Osnovni algoritmi teorije brojeva u kriptografiji

Bertok, Pamela

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

Pamela Bertok

Algoritmi teorije brojeva u kriptografiji

Diplomski rad

Osijek, 2021.

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

Pamela Bertok

Algoritmi teorije brojeva u kriptografiji

Diplomski rad

Mentor: prof. dr. sc. Ivan Matić

Osijek, 2021.

Sadržaj

Uvod

Problemom sigurne razmjene poruka između dviju udaljenih osoba bavili su se već stari Egipćani prije više od 3000. godina. Razvojem pisma i tehnologije postajalo je lakše uspostaviti sigurnu komunikaciju. Glavna ideja nije se promijenila do danas, a to je naći algoritam koji će efikasno i s visokom razinom sigurnosti omogućiti dvjema osobama prenošenje poruka koje samo one razumiju.

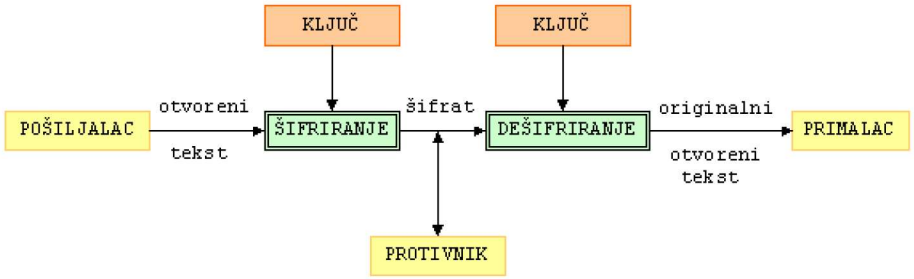
U prvom poglavlju ovog diplomskog rada upoznajemo se s osnovnim pojmovima kriptografije, podjelom kriptosustava te napadima na iste. Nadalje, opisani su pojedini kriptosustavi kao što su *Šifra pomaka, Afina šifra, Supstitucijske šifre, Vigenereova i Hillova šifra* i njihova kriptoanaliza. Sljedeće poglavlje govori o osnovnim algoritmima teorije brojeva koji se primjenjuju u kriptografiji i njihovoj složenosti; Euklidov algoritam, verižni razlomci i kvadratne kongruencije. Na kraju tog poglavlja reći ćemo nešto o faktorizaciji i prostosti kao glavnim problemima kriptografije javnog ključa, o kojoj ćemo govoriti u trećem poglavlju. Također, treće poglavlje prikazuje dva najpoznatija kriptosustava s javnim ključem; RSA i Rabinov kriptosustav.

1 Klasična kriptografija

Kriptografija je znanstvena disciplina koja proučava metode za slanje poruka koje može pročitati samo osoba kojoj su one namijenjene. Sama riječ kriptografija je kombinacija dviju grčih riječi; "krypto", što znači skriven i "graphene", što znači pisati. Stoga riječ kriptografija možemo prevesti kao "tajnopis". Zadatak kriptografije je omogućavanje dvjema osobama koje nazivamo *pošiljalac* i *primalac* komunikaciju nesigurnim kanalom tako da treća osoba (*protivnik*), koja može nadzirati taj komunikacijski kanal, ne može razumjeti njihove poruke.

1.1 Osnovni pojmovi

Pošiljalac šalje primaocu poruku koju nazivamo *otvoreni tekst*, a to može biti tekst na odabranom jeziku, numerički podaci ili bilo što drugo. Postupak transformacije otvorenog teksta koristeći unaprijed dogovoreni *ključ* naziva se *šifriranje*, a dobiveni rezultat je *šifrat*. Tijekom slanja šifrata putem nesigurnog kanala, protivnik prisluškujući može saznati sadržaj istog, ali ne može odrediti otvoreni tekst. Međutim, primalac koji zna ključ, može odrediti sadržaj otvorenog teksta te taj postupak nazivamo *dešifriranje*.



Slika 1: *Shematski prikaz klasične kriptografije ([3]).*

Kriptoanaliza je znanstvena disciplina koja se bavi proučavanjem postupaka za čitanje skrivenih poruka bez poznavanja ključa, dok se grana znanosti koja se sastoji od kriptografije i kriptoanalize naziva *kriptologija*. Pretpostavimo da pošiljalac želi poslati poruku primaocu preko nesigurnog kanala. Prvo odabiru ključ K. Svaki znak otvorenog teksta šifrira se koristeći pravilo šifriranja e_K određeno unaprijed zadanim ključem K. Kad primalac

primi šifrat, on ga dešifrira koristeći pravilo dešifriranja d_K te dobiva izvorni otvoreni tekst.

Definicija 1. ([3]) *Kriptosustav* je uređena petorka $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$, gdje je $\mathcal{P}$ konačan skup osnovnih elemenata otvorenog teksta, $\mathcal{C}$ je konačan skup osnovnih elemenata šifrata, $\mathcal{K}$ je konačan skup ključeva, $\mathcal{E}$ je konačan skup funkcija šifriranja i $\mathcal{D}$ konačan skup funkcija dešifriranja. Za svaki $K \in \mathcal{K}$ postoji funkcija šifriranja $e_K \in \mathcal{E}$ i odgovarajuća funkcija dešifriranja $d_K \in \mathcal{D}$. Pritom su $e_K : \mathcal{P} \to \mathcal{C}$ i $d_K : \mathcal{C} \to \mathcal{P}$ funkcije takve da je $d_K(e_K(x)) = x$, za svaki $x \in \mathcal{P}$.

Kriptosustave klasificiramo na sljedeći način:

1. **Tip operacija koje se koriste pri šifriranju**
   a) Supstitucijske šifre
   b) Transpozicijske šifre
2. **Način na koji se obrađuje otvoreni tekst**
   a) Blokovne šifre
   b) Protočne šifre
3. **Tajnost i javnost ključeva**
   a) Simetrični kriptosustavi
   b) Kriptosustavi s javnim ključem

Općenito se u kriptoanalizi pretpostavlja da kriptoanalitičar zna koji se kriptosustav koristi. To se naziva *Kerckhoffsovo načelo*. Razlikujemo 4 osnovna modela napada na kriptosustave:

1. **Samo šifrat**
2. **Poznati otvoreni tekst**
3. **Odabrani otvoreni tekst**
4. **Odabrani šifrat**
5. **Potkupljivanje, ucjena**

1.2 Šifra pomaka

Za početak, opisat ćemo *Šifru pomaka* koja se temelji na modularnoj aritmetici. Radi razumijevanja, definirat ćemo neke osnovne pojmove.

Definicija 2. ([2]) Neka su a, b i m cijeli brojevi te neka je $m > 0$. Tada pišemo $a \equiv b \pmod{m}$ ukoliko m dijeli razliku $b - a$. Izraz $a \equiv b \pmod{m}$ nazivamo *kongruencija*, a čitamo "a je kongruentan b modulo m". Broj "m" naziva se *modul*.

Pretpostavimo da dijelimo dva cijela broja a i b sa m te promatramo cjelobrojne koeficijente i ostatke. Imamo $a = q_1 m + r_1$ i $b = q_2 m + r_2$, pri čemu su r_1 i r_2 ostaci takvi da vrijedi $0 \leqslant r_1, r_2 \leqslant m - 1$. Primjetimo da je $a \equiv b \pmod{m}$ ako i samo ako je $r_1 = r_2$. Zaključujemo, ako dva cijela broja daju isti ostatak pri dijeljenju s m, tada su a i b kongruentni modulo m i pišemo $a \equiv b \pmod{m}$.

Primjer 1. Da bismo izračunali 95 mod 13 pišemo: $95 = 13 \cdot 7 + 4$. Dakle vrijedi $0 \leqslant 4 \leqslant 13$, pa je $95 \pmod{13} = 4$.

Napomena 1. Budući da vrijedi

$$m \mid (b - a) \iff -m \mid (b - a)$$

promatramo samo pozitivne module m.

Skup $\{0, 1, 2, \ldots, 25\}$ označavat ćemo sa $\mathbb{Z}_m$ i neka su operacije zbrajanja, oduzimanja i množenja na njemu definirane jednako kao u skupu $\mathbb{Z}$, ali na način da se rezultat, ukoliko nije iz skupa $\{0, 1, 2, \ldots, 25\}$, na kraju zamijeni s njegovim ostatkom pri dijeljenju s 26. Taj skup, uz operacije $+_{26}$ i $\cdot_{26}$, nazivamo *prsten*. Možemo uvesti jednu oznaku: ako dva cijela broja a i b daju isti ostatak pri dijeljenju s 26, to zapisujemo s $a \equiv b \pmod{m}$ i čitamo "a i b su kongruentni modulo 26".

Definirajmo sada *Šifru pomaka*:

Definicija 3. ([4]) Neka je $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}$. Za $0 \leqslant K \leqslant 25$ definiramo:

$$e_K(x) = (x + K) \bmod 26$$
$$\text{i}$$
$$d_K(y) = (y - K) \bmod 26.$$

Definirana je na $\mathbb{Z}_{26}$ jer u engleskoj abecedi postoji 26 slova. Zato je potrebno navesti korespondenciju između abecednih znakova i ostataka modulo 26 na sljedeći način:

A	B	C	D	E	F	G	H	I	J	K	L	M
0	1	2	3	4	5	6	7	8	9	10	11	12

N	O	P	Q	R	S	T	U	V	W	X	Y	Z
13	14	15	16	17	18	19	20	21	22	23	24	25

U ovoj su šifri slova osnovni elementi otvorenog teksta, a ključ K označava broj mjesta za koje ih pomičemo udesno.

Napomena 2. Za ključ $K = 3$, kriptosustav se često naziva Cezarovom šifrom. Navodno je rimski vojskovođa Gaj Julije Cezar koristio u komunikaciji sa svojim prijateljima šifru u kojoj su slova otvorenog teksta zamijenjena slovima koja su tri mjesta dalje od njih u abecedi($A \mapsto D, B \mapsto E, C \mapsto F$, i tako dalje).

Ilustrirajmo to na primjeru.

