# Specijalni kurs

# Kriptografija

## 1. Navesti razlike između simetričnih i asimetričnih kriptosistema

**Kriptosistem** je par koji čine algoritam za kriptovanje i algoritam za dekriptovanje. Algoritmi su najčešće svima poznati i uvek zavise od parametra koji se zove **ključ** i koji se potpuno ili delimično čuva u tajnosti. U komunikaciji obično imamo sledeći scenario: Alisa šalje poruku Bobanu, a Cica krade šifrat i pokušava da ga dekriptuje ne znajući ključ, što se naziva **kriptoanaliza**. Kriptosistemi mogu biti:

- **simetrični** - Alisa i Boban koriste isti ključ za (de)kriptovanje. Unapred dogovore ključ, čuvaju ga u tajnosti i periodično ga menjaju. Ovi kriptosistemi su nepraktični kada imamo veliki broj korisnika gde svako komunicira sa svakim. Nedostaci su i to što je potreban dodatni kanal komunikacije za razmenu ključa, kao i to što su ovi sistemi lakši za kriptoanalizu.
- **asimetrični** (**kriptosistemi sa javnim ključem**) - Alisa i Boban prave sopstvene ključeve, pri čemu ključ za kriptovanje objavljuju, a ključ za dekriptovanje čuvaju u tajnosti. Nedostatak ovih kriptosistema je sporo izvršavanje pa se obično koriste za prenos kratke poruke kada je potrebna visoka bezbednost.

Najbolje rezultate zapravo daje kombinovanje ova dva pristupa tako što se za slanje poruke koristi simetrični kriptosistem, a njegov ključ se razmenjuje asimetričnim kriptosistemom.

## 2. Cezarova i afina šifra i njihova kriptoanaliza

**Cezarova šifra**: Slova A-Z kodiramo sa $\{0, \ldots, 25\} = \mathcal{Z}_{26}$ i koristimo protočnu šifru $f(P) = P + b \; mod \; 26$, gde je $P \in \mathcal{Z}_{26}$ kodirani simbol, a $b$ tajni ključ. Algoritam za dekodiranje bi bio $f^{-1}(C) = C - b \; mod \; 26$. Na primer, ako je tajni ključ $b = 16$ i Alisa šalje poruku "PAYMENOW" dobijamo šifrat:
"PAYMENOW" $\rightarrow$ 15 0 24 12 4 13 14 22 $\xrightarrow{f}$ 5 16 14 2 20 3 4 12 $\rightarrow$ "FQOCUDEM"

**Afina šifra** predstavlja uopštenje Cezarove šifre:

$$f(P) = aP + b \; mod \; 26, \; f^{-1}(C) = a'C + b' \; mod \; 26,$$

gde je $a' = a^{-1}$ inverz od $a$ u $\mathcal{Z}_{26}^*$ i $b' = -a^{-1}b$.

Kriptoanaliza: poznato je da je najfrekventnije slovo u tekstu na engleskom jeziku slovo 'E'. Cica pronalazi najfrekventnije slovo u šifratu (npr. slovo 'K') i pretpostavlja da je $f$('E') = 'K',

odnosno $f^{-1}$('K') = 'E'. Slično uspostavlja vezu između drugog najfrekventnijeg slova u tekstu na engleskom jeziku i drugog najfrekventnijeg slova u šifratu (npr. $f$('T') = 'D'). Cica rešavanjem sistema pronalazi ključ. Ako sistem nema rešenje ili ono nije jedinstveno, umesto drugog najfrekventnijeg slova može da koristi treće, četvrto i slično.

## 3. Jednokratna šifra (One Time Pad)

**Jednokratna šifra (One Time Pad)** je najjednostavnija protočna šifra. Poruka $M$ kodira se binarno. Ključ $K$, koji je takođe zapisan binarno, mora biti iste dužine kao $M$. Kriptovanje se vrši bit po bit, sabiranjem bita iz $M$ i bita iz $K$ po modulu 2. Dekriptovanje se radi identično sa istim ključem. Ključ sme da se koristi samo jednom, a ponovna upotreba istog ključa dovodi do curenja podataka.

## 4. Matrično kriptovanje digrafa

Umesto sa pojedinačnim slovima, bolje je raditi sa većim blokovima slova. **Digrafovi** par slova kodiraju brojem iz skupa $\{0, \ldots, 26^2 - 1 = 675\}$. Na primer, digraf "NO" se kodira sa 26 · 'N' + 'O' = 26 · 13 + 14 = 352, a zatim se kriptuje sa 159 · 352 + 580 = 440 mod 676 što je ekvivalent "QY", pri čemu su 159 i 580 neki ključevi.

Neka je $A \in M_2(\mathcal{Z}_n)$ invertibilna matrica, tj. takva da je $\det A$ invertibilno u $\mathcal{Z}_n$. Algoritam za kriptovanje i dekriptovanje digrafa onda može biti:

$$\begin{pmatrix} x \\ y \end{pmatrix} \xrightarrow{f} A \begin{pmatrix} x \\ y \end{pmatrix} \; i \; \begin{pmatrix} x \\ y \end{pmatrix} \xrightarrow{f^{-1}} A^{-1} \begin{pmatrix} x \\ y \end{pmatrix}$$

Na primer, neka je $n = 26$ i $A = \begin{pmatrix} 2 & 3 \\ 7 & 8 \end{pmatrix}$ i neka Alisa šalje poruku "NO|AN|SW|ER": $\begin{pmatrix} 2 & 3 \\ 7 & 8 \end{pmatrix} \begin{pmatrix} 13 & 0 & 18 & 4 \\ 14 & 13 & 22 & 17 \end{pmatrix} = \begin{pmatrix} 16 & 13 & 24 & 7 \\ 21 & 0 & 16 & 8 \end{pmatrix}$ što je "QVNAYGHI". Sa druge strane, ako Boban dobije poruku "FW|MD|IQ" on će je pročitati pomoću $A^{-1} = \begin{pmatrix} 14 & 11 \\ 17 & 10 \end{pmatrix}$: $\begin{pmatrix} 14 & 11 \\ 17 & 10 \end{pmatrix} \begin{pmatrix} 5 & 12 & 8 \\ 22 & 3 & 16 \end{pmatrix} = \begin{pmatrix} 0 & 19 & 2 \\ 19 & 0 & 10 \end{pmatrix}$ što je "ATTACK".

## 5. Jednosmerne funkcije. Navesti primer jednosmerne funkcije

Kažemo da je funkcija $f : X \to Y$ **jednosmerna** ako se za poznato $x \in X$ lako (brzo) može izračunati $f(x)$, ali je za poznato $y \in Y$ teško (neizvodljivo u realnom vremenu) izračunati $f^{-1}(y)$. Dovoljno je da je $f$ injekcija, a najčešće je bijekcija. Primer korišćenja jednosmerne funkcije $f$: Neki internet servis za svakog korisnika čuva par $(N,\ f(P))$ gde je $N$ korisničko ime, a $P$ šifra. Kada se prilikom prijavljivanja unesu $N$ i $P'$ lako se računa $f(P')$ i proverava da li se poklapa sa $f(P)$. U slučaju da Cica ukrade podatke $(N,\ f(P))$, ona ne može da izračuna $P$. Najčešće je vremenska složenost funkcija $f$ i $f^{-1}$ redom $O(n^k)$ i $O(l^n)$. Kriptosistemi sa javnim ključem zasnivaju se na jednosmernim funkcijama.

## 6. Difi-Helmanov algoritam za usaglašavanje ključeva

Ako je $p$ prost broj, tada je $(\mathcal{Z}_p,\ +_p,\ \cdot_p)$ polje, gde je $\mathcal{Z}_p = \mathcal{Z}/(p\mathcal{Z}) = \{0,\ 1,\ \ldots,\ p-1\}$. Multiplikativna grupa $\mathcal{Z}_p^* = (\mathcal{Z}_p\backslash\{0\},\ \cdot_p)$ je ciklična, tj. postoji generator (primitivni koren) $g \in \mathcal{Z}_p\backslash\{0\}$ tako da se svi elementi $\mathcal{Z}_p\backslash\{0\}$ mogu videti kao stepeni $g$. **Difi-Helmanova razmena ključa** zasniva se na:
♣ ako znamo $g \in \mathcal{Z}_p^*$ i $n \in N$ lako je odrediti $g^n$
♠ ako znamo $g$ i $g^n$ teško je odrediti $n$

Algoritam:

- Alisa i Boban biraju prost broj $p$ (približno 200-cifren) i generator $g \in \mathcal{Z}_p^*$ i objavljuju $p$ i $g$
- Alisa bira svoj tajni ključ $a_A \in N$, računa i objavljuje javni ključ $g^{a_A}\ mod\ p$
- Boban bira svoj tajni ključ $a_B \in N$, računa i objavljuje javni ključ $g^{a_B}\ mod\ p$
- Alisa i Boban mogu izračunati usaglašeni ključ $K = (g^{a_A})^{a_B}\ mod\ p$ = $(g^{a_B})^{a_A}\ mod\ p$
- Cica zna samo $p,\ g,\ g^{a_A}$ i $g^{a_B}$ i ne može lako (brzo) da odredi $K$

