4. poglavje

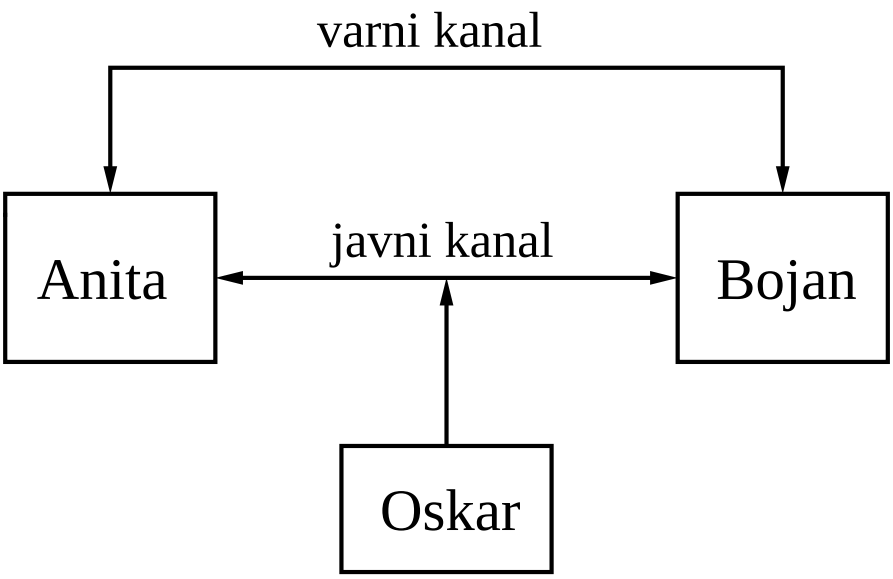
RSA sistem in faktorizacija

- Uvod
  - pomankljivosti simetrične kriptografije
  - kriptografija z javnimi ključi
- Teorija števil
- Opis in implementacija RSA
- Gostota praštevil
- Generiranje praštevil
- Gaussov izrek (o kvadratni recipročnosti)

Uvod

Pomankljivosti simetrične kriptografije

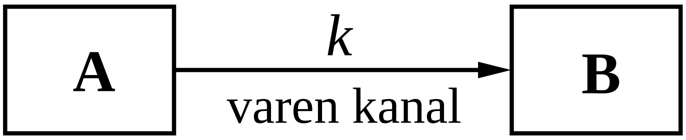
Sodelujoči si delijo *tajno* informacijo.



Dogovor o ključu

Kako Anita in Bojan vzpostavita tajni ključ k?

1. metoda: delitev point-to-point



Varni kanal je lahko:
– kurir
– izmenjava na štiri oči (v temnem hodniku/ulici)

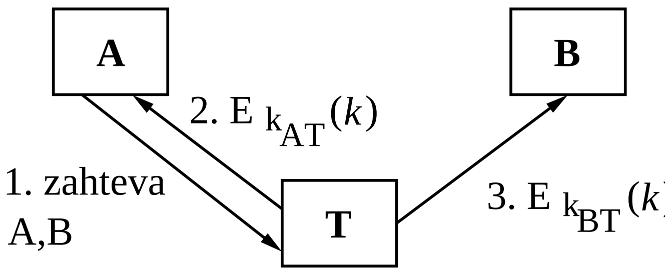
To ni praktično za večje aplikacije.

2. metoda: z neodvisnim centrom zaupanja T

– *Vsak* uporabnik A deli tajni ključ k_{AT} s centrom zaupanja T za simetrično šifrirno shemo E.

– Za vzpostavitev tega ključa mora A obiskati center zaupanja T *samo enkrat*.

– T nastopa kot **center za distribucijo ključev**: (angl. key distribution centre - KDC):



1. A pošlje T zahtevek za ključ, ki si ga želi deliti z B.
2. T izbere ključ k, ga zašifrira za A s ključem k_{AT}.
3. T zašifrira ključ k za osebo B s ključem k_{BT}.

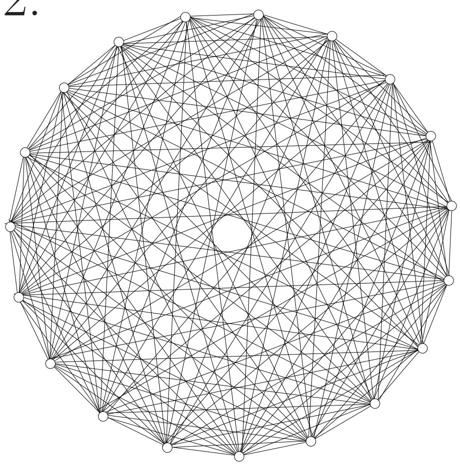
Problemi pri uporabi KDC

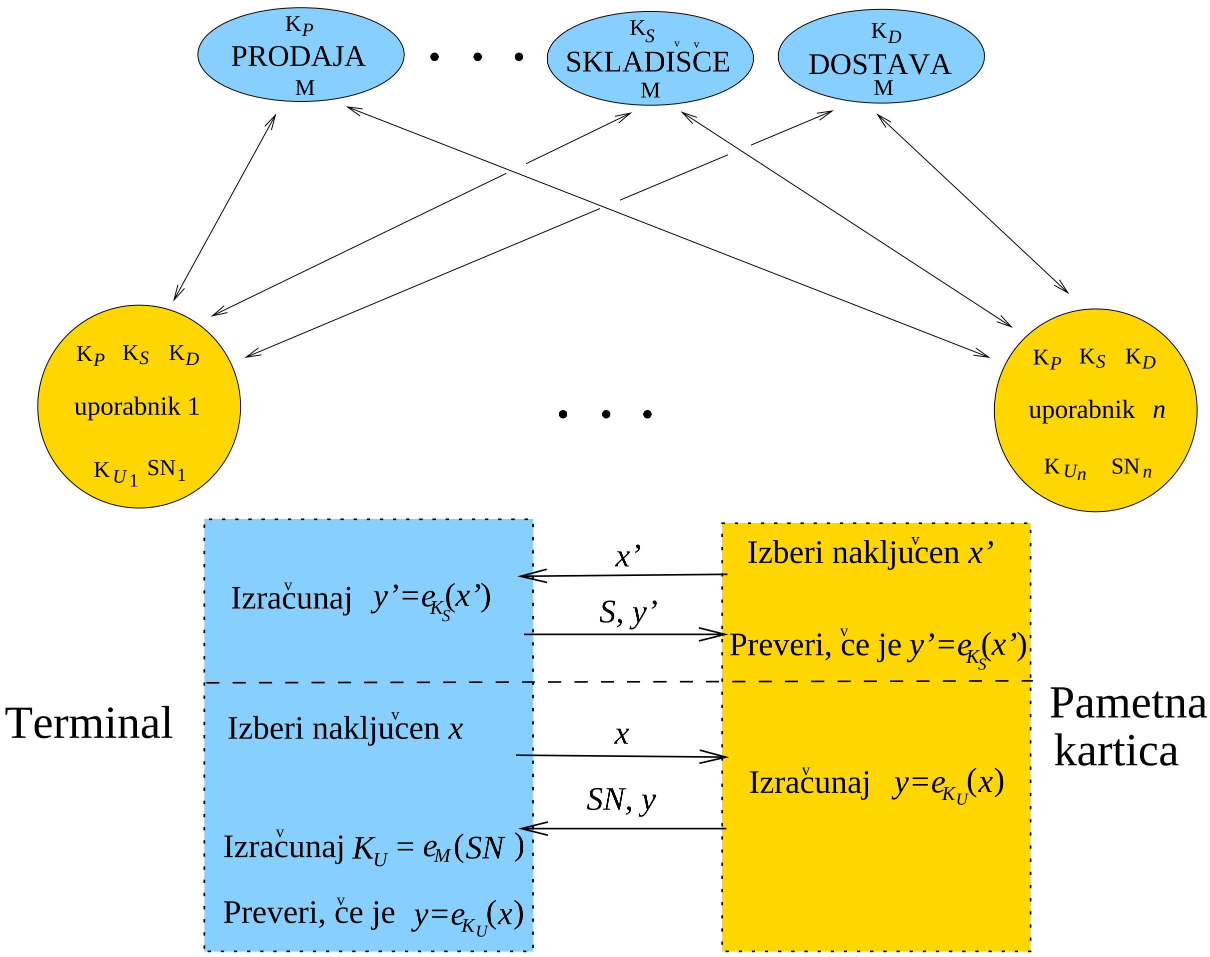
- centru zaupanja T moramo brezpogojno zaupati:
  - to ga naredi za očitno tarčo.
- Zahteva za stalno zvezo (on-line) s centrom T:
  - potencialno ozko grlo,
  - kritično za zanesljivost.

