud på, som også navnet antyder, at man anvender flere forskellige kryptoalfabeter, og veksler regelmæssigt mellem disse. Hvilke alfabeter, der bruges, bestemmes af et kodeord, eller i virkeligheden en nøgle.

Det tidligste eksempel på polyalfabetisk substitution stammer fra franskmanden Blaise de Vigenère (1523-1596). Centralt i dette system er tavlen nedenfor - til ære for Vigenère kaldes denne tavle for et *Vigenère-tableau*.

	ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ
A	ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ
B	BCDEFGHIJKLMNOPQRSTUVWXYZÆØÅA
C	CDEFGHIJKLMNOPQRSTUVWXYZÆØÅAB
D	DEFGHIJKLMNOPQRSTUVWXYZÆØÅABC
E	EFGHIJKLMNOPQRSTUVWXYZÆØÅABCD
F	FGHIJKLMNOPQRSTUVWXYZÆØÅABCDE
G	GHIJKLMNOPQRSTUVWXYZÆØÅABCDEF
H	HIJKLMNOPQRSTUVWXYZÆØÅABCDEFG
I	IJKLMNOPQRSTUVWXYZÆØÅABCDEFGH
J	JKLMNOPQRSTUVWXYZÆØÅABCDEFGHI
K	KLMNOPQRSTUVWXYZÆØÅABCDEFGHIJ
L	LMNOPQRSTUVWXYZÆØÅABCDEFGHIJK
M	MNOPQRSTUVWXYZÆØÅABCDEFGHIJKL
N	NOPQRSTUVWXYZÆØÅABCDEFGHIJKLM
O	OPQRSTUVWXYZÆØÅABCDEFGHIJKLMN
P	PQRSTUVWXYZÆØÅABCDEFGHIJKLMNO
Q	QRSTUVWXYZÆØÅABCDEFGHIJKLMNOP
R	RSTUVWXYZÆØÅABCDEFGHIJKLMNOPQ
S	STUVWXYZÆØÅABCDEFGHIJKLMNOPQR
T	TUVWXYZÆØÅABCDEFGHIJKLMNOPQRS
U	UVWXYZÆØÅABCDEFGHIJKLMNOPQRST
V	VWXYZÆØÅABCDEFGHIJKLMNOPQRSTU
W	WXYZÆØÅABCDEFGHIJKLMNOPQRSTUV
X	XYZÆØÅABCDEFGHIJKLMNOPQRSTUVW
Y	YZÆØÅABCDEFGHIJKLMNOPQRSTUVWX
Z	ZÆØÅABCDEFGHIJKLMNOPQRSTUVWXY
Æ	ÆØÅABCDEFGHIJKLMNOPQRSTUVWXYZ
Ø	ØÅABCDEFGHIJKLMNOPQRSTUVWXYZÆ
Å	ÅABCDEFGHIJKLMNOPQRSTUVWXYZÆØ

Lad os illustrere Vigenère's metode ved at enkryptere teksten

```
DETTE ER ET POLYALFABETISK SYSTEM
```

ved brug af nøglen

```
TABLEAU
```

Man starter med at skrive nøgleordet gentagne gange i en linie for sig. Nedenunder skrives klarteksten:

```
TABLE AU TA BLEAUTABLEAUTE BLEAUT
DETTE ER ET POLYALFABETISK SYSTEM
```

Endelig skriver man i tredie linie kryptoteksten. Det første bogstav enkrypteres ved at gå ind i Vigenère-tableauets T-søjle og vælge det bogstav, der står ud for D - i dette tilfælde W. Det andet bogstav bliver bogstavet ud for E i tableauets A-søjle, her E, osv.

```
TABLE AU TA BLEAUTABLEAUTE BLEAUT
DETTE ER ET POLYALFABETISK SYSTEM
WEUBI EI XT QZPYUBFAMITÅIO TGWTYC
```

Endelig plejer man for en god ordens skyld at slette alle mellemrummene:

```
WEUBIEIXTQZPYUBFAMITÅIOTGWTYC
```

Opgave 5.1

Enkryptér teksten `RØDGRØD MED FLØDE` vha. nøgleordet `HEST`. Brug Vigenère-tableauet.

Opgave 5.2

Dekryptér meddelelsen `IVE YER ZEMXS DRXX MRI, DLQVI`. Brug nøgleordet `FRUGT` og Vigenère-tableauet.

Nu er der ikke noget helligt ved Vigenère-tableauet - faktisk er der et par åbenlyse fejl. F.eks. er det uheldigt, at kryptoalfabetet i A-søjlen er lig klartekst-alfabetet!

Faktisk kan man selv lave sit tableau; det kræver blot 29 mere eller mindre tilfældige klartekstalfabeter. Men det mest udbredte tableau - og det er der en grund til - er nok Beaufort-tableauet. Dette er opkaldt efter den engelske admiral Beaufort, som også har lagt navn til Beaufort-skalaen, som angiver vindstyrke.

I Beaufort-tableauet har man vendt søjlerne i Vigenère-tableauet på hovedet:

	ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ
A	ÅABCDEFGHIJKLMNOPQRSTUVWXYZÆØ
B	ØÅABCDEFGHIJKLMNOPQRSTUVWXYZÆ
C	ÆØÅABCDEFGHIJKLMNOPQRSTUVWXYZ
D	ZÆØÅABCDEFGHIJKLMNOPQRSTUVWXY
E	YZÆØÅABCDEFGHIJKLMNOPQRSTUVWX
F	XYZÆØÅABCDEFGHIJKLMNOPQRSTUVW
G	WXYZÆØÅABCDEFGHIJKLMNOPQRSTUV
H	VWXYZÆØÅABCDEFGHIJKLMNOPQRSTU
I	UVWXYZÆØÅABCDEFGHIJKLMNOPQRST
J	TUVWXYZÆØÅABCDEFGHIJKLMNOPQRS
K	STUVWXYZÆØÅABCDEFGHIJKLMNOPQR
L	RSTUVWXYZÆØÅABCDEFGHIJKLMNOPQ
M	QRSTUVWXYZÆØÅABCDEFGHIJKLMNOP
N	PQRSTUVWXYZÆØÅABCDEFGHIJKLMNO
O	OPQRSTUVWXYZÆØÅABCDEFGHIJKLMN
P	NOPQRSTUVWXYZÆØÅABCDEFGHIJKLM
Q	MNOPQRSTUVWXYZÆØÅABCDEFGHIJKL
R	LMNOPQRSTUVWXYZÆØÅABCDEFGHIJK
S	KLMNOPQRSTUVWXYZÆØÅABCDEFGHIJ
T	JKLMNOPQRSTUVWXYZÆØÅABCDEFGHI
U	IJKLMNOPQRSTUVWXYZÆØÅABCDEFGH
V	HIJKLMNOPQRSTUVWXYZÆØÅABCDEFG
W	GHIJKLMNOPQRSTUVWXYZÆØÅABCDEF
X	FGHIJKLMNOPQRSTUVWXYZÆØÅABCDE
Y	EFGHIJKLMNOPQRSTUVWXYZÆØÅABCD
Z	DEFGHIJKLMNOPQRSTUVWXYZÆØÅABC
Æ	CDEFGHIJKLMNOPQRSTUVWXYZÆØÅAB
Ø	BCDEFGHIJKLMNOPQRSTUVWXYZÆØÅA
Å	ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ

Enkryptering og dekryptering foregår som ved Vigenère-tableauet:

Opgave 5.3

Enkryptér teksten `BRÆNDT BARN LUGTER ILDE` vha. Beaufort-tableauet og nøglen `ILD`.

Opgave 5.4

a) Dekryptér teksten `LRHJFRJAÅ` vha. Beaufort-tableauet og nøglen `OST`.

b) Enkryptér teksten `LRHJFRJAÅ` vha. Beaufort-tableauet og nøglen `OST`.

Som resultatet af opgave 5.4 antyder, så er enkryptering og dekryptering det samme i Beaufort-tableauet - en stor fordel ved praktisk anvendelse! Dette kræver vist et bevis:
Lad os udtrykke, vha. modulær aritmetik i $\mathbf{Z}_{29}$, hvorledes enkryptering og dekryptering foregår i de to tableauer. Vi enkrypterer bogstavet med værdien *a* og får kryptobogstavet *b* ud. Nøgleværdien kaldes *k*.
I Vigenère-tableauet ses, at

$$b = a + k \qquad \text{og} \qquad a = b - k$$

Sammenhængen mellem a og b er derfor ikke symmetrisk, og enkryptering og dekryptering er derfor ikke det samme.
I Beaufort-tableauet gælder derimod, at

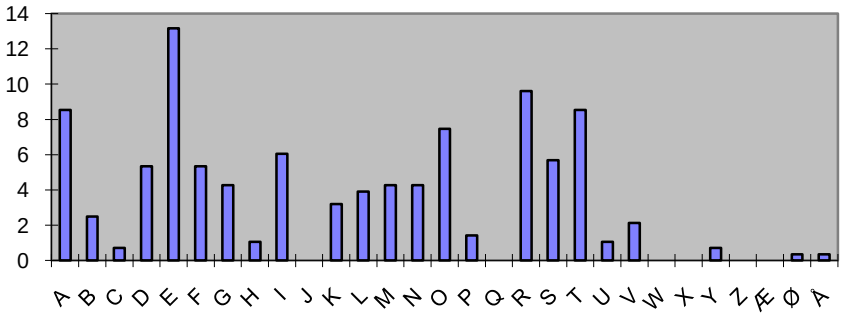
$$b = k - a - 1 \qquad \text{og} \qquad a = k - b - 1$$

og enkryptering og dekryptering er derfor det samme!