## 7. Algoritam za stepenovanje ponovljenim kvadriranjem

Brzi algoritam za ♣ zasniva se na **ponovljenom kvadriranju** i sadrži sledeće korake (sve operacije su po modulu $p$):

1. Redukujemo stepen na $n < p-1$ zbog Male Fermaove teoreme: $g^{p-1} = 1$ u $\mathcal{Z}_p$
2. Zapisujemo $n$ binarno:

$$n = \overline{n_r n_{r-1} \ldots n_1 n_0} = \sum_{i=0}^{r} n_i \cdot 2^i,\ n_i \in \{0,\ 1\}$$

3. Računamo

$$1,\ g,\ g^2,\ (g^2)^2 = g^{2^2},\ \ldots,\ (g^{2^{r-1}})^2 = g^{2^r}$$

4. $g^n$ je proizvod onih stepena $g$ za koje je $n_i = 1$:

$$g^n = g^{\sum_{i=0}^{r} n_i \cdot 2^i} = \prod_{i=0}^{r} g^{n_i \cdot 2^i} = \prod_{0 \le i \le r,\ n_i = 1} g^{2^i}$$

Za množenje $k$-bitnog broja $a$ i $l$-bitnog broja $b$ treba $kl$ operacija, gde je $k \sim \log a$ i $l \sim \log b$, a isto važi i za celobrojno deljenje i uzimanje ostatka po modulu. Vremenska složenost algoritma je onda:

1. $O(\log^2 p)$ jer je $\log p$ broj bitova broja $p$
2. $O(r \log n) = O(\log^2 p)$ jer $r$ puta ponavljamo deljenje sa 2 i ostatak po modulu 2, a $r \sim \log n \le \log p$
3. $r$ kvadriranja, a za svako kvadriranje treba po $O(\log^2(g^{2^i})) = O(\log^2 p)$ operacija
4. najviše $r$ množenja koja zahtevaju po najviše $O(\log^2 p)$ operacija

Dakle, ukupno $O(r \log^2 p) = O(\log^3 p)$. Za množenje $g^n = g \dots g$ bi trebalo $O(n \log^2 p)$ operacija.

Algoritam radi i za modul $m$ koji nije prost, ali može da radi sporije jer se umesto Male Fermaove koristi Ojlerova teorema.

## 8. Definisati diskretni logaritam. Navesti 3 kriptosistema koji se zasnivaju na problemu diskretnog logaritma

Neka je $G$ grupa (npr. $F_q^*$) i neka su $a,\ g \in G$. Najmanji prirodan broj $n$, ako postoji, takav da je $a = g^n$ zovemo **diskretni logaritam** od $a$ u osnovi $g$ i označavamo sa $\log_g a$. Ne postoji formula za izračunavanje diskretnog logaritma, već redom računamo stepene i čekamo da se pojavi tražena vrednost. U najgorem slučaju $g$ je generator i tada moramo proći sve stepene što zahteva $O(q)$ testiranja, a vremenska složenost stepenovanja je $O(\log^4 q)$. Kriptosistemi koji se zasnivaju na problemu diskretnog logaritma su Difi-Helman, Mesi-Omura i ElGamal.

## 9. Algoritam Geljfond-Šenksa (Baby-step-giant-step algoritam)

**Algoritam Geljfond-Šenksa** smanjuje vremensku i prostornu složenost pretraživanja diskretnog logaritma sa $O(q)$ na $O(\sqrt{q} \log^2 q)$. Dakle, cilj je izračunati $n = \log_g a$ u $F_q^*$, gde su $a$ i $g$ poznati. Prvo zapišemo $n$ kao $mi + j$, gde je $m = [\sqrt{q}]$ i $0 \leq i \leq m$, $0 \leq j \leq m - 1$. Tada važi $\log_g a = n \Leftrightarrow g^n = a \Leftrightarrow g^{mi+j} = a \Leftrightarrow g^j = a(g^{-m})^i$. Računamo parove $(j,\ g^j)$ i $(i,\ a(g^{-m})^i)$ za sve $i,\ j$ i čuvamo ih sortirane po drugoj koordinati. Kada nađemo $i$ i $j$ za koje se poklapa druga koordinata lako dolazimo do $n$.

## 10. Polig-Helmanov algoritam

Neka su $q,\ B \in N$. Za broj $q = p_1^{\alpha_1} \dots p_k^{\alpha_k}$ kažemo da je $B$**-gladak** ako za sve $1 \leq i \leq k$ važi $p_i^{\alpha_i} \leq B$. **Polig-Helmanov metod** omogućava brzo izračunavanje diskretnog logaritma u $F_q^*$, sa pretpostavkama da je $B \ll n$ (npr. $B \sim \log n$) i da je $q - 1$ $B$-gladak. Dakle, cilj je izračunati $n = \log_g a$ u $F_q^*$, gde su $a$ i $g$ poznati. Zbog Male Fermaove teoreme $n$ tražimo kao ostatak po modulu $q - 1$. Ako je $q - 1 = p_1^{\alpha_1} \dots p_k^{\alpha_k}$, po Kineskoj teoremi o ostacima dovoljno je odrediti $n$ kao ostatak po modulu $p_i^{\alpha_i}$ za svako $i$. Računanje $n$ po modulu $p^\alpha$:

- Računamo $\zeta_p = g^{\frac{q-1}{p}}$, a zatim i $\zeta_p^0$, $\zeta_p^1$, ..., $\zeta_p^{p-1}$ i sve parove $(j,\ \zeta_p^j)$ čuvamo u nekoj tabeli. Vrednosti $\zeta_p^j$ zovemo $p$-ti koren iz 1, jer su rešenja jednačine $x^p \equiv 1\ (mod\ q - 1)$.
- Zapišemo $n \equiv n_0 + n_1 p + \dots + n_{\alpha-1} p^{\alpha-1}\ (mod\ p^\alpha)$. Potrebno je odrediti vrednosti $n_i$.
- Računamo $a^{\frac{q-1}{p}}$, međutim važi $a^{\frac{q-1}{p}} = g^{\frac{n(q-1)}{p}} = \zeta_p^n = \zeta_p^{n_0}$. U tabeli imamo sve vrednosti $\zeta_p^j$ pa lako pronalazimo vrednost $n_0$.
- Računamo $(\frac{a}{g^{n_0}})^{\frac{q-1}{p^2}} = g^{\frac{(n-n_0)(q-1)}{p^2}} = \zeta_p^{\frac{n-n_0}{p}} = \zeta_p^{n_1}$ pa iz tabele dobijamo $n_1$. Slično, $(\frac{a}{g^{n_0+n_1 p}})^{\frac{q-1}{p^3}}$ daje $n_2$, $(\frac{a}{g^{n_0+n_1 p+n_2 p^2}})^{\frac{q-1}{p^4}}$ daje $n_3$ i tako dalje.

Ponovo nemamo računanje diskretnog logaritma, već traženje vrednosti u tabeli. Dodatno, sada i čuvamo tabelu, odnosno trošimo memoriju, međutim dužina tabele je manja od $B$. Za

svako $p$ imamo $\alpha$ puta ponavljanje računa i na kraju Kinesku teoremu. Vremenska složenost je polinom od $B$, što je prihvatljivo za malo $B$. Dakle, ako Alisa i Boban izaberu javni ključ $q$ tako da je $q-1$ $B$-gladak, Cica može da dekriptuje poruku za dovoljno malo $B$. Alisa i Boban mogu proveriti $B$-glatkost za neko malo $B$, čime pokrivaju i sve vrednosti manje od $B$, ali uvek postoji opasnost od $(B+1)$-glatkosti gde je vremenska složenost praktično ista kao za $B$.

## 11. Mesi-Omura kriptosistem

**Mesi-Omura kriptosistem** koristi se za razmenu ključeva ili poruka. Ako je poruka duža, deli se na blokove. Algoritam:

- Fiksira se konačno polje $F_q$ i to je svima poznato, odnosno $q$ je javni ključ.
- I Alisa i Boban biraju svoje tajne ključeve $e_A$ i $e_B$ tako da važi $NZD(e_A,\ q-1)$ = $NZD(e_B,\ q-1)$ = 1.
- Svako od njih računa svoj tajni ključ $d_i \equiv e_i^{-1}\ (mod\ q-1)$, $i \in \{A,\ B\}$.
- Ako je $M$ blok poruke kodiran elementom polja $F_q$ koju treba poslati Alisa, ona računa $M^{e_A}$ i to šalje Bobanu.
- Boban ne može da pročita $M^{e_A}$, ali može da izračuna $(M^{e_A})^{e_B}$ = $M^{e_Ae_B}$ i to šalje nazad Alisi.
- Alisa računa $(M^{e_Ae_B})^{d_A} = M^{e_B}$ i šalje opet Bobanu. Ovde se koristi $e_Ad_A \equiv 1\ (mod\ q-1)$ što povlači $M^{e_Ad_A} = M$ u $F_q$.
- Boban računa $(M^{e_B})^{d_B} = M$ i dolazi do početne poruke.

