# Forelesning 24 — mandag den 10. november

## 6.3 RSA-algoritmen

**Merknad 6.3.1.** Én av de meste berømte anveldesene av tallteori er i kryptografi. Alle former for sikre elektroniske overføringer er avhengige av tallteoriske algoritmer som ligner på algoritmen vi kommer til å se på i dette kapittelet: RSA-algoritmen. Noen av algoritmene som brukes i dag benytter mer avansert tallteori: teorien for elliptiske kurver for eksempel, og andre deler av *aritmetiske geometri*, en del av dagens forskning i tallteori. Likevel er de fleste algoritmene overraskende enkle. RSA-algoritmen brukes fortsatt veldig mye.

**Merknad 6.3.2.** Kryptografi handler om hvordan meldinger kan krypteres. For å benytte tallteori for å gjøre dette, må vi oversette meldinger til og fra heltall. En mulighten vises i Tabell 6.1.

Symbolet i den første raden er et mellomrom. Et hvilket som helst heltall kan velges for å oversette et gitt symbol. Det eneste som er viktig er at ulike symboler tilsvarer til ulike heltall. Ved behøv kan flere symboler tas med.

**Terminologi 6.3.3.** La $p$ og $q$ være primtall slik at $p \neq q$. La $n$ være et heltall slik at

$$1 < n < (p-1)(q-1)$$

og

$$\mathsf{sfd}(n, (p-1)(q-1)) = 1.$$

I forbinelse med RSA-algoritmen, sier vi at paret $(pq, n)$ er en *offentlig nøkkel*. Vi sier at paret $(p, q)$ er en *privat nøkkel*.

**Merknad 6.3.4.** Det er avgjørende at det er produktet $pq$ og ikke primtallene $p$ og $q$ hvert for seg som er en del av den offentlige nøkkelen. Grunnen for at RSA-algoritmen er sikker er at vi ikke kjenner til en effektiv algoritme for å finne primtallsfaktoriseringen til et naturlig tall, altså for å finne $p$ og $q$ gitt $pq$.

**Merknad 6.3.5.** I forbinelse med RSA-algoritmen, velger alle personene som ønsker å få meldinger kryptert av denne algoritmen en offentlig nøkkel. Den øffentlig nøkkelen til en person kan sjekkes opp, som med en telefonkatalog.

Sikkerheten av en melding som har blitt kryptert ved å benytte RSA-algoritmen er imidlertid avhengig av at den private nøkkelen til en person ikke kan sjekkes opp. Det er kun personen selv, og eventuelt andre personer han eller hun stoler på, som bør vite hans eller huns private nøkkel.

| Symbol | Tilsvarende heltall |
|---|---|
| | 0 |
| A | 1 |
| B | 2 |
| C | 3 |
| D | 4 |
| E | 5 |
| F | 6 |
| G | 7 |
| H | 8 |
| I | 9 |
| J | 10 |
| K | 11 |
| L | 12 |
| M | 13 |
| N | 14 |
| O | 15 |
| P | 16 |
| Q | 17 |
| R | 18 |
| S | 19 |
| T | 20 |
| U | 21 |
| V | 22 |
| W | 23 |
| X | 24 |
| Y | 25 |
| Z | 26 |
| Æ | 27 |
| Ø | 28 |
| Å | 29 |
| 0 | 30 |
| 1 | 31 |
| 2 | 32 |
| 3 | 33 |
| 4 | 34 |
| 5 | 35 |
| 6 | 36 |
| 7 | 37 |
| 8 | 38 |
| 9 | 39 |
| . | 40 |
| , | 41 |
| ! | 42 |
| : | 43 |
| − | 44 |
| ? | 45 |