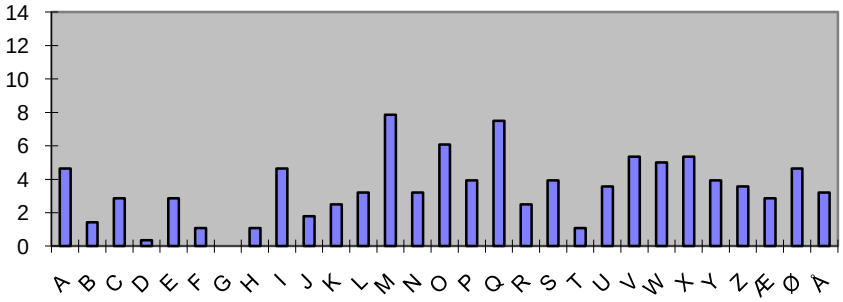
Hvad har vi egentligt fået ud af denne polyalfabetiske substitution? Jo, det statistiske angreb er blevet meget sværere at gennemføre:

Søjlediagrammerne nedenunder viser bogstavsfordelingen i en klartekst og i kryptoteksten enkrypteret med et kodeord på henholdsvis 4, 8 og 12 bogstaver.

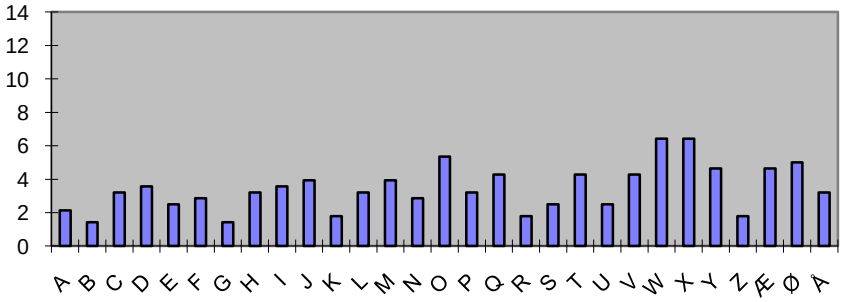
Klartekst:



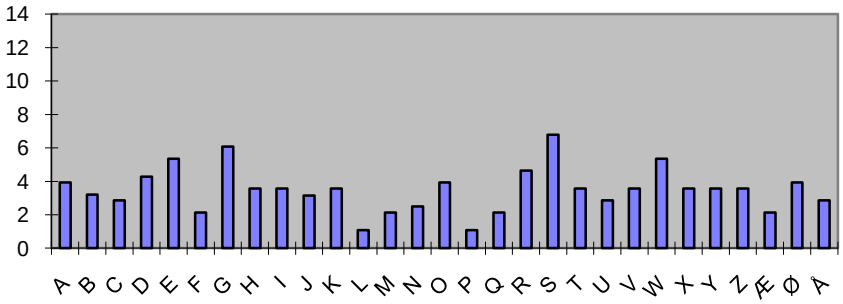
Kodeord med 4 bogstaver:



Kodeord med 8 bogstaver:



Kodeord med 12 bogstaver:



Som det ses, bliver frekvensfordelingen mere og mere jævn, efterhånden som kodeordets længde stiger.

Faktisk er det så svært at bryde en polyalfabetisk substitution, at man i forrige århundrede kaldte dette system for *le charré indechifferable* - den ubrydelige kode. Dette viste sig dog at være for godt til at være sandt - det er svært og møjsommeligt at bryde et polyalfabetisk system, men ikke umuligt.

En kryptoanalyse af en polyalfabetisk substitution består af to dele - først og fremmest skal man kende længden af nøgleordet. Dernæst opdeler man kryptoteksten i dele - en for hver bogstav i nøgleordet - således at de enkelte dele er krypteret monoalfabetisk, og man foretager her en statistisk analyse.

Lad os se lidt nærmere på det sidste skridt:

Vi har jo tidligere krypteret nedenstående tekst:

```
TABLE AU TA BLEAUTABLEAUTE BLEAUT
DETTE ER ET POLYALFABETISK SYSTEM
WEUBI EI XT QZPYUBFAMITÅIO TGWTYC
```

Her har nøgleordet længden 7. Dette betyder, at bogstaverne 1, 8, 15, 22, ... er enkrypteret vha. det samme monoalfabetiske system, nemlig alfabetet i tableauets T-søjle. Tilsvarende er bogstaverne 2, 9, 16, ... enkrypteret vha. A-søjlen.

Er kryptoteksten lang nok, så vil vi ved denne opdeling kunne få nok bogstaver i hver enkelt gruppe til at kunne lave et succesfuldt statistisk angreb på hvert enkelt monoalfabetiske system. Normalt kan dette angreb foretages blot ved at identificere 'e' som det hyppigst forekommende kryptobogstav - da alfabeterne både i Vigenère- og Beaufort-tableauet er 'additive', så giver identifikationen af 'e' hele alfabetet.

Der findes to meget forskellige måder til at finde nøgleordets længde. Den ene er opfundet af tyskeren Kasiski i slutningen af forrige århundrede og er en *ad hoc* metode, dvs. den giver kun resultater i heldige tilfælde. Den anden metode, er opfundet af amerikaneren Friedmann i 1920'erne. Den er statistisk, eller rettere sandsynlighedsteoretisk, og virker næsten altid.

Vi starter med at beskrive Kasiskis metode. Denne metode benytter, at man nogen gange kommer ud for, at gentagne bogstavssekvenser i klarteksten krypteres ens. Vi giver et eksempel:

Klarteksten

ÆBLET FALDER IKKE LANGT FRA STAMMEN, PÆREN IKKE LANGT FRA HESTEN

krypteres vha. et Vigenère-tableau på to måder - med nøglerne HAVE og ORDSPROG. Man får

```
HAVEHAVEHAVEHAVEHAVEHAVEHAVEHAVEHAVEHAVEHAVEHAVEH
ÆBLETFALDERIKKELANGTFRASTAMMENPÆRENIKKELANGTFRAHESTEN
EBDIÆFVPKEJMRKZPHNØXMRVWÆAEQLNHBYEFMRKZPHNØXMRVLLSLIU
```

og

```
ORDSPROGORDSPROGORDSPROGORDSPROGORDSPROGORDSPROGORDSP
ÆBLETFALDERIKKELANGTFRASTAMMENPÆRENIKKELANGTFRAHESTEN
LSOWFWORRVUÆZSROBJIUFOYERPBTBADCVQÆZØSROBJIUFOYSGSWWÅ
```

I begge kryptoteksterne er der markeret nogle gentagelser. I begge tilfælde stammer de gentagelsen 'ikke langt fra' i klarteksten. Vil sådanne gentagelser ikke altid opstå? Nej:

```
FRUGTFRUGTFRUGTFRUGTFRUGTFRUGTFRUGTFRUGTFRUGTFRUGTFRU
ÆBLETFALDERIKKELANGTFRASTAMMENPÆRENIKKELANGTFRAHESTEN
CSCKJKRCJXWZBQXQREMJKFUYJFADKDUOIKDNØBKBFBÆZYWRØKIYVE
```

Hér, ved brugen af nøglen FRUGT opstår gentagelserne ikke.

I de to første tilfælde opstår gentagelserne, fordi nøglen rammer ind i den gentagne klartekst på samme sted, og alle de gentagne karakterer bliver derfor krypteret ens. I det sidste tilfælde passer nøglen og den gentagne tekst ikke sammen.

Faktisk kan man let se, at forskellen i bogstaver mellem de to gentagelser er 24. Derfor vil vi få gentagelser ved nøglelængder på 4 og 8, idet disse tal går op i 24, men ikke ved nøglelængden 5.

Kasiskis metode går nu ud på, ud fra en given kryptotekst, at finde gentagelser, og finde afstanden mellem disse gentagelser. Disse afstande vil være et multiplum af nøglelængden, og man kan gå gætte på denne.
Kasiskis metode har et par bagdele. Dels kræver den relativt længde kryptotekster, før man er sikker på at der optræder gentagelser. Endvidere kan der, mere eller mindre tilfældigt, opstå gentagelser i kryptoteksten, som ikke stammer fra gentagelser i klarteksten. Erfaringen viser, at man skal kræve, at de gentagne kryptotekstfragmenter har en længde på mindst 3 og helst meget længere. Endeligt er det et meget trælsomt arbejde at finde disse gentagelser manuelt.

Friedmanns metode er meget nemmere at udføre i praksis end Kasiskis metode, og den giver et meget bedre bud på nøglelængden.

