Analiza algoritama korištenih u sustavima kriptovaluta

Furko, Filip

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Filip Furko

Analiza algoritama korištenih u sustavima kriptovaluta

ZAVRŠNI RAD

Varaždin, 2021.

SVEUČILIŠTE U ZAGREBU

FAKULTET ORGANIZACIJE I INFORMATIKE

V A R A Ž D I N

Filip Furko

Matični broj: 0016139628

Studij: Primjena informacijske tehnologije u poslovanju

Analiza algoritama korištenih u sustavima kriptovaluta

ZAVRŠNI RAD

Mentor:

Izv. prof. dr. sc. Sandro Gerić

Varaždin, rujan 2021.

Filip Furko

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

U ovom završnom radu bit će obrađeni algoritmi i tehnologije korištene kod sustava plaćanja temeljenih na kriptovalutama. Glavni orijentir razumijevanja cijelog sustava će biti prva i najpoznatija kriptovaluta, Bitcoin, te će se usporedbom s drugim sustavima komparirati način rada, složenost, ali i prednosti i nedostaci istih. Kako bi razumjeli na koji se način provode transakcije između dva korisnika mreže kriptovaluta, važno se upoznati s nekim osnovnim pojmovima koje opisuju tu mrežu. Na početku rada bit će spomenuta kratka povijest nastanka kriptovaluta, zatim će biti kratko obrađene različite vrste kriptovaluta, što je i kako se kreira elektronički potpis, gdje se čuvaju kriptovalute i kako se u konačnici primjenjuju u svakidašnjem životu. Kao glavni fokus rad će detaljnije proći kroz najpoznatije kriptografske algoritme, kako funkcionira rudarenje ili kopanje odnosno potvrda transakcija kod različitih sustava kriptovaluta te na koji način se transakcije povezuju u jednu cjelinu odnosno lanac vrijednosti. Na samome kraju rada algoritmi će biti komparativno uspoređeni gdje će biti opisana neka od njihovih svojstava.

Ključne riječi: kriptovalute, algoritmi, kriptiranje, složenost, rudarenje kriptovaluta, transakcije, lanac vrijednosti

Sadržaj

1. Uvod

U posljednja dva desetljeća, izazvano pojavom i širenjem Interneta, kao i tehnologije, dogodilo se veliko tehnološko širenje koje za sobom povlači i tehnološki napredak. Činjenica današnjice je da se Internet uključuje u sva područja ljudskih aktivnosti, od komunikacije i zabave, učenja i trgovine, pa čak i u izvršavanje poslovnim obaveza. Ta činjenica dodatno je poduprta zadnjih dvije godine teškim globalnim stanjem gdje su svi stanovnici bili prisiljeni obavljati gotovo sve obaveze od kuće. Tehnološki napredak također je utjecao i na financijski sustav gdje se način skladištenja sredstava i plaćanja roba i usluga vrlo modernizirao i naravno promijenio. Dugi niz godina tragalo se za kvalitetnom implementacijom digitalnih valuta koje bi olakšale neke životne procese. Primjena tih valuta nažalost zahtijeva određeno učenje, prilagodbe i promjene postojećih navika i načina na koji promatramo valute, stoga je njihov razvoj i prihvaćanje otežano. Taj problem riješen je s uvođenjem danas općepoznatog pojma, kriptovalute, čija primjena i osobine nisu poznate svima, što je i jedan od razloga pisanja ovoga završnog rada. U ovome radu bit će detaljnije opisano što su to kriptovalute te kako su one nastale, gdje se one čuvaju i kako se s njima koristi. Nakon toga, bit će spomenuta njihova primjena sada i u budućnosti, te će zatim glavni fokus biti na tehnološkoj osnovici samih kriptovaluta. U analizi tehnoloških osnovica bit će pokazano kako i na koji način funkcioniraju najpoznatiji kriptografski algoritmi i hash funkcije te kako se koristi elektronički potpis. U nastavku će se detaljnije opisati procesi rudarenja kriptovaluta te uz pomoć kojih parametara i algoritama taj proces uopće funkcionira. Nakon toga bit će opisan lanac vrijednosti, što i zašto on predstavlja osnovicu samog pojma kriptovaluta. Na samom kraju rada bit će komparativno uspoređeni neki od najkorištenijih algoritama koji stvaraju konsenzus između sudionika u lancu vrijednosti i tako pokreću cijelu mrežu kriptovaluta.

2. Kriptovalute

Pojava digitalne valute i nova sigurnosna otkrića današnjice usko su povezana s razvojem područja kriptografije koje predstavlja osnovne izazove vezane uz upotrebu bitova za predstavljanje vrijednosti robe i usluga koje se mogu razmjenjivati. S obzirom na to da digitalni novac nije u stvarnom fizičkom obliku pred nama, najvažnija pitanja koja se postavljaju su: "Može li se vjerovati da je digitalna valuta stvarna, a ne krivotvorena?" te "Može li se biti siguran da nitko drugi ne može tvrditi da novac pripada njima umjesto vama?". Izdavatelji papirnatih novčanica riješili su problem krivotvorenja koristeći naprednu tehnologiju ispisa. Tradicionalne valute također se nalaze i u fizičkom i u digitalnom obliku što znači da slučaju krivotvorenja plaćanje nadzire središnje tijelo. Kod digitalnih valuta, kriptografija je ta koja pruža osnovu za povjerenje vezanu za legitimnost korisnika koji tvrde da njihova valuta ima određenu vrijednost. Kriptirani elektronički potpis omogućuje korisnicima da potpišu svoju digitalnu imovinu ili transakcije kako bi potvrdili vlasništvo nad svojim sredstvima, te se uz odgovarajuću arhitekturu, elektronički potpis također se može koristiti za rješavanje problema dvostrukog trošenja koji je česta pojava u svijetu kriptovaluta. (Antonopoulos, 2016.)

Budući da se kriptografija široko koristila osamdesetih godina, mnogi su istraživači pokušali koristiti kriptografiju za stvaranje digitalnih valuta. Prvi pokušaji razvoja digitalnih valuta i izdavanja digitalnih valuta zabilježeni su u povijesti te će u nastavku ovog poglavlja oni biti detaljnije obrađeni.

