RSA kriptosustav i njegova kriptoanaliza

Spaić, Ines

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Integrirani sveučilišni nastavnički studij matematike i informatike

Ines Spaić

RSA kriptosustav i njegova kriptoanaliza

Diplomski rad

Osijek, 2017.

Sveučilište J.J. Strossmayera u Osijeku
Odjel za matematiku
Integrirani sveučilišni nastavnički studij matematike i informatike

Ines Spaić

RSA kriptosustav i njegova kriptoanaliza

Diplomski rad

Mentor: doc. dr. sc. Ivan Soldo

Osijek, 2017.

Sadržaj

Uvod

Kriptografija je znanstvena disciplina koja se bavi načinima slanja poruke u tajnosti u šifriranom ili dešifriranom obliku tako da ju samo osobe kojima je ta poruka namijenjena mogu dešifrirati ili šifrirati. Temeljni zadatak kriptografije je omogućiti dvjema osobama, *pošiljaocu* i *primaocu* poruke, komunikaciju preko nesigurnog komunikacijskog kanala na način da neka treća osoba, *protivnik*, koja ima mogućnost nadziranja kanala, ne može razumjeti njihovu poruku. Riječ kriptografija nastala je od pridjeva *kryptos* što znači *skriven* i od riječi *graphein* što znači *pisati*. Originalna poruka koja se treba šifrirati i poslati nekoj osobi se naziva *otvoreni tekst*, a šifrirana poruka naziva se *šifrat*. Konačna poruka šifrirana i poslana nekoj osobi zove se *kriptogram*. Proces transformiranja izvorne poruke u šifrat naziva se *šifriranje ili kriptiranje*, a obrnuti proces se naziva *dešifriranje ili dekriptiranje*. Proučavanje matematičkih tehnika pokušaja razbijanja kriptografske metode naziva se *kriptoanaliza*. Pojam *kriptologija* koristi se za ujedinjenje proučavanja kriptografije i kriptoanalize. Riječ kriptologija dolazi od grčkog pridjeva *kryptos* što znači *skriven* i riječi *logos* što znači *riječ*.

Kriptologija je vrlo mlada znanstvena grana. Iako se već tisućama godina poruke pokušavaju sakriti od "očiju" ljudi kojima te poruke nisu namjenjene, sustavno proučavanje kriptologije kao znanstvene grane počelo je tek prije stotinjak godina. Prvi poznati dokaz korištenja pronađen je u natpisu uklesanom oko 1900. godine prije Krista, u glavnoj sobi groba plemića Khnumhotepa II, u Egiptu. Iako taj natpis nije bio forma tajnog pisanja, u sebi je sadržavao nekakvu transformaciju originalnog teksta te je najstariji poznati tekst koji tako što radi.

Kriptografski algoritam ili *šifra* je matematička funkcija koja se koristi za šifriranje i dešifriranje. Općenito, radi se o dvije funkcije, jednoj za šifriranje, a drugoj za dešifriranje. Te funkcije preslikavaju osnovne elemente otvorenog teksta u osnovne elemente šifrata, i obratno. Funkcije se biraju iz određene familije funkcija u ovisnosti o ključu. Skup svih mogućih vrijednosti ključeva nazivamo prostor ključeva. Oko 100. god. prije Krista, Julius Cesar je koristio oblik enkripcije da prenese tajne poruke svojim vojnim generalima u ratu. Ova šifra poznata je kao Caesarova šifra i jedna je od najznačajnijih povijesnih šifri. Tijekom 16. stoljeća, Vigenere je dizajnirao šifru koja je navodno bila prva šifra u kojoj se koristio ključ za šifriranje (vidi [10]).

Definicija 1 (vidi [3]). *Kriptosustav je uređena petorka* $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$*, za koju vrijede sljedeća svojstva:*

1. $\mathcal{P}$ *je konačan skup svih mogućih otvorenih tekstova*
2. $\mathcal{C}$ *je konačan skup svih mogućih šifriranih tekstova*
3. $\mathcal{K}$ *je konačan skup svih mogućih ključeva*
4. *Za svaki* $K \in \mathcal{K}$ *postoji funkcija šifriranja* $e_K \in \mathcal{E}$ *i odgovarajuća funkcija dešifriranja* $d_K \in \mathcal{D}$*, gdje su* $e_K : \mathcal{P} \longrightarrow \mathcal{C}$ *i* $d_K : \mathcal{C} \longrightarrow \mathcal{P}$ *funkcije sa svojstvom:*

$$d_K(e_K(m)) = m, \text{ za svaki otvoreni tekst } m \in \mathcal{P}.$$

Iz svojstva $d_K(e_K(m)) = m$ slijedi kako funkcije e_K moraju biti injekcije. Kada bi bilo

$$e_K(m_1) = e_K(m_2) = c,$$

za dva različita otvorena teksta m_1 i m_2, onda primalac ne bi mogao odrediti treba li c dešifrirati u m_1 ili u m_2, odnosno $d_K(c)$ ne bi bio definiran. Uočimo, ako je $\mathcal{P} = \mathcal{C}$ onda su funkcije e_K permutacije.

Ako želimo sačuvati tajnost neke informacije, mogućnosti su ili sakriti postojanje informacije ili ju učiniti nerazumljivom za druge. Moderna kriptografija koristi sofisticirane matematičke jednadžbe (algoritme) i tajne ključeve za šifriranje i dešifriranje podataka.

Danas se kriptografija koristi za pružanje tajnosti i integriteta našim podacima te autentičnosti i anonimnosti prilikom komuniciranja. Moderni kriptografi ističu da sigurnost ne bi trebala ovisiti o tajnosti metode šifriranja (ili algoritmu) nego samo o tajnosti ključeva. Tajni ključevi ne smiju se otkriti kada se uspoređuju otvoreni i šifrirani tekstovi. Suvremeni algoritmi temelje se na matematički teškim problemima kao što su npr. faktorizacija na proste brojeve, diskretni logaritmi, itd. Suvremeni kriptografski algoritmi su previše složeni da bi ih izvršili ljudi. Danas se algoritmi izvode pomoću računala ili specijaliziranih hardverskih uređaja. Dizajn sigurnih sustava korištenjem tehnika šifriranja fokusira se uglavnom na zaštitu tajnih ključeva koji mogu biti zaštićeni tako da se šifriraju pod drugim ključevima ili fizičkom zaštitom, a algoritam koji se koristi za šifriranje podataka je javno objavljen

i podvrgnut intenzivnom pregledu. Danas su mnogi dobri kriptografski algoritmi dostupni u velikim knjižarama, knjižnicama i na internetu ili u patentnom uredu.

Postoji nekoliko načina podjele kriptosustava (vidi [3]). Neki od njih su primjerice:

1. Podjela kriptosustava obzirom na tip operacija koje se koriste pri šifriranju:

   - *Supstitucijske šifre*
     Svaki element otvorenog teksta (bit, slovo, grupa bitova ili slova) zamjenjuje se nekim drugim elementom, a transformacija je unaprijed utvrđena. Ovisno o broju transformacija razlikujemo monoalfabetske i polialfabetske.
     Primjerice GRAD $\rightarrow$ LWFI
   - *Transpozicijske šifre*
     Elementi otvorenog teksta se premještaju odnosno permutiraju. Primjerice GRAD $\rightarrow$ ADGR
   - Kriptosustavi koji kombiniraju ove dvije metode

2. Podjela kriptosustava obzirom na način na koji se obrađuje otvoreni tekst:

   - *Blokovne šifre*
     Obrađuje se jedan po jedan blok elemenata otvorenog teksta koristeći jedan te isti ključ K.
   - *Protočne šifre*
     Elementi otvorenog teksta obrađuju se jedan po jedan koristeći pritom paralelno generirani niz ključeva.

Obzirom na temu ovog diplomskog rada navest ćemo i podjelu kriptosustava obzirom na tajnost ključeva. Tu imamo:

1. *Simetrične kriptosustave (kriptosustavi s tajnim ključem):*
   Za šifriranje i dešifriranje uglavnom se koristi isti ključ ili ako se ne koristi, ključ za dešifriranje lako se izvodi iz ključa za šifriranje. Simetrični kriptosustavi mogu se podijeliti na šifre *signala* ili *stream šifre* i *blokove šifriranja*. Korištenjem stream šifre može se šifrirati jedan dio teksta istodobno, dok se korištenjem blok šifre uzimaju brojni bitovi (obično 64 bita u modernim šiframa) i šifriraju se kao jedna jedinica. Primjer simetričnog kriptosustava je DES (vidi [7]).

