KRIPTOGRAFIJA V FINANCAH: KAKO MATEMATIKA ŠČITI BITCOIN?

NINA JANKOVIČ

Fakulteta za matematiko in fiziko
Univerza v Ljubljani

Bitcoin je decentraliziran digitalni denar, ki temelji na naprednih kriptografskih mehanizmih, kot so zgoščevalne funkcije in digitalni podpisi. S pomočjo teh mehanizmov se zagotavlja varnost in zanesljivost transakcij. Članek predstavi osnovne kriptografske temelje Bitcoina, vključno z uporabo funkcije SHA-256 za zaščito integritete podatkov in ECDSA za preverjanje lastništva. Prav tako preuči omrežno arhitekturo Bitcoina, ki temelji na sistemu P2P, in postopek rudarjenja, ki zagotavlja potrjevanje transakcij ter varnost celotnega omrežja. Članek prav tako opiše igro rudarjenja, ki jo rudarji uporabljajo v tem konkurenčnem okolju.

CRYPTOGRAPHY IN FINANCE: HOW DOES MATHEMATICS SECURE BITCOIN?

Bitcoin is a decentralized digital currency based on advanced cryptographic mechanisms such as hash functions and digital signatures. These mechanisms ensure the security and reliability of transactions. This article presents the basic cryptographic foundations of Bitcoin, including the use of the SHA-256 function to protect data integrity and ECDSA for verifying ownership. It also examines Bitcoin's network architecture, which operates on a peer-to-peer (P2P) system, and the mining process, which ensures transaction validation and overall network security. Additionally, the paper discusses the mining game the miners play in this competitive environment.

1. Uvod

Čeprav so pred Bitcoinom obstajali različni poskusi digitalnega denarja, je prav Bitcoin prva uspešna decentralizirana digitalna valuta, ki temelji na veriženju blokov (angl. *blockchain*) — na nespremenljivi 'knjigi transakcij'.

Bitcoin je bil prvič opisan leta 2008 v članku z naslovom 'Bitcoin: A Peer-to-Peer Electronic Cash System', ki ga je podpisal anonimnež Satoshi Nakamoto. Nakamoto je združil več predhodnih izumov, kot je Hashcash, da je ustvaril popolnoma decentraliziran elektronski plačilni sistem, ki ne potrebuje centralne oblasti za izdajanje valute ali potrjevanje transakcij. Ključna inovacija je bila uporaba distribuiranega računalniškega sistema ('dokaz o delu', angl. *proof-of-work*) za izvedbo globalne 'loterije' približno vsakih 10 minut, ki omogoča decentraliziranemu omrežju, da doseže soglasje o stanju transakcij. Ta pristop elegantno rešuje problem dvojne porabe, pri katerem bi lahko enotna valuta bila porabljena dvakrat. Pred tem je bil problem dvojne porabe slabost digitalnega denarja, ki se je reševal tako, da so vse transakcije potekale skozi centralizirano plačilno službo [2].

Omrežje Bitcoin je bilo vzpostavljeno leta 2009 na podlagi referenčne implementacije, ki jo je objavil Nakamoto in so jo kasneje večkrat spremenili drugi programerji. Število in moč naprav, ki izvajajo algoritem dokaza o delu (rudarjenje), ki zagotavlja varnost in odpornost omrežja Bitcoin, sta eksponentno narasla, tako da danes njihova združena računska moč presega skupno število računalniških operacij vseh najzmogljivejših superračunalnikov na svetu [2].

Satoshi Nakamoto se je umaknil iz javnosti aprila 2011, odgovornost za razvoj kode in omrežja pa je prepustil uspešni skupini prostovoljcev. Identiteta osebe ali oseb, ki stojijo za Bitcoinom, še vedno ni znana. Vendar pa niti Satoshi Nakamoto niti kdorkoli drug ne izvaja individualne kontrole nad sistemom Bitcoin, ki deluje na osnovi popolnoma preglednih matematičnih načel, odprte kode in soglasja med udeleženci. Sam izum je prelomnega pomena in je že sprožil novo znanost na področjih distribuiranega računalništva, ekonomije in ekonometrije [2].

Sčasoma je Bitcoin pritegnil pozornost številnih vlagateljev, razvijalcev in podjetij, ki so začeli raziskovati različne možnosti uporabe tehnologije veriženja blokov, kar je pripeljalo do širšega sprejetja in razvoja številnih drugih kriptovalut in decentraliziranih finančnih sistemov. Kljub številnim

izzivom, kot so nihanja cene, regulativni okviri in varnostni pomisleki, Bitcoin še vedno ostaja vodilna in najbolj prepoznavna kriptovaluta na svetu [13].

Veriženje blokov, ki omogoča, da so vsi podatki o transakcijah shranjeni v verigi blokov, predstavlja jedro Bitcoinovega sistema. Vsak blok vsebuje zbirko transakcij, ki so potrjene s strani omrežja, in s tem zagotavlja celovitost in nepreklicnost zgodovine transakcij [12]. Namesto, da bi se Bitcoin zanašal na centralizirane institucije, kot so banke, omogoča varne in preverljive transakcije s pomočjo matematičnih in kriptografskih metod. Tehnologija veriženja blokov je danes temelj številnih decentraliziranih finančnih sistemov, pri čemer kriptografija zagotavlja varnost in zanesljivost omrežja [14].

V tem članku bomo podrobneje preučili ključne kriptografske koncepte, ki omogočajo delovanje Bitcoina, vključno z uporabo zgoščevalnih funkcij (angl. *hash functions*), digitalnih podpisov (angl. *digital signatures*) in sistemov za omrežno potrjevanje transakcij. Prav tako bomo raziskali vlogo rudarjenja pri ohranjanju varnosti omrežja ter pregledali arhitekturo, ki podpira Bitcoinovo decentralizirano naravo. Z razumevanjem teh osnovnih principov si bralec lahko pridobi vpogled v kompleksnost in moč Bitcoina kot tehnološke platforme, ki je spremenila način, kako razmišljamo o denarju in transakcijah.