Upravljanje ključev

- v mreži z n uporabniki, mora vsak uporabnik deliti različen ključ z vsakim uporabnikom,
- zato mora hraniti vsak uporabnik $n-1$ različnih tajnih ključev,
- vseh tajnih ključev je $\binom{n}{2} \approx n^2/2$.

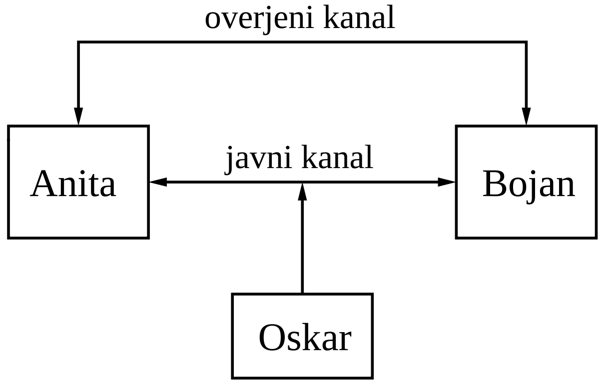
(Tudi preprečevanje tajenja je nepraktično.)




K_P
PRODAJA
M
K_S
SKLADIŠČE
M
K_D
DOSTAVA
M
K_P K_S K_D
uporabnik 1
K_U1 SN_1
K_P K_S K_D
uporabnik n
K_Un SN_n
Izračunaj y'=e_KS(x')
Izberi naključen x'
x'
S, y'
Preveri, če je y'=e_KS(x')
Terminal
Izberi naključen x
x
Izračunaj y=e_KU(x)
SN, y
Izračunaj K_U = e_M(SN)
Preveri, če je y=e_KU(x)
Pametna kartica

Kriptografija z javnimi ključi

Udeleženci si predhodno delijo *overjeno/avtentično* informacijo.



L. **1976** sta jo predlagala Whitfield **Diffie** in Martin **Hellman** (L. 1970 pa tudi James Ellis, ki je bil član Communication Electronics Security Group pri British Government Communications Headquarters).

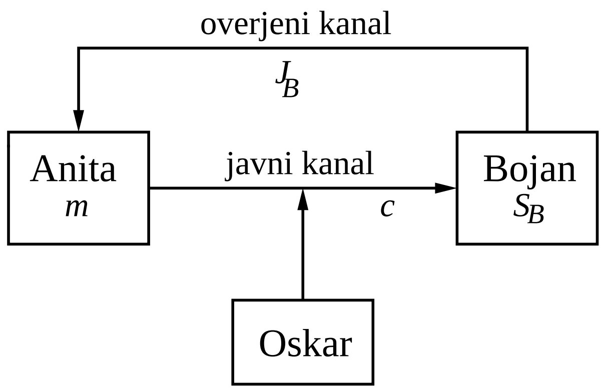
Generiranje para ključev

Vsaka oseba A naredi naslednje:

- generira par ključev (J_A, S_A),
- S_A je A-jev zasebni/tajni ključ,
- J_A je A-jev javni ključ.

Varnostna zahteva: za napadalca mora biti nemogoče priti do kluča S_A iz ključa J_A.

Šifriranje z javnimi ključi



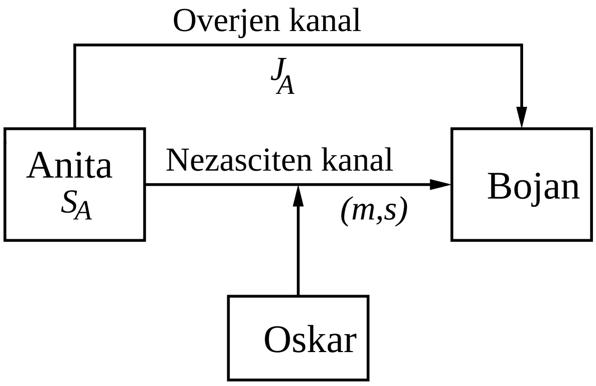
Da bi Bojanu poslala zaupno sporočil m, Anita:

- dobi overjenjo kopijo Bojanovega javnega kjuča J_B,
- izračuna $c = E(J_B, m)$, kjer je E šifrirna funkcija,
- pošlje Bojanu tajnopis c.

Za odšifriranje tajnopisa c Bojan naredi nasledne

- Izračuna $m = D(S_B, c)$, kjer je D odšifrirna funkcija.

Digitalni podpisi



Za podpis sporočila m Anita naredi naslednje:

- izračuna $s = \mathsf{Sign}(S_A, m)$,
- pošlje m in s Bojanu.

Bojan preveri Anitin podpis s sporočila m:

- pridobi si overjeno kopijo javnega ključa J_A,
- sprejme podpis, če je $\mathsf{Verify}(J_A, m, s) = \mathsf{Accept}$.

Prednosti kriptosistemov z javnimi ključi

- Ni zahteve po varnem kanalu.
- Vsak uporabnik ima 1 par ključev.
- Poenostavljeno upravljanje s ključi.
- Omogoča preprečevanje tajenja.

Pomanjkljivosti kriptosistemov z javnimi ključi

- Sheme z javnimi ključi so počasnejše.
- Javni ključi so večji od simetričnih.

V praksi uporabljamo skupaj sheme s simetričnimi in javnimi ključi in jim rečemo **hibridne sheme**

Primer: Da bi Bojanu poslala podpisano tajno sporočilo m, Anita naredi naslednje:

- izračuna $s = \mathsf{Sign}(S_A, m)$,
- izbere tajni ključ k simetrične šifrirne sheme (AES),
- pridobi overjeno kopijo Bojanovega javnega ključa J_B,
- pošlje $c_1 = E(J_B, k)$, $c_2 = \mathsf{AES}(k, (m, s))$.

Za odkritje sporočila m in preverjanje avtentičnosti, Bojan:

- odšifrira c_1: $k = D(S_B, c_1)$,
- odšifrira c_2 z uporabo ključa k, da dobi (m, s),
- pridobi overjeno kopijo javnega ključa J_A,
- preveri podpis s sporočila m.

Že l. 1977 so Ronald L. **Rivest**, Adi **Shamir** in Leonard M. **Adleman** naredili prvo realizacijo takšnega kriptosistema (**RSA**)
(tajno pa že l. 1973 **C. Cocks** pri GCHQ).

Temu so sledili številni drugi nesimetrični kriptosistemi, med katerimi pa so danes najbolj pomembni naslednji:

- RSA (faktorizacija),
- Merkle-Hellman Knapsack (metoda nahrbtnika)
- Chor-Rivest
- McEliece (linearne kode),
- ElGamal (diskretni logaritem),
- eliptične krivulje.

Javni kriptosistemi **niso** nikoli brezpogojno varni, zato študiramo računsko/časovno zahtevne sisteme.

Teorija števil

Evklidov algoritem in reševanje Diofantske enačbe

$$ax + by = d, \qquad \text{kjer} \;\; D(a,b) \mid d.$$

Evklidov algoritem je zasnovan na preprostem dejstvu, da iz $k \mid a$ in $k \mid b$ sledi $k \mid a - b$.

Če je $D(a,b) = 1$ in poznamo eno rešitev (x_0, y_0), tj.

$$ax_0 + by_0 = d,$$

potem ima poljubna rešitev (x, y) naslednjo obliko:

$$x = x_0 - kb, \quad y = y_0 + ka, \qquad \text{za } k \in \mathbb{Z}.$$

Zgodovina Evklidovega algoritma

Evklidov algoritem poišče največji skupni delitelj dveh naravnih števil in je zasnovan na dejstvu, da če število d deli števili a in b, potem deli tudi njuno razliko $a-b$.