2. *Asimetrične kriptosustave (kriptosustavi s javnim ključem):*
   Koristi se različiti ključ za šifriranje i dešifriranje, a ključ za dešifriranje ne može se izvesti iz ključa za šifriranje. Ključ za šifriranje je javni ključ. Drugim riječima, bilo tko može šifrirati poruku pomoću tog ključa, ali samo osoba koja ima odgovarajući ključ za dešifriranje, može dešifrirati tu poruku.

Općenito su simetrični kriptosustavi mnogo brži nego asimetrični. U primjenama se često koriste zajedno, na način da se algoritmom javnog ključa šifrira nasumično generiran ključ za šifriranje, a taj ključ šifrira poruku koristeći simetrični kriptosustav. Ta kombinacija često se zove digitalna omotnica. U ovom diplomskom radu opisat ćemo jedan od najpoznatijih kriptosustava s javnim ključem, RSA kriptosustav. Prezentirat ćemo upotrebu, sigurnost, učinkovitost, te kriptoanalizu RSA kriptosustava.

1 Kriptosustavi s javnim ključem

Javnost se 1970.-ih godina počela upoznavati s kriptografijom. U radu objavljenom 1976. godine Whit Diffie i Martin Hellman osmislili su metodu kojom su komunicirale dvije osobe koje se nikad prije nisu vidjele ili razmijenile ključeve kojima bi razvile zajednički tajni ključ slanjem poruka preko otvorenog (nesigurnog) kanala (vidi [7]). Sve do razvoja te ideje svi kriptosustavi tražili su mehanizme za sigurnu izradu tajnih ključeva. To je nužno jer kada se zna simetrični ključ za šifriranje vrlo je lagano doći do ključa za dešifriranje. Uvođenjem Diffie-Hellmanove ideje osobe mogu osigurati razmjenu ključa na otvorenom i tako osigurati privatnost. Možda se čini kontradiktornim, ali to je jedan izvrstan sustav koji koristi dva bitno različita ključa: jedan za šifriranje i može se javno objaviti, a drugi za dešifriranje koji se privatno čuva. Ključ za dešifriranje bi se konstruirao tako da ga je nemoguće dobiti iz ključa za šifriranje, tj. postojao bi par nesimetričnih ključeva. Pokažimo na primjeru kako kriptosustav s javnim ključem funkcionira.

Primjer 1 (vidi [7])**.** *Uzmimo da dvije osobe žele razmjeniti poruku, a da nitko drugi ne može otkriti što se u toj poruci nalazi. U literaturama su te dvije osobe najčešće nazvane Bob i Alice. Neka Bob ima otvoreni sef u svom uredu u koji svi djelatnici mogu staviti nekakvu poruku, uključujući i Alice, te ga mogu i zaključati. Samo Bob može ponovo otvoriti sef jer jedino on ima potrebnu kombinaciju za otključavanje. Da bismo dobili opći pregled temeljen na Diffie-Hellmanovoj ideji, trebamo uvesti pojam jednosmjerne funkcije, koju možemo promatrati kao metodu šifriranja kod koje je nemoguće provesti dešifriranje, tj. inverzan postupak. (Primjerice, ako netko napiše poruku na komad papira, a zatim tu poruku zapali primjer je jednosmjerne funkcije kojoj je nemoguće odrediti inverz, tj. vratiti papir s porukom i otkriti ju). Gledajući to s matematičke strane, to su funkcije čije su vrijednosti lagane za izračunati, međutim računanje inverza je neisplativo. (Ukoliko zapalimo poruku kako će ona doći do primatelja? Kako će on pročitati sadržaj poruke?) Trebaju nam dodatne informacije ugrađene u našu jednosmjernu funkciju tako da primatelj može obnoviti poruku. Te dodatne informacije zovu se "zamke". S matematičkog aspekta "zamka" je u jednosmjernoj funkciji dodatna informacija koja nalaženje inverza te funkcije čini isplativim zadatkom. Bez nje je to neisplativo. Za sada razmišljajmo o "zamci" kao o informaciji koja nam omogućava naći inverz funkcije tj. dešifriranje poruke, ali ukoliko ju ne znamo, ne možemo to učiniti. Lako je naći jednosmjerne funkcije, ali naći one sa "zamkama" zahtijeva malo više truda. Sada pogledajmo kako Diffie-*

Hellmanova ideja funkcionira na primjeru Boba i Alice.
Bob i Alice nikada se nisu sreli, no žele uspostaviti tajnu komunikaciju. Oboje imaju jedinstvene javne ključeve koje svi možemo vidjeti kao dugačak niz bitova, objavljenih u nekoj javnoj bazi ključeva. Oboje imaju i svoje privatne ključeve koje znaju samo oni. Alice piše poruku i koristi Bobov javni ključ kroz jednosmjernu funkciju, tako da jedino Bobov privatni ključ to može otključati. Kada Alice pošalje poruku, jedina osoba na svijetu koja ju može pročitati je Bob. Pretpostavimo da postoji još jedna osoba koja presretne poruku, znatiželjni Mario. Bez Bobovog privatnog ključa on može samo pokušati pogoditi sadržaj poruke, no za pogodak bi trebali milijuni godina, pa je njegovo pokušavanje beskorisno. Budući da Bob jedini ima oba elementa ključa, može pročitati poruku odmah. Poruka primjerice može sadržavati simetrični ključ K i poveznicu na algoritam simetričnog ključa kao npr. DES. Bob koristi javni ključ od Alice i jednosmjernu funkciju da šifrira odgovor, koji će reći da se slaže s uporabom DES-a sa simetričnim ključem K za razgovor. Zatim to šalje Alice koja koristi svoj privatni ključ da dešifrira poruku što ona jedino i može. U Diffie-Hellmanovoj ideji, K je zajednički tajni ključ kojeg samostalno generiraju Bob i Alice. Bob i Alice završili su ključnu razmjenu i suglasni su na ključ K. Stoga su, preko neosiguranog kanala, uspostavili sigurno sredstvo komunikacije.

Navedimo sada preciznu definiciju kriptosustava s javnim ključem.

Definicija 2. *Kriptosustav s javnim ključem je kriptosustav za kojeg vrijede sljedeća svojstva:*

1. *Za svaki ključ $K \in \mathcal{K}$ i za svaki otvoreni tekst $m \in \mathcal{P}$ lako se izračunaju $e_K(m)$ i $d_K(e_K(m))$.*

2. *Za skoro svaki ključ $K \in \mathcal{K}$, svaki algoritam koji je jednostavan za izračunati i ekvivalentan je s funkcijom dešifriranja d_K računalno je neisplativo izvesti preko e_K. Bez funkcije d_K teško je dešifrirati.*

3. *Algoritam šifriranja funkcijom e_K je javan dok je algoritam dešifriranja funkcijom d_K privatan.*

Kriptosustav s javnim ključem zamišljen je kao sustav s tri algoritma:

- algoritam za generiranje ključa
- algoritam šifriranja

- algoritam dešifriranja.

Algoritam za generiranje ključa implicitno ili eksplicitno definira skup ključeva $\mathcal{K}$, dok algoritmi šifriranja i dešifriranja definiraju skup otvorenih tekstova $\mathcal{P}$ i skup šifriranih tekstova $\mathcal{C}$, odnosno šifrata.

2 RSA kriptosustav

RSA kriptosustav, najpoznatiji i najkorišteniji kriptosustav s javnim ključem u svijetu, ime je dobio po svojim izumiteljima Ronu Rivestu, Adi Shamiru i Lenu Adlemanu. Uveden je 1977. godine u znanstvenom članku tvrtke Gardner, a godinu dana kasnije izumitelji su objavili izvorni originalni znanstveni članak. U svom radu opisuju kriptosustav javnog ključa, uključujući generiranje ključa i šifre javnih ključeva čija se sigurnost temelji na pretpostavljenim poteškoćama rastavljanjem brojeva na njihove proste faktore. RSA kriptosustav koristi se od 18. st. do danas, pri tome se ažurira na našu suvremenu računalnu informacijsku tehnologiju. Danas se koristi primjerice u bankarstvu, za sigurnost e-pošte, prodaje preko interneta itd. Kao jedan od kriptosustava javnog ključa korištenih u sigurnosnim protokolima (TLS, SSL), RSA kriptosustav koristi se na internetu milijunima puta dnevno (vidi [4]).
U vrijeme kada je elektornička pošta postala razvijena, RSA kriptosustav je implementirao dvije važne ideje (vidi [6]):

- *Šifriranje javnih ključeva*
  Ova ideja izostavlja potrebu za dostavom ključeva primatelju preko drugog sigurnog kanala prije prenošenja izvorno namijenjene poruke.
  U RSA kriptosustavu ključevi za šifriranje su javni, a ključevi za dešifriranje su privatni tako da samo osoba s ispravnim ključem za dešifriranje može dešifrirati šifrirane poruke.
  Svatko ima svoje ključeve za šifriranje i dešifriranje i oni moraju biti načinjeni tako da se ključ za dešifriranje ne može lako dobiti iz javnog ključa za šifriranje.

