Kungl Tekniska Högskolan

Institutionen för Numerisk analys och datalogi

Kryptering och primtalsfaktorisering

Johan Håstad
Nada, KTH
johanh@nada.kth.se

Ett Exempel

$$
\begin{aligned}
N = \ & 93248941901237910481523323193941354 \\
& 41141253923482543847923483201340943 \\
& 01913415116613951851034125615302323 \\
& 24525239230624210960123234120156 \\
& 809104109501303498614012865123
\end{aligned}
$$

Kan vi avgöra om N är sammansatt?

Om så kan vi faktorisera N?

Spelar det någon roll?

Fermatprimtal

$F_i = 2^{2^i} + 1.$

$$
\begin{aligned}
F_0 &= 3 \\
F_1 &= 5 \\
F_2 &= 17 \\
F_3 &= 257 \\
F_4 &= 65537 \\
F_5 &= 4294967297
\end{aligned}
$$

Pierre de Fermat noterade att det första 5 är primtal och påstods ha trott att alla F_i är primtal.

Källkritik?!

Vad gjorde Fermat?

"Enda bevis publicerat av Fermat var för satsen att varje primtal på formen $4k+1$ kan på ett unikt sätt skrivas som summan av två kvadrater."

$$\begin{aligned} 29 &= 5^2 + 2^2 \\ 41 &= 6^2 + 5^2 \\ 53 &= 7^2 + 2^2 \\ 113 &= 8^2 + 7^2 \end{aligned}$$

Fermat hade (helt) fel

$F_5 = 4294967297 = 641 \cdot 6700417$ och inga andra kända tal i serien är primtal.

Borde Fermat insett att F_5 inte är primtal?

Fermats lilla sats

Sats: Om $1 \leq a \leq p-1$ så är resten av a^{p-1} vid division med p lika med 1.

$\equiv_p$ betyder rest vid divison med p

$$
\begin{aligned}
1^6 &= 1 \equiv_7 1 \\
2^6 &= 64 \equiv_7 1 \\
3^6 &= 729 \equiv_7 1 \\
4^6 &= 4096 \equiv_7 1 \\
5^6 &= 15625 \equiv_7 1 \\
6^6 &= 46656 \equiv_7 1
\end{aligned}
$$

men

$2^8 = 256 \equiv_9 4$ och 9 är således inte primtal.

$$N = 257$$

$$\begin{aligned} 3^{256} \;=\;& 139008452377144732764939786789661 \\ & 303114218850808529137991604824430 \\ & 036072629766435941001769154109609 \\ & 521811665540548899435521 \end{aligned}$$

och ger rest 1 vid division med 256. Verkar som 257 är primtal...

Åter till F_5

$3^{4294967296}$ ger rest 3029026160 vid division med 4294967297 och vi kan med säkerhet säga att F_5 inte är primtal.

Hur inser man detta?

$3^{4294967296}$ är ett tal med ett par miljarder siffror...

Bättre räkningar

Vad är sista siffran i 1541^{2312}?

Vad är sista siffran i 1543^{2312}?

Generalisering

Sista siffran är resten vid division med 10.

För att beräkna den behöver vi bara sista siffran i operanderna och kan slänga alla andra siffor.

Allmänt: Vill vi veta resten av svaret av en beräkning vid division med N, kan vi kasta alla multipler av N på vägen.

Bättre räkningar för $N = 257$

$$
\begin{aligned}
3^{256} &= 9^{128} \\
&= 81^{64} \\
&= 6561^{32} \\
&\equiv_{257} 136^{32} \\
&= 18496^{16} \\
&\equiv_{257} 249^{16} \\
&= 62001^{8} \\
&\equiv_{257} 64^{8} \\
&= 4096^{4} \\
&\equiv_{257} 241^{4} \\
&= 58081^{2} \\
&\equiv_{257} 256^{2} \\
&= 55536 \equiv_{257} 1
\end{aligned}
$$

Enbart 8 multiplikationer av tresiffriga tal.

Bättre beräkning F_5

Fermat kunde ha räknat ut resten av $3^{4294967296}$ vid division med 4294967297 med 32 multiplikationer av 10 siffriga tal och 32 resttagningar.

Tar högst någon timme för en van handräknare.

Skulle dock inte ha gett faktoriseringen som kräver MYCKET jobb att få fram för hand.

Datorkörningar

En dator klarar att göra ett “Fermattest” på 1000 siffriga tal på högst någon minut, en bra implementation på någon sekund.

Slutsats antingen *“Definitivt inte primtal”* eller *“Kanske primtal”*.

Bättre upp

Gary Miller och Michael Rabin visade 1976 att om vi utvidgar testet en aning och provar 50 slumpvis a så fås

"Definitivt inte primtal" eller *"Troligen primtal"*.

Sannolikhet att ha fel är 2^{-100}.

Behövs slump?

Felsannolikhet 2^{-100} gör detta till en akademisk fråga av litet praktiskt intresse.

Stort filosofiskt och teoretiskt intresse. Finns det ett effektivt deterministiskt primtalstest?

Manindra Agarwal, 2002

Det finns ett deterministiskt primtalstest som på tal med n siffor gör $\approx n^6$ operationer.

Mycket bättre än provdivision som går i tid $\sqrt{N} \approx 10^{n/2}$, men mycket sämre än Miller-Rabin som går i tid $\approx n^3$.

Komplexitetsteoretisk slutsats

Primtal ligger i klassen av effektivt beräkningsbara funktioner, kallad P.

Detta är inte känt för faktorisering.

Allmänt så tror man att faktorisering inte ligger i P men ingen är i närheten av att visa detta.

Relation till kryptering

Vi har använt satser om rester vid division. Betrakta följande:

a^{21} och a ger samma rest vid divison med 11.

Sätt b till resten av a^7 vid divison med 11.

Då $21 = 7 \cdot 3$ ger resten av b^3 vid division med 11 tillbaka a.

$a = 2$ ger $a^7 = 128 \equiv_{11} 7 = b$ och $b^3 = 343 \equiv_{11} 2$.

Detta är en krypteringsmetod!

Kryptering

Välj stort tal N och e och d så att a^{ed} och a alltid ger samma rest vid division med N.

Ta ett medelande och koda som ett stort heltal M.

Kryptotexten blir resten av M^e vid division med N, kalla den C.

Klartexten återskapas som resten av C^d med division med N.

Exempel

Meddelande HEJ JOHAN. A=1, B=2 etc ger

$$M = 805101015080114$$

Vi kan ha $N = 23942194232123139$ och lämpliga e och d.

Längre meddelanden delas upp i bitar, ett blockkrypto.

Egenskaper

Dekrypteringen blir rätt på grund av matematisk sats.

Kryptering och dekryptering är tämligen effektiva, ungefär som ett Miller-Rabin test.

Att skapa nycklar N, e och d är lätt, följer av matematiska satser.

Kan vi publicera N och e?

Kryptosystemet RSA

Välj N som produkten av två stora (kanske 150 decimala siffror) primtal.

Beräkna lämpliga e och d.

Publicera N och e.

Vi publicerar KRYPTERINGSNYCKELN, alla kan kryptera.

RSA egenskaper

Lätt att skapa nycklar via primtalstest.

Enda kända sättet att forcera är att faktorisera N, vilket i dag går trögt, bästa klarar runt 160 siffror.

Lätt att se att N inte är primtal, svårt att faktorisera.

Kan eventuellt använda $e = 3$!

Slutord

Inte svårt att skriva egen implementation av RSA, kräver bara aritmetik på stora heltal.

Rimligt projekt att förstå talteorin bakom att RSA fungerar.