Kriptografija javnog ključa i diskretni logaritmu

Storić, Luka

Sveučilište J. J. Strossmayera u Osijeku

Odjel za matematiku

Diplomski studij Financijska matematika i statistika

Luka Storić

KRIPTOGRAFIJA JAVNOG KLJUČA I DISKRETNI LOGARITMI

Diplomski rad

Osijek, 2022.

Sveučilište J. J. Strossmayera u Osijeku

Odjel za matematiku

Diplomski studij Financijska matematika i statistika

Luka Storić

KRIPTOGRAFIJA JAVNOG KLJUČA I DISKRETNI LOGARITMI

Diplomski rad

Mentor: prof. dr. sc. Ivan Matić

Osijek, 2022.

Zahvaljujem se mentoru prof.dr.sc. Ivanu Matiću što je sa svojim stručnim znanjem i vodstvom omogućio realizaciju ovog diplomskog rada.

Jedno veliko hvala mojim roditeljima i Nini, jer bez Vas ovo ništa ne bi bilo moguće.

Sadržaj

1 Uvod

U ovom radu bavit ćemo se kriptografijom javnog ključa i problemom diskretnog logaritma.

U drugom poglavlju upoznajemo se s osnovnim pojmovima u kriptografiji. Trenutne primjene kriptografije daleko nadilaze nacionalnu sigurnost, budući da uporaba računala i elektronski prijenos informacija postaju sve više dio svakodnevnog života.

Nadalje, u trećem poglavlju detaljinije opisujemo ElGamalov kriptosustav te njegovu sigurnost.

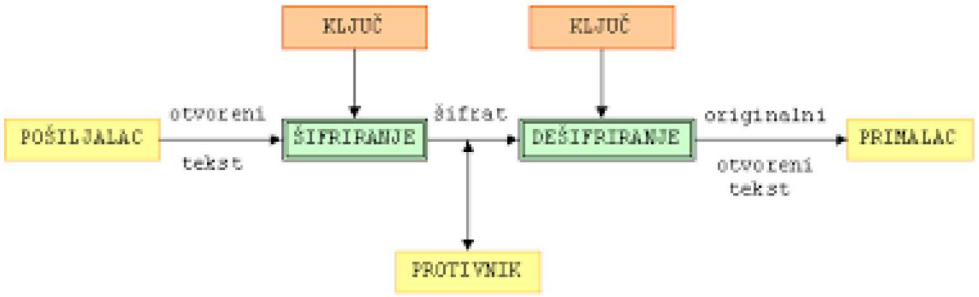
Četvrto poglavlje govori o algoritmima za problem diskretnih logaritama, tj. o Shanksovu algoritmu, Pollardovu rho algoritmu, Pohlig-Hellmanovu algoritmu i metodu računanja indeksa.

Nerješivost problema diskretnog logaritma pruža osnovu za sigurnost mnogih kriptosustava javnog ključa.

2 Osnovni pojmovi u kriptografiji

Kriptografija, umjetnost i znanost sigurnog prijenosa informacija. Glavna svrha kriptografije (od grčki krypto- skrivati te grafo- pisati) je omogućiti dvjema osobama koji se obično nazivaju Alice i Bob (pošiljalac i primalac), da komuniciraju preko nesigurnog kanala na takav način da Oscar (protivnik) ne može razumjeti što se govori.

Informacije koje Alice želi poslati Bobu nazivamo "otvoreni tekst". Alice transformira otvoreni tekst pomoću unaprijed određenog ključa i takav proces nazivamo šifriranje. Dobiveni šifrirani tekst šalje preko kanala i takav proces nazivamo šifrat. Nakon što je Oscar putem prisluškivanja vidio šifrirani tekst u kanalu, ne može odrediti što je otvoreni tekst. Ali Bob zna ključ za šifriranje, može dešifrirati šifrirani tekst i rekonstruirati otvoreni tekst.



Slika 1: Shema simetrične kriptografije

Kriptosustav je uređena petorka $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ tako da je:

- $\mathcal{P}$ konačan skup svih osnovnih elemanata otvorenog teksta
- $\mathcal{C}$ konačan skup svih osnovnih elemenata šifrata
- $\mathcal{K}$ konačan skup svih ključeva (tj. prostor ključa)
- Za svaki $K \in \mathcal{K}$ postoji funkcija šifriranja $e_K \in \mathcal{E}$ i odgovarajuća funkcija dešifriranja $d_K \in \mathcal{D}$. Pritom za funkcije e_K i d_K vrijedi

$$d_K(e_K(x)) = x,$$

  za svaki otvoren tekst $x \in \mathcal{P}$.

Ključ e_K zovemo javni ključ dok d_K zovemo tajni ključ.

Osnovna podjela kriptosustava s obzirom na vrstu ključa:

- simetrični kriptosustavi
- kriptosustavi s javnim ključem

Ključ za dešifriranje simetričnih kriptosustava može se izračunati poznavajući ključ za šifriranje i obratno. Sigurnost takvih kriptosustava leži u tajnosti ključa, tj. zato se zovu kriptosustavi s tajnim ključem.
Ključ za dešifriranje kriptosustava s javnim ključem ne može se izračunati iz ključa za šifriranje. Ključ za šifriranje je javni ključ, ali samo osoba koja ima odgovarajući ključ za dešifriranje može dešifrirati tu poruku.

3 ElGamalov kriptosustav s javnim ključem

ElGamalov kriptosustav temelji se na problemu diskretnog logaritma.
Započinjemo opisivanjem ovog problema u konačnoj multiplikativnoj grupi $(G, \cdot)$.
Za element $\alpha \in G$ reda n, definiramo

$$\langle \alpha \rangle = \{\alpha^i : 0 \leq i \leq n-1\}.$$

Lako je uočiti da je $\langle \alpha \rangle$ ciklička podgrupa od G reda n.
Uzet ćemo da je G multiplikativna grupa konačnog polja $\mathbb{Z}_p$ (gdje je p prost broj), i α primitivni korijen modulo p. U ovoj situaciji imamo da je $n = |\alpha| = p - 1$.
Također možemo uzeti da je α reda q, gdje je q prost broj u multiplikativnoj grupi $\mathbb{Z}_p^*$ (gdje je p prost broj i $p - 1 \equiv 0 \pmod{q}$). Takav α može se dobiti kao potencija primitivnih elemenata s eksponentom $(p-1)/q$.