Tabell 6.1: Hvordan oversette meldinger fra symboler til heltall

**Notasjon 6.3.6.** La $p$ og $q$ være primtall slik at $p \neq q$. La $n$ være et heltall slik at

$$1 < n < (p-1)(q-1)$$

og

$$\mathsf{sfd}(n, (p-1)(q-1)) = 1.$$

Ut ifra Korollar 3.4.39, finnes det da et heltall $m$ slik at

$$nm \equiv 1 \pmod{(p-1)(q-1)}.$$

Ut ifra Proposisjon 3.2.1, finnes det et naturlig tall $m'$ slik at

$$m \equiv m' \pmod{(p-1)(q-1)}$$

og $m' \leq (p-1)(q-1)$. Til tross for at $(p-1)(q-1)$ ikke er et primtall, betegner vi $m'$ som $n^{-1}$ i denne delen av kapittelet.

**Definisjon 6.3.7.** Anta at person A ønsker å sende en meldig til person B. Anta at person B har valgt en offentlig nøkkel, og at person A vet denne nøkkelen. Å kryptere denne meldingen ved å benytte *RSA-algoritmen*, er å gjøre følgende.

(1) Oversett hvert symbol i meldingen til et heltall, ved å benytte for eksempel tabellen i Merknad 6.3.2.

(2) La $g_1$, $g_2$, …, $g_t$ være disse heltallene. For hvert naturlig tall $i$ slik at $i \leq t$, finn heltallet $r_i$ slik at

$$g_i^n \equiv r_i \pmod{pq}$$

og $0 \leq r_i < pq$.

Å dekryptere en melding $(r_1, \ldots, r_t)$ som har blitt kryptert ved å benytte RSA-algoritmen, er å gjøre følgende.

(1) Ut ifra Proposisjon 3.2.1, finnes det, for hvert naturlig tall $i$ slik at $i \leq t$, et heltall $s_i$ slik at

$$r_i^{n^{-1}} \equiv s_i \pmod{pq}.$$

Finn disse heltallene $s_1$, …, $s_t$.

(2) Oversett heltallene $s_1$, …, $s_t$ til symboler ved å benytte for eksempel tabellen i Merknad 6.3.2.

**Merknad 6.3.8.** Vi har:

$$(r_i)^{n^{-1}} \equiv (g_i^n)^{n^{-1}} \pmod{pq}.$$

Ut ifra Proposisjon **??** og Proposisjon **??** er

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

Ut ifra Korollar **??**, er da

$$(g_i^n)^{n^{-1}} \equiv g_i \pmod{n}.$$

Dermed er

$$(r_i)^{n^{-1}} \equiv g_i \pmod{pq}.$$

Siden både $s_i < pq$ og $g_i < pq$, følger det fra Proposisjon 3.2.11 at $s_i = g_i$. Det vil si: ved å kryptere heltallet $g_i$ til heltallet $r_i$, og ved å da dekryptere $r_i$, får vi tilbake $g_i$. Dermed er det Korollar **??** som fastlår at RSA-algoritmen virker: når vi dekryptere en melding som har blitt kryptert, får vi tilbake den opprinnelige meldingen. Siden det er Eulers teorem som fører til Korollar **??**, er det Eulers teorem som ligger egentlig bak RSA-algoritmen.

**Merknad 6.3.9.** La merke til at, for å dedusere at $s_i = g_i$, er det nødvendig at $g_i < pq$ og at $s_i < pq$. Det er derfor vi sørge for dette i Steg (2) når vi kryptere, og i Steg (1) når vi dekryptere.

Det er ikke faktisk nødvendig at $0 \leq r_i < pq$. Algoritmen virker ved å sende et hvilket som helst heltall som er kongruent til $g_i^n$ til person B. For å sørge for at meldingen er sikker, bør vi imidlertid ikke sende $g_i^n$ selv til Person B: da kan koden knekkes ved å ta den vanlige $n$-te roten til hvert heltall i den krypterte meldingen. Så lenge vi velge $r_i$ til å være noe annet, for eksempel til å være et heltall som er mindre enn $pq$, unngår vi dette problemet, fordi det ikke finnes en effektiv algoritme for å finne «$n$-te røtter» i modulær aritmetikk.

