Matematik, KTH
Bengt Ek
augusti 2016

Material till kurserna
SF1630 och SF1679, Diskret matematik:

RSA-kryptering och primalitetstest

Hemliga koder (dvs koder som används för att göra meddelanden oläsbara för obehöriga) var länge främst av intresse för diplomater, spioner, militärer och kriminella, men numera har de blivit mycket viktiga i samband med datakommunikation.

Med ett **kryptosystem**

$$\mathcal{M} \underset{D}{\overset{E}{\rightleftarrows}} \mathcal{C}$$

avser vi två ändliga mängder $\mathcal{M}$ (de möjliga meddelandena, klartexterna) och $\mathcal{C}$ (motsvarande möjliga chiffertexter) och två funktioner $E : \mathcal{M} \to \mathcal{C}$ för kryptering och $D : \mathcal{C} \to \mathcal{M}$ för avkryptering, sådana att

$$D(E(m)) = m \text{ för alla } m \in \mathcal{M}.$$

Vi antar att $\mathcal{M} = \mathcal{C} = \mathbb{Z}_n = \{0, 1, 2, \ldots, n-1\}$ för något (stort) $n \in \mathbb{N}$. Om de meddelanden vi vill kryptera har någon annan form (t.ex. vanlig text) kan de omformas till stora tal, t.ex. med hjälp av ASCII-koderna för ingående symboler (och långa meddelanden kan brytas upp i mindre bitar för att rymmas i $\mathbb{Z}_n$).

I ett klassiskt kryptosystem (som substitutionschiffer ("byt varje 'a' mot ett 'k', varje 'b' mot ett 's', osv") och många andra, mer sofistikerade krypton) innebär kunskap om E att man känner D. De måste alltså båda hållas hemliga. ("I princip" gäller det för varje kryptosystem, ty om man känner E kan man pröva alla $m \in \mathcal{M}$ tills $E(m)$ är den chiffertext man vill tolka, men i praktiken är det inte möjligt då $\mathcal{M}$ är mycket stort).

1976 föreslogs en annan typ av kryptosystem, nu kända som **kryptosystem med offentlig nyckel** (eng. **public-key cryptosystems**) av W. Diffie och M. Hellman. I ett sådant system har varje användare, A, sin egen krypteringsfunktion E_A och sin egen avkrypteringsfunktion D_A. De är så "komplicerade" att även om man känner E_A är det mycket svårt att finna D_A. Alla E_A:n kan offentliggöras och varje deltagare håller sitt eget D_A hemligt. Det betyder att vem som helst kan producera en chiffertext till A, men bara någon som

har tillgång till D_A (dvs förhoppningsvis bara A själv) kan läsa den. Ett av de första sådana systemen, och troligen det bäst kända, är RSA-systemet (uppkallat efter upphovsmännen Ron Rivest, Adi Shamir och Leonard Adleman). Eftersom det bygger på primtal och deras matematik, förtjänar det en plats i vår kurs.

RSA-systemet

Låt p och q vara olika (stora) primtal och $n = p \cdot q$, $m = \phi(n)$ (ϕ är Eulers funktion). Eftersom $\mathbb{Z}_n \approx \mathbb{Z}_p \times \mathbb{Z}_q$ (enligt Kinesiska restsatsen) och $(a, b) \in \mathbb{Z}_p \times \mathbb{Z}_q$ är inverterbart omm $a, b \neq 0$, finner vi att antalet inverterbara element i $\mathbb{Z}_n$ är $m = \phi(n) = (p-1)(q-1)$.