2.1. Povijest kriptovaluta

Kao što je već rečeno, povijest kriptovaluta započela je djelom Davida Chauma 90-ih godina u kojem je prvi puta spomenuo tehnologiju koja se zvala „slijepi potpis“. U tom je radu Chaum predložio novu shemu šifriranja sa svrhom sakrivanja sadržaj poruke prije samog potpisivanja, što je rezultiralo time da niti potpisnik ne može odrediti sadržaj koji potpisuje. Taj slijepi potpis mogao se javno verificirati kao i svaki drugi potpis. Ova inovativna ideja koju je Chaum predložio po prvi je puta omogućio svim korisnicima anonimnu upotrebu digitalne valute bez mogućnosti praćenja ili otkrivanja njihovog identiteta. U kasnijim radovima, Chaum unaprjeđuje ovu ideju omogućujući novu funkcionalnost provođenja izvan-mrežnih transakcija. Negativna strana ovog sustava je bila ta da je sustav još uvijek zahtijevao povjerljivog posrednika za izdavanje i korištenje digitalne valute. Kako bi ovu ideju komercijalizirao, Chaum je 1990. predstavio svoju ideju „DigiCash“ koja je zapravo započela prvu generacija kriptovaluta, ali ona nažalost nije uspjela u tome trenutku doći do šire

publike. Nakon toga, glavni je cilj postao izgradnja anonimnog sustava koji koristi digitalni novac sa svrhom smanjenja korupcije te organiziranog kriminala namijenjen vladama i bankama. Dodatno, DigiCash je predstavio veliku inovaciju u svijetu koja se očitovala u obliku mogućnosti beskontaktnog plaćanja cestarine, što se smatrala i njegova prva primjena. Ova ideja privukla je nekoliko banaka, vladu Nizozemske te poznate kompanije kao što je Visa i Microsoft. Usprkos svemu, no i ova je ideja doživjela neuspjeh. (Halaburda, Sarvary, 2016 ; Judmayer, Stifter, Krombholz, Weippl, 2017.)

Uslijed velikih iskoraka Davida Chuma, osniva se nazvan „cyperpunk“ čiji je glavni cilj bio pravedno korištenje tehnologije i kriptografije koja podupire privatnost, te će neke od ideja koje su bile inspirirane ovom idejom biti navedene u nastavku poglavlja.

Američka kompanija Citibank je 1999. godine pokušala razviti sustav elektroničkog novca za vlastite potrebe, pod nazivom „Elektronički monetarni sustavi“ ili kraće, e-cash. Ovo je bio drugi pokušaj komercijalizacije razvoja decentralizirane digitalne valute, odnosno prvi pravi pokušaj nakon DigiCash-a. Ova digitalna valuta imala je iznimno neobične karakteristike, odnosno, nakon nekog vremena novac bi nestao te bi osoba koja posjeduje novac morala otići do banke i zamijeniti ga. Tim postupkom htjelo se izbjeći pranje novca. Testni pokušaji izveli su se i 1997. kao i 2001. godine, ali ne s puno uspjeha. (Halaburda, Sarvary, 2016 ; Lee, 2015.)

Na temelju svih do tada napisanih radova kao i neuspjeha, Hal Finney 2004. godine pokreće prvi valutni sustav temeljen na dokazu o radu za višekratnu uporabu (eng. *Reusable Proof of work*) ili kraće RPOW. Njegova ideja bila je stvoriti novac u obliku žetona čija će se vrijednost kretati kao i zlato. Zanimljiva je činjenica da nakon što je Satoshi Nakamoto 2009. godine pokrenuo svoj projekt Bitcoin baš je Hal Finney bio prvi korisnik koji je dobio jedan Bitcoin. (Judmayer i sur, 2017.)

2005. Nick Szabo osmislio je novu digitalnu valutu pod nazivom Bit Gold. Njegov se sustav također oslanjao na kriptografske slagalice, što je slučaj i danas, koje kada bi bile riješene, bile bi poslane u javni registar te bi rezultat bio dodijeljen javnom ključu čvora koji je riješio slagalicu. Na ovakav način, riješen je problem mrežnog slaganja odnosno ostvarivanja konsenzusa vezan za jedinice valute koje se puštaju u opticaj. Korištenjem distribuiranog registra te dokaza o radu stvoren je sustav vrlo sličan današnjem Bitcoin-u koji je dolaskom do rješenja u opticaj puštao nove jedinice valute. Jedina očita razlika je bila ta da nije bila poznata niti određena maksimalna količina valutnih jedinica baš kao ni koliko se brzo te nove jedinice stvaraju. Taj problem stvorio je i pitanje o postojanju super-računala koji bi srušio tržište i cijenu ove digitalne valute, stoga niti ovaj projekt nije doživio široku primjenu. (Halaburda, Sarvary, 2016 ; Judmayer i sur., 2017.)

Iako su ovo bile samo ideje i teorijska razmatranja koja nikada nisu u potpunosti dovršena, teško je reći koliko bi bile učinkovita u svakodnevnoj praksi, što nažalost potvrđuje i činjenica da nikada nisu privukle veliku medijsku pažnju i primjenu od strane šire javnosti. Usprkos tome, Bitcoin je, kao trenutno najpoznatija kriptovaluta, preuzeo neke elemente gore navedenih sustava te ih spojio u nešto novo, funkcionalno i primjenjivo.

2.2. Različite kriptovalute

Nakon obrađene povijesti vezane za nastajanje i razvoj kriptovaluta, potrebno je pobliže proučiti kako se razlikuju same kriptovalute, te ih međusobno usporediti prema različitim pokazateljima.