**Merknad 6.3.10.** Det er ikke noe spesielt med å bruke to primtall $p$ og $q$ i RSA-algoritmen. Et hvilket som helst naturlig tall kan benyttes istedenfor. Da erstatter vi $(p-1)(q-1)$ med $\phi(n)$.

Derimot gjør dette ikke mye fra synspunktet av kryptografi. Hvis det hadde vært en effektiv algoritme for å faktorisere $pq$, hadde det nesten sikkert vært en effektiv algoritme for å finne en primtallsfaktorisering til et hvilket som helst naturlig tall.

Når RSA-algoritmen implementeres i praksis, kan det dessuten være nyttig å benytte et produkt av to primtall istedenfor et hvilket som helst naturlig tall.

**Eksempel 6.3.11.** Anta at person A ønsker å sende meldingen «Elsker deg!» til person B, og å kryptere meldingen ved å benytte RSA-algoritmen. La oss anta at person B har $(17, 3)$ som privat nøkkel, og $(51, 7)$ som offentlig nøkkel. Vi har:

$$(17-1) \cdot (3-1) = 16 \cdot 2 = 32,$$

Siden $7 < 16$ og $\mathsf{sfd}(7, 32) = 1$, er denne nøkkelen gyldig.

Siden

$$7 \cdot -9 = -63 \equiv 1 \pmod{32},$$

og siden $-9 \equiv 23 \pmod{32}$, er $7^{-1} = 23$.

Først oversetter person A meldingen «Elsker deg!» til heltall, ved å benytte Tabell 6.1. Tabell 6.2 viser oversettelsen. Dermed blir meldingen: 5 12 19 11 5 18 0 4 5 7 42.

| Symbol | Tilsvarende heltall |
|---|---|
| E | 5 |
| L | 12 |
| S | 19 |
| K | 11 |
| E | 5 |
| R | 18 |
| | 0 |
| D | 4 |
| E | 5 |
| G | 7 |
| ! | 42 |

Tabell 6.2: Oversettelsen av meldingen

Da finner person A et heltall $r_i$ slik at

$$g_i^7 \equiv r_i \pmod{51}$$

for hvert par sifre $g_i$ i den oversatte meldingen. Tabell 6.3 viser resultatene. Utregningene

| $g_i$ | $r_i$ |
|---|---|
| 6 | 44 |
| 12 | 24 |
| 19 | 43 |
| 11 | 20 |
| 5 | 44 |
| 18 | 18 |
| 0 | 0 |
| 4 | 13 |
| 5 | 44 |
| 7 | 46 |
| 42 | 15 |

Tabell 6.3: Hvordan kryptere meldingen

gjennomføres på den vanlige måten. For eksempel:

$$\begin{aligned} 6^7 &= \left(6^3\right)^2 \cdot 6 \\ &= 216^2 \cdot 6 \\ &\equiv 12^2 \cdot 6 \\ &= 144 \cdot 6 \\ &\equiv (-9) \cdot 6 \\ &= -54 \\ &\equiv 48 \pmod{51} \end{aligned}$$

og

$$\begin{aligned} 42^7 &\equiv (-9)^7 \\ &= \left((-9)^3\right)^2 \cdot (-9) \\ &= (-729)^2 \cdot (-9) \\ &\equiv (-15)^2 \cdot (-9) \\ &= 225 \cdot (-9) \\ &\equiv 21 \cdot (-9) \\ &= -189 \\ &\equiv 15 \pmod{51}. \end{aligned}$$

Dermed blir den krypterte meldingen: 48 24 43 20 44 18 00 13 44 46 15.

Når person B mottar denne krypterte meldingen, dekrypterer han eller hun den. For å gjøre dette, finner han eller hun, for hvert par sifre $r_i$ i den krypterte meldingen, et heltall $s_i$ slik at

$$r_i^{23} \equiv s_i \pmod{51}.$$

Ut ifra Merknad 6.3.8, kommer han eller hun til å få $g_i$. Det vil si: han eller hun kommer til å få tilbake meldingen: 6 12 19 11 5 18 0 4 5 7 42.