Låt nu $x \in \mathbb{Z}_n$. Enligt Eulers sats gäller $x^m = 1$ (i $\mathbb{Z}_n$) om(m) $\text{sgd}(x, n) = 1$, dvs omm $p, q \nmid x$. Antag att $x \in \mathbb{Z}_n$ motsvarar $(a, b) \in \mathbb{Z}_p \times \mathbb{Z}_q$ (dvs $f(x) = (a, b)$ med beteckningen i materialet om kinesiska restsatsen). Då gäller för varje $k \in \mathbb{N}$ att $f(x^{km+1}) = (a^{km+1}, b^{km+1})$. Om $a \neq 0$ är $a^{km+1} = (a^{p-1})^{k(q-1)} \cdot a = 1^{k(q-1)} \cdot a = a$ (i $\mathbb{Z}_p$) medan $0^{km+1} = 0$, så $a^{km+1} = a$ för alla $a \in \mathbb{Z}_p$. På samma sätt är $b^{km+1} = b$ (i $\mathbb{Z}_q$), så $f(x^{km+1}) = (a, b) = f(x)$ för alla $x \in \mathbb{Z}_n$. Eftersom f är injektiv ger det följande

Sats:
Låt p och q vara skilda primtal, $n = p \cdot q$, och $m = (p-1)(q-1)$.
Om $x \in \mathbb{Z}_n$ och $s \in \mathbb{N}$ uppfyller $s \equiv_m 1$, så

$$\boldsymbol{x^s = x} \text{ i } \mathbb{Z}_{\boldsymbol{n}},$$
$$\text{dvs för alla } \boldsymbol{x \in \mathbb{Z}}, \quad \boldsymbol{x^s \equiv x \pmod{n}}.$$

Det ger följande sätt **att konstruera ett RSA-system:**

1. Tag två olika primtal p, q.
2. Låt $\boldsymbol{n = p \cdot q}$ och $\boldsymbol{m = (p-1)(q-1)}$.
3. Finn $e \in \mathbb{N}$ med $\text{sgd}(e, m) = 1$, och $d \in \mathbb{N}$ med $\boldsymbol{e \cdot d \equiv 1 \pmod{m}}$.
4. Låt $E, D : \mathbb{Z}_n \to \mathbb{Z}_n$ ges av $\boldsymbol{E(x) = x^e}$ och $\boldsymbol{D(x) = x^d}$.
   så $E(x) \equiv x^e \pmod{n}$, $D(x) \equiv x^d \pmod{n}$
5. Offentliggör n och e, håll d hemligt (och kasta bort m (så ingen får tag på det)).

Observera att modulerna är olika. Vi beräknar E, D med modul n och finner d med modul m (innan vi kastade bort det).

För att finna e i steg 3, pröva olika värden och kontrollera med Euklides algoritm villkoret $\text{sgd}(e, m) = 1$. Den ger då också motsvarande d.

Eftersom $D(E(x)) = (x^e)^d = x^{ed} = x^{de} = (x^d)^e = E(D(x)) = x$ enligt satsen ovan är $D = E^{-1}$ i det här systemet.

D är svår att finna (dvs d är svårt att finna om man bara känner n och e), ty det är (såvitt vi vet) svårt att faktorisera stora heltal (för att finna m) (på rimlig tid). I praktiken verkar kring 150 siffror (i bas 10) i p och q anses tillräckligt.

Exempel. Om vi tar (de inte så stora) primtalen $p = 17$, $q = 23$, finner vi $n = 17 \cdot 23 = 391$ och $m = (17-1)(23-1) = 352$.
Med (t.ex.) $e = 15$ ($\text{sgd}(352, 15) = 1$, ty $352 = 15 \cdot 23 + 7$, $15 = 7 \cdot 2 + 1$), får vi $d = 47$ ($1 = 15 - 2 \cdot 7 = 15 - 2(352 - 23 \cdot 15) = -2 \cdot 352 + 47 \cdot 15$, så $15 \cdot 47 \equiv_{352} 1$).
För att kryptera det ytterst hemliga meddelandet $x = 147$, beräknar vi $E(147) = 147^{15} = 167$ (i $\mathbb{Z}_{391}$). Så chiffertexten är 167. Om vi avkrypterar med D får vi $D(167) = 167^{47} = 147$ (också i $\mathbb{Z}_{391}$).