2. Osnovni koncepti

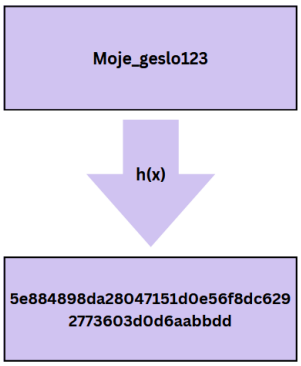
2.1 Kriptografski elementi

Za razumevanje ozadja delovanja Bitcoina je ključno, da najprej razumemo osnovne kriptografske koncepte.

Definicija 1. [10] **Zgoščevalna funkcija** je funkcija $h : A \to B$, kjer je $A = \{a \in \{0,1\}^j \mid j \in \mathbb{N}\}$ množica vseh bitnih zaporedij poljubne dolžine in $B = \{0,1\}^k$ množica vseh bitnih zaporedij določene (običajno kratke) dolžine k. Vhodi v zgoščevalne funkcije se imenujejo sporočila (angl. *messages*), izhodi pa izvlečki (angl. *digests*).

Za lažjo predstavo: zgoščevalna funkcija poljubno dolg vhod zgosti (stisne) v krajši izhod fiksne dolžine.

Zgled 2. Zgoščevalne funkcije se pogosto uporabljajo za zagotavljanje celovitosti podatkov. Na primer, če želimo poslati sporočilo, kot je prikazano na sliki 1, lahko zgoščevalno funkcijo uporabimo za ustvarjanje kratkega izvlečka, ki predstavlja vsebino sporočila. Ta izvleček lahko nato uporabimo za preverjanje, če je bilo sporočilo med prenosom spremenjeno.



Slika 1. Prikaz uporabe zgoščevalne funkcije. Geslo ‘Moje_geslo123’ je zgoščeno v izvleček ‘5e884898da28047151d0e56f8dc6292773603d0d6aabbdd’.

Na zgoščevalne funkcije lahko torej gledamo kot na nekakšen algoritem, ki za vhod dobi poljubno sporočilo, kot izhod pa vrne fiksno dolgo vrednost. Imajo tri glavne lastnosti:

- vhod je niz poljubne dolžine,
- izhod je fiksne dolžine,
- računanje izvlečka je enostavno in hitro.

Kripotgrafska zgoščevalna funkcija je tip zgoščevalnih funkcij, ki je opremljena z dodatnimi varnostnimi lastnostmi. Le-teh je veliko, kar se pa Bitcoina tiče, so najpomembnejše lastnosti odpornost praslik, odpornost drugih praslik (šibka odpornost na trke) in odpornost na trke (krepka odpornost na trke).

Definicija 3 (Odpornost praslik). [9] Za dani izvleček y je računsko neizvedljivo[1] poiskati sporočilo x, da velja $h(x) = y$.

Odpornost praslik ščiti pred tem, da bi iz zgoščene vrednosti lahko rekonstruirali izvirno sporočilo. To je ključno za zaščito gesel, digitalnih podpisov in drugih aplikacij, kjer je pomembna enosmernost zgoščevalne funkcije.

Definicija 4 (Odpornost drugih praslik). [9] Za dano sporočilo x je računsko neizvedljivo poiskati drugo sporočilo x', da velja $h(x) = h(x')$.

Odpornost drugih praslik ščiti pred zamenjavo določenega sporočila z drugim, ki ima enak izvleček.

Zgled 5. Recimo, da imamo sporočilo

$$x = \text{‘Janezu Novaku izplačajte 1.000 EUR’.}$$

Zgoščevalna funkcija h ustvari izvleček

$$h(x) = \text{ad12bde650845336bde3}.$$

Napadalec poskuša najti drugo sporočilo x', na primer ‘Maji Horvat izplačajte 1.000 EUR’, ki ima enak izvleček kot x. Če uspe, lahko zamenja prvotno sporočilo s ponarejenim, ne da bi sistem zaznal spremembo. Odpornost drugih praslik takšen napad preprečuje.

Definicija 6 (Odpornost na trke). [9] Računsko neizvedljivo je poiskati dve različni sporočili x in x' z enakim izvlečkom.

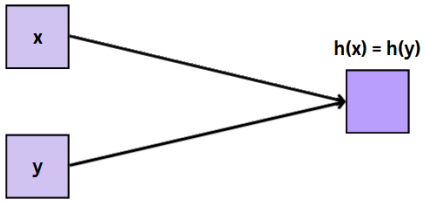
Slika 2 prikazuje vizualizacijo trka. Odpornost na trke ščiti pred tem, da bi med poljubnimi pari sporočil lahko našli trk. To je še posebej pomembno v sistemih, kjer je mogoče sporočila prosto izbirati (npr. digitalni podpisi, veriženje blokov).

Zgled 7. Napadalec išče kateri koli par različnih sporočil x in x', ki ima isti izvleček. Na primer, sporočilo $x =$ ‘Janezu Novaku izplačajte 1.000 EUR’ in sporočilo $x' =$ ‘Maji Horvat izplačajte 1.000 EUR’ bi lahko imela enak izvleček

$$h(x) = h(x') = \text{ad12bde650845336bde3}.$$

Če je napadalec uspešen, lahko ustvari dve različni sporočili z istim izvlečkom, kar omogoča zlorabo. Takšen napad preprečuje lastnost odpornost na trke.

[1] Izraz *računsko neizvedljivo* razložimo v nadaljevanju.



Slika 2. Trk je par različnih sporočil x in y z istim izvlečkom.

Pri vseh zgornjih definicijah se pojavi besedna zveza *računsko neizvedljivo* in ne *ne obstaja*. Za lažjo razlago pomembne razlike med izrazoma se osredotočimo na trke.