Sada definiramo problem diskretnog logaritma u podgrupi $\langle \alpha \rangle$ grupe $(G, \cdot)$.

Problem 3.1. ***Diskretni logaritam***

Multiplikativna grupa $(G, \cdot)$, element $\alpha \in G$ reda n, i element $\beta \in \langle \alpha \rangle$.
Pronađi jedinstveni cijeli broj a, $0 \leq a \leq n-1$, takav da je

$$\alpha^a = \beta.$$

Taj ćemo cijeli broj a označiti s $\log_\alpha \beta$; naziva se diskretni logaritam od β.

U kriptografiji se trenutno vjeruje kako je diskretne logaritme u slučajevima nekih grupa vrlo teško izračunati, ali inverzna operacija potenciranja može se učinkovito izračunati metodom kvadriraj i množi.

Algoritam 3.1. *([7], Algorithm 5.5) Algoritam kvadriraj i množi* (x, c, n)

$z \longleftarrow 1$
za $i \longleftarrow l - 1$

$$\textit{Učini} \begin{cases} z \longleftarrow z^2 \bmod n \\ \textit{ako je } c_i = 1 \\ \textit{onda } z \longleftarrow (z \cdot x) \bmod n \end{cases}$$

vrati (z)

ElGamal je predložio kriptosustav s javnim ključem koji se temelji na problemu diskretnog logaritma u $(\mathbb{Z}_p^*, \cdot)$.

Operacija šifriranja u ElGamalovom kriptosustavu je nasumična, jer šifrirani tekst ovisi o otvorenom tekstu x i o slučajnoj vrijednosti k koju je izabrala Alice.
Stoga, bit će mnogo šifriranih tekstova (zapravo $p-1$) koji su šifriranja istog otvorenog teksta.

Neformalno, tako funkcionira ElGamal kriptosustav: otvoreni tekst x je "maskiran" množeći ga s β^k. Vrijednost α također se prenosi kao dio šifriranog teksta.
Bob, koji zna privatni ključ α, može izračunati β^k iz α^k. Tada može "ukloniti masku" dijeljenjem y_2 s β^k da bi se dobio x.

Definicija 3.1. *Pretpostavimo da je $a \in \mathbb{Z}_m$. Multiplikativni inverz od a modulo m, označen $a^{-1} \mod m$ je element $a' \in \mathbb{Z}_m$ takav da je $aa' \equiv a'a \equiv 1 \pmod m$. Ako je m fiksan ponekad pišemo a^{-1} za $a^{-1} \mod m$.*

Algoritam 3.2. *([7], Cryptosystem 6.1) ElGamalov kriptosustav javnog ključa u $\mathbb{Z}_p^*$*

Neka je p prost broj takav da je problem diskretnog logaritma u $(\mathbb{Z}_p^*, \cdot)$ nerješiv, i neka je $\alpha \in \mathbb{Z}_p^*$ primitivni korijen modulo p. Neka je $\mathcal{P} = \mathbb{Z}_p^*$, $\mathcal{C} = \mathbb{Z}_p^* \times \mathbb{Z}_p^*$ i definiramo

$$\mathcal{K} = \{(p, \alpha, a, \beta) : \beta \equiv \alpha^a \pmod p\}.$$

Vrijednost p , α i β su javne, a vrijednost a je tajna. Za $\mathcal{K} = (p, \alpha, a, \beta)$ i za tajni slučajno odabrani $k \in \mathbb{Z}_{p-1}$, definiramo

$$e_k(x, k) = (y_1, y_2),$$

gdje

$$y_1 = \alpha^k \mod p$$

i

$$y_2 = x\beta^k \mod p.$$

Za y_1 , $y_2 \in \mathbb{Z}_p^*$, definiramo

$$d_k(y_1, y_2) = y_2(y_1^a)^{-1} \mod p.$$

Primjer 3.1. *Neka su Alice i Bob dogovorili (javno) da koriste grupu* $\mathbb{Z}_{31}^*$ *i fiksni element* $\alpha = 3$. *Neka je* $a = 7$, *tada je*

$$\beta = 3^7 \bmod 31 = 17.$$

Alice želi proslijediti Bobu tajnu informaciju $x = 24$, odabire slučajni broj $k = 5$ i računa:

$$y_1 = 3^5 \bmod 31$$

$$y_1 = 26$$

i

$$y_2 = 24 \cdot 17^5 \bmod 31$$

$$y_2 = 4.$$

Bob sada prima brojeve $y = (26, 4)$ i računa

$$x = 4 \cdot (26^7)^{-1} \bmod 31$$
$$x = 4 \cdot (26)^{-1} \bmod 31$$
$$x = 4 \cdot 6 \bmod 31$$
$$x = 24.$$

ElGamalov kriptosustav bit će nesiguran ako Oscar može izračunati vrijednost $a = \log_\alpha \beta$, jer tad Oscar može dešifrirati šifrirane tekstove točno kao što to radi Bob. Stoga je nužan uvjet da ElGamalov kriptosustav bude siguran da je problem diskretnog logaritma $\mathbb{Z}_p^*$ neizvediv, odnosno teško rješiv. Ovo se općenito smatra slučajem kada je p pažljivo odabran i α primitivan korijen modulo p. Da bi se spriječili poznati napadi, p treba imati najmanje 300 znamenki, a $p - 1$ treba imati barem jedan "veliki" prosti faktor.

3.1 Semantička sigurnost ElGamal sustava

Primjećujemo da osnovni ElGamal kriptosustav nije semantički siguran. Prisjetimo se da je $\alpha \in \mathbb{Z}_p^*$ primitivni korijen i $\beta = \alpha^a \mod p$, gdje je a tajni ključ. Obzirom na element x otvorenog teksta i slučajno odabran broj k, izračunali smo $e_k(x, k) = (y_1, y_2)$ gdje je $y_1 = \alpha^k \mod p$ i $y_2 = x\beta^k \mod p$.

