Univerzitet u Banjoj Luci

Prirodno-matematički fakultet

Studijski program: Matematika i informatika

Smjer: Informatika

SEMINARSKI RAD

Predmet: Informacione tehnologije i društvo

Tema: RSA kriptosistem

Profesor:

vanr. prof. dr Dragan Matić

Student:

Teodora Milanović, 22/19

Sadržaj

1. Uvod **1**
1.1 Osnovni pojmovi....1
1.2 Kriptosistem javnog ključa....4

2. RSA kriptosistem **6**
2.1 Opis algoritma 6
2.2 Definicija RSA kriptosistema....7
2.3 Implementacija algoritma....9
2.4 Efikasnost RSA kriptosistema....12

3. Kriptoanaliza RSA kriptosistema **13**
3.1 Faktorizacija 13
3.2 Mali tajni eksponent....15
3.3 Mali javni eksponent....18
3.4 Ciklični napadi....19

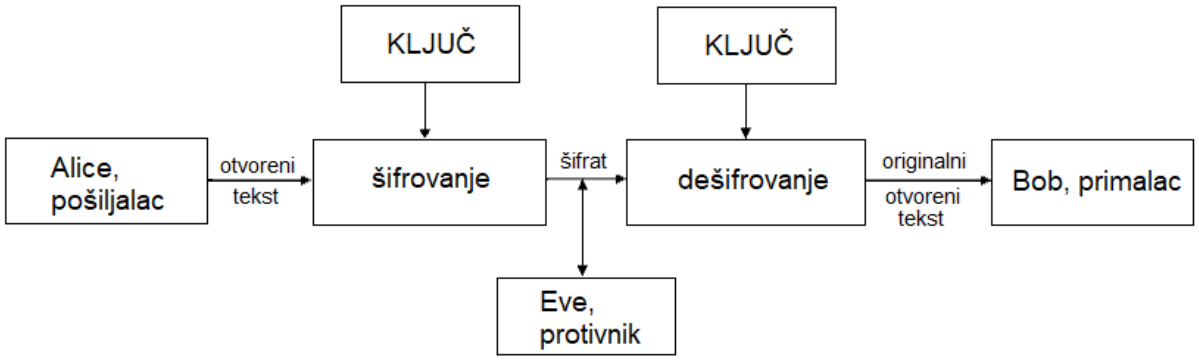
4. Sigurnost RSA kriptosistema **21**

Literatura **22**

1. Uvod

1.1 Osnovni pojmovi

Kriptografija ili **šifrovanje** je nauka koja se bavi metodima očuvanja tajnosti informacija tj. analiziranjem i pronalaženjem metoda za slanje poruka u obliku u kojem ih neće moći pročitati niko osim onih kojima su namijenjene. Riječ dolazi od grčkog pridjeva kriptós (κρυπτός) - "skriven" i glagola gráfo (γράφω) - "pisati".
Kriptografske tehnike koje se koriste da bi se implementirali bezbjednosni servisi su **šifra** tj. algoritam za šifrovanje i dešifrovanje, i **digitalni potpis** tj. skup podataka koji služe za identifikaciju potpisnika i potvrdu autentičnosti poruke.
Osnovni zadatak kriptografije jeste omogućiti dvjema osobama od kojih je jedna pošiljalac, a druga primalac poruke, da komuniciraju preko nesigurnog komunikacionog kanala (telefonska linija, radiotalasi, računarska mreža, itd.) na način da treća osoba (tzv. napadač) ne može razumjeti njihove poruke. U stranoj literaturi su pošiljac i primalac nazvani Alice i Bob, a napadač ili protivnik Eve.
Poruku koju pošiljalac želi poslati primaocu zovemo **otvoreni tekst** (engl. plaintext). To može biti tekst na njihovom maternjem ili nekom drugom jeziku, numerički podaci ili bilo šta drugo. Ključ šifrovanja i otvoreni tekst su ulazni parametri u šifarski sistem. Pošiljalac transformiše otvoreni tekst, koristeći unaprijed dogovoreni ključ. Taj proces se zove **šifrovanje** (kriptovanje), a dobijeni rezultat **šifrat** (šifrovana poruka, kriptogram). Dužina ključa ili broj simbola koji predstavljaju ključ zavisi od šifarskog sistema i predstavlja jedan od parametara sigurnosti tog sistema. Obrnut proces, **dešifrovanje**, rekonstruiše otvoreni tekst na osnovu šifrata. U ovom slučaju se koristi ključ dešifrovanja, a argumenti funkcije za dešifrovanje su ključ i šifrat.



Slika 1.1 Klasična kriptografija

U kriptografiji, **kriptosistem** je skup kriptografskih algoritama potrebnih za implementaciju određene sigurnosne usluge, najčešće radi postizanja povjerljivosti (šifrovanja). Kriptosistem se obično sastoji od tri algoritma: jednog za stvaranje ključeva, jednog za šifrovanje i jednog za dešifrovanje. Pošto se izraz šifra koristi za označavanje algoritama za šifrovanje i dešifrovanje, pojam kriptosistem se najčešće koristi kada je algoritam generisanja ključeva važan. Iz tog razloga, termin kriptosistem se obično koristi za označavanje tehnika javnih ključeva; međutim, i „šifra“ i „kriptosistem“ koriste se za simetrične tehnike.

Definicija 1.1

Matematički se kriptosistem može definisati kao uređena petorka $(\mathcal{P},C,\mathcal{K},\mathcal{E},\mathcal{D})$ za koju vrijedi:

1. $\mathcal{P}$ je konačan skup svih mogućih otvorenih tekstova.
2. C je konačan skup svih mogućih šifrata.
3. $\mathcal{K}$ je konačan skup svih mogućih ključeva.
4. Za svaki ključ k $\in \mathcal{K}$ postoji algoritam šifrovanja $\mathcal{E}_k \in \mathcal{E}$, i odgovarajući algoritam dešifrovanja $\mathcal{D}_k \in \mathcal{D}$, gdje su $\mathcal{E}_k$: $\mathcal{P} \rightarrow C$ i $\mathcal{D}_k$: $C \rightarrow \mathcal{P}$ funkcije sa svojstvom da za svaki e $\in \mathcal{K}$ postoji d $\in \mathcal{K}$ tako da $\mathcal{D}_d$ ($\mathcal{E}_e$ (p)) = p za svaki p $\in \mathcal{P}$.

U odnosu na tajnost ključa, kriptosistemi se dijele na:

1. **Simetrične kriptosisteme** kod kojih se ključ za dešifrovanje može izračunati poznavajući ključ za šifrovanje i obrnuto. Najčešće su ovi ključevi identični. Sigurnost im leži u tajnosti ključa. Zbog toga se oni i zovu kriptosistemi s tajnim ključem.
2. **Asimetrične kriptosisteme** kod kojih se ključ za dešifrovanje ne može (barem ne u nekom razumnom vremenu) odrediti iz ključa za šifrovanje. Ovdje je ključ za šifrovanje javni ključ. Tačnije rečeno, bilo ko može šifrovati poruku pomoću njega, ali samo osoba koja poznaje odgovarajući ključ za dešifrovanje (privatni ili tajni ključ) može dešifrovati tu poruku. Ovi sistemi se zovu i kriptosistemi javnog ključa.