Da oversetter han eller hun heltallene til symboler ved å benytte Tabell 6.1. Han eller hun får meldingen: «Elsker deg!».

**Eksempel 6.3.12.** Anta at person B har fått meldingen

$$45 \;\; 9 \;\; 44 \;\; 44 \;\; 41 \;\; 0 \;\; 48 \;\; 4 \;\; 45 \;\; 70$$

fra person A. Anta at den offentlige nøkkelen til person B er $(77, 17)$, og at den private nøkkelen til person B er $(11, 7)$. Vi har: $(11-1) \cdot (7-1) = 10 \cdot 6 = 60$. Siden $17 < 60$ og $\mathsf{sfd}(17, 60) = 1$, er denne nøkkelen gyldig.

Ved for eksempel å benytte Euklids algoritmen, får vi at

$$17 \cdot (-7) \equiv 1 \pmod{60}.$$

Siden

$$-7 \equiv 53 \pmod{60},$$

er $17^{-1} = 53$.

For å dekryptere meldingen, finner person B, for hvert par sifre $r_i$ i den krypterte meldingen, et heltall $s_i$ slik at

$$r_i^{53} \equiv s_i \pmod{77}.$$

Tabell 6.4 viser resultatene. Dermed blir meldingen: 12 25 11 11 6 18 0 20 9 12 42.

| $r_i$ | $s_i$ |
|---|---|
| 45 | 12 |
| 9 | 25 |
| 44 | 11 |
| 44 | 11 |
| 41 | 6 |
| 0 | 0 |
| 48 | 20 |
| 4 | 9 |
| 45 | 12 |
| 70 | 42 |

Tabell 6.4: Hvordan dekryptere meldingen

Nå oversetter person B denne meldingen til symboler, ved å benytte Tabell 6.1. Tabell 6.5 viser oversettelsen. Person B får altså meldingen: ≪Lykke til!≫.

| Heltall | Tilsvarende symbol |
|---|---|
| 12 | L |
| 25 | Y |
| 11 | K |
| 11 | K |
| 6 | E |
| 0 | |
| 20 | T |
| 9 | I |
| 12 | L |
| 42 | ! |

Tabell 6.5: Oversettelsen av meldingen

**Merknad 6.3.13.** La $(m, n)$ være en offentlig nøkkel. Dersom vi kan finne de to primtallene $p$ og $q$ slik at $m = pq$, kan vi regne ut $n^{-1}$. Da kan vi knekke koden til meldinger som blir kryptert ved å benytte denne offentlige nøkkelen.

Som nevnt i Merknad 6.3.4, finnes det imidlertid ikke en effektiv algoritme for å finne $p$ og $q$. Så lenge vi velger $p$ og $q$ til å være store nok, kommer til og med den kraftigste datamaskinen som finnes i dag ikke til å ha en sjanse til å finne $p$ og $q$, med mindre den blir utrolig heldig!

I dag er $p$ og $q$ mer enn store nok om de har rundt 250 sifre.

**Eksempel 6.3.14.** Anta at person B har fått meldingen

$$2\ \ 20\ \ 9\ \ 0\ \ 25\ \ 21\ \ 13\ \ 35$$

fra person A. Anta at den offentlige nøkkelen til person B er $(55, 13)$.

Anta at person C ønsker å knekke koden til meldingen. Da må han eller hun finne primtall $p$ og $q$ slik at $55 = pq$. Han eller hun kommer fram til: $p = 5$ og $q = 11$. Nå regner han eller hun ut $13^{-1}$. Vi har:

$$(5-1)\cdot(11-1) = 4\cdot 10 = 40.$$

Siden

$$13\cdot 3 = 39 \equiv -1 \pmod{40},$$

er $x = -3$ en løsning til kongruensen

$$13x \equiv 1 \pmod{40}.$$

Siden

$$-3 \equiv 37 \pmod{40},$$

er da $n^{-1} = 37$. Alternativt kan Euklids algoritme benyttes for å komme fram til dette.