## 12. Alisa šalje Bobanu poruku pomoću Mesi-Omura kriptosistema i pretpostavljamo da je Cica videla celokupnu komunikaciju. Objasniti zašto Cica ipak ne može da dekriptuje poruku

Ako Cica presretne komunikaciju najviše što može da zna je $M^{e_A}$, $M^{e_B}$, $M^{e_Ae_B}$ i javni ključ $q$. Da bi došla do $M = (M^{e_A})^{d_A}$ mora da izračuna $e_A = \log_{M^{e_B}}(M^{e_Ae_B})$ i $d_A \equiv e_A^{-1}\ (mod\ q-1)$, što ne može lako (brzo) da uradi. Kao dodatna zaštita ključevi $e_A$, $e_B$, $d_A$ i $d_B$ mogu se menjati kod svakog bloka. Ponekad se kaže da Mesi-Omura nije ni simetričan ni asimetričan kriptosistem jer se smatra da ima samo javni ključ $q$, a ostalo su parametri koji se jednokratno generišu.

## 13. ElGamalov kriptosistem

U **ElGamalovom kriptosistemu** javni ključ $q$ je stepen nekog prostor broja i $g \in F_q^*$ je generator. Algoritam:

- Boban bira svoj tajni ključ $e_B$ i pomoću njega pravi javni ključ $g^{e_B}$ koji šalje Alisi, tj. objavljuje kao kod Difi-Helmana. Ovaj ključ se šalje samo jednom na početku.
- Neka je $M \in F_q$ kodirani blok koji Alisa želi da pošalje. Ona generiše slučajan prirodan broj $k < q$ koji će koristiti samo jednom za blok $M$. Za naredni blok bira novo $k$.
- Alisa šalje Bobanu par informacija $g^k$ i $Mg^{e_Bk}$ = $M(g^{e_B})^k$.

- Boban računa $g^{e_B k} = (g^k)^{e_B}$, a zatim i njegov inverz i dobija $M = Mg^{e_B k}(g^{e_B k})^{-1}$.

Da bi pročitala poruku, Cica mora da reši problem diskretnog logaritma, što ne može lako (brzo) da uradi.

## 14. Kako se generiše slučajan veliki prost broj

Znamo kako radi generator pseudoslučajnih prirodnih brojeva, ali nasumično izabran broj verovatno nije prost, a ne postoji ni formula za $n$-ti prost broj $p_n$. Prvo biramo neparan (veliki) pseudoslučajan broj $n$. Zatim prost broj tražimo u nizu $n$, $n+2$, $n+4$, ... Očekujemo da prethodni korak ne traje dugo, jer je razmak između uzastopnih prostih brojeva reda: $p_{m+1} - p_m$ = $(m+1)\log(m+1) - m\log m \sim \log m \sim \log n$. Potreban nam je efikasan način da proverimo da li je neki broj prost. Elementarno rešeto je sporo - ima vremensku složenost $O(\sqrt{n})$.

## 15. Testovi primalnosti. Šta su ulazni i izlazni podaci kod testa primalnosti

**Testovi primalnosti** su napravljeni tako da ako broj $n$ padne na testu onda je on složen. Ako broj $n$ prođe test on može, ali ne mora biti prost. Verovatnoća da broj $n$ koji je prošao test bude prost je:

$$v = \frac{card\{n \in [1,\ N] \mid n\ je\ prost\}}{card\{n \in [1,\ N] \mid n\ je\ prošao\ test\}}$$

Test primalnosti je bolji ako je $v$ veće. Obično zavisi od nekih parametara. Ponavljanje testa za razne (nezavisne) parametre povećava verovatnoću da je broj koji je preživeo sva testiranja zaista prost. Test primalnosti mora da radi brzo. Dakle, ulaz za test predstavlja broj $n$, a izlaz broj $v$ koji predstavlja verovatnoću da je $n$ prost.

## 16. Definisati pseudoproste i Karmajklove brojeve. Šta je glavni nedostatak Karmajklovog testa primalnosti

**Mala Fermaova teorema** kaže da za prost broj $n$ i $a$ za koje je $NZD(a,\ n) = 1$ važi $a^{n-1} \equiv 1\ (mod\ n)$ $(\star)$. Međutim, nije nemoguće da MFT važi i ako $n$ nije prost broj. Ako za prirodan broj $a$ i složen broj $n$ tako da $NZD(a,\ n) = 1$ važi MFT, kažemo da je $n$ **pseudoprost broj** u bazi $a$. **Ojlerova teorema** kaže da važi $a^{\phi(n)} \equiv 1\ (mod\ n)$, ali $\phi(n)$ obično ne znamo. Ako je $n$ pseudoprost u bazi $a$, onda mora da važi i $a^{n-\phi(n)-1} \equiv 1\ (mod\ n)$. Ovde možemo tražiti sve baze $a$ za koje je $n$ pseudoprost broj jer je vrednost $n - \phi(n) - 1$ obično mnogo manja od $n-1$. Teorema:

1. Ako je $n$ pseudoprost i u bazi $a$ i u bazi $b$ onda je pseudoprost i u bazi $ab$
   △:
   $(ab)^{n-1} = a^{n-1}b^{n-1} \equiv 1 \cdot 1\ (mod\ n)$ ■
2. Ako je $n$ pseudoprost u bazi $a$, a nije pseudoprost u bazi $b$, onda nije pseudoprost ni u bazi $ab$
   △:
   $(ab)^{n-1} = a^{n-1}b^{n-1} \equiv b^{n-1} \not\equiv 1\ (mod\ n)$ ■

3. Ako $n$ nije pseudoprost u bazi $a$, a onda nije pseudoprost ni u bazi $b$, bar za pola $b$-ova iz $Z_n^*$ = $\{b \in Z_n \mid NZD(b,\ n) = 1\}$
△:

   Za svaku bazu $c$ u kojoj je $n$ pseudoprost postoji baza $b = ca$ u kojoj $n$ nije pseudoprost na osnovu 2. ■

**Karmajklov broj** $n$ je složen broj koji je pseudoprost u svakoj bazi $a \in Z_n^*$. Po prethodnoj teoremi verovatnoća da broj $n$ koji nije ni prost ni Karmajklov prođe test $(\star)$ u jednom testiranju sa slučajno izabranom bazom je najviše $\frac{1}{2}$, a u $k$ testiranja sa slučajno i nezavisno izabranim bazama je najviše $\frac{1}{2^k}$. Test je vrlo efikasan, ali ne odvaja proste od Karmajklovih brojeva. Svaki Karmajklov broj je oblika $p_1p_2 \ldots p_k$, gde je $k \geq 3$ i $p_i$-ovi su međusobno različiti prosti brojevi. Beskvadratan broj $n$ je Karmajklov akko za svaki prost broj $p_i$ važi $p_i|n \ \Rightarrow \ p_i - 1|n - 1$. Dakle, Karmajklovi brojevi zaista postoje i moramo da imamo test koji ih odvaja od prostih, što Karmajklov test ne može da uradi i to predstavlja njegov glavni nedostatak.

## 17. Miler-Rabinov test primalnosti

Posledica MFT: Ako je $p$ neparan prost broj i $a \in Z$ tako da $NZD(a,\ p) = 1$, onda je $a^{\frac{p-1}{2}} \equiv \pm 1\ (mod\, p)$. Broj na desnoj strani zovemo **Ležandrov simbol** i označavamo sa $(\frac{a}{p})$.
△:
$a^{p-1} \equiv 1\ (mod\, p) \ \Rightarrow \ p|a^{p-1} - 1$ = $(a^{\frac{p-1}{2}} - 1)(a^{\frac{p-1}{2}} + 1)$, pa $p$ deli bar jednu zagradu jer je prost. ■

Ako za prirodan broj $a$ i neparan složen broj $p$ tako da $NZD(a,\ p) = 1$ važi MFT kažemo da je $p$ **Ojlerov pseudoprost broj** u bazi $a$. Ako je $p$ Ojlerov pseudoprost u bazi $a$, onda je on i pseudoprost u bazi $a$.
△:
$a^{p-1} = (a^{\frac{p-1}{2}})^2 \equiv_p (\pm 1)^2 = 1$ ■

Pokazali smo $a^{p-1} \equiv 1\ (mod\, p) \ \Rightarrow \ a^{\frac{p-1}{2}} \equiv \pm 1\ (mod\, p)$, ali ako je $a^{\frac{p-1}{2}} \equiv 1\ (mod\, p)$ i $\frac{p-1}{2}$ parno, ovo možemo ponoviti i dobiti $a^{\frac{p-1}{4}} \equiv \pm 1\ (mod\, p)$. Postupak ponavljamo sve dok je $a^{\frac{p-1}{2^i}} \equiv \pm 1\ (mod\, p)$ i $\frac{p-1}{2^i}$ parno i dobijamo $a^{\frac{p-1}{2^{i+1}}} \equiv \pm 1\ (mod\, p)$. Dobijamo kongruentan niz vrednosti $1,\ 1,\ \ldots,\ 1,\ -1,\ \ldots$, gde su uzastopne jedinice na početku $k \geq 0$ puta, a posle $-1$ dolazi bilo šta.