Za razliku od kriptografije, **kriptoanaliza** (od grčkog kryptós-skriveno i analýein-razmrsiti) je nauka koja se bavi razbijanjem šifri, odnosno otkrivanjem sadržaja otvorenog teksta na osnovu šifrata, a bez poznavanja ključa. Različite tehnike kriptoanalize nazivaju se napadi.

Razlikujemo četiri osnovna nivoa kriptoanalitičkih napada:

1. Napad poznatim šifratom: Kriptoanalitičar posjeduje samo šifrat. Zadatak mu je otkriti otvoreni tekst ili u najboljem slučaju ključ kojim je poruka šifrovana.
2. Napad poznatim otvorenim tekstom: Kriptoanalitičar posjeduje i šifrat i otvoreni tekst, a zadatak mu je otkriti ključ ili algoritam kojim je poruka šifrovana.
3. Napad odabranim otvorenim tekstom: Kriptoanalitičar može odabrati tekst koji će biti šifrovan te dobiti njegov šifrat. Ovaj je napad jači od prethodnog ali je manje realističan.
4. Napad odabranim šifratom: Kriptoanalitičar ima pristup alatu za dešifrovanje, pa može odabrati šifrat i dobiti odgovarajući otvoreni tekst. Ovaj napad je klasičan kod kriptosistema s javnim ključem. Tu je zadatak kriptoanalitičara otkriti ključ za dešifrovanje (tajni ključ).

1.2 Kriptosistem javnog ključa

Glavni problem u simetričnom šifrovanju je rukovođenje tajnim šiframa, ključevima koji su korišteni za šifrovanje i dešifrovanje poruke. Zbog toga su **Whitfield Diffie** i **Martin Hellman**. 1976. godine iznijeli ideju o kriptosistemu koji je potpuno drugačijeg tipa nego simetrični kriptosistem. Nazvali su ga kriptosistem javnog ključa. Ideja se sastojala u tome da se koriste funkcije za šifrovanje iz kojih je praktički nemoguće, u nekom razumnom vremenu, izračunati funkciju za dešifrovanje. Za razliku od simetrične kriptografije, koriste se dva ključa, javni i privatni. Stvaranje javnog i privatnog ključa je povezano matematičkim postupkom, nakon čega se dobijeni tekst javnog ključa može slobodno podijeliti, ali se njime skriven tekst može otključati samo privatnim ključem. Prednost ovog načina šifrovanja je u tome što ne mora da se brine u slučaju da neko presretne javni ključ, jer pomoću njega može samo da šifruje podatke.

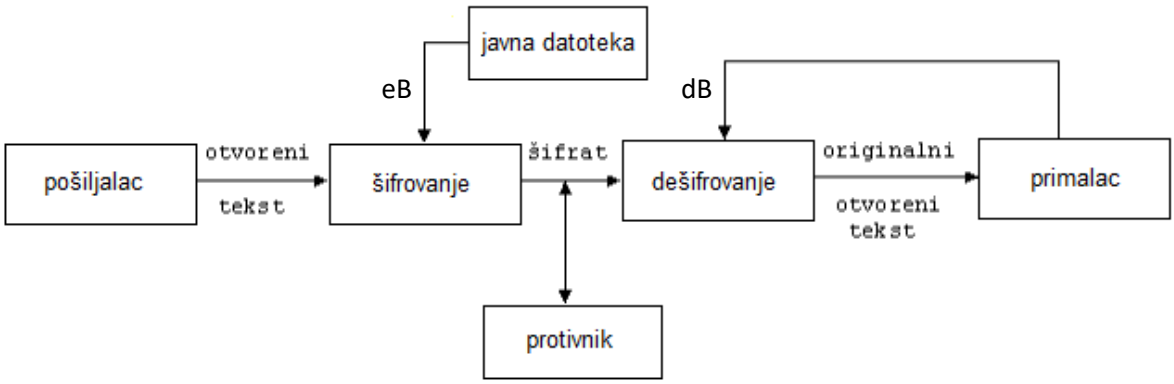
U svrhu realizacije ideje kriptosistema s javnim ključem, koriste se lične jednosmjerne funkcije. Za funkciju f: X → Y kažemo da je jednosmjerna funkcija (one-way function), ako je f (x) lako izračunati za svaki x ∈ X, ali je f^{-1} (y) jako teško izračunati. Ukoliko nam je poznat neki dodatni podatak (trapdoor - tajni ulaz), te je f^{-1} lako izračunati, onda za funkciju f kažemo da je lična jednosmjerna funkcija (trapdoor one-way function).

Kriptosistem s javnim ključem sastoji se od dva skupa funkcija, e_k (šifrovanje) i d_k (dešifrovanje) gdje K prolazi skupom svih mogucih korisnika za koje vrijedi:

- za svaki K je d_k inverzno od e_k ,
- e_k je javan, ali d_k je poznat samo osobi K,
- e_k je lična jednosmjerna funkcija.

Ako pošiljalac A želi poslati poruku x primaocu B, onda B najprije pošalje A svoj javni ključ eB. Potom A šifrira svoju poruku pomoću eB i pošalje primaocu šifrat $y = eB(x)$. Konačno, B dešifrira šifrat koristeći svoj tajni ključ dB i dobija $dB(y) = dB(eB(x)) = x$.
Ukoliko se komunikacija odvija izmedu više ljudi tada svi korisnici stave svoje javne ključeve u neku datoteku koja je dostupna svima i koja se formira u obliku telefonskog imenika. Datoteka mora biti takva da se javni ključevi ne mogu mijenjati. Ako A želi poslati poruku B, dovoljno je da iz datoteke pročita njegov javni ključ eB i izvrši šifrovanje. Pošto svako može pristupiti funkciji eB, tada se svako može i predstaviti kao osoba A. Zato dolazi do pitanja vjerodostojnosti ili autentičnosti poruke koje se može riješiti pomoću potpisa. Neka je P potpis od A koji može da uključuje bilo koji lični podatak. Potrebno je da na početku ili kraju poruke koja se šalje dopisati eBdA (P). Primljeni šifrat, koji se sastoji od poruke i dijela eBdA (P), B dešifruje pomoću dB i

dobija tekst poruke i nerazumljivi dio koji je dA (P). Kako B zna da bi ta poruka trebalo da potiče od A, on koristi njen javni ključ eA i dešifruje dA (P), te dobija P. Zbog nepoznavanja dA, niko drugi, osim A, nije mogao poruku potpisati na taj način.