Naglim i svakodnevnim razvojem tehnologije, ljudi sve više i više prihvaćaju tehnologiju kao normalan dio svakodnevice, pa tako traže i tehnološka rješenja za moderne probleme zastarjelosti fiducijarnog novca. Kao što je već spomenuto, prvi takav iskorak napravio je David Chaum u svome djelu „eCash“ 90-ih godina prošloga stoljeća što je kasnije Satoshi Nakamoto iskoristio kako bi svo svoje znanje o kriptovalutama učinio javno dostupnim, odnosno eng. *open source*, što je i razvilo generalnu ideju o rasprostranjenom korištenju kriptovaluta. Sukladno tome danas postoji više od devet tisuća različitih kriptovaluta koje se svakodnevno koriste, troše i kupuju na različitim razvijenim platformama specijalno definiranim za tu svrhu. Ti novi novčići kriptovaluta nastali iz Nakamotovog Bitcoina nazivaju se alternativnim kriptovalutama eng. *altcoins* što bi predstavljalo sve kriptovalute koji nisu Bitcoin. Izraz „novčić“ u nastavku rada predstavljat će imaginarnu mjernu jedinicu jedne kriptovalute, te je važno spomenuti da se u literaturi može još pronaći i izraz token (eng. *token*). Alternativne kriptovalute dijele neka obilježja s Bitcoinom, ali se i razlikuju od njega na druge načine. Na primjer, neke kriptovalute koriste drugačije mehanizme konsenzusa (eng. *consensus algorithms*) za stvaranje blokova u mreži ili validiranje odnosno potvrđivanje transakcija. Uz to, kriptovalute se mogu razlikovati na način da pružaju nove ili dodatne mogućnosti korisniku, poput stvaranja pametnih ugovora kao što je slučaj kod Ethereuma ili provođenje brzih i jednostavnih transakcija kao što je slučaj kod Ripple sustava. Zanimljiva činjenica je da su prema CoinMarketCap-u, još jednoj platformi za trgovanje u kripto svijetu, alternativne kriptovalute činile preko 40% ukupnog tržišta kriptovalutama u ožujku 2021. godine. (Coinmarketcap.com, 2021.) Uzevši u obzir činjenicu da ostale kriptovalute ne dostižu niti približnu vrijednost po jednom novčiću kao što je kod Bitcoina, definitivno im treba se dati na važnosti i pratiti njihove promjene. Tako na primjer treba spomenuti kriptovalutu pod imenom „SHIBA INU“ koja je u travnju 2021. godine imala veliku eksploziju u popularnosti i vrijednosti nakon velike medijske pažnje, a tada je njegova

vrijednost jednog novčića iznosila manje od 0.00000001 američkih dolara. Time se može zaključiti da se različite kriptovalute mogu razlikovati po njihovoj vrijednosti, a sama promjena vrijednosti nije proporcionalna s cijenom, već s potražnjom, te u nekim slučajevima ovise i o ponudi svjetske zalihe, koja predstavlja još jedan od različitih faktora kriptovaluta. (Frankenfield, 2021.)

Nadalje, različite kriptovalute se mogu razlikovati prema algoritmima za kriptiranje koji su u procesu transakcije zaslužni za „sakrivanje“ odnosno kriptiranje sadržaja poruke to jest adrese same transakcije. Tako se razlikuju najpoznatiji SHA256 algoritam koji je zaslužan za kriptiranje i dekriptiranje adresa Bitcoin transakcija, ali i nekih drugih, kao što su Mastercoin, MazaCoin, Namecoin, Peercoin i tako dalje. Druga najpoznatija kriptovaluta je Ethereum koji koristi Keccak-256 algoritam, dok Litecoin koristi Scrypt algoritam. Još jedan od pokazatelja raznolikosti kriptovaluta je način nastajanja novčića gdje postoje slučajevi kao što su IOTA ili XRP koji ne koriste tehnologiju lanca vrijednosti. Sve navedeno i još toga bit će obrađeno u nastavku ovoga rada u svrhu boljeg shvaćanja transakcija kriptovaluta.

2.3. Novčanici za kriptovalute

Novčanici za kriptovalute (eng. *Crypto Wallets*) ključni su za korištenje kriptosustava temeljenih na tehnologiji lanaca vrijednosti. Svaki korisnik, koji namjerava koristiti platformu lanca vrijednosti za bilo koju razmjenu transakcija, mora koristiti nekakav oblika novčanika za kriptovalute. Za razliku od tradicionalnih fizičkih novčanika koje korisnici drže u svome džepu, kriptovalute nisu zapravo „fizički“ pohranjene u ovom obliku novčanika. Ustvari, kriptovalute same po sebi nisu pohranjene na bilo kojem mjestu niti postoji bilo gdje u bilo kojem fizičkoj formi, ali se njihovo postojanje očituje kao zapis o transakcijama pohranjen u lancu vrijednosti kriptosustava. Pri kreiranju korisničkog računa koji se povezuje s novčanikom kreira se par ključeva koji se sastoji od jedinstvenog privatnog ključa i jedinstvenog javnog ključa te se on pohranjuje u novčanik za kriptovalute. Uz to, korisnik može potrošiti svoju kriptovalutu otključavanjem sredstava u novčaniku pomoću ključeva pohranjenih u njegovom novčaniku, te kako bi uspješno izvršio bilo kakvu transakciju, on mora otpisati svoje posjedovanje kriptovalute uz pomoć digitalnog potpisa preko adrese novčanika. U tom procesu ne postoji prava razmjena stvarnih kovanica, već razmjena vrijednosti podataka koji se nalaze u transakciji koji su kreirani u lancu vrijednosti (eng. *blockchain*). Novčanik za kriptovalute je mnogo sigurniji od klasične razmjene dobara zbog mnogo faktora, ali i on sam ima svoje nedostatke. Sami novčanik za kriptovalute definiran je dugačkim zapisom ili eng. *string-om,* koji predstavlja ključ pripadajućeg novčanika koji se generira uz pomoć kriptografije. (Suratkar, Shirole, 2020.)

Kada se govori o podjeli novčanika, postoje dva glavna tipa: novčanik kao (1) softver ili (2) hardver. Softverski ili „meki“ novčanici su programi ili aplikacije koji se mogu preuzeti na mobilnom uređaju ili računalu, dok su hardverski novčanici jedan oblik fizičkog skladišta u koji se pohranjuju podaci o kriptovalutama na posebno dizajniranom tvrdom disku koji se nalazi u uređaju. Softverski novčanici za kriptovalute dijele se na „vruće“ i „hladne“, te je važno napomenuti da „vrući“ novčanici moraju biti spojeni na Internet kako bi funkcionirali. S obzirom na to da ništa na Internetu nije potpuno sigurno, te je zaključeno da su sredstva zadržana u „vrućem“ novčaniku, uvijek je prisutan rizik od krađe ili gubitka sve imovine sačuvane na njemu. Taj ishod može se spriječiti pravilnim i sigurnim korištenjem mreže, te ne odavanjem privatnih informacijama direktno ili indirektno bilo kome na Internetu. „Hladni“ novčanik dobio je naziv po poboljšanoj sigurnosti u usporedbi sa „vrućim“ novčanikom koja se očituje u tome da se novčanik nalazi izvan mreže, što znači da nije spojen na Internet i ne može se namjerno ili slučajno kompromitirati na Internetu. Jedini način na koji bi napadač mogao doći do novčića spremljenih u „hladnom“ novčaniku je da fizički ukrade ili provali u uređaj. Svi popularni hardverski novčanici dizajnirani su tako da budu što sigurniji i mogu se sigurnosno kopirati na različite načine. Osim što pružaju mogućnost obavljanja transakcija, novčanici daju uvid u trenutno stanje i vrijednost kriptovaluta vlasnika novčanika, kao i sve prijašnje obavljene transakcije. Sva navedena svojstva ažuriraju se svake sekunde u realnom vremenu, s obzirom na to da se sama vrijednost određene kriptovalute mijenja iz sekunde u sekundu. Tablica 1. prikazuje neke primjere novčanika korištenih za kriptovalute. Tablica sadrži nazive novčanika, tko ih kontrolira, koje valute podržavaju, prikazuje razinu anonimnosti, platforme koje ih podržavaju te način na koji se može oporaviti i vratiti novčanik. (Suratkar, Shirole, 2020.)