Den kinesiske vismand Sun Tzu sagde, at man altid skal angribe sin fjende på hans stærkeste sted, og dette lever Friedmanns metode klart op til. Som vi så tidligere, ligger den polyalfabetiske substitutions styrke netop i, at jo længere nøglen er, jo længere 'fladere' er frekvensfordelingen.
Fladheden af en bogstavfordeling defineres som fordelingens afvigelse fra den totalt flade, symmetriske fordeling, hvor alle bogstaver optræder lige hyppigt, og med frekvensen 1/29.

Definition 12

Fladheden F af en bogstavfordeling defineres som

$$F = \sum_{i=0}^{28} (p_i - \tfrac{1}{29})^2 ,$$

hvor p_i er frekvensen af det *i*'te bogstav.

Det lidt mystiske græske sigma betegner her en sum - hvert enkelt led i summen har formen $(p_i - \frac{1}{29})^2$, og indexet *i* går fra værdien 0 (svarende til bogstavet A), gennem værdierne 1, 2, 3, ... og ender med værdien 28.

Man kunne også have valgt andre mål for fladheden, f.eks.

$$\sum_{i=0}^{28} \left| p_i - \tfrac{1}{29} \right| .$$

Det viser sig dog, at numerisk-tegnet ovenfor er en matematisk set væmmelig størrelse, og at det er meget behageligere af arbejde med vores definition, hvor man summerer kvadraterne af de enkelte afvigelser.

Sætning 13

Fladheden for en bogstavfordeling opfylder

$$F = \sum_{i=0}^{28} {p_i}^2 \quad - \tfrac{1}{29}$$

Bevis:

Ganger vi hvert enkelt led i summen ud, så fås

$$(p_i - \tfrac{1}{29})^2 = {p_i}^2 - \tfrac{2}{29} p_i + (\tfrac{1}{29})^2$$

og ved summation

$$F = \sum_{i=0}^{28} (p_i - \tfrac{1}{29})^2 = \sum_{i=0}^{28} ({p_i}^2 - \tfrac{2}{29} p_i + (\tfrac{1}{29})^2) =$$

$$\sum_{i=0}^{28} {p_i}^2 \; - \sum_{i=0}^{28} \tfrac{2}{29} p_i + \sum_{i=0}^{28} (\tfrac{1}{29})^2 = \sum_{i=0}^{28} {p_i}^2 \; - \tfrac{2}{29} \sum_{i=0}^{28} p_i + 29 \cdot (\tfrac{1}{29})^2 =$$

$$\sum_{i=0}^{28} {p_i}^2 \; - \tfrac{2}{29} + \tfrac{1}{29} = \sum_{i=0}^{28} {p_i}^2 \; - \tfrac{1}{29}$$

hvor vi brugte, at

$$\sum_{i=0}^{28} p_i = 1$$

(summen af alle hyppighederne må jo være 1).

□

Definition 14

Index of coincidence for en bogstavfordeling defineres som

$$IC = \sum_{i=0}^{28} {p_i}^2$$

Det viser sig i sætning 13, at *Index of coincidence* og fladheden for en fordeling hænger tæt sammen. Da det er meget lettere at arbejde med *Index of coincidence*, vil vi betragte denne størrelse fremover.

Index of coincidence er engelsk og kan bedst oversættes med 'sandsynligheden for sammentræf'. Dette passer også med nedenstående tolkning af *IC*:

Sætning 15

Betragt en bogstavfordeling. Da vil sandsynligheden for, at to tilfældigt udtrukne bogstaver fra teksten er ens, være lig

$$IC = \sum_{i=0}^{28} {p_i}^2 .$$

Bevis:

Sandsynligheden for, at de to bogstaver begge er et 'A', er jo produktet for sandsynligheden for, at det ene er et 'A' og sandsynligheden for, at det andet er et 'A', dvs. $p_0 \cdot p_0 = {p_0}^2$.

Tilsvarende ses, at sandsynligheden for, at begge bogstaver er bogstav nummer *i*, er ${p_i}^2$. Adderes alle disse sandsynligheder, så fås *IC*.

□

I praksis udregnes en tilnærmelse til *IC* for en kryptotekst med *n* bogstaver med hyppighederne $h_0, h_1, h_2, \ldots$ som følger:

Sætning 16

$$IC = \frac{\sum_{i=0}^{28} h_i (h_i - 1)}{n(n-1)}$$

Bevis:

Vi udtager to tilfældige bogstaver fra kryptoteksten og beregner sandsynligheden for, at de to bogstaver er ens:
Antallet af måder, hvorpå vi kan udtage to bogstaver, er

$$K(n,2) = \frac{n!}{2! \cdot (n-2)!} = \frac{n(n-1)}{2}$$

Antallet af måder, hvorpå begge bogstaver kan være bogstav nummer *i*, er

$$K(h_i,2) = \frac{h_i(h_i-1)}{2},$$

og det totale antal måder, hvorpå de to bogstaver er ens, er

$$\sum_{i=0}^{28} K(h_i,2) = \sum_{i=0}^{28} \frac{h_i(h_i-1)}{2}.$$

Divideres disse to antal med hinanden, så fås resultatet.

□

En umiddelbar anvendelse af *IC* er en test til vurdering af, hvorvidt en kryptotekst er krypteret ved en monoalfabetisk substitution eller ej. Ved en monoalfabetisk substitution vil frekvensfordelingen af bogstaverne nemlig ikke ændre sig, hvorfor både fladheden og *IC* er uændret fra klarteksten.
For en dansk klartekst er *IC* lig 0,0701, hvilket kan beregnes ud fra definition 14 og frekvensfordelingen på side 14.
Man beregner *IC* for kryptoteksten vha. sætning 16, og ligger denne *IC* tæt på 0,0701, så er det sandsynligt, at kryptoteksten er monoalfabetisk substitueret.
Det modsatte tilfælde, hvor alle bogstaver optræder med lige stor sandsynlighed, giver en *IC* på 0,0345.

Hvorledes man kan nu anvende *IC* og Friedmanns metode til at finde nøglelængden *l* for en given kryptotekst?

Man gætter på en værdi for *l* - enten ud fra en Kasiski-analyse af teksten, eller også prøver man sig systematisk frem med alle værdier af *l*. Kryptoteksten opdeles i *l* rækker, og man beregner *IC* for hver række. Ligger alle disse *IC*'er tæt på 0,0701, så har man en monoalfabetisk substitution i hver række, og værdien af *l* er korrekt.
Eventuelt kan man beregne den gennemsnitlige værdi for hver rækkes *IC*'s afvigelse fra 0,0701,

$$\frac{1}{l}\sum_{j=1}^{l}(IC_j - 0{,}0701)^2$$

og vælge den værdi af *l*, hvor størrelsen ovenfor bliver mindst.

Opgaver

5.5 Betragt nedenstående kryptotekst:

```
ÅHÅRKXÆOGTTNLPQADFTZXHQØSYRJXVDÆHØDPOFÆØSDWPOVYQUÆOJVXEMHJVFXF
AÅYJSROXÆLWDENVECIVIYNVPQLWYOWCIBØDYTJØWÆUXØØIOWUOGRØSOFXIVXYW
YQMØDZCFPMNMIORJØECÆÅHWÆOQQLXTCÆOWULWDÆNMLMMAJGJÅXBØÆWSWWICWVX
ØKYVNMVHSYYKPMGXCTZWCHWJÆFXKQNDHDPOPXQYXTTØQQZXECPØSYIIEUWKJUJ
ATUFÅTMXÆHÅLAÅZLIBOLÅOAWBTETQVMNÆ
```

En Kasiski-analyse af denne tekst viser, at antager man en nøglelængde på 2, 4 eller 8, så findes der 3 gentagelser, og at ved andre nøglelængder findes der ingen gentagelser.
IC for teksten beregnes til 0,03615, så teksten er nok ikke krypteret monoalfabetisk.
Antager man nøglelængder på 2, 3, 4, ..., 10, så får man følgende *IC*'er for hver af rækkerne:

$l = 2$	0,04070	0,04132				
$l = 3$	0,03546	0,03389	0,03436			
$l = 4$	0,05176	0,04513	0,04472	0,04845		
$l = 5$	0,03636	0,03636	0,03117	0,03701	0,03247	
$l = 6$	0,03515	0,03423	0,03793	0,03978	0,04058	
	0,03478					
$l = 7$	0,03333	0,02308	0,04102	0,03333	0,03333	
	0,04231	0,03333				
$l = 8$	0,05546	0,06050	0,04874	0,06723	0,08403	
	0,05042	0,06891	0,06387			
$l = 9$	0,04839	0,04516	0,02581	0,03011	0,02581	
	0,04516	0,03871	0,02366	0,03011		
$l = 10$	0,03175	0,04233	0,05026	0,03704	0,03439	0,04233
	0,04762	0,03439	0,02910	0,03968		

a) Giv et bud på nøglelængden *l*.

b) Antag, at $l = 8$ og at vi har brugt et Vigenère-tableau.
De almindeligste bogstaver i de 8 rækker er da, efter hyppighed,

1. række: D, V, W, G, I, X
2. række: H, J, T, X , D, Y
3. række: C, T, O, S, Ø, D
4. række: J, F, N, P, W, T
5. række: O, X, Ø, V, Y, Å
6. række: V, W, X, F, P, Q
7. række: Q, C, M, Æ, U, A
8. række: L, M, Ø, I, N, O

Hvad er nøgleordet?
(Vink: Det er letsindigt at antage, at E svarer til det hyppigste bogstav i hver række. Men det er rimeligt at antage, at E, R, N, T befinder sig iblandt de 6 bogstaver ovenfor, og at de to sidste nok er blandt I, D, A, S, O.)

c) Dekryptér teksten.