- *Digitalni potpis* (vidi [9])
  Primatelj (npr. Bob) će u nekom trenutku možda trebati potvrditi da je poruku dobio od određene osobe (npr. Alice), a ne od nekog drugog pošiljatelja, te Alice ne može poreći slanje poruke i obratno.
  To se može postići korištenjem pošiljateljevog ključa za dešifriranje, a potpis kasnije može potvrditi bilo tko koristeći javni ključ za šifriranje. Na taj način potpisi se ne mogu krivotvoriti, a pošiljatelj ne može zanijekati poruku.

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

2.1 RSA kriptosustav i javni ključ

Imajući u vidu prethodnu definiciju o kriptosustavu s javnim ključem navedimo sada definiciju RSA kriptosustava.

Definicija 3 (vidi [4]). *Neka je $N = pq$ produkt dvaju prostih brojeva p i q te neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_N$. Definirajmo prostor ključeva kao:*

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

gdje je $\varphi(N) = (p-1)(q-1)$ Eulerova funkcija. Za svaki ključ $K \in \mathcal{K}$, $K = (N, p, q, e, d)$, funkcija šifriranja $e_K : \mathbb{Z}_N \longrightarrow \mathbb{Z}_N$ definirana je s

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

dok je funkcija dešifriranja $d_K : \mathbb{Z}_N \longrightarrow \mathbb{Z}_N$ definirana 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 šifriranja $e_K(x) = x^e \bmod N$, gdje je faktorizacija broja N nepoznata, te $(e, \varphi(N)) = 1$ 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 trebaju zadovoljavati kongruenciju

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

slijedi nam

$$ed = 1 + k\varphi(N), \tag{1}$$

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

Teorem 1 (vidi [7, Theorem 1.18.]). *Ako su cijeli brojevi a i prirodan broj n relativno prosti, onda vrijedi:*

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

Dokaz. Vidi [3]. □

Kod nas je $n = N$, pa imamo

$$a^{\varphi(N)} \equiv 1(\text{mod } N).$$

Dobivenim javnim ključem (N, e) i porukom otvorenog teksta $m \in \mathbb{Z}_N^*$, tj. $m \in \mathbb{Z}_N$ i $(m, N) = 1$, algoritam šifriranja izračunava šifriranu poruku

$$c = m^e \text{ mod } N.$$

Budući je $ed = 1 + k\varphi(N)$ i

$$\begin{aligned} c^d(\text{mod}\, N) &\equiv (m^e)^d(\text{mod}\, N) \\ &\equiv m^{ed}(\text{mod } N) \\ &\equiv m^{1+k\varphi(N)}(\text{mod}\, N) \\ &\equiv m(m^{\varphi(N)})^k(\text{mod}\, N) \\ &\equiv m(\text{mod } N), \end{aligned}$$

koristeći algoritam dešifriranja dobivamo otvoreni tekst. Kako je $m \in \mathbb{Z}_N$, slijedi $c^d(\text{mod}\, N) = m$.

Primjer 2. *Uzmimo $p = 7$, $q = 13$. Tada je $N = 91$, a $\varphi(N) = 6 \cdot 12 = 72$. Kako enkripcijski eksponent e mora biti relativno prost sa 72, pa recimo da je $e = 5$. Sada primjenom Euklidovog algoritma (ako nam nije očigledno), slijedi $d = 29$. Sada je $(N, e) = (91, 5)$ naš javni ključ. Pretpostavimo da nam netko želi poslati poruku $m = 72$. To znači da mora izračunati $e_K(m) = 72^5 \text{ mod } 91$.*

$$72^5 = 72^3 \cdot 72^2 \equiv 11(\text{mod } 91).$$

Dobiveni šifrat je $c = e_K(m) = 11$. Kada primaoc primi ovaj šifrat, dešifrira ga pomoću tajnog ključa d:

$$m = d_K(c) = 11^{29} \equiv 72(\text{mod } 91).$$

Dakle, $m = 72$.

Primjer 3 (vidi [3])**.** *U RSA kriptosustavu s javnim ključem* $(2047, 411)$ *dešifrirajte poruku "BP" u engleskom alfabetu.*
Rješenje:
Faktorizirajmo N*. Imamo* $N = 23 \cdot 89$*,* $e = 411$*, pa je* $\varphi(N) = \varphi(23 \cdot 89) = 1936$*. Iz* $(\varphi(N), e) = 1$ *slijedi kako postoje* $d, l \in \mathbb{Z}$ *takvi da je* $411d + 1936l = 1$*. Primjenom Euklidovog algoritma dolazimo do* d*:*

$$\begin{aligned} 1936 &= 411 \cdot 4 + 292 \\ 411 &= 292 \cdot 1 + 119 \\ 292 &= 119 \cdot 2 + 54 \\ 119 &= 54 \cdot 2 + 11 \\ 54 &= 11 \cdot 4 + 10 \\ 11 &= 10 \cdot 1 + 1 \\ 10 &= 1 \cdot 10 + 0. \end{aligned}$$

	-1	0	1	2	3	4	5	6
q_i			4	1	2	2	4	1
d_i	0	1	-4	5	-14	33	-146	179

Dakle, $d = 179$*. Šifratu "BP" pridružujemo numerički ekvivalent tako da slovima A,B,...,Z redom pridružujemo brojeve* $1, 2, \ldots, 26$*. Slovima A,B,...,Z možemo pridruživati brojeve počevši i od 0, tj. slovima A,B,...,Z možemo pridružiti redom brojeve* $0, \ldots, 25$*. Dobivamo:*
$B\ P \to 2\ 16$*. i dešifriramo pomoću funkcije*

$$d_K(c) = c^{179} \bmod 2047, \quad za\ \ c = 2, 16.$$

Dobivamo

$$\begin{aligned} d_K(2) &= 2^{179} \bmod 2047 = 8 \\ d_K(16) &= 16^{179} \bmod 2047 = 2, \end{aligned}$$

pa je otvoreni tekst jednak "HB". Ako želimo provesti inverzan postupak, šifriramo funkcijom

$$e_K(m) = m^{411} \bmod 2047, \quad za\ \ m = 8, 2.$$

Dobivamo

$$\begin{aligned} e_K(8) &= 8^{411} \bmod 2047 = 2 \\ e_K(2) &= 2^{411} \bmod 2047 = 16, \end{aligned}$$

pa je šifrat jednak "BP".

Za poruke otvorenog teksta koje su relativno proste s modulima točnost algoritma dešifriranja može se jednostavno pokazati koristeći sljedeći teorem poznat kao Kineski teorem o ostacima.

Teorem 2 (Kineski teorem o ostacima, vidi [7, Theorem 1.12])**.** *Neka su N_i, $i \leq k \in \mathbb{N}$ u parovima relativno prosti prirodni brojevi, te neka je*

$$N = \prod_{i=1}^{k} N_i.$$

Neka su $r_1, ..., r_k$ cijeli brojevi. Tada sustav kongruencija:

$$\begin{aligned} x &\equiv r_1 (\bmod N_1), \\ &\vdots \\ x &\equiv r_k (\bmod N_k) \end{aligned}$$

ima jedinstveno rješenje modulo N.

Dokaz. vidi [7]. □

Primjer 4 (vidi [7, Example 1.8.])**.** *Riješimo sustav kongruencija:*

$$\begin{aligned} x &\equiv 2 (\bmod 3), \\ x &\equiv 2 (\bmod 5), \\ x &\equiv 3 (\bmod 7). \end{aligned}$$

Primjenom Kineskog teorema o ostacima, slijede linearne kongruencije:

$$35x \equiv 2 (\bmod 3), \tag{2}$$
$$21x \equiv 2 (\bmod 5), \tag{3}$$
$$15x \equiv 3 (\bmod 7). \tag{4}$$

Rješenje kongruencije (2)*, tj. rješenje kongruencije $2x \equiv 2 (\bmod 3)$, kako je $(35, 3) = 1$, postoje cijeli brojevi u i v takvi da je $35u + 3v = 1$. Primjenom Euklidovog algoritma dobije se $u = -1$, pa su sva rješenja kongruencije* (2) *dana s $x_1 \equiv -1 \cdot 2 \equiv$*

$1(\bmod 3)$. *Slično se dobije i rješenje kongruencija* (3) *i* (4). *Sva rješenja kongruencije* (3) *dana su s* $x_2 \equiv 1 \cdot 2(\bmod 5) \equiv 2(\bmod 5)$. *Sva rješenja kongruencije* (4) *dana su s* $x_3 \equiv 1 \cdot 3(\bmod 5) \equiv 3(\bmod 5)$. *Prema tome, sva rješenja sustava kongruencija* (2), (3) *i* (4) *dana su s*

$$x \equiv 35 \cdot 1 + 21 \cdot 2 + 15 \cdot 3 \equiv 122(\ \bmod 3 \cdot 5 \cdot 7).$$

Definirajući javne i privatne eksponente kao inverzne module $\varphi(N)$ (što je originalno i napravljeno za RSA) pruža nam dovoljan, ali ne i nužan uvjet kojim bi pravilo za dešifriranje vratilo otvoreni tekst iz šifriranog teksta. Nužan uvjet je taj da su javni i privatni eksponenti jedan drugom inverzi modulo Charmichaelova lambda funkcija λ.

