RSA-kryptering

Torbjörn Tambour

RSA-metoden för kryptering har den speciella och betydelsefulla egenskapen att metoden för kryptering är offentlig, medan metoden för dekryptering är hemlig. Detta kan om man funderar lite verka märkligt; om man vet hur man krypterar, så borde man väl veta hur man går baklänges, dvs dekrypterar, också, men så är det alltså inte. Den allmänna principen för dekrypteringen är känd av alla, men den är beroende av vissa tal som hålls hemliga och som utomstående inte kan räkna ut eller åtminstone har synnerligen stora svårigheter att räkna ut. Metoden bygger på ett resultat i talteorin, nämligen Fermats lilla sats. Vi har således ett exempel på ett resultat i den mer abstrakta delen av matematiken som oväntat får en konkret och viktig tillämpning. Akronymen RSA kommer av namnen på uppfinnarna, Ron Rivest, Adi Shamir och Len Adleman, och metoden publicerades 1979.

1 Kongruenser

Vi skall börja med att repetera *kongruenser*. Låt n vara ett fixt positivt heltal. Två heltal a och b säges vara kongruenta modulo n om $a - b$ är delbart med n. Man skriver detta $a \equiv b \mod n$. Observera att $a \equiv b$ är detsamma som att $a = b + kn$ för något heltal k. Ett annat sätt att formulera detta är att två tal kongruenta modulo n om och endast om de ger samma rest vid division med n. Om a ger resten r, så är $a \equiv r \mod n$ och det följer att givet ett tal a så finns precis ett tal r sådant att $0 \leq r \leq n - 1$ sådant att $a \equiv r \mod n$, nämligen resten vid division med n (ty om $|r - s| < n$ och $r \equiv s \mod n$, så måste $r = s$).

Sats 1 *Följande gäller för kongruenser:*

a) $a \equiv a \mod n$ *för alla* a

b) om $a \equiv b \mod n$*, så är* $b \equiv a \mod n$

c) om $a \equiv b$ *och* $b \equiv c \mod n$*, så är* $a \equiv c \mod n$

d) om $a \equiv b$ *och* $c \equiv d \mod n$*, så gäller* $a \pm c \equiv b \pm d$ *och* $ac \equiv bd \mod n$

e) om $a \equiv b$*, så är* $a^k \equiv b^k \mod n$ *för alla heltal* $k \geq 0$
f) om $SGD(a, n) = 1$*, så finns ett heltal* x *sådant att* $ax \equiv 1 \mod n$*.*

Bevis. De tre första påståendena är mer eller mindre självklara. Om $a \equiv b, c \equiv d$, så kan vi skriva $a = b + kn, c = d + ln$ för några heltal k, l. Alltså är $a \pm c = b \pm d + (k \pm l)n$ och första delen av d) följer. Multiplikation ger istället $ac = bd + (bl + dk + kln)n$, varav $ac \equiv bd$. Upprepad användning av d) ger e). Del f) är betydligt intressantare. Enligt Euklides algoritm finns heltal x, y sådana att $ax + ny = 1$. Modulo n betyder detta att $ax \equiv 1$.

Talet x i e) brukar kallas för *inversen till* a *modulo* n och en vanlig beteckning är $x = a^{-1}$ (här måste det alltså vara underförstått vilket n som ligger i botten). Men definierar förstås $a^{-k} = (a^{-1})^k$ när k är ett positivt heltal och det är lätt att visa att de vanliga potenslagarna gäller.

Sats 2 (Fermats lilla sats) *Om* p *är ett primtal och* a *ett heltal som inte är delbart med* p*, så gäller att* $a^{p-1} \equiv 1 \mod p$*.*

En ekvivalent formulering är att $a^p \equiv a \mod p$ för alla a. I sista avsnittet finns inte mindre än två bevis för Fermats lilla sats. Vi kommer också att behöva

Sats 3 *Låt* p *och* q *vara två olika primtal och låt* M *vara ett tal som är delbart med både* $p - 1$ *och* $q - 1$*. Om* a *inte är delbart med* pq*, så gäller att* $a^M \equiv 1 \mod pq$*.*

Bevis. Vi har $a^M = (a^{p-1})^{M/(p-1)} \equiv 1^{M/(p-1)} \equiv 1 \mod p$ enligt Fermats lilla sats eftersom M är delbart med $p - 1$ (lägg märke till att detta villkor är väsentligt!). På samma sätt får vi $a^M \equiv 1 \mod q$. Alltså är $a^M - 1$ delbart med både p och q. Men p och q är olika primtal och då måste $pq | a^M - 1$, dvs $a^M \equiv 1 \mod pq$.