6. En ubrydelig kode, og dens nære slægtninge

Vi har set, at polyalfabetisk substitution er relativt nem at bryde, og at dette skyldes, at man bruger et gentaget kodeord til at enkryptere klarteksten. Kan man modificere dette system, således at kodeordet ikke gentages, og opnå et sikrere system. Svaret er ja - man kan faktisk lave et ubrydeligt system - kaldet *one time pad*.

One time pad

One time pad er engelsk og betyder noget i stil med 'engangsskriveblok'. Dette er et meget passende navn, idet ubrydeligheden ligger i, at man kun anvender nøglen én gang.

Ved den polyalfabetiske substitution anvendte man en repeterende sekvens af bogstaver til at enkryptere med, f.eks.

```
     TABLE AU TA BLEAUTABLEAUTE BLEAUT
     DETTE ER ET POLYALFABETISK SYSTEM
WEUBI EI XT QZPYUBFAMITÅIO TGWTYC
```

hvor sekvensen er `TABLEAUTABLEAUTABLEAU...`

Hvad nu, hvis man i stedet anvender en sekvens af tilfældige bogstaver? En sådan sekvens kunne f.eks. laves ved at putte 29 lapper papir - hver med et af alfabetets bogstaver på - ned i en pose, og derefter trække en lap papir af gangen. Husk, at lægge lappen tilbage!
(Det viser sig, at man ikke kan lade en computer generere tilfældige bogstaver. Computeren producerer nemlig *pseudo-tilfældige* sekvenser, som ganske vist ser tilfældige ud, men absolut ikke er det. Ofte vil der være periodiske tendenser - se senere!)

Lad os antage, at vi har den tilfældige sekvens: `JÆPQABE...` Vi vil sende budskabet `KRIG` enkrypteret vha. denne sekvens. Dette sker f.eks. ved brug af Vigenère-tableauet:

```
JÆPQABE
KRIG...
BOXW...
```

Uheldigvis opsnapper fjenden krypto-meddelelsen `BOXW`. Men fjendens kryptoanalytiker kan ikke dekryptere meddelelsen uden kendskab til nøglen. Han kan jo altid antage, at nøglen er `JÆPQ` og få budskabet `KRIG` frem, men nøglen kunne jo ligeså godt være `PDXD` eller `KIMÆ`, givende budskaberne `PLAT` eller `UGLE`.
Faktisk kan man få alle mulige ord med 4 bogstaver frem ved passende brug af nøgle...
Kryptoanalytikeren må konstatere, at koden ikke kan brydes!

Uheldigvis ligger sikkerheden for *One time pad* i, at den kun bruges én gang. Lad os antage, at vi er dovne og bruger nøglen `JÆPQABE...` igen, men til at sende budskabet `FRED`. Enkrypteret bliver dette til `OOTT` - og naturligvis falder dette bidskab også i fjendens hænder.

Den fjendtlige kryptoanalytiker kan nu eliminere den ukendte nøgle! Hertil skal vi oversætte alt til modulær aritmetik. Nøglen betegnes $n_1 n_2 n_3 n_4 \ldots$ - altså en sekvens af tal fra $\mathbf{Z}_{29}$. Det

første budskab var $a_1a_2a_3a_4$, som enkrypteredes til $A_1A_2A_3A_4$, og det andet budskab var $b_1b_2b_3b_4$, som blev til $B_1B_2B_3B_4$. Vi brugte Vigenère-tableauet, så der gælder

$$A_i = a_i + n_i \qquad \text{og} \qquad B_i = b_i + n_i$$

for hvert enkelt bogstav i budskaberne. Vi trækker ligningerne fra hinanden og får

$$A_i - B_i = a_i - b_i$$

Nøglen er altså elimineret!
Kryptoanalytikeren kender nu venstresiden $A_i - B_i$, og han kan nu ved at gennemsøge en ordbog for alle muligheder for ord på 4 bogstaver finde par af ord, hvor bogstaverne passer med ligningerne ovenfor. (Normalt får han en computer til at lave dette trælse arbejde).
En mulighed, han vil finde, er KRIG og FRED, men han vil måske finde andre. Lad os estimere antallet af mulige ordpar.
Der er 29 bogstaver i alfabetet, dvs. der er $29^4 = 331776$ mulige 'ord' med fire bogstaver. Et slag på tasken viser, at der kun er ca. 3000 af disse kombinationer, der faktisk er ord! Dvs. kun ca. 1% af alle mulige kombinationer giver mening.
Kryptoanalytikeren gennemløber samtlige 3000 muligheder for det første klartekstord, og for hver mulighed undersøger han, om det andet 'ordø faktisk findes. Dette gør det i 1% af tilfældene, dvs. han får 30 muligheder for de to budskaber. Herefter er det en smal sag at finde den rigtige kombination af budskaber ud fra sammenhængen.
Denne metode virker naturligvis meget bedre ved længere budskaber.

One time pad-systemet er altså ubrydeligt, men også upraktisk! For det første fylder nøglen lige så meget som selve budskabet, for det andet kan nøglen kun bruges én gang. Problemet er bl.a., at afsenderen skal have transporteret nøglen hen til modstanderen, og dette kræver en sikker kommunikationskanal. Denne findes jo ikke, for fandtes den, så var det jo absurd at anvende kryptering!

Kodemaskiner

I stedet for at insistere på en tilfældig følge af karakterer i nøglen kunne man jo være mere beskeden og ønske sig en nøgle, som ganske vist var periodisk, men med en meget lang periode, således at Friedmanns og Kasiskis metoder ikke virker.

Dette er bl.a. ideen bag de såkaldte *kodemaskiner*, som især var udbredte under og efter 2. Verdenskrig. På forskellig vis producerer de nøgler med meget lange perioder, ofte vha. tandhjul eller elektriske himstregimser.

En af de berømteste kodemaskiner er svenskeren Boris Hägelins M-209. Denne blev bruge af vestmagterne under 2. Verdenskrig og i 50'erne. Se den relevante litteratur om emnet. Også Hitler-Tysklands *Enigma* virkede på denne måde.

Fælles for alle disse maskiner er, at de er meget vanskelige at kryptoanalysere. Det kan dog gøres, bl.a. ved angreb, hvor man kender (en del af) klarteksten. Dette kræver dog en del regnerier, og var nok svært gennemførligt for 50 år siden. Med nutidens teknologi kan en gymnasieelev bryde Hägelin M-209 på en time!

7. Transposition

En anden, klassisk krypteringsmetode bygger på *transposition*. I modsætning til substitution ændrer man her ikke på bogstavernes værdi, men på deres rækkefølge. Lad os se på et par eksempler:

Permutativ transposition

En *permutation* på *n* symboler er ombytning af de *n* elementer i $\mathbf{Z}_n$. Eksempelvis kan vi betragte permutationen på 5 symboler, som ændrer rækkefølgen af 0,1,2,3,4 til 3,1,0,4,2.

En permutation på *n* symboler giver anledning til et kryptosystem. Dette virker ved, at man opdeler sin klartekst i gruppe af *n* karakterer, og anvender permutationen på hver gruppe.

Betragter vi permutationen fra før, og klarteksten

alt har sin stund, og hver ting under himmelen sin tid

så er første skridt at opdele teksten i grupper af 5 bogstaver

altha rsins tundo ghver tingu nderh immel ensin tixxx

Her tilføjes der xxx for at fylde den sidste gruppe ud.
Hernæst anvendes permutationen på hver enkelt gruppe:

hlaat nsasi duton ehgrv gitun rdnhe emilm inens xitxx

ÜBCHI

Dette system, som reelt er en *dobbelt kolumnær transposition*, blev anvendt af tyskerne under 1. verdenskrig, og meget hurtigt brudt af franskmændene. System virker på følgende måde:

Kodeordet - her `DIE WACHT AM RHEIN` - oversættes til en talfølge ved at nummerere hvert bogstav efter bogstavernes alfabetiske orden:

```
D  I  E  W  A  C  H  T  A  M  R  H  E  I  N
4  9  5 15  1  3  7 14  2 11 13  8  6 10 12
```

Selve klarteksten - `Tenth division X attack Montigny sector at daylight X Gas barrage to precede you` - (bemærk, at X'erne bruges som punktummer) opskrives fra venstre mod højre i et skema under kodeordet:

4	9	5	15	1	3	7	14	2	11	13	8	6	10	12
t	e	n	t	h	d	i	v	i	s	i	o	n	x	a
t	t	a	c	k	m	o	n	t	i	g	n	y	s	e
c	t	o	r	a	t	d	a	y	l	i	g	h	t	x
g	a	s	b	a	r	r	a	g	e	t	o	p	r	e
c	e	d	e	y	o	y								

Herefter opskrives den første søjle (hkaay), anden søjle, etc.

hkaay ityg dmtro ttcgc naosd nyhp iodry ongo ettae
xstr sile aexe igit vnaa tcrbe

Denne proces gentages:

4	9	5	15	1	3	7	14	2	11	13	8	6	10	12
h	k	a	a	y	i	t	y	g	d	m	t	r	o	t
t	c	g	c	n	a	o	s	d	n	y	h	p	i	o
d	r	u	o	n	g	o	e	t	t	a	e	x	s	t
r	s	i	l	e	a	e	x	e	i	g	i	t	v	n
a	a	t	c	r	b	e	k	a	i	s				

og kryptoteksten bliver

ynner gdtea iagab htdra aguit rpxt tooee thei kcrsa
oisv dntii totn myags ysexk acolc

Dobbelt anagrammering

Dobbelt anagrammering er den generelle løsning til transpositions-systemer.