Iskazat ćemo teorem pod nazivom Eulerov kriterij bez dokaza.

Teorem 3.1. *([2], Teorem 4.2) (Eulerov kriterij). Za svaki cijeli broj* a *i neparani prosti broj* p *vrijedi*

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

Za provjeru radi li se o kvadratnom ostatku modulo p koristimo Eulerov kriterij za elemente iz $\mathbb{Z}_p$. β je kvadratni ostatak modulo p ako i samo ako je a paran.
Slično, y_1 je kvadratni ostatak modulo p ako i samo ako je k paran. Možemo odrediti parnost od a i k, prema tome možemo izračunati i parnost od $a \cdot k$. Stoga možemo odrediti je li $\beta^k (= \alpha^{ak})$ kvadratni ostatak. Pretpostavimo da želimo razlikovati enkripcije x_1 od x_2, gdje je x_1 kvadratni ostatak modulo p a x_2 nije. Slijedi da je (y_1, y_2) enkripcija od x_1 ako i samo ako su β^k i y_2 kvadratni ostatci ili oba kvadratni neostatci.

Gore navedeni napad ne funkcionira ako je β kvadratni ostatak i svaki otvoreni tekst x mora biti kvadratni ostatak.
Ako vrijedi da je $p = 2q + 1$, gdje je q prost broj. Tada se može pokazati da je ograničavanje β, y_1, i x na kvadratne ostatke ekvivalentno implementaciji ElGamalovog kriptosustava u podgrupi kvadratnih ostataka modulo p (koja je ciklička podgrupa od $\mathbb{Z}_p^*$ reda q).

4 Algoritmi za problem diskretnog logaritma

Pretpostavimo da je $(G, \cdot)$ multiplikativna grupa i $\alpha \in G$ reda n. Stoga se problem diskretnog logaritma može izraziti u sljedećem obliku: Za dano $\beta \in \langle\alpha\rangle$, pronađimo jedinstveni eksponent a, $0 \leq a \leq n-1$ tako da je $\alpha^a = \beta$.

Započnimo analizom nekih elementarnih algoritama koji se mogu koristiti za rješavanje problema diskretnog logaritma. U ovim analizama pretpostavit ćemo da za izračunavanje umnoška dvaju elemenata u grupi G zahtjeva konstantno (tj. $\mathcal{O}(1)$) vrijeme.
Primjetimo da se problem diskretnog logaritma može riješiti pretraživanjem u $\mathcal{O}(n)$ vremenu i $\mathcal{O}(1)$ prostoru, jednostavnim izračunavanjem $\alpha, \alpha^2, \alpha^3, \ldots$ sve dok se ne pronađe $\beta = \alpha^a$. Također možemo unaprijed izračunati sve moguće vrijednosti α^i, a zatim sortirati listu uređenih parova (i, α^i) s obzirom na njihove druge koordinate. Binarnim pretraživanjem sortirane liste s obzirom na β, možemo pronaći vrijednost a tako da je $\alpha^a = \beta$.

Algoritme za rješavanje problema diskretnom logaritma možemo razvrstati u iduće tri kategorije:

1. Algoritmi koji rade u proizvoljnim grupama tj. ne koriste niti jedno posebno svojstvo grupe, to su: Shanksov algoritam, Pollardov ρ i λ algoritam ...
2. Algoritmi koji radi u grupama čiji red nije djeljiv velikim prostim brojevima, to je Pohlig-Hellmanov algoritam.
3. Algoritmi koji koriste metode koji koriste prikaz elemenata grupe u obliku produkta elemenata odabranih iz relativno malih skupova (tzv. faktorskih baza). Tipični predstavnici ove kategorije su algoritmi koji su varijacije metode računanja indeksa.

U nastavku poglavlja analizirat ćemo neke od navedenih algoritama.

4.1 Shanksov Algoritam

Neka je G konačna ciklička grupa reda n i neka je α generator za G. Tada za svaki $\beta \in G$ imamo da je $\log_\alpha \beta$ element od $\mathbb{Z}_n$, te se stoga može zapisati u obliku $m_j + i \mod n$, gdje je $m = \lceil \sqrt{n} \rceil$; $0 \leq i \leq m-1$ i $0 \leq j \leq m-1$.

Da bismo pronašli $\log_\alpha \beta$, započinjemo računanjem odvojene liste L_1 gdje je

$$L_1 = \{(j, \alpha^{mj}) : 0 \leq j \leq m\}.$$

Za svaki zadani $\beta \in G$ izračunali smo jedan $(i, \beta\alpha^{-i})$, a zatim bismo skenirali popis L_1 za odgovarajući drugi iznos. Ako ga ne pronađemo, prelazimo na sljedeći i i učinimo isto. Kad ga pronađemo, onda možemo riješiti $\log_\alpha \beta$.
Ako je $(i, y) \in L_1$ i nađemo par $(i, y) \in L_2$ imamo

$$\alpha^{mj} = y = \beta\alpha^{-i}$$

ili

$$\alpha^{mj+i} = \beta.$$

Obratno, za bilo koji $\beta \in \langle\alpha\rangle$ imamo $0 \leq \log_\alpha \beta \leq n-1$. Ova jednostavna opcija može se implementirati u $\mathcal{O}(\sqrt{n}\log n)$ vremenu i $\mathcal{O}(\sqrt{n})$ prostoru.

Algoritam 4.1. *([7], Algorithm 6.1) Shanksov algoritam za diskretni logaritam u* $\mathbb{Z}_p$

1. *Neka je* $m = \lceil\sqrt{n}\rceil$, $n = p-1$
2. *Izračunaj* $\alpha^{mj} \mod p$, $0 \leq j \leq m-1$
3. *Poredajte* m *preostalih parova* $(j, \alpha^{mj} \mod p)$ *s obzirom na njihove druge koordinate, dobivajući listu* L_1
4. *Izračunaj* $\beta\alpha^{-i}$, $0 \leq i \leq m-1$
5. *Poredajte* m *uređenih parova* $(i, \beta\alpha^{-i})$ *s obzirom na njihove druge koordinate, dobivajući listu* L_2
6. *Nađi par* $(i, y) \in L_1$ *i par* $(j, y) \in L_2$ *koji imaju jednaku drugu koordinatu*
7. $\log_\alpha \beta = (mj + i) \mod n$.