Primjer 2. Treba dekriptirati šifrat LBYKTIHFTDT dobiven pomoću *Šifre Pomaka*.
Kako je prostor ključeva vrlo mali, možemo primjenjujući grubu silu, odnosno ispitujući sve moguće ključeve, odrediti naš ključ K.
Za K=0 dobivamo: L B Y K T I H F T D T
Za K=1 dobivamo: M C Z L U J I G U E U
⋮
Za K=7 dobivamo: S I F R A P O M A K A Dakle, ključ je 7, a otvoreni tekst je ŠIFRA POMAKA.

1.3 Afina šifra

Mogli smo primjetiti da *Šifra pomaka* nije sigurna. Naime, nju možemo kriptoanalizirati iscrpnim pretraživanjem svih mogućih ključeva (ukupno 26). Dakle, nužan uvjet da bi kriptosustav bio siguran je da je takvo pretraživanje

ključeva neizvedivo, odnosno prostor ključeva $\mathcal{K}$ mora biti velik. Kako bismo dobili barem malo sigurniju šifru, promatrat ćemo funkcije šifriranja koje imaju više od jednog parametra, a najjednostavnija takva je afina funkcija $e(x) = ax + b$. Međutim, ovdje moramo postaviti ograničenje; naime, na promatranoj domeni ova funkcija ne mora biti injekcija, odnosno ne mora imati inverz. Kako bismo dokazali injektivnost, dovoljno je da za svaki $y \in \mathbb{Z}_{26}$ kongruencija $ax + b \equiv y \pmod{m}$ ima jedinstveno rješenje po x.

Teorem 1. ([4]) Kongruencija $ax \equiv b \ (mod\ m)$ ima jedinstveno rješenje $x \in \mathbb{Z}_m$, $\forall y \in \mathbb{Z}_m$ ako i samo ako je $(a, m) = 1$.

Znamo da je $26 = 2 \cdot 13$, a $(a, 26) = 1$, pa zaključujemo da su vrijednosti parametra a iz skupa $\{1, 3, 5, 7, 9, 11, 15, 17, 19, 21, 23, 25\}$, dok parametar b može biti bilo koji element skupa $\mathbb{Z}_{26}$. Dakle, ukupan broj ključeva afine šifre je $12 \cdot 26 = 312$, što je i dalje premalo da bi kriptosustav bio siguran.
Prije nego formalno definiramo *Afinu Šifru*, potrebno je definirati još jedan pojam.

Definicija 4. ([4]) Neka je $a \in \mathbb{Z}_m$. Multiplikativni inverz od a *modulo* m, u oznaci $a^{-1} \bmod m$, je element $a' \in \mathbb{Z}_m$ za koji vrijedi $aa' \equiv a'a \equiv 1 \pmod{m}$.

Budući da broj 26 nije prost broj, nemaju svi elementi iz $\mathbb{Z}_m$ multiplikativni inverz. Imaju ga samo oni elementi koji su relativno prosti s 26. Navedimo te brojeve i njihove inverze:

a	1	3	5	7	9	11	15	17	19	21	23	25
a'	1	9	21	15	3	19	7	23	11	5	17	25

Konačno, definirajmo *Afinu Šifru*.

Definicija 5. ([4]) Neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$, te neka je $\mathcal{K} = \{(a, b) \in \mathbb{Z}_{26} \times \mathbb{Z}_{26} : (a, 26) = 1\}$. Za $K = (a, b) \in \mathcal{K}$ definiramo

$$e_K(x) = (ax + b) \bmod 26$$
$$\text{i}$$
$$d_K(y) = a^{-1}(y - b) \bmod 26.$$

A	I	O	E	N	S	R	J	T	U	D
115	98	90	84	66	56	54	51	48	43	37

K	V	L	M	P	C	Z	G	B	H	F
36	35	33	31	29	28	23	16	15	8	3

Tablica 1: Frekvencija slova u hrvatskom jeziku (u promilima).

E	T	A	O	I	N	S	H	R	D	L	C	U
127	91	82	75	70	67	63	61	60	43	40	28	28

M	W	F	G	Y	P	B	V	K	J	Q	X	Z
24	23	22	20	20	19	15	10	8	2	1	1	1

Tablica 2: Frekvencija slova u engleskom jeziku (u promilima).

1.4 Supstitucijske šifre

Šifra pomaka i *Afina šifra* specijalni su slučajevi *Supstitucijske šifre* koju definiramo na sljedeći način:

Definicija 6. ([4]) Neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}$. Prostor ključeva $\mathcal{K}$ sastoji se od svih permutacija skupa $\{0, 1, 2, \ldots, 25\}$. Za svaku permutaciju $\pi \in \mathcal{K}$ definiramo

$$e_\pi(x) = \pi(x), \quad d_\pi(x) = \pi^{-1}(y),$$

gdje je π^{-1} inverzna permutacija od π.

Ovdje za alfabet šifrata možemo izabrati bilo koju permutaciju slova A, B, ..., Z. Primjetimo da ovdje imamo $26! = 1 \cdot 2 \cdots 26 \approx 4 \cdot 10^{26}$ mogućih ključeva. S obzirom na to, supstitucijsku šifru lakše je dešifrirati koristeći statistička svojstva jezika na kojem je pisan otvoreni tekst. Najčešće koristimo *analizu frekvencije slova.* Ova metoda zasnovana je na broju pojavljivanja svakog slova u šifratu te se ti brojevi uspoređuju s poznatim podatcima o učestalosti slova u jeziku kojim je pisan otvoreni tekst. Osim toga, korisno je promatrati *bigrame* i *trigrame.*
Tablica 1 i Tablica 2 prikazuju nam frekvencije slova u hrvatskom i engleskom jeziku.

Ako unutar šifrata primjetimo pojavljivanje neke riječi, tada se to naziva *Cezarova šifra s ključnom riječi.*

1.5 Vigenèreova šifra

Šifre koje smo do sada promatrali zovu se *monoalfabetske*, odnosno svakom slovu otvorenog teksta odgovara jedinstveno slovo šifrata. Sada predstavljamo *Vigenèreovu šifru*, šifru koju ubrajamo u *polialfabetske šifre*, kod koje se svako slovo otvorenog teksta može preslikati u jedno od m slova, pri čemu je m duljina ključa, a u koje slovo će se preslikati ovisi o položaju slova unutar otvorenog teksta. Definiramo ju na sljedeći način:

Definicija 7. ([4]) Neka je $m \in \mathbb{N}$. Definiramo $\mathcal{P} = \mathcal{C} = \mathcal{K} = \mathbb{Z}_{26}^m$. Za ključ $K = (k_1, k_2, \ldots, k_m)$ definiramo

$$e_K(x_1, x_2, \ldots, x_m) = (x_1 +_{26} k_1, x_2 +_{26} k_2, \ldots, x_m +_{26} k_m),$$
$$d_K(y_1, y_2, \ldots, y_m) = (y_1 -_{26} k_1, y_2 -_{26} k_2, \ldots, y_m -_{26} k_m).$$

Pogledajmo na primjeru o čemu se radi.

Primjer 3. Pretpostavimo da je $m = 6$ i ključna riječ MARIJA. To odgovara numeričkom ekvivalentu $K = (12, 0, 17, 8, 9, 0)$. Neka je otvoreni tekst niz

kriptografijajeznanstvenadisciplina.

10	17	8	15	19	14	6	17	0	5	8	9
12	0	17	8	9	0	12	0	17	8	9	0
22	17	25	23	3	14	16	17	17	13	17	9

0	9	4	25	13	0	13	18	19	21	4	13	0
12	0	17	8	9	0	12	0	17	8	9	0	12
12	9	21	8	22	0	25	18	11	4	13	13	12

3	8	18	2	8	15	11	8	13	0
0	17	8	9	0	12	0	17	8	9
3	25	1	11	8	2	11	25	21	9

Abecedni ekvivalent šifratu tada bi bio WRZXDOQRRNRJMJVIWAZSLENNMDZBLICLZVJ.

Prema podjeli šifri s obzirom na način obrade otvorenog teksta, ovu šifru shvaćamo kao primjer blokovne šifre. Osim ove, postoje i druge varijante

Vigenèreove šifre, a jedna od njih je ona s *autoključem*. Kod nje otvoreni tekst generira ključ, odnosno originalni ključ koristi se samo za šifriranje prvog bloka od m slova, a za daljnje šifriranje koristi se prethodni blok otvorenog teksta. Zbog toga ovu šifru ubrajamo u protočne šifre.

Prvi korak kriptoanalize je određivanje duljine ključne riječi. Prva metoda koju ćemo spomenuti naziva se *Kasiskijev test*. Tražimo parove istih odsječaka duljine veće od 2 te bilježimo udaljenosti između njihovih početnih položaja. Dobivamo udaljenosti $d_1, d_2, d_3, \ldots$ i možemo pretpostaviti da m dijeli barem većinu tih udaljenosti. Kada smo odredili m, dobivamo sličnu situaciju kao kod supstitucijske šifre. Međutim, situacija je ipak malo kompliciranija nego kod nje, pa zato opisujemo drugu metodu, tzv. *"indeks koincidencije"*.

Definicija 8. ([3]) Neka je $x = x_1x_2\cdots x_n$ niz od n slova. *Indeks koincidencije* od x označavamo s $I_c(x)$ te definiramo kao vjerojatnost da su dva slučajno odabrana elementa iz x jednaka.

Ako s $f_0, f_1, \ldots, f_{25}$ označimo redom frekvencije slova A, B, $\ldots$, Z u x, onda dva elementa iz niza možemo izabrati na $\frac{n(n-1)}{2}$ načina, a za svaki $i = 0, 1, \ldots, 25$ postoji $\frac{f_i(f_i-1)}{2}$ načina dvostrukog odabira i-tog slova u abecedi. Dakle, vrijedi:

$$I_c(x) = \frac{\sum_{i=0}^{25}(f_i(f_i-1))}{n(n-1)}.$$

Ako je n dovoljno velik, a x predstavlja tekst na hrvatskom jeziku, vrijedi:

$$I_c(x) \approx \sum_{i=0}^{25} p_i^2 \approx 0.064.$$

gdje su p_i očekivane vrijednosti pojavljivanja slova A, B,$\ldots$, Z u hrvatskom jeziku. Pretpostavimo da imamo šifrat $y = y_1y_2\cdots y_n$ koji je dobiven Vigenereovom šifrom. Rastavimo ga na m podnizova $z_1, z_2, \ldots, z_m$ tako da y napišemo u matricu dimenzije $m \times \frac{n}{m}$. Ako je m jednak duljini ključa, onda elemente istog retka šifriramo pomoću istog ključa i $I_c(z_i)$ trebao bi biti približno 0.064.
Sada nas zanima kako odrediti ključnu riječ ako znamo njezinu duljinu. U tu svrhu koristimo *međusobni indeks koincidencije*.

