Napadi na RSA kriptosustav

Andrej Dujella

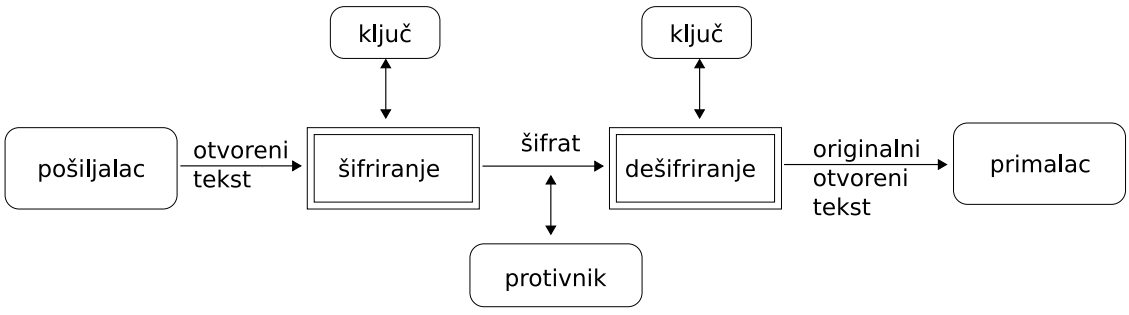
PMF - Matematički odjel
Sveučilište u Zagrebu, Hrvatska
e-mail: duje@math.hr
URL: http://web.math.hr/~duje/

Kako uspostaviti sigurnu komunikaciju preko nesigurnog komunikacijskog kanala?

Metode za rješavanje ovog problema proučava znanstvena disciplina koja se zove **kriptografija**.

Osnovni zadatak kriptografije je omogućavanje komunikacije dvaju osoba (zovemo ih *pošiljalac* i *primalac* - u kriptografskoj literaturi za njih su rezervirana imena *Alice* i *Bob*) na takav način da treća osoba (njihov *protivnik* - u literaturi se najčešće zove *Eva* ili *Oskar*), koja može nadzirati komunikacijski kanal, ne može razumjeti njihove poruke.

Poruku koju pošiljalac želi poslati primaocu zovemo *otvoreni tekst*. Pošiljalac transformira otvoreni tekst koristeći unaprijed dogovoreni *ključ* K. Taj se postupak zove *šifriranje*, a dobiveni rezultat *šifrat*. Nakon toga pošiljalac pošalje šifrat preko nekog komunikacijskog kanala. Protivnik prisluškujući može saznati sadržaj šifrata, ali kako ne zna ključ, ne može odrediti otvoreni tekst. Za razliku od njega, primalac zna ključ kojim je šifrirana poruka, pa može *dešifrirati* šifrat i odrediti otvoreni tekst.



shema simetrične kriptografije

Definicija: *Kriptosustav* je uređena petorka $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, gdje je $\mathcal{P}$ konačan skup svih otvorenih tekstova, $\mathcal{C}$ konačan skup svih šifrata, $\mathcal{K}$ konačan skup svih mogućih ključeva, $\mathcal{E}$ skup svih funkcija šifriranja i $\mathcal{D}$ skup svih funkcija dešifriranja. Za svaki $K \in \mathcal{K}$ postoji $e_k \in \mathcal{E}$ i odgovarajući $d_K \in \mathcal{D}$. Pritom su $e_K : \mathcal{P} \to \mathcal{C}$ i $d_K : \mathcal{C} \to \mathcal{P}$ funkcije sa svojstvom da je $d_K(e_K(x)) = x$ za svaki $x \in \mathcal{P}$.

Shema koju smo u uvodu opisali predstavlja tzv. *simetrični kriptosustav*. Funkcije koje se koriste za šifriranje e_K i dešifriranje d_K ovise o ključu K kojeg Alice i Bob moraju tajno razmjeniti prije same komunikacije. Kako njima nije dostupan siguran komunikacijski kanal, ovo može biti veliki problem.

Diffie i Hellman (1976): protokol za razmjenu ključeva, zasnovan na činjenici da je u nekim grupama potenciranje puno jednostavnije od logaritmiranja (*problem diskretnog logaritma*).

Diffie i Hellman se smatraju začetnicima *kriptografije javnog ključa*. Ideja javnog ključa se sastoji u tome da se konstruiraju kriptosustavi kod kojih bi iz poznavanja funkcije šifriranja e_K bilo praktički nemoguće (u nekom razumnom vremenu) izračunati funkciju dešifriranja d_K. Tada bi funkcija e_K mogla biti javna.