V literaturi naletimo nanj prvič 300 p.n.š. v 7. knjigi Evklidovih **Elementov**.

Nakateri strokovnjaki so mnenja, da je njegov avtor Eudoxus (c. 375 p.n.š.). Gre za *najstarejši* netrivialen algoritem, ki je preživel do današnjih dni (glej Knuth).

Eno rešitev lahko poiščemo z **razširjenim Evklidovim algoritmom**.

Privzemimo, da je $a > b$ in zapišimo zgornjo enačbo malo bolj splošno (z zaporedji):

$$ap_i + bq_i = r_i.$$

Poiščimo dve trivialni rešitvi:

$$p_1 = 1, \quad q_1 = 0, \quad r_1 = a$$

in

$$p_2 = 0, \quad q_2 = 1, \quad r_2 = b.$$

Zaradi rekurzije

$$r_{i+1} = r_i - s_i r_{i-1}$$

(kjer je s_i izbran tako, da je $r_{i+1} < r_i$)
si lahko izberemo še

$$p_{i+1} = p_i - s_i p_{i-1} \quad \text{in} \quad q_{i+1} = q_i - s_i q_{i-1}.$$

Ko računamo a^{-1} (po modulu praštevila p), računamo samo r_i ter p_i (ne pa tudi q_i).

Zgled za razširjeni algoritem:

$4864 = 1{\cdot}3458{+}1406$	$p_2 := p_1 - 1 \cdot p_0 = 1$
$3458 = 2{\cdot}1406{+}646$	$p_3 := p_2 - 2 \cdot p_1 = -2$
$1406 = 2{\cdot}646{+}114$	$p_4 := p_3 - 2 \cdot p_2 = 5$
$646 = 5{\cdot}114{+}76$	$p_5 := p_4 - 5 \cdot p_3 = -27$
$114 = 1{\cdot}76{+}38$	$p_6 := p_5 - 1 \cdot p_4 = 32$
$76 = 2{\cdot}38{+}0$	$p_7 := p_6 - 2 \cdot p_5 = -91$

$4864 = 1{\cdot}3458{+}1406$	$p_2 = 1 \quad q_2 = -1$
$3458 = 2{\cdot}1406{+}646$	$p_3 = -2 \quad q_3 = 3$
$1406 = 2{\cdot}646{+}114$	$p_4 = 5 \quad q_4 = -7$
$646 = 5{\cdot}114{+}76$	$p_5 = -27 \quad q_5 = 38$
$114 = 1{\cdot}76{+}38$	$p_6 = 32 \quad q_6 = -45$
$76 = 2{\cdot}38{+}0$	$p_7 = -91 \quad q_7 = 128$

$$4864 \cdot (-91) + 3458 \cdot (128) = 38$$

Čeprav uporabljamo ta algoritem že stoletja, pa je presenetljivo, da ni vedno najboljša metoda za iskanje največjega skupnega delitelja.

R. Silver in **J. Terzian** sta leta **1962**
(v lit. J. Stein, *J. Comp. Phys.* **1** (1967), 397-405)
predlagala **binarni algoritem**:

B1. *Poišči tak največji* $k \in \mathbb{Z}$, *da bosta števili* a *in* b *deljivi z* 2^k; $a \leftarrow a/2^k$ *in* $b \leftarrow b/2^k$, $K \leftarrow 2^k$.

B2. *Dokler* $2|a$ *ponavljaj* $a \leftarrow a/2$ *in dokler* $2|b$ *ponavljaj* $b \leftarrow a/2$.

B3. *Če je* $a = b$, *je* $D(a,b) = a * K$, *sicer pa v primeru* $a > b$, *priredi* $a \leftarrow a-b$, *sicer* $b \leftarrow b-a$ *in se vrni na korak B2.*

Lehmerjev algoritem deli z majhnimi namesto velikimi števili (izboljšave J. Sorenson, Jaebelan,...).

Dobro vprašanje je kako prenesti te ideje v $\mathrm{GF}(2^n)$.

R. Schroeppel je že naredil prvi korak s svojim algoritmom **almost inverse**.

Kitajski izrek o ostankih. *Če so števila* $m_1, m_2, ..., m_r$ *paroma tuja, tj.* $D(m_i, m_j) = 1$ *za* $i \neq j$, *in* $a_1, a_2, ..., a_r \in \mathbb{Z}$, *potem ima sistem kongruenc*

$$\begin{aligned} x &\equiv a_1 \pmod{m_1} \\ x &\equiv a_2 \pmod{m_2} \\ &\vdots \\ x &\equiv a_r \pmod{m_r} \end{aligned}$$

enolično rešitev po modulu $M = m_1 \cdot m_2 \cdots m_r$,

$$x = \sum_{i=1}^{r} a_i \cdot M_i \cdot y_i \pmod{M},$$

kjer je $M_i = M/m_i$, $y_i = M_i^{-1} \bmod m_i$, $i = 1, ..., r$.

(angl. Chinese Reminder Theorem oziroma CRT)

Red elementa g v končni multiplikativni grupi je najmanjše celo število m tako, da $g^m = 1$.

Lagrangev izrek: *Naj bo G multiplikativna grupa reda n in $g \in G$, potem red g deli n.*

Naj bo p praštevilo. Generatorju multiplikativne grupe $\mathbb{Z}_p^*$ pravimo **primitiven element**.

DN: Koliko primitivnih elementov ima $\mathbb{Z}_p$?
Naj bo α primitiven element, potem za $\forall\beta \in \mathbb{Z}_p^*$ obstaja tak $i \in \{0, 1, \ldots, p-2\}$, da je $\beta = \alpha^i$.
Pokaži, da je red elementa β enak $\dfrac{p-1}{D(p-1, i)}$.

Eulerjevo funkcijo φ definiramo s

$$\varphi(n) = |\{x \in \mathbb{N} \,|\, x < n \text{ in } D(x,n) = 1\}|.$$

Potem za praštevilo p, naravno število n in poljubni tuji si števili a in b velja

$$\varphi(p^n) = p^n - p^{n-1} \quad \text{in} \quad \varphi(ab) = \varphi(a)\varphi(b).$$

Če poznamo faktorizacijo števila n, poznamo tudi $\varphi(n)$.

Fermatov izrek

Za praštevilo p in $b \in \mathbb{Z}_p$ velja $\quad b^p \equiv b \pmod{p}$.

Eulerjev izrek

Če je $a \in \mathbb{Z}_n^*$ oziroma $D(n,a) = 1$, potem velja

$$a^{\varphi(n)} \equiv 1 \pmod{n}.$$

Opis in implementacija RSA

Generiranje ključev: najprej izberemo

- praštevili $\boldsymbol{p}$, $\boldsymbol{q}$ ter izračunamo *modul* $\boldsymbol{n} := pq$, in
- *šifrirni eksponent* $\boldsymbol{e}$, tako da je $D(e, \varphi(n)) = 1$,

nato pa izračunamo *odšifrirni eksponent* $\boldsymbol{d}$ iz kongruence

$$ed \equiv 1 \pmod{\varphi(n)}$$

z razširjenim Evklidovim algoritmom (ali pa potenciranjem).

Javni ključ je $\boldsymbol{(e, n)}$, **zasebni ključ** pa $\boldsymbol{(d, p, q)}$.

Šifriranje: $E(e,n)(x) = x^e \pmod{n}$.

Odšifriranje: $D(d,p,q)(y) = y^d \pmod{n}$.

Šifriranje in odšifriranje sta inverzni operaciji.
Za $x \in \mathbb{Z}_n^*$ to sledi iz Eulerjeve kongruence:

$$(x^e)^d \equiv x^{r\varphi(n)+1} \equiv (x^{\varphi(n)})^r x \equiv x \pmod{n},$$

za $x \in \mathbb{Z}_n \backslash \mathbb{Z}_n^*$ pa se prepričajte sami za DN.

Generiranje podpisa:
za podpis sporočila $m \in \{0,1\}^*$, Anita:

1. izračuna $M = H(m)$,
   kjer je H zgoščevalna funkcija (npr. SHA-1),
2. izračuna $s = M^d \bmod n$,
3. Anitin podpis za m je s.