Definicija 9. ([3]) Neka su $x = x_1x_2\cdots x_n$ i $y = y_1y_2\cdots y_{n'}$ dva niza od n, odnosno n' slova. *Međusobni indeks koincidencije* od x i y, u oznaci $MI_c(x, y)$, definiramo kao vjerojatnost da je slučajno odabrani element od x jednak slučajno odabranom elementu od y. Ako frekvencije od A, B, C,..., Z u x i y označimo s $f_0, f_1, \ldots, f_{25}$, odnosno $f'_0, f'_1, \ldots, f'_{25}$, onda je:

$$MI_c = \frac{\sum_{i=0}^{25} f_i f'_i}{nn'}.$$

Ako znamo na kojem je jeziku pisan otvoreni tekst, do ključne riječi možemo doći računajući $MI_c(x, z_j^g)$, gdje je x niz koji odgovara tekstu na jeziku otvorenog teksta, a z_j^g predstavlja šifrirani niz z_j slovima A, B, ..., Z (tj. pomakom za 0, 1, 2, ..., 25 mjesta). Ako nam je poznato da je otvoreni tekst pisan hrvatskim jezikom, relativne su frekvencije $\frac{f_i}{n}$ za x približno jednake p_i, pa vrijedi

$$MI_c(x, z_j^g) \approx \frac{\sum_{i=0}^{25} p_i f'_{i-g}}{n'}.$$

Kako bi odredili j-to slovo k_j ključne riječi K, za $0 \leq g \leq 25$ izračunamo

$$M_g = \frac{\sum_{i=0}^{25} p_i f'_{i-g}}{n'}$$

te odredimo h takav da je $M_h = max\{M_g : 0 \leq g \leq 25\}$, te stavimo $k_j \equiv -h($ (mod 26)).

1.6 Hillova šifra

Riječ je o kriptosustavu kod kojeg se m uzastopnih slova otvorenog teksta zamjenjuje s m uzastopnih slova u šifratu. Kriptosustav je definiran na sljedeći način:

Definicija 10. ([4]) Neka je m fiksan prirodan broj. Neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_{26}^m$, te

$$\mathcal{K}=\{m \times m \text{ invertibilne matrice nad } \mathbb{Z}_{26}\}.$$

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

$$e_K(x) = xK, \qquad d_K(y) = yK^{-1}$$

pri čemu promatramo operacije u $\mathbb{Z}_{26}$.

Hillov kriptosustav s 3×3 matricama skriva sve informacije o frekvencijama slova i bigrama. Za $m \geq 5$ je ovaj kriptosustav gotovo siguran na napad samo šifrat, ali ne i na napade pomoću poznatog i odabranog otvorenog teksta.
Pretpostavimo da kriptoanalitičar zna vrijednost m te da ima barem m različitih parova m-torki $x_i = (x_{i1}, \ldots, x_{im})$, $y_i = (y_{i1}, \ldots, y_{im})$, takvih da je $y_i = e_K(x_i), i = 1, \ldots, m$. Definiramo dvije matrice dimenzije $m \times m$: $X = [x_{ij}], Y = [y_{ij}]$. Tada vrijedi $Y = XK$, gdje je K $m \times m$ matrica koja predstavlja ključ. Ukoliko je X invertibilna matrica, možemo izračunati $K = X^{-1}Y$. Ako X nije invertibilna matrica, treba pokušati s nekim drugim skupom od m parova *otvoreni tekst - šifrat*.
Ukoliko ne znamo m, te pretpostavljamo da nije jako velik, možemo probati redom s $m = 2, 3, \ldots$, dok ne nađemo ključ.

2 Osnovni algoritmi teorije brojeva

Kada govorimo o problemima u matematici, uglavnom ih karakteriziramo kao "lake" ili "teške", no vrlo često to nije tako jednostavno.

2.1 Složenost algoritma

Algoritam se definira kao procedura koja rješava određenu klasu problema tako da za neke ulazne podatke daje izlazne podatke u konačnom vremenu. Za određenu klasu problema postoji nekoliko algoritama pomoću kojih se isti rješavaju, ali bitna razlika je u tome što su neki efikasniji od drugih. Zanimat će nas veličina ulaznih podataka, koliko koraka je potrebno za izvršavanje algoritma, a ponekad i koliko prostora (memorije) zahtjeva. Svaki korak predstavlja jednu od elementarnih operacija na nulama i jedinicama (bitovima). Broj tih koraka u najlošijem slučaju govori nam o složenosti algoritma.
Točan broj koraka vrlo često je teško, pa čak i nemoguće odrediti, stoga kod analize složenosti nekog algoritma koristimo sljedeću definiciju:

Definicija 11. ([3]) Neka su $f, g : \mathcal{S} \to \mathbb{R}$ dvije funkcije. Tada pišemo:

1. $f(n) = O(g(n))$ ako postoje $B, C > 0$ takvi da je $|f(n)| \leqslant C|g(n)|$ za sve $n \in S$ takve da je $n > B$,
2. $f(n) \sim g(n)$ ako je $\lim\limits_{n\to\infty} \dfrac{f(n)}{g(n)} = 1$,
3. $f(n) = o(g(n))$ ako je $\lim\limits_{n\to\infty} \dfrac{f(n)}{g(n)} = 0$.

Kada govorimo o efikasnosti, odnosno neefikasnosti algoritama, razlikujemo *polinomijalan* i *eksponencijalan*. Definirajmo ih.

Definicija 12. ([1]) Polinomijalan algoritam je algoritam čiji je broj operacija u najlošijem slučaju funkcija oblika $O(n^k)$, gdje je n duljina ulaznog podatka, a k je konstanta. Za algoritam koji nije polinomijalan, kažemo da je eksponencijalan.

Duljina ulaznog podatka predstavlja broj bitova koji su potrebni za prikaz podatka. Ulazni podaci uglavnom su prirodni brojevi N, pa je duljina tog broja $\lfloor \log_2 N \rfloor + 1$.

Dakle, polinomijalni algoritmi predstavljaju zaista efikasne algoritme, a eksponencijalni neefikasne. U primjenama je od velike važnosti prosječna složenost algoritma.

Definicija 13. ([3]) Subeksponencijalan algoritam je algoritam čija je složenost funkcija oblika $O(e^{o(n)})$, gdje je n duljina ulaznog podatka.

2.2 Osnovne računske operacije s prirodnim brojevima

Kada zbrajamo dva prirodna broja x i y, zbrajamo bit po bit i gledamo je li se dogodio prijenos ili ne. Kod operacije oduzimanja postupamo analogno. Pogledajmo rezultat zbrajanja dvaju brojeva.
Brojevi x i y su brojevi zapisani u nekoj bazi b i oblika su

$$x = (x_n, \ldots, x_1, x_0)_b, \qquad y = (y_n, \ldots, y_1, y_0)_b.$$

Njihov zbroj tada je jednak

$$x + y = w = (w_{n+1}, w_n, \ldots, w_1, w_0)_b.$$

Algoritam za zbrajanje dvaju brojeva je sljedeći:
Algoritam za zbrajanje
$c = 0$
for $(0 \leqslant i \leqslant n)$ do
    if$(x_i + y_i + c < b)$ then $w_i = x_i + y_i + c; c = 0$
        else $w_i = x_i + y_i + c - b; c = 1$
$w_{n+1} = c$

Zbrajanje i oduzimanje takva dva prirodna broja obavlja se u $O(\ln N)$ bitnih operacija, dakle polinomijalni su. Stoga nema potrebe tražiti nešto "bolje", odnosno optimizirati.

Sada nas zanima kako izračunati produkt dvaju brojeva. Ako su x i y opet brojevi oblika $x = (x_n, \ldots, x_1, x_0)_b$, $y = (y_t, \ldots, y_1, y_0)_b$, onda je njihov produkt

$$x \cdot y = (w_{n+t+1}, \ldots, w_0)_b.$$

Dakle, ako su x_j i y_i znamenke u bazi b, onda produkt $x_j \cdot y_i$ možemo zapisati kao $(uv)_b$, gdje su u i v znamenke u bazi b. Pogledajmo kako radi algoritam

za najjednostavnije "školsko" množenje:

Algoritam za "školsko" množenje

for $(0 \leqslant i \leqslant n-t)$ do q_j=0
while $(x \geqslant yb^{n-t})$ do $q_{n-t} = q_{n-t} + 1; x = x - yb^{n-t}$
for $(n \geqslant i \geqslant t-1)$ do
    if$(x_i = y_t)$ then $q_{i-t-1} = b - 1$
        else $q_{i-t-1} = \lfloor (x_i b + x_{i-1})/y_t \rfloor$
    while $(q_{i-t-1}(y_t b + y_{t-1}) > x_i b^2 + x_{i-1} b + x_{i-2})$ do
        $q_{i-t-1} = q_{i-t-1} - 1$
    $x = x - q_{i-t-1} y b^{i-t-1}$
    if$(x < 0)$ then $x = x + yb^{i-t-1}; q_{i-t-1} = q_{i-t-1} - 1$
$r = x$

Kod dijeljenja ovakvih dvaju brojeva, uz pretpostavku da je $n \geqslant t \geqslant 1$, tražimo kvocijent $q = (g_{n-t}, \ldots, q_0)_b$ i ostatak $r = (r_t, \cdots, r_0)_b$ takve da je $x = qy + r$, pri čemu je $0 \leqslant r < y$. Takvo dijeljenje nazivamo dijeljenje s ostatkom.