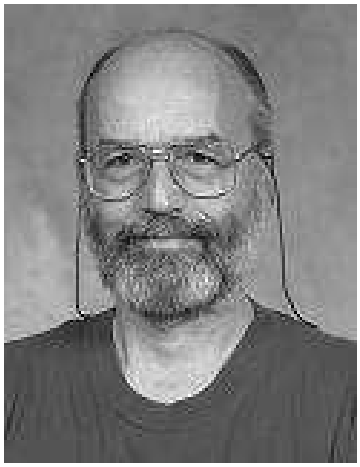
Dakle, u kriptosustavu s javnim ključem svaki korisnik K ima dva ključa: javni e_K i tajni d_K. Ako Alice želji poslati Bobu poruku x, onda je ona šifrira pomoću Bobovog javnog ključa e_B, tj. pošalje Bobu šifrat $y = e_B(x)$. Bob dešifrira šifrat koristeći svoj tajni ključ d_B, $d_B(y) = d_B(e_B(x)) = x$.

Uočimo da Bob mora posjedovati neku dodatnu informaciju (tzv. *trapdoor* - skriveni ulaz) o funkciji e_B, da bi samo on mogao izračunati njezin inverz d_B, dok je svima drugima (a posebno Evi) to nemoguće. Takve funkcije čiji je inverz teško izračunati bez poznavanja nekog dodatnog podatka zovu se *osobne jednosmjerne funkcije*.

Najpoznatiji kriptosustav s javnim ključem je **RSA kriptosustav** iz 1977. godine, nazvan po svojim tvorcima Ronaldu Rivestu, Adi Shamiru i Leonardu Adlemanu.

Njegova sigurnost je zasnovana na prvenstveno na teškoći faktorizacije velikih prirodnih brojeva. Parametri RSA kriptosustava su modul n koji je produkt dva velika prosta broja p i q, te eksponenti e i d koji se koriste za šifriranje i dešifriranje.

Ronald Rivest



Adi Shamir

Leonard Adleman

RSA kriptosustav:

Neka je $n = pq$, gdje su p i q prosti brojevi. Neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$, te

$$\mathcal{K} = \{(n, p, q, d, e) : n = pq,\ de \equiv 1 \pmod{\varphi(n)}\}.$$

Za $K \in \mathcal{K}$ definiramo

$$e_K(x) = x^e \bmod n, \qquad d_K(y) = y^d \bmod n.$$

Vrijednosti n i e su javne, a vrijednosti p, q i d su tajne, tj. (n, e) je javni, a (p, q, d) je tajni (privatni) ključ.

Ovdje je $\varphi(n) = (p-1)(q-1) = n - p - q + 1$ Eulerova funkcija. Iz njezinog svojstva da je $a^{\varphi(n)} \equiv 1 \pmod{n}$ za $\text{nzd}(a, n) = 1$, slijedi da su funkcije e_K i d_K jedna drugoj inverzne.

Standardna verzija RSA:

- p i q imaju približno jednaki broj bitova,
- $e < n$.

Sigurnost RSA kriptosustava leži u pretpostavci da je funkcija

$$e_K(x) = x^e \bmod n$$

jednosmjerna. Dodatni podatak ("trapdoor") koji omogućava dešifriranje je poznavanje faktorizacije $n = pq$.

Zaista, onaj tko zna faktorizaciju broja n, taj može izračunati $\varphi(n) = (p-1)(q-1)$, te potom dobiti eksponent d rješavajući linearnu kongruenciju

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

(pomoću proširenog Euklidova algoritma).

No, otvoreno je pitanje je li razbijanje RSA kriptosustava, tj. određivanje x iz poznavanja $x^e \bmod n$, ekvivalentno faktorizaciji od n.

Izbor parametara:

1. Tajno izaberemo dva velika prosta broja p i q slične veličine (recimo 512 bitova). Najprije generiramo slučajan prirodan broj m sa željenim brojem bitova, pa zatim pomoću nekog testa prostosti (npr. Miller-Rabinovog) tražimo prvi prost broj veći ili jednak m.