Intuitivno zagotovo vemo, da trki obstajajo; vhod je namreč poljubno dolg niz, izhod pa niz določene fiksne dolžine. Potemtakem drži, da je v zgoščevalni funkciji $h : A \to B$ množica A neskončna, medtem ko je množica B končna. Tako zagotovo obstaja trk. Izkaže se, da lahko za iskanje trka za zgoščevalno funkcijo z 256-bitnim izhodom (teoretično) uporabimo naslednjo metodo: izberimo $2^{256} + 1$ različnih sporočil in izračunamo njihove izvlečke. Ker je število možnih izvlečkov (2^{256}) manjše od števila sporočil, mora obstajati vsaj en par sporočil z istim izvlečkom. S to metodo zagotovo najdemo trk, vendar na zelo neučinkovit način, saj bi izračun $2^{256} + 1$ izvlečkov trajal zelo dolgo. V praksi lahko trk najdemo z veliko verjetnostjo že z mnogo manjšim številom sporočil. Na primer, če naključno izberemo $2^{130} + 1$ sporočil, je verjetnost, da obstaja vsaj en trk, 99,8 %. Kljub temu je tudi $2^{130} + 1$ izračunov izvlečkov praktično neizvedljivo. Za zgoščevalno funkcijo z 256-bitnim izvlečkom bi bilo potrebno v povprečju izračunati približno 2^{128} izvlečkov, kar bi trajalo več kot 10^{27} let, če bi računalnik izvajal 10.000 zgostitev na sekundo [4].

Med lastnostmi kriptografske zgoščevalne funkcije velja naslednja povezava.

Trditev 8. *Naj bo $h : A \to B$ zgoščevalna funkcija. Potem odpornost drugih praslik sledi iz odpornosti na trke.*

Naslednji kripotgrafski gradnik Bitcoina je **digitalni podpis**. To je digitalni analog fizičnega lastnoročnega podpisa. Ima štiri glavne lastnosti:

- digitalni podpis dokazuje, da je podpisnik zares podpisal dokument,
- vsebine digitalno podpisanega dokumenta ni mogoče spreminjati,
- podpisa ni mogoče kopirati in ponarejati,
- podpisnik kasneje ne more zanikati, da je podpisal dokument.

Definicija 9. [9] Sistem za digitalno podpisovanje je sestavljen iz treh verjetnostnih polinomsko časovnih algoritmov (Gen, Sig, Ver), ki imajo naslednje lastnosti.

1. Algoritem za generiranje ključev Gen kot vhod prejme varnostni parameter 1^n in vrne par ključev *(pk, sk)*, imenovana javni ključ (angl. *public key*) in zasebni ključ (angl. *private key*). Predpostavimo, da imata *pk* in *sk* dolžino vsaj n, kar zagotavlja, da so ključi dovolj veliki za zahtevano varnostno raven. Poleg tega naj bo n mogoče določiti iz *pk* ali *sk*, kar omogoča enostavno preverjanje veljavnosti ključev.

2. Algoritem za podpisovanje Sig kot vhod vzame zasebni ključ *sk* in sporočilo m iz nekega prostora sporočil (ki je lahko odvisen od *pk*). Kot izhod vrne podpis σ, kar zapišemo kot $\sigma \leftarrow \mathrm{Sig}_{sk}(m)$.

3. Deterministični algoritem za preverjanje Ver vzame kot vhodni podatek javni ključ *pk*, sporočilo m in podpis σ. Kot izhod vrne bit b, kjer $b = 1$ pomeni veljaven podpis in $b = 0$ pomeni neveljaven podpis. To zapišemo kot $b = \mathrm{Ver}_{pk}(m, \sigma)$.

Zahtevamo, da za vsako sporočilo m velja:

$$\mathrm{Ver}_{pk}(m, \mathrm{Sig}_{sk}(m)) = 1.$$

Če obstaja taka funkcija ℓ, da je za vsak *(pk, sk)*, ki ga vrne $\mathrm{Gen}(1^n)$, prostor sporočil enak $\{0,1\}^{\ell(n)}$, potem rečemo, da je (Gen, Sig, Ver) sistem za digitalno podpisovanje za sporočila dolžine $\ell(n)$.

Opomba: V praksi pogosto vključimo zgoščevalno funkcijo v sistem, da omogočimo podpisovanje poljubno dolgih sporočil $m \in \{0,1\}^*$. V tem primeru se sporočilo najprej zgosti v izvleček fiksne dolžine, nato pa se podpiše izvleček.

Rečemo, da je podpis σ veljaven, če velja $\mathrm{Ver}_{pk}(m, \sigma) = 1$.

Zgled 10. [9] Sistem za digitalno podpisovanje se uporablja na naslednji način. Pošiljatelj S zažene algoritem $\mathrm{Gen}(1^n)$, da pridobi ključa *(pk, sk)*. Javni ključ *pk* je nato objavljen kot lastnina S; S ga lahko objavi na svoji spletni strani ali ga umesti v nek javni imenik. Tu predpostavimo, da lahko kdor koli pridobi legitimno kopijo S-ovega javnega ključa. Ko želi S potrditi avtentičnost sporočila m, izračuna podpis $\sigma \leftarrow \mathrm{Sig}_{sk}(m)$ in pošlje $(m,\ \sigma)$. Prejemnik, ki pozna *pk*, lahko z $(m,\ \sigma)$ preveri avtentičnost m tako, da preveri, če velja $\mathrm{Ver}_{pk}(m, \sigma) = 1$. Tako se ugotovi:

- S je poslal m,
- m ni bil spremenjen med pošiljanjem.

Česar nam pa tak sistem ne omogoča, je zagotovilo o tem, *kdaj* je bilo sporočilo m podpisano.

Ponaredek je sporočilo m skupaj z veljavnim podpisom σ, kjer m-ja ni podpisal S (za fiksni javni ključ *pk*, ki ga ustvari S). Sistem za digitalno podpisovanje je **varen**, če napadalec ne more izdati ponaredka, tudi če pridobi digitalni podpis za poljubna druga sporočila.

Pred formalno definicijo varnega sistema digitalnega podpisovanja definirajmo še zanemarljive funkcije *negl*.

Definicija 11. [9] Funkcija f, ki slika iz naravnih v nenegativna realna števila, je **zanemarljiva**, če za vsak pozitiven polinom p obstaja tak N, da za za vsa cela števila $n > N$ velja $f(n) < \frac{1}{p(n)}$.