Jasno je da je složenost "školskog" množenja i dijeljenja $O(\ln^2 N)$. Sad se možemo pitati može li "bolje" od ovoga. Postoje algoritmi za množenje i dijeljenje koji su efikasniji od navedenih algoritama. Takvi algoritmi imaju složenost $O(\ln N(\ln\ln N)(\ln\ln\ln N))$. Budući da je dio $(\ln\ln N)(\ln\ln\ln N)$ zanemariv u odnosu na $\ln N$, odnosno da se radi o sporo rastućoj funkciji, onda to možemo zapisati u obliku:

$$\ln N(\ln\ln N)(\ln\ln\ln N) = O((\ln N)^{1+\varepsilon})$$

za prozvoljan $\varepsilon > 0$.

Iz toga možemo zaključiti da je množenje tek nešto složenije od zbrajanja. No, to vrijedi samo ako imamo brojeve od nekoliko tisuća znamenaka. Međutim, za naše potrebe, odnosno za primjene u kriptografiji, postoje algoritmi koji su bolji za množenje prirodnih brojeva od onog najjednostavnijeg množenja. Spomenut ćemo tzv. *Karacubinu metodu*, u kojoj se promatraju dva parna, $2n$-bitna broja x i y. Prilikom množenja tih dvaju brojeva, koji su prethodno zapisani kao zbroj lijeve i desne polovice svakoga od njih, i potrebnim raspisivanjem, došlo se do zaključka da se to množenje može svesti na računanje 3 n-bitna produkta. Složenost takvog algoritma je $O(n^{log_2 3})$,

odnosno računanje produkta brojeva x i y provodi se uz $O((\ln N)^{log_2 3})$ bitnih operacija, što je bolje nego $(\ln N)^2$, dakle odgovor na pitanje može li efikasnije od najjednostavnijeg "školskog" množenja je potvrdan.

Kada budemo govorili o šifriranju i dešifriranju kriptosustava s javnim ključem, vidjet ćemo da se većina operacija provodi u prstenu $\mathbb{Z}_m$. Zbrajanje je vrlo jednostavno. Naime, ako su x i y cijeli brojevi takvi da je $0 \leqslant x, y \leqslant m$, onda vrijedi

$$x +_m y = \begin{cases} x + y, & \text{ako je } x + y < m \\ x + y - m, & \text{ako je } x + y \geqslant m. \end{cases}$$

Kada je riječ o množenju, ono nije tako jednostavno u $\mathbb{Z}_m$. Razlog tomu je što osim običnog množenja još radimo i redukciju po modulu, odnosno dijelimo s ostatkom, a to je nešto složenije. Dakle, trebali bismo prvo računati produkt dva broja, a zatim ostatak pri dijeljenju tog produkta sa m. Postoje razna poboljšanja te metode, a jedna od najpoznatijih je *Montgomeryjeva redukcija* čija je glavna ideja izbjeći dijeljenje s ostatkom i svesti na dijeljenje s potencijom baze. Ono što će nama biti od važnosti je tzv. *modularno potenciranje*, jer će se upravo to koristiti u primjenama u kriptografiji javnog ključa. Takvo potenciranje je specijalan slučaj potenciranja. Ono što odmah možemo zaključiti je da će računanje x^n na način da množimo $x \cdot x \cdot x \cdots x$ $n-1$ puta biti neefikasno i riječ je o eksponencijalnom algoritmu. Efikasnija metoda je binarna metoda.

Primjer 4. Želimo izračunati 3^{13}. Prvo što moramo napraviti je broj 13 zapisati u binarnom brojevnom sustavu: $13 = 1101_2$. Sada prateći pozicije jedinica dobivamo sljedeći zapis: $3^{13} = 3 \cdot (3^2)^2 \cdot ((3^2)^2)^2$.

Pogledajmo na koji način radi algoritam.

Binarna metoda s desna na lijevo

$z = 1; y = x$
for $(0 \leqslant i \leqslant d-1)$ do
    if $(n_i = 1)$ then $z = z \cdot y$
    $y = y^2$
$z = z \cdot y$

Dakle, algoritam prolazi kroz petlju onoliko puta koliko ima binarnih znamenki. Do množenja će doći ako je binarna znamenka jednaka 1, a ako

nije taj korak preskačemo. Osim na ovaj način, možemo krenuti s lijeva na desno te će ta metoda biti brža ako je x mali broj, što je u primjenama često slučaj.
Složenost ovog algoritma je $O(\ln N \cdot \ln^2 m)$, pri čemu uzimamo da je složenost množenja i dijeljenja brojeva manjih od m jednaka $O(\ln^2 m)$. Postoje neka poboljšanja, a jedno od njih je grupiranje od po dvije znamenke, odnosno rad sa bazom 4.

Kada pogledamo primjenu u kriptografiji, često ćemo imati slučaj da su baza ili eksponent fiksni brojevi, a onda je moguće i dodatno poboljšati situaciju:
1° *Fiksna baza*
Unaprijed računamo vrijednosti $x_{ij} = x^{j \cdot b^i}$, gdje je b baza. Tada x^n računamo kao $x^n = \prod_{i=0}^{d} x_i n_i$. Što veću bazu uzmemo, to će broj operacija biti veći.

2° *Fiksan eksponent*
U ovom slučaju koristi se tzv. *lanac zbrojeva.* To je niz brojeva $u_0, u_1, \ldots, u_s$ s pridruženim nizom $w_1, w_2, \ldots, w_s$ parova $w_i = (i_1, i_2)$ sa svojstvom da je $u_0 = 1, u_s = u, u_i = u_{i_1} + u_{i_2}$ za $1 \leqslant i \leqslant s$.

Primjer 5. Pogledajmo lanac zbrojeva za $n = 17$ i $w = (1, 2, 3, 6, 9, 15)$:
$u_0 = 1$
$u_1 = 1 + 1 = 2$
$u_2 = 2 + 1 = 3$
$u_3 = 3 + 3 = 6$
$u_4 = 6 + 3 = 9$
$u_5 = 9 + 6 = 15$
$u_6 = 15 + 2 = 17.$

2.3 Euklidov algoritam

Sljedeći algoritam ima vrlo veliku primjenu općenito, pa tako i u kriptografiji. To je algoritam koji omogućuje nalaženje najvećeg zajedničkog djelitelja dvaju brojeva a i b u oznaci (a, b). Općenito, nalaženje najvećeg zajedničkog djelitelja brojeva a i b može se svesti na faktorizaciju, međutim

to vrlo često nije efikasno, pa iz tog razloga ćemo trebati nešto bolje. Pogledajmo kako radi Euklidov algoritam:

Euklidov algoritam

while $(b > 0)$ do

$(a, b) = (b, a \pmod b)$

return a

Kad algoritam raspišemo po koracima i ako se pogleda koliko koraka će biti potrebno za određivanje (a, b) u najlošijem slučaju, dolazi se do zaključka da ovaj algoritam zahtjeva $O(\ln N)$ koraka i složenosti je $O(\ln^3 N)$. Ovo se može poboljšati do složenosti $O(\ln^2 N)$, iz čega zaključujemo da je Euklidov algoritam vrlo efikasan.

Kao što je i ranije rečeno, primjene Euklidovog algoritma su brojne. Kada bude riječi o kriptografiji javnog ključa, susrest ćemo se s vrlo bliskim problemom, a to je određivanje modularnog inverza. Ovi problemi povezani su idućim rezultatom (dokaz možemo pronaći u ([3]):

Teorem 2. Postoje cijeli brojevi x, y takvi da je $ax + by = (a, b)$.

Ako su a i b relativno prosti, onda vrijedi $ax + by = 1$.
Osim pronalaska broja (a, b), ovaj algoritam može se upotrijebiti i za traženje cijelih brojeva x i y takvih da je $ax + by = (a, b)$, odnosno za rješavanje linernih diofantskih jednadžbi. To se onda naziva *prošireni Euklidov algoritam.*

Euklidov algoritam možemo pronaći i kod rješavanja sustava linearnih kongruencija, o čemu govori *Kineski teorem o ostacima* ([4]). Za početak ćemo iskazati teorem:

Teorem 3. (Kineski teorem o ostacima) Neka su $m_1, m_2, \ldots, m_k$ u parovima relativno prosti prirodni brojevi, tj. $(m_i, m_j) = 1$ za sve $i \neq j$. Tada za proizvoljne cijele brojeve $x_1, x_2, \ldots, x_k$ postoji cijeli broj x takav da vrijedi

$$x \equiv x_i \pmod{m_i}, i = 1, \ldots, k.$$

Broj x je jedinstven modulo $M = m_1 \cdots m_k$.

Pomoću Euklidova algoritma može se naći modularni inverz a_i takav da vrijedi $a_i \cdot M_i \equiv 1 \pmod{m_i}$, gdje je $M_i = \frac{M}{m_i}$. Iz toga onda slijedi da

$$x = \sum_{i=1}^{k} a_i M_i x_i \pmod m$$

zadovoljava uvjete teorema i to je jedinstveno rješenje kongruencije iz teorema. Složenost prethodno opisanog algoritma je $O(\ln^2 M)$. Primjena spomenutog teorema je velika, primjerice ako su u pitanju jako veliki brojevi, veći nego što računalo prirodno podržava, možemo to svesti na jednostavniji račun. Brojeve $m_1, \cdots, m_k$ u pravilu biramo sami.

