ZERO KNOWLEDGE PROOFS

1. ZKP i ilustrativni primeri

ZKP je kriptografska tehnika pomoću koje dokazivač dokazuje verifikatoru da zna neku informaciju bez otkrivanja same informacije.
Nisu dokazi u klasičnom matematičkom smislu, već su probabilistički dokazi tj. postoji minimalna verovatnoća da dokazivač ne zna informaciju, a da će dokaz biti prihvaćen od strane verifikatora.
Postoje dva glavna tipa dokaza sa nula znanja: interaktivni i neinteraktivni

Primeri:

- *Gde je Valdo?* – dokazivanje poynavanja lokacije nečega/nekoga unutar složenog skupa podataka, bez otkrivanja stvarne pozicije
- *Ali Babina pećina* – izazov je da dokazivač ubedi verifikatora da zna tajnz reč bez otkrivanja iste. Dokazivač ulazi u pećinu i bira jedan put, dok verifikator čeka napolju. Zatim verifikator nasumično traži od dokazivača da izađe kroz jedan od puteva. Dokazivač može otvoriti vrata koristeći tajnu reč, ako je to potrebno da bi ispunio zahtev verifikatora, čime dokazuje poznavanje reči bez njihovog otkrivanja.
- *Prijatelj daltonista* – kako dokazati daltonisti da sud ve lopte različitih boja bez otkrivanja samih boja. Jedan pristup je korišćenje serija zamene izmedju lopti, koje kontroliše daltonista, gde on može testirati da li dokazivač i dalje može identifikovati lopte kao različite.

2. Primene ZKP-a

Blockchain – za povećanje privatnosti transakcija
Finansije – za bezbednu razmenu podataka i verifikaciju bez otkrivanja podataka, npr. aplikant za kredit može dokazati da mu je plata unutar određenog opsega, bez otkrivanja tačnog iznosa. Slično tome, može dokazati da je iznos plaćanja unutar granice, ali ne prikazuje tačan iznos
Online glasanje – poboljšanje integriteta i privatnosti u sistemima za online glasanje; primer MACI (Minimum Anti-Collusion Infrastructure)
DIDs – za što bolju verifikaciju identiteta bez otkrivanja više informacija nego što je potrebno. Decentralizovana identifikacija daje pojedincu mogućnost da kontroliše pristup ličnim identifikatorima(npr. dokazivanje svog državljanstva bez otkrivanja poreskog ID-a ili detalja pasoša)
Mašinsko učenje – verifikovanje rezultata ML modela bez otkrivanja podataka ili modela;
primena mašinskog učenja na neke osetljive podatke gde bi korisnik mogao znati rezultat inferencije modela na svojim podacima, a da pritom ne otkriva svoj ulaz trećoj strani (npr. u medicinskoj industriji).
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 pristupiti 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. Ovo stablo omogućava efikasnu pretragu sadržaja velike strukture podataka. Ispitivanje pripadnosti list-čvora stablu se odvija 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 koji bi mogli otkriti sam sadržaj skupa ili informacije koje nisu namenjene javnosti.

4. Completeness, soundness i ZK

ZK dokaz mora zadovoljiti sva 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 (Zp*,·)

$Z_p^* = \{1, \dots, p-1\}$ gde je p prost broj sa operacijom · definisanom kao $a \cdot b = a * b \bmod p$
Svojstva grupe:
Zatvorenost - ako su elementi a i b iz grupe, tada je i $a \cdot b$ takođe u grupi;
Asocijativnost - ako su elementi a, b, c iz grupe, tada je $a \cdot (b \cdot c) = (a \cdot b) \cdot c$ takođe u grupi;
Neutral - ako je element a iz grupe, tada je $a \cdot 1 = a$ takođe u grupi;
Inverz - ako je element a iz grupe, tada postoji element b iz grupe tako da je $a \cdot b = 1$;

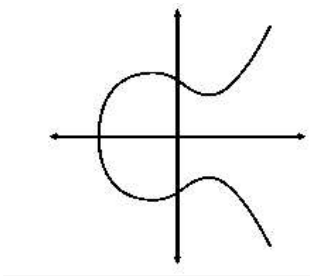
Neki element *a* je generator ciklične grupe Z_p^* ukoliko se svi ostali elementi te grupe mogu dobiti preko njega, tako da je $a_i = a^i \ mod \ p$
Ciklična grupa mora sadržati makar jedan generator.
$Z_p^* = \{1, \dots, p-1\}$ gde je p prost broj će uvek biti ciklična grupa.