Preverjanje podpisa:
Bojan preveri Anitin podpis s za m, tako da:

1. vzame overjeno kopijo Anitinega
   javnega ključa (n, e),
2. izračuna $M = H(m)$,
3. izračuna $M' = s^e \bmod n$,
4. sprejme (m, s) če in samo če je $M = M'$.

Potenciranje z redukcijo pri RSA je enosmerna funkcija z bližnjico.

Bližnjica: poznavanje števila d oziroma $\varphi(n)$ oziroma števil p in q.

RSA v praksi

- Modul $n = pq$ mora biti dovolj velik, da je njegova faktorizacija računsko prezahtevna.
- Implementacije RSA z dolžino ključev 512 bitov ne jamčijo več dolgoročne varnosti.

Časovna zahtevnost računskih operacij

Naj ima število n v binarni representaciji k bitov, tj.

$$k = \lfloor \log_2 n \rfloor + 1.$$

Potem je časovna zahtevnost

seštevanja $O(k)$,
Evklidovega algoritma $O(k^2)$,
modularne redukcije $O(k^2)$,
potenciranja pa $O(k^3)$.

Potenciranje opravimo učinkovito z metodo "**kvadriraj in množi**".

Izbira šifrirnega eksponenta

$$e = 5, 17, 2^{16} + 1$$

Pospešitev odšifriranja z uporabo kitajskega izreka o ostankih (CTR) za faktor 4:

namesto da računamo $y^d \bmod n$ direktno, najprej izračunamo

$C_p := y^{d \bmod (p-1)} \bmod p$ in $C_q := y^{d \bmod (q-1)} \bmod q$,

nato pa po CRT še

$$C := t_p C_p + t_q C_q \bmod n,$$

kjer $p \mid t_p - 1, t_q$ in $q \mid t_p, t_q - 1$.

Algoritem RSA je cca. 1500-krat počasnejši od DES-a. Uporablja se za prenos ključev simetričnega algoritma.

(Za 512-bitno število n lahko dosežemo z RSA-jem hitrost 600 Kb na sekundo, medtem ko DES zmore 1 Gb na sekundo.)

Nekaj lažjih nalog

1. Koliko množenj potrebujemo, da izračunamo m^d?
2. Prepričaj se, ali je dovolj, da pri RSA uporabimo le Fermatovo kongruenco.

3. Pokaži, da $p \mid \binom{p}{i}$, za $1 < i < p$.
4. Naj bo p praštevilo, potem za poljubni števili a in b velja
$$(a+b)^p \equiv a^p + b^p \pmod{p}.$$
5. Naj bo p praštevilo, potem za poljubno število m velja $m^p \equiv m \pmod{p}$.

Gostota praštevil

Izrek o gostoti praštevil

[de la Vallée Poussin, Hadamard, 1896]

Funkcija $\pi(x)$ *je asimptotično enaka* $\dfrac{x}{\log x}$,

ko gre $x \to \infty$.

(angl. **Prime Number Theorem** oziroma PNT)

Domnevo za PNT je prvi postavil leta 1791 (še kot najstnik) **Frederic Gauss (1777-1855)**, za testiranje pa je kasneje uporabljal tudi tablice **Jurija Vege** iz leta 1796:

$$\pi(x) \approx \int_2^x \frac{1}{\log n} dn \ .$$

Legendre pa jo je objavil v svoji knjigi iz leta 1808:

$$\pi(x) \approx \frac{x}{\log x - 1.08366} \ .$$

Namesto da bi šteli praštevila, ki so manjša ali enaka številu n, raje poglejmo, kakšna je njihova *gostota*:

$$\pi(n)/n.$$

Primerjamo

$$\pi(10^{10})/10^{10} = .04550525$$

z

$$1/\ln(10^{10}) = .04342945.$$

To je bil **problem 19. stoletja.**

Peter Gustav Lejeune-Dirichlet (1805-1859) (začetki analitične teorije števil): za vsaki tuji si celi števili a in b aritmetično zaporedje

$$a, \quad a+b, \quad a+2b, \quad a+3b, \quad \ldots, a+nb, \quad \ldots$$

vsebuje neskončno praštevil.

Pafnuti Lvovich Chebyshev (1821-1884) je leta 1850 pokazal, da, če limita obstaja, potem leži na intervalu

$$[0.92129, 1.10555].$$

Leta 1859 je **Georg Friedrich Bernhard Riemann (1826-1866)** naredil briljanten napredek na področju analitične teorije števil s študijem Riemannove **zeta funkcije**.

Leta 1896 sta končno dokazala domnevo

Charles-Jean-Gustave-Nicholas de la Vallée-Poussin (1866-1962)

in

Jacques Hadamard (1865-1963).

V Prilogi A si lahko ogledate dokaz izreka, ki sledi D. Zagieru, ki je uporabil analitični izrek namesto Tauberjevih izrekov. (Newman's Short Proof of the Prime Number Theorem, *American Mathematical Monthly*, October 1997, strani 705-709).

Še nekaj zanimivih referenc:

J. Korevaar, On Newman's quick way to the prime number theorem, *Mathematical Intelligencer* 4, **3** 1982, 108-115.

P. Bateman and H. Diamond, A hundred years of prime numbers, *American Mathematical Monthly* **103** 1996, 729-741.

Elementarni dokaz izreka o gostoti praštevil

Prvi dokaz so poenostavili **Landau** in drugi v začetku 20. stoletja. Vsi so uporabljali zapletene metode realne in kompleksne analize.

Leta 1949 sta **Atle Selberg** in **Paul Erdös** odkrila neodvisno elementaren dokaz (brez kompleksne analize).

Leta 1956 je **Basil Gordon** dokazal izrek o gostoti praštevil s pomočjo Stirlingove formule za $n!$.

The Times London, sept. 25, 1996:

Selberg and Erdös agreed to publish their work in back-to-back papers in the same journal, explaining the work each had done and sharing the credit. But at the last minute Selberg ... raced ahead with his proof and published first. The following year Selberg won the Fields Medal for this work. Erdös was not much concerned with the competitive aspect of mathematics and was philosophical about the episode.

`http://www-groups.dcs.st-and.ac.uk/`
`~history/Mathematicians/Erdos.html`

Posledica: *Če je p_n n-to praštevilo, velja*

$$\lim_{n\to\infty} \frac{n\log n}{p_n} \;=\; 1.$$

Dokaz: Logaritmirajmo limito iz izreka o gostoti praštevil

$$\lim_{x\to\infty} \left(\log\pi(x) + \log\log x - \log x\right) \;=\; 0$$

oziroma

$$\lim_{x\to\infty} \left\{ \log x \left(\frac{\log\pi(x)}{\log x} + \frac{\log\log x}{\log x} - 1 \right) \right\} \;=\; 0\ .$$

Ker gre $\log x \to \infty$, velja

$$\lim_{x\to\infty} \left\{ \frac{\log \pi(x)}{\log x} + \frac{\log\log x}{\log x} - 1 \right\} = 0$$

oziroma ker gre $\log\log x / \log x \to 0$, tudi

$$\lim_{x\to\infty} \frac{\log \pi(x)}{\log x} = 1.$$

Pomnožimo še z limito iz izreka o gostoti praštevil in dobimo

$$\lim_{x\to\infty} \frac{\pi(x)\log \pi(x)}{x} = 1,$$

kar pa je že želena limita, če vzamemo $x = p_n$ oziroma $\pi(x) = n$. ■

RSA sistem in faktorizacija

- Probabilistično testiranje praštevilčnosti (ponovitev Monte Carlo algoritem, Solovay-Strassen algoritem in Miller-Rabinov test)
- Napadi na RSA (odšifrirni eksponent, Las Vegas algoritem)
- Rabinov kriptosistem
- Algoritmi za faktorizacijo (naivna metoda, metoda $p-1$, Dixonov algoritem in kvadratno rešeto)

Generiranje praštevil

Za inicializacijo RSA kriptosistema potrebujemo velika (npr. 80-mestna) naključna praštevila.

V praksi generiramo veliko naključno število in testiramo, ali je praštevilo z Monte Carlo algoritmom (npr. Solovay-Stassen ali Miller-Rabin).