**Miler-Rabinov test primalnosti**: Neka je $n$ neparan i $a \in Z$ tako da $NZD(a,\ n)$ = 1. Zapišemo $n - 1$ = $2^r d$, gde je $d$ neparan i za svako $0 \leq j \leq r - 1$ računamo $a_j = a^{2^j d}\ (mod\, n)$. Broj $n$ prolazi Miler-Rabinov test u bazi $a$ ako je ispunjen jedan od uslova:

1. $a_0 = 1$
2. postoji $0 \leq s \leq r - 1$ tako da $a_s = -1$

Primetimo da je $a_{j+1} \equiv a_j^2\ (mod\, n)$, pa na osnovu 1. sledi da su svi $a_j = 1$, a na osnovu 2. da su svi $a_j = 1$ za $j > s$. Za složen broj $n$ koji prolazi Miler-Rabinov test u bazi $a$ kažemo da je **jako pseudoprost broj** u bazi $a$.

## 18. Veza između pseudoprostih i jako pseudoprostih brojeva. Efikasnost Karmajklovog i Miler-Rabinovog testa

Ako je broj $n$ jako pseudoprost u bazi $a$, onda je on i pseudoprost u bazi $a$.
△:
$a^{n-1} \equiv_n a_{r-1}^2 = 1$ ■

Kao posledica, efikasnost Miler-Rabinovog testa je veća nego efikasnost Karmajklovog testa. Teorema:

1. Ne postoji složen broj $n$ koji je jako pseudoprost u svakoj bazi $a \in Z$ tako da $NZD(a,\ n)$ = 1.
2. Ako je $n$ neparan složen broj, onda on može biti jako pseudoprost za najviše četvrtinu baza $a \in Z_n^*$.

Dakle, efikasnost Miler-Rabinovog testa je najmanje $1 - \frac{1}{4^k}$, gde je $k$ broj testiranja.

## 19. Rivest-Šamir-Ejdelman kriptosistem

**Rivest-Šamir-Ejdelman (RSA) algoritam** pomoću kojeg Alisa šalje Bobanu poruku ili ključ:

- Boban tajno bira dva velika prosta broja $p$ i $q$, računa $n = pq$ i računa $\phi(n)$ = $(p-1)(q-1)$. Zatim, bira broj $1 \leq e \leq \phi(n)$ tako da važi $NZD(e,\ \phi(n))$ = 1 i računa $d = e^{-1}\ (mod\ \phi(n))$. Alisi šalje javni ključ $(n,\ e)$, a $d$ čuva kao svoj tajni ključ. U ovom trenutku Boban može da zaboravi $p$ i $q$, ali nikako ne sme da ih objavljuje.
- Neka je poruka koju Alisa želi da pošalje $M < n$. Ona računa $N = M^e\ (mod\ n)$ i to šalje Bobanu.
- Boban pomoću tajnog ključa računa $N^d \equiv M^{ed} \equiv M\ (mod\ n)$. Ovde se koristi $ed \equiv 1\ (mod\ \phi(n))$ i Ojlerova teorema.

Cica vidi $n$, $e$ i $N$, ali ne može da dođe do poruke $M$ sve dok ne odredi $d$.

## 20. Prividno jednosmerne funckije. Navesti primer prividno jednosmerne funkcije

Funkcija $f(M) = M^e\ (mod\ n)$ je primer **prividno jednosmerne funkcije**. To znači:

- $f$ je jednosmerna za Alisu i Cicu, tj. one ne mogu da odrede $f^{-1}$ u realnom vremenu.
- $f$ nije jednosmerna za Bobana, tj. on bi mogao da odredi $f^{-1}$ jer ima podatak o faktorizaciji $n = pq$.

## 21. Fermaov metod faktorizacije

**Fermaov metod faktorizacije** pretpostavlja da je $n = pq$, gde su $p$ i $q$ slične veličine. Brojevi $p$ i $q$ ne moraju biti prosti, ali jesu ako govorimo o RSA. Zapišemo $n$ = $pq$ = $s^2 - t^2$, gde su $s = \frac{p+q}{2}$ i $t = \frac{p-q}{2}$ prirodni brojevi. Problem se svodi na nalaženje $s$ i $t$, pri čemu je $s > \sqrt{n}$ i $t$

malo. U nizu $s_1 = [\sqrt{n}] + 1$, $s_2 = [\sqrt{n}] + 2$, ..., $s_i = [\sqrt{n}] + i$, ... tražimo najmanje $s_i$ tako da je $s_i^2 - n$ potpun kvadrat, tj. $t_i = \sqrt{s_i^2 - n}$ ceo broj. Tada je $p = s_i + t_i$ i $q = s_i - t_i$.

## 22. Kriptoanaliza RSA Fermaovim metodom

U RSA važi: odrediti tajni ključ $d \Leftrightarrow$ naći inverz za množenje $\cdot_{\phi(n)} \Leftrightarrow$ naći $\phi(n) \Leftrightarrow$ rastaviti $n$ $\Leftrightarrow$ naći pravi delilac od $n$. Osnovna pretpostavka RSA je da ne postoji efikasan način da se reši jedan od ovih problema, čime bi se rešio i svaki od njih. Elementarno rešeto je presporo jer je vreme kriptovanja $O(\log^3 n)$. Cica bi Fermaovim metodom faktorizacije mogla da dobije $p$ i $q$ jer je $n$ deo javnog ključa. Sada veoma lako računa $\phi(n) = (p-1)(q-1)$ na osnovu čega nalazi tajni ključ $d = e^{-1}\ (mod\ \phi(n))$ jer je i $e$ deo javnog ključa. Na kraju, čita poruku $M$ na isti način kao i Boban: $N^d \equiv M^{ed} \equiv M\ (mod\, n)$.

## 23. Polardov $(p-1)$-metod

**Polardov metod** omogućava da brzo faktorišemo prirodan broj $n$ pod pretpostavkama da $n$ ima prost činilac $p$ tako da je $p-1$ $B$-gladak i da je $B \ll n$ (npr. $B \sim \log n$). Pre primene Polardovog algoritma treba izračunati vrednost $m$ = $NZS(1,\ 2,\ \ldots,\ B)$. U kanonskoj faktorizaciji $m$ učestvuju samo prosti brojevi $p$ koji dele neki od brojeva $1,\ 2,\ \ldots,\ B$, pa je $p \leq B$. Ako je $\alpha$ najveći stepen tako da $p^\alpha | m$ onda $p^\alpha$ deli neki od brojeva $1,\ 2,\ \ldots,\ B$ pa je $p^\alpha \leq B$, odnosno $\alpha \leq \log_p B$. Dakle:

$$m = \prod_{p\ prost,\ p \leq B} p^{[\log_p B]}$$

Svi $B$-glatki brojevi dele $m$. Na osnovu $p-1 | m$ i MFT ($a^{p-1} \equiv 1\ (mod\, p)$) važi $a^m \equiv 1\ (mod\, p)$. Metod:

1. Biramo $2 \leq a \leq n-1$ tako da $NZD(a,\ n) = 1$
2. Računamo $x = a^m - 1\ (mod\, n)$. Ako je $x = 0$ vraćamo se na prethodni korak i biramo novo $a$. Ako je $x = 1$ pretpostavka o $B$-glatkosti nije ispunjena i eventualno se može pokušati sa većim $B$. Ako je $x \neq \{0,\ 1\}$ prelazimo na sledeći korak.
3. Računamo $g$ = $NZD(n,\ x)$. Ako je $g \neq n$ onda će $g$ biti traženi pravi delilac od $n$. Vrednost $g$ ne zavisi od $p$, već samo od $n$ i $a$ koje biramo proizvoljno. Ako je $g = n$ probati sa drugim $a$, tj. vratiti se na prvi korak.

Ako je pretpostavka o $B$-glatkosti ispunjena, algoritam će dati $g \in \{2,\ 3,\ \ldots,\ n-1\}$.

## 24. Zašto se javni ključ $n = pq$ u RSA ne može izabrati tako da ne bude osetljiv na napad Polardovim metodom

RSA je ranjiv ako je $n = pq$ tako da je jedan od brojeva $p - 1$ ili $q - 1$ $B$-gladak. Možemo da proverimo $B$-glatkost $p - 1$ za neko fiksirano $B$, a samim tim i za sve vrednosti manje od $B$, ali i dalje postoji opasnost od $(B + 1)$-glatkosti. Ne možemo tražiti najmanje $B$ koje ispunjava prethodni uslov jer bismo se opet vratili na faktorizaciju.

## 25. Intergritet poruke i heš algoritam

**Integritet poruke** podrazumeva da Boban treba da bude siguran da Alisina poruka nije usput promenjena, slučajno ili namerno. Promenjena poruka može da bude nečitljiva, ali čak i ako je čitljiva Boban neće primetiti da je promenjena. Iz tog razloga se u kriptosistem obično uključuje i **heš algoritam** koji Alisa primenjuje na poruku pre kriptovanja. Boban primenjuje heš algoritam na dešifrovanu poruku i upoređuje dobijenu vrednost sa Alisinim rezultatom. Cica može da promeni šifrat, ali ne zna kako da promeni Alisinu vrednost heš algoritma. Heš algoritam se sastoji od više uzastopnih primena **heš funkcije**. Heš funkcija $h$ je jednosmerna i otporna na koliziju. Slaba otpornost podrazumeva da je za dato $x$ teško odrediti $y \neq x$ tako da $h(x) = h(y)$. Jaka otpornost podrazumeva da je teško odrediti različite $x$ i $y$ tako da $h(x) = h(y)$. Ulaz heš funkcije $h(x,\ y)$ su argumenti fiksirane dužine $k$ i $m$, a izlaz je dužine $m$. Ulaz heš algoritma je poruka promenljive dužine, a izlaz ima fiksiranu dužinu koja je obično mnogo manja od dužine ulaza. Heš algoritmi se najčešće dobijaju **MD (Merkle-Damgor) konstrukcijom**:

- Poruka se deli na blokove $M_1,\ \ldots,\ M_n$ dužine $m$.
- Izabere se neka inicijalna vrednost $K_0$ (npr. 0...0) i računa se $K_1 = h(K_0,\ M_1)$. MAC (Message Authentication Code) je heš algoritam koji umesto podrazumevane inicijalne vrednosti koristi tajni ključ.
- Redom se računaju vrednosti $K_i = h(K_{i-1},\ M_i)$, a $K_n$ će biti izlaz algoritma.