Definicija 12. [9] Denimo, da je $\Pi = (\mathrm{Gen}, \mathrm{Sig}, \mathrm{Ver})$ sistem za digitalno podpisovanje. Naj ima napadalec $\mathcal{A}$ dostop do javnega ključa *pk*, ki je bil ustvarjen z $\mathrm{Gen}(1^n)$, in do oraklja $\mathrm{Sig}_{sk}(\cdot)$, pri čemer ne sme uporabiti oraklja na sporočilu m. Naj bo $\textit{Sig-forge}_{\mathcal{A},\Pi}(n)$ podpis, ki ga ponaredi napadalec $\mathcal{A}$. Sistem Π je **varen**, če za vse verjetnostne polinomsko časovne napadalce $\mathcal{A}$ obstaja taka zanemarljiva funkcija *negl*, da velja

$$P\left(\mathrm{Ver}(m, \text{Sig-forge}_{\mathcal{A},\Pi}(n)) = 1\right) \leq \mathit{negl}(n).$$

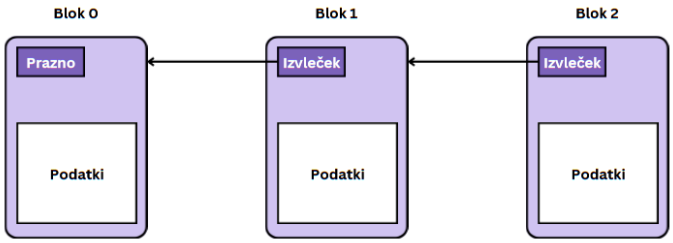
V tem kontekstu je **orakelj** nekakšna 'črna skrinjica', ki izvaja določeno operacijo ali funkcijo, ne da razkrije svoje notranje delovanje. Specifično v zgornjem primeru ima napadalec $\mathcal{A}$ dostop do oraklja $\mathrm{Sig}_{sk}(\cdot)$, ki mu omogoča, da pošlje poljubno sporočilo m in prejme podpis $\sigma \leftarrow \mathrm{Sig}_{sk}(m)$. Kljub temu, da napadalec lahko pridobi podpise za poljubna sporočila, orakelj ne razkrije zasebnega ključa ali notranjih podrobnosti o tem, kako je podpis ustvarjen.

Orakelj torej modelira situacijo, v kateri ima napadalec omejen dostop do sistema (npr. lahko pridobi podpise za poljubna sporočila), vendar ne more neposredno pridobiti zasebnih ključev ali drugih občutljivih informacij.

V nadaljevanju privzemimo, da obstaja varen sistem za digitalno podpisovanje. Natančneje, nemogoče je ustvariti veljaven digitalni podpis brez zasebnega kjuča podpisnika.

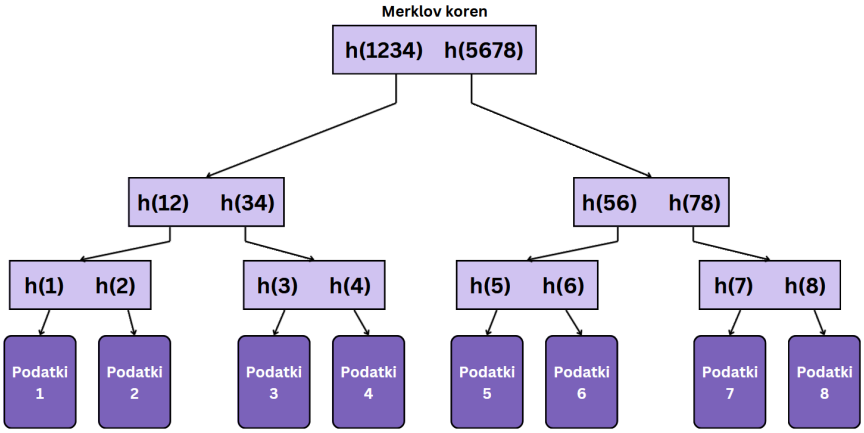
2.2 Podatkovne strukture

Pomembna podatkovna struktura, ki je temelj delovanja Bitcoina, je **zgoščevalni kazalec** (angl. *hash pointer*). Zgoščevalni kazalec vsebuje izvleček prejšnjega bloka in kazalec na ta blok [3]. Izvleček se izračuna na podlagi podatkov v prejšnjem bloku, kar zagotavlja integriteto verige blokov. Če se podatki v katerem koli bloku spremenijo, se spremeni tudi njegov izvleček, kar povzroči neskladje v verigi in razkrije poskus manipulacije. Prikaz zgoščevalnih kazalcev je na sliki 3.



Slika 3. Veriga blokov z zgoščevalnimi kazalci (črne puščice). Prvi blok v verigi (Blok 0) nima kazalca na prejšnji blok. Vsak naslednji blok vsebuje izvleček prejšnjega bloka, kar vzpostavlja povezavo med bloki in zagotavlja varnost verige.

Zgoščevalni kazalci so gradniki veriženja blokov. Pred razlago njihovega delovanja si oglejmo še **Merklovo drevo**, prikazano na sliki 4. Ta podatkovna struktura omogoča učinkovito organizacijo in preverjanje velikih količin podatkov. Gre za posebno vrsto binarnega drevesa, kjer vsak list predstavlja izvleček določenega nabora podatkov, njihovi predniki pa tvorijo izvlečke svojih otrok [1]. Koren drevesa, imenovan tudi Merklov koren, deluje kot nekakšen digitalni prstni odtis vseh transakcij v drevesu.



Slika 4. V Merklovem drevesu se podatkovni bloki združijo v pare, izvleček vsakega bloka se shrani v vozlišče starša. Starši se nato spet združijo v pare, njihovi izvlečki se shranijo v višjo raven drevesa. Postopek se ponavlja vse do korena drevesa, ki predstavlja končen izvleček celotne strukture.