Et *anagram* er en ombytning af bogstaverne i et ord, f.eks. er KAT et anagram af TAK.

Lad os antage, at vi skal analysere en kryptotekst på n bogstaver, og at denne kryptotekst er en transponeret version af klarteksten. Gennemsøger vi alle muligheder, så ser vi, at der findes $n!$ mulige klartekster. Selvom måske kun 1% af disse er meningsfulde, så giver det alligevel alt for mange muligheder. Endvidere er det i praksis umuligt at opskrive alle disse muligheder.

Har vi derimod to (eller flere) kryptotekster, som er fremkommet ved den samme transposition, så kan vi lave dobbelt anagrammering. Dette virker, fordi af de $n!$ mulige transpositioner, vil kun $1\% \cdot 1\% = 0{,}01\%$ være meningsfyldte. Endvidere kan man spille de to kryptotekster ud mod hinanden, som vi senere skal se.

I praksis laver man dobbelt anagrammering ved at bruge n papirlapper. På den første lap skrive første bogstav fra hver af kryptoteksterne, osv. Problemet er nu at placere papirlapperne i en rækkefølge, således at *begge* budskaber giver mening.

Eksempel

FBI-agenterne Sculder og Mully opsnapper nogle enkrypterede budskaber:

```
     UINFBEOEYARLNDAGINRSD
VRLAROMØDRVBYEARRNEBK
```

For at starte kryptoanalysen et sted antager vi, at ordet UFO indgår i mindst af en af klarteksterne. Dette kan kun være den første, og idet bogstaverne i den anden tekst skal følge med, får vi

```
UFO
VAM
```

Et af de få ord, der starter med VAM, er VAMPYR, så den anden tekst, og dermed den første tekst, bliver

```
UFOANG
VAMPYR
```

Igen kan vi antage, at det andet ord i første linie er ANGRIBER, givende

```
UFOANGRIBER
VAMPYRERPLY
```

PLY antyder PLYNDRER, osv. osv. osv.

Opgave 7.1

Færdiggør ovenstående kryptoanalyse.

8. Polygrammisk substitution

En af vejene frem til et praktisk anvendeligt, men alligevel i praksis ubrydeligt, kryptosystem, er at substituere, ikke enkelte bogstaver, men derimod blokke af bogstaver. En blok på to bogstaver kaldes et *digram*, en blok på tre bogstaver et *trigram*, osv., så derfor er en blok på flere bogstaver ganske simpelt et *polygram*.

Playfair-systemet

Playfair-systemet blev opfundet af den engelske admiral Beaufort i midten af forrige århundrede - det er også ham, Beaufort-tableauet er opkaldt efter. Playfair var en af Beauforts gode venner, og han advokerede så meget for Beauforts system, at systemet endte med at blive opkaldt efter Playfair!

Systemet kræver et alfabet på 25 bogstaver, så vi bruger det sædvanlige engelske alfabet, hvor I og J opfattes som det samme bogstav.
Man vælger et nøgleord, og som ved den monoalfabetiske substitution opskrives nøgleordet, gentagne bogstaver slettes, og resten af alfabetet fyldes på.
Lad os vælge nøgleordet `SHAKESPEARE`.

```
SHAKEPRBCDFGILMNOQTUVWXYZ
```

Denne bogstavsekvens placeres nu i et 5x5-kvadrat:

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

Vi skal nu enkryptere en klartekst. Denne opdeles i blokke af to bogstaver, således at hvis en blok består af to ens bogstaver, så indskydes der et 'x' mellem de to bogstaver. Endelig kan man komme ud for, at den sidste blok kun består af et enkelt bogstav - her tilføjes atter et 'x'.

```
TO BE OR NOT TO BE
```

bliver således til

```
TO BE OR NO TX TO BE
```

Bogstaverne TO ligger i samme række i Playfair-kvadratet, så disse erstattes med de to bogstaver lige efter i samme række, nemlig QU:

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

BE ligger hverken i samme række eller søjle, så de to bogstaver udspænder et rektangel. Man erstatter BE med de to andre hjørner i dette rektangel, således at bogstavet i samme række som B'et kommer først: DA :

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

OR ligger i samme søjle, så man tager bogstaverne nedenunder: WG:

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

Bemærk, at søjlerne og rækkerne er cykliske, f.eks. ville BX enkrypteres til IA.

Vores besked bliver altså enkrypteret til

```
     TO BE OR NO TX TO BE
UQ DA WG OQ QY UQ DA
```

Bemærk, at klartekstbogstavet T blev oversat til både U og Q, alt efter, hvad T's makker var.

Playfair kan naturligvis brydes ved at lave statistik - ikke på de enkelte bogstaver, men på digrammerne. Desværre er dette lettere sagt end gjort - der er $25^2 = 625$ digrammer, og deres fordeling er mere flad. F.eks. er de hyppigste bogstaver på engelsk E og T med hyppigheder på 12% og 9%, mens de hyppigste digrammer på engelsk er TH og HE med hyppighederne 3,25% og 2,5%.

Algebraisk digrammisk substitution

Anvender vi modulær aritmetik, så er det meget nemt at lave et utal af digrammiske (og polygrammiske) substitutions-systemer.

Vi opfatter altså vores alfabet som $\mathbf{Z}_n$.

Klarteksten opdeles i bidder af to karakterer, og hver bid enkrypteres for sig. Kaldes de to karakterer i digrammet for *x* og *y*, så opskriver i enkrypteringsligningerne:

$$s = a_1 x + b_1 y + c_1 \pmod{n}$$
$$t = a_2 x + b_2 y + c_2 \pmod{n}$$

hvor $a_1, a_2, b_1, b_2, c_1, c_2 \in \mathbf{Z}_n$, og *s* og *t* udgør karaktererne i det enkrypterede digram.

Som i det monoalfabetiske tilfælde bør vi nu undersøge, om det er muligt at opskrive dekrypteringsligningerne, eller alternativt, om ligningssystemet ovenfor har en entydig løsning. Dette er indholdet af følgende sætning:

Sætning 17

Nedenstående ligningssystem i $\mathbf{Z}_n$

$$a_1 x + b_1 y = c_1$$
$$a_2 x + b_2 y = c_2$$

har netop en løsning hvis determinanten

$$D = \begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}$$

er indbyrdes primisk med *n*. I så fald er den entydige løsning lig

$$x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} \qquad \text{og} \qquad y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}}$$

Bevis:

Både sætning og bevis minder meget om den tilsvarende sætning fra plangeometrien om to ligninger med to ubekendte. Her er vi dog nødt til at gennemføre beviset uden brug af tværvektorer:

$$a_1 x + b_1 y = c_1 \quad \wedge \quad a_2 x + b_2 y = c_2$$
$$\Downarrow$$
$$a_1 b_2 x + b_1 b_2 y = c_1 b_2 \quad \wedge \quad a_2 b_1 x + b_2 b_1 y = c_2 b_1$$
$$\Downarrow$$
$$a_1 b_2 x + a_2 b_1 x = c_1 b_2 - c_2 b_1$$
$$\Downarrow$$

$$(a_1b_2 - a_2b_1)x = c_1b_2 - b_2c_1$$

Tilsvarende får vi ved multiplikation med henholdsvis a_2 og a_1 og efterfølgende subtraktion, at

$$(a_1b_2 - a_2b_1)y = a_1c_2 - a_2c_1$$

Det ses, at begge ligninger kan løses, netop når determinanten $a_1b_2 - a_2b_1$ har en multiplikativ invers modulo *n*.

□

DES

Det mest udbredte kryptosystem er *DES* - en forkortelse for *Data Encryption Standard*. Dette system udvikledes i 70'erne i USA, og anvendes den dag i dag til især finansielle transaktioner. Bl.a. anvender det danske *DanKort* DES til enkryptering af den 4-cifrede PIN-kode.

Vi skal ikke gå ind i detaljerne i DES, selvom de er offentlige tilgængelige.

Alfabetet i DES er det sædvanlige ASCII-alfabet, som anvendes i stort set alle computere. Der findes 256 karakterer, herunder store og små bogstaver, tal, tegn, osv.

DES enkrypterer i blokke af 8 karakterer, principielt på samme måde som beskrevet tidligere.

DES kræver et nøgleord på 7 karakterer.