## 26. Autentikacija, digitalni potpis i sertifikat

**Autentikacija** je proces kojim se dokazuje da poruka dolazi od pravog pošiljaoca. Obuhvata:

- **digitalni potpis** koji povezuje poruku sa javnim ključem.
- **sertifikat** koji povezuje javni ključ sa konkretnom osobom. U pitanju je dokument koji se izdaje od ovlašćenog lica u kome piše "Osoba X koristi ključ Y".

## 27. Digitalni potpis pomoću RSA kriptosistema

Alisa treba da pošalje poruku Bobanu, a Boban treba da bude siguran u njen identitet. Koraci:

- Alisa generiše javni ključ $(n_A,\ e_A)$ i tajni ključ $d_A$ kao u RSA.
- Boban generiše javni ključ $(n_B,\ e_B)$ i tajni ključ $d_B$ kao u RSA.

- Alisa želi da pošalje kodiranu poruku $M < n_A,\ n_B$.
- Alisa računa $M_1 = M^{d_A}\ (mod\, n_A)$ i $M_2 = M_1^{e_B}\ (mod\, n_B)$ i šalje Bobanu $M_2$.
- Boban računa $M_3 = M_2^{d_B}\ (mod\, n_B)$ i $M_4 = M_3^{e_A}\ (mod\, n_A)$.

Važi $M_3 \equiv_{n_B} M_2^{d_B} \equiv_{n_B} M_1^{e_B d_B} \equiv_{n_B} M_1$, odnosno $M_3 = M_1$ pa važi $M_4 \equiv_{n_A} M_3^{e_A} \equiv_{n_A} M_1^{e_A} \equiv_{n_A} M^{e_A d_A} \equiv_{n_A} M$, odnosno $M_4 = M$.

## 28. Čemu služi heširanje prilikom digitalnog potpisa

Potpisivanje cele poruke $M$ je dobro rešenje, jer Cica ne može da izdvoji Alisin potpis, međutim ovo rešenje je sporo. U praksi se obično ključ i digitalni potpis razmenjuju asimetričnim kriptosistemom, a poruka simetričnim, pa je ovaj način neizvodljiv. Zbog toga se najčešće potpisuje $H(M)$, gde je $H$ heš algoritam. Sada poruka može ići simetričnim, a potpis asimetričnim kriptosistemom, ali je potpis vezan za heširanu poruku.

## 29. Kakvo poboljšanje donose eliptičke krive u sigurnosti kriptosistema, a kakve u kriptoanalizi

Mnogi kriptosistemi sa javnim ključem, kao što su Difi-Helman, Mesi-Omura i ElGamal, imaju unapređenu verziju koja se zasniva na **eliptičkim krivama**. One omogućavaju da se postigne ista zaštita sa manjim ključem koji koristi EK. Na primer, 256-bitni ključ zasnovan na EK menja 3072-bitni ključ. EK se koriste i u kriptoanalizi, posebno za napad na RSA. Čak i ako neki kriptosistem ne koristi EK, on može biti napadnut algoritmom koji koristi EK. Na primer, Polardov $(p-1)$-metod faktorizacije je spor ako $n$ nema prost činilac $p$ tako da je $p-1$ $B$-gladak. Sa EK će biti dovoljno da za malo $s$ neki od $p+s$ bude $B$-gladak.

## 30. Definisati i nacrtati eliptičku krivu nad poljem realnih brojeva

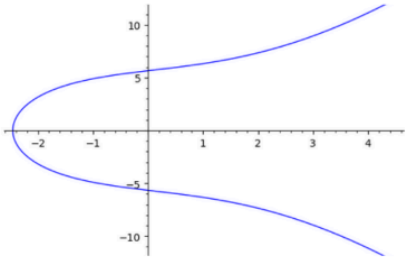
**Eliptička kriva nad** $R$ je kriva definisana jednačinom

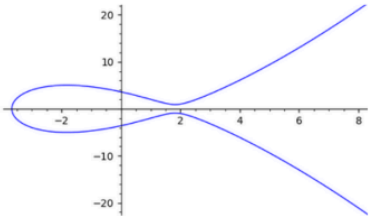
$$E: y^2 = x^3 + ax + b,\ \ a,\ b \in R,\ \ \Delta = -16(4a^3 + 27b^2) \neq 0$$

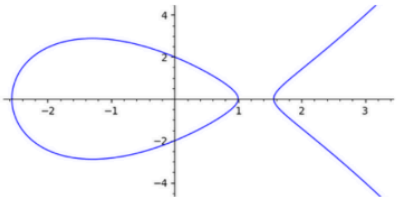
Na skup rešenja dodaje se i jedna "beskonačno daleka" tačka $\mathcal{O}$, pa je puna definicija:

$$E(R) = \{(x,\ y) \in R \times R \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

Primeri eliptičkih krivih:



## 31. Definisati eliptičku krivu nad konačnim poljem

**Eliptička kriva nad** $F_q$, pri čemu je $q$ stepen prostog broja $p \neq 2,\ 3$, definisana je jednačinom

$$E(F_q) = \{(x,\ y) \in F_q \times F_q \mid y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\},\ \ a,\ b \in F_q,\ \ \Delta = -16(4a^3 + 27b^2) \neq 0$$

Uobičajeno je da se piše $E(F_q)$, ali je ispravnije $E(F_q;\ a,\ b)$ jer zavisi od sva tri parametra. Sada je operacije teže videti geometrijski, ali se $\oplus$ i $\ominus$ mogu računati algebarski.

## 32. Definisati operacije na eliptičkoj krivoj. Grupni zakon na eliptičkoj krivoj

Za $P = (x,\ y) \in E(R)$ definišemo **inverz** $\ominus P = (x,\ -y)$ i $\ominus\mathcal{O} = \mathcal{O}$

Za $P,\ Q \in E(R)$ definišemo **sabiranje** $P \oplus Q$:

1. ako je $P = \mathcal{O}$: $\mathcal{O} \oplus Q = Q$
2. ako je $Q = \mathcal{O}$: $P \oplus \mathcal{O} = P$
3. ako je $Q = \ominus P \neq \mathcal{O}$: $P \oplus Q = \mathcal{O}$
4. ako je $P,\ Q \neq \mathcal{O},\ Q \neq P,\ \ominus P$: povučemo pravu $l$ kroz $P$ i $Q$ i tada je $P \oplus Q = \ominus R$, pri čemu važi jedno od sledeća dva:
   - $l$ seče EK u još tačno jednoj tački $R \neq P,\ Q$
   - $l$ je tangentna na EK u jednoj od tačaka $P$ i $Q$ pri čemu je $R$ upravo ta tačka
5. ako je $P = Q \neq \mathcal{O},\ \ominus P$: povučemo tangentu $l$ na EK u tački $P$. Ona će preseći EK u još tačno jednoj tački $R \neq P$. Tada je $P \oplus Q = P \oplus P = 2P = \ominus R$

**Grupni zakon na eliptičkoj krivoj**: $(E(R),\ \oplus,\ \ominus,\ \mathcal{O})$ je Abelova grupa.

## 33. Haseova teorema za broj tačaka na eliptičkoj krivoj. Zašto rad sa grupom $(E(F_q),\ \oplus)$ nudi više mogućnosti od grupe $(F_q \backslash \{0\},\ \cdot)$

**Haseova teorema**: Kardinalnost grupe $E(F_q)$ je $q + 1 + s$, gde je $s \leq 2\sqrt{q}$. Dodatno, za svaku celobrojnu vrednost $s \in [-2\sqrt{q},\ 2\sqrt{q}]$ postoji EK $E(F_q)$ tako da je $|E(F_q)| = q + 1 + s$.

Ideja je da se umesto grupe $(F_q \backslash \{0\},\ \cdot)$ koristi grupa $(E(F_q),\ \oplus)$ jer umesto fiksirane kardinalnosti $q - 1$ imamo slobodu $[q + 1 - 2\sqrt{q},\ q + 1 + 2\sqrt{q}]$. Takođe, stepenovanje $g^n$ se menja sa $nP = P \oplus \ldots \oplus P$. Vrednost $nP$ računamo ponovljenim dupliranjem tačke pomoću $O(\log n)$ operacija $\oplus$, a svaka operacija $\oplus$ se realizuje sa nekoliko sabiranja, oduzimanja i množenja.