Slika 1.2 Shema kriptosistema javnog ključa

Glavne prednosti kriptosistema s javnim ključem u poređenju sa simetričnim su:

- nije potreban siguran komunikacioni kanal za razmjenu ključeva,
- dugovječnost: kod asimetričnih kriptosistema par ključeva može se koristiti duže vrijeme bez promjene, čak godinama, dok se kod simetričnih kriptosistema ključevi moraju mijenjati pri svakoj upotrebi,
- manji broj ključeva: za komunikaciju grupe od n ljudi potrebno je 2n ključeva jer su svakom od korisnika potreba samo dva ključa, javni i tajni, za komunikaciju s preostalih n – 1 ljudi,
- mogućnost potpisa poruke,
- nepobitnost: pošiljalac ne može poreći da je poslao poruku. Kod asimetričnih kriptosistema nepobitnost se osigurava digitalnim potpisom, dok je kod simetričnih kriptosistema potrebna neka treća osoba od povjerenja.

Glavni nedostaci kriptosistema s javnim ključem:

- algoritmi s javnim ključem su znatno sporiji od modernih simetričnih algoritama (čak 1000 puta),
- ključevi asimetričnih kriptosistema znatno su veći nego oni kod simetričnih,
- slabi su na napad “odabrani otvoreni tekst”: ako je y = e(x), gdje otvoreni tekst može poprimiti jednu od n vrijednosti, tada je šifrovanjem svih n mogućih otvorenih tekstova i poređenjem s y moguće otkriti x. Pritom tajni ključ d neće biti otkriven. Taj napad moguće je izvršiti jedino ako je n mali.

2. RSA kriptosistem

2.1 Opis algoritma

RSA je algoritam za asimetričnu kriptografiju, prvenstveno namijenjen šifrovanju podataka ali se danas koristi i u sistemima elektronskog potpisa. RSA danas predstavlja industrijski standard u oblasti asimetične kriptografije i zaštiti podataka. Nastao je 1977. na MIT univerzitetu. Tvorci ovog algoritma su **Ronald Rivest**, **Leonard Ejdlman** i **Adi Šamir**, gdje **RSA** predstavlja akronim njihovih prezimena. Algoritam je patentiran od strane MIT-a 1983. u SAD , pod šifrom U.S. Patent 4,405,829. Patentna prava su istekla 21. septembra 2000. Kliford Koks, britanski matematičar koji je radio za britansku obavještajnu agenciju Head Communications Headquarters (GCHQ), objavio je još 1973. godine u internim dokumentima potpuno ekvivalentni sistem za asimetričnu kriptografiju, ali zbog poverljivosti tih dokumenata to je tek objavljeno 1997.

U RSA algoritmu ključnu ulogu imaju veliki prosti brojevi. Sigurnost RSA zasniva se na složenosti faktorizacije velikih brojeva. Smatra se da je određivanje originalne poruke na osnovu šifrata i ključa za šifrovanje ekvivalentno faktorizaciji proizvoda dva velika prosta broja.

U vrijeme kada je elektronska pošta postala razvijena, RSA kriptosistem je implementirao dvije važne ideje:

1. Šifrovanje javnih ključeva - Ova ideja izostavlja potrebu za dostavom ključeva primaocu preko drugog sigurnog kanala prije prenošenja izvorno namijenjene poruke. U RSA kriptosistemu ključevi za šifrovanje su javni, a ključevi za dešifrovanje su privatni tako da samo osoba s ispravnim ključem za dešifrovanje može dešifrovati šifrovane poruke. Svako ima svoje ključeve za šifrovanje i dešifrovanje i oni moraju biti načinjeni tako da se ključ za dešifrovanje ne može lako dobiti iz javnog ključa za šifrovanje.
2. Digitalni potpis - Primalac (npr. Bob) će u nekom trenutku možda trebati potvrditi da je poruku dobio od određene osobe (npr. Alice), a ne od nekog drugog pošiljaoca, te Alice ne može poreći slanje poruke i obrnuto. To se može postići korištenjem ključa za dešifrovanje, a potpis kasnije može potvrditi bilo ko koristeći javni ključ za šifrovanje. Na taj način potpisi se ne mogu krivotvoriti, a pošiljalac ne može poreći slanje poruke.

Ovo nije korisno samo za elektronsku poštu, nego i za ostale elektronske transakcije i prenose, kao na primjer kupovinu i prodaju preko interneta koja je u današnje vrijeme vrlo popularna.

2.2 Definicija RSA kriptosistema

Definicija 2.1

Neka je N = pq produkt dva prosta broja p i q te neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_N$. Definišemo prostor ključeva kao:

$$\mathcal{K} = \{(N, p, q, e, d) : ed \equiv 1(\bmod \phi(N))\}$$

gdje je $\phi(N) = (p-1)(q-1)$ Eulerova funkcija. Eulerova funkcija je funkcija koja prirodnom broju n pridružuje broj $\varphi(n)$ koji predstavlja broj elemenata u nizu 1,2,...,n koji su relativno prosti s n.

Za svaki ključ $K \in \mathcal{K}$, K = (N, p, q, e, d), funkcija šifrovanja $e_K : \mathbb{Z}_N \rightarrow \mathbb{Z}_N$ definisana je s:

$$e_K(x) = x^e \bmod N,$$

dok je funkcija dešifrovanja $d_K : \mathbb{Z}_N \rightarrow \mathbb{Z}_N$ definisana kao:

$$d_K(y) = y^d \bmod N,$$

za bilo koje $x, y \in \mathbb{Z}_N$. Par (N, e) je javni RSA ključ, a trojka (d, p, q) je privatni (tajni) RSA ključ.

Funkcija šifrovanja $e_K(x) = x^e \bmod N$, gdje je faktorizacija broja N nepoznata, te $(e, \phi(N)) = 1$ zove se RSA funkcija. Produkt N = pq zove se RSA modul (ili samo modul). Prosti brojevi p i q nazivaju se RSA prosti brojevi, e se naziva javni eksponent, a d se naziva privatni eksponent. Kako privatni i javni eksponent treba da zadovoljavaju kongruenciju

$$ed \equiv 1(\bmod \phi(N)),$$

slijedi nam

$$ed = 1 + k\phi(N), \qquad (1)$$

za neki cijeli broj k. Jednakost (1) naziva se jednačina RSA ključa ili jednostavnije jednačina ključa. Tačnost algoritma dešifrovanja za elemente otvorenog teksta koji su relativno prosti s modulom proizlazi iz Eulerovog teorema.