Pogledajmo kako algoritam funkcionira na sljedećem primjeru.

Primjer 4.1. *Pretpostavimo da želimo pronaći* $\log_3 54$ *u* $(\mathbb{Z}_{71}^*, \cdot)$, *tako da imamo* $\alpha = 3$, $n = 70$, $\beta = 54$ *i* $m = \lceil\sqrt{70}\rceil = 9$. *Tada je*

$$\alpha^9 \bmod 71 = 16.$$

Prvo izračunavamo uređene parove $(j, 16^9 \mod 71)$ za $0 \leq j \leq 9$. Dobivamo listu

$$\begin{array}{lll} (0,1) & (1,16) & (2,43) \\ (3,49) & (4,3) & (5,48) \\ (6,58) & (7,5) & (8,9) \end{array}$$

koju sortiramo da bi dobili L_1.

Druga lista sadrži uređene parove $(i, 54 \cdot 3^{-1} \mod 71)$ za $0 \leq j \leq 9$, tj. tada slijedi:

$$\begin{array}{lll} (0,54) & (1,18) & (2,6) \\ (3,2) & (4,48) & (5,16) \\ (6,29) & (7,57) & (8,19). \end{array}$$

Nakon sortiranja ove liste, dobivamo L_2.

Ako istovremeno nastavimo kroz ove dvije sortirane liste, nalazimo da je $(1,16) \in L_1$ i $(5,16) \in L_2$. Nadalje, možemo izračunati

$$\log_3 54 = (9 \cdot 1 + 5) \bmod 70$$
$$\log_3 54 = 14 \bmod 70.$$

Stoga, može se provjeriti $3^{14} \equiv 54 \pmod{71}$.

Također uočimo da je $(5,48) \in L_1$ i $(4,48) \in L_2$, i računamo

$$\log_3 54 = (9 \cdot 5 + 4) \bmod 70$$
$$\log_3 54 = 49 \bmod 70.$$

Stoga, može se provjeriti $3^{49} \equiv 54 \pmod{71}$.

4.2 Pollardov ρ algoritam

Pollard predlaže elegantan algoritam za određivanje diskretnog logaritma temeljenog na Monte Carlo ideji i nazvao ga ρ metoda. ρ metoda funkcionira tako da prvo definiramo niz elemenata koji će se periodički ponavljati, a zatim traži podudarnost u nizu. S velikom vjerojatnošću podudaranje će dovesti do rješenja problema diskretnog logaritma.
Dvije ključne ideje su funkcija iteracije za generiranje niza i algoritam za pronalaženje ciklusa za otkrivanje podudaranja.

Neka je G ciklička grupa, $\alpha \in G$ reda n i neka je $\beta \in \langle \alpha \rangle$ čiji diskretni logaritam želimo pronaći. Budući da je $\langle \alpha \rangle$ ciklička reda n, možemo tretirati $\log_\alpha \beta$ kao element od $\mathbb{Z}_n$.

Formiramo niz $x_1, x_2, x_3, \ldots$ interativnom primjenom funkcije slučajnog odabira, f. Jednom kad dobijemo dva elementa x_i i x_j u nizu tako da je $x_i = x_j$ i $i < j$, možemo izračunati $\log_\alpha \beta$.
Tražit ćemo podudarnost oblika $x_i = x_{2i}$. Neka su $S_1 \cup S_2 \cup S_3$ particije od G na tri podskupa približno jednake veličine.

Definiramo funkciju f: $\langle\alpha\rangle \times \mathbb{Z}_n \times \mathbb{Z}_n \to \langle\alpha\rangle \times \mathbb{Z}_n \times \mathbb{Z}_n$:

$$f(x,a,b) = \begin{cases} (\beta x, a, b+1) & ako\ je \quad x \in S_1 \\ (x^2, 2a, 2b) & ako\ je \quad x \in S_2 \\ (\alpha x, a+1, b) & ako\ je \quad x \in S_3. \end{cases}$$

Nadalje, svaka trojka (x,a,b) koju formiramo mora imati svojstvo da je $x = \alpha^a\beta^b$. Započinjemo s početnom trojkom koja ima to svojstvo, $(1,0,0)$.
Uočavamo da $f(x,a,b)$ zadovoljava željeno svojstvo ako (x,a,b) radi. Stoga definiramo

$$(x_i, a_i, b_i) = \begin{cases} (1,0,0) & ako\ je \quad i = 0 \\ f(x_{i\text{-}1}, a_{i\text{-}1}, b_{i\text{-}1}) & ako\ je \quad i \geq 1. \end{cases}$$

Uspoređujemo trojke (x_{2i}, a_{2i}, b_{2i}) i (x_i, a_i, b_i) sve dok ne pronađemo vrijednost $i \geq 1$ tako da je $x_{2i} = x_i$.

Kada se to dogodi, imamo:

$$\alpha^{a_{2i}}\beta^{b_{2i}} = \alpha^{a_i}\beta^{b_i}$$

Ako označimo $c = \log_\alpha \beta$ onda mora vrijediti:

$$\alpha^{a_{2i}+cb_{2i}} = \alpha^{a_i+cb_i}.$$

Budući da je α reda n, slijedi:

$$a_{2i} + cb_{2i} \equiv a_i + cb_i \pmod{n}.$$

Ovo možemo zapisati kao:

$$c(b_{2i} - b_i) \equiv a_i - a_{2i} \pmod{n}.$$

Ako je $(b_{2i} - b_i, n) = 1$, tada možemo odrediti c na slijedeći način:

$$c = (a_i - a_{2i})(b_{2i} - b_i)^{-1} \pmod{n}.$$

Algoritam 4.2. *([7], Algorithm 6.2) Pollardov ρ algoritam (G, n, α, β).*

postupak f(x,a,b)

ako je $x \in S_1$

tada $f \leftarrow (\beta \cdot x, a, (b+1) \bmod n)$

inače $x \in S_2$

tada $f \leftarrow (x^2, 2a \mod n, 2b \bmod n)$

inače $f \leftarrow (\alpha \cdot x, (a+1) \bmod n, b)$

vrati f

definiramo particiju $G = S_1 \cup S_2 \cup S_3$

$(x, a, b) \leftarrow f(1, 0, 0)$

$(x', a', b') \leftarrow f(x, a, b)$

dok je $x \neq x'$

$$\textit{radi} \begin{cases} (x,a,b) \leftarrow f(\ x,a,b) \\ (x',a',b') \leftarrow f(\ x',a',b') \\ (x',a',b') \leftarrow f(\ x',a',b') \end{cases}$$

ako je $(b' - b, n) \neq 1$

tada vrati ("neuspjeh")

inače vrati $((a - a')(b' - b)^{-1} \bmod n)$.