Definicija 4 (vidi [7])**.** *Carmichaelova lambda funkcija od* N *je najmanji broj takav da je*

$$a^{\lambda(N)} \equiv 1(\text{mod } N),$$

pri čemu je cijeli broj a *relativno prost s* N.

Dovoljno je definirati privatne i javne eksponente kao inverze jedan drugome modulo bilo koji višekratnik od $\lambda(N)$. Za RSA modul $N = pq$ Charmichael lambda funkcija dana je s $\lambda(N) = NZV(p-1, q-1)$. Broj $\varphi(N)$ je višekratnik od $\lambda(N)$ budući da je

$$\begin{aligned} \varphi(N) &= (p-1)(q-1) \\ &= NZD(p-1, q-1)NZV(p-1, q-1) \\ &= NZD(p-1, q-1)\lambda(N), \end{aligned}$$

što nam dopušta da koristimo $\varphi(N)$ u algoritmu za generiranje ključa.

Primjer 5 (vidi [5, Primjer 19.])**.** *Najmanji Carmichaelov broj je* 561. *Rastavimo* 561 *na proste faktore* $561 = 3 \cdot 11 \cdot 17$. *Neka je* a *prirodan broj koji je relativno prost s* 561. *Očito je da* a *nije djeljiv sa* 3, 11 *i* 17, *pa je* $a^{m-1} \equiv 1(\text{mod } m)$, *za* $m \in \{3, 11, 17\}$. *Kako je* $560 = 2 \cdot 280 = 10 \cdot 56 = 16 \cdot 35$, *dobivamo da je* $a^{560} \equiv 1(\text{mod } m)$, *za* $m \in \{3, 11, 17\}$, *odakle slijedi* $a^{560} \equiv 1(\text{mod } 561)$.

2.2 Implementacija RSA kriptosustava

Vratimo se sada na primjer Boba i Alice, uvodeći oznake za javne i privatne ključeve iz prethodne definicije. Podijelimo algoritam na dva dijela, pretpostavljajući da Alice želi poslati poruku Bobu.

1. *Generiranje RSA ključa*

   - Bob generira dva velika, slučajno odabrana prosta broja p i q za koje vrijedi $p \neq q$, $N = pq$ i $\varphi(N) = (p-1)(q-1)$.
   - Odabire bilo koji $e \in \mathbb{N}$ takav da je $1 < e < \varphi(N)$ i $(e, \varphi(N)) = 1$. Zatim pomoću proširenog Euklidovog algoritma izračuna jedinstveni $d \in \mathbb{N}$ takav da je $q < d < \varphi(N)$ i $ed \equiv 1(\text{mod } \varphi(N))$.
   - Bob objavljuje (N, e) u nekoj javnoj bazi ključeva, a d, p, q i $\varphi(N)$ su znani samo njemu.

2. *Šifrirani RSA javni ključ*

   - *Postupak šifriranja*:
     Pretpostavimo da je otvoreni tekst $m \in \mathcal{M}$ numeričkog oblika tako da je $m < N$ te da je $\mathcal{M} = \mathcal{C} = \mathbb{Z}/N\mathbb{Z}$ i pretpostavimo da je $(m, N) = 1$.
     (a) Alice dohvaća Bobov javni ključ iz baze ključeva.
     (b) Šifrira poruku m računajući $c \equiv m^e(\text{mod } N)$ koristeći metodu "Kvadriraj i množi" (vidi [7]) i šalje $c \in \mathcal{C}$ Bobu.
   - *Postupak dešifriranja*:
     Kada Bob primi poruku c, koristi dekripcijski eksponent d, te dobiva otvoreni tekst $m \equiv c^d(\text{mod } N)$.

     Napomena 1. *Da bi vidjeli kako Bobovo dešifriranje daje otvoreni tekst m, promotrimo sljedeće. Kako je,*

     $$ed \equiv 1(\text{ mod } \varphi(N)),$$

     postoji $g \in \mathbb{Z}$ takav da je

     $$ed \equiv 1 + g\varphi(N).$$

     Ako $p \nmid m$, prema Malom Fermatovom teoremu slijedi $m^{p-1} \equiv 1(\text{mod } p)$. Stoga je

     $$m^{ed} = m^{1+g(p-1)(q-1)} \equiv m(m^{g(q-1)})^{p-1} \equiv m(\text{ mod } p). \tag{5}$$

Ako $p|m$, tj. $m \equiv 0 (\text{mod } p)$, također vrijedi (5). Stoga je $m^{ed} \equiv m (\text{mod } p)$, za bilo koji m. Slično se pokaže i $m^{ed} \equiv m (\text{mod } q)$. Kako je $p \neq q$, imamo $m^{ed} \equiv m (\text{mod } N)$. Iz toga nam slijedi

$$c^d \equiv (m^e)^d \equiv m (\text{mod } N).$$

Primjer 6. *Pokažimo kako Alice želi poslati Bobu poruku*

MATEMATIKA.

- *Bob odabire proste brojeve $p = 17$ i $q = 53$, odakle dobiva $N = pq = 901$, te $\varphi(N) = (p-1)(q-1) = 832$. Zatim odabire enkripcijski eksponent $e = 11$ i računa dekripcijksi eksponent d takav da je $ed \equiv 1 (\text{mod } \varphi(N))$. Tako dobiva $d = 227$.*

- *Bob u javnoj bazi ključeva ostavlja svoj javni ključ $(N, e) = (901, 11)$, a p, q, d i $\varphi(N)$ čuva u tajnosti.*

- *Alice dohvaća Bobov javni ključ kako bi šifrirala poruku*

  MATEMATIKA.

  Numerički ekvivalent otvorenog teksta dobiva se tako da se slovima A,...,Z redom pridruže brojevi 01,...,26, s tim da razmak označimo s 00. Numerički ekvivalent prethodne poruke je

  $$x = 13012005130120091101.$$

  Budući da je $x > N$ prije šifriranja x se dijeli u blokove od po 2 znamenke. Sada je

  $$\begin{aligned} x &= (x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}) \\ &= (13, 01, 20, 05, 13, 01, 20, 09, 11, 01). \end{aligned}$$

- *Uz pomoć Bobovog javnog ključa* (N, e) *Alice računa* $y_i = x_i^e \bmod N$ *za sve* $i = 1, ..., 10$.

$$\begin{aligned} y_1 &= 13^{11} \bmod 901 = 599 \\ y_2 &= 01^{11} \bmod 901 = 1 \\ y_3 &= 20^{11} \bmod 901 = 330 \\ y_4 &= 05^{11} \bmod 901 = 232 \\ y_5 &= 13^{11} \bmod 901 = 599 \\ y_6 &= 01^{11} \bmod 901 = 1 \\ y_7 &= 20^{11} \bmod 901 = 330 \\ y_8 &= 09^{11} \bmod 901 = 865 \\ y_9 &= 11^{11} \bmod 901 = 590 \\ y_{10} &= 01^{11} \bmod 901 = 1. \end{aligned}$$

Dobiveni šifrat

$$y = 599001330232599001330865590001$$

Alice dalje šalje Bobu.

- *Dijelećи y na blokove, Bob na sličan način pomoću dekripcijskog eksponenta* $d = 227$ *računa*

$$x_i = y_i^{227} \bmod 901, \quad i = 1, ..., 10$$

i dobiva otvoreni tekst

MATEMATIKA.

