Kryptologi og RSA

Jonas Lindstrøm Jensen (jonas@imf.au.dk)

1 Introduktion

Der har formodentlig eksisteret kryptologi lige så længe, som vi har haft et sprog. Ønsket om at kunne sende beskeder, som uvedkommende ikke skal kunne læse, opstår i hvert fald i mange situationer, og forskellige metoder til at opnå dette, har været anvendt op gennem historien. En klassiske metode, der går tilbage til Romerne, går ud på, at man erstatter hvert bogstav med det bogstav, der står fx tre pladser længere henne i alfabetet, og hvis man rammer enden, starter man fordra. Så istedet for ANGRIB sender man beskeden DQJULE. Afsender og modtager skal så inden blive enige om, hvor langt man rykker bogstaverne. Siden er der blevet udviklet mange metoder, og i denne lille note vil vi fokusere på en af de nyeste, nemlig RSA. Metoden er opkaldt efter opfinderne Ron Rivest, Adi Shamir og Leonard Adleman. Vi vil i denne note først præsentere nogle klassiske krypteringsmetoder, derefter forklare den nødvendige talteori for at forstå RSA, og til sidst forklare, hvorledes RSA fungerer.

2 Kryptologi

Kryptologi er læren om, hvorledes man kan maskere meddelelser, således at kun rette vedkommende skal kunne læse den. Vi vil her præsentere nogle metoder til, hvorledes det kan gøres. Afsender og modtager bliver enige om en nøgle, der bruges både til at *kryptere* og *dekryptere* beskeden, og pointen er naturligvis, at andre helst ikke skal kunne regne nøglen ud, og dermed kunne dekryptere beskeden.[1]

Cæsar-kryptering

Denne metode omtalte vi i introduktionen. Her krypteres en besked ved at hvert bogstav erstattes med det bogstav, der står n pladser pladser længere henne i alfabetet, hvor man starter forfra, hvis man når slutningen af alfabetet. Hvis $n = 3$ får vi altså

ANGRIB
DQJULE

Nøglen er her n, og modtageren rykker nu alle bogstaver n pladser tilbage, og får den oprindelige besked. Denne form for kryptering er ret nem at bryde, idet man

[1] En mere fuldstændig gennemgang af kryptologi findes i bogen "Kryptologi – fra viden til videnskab" af Peter Landrock og Knud Nissen, (ABACUS,1990).

jo bare kan prøve de 27 muligheder der er – hvis der er 28 bogstaver i alfabetet, og man rykker et bogstav 28 pladser, får man jo det samme bogstav.

Kryptering ved substitution

Her bliver afsender og modtager enige om, hvilket bogstav hvert bogstav skal ændres til. Hvis vi vælger

```
ABCDEFGHIJKLMNOPQRSTUVWXYZÆØÅ
QWERTYUIOPÅASDFGHJKLÆØZXCVBNM
```

får vi fx

```
MØD MIG VED HULEN KLOKKEN FIRE
SNR SOU ØTR IÆATD ÅAFÅÅTD YOJT
```

Denne form for kryptering er noget sværere at bryde, idet der jo er $28 \cdot 27 \cdot 26 \cdots 2 \cdot 1 = 28! \approx 8.84 \cdot 10^{30}$ forskellige måder vi kan gøre det på (hvorfor?). Dog har en uvedkommende modtager stadig en god chance for at bryde koden, idet alle bogstaver ikke optræder lige ofte i en tekst. Det mest almindelige bogstav er E, så det bogstav der optræder oftest i den krypterede tekst svarer højst sandsynlig til E, og ved at gætte sig lidt frem, kan man ofte bryde denne form for kryptering alligevel.

Viginère-kryptering

Der er blevet udviklet adskillige krypteringsmetoder, hvor man forsøger at undgå, at man kan finde E'et. En af disse er Viginère-kryptering opkaldt efter Blaise de Vigenère (1523-1596). Her er nøglen et ord, fx ABE. Dette ord skrives gentagne gange over den tekst, man vil kryptere, og hvert bogstav i teksten rykkes så det antal pladser i alfabetet, svarende til den plads bogstavet over har i alfabetet.

```
ABE ABE ABE ABEAB EABEABE ABEA
MØD MIG VED HULEN KLOKKEN FIRE
NAI NKL WGI IWQFP PMQPLGS GKWF
```

Det er lidt mere kringlet at bryde denne kode, og koder af denne type blev også benyttet under Anden Verdenskrig, fx brugte Tyskernes berømte Enigma-maskine en metode svarende lidt til denne, men dog med en nøgle, der var meget, meget længere, hvilket gjorde det sværere at bryde krypteringen. Det lykkedes at bryde krypteringen, og det fik betydning under krigen.[2]

RSA

De tre krypteringsmetoder vi har præsenteret ovenfor har det til fælles, at her er nøglen man bruger til at kryptere og dekryptere nøjagtigt den samme. Modsat hvad man skulle tro, er det dog ikke nødvendigt, at nøglerne er ens, og i RSA skal man bruge to forskellige nøgler. Det giver nogle andre anvendelsesmuligheder end med de systemer, vi hidtil har set på.

