Blockchain i njegove primjene

Pešić, Matea

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Matea Pešić

BLOCKCHAIN I NJEGOVE PRIMJENE

Diplomski rad

Voditelj rada:
prof. dr. sc. Filip Najman

Zagreb, studeni, 2022.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Svojim najbližima

Sadržaj

Uvod

U današnje vrijeme, većina ljudi za svoje transakcije koriste pouzdanog posrednika, poput banke. Banka pritom predstavlja središnju točku transakcije koja izvršava sve provjere integriteta vezane uz nju.
2009. godine autor koji se predstavlja pod pseudonimom Satoshi Nakamoto u knjizi *„Bitcoin: A Peer-to-Peer Electronic Cash System"* predstavlja novi, decentralizirani sustav u kojem se uklanja središnja točka. Prvi je puta predloženo korištenje ulančanih blokova kao polazna točka za sustav digitalnog plaćanja.

Blockchain tehnologija je temelj tog sustava te omogućava distribuciju između svih čvorova koji sudjeluju u sustavu. Ta tehnologija pruža maksimalnu zaštitu integriteta zapisa korištenjem kriptografskih metoda te svaki čvor sustava sadrži kopiju svih relevantnih informacija čime se uklanja potreba za posrednikom. Umjesto oslanjanja na središnje tijelo za siguran rad s drugim korisnicima, blockchain koristi inovativne konsenzusne protokole preko mreže čvorova, za provjeru transakcija i bilježenje podataka na način koji se ne može korumpirati. Sastavljen je od blokova međusobno povezanih u lanac gdje svaki blok sadrži niz zapisa. Blokovi se povezuju algoritmom koji koristi kriptografsku hash funkciju.

Blockchain tehnologija nam omogućava da podaci postanu korisniji ne samo jednoj organizaciji, nego svim partnerima u mreži. S druge strane, podacima kao i ostalim procesima između partnera se upravlja na kontroliran način uz visoku razinu sigurnosti i povjerenja. Iako su algoritmi koji se nalaze u tehničkim detaljima vrlo zamršeni i kompliciran, cijeli koncept je jednostavan. Ako se neka transakcija obavi i spremi u decentraliziranu mrežu, blockchain će omogućiti da se izmijenjeni podaci odmah vide te se može utvrditi je li njima manipulirano.

U ovom radu ćemo se osvrnuti na sve prednosti i mane blockchain tehnologije te ćemo pomnije promotriti njen nastanak i strukturu. Opisujemo povezanost čvorova u mreži, mrežne konsenzuse potrebne za uspješno funkcioniranje tehnologije, moguće prijetnje koje se javljaju te konstrukciju blokova uključujući razne kriptografske alate potrebne za uspješno spajanje blokova, od kojih je možda najvažniji alat kriptografska hash funkcija i *Merkle stabla*. Također ćemo govoriti o raznim primjenama blockchain tehnologije, fokusirajući se na kriptovalute. Prva, i do danas najuspješnija primjena tehnologije ulančanih blokova, zasigurno je digitalna valuta Bitcoin.

Diplomski rad napravljen je u sklopu aktivnosti Projekta KK.01.1.1.01.0004 - Znanstveni centar izvrsnosti za kvantne i kompleksne sustave te reprezentacije Liejevih algebri.

Poglavlje 1

Povijest blockchaina

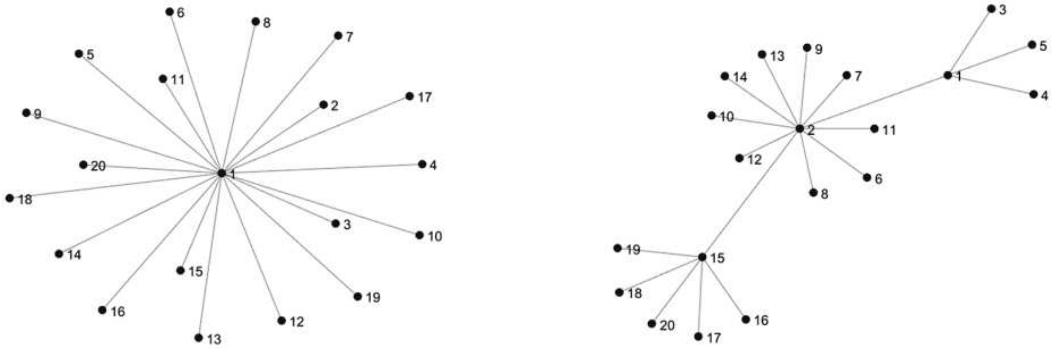
1.1 Tehnologija distribuirane knjige

Kako bismo razumjeli blockchain tehnologiju, prvo moramo razumjeti tehnologiju distribuirane knjige (eng. *distributed ledger technology, DLT*). Jednostavno rečeno, blockchain je jedan od oblika DLT-ja. To je dijeljena, distribuirana knjiga dizajnirana za bilježenje transakcija u poslovnoj mreži i praćenje nastale promjene u vlasništvu imovine. Odgovarajuća imovina može biti materijalna (novac, dionice, nekretnine) i nematerijalna (intelektualno vlasništvo, uključujući patente, zaštitne znakove, autorska prava i prepoznatljivost marke). U teoriji, vlasništvo nad bilo čime vrijednim može se pratiti na mreži blockchain-a, smanjujući rizik i troškove za sve uključene stranke uklanjanjem posrednika transakcija. Osim toga, blockchain je moguće koristiti za uspostavljanje digitalnog identiteta pojedinaca i poduzeća te radikalno reorganiziranje poslovanja.

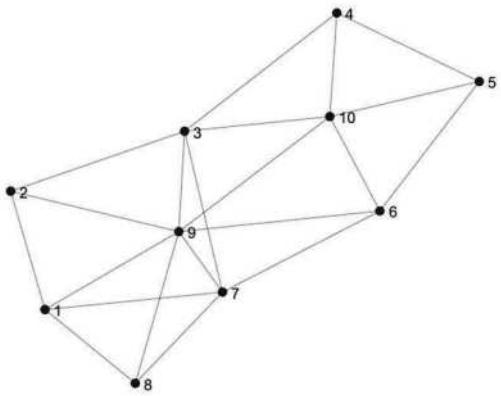
Možemo razmišljati o tri komplementarna načina organiziranja informacija: centralizirani sustav, decentralizirani sustav i distribuirani sustav.
Trenutačno, centralizirani sustav, prikazan na slici 1.1, je temeljna struktura većine industrija. Uzmimo za primjer bankarstvo u jednoj državi, pri čemu je jedan čvor središte koje označava središnju banku, a ostali čvorovi, koji su povezani sa središnjim, su pojedinačne komercijalne banke. Zatim, decentralizirani sustav, također prikazan na slici 1.1, s nekoliko središnjih čvorova tipičan je za aranžmane unutar više različitih država, kao što je prekogranično bankarstvo. Konačno, distribuirani sustav, prikazan na slici 1.2, karakterizira izravnu ili *peer-to-peer* poslovnu organizaciju (trenutačno rijedak slučaj) i potencijalno se može smatrati najsnažnijim od tri spomenuta sustava.

S obzirom na nagli razvoj DLT-a, možemo razmišljati o alternativnim načinima organiziranja poslovnih aktivnosti. Postoje dvije mogućnosti:

1. Trenutni sustav, uređen na takav način da svaki sudionik čuva svoju knjigu, a knjige se povremeno usklađuju jedna s drugom. Ovaj sustav ima svoje prednosti, uključujući,



Slika 1.1: Centralizirani i decentralizirani sustav



Slika 1.2: Distribuirani sustav

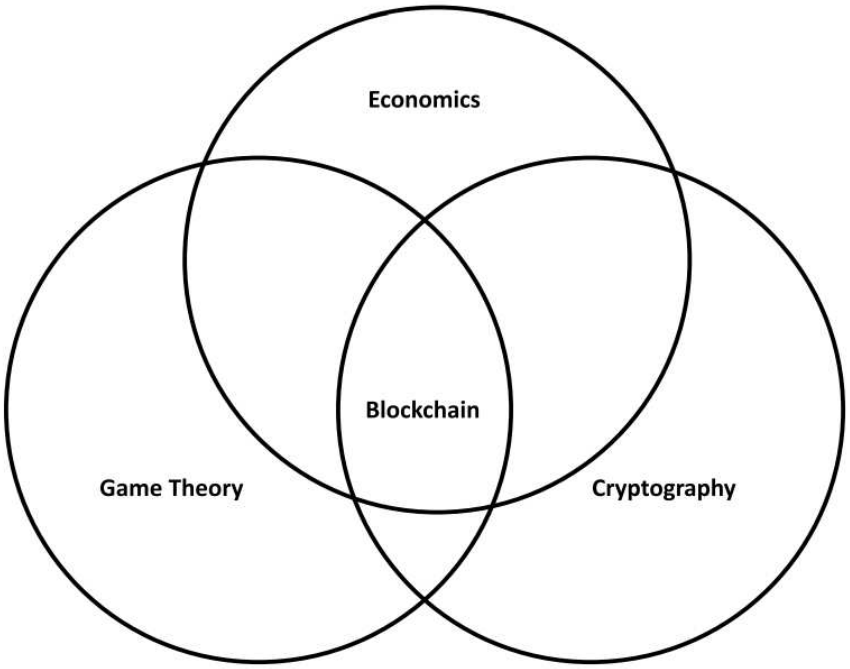
ali ne ograničavajući se na, čvrstu kontrolu informacija. Ipak, po svojoj prirodi, to rezultira pogreškama i potencijalnim prijevarama.

2. Budući sustav, koji se oslanja na sudionike koji održavaju (na ovaj ili onaj način) zajedničku nepromjenjivu knjigu, gdje svaki sudionik može bilježiti transakcije i dohvatiti relevantne informacije koje ima pravo znati. Ako se pravilno izvede, takav sustav može smanjiti ukupna trenja poslovnog procesa i povećati njegovu robusnost. Međutim, racionalizacija poslovnog procesa dolazi sa značajnim troškovima, potrebnim za održavanje konsenzusa o dijeljenoj glavnoj knjizi i pravilnom šifriranju privatnih podataka.

Interes za DLT-jem naglo je porastao potaknut usponom Bitcoina i ostalih kriptovaluta. Jasna je činjenica da je trenutno DLT puno veći od svoje uske primjene na kriptovalute. Iako DLT mnogo obećava, teško ga je pravilno iskoristiti iz razloga što se DLT nalazi na sjecištu tri donekle nepovezana polja:

1. Kriptografije - koja osigurava integritet transakcija
2. Teorije igara - koja uspostavlja konsenzus o stanju distribuirane knjige
3. Ekonomije - koja osmišljava odgovarajuće ekonomske inicijative

te pritom zahtijeva dobro razumijevanje sva tri od navedenih polja (slika 1.3,). Relativno je lako razumjeti bilo koja dva od tri polja, ali savladati sve njih je malo teži zadatak.



Slika 1.3:

DLT je nastao zahvaljujući nizu izvanrednih otkrića u nekoliko područja. Smatra se jednom od temeljnih inovacija kako zatvara jednu od temeljnih praznina. Konkretno, jaz u povjerenju u modernoj internet-orijentiranoj ekonomiji poput razvoja kriptovaluta u kojima DLT igra središnju ulogu.