## 34. Problem diskretnog logaritma nad eliptičkim krivama

**Problem diskretnog logaritma nad EK**: Ako je poznato $P$ i $nP$ odrediti $n$. U praksi, ovaj problem se rešava još sporije od diskretnog logaritma u $F_q^*$.

## 35. Kodiranje i dekodiranje podataka pomoću eliptičke krive

Ako je $q = p$ prost broj, ovaj metod će raditi uspešno sa verovatnoćom $1 - \frac{1}{2^k}$, pri čemu $k$ sami biramo. U praksi obično $k \in [30,\ 50]$. Poruka koja treba da se kodira se po potrebi deli na blokove $m$ koji se prevode u numerički ekvivalent $M$. Maksimalna veličina bloka $N$ je takva da važi $Nk < q$. $M$ treba kodirati tačkom $(x_0,\ y_0)$ sa krive $E : y^2 = x^3 + ax + b$ nad $Z_p$. Pokušamo sa $x_0 = Mk$, pa ako $y^2 = x_0^3 + ax_0 + b$ ima rešenja, izaberemo jedno takvo $y_0$. Ukoliko nema rešenja, pokušavamo dalje sa $Mk + 1$, $Mk + 2$, ..., $Mk + k - 1$ sve dok ne pronađemo $(x_0,\ y_0)$. Kvadratna kongruencija ima rešenje u $\frac{1}{2}$ slučajeva, pa je verovatnoća da ćemo u $k$ pokušaja bar jednom biti uspešni $1 - \frac{1}{2^k}$. Kodirana poruka je tačka $(x_0,\ y_0)$, mada se u nekim implementacijama koristi samo $x_0$. U opštem slučaju kada $q = p^\alpha$ i $F_q \cong \{a_0 + \ldots + a_{\alpha-1}t^{\alpha-1} \mid 0 \leq a_0,\ \ldots,\ a_{\alpha-1} \leq p - 1\}$ postupak je isti ali se $Mk + j$ zapiše u osnovi $p$ kao $Mk + j$ = $a_0 + \ldots + a_r p^r$ ($r \leq \alpha - 1$ jer $M < q$) i pokuša se da se za polinom $x_0 = x_0(t)$ = $a_0 + \ldots + a_{r-1}t^{r-1}$ nađe polinom $y_0 = y_0(t)$ tako da $(x_0,\ y_0)$ pripada EK.

Dekodiranje: Od tačke $(x_0,\ y_0)$ treba rekonstruisati poruku $m$. U slučaju $q = p$ važi $M$ = $[\frac{x_0}{k}]$ jer $[\frac{x_0}{k}]$ = $[M + \frac{j}{k}]$ pri čemu ne znamo šta je $j$, ali znamo da je iz $[0,\ k - 1]$. U opštem slučaju $q = p^\alpha$, polinom $x_0(t)$ = $a_0 + \ldots + a_{r-1}t^{r-1}$ prevodimo u broj $a_0 + \ldots + a_{r-1}p^{r-1}$ i dalje radimo kao u prethodnom slučaju.

## 36. Difi-Helmanovo usaglašavanje ključa nad eliptičkim krivama

Javni ključ čine konačno polje $F_q$, eliptička kriva $E : y^2 = x^3 + ax + b$ nad $F_q$ i tačka $P$ = $(x_0,\ y_0) \in E(F_q)$, odnosno parametri $(q,\ a,\ b,\ x_0)$. Poželjno je da $P$ bude generator grupe $(E(F_q),\ \oplus)$. Alisa i Boban biraju svoje tajne ključeve $a_A,\ a_B < |E(F_q)|$, a zatim računaju javne ključeve $a_AP,\ a_BP \in E(F_q)$ i razmenjuju ih. Usaglašeni ključ će biti $K$ = $(a_Aa_B)P \in E(F_q)$. Alisa može da izračuna $K$ = $a_A(a_BP)$, a Boban $K$ = $a_B(a_AP)$. Cica vidi samo $a_AP$ i $a_BP$, ali ne i $K$.

## 37. ElGamalov kriptosistem nad eliptičkim krivama

Javni ključ čine konačno polje $F_q$, eliptička kriva $E : y^2 = x^3 + ax + b$ nad $F_q$ i tačka $P$ = $(x_0,\ y_0) \in E(F_q)$, odnosno parametri $(q,\ a,\ b,\ x_0)$. Poželjno je da $P$ bude generator grupe $(E(F_q),\ \oplus)$. Boban bira svoj tajni ključ $e < |E(F_q)|$ i pomoću njega računa javni ključ $eP \in E(F_q)$. Za svaki kodirani blok poruke $M \in E(F_q)$ Alisa generiše slučajan broj $k < |E(F_q)|$ i šalje Bobanu tačke $kP$ i $M \oplus keP$, gde $keP$ dobija množeći tačku $eP$ sa $k$. Boban tačku $keP$ može dobiti tako što $kP$ pomnoži sa $e$. On sabira tačke $M \oplus kep$ i $\ominus keP$ i dolazi do $M$. Cica vidi samo $eP$, $kP$ i $M \oplus kep$ i mora da reši problem diskretnog logaritma nad EK da bi došla do poruke $M$.

### 38. Lenstrin metod faktorizacije

Ako hoćemo da faktorišemo broj $n$ za koji verujemo da je složen, možemo pretpostaviti suprotno - $n$ je prost i onda važi da je $Z_n$ polje. Izaberemo neku eliptičku krivu $E(Z_n)$ i neku tačku $P$ sa te krive. Krenemo da računamo $2P,\ 3P,\ 4P,\ \ldots$ ili $2P,\ 4P,\ 8P,\ \ldots$ i negde će se pojaviti problem sa deljenjem. Kada se u imeniocu pojavi broj $g$ koji nije invertibilan po modulu $n$, onda će $NZD(g,\ n) > 1$ biti pravi delilac $n$.

**Lenstrin metod**: Pretpostavka je da $n$ ima prost činilac $p$ tako da je kardinalnost $|E(Z_p)|$ $B$-gladak broj za neko malo $B$. Ideja je da ne treba pogađati koje $m$ radi, već pokušati sa $m$ = $NZS(1,\ \ldots,\ B)$ ili $m$ = $B$!. Znamo da $|E(Z_p)|$ deli ove $m$, pa će biti $mP$ = $\mathcal{O}$ na $E(Z_p)$, tj. pojaviće se imenilac $g$ koji je deljiv sa $p$. Nećemo računati $mod\ p$, već $mod\ n$. Dakle, računamo $mP$ na $E(Z_n)$. Može da se desi:

- izračunali smo $mP$ bez problema: pretpostavka nije ispunjena, pokušati sa većim $B$ ili drugom EK.
- kao imenilac pojavi se $g$ deljivo sa $n$: promeniti $P$.
- kao imenilac pojavi se $g$ koje nije deljivo sa $n$, ali nije ni invertibilno po modulu $n$: $NZD(g,\ n)$ je pravi delilac $n$.

# Zero Knowledge proofs

### 1. Zero Knowledge proofs i ilustrativni primeri

**Zero Knowledge proofs - ZKP (dokazi sa nula znanja)** je kriptografska tehnika pomoću koje **dokazivač** dokazuje **verifikatoru** da zna neku informaciju bez otkrivanja same informacije. U pitanju su probabilistički dokazi, tj. postoji minimalna verovatnoća da dokazivač ne zna informaciju, a da će dokaz biti prihvaćen od strane verifikatora. Dokazi sa nula znanja mogu biti **interaktivni** i **neinteraktivni**. Neinteraktivni ZKP mogu biti **SNARKs** ili **STARKs**. Primeri:

- **Gde je Valdo?** - dokazivanje poznavanja lokacije nečega ili nekoga unutar složenog skupa podataka bez otkrivanja stvarne pozicije.
- **Ali Babina pećina** - dokazivač treba da ubedi verifikatora da zna tajnu reč, bez otkrivanja iste. Dokazivač ulazi u pećinu i bira jedan put, a verifikator čeka napolju i nasumično traži od dokazivača da izađe kroz jedan od puteva. Dokazivač može otvoriti vrata koristeći tajnu reč, čime dokazuje poznavanje reči bez njenog otkrivanja.