I exemplet innehöll beräkningarna 147^{15}, som har 33 siffror i bas 10. I ett verkligt system skulle man ha x med hundratals siffror och e likaså, så x^e skulle vara alldeles för stort för att beräkna, ens på en riktigt stor dator.
I stället kan man beräkna successiva kvadrater, x, x^2, x^4, $x^8, \ldots$, och i varje steg ta resultatet (mod n) (dvs räkna i $\mathbb{Z}_n$). Då räcker det att handskas med tal som inte är större än n^2, vilket inte är svårt för en dator. Sedan multipliceras rätt x^i:n ihop (fortfarande (mod n) i varje steg) för att få resultatet.

Exemplet fortsatt. Vi kan beräkna $D(167) = 167^{47}$ så här i $\mathbb{Z}_{391}$:
$x = 167$,
$x^2 = 27\,889 = 128$,
$x^4 = 128^2 = 16\,384 = 353$,
$x^8 = 353^2 = 124\,609 = 271$,
$x^{16} = 271^2 = 73\,441 = 324$,
$x^{32} = 324^2 = 104\,976 = 188$.
Eftersom $47 = (101111)_2$ får vi $x^{47} = x^{32} \cdot x^8 \cdot x^4 \cdot x^2 \cdot x = 188 \cdot 271 \cdot 353 \cdot 128 \cdot 167 =$
$= 50\,948 \cdot 353 \cdot 128 \cdot 167 = 118 \cdot 353 \cdot 128 \cdot 167 = 41\,654 \cdot 128 \cdot 167 = 208 \cdot 128 \cdot 167 =$
$= 26\,624 \cdot 167 = 36 \cdot 167 = 6\,012 = 147$ (i $\mathbb{Z}_{391}$).

Elektronisk signatur

Att ”vem som helst” kan kryptera meddelanden till användaren A, med den offentliga nyckeln E_A, ger ett nytt problem. Nämligen, då B skickar ett meddelande till A, hur kan A vara säker på att det verkligen kommer från B och inte från en bedragare?

Det problemet löses med en så kallad **elektronisk signatur**, som använder att $D(E(x)) = E(D(x)) = x$.

Antag att B vill försäkra sig om att A kommer att vara säker på att det var B som sände meddelandet x. Hon kan då sända $D_B(E_A(x))$ (eller $E_A(D_B(x))$) i stället för $E_A(x)$. Eftersom A (liksom alla inblandade) har E_B, kan han använda den för att få $E_B(D_B(E_A(x))) = E_A(x)$ och sedan med sin egen hemliga D_A få x. Bara B kan använda D_B, så bara B kunde vara avsändaren och bara A kunde läsa meddelandet, eftersom D_A behövdes.

På liknande sätt kan A, om han vill göra meddelandet y allmänt känt, sända $D_A(y)$ till alla. De kan läsa meddelandet med den offentliga E_A och bara någon med den hemliga D_A kan vara avsändaren.

Primalitetstest

För att implementera ett RSA-system, måste varje användare ges två egna mycket stora primtal.

Det finns tillräckligt många primtal. Enligt den berömda primtalssatsen, bevisad 1896, är tätheten av primtal nära N omkring $\frac{1}{\ln N}$ för stora N, så nära $N = 10^{150}$ är ungefär ett tal av 350 ett primtal. Det betyder att en dator kan finna primtal ganska lätt genom att testa flera kandidater, förutsatt att den har en någotsånär effektiv metod för att avgöra om ett heltal med omkring 150 siffror (i bas 10) är ett primtal. Så vi behöver ett bra **primalitetstest**.