1.2 Retrospektivni pogled

Ideja prijenosa vrijednih predmeta i vlasničkih prava putem lanaca pomoću nekog oblika konsenzusa nije ništa novo. Na primjer, genealoška stabla kraljevskih i aristokratskih obitelji mogu se smatrati (blok) lancima, budući da prenose prava rođenja prema više ili manje strogom skupu pravila od jednog vladara do drugog. Potrebno je izvrsno vladati genealogijom i heraldikom da bi se ispravno protumačili nevjerojatno detaljni podaci kodirani u tim obiteljskim stablima. Međutim, jedna značajka je nesporna - slično kao i kod Bitcoin transakcija - vlasništvo se prenosi s jednog vlasnika na sljedećeg prema skupu unaprijed određenih pravila. ove prijenose mora prihvatiti grupa ljudi koja provodi pravila, uključujući i druge kraljevske dvore i potvrditi ih.

Kao još jedan primjer ulančanih blokova, ovaj put u ekonomiji, navodimo popis vlasničkih listova u Engleskoj koji su u kontinuiranoj upotrebi još od srednjeg vijeka. Prema izjavi vlade Ujedinjenog Kraljevstva: "Vlasnički listovi su papirnati dokumenti koji pokazuju lanac vlasništva nad zemljom i imovinom. Mogu uključivati: prijenose, kupoprodajne ugovore, oporuke, hipoteke i zakupe." U mnogim su slučajevima ti lanci vrlo dugi, budući da se naslovi mogu pratiti do srednjeg vijeka. Jasno je da su naslovi ulančani blokovi. Umjesto rudara, nasljeđe ovjeravaju javni bilježnici i čuvaju u središnjem repozitoriju.

1.3 Perspektivni pogled

Kao što je ranije spomenuto, za bilježenje transakcija i praćenje imovine na vrlo siguran način koristi se distribuirana knjiga koju dijele sudionici. DLT smanjuje transakcijske troškove, uključujući, ali ne ograničavajući se na trošak provjere transakcije, i poboljšava umrežavanje mogućnosti poslovnog ekosustava. Glavne prednosti DLT-a su: povećanje transakcijske sigurnost, transparentnosti i pouzdanosti; poboljšanje kvalitete i točnosti podataka te smanjenje prijevara i kibernetičkog kriminala. Uvođenje izvorne digitalne imovine, postignuto tokenizacijom imovine u distribuiranoj knjizi, pojednostavljuje i automatizira poslovne procese dopuštajući sudionicima da pristupe povijesti transakcija i utvrde trenutno vlasništvo nad odgovarajućim tokenima.

Izvršenje pametnog ugovora može dovesti do promjene vlasništva. U teoriji, to će omogućiti eliminaciju centraliziranog posrednika i upravljanje tržištem ili s lancom opskrbe na decentraliziran način. Međutim, u praksi je to lakše reći nego učiniti. Profitabilna primjena DLT-a zahtijeva pristup u fazama. Prvo, potrebno je standardizirati i digitalizirati interni poslovni proces izdavanjem digitalnih tokena te reorganizacijom obračuna i izvještavanja. Drugo, tvrtke mogu početi dijeliti digitalne tokene diljem cijelog financijskog ekosustava, čije su usluge jedno su od najistaknutijih područja u kojima DLT može promijeniti tok djelovanja.

Poglavlje 2

Kriptovalute i distribuirane knjige

Godine 2008., anonimni istraživač koji koristi pseudonim Satoshi Nakamoto objavljuje članak *"Bitcoin: A Peer-to-Peer Electronic Cash System"*. U njemu opisuje funkcioniranje decentralizirane i *peer-to-peer* tehnologije koja može održavati glavnu novčanu knjigu. Takva platforma omogućuje novi oblik novca izuzet od središnje uprave ili posrednika. Također jamči visoku dostupnost i neovisnost o lokaciji, a otporna je i na cenzuru. Satoshi Nakamoto objavljuje prvu implementaciju Bitcoina 2009. godine, dok je vrijednost svih bitcoina (BTC) 2022. godine procijenjena na 1.03 trilijuna USD i inspirira tisuće drugih sličnih projekata.

Glavni doprinos Satoshija Nakamota je određivanje arhitekture distribuirane glavne knjige koju vode njezini korisnici, bez ikakvog prethodnog zahtjeva za dopuštenje. Ova distribuirana glavna knjiga, koja se također naziva blockchain zbog svoje strukture podataka, održava potpuni provjereni trag svake transakcije u sustavu i nudi slobodan uvid u kompletno računovodstvo. Blockchain je distribuiran među brojnim korisnicima te je iz tog razloga vrlo otporan na većinu vrsta napada i može nametnuti temeljna pravila Bitcoina — kao što je monetarna politika BTC-a — bez singularne točke povjerenja. Satoshi ga dizajnira takvim da potiče suradnička ponašanja i kažnjava korisnike koji se ne pridržavaju pravila unatoč njihovoj anonimnost.

Budući da se novi sudionici mogu pridružiti ili napustiti mrežu proizvoljno i anonimno, kako se može uspostaviti odnos povjerenja čiji su standardi dovoljno visoki da mogu podržati financijski sustav? Kako otvoriti račun u nedostatku banke? Kako dokazati vlasništvo nad računom u anonimnom sustavu bez ikakvih pravnih identiteta? Gdje pohraniti knjigovodstvenu knjigu za održavanje stanja i transakcija u nedostatku pouzdanog pružatelja pohrane podataka? Kako izbjeći neovlašteno mijenjanje podataka u prisustvu nepoštenih sudionika mreže i bez revizora koji pregledava račune i obvezuje se na njihov integritet?

Unatoč ozbiljnosti ovih izazova, svi sastavni dijelovi Bitcoina, uključujući digitalne

potpise, Merkle stabla (eng. *Merkle trees*), o kojima ćemo detaljnije, te dokaz o radu (eng. *proof-of-work*) temeljen na kriptografskim hash funkcijama su poznati već neko vrijeme. Gradnja ovih blokova kako bi se stvorio siguran sustav uz odgovarajuće poticaje bilo je najimpresivnije postignuće Satoshija Nakamota. Nadalje ćemo proučavati razvoj tog postignuća, uključujući nastanak Bitcoina, glavnih principa i izazova tehnologije distribuirane knjige te trendove koje je pokrenula ova tehnološka inovacija.

2.1 Distribuirani identitet

Pravna identifikacija

U današnjem gospodarstvu vladi se vjeruje da će potvrditi identitet i osigurati materijal za potvrdu identiteta kao što su osobne iskaznice, putovnice i vozačke dozvole. Suočavajući se s identifikacijskim dokumentom, službenik granične kontrole procjenjuje dva ključna zahtjeva: Prvo, je li dokument originalan — da ga je autentično izdala vlada te da još uvijek vrijedi. Za to postoji niz mjera protiv krivotvorenja, uključujući sigurnosnu tintu, poseban papir i materijal, dizajne koji se teško kopiraju, tehnike ispisa, posebni identifikatori i mikročipovi koji uključuju neki oblik kriptografske tehnologije. Drugo, službenik provjerava pripada li putovnica nositelju: odgovara li portretna fotografija, odgovara li boja očiju, visina, otisci prstiju i godine. Budući da je malo vjerojatno da će kombinacija ovih nekoliko osobina odgovarati više pojedinaca, vlasništvo putovnice može se potvrditi.

Ambicija Satoshija Nakamota da u potpunosti distribuira Bitcoin sukobila se s ovim načelom. Nije moguće imati centralno upravljanje izdavatelja identiteta u sustavu čija je bit nepostojanje središnje stranke. Decentralizirani sustav mora se oslanjati na identifikacijski protokol jamčeći neovisnost od središnje vlasti i dopuštajući sudjelovanje bilo kojem korisniku bez prethodnog zahtjeva za dopuštenjem. Napredak postignut s kriptografijom s javnim ključem u prethodnim desetljećima omogućio je Satoshiju Nakamotu da definira jednostavan model temeljen na takozvanoj kriptografiji s javnim ključem.

Digitalna identifikacija

U tradicionalnom pravnom okruženju papirnati primjerak ugovora ovjerava se vlastoručnim potpisom. Temeljna pretpostavka je da samo legitimni ugovaratelj zna kako dizajnirati vlastiti potpis — drugim riječima, da ga je teško replicirati, ali bilo koji vanjski promatrač može to lako usporediti s nekom identifikacijskom dokumentacijom za procjenu njegove valjanosti. U praksi je postalo jasno da je ova pretpostavka nerazumna: krivotvorenje potpisa je relativno jednostavan proces bilo da se postiže ručno ili pomoću računalnih

grafičkih alata, a procjena autentičnosti potpisa je težak zadatak koji može dovesti do visokog rizika od lažno pozitivnih ili lažno negativnih rezultata.

Proces digitalnog potpisa ima za cilj osloniti se na matematičke konstrukcije kako bi se formalno pokazalo da je autentičnost informacija potvrdila određena strana, bez rizika od krive karakterizacije. Oslanja se na dvije povezane stvari: tajni ključ (također nazvan privatni ključ), koji mora biti poznat isključivo osobi koja potpisuje dokument te javni ključ izveden iz tajnog ključa, koji djeluje kao javni identifikator. Digitalni potpis može se izračunati samo iz tajnog ključa i može se provjeriti usporedbom s javnim ključem. Temeljna pretpostavka algoritama potpisa je da je računski neizvedivo izvesti potpis poruke koji odgovara javnom ključu bez poznavanja tajnog ključa. Posljedično, za poruku ovjerenu potpisom koji je lako provjeriti javnim ključem pretpostavlja se da potječe od entiteta koji u vlasništvu ima odgovarajući tajni ključ.

Satoshi Nakamoto definira digitalni novčanik kao par tajnog i javnog ključa. Novčanik je neka vrsta decentraliziranog bankovnog računa koji može primati sredstva, izvršavati vanjske prijenose i biti povezan sa stanjem i povijesti transakcija. Javni ključ novčanika djeluje slično kao broj bankovnog računa: može se dijeliti s drugim sudionicima mreže kako bi se omogućili prijenosi. Tajni ključ, za koji se pretpostavlja da je poznat samo vlasniku novčanika, koristi se za autorizaciju prijenosa putem računanja digitalnog potpisa koji ovjerava prijenos. Vanjski promatrač može procijeniti legitimnost prijenosa provjerom valjanosti digitalnog potpisa u usporedbi s podacima o prijenosu i javnim ključem pošiljatelja.

2.2 Decentralizirani sustav

Model klijent-poslužitelj

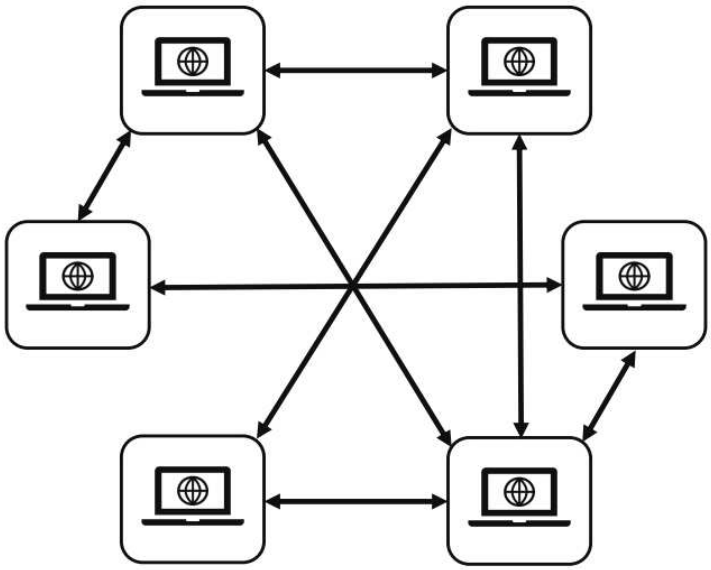
Povijesno gledano, naša interakcija s drugim korisnicima i uslugama na Internetu uglavnom se temeljila na modelu klijent-poslužitelj. Poslužitelji su centralizirana računala koja dopuštaju veze i djeluju kao emiteri informacija, pohrane podataka i procesori. Umjesto poticanja direktne komunikacije među korisnicima, model klijent-poslužitelj oslanja se na posredničkog poslužitelja koji kontrolira protok informacija. Ova višeslojna arhitektura se u prošlosti pokazala lakšim za rad s obzirom na jednostavnost topologije: poznavanje IP adresa poslužitelja i komunikacija s jedinstvenom krajnjom točkom uklanja potrebu za otkrivanje mreže, emitiranje podataka i sinkronizaciju između više čvorova, upravljanje povremenim vezama, upravljanje trajnom pohranom itd.