Primjer 4.2. *Neka je* $p = 251$ *prost,* $\alpha = 29$ *i* $\beta = 132$. *Izračunat ćemo* $\log_\alpha \beta$. *Pretpostavimo da definiramo skupove* S_1, S_2 *i* S_3 *na sjedeći način:*

$$S_1 = \{x \in \mathbb{Z}_{251}^* : x \equiv 1 \pmod 3\}$$

$$S_2 = \{x \in \mathbb{Z}_{251}^* : x \equiv 0 \pmod 3\}$$

$$S_3 = \{x \in \mathbb{Z}_{251}^* : x \equiv 2 \pmod 3\}.$$

Za $i = 1, 2, \ldots,$ dobivamo trojke (x_{2i}, a_{2i}, b_{2i}) i (x_i, a_i, b_i), tj. slijedi:

i	(x_i, a_i, b_i)	(x_{2i}, a_{2i}, b_{2i})
1	$(132, 0, 1)$	$(105, 0, 2)$
2	$(105, 0, 2)$	$(2, 0, 5)$
3	$(232, 0, 4)$	$(126, 1, 6)$
4	$(2, 0, 5)$	$(204, 4, 24)$
5	$(58, 1, 5)$	$(241, 16, 96)$
6	$(126, 1, 6)$	$(209, 32, 194)$
7	$(63, 2, 12)$	$(115, 33, 195)$
8	$(204, 4, 24)$	$(93, 66, 142)$
9	$(201, 8, 48)$	$(120, 132, 35)$
10	$(241, 16, 96)$	$(115, 28, 140)$
11	$(186, 16, 97)$	$(93, 56, 32)$
12	$(209, 32, 194)$	$(120, 112, 64)$
13	$(37, 33, 194)$	$(115, 198, 10)$
14	$(115, 33, 195)$	$(93, 146, 22)$
15	$(120, 33, 196)$	$(120, 42, 45)$

Uočavamo da je $x_{15} = x_{30} = 120$. Nadalje, računamo

$$\begin{aligned} c &= (33 - 42)(45 - 196)^{-1} \pmod{250} \\ &= (-9 \cdot -151^{-1}) \pmod{250} \\ &= (9 \cdot 101) \pmod{250} \\ &= 159. \end{aligned}$$

Stoga, može se provjeriti $29^{159} \equiv 132 \pmod{251}$.

4.3 Pohlig-Hellmanov algoritam

Pohlig-Hellmanov algoritam se primjenjuje na grupe čije je glavni poredak potencija. Za ovaj algoritam vrijedi: p je prost broj, $\alpha \in \mathbb{Z}_p$ je primitivni element (tj. da se svaki ne-nul element od $\mathbb{Z}_p$ može zapisati kao α^k za neki cijeli broj k) i $\beta \in \mathbb{Z}_p^*$. Naš cilj je odrediti $a = \log_\alpha \beta$, gdje je bez smanjenja općenitosti $0 \leq a \leq n - 1$.

Pretpostavimo da je

$$n = \prod_{i=1}^{k} p_i^{c_i},$$

gdje su p_i različiti prosti brojevi. Glavni korak je izračunavanje $a \bmod p_i^{c_i}$ za svaki i, $1 \leq i \leq k$, tada možemo izračunati $a \bmod n$ koristeći Kineski teorem o ostatcima.

Teorem 4.1. *([2], Teorem 3.7) (Kineski teorem o ostatcima). Neka su $m_1, m_2, ..., m_r$ u parovima relativno prosti prirodni brojevi te neka su $a_1, a_2, ..., a_r$ cijeli brojevi.*
Tada sustav kongruencija

$$x \equiv a_1 \pmod{m_1}, x \equiv a_2 \pmod{m_2}, ..., x \equiv a_r \pmod{m_r} \tag{1}$$

ima rješenje. Ako je x_0 jedno rješenje, onda su sva rješenja od (1) dana s $x \equiv x_0 \pmod{m_1 m_2 \cdots m_r}$.

Dokaz: Neka je $m = m_1 m_2 \cdots m_r$ te neka je $n_j = \frac{m}{m_j}$ za $j = 1, ..., r$. Zbog uvjeta da su $m_1, ..., m_r$ u parovima relativno prosti, imamo da je $(m_j, n_j) = 1$, pa postoji cijeli broj x_j takav da je $n_j x_j \equiv a_j \pmod{m_j}$. Promotrimo broj

$$x_0 = n_1 x_1 + \cdots + n_r x_r.$$

Za njega vrijedi $x_0 \equiv 0 + \cdots + 0 + n_j x_j + 0 + \cdots + 0 \equiv a_j \pmod{m_j}$. Prema tome, x_0 je rješenje od (1). Ako su sada x, x_0 dva rješenja od (1), onda je $x \equiv x_0 \pmod{m_j}$ za $j = 1, ..., r$ pa jer su m_j u parovima prosti, dobivamo da je $x \equiv x_0 \pmod{m}$. □

Pretpostavimo da je $q = p_i$ i $c = c_i$ za svaki i, $1 \leq i \leq k$. Pokazat ćemo kako izračunati $x \equiv a \pmod{q^c}$.
Prvo računamo x kao

$$x = \sum_{i=0}^{c-1} a_i q^i,$$

gdje je $0 \leq a_i \leq q - 1$ za $0 \leq i \leq c - 1$.
Iz toga slijedi da je

$$a = a_0 + a_1 q + \cdots + a_{c-1} q^{c-1} + s q^c,$$

za svaki cijeli broj s.