2.4 Verižni razlomak

Još jednu primjenu Euklidovog algoritma nalazimo u ovom potpoglavlju. Naime, u prvom koraku Euklidovog algoritma imamo $a = bq_1 + r_1$. Podijelimo li tu jednakost sa b, dobivamo:

$$\frac{a}{b} = q_1 + \frac{1}{\frac{b}{r_1}}.$$

U drugom koraku imamo $b = r_1 g_1 + r_2$ i istim postupkom dolazimo do:

$$\frac{a}{b} = q_1 + \frac{1}{g_2 + \frac{1}{\frac{r_1}{r_2}}}.$$

Ponavljanjem postupka dobivamo prikaz racionalnog broja $\frac{a}{b}$, tzv. *jednostavni verižni razlomak*:

$$\frac{a}{b} = q_1 + \cfrac{1}{q_2 + \cfrac{1}{q_3 + \cfrac{1}{\ddots + \cfrac{1}{q_m}}}}.$$

Svaki realan broj α možemo na ovaj način razviti u jednostavan verižni razlomak:

$$\alpha = a_0 + \cfrac{1}{a_1 + \cfrac{1}{a_2 + \ddots}}$$

pri čemu je $a_0 \in \mathbb{Z}$, a $a_1, a_2, \ldots \in \mathbb{N}$. To zapisujemo u obliku $[a_0; a_1, a_2, \ldots]$. Brojevi $a_0, a_1, \ldots$ zovu se *parcijalni kvocijenti*. Računamo ih na sljedeći način:

$$a_0 = \lfloor \alpha \rfloor, \quad \alpha = a_0 + \frac{1}{\alpha_1}, \quad a_1 = \lfloor \alpha_1 \rfloor, \quad \alpha_1 = a_1 + \frac{1}{\alpha_2}, \quad a_2 = \lfloor \alpha_2 \rfloor, \ldots$$

Postupak završava kada je $a_k = \alpha_k$. Ovakav razvoj broja α je konačan ako i samo ako je α racionalan broj. Kad je α iracionalan, onda poznajemo samo njegovu aproksimaciju pa će u razvoju tog broja u jednostavni verižni razlomak biti točni samo početni parcijalni kvocijenti jer mi uzimamo najveće cijelo pa uvijek izgubimo barem jednu znamenku. Nakon par koraka izgubimo sve znamenke pa je onda ostatak tih a_i-ova sasvim slučajan. Postoji algoritam koji nam kaže kada trebamo stati. On uzima dvije aproksimacije, $\frac{a}{b}$ i $\frac{a'}{b'}$ i staje u trenutku kad se pripadni parcijalni kvocijenti više ne poklapaju. Pogledajmo to na primjeru:

Primjer 6. Broj $\sqrt{2}$ treba razviti u jednostavan verižni razlomak. Uzmimo aproksimacije $1.41 < \sqrt{2} < 1.42$.
$1.41 = \frac{141}{100}, \qquad 1.42 = \frac{142}{100}$
$q = 1, \qquad r = 41, \qquad r' = 21$
$a_0 = 1, a = 100, b = 41, a' = 50, b' = 21 \leftarrow$ obični Euklidov algoritam
$q = 2, \qquad r = 18, \qquad r' = 8$
$a_1 = 2, a = 41, b = 18, a' = 21, b' = 8$
$q = 2, \qquad r = 5, \qquad r' = 5$
$a_2 = 2, a = 18, b = 5, a' = 8, b' = 5$
$g = 3, \qquad r = 3, \qquad r' = -7, \qquad g' = 1 \leftarrow$ ovdje stajemo
Dobili smo razvoj $[1; 2, 2]$.

Za primjene verižnih razlomaka važnije je da kada je $\alpha = \sqrt{d}$ njegov razvoj u jednostavan verižni razlomak ima oblik $[a_0; \overline{a_1, a_2, \ldots, a_{r-1}, 2a_0}]$, pri čemu je $a_0 = \lfloor \sqrt{d} \rfloor$. Također vrijedi da je jednako čitamo li brojeve $a_1, \cdots, a_{r-1}$ slijeva na desno i obratno.

Spomenimo sada primjenu verižnih razlomaka u kriptografiji javnog ključa, posebno, u RSA kriptosustavu. Racionalne brojeve
$\frac{p_k}{q_k} = a_0 + \cfrac{1}{a_1 + \cfrac{1}{\ddots + \cfrac{1}{a_k}}}$ nazivamo *konvergente verižnog razlomka.* Razlika između dviju susjednih konvergenti je vrlo mala i te konvergente jako dobro aproksimiraju broj α, odnosno vrijedi:

$$\left| \alpha - \frac{p_k}{g_k} \right| < \frac{1}{g_k^2}.$$

Ako imamo tako dobar broj $\frac{p_k}{q_k}$, možemo se pitati mora li on baš dolaziti od verižnih razlomaka. Sljedeći teorem ([3]) govori o tome:

Teorem 4. Neka je α proizvoljan realan broj i c pozitivan realan broj. Ako racionalan broj $\frac{p}{q}$ zadovoljava nejednakost $\mid \alpha - \frac{p}{g} \mid < \frac{c}{g_k^2}$, onda je $\frac{p}{q} = \frac{rp_k \pm sp_{k-1}}{rq_k \pm sq_{k-1}}$, za neke nenegativne cijele brojeve k, r, s takve da je $rs < 2c$.

Ako je $c = \frac{1}{2}$, ova činjenica bit će od velike važnosti za Wienerov napad na RSA kriptosustav. Općenito, ako c nije mali broj, onda je to samo teorijski rezultat.

Verižni razlomci mogu se koristiti i za određivanje cijelih brojeva x i y takvih da je $x^2 + y^2 = p$, pri čemu je p dani prost broj oblika $4k + 1$. Fermat je znao da se neparan prost broj može prikazati u obliku zbroja kvadrata dva cijela broja ako i samo ako vrijedi $p \equiv 1 \pmod 4$, a Hermiteovom konstrukcijom koja koristi verižne razlomke te brojeve se onda može i pronaći.

2.5 Prostost

Sljedeći problem teorije brojeva koji se koristi u kriptografiji javnog ključa je prostost, odnosno testiranje prostosti. Krenimo od definicije prostog broja.

Definicija 14. ([2]) Za prirodan broj $p, p \geq 2$ kažemo da je *prost* ako ima točno dva djelitelja; 1 i p. Ako broj nije prost, kažemo da je *složen*.

Najjednostavniji i najprimitivniji način za određivanje je li broj n prost ili nije sastoji se od toga da redom krenemo sa $d = 2, 3, \ldots$ i provjeravamo je li n djeljiv s tim brojevima. Malo poboljšanje je da provjeravamo samo za $d \leqslant \sqrt{n}$, pa ako je n djeljiv s nekim takvim d, onda je on složen, u suprotnom je n prost broj. Međutim, potrebno vrijeme je $O(\sqrt{n})$, pa je ovaj algoritam neefikasan za malo veće n.
U kriptosustavima s javnim ključem prosti brojevi imaju značajnu ulogu. Preciznije, u RSA kriptosustavu vidjet ćemo da su dva tajna prosta broja p i q značajna za izgradnju javnog modula n, pri čemu su ti prosti brojevi izrazito veliki.

Za određivanju je li neki prirodan broj p prost ili složen koriste se testovi prostosti, odnosno kriteriji koje taj broj mora zadovoljiti kako bismo mogli

reći da je prost. Dakle, ako broj p ne zadovolji kriterije, onda sa sigurnošću znamo da je on složen, međutim ako ih zadovolji možemo reći da je on vjerojatno prost. U primjenama je to uglavnom dovoljno. Postoje rezultati koji govore o tome kad je neki broj prost, ali oni nisu tako jednostavni za provjeru. S druge strane, postoje različita svojstva koja se lakše provjeravaju, a jedno od njih nalazimo u sljedećem teoremu ([2]):

Teorem 5. (Mali Fermatov teorem) Neka je p prost broj. Ako $p \nmid a$, onda je $a^{p-1} \equiv 1 \pmod{p}$.

Međutim, postoje složeni brojevi koji zadovoljavaju gornju relaciju, pa to znači da ovo ne možemo iskoristiti kao test prostosti. Ali obrat ovog teorema može nam poslužiti kao test složenosti, odnosno tražit ćemo broj a takav da vrijedi $a^{p-1} \not\equiv 1 \pmod{p}$, i ako uspijemo pronaći takav broj, to znači da je n složen.

2.6 Faktorizacija

Testiranje prostosti uglavnom će nam dati informaciju o tome je li broj p prost ili složen, međutim ako je broj složen taj test nam neće dati faktore od p. Faktorizacija velikog broja glavni je problem kriptosustava s javnim ključem. Kao i kod prostosti, možemo krenuti s najprimitivnijim pristupom i tražiti netrivijalan faktor dijeljenjem našeg broja sa svim prostim brojevima koji su manji ili jednaki korijenu tog broja. Međutim, vidjeli smo i ranije da je to vrlo neefikasna metoda, pa ćemo spomenuti neke od najpoznatijih metoda faktorizacije i njihove glavne ideje.

2.6.1 Pollardova ρ metoda

Glavna ideja za pronalazak faktora p broja n je da prvo konstruiramo periodičan niz modulo p cijelih brojeva (x_i), onda nađemo i, j takve da vrijedi $x_i \equiv x_j \pmod{p}$ i na kraju odredimo p kao najveći zajednički djelitelj razlike $x_i - x_j$ i početnog broja n. Niz cijelih brojeva (x_i) konstruira se pomoću preslikavanja $f : \mathbb{Z}_n \to \mathbb{Z}_n$, gdje za funkciju f možemo uzeti kvadratni polinom. Sada izaberemo početnu vrijednost x_0 i onda računamo vrijednosti x_{i+1} iterativno, odnosno $x_{i+1} = f(x_i)$. Ovdje također postoje poboljšanja. Naime,

puno je efikasnije umjesto $(x_i - x_j, n)$ računati $(x_{2i} - x_i, n)$. Vrijednosti x_i i x_{2i} računamo na sljedeći način:

$$x_i = f(x_{i-1}) \pmod{n}, \qquad x_{2i} = f(f(x_{2i-2})) \pmod{n}.$$

Složenost ove metode je eksponencijalna, ali je ipak bolja od običnog dijeljenja.

2.6.2 Pollardova $\rho - 1$ metoda

Ova metoda zasnovana je na Malom Fermatovom teoremu kojeg smo ranije i spomenuli. Ako uzmemo da je m višekratnik broja $p - 1$, onda vrijedi i $a^m \equiv 1 \pmod{p}$. Sada je problem što ne znamo m. Međutim, ako saznamo m, onda je $(a^m - 1, n)$ jedan faktor od n. Za nalaženje broja m potrebna nam je sljedeća definicija:

Definicija 15. ([3]) Za prirodan broj kažemo da je B-gladak ako su mu svi prosti faktori manji ili jednaki B.