   Treba paziti da $n = pq$ bude otporan na metode faktorizacije koje su vrlo efikasne za brojeve specijalnog oblika. Tako bi brojevi $p \pm 1$ i $q \pm 1$ trebali imati barem jedan veliki prosti faktor, jer postoje efikasne metode za faktorizaciju brojeva koji imaju prosti faktor p takav da je jedan od brojeva $p-1$, $p+1$ "gladak", tj. ima samo male proste faktore. Također, p i q ne smiju biti jako blizu jedan drugome, jer ih se onda može naći koristeći činjenicu da su približno jednaki $\sqrt{n}$.

2. Izračunamo $n = pq$ i $\varphi(n) = (p-1)(q-1) = n - p - q + 1$.

3. Izaberemo broj e takav da je nzd$(e, \varphi(n)) = 1$, te pomoću proširenog Euklidova algoritma izračunamo d takav da je $de \equiv 1 \pmod{\varphi(n)}$. Obično se uzima da je $e < \varphi(n)$. Broj e se može izabrati slučajno, a ima smisla izabrati ga i što manjim, tako da bi šifriranje $x^e \bmod n$ (tzv. modularno potenciranje) bilo što brže. Broj operacija u šifriranju ovisi o veličini broja e, te o broju jedinica u binarnom zapisu od e. Stoga je dugo vremena $e = 3$ bio popularan izbor. No, vidjet ćemo da izbor vrlo malog eksponenta e predstavlja opasnost za sigurnost, te se danas preporuča izbor $e = 2^{16} + 1 = 65537$.

4. Stavimo ključ za šifriranje (n, e) u javni direktorij.

Ideja: izabrati parametre RSA kriptosustava tako da jedan od eksponenata e ili d bude mali. Budući da je broj operacija za modularno potenciranje linearan u broju bitova eksponenta, to bi moglo smanjiti vrijeme potrebno za šifriranje, odnosno dešifriranje.

To bi posebno moglo biti od interesa u situacijama kad postoji veliki nesrazmjer u snazi dvaju uređaja koji sudjeluju u komunikaciji, kao što je npr. slučaj kad "pametna kartica" komunicira s centralnim računalom. U toj situaciji bilo bi poželjno da:

- kartica ima mali tajni eksponent d,
- računalo ima mali javni eksponent e,

da bismo minimizirali onaj dio računanja koje treba provesti kartica.

Wiener (1990) -
napad na RSA s malim eksponentom d:

$ed - k\varphi(n) = 1$

$\varphi(n) \approx n \quad \Rightarrow \quad \frac{k}{d} \approx \frac{e}{n}$

Pretpostavimo da je $p < q < 2p$. Ako je $d < \frac{1}{3}n^{0.25}$, tada je

$$\left|\frac{k}{d} - \frac{e}{n}\right| < \frac{1}{2d^2}.$$

Po klasičnom Legendreovom teoremu iz diofantskih aproksimacija, d mora biti nazivnik neke konvergente p_m/q_m u razvoju u verižni razlomak broja e/n, pa se stoga d može efikasno izračunati iz javnog ključa (n, e).

Ukupan broj konvergenti je reda $O(\log n)$, a svaka konvergenta se može testirati u polinomijalnom vremenu.

Primjer: Pretpostavimo da su u RSA kriptosustavu zadani modul

$$n = 7978886869909,$$

javni eksponent

$$e = 3594320245477,$$

te da je poznato da tajni eksponent d zadovoljava $d < \frac{1}{3}n^{0.25} < 561$.

Da bismo primijenili Wienerov napad, računamo razvoj broja $\frac{e}{n}$ u verižni razlomak (pomoću Euklidovog algoritma). Dobivamo:

$$[0; 2, 4, 1, 1, 4, 1, 2, 31, \ldots] = \cfrac{1}{2 + \cfrac{1}{4 + \cfrac{}{\ddots}}}.$$

Potom računamo pripadne konvergente:

$$0, \frac{1}{2}, \frac{4}{9}, \frac{5}{11}, \frac{9}{20}, \frac{41}{91}, \frac{50}{111}, \frac{141}{313}, \frac{4421}{9814}, \ldots.$$

Konačno, provjeravamo koji od nazivnika 2, 9, 11, 20, 91, 111, 313 zadovoljava kongruenciju $(x^e)^d \equiv x \pmod{n}$ za npr. $x = 2$. Tako dobivamo da je tajni eksponent $d = 313$.