U decentraliziranom sustavu nastoji se ukloniti ovisnost o određenim centraliziranim sustavima poslužitelja. Centralizirani poslužitelji mogu zakazati, mogu pokvariti poslovne procese ili mogu lažno prikazati informacije sa zlom namjerom, zbog napada ili greškom: oni su suprotni cilju decentralizacije i predstavljaju jednu točku neuspjeha i povjerenja

koje je napadači više puta iskorištavaju za destabilizaciju mrežnih usluga. Oštećenje podataka, krađa podataka, uskraćivanje usluge ili cyber napadi samo su neki od primjera koje poslužitelji doživljavaju i od kojih pate klijenti. Osim toga, kontrola nad središnjim poslužiteljima lako se može zloupotrijebiti od strane pružatelja usluga ili moćnih trećih strana kao što su vlade i regulatori.

Peer-to-peer model

Bitcoin se temelji na takozvanoj *peer-to-peer* mreži, grafički prikazanoj na slici 2.1, gdje svaki čvor ili korisnik može preuzeti ulogu i klijenta i poslužitelja istovremeno. Čvorovi prikupljaju informacije od drugih čvorova, ali također mogu ponuditi informacije drugim korisnicima održavajući međusobne bilateralne veze. Ovaj model daje prednost simetričnoj mreži u kojoj svaki čvor ima kapacitet konzumiranja podataka iz drugih čvorova, ali i da postane aktivni posrednik i poslužitelj podataka. Zbog dupliciranja poslužitelja i mogućnosti guste međusobne povezanosti, dobro razvijene *peer-to-peer* mreže otporne su na mnoge oblike mrežnih kvarova, visoko su dostupne i postojane. Budući da se podaci mogu masovno replicirati preko geografski raspoređenih poslužitelja, namjerno prekidanje mreže ili izazivanje kvarova teško je izvesti. Neke od najpoznatijih primjena *peer-to-peer* mreža uključuju dijeljenje datoteka, distribuirano računalstvo ili telekomunikacijske sustave.



Slika 2.1: Peer-to-peer mreža

Mrežni čvorovi

Svaki korisnik interneta može razviti Bitcoin čvor koji djeluje i kao klijent mreže i kao poslužitelj. Svaki čvor je u interakciji s drugim čvorovima kako bi održao dinamičan pogled na temeljnu knjigu Bitcoina, zvanu blockchain, te za reviziju njezina sadržaja i integriteta. Neki čvorovi, zvani arhivski čvorovi, mogu odlučiti pohraniti i ponuditi potpunu kopiju povijesti blockchaina od njegova osnutka 2009. godine do najnovijih suvremenih ažuriranja. Čvorovi odašilju njihove IP adrese koje služe kao ulazne točke za nove sudionike mreže koji žele preuzeti vlastitu kopiju blockchaina.

Bitcoin čvorovi čine više od pohranjivanja i posluživanja blockchain podataka. Djeluju i kao protokol izvršitelji: oni aktivno nadziru promet, provjeravaju valjanost transakcija, valjanost blockchaina i općenito dobru primjenu pravila protokola kako bi nadzirali svako nekorektno ponašanje. Kooperativni čvorovi presreću nevažeće transakcije, netočne računovodstvene operacije i druge oblike nedopuštenih radnji i namjerno ih zanemaruju kako ne bi zagadili mrežu. Budući da su čvorovi izgrađeni od softvera tipa otvorenog koda (eng. *open-source*) i mogu biti uređivani, kompajlirani i izvršeni od strane bilo kojeg korisnika, korisnik uvijek može biti siguran da njegov čvor strogo provodi pravila protokola i lokalno provjerava je li radnje koje poduzima mreža ne krše nijedno pravilo.

Konfiguracije čvora u mreži

U praksi, čvor slijedi sljedeći životni ciklus:

1. Faza pokretanja: novi čvor povezuje se s jednim ili više izvornih čvorova čije su IP adrese javno dostupne; izvorni čvorovi mogu biti bilo koji kooperativni čvorovi već povezani s mrežom.

2. Otkrivanje čvorova: čvor traži izvorne čvorove za IP adrese drugih, njima poznatih, čvorova unutar mreže, po mogućnosti rekurzivno kako bi se izgradila lokalna karta dostupnih kooperativnih čvorova.

3. Oporavak stanja: čvor preuzima povijest glavne knjige, djelomično ili u cijelosti, iz bilo koje kombinacije otkrivenih kooperativnih čvorova.

4. Mrežno servisiranje: čvor po izboru sam postaje kooperativni čvor i počinje prosljeđivanje informacija primljenih od peer čvorova i omogućavanje drugim vršnjačkim čvorovima da zatraže kopije lokalno pohranjene knjige. Pritom čvor potvrđuje pravila konsenzusa i blokira sve informacije koje odstupaju od njih.

Nakon što je čvor postavljen i ažuriran s ostatkom mreže, počinje slušati sav mrežni promet kako bi saznao o novim uputama i transakcijama koje emitira i obrađuje decentralizirani sustav. Također djeluje kao lokalni pristupnik mreži, iz kojeg se mogu emitirati

sve vrste uputa povezanih s protokolom: ovo je način na koji se nove transakcije kriptovalute prenose cijelom skupu sudionika mreže: poput neke vrste eksponencijalne lančane reakcije, čvor emitira uputu skupu povezanih vršnjaka koji ga odmah dijele sa svojim vršnjacima, i tako dalje sve dok većina čvorova nije obaviještena o novom zahtjevu.

2.3 Distribuirana knjiga

Distribuirana knjiga je računovodstvena baza podataka koju zajednički održava više labavo povezanih i autonomnih strana koje komuniciraju preko *peer-to-peer* mreže. Glavna upotreba ove tehnologije je dijeljenje jedinstvenog računovodstva između različitih, možda nepovjerljivih strana i jamstvo da nitko ne može preuzeti isključivu kontrolu nad mrežom, manipulirati njenim izvršavanjem, selektivno je cenzurirati ili pristupiti njenim privatnim podacima. Iako postoji mnogo različitih arhitektura i protokola koji mogu podržati distribuiranu knjigu, model Satoshija Nakamota, uključujući blockchain podatkovnu strukturu, bio je prvi uspješan pokušaj izgradnje potpuno decentralizirane, sigurne knjige bez potrebnih dopuštenja.

Iako se blockchain može prikazati kao centralizirana ili polucentralizirana pohrana podataka, on postiže svoj pun potencijal kada se otvoreno dijeli na *peer-to-peer* mreži i distribuira među korisnicima slijedeći odgovarajući algoritam konsenzusa. Tko su ti korisnici ovisi o samoj prirodi glavne knjige s obzirom na to da je njen okvir za izdavanje dozvola definiran u samom protokolu: u slučaju mreža kriptovaluta kao što su Bitcoin ili Ethereum, govori se o glavnoj knjizi bez dopuštenja jer im se svaki korisnik može pridružiti, preuzeti cijeli blockchain i preuzeti aktivnu ulogu održavanja bez preliminarnog procesa autorizacije.

Bez obzira na okvir i posebna pravila, distribucija je najbolja zaštita od rizika gubitka podataka, korupcije i manipulacije. Posjedovanjem mnogo suvišnih kopija istih informacija (koje su idealno geografski raspoređene i pod kontrolom različitih, neovisnih entiteta) takve mreže postaju prezaštićene za napad jer svaki pojedini sudionik u potpunosti doprinosi njenom punom obliku i obliku.

Struktura podataka blockchaina

Blockchain je kriptografski lanac blokova podataka. Svaki blok sadrži skup transakcija koje kronološki slijede transakcije prethodnih, nadređenih blokova i prethode mogućim budućim transakcijama svojih podređenih blokova. S izuzetkom izvornog bloka, također nazvan blok podrijetla (eng. *genesis block*), svaki sljedeći blok održava eksplicitni kriptografski pokazivač na svog roditelja tako da se može lako preputovati lancem te rekonstruirati kompletan graf transakcija. Blockchain stoga održava iscrpan popis potvrđenih

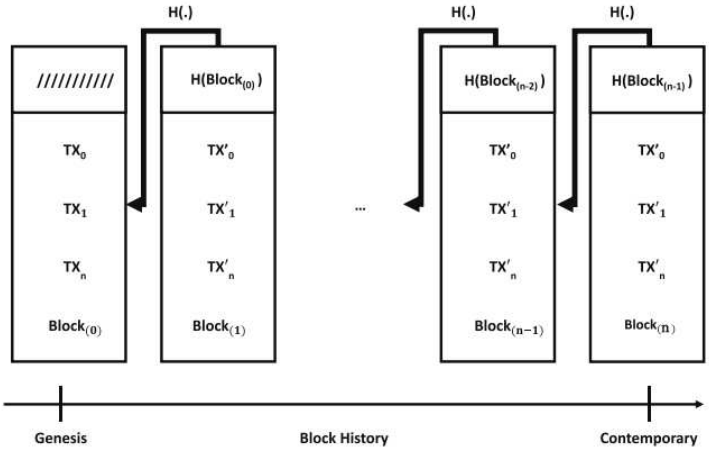
transakcija od nastanka do trenutnih ažuriranja, kao što je prikazano na slici 2.2. Pojednostavljena podatkovna struktura lanca blokova može se specificirati na sljedeći način:

1. Neka *hash* poruka bude kriptografska hash funkcija. Po konstrukciji, hash je bijektivan jer se praktički ne može generirati poruka koja daje zadanu izlaznu vrijednost niti pronaći dvije različite poruke s istom izlaznom vrijednošću. Hash se može koristiti za izračunavanje digitalnog ekvivalenta otisku prsta, tako da svaka promjena izvorne poruke, koliko god mala bila, poništava njen hash.

2. Neka je $block_0$: $transactions_0$ izvorni blok blockchaina, gdje je $transactions_0$ lista transakcija unutar bloka.

3. Neka je $block_i$: $hash(block_{i-1})$, $transactions_i$ struktura podataka bilo kojeg djeteta izvornog bloka, gdje je $i > 0$ te je $transactions_i$ lista transakcija unutar $hash(block_i)$

4. Blockchain je niz blokova $(block)_{i=0}^{N}$ gdje je N indeks istovremenog bloka (eng. *contemporaneous block*)



Slika 2.2: Blockchain model podataka

Izgradnjom glavne knjige s gornjom strukturom, moguće je stvoriti povjerenje u cjelovitost podataka usprkos nepouzdanom okruženju unutar kojeg je knjiga često izložena i održavana. Sve dok je najnoviji $blok_N$ poznat, kompletna povijest je strogo definirana i ne može biti predmet bilo kakvog manipuliranja bez ugrožavanja valjanosti lanca. Razlog je evidentan u njegovoj rekurzivnoj definiciji: $blok_N$ sadrži hash $bloka_{N-1}$, dakle $blok_{N-1}$ ne može se neprimjetno manipulirati. $blok_{N-1}$ sadrži hash $blok_{N-2}$, stoga vrijedi isto jamstvo integriteta podataka. Rekurzivnom primjenom sličnog razmišljanja zaključuje se da ovaj

lanac hashiranja osigurava podatke svih povijesnih blokova transakcija natrag do izvornog bloka. Napadi s ciljem promjene ravnoteže, izmjene ili otkazivanje transakcija stoga ne dolaze u obzir u lancu koji se već dijeli među ostalim sudionicima mreže osim ako se ne postigne novi konsenzus između mrežnih čvorova na istovremenom bloku.