2 RSA

Vi kan nu beskriva RSA-metoden. Vi har alltså ett meddelande som vi vill kryptera och först måste vi översätta det till ett tal. Exempelvis kan vi låta A vara 01, B 02 osv. Antag att meddelandet är a översatt till ett tal. Vi låter p och q vara två olika primtal och antar till att börja med att $1 \leq a \leq pq$. Låt M vara minsta gemensamma multipeln till $p - 1$ och $q - 1$. Tag ett tal k sådant att $SGD(k, M) = 1$ och låt l vara sådant att $kl \equiv 1 \mod M$. Vi har nu

- *Kryptering:* $a \mapsto a^k \mod pq$
- *Dekryptering:* $b \mapsto b^l \mod pq$

Här betyder $a^k \mod pq$ resten av a^k vid division med pq. Innan vi börjar reda ut varför detta fungerar skall vi ta ett enkelt exempel. Låt $p = 11, q = 13$. Då är $M = MGM(10, 12) = 60$. Tag t ex $k = 7$. Låt meddelandet vara $a = 42$. Det krypterade meddelandet får vi genom att reducera 42^7 modulo 143. En stunds räknande (med hjälp av reglerna i Sats 1) ger att $42^7 \equiv 81 \mod 143$. Det krypterade meddelandet är således $b = 81$.

För att kunna dekryptera detta måste vi finna ett tal l så att $7l \equiv 1 \mod 60$. Vi skall alltså lösa den diofantiska ekvationen $7l + 60m = 1$ och som vanligt gör man det med hjälp av Euklides algoritm. Man finner $l = 43$ som en möjlighet. Dekrypteringen innebär således att vi skall beräkna $81^{43} \mod 143$. Ytterligare en stunds räknande ger mycket riktigt att $81^{43} \equiv 42 \mod 143$.

Låt oss så visa varför metoden fungerar. Det krypterade meddelandet är alltså $a^k \mod pq$ och dekrypteringen innebär att vi skall beräkna $(a^k)^l \mod pq$. Men $kl = 1 + cM$ för något heltal c eftersom $kl \equiv 1 \mod M$, vilket ger

$$(a^k)^l \mod pq \equiv a^{kl} \equiv a^{1+cM} \equiv a \cdot (a^M)^c \equiv a \cdot 1^c \equiv a \mod pq$$

enligt Sats 3. Nu är både första och sista ledet $\leq pq$ och då måste de vara lika. Men hur gör man om meddelandet a från början är $> pq$? Jo, man delar upp det i mindre bitar.

Säkerheten i RSA-metoden har att göra med svårigheten att *faktorisera stora tal*. Det som är offentligt känt i krypteringsalgoritmen är talen pq och k. För att dekryptera behöver man talet l. För att bestämma det behöver man lösa kongruensen $k \cdot l \equiv 1 \mod M$ och då behöver man förstås veta vad $M = MGM(p - 1, q - 1)$ är. Den enda kända metoden att bestämma M är att ta reda på p och q, dvs att faktorisera talet pq. Att faktorisera stora tal är ett av de svåraste problem man känner till och det är därför man betraktar RSA som en mycket säker metod. I praktiken måste naturligtvis p och q vara stora primtal, säg med minst 100 siffror. Det är klart att det kan finnas snabba och effektiva faktoriseringsmetoder, eller att det finns en metod att bestämma M utan att faktorisera, men sådana känner man inte till idag.

För att RSA skall fungera måste man alltså ha tillgång till stora primtal. Sådana hittar man i praktiken genom ett slags trial-and-error. Man tar ett slumpvis utvalt tal med många siffror och testar om det är ett primtal. Primtalstest är ett enklare problem än faktorisering och ett sätt man ibland

använder bygger på Fermats lilla sats eller snarare på en "omvändning" till den: Låt a vara ett slumpvis valt tal. Om $a^{p-1} \equiv 1 \mod p$ så är det stor sannolikhet för att p är ett primtal (men det är inte alls säkert!).

Det pågår mycket forskning kring faktorisering av stora tal och primtalstestning, både för att knäcka RSA och för att göra den säkrare. Den naivaste metoden att faktorisera ett tal N är väl att prova att dividera med alla tal $\leq \sqrt{N}$ (om N är sammansatt, så har det minst en faktor som är $\leq \sqrt{N}$; varför?). Säg att N har 200 siffror, så att $\sqrt{N}$ har ungefär 100 siffror. Det finns då 10^{100} tal att dividera med. Hur snabbt kan en dator göra en räkneoperation? Låt oss säga att den minsta möjliga tiden är den det tar för en ljusstråle att färdas sträckan 10^{-15} meter (räckvidden hos kärnkrafterna). Detta blir $10^{-15}/3 \cdot 10^9 \approx 10^{-25}$ sekunder. Universum är ca 10^{20} s gammalt och under denna tid hinner man alltså göra ungefär 10^{45} operationer, om man inte gör dem parallellt. Om alla väteatomer, ca 10^{53} stycken, i solen vore involverade i räknandet, så skulle vi ha hunnit göra 10^{98} operationer sedan universum kom till och vi måste vänta tills det blir 100 gånger så gammalt som nu för att vi skall bli klara ...