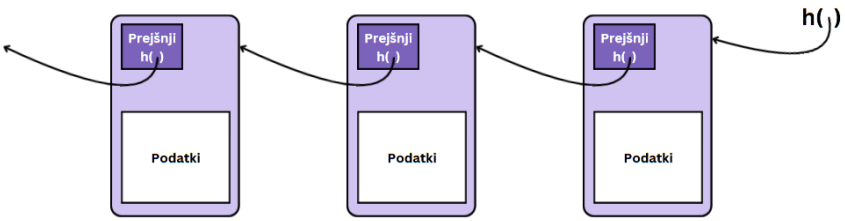
Ena ključnih prednosti Merklovega drevesa je omogočanje učinkovitega preverjanja posameznih transakcij. Namesto, da bi moral uporabnik preveriti celoten seznam transakcij v bloku, lahko s

pomočjo Merklovega dokaza (angl. *Merkle proof*) sledi poti od določene transakcije do Merklovega korena [1]. Če se izvleček, izračunan iz dokazne poti, ujema s korenom, lahko uporabnik potrdi, da je transakcija vsebovana v drevesu. To je bistveno bolj učinkovito, saj uporabniku ni treba prenesti in preveriti celotnega seznama transakcij, ampak le majhen del podatkov, kar zmanjša porabo virov in poveča hitrost preverjanja. Zaradi te lastnosti se Merklova drevesa pogosto uporabljajo v kriptografskih sistemih in pri veriženju blokov, saj omogočajo hitro in varno preverjanje podatkov.

Izkaže se še ena lepa lastnost Merklovih dreves, ki je dokazana v [9] (izrek 4.12):

Trditev 13. *Naj bo H zgoščevalna funkcija in M_t funkcija, ki t vrednostim $x_1, x_2, \ldots, x_t$ priredi Merklovo drevo. Če je H odporna na trke, je taka tudi M_t za poljuben t.*

Še zadnja podatkovna struktura, ki si jo moramo ogledati, je **bločna veriga** (angl. *block chain*)[2], predstavljena na sliki 5. Temelji na verigi blokov, kjer je vsak blok povezan s prejšnjim z zgoščevalnim kazalcem. Zelo podobna znana podatkovna struktura je povezani seznam. Z bločno verigo se razlikuje le v tipu kazalcev (povezani seznam vsebuje 'navadne' kazalce).



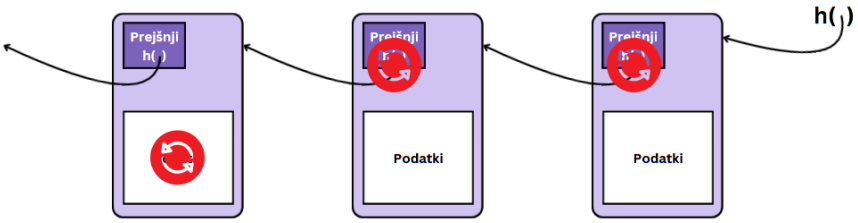
Slika 5. Povezani seznam, zgrajen z zgoščevalnimi kazalci (namesto kazalci) je podatkovna struktura, poznana pod imenom 'bločna veriga'. Sestavljena je torej iz zaporedja blokov, kjer vsak blok vsebuje podatke in zgoščevalne kazalce na prejšnji blok v seznamu. To pomeni, da vsak blok vsebuje vrednost prejšnjega bloka in tudi izvleček te vrednosti, s katero lahko preverimo, če se je vrednost spremenila.

Vsak blok vsebuje podatke (npr. transakcije) in izvleček prejšnjega bloka, kar ustvarja neprekinjeno verigo, ki je odporna na manipulacijo oziroma naknadno spremembo. Razlog za to lastnost so zgoščevalni kazalci v verigi. Če bi kdo poskusil spremeniti podatke v katerem koli bloku, kot je prikazano na sliki 6, bi se spremenil tudi izvleček teh podatkov. Ker vsak predhodni blok vsebuje izvleček prejšnjega bloka, bi bila veriga pokvarjena, saj se izvlečki ne bi več ujemali. Zaradi neposredne povezanosti sosednjih blokov bi moral napadalec ob spremembi vrednosti enega bloka spremeniti tudi vse predhodne bloke, da napadalca ne odkrijejo [4].

Seveda je to praktično nemogoče zaradi definicije 6, saj uporabljamo zgoščevalne funkcije brez trkov. To zagotavlja nespremenljivost in varnost bločne verige.

Bločna veriga se uporablja kot glavna podatkovna struktura v decentraliziranem sistemu, kar pomeni, da deluje brez posrednikov ali centralnega upravljanja. To omogoča, da vsi udeleženci v omrežju sodelujejo pri odločanju, kar zmanjša tveganje za vdor v podatkovno bazo ali zlorabo podatkov.

[2]Veriženje blokov je izraz, ki označuje *celoten koncept decentraliziranega sistema*, ki temelji na povezovanju blokov podatkov, medtem ko bločna veriga označuje *podatkovno strukturo*, ki hrani informacije v kronološkem zaporedju.



Slika 6. Napad na bločno verigo. Napadalec spremeni podatke v enem bloku, kar povzroči nov izvleček. Da ostane neopazen, ga mora popraviti. Posledično mora popraviti še vrednost predhodnega bloka vse do korena.

Glavne prednosti bločne verige so:

- nespremenljivost, kot je opisana zgoraj in na sliki 6,
- transparentnost: vsaka transakcija je vidna vsem udeležencem v omrežju, kar povečuje preglednost in zmanjšuje možnost goljufij,
- varnost: uporaba kriptografskih metod (zgoščevalne funkcije, zgoščevalni kazalci),
- hitrost in učinkovitost: transakcije so hitrejše in cenejše v primerjavi s tradicionalnimi sistemi, kot so bančne transakcije,
- sledljivost: transakcijam lahko preko blokov enostavno sledimo,
- poenostavitev: vse transakcije so zabeležene v eni sami 'knjigi' (angl. *ledger*), kar zmanjša zapletenost in nered [8].

Ta tehnologija se ne uporablja samo pri kriptovalutah, temveč tudi v različnih drugih panogah, kot so dobavne verige, zdravstvo in nepremičnine, kjer je pomembna transparentnost, varnost in sledljivost podatkov.

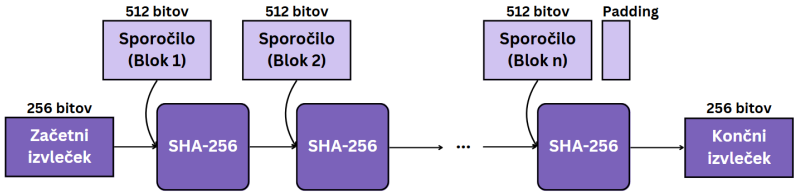
3. Bitcoin

Delovanje Bitcoina temelji na zgoraj opisanih kriptografskih konceptih in podatkovnih strukturah, ki skupaj zagotavljajo varnost in zanesljivost transakcij. V tem poglavju bomo podrobneje pogledali specifične kriptografske mehanizme, ki omogočajo delovanje Bitcoina, in vlogo rudarjenja pri potrjevanju blokov.