Tablica 1. Novčanici za kriptovalute i njihova svojstva

Značajke / Ime novčanika	Coinbase	Cryptonator	GateHub	Guarda	Jaxx	Blockchain Wallet
Broj podržanih kriptovaluta	28	18	9	50	78	3
Pohrana ključevima	Na serverskoj strani	Na korisničkoj strani	Na serverskoj strani	Na korisničkoj strani	Na korisničkoj strani	Na korisničkoj strani
Osnovne valute	USD, EUR i još 32 drugih	USD, EUR, RUB, UAH	EUR, USD	EUR, USD, GBP, RUB, DKK	USD, EUR	USD, EUR
Razina anonimnosti (kod prijave)	korisnička identifikacijska oznaka	samo e-mail	samo e-mail	nema prijave	nema prijave	korisnička identifikacijska oznaka
Podržane platforme	Google Chrome, Mozilla Firefox, Microsoft Edge, Opera, Brave	Google Chrome, Mozilla Firefox, Android uređaji, iOS uređaji, OSX	Google Chrome, Mozilla Firefox	Google Chrome, Mozilla Firefox, Android uređaji, iOS uređaji, Linux, Windows, Mac	Google Chrome, Mozilla Firefox, Android uređaji, iOS uređaji, Linux, Windows, Mac	Google Chrome, Mozilla Firefox, Android uređaji, iOS
Oporavak "izgubljenog" novčanika	sigurnosna fraza	e-mail podrška	ključ za oporavak od 32 znaka	sigurnosna fraza	sigurnosna fraza	sigurnosna fraza

(Prema: Suratkar, Shirole, Bhirud, 2020.)

2.4. Transakcije kriptovalutama

Transakcije koje se provode korištenjem tehnologije temeljene na kriptovalutama, odnosno korištenjem lanca vrijednosti, odvijaju se na potpuno drugačiji način nego klasične „ne-fizičke“ novčane transakcije koje koriste na primjer kreditne kartice. U takvom okruženju, validnost transakcije provjerava banka kartične kompanije korisnika, dok se u sustavima lanca vrijednosti transakcije provjeravaju od strane korisnika mreže, takozvanih čvorova. Uzevši sve to u obzir, proces transakcije bit će pokazan na primjeru Bitcoina, koji predstavlja prvi takav decentralizirani sustav. Sama transakcija sastoji se od četiri glavna dijela: verzija (eng. *version*), vrijeme dodavanja (eng. *locktime)*, te ulaznih i izlaznih podataka. Verzija označava izjavu o pravilima koja transakcija slijedi, odnosno koja je verzija Bitcoin softvera korištena. Budući da nisu sve transakcije kreirane u istoj verziji, unesene vrijednosti u različitim verzijama mogu varirati, stoga je bitno da se taj podatak provjeri. *Locktime* označava „najbrže vrijeme” da se transakcija pridoda u lanac vrijednosti (eng. *blockchain*) ili pak vrijeme zaključavanja lanca vrijednosti prije transakcije. Na taj se način može odrediti da se iz nekog razloga transakcija pridoda u lanac vrijednosti za na primjer 10 sekundi, što znači da će vrijeme zaključavanja (eng. *locktime*) biti zapisano s vrijednošću koja odgovara „10 sekundi“. Proces transakcije odvija se na način da jedan korisnik prenosi „novac“ iz svog novčanika za kriptovalute na drugi novčanik za kriptovalute drugog korisnika preko Bitcoin adrese, te svaka transakcija mora imati digitalni potpis kako bi se osigurala valjanost, što je spomenuto i u prethodnom poglavlju. Jednostavnije rečeno Bitcoin transakcija predstavlja niz digitalno potpisanih podataka, odnosno datoteka, koje se emitiraju u mrežu. Ako su podaci valjani, unijet će se kao dio bloka transakcija na samome čelu lanca vrijednosti. Valjanost podataka sudionika u ovoj novčanoj razmjeni ne provjerava banka, već mreža korisnika iste te mreže, koji imaju status i odobrenje od same mreže da mogu potvrđivati ostale transakcije. Cijeli taj koncept će također biti objašnjen u sljedećem poglavlju, a proces transakcije će biti prikazan u nastavku i na primjeru. (Xi, Fan, Shenwen, 2020; Judmayer i sur., 2017; Dong-ha, 2019.)