[2] Se "The Code Book" af Simon Singh for en historisk gennemgang af kryptologi og mere om Enigma-maskinen.

3 Talteori

Det er nødvendigt at kunne lidt talteori for at forstå RSA, så det vil vi præsentere i dette afsnit. Hvis du gerne vil se flere detaljer om talteori, så se Johan P. Hansen og Henrik Spalks bog "Algebra og talteori", Gyldendal (2002).

Hvis man har to positive hele tal, n og q, kan man dividere n med p. Hvis q går op i n giver det et helt tal og hvis ikke, så er der en *rest*, der nødvendigvis må være mindre end q (hvorfor?). Hvis to tal n og m giver samme rest ved division med q, så skriver vi

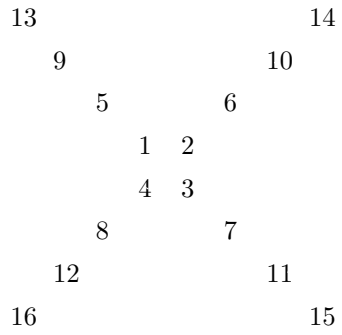
$$n \equiv m \,(\mathrm{mod}\, q)$$

og siger at n og m er kongruent modulo q.

Eksempel 1. Lad $q = 4$. Så giver 7 og 15 begge en rest på 3 ved division med 4, så derfor er

$$7 \equiv 15 \,(\mathrm{mod}\, 4).$$

Man kan også se eksemplet ovenfor sålede: Skriv tallene i en spiral, så der hver gang bruges fire tal til at nå en omgang rundt. To tal er så kongruente, hvis de ligger på samme 'arm',



Vi er i den heldige situation, at kongruens opfører sig næsten lige som det lighedstegn i kender. Fx kan man stadig bevare en kongruens, hvis man lægger det samme tal til på begge sider.

Eksempel 2. Vi ved fra ovenstående eksempel at

$$7 \equiv 13 \,(\mathrm{mod}\, 3),$$

men det gælder også at

$$8 \equiv 14 \,(\mathrm{mod}\, 3)$$

eller at

$$14 \equiv 26 \,(\mathrm{mod}\, 3).$$

Man kan desuden erstatte et tal med et vilkårligt andet tal, der giver samme rest ved division med 3, fx er

$$7 + 16 \equiv 1 + 1 \equiv 2 \,(\mathrm{mod}\, 3).$$

Vi får brug for lidt flere begreber, blandt andet følgende to.

- *Indbyrdes primiske*: To tal er indbyrdes primiske, hvis der ikke er andre tal end 1, der går op i begge tal.
- *Eulers ϕ-funktion*: For et tal n angiver $\phi(n)$ hvor mange af tallene $m = 1, 2, \ldots, n$ der er indbyrdes primisk med n.

Eksempel 3. Det ses at 4 og 9 er indbyrdes primiske, men at 4 og 6 ikke er det, idet 2 går op i begge tal. Hvis $n = 9$ ser vi, at $1, 2, 4, 5, 7$ og 8 er indbyrdes primiske med 9, så $\phi(9) = 6$.

Hvis nu p er et primtal[3], så er $\phi(p) = p - 1$ idet alle tallene $1, 2, \ldots, p - 1$ er indbyrdes primiske med p. Det gælder også at hvis p og q er forskellige primtal, så er $\phi(pq) = (p - 1)(q - 1)$.

Vi får også brug for Eulers sætning opkaldt efter Leonard Euler (1707–1783).

Sætning 4. *Lad a og n være indbyrdes primiske. Så er*

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

Eksempel 5. I sidste eksempel så vi, at $\phi(9) = 6$ og at 2 er indbyrdes primisk med 9. Så her er $a = 2, n = 9$ og $\phi(n) = 6$ og Eulers sætning siger så, at

$$2^6 \equiv 1 \mod 9.$$

Det er heldigvis sandt idet $2^6 = 64$.