3.1 Kriptografski temelji Bitcoina

Ena ključnih komponent Bitcoina je zgoščevalna funkcija SHA-256 (Secure Hash Algorithm 256-bit) [12], ki se uporablja za pretvorbo podatkov v edinstven, fiksno dolg izvleček. SHA-256 sprejme vhodne podatke poljubne dolžine (npr. podatke o transakcijah) in jih s ponavljajočimi se kriptografskimi operacijami pretvori v 256-bitno vrednost. Ta izvleček deluje kot digitalni prstni odtis — vsaka najmanjša sprememba v vhodnih podatkih povzroči popolnoma drugačen rezultat. Zaradi tega je SHA-256 ključen za zagotavljanje celovitosti podatkov v omrežju Bitcoin. Poenostavljen postopek delovanja SHA-256 je prikazan na sliki 7.

Lastnosti kriptografskih zgoščevalnih funkcij, ki smo jih že omenili v poglavju 2.1, so ključne za varnost omrežja Bitcoin, saj preprečujejo napadalcem, da bi izvedli zlonamerna dejanja, kot so dvojna poraba sredstev ali spreminjanje zgodovine transakcij. Za SHA-256 velja, da je *računsko neizvedljivo najti* primere, ki bi kazali na kršitev teh lastnosti [5]. Odkritje takšne ranljivosti bi



Slika 7. Zgoščevalna funkcija SHA-256 deluje tako, da vhodno sporočilo (večje od 512 bitov) razdeli na 512-bitne bloke. Če je zadnji blok krajši od 512 bitov, se dopolni ('padding'). Vsak blok se nato obdela z zaporedjem matematičnih operacij, ki vključujejo logične funkcije, premike in dodajanje konstant. Po obdelavi vseh blokov se ustvari končni 256-bitni izvleček, ki je edinstven za dano vhodno sporočilo [6].

zahtevalo takojšnje ukrepe s strani Bitcoinove skupnosti. Verjetno bi sledil prehod na novo kriptografsko zgoščevalno funkcijo, ki bi zagotavljala zahtevano varnost.

Sistem za digitalno podpisovanje, ki ga uporablja Bitcoin, se imenuje algoritem za digitalne podpise z eliptičnimi krivuljami oz. ECDSA (angl. *Elliptic Curve Digital Signature Algorithm*). To je praktično digitalno podpisovanje, ki temelji na kriptografiji eliptičnih krivulj oz. ECC (angl. *Elliptic-curve cryptography*).

Kriptografija javnega ključa omogoča varno šifriranje, dešifriranje in digitalno podpisovanje, pri čemer so najbolj razširjene sheme zasnovane na dveh matematičnih problemih: faktorizaciji (npr. RSA) ter izračunu diskretnega logaritma v neki grupi. Slednje vključujejo tako uporabo multiplikativnih grup po modulu praštevila (npr. DSA) kot tudi uporabo grup na eliptičnih krivuljah[3] (ECC). ECC torej temelji na algebraični strukturi eliptičnih krivulj nad končnimi obsegi. Osnovana je na problemu diskretnega logaritma eliptične krivulje (angl. *Elliptic Curve Discrete Logarithm Problem*), ki je znan kot NP-težek problem [15]. Za podrobnejšo razlago ECC si lahko bralec ogleda [15] in poglavje 8.3.4 v [9].

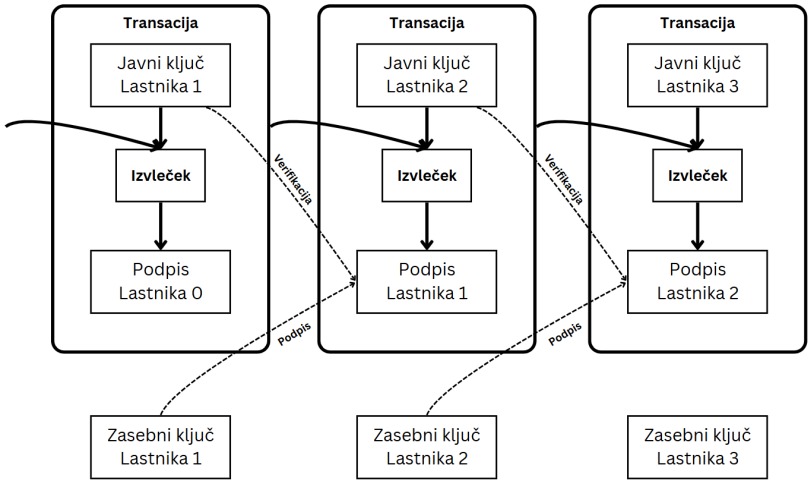
Kriptografija javnega ključa je sodobnejša od tradicionalnih metod in ponuja večjo varnost, saj uporablja daljše ključe in zahteva bistveno več računalniške moči za dešifriranje. Kljub visoki varnosti pa ta pristop zaradi svoje računske zahtevnosti ne more v celoti nadomestiti 'tradicionalne' kriptografije. Zato se v praksi uporablja predvsem za digitalne podpise in upravljanje ključev, medtem ko se za šifriranje večjih količin podatkov še vedno uporablja hitrejša 'tradicionalna' kriptografija [15].

3.2 Kovanec in omrežje Bitcoin

Podpoglavje je povzeto po [11].

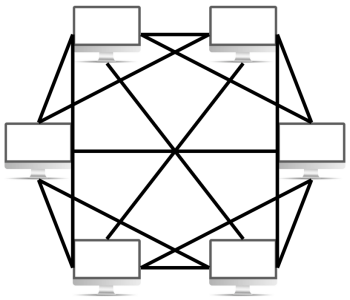
Elektronski kovanec Bitcoin definirajmo kot verigo digitalnih podpisov. Lastnik ga lahko prenese na naslednjega lastnika tako, da digitalno podpiše izvleček prejšnje transakcije in javni ključ prejemnika, nato pa te podatke doda na konec kovanca. Prejemnik preveri verodostojnost podpisa in s tem potrdi lastništvo. Prenos kovancev in njihovo verigo lahko vidimo na sliki 8.