Ti algoritmi so hitri, vendar pa so probabilistični in ne deterministični. Po izreku o gostoti praštevil je verjetnost, da je naključno 512-bitno liho število praštevilo, približno $2/\log p \approx 2/177$.

S praštevili, ki so "osnovni gradniki" matematike, so se ukvarjali učenjaki vse od antičnih časov dalje.

Odločitveni problem `praštevilo`
Za dano število n ugotovi ali je praštevilo.

Leta **240 pr. n. št.** se je grški matematik in filozof **Eratostenes**, bibliotekar aleksandrijske knjižnice, domislil prve neoporečne metode (*čas. zahtev.* $O(n)$). V primeru zelo dolgih števil bi za rešitev tega problema potrebovali več časa kot je staro vesolje.

Od tedaj so matematiki poskušali najti algoritem, ki bi dal odgovor v smiselnem času.

Karl Frederich Gauss (1777-1855) je v svoji knjigi Disquisitiones Arithmeticae (1801) zapisal:

> *"Menim, da čast znanosti narekuje,*
> *da z vsemi sredstvi iščemo rešitev tako*
> *elegantnega in tako razvpitega problema."*

Od prihoda računalnikov dalje poudarek ni več na iskanju matematične formule, ki bi dajala praštevila, ampak na iskanju učinkovitega algoritma za razpoznavanje praštevil.

Večji korak naprej je v 17. stoletju napravil **Fermat**, z že omenjenim **Fermatovim malim izrekom**:

$$a^{p-1} \equiv 1 \pmod{p}$$

za vsak $a \in \mathbb{N}$ in vsako praštevilo p, ki ne deli a.

Po zaslugi kriptografije so postale raziskave problema `praštevilo` v zadnjih desetletjih še intenzivnejše:

1976	**Miller**: deterministični algoritem polinomske časovne zahtevnosti (temelji na Riemannovi hipotezi)
1977	**Solovay in Strassen**: verjetnostni algoritem časovne zahtevnosti $O(\log^3 n)$.
1980	**Rabin**: modifikacija Millerjevega testa v verjetnostni alg. (pravilnost dokazana)
1983	**Adleman, Pomerance in Rumely**: det. alg. čas. zathev. $O(\log n^{O(\log\log\log n)})$
1986	**Golwasser in Kilian**: polinomski verj. alg. za skoraj vse podatke z uporabo eliptičnih krivulj
2002	**Agrawal, Kayal in Saxena (AKS)**: det. alg. s časovno zahtevnostjo $O(\log^{12} n)$ v praksi $O(\log^6 n)$, tudi $O(\log^3 n)$ a brez dokaza.

Naj bo p liho praštevilo, $0 \leq x \leq p-1$.
Potem je x **kvadratni ostanek** po modulu p,
tj. $x \in \mathbf{QR}(p)$, če ima kongruenca

$$y^2 \equiv x \pmod{p}$$

rešitev $y \in \mathbb{Z}_p$.

Eulerjev kriterij
Naj bo p liho praštevilo. Potem je

$$x \in \mathrm{QR}(p) \quad \Longleftrightarrow \quad x^{(p-1)/2} \equiv 1 \pmod{p}.$$

Torej obstaja polinomski algoritem za odločitveni problem **kvadratnega ostanka**.

Dokaz:

Naj bo p liho praštevilo in a nenegativno celo število. Potem je **Legendrov simbol** definiran z

$$\left(\frac{a}{p}\right) = \begin{cases} 0, & \text{če } p \mid a, \\ 1, & \text{če je } a \in \mathrm{QR}(p), \\ -1, & \text{sicer.} \end{cases}$$

Po Eulerjevem kriteriju velja

$$\left(\frac{a}{p}\right) \equiv a^{(p-1)/2} \pmod{p}.$$

Legendrov simbol posplošimo v Jacobijev simbol. Število n naj bo celo liho število z naslednjo praštevilsko faktorizacijo $n = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$.

Za nenegativno celo število a definiramo **Jacobijev simbol** z

$$\left(\frac{a}{n}\right) \quad := \quad \Pi_{i=1}^{k} \left(\frac{a}{p_i}\right)^{e_i}.$$

Eulerjevo psevdopraštevilo: $91 = 7{\cdot}13$, vseeno pa obstaja tak $a = 10$, da je

$$\left(\frac{10}{91}\right) = -1 = 10^{45} \mod 91.$$

DN: Pokaži, da je za poljubno sestavljeno število n, m Eulerjevo psevdopraštevilo glede na bazo a za največ polovico naravnih števil, ki so manjša od n (glej nalogo 4.14).

DA-naklonjen Monte Carlo algoritem je probabilistični algoritem za odločitveni problem (tj. DA/NE-problem), pri katerem je “DA” odgovor (vedno) pravilen, “NE” odgovor pa je lahko nepravilen.

Verjetnost napake za **DA-naklonjen Monte Carlo** algoritem je ε, če za vsak odgovor “DA” algoritem odgovori z “NE” z verjetnostjo kvečjemu ε.

Solovay-Strassen algoritem

1. *Izberi naključno število* $a \in \mathbb{Z}_n$, $\quad x := \left(\frac{a}{n}\right)$.
2. **if** $x = 0$ **then return**(*"n je sestavljeno število"*).
3. $y := a^{(n-1)/2} \bmod n$,
   if $x \equiv y \pmod{n}$
   then return(*"n je praštevilo"*)
   else return(*"n je sestavljeno število"*).

Verjetnost napake pri Solovay-Strassen algoritmu je kvečjemu $1/2$ (glej nalogo 4.14 v Stinsonu).

Monte Carlo verjetnostni algoritem za odločitveni problem, ali je število sestavljeno: test ponovimo m-krat z naključnimi vrednostmi a. Verjetnost, da bo odgovor napačen m-krat zapored napačen je ε^m, vendar pa iz tega še ne moremo zaključiti, da je verjetnost, da je n praštevilo, $1 - \varepsilon^m$.

Dogodek A:

"naključen lih n določene velikosti je sestavljen"

in dogodek B:

"algoritem odgovori 'n je praštevilo' m-krat zapored."

Potem očitno velja $P(B/A) \leq \varepsilon^m$, vendar pa nas v resnici zanima $P(A/B)$, kar pa ni nujno isto.

Naj bo $N \leq n \leq 2N$ in uporabimo izrek o gostoti praštevil

$$\frac{2N}{\log 2N} - \frac{N}{\log N} \approx \frac{N}{\log N} \approx \frac{n}{\log n}\ .$$

Sledi $P(A) \approx 1 - 2/\log n$. Bayesovo pravilo pravi:

$$P(A/B) = \frac{P(B/A)P(A)}{P(B)}\ .$$

Imenovalec je enak $P(B/A)P(A) + P(B/\overline{A})P(\overline{A})$.

Upoštevajmo še $P(B/\overline{A}) = 1$ in dobimo

$$P(A/B) = \frac{P(B/A)(\log n - 2)}{P(B/A)(\log n - 2) + 2} \leq$$

$$\leq \frac{2^{-m}(\log n - 2)}{2^{-m}(\log n - 2) + 2} = \frac{(\log n - 2)}{\log n - 2 + 2^{m+1}},$$

kar pomeni, da gre iskana verjetnost eksponentno proti 0.

Monte Carlo verjetnostni algoritem za odločitveni problem ali je število sestavljeno:

Test ponovimo k-krat z različlnimi vrednostmi a. Verjetnost, da bo ogovor k-krat zapored napačen, je za nas ocenjena z ε^k.

DN: Iz naslednjega izreka izpeljite, da za izračun Jacobijevega simbola ne potrebujemo praštevilske faktorizacije števila n.

Gaussov izrek

Izrek o kvadratni recipročnosti (1796)

Če sta p in q različni lihi praštevili, potem velja

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{p-1}{2}\cdot\frac{q-1}{2}}$$

ter za praštevilo 2

$$\left(\frac{2}{p}\right) = (-1)^{\frac{p^2-1}{8}}.$$

Zakaj je ta izrek tako pomemben?

Pomaga nam, da odgovorimo, kdaj imajo kvadratne kongruence rešitev, saj velja multiplikativno pravilo

$$\left(\frac{ab}{p}\right) = \left(\frac{a}{p}\right)\left(\frac{b}{p}\right).$$

Predstavlja pa tudi nepričakovano zvezo med pari praštevil (pravilo, ki ureja praštevila).