Kao što je već rečeno, transakcije u Bitcoin mreži se koriste za prijenos valutnih jedinica s jedne adrese na drugu, pazeći na to da strana pošiljatelja posjeduje neku količinu valute. Ta količina je u literaturi označena kraticom „UTXO“ što na engleskom jeziku označava eng. *unspent transaction output* ili u prijevodu nepotrošeni transakcijski izlaz. Posjedovanje u tom kontekstu znači kontrola nad privatnim ključem adrese (to jest javnim ključem) koja trenutno drži valutne jedinice koje se prenose. Transakcija u Bitcoinu sastoji se od jednog ili više ulaznih podataka (eng. *input*; u konkretnom kodu se naziva i *vin*) i jednog ili više izlaznih rezultata (eng. *output;* u konkretnom kodu naziva se i *vout)*. Mehanizam koji provjerava vlasništvo ključeva naziva se eng. *script*, te se sastoji od dva stanja, stanje otključavanja *script*-a i stanje zaključavanja *script*-a. Ulazni podaci otključavaju prethodne izlazne podatke davanjem valjanog elektroničkog potpisa. Time se može zaključiti da ulazi transakcije služe kao dokaz da nositelj odgovarajuće adrese Bitcoin-a koji je prethodno primio Bitcoin također posjeduju traženi privatni ključ. Privatni ključ potreban je za generiranje potpisa koji otključava sredstva koja se mogu koristiti, odnosno prijenos na drugi Bitcoin račun. Primjerice, ako Alice želi prenijeti pet Bitcoin-a Bobu, prvo treba Bobovu Bitcoin adresu, uz pretpostavku da se ova adresa prenosi preko nekog pouzdanog komunikacijskog kanala, kao što je kanal za online kupnju. Prvo Alice stavlja Bobovu adresu u izlaznu transakciju koju želi izvršiti zajedno s brojem kriptovaluta/novčića koje želi prebaciti. U sljedećem koraku, Alice mora dokazati da posjeduje traženi broj Bitcoina i da ih stvarno želi prebaciti Bobu, što radi na način da stvara ulazne podatke za trenutnu transakciju za svaki izlazne podatke koje želi otključati. Ti ulazni podatci jedinstveno identificiraju prethodne izlazne rezultate prema njihovom identifikacijskom broju i broju transakcije. Da bi otključala te rezultate, Alice mora dokazati da je ona zakonita vlasnica, što ona radi davanjem kriptografskih potpisa zajedno sa svakim ulazom (eng. *input*). Čim se transakcija „izgradi“, Alice je prenosi u Bitcoin mrežu koja se temelji na međusobnoj suradnji članova mreže i čeka da bude uključena u novo generirani blok u lancu vrijednosti. Jednom kada je transakcija uključena na čelo lanca vrijednosti, transakcija se smatra potvrđenom. Broj potvrde se definira brojem blokova koji se nadograđuju jedan na drugi te oni sadrže same transakcije i informacije o njima. (Judmayer i sur., 2017; Dong-ha, 2019)

2.4. Primjena kriptovaluta u svijetu

Pojam kriptovaluta popularna je tema koja je svoje prve korake šire medijske pozornosti ostvarila tek posljednjih par godina. Eksplozije i medijska pažnja javlja se samo pri velikim skokovima same vrijednosti određene kriptovalute što je dobar pokazatelj da većina ljudi vidi ovaj pojam samo kao mogućnost ostvarivanja financijske koristi. U praksi to zapravo niti nije jedina primjena kriptovaluta, što će biti i obrađenu u ovome poglavlju.

2.4.1. Elektroničko glasovanje

Prvi iskorak koji bi imao vrlo pozitivne učinke u svakodnevnom svijetu je elektroničko glasovanje. Elektroničko glasovanje, ne samo u svrhu izbora, već je popularan termin u svijetu interneta jer korisnicima daje mogućnost anonimnog glasovanja te sakupljanja mišljenja i podataka bez potrebe da se otkrije njihov identitet. Uz to, brojke glasača koji izađu i zapravo glasuju se, barem u Hrvatskoj, a i u većini ostalih zemalja, sve više smanjuju, što nije dobar pokazatelj. Poražavajuća činjenica je i ta što se jedan dio glasačko sposobnih osoba ne odluči na izlazak zbog same potrebe za odlazak na glasačko mjesto ili pak samo čekanje u red na istome. Taj problem rješava elektroničko glasovanje koje omogućuje glasovanje iz vlastitog doma, kao i automatsko brojenje glasova koje znatno ubrzava sam proces te ga čini sigurnijim od krivotvorenja od strane brojača glasova. Uz sve te pozitivne strane, jedan od problema javlja se i kod potvrđivanja identiteta samog glasača, što zna biti česti trend u ovakvom obliku glasovanja. Takav slučaj, odnosno sumnje, pokazale su se i na američkim e-izborima 2020. godine, gdje se saznalo da su neki od glasača bile preminule osobe, što je naravno nemoguće.

Bilo kako bilo, sama arhitektura sustava lanca vrijednosti pridonosi decentraliziranost, automatiziranost te nemogućnost lažiranja rješava sve gore navedene probleme. Europska unija od 2014. godine razvija elektronički sustav glasovanja temeljen na lancu vrijednosti pod nazivom eng. „*Crypto voting*“. Ovaj projekt želi implementirati naprednu ideju o elektroničkom glasovanju kroz korištenje dvaju povezanih jednostranih lanca vrijednosti sa svrhom registriranja birača i njihovih glasova te prebrojavanje glasova dodijeljenih različitim kandidatima. Tajnost i sigurnost nečijeg glasa glavni je aspekt ovog projekta koji se koristi za prikupljanje potpisa koji po svojoj prirodi nisu anonimni, već javno dostupni na mjestima kao što su: referendumi, popularni zakoni o inicijativama, predstavljanje izbornih lista ili pojedinih kandidata za primarne izbore ili za stvarne političke izbore.

Cilj ovog projekta jest proučiti i razviti novi sustav elektroničkog glasovanja integriran s jednim ili više postupaka upravljanja izbornim događajima (uspostava sustava, distribucija vjerodajnica, glasovanje, prikupljanje glasačkih listića, prebrojavanje glasova, objavljivanje rezultata), korištenjem dvaju povezanih lanca vrijednosti. Jedan će lanac određivati i registrirati odnosno zapisivati birače s pravom glasa kao i njihove glasove, dok će drugi lanac te iste glasove brojati.

Ovaj transparentniji način glasovanja provodit će se u tri faze:

- Provođenje pripremnih aktivnosti i utvrđivanje izbornih popisa;
- Upravljanje glasovima (prikupljanje);
- Brojanje glasova, pri čemu glasovanje podrazumijeva automatsko prebrojavanje glasova. (netservice.eu, 2014. ; Alam, Zia Ur Rashid, Abdus Salam, Islam, 2018.)

2.4.2. Lanac opskrbe i logistika

Nadalje, mehanizam lanca vrijednosti očituje se i u drugim područjima kao što su energetika, zdravstvo, proizvodnja, poljoprivreda, poslovanje, distribucija digitalnog sadržaja, pametni gradovi, Internet stvari, opskrbni lanac, logistika te turizam. (Bodkhe, Tanwar, Parekh, Khanpara, Tyagi, Kumar, Alazab, 2020.)