Primjerice, ako uzmemo da je $B = 5$, onda je broj 60 B-gladak broj jer su njegovi prosti faktori $2, 3$ i 5.
Ako znamo da je broj $p - 1$ B-gladak, tada za m možemo uzeti najmanji zajednički višekratnik brojeva $1, 2, \ldots, B$. Pogledajmo to na primjeru:

Primjer 7. Neka je $n = 52, B = 5, a = 2$.
$m = 2 \cdot 3 \cdot 5 = 30$, pa imamo $2^{30} \pmod{52} = 12$ i $(12, 52) = 4$.
Zaista, $52 = 4 \cdot 13$.

2.6.3 Fermatova faktorizacija

Spomenut ćemo još jednu metodu, tzv. *Fermatovu faktorizaciju.* Nju možemo primjeniti na broj koji je produkt dvaju vrlo bliskih brojeva. Ako imamo broj $n = ab$, onda taj broj možemo zapisati u obliku $n^2 = x^2 - y^2$, pri čemu su x i y takvi da vrijedi $x = \frac{a+b}{2}$ i $y = \frac{a-b}{2}$. Jedan od tih brojeva onda je blizu $\sqrt{n}$, pa krenemo redom od $\lfloor \sqrt{n} \rfloor + 1$ i tražimo faktorizaciju. Pogledajmo na primjeru:

Primjer 8. Neka je $n = 310717$. Imamo $\lfloor \sqrt{310717} \rfloor + 1 = 558$. Sada je $558^2 - 310717 = 647$, a to nije kvadrat. Nastavljamo dalje sa $x = 559$.

Imamo $559^2 - 310717 = 1764$, a vrijedi $42^2 = 1764$. Sada je $310717 = 559^2 - 42^2 = (559 + 42)(559 - 42) = 517 \cdot 601$.

Postoji modifikacija ove metode u kojoj se ne traže brojevi x, y takvi da je $n = x^2 - y^2$, nego se traže brojevi x, y takvi da $n|x^2 - y^2$, odnosno da vrijedi $y^2 \equiv x^2 \pmod{n}$. Ako vrijedi $s \not\equiv t \pmod{n}$, onda su $(s + t, n)$ i $(s - t, n)$ netrivijalni faktori od n.

2.7 Kvadratne kongruencije

Još jedan dio teorije brojeva čiju primjenu susrećemo u kriptografiji su kvadratne kongruencije.

Definicija 16. ([4]) Neka su a i n relativno prosti cijeli brojevi. Ako kongruencija $x^2 \equiv a \pmod{n}$ ima rješenja, tada kažemo da je a kvadratni ostatak modulo n. U suprotnom kažemo da je a kvadratni neostatak modulo n.

Kvadratnih ostataka i neostataka ima jednako, odnosno $\frac{p-1}{2}$.

Definicija 17. ([4]) Neka je p neparan prost broj. Legendreov simbol $(\frac{a}{p})$ jednak je 1 ako je a kvadratni ostatak modulo p, jednak je -1 ako je a kvadratni neostatak modulo p, a 0 ako je $a \equiv 0 \pmod{p}$.

Legendreov simbol može se izračunati preko Eulerova kriterija koji glasi:

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

Sada imamo $a^{\frac{(p-1)}{2}} = \sqrt{a^{p-1}}$, a Mali Fermatov teorem nam kaže da je $a^{p-1} \equiv 1 \pmod{m}$.

Ovaj kriterij nam daje polinomijalan algoritam, ali postoji mogućnost poboljšanja uz korištenje Gaussovog kvadratnog zakona reciprociteta:

$$\left(\frac{p}{q}\right)\left(\frac{q}{p}\right) = (-1)^{\frac{(p-1)(q-1)}{4}}.$$

Međutim, ovdje stoji zahtjev da su p i q prosti, pa uvodimo sljedeću definiciju:

Definicija 18. ([4]) Neka je m neparan prirodan broj i $m = \prod_{i=1}^{k} p_i^{\alpha_i}$ njegov rastav na proste faktore, te neka je a cijeli broj. Jacobijev simbol $(\frac{a}{m})$ se definira s $(\frac{a}{m}) = \prod_{i=1}^{k} \left(\frac{a}{p_i}\right)^{\alpha_i}$, gdje je $\frac{a}{p_i}$ Legendreov simbol.

Algoritam za računanje Jacobijevog simbola vrlo je sličan Euklidovom algoritmu pa mu je i složenost jednaka kao i složenost Euklidovog algoritma. Ono što nas sada zanima je kako pronaći x, odnosno kako izračunati kvadratni korijen od a modulo p. Pretpostavit ćemo da p nije mali broj jer je u tom slučaju račun vrlo jednostavan. Navest ćemo dva slučaja:
1. Broj p je oblika $p = 4k + 3$.

Propozicija 1. ([3]) Ako je $p \equiv 3 \pmod 4$, onda je $x = a^{\frac{(p+1)}{4}}$ rješenje kongruencije $x^2 \equiv a \pmod p$.

2. Broj p je oblika $p = 8k + 5$.

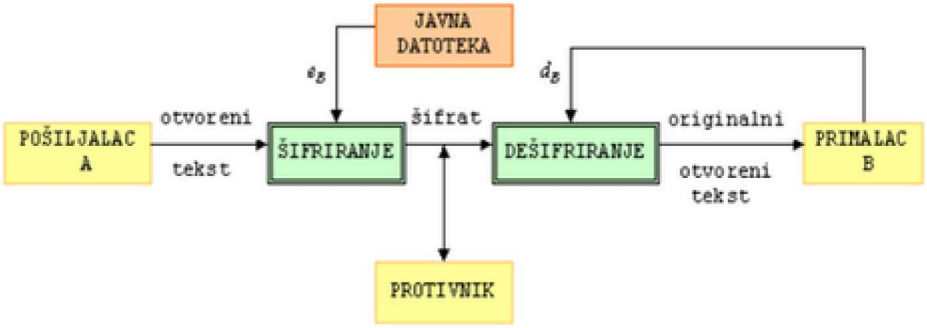
Propozicija 2. ([3]) Ako je $p \equiv 5 \pmod 8$, onda je jedan od brojeva $a^{\frac{(p+3)}{8}}$ i $2^{\frac{(p-1)}{4}} a^{\frac{(p+3)}{8}}$ rješenje kongruencije $x^2 \equiv a \pmod p$.

3 Kriptografija javnog ključa

U klasičnoj kriptografiji o kojoj je bilo riječi u prvom poglavlju, dvije osobe, pošiljalac i primalac, potajno izabiru ključ K i taj ključ onda generira funkcije za šifriranje e_K i dešifriranje d_K. U kriptosustavima tog tipa funkcija d_K je ili potpuno ista kao e_K ili se vrlo lako može dobiti iz nje. Takvi kriptosustavi nazivaju se simetrični kriptosustavi. Nedostatak je što zahtjevaju prethodnu razmjenu ključa K između tih dviju osoba prije prenošenja nekog šifrata. Međutim, u praksi to može biti vrlo često neizvedivo ili teško izvedivo. Ideja koja stoji iza kriptosustava s javnim ključem je mogućnost nalaženja kriptosustava u kojem je praktički nemoguće izračunati d_K iz danog e_K. U tom slučaju funkcija e_K može biti javna. Dakle, u kriptosustavu s javnim ključem svaki korisnik ima dva ključa: javni e_K i tajni d_K. Ako pošiljalac želi poslati poruku, on mora znati samo javni ključ koji mu je prethodno poslao primalac. Tada on otvoreni tekst šifrira pomoću e_K i šalje primaocu šifrat $y = e_K(x)$. Tada primalac može dešifrirati šifrat koristeći svoj tajni ključ d_K. Kako bi ova situacija zaista bila realistična, primalac treba imati neku dodatnu informaciju, tzv. *skriveni ulaz*, koji dopušta lako računanje inverza od e_K, međutim nitko drugi osim njega ne može imati tu informaciju. Takve funkcije kod kojih je teško izračunati inverz bez poznavanja nekog dodatnog podatka nazivaju se *osobne jednosmjerne funkcije*. Općenito u primjenama, kriptosustavi s javnim ključem ne koriste se za šifriranje poruka nego za šifriranje ključeva koji se koriste u simetričnim kriptosustavima. Također, ovakvi kriptosustavi omogućavaju tzv. *digitalni potpis*. Drugim riječima, ako osoba A pošalje osobi B šifrat $a = d_A(e_B(x))$, osoba B može biti sigurna da joj je to poslala osoba A, odnosno osoba A to ne može zanijekati. Definirajmo sada kriptosustav s javnim ključem.

Definicija 19. ([3]) Kriptosustav s javnim ključem sastoji se od familije funkcija za šifriranje e_K i funkcija za dešifriranje d_K koje zadovoljavaju iduća svojstva:

1. d_K je inverz od e_K za svaki K;
2. e_K je javan za svaki K, ali je d_K poznat samo osobi K
3. e_K je osobna jednosmjerna funkcija za svaki K.



Slika 2: *Shematski prikaz kriptosustava s javnim ključem ([3]).*

Napomenimo da kriptosustav s javnim ključem nikad ne može ponuditi bezuvjetnu sigurnost. Razlog tomu je što protivnik može redom šifrirati svaki mogući otvoreni tekst koristeći javno pravilo šifriranja dok ne pronađe jedinstveni x takav da je $y = e_K(x)$. Naravno, ovakav napad je moguć samo ako je skup vrijednosti otvorenog teksta mali broj.

3.1 RSA kriptosustav

O teškoći i metodama faktorizacije rekli smo nešto u prethodnom poglavlju. Najpoznatiji i najrašireniji kriptosustav s javnim ključem je RSA kriptosustav i zasnovan je na problemu faktorizacije velikih prirodnih brojeva. Definirajmo ga:

Definicija 20. ([3]) Neka je $n = pg$, pri čemu su p i q prosti brojevi. Neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$ i $\mathcal{K} = \{(n, p, q, d, e) \in \mathbb{Z}^5 : n = pq, de \equiv 1 \pmod{\varphi(n)}\}$. Za $K \in \mathcal{K}$ definiramo funkciju šifriranja

$$e_K(x) = x^e \pmod{n}$$

i funkciju dešifriranja

$$d_K(y) = y^d \pmod{n}$$

za $x, y \in \mathbb{Z}_n$. Vrijednosti n i e su javne, a p, q i d su tajne, pa se (n, e) naziva javni, a (p, q, d) tajni ključ.