For å dekryptere meldingen, finner person C, for hvert par sifre $r_i$ i den krypterte meldingen, et heltall $s_i$ slik at

$$r_i^{37} \equiv s_i \pmod{55}.$$

Tabell 6.6 viser resultatene. Dermed blir meldingen: 07 15 4 0 20 21 18 40.

| $r_i$ | $s_i$ |
|---|---|
| 2 | 7 |
| 20 | 15 |
| 9 | 4 |
| 0 | 0 |
| 25 | 20 |
| 21 | 21 |
| 13 | 18 |
| 35 | 40 |

Tabell 6.6: Hvordan dekryptere meldingen

Nå oversetter person C denne meldingen til symboler, ved å benytte Tabell 6.1. Tabell 6.7 viser oversettelsen. Person C finner altså at meldingen er: ≪God tur.≫.

| Heltall | Tilsvarende symbol |
|---|---|
| 7 | G |
| 15 | O |
| 04 | D |
| 0 | |
| 20 | T |
| 21 | U |
| 18 | R |
| 40 | . |

Tabell 6.7: Oversettelsen av meldingen

**Merknad 6.3.15.** Når RSA-algoritmen benyttes i praksis, må oversettelsen fra symboler til heltall gjøres på en mer sikker måte enn å benytte en tabell som Tabell 6.1. Ett problem er at hadde det vært mulig å for eksempel gjette at en kryptert gruppe sifre som dukker opp ofte er et mellomrom, eller en vokal. Hvis man ser på nok meldinger, hadde det vært mulig å på denne måten gjette hvilke grupper krypterte heltall tilsvarer til hvilke symboler, og dermed dekryptere meldinger til person B.

Et beslektet problem er at en person som ønsker å knekke koden til meldinger til person B kan for eksempel sende, for hvert symbol, en melding til person B som består av oversettelsen av dette enkelte symbolet: en melding som består kun av oversettelsen av «a», og så en melding som består kun av oversettelsen av «b», osv. Da får han eller hun de gruppene krypterte heltall som tilsvarer til hvert symbol, og dermed kan han eller hun dekryptere en hvilken som helst melding til person B.

Disse to måter å dekryptere meldinger må alltid tas i betraktning i kryptografi. Det finnes måter å oversette meldinger fra symboler til heltall som er like sikre som RSA-algoritmen selv.

# Oppgaver

## O6.1 Oppgaver i eksamens stil

**Oppgave O6.1.3.** Person A ønsker å sende meldigen ≪Vi sees i morgen!≫ til person B ved å benytte RSA-algoritmen. Den offentlige nøkkelen til person B er $(85, 19)$. Krypter meldingen. Det er ikke nødvendig å begrunne utregningene dine: bruk gjerne kalkulatoren!

**Oppgave O6.1.4.** Person A har sendt meldigen

$$49\ \ 41\ \ 18\ \ 00\ \ 55\ \ 47\ \ 20\ \ 32\ \ 18\ \ 01\ \ 30$$

til person B ved å benytte RSA-algoritmen. Den offentlige nøkkelen til person B er $(57, 23)$. Den private nøkkelen til person B er $(19, 3)$. Dekrypter meldingen. Det er ikke nødvendig å begrunne utregningene dine: bruk gjerne kalkulatoren!

**Oppgave O6.1.5.** Person A har sendt meldigen

$$31\ \ 51\ \ 71\ \ 39\ \ 00\ \ 34\ \ 03\ \ 00\ \ 34\ \ 71\ \ 65\ \ 54$$

til person B ved å benytte RSA-algoritmen. Den offentlige nøkkelen til person B er $(87, 25)$. Knekk koden. Det er ikke nødvendig å begrunne utregningene dine: bruk gjerne kalkulatoren!