Teorema pomoću koje brzo nalazimo generatore ciklične grupe:
Neka je g element ciklične grupe Z_p^*. On će biti generator te grupe akko $g^{((p-1)/q)} \neq 1 \bmod p$, za **svaki** prost broj q tako da $q \mid (p-1)$

6. Problem diskretnog logaritma

Neka je g generator ciklične grupe $Z_p^* = \{g, \dots, g^{p-1}\}$ gde je p prost broj.
Ako su g i p uzajamno prosti brojevi, tada na osnovu MFT važi $g^{p-1} = 1 \ mod \ p$ i $Z_p^* = \{1, g, \dots, g^{p-2}\}$
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 grupe Z_p^*, a g njen generator.

7. Eliptičke krive nad konačnim poljem



EK su grupe definisane nad konačnim poljem. EK:
$y^2 = x^3 + ax + b$, a, b iz F_q^*
EK su Abelove grupe $(E(F_q^*), +)$, pretpostavljamo da su EK nad konačnim poljem ciklične.
Operacije -, +, nP nad tačkama EK.
Tačke EK nad konačnim poljem su bukvalno samo tačke na grafiku (lin. reg.), nema krive.

Dokazano je da skup tačaka u ECC-u ("Elliptic Curve Cryptography") uvek formira Abelovu grupu sa sledećim svojstvima:
Zatvorenost: Ako tačke P i Q pripadaju E(K), onda i P + Q pripada E(K);

Asocijativnost: (P + Q) + R = P + (Q + R); Identitet: Postoji identični element 0 takav da je P + 0 = P;
Inverz: Svaki element P ima inverz Q takav da je P + Q = 0;
Komutativnost: P + Q = Q + P;
Pretpostavićemo da je eliptična kriva nad konačnim poljem ciklična.

8. Add and Double algoritam

Računanje nP je jako sporo za veliko n, jer sabiramo P sa samim sobom n puta
nP računamo tako što n zapišemo binarno i time svodimo na logaritamski broj sabiranja.
Npr. $79*P = 2^6*P + 2^3* P + 2^2*P + 2^1*P + 2^0*P$

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 je varijanta Pippengerovog algoritma koji nazivamo bucket metod.
Koraci algoritma:
- *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 eliptičke krive

10. Problem diskretnog logaritma nad eliptičkim krivama

- poznato je *P* i *mP*, a treba pronaći *m*
- analogno kod ciklične grupe Z_n* za g^n
- glavna operacija nad eliptičkim krivama
- jednosmerna funkcija
- 29. pitanje u kriptografiji

11. Uparivanje na eliptičkim krivama

Neka je dato $E(F_q)$ i G1 i G2 su podgrupe reda p, gde je p prost broj od EK.
Neka je g1 generator od G1, a g2 od G2.
Funkcija e: G1 x G2 → G_T, gde je G_T multiplikativna podgrupa od pola F_q reda p je uparivanje nad EK ako je zadovoljeno:
1) e (g1, g2) ≠ 1
2) e (P+Q, R) = e (P,R) * e (Q, R)
3) e (P, Q+R) = e (P,Q) * e (P, R)

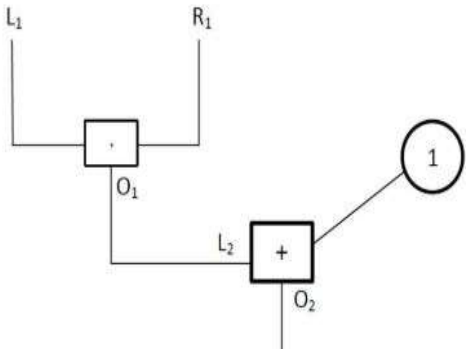
Tipovi bilinearnog uparivanja:
1. Ako je G1 = G2 i *e* je simetrično bilinearno uparivanje.
2. Ako G1 ≠ G2 i postoji efikasan homomorfizam ϕ: G2 → G1, ali ne i u suprotnom smeru
3. Ako G1 ≠ G2 i ne postoji efikasan homomorfizam između G2 i G1

12. STARKs & SNARKs

ZK SNARK – sažeti dokaz, brza verifikacija, koriste eliptičke krive (trusted setup - "povereno postavljanje" (trusted setup), što znači da se oslanja na početne parametre koji se veruju), veličina dokaza i vreme za verifikaciju zavise od aritmetičkog kola
ZK STARK – ne koriste eliptičke krive i ne zahtevaju povereno postavljanje, koristeći umesto toga heš funkcije, ovo ih čini transparentnijim i manje podložnim potencijalnim sigurnosnim rizicima vezanim za povereno postavljanje.