Teorema 2.1 (Euler)
Neka je a cijeli broj te n prirodan broj. Ako su brojevi a i n relativno prosti, tada je

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

Provjerimo sada jesu li funkcije eK i dK jedna drugoj inverzne.

$$d_K(e_K(x)) \equiv (e_K(x))^d \equiv (x^e)^d \equiv x^{de} \pmod{n}$$

Kako je $de \equiv 1 \pmod{\phi(n)}$, to znači da postoji prirodan broj c takav da je $de = c \cdot \phi(n) + 1$ pa imamo:

$$x^{de} = x^{c \cdot \phi(n)+1} = x^{c \cdot \phi(n)} \cdot x = [x^{\phi(n)}]^c \cdot x.$$

U zavisnosti od n i x razlikujemo dva slučaja:

1. **(x,n) = 1**
   Kako je tada, prema Eulerovom teoremu, $x^{\phi(n)} \equiv 1 \pmod{n}$, to je
   $$x^{de} \equiv 1^c \cdot x \equiv x \pmod{n}.$$
2. **(x,n) ≠ 1**
   Ako je (x,n) = n, tada je x = 0 pa je kongruencija trivijalno zadovoljena. Neka je (x,n) = p ili (x,n) = q. Bez smanjenja univerzalnosti možemo pretpostaviti da je (x,n) = p, pa je $x^{de} \equiv 0 \equiv x \pmod{p}$. Kako je (x,pq) = p, gdje su p i q prosti, to je (x,q)=1 pa je prema Eulerovom teoremu
   $$x^{\phi(q)} \equiv 1 \pmod{q} \Rightarrow x^{q-1} \equiv 1 \pmod{q}$$
   $\phi: \mathbb{N} \rightarrow \mathbb{N}$ je Eulerova funkcija. Ako je q prost broj tada se $\phi(q)$ računa kao $\phi(q) = q-1$. Sada je:
   $$x^{de} = (x^{(q-1)})^{(p-1)c} \cdot x \equiv x \pmod{q}.$$
   Konačno je $x^{de} \equiv x \pmod{pq}$, tj. $x^{de} \equiv x \pmod{n}$.

2.3 Implementacija algoritma

RSA algoritam uključuje četiri koraka:

1. Generisanje ključeva
2. Distribucija ključeva
3. Šifrovanje poruke
4. Dešifrovanje poruke

Generisanje ključeva

1. Generisaćemo slučajno dva velika prosta broja p i q.
    - Iz sigurnosnih razloga, cijeli brojevi p i q treba da budu odabrani nasumično i treba da budu slični po veličini, ali da se razlikuju u dužini za nekoliko cifara kako bi faktorizacija bila teža. P i q se čuvaju u tajnosti.
2. Izračunati $n = pq$
    - n se koristi kao modul i za javne i za privatne ključeve. Njegova dužina, obično izražena u bitovima, je dužina ključa. Takođe, n je dio javnog ključa.
3. i Ojlerovu funkciju $\phi(n) = (p-1)(q-1)$.
    - $\phi(n)$ se čuva u tajnosti.
4. Odabere se cjelobrojna vrijednost e pri čemu $1 < e < \phi(n)$ takav da je $(e,\phi(n))=1$.
    - Poželjno je odabrati što manji e kako bi se šifrovanje $x^e \bmod n$ odvijalo brže. Kako broj operacija u šifrovanju zavisi od veličine broja e i broja jedinica u binarnom zapisu od e, jedan od čestih izbora jeste e = 3. Ali, s obzirom da upravo izbor malog broja e narušava sigurnost kriptosistema, najbolji izbor je broj $e = 216 + 1 = 65537$. Taj broj je pogodan jer je prost te dovoljno velik za izbjegavanje napada malim eksponentom. Bilo koji veći broj zahtijevao bi barem još jednu operaciju množenja ili dijeljenja, čime bi se proces računanja dodatno odužio. Takođe, e je dio javnog ključa.
5. Upotrebom Euklidovog algoritma izračunati jedinstveni cijeli broj d, $1 < d < \phi(n)$ takav da je $de \equiv 1 \pmod{\phi(n)}$.
    - d se čuva u tajnosti.
    - multiplikativni inverz d od e se može odrediti iz Euklidovog algoritma, jer ima svojstvo da postoji neki cijeli broj c za koji je $e \cdot d + c \cdot \phi(n) = 1$.
6. Javni ključ je par (n,e), a tajni ključ je d.

Distribucija ključeva

Pretpostavimo da Bob želi da pošalje informacije Alice. Ako odluče koristiti RSA, Bob mora znati Alicein javni ključ za šifrovanje poruke, a Alice mora koristiti njen privatni ključ za dešifrovanje poruke. Da bi Bob mogao poslati svoje šifrovane poruke, Alice prenosi svoj javni ključ (n, e) Bobu putem pouzdane, ali ne nužno tajne rute. Alicein privatni ključ (d) nikada se ne distribuira.

Šifrovanje poruke

Da bi osoba A šifrovala poruku x osobi B mora da:

1. Pristupi javnom ključu (n,e) osobe B.
2. Izračuna $e_K(x)= x^e \pmod n$.
3. Pošalje šifrat $e_K(x)$ osobi B.

Dešifrovanje poruke

Kako bi dobila otvoreni tekst x iz eK(x), osoba B treba da:

1. Upotrijebi tajni ključ d za dobijanje $x = (e_K(x))^d \pmod n$.

Primjer 2.1

Generisanje ključeva:

1. Osoba B odabere p = 11 i q = 17.
2. Izračuna $n = pq = 11 \cdot 17 = 187 \rightarrow$ **n = 187**
3. i $\phi(n) = (p-1)(q-1) = 10 \cdot 16 = 160 \rightarrow$ **ϕ(n) = 160**
4. Zatim odabere **e = 3** koji je relativno prost s 160.
5. Pomoću proširenog Euklidovog algoritma računa d tako da je $de \equiv 1 \pmod{\phi(n)}$, tj. $d \cdot 3 \equiv 1 \pmod{160}$, slijedi da je **d = 107**.

B-ov javni ključ je (n = 187, e = 3), dok je tajni ključ d =107.

Šifrovanje poruke:

1. Za šifrovanje otvorenog teksta x=72 A računa šifrat:

$$e_K(x)= x^e \pmod n = 72^3 \pmod{187} = 183,$$

te ga potom šalje osobi B.

Dešifrovanje poruke:

1. Za dešifrovanje $e_K(x)$, B računa:

$$(e_K(x))^d \pmod n = 183^{107} \pmod{187} = 72.$$

Primjer 2.2

Pokažimo kako bi izgledalo da Alice želi poslati Bobu poruku INFORMATIKA.

- Bob bira proste brojeve p = 17 i q = 53, odakle dobija n = pq = **901**, te $\phi(n) = (p-1)(q-1) =$ **832**. Zatim bira enkripcioni eksponent **e = 11** i računa dekripcioni eksponent d takav da je $de \equiv 1 (\mathrm{mod}\ \phi(n))$. Tako dobija **d = 227**.
- Bob u javnoj bazi ključeva ostavlja svoj javni ključ (n, e) = (901, 11), a p, q, d i $\phi(n)$ čuva u tajnosti.
- Alice uzima Bobov javni ključ kako bi šifrovala poruku INFORMATIKA. Numerički ekvivalent otvorenog teksta dobija se tako da se slovima A,. . . ,Z redom pridruže brojevi 01,. . . ,26, s tim da razmak označimo s 00. Numerički ekvivalent prethodne poruke je **x = 09140615181301200911 01**. Budući da je x > n prije šifrovanja x se dijeli u blokove od po 2 cifre. Sada je x = (x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11) = (09, 14, 06, 15, 18, 13, 01, 20, 09, 11, 01).
- Uz pomoć Bobovog javnog ključa (n, e) Alice računa $y_i = x_i^e \bmod n$ za sve i = 1, ..., 11.