Eisensteinova lema. *$p>2$ praštevilo, $p \nmid q \in \mathbb{N}$. Naj bo $A := \{2,4,6,\dots,p-1\}$ in $r_a := qa \bmod p$ za $a \in A$. Potem je*

$$\left(\frac{q}{p}\right) = (-1)^{\sum\limits_{a\in A} r_a}.$$

Dokaz: Za $a, a' \in A$, $a \neq a'$, ne more veljati

$r_a(-1)^{r_a} = r_{a'}(-1)^{r_{a'}}$ oziroma $qa \equiv \pm qa' \pmod{p}$,

saj bi od tod sledilo $a = \pm a'$, kar pa ni mogoče.

Opozorimo še, da so vsa števila $r_a(-1)^{r_a} \bmod p$ soda, torej pretečejo ravno vse elemente množice A.

Od tod dobimo

$$\prod a \equiv (-1)^{\sum r} \prod r \pmod{p},$$

očitno pa neposredno iz definicije sledi tudi

$$q^{\frac{p-1}{2}} \prod a \equiv \prod r \pmod{p}.$$

Torej velja $q^{\frac{p-1}{2}} \equiv (-1)^{\sum r} \pmod{p}$ in po Eulerjevem kriteriju še

$$\left(\frac{q}{p}\right) \equiv q^{\frac{p-1}{2}} \pmod{p}. \quad \blacksquare$$

Oglejmo si Eisensteinov *dokaz Gaussovega izreka o kvadratni recipročnosti.* Očitno velja

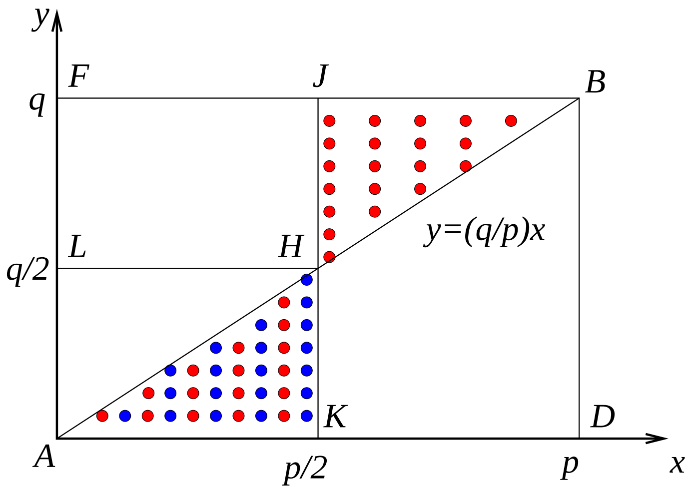
$$\sum qa = p \sum \left\lfloor \frac{qa}{p} \right\rfloor + \sum r \ .$$

Ker so elementi a vsi sodi in je p lih, velja

$$\sum r \equiv \sum \left\lfloor \frac{qa}{p} \right\rfloor \pmod{2}$$

in zato iz Eisensteinove leme sledi

$$\left(\frac{q}{p}\right) = (-1)^{\sum \left\lfloor \frac{qa}{p} \right\rfloor} \ .$$



Vsota $\sum \left\lfloor \frac{qa}{p} \right\rfloor$ je enaka številu celoštevilčnih točk sodo x-koordinato, ki ležijo v notranjosti trikotnika ABD. Sedaj pa si oglejmo točke z x-koordinato večjo od $p/2$. Ker pa je $q-1$ sod, je parnost števila $\left\lfloor \frac{qa}{p} \right\rfloor$ točk z isto x-koordinato pod diagonalo AB enako številu točk z isto sodo x-koordinato nad diagonalo AB.

To pa je po drugi strani enako številu točk pod diagonalo AB z liho x-koordinato $p - a$ (bijektivna korespondenca med točkami s sodo x-koordinato v BHJ in liho x-koordinato v AHK). Od tod sledi, da ima vsota $\sum \left\lfloor \frac{qa}{p} \right\rfloor$ enako parnost kot številu μ celoštevilčnih točk v notranjosti trikotnika AHK, tj.

$$\left(\frac{q}{p}\right) = (-1)^{\mu}.$$

Če zamenjamo p in q, dobimo še število ν celoštevilčnih točk v notranjosti trikotnika AHL, kar nam da

$$\left(\frac{p}{q}\right) = (-1)^{\nu}$$

in skupaj s prejšnjo relacijo Gaussov izrek. ■

Še en Monte Carlo algoritem za testiranje sestavljenosti števil.

Miller-Rabinov test: *testiramo liho število n.*

1. $n - 1 = 2^k m$, *kjer je m liho število,*
2. *izberemo naključno naravno število $a < n$,*
3. *izračunamo $b \equiv a^m \pmod{n}$,*
4. **if** $b \equiv 1 \pmod{n}$ **then** *n je praštevilo;* **exit;**
5. **for** $i = 0$ **to** $k - 1$ **do**

   if $b \equiv -1 \pmod{n}$

   then *n je praštevilo;* **exit;**

   else $b \equiv b^2 \pmod{n}$,
7. *število n je sestavljeno.*

Izrek: *Miller-Rabinov algoritem za problem sestavljenih števil je DA-naklonjen Monte Carlo algoritem.*

Dokaz: Predpostavimo, da algoritem odgovori "n je sestavljeno število" za neko praštevilo p.

Potem je $a^m \not\equiv 1 \pmod{n}$.

Sledi $a^{2^i m} \not\equiv -1 \pmod{n}$ za $i \in \{0, 1, \ldots, k-1\}$.

Ker je $n = 2^k m + 1$ praštevilo, iz Fermatovega izreka sledi

$$a^{2^k m} \equiv 1 \pmod{n}$$

in je $a^{2^{k-1} m}$ koren od 1 po modulu n.

Iz $x^2 \equiv 1 \pmod{n}$ oziroma $n | x^2 - 1 = (x-1)(x+1)$ sledi

$$x \equiv 1 \pmod{n} \quad \text{ali} \quad x \equiv -1 \pmod{n}$$

oziroma v našem primeru $a^{2^{k-1}m} \equiv 1 \ (\mathrm{mod}\ n)$. Na isti način pridemo do

$$a^m \equiv 1 \ (\mathrm{mod}\ n),$$

kar je protislovje, saj bi algoritem v tem primeru odgovoril "n je praštevilo". ■

Za konec omenimo brez dokaza še, da je verjetnost napake Miller-Rabinovega algoritma kvečjemu 1/4.

Napadi na RSA

Odličen pregledni članek "Twenty Years of Attacks on the RSA kriptosysystem", je objavil Dan Boneh v *Notices of AMS*, Feb. 1999, pp. 203-212.

Mi bomo omenili le nekaj osnovnih napadov.

Če poznamo $\varphi(n)$ in n, dobimo p, q iz naslednjega sistema dveh enačb

$$n = pq \qquad \text{in} \qquad \varphi(n) = (p-1)(q-1).$$

Odšifrirni eksponent kriptosistema RSA

Trditev: *Vsak algoritem A, ki najde odšifrirni eksponent d, lahko uporabimo kot podprogram v probabilističnem algoritmu, ki najde faktorje števila n.*

Od tod sledi, da iskanje odšifrirnega eksponenta ni nič lažje kot problem faktorizacije.

Opozorilo: če "izgubimo" d, moramo poleg šifrirnega eksponenta zamenjati tudi modul n.

Naj bo $\varepsilon \in [0,1)$. **Las Vegas algoritem** je probabilističen algoritem, ki za dani primer problema, lahko *ne da odgovora* z verjetnostjo ε (se pravi, da konča s sporočilom "ni odgovora"). Če pa algoritem odgovori, potem je *odgovor gotovo pravilen.*

DN: Pokaži, da je povprečno pričakovano število ponovitev algoritma vse dokler ne dobimo odgovora, enako $1/(1-\varepsilon)$ (glej nalogo 4.15).

Če Las Vegas algoritem faktorizira število n z verjetnostjo vsaj ε in ga ponovimo m-krat, potem bo število n faktorizirano z verjetnostjo vsaj $1-\varepsilon^m$.