2.3 Sigurnost RSA kriptosustava

RSA kriptosustav jedan je od najsigurnijih kriptosustava, ali niti on nije sasvim siguran od napada kriptoanalitičara. Metode poput *brute-force* metode (vidi [7]) jednostavne su i dugotrajne i mogu otkriti dio poruke, ali vjerojatno ne i cijelu poruku. Ne zna se kako dokazati je li neki algoritam šifriranja nepopustljiv, ali se onda provjerava može li netko slomiti taj algoritam. Do sada nije poznato da je netko razbio RSA algoritam što ga čini sigurnim u praksi. Sigurnost RSA kriptosustava oslanja se na teškoće rješavanja tzv. RSA problema. S obzirom na RSA javni ključ

(N, e) i šifrirani tekst $y = x^e \bmod N$, teško je računanje otvorenog teksta x. To je zapravo problem računanja $e - tog$ korijena modulo N ili invertiranja funkcije d_K. Postoji RSA pretpostavka koja tvrdi da je RSA problem teško riješiti kada je otvoreni tekst $x \in \mathbb{Z}_N$ slučajno odabran i modul je dovoljno velik sa slučajno odabranim prostim brojevima.

2.4 Učinkovitost RSA kriptosustava

Promotrimo sada efikasnost RSA kriptosustava. Posebno ćemo razmatriti trajanje algoritma za generiranje prostih brojeva i modularnog potenciranja što su dominantne operacije za algoritam generiranja ključa i za algoritme šifriranja i dešifriranja.

Odabir prostih brojeva

Algoritam za generiranje ključa treba generirati dva nasumična prosta broja koji su otprilike iste veličine. Može se generirati $n - bitni$ 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 generirati. Postoje mnogi algoritmi za generiranje prostih brojeva, primjerice *Pollardov Rho algoritam*, *Faktorizacija razlikom kvadrata*, *Gordonov algoritam za generiranje prostih brojeva*, itd. (za detalje vidi [7]).

Modularno potenciranje

Šifriranje i dešifriranje u RSA kriptosustavu sastoji se od modularnog potenciranja. Te operacije mogu biti vrlo skupe kada su eksponent i modul veliki. Postoji mnogo različitih algoritama, ali složenost ovog izračuna može se smanjiti na računanje broja modularnih množenja pomoću tzv. metode *Kvadriraj i množi*:

1. Prikažemo e u bazi 2: $e = 2^{k-1}e_{k-1} + ... + 2e_1 + e_0$.
2. Zatim se primjeni algoritam:
   $y = 1$
   $for(k - 1 \geq i \geq 0)$
   $\{$
   $\quad y = y^2(\mod N)$
   $\quad\quad if(e_i = 1)$
   $\quad\quad\quad y = y \cdot x \mod N$
   $\}$

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

3 Kriptoanaliza RSA kriptosustava

Postoji mnogo različitih tipova napada na RSA. Više je vrsta napada iz različitih kanala, koji izrabljuju neka fizička svojstva uređaja na kojem je RSA implementiran. Neki od njih uključuju: napad bugovima, 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 neke vrste manipulacija. Primjerice, 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 koje ćemo proučavati temelje se na matematičkoj strukturi RSA kriptosustava (modulu, jednadžbi ključa) i korištenju određenih izbora parametara (korištenje malog javnog ili privatnog eksponenta). Također, uključeni su i napadi koji koriste neko znanje o privatnom ključu, ali nećemo se baviti kako su se takve informacije dobile.

3.1 Faktorizacija

Prvi napad koji ćemo promatrati je faktorizacija broja N na način $N = pq$, gdje su p i q prosti brojevi. Budući da iz faktora od N dolazimo do $\varphi(N) = (p-1)(q-1)$, a samim time i do d iz $d = e^{-1} \bmod N$, kriptoanalitičari bi razbili kôd faktoriziranjem 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 kriptosustava, pod uvjetom da se RSA pravilno koristi.
Sljedeću tablicu autori RSA kriptosustava predstavili su 1978. godine. Pretpostavili su da operaciji u Schroeppelovom algoritmu faktoriziranja treba jedna mikrosekunda za određivanje prostih faktora broja N. Tako su predstavili sljedeću tablicu za različite duljine broja N:

Broj znamenki	**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 godina
200	$1.2 \cdot 10^{23}$	$3.8 \cdot 10^{9}$ godina
300	$1.5 \cdot 10^{29}$	$4.9 \cdot 10^{15}$ godina
500	$1.3 \cdot 10^{39}$	$4.2 \cdot 10^{25}$ godina

Preporučljivo je da N ima više od 200 znamenki, tj. da se za p i q odabiru brojevi koji imaju oko 100 znamenki, ali duljina od N može varirati na temelju važnosti brzine šifriranja u odnosu na sigurnost. RSA kriptosustav dopušta korisniku da sam izabire duljinu ključa, a samim time i razinu sigurnosti.
Naš je zadatak istražiti napade na RSA kriptosustav koji dešifriraju poruke bez izravnog faktoriziranja RSA modula N. Postoje slučajevi u kojima se $N = pq$ lako može faktorizirati. Na primjer, ako $p - 1$ i $q - 1$ imaju male proste faktore ili ako su p i q približnih vrijednosti pa takve slučajeve treba izbjegavati. Kao što je prethodno navedeno, ako postoji učinkovit algoritam za faktorizaciju broja N, onda je RSA kriptosustav nesiguran. Više o metodama faktorizacije prostih brojeva može se vidjeti u [8].

3.2 Najraniji napadi

Navest ćemo neke od ranije poznatih napada na RSA kriptosustav. Neki od tih ranih napada su primjeri propusta u protokolima. Neuspjeh protokola pojavljuje se kada se sigurnosnim dijelom kriptosustava ne postupa ispravno, što rezultira u neuspjehu željenih sigurnosnih ciljeva protokola. Kvarovi protokola omogućuju protivniku da izračuna otvoreni tekst s obzirom na nekoliko šifriranih tekstova.

Napad zajedničkim modulom

Kako bi se izbjeglo generiranje različitih modula $N = pq$ za svakog korisnika unutar nekog sustava, osmišljeno je da svaki korisnik ima javni ključ s istim modulom N. Pouzdano središnje tijelo generira RSA modul i dijeli parove ispravnih ključeva svim korisnicima, gdje svi ključevi imaju isti modul. Namjera je bila da samo središnje ključno tijelo zna faktorizaciju tog zajedničkog modula. Simmons je 1983. godine pokazao da taj sustav ne funkcionira ako je isti otvoreni tekst šifrirao s dva različita javna ključa koji su imali iste module i relativno proste javne eksponente. S obzirom na dva šifrirana teksta i dva javna ključa, pokazao je da se lako može doći do otvorenog teksta.
Neka su (e_1, N) i (e_2, N) javni ključevi takvi da je $(e_1, e_2) = 1$. Tada lako možemo doći do cijelih brojeva u i v takvih da je $ue_1 + ve_2 = 1$. Za bilo koji otvoreni tekst x dan s $y_1 = x^{e_1} (\mathrm{mod}\, N)$ i $y_2 = x^{e_2} (\mathrm{mod}\, N)$, otvoreni tekst može se dobiti računajući $y_1^u y_2^v (\mathrm{mod}\, N)$, budući da računajući u $\mathbb{Z}_N$ imamo

$$y_1^u y_2^v = x^{ue_1} x^{ve_2} = x^{ue_1 + ve_2} = x.$$

Promatrajući dva šifrirana teksta, taj napad može izvršiti bilo koji korisnik koji ima pristup javnom ključu. 1984. godine DeLaurentis je pokazao da je protokol potpuno nesiguran. Pokazao je da je dovoljno poznavanje bilo kojeg javnog i privatnog ključa za izračunavanje valjanog privatnog ključa za bilo koji drugi javni ključ s istim modulom. Taj rezultat dan je sljedećim teoremom.

Teorem 3 (vidi [4, Theorem 3.1.])**.** *Neka je (e, N) valjani javni ključ s odgovarajućim privatnim ključem (d, N), te neka je (e_1, N) drugi valjani javni ključ tako da je $e_1 \neq e$. Za dane e, d, N i e_1, valjani dekripcijski eksponent za javni ključ (e_1, N) dan s*

$$d_1 = e_1^{-1} \bmod \frac{ed-1}{(e_1, ed-1)},$$

može se izračunati u polinomijalnom vremenu $\log(N)$.