Den eneste måde at bryde DES på, er at udføre en udtømmende søgning, dvs. gennemsøge alle mulige $256^7 = 7{,}2 \cdot 10^{16}$ nøgler. Dette er i praksis umuligt - tager undersøgelsen af en nøgle 1 millisekund, så vil alle nøgler blive gennemsøgt i løbet af 2,29 millioner år.

Desværre for DES bliver computere hurtigere og hurtigere, og DES er snart forældet. Man taler derfor om at indføre *triple-DES*, som er et DES-agtigt system, men som anvender en meget længere nøglelængde.

9. Public key kryptosystemer

Vi har hidtil beskæftiget os med *klassiske kryptosystemer* eller *private key kryptosystemer*. Disse betegnelser dækker over, at enkryption og dekryption er (stort set) de samme processer, og at man skal kende hele nøglen både for at kunne enkryptere og dekryptere.

I dette moderne informationssamfund er dette ikke særligt hensigtsmæssigt. Næsten al kommunikation foregår over telefonnettet (eller måske endda *Internettet*), og der er her ikke den samme sikkerhed imod uautoriseret aflytning som ved traditionelle kommunikationsmetoder.

Svaret herpå er naturligvis kryptologi - ved at kryptere sin kommunikation opnår man den fornødne sikkerhed. Men hvordan etablerer man denne kommunikation. I de klassiske kryptosystemer skal afsender og modtager udveksle nøgler, og hvordan skal denne nøgleudveksling foregå ?

Informationssamfundet stiller også nye spørgsmål. F.eks. bliver *home-banking* mere og mere udbredt. Her kommunikerer man via telefonnettet direkte med sin bank, og man kan lave forskellige finansielle transaktioner. Her er der nogle juridiske problemer? Hvordan kan banken vide, at beskeden om denne eller hin transaktion kommer fra den autoriserede bruger? Og hvordan kan banken bevise (i en retssag), at det var den autoriserede bruger, og ikke banken selv, som opkøbte alle aktierne i *Afgrund A/S* en halv time, før selskabet krakkede?

Amerikanerne Diffle og Hellman løste i 1976 disse problemer, i hvert fald i teorien. Løsningen hedder *Public key kryptosystemer* - forkortet PKK.

I et PKK vælger hver enkelt bruger *A* en nøgle. Denne nøgle producerer to procedurer, nemlig en enkryptering E_A og en dekryptering D_A. Disse skal opfylde nogle krav:

1) $E_A(D_A(M)) = M$ og $D_A(E_A(M)) = M$ for alle mulige meddelelser *M*.
2) Det skal være umuligt (eller svært) at finde D_A ud fra kendskab til E_A.
3) Det skal være umuligt (eller svært) at finde en meddelelse *M* ud fra kendskab til $E_A(M)$ og E_A.

Her betyder ordet *svært*, at det vil tage lang tid (milliarder år) at løse problemet på en computer.

E_A kaldes den *offentlige nøgle* og D_A den *private nøgle*.

Som navnet antyder er ideen med et PKK, at alle offentlige nøgler er frit tilgængelige, f.eks. som numre i en telefonbog. De private nøgler skal hver bruger naturligvis hemmeligholde.

Med et sådant system kan problemerne fra før løses:

Konfidentialitet, eller hemmeligholdelse af budskaber løses på følgende måde:
A vil sende et budskab *M* til *B*. *A* finder *B*'s offentlige nøgle E_B, og sender budskabet $E_B(M)$ til *B*.
B kan let læse budskabet - han anvender D_B på meddelelsen $E_B(M)$ og får klarteksten *M* frem.

Fjenden *C* kan ikke læse budskabet, idet han ikke kender *B*'s hemmelige nøgle D_B.

Autencitet, eller sikkerhed for afsenderens identitet, kan også opnås:
A sender budskaber $D_A(M)$ til *B*. *B* kan sagtens læse budskabet, idet han kender E_A og derfor kan læse $E_A(D_A(M)) = M$.
B kan faktisk bevise, at det var *A*, der sendte *M*. Han kan nemlig i retten fremlægge $D_A(M)$, og retten kan så overbevise sig selv om, at *M* kan opnås ud fra $D_A(M)$.
Den lumske *C* kan ikke sende et ondt budskab *N* i *A*'s navn. Han kender nemlig ikke D_A.

Men:
I det første scenario har vi ikke autencitet. *C* kan jo sagtens sende det onde budskab *N* i formen $E_B(N)$.
I det andet scenario har vi ikke konfidentialitet. Den onde *C* kender jo også E_A, og kan derfor opsnappe $D_A(M)$ og læse $E_A(D_A(M)) = M$.

Heldigvis kan vi opnå både autencitet og konfidentialitet. *A* kan jo bare sende budskabet $E_B(D_A(M))$ til *B*. *B* kender både D_B og E_A og kan derfor læse

$$E_A(D_B(E_B(D_A(M)))) = E_A(D_A(M)) = M.$$

C er helt hjælpeløs. Han kender ikke D_B og kan derfor ikke læse *M*. Og han kender ikke D_A og kan derfor ikke forfalske sit onde budskab *N*.

I praksis lyder det umuligt at konstruere et PKK. Men der findes alligevel nogle stykker. Det mest berømte og anvendte er RSA-systemet, opkaldt efter sine skabere Rivest, Shimair og Adleman. Systemet bygger på primtal og primtalsfaktorisering, som vi derfor vil se nærmere på i de næste kapitler.

10. Primtal og primtalsopløsning

Et primtal er et helt positivt tal, som ikke har andre divisorer end 1 og tallet selv.

De første primtal er

2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47...

Hvor mange primtal findes der? Jah, der findes temmeligt mange:

Sætning 18

Der findes uendeligt mange primtal.

Bevis:

Beviset er et modstridsbevis. Vi antager nemlig, at der kun findes endelig mange primtal, og at vi har en liste med dem nedenunder:

$p_1, p_2, ..., p_n$.

Vi vil nu konstruere et nyt tal, som vi kalder *M*:

$M = p_1 \cdot p_2 \cdot ... \cdot p_n + 1$

Vi ser med det samme, at *M* ikke har nogle af tallene $p_1, p_2, ..., p_n$ som divisorer - division med tallet p_i vil altid give resten 1 og ikke 0.

Enten er *M* selv et primtal, som ikke står på listen, eller også findes der primtal, som går op i *M* . Uanset hvad, så har vi fundet primtal, som ikke var i vores oprindelige liste.

□

Primtal er vigtige, fordi de er tallenes atomer - ethvert helt tal kan skrives som et produkt af primtal, og denne faktorisering kan kun laves på én måde.

Inden vi viser dette, så lad os tage et par eksempler:

Tallet 47 er selv et primtal, så 47's primtalsopløsning er 47.

Tallet 6 er lige, så 2 går op i 6. Dividerer vi 6 med 2, så får vi primtallet 3. Dvs. 6's primtalsopløsning er $6 = 2 \cdot 3$.

Tallet 16 er lige, så vi kan dividere med 2: $16 = 2 \cdot 8$. Kvotienten 8 er lige, så vi kan dividere med 2 igen: $16 = 2 \cdot 8 = 2 \cdot 2 \cdot 4$. Gentages spøgen, så fås, at

$$16 = 2 \cdot 8 = 2 \cdot 2 \cdot 4 = 2 \cdot 2 \cdot 2 \cdot 2 = 2^4.$$

Primtalsopløsningen af 24 ses at være $24 = 2^3 \cdot 3$. Vi dividerer 24 med 2, indtil divisionen ikke går op, og dividerer kvotienten med 3.

60 opløses som $2^2 \cdot 3 \cdot 5$.

Helt generelt gælder der:

Sætning 19

Ethvert helt tal *n* kan på netop én måde skrives som

$$n = p_1^{k_1} \cdot p_2^{k_2} \cdot \ldots \cdot p_j^{k_j}$$

hvor $p_1, p_2, \ldots, p_j$ er forskellige primtal.

Inden vi beviser sætningen, så får vi brug for en hjælpesætning:

Sætning 20

Hvis $(c,b) = 1$ og $c \mid ab$, så $c \mid a$.

Bevis:

Der gælder, at *c* er en divisor i ab (antagelsen i sætningen) og at *c* også er en divisor i ac.
Derfor må *c* være en divisor i $(ab,ac) = a \cdot (b,c) = a \cdot 1 = a$.

□

Bevis for sætning 19:

Antag, at vi har to forskellige primtalsopløsninger af et tal *N*:

$$N = p_1^{k_1} \cdot p_2^{k_2} \cdot \ldots \cdot p_m^{k_m} = q_1^{l_1} \cdot q_2^{l_2} \cdot \ldots \cdot q_n^{l_n}$$

Vi har så, at $p_1 \mid N$ og derfor $p_1 \mid q_1^{l_1} \cdot q_2^{l_2} \cdot \ldots \cdot q_n^{l_n}$. Nu gælder der jo, at to primtal er indbyrdes primiske, hvis de er forskellige. Sætning 20 fortæller derfor, at et af tallene q_j faktisk er p_1. Vi kan da dividere med p_1 på begge sider og få to nye primtalsopløsninger af tallet N / p_1, og hvor der er en primfaktor mindre.