Poljoprivrednom sustavu potrebni su kritični podaci upravljanja, kao što je upravljanje opskrbnim lancem koje igra značajnu ulogu u ljudskom životu. Tradicionalni logistički sustavi koji se koriste u opskrbi hranom i poljoprivredi čuvaju narudžbe i isporučuju ih na odredište. Ti konvencionalni sustavi imaju nedostatke u pogledu različitih obilježja kao što su vođenje financijskih knjiga, detaljno praćenje isporučene robe i transparentnost. Međutim, u suvremenom digitalnom dobu ove značajke mogu poboljšati sigurnost i kvalitetu hrane, a time i povećati potražnju potrošača za kvalitetom hrane. Ti se problemi mogu riješiti na učinkovit način uporabom tehnologije lanca vrijednosti. Decentralizirano rješenje utemeljeno na lancu vrijednosti nazvano je „AgriBlockIoT“, te je ono detaljnije obrađenu u djelu „*Blockchainbased traceability in agri-food supply chain management: A practical implementation*“. Ovaj projekt integrirao je razne senzore koji koriste Internet stvari (eng. *Internet Of Things*) koji bi stvarali i obrađivali podatke duž lanca vrijednosti. Do pohranjenih podataka će se lako moći doći, te će se raditi na razvijanju mogućnosti sklapanja pametnih ugovora, s ciljem postizanja transparentnosti i nepromjenjivosti evidencija narudžbi, ugovora i ostalog. (Bodkhe i sur., 2020; Caro, Vecchio, Giaffreda, 2018.)

2.4.3. Zdravstvo i zdravstvena zaštita

Zdravlje je najvrjednije sredstvo bilo koje države, stoga je njeno unaprjeđenje i poboljšanje veliki prioritet. U tradicionalnom zdravstvenom sustavu, svi podaci vezani uz pacijente čuvaju se u centraliziranoj bazi podataka ili pak samo kao fizički dokumenti kojima naravno nemaju svi pristup. Osim toga, privatnost i sigurnost informacija o pacijentima moraju se održavati, jer su ranjive do različitih vrsta napada i krađa. Centralizirana arhitektura ne može ispunite ove zahtjeve u potpunosti, stoga je bilo potrebno pronaći novo rješenje. Autori djela „*Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control*" predložili su rješenje temeljeno na lancu vrijednosti pod nazivom „*Goalthcare Data Gateway*" (HDG). U ovom sustavu pacijenti mogu sigurno dijeliti i kontrolirati svoje podatke sigurnim podatkovnim kanalima. Ovaj predloženi sustav obrađuje i upravlja podacima o pacijentima bez ikakve brige o gubitcima podataka ili neovlaštenim dijeljenima podataka. Ova pametna zdravstvena zaštita usmjerena je na praćenje i dijagnosticiranje zdravlja pacijenata na daljinu kroz bežične komunikacijske kanale. Prikupljanje potrebnih informacije o različitim pacijentima vršilo bi se kroz uređaje i senzore koje bi pacijenti nosili na sebi. To bi značilo da bi ova tehnologija mogla podupirati cjelokupni zdravstveni sustav pomoću distribuirane knjige koju dijele različiti korisnici kao što su pacijenti, liječnici, medicinske trgovine i agencije za osiguranje, ali naravno ima još puno mjesta za napredak i poboljšanja. (Yue, Wang, Jin, Li, Jiang, 2016; Bodkhe i sur., 2020)

3. Tehnološka osnovica kriptovaluta

U ovome poglavlju glavni fokus bit će tehnološka osnovica samih kriptovaluta, odnosno dublji pogled u detalje rada cjelokupne mreže lanca vrijednosti. Na početku će biti predstavljeno što su hash funkcije, koja su njihova svojstva te će dodatno biti obrađena dva najpoznatija korištena algoritma, SHA256 i Keccak256. Kod analize algoritama fokus će biti na prolaženje svih koraka kroz koji podaci prolaze te kako izgleda njihov izlaz nakon provođenja algoritma. Uz to, bit će pojašnjen pojam elektroničkog potpisa, kao i njegova primjena. Nadalje, poglavlje će sadržavati sve informacije o podjeli i procesu rudarenja kriptovaluta te će se cijeli proces opisati uz pomoć bitnih pojmova kao što su Dokaza o radu, te Dokaza o ulogu, bez kojih sustav lanca vrijednosti ne bi funkcionirao. Na kraju će ukratko biti opisana i primjena SHA256 algoritma u procesu rudarenja kriptovalute Bitcoin te će poglavlje završiti analizom lanca vrijednosti kao i njegovim dijelovima.

3.1. Kriptografski algoritmi i hash funkcije

Prethodno poglavlje pokazalo je kako i na koji način se primjena kriptovaluta u svijetu može isplatiti. Jedni od najbitnijih faktora zbog kojega ljudi u današnje vrijeme koriste kriptovalute su naravno anonimnost i sigurnost, te se može reći da se navedeni pojmovi velikom većinom ostvaruju uz pomoć kriptografije. Kriptografija kao znanost bavi se kontroliranjem prava pristupa korisnicima na način da sakriva poruku ili sadržaj koji se kodira uz pomoć jednog ili više algoritama te prenosi. Da bi se sadržaj dekodirao potrebno je da primatelj zna koji je algoritam pošiljatelj koristio pri enkripciji, na koji je način to izveo te još neke od faktora. Iako kriptografija predstavlja samo mali dio sigurnosnog sustava, ona predstavlja vrlo kritičan dio, stoga je njeno pravilno izvođenje od vrlo velike važnosti kada se radi o očuvanju podataka kao što su informacije o transakcijama. Konkretno, u kriptografiji koja je povezana s kriptovalutama prvi pojam koji se javlja je pojam hash funkcije. Hash funkcija je jednosmjerna funkcija koja kao ulaznu vrijednost uzima proizvoljno dugačak niz bitova (ili bajtova) i proizvodi rezultat, odnosno izlaz fiksne veličine. Važno je napomenuti da će se u daljnjem tekstu često koristiti izraz „*hash*“, stoga je bitno zapamtiti taj izraz.