Funkcija $\varphi(n)$ naziva se *Eulerova funkcija* i broji sve pozitivne cijele brojeve manje od n koji su relativno prosti sa n. Vrijedi:

$$\varphi(n) = \varphi(pq) = \varphi(p)\varphi(q) = (p-1)\cdot(q-1) = n - p - q + 1.$$

Može se pokazati da su funkcije e_K i d_K zaista jedna drugoj inverzne, koristeći sljedeći teorem ([2]):

Teorem 6. (Eulerov teorem) Ako su $x \in \mathbb{N}$ i $n \in \mathbb{Z}$ takvi da je $(x, n) = 1$, onda je

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

Dakle, imamo $d_K(e_K(x)) \equiv x^{de} \pmod{n}$. Iz kongruencije $de \equiv 1 \pmod{\varphi(n)}$ slijedi da postoji $k \in \mathbb{N}$ takav da je $de = k\varphi(n) + 1$. Pretpostavimo sada da je $(x, n) = 1$. Prema Eulerovom teoremu vrijedi:

$$x^{de} = x^{k\varphi(n)} = (x^{\varphi(n)})^k \cdot x \equiv x \pmod{n}.$$

Ako je $(x, n) = n$, onda je $x^{de} \equiv 0 \equiv x \pmod{n}$.
Ako je $(x, n) = p$, onda je $x^{de} \equiv 0 \equiv x \pmod{p}$. Budući da je $(pq, x) = p$, a p i q su prosti, vrijedi $(q, x) = 1$, pa prema Eulerovom teoremu imamo da je $x^{\varphi(q)} = x^{q-1} \equiv 1 \pmod{q}$. Sada je $x^{de} = (x^{q-1})^{(p-1)k} \cdot x \equiv x \pmod{q}$ pa je $x^{de} \equiv x \pmod{n}$.
Ako je $(x, n) = q$, onda je $x^{de} \equiv 0 \equiv x \pmod{q}$. Budući da je $(pq, x) = q$, a p i q su prosti, vrijedi $(p, x) = 1$, pa prema Eulerovom teoremu imamo da je $x^{\varphi(p)} = x^{p-1} \equiv 1 \pmod{p}$. Sada je $x^{de} = (x^{p-1})^{(q-1)k} \cdot x \equiv x \pmod{p}$ pa je $x^{de} \equiv x \pmod{n}$. Vidimo da u svakom slučaju vrijedi $x^{de} \equiv x \pmod{n}$, odnosno $d_K(e_K(x)) = x$.

Pogledajmo jedan jednostavan primjer.

Primjer 9. Neka je $p = 5$ i $q = 13$. Tada je $n = pq = 5 \cdot 13 = 65$ i $\varphi(n) = (p-1)(q-1) = (5-1)(13-1) = 4 \cdot 12 = 48$. Javni eksponent e mora biti relativno prost sa $\varphi(n)$, pa ćemo uzeti da je $e = 5$. Sada je javni ključ $(n, e) = (65, 5)$. Kako je $(e, \varphi(n)) = 1$, znamo da postoje cijeli brojevi d i k takvi da vrijedi $ed + k\varphi(n) = 1$, odnosno $ed \equiv 1 \pmod{\varphi(n)}$. Dobivamo kongruenciju $5e \equiv 1 \pmod{\varphi(n)}$ i slijedi da je $d = 29$. Pretpostavimo da osoba A želi poslati poruku $x = 55$. Trebamo izračunati $x^e \pmod{n}$, tj. $55^5 \pmod{65}$. Dobivamo šifrat $y = e_K(x) = 35$. Osoba B dešifrira to pomoću funkcije d_K, tj. $x = d_K(y) = 35^{29} \pmod{65}$ i dobiva početnu poruku $x = 55$.

U primjeru smo radi jednostavnosti uzeli male proste brojeve p i q, ali u primjenama ti brojevi trebaju biti puno veći (oko 100 znamenaka) kako treća osoba, protivnik, ne bi mogao lako razbiti kriptosustav. Sada ćemo reći nešto o tome na koji način izabrati parametre.

1. Odabir dva tajna prosta broja p i q nije jednostavan. Prvo treba generirati slučajan broj m s određenim brojem znamenaka. Zatim koristeći ranije spomenute testove prostosti ispitati da li je taj m prost te ako nije pronaći prvi prost broj koji je veći od njega. Važno je da se to može brzo testirati, a onda, ako je potrebno, brzo pronaći prvi prost broj veći od njega. Sljedeće na što treba paziti je da broj $n = pq$ nije lako fatorizirati, odnosno ne bi trebao biti jedan od brojeva specijalnog oblika koje smo naveli ranije jer za njihovu faktorizaciju danas postoje vrlo efikasni algoritmi. To znači da bi svaki od brojeva $p \pm 1$ i $g \pm 1$ trebao imati barem jedan veliki prosti faktor. Osim toga, brojevi p i q ne smiju biti previše blizu jedan drugome jer se u tom slučaju može ispitivati sve brojeve koji su blizu $\sqrt{n}$. Općenito se preporučuje da p i q budu 512-bitni prosti brojevi, pa će n biti 1024-bitni modul, a faktorizacija brojeva ove veličine je daleko iznad mogućnosti najboljih postojećih algoritama za faktorizaciju.

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

3. Izabiremo broj e koji je relativno prost s $\varphi(n)$ i koristeći prošireni Ekulidov algoritam dolazimo do kongruencije $de \equiv 1 \pmod{\varphi(n)}$. S obzirom da se za šifriranje koristi ranije opisano modularno potenciranje, najbolje bi bilo uzeti što manji eksponent e, međutim za sigurnost RSA kriptosustava to nije najbolja opcija.

4. Preostaje staviti javni ključ (n, e) u javnu datoteku.

Vidjeli smo ranije da ako znamo faktorizirati n onda možemo naći $\varphi(n)$ i d. Također, ako poznajemo n i uz to $\varphi(n)$, onda iz dviju jednakosti

$$n = pg$$

i

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

možemo pronaći p i q. Postavlja se pitanje, ako je poznat tajni eksponent d, mogu li se iz toga saznati p i q. Naime, postoji vrlo efikasan algoritam, tzv. vjerojatnosni algoritam, koji u polinomijalnom vremenu iz poznavanja tajnog eksponenta faktorizira n. Taj algoritam neće uvijek dati odgovor, ali u slučaju da ga daje tada je odgovor sigurno točan. Također, postoji i algoritam koji iz poznavanja d daje faktorizaciju od n, ali uz uvjet da je $ed < n^2$. Ako protivnik slučajno ili nekom metodom otkrije d, osim e treba promijeniti i n.

Recimo sada nešto o sigurnosti RSA kriptosustava. Kada razmišljamo o odabiru javnog i tajnog eksponenta, dolazimo do zaključka da bi osobama koje komuniciraju bilo lakše ako su ti parametri mali brojevi. Međutim, u tom slučaju je vrlo lako protivniku razbiti taj kriptosustav i saznati tekst poruke. O slučaju kada je d mali tajni eksponent (u odnosu na n) govori nam sljedeći teorem ([3]):

Teorem 7. Neka je $n = pq$ i $p < q < 2p$. Pretpostavimo da je $e < \varphi(n)$ i $d < \frac{1}{3}n^{0.25}$. Tada postoji polinomijalni algoritam koji iz poznavanja n i e izračunava d.

Dokaz: Iz $ed \equiv 1 \pmod{\varphi(n)}$ slijedi da postoji $k \in \mathbb{N}$ takav da je $ed - k\varphi(n) = 1$. Dijeljenjem te jednakosti s $d\varphi(n)$ dobivamo:

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

Iz toga se vidi da je $\frac{k}{d}$ dobra aproksimacija od $\frac{e}{\varphi(n)}$. Kako ne znamo $\varphi(n)$, možemo ju aproksimirati s n. Iz $\varphi(n) = n - p - q + 1$ i $p + q - 1 < 3\sqrt{n}$ slijedi $\left|n - \varphi(n)\right| < 3\sqrt{n}$. Zamijenimo sada $\varphi(n)$ sa n i dobivamo:

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

$$= \frac{3k}{d\sqrt{n}}.$$

Kako je $k\varphi(n) = ed - 1 < ed$ i $e < \varphi(n)$ slijedi $k < d < \frac{1}{3}n^{0.25}$ i dobivamo

$$\left|\frac{e}{n} - \frac{k}{d}\right| \leqslant \frac{1}{d\sqrt[4]{n}} < \frac{1}{2d^2}.$$

Kod verižnih razlomaka smo naveli teorem koji nam govori da ova relacija povlači da $\frac{k}{d}$ mora biti konvergenta razvoja u verižni razlomak od $\frac{e}{n}$. Dakle, možemo odrediti konvergente u razvoju $\frac{e}{n}$ i tražiti koja od njih zadovoljava kongruenciju $(x^e)^d \equiv x \pmod{n}$ za slučajno odabrani broj x. Konvergenti koje provjeravamo nema puno jer nazivnici eksponencijalno rastu, a jedan od tih nazivnika je naš tajni eksponent d. Postoji i drugi način za određivanje je li promatrana konvergenta prava, a taj način dodatno daje i faktorizaciju broja n. Polazimo od pretpostavke da je $\frac{k}{d}$ konvergenta koja nam treba. Izračunamo $\varphi(n) = (p-1)(q-1) = \frac{ed-1}{k}$. Tada se može izračunati $\frac{p+q}{2}$ iz $\frac{pg-(p-1)(q-1)+1}{2} = \frac{p+q}{2}$, te $\frac{q-p}{2}$ iz $(\frac{p+q}{2})^2 - pq = (\frac{q-p}{2})^2$. Ako dobijemo da su $\frac{p+q}{2}$ i $\frac{q-p}{2}$ cijeli brojevi, zaključujemo da smo dobili traženu konvergentu. Tada iz $\frac{p+q}{2}$ i $\frac{q-p}{2}$ možemo lako dobiti faktorizaciju $n = pq$.