Prvi korak algoritma je izračunati a_0, koji slijedi iz činjenice da je

$$\beta^{\frac{n}{q}} \equiv \alpha^{\frac{a_0 n}{q}} \pmod{p}. \tag{2}$$

Dokaz kongruencije (2):

$$\begin{aligned} \beta^{\frac{n}{q}} &\equiv (\alpha^a)^{\frac{n}{q}} \pmod{p} \\ &\equiv (\alpha^{a_0 + a_1 q + \cdots + a_{c-1} q^{c-1} + s q^c})^{\frac{n}{q}} \pmod{p} \\ &\equiv (\alpha^{a_0 + Kq})^{\frac{n}{q}} \pmod{p} \end{aligned}$$

(gdje je K cijeli broj)

$$\begin{aligned} &\equiv \alpha^{\frac{a_0 n}{q}} \alpha^{Kn} \pmod{p} \\ &\equiv \alpha^{\frac{a_0 n}{q}} \pmod{p}. \end{aligned}$$

Iz ove jednadžbe lako možemo odrediti a_0.
Sljedeći korak je izračunati $a_1, ..., a_{c-1}$ (ako je $c > 1$). Koji se mogu odrediti iz prikladne generalizirane kongruencije (2).
Prvo, definirajmo $\beta_o = \beta$ i

$$\beta_j = \beta_\alpha^{-(a_0+a_1q+\cdots+a_{j-1}q^{j-1})} \pmod p,$$

za $0 \leq j \leq c-1$. Koristimo prikladnu generaliziranu kongruenciju (2) :

$$\beta_j^{\frac{n}{q^{j+1}}} \equiv \alpha^{\frac{a_j n}{q}} \pmod p. \tag{3}$$

Primjetimo da kad je $j = 0$, kongruencija (3) se svodi na kongruenciju (2).

Dokaz kongruencije (3):

$$\begin{aligned}
\beta_j^{\frac{n}{q^{j+1}}} &\equiv (\alpha^{a-(a_0+a_1q+\cdots+a_{j-1}q^{j-1})})^{\frac{n}{q^{j+1}}} \pmod p \\
&\equiv (\alpha^{a_jq^j+\cdots+a_{c-1}q^{c-1}+sq^c})^{\frac{n}{q^{j+1}}} \pmod p \\
&\equiv (\alpha^{a_jq^j+K_jq^{j+1}})^{\frac{n}{q^{j+1}}} \pmod p
\end{aligned}$$

(gdje je K_j cijeli broj)

$$\begin{aligned}
&\equiv \alpha^{\frac{a_jn}{q}}\alpha^{K_jn} \pmod p \\
&\equiv \alpha^{\frac{a_jn}{q}} \pmod p.
\end{aligned}$$

S obzirom na zadani β_j, jednostavno možemo izračunati a_j iz kongruencije (3).
Primjetimo da se β_{j+1} može izračunati iz β_j pomoću jednostavne relacije ponavljanja nakon što izračunamo a_j. To slijedi iz sljedeće relacije:

$$\beta_{j+1} \equiv \beta_j\alpha^{-a_jq_j} \pmod p. \tag{4}$$

Sad možemo izračunati $a_0, \beta_1, a_1, \beta_2, ..., \beta_{c-1}, a_{c-1}$ naizmjenično primjenjujući kongruencije (3) i (4).

Primjer 4.3. *Pretpostavimo da je* $p = 127$, $\alpha = 7$ *i* $\beta = 14$. p *je prost broj i* α *je primitivni element modulo* p, *imamo da je* $n = p - 1$. *Želimo odrediti* $a = \log_7 14$.

1. Naći proste faktore od n,
   $n = p - 1 = 127 - 1 = 126 = 2^1 3^2 7^1$
   $(q = 2, 3, 7)$.

2. Za $q = 2$ imamo $a = 2^0 a_0$.
   a_0:

$$\beta^{\frac{n}{q}} \equiv (\alpha^{\frac{n}{q}})^{a_0} \pmod{p}$$

$$14^{\frac{126}{2}} \equiv (7^{\frac{126}{2}})^{a_0} \pmod{127}$$

$$-1 \equiv (-1)^{a_0} \pmod{127}$$

   slijedi da je $a_0 = 1$, tj. $a = 1$

$$a \equiv 1 \pmod{2}.$$

3. Za $q = 3$ imamo $a = 3^0 a_0 + 3^1 a_1$.
   a_0:

$$\beta^{\frac{n}{q}} \equiv (\alpha^{\frac{n}{q}})^{a_0} \pmod{p}$$

$$14^{\frac{126}{3}} \equiv (7^{\frac{126}{3}})^{a_0} \pmod{127}$$

$$107 \equiv 107^{a_0} \pmod{127}$$

slijedi da je $a_0 = 1$.
a_1:

$$\beta_1 \equiv \beta\alpha^{-a_0} \pmod{p}$$

$$\beta_1 \equiv 14 \cdot 7^{-1} \pmod{127}$$

$$\beta_1 \equiv 2 \pmod{127}$$

$$2^{\frac{126}{9}} \equiv (7^{\frac{126}{3}})^{a_1} \pmod{127}$$

$$2^{14} \equiv (7^{42})^{a_1} \pmod{127}$$

$$1 \equiv (107)^{a_1} \pmod{127}$$

slijedi da je $a_1 = 0$, tj. $a = 3^0 \cdot 1 + 3^1 \cdot 0$, $a = 1$.

$$a \equiv 1 \pmod{9}.$$

Za $q = 7$ imamo $a = 7^0 a_0$.
a_0:

$$\beta^{\frac{n}{q}} \equiv (\alpha^{\frac{n}{q}})^{a_0} \pmod{p}$$

$$14^{\frac{126}{7}} \equiv (7^{\frac{126}{7}})^{a_0} \pmod{127}$$

$$14^{18} \equiv (7^{18})^{a_0} \pmod{127}$$

$$8 \equiv 64^{a_0} \pmod{127}$$

slijedi da je $a_0 = 4$,tj. $a = 7^0 \cdot 4$, $a = 4$.

$$a \equiv 4 \pmod{7}.$$

Imamo sustav kongruencija:

$$a \equiv 1 \pmod{2}$$
$$a \equiv 1 \pmod{9}$$
$$a \equiv 4 \pmod{7}.$$

Koristeći Kineski teorem o ostatcima, dobijemo

$$a \equiv 109 \pmod{126}.$$

Stoga, može se provjeriti $14 \equiv 7^{109} \pmod{127}$.