Tipično korištenje hash funkcije je elektronički potpis, što će biti spomenuto i u narednom poglavlju. Ljepota i praktičnost korištenja hash funkcija očituje se već i na jednostavnom primjeru. Recimo da imate poruku *m*, te se ta ista poruka može potpisati. Međutim, operacije većine programa digitalnog potpisa koje su vezane za izračun javnog ključa prilično su skupe u računalnom smislu. To znači da je umjesto potpisivanja same poruke *m* puno isplativije primjenjivanje hash funkciju *h* na poruku *m* što rezultira potpisivanjem h(*m*), što naravno u sustavu lanca vrijednosti označava potpisanu datoteku. Na taj način, rezultat *h* obično iznosi izraz duljine od između 128 i 1024 bita, u usporedbi s višestrukim tisućama ili milijunima bitova za samu poruku *m*. Potpisivanje h(*m*) je stoga mnogo brže od izravnog potpisivanja *m*, gdje je najbitniji faktor ove konstrukcija sigurnost, te nemogućnost izvedbe ili konstrukcije dvije poruke m_1 i m_2 koje imaju istu vrijednost. Baš iz tog razloga, sigurnost se očituje u prvom svojstvu hash funkcija, a to je jednosmjernost funkcije. To svojstvo znači da je s obzirom na ulazni niz podataka *m* lako izračunati h(*m*), ali s obzirom na vrijednost *x* nije moguće pronaći *m,* stoga se može zaključiti da vrijedi h(*m*) ≠ *x*. Drugim riječima, jednosmjerna funkcija je ona funkcija koja se može izračunati, ali od nje nije moguće dobiti inverz (ili ga je barem jako teško pronaći). Od brojnih svojstava koje bi trebala imati dobra hash funkcija, ona koja najčešće se spominje otpornost na sudare (eng. *collision resistance*). Sudar označava dva različita ulaza m_1 i m_2 za koje vrijedi $h(m_1) = h(m_2)$. Naravno, svaka hash funkcija ima beskonačan broj tih sudara. To je naravno istina pod pretpostavkom da postoji beskonačan broj mogućih ulaznih vrijednosti (eng. *input*) i samo ograničen broj mogućih izlaznih vrijednosti (eng. *output*). Dakle, teško je da postoji hash funkcija koja nikada nije bez sudara, što znači da je cilj stvoriti ne-savršenu ali vrlo efektivnu hash funkciju koja će ulaganje resursa ili vremena u njezino probijanje učiniti neisplativim. Otpornost na sudare svojstvo je koje hash funkcije razlikuje prema primjeni, ovisno o sposobnosti otpornosti, te je zbog toga vrlo bitna.

Nadalje, kriptografske hash funkcije moraju biti efikasne u svojem radu, što znači da u najmanjem broju koraka, odnosno petlji moraju doći do željenog izlaza (eng. *computational efficiency*). Uz to, bitno svojstvo koje čini hash funkcije korisnim u svakodnevnoj primjeni jest raznolikost, odnosno izgled izlaznih podataka nema nikakve poveznice s izgledom ulaznih podataka. Zbog toga hash funkcije gotovo uvijek imaju izgled kao niz slučajnih slova i brojeva, što one zapravo i jesu, samo što su nastale vrlo smislenim redoslijedom koji izvršava sam algoritam, te će taj algoritam biti opisan u nastavku. Neki od algoritama koji se koriste u različitim kriptovalutama su SHA256, Keccak256, Scrypt, Equihash, Cryptonight i X11, a u narednim poglavljima bit će obrađena dva najpoznatija. (Kohno, Ferguson, Schneier, 2010.)

3.1.1. SHA256 algoritam

SHA256 predstavlja prvu i najviše korištenu kriptografsku funkciju koja u svojoj srži koristi algoritam stvoren 2001. godine. Ovaj algoritam dio je SHA-2 obitelji kriptografskih funkcija koja se pokazala puno više efikasnija i sigurnija od SHA-1 obitelji zbog mnogih proboja u zaštiti samih algoritama. Za njegovu visoku korištenost zaslužan je naravno Bitcoin koji je od samoga početka temeljen na ovoj arhitekturi. Najznačajnija primjena ovog algoritma pojavljuje se u kriptiranju detalja svake transakcije koja se provodi kroz Bitcoin mrežu. SHA256 temeljen je na Merkle–Damgård konstrukciji koja predstavlja metodu izgradnje kriptografskih hash funkcija „otpornih“ na kolizije iz jednosmjernih funkcija također „otpornih“ na kolizije (ne savršeno otpornih).

Za početak važno je naglasiti da algoritam procesuira podatke u binarnom zapisu te ulazna riječ ili podatak mora biti prevedena uz pomoć 8-bitne ASCII tablice kako bi vrijednost svakog znaka poruke dobila binarni zapis čitljiv od strane računala. To se izvodi na način da se svakom znaku pridruži izraz duljine četiri bita te tada cjelokupnu riječ tvori niz bitova. Sami algoritam prima ulazni podatak proizvoljne veličine „*L“* koja se nalazi unutar domene $0 \leq L < 2^{64}$, a kao finalni rezultat izbacuje izlaz veličine 256 bitova, što se može vidjeti i u nazivu algoritma. Da bi algoritam proveo barem jednu komputaciju, mora primiti vrijednost izraza ili takozvani blok ukupne duljine od barem 512 bitova, gdje je svaka pojedina „riječ“ podijeljena u 32-bitne zapise. Taj se proces ostvaruje uz pomoć takozvanog „Postavljanja i raščlanjivanja poruke“ (eng. *Message Padding and Parsing*). Postavljanja i raščlanjivanja poruke čine dva od tri predprocesa provođenja algoritma te je njihova svrha dodatno obraditi i prevesti podatke u jezik lakše čitljiv računalu. Predproces postavljanja (eng. *padding*) može se izvršiti u bilo kojem trenutku prije samog procesa komputacije. Zapis u bloku ostvaruje duljinu od 512 bitova na način da se dijeli u blokove višekratnika broja 512. Postoje slučajevi gdje je zapis prekratak ili premašuje vrijednost od 512 bitova, te je sukladno tome potrebno zadnji blok nadopuniti. U tom slučaju zadnji blok sadrži ostatak bitova te se nedovršene riječi bloka nadopunjuju s bitnim stanjem 1 iza kojeg slijedi onoliki niz broja nula koliko je potrebno da se ostvari željeni broj od 448 bitova, te obavezno na kraju dodaje 64-bitni zapis koji na primjeru označava decimalni broj bitova originalne poruke u heksadecimalnom obliku.