Hur testar vi om ett litet heltal, till exempel 389, är ett primtal? Den mest direkta metoden skulle vara att söka faktorer, genom att pröva alla primtal upp till 19 (eftersom nästa primtal, 23, har $23^2 > 389$) (prövningen behöver inte vara en fullständig division, t.ex. gäller $19 \mid 389 \Leftrightarrow 19 \mid (389 - 19) = 370 \Leftrightarrow 19 \mid 37$, (eftersom $\text{sgd}(19, 10) = 1$), så $19 \nmid 389$). 389 är ett primtal, men den metoden kan inte användas för riktigt stora tal (åtminstone inte med datorer av nuvarande typ). Även om vi redan känner alla primtal upp till 10^{75} och kan avgöra delbarhet på 10^{-12} sekunder, kunde testet av ett tal nära 10^{150} ta mer än 10^{53} år. Att testa för alla tänkbara faktorer skulle nästan innebära att faktorisera ett stort heltal och om vi kunde det vore RSA hur som helst inte säkert.

Men vi känner till ett test för att avgöra om N är ett primtal, utan att faktorisera:

Fermattestet, bas b, $1 < b < N$:

$$\textbf{Är } b^{N-1} \equiv 1 \pmod{N}\textbf{?}$$

Enligt Fermats (lilla) sats är svaret "ja" för alla b om N är ett primtal, så om svaret är "nej" vet vi säkert att N inte är ett primtal. **Men** det finns heltal, så kallade (fermat-) **pseudoprimtal** för bas b, som klarar fermattestet för bas b utan att vara primtal.

Exempel. Eftersom $2^{10} = 1024 = 341 \cdot 3 + 1$ är $2^{340} \equiv_{341} 1^{34} = 1$, så 341 är ett fermatpseudoprimtal för bas 2. Men $3^{340} \equiv_{341} 56 \neq 1$, så 341 är inte ett fermatpseudoprimtal för bas 3 (och inget primtal, $341 = 11 \cdot 31$).

Som exemplet visar ger fermattestet att 341 inte är ett primtal, om vi prövar det med både bas 2 och bas 3. Ett **probabilistiskt** primalitetstest av N kunde då vara att ta ett antal slumpmässigt valda baser $b_1, b_2, \ldots, b_k$ och utföra testet för dem alla. Om N är ett primtal skulle det klara alla testen, men det vore osannolikt att ett icke-primt tal skulle göra det (i meningen att bara en mycket liten andel av alla icke-prima N av liknande storlek, säg, skulle klara alla). Om vi ökar antalet baser, k, bör sannolikheten att ett icke-primtal slinker igenom bli "praktiskt taget noll". Eller?

I själva verket finns det icke-primtal N som klarar **alla** fermattest med baser b som uppfyller $\text{sgd}(b, N) = 1$. Sådana tal kallas **carmichaeltal** och de är precis de kvadratfria (dvs för inget primtal p gäller $p^2 \mid N$) sammansatta heltal som uppfyller att för alla primtal p med $p \mid N$ gäller $(p - 1) \mid (N - 1)$. De har alla minst tre primfaktorer och det finns oändligt många sådana.

De minsta carmichaeltalen är

$$561 = 3 \cdot 11 \cdot 17,\ 1105 = 5 \cdot 13 \cdot 17,\ 1729 = 7 \cdot 13 \cdot 19,\ 2465 = 5 \cdot 17 \cdot 29,$$

$$2821 = 7 \cdot 13 \cdot 31,\ 6601 = 7 \cdot 23 \cdot 41,\ 8911 = 7 \cdot 19 \cdot 67.$$

Om primfaktorerna i ett carmichaeltal är mycket stora, är det mycket osannolikt att vi skulle upptäcka att det inte är ett primtal med det probabilistiska testet som beskrevs ovan.

Miller-Rabins test med bas b (från 1980) är en modifikation av fermattestet som, när det utförs med flera olika baser, gör det mycket osannolikt att ett icke-primtal skulle bedömas vara primt.