13. Aritmetizacija i sistem ograničenja (system constraints) kod ZKP-a

Aritmetizacija predstavlja prevođenje problema u aritmetičko kolo, koje prevodimo u ograničenja, koje prevodimo u polinome.
Aritmetizacija koristi operacije + i * gde ulazni podaci i izlaz moraju biti iz konačnog polja.
Verifikator proverava da li se izlaz aritmetičkog kola podudara sa javnom heširanom vrednošću dokazivača.



Gate constraints:
(1) $L_1 * R_1 - O_1 = 0$
(2) $L_2 + 1 - O_2 = 0$

Copy constraints:
(1) $L_1 = R_1$
(2) $O_1 = L_2$

Konstrukcija polinoma:
$L(1) = L_1$, $L(2) = L_2$, $R(1) = R_1$, $R(2) = R_2$, $O(1) = O_1$, $O(2) = O_2$
PLONK: $Lq_l + Rq_r + Oq_o + q_c + LRq_M = 0$

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.

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

Komitovanje predstavlja opredeljenje za neku vrednost tako da je ne smemo otkriti.
Komitovanje preko polinoma igra ključnu ulogu pri izgradnji efikasnih ZNP.
Omogućava dokazivanje ispravnosti polinoma, bez otkrivanja polinoma.
Najčešći tip polinomskog komitovanja je KZG, a koriste se još i Dory20, Dark20 i FRI.

15. Trusted setups kod Groth16 i PLONK-a

Trusted setup je procedura koja se obavlja jednaput radi generisanja podataka koji se koriste svaki put kada se neki kriptografski protokol pokrene.
Groth16 zahteva specifičan trusted setup za svako aritmetičko kolo, gde se koristi nasumično odabrana pomoćna tačka sa EK kako bi se sprečilo lažiranje dokaza.
PLONK nudi univerzalni 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 (pomoćne tačke sa EK).

16. Non-Interactive Preprocessing argument system

Najpre se obavlja faza preprocesiranja u kojoj se generišu informacije koje dokazivač koristi da konstruiše dokaz koji će poslati verifikatoru.
Verifikator koristi samo taj dokaz i preprocesirane informacije da proveri ispravnost tvrdnje.

17. KZG

KZG je kriptografska šema koja se koristi pri komitovanju preko polinoma.
Omogućava dokazivanje ispravnosti polinoma, bez otkrivanja polinoma.

Generiše se komit za polinom i šalje se verifikatoru koji proverava da li je vrednost polinoma u određenoj tački zaista 0.
Faze KZG:

- setup – odaberemo nasumičnu tačku i parametre $H_0 = G$, $H_1 = Gs$, $H_2 = Gs^2$ …
- commit - com(f) = f(s) * 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 $x - x_0$ | $f(x) - y$
  $f(s) - f(z) = (s - z) * h(s)$

18. PLONK

PLONK je ZNP sistem koji pripada SNARK grupi sistema.
To je univerzalni sistem, što znači da je potrebno inicirati trusted setup samo jednom koji će da važi za svako aritmetičko kolo. Trusted setup se sastoji od inicijalnih parametara koji se koriste tokom verifikacije.
PLONK se oslanja na komitovanje preko polinoma, gde dokazujemo ispravnost polinoma bez njegovog otkrivanja. Komitovanje preko polinoma generiše komit za neki polinom.
PLONK koristi permutacije baziranje na Langranžovim 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 Systems.

19. Protokol Semafor

Semafor je ZKP protokol koji omogućava korisnicima da šalju poruke kao dokazani članovi grupe, bez otkrivanja identiteta. Takođe, onemogućuje ponovljeno slanje poruka.
Koristi se u privatnom glasanju, otkrivanju nepravilnosti, anonimni DAO i mikseri.
Semafori omogućavaju korisnicima da kreiraju objekat semafora i da ga dodaju u grupu kako bi poslali proverljiv anonimni signal.
Kolo semafora predstavlja jezgro ovog protokola i sastoji se od:

1. dokaza pripadnosti
2. anulirajućeg heša
3. signala

Kolo hešira anulirajući heš kako bi generisao identitet komita preko koga proverava dokaz pripadnosti u korenu Merkle drveta.