[3]Eliptična krivulja je definirana z enačbo $y^2 + ay = x^3 + bx^2 + cx + d$.



Slika 8. Prikaz transakcij in digitalnih podpisov v sistemu elektronskih kovancev. Kovanec se najprej prenese od lastnika 1 do lastnika 2, kjer lastnik 1 podpiše izvleček prejšnje transakcije in javni ključ lastnika 2, kar se vključi na konec kovanca ('Podpis Lastnika 1'). Da lastnik 2 preveri, če je bil kovanec res v lasti lastnika 1, preprosto preveri veljavnost enačbe $\mathrm{Sig}_{sk_{\text{lastnik 1}}}(m) = \sigma_{\text{lastnik 1}}$, kjer je m 'Javni ključ Lastnika 2' in 'Izvleček' prejšnjega bloka.

Za zagotovitev časovnega zaporedja transakcij se vsak izvleček posameznega bloka podatkov javno objavi v omrežju Bitcoin. Vsak tak žig vključuje izvleček prejšnjega bloka, s čimer tvori veriženje blokov in preprečuje ponarejanje transakcij. Omrežje deluje na osnovi *peer-to-peer* oz. P2P (prikazani na sliki 9), kar pomeni, da ni centralne avtoritete, temveč vsa vozlišča enakovredno sodelujejo pri preverjanju in potrjevanju transakcij.



Slika 9. Omrežje P2P je sestavljeno iz povezanih vozlišč (računalnikov), ki si med seboj delijo informacije brez centralnega administrativnega sistema.

Za učinkovito shranjevanje in preverjanje transakcij Bitcoin uporablja Merklovo drevo, ki omogoča kompaktno predstavitev podatkov. V vsakem bloku so transakcije organizirane hierarhično, pri čemer se zgolj Merklov koren shrani v glavo bloka. To omogoča hitro preverjanje prisotnosti transakcije v bloku brez potrebe po shranjevanju celotne zgodovine transakcij, kar zmanjšuje potrebo po pomnilniškem prostoru.

3.3 Rudarjenje in potrjevanje transakcij

Omrežje Bitcoin za potrjevanje transakcij uporablja proces, imenovan rudarjenje (angl. *mining*). Rudarji tekmujejo pri reševanju matematičnega problema, ki zahteva iskanje določene naključne vrednosti (*nonce*), tako da bo izvleček celotnega bloka (izračuna s SHA-256) ustrezal kriterijem

težavnosti (tj. imel določeno število vodilnih ničel). Ta proces, imenovan dokazilo o delu (angl. *proof-of-work*, POW), zagotavlja varnost omrežja, saj je rešitev uganke izjemno zahtevna, preverjanje rešitve pa je enostavno [11, 12, 13].

Ko rudar najde veljaven blok, ga razpošlje celotnemu omrežju, kjer ga ostali udeleženci preverijo. Če je blok veljaven, se doda v verigo blokov in postane uradni del zgodovine transakcij. V praksi postopek poteka po sledečem postopku [11]:

1. Nove transakcije se pošljejo na vsa vozlišča v omrežju.
2. Vsako vozlišče nove transakcije zbere v blok.
3. Vsako vozlišče išče POW za svoj blok.
4. Ko vozlišče najde POW, pošlje blok vsem ostalim vozliščem.
5. Vozlišča ta blok sprejmejo le, če so vse transakcije v njem veljavne in še niso bile porabljene.
6. Vozlišča pokažejo, da so nov blok sprejela, tako, da ustvarjajo naslednji blok v verigi, pri čemer uporabljajo izvleček sprejetega bloka kot 'predhodni izvleček'.

Prvi rudar, ki je v tem postopku uspešen, je za svoje delo nagrajen v kovancih Bitcoin. Pri trenutni implementaciji Bitcoina ta nagrada izvira iz *ex-nihilo* ustvarjanja novih kovancev Bitcoin in provizij, ki jih lahko uporabniki Bitcoina dodajo svojim transakcijam. Verjetnost, da rudar reši matematični problem, je odvisna od njegove računske moči, zato se kompleksnost rudarjenja prilagaja skupni računski moči vseh rudarjev. Natančneje, kompleksnost se dinamično prilagaja tako, da rudarji najdejo POW za blok (in tako ustvarijo nove kovance Bitcoin) približno vsakih 10 minut. Ko je blok vstavljen v verigo blokov, se matematični problemi za rudarje spremenijo. Tako se tekma med rudarji ponovno začne [7].
Rudarjenje pri Bitcoinu opravlja dve ključni nalogi.

1. Rudarji ustvarjajo nove kovance Bitcoin v vsakem bloku, kar je podobno tiskanju denarja s strani centralne banke. Količina novih kovancev Bitcoin, ustvarjenih na blok, je fiksna in se postopoma zmanjšuje s časom.
2. Rudarjenje zagotavlja zaupanje v omrežje, saj potrdi transakcije le, če je bilo dovolj računske moči vloženo v blok, ki jih vsebuje. Več blokov pomeni več računske moči, kar povečuje zaupanje v celoten sistem [2].

3.3.1 Igra rudarjenja

Na kratko opišimo poenostavljeno igro rudarjenja, ki temelji na dinamiki, opisani v članku [7].

Primer 14 (Igra rudarjenja). Recimo, da je $N = \{1, \ldots, n\}$ množica rudarjev v omrežju Bitcoin, kjer velja $n \geq 2$. Vsak rudar $i \in N$ ima računsko moč $h_i > 0$, kjer velja $\sum_{j \in N} h_j = 1$. Rudarji med seboj tekmujejo, kdo prvi najde rešitev za matematični problem, ki ga je potrebno rešiti za ustvarjanje novega bloka[4]. Rudar lahko poljubno izbira množico transakcij, ki jih vključi v blok. Zanje mora poiskati rešitev, ki jo deli s preostankom omrežja Bitcoin. Le-ta se mora z rudarjevo rešitvijo strinjati. Čas, potreben za dosego soglasja za blok, naj bo $\tau(Q) = zQ$, kjer je Q velikost bloka (torej število transakcij v njem) in $z > 0$ konstanta. Nagrada za prvega rudarja, ki najde rešitev, je sestavljena iz dveh delov:

[4]Verjetnost, da posamezen rudar reši matematični problem, je odvisna od njegove računske moči, medtem ko je zahtevnost rudarjenja prilagojena skupni računski moči vseh rudarjev. Natančneje, zahtevnost se dinamično prilagaja tako, da do rešitve bloka (in s tem do ustvarjanja kovancev Bitcoinov) v povprečju pride vsakih 10 minut.