Trditev sledi iz algoritma, ki uporablja naslednje: za $n = pq$, kjer sta p, q lihi praštevili,

$$x^2 \equiv 1 \pmod{n}, \quad \text{tj.} \quad pq|(x-1)(x+1),$$

dobimo štiri rešitve; dve (trivialni) rešitvi iz enačb

$$x \equiv 1 \pmod{n} \ \text{ in } \ x \equiv -1 \pmod{n}$$

in s pomočjo kitajskega izreka o ostankih iz

$$x \equiv 1 \pmod{p}, \qquad x \equiv -1 \pmod{q}$$

in

$$x \equiv -1 \pmod{p}, \qquad x \equiv 1 \pmod{q}$$

še dve (netrivialni) rešitvi.

Algoritem za faktorizacijo z danim šifr. eksp. d

1. Izberi naključno naravno število $w < n$,
2. izračunaj $x = D(w, n)$,
3. **if** $1 < x < n$ **then exit**(uspeh $x = p$ ali $x = q$)
4. izračunaj $d = A(e, n)$ in zapiši $de - 1 = 2^s r$, r lih,
5. izračunaj $v = w^r \mod n$,
6. **if** $v \equiv 1 \pmod{n}$ **then exit**(neuspeh)
7. **while** $v \not\equiv 1 \pmod{n}$ **do** $v_0 = v$, $v = v^2 \mod n$
8. **if** $v_0 \equiv -1 \pmod{n}$ **then exit**(neuspeh)
   else izračunaj $x = D(v_0 + 1, n)$
   (uspeh: $x = p$ ali $x = q$) .

Naključne napake

(Boneh, DeMillo in Lipton, 1997)

Če uporabimo CRT in pride pri samo enem izmed C_p in C_q do napake, npr. C_p je pravilen, $\hat{C}_q$ pa ni, potem je $\hat{C} = t_p C_p + t_q \hat{C}_q$ očitno nepravilen podpis, saj je $\hat{C}^e \neq M \bmod N$. Vendar pa je

$$\hat{C}^e = M \bmod p, \quad \text{medtem, ko je} \quad \hat{C}^e \neq M \bmod q$$

in nam $D(n, \hat{C}^e - M)$ odkrije netrivialni faktor števila n.

Rabinov kriptosistem

Temelji na tem, da je težko najti faktorizacijo produkta dveh velikih praštevil p in q.

$$n = pq, \quad p \neq q, \quad p, q \equiv 3 \pmod 4, \quad \mathcal{P} = \mathcal{C} = \mathbb{Z}_n$$

$$\mathcal{K} = \{(n, p, q, B);\ 0 \leq B \leq n-1\}$$

Za izbrani ključ $K = (n, p, q, B)$ naj bo:

$$e_K(x) = x(x+B) \pmod n,$$

$$d_K(y) = \sqrt{y + B^2/4} - B/2.$$

Javni ključ *je* (n, B), **zasebni ključ** *pa* (p, q).

Trditev: *Naj bo* $\omega^2 \equiv 1 \pmod{n}$ *netrivialen koren (kongruenca ima 4 rešitve:* 1, -1 *in še dve netrivialni), in* $x \in \mathbb{Z}_n$, *potem velja:*

$$e_K(\omega(x + B/2) - B/2) = e_K(x).$$

Imamo 4 čistopise, ki ustrezajo tajnopisu $e_K(x)$:

$$x, \quad -x - B, \quad \omega(x + \frac{B}{2}) \;\; \text{in} \quad -\omega(x + \frac{B}{2}).$$

V splošnem se ne da ugotoviti, kateri je pravi.

Odšifriranje

Imamo tajnopis y in iščemo x, ki zadošča naslednji enačbi:

$$x^2 + Bx \equiv y \pmod{n}.$$

Poenostavimo: $x = x_1 - B/2$,

$$x_1^2 \equiv y + B^2/4 \pmod{n}, \qquad C = y + B^2/4.$$

Iščemo kvadratne korene enačbe $\boxed{x_1^2 \equiv C \pmod{n}}$.

To je ekvivalentno sistemu:

$$\left.\begin{array}{l} x_1^2 \equiv C \pmod{p} \\ x_1^2 \equiv C \pmod{q} \end{array}\right|$$

Eulerjev izrek:
$C^{(p-1)/2} \equiv 1 \pmod{p}$

predpostavka: $p \equiv 3 \pmod{4}$

$\downarrow \qquad \Rightarrow \quad (\pm C^{(p+1)/4})^2 \equiv C \pmod{p}$

$$\left.\begin{array}{l} x_1 \equiv x_{1,2} \pmod{p} \\ x_1 \equiv x_{3,4} \pmod{q} \end{array}\right|$$

$\Rightarrow$ korena prve enačbe sta:
$x_{1,2} = \pm C^{(p+1)/4}$

korena druge enačbe pa:
$x_{3,4} = \pm C^{(q+1)/4}$

$\Downarrow$ KIO
x_1, x_2, x_3, x_4

Primer: $n = 77 = 7 \cdot 11, \quad B = 9$

$e_K(x) = x^2 + 9x \pmod{77}$

$$d_K(y) = \sqrt{y + B^2/4} - B/2 = \sqrt{1 + y} - 43 \pmod{77}$$

Tajnopis: $y = 22$. Poiskati moramo rešitve:

$$\begin{array}{l|}x^2 \equiv 23 \pmod{7} \\ x^2 \equiv 23 \pmod{11} \\ \hline\end{array} \qquad \begin{array}{c}(x \equiv \pm 4 \pmod{7}) \\ (x \equiv \pm 1 \pmod{11})\end{array}$$

Dobimo štiri sisteme dveh enačb z dvema neznankama, npr.:

$$x \equiv 4 \pmod{7}, \quad x \equiv 1 \pmod{11}$$

Po kitajskem izreku o ostankih velja:

$$x = 4 \cdot 11 \cdot (11^{-1} \bmod 7) + 1 \cdot 7 \cdot (7^{-1} \bmod 11).$$

Vse rešitve so:

$$x_1 = 67 \pmod{77}, \quad x_2 = 10 \pmod{77},$$
$$x_3 = 32 \pmod{77}, \quad x_4 = -32 \pmod{77}.$$

Odšifrirani tekst je:

$$\begin{aligned} d_K(y) = \ & 67 - 43 \pmod{77} && = 24 \\ & 10 - 43 \pmod{77} && = 44 \\ & 32 - 43 \pmod{77} && = 66 \\ & 45 - 43 \pmod{77} && = 2, \end{aligned}$$

vse štiri rešitve pa se zašifrirajo v 22.

Varnost Rabinovega kriptosistema

Hipotetični algoritem A za dekripcijo Rabinovega kriptosistema lahko uporabimo kot podprogram v algoritmu tipa Las Vegas za faktorizacijo števila n z verjetnostjo vsaj $1/2$.

1. Izberemo r, $1 \leq r \leq n-1$,
2. $y := r^2 - B^2/4 \pmod{n} \quad (y = e_K(r - B/2))$,
3. $x := A(y)$,
4. $x_1 := x + B/2 \qquad\qquad (x_1^2 \equiv r^2 \pmod{n})$,
5. če velja $x_1 \equiv \pm r \pmod{n}$, potem ni odgovora, sicer ($x_1 \equiv \pm\omega \cdot r \pmod{n}$, kjer je $\omega \equiv 1 \pmod{n}$ netrivialni koren) $D(x_1 + r_1, n) = p$ (ali q).

V zadnjem primeru $n \mid (x_1 - r)(x_1 + r)$, vendar $n \not| (x_1 - r)$ in $n \not| (x_1 + r) \quad \Rightarrow D(x_1 + r, n) \neq 1$.

Verjetnost, da uspemo v enem koraku:

Def: $r_1 \sim r_2 \Leftrightarrow r_1^2 \equiv r_2^2 \pmod{n}$ $(r_1, r_2 \neq 0)$.
To je ekvivalenčna relacija, ekvivalenčni razredi v $Z_n \backslash \{0\}$ imajo moč 4: $[r] = \{\pm r, \pm \omega r\}$.
Vsak element iz [r] nam da isto vrednost y.
Podprogram A nam vrne x, $[x] = \{\pm x, \pm \omega x\}$,
$r = \pm x$: 4 ni odgovora $\quad r = \pm \omega x$: dobimo odgovor.