Dokaz. Neka je k neki pozitivan cijeli broj. Izrazom

$$ed - 1 = k\lambda(N)$$

dana je jednadžba ključa za poznat par privatnih i javnih ključeva (e, N), (d, N). Kako je e_1 valjani javni eksponent, on zadovoljava $(e_1, \lambda(N)) = 1$ pa je onda $(e_1, k\lambda(N)) = k'$, za neki cijeli broj k' koji dijeli k. Stavimo da je $k^* = k/k'$. Tada dobivamo

$$\frac{ed-1}{(e_1, ed-1)} = \frac{k\lambda(N)}{k'} = k^*\lambda(N).$$

Privatni eksponent d_1 zadovoljava jednadžbu

$$e_1 d_1 = 1 + k_1(k^*\lambda(N)),$$

za neki pozitivan cijeli broj k_1. Prema tome, $e_1 d_1 \equiv 1 (\bmod \lambda(N))$, pa je d_1 valjani privatni eksponent za javni ključ (e_1, N). Budući da se sva računanja mogu završiti u polinomijalnom vremenu $\log(N)$ teorem je dokazan. □

Primjer 7. *Neka Mario koristi javni ključ $(3, 51) = (e, N)$ i njemu odgovarajući ključ $(5, 51) = (d, N)$. Presreo je poruku koju je Alice poslala Bobu i želi ju pročitati. U bazi ključeva pronašao je Bobov javni ključ $(5, 51) = (e_1, N)$, te uočio da Bobov javni ključ ima isti modul $N = 51$ kao i njegov. Odlučio je uz pomoć svog javnog i privatnog ključa, te Bobovog javnog ključa doći i do Bobovog privatnog ključa (d_1, N) te tako otkriti što piše u poruci. Koristeći $d_1 = e_1^{-1} \bmod \frac{ed-1}{(e_1, ed-1)}$, dolazi do $d_1 = 13$, odnosno otkrio je Bobov privatni ključ $(13, 51)$ i sada može dešifrirati poruku koja je namjenjena Bobu.*

DeLaurentis je pokazao kako se s danim parom privatnog i javnog ključa modul može faktorizirati u polinomijalnom vremenu. Za dane e i d taj algoritam iz $ed - 1 = k\varphi(N)$ jednostavno računa višekratnik od $\varphi(N)$, iz čega možemo doći do faktorizacije za N. Taj protokol je nesiguran jer svaki korisnik može faktorizirati modul koristeći samo svoj par privatno-javnog ključa. Kao rezultat tih napada, jasno je da svaki RSA modul treba biti poznat samo jednom korisniku, tj. da ga ne bi smjelo koristiti više korisnika.

Zasljepljujući napad

Pretpostavimo da je (d, N) Bobov privatni ključ, a (e, N) njemu odgovarajući javni ključ. Pretpostavimo da Bobov protivnik Mario želi njegov potpis na poruci $m \in \mathbb{Z}_N^*$, te da je Bob odbio potpisati poruku m. Tada Mario može pokušati odabrati slučajan $r \in \mathbb{Z}_N^*$ i postaviti $m' = r^e m \bmod N$ i pitati Boba da potpiše slučajnu poruku m'. Ako Bob na tu poruku stavi svoj potpis, označimo ga s S', kako je $S' = (m')^d \bmod N$, Mario jednostavno može izračunati $S = S'/r \mod N$, te dobiti Bobov potpis na izvornoj poruci m. Zaista je

$$S^e = (S')^e/r^e = (m')^{ed}/r^e \equiv m'/r^e = m \bmod N.$$

Primjer 8. *Neka je* $(5, 51)$ *Bobov privatni ključ, a* $(3, 51)$ *njemu odgovarajući javni ključ. Mario je pitao Boba da mu potpiše poruku* $m = 12$*, no Bob se sa sadržajem te poruke ne slaže, te odbija potpisati poruku. Mario odlučuje prevariti Boba i pomoću nove poruke doći do Bobovog potpisa na poruci* m*. Odabire cijeli broj* $r = 9$ *i postavlja novu poruku* m' *tako da vrijedi* $m' = r^e m \bmod N$*, te dobiva novu poruku* $m' = 27$*. Zamolio je Boba da mu potpiše poruku* m'*, na što Bob pristaje i potpisuje poruku. Označimo njegov potpis sa* S'*. Kako je* $S' = (m')^d \bmod N$*, slijedi* $S' = 6$*. Mario je sretan jer je nadmudrio Boba i računajući* $S = S'/r \mod N$*, dolazi do Bobovog potpisa* $S = 12$ *na izvornoj poruci* m*.*

Ova tehnika nazvana je zasljepljujuća jer omogućuje Mariu dobiti valjani potpis na svojoj poruci, tražeći Boba da potpiše slučajnu poruku. Pritom Bob nema nikakvih informacija o poruci koju potpisuje.

Ciklički napadi

Posljednji od ranih napada koje ćemo promotriti su ciklički napadi. 1977. godine Simmons i Norris primijetili su da se otvoreni tekst uvijek može dobiti ponovnim šifriranjem njegovog šifrata, sve dok se ne vrati na sebe, tj. ciklusi se vraćaju na izvorni šifrirani tekst.

S obzirom na šifrirani tekst $c = m^e \bmod N$ i javni ključ (e, N), ako je nakon $l+1-og$ ponovnog šifriranja šifrirani tekst otkriven tj.

$$c^{e^{l+1}} \equiv c(\bmod N),$$

slijedi da je

$$c^{e^l} \equiv m(\bmod N).$$

Do otvorenog teksta dolazi se nakon l ponovnih šifriranja. Za izvorni ciklički napad bilo je dovoljno pronaći najmanji l koji se u nekim literaturama naziva i *eksponent oporavka* za otvoreni tekst m. Bez dokaza navodimo sljedeći teorem.

Teorem 4 (vidi [4, Theorem 3.4.])**.** *Neka je (e, N) odgovarajući RSA javni ključ. Za bilo koji otvoreni tekst $m \in \mathbb{Z}_N^*$ eksponent oporavka dijeli $\lambda(\lambda(N))$.*

Prethodni teorem implicira kako je najveći mogući eksponent oporavka $\lambda(\lambda(N))$. Ako su prosti brojevi odabrani tako da je $\lambda(\lambda(N))$ dovoljno mali, napad je izvediv za sve otvorene tekstove. Ako $\lambda(\lambda(N))$ ima samo male proste faktore, napad može biti izvediv za neke otvorene tekstove. Očekuje se da će većina otvorenih tekstova imati veliki eksponent oporavka budući da je $\lambda(\lambda(N))$ velik i ima velike proste faktore. Pokazano je da za gotovo sve izbore uravnoteženih prostih brojeva i javnih eksponenata, svi osim konačno mnogo otvorenih tekstova imat će eksponent oporavka $l > N^{1-\epsilon}$, za neki dovoljno mali ϵ. Odnosno, za dovoljno veliki N, očekuje se da će ciklički napad biti nemoguć.
1979. godine, Williams i Schmid generaliziraju ciklički napad na traženje ciklusa modulo p (ili q) umjesto modulo N kao što je izvorna metoda tražila. Ovdje se traži najmanji k tako da zadovoljava relaciju

$$g = (c^{e^k} - c, N) > 1.$$

Ako je $1 < g < N$, tada je ciklički modul p (ili q) nađen i g otkriva faktorizaciju modula, odnosno $g = p$ (ili $g = q$). Ako je $g = N$, onda je ciklički modul N nađen baš kao i u izvornom napadu i vrijedu $c^{e^{k-1}} \equiv m(\bmod N)$. Učinkovitost modificiranog napada ovisi o veličini k. Ako je prost broj p izabran tako da $\lambda(\lambda(p)) = \lambda(p-1)$ ima samo male proste faktore ili je $\lambda(\lambda(p))$ dovoljno mala, tada ciklički modul p može biti nađen s relativno malim k. Analogno vrijedi i za q. Ako su prosti brojevi slučajno odabrani, očekuje se da će svi osim vrlo malog broja otvorenih tekstova imati $k > N^{1/2-\epsilon}$. Za dovoljno velike slučajno odabrane proste brojeve očekuje se da će modificirani ciklički napad biti nemoguć.

Primjer 9. *Pretpostavimo da Alice i Bob ne paze u određivanju njihovih ključeva. Neka Alice šifrira poruku* $m = 5$ *pomoću Bobovog javnog ključa* $(3, 51)$ *i tako dobiva* $c = 23$*. Mario presreće Alicinu poruku i pokušat će ju razbiti pomoću cikličkog napada.*

$$\begin{aligned} e(23) &= 23^3 \bmod 51 = 29 \\ &= 29^3 \bmod 51 = 11 \\ &= 11^3 \bmod 51 = 5 \\ &= 5^3 \bmod 51 = 23. \end{aligned}$$

Kada nakon cikličkog šifriranja Mario otkrije izvornu vrijednost $(c = 23)$ *koju je presreo, vratit će se jedan korak u računanju kako bi otkrio dešifriranu poruku. Ono što je Mario šifrirao na vrijednost* 23*, mora biti jednako onome što je Alice šifrirala na vrijednost* 23*. Kod za šifriranje je otkriven i Mario može čitati "sigurnosne" poruke između Alice i Boba.*

S obzirom na dovoljno vremena, ciklički napad uvijek će moći otkriti poruke koje su šifrirane RSA algoritmom. Potrebno je puno vremena kako bi se pomoću cikličkog napada otkrila šifrirana poruka. Tako RSA kriptosustav sigurnim od cikličkog napada čine:

1. "Jaki" prosti brojevi

   - p je jak prosti broj ako brojevi $p - 1$ i $p + 1$ imaju velike proste faktore u i v.
   - $u - 1$, $u + 1$, $v - 1$ i $v + 1$ također imaju velike proste faktore.