3 Bevis för Fermats lilla sats

Vi skall avslutningsvis ge inte mindre än två olika bevis för Fermats lilla sats, ett enkelt och lättsmält med induktion och ett lite mer krävande, men som bättre visar varför satsen är sann.

Bevis 1. Vi kommer att behöva binomialsatsen

$$(x+y)^n = x^n + \binom{n}{1}x^{n-1}y + \ldots + \binom{n}{k}x^{n-k}y^k + \ldots + y^n$$

då exponenten är vårt primtal p. Vi behöver också veta att binomialkoefficienterna $\binom{p}{k}$ är delbara med p om $1 \leq k \leq p-1$. Varför är det så? Jo, vi har ju

$$\binom{p}{k} \cdot (p-k)!k! = p!.$$

Här är högerledet delbart med p, så då måste vänsterledet vara det också. Alltså är någon av faktorerna till vänster delbar med p, eftersom p är ett primtal. Men om $k! = 1 \cdot 2 \cdot \ldots \cdot k$ vore delbart med p, så vore någon av faktorerna till höger delbar med p, vilket är omöjligt då de alla är $< p$. På samma sätt ser man att $(p-k)!$ inte kan vara delbart med p och det följer att $\binom{p}{k}$ är delbart med p.

Vi skall nu använda induktion över a för att visa att $p|(a^p - a)$. När $a = 1$ så är $a^p - a = 0$ och påståendet är trivialt. Antag att det är sant för något visst a. Vi skall bevisa att det då är sant även för $a + 1$. Enligt binomialsatsen är

$$(a+1)^p = a^p + \binom{p}{1}a^{p-1} + \binom{p}{2}a^{p-2} + \ldots + \binom{p}{p-1}a + 1,$$

så att

$$(a+1)^p - (a+1) = (a^p - a) + \binom{p}{1}a^{p-1} + \binom{p}{2}a^{p-2} + \ldots + \binom{p}{p-1}a.$$

Enligt induktionsantagandet är $a^p - a$ delbart med p och enligt ovan är $\binom{p}{k}$ delbart med p om $1 \leq k \leq p-1$. Alltså är hela högerledet delbart med p och satsen följer med induktion.

Bevis 2. Låt a vara ett tal som inte är delbart med p. Vi skall betrakta de $p-1$ talen

$$a \cdot 1, a \cdot 2, \ldots, a \cdot (p-1).$$

Låt deras rester vid division med p vara $b_1, b_2, \ldots, b_{p-1}$. Notera att $1 \leq b_i \leq p-1$ och att $ai \equiv b_i \mod p$. Vi skall först visa att alla b_i:na är olika. För antag att $b_i = b_j$. Då är $ai \equiv aj \mod p$ och alltså har vi $p|a(i-j)$. Men p delar inte a, så $p|i-j$, varav följer $i = j$ eftersom både i och j ligger mellan 1 och $p-1$.

Då talen b_i är $p-1$ stycken till antalet och är olika, så måste de utgöra alla tal $1, 2, \ldots, p-1$ fast förmodligen i en annan ordning. Hur som helst är

$$b_1 \cdot b_2 \cdot \ldots \cdot b_{p-1} = 1 \cdot 2 \cdot \ldots \cdot (p-1) = (p-1)!.$$

Produkten av alla ai är $a^{p-1}(p-1)!$, så av $ai \equiv b_i \mod p$ följer

$$a^{p-1}(p-1)! \equiv b_1 \cdot \ldots \cdot b_{p-1} \equiv (p-1)! \mod p.$$

Alltså gäller att $p|(a^{p-1}-1)(p-1)!$. Men p kan inte dela $(p-1)!$, för då skulle det dela någon av faktorerna, vilka alla är $< p$. Det följer att $p|a^{p-1} - 1$ och med andra ord att $a^{p-1} \equiv 1 \mod p$.

Det andra beviset är möjligen lite mer abstrakt än det första, men det har fördelen att ge information om varför Fermats lilla sats är sann och kan dessutom generaliseras.