4.4 Metoda računanja indeksa

Metoda računanja indeksa ima tri osnovne faze:

1. Generiranje glatkih odnosa koji uključuju elemente u faktorskoj bazi.
2. Rješavanje odgovarajućeg linearnog sustava jednadžbi za pronalaženje logaritma faktorske baze.
3. Korištenje logaritma faktorske baze za određivanje diskretnog logaritma bilo kojeg elementa grupe.

Važno je napomenuti da prve dvije faze metode računanja indeksa ni na koji način ne ovise o elementu čiji logaritam želimo pronaći. U pretpostavci gdje je polje $\mathbb{F}$ generirano s α, gdje pokušavamo pronaći $\log_\alpha \beta$ za $\beta \in \mathbb{F}$, dok ne dođemo do treće faze ne zanima nas što je β. Zbog toga se prve dvije faze često nazivaju fazama predračunavanja, budući da ih možemo riješiti u svakom trenutku kad se sazna o kojem je polju riječ.

Sad ćemo opisati osnovni oblik metode računanja indeksa.
Za grupu G s generatorom α, odabiremo mali broj "prostih brojeva" elemenata koje ćemo smjestiti u faktorsku bazu B zajedno s α. Kad kažemo prost broj, mislimo na to da se element ne može prikazati u obliku produkta elemenata "manjih" od sebe. Na primjer ako je naša grupa polje s prostim brojem elemenata ($\mathbb{Z}_p$), tada u našu faktorsku bazu uključujemo male proste prirodne brojeve. Broj elemenata odabranih za B je mali u usporedbi s veličinom same grupe, jer ako je $|B|$ velik izračun logaritma od faktorske baze bi bio težak. Zatim pokušavamo pronaći potencije generatora α tog faktora u cijelosti među elementima u B. Ako uspijemo, tada imamo glatku relaciju, tj. kongruenciju u grupi koja povezuje logaritme elemenata faktorske baze. Nakon što dobijemo dovoljno mnogo relacija prelazimo na drugu fazu.

U drugoj fazi postavljamo odgovarajući linearni sustav jednadžbi i rješavamo logaritme faktorske baze. Taj sustav moramo riješiti po modulu $p-1$, koji i sam može biti složen. Ako je tako, možda će biti potrebno faktorizirati $p-1$, a zatim primjeniti Kineski teorem o ostatku da bi dobili konačno rješenje.

Kad dobijemo rješenje, prelazimo na treću fazu i pokušavamo pronaći logaritam elementa polja. Množimo element generatorom polja u nadi da će se umnožak moći faktorizirati pomoću faktorske baze. Ako se to dogodi, uzimamo logaritam kongruencije i rješavamo linearnu jednadžbu za $\log_\alpha \beta$. Ako ne, odbacujemo i pokušavamo ponovo dok ne uspijemo.

Algoritam 4.3. *([4], Table 4) Metoda računanja indeksa za pronalaženje* $\log_\alpha \beta = a$

Faza 1: Pronalaženje logaritama faktorske baze:

- *Konstruirati faktorsku bazu* $B = \{p_1, p_2, ..., p_m\}$.
- *Izračunati relacije* $\alpha^{x_j} \equiv p_1^{a_{1j}} p_2^{a_{2j}} \dots p_m^{a_{mj}} \pmod p$ *za* $1 \leq j \leq t$, *gdje je* $t > m$.

Faza 2: Rješavanje logaritama faktorske baze:

- $x_j \equiv a_{1j} \log_\alpha p_1 + a_{2j} \log_\alpha p_2 + \dots + a_{mj} \log_\alpha p_m \pmod{p-1}$
- *Riješite linearan sustav*

$$\begin{bmatrix} a_{11} & a_{12} & \dots & a_{1t} \\ a_{21} & a_{22} & \dots & a_{2t} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mt} \end{bmatrix} \begin{bmatrix} \log_\alpha p_1 \\ \log_\alpha p_2 \\ \vdots \\ \log_\alpha p_m \end{bmatrix} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_t \end{bmatrix} \pmod{p-1}$$

da se dobiju logaritmi faktorske baze.
Faza 3: Izračunajte $\log_\alpha \beta = a$

- *Odaberemo nasumično* s, $1 \leq s \leq p-2$ *i izračunajmo* $\gamma \equiv \beta\alpha^s \pmod p$.
- *Ako je* $\gamma = p_1^{c_1} p_2^{c_2} \dots p_m^{c_m} \pmod p$ *imamo* $\log_\alpha \beta - s \equiv c_1 \log_\alpha p_1 + c_2 \log_\alpha p_2 + \dots + c_m \log_\alpha p_m \pmod{p-1}$.

Pogledajmo na sljedećem primjeru koji će pokazati jednostavnost i učinkovitost metode računanja indeksa.