U slučaju kad je n jako velik, dobit ćemo jako puno konvergenti čije nazivnike moramo provjeravati. Postoji karakterizacija prave konvergente koja će nam pomoći da ne provjeravamo sve. Pretpostavka je da je $n > 10^8$. Konvergenta $\frac{k}{d}$ zadovoljava

$$\frac{2e}{n\sqrt{n}} < \frac{k}{d} - \frac{e}{n} < \frac{2.122e}{n\sqrt{n}}.$$

Osim korištenja malog tajnog eksponenta, nije dobro koristiti ni mali javni eksponent. Prije je najčešće bio korišten $e = 3$, kao prvi najmanji dopušteni broj (1 i 2 ne dolaze u obzir). Međutim, pokazano je kako to narušava sigurnost RSA kriptosustava. Pretpostavimo da netko želi trima različitim korisnicima poslati istu poruku m, pri čemu svaki od korisnika ima svoj javni modul n_1, n_2, n_3 i da svi koriste javni eksponent $e = 3$. Protivnik tada može saznati šifrate:

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

On tada može koristeći Kineski teorem o ostacima riješiti sustav linearnih kongruencija:

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

Dobiva kongrunciju $x \equiv m^3 \pmod{n_1 n_2 n_3}$. Kako m mora biti manji od n_1, n_2, n_3, vrijedi da je $m^3 < n_1 n_2 n_3$ pa slijedi da je $x = m^3$, odnosno $m = \sqrt[3]{x}$. Vidimo zašto se ne preporučuje korištenje malog javnog eksponenta.

Općenito, najčešće se koristi $e = 2^{16} + 1 = 65537$ jer sadrži malo jedinica u binarnom sustavu pa šifriranje ide vro brzo. Još uvijek nije poznata metoda kojom bi se mogao u potpunosti probiti RSA kriptosustav. Postoje slabosti tog kriptosustava na koje treba obratiti pozornost, ali se općenito smatra sigurnim kriptosustavom.

3.2 Rabinov kriptosustav

U ovom potpoglavlju kratko ćemo opisati Rabinov kriptosustav. Ovaj kriptosustav zasnovan je na problemu računanja kvadratnog korijena u $\mathbb{Z}_n$, tj. traženja $x \in \mathbb{Z}$ takvog da je $x^2 \equiv a \pmod{n}$, odnosno da je a kvadratni ostatak modulo n. Taj problem povezan je s problemom faktorizacije. Kao i u RSA kriptosustavu, neka je $n = pq$, pri čemu su p i q prosti brojevi. U potpoglavlju o kvadratnim kongruencijama vidjeli smo da se takve kongruencije vrlo efikasno rješavaju za brojeve specijalnog oblika. Za početak, definirajmo Rabinov kriptosustav.

Definicija 21. ([3]) Neka je $n = pq$ gdje su p i q prosti brojevi takvi da je $p \equiv q \equiv 3 \pmod{4}$. Neka je $\mathcal{P} = \mathcal{C} = \mathbb{Z}_n$, te

$$\mathcal{K} = \{(n, p, q) \in \mathbb{Z}^3 : n = pq\}.$$

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

$$e_K(x) = x^2 \pmod{n}$$

i

$$d_K = \sqrt{y} \pmod{n}.$$

Vrijednost n je javna, a vrijednosti p i q su tajne.

Zahtjev iz definicije $p \equiv q \equiv 3 \pmod{4}$ nije nužan, ali smo ranije vidjeli da je to jedan od specijalnih slučajeva, a samim tim će šifriranje biti lakše. Zaista, ako je taj zahtjev ispunjen, rješenje kongruencije $x^2 \equiv a \pmod{p}$ dobivamo kao $x = \pm a^{\frac{(p+1)}{4}} \pmod{p}$. Isto tako, rješenje kongruencije $x^2 \equiv a \pmod{q}$ dobijemo kao $x = \pm a^{\frac{(q+1)}{4}} \pmod{q}$. Kombinacijom tih rješenja i upotrebom Kineskog teorema o ostacima dobivamo rješenja kongruencije $x^2 \equiv a \pmod{pq} \equiv a \pmod{n}$. Tu primjećujemo nedostatak Rabinovog kriptosustava. Naime, funkcija šifriranja e_K nije injekcija, pa dešifriranje ne

može biti jednoznačno određeno, odnosno dobivamo 4 rješenja (4 kvadratna korijena) modulo n. Općenito, primalac ne može znati koje od rješenja je zaista pravo, osim ako otvoreni tekst nije neki smisleni tekst, ali ranije smo rekli da kriptosustavi s javnim ključem služe uglavnom za šifriranje ključeva koje kasnije koriste simetrični kriptosustavi. Druga mogućnost je da se u otvoreni tekst ubaci nešto po čemu će primalac prepoznati da je to pravo rješenje.
Pretpostavimo da osoba A želi osobi B poslati poruku koristeći Rabinov kriptosustav. Osoba A prvo uzima javni ključ osobe B. Zatim poruku koju želi poslati prevodi u broj $x \in \mathbb{Z}_n$ i računa kvadratnu kongruenciju $y \equiv x^2 \pmod{n}$. Naposljetku šifrat šalje osobi B.
Pretpostavimo sada da je osoba B primila taj šifrat od osobe A. Koristeći prošireni Euklidov algoritam, osoba B računa cijele brojeve x, y takve da vrijedi $px + qy = 1$, pri čemu je $n = pq$. Nakon toga, rješava kongruencije $r \equiv y^{\frac{(p+1)}{4}} \pmod{p}$ i $s \equiv y^{\frac{(q+1)}{4}} \pmod{q}$. Konačno, računamo $u \equiv (xps + yqr) \pmod{n}$ i $v \equiv (xps - yqr) \pmod{n}$. Kvadratni korijeni koje tražimo su: $m_1 = u, m_2 = -u, m_3 = v, m_4 = -v$. Pogledajmo na primjeru kako funkcionira Rabinov kriptosustav.

Primjer 10. Neka je otvoreni tekst $x = 5$. Zapisat ćemo ga u binarnom zapisu: $x = 101_2$. Kako bi osoba B znala koji od četiri kvadratna korijena je pravi, ponovit ćemo zadnja 3 bita: $101101_2 = 45$. Neka je $p = 11$, a $q = 7$. To su dva prosta broja za koja vrijedi da je $p \equiv 3 \pmod{4}$. Tada je $n = pq = 77$. Javni je ključ $n = 77$, a tajni $11, 7$.
Šifriranje se vrši na način: $y \equiv 45^2 \pmod{77} \equiv 23 \pmod{77}$.
Kod dešifriranja, prvo ćemo izračunati x i y takve da je $11x + 7y = 1$. Dobivamo da je $x = 2$ i $y = -3$. Sada računamo:

$$r = 23^{\frac{11+1}{4}} \pmod{11} = 1 \pmod{11}, \quad s = 23^{\frac{7+1}{4}} \pmod{7} = 4 \pmod{7}.$$

Na kraju, računamo u, v:

$$u = (2 \cdot 11 \cdot 4 + (-3) \cdot 7 \cdot 1) \pmod{77} = 67$$

i

$$v = (2 \cdot 11 \cdot 4 - (-3) \cdot 7 \cdot 1) \pmod{77} = 32.$$

Kvadratni korijeni koje tražimo su:

$$m_1 = 67, m_2 = 10, m_3 = 32, m_4 = 45.$$

Kad osoba B zapiše ovo u binarnom zapisu, primjetit će pravilnost kod m_4 i znat će da je to traženi kvadratni korijen.

Literatura

[1] J. Buhler, *Algorithmic Number Theory*, Third International Symposium, ANTS-III, Portland, Orgeon, USA, 1998.

[2] A. Dujella, *Teorija brojeva*, Školska knjiga, 2019.

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

[4] D. R. Stinson, *Cryptography Theory and Practice, Third Edition*, Chapman Hall/CRC, Boca Raton, 2006.

Sažetak

Ovaj rad proučava algoritme teorije brojeva i njihovu primjenu u kriptografiji javnog ključa te se može promatrati kroz tri osnovne cjeline.
Prvi dio odnosi se na osnovne pojmove kriptografije te definicije i kriptoanalizu nekoliko jednostavnih kriptosustava.
Drugi dio rada govori o teoriji brojeva i algoritmima teorije brojeva čiju primjenu nalazimo u kriptografiji. Takvih algoritama i pojmova iz teorije brojeva ima mnoštvo, a u ovom radu opisani su: složenost algoritama, osnovne računske operacije, Euklidov algoritam, verižni razlomci, kvadratne kongruencije te prostost i faktorizacija.
Treći dio rada govori o kriptografiji javnog ključa te opisuje dva najpoznatija kriptosustava s javnim ključem, a to su RSA i Rabinov kriptosustav. Kod svakoga od njih nalazimo opis, definiciju, implementaciju i primjer.

Ključne riječi

Kriptografija, algoritam, složenost algoritma, Euklidov algoritam, teorija brojeva, kriptosustav s javnim ključem

Summary

This thesis studies algorithms of number theory and their application in public-key cryptography and can be observed through three basic units.
The first part refers the basic concepts of cryptography, definition and cryptanalysis of several simple cryptosystems.
The second part of this thesis discusses number theory and algorithms of number theory whose application we find in cryptography. There are many such algorithms and concepts from number theory, and here are described: algorithmic complexity, basic computational operations, Euclid's algorithm, continued fractions, quadratic congruences, primality and factoring.
The third part discusses public key cryptography and describes the two most well-known public key cryptosystems, namely RSA and Rabin's cryptosystem. In each of them we find a description, definition, implementation and example.

Keywords

Cryptography, algorithm, algorithmic complexity, Euclid's algorithm, number theory, public-key cryptosystem

Životopis

Rođena sam 15. travnja 1997. godine u Osijeku. Završila sam Osnovnu školu Vladimir Nazor u Đakovu, te nakon toga upisala Prirodoslovno-matematičku gimnaziju u Đakovu. Nakon završetka, upisala sam sveučilišni nastavnički studij matematike i informatike na Odjelu za matematiku u Osijeku.