$y_1 = 09^{11} \bmod 901 = 865$
$y_2 = 14^{11} \bmod 901 = 724$
$y_3 = 06^{11} \bmod 901 = 396$
$y_4 = 15^{11} \bmod 901 = 791$
$y_5 = 18^{11} \bmod 901 = 154$
$y_6 = 13^{11} \bmod 901 = 599$
$y_6 = 01^{11} \bmod 901 = 1$
$y_7 = 20^{11} \bmod 901 = 330$
$y_8 = 09^{11} \bmod 901 = 865$
$y_9 = 11^{11} \bmod 901 = 590$
$y_{10} = 01^{11} \bmod 901 = 1$

Dobijeni šifrat **y = 865724396791154599001330865590001** Alice dalje šalje Bobu.

- Dijeleći y na blokove, Bob na sličan način pomoću dekripcionog eksponenta d = 227 računa $x_i = y_i^{227} \bmod 901$, i=1, ..., 11:

$x_1 = 865^{227} \bmod 901 = 09$
$x_2 = 724^{227} \bmod 901 = 14$
$x_3 = 396^{227} \bmod 901 = 06$
$x_4 = 791^{227} \bmod 901 = 15$
$x_5 = 154^{227} \bmod 901 = 18$
$x_6 = 599^{227} \bmod 901 = 13$
$x_6 = 1^{227} \bmod 901 = 1$
$x_7 = 330^{227} \bmod 901 = 20$
$x_8 = 865^{227} \bmod 901 = 9$
$x_9 = 590^{227} \bmod 901 = 11$
$x_{10} = 1^{227} \bmod 901 = 1$

i dobija otvoreni tekst INFORMATIKA.

2.4 Efikasnost RSA kriptosistema

Prosti brojevi koji se koriste u ovom algoritmu uglavnom sadrže nekoliko stotina cifara i zbog toga se ovdje javlja više problema praktične prirode. Da bi se pomnožili toliko veliki brojevi, moraju se koristiti posebni algoritmi za množenje. Sem toga lako se da primijetiti da je za takve operacije potrebno više vremena, pa su tako ovi algoritmi šifrovanja mnogo sporiji u odnosu na simetrične algoritme. DES algoritam šifrovanja je oko 100 do 1000 puta brži u odnosu na RSA algoritam.
Posebno se može posmatrati trajanje algoritma za generisanje prostih brojeva i za modularno potenciranje, koje su dominantne operacije za algoritam generisanja ključa i za algoritme šifrovanja i dešifrovanja.

Modularno potenciranje
Šifrovanje i dešifrovanje u RSA kriptosistemu sastoji se od modularnog potenciranja. Te operacije mogu biti vrlo skupe kada su eksponent i modul veliki. Postoji mnogo različitih algoritama, ali složenost se može smanjiti tj. šifrat $e_K(x) = x^e \pmod n$ se može efikasno izračunati pomoću algoritma **„kvadriraj i množi”:**

1. Prikažemo e u bazi 2: $e = e_0 + 2e_1 + \cdots + 2^{s-1} e_{s-1}$.
2. Primijenimo sljedeći algoritam:

```
y=1
za i = s−1,...,1,0
        y = y^2 (mod n)
        ako je e_i = 1, onda y = y · x (mod n)
```

Iz toga je vidljivo da je ukupan broj množenja manji ili jednak od 2s, pa je ukupan broj operacija $O(\log e \cdot \log^2 n)$. To znači da je taj algoritam polinomijalan.

Izbor prostih brojeva
Algoritam za generisanje ključa treba generisati dva nasumična prosta broja koji su otprilike iste veličine. Može se generisati n − bitni slučajno odabrani prosti broj u očekivanom vremenu $O(n^4/ \log(n) + tn^3)$. Za velike vrijednosti modula to može biti vrlo skup posao, pogotovo ako postoji mnogo prostih brojeva koje treba generisati. Postoje mnogi algoritmi za generisanje prostih brojeva, npr. Pollardov Rho algoritam, Faktorizacija razlikom kvadrata, Gordonov algoritam za generisanje prostih brojeva, itd.

3. Kriptoanaliza RSA kriptosistema

Postoji mnogo različitih tipova napada na RSA. Neki od njih uključuju: napad bug-ovima, tajmirane napade, napade na napajanje, itd. Druge vrste napada usmjerene su na ljudsku komponentu sigurnosti. Socijalni inženjerski napadi mogu se koristiti za iskorištavanje ljudskog ponašanja. Neke su informacije izdvojene od korisnika pomoću određenih vrsta manipulacija. Na primjer, lozinka koja osigurava sigurnost RSA privatnog ključa može se saznati zvanjem osobe usred noći sa zabrinutim glasom i traženjem lozinke zbog nekog hitnog slučaja na poslu. Neki napadi temelje se na matematičkoj strukturi RSA kriptosistema (modulu, jednačini ključa) i korištenju određenih izbora parametara (korištenje malog javnog ili privatnog eksponenta).

3.1 Faktorizacija

Očigledan napad na RSA je faktorizacija od n na način n = pq, gdje su p i q prosti brojevi. Ako napadač faktorizuje n tada može jednostavno otkriti

$$\varphi(n) = (p-1)(q-1),$$

te pomoću Euklidovog algoritma odrediti tajni eksponent d iz

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

Rastavljanje brojeva na proste faktore puno je teže od provjeravanja je li broj prost ili složen. Međutim, postoje mnogi algoritmi za rastavljanje broja na proste faktore. Iako se oni stalno poboljšavaju, još uvijek su daleko od prijetnje sigurnosti RSA kriptosistema, pod uslovom da se RSA pravilno koristi. Sljedeću tabelu autori RSA kriptosistema predstavili su 1978. godine. Pretpostavili su da operaciji u Schroeppelovom algoritmu faktorizovanja treba jedna mikrosekunda za određivanje prostih faktora broja n. Tako su predstavili sljedeću tabelu za različite dužine broja n:

Broj cifara	**Broj operacija**	**Trajanje**
50	$1.4 \cdot 10^{10}$	3.9 sati
75	$9.0 \cdot 10^{12}$	104 dana
100	$2.3 \cdot 10^{15}$	74 godine
200	$1.2 \cdot 10^{23}$	$3.8 \cdot 10^{9}$ godina
300	$1.5 \cdot 10^{29}$	$4.2 \cdot 10^{25}$ godina
500	$1.3 \cdot 10^{39}$	$1.2 \cdot 10^{23}$ godina

Zato se preporučuje da u RSA tajno izabrani parametri p i q budu veliki prosti brojevi od barem 100 cifara. Oni se biraju tako da se prvo generiše slučajan prirodan broj m s traženim brojem cifara, nakon čega se traži prvi prosti broj veći ili jednak od m koristeći pritom neki test prostosti. Tada n = pq ima oko 200 cifara, a kako trenutno najbrži algoritam za faktorizaciju treba

$$\exp(O((\log n)^{1/3}(\log \log n)^{2/3}))$$

operacija, brojevi od preko 200 cifara sigurni su od ovog napada. To jest, nije poznat niti jedan polinomijalni algoritam za faktorizaciju.