- fiksni del $R \geq 0$, ki predstavlja nagrado za ustvarjanje novega bloka,
- spremenljivi del ρQ, kjer je $\rho \geq 0$ gostota provizij, ki jih rudar prejme za vključene transakcije.

Reševanje matematičnega problema sledi strategiji 'poskusi in ugani', kjer rudarji v problem vnašajo različne vrednosti in preverjajo, ali so našli pravilno rešitev. Nastanek rešitve je mogoče modelirati kot slučajno spremenljivko, ki sledi homogenemu Poissonovemu procesu. Tu za parameter vzamemo $\lambda = \frac{1}{600}$[5].

Cilj vsakega rudarja je seveda najti rešitev za matematični problem, ki omogoči ustvarjanje novega bloka v Bitcoin bločni verigi, in s tem zaslužiti nagrado. Glavni cilj te naloge na splošno pa je razumeti, kako se rudarji obnašajo v konkurenci, ko iščejo rešitev za matematični problem, ob tem pa želijo maksimirati svoje dobičke.

Za natančnejšo analizo tega problema in iskanje Nashevega ravnoveja v igri rudarjenja si lahko bralci ogledajo članek [7], kjer je podrobno obravnavana strategija rudarjev v omrežju Bitcoin, vključno s tem, kako se iskanje rešitve za matematični problem modelira kot Poissonov proces. V članku so raziskane tudi simulacije, ki kažejo, da trenutni rudarji v omrežju Bitcoin ne sledijo strategiji v Nashevem ravnovesju, kar lahko vodi v podoptimalne odločitve in neravnovesje v omrežju.

4. Zaključek

Bitcoin je brez dvoma ena najbolj prelomnih inovacij v svetu digitalnih financ, katere varnost temelji na trdnih kriptografskih principih. Široka uporaba funkcij, kot sta SHA-256 za zgoščevanje podatkov in ECDSA za podpisovanje transakcij, omogoča nemoteno delovanje decentraliziranega sistema, kjer ni centralne avtoritete. Proces rudarjenja, ki vključuje dokazilo o delu, je ključnega pomena za zagotavljanje zanesljivosti in varnosti transakcij, saj rudarji tekmujejo v reševanju kompleksnih matematičnih nalog. Kljub številnim prednostim pa Bitcoin kljub vsemu ni brez izzivov, kot so visoki stroški rudarjenja in povečanje energetske porabe. Razumevanje teh mehanizmov in njihovega vpliva na delovanje Bitcoina je ključno za napovedovanje prihodnjega razvoja tega finančnega sistema.

LITERATURA

[1] 0xIchigo. Cryptographic tools 101 - Hash functions and Merkle trees explained, 2023. Dostopala 3. 2. 2025. Dostopno na `https://www.helius.dev/blog/cryptographic-tools-101-hash-functions-and-merkle-trees-explained`.

[2] Andreas M. Antonopoulos. *Mastering Bitcoin: Unlocking Digital Cryptocurrencies*. O'Reilly Media, 1nd edition, 2014.

[3] Educative. What are hash pointers in blockchain?, 2024. Dostopala 1. 2. 2025. Dostopno na `https://www.educative.io/answers/what-are-hash-pointers-in-blockchain`.

[4] Arvind Narayanan et al. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, 2016. Predobjavni osnutek knjige je dostopen na `https://d28rh4a8wq0iu5.cloudfront.net/bitcointech/readings/princeton_bitcoin_book.pdf`. Končna objavljena različica je strokovno recenzirana, lektorirana in oblikovana.

[5] Christoph Dobraunig et al. Analysis of SHA-512/224 and SHA-512/256. *Advances in Cryptology - ASIACRYPT 2015*, strani: 612–630, 12 2015.

[6] Thi Hong Tran et al. A high-performance multimem SHA-256 accelerator for society 5.0. *IEEE Access*, PP:1–1, 03 2021.

[7] Nicolas Houy. The Bitcoin mining game. *Ledger*, 1:53–68, 12 2016.

[8] Julija Strebko in Andrejs Romanovs. The advantages and disadvantages of the blockchain technology. *Proceedings of the 2018 IEEE 6th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*, strani: 1–6, 11 2018.

[5] To pomeni, da število novih blokov v času t sledi Poissonovi porazdelitvi s parametrom λt, torej Poiss(λt). Pri dani vrednosti λ to pomeni, da je pričakovano število novih blokov na enoto časa (v našem primeru na sekundo) $\frac{1}{600}$, kar ustreza povprečnemu času 10 minut na blok.

[9] Jonathan Katz in Yehuda Lindell. *Introduction to Modern Cryptography*. Chapman and Hall/CRC, 2nd edition, 2014.

[10] Spencer Miller. MMH^*_{32} from scratch: an introduction to hash functions. University of Chicago REU Paper, 2020.

[11] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Dostopala 3. 2. 2025. Dostopno na `https://bitcoin.org/bitcoin.pdf`.

[12] Wikipedia. Bitcoin. Dostopala: 31. 1. 2025. Dostopno na `https://en.wikipedia.org/wiki/Bitcoin`.

[13] Wikipedia. Bitcoin protocol. Dostopala: 4. 2. 2025. Dostopno na `https://en.wikipedia.org/wiki/Bitcoin_protocol`.

[14] Wikipedia. SHA-2. Dostopala: 4. 2. 2025. Dostopno na `https://en.wikipedia.org/wiki/SHA-2`.

[15] Yuhan Yan. The overview of elliptic curve cryptography (ECC). *Journal of Physics: Conference Series*, 2386, 12 2022.