Verheul - van Tilborg (1997): Proširenje Wienerovog napada koje je primjenjivo kada d ima nekoliko bitova više od $n^{0.25}$. Za $d > n^{0.25}$, njihov napad koristi pretraživanje "grubom silom" za $2t + 8$ bitova (uz izvjesne pretpostavke na parcijalne kvocijente u verižnom razlomku), gdje je $t = \log_2(d/n^{0.25})$.

Boneh - Durfee (1999),
Blömer - May (2001):
Napadi nasnovani na Coppersmithovoj tehnici koja koristi LLL-algoritam za nalaženje malih korijena modularnih polinomijalnih jednadžbi. Ovi napadi su "heuristički", te u praksi rade zadovoljavajuće ako je $d < n^{0.292}$.

Smatra se da bi trebalo koristi tajni eksponent $d > \sqrt{n}$, jer je poznato da su svi ovi gore spomenuti napadi sasvim neprimjenjivi u toj situaciji.

D. (2004): Mala modifikacija Verheul - van Tilborgovog napada, zasnovana na Worleyevom resultatu (1981) iz diofantskih aproksimacija, koji povlači da svi racionalni brojevi p/q koji zadovoljavaju nejednakost

$$\left|\alpha - \frac{p}{q}\right| < \frac{c}{q^2},$$

za neki pozitivan realan broj c, imaju oblik

$$\frac{p}{q} = \frac{rp_{m+1} \pm sp_m}{rq_{m+1} \pm sq_m}$$

za neki $m \geq -1$ i nenegativne cijele brojeve r i s takve da je $rs < 2c$.

D. - Ibrahimpašić (2008): Worleyev rezultat je najbolji mogući, u smislu da se uvjet $rs < 2c$ ne može zamijeniri sa $rs < (2 - \varepsilon)c$ za $\varepsilon > 0$.

U oba spomenuta proširenja Wienerovog napada, kandidati za tajni eksponent su oblika $d = rq_{m+1} + sq_m$. Testiraju se sve mogućnosti za d, a broj svih mogućnosti je ugrubo (broj mogućnosti za r) $\times$ (broj mogućnosti za s), što je $O(D^2)$, gdje je $D = d/n^{0.25}$.

Preciznije, broj mogućih parova (r, s) u Verheul - van Tilborgovom napadu je $O(D^2A^2)$, gdje je $A = \max\{a_i : i = m+1, m+2, m+3\}$, dok je u mojoj varijanti iz 2004. godine $O(D^2 \log A)$ (a_i su parcijalni kvocijenti u razvoju u verižni razlomak).

Novu modifikaciju Verheul - van Tilborgovog napada su predložili Sun, Wu i Chen (2007). Ona zahtijeva (heuristički) pretraživanje "grubom silom" za $2t-10$ bitova, pa je i njena složenost također $O(D^2)$. Drastično poboljšanje se ne može očekivati, budući da su Steinfeld, Contini, Wang i Pieprzyk (2005) pokazali da u među algoritmima ovog tipa ne postoji algoritam sa subeksponencijalnom ovisnošću o D.

Testiranje

Dvije su glavne metode za testiranje:

1) izračunamo p i q uz pretpostavku da je testirana konvergenta jednaka d/k:

$$\varphi(n) = (de - 1)/k, \quad p + q = n + 1 - \varphi(n),$$

$$(q - p)^2 = (p + q)^2 - 4n,$$

$$p = ((p+q)-(q-p))/2, \quad q = ((p+q)+(q-p))/2;$$

2) za testirani d, testiramo je li zadovoljena kongruencija

$$(x^e)^d \equiv x \pmod{n},$$

recimo za $x = 2$.

Prikazat ćemo novu ideju, a to je primjena metode "susret u sredini" (*meet-in-the-middle*) na ovaj drugi test.

Želimo testirati je li

$$2^{e(rq_{m+1}+sq_m)} \equiv 2 \pmod{n}.$$

Primijetimo da je indeks m skoro fiksiran. Naime, ako je m' najveći neparan prirodan broj takav da je

$$\frac{p_{m'}}{q_{m'}} > \frac{e}{n} + \frac{2.122e}{n\sqrt{n}},$$

onda je $m \in \{m', m'+1, m'+2\}$.

Uvedimo oznake:

$$2^{eq_m+1} \bmod n = a, \quad (2^{eq_m})^{-1} \bmod n = b.$$

Tada zapravo možemo testirati kongruenciju

$$a^r \equiv 2b^s \pmod{n}.$$

To možemo napraviti tako da izračunamo a^r mod n za sve r, sortiramo rezultate, a potom računamo $2b^s$ mod n redom za svaki s i provjeravamo pojavljuje li se rezultat u prethodno dobivenoj sortiranoj listi. Na ovaj način broj koraka u testiranju postaje ugrubo (broj mogućnosti za r) + (broj mogućnosti za s).