   Ako se odaberu jaki prosti brojevi p i q povećava se broj ciklusa potrebnih za prekid šifriranja.

2. Veliki prosti brojevi

   - ključevi veći od 60 bitova neće se moći razbiti pomoću cikličkog napada unutar 24 sata.
   - Kako RSA trenutno koristi ključeve od 1024 bita ili više, potrajalo bi mnogo godina dok se ne razbije kôd.

3.3 Mali privatni eksponent

Ilustrirajmo sada opasnosti korištenja RSA kriptosustava s malim privatnim eksponentom d. U [4] M. Wiener pokazuje da mali d uzrokuje potpuni prekid kriptosustava. U RSA kriptosustavu, svi privatni eksponenti $d < 2^l$, gdje l ovisi o trenutnom najsuvremenijem računanju, mogu biti jednostavno pogođeni. Primjerice, trenutno je moguće otkriti sve privatne eksponente $d \leq 2^{60}$, ali ne i za one $d \leq 2^{80}$. Navedimo sada teorem koji ćemo iskoristiti u dokazu poznatog rezultata o Wienerovu napadu.

Teorem 5 (Legendre, vidi [2, Teorem 1.7]). *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.*

Teorem 6 (M. Wiener, vidi [2, Teorem 3.1]). *Neka je $N = pq$ i $p < q < 2p$, te neka je $d < \frac{1}{3}N^{0.25}$. Tada postoji polinomijalni algoritam koji iz poznavanja N i e izračunava d.*

Dokaz. Kako je $ed = 1 \bmod \varphi(N)$, postoji prirodan broj k takav da je $ed - k\varphi(N) = 1$. Odavde je

$$\left|\frac{e}{\varphi(N)} - \frac{k}{d}\right| = \frac{1}{d\varphi(N)}. \tag{6}$$

To znači da $\frac{e}{\varphi(N)}$ aproksimira $\frac{k}{d}$. Kako je $\varphi(N)$ nepoznat, s N ćemo aproksimirati $\varphi(N)$. Kako je $\varphi(N) = N-p-q+1$ i $p+q-1 < 3\sqrt{N}$ slijedi da je $|N-\varphi(N)| < 3\sqrt{N}$. Ako $\varphi(N)$ zamjenimo s N dobivamo

$$\begin{aligned}\left|\frac{e}{N} - \frac{k}{d}\right| &= \left|\frac{ed - k\varphi(N) - kN + k\varphi(N)}{Nd}\right| \\ &= \left|\frac{1 - k(N - \varphi(N))}{Nd}\right| \leq \left|\frac{3k\sqrt{N}}{Nd}\right| = \frac{3k}{d\sqrt{N}}.\end{aligned}$$

Sada je $k\varphi(N) = ed - 1 < ed$. Kako je $e < \varphi(N)$, uočimo da je $k < d < \frac{1}{3}N^{1/4}$, pa dobivamo

$$\left|\frac{e}{N} - \frac{k}{d}\right| \leq \frac{1}{dN^{1/4}} < \frac{1}{2d^2}. \tag{7}$$

Prema Legendreovom teoremu, $\frac{k}{d}$ je neka konvergenta razvoja u verižni razlomak od $\frac{e}{N}$. Nakon što izračunamo sve konvergente, testiramo koja od njih zadovoljava

$x^{ed} \equiv x(\mod N)$ za nasumično odabrani x. To daje polinomijalni algoritam za otkrivanje tajnog eksponenta d. □

Primjer 10 (vidi [2, Primjer 3.1.]). *Pretpostavimo da su u RSA kriptosustavu zadani javni eksponent* $e = 3594320245477$*, te modul* $N = 7978886869909$*. Neka je poznato da tajni eksponent* d *zadovoljava* $d < \frac{1}{3}N^{0.25}$*. Odredimo* d*. Da bi primijenili Wienerov napad, računamo razvoj broja* $\frac{e}{N}$ *u verižni razlomak. Dobivamo:*

$$[0, 2, 4, 1, 1, 4, 1, 2, 31, 21, 1, 3, 1, ...]$$

a zatim računamo pripadne konvergente. Dobivamo:

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

Zahtijevamo $d < 561$*, pa provjeravamo koji od nazivnika* $2, 9, 11, 20, 91, 111, 313$ *zadovoljava kongruenciju* $(x^e)^d \equiv x(\mathrm{mod}\ N)$ *za primjerice,* $x = 2$*. Tako dobivamo da je tajni eksponent* $d = 313$.

Budući da je tipično da N bude 1024 bitni, da bi se izbjegao ovaj napad d mora biti bar 256 bitni. To nije dobro kod uređaja s niskom snagom poput "smart kartica", gdje bi mali d rezultirali s velikim uštedama. Wiener opisuje niz tehnika koje omogućuju brzo dešifriranje i nisu osjetljivi na njegov napad. Neki od njih su (vidi [1]):

- *veliki e:*
  Umjesto da smanjujemo e modulo $\varphi(N)$, pretpostavimo da koristimo (N, e') za javni ključ, gdje je $e' = e + t\varphi(N)$, za neki veliki t. Eksponent e' može se koristiti umjesto eksponenta e za šifriranje 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 rezultiraju povećanjem vremena šifriranja.

- *Korištenje Kineskog teorema o ostacima*:
  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 dešifriranje šifriranog teksta y može provesti na sljedeći način: najprije računamo $x_p = y^{d_p} \bmod p$ i $x_q = y^{d_q} \bmod q$. Tada koristimo Kineski teorem o ostacima da izračunamo jedinstvenu vrijednost $m \in \mathbb{Z}_N$ koja zadovoljava $x = x_p \mod p$ i $x = x_q \mod q$. Dobiveni otvoreni x zadovoljava $x = y^d \mod N$. Iako su d_p i d_q mali, vrijednost $d \mod \varphi(N)$ može biti velika.

Znamo da Wienerov napad ne djeluje protiv tih metoda, ali ne znamo je li koja od tih metoda sigurna.

3.4 Mali javni eksponent

Korištenje vrlo malog javnog eksponenta e može drastično uštedjeti vrijeme, ali i smanjiti troškove šifriranja. Najmanji mogući javni eksponent e je 3, ali da bi se obranilo od mogućeg napada preporučljiva vrijednost za e je $2^{16}+1$. 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 odabrani $e \leq \varphi(N)$.

Hastadov napad

Još jedan kvar protokola pojavljuje se kada je nekoliko otvorenih tekstova šifrirano s malim javnim eksponentima i različitim modulima. Napadi na te propuste u protokolu često se nazivaju *Hastad emitirani napadi*. Promotrit ćemo dvije vrste napada: *napad na zajednički otvoreni tekst* i *napad na povezane poruke*.

- *Napad na zajednički otvoreni tekst*
  Pad protokola nastaje kada se ista poruka m šifrira s nekoliko različitih javnih ključeva (e, N_i), $i = 1, ..., l$, $l \in \mathbb{N}$ koji imaju isti javni eksponent e i različite module N_i. Napad na ovaj protokol prvi je opisao Hastad 1985. godine.

  Teorem 7 (vidi [4, Theorem 3.2])**.** *Neka su* $((e, N_1), ..., (e, N_l))$, $l \geq e$ *valjani RSA javni ključevi s u parovima relativno prostim modulima, 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* $m < N_0$ *danu s* $c_i = m^e mod N_i$ *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 Kineskog teorema o ostatacima možemo izračunati $x \equiv m^e (mod N)$, 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 izračuni mogu napraviti u polinomijalnom vremenu $\log(N)$ teorem je dokazan. □

  Sada primjerom ilustrirajmo napad na RSA kriptosustav s malim javnim eksponentom e.