Ova shema jamči da je odluka o istovremenom bloku ekvivalentna odluci o cjelokupnoj povijesti blockchaina. Proučavajući povijest blockchaina, smanjujemo složenost protokola za dizajn konsenzusnog algoritma te na taj način olakšavamo postizanje dogovora s drugima sudionika mreže oko najnovijeg bloka.

2.4 Problem dvostruke potrošnje

Do sada smo proučili glavna načela DLT-a s podatkovnom strukturom lanca blokova. Komunikacija između čvorova je bez posrednika ili *peer-to-peer*; knjigu održava bilo koji dobrovoljni čvor na način da postoji mnogo suvišnih kopija; podaci su ovjereni s odgovarajućim shemama raspršivanja tako da je cijela povijest evidentna. Korisnički računi su pseudonimni parovi tajno-javnih ključeva, gdje je javni ključ identifikator računa koji se može dijeliti, a tajni ključ se koristi za autorizaciju transakcija s digitalnim potpisom.

Gore navedeni model je moćan, ali sam po sebi prirodno ne nameće osnovne transakcijske principe. Platna mreža mora pružiti visoka jamstva da korisnik ne može potrošiti više nego što posjeduje. Ovo može zvučati kao problem koji je jednostavno riješiti - zapravo, općenito se svodi na jednostavnu provjeru da zbroj odlaznih transfera nije veći od ukupne vrijednosti dolazni prijenosa, ali distribuirane knjige dolaze sa svojim složenostima koje čine problem kompliciranijim. U nastavku ilustriramo glavni problem, koji se obično naziva dvostruka potrošnja.

Ilustracija problema u centraliziranom sustavu

Razmotrimo primjer tradicionalnog bankovnog računa sa stanjem računa 1000 kn. Također pretpostavimo da bi vlasnik računa htio kupiti crveni i plavi bicikl, svaki vrijedan 1000 kn. Intuitivno, vlasnik može kupiti samo crveni ili plavi bicikl s obzirom na to da je ukupna vrijednost oba bicikla 2000 kn, što premašuje njegovo ukupno stanje. Neka vlasnik pokuša prevariti sustav kreiranjem dva zahtjeva za plaćanje u kratkom vremenskom razdoblju, prvi za crveni bicikl, a drugi za plavi bicikl. Banka prima obavijest za oboje zahtjeve za plaćanje, poreda ih (na primjer, na temelju vremenske oznake ili prioriteta) i inicira obradu inicijalnog zahtjeva za plaćanje. Nakon jednostavne provjere banka može provjerite je li stanje dostatno za prvu uplatu, obradi ga i ažurira stanje na 0 kn. Zatim se pokreće obrada drugog zahtjeva za plaćanje, ali izvršavanje ne uspijeva zbog nedovoljnog stanja računa. Vlasnik je obaviješten da njegova transakcija nije izvršena.

Ilustracija problema u decentraliziranom sustavu

Razmotrimo sada sličnu situaciju u decentraliziranom okruženju poput Bitcoin-a. Vlasnik računa sada drži 1 BTC u Bitcoin knjizi i želio bi kupiti i crveni bicikl i plavi bicikl, svaki vrijedan 1 BTC. Intuicija sugerira da vlasnik može kupiti samo jedno ili drugo, budući da je ukupna vrijednost oba bicikla 2 BTC. Vlasnik ima za cilj prevariti sustav i kreira dvije transakcije u kratkom vremenskom razdoblju. Prva transakcija šalje 1 BTC za plaćanje crvenog bicikla dok druga transakcija šalje 1 BTC za plaćanje plavog bicikla.

Nakon toga započinje peer-to-peer emitiranje oba zahtjeva za transakciju i svaki mrežni čvor obaviješten o bilo kojoj od transakcija odmah je emitira vlastitom skupu susjednih čvorova. Nakon toga, jedna, druga ili obje transakcije brzo postaju poznate svim sudjelujućim čvorovima na Internetu. Zbog toga što nema determinizma, ne postoji jamstvo da svi čvorovi prvo vide transakciju crvenog bicikla ili transakciju plavog bicikla. Čvorovi se ne mogu zajednički dogovoriti o tome koju transakciju smatrati ispravnom i koju poništiti. Zapravo, izvorni pošiljatelj može pokušati prevariti sustav slanjem transakcije crvenog bicikla na određenu particiju mreže tijekom slanja transakciju plavog bicikla na drugu particiju mreže, tako da različite grupe čvorova vide crvene i plave transakcije različitim redoslijedom.

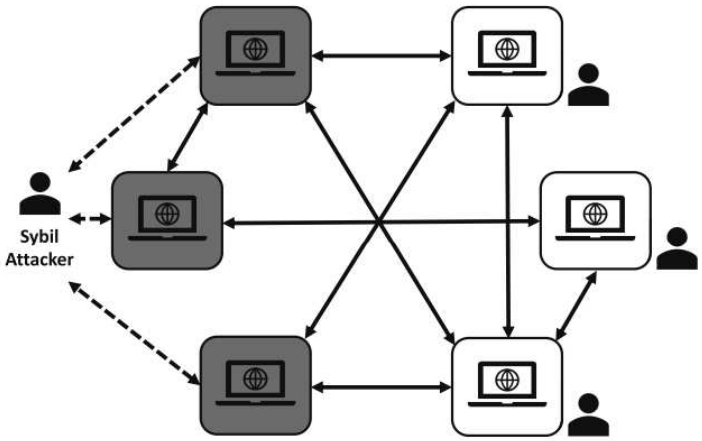
Budući da ne postoji prirodni redoslijed transakcija, ne postoji ni dogovor na razini cijele mreže koju od dvije transakcije treba smatrati valjanom, a koju treba odbiti. Ne može se vjerovati ni vremenskoj oznaci koju bi pošiljatelj priložio svojim danim transakcijama jer nema načina da se provede valjanost vremenske oznake. Posljedica ovog nedeterminizma je problem koji se često naziva problemom dvostruke potrošnje (eng. *double spending problem*).

2.5 Mrežni konsenzus

Očito je da distribuirana knjiga treba robusno rješenje protiv problema dvostruke potrošnje. Drugim riječima, mora ugraditi algoritam konsenzusa koji omogućuje svakom sudioniku mreže da postigne visoku sigurnost da njihov stav o tome koju transakciju zadržati dijele i ostali sudionici mreže. Algoritam konsenzusa je upravo dogovor o odluci kada je uključeno više od jednog čvora i osiguravanje da je sporazum otporan na različite vrste manipulacija ili napada. U slučaju dvostruke potrošnje, konsenzus ima za cilj promicanje odluka o tome koju transakciju zadržati, a koju zanemariti. Potreba za konsenzusom ne jamči čuvanje najranije transakcije; ono samo pokreće opći dogovor s kojom transakcijom se ”mreža” slaže.

Sybil napadi

Promotrimo jednostavan model gdje je potrebna većina glasova sudionika mreže da bi se odobrila određena transakcija. Prema ovom modelu, prvo se čeka da se postigne većina glasovna da bi se zaključilo da je transakcija odobrena. U praksi, trgovac biciklima prodaje bicikl tek nakon što je potvrđeno 51% glasova u korist transakcije koja plaća njegov proizvod. Pitanje je bi li takav model bio dostatan?



Slika 2.3: Sybil napad

Pažljiva analiza brzo otkriva da ne bi. U sustavu bez potrebe za dopuštenjem je teško nametnuti ispravan sustav glasovanja. Sustav ne može jamčiti da sudionik mreže glasuje samo jednom kada ima tehničke mogućnosti umnožiti svoje Internetske veze i umjetno pojačati njegovu prisutnost po volji i to neotkriveno. Ovaj problem često se naziva *Sybil napadom* (eng. *Sybil attack*). Sudionik mreže napada sustav stvaranjem proizvoljnog broja pseudonimnih identiteta koje koristi kako bi povećao svoj utjecaj na ishod glasovanja, kao što je prikazano na slici 2.3. Sybil napadi su posebno učinkoviti zbog toga što se stvaranje identiteta može postići uz niske troškove i stoga masovno skalirati te manipulirati procesom glasanja.

Alternativni model bi bio da se izabere vođa među sudionicima mreže koji bi imao ovlast da jednostrano odlučuje o transakcijama koje će odobriti ili odbiti. U distribuiranom sustavu, svaki sudionik mreže bi imao mogućnosti postati vođa te ta pozicija ne bi trebala biti stalna kako bi se izbjegla recentralizacija donošenja odluka. Ovaj pristup samo pomiče problem za jedan korak naprijed te stvara novi izazov: proces izbora vođe. Izabrati vođu nije ništa lakše nego glasovati u korist određene transakcije: još uvijek je podložan Sybil napadima i stoga ne može izabrati legitimnog vođu.

Algoritam konsenzusa Satoshija Nakamota precizno rješava gore navedeni izazov. Uspostavlja novi algoritam konsenzusa za distribuirane knjige bez dopuštenja, što čini Sybil napade skupima. Kako bi se uspostavio ispravan sustav poticaja, sudionici mreže mogu biti nagrađeni pravilnim ponašanjem, a kažnjeni zbog nedoličnog. Ovo je takozvani algoritam dokaza o radu (eng. *Proof-of-Work, PoW*).

Dokaz o radu

U kontekstu u kojem svaki sudionik mreže može povećati svoju percipiranu prisutnost po želji, na primjer umjetnim umnožavanjem svojih internetskih veza i gdje sudionici nemaju razloga vjerovati jedni drugima, kako se može postaviti proces pod kojim se može postići dogovor za cijelu mrežu? Treba li vođa donositi odluke i ako da, kako se složiti tko bi trebao biti taj vođa? Treba li postojati model demokracije bez vođe, i ako da, kako osigurati da svaki sudionik glasuje samo jednom?

Čini se jasnim da napadi Sybil sprječavaju svaki oblik naivnog demokratskog modela zbog nepovjerenja između anonimnih sudionika i njihove sposobnosti zlouporabe glasovanja. Da bi radio, Bitcoin mora dokazano ograničiti sposobnost sudionika da proizvoljno povećaju svoju glasačku moć. Bitcoin mora postaviti sustav kojem svaki sudionik može doprinijeti, ali glasovanje ima svoju cijenu. Bitcoin mora stvoriti demokratski sustav bez identiteta koji ne dopušta prijevare.

Satoshi Nakamoto postavlja Bitcoin protokol tako da je valjanost transakcije bloka uvjetovano je preliminarnim rješavanjem skupog problema. Protokol jamči da se samo blokovi s važećim dokazom rezolucije problema — ili algoritmom *proof-of-work*, prihvaćaju kao valjani blokovi. Mrežni čvorovi odbijaju blokove bez važećeg *proof-of-work-a*.