RSA kriptosistem dopušta korisniku da sam izabere dužinu ključa, a samim time i nivo sigurnosti. Postoje slučajevi u kojima se n = pq može lako faktorizovati pa bi prema tome brojevi p ± 1 i q ± 1 trebalo da imaju barem jedan veliki prosti faktor. Osim toga, p i q ne smiju biti vrlo blizu jedan drugome jer ih se u tome slučaju može odrediti tako što se posmatraju brojevi koji su približno jednaki $\sqrt{n}$.

RSA problem

Definicija 3.1.
Neka je dat pozitivan cijeli broj n koji je produkt dva prosta broja p i q, pozitivan cijeli broj e takav da je $(e,(p-1)(q-1))=1$ i cijeli broj $e_K(x)$. RSA problem je pronaći cijeli broj x takav da je $e_K(x)= x^e \pmod n$.

RSA problem se sastoji od određivanja poruke x ako nam je poznat javni ključ (n, e), te šifrat $y = x^e \bmod n$. Odnosno, to je problem računanja e-tog korijena modul n ili problem određivanja inverza RSA funkcije. Uslovi koji su postavljeni pri izboru parametara n i e osiguravaju da za svaki cijeli broj $e_K(x) \in \{1,2,...,n-1\}$ postoji tačno jedan broj $x \in \{1,2,...,n-1\}$ takav da je $e_K(x) \equiv x^e \pmod n$. Takođe, funkcija e_K: $\mathbb{Z}_N \rightarrow \mathbb{Z}_N$ je permutacija. Pošto je p−1 paran broj, odabrani e mora biti neparan broj. Na temelju toga je $e \neq 2$ pa slijedi da problem pronalaženja drugog korijena nije specijalan slučaj RSA problema. Ako su faktori od n poznati onda se RSA problem može lako riješiti. Zato je važno da n bude otporan na metode faktorizacije (da p i q nisu približnih vrijednosti, te da p−1 i q−1 imaju bar jedan veliki prosti faktor). Takođe, problem je teško rješiv kada je otvoreni tekst $x \in \mathbb{Z}_n$ slučajno izabran, te modul n dovoljno velik broj pri čemu su p i q takođe slučajno izabrani veliki brojevi takvi da je n = pq. Taj uslov nazivamo RSA pretpostavkom koju smatramo istinitom jer nije dokazano da je neistinita otkad je RSA stvoren.

Iako je lakše riješiti RSA problem nego problem faktorizacije cijelog broja, u praksi se pretpostavlja da su ti problemi ekvivalentni.

3.2 Mali tajni eksponent

Osim što je potrebno izbjegavati male brojeve p i q, potrebno je i izbjegavati mali tajni eksponent d. Napad koji služi za razbijanje RSA kriptosistema u slučaju da je izabran mali tajni eksponent d zove se Wienerov napad. Nazvan je po kanadskom kriptologu Michaelu Wieneru. Wiener pokazuje da mali d uzrokuje potpuni prekid kriptosistema. U RSA kriptosistemu, svi tajni eksponenti $d < 2^l$, gdje l zavisi od trenutnog najsavremenijeg računanja, mogu biti jednostavno pogođeni. Na primjer, trenutno je moguće otkriti sve tajne eksponente $d \leq 2^{60}$, ali ne i za one $d \leq 2^{80}$.

Teorema 3.1 (Wiener)

Neka je n = pq i p < q < 2p te neka je e < ϕ(n) i $d < \frac{1}{3} n^{0.25}$. Tada postoji polinomijalni algoritam koji iz poznavanja n i e računa d.

Verižni razlomak je razlomak oblika

$$x = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \cfrac{1}{a_3 + \cfrac{1}{a_4 + \ddots}}}}$$

koji se kraće zapisuje kao x = [a_0; a_1, a_2 ... a_n], gdje su a_0,a_1,...,a_n koeficijenti iz Euklidovog algoritma primijenjenog na x.

Teorema 3.2 (Legendre)

Neka je $\alpha \in \mathbb{Q}$, te neka su p i q cijeli brojevi za koje vrijedi $q \geq 1$ i $|\alpha - \frac{p}{q}| < \frac{1}{2q^2}$. Tada je $\frac{p}{q}$ neka konvergenta od α.

Konvergentama verižnog razvoja nazivamo razlomke $\frac{p_k}{q_k}$, za $k \leq n$ vrijedi:

$p_0 = a_0$ $\quad$ $q_0 = 1$

$p_1 = a_0a_1 + 1$ $\quad$ $q_1 = a_1$

$p_k = a_kp_{k-1} + p_{k-2}$ $\quad$ $q_k = a_kq_{k-1} + q_{k-2}$.

Dokaz.

Kako je ed = 1 mod ϕ(n), postoji prirodan broj k takav da je ed − kϕ(n) = 1. Odavde je

$$\left|\frac{\boldsymbol{e}}{\phi(n)} - \frac{\boldsymbol{k}}{\boldsymbol{d}}\right| = \frac{\mathbf{1}}{\boldsymbol{d}\phi(n)}$$

To znači da $\frac{\boldsymbol{e}}{\phi(n)}$ aproksimira $\frac{\boldsymbol{k}}{\boldsymbol{d}}$. Kako je ϕ(n) nepoznat, s n ćemo aproksimirati ϕ(n). Kako je ϕ(n) = n−p−q+1 i p+q−1 < $3\sqrt{n}$ slijedi da je |n−ϕ(n)| < $3\sqrt{n}$. Ako ϕ(n) zamijenimo s n dobijamo:

$$\left|\frac{e}{n} - \frac{k}{d}\right| = \left|\frac{ed - k\phi(n) - kn + k\phi(n)}{nd}\right| = \left|\frac{1 - k(n - \phi(n))}{nd}\right| \leq \left|\frac{3k\sqrt{n}}{nd}\right| = \frac{3k}{d\sqrt{n}}$$

Sada je kϕ(n) = ed − 1 < ed. Kako je e < ϕ(n), uočimo da je k < d < $\frac{1}{3}$ $n^{1/4}$, pa dobijamo

$$\left|\frac{e}{n} - \frac{k}{d}\right| \leq \frac{1}{dn^{1/4}} < \frac{1}{2d^2}$$

Prema Legendreovom teoremu, $\frac{\boldsymbol{k}}{\boldsymbol{d}}$ je neka konvergenta razvoja u verižni razlomak od $\frac{\boldsymbol{e}}{\boldsymbol{n}}$. Nakon što izračunamo sve konvergente, testiramo koja od njih zadovoljava $x^{ed} \equiv x(\text{ mod } n)$ za nasumično odabrani x. To daje polinomijalni algoritam za otkrivanje tajnog eksponenta d.