Preciznije, vremenska složenost faze testiranja smanjuje sa $O(D^2)$ na $O(D \log D)$ (uz prostornu složenost $O(D)$).

Gore opisani napad smo implementirali u programskom paketu PARI i C++ (uz pomoć V. Petričevića), te pokazali da napad radi efikasno za vrijednosti od D do 2^{30}, tj. za $d < 2^{30} n^{0.25}$.

Za veće vrijednosti od D zahtjevi na memoriju (na standardnim računalima) postaju preveliki.

$\log_2 n$	$\log_2(2^{30} n^{0.25})$	$\log_2(n^{0.292})$
512	158	150
768	222	224
1024	286	299
2048	542	598

Moguće je smanjiti zahtjev na memoriju uz povećanje vremena izvršavanja (*space-time trade-off*), koristeći asimetrične varijante Worleyevog rezultata (s različitim ogradama za r i s).

ograda za r	ograda za s	vjerojatnost uspjeha
$4D$	$4D$	98%
$2D$	$2D$	89%
D	D	65%
D	$4D$	86%
$4D$	D	74%
$D/2$	$2D$	70%
$2D$	$D/2$	47%
$D/4$	$4D$	54%
$4D$	$D/4$	28%

Daljnje mogućnosti za poboljšanja:

- korištenje još boljih aproksimacija za $\frac{k}{d}$ nego što je $\frac{e}{n}$, npr. $\frac{e}{n+1-2\sqrt{n}}$,
- korištenje hash funkcija umjesto sortiranja.

S tim poboljšanjima, za 1024-bitni RSA modul n, raspon u kojem se ovaj novi napad može primijeniti je praktički isti kao najbolji poznati napadi zasnovani na LLL-algoritmu.

Neka je zadan polinom $f(x) \in \mathbb{Z}[x]$ stupnja δ i neka je poznato da postoji "malo" rješenje kongruencije $f(x) \equiv 0 \pmod{N}$, tj. rješenje x_0 za koje vrijedi $|x_0| < N^{1/\delta}$. Pitanje je možemo li efikasno naći x_0. Coppersmith je pokazao da je odgovor na ovo pitanje potvrdan.

Ideja: konstruirati novi polinom $h(x) = h_0 + h_1 x + \cdots + h_n x^n \in \mathbb{Z}[x]$ za kojeg će također vrijediti $h(x_0) \equiv 0 \pmod{N}$, ali koji će imati male koeficijente, tj. za koga je "norma" $\|h(x)\| := \left(\sum_{i=0}^{n} h_i^2\right)^{1/2}$ mala. Tada se može iskoristiti sljedeća jednostavna činjenica: ako za prirodan broj X vrijedi $\|h(xX)\| < \frac{N}{\sqrt{n}}$ i $|x_0| < X$ zadovoljava kongruenciju $h(x_0) \equiv 0 \pmod{N}$, onda je x_0 nultočka polinoma h, tj. vrijedi ne samo kongruencija, već i jednakost $h(x_0) = 0$.

Polinom $h(x)$ s traženim svojstvom može se naći pomoću LLL-algoritma, koji nalazi male vektore u rešetki (A. K. Lenstra, H. W. Lenstra, L. Lovász (1982)).

Boneh i Durfee su opisali jedan napad na RSA ovakvog tipa koji je primjenjiv u slučaju da je $d < n^{0.292}$. Slično kao kod Weinerova napada, kreće se od jednakosti $ed - k\varphi(n) = 1$, koja se može zapisati i kao

$$ed - k(n + 1 - p - q) = 1.$$

Stavimo $s = p+q$, $a = n+1$. Sada je nalaženje malog tajnog eksponenta d, recimo $d < n^{\delta}$, ekvivalentno nalaženju malih rješenja k i s kongruencije

$$f(k, s) = k(s - a) \equiv 1 \pmod{e}.$$

Zaista, za s i k imamo sljedeće ocjene:

$$|s| < 3\sqrt{n} \approx e^{0.5}, \quad |k| < \frac{de}{\varphi(n)} \approx e^{\delta}.$$

Dakle, situacija je slična kao kod gore navedenog Coppersmithova rezultata, samo što se ovdje radi o polinomu od dvije varijable, pa se Coppersmithov teorem ne može direktno primijeniti da bi se strogo dokazala korektnost ovog napada. Ipak, pokazalo se da on u praksi radi sasvim zadovoljavajuće.