PoW prisiljava čvor blockchain mreže na rješavanje kriptografskog problema kao dokaz poštenosti. Kako bi riješili problem, rudari moraju pronaći nasumičnu vrijednost, takozvani *guess*. Takav broj, konkateniran s transakcijama unutar bloka te hash-om prethodnog bloka, mora dati cjelokupni hash manji od ciljanog broja. Ciljani broj je određen s blockchain *difficulty* vrijednosti koja regulira prosječno vrijeme rudarenja utrošeno od strane rudara kako bi riješili zadatak. Kako bi se pružio dokaz o radu, u posebnu varijablu zapisuje se broj koraka potrebnih za rješavanje problema. Na taj način svi ostali čvorovi blockchain mreže jednostavno validiraju ispravnost dokaza. Ovaj pristup još uvijek dopušta da bilo koji sudionik mreže može pridonijeti mreži s novim blokom transakcija sve dok novostvoreni blok sadrži valjani dokaz rada.

Konstruiranje blockchaina

Konstrukcija blockchaina odvija se prema sljedećem procesu:

1. Preuzmemo postojeći lanac blokova sve do istovremenog bloka od drugih korisnika mreže.

2. Procijenimo valjanost lanca, njegove transakcije i njegov dokaze o radu u odgovarajućim blokovima.

3. Odaberemo skup transakcija koje čekaju na uključenje u blockchain. Ove transakcije, koje su razmijenjene između korisnika mreže nakon emitiranja njihovih odgovarajućih pošiljatelja, pohranjuju se u memorijsko spremište (eng. *memory pool*) odgovarajućih mrežnih čvorova koji čekaju da budu uključeni u novi blok.

4. Izgradimo blok s odabranim transakcijama, postavimo nadređeni (eng. *parent*) hash unutar bloka i počinjemo s izračunavanjem dokaza o radu.

5. Ako i kada identificiramo valjani dokaz, uključujemo ga unutar bloka i emitiramo blok drugim čvorovima.

6. Drugi čvorovi zatim provjeravaju integritet bloka, uključujući njegov dokaz o radu i sukladno tome donose odluku o tome hoće li ga uključiti u svoju lokalnu kopiju blockchaina.

Istovremeno i svi drugi korisnici mreže rade isti postupak kako bi predali svoje mišljenje o sljedećem bloku. Međutim, ostaju mnoga pitanja. S obzirom na troškove povezane s proizvodnjom bloka, zašto bi se itko dobrovoljno bavio ovim zadatkom? Koje transakcije odabrati za pohranjivanje unutar bloka? Koji blok odabrati u slučajevima konflikta?

Rudarenje blokova

Proizvodnja blokova, koja je uvjetovana važećim dokazom o radu, je skup i zahtjevan posao. Korisnici mreže neće dobrovoljno sudjelovati u proizvodnji bez odgovarajuće sheme poticaja. Zbog toga Nakamoto predlaže nagradu kao dio protokola koju kreator dobiva zbog odobrenog bloka od strane ostatka mreže, Bitcoin-e.

Zbog labave usporedbe s rudarenjem zlata, proizvodnja blokova je metaforički nazvana rudarenje, a korisnici koji obavljaju posao nazivaju se rudari. Neki korisnici Bitcoin mreže postali bi aktivni održavatelji, odnosno rudari i pratili bi mrežu kako bi prikupili transakcije od drugih korisnika, grupirali ih u blokove, započeli s rješavanjem dokaza o radu i ovisno o njegovom rješavanju, emitirali blok ostatku mreže kako bi pokupili nagradu.

Poglavlje 3

Kriptografske hash funkcije

U ovom poglavlju opisujemo prethodno spomenutu kriptografsku strukturu iznimne važnosti, hash funkcije. Hash funkcija može preslikati poruku proizvoljne duljine u poruku fiksne duljine, recimo 256 bita. Praktički je nemoguće pronaći neku drugu poruku s istim hashom iako, očito, takve poruke postoje. Korištenjem hash funkcija može se značajno smanjiti računalno opterećenje, kao što je digitalno potpisivanje poruke jer postaje moguće potpisati hash poruke poruka, a ne samu poruku. Jednostavno rečeno, hash se može smatrati otiskom prsta osobe — ne govori vam ništa o tome tko je vlasnik, ali vlasnik može lako dokazati da je otisak prsta doista njezin.

3.1 Osnovni alati kriptografije

Prije opisa hash funkcija, opišimo prvo neke osnovne alate kriptografije. Površno ćemo se osvrnuti na pojmove kriptografije simetričnog, asimetričnog ključa te kriptografskih jednosmjernih funkcija kako bi olakšali razumijevanje hash funkcija i njihove primjene. Mnogi protokoli, poput Bitcoina i Etheruma oslanjaju se na činjenicu da asimetrična kriptografija omogućava digitalno potpisivanje poruka pri čemu je valjani potpis nepobitan dokaz da potpisnik posjeduje tajni ključ povezan s njezinim javnim ključem.

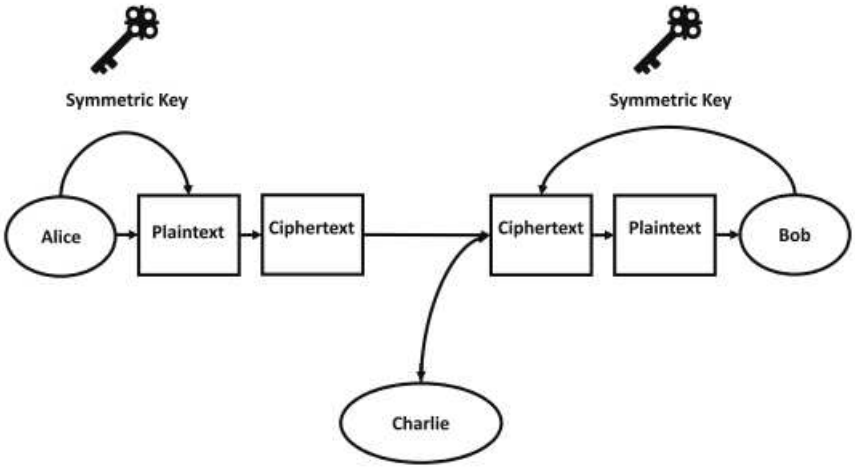
Kriptografija simetričnog ključa

Poruka otvorenog teksta sadrži informacije u svom standardnom obliku. Šifrirani tekst ili kriptogram je transformirana poruka. Ključ (ključevi) je (su) tajni parametri za enkripciju, koji su poznati samo pošiljatelju i namijenjenom primatelju. Tajni ključ(evi) se koriste za izvođenje transformacije iz otvorenog teksta u kriptogram (šifriranje) i iz kriptograma u otvoreni tekst (dešifriranje). Šifre se temelje na općem algoritmu koji koristi tajni

ključ za izvođenje šifriranja i dešifriranje. Kriptoanaliza predstavlja metodu za pretvaranje kriptograma natrag u izvorni otvoreni tekst bez prethodnog znanja o ključu.

Generalna ideja prikazana je na slici 3.1. Alice koristi tajni ključ za šifriranje poruke i emitira ga Bobu, koji koristi isti tajni ključ za dešifriranje kriptograma. Charlie, protivnik, pokušava prisluškivati Aliceinu i Bobovu komunikaciju pomoću kriptoanalize. Ako uspije, ne samo da može čitati njihove poruke nego i pokušati ih krivotvoriti.

Prema Kerckhoffovom principu, kriptografska shema je sigurna kada je sve o njoj, osim ključa, javno poznato, a ipak je se ne može probiti.



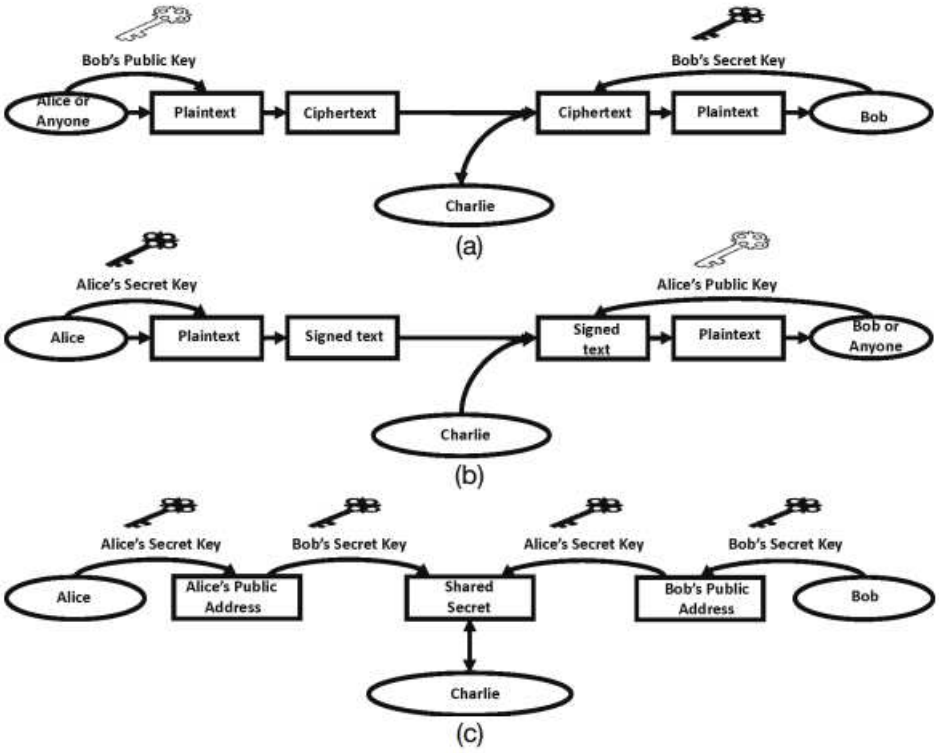
Slika 3.1: Kriptografija simetričnog ključa

Kriptografija asimetričnog ključa

Kod kriptografije simetričnog ključa, pretpostavka je da se isti ključ koristi za šifriranje i dešifriranje. Međutim, postoji i drugačiji pristup. U 1970-ima nekoliko je istraživača predložilo korištenje *različitih ključeva* za šifriranje i dešifriranje. Spoznaja da je to moguće i, u mnogim slučajevima, poželjno, otvara novo poglavlje u kriptografiji — tzv. kriptografiju s javnim ključem koja se oslanja se na infrastrukturu javnog ključa (eng. *public-key infrastructure, PKI*). Na slici 3.2 grafički prikazujemo proces šifriranja i dešifriranja poruke; na slici 3.2 (b) prikazujemo digitalni algoritam potpisa, dok na slici 3.2 (c), pokazujemo kako dvije strane mogu stvoriti zajedničku tajnu.

Promotrimo sljedeći primjer. Kao i prethodno Alice i Bob komuniciraju preko poruka koje žele zaštititi od nepoželjnih osoba koje ih prisluškuju, kao što je Charlie. Na slici 3.2 a) Alice koristi Bobov javni ključ koji je lako dostupan za šifriranje poruke, koju samo Bob

može dešifrirati pomoću svog tajnog ključa koji je poznat samo Alice. Charlie pokušava dešifrirati poruku namijenjenu Bobu i eventualno je zamijeniti krivotvorenom. Na slici 3.2 b) Alice koristi svoj tajni ključ da potpiše poruku ili njen hash. Hashiranje je algoritam za stvaranje jedinstvenog sažetka fiksne duljine iz poruke proizvoljne duljine. Njezin potpis može provjeriti Bob (ili bilo tko drugi). Charlie pokušava stvoriti krivotvorenu poruku i pretvara se da ju je Alice potpisala. Konačno, na slici 3.2 c) Alice i Bob koriste svoje tajne i javne ključeve za stvaranje zajedničku tajnu, dok je Charlie pokušava razotkriti.