Proceduren gentages: En primfaktor fra *p*-siden udvælges, sætning 20 fortæller, at denne faktor findes på *q*-siden et eller andet sted, primfaktoren divideres væk, og vi har en mindre primfaktor.

Denne spøg fortsætter, indtil der ikke er flere primfaktorer på *p*-siden. Så står der 1 hér, og *q*-siden må derfor også være 1.

Alt i alt kan vi konstatere, at vi har fjernet ens primfaktorer hele tiden fra de to originale primtalsopløsninger. Disse to opløsninger må derfor være ens.

□

I praksis gennemfører man en primtalsopløsning som beskrevet ovenfor. Man dividerer tallet *n* så mange gange som muligt med tallet 2, herefter med tallet 3, osv.

Et tallet *n* meget stort, f.eks. med 100 cifre, så kan man risikere at skulle prøve samtlige primtal under tallet *n* for at finde en divisor. Dette lyder jo ret bøvlet, så vi kan formulere den talteoretiske baggrund for RSA-systemet:

Sætning 21

Primtalsopløsning er et svært problem.

Umiddelbart ser det også ud til, at det at afgøre, om et givet tal er et primtal, er et svært problem. Det er det også, men man har fundet på forskellige tests, som sandsynliggør, med en given ønsket sandsynlighed, at et givet tal faktisk er et primtal. Detaljerne udelades.

Opgaver

10.1 Opskriv samtlige primtal mellem 1 og 100.
Du kan eventuelt bruge *Erastothenes' si*. Opskriv alle tallene mellem 1 og 100. Slet 1, da dette jo ikke er et primtal. Det første primtal er 2, så slet alle tal, som 2 går op i - dvs. hvert andet tal, startende med 4.
Det næste uslettede tal er 3, så slet hvert tredie tal, startende med 6, 9, ...

10.2 Primfaktorisér nedenstående tal:

a)	495	b)	3861	c)	266	d)	748
e)	71	f)	2116	g)	4600	h)	217

10.3 Find største fælles divisor mellem tallene *a* og *b*, givet ved

$$a = 2^{13} \cdot 3^8 \cdot 5^7 \cdot 7 \cdot 11^4 \cdot 17^9 \cdot 23$$

og

$$b = 2^5 \cdot 3^3 \cdot 5^{22} \cdot 7^7 \cdot 11^3 \cdot 13^6 \cdot 29$$

Opstil en generel metode til at bestemme to tals største fælles divisor ud fra kendskab til tallenes primfaktoriseringer.

11. Eulers φ-funktion

Vi skal nu have den sidste ingrediens til RSA-kryptosystemet, nemlig Eulers φ-funktion. Denne er opkaldt efter matematikeren Leonhard Euler, der levede i det 18. århundrede.

Definition 22

Lad *n* være et helt, positivt tal. Så defineres $\varphi(n)$ som antallet af elementer i $\mathbf{Z}_n$, der er indbyrdes primiske med *n*.

Eksempel

Ved direkte udregning ser vi, at $\varphi(6) = 2$ - der er nemlig kun tallene 1 og 5, der er indbyrdes primiske med 6.
11 er et primtal, så alle tal i $\mathbf{Z}_{11}$ på nær 0 er indbyrdes primiske med 11. Dvs. $\varphi(11) = 10$.

Der findes en generel formel til beregning af $\varphi(n)$ ud fra kendskab til primtalsopløsningen af *n*:

Sætning 23

Lad primtalsopløsningen af *n* være $n = {p_1}^{k_1} \cdot {p_2}^{k_2} \cdot \ldots \cdot {p_m}^{k_m}$.
Så er

$$\varphi(n) = {p_1}^{k_1}(1-\frac{1}{p_1}) \cdot {p_2}^{k_2}(1-\frac{1}{p_2}) \cdot \ldots \cdot {p_m}^{k_m}(1-\frac{1}{p_m})$$

Vi vil ikke bevise denne sætning, men kun nogle specialtilfælde af den:

Sætning 24

Lad *p* og *q* være to primtal.

a) $\varphi(p) = p-1$

b) $\varphi(p^n) = p^n - p^{n-1}$

c) $\varphi(pq) = (p-1) \cdot (q-1)$

Bevis:

a) Hvis *p* er et primtal, så er alle tallene i $\mathbf{Z}_p$ pånær 0 indbyrdes primiske med *p*. Men der er jo $p-1$ af disse tal.

b) Der er p^n elementer i $\mathbf{Z}_{p^n}$. Hvert *p*'ende af disse tal er divisibel med *p* og er derfor ikke indbyrdes primisk med p^n. Tilbage er der derfor

$$p^n - \frac{p^n}{p} = p^n - p^{n-1}$$

elementer, som er indbyrdes primiske med p^n.

c) Af de pq elementer, som findes i $\mathbf{Z}_{pq}$, er hvert *p*'ende divisibelt med *p* og hvert *q*'ende divisibelt med *q*. Der er altså *p* elementer, som er divisible med *p*, og *q* elementer divisible med *q*. Her har vi dog talt elementet 0 med to gange, idet 0 divideres af både *p* og *q*. Resten er indbyrdes primiske med pq, dvs.

$\varphi(pq) = pq - p - q + 1 = (p-1)(q-1)$.

□

Eulers φ-funktion er især vigtig, fordi den tillader os at beregne multiplikative inverse elementer. Hertil skal vi bruge *Eulers sætning*:

Sætning 25

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

Bevis:

Lad $r_1, r_2, r_3, \ldots, r_{\varphi(n)}$ være de $\varphi(n)$ elementer i $\mathbf{Z}_n$, som er indbyrdes primiske med *n*.

Idet både r_i og *a* er indbyrdes primiske med *n*, så er produktet ar_i indbyrdes primisk med *n*. (Både *a* og r_i har ingen fælles primfaktorer med *n*, så det samme gælder for ar_i).

Endvidere gælder, at hvis $i \neq j$, så er $ar_i \neq ar_j$. Vi har jo den multiplikativt inverse *b* til *a*, og kan derfor gøre følgende:

$ar_i = ar_j \Rightarrow bar_i = bar_j \Rightarrow r_i = r_j \Rightarrow i = j$.

Vi kan derfor konstatere, at de $\varphi(n)$ forskellige tal $ar_1, ar_2, ar_3, \ldots, ar_{\varphi(n)}$ i virkeligheden er tallene $r_1, r_2, r_3, \ldots, r_{\varphi(n)}$, bare skrevet i en anden rækkefølge. Derfor er deres produkter ens:

$r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(n)} \equiv ar_1 \cdot ar_2 \cdot \ldots \cdot ar_{\varphi(n)} \pmod{n}$

$r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(n)} \equiv a^{\varphi(n)} r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(n)} \pmod{n}$

$1 \equiv a^{\varphi(n)} \pmod{n}$.

I den sidste implikation dividerede vi med $r_1 \cdot r_2 \cdot \ldots \cdot r_{\varphi(n)}$, hvilket er tilladt, idet alle elementerne er indbyrdes primiske med *n*.

□

Sætning 26

Lad $(a,n) = 1$. Så er den multiplikativt inverse til *a* modulo *n* givet ved $a^{\varphi(n)-1}$.

Bevis:

$a \cdot a^{\varphi(n)-1} = a^{\varphi(n)} \equiv 1 \pmod{n}$.

□

Et specialtilfælde af Eulers sætning er *Fermats lille sætning*:

Sætning 27

Lad p være et primtal, og $(a, p) = 1$. Så er $a^{p-1} \equiv 1 \pmod{n}$

Bevis:

Idet p er et primtal, så er $\varphi(p) = p - 1$.

□

Eksempel

Hvad er den multiplikativt inverse til 7 modulo 24?

For det første konstaterer vi, at $(7,24) = 1$, så 7 har altså en multiplikativt invers modulo 24.

For det andet primfaktoriserer vi 24: $24 = 2^3 \cdot 3$, og anvender sætning 23 til at beregne $\varphi(24)$:

$$\varphi(24) = 2^3(1-\frac{1}{2}) \cdot 3(1-\frac{1}{3}) = 8 \cdot \frac{1}{2} \cdot 3 \cdot \frac{2}{3} = 8.$$

Endelig beregner vi

$$7^{\varphi(24)-1} = 7^{8-1} = 7^7 = 7^2 \cdot 7^2 \cdot 7^2 \cdot 7 = 49 \cdot 49 \cdot 49 \cdot 7 \equiv 1 \cdot 1 \cdot 1 \cdot 7 = 7.$$

Ergo

$$7^{-1} \equiv 7 \pmod{24}.$$

Indenfor RSA-systemet er vi interesseret i tal af formen pq, hvor p og q er forskellige primtal. Lad os derfor bevise nedenstående sætning, der ofte kalde den *kinesiske restklassesætning*.

Sætning 28

Lad $n = pq$, hvor p og q er forskellige primtal. Så gælder for alle hele tal a og b, at

$$a \equiv b \pmod{p} \ \wedge \ a \equiv b \pmod{q} \ \Rightarrow \ a \equiv b \pmod{pq}$$

Bevis:

$a \equiv b \pmod{p} \ \Rightarrow \ p \mid (b-a)$

$a \equiv b \pmod{q} \ \Rightarrow \ q \mid (b-a)$

Både p og q er altså primtalsdivisorer i $b-a$. Men da p og q er forskellige primtal, så må pq være en divisor i $b-a$, hvilket beviser sætningen.