Remark 6 (🖩). Hvis du har en TI-89 (eller anden grafregner fra Texas-Intrument) kan du finde resten af division af n med q ved ar bruge kommandoen `mod`, der findes under `Math`, ved at skrive `mod(n,q)`. Grafregnere af andre mærker har helt sikkert en tilsvarende funktion. Hvis du ikke kan finde den, kan du gøre således: Udregn først n/q. Hvis det ikke giver et helt tal, husk da det af resultatet der står før kommaet, kald det m. Resten kan nu findes som

[3]Husk at p er et primtal hvis kun 1 og p går op i p.

```
n - q*m
```

Hvis vi har $n = 219$ og $q = 7$ får vi

```
n/q = 31.287
```

så vi lader $m = 31$, og resten er nu

```
219 - 7*31 = 2
```

4 RSA

Vi er nu klar til at forklare, hvorledes RSA fungerer. En typisk situation hvor RSA benyttes er, hvis en kunde vil sende information til sin bank, fx i forbindelse med netbank. Husk på at ideen med systemet er, at der skal bruges forskellige nøgler til at kryptere og dekryptere en besked. På den måde kan banken gøre den nøgle, der kan bruges til at kryptere med, fuldstændigt offentlig, hvorimod de holder den anden nøgle hemmelig.

Matematisk set foregår det således: Banken vælger to store primtal, p og q, som de holder hemmelige. De offentliggører produktet af de to, $m = pq$ og et tal k der er indbyrdes primisk med $\phi(m)$. Så situationen er altså:

$$\textbf{Offentlig: } m, k \qquad \textbf{Hemmelig: } p, q, \phi(m).$$

En kunde vil gerne sende beskeden $B < m$, og sender istedet $C < m$ hvor

$$C \equiv B^k \pmod{m}.$$

Bemærk at vores meddelelse her er et tal, hvor vi tidligere så på en tekst. Det er dog ikke svært at lave en tekst om til tal – fx ved at erstatte hvert bogstav med dens plads i alfabetet.

Eksempel 7. Banken vælger primtallene 3 og 5 og offentliggører produktet 15. Idet $\phi(15) = \phi(3 \cdot 5) = (3-1)(5-1) = 8$ kan de vælge $k = 3$.

En kunde vil gerne sende beskeden 2 og sender istedet 8 idet $2^3 \equiv 8 \pmod{15}$.

Nu skal banken gerne kunne finde frem til den oprindelige besked. Først finder de to positive tal u, v således at[4]

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

og dermed

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

Banken kan nu finde den oprindelige besked B ved at udregne C^u idet

$$C^u = (B^k)^u = B^{ku} = B^{1+\phi(m)v} = BB^{\phi(m)v} = B(B^{\phi(m)})^v = B \cdot 1^v \equiv B \pmod{m}$$

hvor vi i kongruensen til sidst benytter Eulers sætning fra sidste kapitel.

[4]Det kan gøres vha. en metode kaldet *Euklids algoritme*

Eksempel 8. Banken modtager fra foregående eksempel beskeden 8. Idet $\phi(15) = 8$ skal de nu finde u, v så

$$3u - 8v = 1.$$

De kan fx vælge $u = 3$ og $v = 1$ – så er

$$8^3 = 512 \equiv 2 \pmod{15}$$

som jo var den oprindelige besked.

5 Sikkerhed

I eksemplet ovenfor, vil det være ret nemt at bryde krypteringen, for som i måske bemærkede, så er m og k offentligt kendte, og man kan dekryptere en besked hvis man kender u som findes ved hjælpe af k og $\phi(m)$. Tidligere fandt vi $\phi(m)$ for nogle små eksempler, og i teorien er det da også muligt at udregne $\phi(m)$, bare man kender m. I praksis er m dog et tal på adskillige hundrede cifre, så det vil tage alt for lang tid at prøve med alle divisorere. Den eneste grund til at banken kan udregne $\phi(m)$ er, at de kender p og q og de ved at $\phi(m) = (p-1)(q-1)$. At gange to kæmpestore tal sammen virker måske uoverskueligt, men det er enormt meget nemmere end at udregne $\phi(m)$.

En anden måde at angribe systemet på er, hvis man man finde p og q. For små tal, er det ikke så svært, fx kan man i vores eksemepl nemt se, at $15 = 3 \cdot 5$, men ligesom for beregning af $\phi(m)$, bliver det et meget tungt stykke arbejde for kæmpestore tal. Efterhånden som computere bliver hurtigere, har det være nødvendigt at øge størrelsen af m, men indtil nogen kommer op med en banebrydende ny måde at finde $\phi(m)$, er systemet ganske sikkert.

6 Opgaver

Opgave 1. *Krypter teksten* `HEJ MED DIG` *med Cæsar-kryptering og nøglen* $n = 2$.

Opgave 2. *Krypter teksten* `HEJ MED DIG` *med Viginère-kryptering hvor nøglen er* `BAD`.