Slika 3.2: Kriptografija asimetričnog ključa

Kriptografske jednosmjerne funkcije

Prije opisa hash funkcija, uvodimo još pojam kriptografskih jednosmjernih funkcija (eng. *cryptographic one-way functions*). One su građevni blokovi asimetrične kriptografije. Jednosmjerna funkcija je funkcija koju je lako izračunati, ali je teško pronaći inverz te funkcije. Na primjer, lako je pronaći $y = f(x)$ za dani x, ali je teško pronaći $x = g(y)$.

Svaka hash funkcija s dužim ulazom nego izlazom nužno ima kolizije. Na primjer, hash funkcija kao što je SHA-256 proizvodi 256 bita izlazne vrijednosti s bilo kojom ulaz-

nom vrijednosti (koja može biti dugačka i do $2^{64} - 1$ bita). Dakle, SHA-256 generira 2256 izlaznih vrijednosti za mnogo veći set od $2^{2^{64}} - 1$ ulaznih vrijednosti, tako da će se neki ulazne vrijednosti hashirati na istu izlaznu vrijednost, to je tzv. princip goluba (eng. *pigeonhole principle*). Nadalje pokazujemo kako izgraditi funkcije otporne na kolizije, što je vrlo važne za kriptografiju.

Hash funkcije otporne na kolizije

Hash funkcija koja je **jako** otporna na kolizije je funkcija koju je lako izračunati, ali je teško pronaći bilo kakvu koliziju. Drugim riječima, teško je pronaći x i x' $x \neq x'$, tako da

$$h(x) = h(x').$$

Hash funkcija koja je **slabo** otporna na kolizije je jednosmjerna funkcija koju je lako izračunati, ali je teško pronaći drugu prasliku za dani x. Dakle, za dani x, teško je pronaći x', $x \neq x'$, takav da

$$h(x) = h(x').$$

Glavna razlika između jako i slabo otpornih hash funkcija je u tome što kod jako otpornih nemamo zadane x i x', dok u slabo otpornim funkcijama imamo zadani x, a potrebno je pronaći odgovarajući x'. Kao što je jasno iz imena, jaka otpornost je zahtjevnija za postizanje od slabe otpornosti na koliziju.

Jako otporne hash funkcije su korisne ako netko želi stvoriti hash operator za unose u opsežnu bazu podataka i biti siguran da dva unosa nemaju istu oznaku. Slabo otporne hash funkcije mogu se koristiti kako bi se osiguralo da se određena lozinka, pohranjena u bazi podataka hashiranih lozinki, ne može lako probiti.

3.2 Digitalni potpisi i hash funkcije

Potpisivanje hasha dodaje još jednu mogućnost napada i predstavlja sljedeći problem: pretpostavimo da je Charlie vidio par $(m, sign(h(m)))$ potpisan od strane Alice i želi se pretvarati da je ona potpisala njegovu poruku m' umjesto toga, što bi bilo lako učiniti ako je $h(m') = h(m)$, jer, očito,

$$sign(h(m)) = sign(h(m')).$$

Stoga bi dobra hash funkcija $h(m)$ trebala otežati pronalaženje poruka m' takvih da $h(m') = h(m)$, tj. to mora biti hash funkcija koja je slabo otporna na kolizije.

Rođendanski napad

Problem rođendana javlja se u teoriji vjerojatnosti. Problem je pronaći vjerojatnost da u skupu od n nasumično odabranih ljudi postoji par s istim rođendanom. Ovaj problem se naziva i rođendanski paradoks, ali ne zato to je paradoks sam po sebi, već zato što se rezultat čini vrlo kontraintuitivan na prvu. Ova je vjerojatnost očito 100% za $n = 367$ jer je moguće samo 366 rođendana (u set uključujemo i 29. veljače). Nadalje u promatranju zanemarujemo prijestupne godine i pretpostavimo da su svi dani u godini jednako vjerojatni. Vjerojatnost od 50% postignuta je već sa samo 23 osobe, tj $n = 23$, a vjerojatnost od 99,9% je postignuta sa 70 osoba, tj $n = 70$.

Rješenje rođendanskog problema je jednostavno. Razmotrimo skup od N ljudi, $N \leq 365$. Moramo izračunati vjerojatnost da njih dvoje (ili više) imaju isti rođendan, što označavamo s P_N. U ovom slučaju, lakše je izračunati komplementarnu vjerojatnost $P'_N = 1 - P_N$ da nikoje dvije osobe nemaju isti rođendan. Označimo proizvoljno ljude od 1 do N. Kažemo da se "Ishod 2" događa kada osoba 2 nema isti rođendan kao osoba 1, "Ishod 3" se događa kada osoba 3 nema isti rođendan kao osoba 1 ili osoba 2, i tako dalje. Jasno je da je P'_N vjerojatnost "Ishoda N", koja se može izračunati korištenjem uvjetne vjerojatnosti na sljedeći način:

$$P'_2 = 1\frac{364}{365} = 1(1 - \frac{1}{365}), P'_3 = P'_2\frac{363}{365} = P'_2(1 - \frac{2}{365}),$$

$$P'_N = P'_{N-1}\frac{365 - (N-1)}{365} = P'_{N-1}(1 - \frac{N-1}{365}),$$

tako da

$$P'_N = \frac{365!}{365^N(365 - N)!}.$$

Prema tome,

$$P_N = 1 - P'_N.$$

Posebno, $P_{23} = 50.7\%, P_{70} = 99.9\%$. Koristeći činjenicu da je $1 - x \approx e^{-x}$, kada $x \ll 1$, nije teško za vidjeti da P_N može biti aproksimiran kao:

$$P_N \approx 1 - e^{N(N-1)/730}$$

Posljedice na kriptografiju su značajne. Algoritam rođendanskog napada omogućuje nam pronaći koliziju hash funkcije u $2^{L/2}$ pokušaja, gdje je L broj bitova koji definiraju klasičnu sigurnost otpora praslike. Navodimo jednostavan primjer:

Pretpostavimo da Charlie zna da se Alice spremna potpisati dokument d. On želi konstruirati drugačiji dokument d' i tvrditi da ga je Alice potpisala. Na primjer, umjesto "Plaćanje po nalogu Boba", želi da na potpisanom dokumentu stoji "Plaćanje po nalogu Charlieja". Naravno, samo mijenjanje Bobovog imena u Charliejevo ne bi pomoglo jer će

hashevi odgovarajućih dokumenata biti potpuno drugačiji zbog otpornosti na koliziju hash funkcije. Međutim, Charlie može postupiti na sljedeći način.

Charlie uzima pravi ugovor d i proizvodi male varijacije dodavanjem razmaka na kraj retka, neznatno mijenjajući tekst ili čineći druge bezazlene izmjene. Charlie preuzima lažni ugovor d' i proizvodi slične male promjene u njemu. K promjena će dati 2^K različitih dokumenata, $d_1, ..., d_{2^K}$, i slično, 2^K različitih dokumenata $d'_1, ..., d'_{2^K}$. Nakon toga pokušava pronaći podudaranje hash vrijednosti $h(d_k)$ i $h(d'_l)$, za proizvoljan par (k, l), gdje $1 \leq k, l \leq 2^K$. Ako postoji podudaranje $h(d_k) = h(d'_l)$ za ta dva ugovora će vrijediti isti potpis. Charlie predstavlja prihvatljivu verziju, d_k, da Alice potpiše. Nakon što je Alice potpisala ugovor, Charlie preuzima potpis, prilaže ga izmijenjenom ugovoru d'_l i tvrdi da ga je Alice potpisala. Ako su hash vrijednosti kraće od 2K bita, tada je vjerojatnost podudaranja vrlo visoka zbog rođendanskog paradoksa.

3.3 Pouzdani hash algoritmi

Iako je dizajniranje pouzdanog, sigurnog hash algoritma (eng. *Secure Hash Algorithm, SHA*) teško, postoje brojni kandidati. Bitcoin i mnogi drugi sustavi koriste SHA-256 i RIPEMD-160 (eng. *RIPE Message Digest*). U sklopu blockchain tehnologije, funkcija SHA-256 je između ostalog korištena pri povezivanju blokova, dodavanju novih blokova te generiranju sažetaka od svih podataka iz jednog bloka. Kao što mu ime sugerira, SHA-256 preslikava proizvoljan niz u niz duljine 256 bita.

U praksi se koristi nekoliko hash algoritama, kao što su MD5, SHA1, SHA2. Međutim, stariji algoritmi, kao što je MD5, su nevažeći i više se ne mogu koristiti u kriptografske svrhe. U nastavku opisujemo algoritam SHA-256 koji koristi Merkle-Damgård transformaciju i pretvara funkciju s ulazima fiksne duljine u funkciju s ulazima proizvoljne duljine primjenjujući ih sekvencijalno. Ulaz je dopunjen prema potrebi kako bi njegova duljina bila višekratnik 512 bita.

Algoritam SHA-256

Kao što je prethodno spomenuto, SHA-256 pretvara poruku promjenjive duljine u izlaz fiksne duljine od 256 bita (32 bajta). Za razumijevanje rada algoritma SHA-256, dovoljno je razumjeti kako se jedan 512-bitni ulaz preslikava u 256-bitni hash. Prije daljnjeg opisa, navodimo neke definicije koje koristimo u daljnjem razmatranju.

Definicija 1. *Neka je W neka riječ te neka je W(i) znak na indeksu i,* $0 \leq i \leq m$*, gdje je m duljina riječi W. Desnu rotaciju riječi W za n bitova označavamo sa* $ROTR^n$ *i definiramo kao:*

$$ROTR^n(W) = W',$$

gdje je riječ W' dobivena formulom:

$$W'(i) = W((i - n) \bmod m)$$

Odnosno, riječ W pomičemo za n bitova udesno te bitove s kraja vraćamo na početak.

Primjer desne rotacije za dva bita za ulaznu riječ W = 1001101:

1001101

⇓

0110011

Definicija 2. *Neka je ponovo W neka riječ te W(i) znak na indeksu i,* $0 \leq i \leq m$*, gdje je m duljina riječi W. Desni pomak riječi W za n bitova označavamo sa* SHR^n *i definiramo kao:*

$$SHR^n(W) = W',$$

gdje je riječ W' dobivena formulom:

$$W'(i) = \begin{cases} W(i-n), & ako\ i - n \geq 0 \\ 0, & inače \end{cases}$$

Odnosno, riječ W pomičemo za n bitova udesno te ostale bitove popunjavamo s nulama.