Primjer 4.4. *Neka je* $p = 3137$ *prost broj. Pretpostavimo da je* $\alpha = 3$ *primitivni element koji se koristi kao baza logaritma po modulu* p *i uzmemo* $B = \{2, 3, 5, 7, 11\}$ *kao faktorsku bazu. Želimo pronaći* $\log_3 125 = ?$

Nakon "nasumičnih" izbora za eksponente, imamo sljedeće odnose:

$$3^{29} \mod 3137 = 70 = 2 \cdot 5 \cdot 7$$

$$3^{65} \mod 3137 = 308 = 2^2 \cdot 7 \cdot 11$$

$$3^{248} \mod 3137 = 1250 = 2 \cdot 5^4$$

$$3^{553} \mod 3137 = 70 = 5^5.$$

Dobivamo kongruencije:

$$\log_3 2 + \log_3 5 + \log_3 7 \equiv 29 \pmod{3136}$$

$$2\log_3 2 + \log_3 7 + \log_3 11 \equiv 65 \pmod{3136}$$

$$\log_3 2 + 4\log_3 5 \equiv 248 \pmod{3136}$$

$$5\log_3 5 \equiv 553 \pmod{3136}.$$

Zapisujemo ih u matričnom obliku

$$\begin{bmatrix} 1 & 1 & 1 & 0 \\ 2 & 0 & 1 & 1 \\ 1 & 4 & 0 & 0 \\ 0 & 5 & 0 & 0 \end{bmatrix} \begin{bmatrix} \log_3 2 \\ \log_3 5 \\ \log_3 7 \\ \log_3 11 \end{bmatrix} = \begin{bmatrix} 29 \\ 65 \\ 248 \\ 553 \end{bmatrix} \bmod 3136$$

i rješavamo linearan sustav.

1. $$\boxed{\log_3 5 = 1365 \pmod{3136}}$$

2. $$\log_3 2 + 4 \cdot 1365 \equiv 248 \pmod{3136}$$

$$\log_3 2 + 5460 \equiv 248 \pmod{3136}$$

$$\log_3 2 \equiv -5213 \pmod{3136}$$

$$\boxed{\log_3 2 \equiv 1060 \pmod{3136}}$$

3. $$\log_3 2 + \log_3 5 + \log_3 7 \equiv 29 \pmod{3136}$$

$$1060 + 1365 + \log_3 7 \equiv 29 \pmod{3136}$$

$$\log_3 7 \equiv -2396 \pmod{3136}$$

$$\boxed{\log_3 7 \equiv 740 \pmod{3136}}$$

4.

$$2\log_3 2 + \log_3 7 + \log_3 11 \equiv 65 \pmod{3136}$$

$$2 \cdot 1060 + 740 + \log_3 11 \equiv 65 \pmod{3136}$$

$$\log_3 11 \equiv -2795 \pmod{3136}$$

$$\boxed{\log_3 11 \equiv 341 \pmod{3136}.}$$

Da bi izračunali $\log_3 125$, "nasumično" odabiremo $s = 132$.
Računamo:

$$125 \cdot 3^{132} \pmod{3137} = 1260 = 2^2 \cdot 5 \cdot 3^2 \cdot 7.$$

Imamo

$$\log_3 125 \equiv (2\log_3 2 + 2\log_3 3 + \log_3 5 + \log_3 7 - s) \pmod{3136}$$

$$\log_3 125 \equiv (2 \cdot 1060 + 2 \cdot 1 + 1365 + 740 - 132) \pmod{3136}$$

$$\log_3 125 \equiv 4095 \pmod{3136}$$

$$\log_3 125 \equiv 959 \pmod{3136}.$$

Stoga, može se provjeriti $3^{959} \equiv 125 \pmod{3137}$.

Literatura

[1] C. Diem, *What is Index Calculus?,* University of Leipzig, Leipzig, 1985.

[2] A. Dujella, *Teorija brojeva*, Školska knjiga, Zagreb, 2021.

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

[4] J.S. Howell, *The Index Calculus Algorithm for Discrete Logarithms* `https://people.clarkson.edu/~jhowell/math/msthesis.pdf`

[5] B. Ibramhipašić, D. Kovačević, Mat-Kol(Banja Luka), Vol. XVII, 2(2011), 43-52 `http://elib.mi.sanu.ac.rs/files/journals/mk/2/mkn2p43-52.pdf`

[6] S. Lindhurst, *An Analysis of Shanks Algorithm for Computing Square Roots in Finite Fields*, University of Wisconsin, Madison, 1997.

[7] D.R.Stinson, *Cryptography theory and practice*, Chapman and Hall/CRC, Boca Raton, 2006.

Sažetak

U ovom radu bavili smo se kriptografijom javnog ključa i problemom diskretnog logaritma. Definirali smo osnovne pojmove u kriptografiji i njezinu glavnu svrhu. Opisali smo ElGamalov kriptosustav s javnim ključem te smo ga prikazali kroz primjer. Nadalje, obradili smo njegovu semantičku sigurnost. Na kraju smo detaljno opisali i kroz primjere prikazali algoritme koji se bave problemom diskretnog logaritma, a to su: Shanksov Algoritam, Pollardov rho algoritam, Pohlig-Hellmanov algoritam i Metoda računanja indeksa.

Ključne riječi: kriptosustav s javnim ključem, problem diskretnog logaritma, ElGamalov kriptosustav, Shanksov Algoritam, Pollardov rho algoritam, Pohlig-Hellmanov algoritam, Metoda računanja indeksa.

Public-key Cryptography and Discrete Logarithms

Summary

In this work, we defined public key cryptography and the discrete logarithm problem. We have defined the basic terms in cryptography and it's main purpose. We described ElGamal's public key cryptosystem and demonstrated it through an example. Furthermore, we have processed it's semantic security. At the end, finally we described in detail and presented through examples the algorithms that deal with the discrete logarithm problem, namely: Shanks' Algorithm, Pollard rho algorithm, Pohlig-Hellman algorithm and Index Calculus Method.

Keywords: public key cryptosystem, discrete logarithm problem, ElGamal's cryptosystem, Shanks' Algorithm, Pollard rho algorithm, Pohlig-Hellman Algorithm and Index Calculus Method.

Životopis

Rođen sam 13.10.1990. godine u Šibeniku, te tamo završavam osnovnoškolsko i srednjoškolsko obrazovanje. Nakon završene srednje škole upisujem Preddiplomski sveučilišni studij Matematika; smjer-nastavnički na Matematičkom odsjeku PMF-a u Zagrebu, gdje stječem akademski naziv; sveučilišni prvostupnik edukacije matematike.

Na Odjelu za matematiku Sveučilišta Josipa Jurja Strossmayera u Osijeku upisujem Diplomski studij financijske matematike i statistike.
U Osijeku na Filozofskom fakultetu upisujem i završavam program Pedagoško-psihološko-didaktičko-metodičke izobrazbe te stječem pedagoške kompetencije.