□

Vi har nu den sidste sætning før RSA-systemet:

Sætning 29

Lad $n = pq$, hvor p og q er forskellige primtal. Da gælder for ethvert $a \in \mathbf{Z}_n$, at

$$a^{\varphi(n)+1} = a \pmod{n}.$$

Bevis:

Hvis $(a,n) = 1$, så følger resultatet direkte af sætning 25, Eulers sætning.

Hvis $(a,n) \neq 1$, så har vi to forskellige tilfælde: Enten er a divisibel med p eller også er a divisibel med q. Vi antager i det følgende, at a er divisibel med p.

I så fald gælder, at

$a \equiv 0 \pmod{p}$ og derfor $a^{\varphi(n)+1} \equiv a \pmod{p}$.

Modulo q har vi, at $(a,q) = 1$, og derfor

$$a^{\varphi(n)+1} \equiv a \cdot a^{\varphi(n)} \equiv a \cdot a^{(q-1)(p-1)} \equiv a \cdot (a^{q-1})^{p-1} \equiv a \cdot 1^{p-1} \equiv a \pmod{q}.$$

Sætning 28 giver nu det ønskede resultat.

□

Opgaver

11.1 Beregn $\varphi(n)$ for nedenstående tal. Brug sætning 23 og resultatet af opgave 10.2:

a)	495	b)	3861	c)	266	d)	748
e)	71	f)	2116	g)	4600	h)	217

11.2 a) Primfaktorisér tallet 1763 (Vink: 41)

b) Bestem $\varphi(1763)$

c) Vis, at $(21,1763) = 1$.

d) Bestem den multiplikativt inverse til 21 modulo 1763.

12. RSA-kryptosystemet

Vi kan nu endelig beskrive RSA-systemet.

Nøgle-konstruktion

Vælg to store primtal p og q - gerne med ca. 100 cifre.
Beregn $n = pq$.
Beregn $\varphi(n) = (p-1)(q-1)$
Vælg et helt tal e, således at $(e, \varphi(n)) = 1$.
Bestem den multiplikativt inverse d til e modulo $\varphi(n)$.

Den offentlige nøgle består af tallene n og e.
Den hemmelige nøgle består af tallene n og d.

En meddelelse er et tal m opfyldende $0 \le m < n$, dvs. et element i $\mathbf{Z}_n$.

Enkrypteringprocessen E_A foregår ved $E_A(m) = m^e \text{ MOD } n$.

Dekrypteringsprocessen D_A foregår ved $D_A(m) = m^d \text{ MOD } n$.

Er dette virkeligt et public key kryptosystem?

Ja, de 3 egenskaber på side 44 er opfyldt:

1) $E_A(D_A(m)) = m^{ed} = m^1 = m \pmod n$
og tilsvarende for $D_A(E_A(m))$. Se sætning 29.
2) Ja, man kan kun finde d ud fra kendskab til n og e ved at beregne $\varphi(n)$ og anvende sætning 26. Men $\varphi(n)$ kan kun beregnes ud fra kendskab til n's primfaktoropløsning, og dette er ifølge sætning 21 et svært problem.
3) Ja, denne betingelse er også opfyldt. Detaljerne udelades.

Lad os give et eksempel:

Eksempel

$p = 37$, $q = 89$, $n = 3293$ og $\varphi(n) = 3168$.
e vælges som 25, og vha. Euklids algoritme ses, at $d = 2281$.
Den offentlige nøgle er altså: $n = 3292$, $e = 25$
Den private nøgle er $n = 3292$, $d = 2281$.

Meddelelsen $m = 1118$ enkrypteres som $1118^{25} = 489$ - modulo 3292.

Opgaver

12.1 Vis, at alle detaljerne i eksemplet ovenfor er korrekte.
Vis endvidere, at $489^{2281} \equiv 1118 \pmod{3293}$.

12.2 Lav dit eget RSA-system. Vælg to små primtal, f.eks. 23 og 29, og gennemfør programmet ovenfor.

Facitliste

1.1 OMNIA GALLIA EST DIVISA PARTES TRES
('Hele Gallien er delt i tre dele' - Fra Cæsars *Gallerkrigene*)

1.2 KRYPTOLOGI ER SJOVERE END INTEGRALREGNING

2.1 a) 9 b) 22 c) 6 d) 0 e) 348 f) 1 g) 1 h) 8 (eller -1)

2.2 Eksempel: $2 \cdot 3 \equiv 4 \cdot 3 \pmod 6$ men $2 \neq 4 \pmod 6$
Generelt skal man søge efter divisorer i tallet *n*.

3.1 NÅR DER INDGÅR IONER I REAKTIONSSKEMAET SKAL DEN SAMLEDE ELEKTRISKE LADNING VÆRE DEN SAMME FØR OG EFTER REAKTIONEN

3.2 FRA FYSIKERNE I FRANKRIG OG USA KOM DER I MARTS EKSPERIMENTELLE RESULTATER, DER VISTE, AT FISSIONEN IKKE KUN RESULTERER I TO KERNER, MEN OGSÅ I NEUTRONER. ANTALLET AF NEUTRONER BLEV SENERE BESTEMT TIL TO ELLER TRE, I GENNEMSNIT TO EN HALV

3.3
1) Lad vær med at skrive mellemrum imellem ordene, men skriv teksten ud i en køre. (forhindrer bl.a. 'i'-tricket)
2) Fjern alle tegn: Kommaer, punktummer osv.
3) Brug i forvejen fastlagte forkortelser for hyppigt optrædende ord.
(I opg. 3.2 kunne man med fordel forkorte 'neutroner' med 'neu' eller endnu bedre 'xqw'. Dette sidste vil forstyrre frekvensfordelingen!)
4) Brug flere forskellige kryptotekstkarakterer for 'e'. (Dette kaldes *homofoner*). Disse karakterer kan tages fra sjældent anvendte bogstaver - f.eks. fra kryptotekstkarakteren for 'x'. 'x' kan erstattes med 'ks'.
5) Slet dobbelte konsonanter. Dobbelte -> dobelte osv.
6) Lav ikke-meningsforstyrrende stavefejl. Helsqt sådawnne, qder forztyrrer freqvenzforxdelincgen.

4.1 29 er et primtal - se opgave 4.6

4.2 a) 1 b) 3 c) 1 d) 2 e) 42 f) 3388

4.3 17 $17 = 51 - 34$

4.4 1, 5, 7, 11 alle elementerne er 'selv-inverse'

4.5 1, 2, 3, 4, 5, 6 (1, 4, 5, 2, 3, 6)

4.6 $p-1$ elementer, nemlig alle elementer i $\mathbf{Z}_p$ på nær 0

5.1 YCVZYCV CLH XBEHW

5.2 DEN SOM INGEN PÆRE HAR, ÆVLER

5.3 GWFXHM GKNX ÅLBUØT CUEG

5.4 a) CAMEMBERT b) CAMEMBERT

5.5 a) 8 b) STOFKEMI
c) KROMATOGRAFIBLEVOPFUNDETAFDENRUSSISKEBOTANIK
ERMICHAELTVESTHANBRUGTEMETODENTILATADSKILLED
EFARVESTOFFERSOMFINDESIGRØNNEBLADEKROMATOGRA
FIBETYDERATSKRIVEMEDFARVESIDENMICHAELTVESTST
IDERMETODENFORBEDRETOGDEREROPSTÅETMANGEFORSK

ELLIGEFORMERFORKROMATOGRAFIBLAGASPAPIROGTYND
TLAGSKROMATOGRAFI

7.1 ufo angriber disneyland og vampyrer plyndrer blodbank

10.1 Erathostenes' si giver

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
31	32	33	34	35	36	37	38	39	40
41	42	43	44	45	46	47	48	49	50
51	52	53	54	55	56	57	58	59	60
61	62	63	64	65	66	67	68	69	70
71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90
91	92	93	94	95	96	97	98	99	100

10.2 a) $3^2 \cdot 5 \cdot 11$ b) $3^3 \cdot 11 \cdot 13$ c) $2 \cdot 7 \cdot 19$ d) $2^2 \cdot 11 \cdot 17$
e) 71 f) $2^2 \cdot 23^2$ g) $2^3 \cdot 5^2 \cdot 23$ h) $7 \cdot 31$

10.3 $(a,b) = 2^5 \cdot 3^3 \cdot 5^7 \cdot 7 \cdot 11^3$
Generelt, hvis $a = {p_1}^{k_1} \cdot {p_2}^{k_2} \cdot \ldots \cdot {p_n}^{k_n}$ og $b = {p_1}^{l_1} \cdot {p_2}^{l_2} \cdot \ldots \cdot {p_n}^{l_n}$, så er
$(a,b) = {p_1}^{m_1} \cdot {p_2}^{m_2} \cdot \ldots \cdot {p_n}^{m_n}$ med $m_i = \min(k_i, l_i)$.

11.1 a) 240 b) 2160 c) 108 d) 320
c) 70 f) 1012 g) 1760 h) 180

11.2 a) $41 \cdot 43$ b) 1680 d) 77