Primjer desnog pomaka za dva bita za ulaznu riječ W = 1001101:

1001101

⇓

0010011

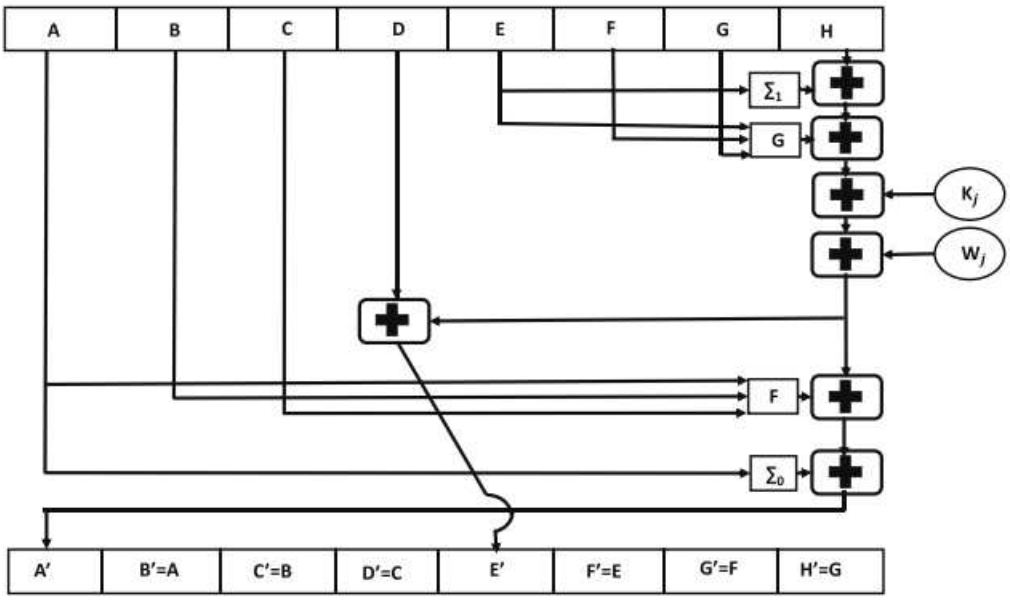
Ulaz SHA-256 algoritma podijeljen je na 16 32-bitnih riječi, označenih s $W_0, ..., W_{15}$. Skup od 16 riječi je proširen na skup od 64 riječi prema pravilu

$$W_j = \sigma_1(W_{j-2}) + W_{j-7} + \sigma_0(W_{j-15}) + W_{j-16}, 16 \leq j \leq 63.$$

gdje su

$$\sigma_0(W) = ROTR^7(W) \oplus ROTR^{18}(W) \oplus SHR^3(W),$$
$$\sigma_1(W) = ROTR^{17}(W) \oplus ROTR^{19}(W) \oplus SHR^{10}(W),$$

Budući da ima 64 riječi, postoje 64 transformacije. Na slici 3.3 je prikazan dijagram toka algoritma SHA-256 za zadanu riječ W_j i konstantnu riječ K_j.



Slika 3.3: Dijagram toka algoritma SHA-256

Definicija 3. *Neka je funkcija F zadana na sljedeći način:*

$$F(A, B, C) = (A \wedge B) \oplus (\neg A \wedge C),$$

te neka su:

$$B_j = A_{j-1}, C_j = B_{j-1}, D_j = C_{j-1},$$

$$E_j = D_{j-1} + H_{j-1} + \sum_1(E_{j-1}) + G(E_{j-1}, F_{j-1}, G_{j-1}) + K_j + W_j,$$

$$F_j = E_{j-1}, G_j = F_{j-1}, H_j = G_{j-1}.$$

i

$$\sum_0(W) = ROTR^2(W) \oplus ROTR^{13}(W) \oplus ROTR^{22}(W),$$

$$\sum_1(W) = ROTR^6(W) \oplus ROTR^{11}(W) \oplus ROTR^{25}(W).$$

Tada se j-ta transformacija, označena sa A_j definira na sljedeći način:

$$A_j = \sum_0(A_{j-1}) + F(A_{j-1}, B_{j-1}, C_{j-1}) + H_{j-1} + \sum_1(E_{j-1}) + G(E_{j-1}, F_{j-1}, G_{j-1}) + K_j + W_j,$$

Navedimo sada neke primjere SHA-256 algoritma:

Mary had a little lamb.

$$\Downarrow$$

$$d2fc16a1f51a653aa01964ef9c923336$$
$$e10653fec195f493458b3b21890e1b97$$

Mary had a little lamb

$$\Downarrow$$

$$efe473564cb63a7bf025dd691ef0ae0a$$
$$c906c03ab408375b9094e326c2ad9a76$$

Mary had a little lamb!

$$\Downarrow$$

$$7e2dbc1ca1859dabe1c1e9547ed4734d$$
$$56ef85ec87ae87ea3f63c4371cf4a79e$$

Primijetimo da "male" promjene, kao što su točka i uskličnik na kraju rečenice u ulazu, čine izlaz neprepoznatljivim, kao što je i bila namjera.

Bitno je razumjeti da u nekim slučajevima algoritam SHA-256 može interpretirati ulaznu vrijednost na dva različita načina: kao tekst i kao heksadecimalni broj. Izlazne vrijednosti su potpuno drugačije. Pokažimo neke primjere algoritma SHA-256 primijenjene na tekstualne nizove u odnosu na heksadecimalne brojeve.
Ako algoritam ulaznu vrijednost interpretira kao tekstualni niz dobivamo sljedeći primjer:

$$efe473564cb63a7bf025dd691ef0ae0a$$
$$c906c03ab408375b9094e326c2ad9a76$$

$$\Downarrow$$

$$f8c72ab0790e80e9191af68a37659e80$$
$$33c3de935f68be0df8ed04bfe35ac3c5$$

Međutim, ako taj isti ulaz algoritam interpretira kao heksadecimalni broj, dobivamo sljedeći izlaz:

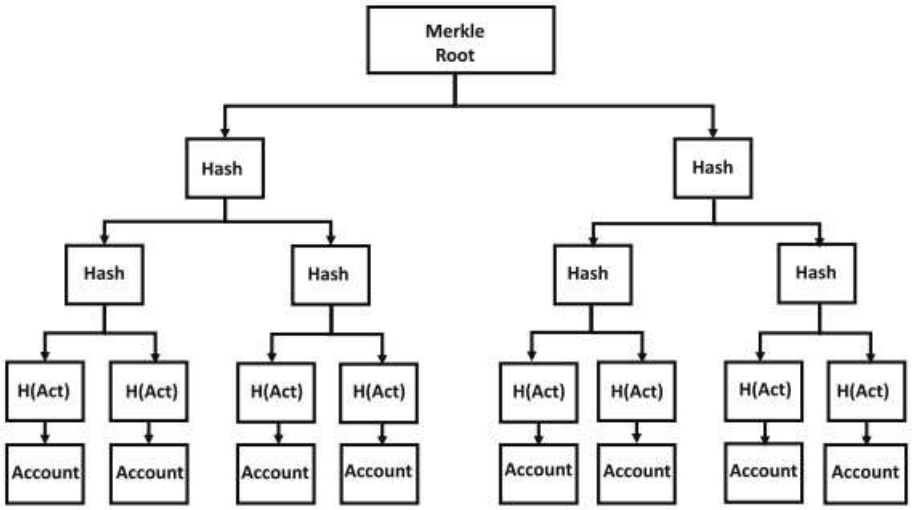
$$efe473564cb63a7bf025dd691ef0ae0a'$$
$$c906c03ab408375b9094e326c2ad9a76$$
$$\Downarrow$$
$$a34e5a180726e23e5ea2aea4b16ce101$$
$$99331eb897003e419c947619426e4da0$$

Primjećujemo da u Bitcoin protokolu i u većini drugih kripto protokola, koji pretežno rade s heksadecimalnim brojevima, a ne s nizovima, ulazne vrijednosti su uvijek hashirane dva puta:

$$x \to y = SHA_{256}(x) \to z = SHA_{256}(y).$$

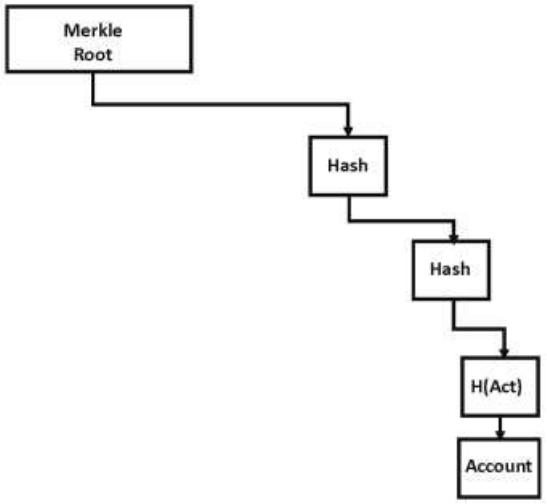
Drugo hashiranje koristi se kao sigurnosni mehanizam protiv takozvanog "length-extension" napada. *Lenght-extension* omogućuje napadaču koji zna duljinu poruke m_1 i njen $Hash(m_1)$ da izračuna hash $Hash(m_1||m_2)$ poruke m_1 konkatenirane s porukom m_2 koju je odabrao napadač, bez znanja m_1.

3.4 Merkle stabla i hash pokazivač



Slika 3.4: Merkle stablo

Ideja hash-a korisna je za izradu podataka koji su istovremeno nepromjenjivi i komprimirani. Pretpostavimo da imamo skup od N blokova podataka. Može ih nadopuniti tako da ukupan broj blokova bude višekratnik broja 2, na primjer 2^n. Zatim ih hashiramo sekvencijalno dok ne dođemo do korijenskog čvora. Ove strukture nazivaju se Merkle stabla (eng. *Merkle trees*) i vrlo su korisna u kriptografiji, a posebno i u blockchainu.



Slika 3.5: Provjera pripadnost određene transakcije skupu

Tipično Merkleovo stablo prikazano je na slici 3.4. Po strukturi, Merkle stabla su otporna na neovlaštene promjene, u smislu da će promjene u bilo kojem listu na putu do korijena promijeniti korijen stabla do neprepoznatljivosti. Ovo svojstvo je korisno ako, na primjer, želimo provjeriti da određena podatkovna točka, recimo transakcija, pripada određenom Merkleovom stablu. Korištenjem Merkleovih stabala, možemo potvrditi da je transakcija član nekog skupa u $O(ln(N))$ koraka. Ideja putanje provjere ilustrirana je na slici 3.5.

U tablici sa slike 3.6 koristimo hash-ove prvih sedam transakcija iz Bitcoin bloka i gradimo uravnoteženo stablo umjesto podstavljenog stabla kako bismo ubrzali proces. Drugim riječima, da bismo došli do korijena stabla, kombiniramo transakcije na sljedeći način:

- U prvih sedam redaka prikazano je prvih sedam transakcija
- U narednim retcima s brojevima $0, ..., k, 0 \leq k \leq 6$ označene su kombinacije transakcija. Na primjer, s brojem 01 označena je kombinacija transakcija 0 i 1 čime dobijemo njihov *"roditeljski"* čvor u stablu. Zatim, na primjer, s brojem 0123 označena je kombinacija *"roditeljskih"* čvorova kombinacija 0 i 1 te 2 i 3, itd...

- U drugom stupcu prikazujemo stvarni korijen hashiranja
- U trećem stupcu prikazujemo hipotetski korijen, izgrađen s blago modificiranim ulazima — za transakciju 0, zadnji heksadecimalni broj 3 zamijenjen je s 0. Tablica pokazuje da je dobiveni korijen potpuno drugačiji od ispravnog, što nam omogućuje da pronađemo slučajeve neovlaštenog mijenjanja ulaza.