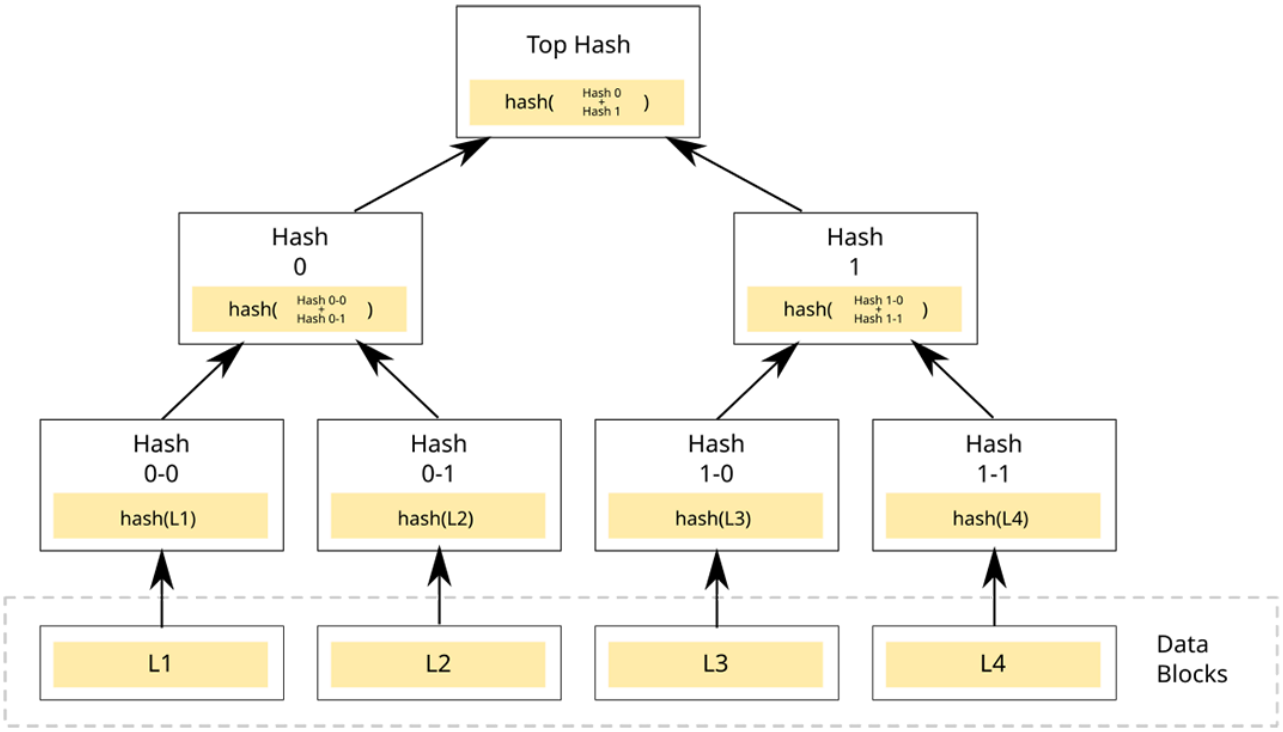
- **Prijatelj daltonista** - kako dokazati daltonisti da su sve lopte različitih boja bez otkrivanja samih boja? Jedan pristup je korišćenje serija zamena između lopti koje kontroliše daltonista, gde on može testirati da li dokazivač i dalje može identifikovati lopte kao različite.

## 2. Primene ZKP-a

- **Blockchain** - povećanje privatnosti transakcija.
- **Finansije** - bezbedna razmena podataka i verifikacija bez otkrivanja podataka. Na primer, aplikant za kredit može dokazati da mu je plata unutar određenog opsega bez otkrivanja tačnog iznosa. Slično, može se dokazati da je iznos plaćanja unutar neke granice, ali se ne prikazuje tačan iznos.
- **Online glasanje** - poboljšanje integriteta i privatnosti. Na primer, MACI (Minimum Anti-Collusion Infrastructure).
- **Decentralizovani identifikatori (DIDs)** - bolja verifikacija identiteta bez otkrivanja više informacija nego što je potrebno. Pruža pojedincu mogućnost da kontroliše pristup ličnim identifikatorima. Na primer, dokazivanje državljanstva bez otkrivanja poreskog ID-a ili detalja pasoša.
- **Mašinsko učenje** - verifikacija rezultata ML modela bez otkrivanja podataka ili samog modela. Omogućava korišćenje ML nad osetljivim podacima gde bi korisnik znao rezultate koje daje model nad njegovim podacima, a da pritom ne otkriva te podatke trećoj strani.
- **Autentifikacija** - dokazivanje da znamo neke informacije bez otkrivanja istih. Jednom kada je ZK dokaz generisan korišćenjem javnih i privatnih ulaza, korisnik ga jednostavno može prezentovati radi autentifikacije svog identiteta kada mu je potrebno da pristupi nekoj usluzi.

## 3. Merkle Tree i ZK dokaz pripadnosti skupu

**Merkle Tree** je binarno stablo kod koga listovi sadrže heš transakcija, a unutrašnji čvorovi sadrže heš kombinaciju dece. Omogućava efikasnu pretragu sadržaja velike strukture podataka. Ispitivanje pripadnosti list-čvora stablu odvija se u logaritamskom vremenu. Blok se sastoji od zaglavlja i tela. Zaglavlje, između ostalog, sadrži koren Merkle stabla, a telo sadrži sve potvrđene informacije o transakcijama u bloku. Dokaz pripadnosti skupu se odvija preko protokola dokazivača i verifikatora, gde dokazivač ne sme otkriti informacije o samom dokazu. Ovo osigurava da se članstvo u skupu može dokazati bez otkrivanja detalja o sadržaju skupa ili informacija koje nisu namenjene javnosti.




## 4. Completeness, Soundness i Zero Knowledge

ZK dokaz mora zadovoljiti tri koraka:

- **Completeness** - verifikator mora prihvatiti ispravan dokaz.
- **Soundness** - verifikator ne bi trebalo da prihvati netačan dokaz.
- **Zero Knowledge** - verifikator kroz javne parametre neće ništa naučiti o dokazu. Ovo osigurava da čak i nakon što je dokaz potvrđen kao ispravan, verifikator neće dobiti nikakve dodatne informacije o sadržaju ili detaljima samog dokaza, osim onih koji su već javno poznati.

## 5. Ciklična grupa $(Z_p^*, \cdot)$

**Ciklična grupa** $(Z_p^*, \cdot)$ je $Z_p^*$ = $\{1, \ldots, p-1\}$, gde je $p$ prost broj, sa operacijom $\cdot$ definisanom kao $a \cdot b$ = $a * b \; mod \; p$, koja pritom sadrži makar jedan generator. Neki element $a$ je **generator** ciklične grupe $Z_p^*$ ako se svi ostali elementi te grupe mogu dobiti preko njega, tako da je $a_i = a^i \; mod \; p$. Svaka grupa $Z_p^*$ gde je $p$ prost broj je ciklična. Svojstva grupe:

- **zatvorenost** - ako su elementi $a$ i $b$ iz grupe, onda je i $a \cdot b$ u grupi.
- **asocijativnosti** - ako su elementi $a$, $b$ i $c$ iz grupe, onda je i $a \cdot (b \cdot c)$ = $(a \cdot b) \cdot c$ u grupi.
- **neutral** - ako je element $a$ iz grupe, onda je i $a \cdot 1$ = $a$ u grupi.
- **inverz** - ako je element $a$ iz grupe, onda postoji element $b$ iz grupe tako da je $a \cdot b = 1$.

Generatore ciklične grupe možemo brzo naći koristeći sledeću teoremu: Element $g$ ciklične grupe $Z_p^*$ će biti generator te grupe akko $g^{\frac{p-1}{q}} \neq 1 \; mod \; p$, za svaki prost broj $q$ tako da $q \mid p-1$.

## 6. Problem diskretnog logaritma

Neka je $g$ generator ciklične grupe $Z_p^*$ = $\{g,\ \ldots,\ g^{p-1}\}$, gde je $p$ prost broj. Ako su $g$ i $p$ uzajamno prosti brojevi, na osnovu MFT važi $g^{p-1} = 1\ mod\ p$ i $Z_p^*$ = $\{1,\ g,\ \ldots,\ g^{p-2}\}$. Broj $x$ će biti **diskretni logaritam** od $b$ u bazi $g$ ako važi $\log_g b = x \Leftrightarrow g^x = b$ u grupi $Z_p^*$, gde je $b$ element te grupe.

## 7. Eliptičke krive nad konačnim poljem

**Eliptičke krive nad konačnim poljem** $F_q^*$ definisane su sa:

$$E(F_q^*) : y^2 = x^3 + ax + b,\ \ a,\ b \in F_q^*$$

Tačke eliptičke krive nad konačnim poljem su samo tačke na grafiku, odnosno nema krive u pravom smislu. Nad tačkama EK definisane su operacije $-$, $+$ i $nP$. Dokazano je da skup tačaka u ECC (Elliptic Curve Cryptography) uvek formira Abelovu grupu sa sledećim svojstvima:

- **zatvorenost** - ako tačke $P$ i $Q$ pripadaju EK, onda i $P + Q$ pripada EK.
- **asocijatinost** - ako tačke $P$, $Q$ i $R$ pripadaju EK, onda važi $(P + Q) + R$ = $P + (Q + R)$.
- **identitet** - postoji identični element $0$ takav da je $P + 0 = P$.
- **inverz** - svaka tačka $P$ koja pripada EK ima inverz $Q$ takav da je $P + Q = 0$.
- **komutativnost** - ako tačke $P$ i $Q$ pripadaju EK, onda važi $P + Q$ = $Q + P$.

Pretpostavljamo da su EK nad konačnim poljem ciklične.

## 8. Add and Double algoritam

Računanje $nP$ je jako sporo za veliko $n$ jer sabiramo $P$ sa samim sobom $n$ puta. Umesto toga, efikasniji je **Add nad Double algoritam**: $nP$ računamo tako što $n$ zapišemo binarno i time svodimo račun na logaritamski broj sabiranja. Na primer: $79P$ = $2^6P + 2^3P + 2^2P + 2^1P + 2^0P$.