Primjer 3.1

Pretpostavimo da su dati modul n = 7064009, javni eksponent e = 5773091, te da za tajni eksponent d vrijedi d < 1/3 · $n^{0.25}$ < 18. Najprije odredimo razvoj broja

$$\frac{e}{n} = \frac{5773091}{7064000}$$

u verižni razlomak (pomoću Euklidovog algoritma). Tada dobijamo: [0, 1, 4, 2, 8, 2, 5, 38, 1, 1, 2, 4, 2, 3]. Zatim računamo pripadne konvergente: 0, 1, $\frac{4}{5}$, $\frac{9}{11}$, $\frac{76}{93}$, $\frac{161}{197}$, $\frac{881}{1078}$, $\frac{33639}{41161}$, $\frac{34520}{42239}$, $\frac{68159}{83400}$, $\frac{170838}{209039}$, $\frac{751551}{919556}$, $\frac{1673860}{2048151}$, $\frac{5773091}{7064009}$. Kako je d < 18, provjerimo koji od imenioca 5 i 11 zadovoljava kongruenciju $(x^e)^d \equiv x(\text{mod } n)$, za npr. x = 2. Tako dobijamo da je tajni eksponent d = 11.

Budući da je n uglavnom 1024 bitni, da bi se izbjegao ovaj napad d mora biti bar 256 bitni. Wiener, također, opisuje niz tehnika koje omogućavaju brzo dešifrovanje i nisu osjetljive na njegov napad. Neki od njih su:

- <u>veliki e:</u>
  Umjesto da smanjujemo e modul $\phi(n)$, pretpostavimo da koristimo (n, e') za javni ključ, gdje je $e' = e + t\phi(n)$, za neki veliki t. Eksponent e' može se koristiti umjesto eksponenta e za šifrovanje poruke. Jednostavno računanje pokazuje nam da ako je $e' > n^{1.5}$, tada, bez obzira koliko mali d bio, navedeni napad neće se moći ostvariti. Velike vrijednosti eksponenta e dovode do povećanja vremena šifrovanja.

- <u>korištenje kineske teoreme o ostacima:</u>
  Pretpostavimo da smo odabrali eksponent d tako da su $d_p = d \bmod (p - 1)$ i $d_q = d \bmod (q - 1)$ mali, recimo da svaki ima po 128 bita. Tada se brzo dešifrovanje šifrovanog teksta y može izvršiti na sljedeći način: prvo računamo $x_p = y^{d_p} \pmod p$ i $x_q = y^{d_q} \pmod q$. Tada koristimo kinesku teoremu o ostacima da izračunamo jedinstvenu vrijednost $m \in \mathbb{Z}_N$ koja zadovoljava $x = x_p \pmod p$ i $x = x_q \pmod q$. Dobijeni otvoreni x zadovoljava $x = y^d \bmod n$.
  Iako su d_p i d_q mali, vrijednost d mod $\phi(n)$ može biti velika.

3.3 Mali javni eksponent

Korištenje vrlo malog javnog eksponenta e može značajno uštedjeti vrijeme, ali i smanjiti troškove šifrovanja. Najmanji mogući javni eksponent e je 3 koji se dugo smatrao dobrim izborom, ali vremenom se pokazalo da je uopšteno RSA kriptosistem s malim javnim eksponentom nesiguran. Da bi se odbranilo od mogućeg napada preporučena vrijednost za e je $2^{16} + 1$, koji je dovoljno velik da bi onemogućio sve poznate napade na RSA s malim eksponentom, a prednost mu je vrlo brzo šifrovanje jer ima malo jedinica u binarnom zapisu. Kada se koristi vrijednost $2^{16} + 1$ potvrda potpisa zahtijeva 17 množenja, što je puno manje od 1000 koliko je potrebno kada se koristi nasumično izabrani $e \leq \phi(n)$.

Hastadov napad

Pad protokola nastaje kada se ista poruka m šifruje s nekoliko različitih javnih ključeva (e, n_i), $i = 1, ..., l$, $l \in N$ koji imaju isti javni eksponent e i različite module n_i . Napad na ovaj protokol prvi je opisao Hastad 1985. godine.

Teorema 3.3

Neka su $((e, n_1), ...,(e, n_l))$, $l \geq$ e valjani RSA javni ključevi čiji su parovi relativno prosti, neka je $n_0 = \min\{n_1, ..., n_l\}$ i neka je $n = \prod_{i=1}^{l} n_i$. Za bilo koju poruku otvorenog teksta $m < n_0$ datu sa $c_i = m^e \bmod n_i$ i (e, n_i) za $i = 1, ..., l$, otvoreni tekst m može se izračunati u polinomijalnom vremenu log(n).

Dokaz.

Kako su moduli u parovima relativno prosti, korištenjem Kineske teoreme o ostacima možemo izračunati $x \equiv m^e \pmod n$, koristeći za ulaz c_i i n_i $(i = 1, ..., l)$. Kako je $m < n_0$ slijedi da je $m^e < n_1 n_2 ... n_l = n$, pa je $x = m^e$. Računajući e − ti korijen od x dolazimo do otvorenog teksta m. Kako se svi proračuni mogu napraviti u polinomijalnom vremenu log(n) teorem je dokazan.

Pretpostavimo da imamo tri korisnika s istim javnim eksponentom e = 3, ali s različitim vrijednostima javnog modula n_1, n_2, n_3. Njihov protivnik može saznati sljedeće šifrate:

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

Pomoću kineske teoreme o ostacima, sada može pronaći rješenje sistema linearnih kongruencija

$$x \equiv c_1 \pmod{n_1},\ x \equiv c_2 \pmod{n_2},\ x \equiv c3 \pmod{n_3},$$

koje je broj x sa svojstvom $x \equiv m^3 \pmod{n1n2n3}$ $\xrightarrow{m^3<n1n2n3}$ $x = m^3$ pa protivnik ukoliko izračuna $\sqrt[3]{x}$ dobija originalnu poruku m.

Primjer 3.2

Tri korisnika koriste različite module $n_1 = 329$, $n_2 = 341$, $n_3 = 377$, dok im je javni eksponent e=3. Prilikom slanja poruke, protivnik je saznao šifrate $c_1 = 43$, $c_2 = 30$, $c_3 = 372$, te želi saznati zajednički otvoreni tekst m.