Transakcije	Pravi hash	Modificrani hash
	93955d40d918d014903843d258aada5c	93955d40d918d014903843d258aada5c
0	720a5d37afac7889268f459a97b148a3	720a5d37afac7889268f459a97b148a0
	a8178a7223372414ac060b4bba4b33b8	a8178a7223372414ac060b4bba4b33b8
1	b4847a756fa76a715af7fd11bfd143d5	b4847a756fa76a715af7fd11bfd143d5
	efb3f60304532ebc80163b5f375fa8a9	efb3f60304532ebc80163b5f375fa8a9
2	4a39a8b0807b99703b6b646c1f7af5bf	4a39a8b0807b99703b6b646c1f7af5bf
	a070eda356c87a7af9bff22eab3b3c38	a070eda356c87a7af9bff22eab3b3c38
3	460605eb00938c84a86a1d6d3c608078	460605eb00938c84a86a1d6d3c608078
	b96b516295b8e4f5452405db8213ca56	b96b516295b8e4f5452405db8213ca56
4	cde630b7a30c2a400c4991b9a17f072a	cde630b7a30c2a400c4991b9a17f072a
	d8974a3a6596fbd86bd1f794a221639c	d8974a3a6596fbd86bd1f794a221639c
5	37d7633394bf03fddd61b48a1505f8b1	37d7633394bf03fddd61b48a1505f8b1
	d87a5a7ea8a8fb566d81605d1ef9ab11	d87a5a7ea8a8fb566d81605d1ef9ab11
6	32462f7812dab97e24b32d667aa1f959	32462f7812dab97e24b32d667aa1f959
	8b038b740bbb16a4e54593367f0bc48f	0f87ad6a849730930d00fc0c12d2af74
01	bde0c17518821f2efd8489daa0720849	a2ff901b27a7cd61efba01db3817b09d
	17efe128dcaede1600101af5ef2223d8	17efe128dcaede1600101af5ef2223d8
23	51578f06ab89725733fa2aac31e972d0	51578f06ab89725733fa2aac31e972d0
	9e27b512cb2ec01cc0eb68df6a683f67	9e27b512cb2ec01cc0eb68df6a683f67
45	4c6c2b6c50bdcf9a2ee3eaeb9575ea4d	4c6c2b6c50bdcf9a2ee3eaeb9575ea4d
	d87a5a7ea8a8fb566d81605d1ef9ab11	d87a5a7ea8a8fb566d81605d1ef9ab11
6	32462f7812dab97e24b32d667aa1f959	32462f7812dab97e24b32d667aa1f959
	b11872bc8ca21a33c52efd1fe1fbd161	3e40422b5f6080810a187564b6e67da2
0123	8203cb181c52db3f22e0b70bb2c165ba	a2e612e36ccf6b9ee410d56dbc356eb3
	b50bffbc033369a859af8a757cf9adfe	b50bffbc033369a859af8a757cf9adfe
456	6334e2c28aedb912eadab99b8986f12e	6334e2c28aedb912eadab99b8986f12e
	1bfedd44486da71aa3d2da322108bbb2	06659e1edeb39dff2e1867f0d1f8d0e2
01234567	31d3f59077af88175ae065a0e5e64518	c91e18d66c5dceecbede07d9f67ca1f2

Slika 3.6: Niz hash pointera

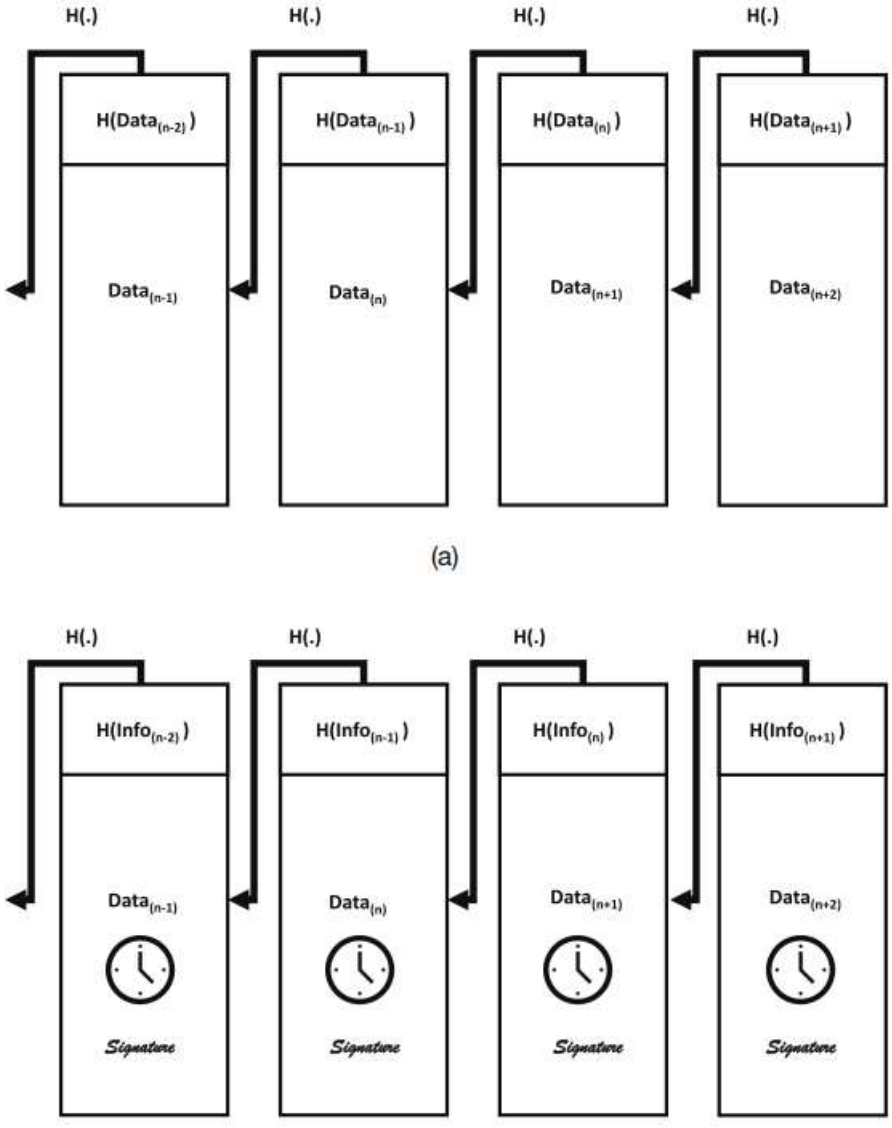
Hash pokazivači i blockchain

Hash pokazivači korisni su za ulančavanje blokova podataka čineći ih nepromjenjivima. Ne samo da povezuju prošlost i sadašnjost bloka, već omogućuju da je mijenjanje prošlosti nemoguće što je ilustrirano na slici 3.7.

S obzirom na prirodu hash funkcija, svaka promjena u prethodnim blokovima potpuno će promijeniti stari hash u novi, čime se uništavaju svi izračuni na temelju starog hasha. Naravno, možemo kombinirati hashiranje i digitalno potpisivanje za stvaranje lanaca blockchaina događaja koji su otporni na manipulacije i zlonamjerne izmjene. Potpisivanjem bloka podataka i hashiranjem para $(m, sign(h(m)))$ stvaramo hash pointer. Zauzvrat, ovaj pokazivač postaje sastavni dio sljedećeg bloka. Na taj način stvaramo blockchain prikazan na slici 3.7.

Po svojoj prirodi, kriptografija postaje prirodni mehanizam za održavanje cjelovitosti vođenja evidencija putem interneta. Ova činjenica kriptografiju čini temeljnim alatom za izradu financijskih i nefinancijskih aplikacija. Dok je simetrična kriptografija ključna za sigurno pohranjivanje podataka u bazama podataka i drugim sličnim aplikacijama, ona je sekundarna za obavljanje transakcija putem interneta. U isto vrijeme, asimetrična kriptografija je primarni instrument u tom pogledu.

Ranije spomenuti PKI, koji uključuje stvaranje parova javnih/tajnih ključeva, distribuciju i korištenje, pruža robustan okvir za izgradnju distribuiranih aplikacija, kao što su kriptovalute i blockchainovi. Za svrhe transakcija, digitalni potpis je glavni instrument. Hash funkcije su posebno korisne jer možemo učinkovito komprimirati poruke pomoću Merkleovog stabla hashova. S ovim alatima na dohvat ruke, lakše razumijemo kako izgraditi peer-to-peer digitalne valute, distribuirane knjige te u konačnici lance kriptiranih blokova, tj. blockchains.



Slika 3.7: Hash pokazivači

Bibliografija

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

[2] Atreccanmi A. Lipton, *Blockchain and Distributed Ledgers, Mathematics, technology, and economics*, World Scientific, 2022.

[3] Karame Ghassan O Wüst Karl i Ritzdorf Hubert Gervais, Arthur, *On the Security and Performance of Proof of Work Blockchains*, Bitcoin.org, 2017.

[4] H. Handschuh H. Gilbert, *Security Analysis of SHA-256 and Sisters*, Lecture Notes in Computer Science, Security Technologies Department (2004.).

Sažetak

Iako Blockchain i distribuirane knjige nisu novi koncepti, moderna tehnologija daje im novi život. Ona otvara nove mogućnosti sigurne bankarske i trgovinske aktivnosti uklanjanjem nepotrebnih posrednika te koristeći složene kriptografske alate. U današnjem svijetu, sigurnost podataka korisnika stavlja se u sve veći plan pa time dolazi do prepoznavanje potencijala ovakve digitalne revolucije. Stoga se Blockchain tehnologija nameće kao optimalno rješenje. Počevši od digitalnih valuta, Blockchain tehnologija ostvarila je značajan napredak u posljednjem desetljeću. S razvojem informatičkih tehnologija kriptovalute su postale sve važniji faktor te dovode u prvi plan sve mogućnosti blockchain tehnologije.

Dok se decentraliziranost i neregularnost sustava smatra najvećom prednosti, također vuče za sobom olakšanu manipulaciju podacima unutar lanca gdje se javlja potreba za pojačanom enkripcijom podataka te algoritmima konsenzusa radi očuvanja integriteta.

Omogućena dobrom podlogom od strane tehnologije, blockchain bi se, u budućnosti, u školstvu i drugim raznim institucijama mogao naći kao sastavni dio svakodnevice. Primjena blockchain tehnologije te realizacija direktnih umjesto centraliziranih transakcija može uvelike uvesti promjene u društveno-ekonomski svijet.

Summary

Although Blockchain and distributed ledgers are not new concepts, modern technology is giving them new life. It opens up new possibilities for secure banking and trading activity by removing unnecessary intermediaries and using complex cryptographic tools. In today's world, the security of users' data is increasingly important, so the potential of this digital revolution is being recognized. Therefore, Blockchain technology imposes as the optimal solution. Starting with digital currency, Blockchain technology has made significant progress in the last decade. With the development of IT technologies, cryptocurrencies have become an increasingly important factor and bring all the possibilities of blockchain technology to the fore.

While the system's decentralization and irregularity are considered the greatest advantage, it also entails easy data manipulation within the chain. Therefore, there arises the need for increased data encryption and consensus algorithms to preserve its integrity.

Enabled by a good foundation from technology, blockchain could be found as an integral part of everyday life in schools and other various institutions in the future. The application of blockchain technology and the realization of direct instead of centralized transactions can greatly introduce changes in the socio-economic world.

Životopis

Rođena sam 16. lipnja 1997. godine u Zadru. Svoje djetinjstvo provodim u Biogradu na Moru gdje sam pohađala vrtić, osnovnu te srednju školu. Još tijekom osnovne škole otkrivam svoje zanimanje za matematiku te još od nižih, pa sve do zadnjeg razreda srednje škole, sudjelujem u natjecanjima iz matematike.

Nakon svih godina natjecanja, jedini logični odabir za mene bio je upisati preddiplomski studij Matematike na Prirodoslovno–matematičkom fakultetu u Zagrebu. 2020. godine stječem titulu sveučilišne prvostupnice matematike te upisujem diplomski studij, smjer Računarstvo i Matematika. Na temelju stečenog znanja tijekom školovanja te programerskih vještina, na kraju pete godine dobivam dvomjesečno pripravništvo u Zagrebačkoj programerskoj tvrtki Memgraph te nakon uspješno odrađenog pripravništva dobivam i posao kod njih.