  Primjer 11. *Neka tri korisnika koriste različite module*

$$N_1 = 629,\ N_2 = 2173\ i\ N_3 = 1159,$$

dok im je javni eksponent $e = 3$*. Bob želi poslati identičnu poruku* m *tim trima osobama, ali prvo će ju šifrirati i dobiva šifrate*

$$c_1 = 529,\ c_2 = 414\ i\ c_3 = 558.$$

Prilikom slanja poruke, Mario je došao do šifrata te sada želi saznati originalnu poruku. Mario će za početak rješavati sustav linearnih kongruencija koristeći Kineski teorem o ostatacima:

$$\begin{aligned} x &\equiv 529(\text{mod } 629) \\ x &\equiv 414(\text{mod } 2173) \\ x &\equiv 558(\text{mod } 1159). \end{aligned}$$

Na taj način dobije sljedeće kongruencije:

$$2518507x \equiv 529(\text{mod } 629) \tag{8}$$
$$729011x \equiv 414(\text{mod } 2173) \tag{9}$$
$$1366817x \equiv 558(\text{mod } 1159). \tag{10}$$

Kako je $2518507 \equiv 620(\text{mod } 629)$ *rješenje kongruencije* (8)*, dobije se* $620x \equiv 529(\text{mod } 629)$*. Iz* $(620, 629) = 1$*, slijedi da postoje* $u, v \in \mathbb{Z}$ *takvi da je* $620u + 629v = 1$*. Pomoću Euklidovog algoritma dobijemo* $u = -70$*,* $v = 69$*, pa su sva rješenja kongruencije* (8) *dana s* $x_1 \equiv 81(\text{mod } 629)$*.*
Kako je $729011 \equiv 1056(\text{mod } 2173)$ *rješenje kongruencije* (9) *dobije se* $1056x \equiv 414(\text{mod } 2173)$*. Iz* $(1056, 2173) = 1$*, slijedi da postoje* $u, v \in \mathbb{Z}$ *takvi da je* $1056u + 2173v = 1$*. Pomoću Euklidovog algoritma dobijemo* $u = -570$*,* $v = 277$*, pa su sva rješenja kongruencije* (9) *dana s* $x_2 \equiv 877(\text{mod } 2173)$*.*
Kako je $1366817 \equiv 356(\text{mod } 1159)$ *rješenje kongruencije* (10)*, dobije se* $356x \equiv 558(\text{mod } 1159)$*. Iz* $(356, 1159) = 1$*, slijedi da postoje* $u, v \in \mathbb{Z}$ *takvi da je* $356u + 1159v = 1$*. Pomoću Euklidovog algoritma dobijemo* $u = -458$*,* $v = 223$*, pa su sva rešenja kongruencije* (10) *dana s* $x_3 \equiv 542(\text{mod } 2173)$*.*

Odatle slijedi da su sva rješenja sustava kongruencija $(8), (9), (10)$ *dana s*

$$\begin{aligned} x &\equiv 2173 \cdot 1159 \cdot 81 + 629 \cdot 1159 \cdot 877 + 629 \cdot 2173 \cdot 542 \\ &\equiv 1584156528(\text{mod } 1584140903) \\ &\equiv 15625. \end{aligned}$$

To znači da je $x = 15625$. *Sada primjenom Teorema 7. znamo da je* $m = \sqrt[3]{x}$, *odnosno* $m = 25$ *i Mario je došao do izvorne poruke.*

- **Napad na povezane poruke**
  Pad protokola dogodio se kada je nekoliko povezanih otvorenih tekstova šifrirano s malim javnim eksponentom e i drugačijim modulima N_i, $i = 1, ..., l$, $l \in \mathbb{N}$. Otvoreni tekstovi m_i su povezani ako je $m_i = f_i(m)$, za neke poznate polinome f_i. Ovdje je jedino m nepoznati dio svakog otvorenog teksta. Bez dokaza navodimo sljedeći teoerem:

  Teorem 8 (vidi [4, Theorem 2.3.]). *Neka su* $(e_1, N_1, ..., (e_l, N_l))$, *valjani RSA javni ključevi s u parovima relativno prostim modulima,* $N_0 = min\{N_1, ..., N_l\}$ *i* $N = N_1 N_2 \cdots N_l$. *Neka su* $f_1(x) \in \mathbb{Z}_{N_1}[x], ..., f_l(x) \in \mathbb{Z}_{N_l}[x]$ *poznati polinomi. Za bilo koji otvoreni tekst* $m < N_0$, *ako je* $l \geq max_i e_i deg(f_i(x))$ *tada se za dani* $c_i = f_i(m)^{c_i} mod N_i$ *i* (e_i, N_i) $(i = 1, ..., l)$ *otvoreni tekst* m *se može izračunati u polinomijalnom vremenu* $\log(N)$.

U današnje doba sve češće provodimo vrijeme na internetu, neki zbog posla, a neki i zbog zabave. Putem interneta možemo razmijenjivati poruke sa svojim prijateljima, kupovati preko raznih internetskih stranica i slično. Puno toga zahtijeva od nas slanje privatnih informacija, primjerice adrese na kojoj živimo, našeg IBAN-a u banci, itd., a sve to šaljemo potpunim stranicma. Svatko bi volio te informacije sačuvati od neke treće osobe koja bi ih mogla presresti i zloupotrijebiti. RSA kriptosustav poznat je kao jedan od prvih praktičnih kriptosustava s javnim ključem i široko se koristi za siguran prijenos podataka putem interneta. Testiran gotovo 40 godina i nije zabilježen niti jedan uspješan napad. Zbog toga je postao vodećim algoritmom za šifriranje internetskih kreditnih kartica, osiguranje e-pošte i autentifikaciju telefonskih poziva.

Literatura

[1] D. Boneh, *Twenty years of attacs on the RSA cryptosystem*, Notices of the AMS **46**(1999), 203–213.

[2] A. Dujella, *Diofantske aproksimacije i primjene*, PMF-matematički odjel, Sveučilište u Zagrebu, skripta, 2012.

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

[4] M. Jason Hinek, *Criptoanalysis of RSA and its variants*, Chapman & Hall, Boca Raton, FL, USA, 2010.

[5] I. Matić, *Uvod u teoriju brojeva*, Odjel za matematiku, Sveučilište u Osijeku, Osijek, 2015.

[6] E. Milanov, *The RSA algorithm*, Int. J. Comput. Trends Technol. **11**(2009), 177–185.

[7] R. Mollin, *An introduction to cryptography*, 2nd edition, Chapman & Hall, Boca Raton, FL, USA, 2007.

[8] H. Riesel, *Prime Numbers and Computer Methods for Factorization*, Springer-Verlag New York Inc., New York, 2012.

[9] R. L. Rivest, A. Shamir, L. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Commun. ACM **21**(1978), 120–126.

[10] S. Simpson, *Cryptography defined/brief history*, A brief history of cryptography, essay, 1997.

Sažetak

U današnje doba digitalizacije i sve učestalijom komunikacijom putem interneta, pametnih telefona i sl., kriptosustavi imaju sve važniju ulogu u životu svakog pojedinca, jer nam omugućavaju zaštitu naše privatnosti. Jedan od najpoznatijih kriptosustava s javnim ključem je RSA kriptosustav. Temelji se na teškoći faktorizacije velikih prirodnih brojeva. Rad se sastoji od tri cjeline. U prvom poglavlju rada definirali smo kriptosustave s javnim ključem, te primjerom ilustrirali kako taj kriptosustav funkcionira. U drugom poglavlju dana je teorijska osnova za RSA kriptosustav, te njegova implementacija. U ovom je poglavlju također opisana sigurnost i učinkovitost RSA kriptosustava. U posljednjem poglavlju naveli smo neke napade na RSA kriptosustav, te prednosti i nedostatke korištenja malih javnih i privatnih eksponenata.

Ključne riječi

Kriptografija, javni ključ, RSA kriptosustav, kriptoanaliza, javni eksponent, privatni eksponent, Wienerov napad

Summary

Nowadays of digitalization, with more communication over the internet, smarth phones etc., cryptosystems have an important role in life of almost every human. They give us the protection of our privacy. The RSA cryptosystem is one of the most known cryptosystems with public key. It is based on the difficulty of integer factorization. This work is based on three chapters. In the first chapter, we have defined cryptosystem with public keys, and with example illustrated how that cryptosystem works. In the next chapter, we presented some theoretical results which are the base for RSA cryptosystem, and gave some of it's implementations. Moreover, we mentioned security and efficiency of RSA cryptosystem. In the last chapter, we described some attacks on RSA cryptosystem, and mentioned advantages and disadvantages, respectively, of small public and private exponents.

Key words

Cryptography, public key, RSA cryptosystem, cryptanalysis, public exponent, private exponent, Wiener attack

Životopis

Zovem se Ines Spaić, rođena sam 5. listopada 1991. godine u Koperu. Osnovnu školu Viktora Cara Emina u Umagu upisala sam 1998. godine. Zbog seljenja s obitelji promijenila sam tri osnovne škole. Osnovnoškolsko obrazovanje završavila sam 2006. godine u OŠ Stari Jankovci u Starim Jankovcima. Nakon toga upisala sam Gimnaziju Matije Antuna Reljkovića u Vinkovcima. 2010. godine upisala sam Integrirani sveučilišni nastavnički studij matematike i informatike na Odjelu za matematiku Sveučilišta J. J. Strossmayera u Osijeku.