Ker izberemo r slučajno, je vsaka od teh možnosti enako verjetna $\Rightarrow$ verjetnost, da uspemo, je $1/2$.

Algoritmi za faktorizacijo števil

Poskušanje
Število n delimo z vsemi lihimi števili do $\sqrt{n}$:

$i := 3$,
until $i \leq \sqrt{n}$ **repeat**
if $i \mid n$, potem smo našli faktor,
else $i := i + 2$.

Algoritem je uporaben za manjše n (npr. $n \leq 10^{12}$).
Časovna zahtevnost za k bitov je $2^{k/2-1}$ deljenj.

Metoda $p - 1$ (Pollard, 1974)

Podatki: n (lih, želimo faktorizirati) in B (meja)

Algoritem temelji na naslednjem preprostem dejstvu:

če je p praštevilo, ki deli n, in za vsako praštevilsko potenco q, ki deli $p-1$, velja $q \leq B$, potem $(p-1)|B!$

Primer: $B = 9$, $p = 37$, $p - 1 = 36 = 2^2 \cdot 3^2$

$2^2 \leq B, 3^2 \leq B \Rightarrow 2^2 \cdot 3^2 | 1 \cdot 2 \cdot 3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8 \cdot 9$

Algoritem

Podatki: n, B
1. $a := 2$
2. $j = 2, \ldots, B$ $\qquad (a \equiv 2^{B!} \pmod{n})$
   $a := a^j \pmod{n}$ $\qquad (\Rightarrow a \equiv 2^{B!} \pmod{p})$
3. $d = D(a-1, n)$ $\qquad$ (Fermat: $2^{p-1} \equiv 1 \pmod{p}$)
4. Če velja $1 < d < n$, je d faktor števila n (saj $p|d$) sicer ni uspeha (to se zgodi, kadar je $d\!=\!1$).

Če $B \geq \sqrt{n}$, vedno uspemo, vendar algoritem ni učinkovit.

Časovna zahtevnost

- $B-1$ potenciranj po modulu n,
  za vsako rabimo $2\log_2 B$ množenj po modulu n,
- največji skupni delitelj z Evklid. alg.: $\mathcal{O}((\log n)^3)$.

Skupaj $\mathcal{O}(B\log B(\log n)^2+(\log n)^3)$, kar pomeni, da je za $B\approx(\log n)^i$ algoritem polinomski.

Primer: $n=143$, $B=4$, $a\equiv 2^{2\cdot 3\cdot 4}\equiv 131 \pmod{143}$. Torej je $a-1=130$ in od tod $D(130,143)=13$.

Za varen RSA izberemo $p=2p_1+1$ in $q=2q_1+1$, kjer sta p_1 in q_1 praštevili.

Dixonov algoritem in kvadratno rešeto

$$(x \not\equiv \pm y \pmod{n}, x^2 \equiv y^2 \pmod{n} \implies D(x-y, n) \neq 1)$$

Sestavimo bazo faktorjev $\mathcal{B} = \{p_1, \ldots, p_B\}$, kjer so p_i "majhna" praštevila. Naj bo C malo večji kot B (npr. $C = B + 10$). Najdemo C kongruenc:

$$x_j^2 \equiv p_1^{\alpha_{1,j}} \times p_2^{\alpha_{2,j}} \times \cdots \times p_B^{\alpha_{B,j}} \pmod{n}, \quad 1 \leq j \leq C$$

Označimo $a_j := (\alpha_{1,j} \bmod 2, \ldots, \alpha_{B,j} \bmod 2)$.
Če najdemo podmnožico $\{a_1, \ldots, a_C\}$, v kateri se vektorji seštejejo v $(0, 0, \cdots, 0) \bmod 2$, potem bo produkt x_j uporabil vsak faktor iz $\mathcal{B}$ sodo mnogokrat.

Primer: $n = 15770708441$, $\mathcal{B} = \{2, 3, 5, 7, 11, 13\}$

$$\begin{array}{rll} 8340934156^2 & \equiv 3\times 7 \pmod{n} & a_1 = (0,1,0,1,0,0) \\ 12044942944^2 & \equiv 2\times 7\times 13 \pmod{n}, & a_2 = (1,0,0,1,0,1) \\ 2773700011^2 & \equiv 2\times 3\times 13 \pmod{n}, & a_3 = (1,1,0,0,0,1) \end{array}$$

Iz $a_1 + a_2 + a_3 = (0,0,0,0,0,0)$ mod 2 sledi

$$(8340934156 \times 12044942944 \times 2773700011)^2 \equiv$$

$$\equiv (2 \times 3 \times 7 \times 13)^2 \pmod{n}$$

oziroma $9503435785^2 \equiv 546^2 \pmod{n}$

in $D(9503435785 - 546, 15770708441) = 115759.$

- Linearno odvisnost med vektorji $\{a_1, a_2, \cdots, a_C\}$ poiščemo npr. z Gaussovo eliminacijo.
- $C > B + 1$: vendar imamo raje več različnih odvisnosti, da bo vsaj ena dala faktorizacijo.
- Števila x_j, za katere se da $x_j^2 \pmod{n}$ faktorizirati v $\mathcal{B}$, iščemo v množici $\{x_j = j + \lfloor\sqrt{n}\rfloor \mid j = 1, 2, ...\}$ z metodo **kvadratnega rešeta** (Pomerance).
- Če je $\mathcal{B}$ velik, je večja možnost, da se da neko število faktorizirati v $\mathcal{B}$, a potrebujemo več kongruenc, da najdemo linearno odvisnost. ($|\mathcal{B}| \approx \sqrt{e^{\sqrt{\ln n \ln \ln n}}}$).

Algoritmi za faktorizacijo v praksi

Kvadratno rešeto	$O(e^{(1+o(1))\sqrt{\ln n \ln \ln n}})$
Eliptične krivulje	$O(e^{(1+o(1))\sqrt{\ln p \ln \ln p}})$
Številsko rešeto	$O(e^{(1.92+o(1))(\ln n)^{1/3}(\ln \ln n)^{2/3}})$

$o(1) \rightarrow 0$, ko $n \rightarrow \infty$

p - najmanjši praštevilski faktor n

V najslabšem primeru, ko je $p \approx \sqrt{n}$, imata kvadratno rešeto in eliptične krivulje približno enako časovno zahtevnost, sicer pa je boljše kvadratno rešeto.

Faktorizacije velikih števil s kvadratnim rešetom:
$(n = p \cdot q,\ p \approx q)$

leto	stevilo	bitov	metoda	opombe
1903	$2^{67} - 1$	67		F. Cole (3 leta ob ned.)
1988		250	QS	100 rac., e-posta
1994	RSA-129	425	QS	1600 rac. 8 mesecev
1999	RSA-155	512	NFS	300 del.p.+Cray; 5 mes.
2002	RSA-158	524	NFS	30 del.p.+Cray; 3 mes
2003	RSA-174	576	NFS	
2005	RSA-200	663	NFS	(55 let na eni del.p.)

Fermatova števila:

$2^{2^{11}} - 1$ eliptične krivulje: 1988 (Brent)

$2^{2^{9}} - 1$ številsko rešeto:
1990 (Lenstra, Lenstra, Manasse, Pollard)

Prof. Vidav je leta 1997 zastavil naslednje vprašanje (morda tudi zato, da preveri trenutne moči namiznih računalnikov): poišči prafaktorje števila

$$10^{64} + 1$$

in namignil, da so vsi prafaktorji, če jih je kaj, oblike $128k + 1$.

Večina osebnih računalnikov z Mathematica/Maple hitro najde en faktor:

$$1265011073$$

55-mestni ostanek pa povzroči težave.
V Waterlooju sem končno našel hiter računalnik (cacr: Alpha ???) ter hitro programsko opremo (glej `http://www.informatik.th-darmstadt.de/TI/LiDIA/`), ki je v manj kot 10-ih minutah našla še preostala prafaktorja

$$15343168188889137818369$$

$$515217525265213267447869906815873.$$