## 9. Multi-Scalar-Multiplication (bucket metod)

Usko grlo u algoritmima za dokazivanje kod većine EK zasnovanih na SNARK sistemima je **Multi-Scalar-Multiplication algoritam**. Naivni algoritam koristi Add and Double algoritam, ali najbrži pristup predstavlja varijanta Pipengereovog algoritma koji nazivamo **bucket metod**:

- **pozicioniranje skalara** - svaki skalar se particioniše na $m$ delova, tako da svaki deo sadrži $w$ bitova.
- **akumulacija** - paralelno obrađivanje skalara i tačaka i akumulacija rezultata u bakete.
- **optimizacija** - tabele rezultata, paralelizacija, izbor EK.

## 10. Problem diskretnog logaritma nad eliptičkim krivama

**Problem diskretnog logaritma nad EK**: Ako je poznato $P$ i $nP$ odrediti $n$. U praksi, ovaj problem se rešava još sporije od diskretnog algoritma u $F_q^*$.

## 11. Uparivanje na eliptičkim krivama

Neka je data EK $E(F_q)$, neka su $G_1$ i $G_2$ podgrupe od $E(F_q)$ reda $p$, gde je $p$ prost broj, i neka su $g_1$ i $g_2$ generatori grupa $G_1$ i $G_2$, redom. Funkcija $e : G_1 \times G_2 \to G_T$, gde je $G_T$ multiplikativna podgrupa od $F_q$ reda $p$, je **uparivanje nad EK** ako važi:

1. $e(g_1,\ g_2) \neq 1$
2. $e(P + Q,\ R) = e(P,\ R) \cdot e(Q,\ R)$
3. $e(P,\ Q + R) = e(P,\ Q) \cdot e(P,\ R)$

Tipovi uparivanja:

- $G_1 = G_2$ (**simetrično uparivanje**)
- $G_1 \neq G_2$ i postoji efikasan homomorfizam $\phi : G_2 \to G_1$, ali ne postoji efikasan homomorfizam u suprotnom smeru.
- $G_1 \neq G_2$ i ne postoji efikasan homomorfizam između $G_1$ i $G_2$.

## 12. STARKs i SNARKs

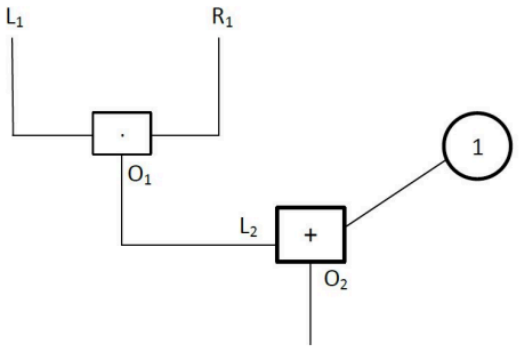
**ZN SNARKs** koriste sažeti dokaz i eliptičke krive, odnosno **povereno postavljanje (trusted setup)** što znači da se oslanjaju na početne parametre kojima se veruje. Veličina dokaza i vreme za verifikaciju zavise od aritmetičkog kola. Karakteriše ih brza verifikacija.

**ZN STARKs** ne koriste eliptičke krive i ne zahtevaju povereno postavljanje, već koriste heš funkcije. Zbog toga su transparentniji i manje podložni potencijalnim sigurnosnim rizicima vezanim za povereno postavljanje.

## 13. Aritmetizacija i sistem ograničenja (System Constraints) kod ZKP-a

**Aritmetizacija** je prevođenje problema u aritmetičko kolo, koje se prevodi u ograničenja, koja se prevode u polinome. Koristi operacije $+$ i $\cdot$ gde ulazni podaci i izlaz moraju biti iz konačnog polja. Verifikator proverava da li se izlaz aritmetičkog kola podudara sa javnom heširanom vrednošću dokazivača. Poenta je da zapišemo deo kola preko polinoma koji pokriva celo kolo tako što podešavamo koeficijente uz delove koje želimo prikazati na $1$, a ostale koeficijente na $0$.

**Example:** I know an $a$ such that $a*a+1=b$, for given $b$.




**System constraints:**

Gate constraints:

(1) $L_1 \cdot R_1 - O_1 = 0$

(2) $L_2 + 1 - O_2 = 0$

Copy constraints:

$L_1 = R_1$

$O_1 = L_2$

$$L_i \cdot q_{L_i} + R_i \cdot q_{R_i} + O_i \cdot q_{O_i} + q_{C_i} + L_i \cdot R_i \cdot q_{M_i} = 0.$$

Define $L(x), R(x), O(x)$:

$$L(1) = L_1,\ L(2) = L_2,\ R(1) = R_1,\ R(2) = R_2,\ O(1) = O_1,\ O(2) = O_2.$$

$$f(x) = L(x) \cdot q_L(x) + R(x) \cdot q_R(x) + O(x) \cdot q_O(x) + q_C(x) + L(x) \cdot R(x) \cdot q_M(x).$$

## 14. Komitmenti pomoću polinoma (Polynomial Commitments) kod SNARK-ova

**Komitovanje** predstavlja opredeljenje za neku vrednost pri čemu je ne otkrivamo ostalima, uz mogućnost da to uradimo kasnije. Komitovanje preko polinoma igra ključnu ulogu pri izgradnji efikasnih ZKP. Omogućava dokazivanje ispravnosti polinoma bez otkrivanja samog polinoma. Najčešći tip polinomskog komitovanja je **KZG**, a koriste se i Dory20, Dark20 i FRI.

## 15. Trusted setups kod Groth16 i PLONK-a

**Trusted setup (povereno postavljanje)** je procedura koja se obavlja jedanput radi generisanja podataka koji se koriste svaki put kada se neki kriptografski protokol pokrene. **Groth16** zahteva specifičan trust setup za svako aritmetičko kolo, gde se koristi nasumično odabrana pomoćna tačka sa EK kako bi se sprečilo lažiranje dokaza. **PLONK** nudi univerzalan i ažurirajući trusted setup, gde se koristi nasumično odabrana pomoćna tačka sa EK koja je nezavisna od kola. Međutim, PLONK ima veće veličine dokaza što utiče na troškove gasa u Eterijum mreži. **Transparentni setup** ne koristi tajne podatke, tj. pomoćne tačke sa EK.

## 16. Non-Interactive Preprocessing Argument sistem

**Non-Interactive Preprocessing Argument sistem** podrazumeva da se pre kreiranja dokaza generišu informacije koje dokazivač koristi za konstrukciju dokaza. Nakon toga, dokazivač šalje dokaz verifikatoru koji vrši verifikaciju. Za razliku od interaktivnih ZKP, ovde se sve završava u jednom koraku i ne postoji dodatna komunikacija između dokazivača i verifikatora.

## 17. KZG

**KZG** je kriptografska šema koja se koristi pri komitovanju preko polinoma. Omogućava dokazivanje ispravnosti polinoma bez otkrivanja samog polinoma. Generiše se komit za polinom i šalje verifikatoru koji proverava da li je vrednost polinoma u određenoj tački zaista $0$. Faze KZG:

- **setup**: biranje nasumične tačke i parametara $H_0 = G$, $H_1 = Gs$, $H_s = Gs^2$, ...

- **commit**: $com(f) = f(s) \cdot G$, gde $s$ pripada konačnom polju $F_p$, a $G$ je generator.
- **evaluate**: $f(x_0) = y$, gde je $x_0$ nula $f(x) - y$ i važi $x - x_0 \mid f(x) - y$ i $f(s) - f(z) = (s - z)h(s)$.

## 18. PLONK

**PLONK** je ZKP sistem koji pripada SNARK grupi. U pitanju je univerzalni sistem, što znači da je potrebno inicirati trusted setup samo jednom i on će važiti za svako aritmetičko kolo. Trusted setup se sastoji od inicijalnih parametara koji se koriste tokom verifikacije. PLONK se oslanja na komitovanje preko polinoma bez njegovog otkrivanja. PLONK koristi permutacije bazirane na Lagranžovim osnovama za definisanje ograničenja u kolu. Ograničenja se sastoje od ulaznih podataka svake kapije i od vektora koeficijenata za svaku kapiju. PLONK se može koristiti u Non-Interactive Preprocessing Argument sistemima.

## 19. Protokol Semafor

**Semafor** je ZKP protokol koji omogućava slanje signala u svojstvu člana grupe, bez otkrivanja sopstvenog identiteta. Takođe, onemogućava ponovno slanje signala. Ima primenu u tajnom glasanju, sistemima zaštite uzbunjivača, anonimnim decentralizovanim organizacijama i slično. Dakle, korisnik može da kreira identitet, doda ga u neku grupu i pošalje proverljiv, anoniman signal. ZKP mogu da osiguraju da je korisnik član date grupe i da već nije poslao signal. **Kolo semafora** predstavlja jezgro protokola. Njegovi delovi su:

- **dokaz pripadnosti** - svaki korisnik ima **identitetsku obavezu** koja se dodaje u Merkle drvo. Ona se formira koristeći identitetski poništavač i tajni ključ tog korisnika. Koren Merkle drveta je javan i svaki član može da dokaže pripadnost grupi, bez otkrivanja ko je.
- **anulirajući heš** - sprečava ponovno slanje signala.
- **signal**