Om N är ett primtal och $x \in \mathbb{Z}_N$ uppfyller $x^2 = 1$, så är $x = \pm 1$ (om N är primt gäller $N \mid (x^2 - 1) \Rightarrow N \mid (x - 1)$ eller $N \mid (x + 1)$). Det är observationen bakom

Miller-Rabins test, bas b, $1 < b < N$:

$$\textbf{Låt } N - 1 = u \cdot 2^r, \quad u \textbf{ udda, } r \geq 1 \textbf{ (om } N \textbf{ är udda).}$$

$$\textbf{Är } b^u \equiv_N 1 \textbf{ eller } (b^u)^{2^i} \equiv_N -1 \textbf{ för något } i,\ 0 \leq i < r\textbf{?}$$

Om N är primt blir svaret ”ja” och man kan visa att om N är sammansatt klarar det Miller-Rabins test för högst $\frac{N}{4}$ av baserna b med $1 < b < N$.

Exempel. För $N = 561$ och $b = 2$ finner vi $N - 1 = 560 = 35 \cdot 2^4$ och
$2^{35} \equiv_{561} 263$,
$2^{70} \equiv_{561} 263^2 = 69\,169 \equiv_{561} 166$,
$2^{140} \equiv_{561} 166^2 = 27\,556 \equiv_{561} 67$,
$2^{280} \equiv_{561} 67^2 = 4\,489 \equiv_{561} 1$.
Eftersom $67 \not\equiv_{561} -1$, visar Miller-Rabins test att 561 inte är ett primtal.
Med isomorfin (från kinesiska restsatsen)

$$f : \mathbb{Z}_{561} \to \mathbb{Z}_3 \times \mathbb{Z}_{11} \times \mathbb{Z}_{17},$$

kan vi se vad som händer här:
$f(2^{35}) = f(263) = (2, 10, 8) = (-1, -1, 8)$,
$f(2^{70}) = f(166) = (1, 1, 13)$,
$f(2^{140}) = f(67) = (1, 1, -1)$,
$f(2^{280}) = f(1) = (1, 1, 1)$,
$f(2^{560}) = f(1^2) = (1, 1, 1)$.
Att $2^{560} \equiv_n 1$ for $n = 3, 11, 17$, innebär att 561 klarar fermattestet med bas 2, men eftersom $2^{35 \cdot 2^i} \equiv_n -1$ inte infaller för samma i för alla $n = 3, 11, 13$ är $2^{35 \cdot 2^i} \not\equiv_{561} -1$ för alla i, så 561 klarar inte Miller-Rabins test med bas 2.

Övningsexempel

1. En användare i ett RSA-system har de offentliga parametrarna $(n, e) = (143, 17)$.

a. Kryptera meddelandet $x = 71$.

b. Finn en möjlig hemlig avkrypteringsparameter d för denna användare.

c. Kontrollera ditt d genom att avkryptera resultatet i a. med det.

2. Ett RSA-system använder $n = 1147 (= 31 \cdot 37)$ och $e \geq 332$.
Vilket är det minsta möjliga värdet för e? Finn också ett motsvarande d.

3. En RSA-användare har parametern $n = p \cdot q = 57\,656\,617$.
Vilka är primtalen p och q, om $m = (p-1)(q-1) = 57\,641\,220$?

4. Visa att om p, q är olika primtal så är $p^{q-1} + q^{p-1} \equiv 1 \pmod{pq}$.

5. Förklara varför det i alla relevanta fall räcker att välja d så att $e \cdot d \equiv_{\frac{m}{2}} 1$ för att det skall kunna användas som avkrypteringsparameter.

Svar

1a. $E(x) = 80$, **b.** $d = 113$ (53 går också bra).

2. $e = 337$, $d = 673$ (eller t.ex. 133).

3. $\{p, q\} = \{6427, 8971\}$.

5. I satsen i texten räcker det att $\text{mgm}(p-1, q-1) \mid s$, så om både p och q är udda (dvs ingen av dem är 2), fungerar $\frac{m}{2}$.