Opgave 3. *Følgende er krypteret vha. substitution – altså hvor hvert bogstav erstattes af et bestemt andet bogstav. Hvad er den oprindelige tekst?*

`QWZVSMW NGBØB VZJQFBØB KLGBRBØ ZW VBØBQ CKHLEWBØB`

`XGSRBØ ZJMØBXBW ZN RSØEQ KØHB KM ZJVØB KJVQSJVBVB`

LØKMØZHHBØ QZHWSVSM BØ ZJWZGGBW ZN NKØQPM LÅ ZW

NØZJZØØB VZJQFBØJBQ FKVBØ KM ABHHBGSMB KLGUQJSJMBØ

RBV ADBGL ZN NZGQFB BHZSGQ KM ADBHHBQSVBØ QWBMBW

HZØFZJW

(Hint: Det mest brugte bogstav på dansk er E, og det næstmest almindelige er R.)

Opgave 4. *Hvilken rest får man hvis man deler*

$$\textit{(i)}\ 7\ \textit{med}\ 3, \quad \textit{(ii)}\ 13\ \textit{med}\ 7, \quad \textit{(iii)}\ 10\ \textit{med}\ 9?$$

Opgave 5. *Hvilke af følgende udsagne er sande?*

1. *(i)* $3 \equiv 5 \pmod 2$,
2. *(ii)* $12 \equiv 18 \pmod 4$,
3. *(iii)* $1001 \equiv 100000001 \pmod 5$*?*

Opgave 6. *Beregn* $\phi(10), \phi(13)$ *og* $\phi(143)$.

Opgave 7. *Vælg* $p = 3$ *og* $q = 11$ *og krypter meddelsen* $B = 3$ *med* $k = 7$*, og dekrypter derfter beskeden igen.*

Opgave 8. *Som ond hacker har du opsnappet beskeden* $C = 2$*. Du ved at* $m = 55$ *og at* $k = 7$*, da de jo var offentlige. Hvad var den oprindelige besked?*

A Euklids algoritme

A.1 Introduktion

I afsnittet om RSA skulle vi på et tidspunkt finde u, v således at

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

Husk at vi valgte k således at k og $\phi(m)$ er indbyrdes primiske, og faktisk er det altid muligt, hvis x og y er indbyrdes primiske tal, at finde u, v så

$$xu + yv = 1.$$

Det gøres ved hjælpe af Euklids algoritme. Algoritmen finder faktisk det største tal d, der går op i både x og y, og finder hele tal u og v således at

$$xu + yv = d,$$

og hvis x og y er indbyrdes primiske, er dette tal jo netop 1.

A.2 Algoritmen

Lad os illustrere metoden med et eksempel. Lad $x = 42$ og $y = 15$, og lav følgende tabel.

1	0	42	
0	1	15	

I feltet nederst til højre skriver vi nu, hvor mange gange 15 går op i 42, i dette tilfælde 2,

1	0	42	
0	1	15	2

Vi har her givet tallene farve for at vise, hvorledes de bruges til at udregne tallene i næste række. Tallene i næste række findes på følgende måde

$$1 - 2 \cdot 0 = 1,$$

$$0 - 2 \cdot 1 = -2,$$

$$42 - 2 \cdot 15 = 12,$$

og 12 går 1 gang op i 15. Tabellen ser nu således ud

1	0	42	
0	1	15	2
1	−2	12	1

Nu rykker vi alle farverne en tak ned,

1	0	42	
0	1	15	2
1	−2	12	1

og finder tallene i næste række som vi gjorde før

$$0 - 1 \cdot 1 = -1,$$

$$1 - 1 \cdot (-2) = 3,$$

$$15 - 1 \cdot 12 = 3,$$

og 3 går 4 gang op i 12. Tabellen ser nu således ud

1	0	42	
0	1	15	2
1	−2	12	1
−1	3	3	4

Vi udregner rykker nu farverne en tak,

1	0	42	
0	1	15	2
1	−2	12	1
−1	3	3	4

og finder tallene i næste række Tallene i næste række findes som før

$$1 - 4 \cdot (-1) = 5,$$

$$-1 - 4 \cdot 3 = -14,$$

$$12 - 4 \cdot 3 = 0,$$

da vi nu er nået 0 er algoritmen færdig, og tabellen kommer til at se således ud,

1	0	42	
0	1	15	2
1	−2	12	1
−1	3	3	4
5	−14	0	

Tabellen skal nu aflæses på følgende måde:

- Det største tal, der går op i både 42 og 15 er $d = 3$.
- Lad $u = -1$ og $v = 3$ idet $-1 \cdot 42 + 3 \cdot 15 = 3$.

A.3 Opgaver

Opgave 9. *Gennemfør Euklids algoritme for tallene* $x = 38$ *og* $y = 9$.

Opgave 10. *Gennemfør Euklids algoritme for tallene* $x = 32$ *og* $y = 12$.