Napad za RSA s malim javnim eksponentom e:

Neka je $e = 3$.

Pretpostavimo da imamo tri korisnika s različitim vrijednostima javnog modula n_1, n_2, n_3, te pretpostavimo da svi oni koriste isti javni eksponent $e = 3$. Nadalje, pretpostavimo da im netko želi poslati identičnu poruku m. Tada njihov protivnik može doznati sljedeće šifrate:

$$c_1 \equiv m^3 \pmod{n_1},\ c_2 \equiv m^3 \pmod{n_2},\ c_3 \equiv m^3 \pmod{n_3}.$$

Nakon toga, on može, koristeći Kineski teorem o ostatcima, naći rješenje sustava linearnih kongruencija

$$x \equiv c_1 \pmod{n_1},\ x \equiv c_2 \pmod{n_2},\ x \equiv c_3 \pmod{n_3}.$$

Na taj način, dobit će broj x sa svojstvom $x \equiv m^3 \pmod{n_1 n_2 n_3}$. No, kako je $m^3 < n_1 n_2 n_3$, zapravo vrijedi jednakost $x = m^3$, pa protivnik može izračunati originalnu poruku m tako na nađe treći korijen iz x.

Upravo opisani napad može se izbjeći tako da se porukama prije šifriranja doda neki "slučajni dodatak" (engl. random pad). Na taj način, nikad nećemo različitim primateljima slati potpuno identične poruke. No, postoje napadi (zasnovani na gore spomenutom Coppersmithovom rezultatu i LLL-algoritmu) koji pokazuju da ni u tom slučaju RSA kriptosustav s vrlo malim eksponentom e nije siguran.

Preciznije, koristi se sljedeći Coppersmithov rezultat (1997):

Neka je $f \in \mathbb{Z}[x]$ normirani polinom s stupnja δ, te $n \in \mathbb{N}$. Ako postoji x_0 takav da je $f(x_0) \equiv 0 \pmod{n}$ i $|x_0| \leq X = n^{1/\delta - \varepsilon}$, onda se x_0 može naći u vremenu koje je polinomijalno u $\log n$ i $1/\varepsilon$.

Hastadov napad (1985):

Pretpostavimo da je, prije šifriranja, na početku svake poruke dodan neki podatak ovisan o korisniku. Npr.

$$c_i = (i \cdot 2^h + m)^e \bmod n_i, \;\; i = 1, \ldots, k.$$

Dakle, imamo k polinoma $g_i(x) = (i \cdot 2^h + x)^e - c_i$, te tražimo m sa svojstvom da je

$$g_i(m) \equiv 0 \pmod{n_i}.$$

Neka je $n = n_1 n_2 \cdots n_k$. Pomoću Kineskog teorema o ostacima možemo naći t_i tako da je

$$g(x) = \sum_{i=1}^{k} t_i g_i(x) \;\text{ i }\; g(m) \equiv 0 \pmod{n}$$

($t_i \equiv 1 \pmod{n_i}$, $t_i \equiv 0 \pmod{n_j}$ za $j \neq i$). Polinom g je normiran i stupnja e. Ako je $k > e$, tj. imamo više korisnika (presretnutih šifrata) nego što je javni eksponent, onda je $m < \min_i n_i < n^{1/k} < n^{1/e}$, pa se m može efikasno naći primjenom gore navedenog Coppersmithovog rezultata.

Može se preporučiti uporaba eksponenta

$$e = 65537,$$

koji je dovoljno velik da bi onemogućio sve poznate napade na RSA s malim eksponentom, a prednost mu je vrlo brzo šifriranje jer ima malo jedinica u binarnom zapisu. Naime, $65537 = 2^{16} + 1$.

Možemo zaključiti da i nakon tri desetljeća intenzivnog proučavanja, još uvijek nije pronađena metoda koja bi razbila RSA kriptosustav.

Svi poznati napadi na RSA zapravo samo pokazuju na što treba paziti i što treba izbjegavati kod izbora parametara i implementacije RSA.

Zasad se, uz korektnu implementaciju, RSA može smatrati sigurnim kriptosustavom.