Rješavanjem sistema linearnih kongruencija

$$x \equiv 43 \pmod{329},\ x \equiv 30 \pmod{341},\ x \equiv 372 \pmod{377},$$

pomoću kineske teoreme o ostacima, dobija se da je

$$\begin{aligned} x &\equiv 341 \cdot 377 \cdot 172 + 329 \cdot 377 \cdot 232 + 329 \cdot 341 \cdot 317 \\ &\equiv 86451373 \\ &\equiv 1860867 \pmod{329 \cdot 341 \cdot 377}. \end{aligned}$$

To znači da je x = 1860867 i $m = \sqrt[3]{x} = 123$.

3.4 Ciklični napadi

Simmons i Norris su 1977. godine primijetili da se otvoreni tekst uvijek može dobiti ponovljenim šifrovanjem njegovog šifrata, sve dok se ne vrati na sebe, tj. ciklusi se vraćaju na izvorni šifrovani tekst.

Neka je $e_K(x) = x^e \pmod{n}$ šifrovani tekst. Neka je k pozitivan cijeli broj takav da je $e_k(x)^{e^k} \equiv e_K(x) \pmod{n}$. Pošto je šifrovanje permutacija na skupu $\{0,1,\ldots,n-1\}$ takav cijeli broj k mora postojati. Iz istog razloga mora postojati slučaj $e_k(x)^{e^{k-1}} \equiv x \pmod{n}$. Ovo razmatranje predvodi sljedeći ciklični napad na RSA kriptosistem. Protivnik računa $e_k(x)^e \pmod{n}$, $e_k(x)^{e^2} \pmod{n}$, $e_k(x)^{e^3} \pmod{n}$,... dok ne dobije $e_K(x)$ prvi put. Ako je $e_k(x)^{e^k} \pmod{n} = e_K(x)$ onda je prethodni broj u ovom krugu $e_k(x)^{e^{k-1}} \pmod{n}$ jednak otvorenom tekstu x.

Primjer 3.3

Pretpostavimo da osoba A i B ne vode računa o sigurnosti tokom određivanja njihovih ključeva. Neka A šifruje poruku m = 5 pomoću B-ovog javnog ključa (3, 51) i tako dobija c = 23. Osoba C presreće poruku od A i pokušava je otkriti pomoću cikličnog napada.

$$e(23) = 23^3 \bmod 51 = 29$$
$$= 29^3 \bmod 51 = 11$$
$$= 11^3 \bmod 51 = 5$$
$$= 5^3 \bmod 51 = 23.$$

Kada nakon cikličnog šifrovanja C otkrije izvornu vrijednost (c = 23) i izvrši prekid šifrovanja, vratiće se jedan korak nazad u računanju kako bi otkrio dešifrovanu poruku tj. x=5. Ono što je C šifrovao na vrijednost 23, mora biti jednako onome što je A šifrovala na vrijednost 23.
Kod za šifrovanje je otkriven i C može čitati poruke između A i B.

Pod uslovom da imamo dovoljno vremena, ciklični napad uvijek će moći otkriti poruke koje su šifrovane RSA algoritmom. Međutim, potrebno je puno vremena kako bi se pomoću cikličnog napada otkrila šifrovana poruka. Tako RSA kriptosistem sigurnim od cikličnog napada čine:

1. "jaki" prosti brojevi
   - p je jak prost broj ako brojevi $p - 1$ i $p + 1$ imaju velike proste faktore u i v.
   - $u - 1$, $u + 1$, $v - 1$ i $v + 1$ takođe treba da imaju velike proste faktore.

   ,ovako se povećava broj ciklusa potrebnih za prekid šifrovanja.
2. veliki prosti brojevi
   - ključevi veći od 60 bitova neće se moći razbiti pomoću cikličnog napada u toku 24 sata.
   - kako RSA trenutno koristi ključeve od 1024 bita ili više, bilo bi potrebno mnogo godina dok se ne razbije kod.

4. Sigurnost RSA kriptosistema

Za sada, RSA kriptosistem možemo smatrati sigurnim kriptosistemom. I nakon tri decenije intenzivnog proučavanja, još uvijek nema metode koja bi razbila RSA kriptosistem. Naglim napretkom današnjih računara javila se potreba za povećanjem sigurnosti, odnosno zaštite podataka. Sigurnost RSA kriptosistema bazira se na teškoći faktorizacije velikih brojeva i na današnjim računarima takav problem nije lako rješiv pa su ti sistemi još uvijek sigurni. Takođe, činjenica da algoritmi za faktorizaciju brojeva postaju svakim danom sve bolji, ali i neumoljiv razvoj kompjutera učinili su da danas 512 bitni RSA algoritam ne bude dovoljan za bezbjedno šifrovanje poruka. Ključeve dužine 512 bitova nije moguće probiti na današnjim ličnim računarima. Postoje složeni računarski sistemi koji mogu probiti 512-bitni RSA u razumnom vremenu, međutim radi se o vrlo skupim i teško dostupnim sistemima. Dužina ključeva za RSA algoritam je uglavnom 1024 bita. Za 1024 bitne algoritme pretpostavlja se da će biti bezbjedni barem još 15-tak godina. Ali za potpunu sigurnost preporučuje se 2048, odnosno 4096 bitova.
Prilikom izbora dužine ključa treba voditi računa o tome koliko nam je zapravo važna sigurnost informacije koju šifrujemo. Ako želimo zaštititi neku običnu poruku od stranca, potpunu sigurnost možemo postići najprimitivnijim tehnikama šifrovanja. Ako ipak želimo zaštiti neke vrlo važne podatke kao kreditne kartice, državne tajne i slično, tada moramo koristiti najbolje i najsigurnije računarske metode. Razvojem novih teoretskih modela računara, koji se pokušavaju i u praksi realizovati, problem faktorizacije će biti lako rješiv. Kao primjer se mogu uzeti kvantni računari. Za razliku od klasičnih računara kod kojih je osnovna jedinica informacije jedan bit, kvantni računari bi koristili ideje iz kvantne mehanike te bi kod njih osnovna jedinica informacije, tzv. qubit, nosila puno više informacija. Klasičnim računarima faktorizacija velikih brojeva je vrlo komplikovan posao, gotovo nemoguć, dok kvantni računar ima potencijal rješavanja tog problema pomoću Shorovog algoritma - polinomijalnog kvantnog algoritma za problem faktorizacije. Kako je time narušena i sigurnost RSA kriptosistema to je motivisalo naučnike za smišljanje novih i jačih načina enkripcije. Međutim, takvi računari još nisu praktično realizovani u obliku koji bi bili konkurencija klasičnim računarima.

Literatura

[1] A.Dujella, M. Maretić, *Kriptografija*, Element, Zagreb, 2007.

[2] D.Boneh, *Twenty Years of Attacks on the RSA Cryptosystem.*

[3] B.Ibrahimpašić, *RSA kriptosustav*, Osječki matematički list 5 (2005), 101-112.

[4] S.Robles, *The RSA Cryptosystem.*

[5] N. Smart, *Cryptography. An Introduction*, McGraw–Hill, New York, 2002.