primjer postavljanja originalna poruka a dodavanje jedinice na kraj

(a) 01100001 01100010 01100011 01100100 01100101

(b) 01100001 01100010 01100011 01100100 01100101 1

L=40, te uz pomoć formule (L+1+K) mod 512 = 448, dolazi se do rješenja K=407 gdje L označava duljinu riječi u bitovima, a K označava broj nula koje trebaju biti nadopisane, jer K treba predstavljati najmanje ne-negativno rješenje jednadžbe

pretvaranje cijelog bloka iz binarnog u heksadecimalni zapis

(c) 61626364 65800000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000000
00000000 00000000 00000000 00000028

16. riječ označena crvenom bojom je označava duljinu originalne poruke, decimalno 40 je 00000000 00101000 binarno a to je 00000000 00000028 heksadecimalno

Ovaj algoritam u svojem radu koristi šest logičkih funkcija/operacija, od kojih svaka radi s 32-bitnim zapisima koje se nazivaju „riječi“, te se te riječi u ovom primjeru označavaju uz pomoć slova *x*, *y* i *z*. Važno je reći da je rezultat svake od ovih funkcija opet 32-bitna riječ. Logičke funkcije korištene u ovom algoritmu koriste logičke operatore NOT; AND; XOR; ROTR; OR; SHR; te modularnu adiciju.

SHR predstavlja skraćenicu koja na engleskom jeziku znači „*shift right*“ odnosno „pomakni desno“ te radi na način da pomiče bitove u desno za određeni broj bitova, što rezultira time da se bitovi s desna gube, dok s lijeva dolazi novi niz bitova stanja nula.

ROTR je sljedeći operator te njegova skraćenica na engleskom jeziku znači „*rotate right*“ ili „rotiraj lijevo“ a radi na način sličan SHR-u, samo što brojevi rotirani s desne strane ne nestaju, već se vraćaju s lijeve strane, što rezultira kružnom kretanju vrijednosti bitova određene riječi.

Sljedeći operator kojeg dodatno treba spomenuti naziva se „XOR“ (eng. *exclusive or*). On ispisuje jedinicu (odnosno vraća vrijednost *true*) ako i samo ako vrijednosti jednadžbe sadrže samo jednu istinitu tvrdnju.

primjer:

	1010
XOR	0111
=	1101

Svrha korištenja ovog operatora je mogućnost spajanja različitih „riječi“ koje se sastoje od bitova kako bi dobili balansiranu reprezentaciju svih bitova, ali u kraćem zapisu.

Posljednji operator koji treba biti objašnjen je modularna adicija koja funkcionira kao klasična adicija/zbrajanje bitova samo što se rezultat zbrajanja mora provesti kroz mod broja 2 na trideset i drugu potenciju (mod 2^{32}) u svrhu ograničavanja izlazne riječi na već spomenutu veličinu od 32 bita. Ona se primjenjuje kroz sve adicije kod ovog algoritma. Kombinacijom svih tih operatora dolazi se do spomenutih šest funkcija koje izgledaju ovako:

```
CH( x, y, z) = (x AND y) XOR ( (NOT x) AND z)

MAJ( x, y, z) = (x AND y) XOR (x AND z) XOR (y AND z)

BSIG0(x) = ROTR^2(x) XOR ROTR^13(x) XOR ROTR^22(x)

BSIG1(x) = ROTR^6(x) XOR ROTR^11(x) XOR ROTR^25(x)

SSIG0(x) = ROTR^7(x) XOR ROTR^18(x) XOR SHR^3(x)

SSIG1(x) = ROTR^17(x) XOR ROTR^19(x) XOR SHR^10(x)
```

Funkcije *BSIG0* i *BSIG1* se koriste u glavnom algoritmu heširanja, dok se *SSIG0* i *SSIG1* koriste u izračunu težine kalkulacija, odnosno broju pozitivnih bitova. Bitno je naglasiti da se u literaturi *BSIG0* i *BSIG 1*, te *SSIG0* i *SSIG1* znaju spominjati i kao $\sum_0$ i $\sum_1$ te σ_0 i σ_1.

U svrhu boljeg i efikasnijeg skrivanja poruke, SHA256 koristi i konstante, koje su naravno poznate. Te konstante (K_t) definirane su vrijednostima prvih 32-bitnih decimalnih vrijednosti trećeg korijena prvih 64 primarnih brojeva, nakon čega znaju biti prevedene u heksadecimalni oblik.

Posljednji podproces zove se inicijalizacija, što bi značilo da se inicijalna hash vrijednost $H^{(0)}$ zapisuje kao osam 32-bitnih riječi u heksadecimalnom zapisu, koje su nastale uzimanjem prvih 32 bita decimalnog zapisa korijena prvih osam prostih brojeva. Ove vrijednosti znaju se nazivati i inicijalne varijable (eng. *initialisation vector – IV*) te nakon provođenja prvih 64 rundi za $H^{(0)}$, nove inicijalne varijable postaju rezultati tih rundi, i tako dalje.

```
H0(0) = 6a09e667
H1(0) = bb67ae85
H2(0) = 3c6ef372
H3(0) = a54ff53a
H4(0) = 510e527f
H5(0) = 9b05688c
H6(0) = 1f83d9ab
H7(0) = 5be0cd19
```

Kako bi algoritam mogao krenuti s procesiranjem potrebno je: (1) nešto što se naziva raspored poruka ili eng. *message schedule*, (2) osam radnih varijabli od kojih se svaka sastoji od 32 bita i označavaju se slovima od *a* do *h* te (3) hash vrijednosti od osam inicijalnih 32 bitnih riječi navedenih gore. Te vrijednosti, od $H_0^{(i)}$ do $H_7^{(i)}$ sadržavat će inicijalnu hash vrijednost $H_{(0)}$ koja će kasnije redom biti zamijenjena svakom sljedećom nadolazećom srednjom hash vrijednošću (nakon obrade svakog bloka poruke) $H_{(i)}$ pa sve do završne hash vrijednosti $H_{(N)}$. SHA256 također koristi pomoćne izraze odnosno riječi, označene s T_1 i T_2.

(1) Procesiranje algoritma započinje inicijalizacijom, odnosno postavljanjem inicijalne hash vrijednosti $H_{(0)}$.

(2) Zatim slijede potprocesi „Postavljanja i raščlanjivanja poruke“ (eng. *Message Padding and Parsing*)

(3) Nakon pripreme slijedi početak komputacije za koju su potrebne gore navedene funkcija i konstante, te se obavezno primjenjuje modularna adicija 2^{32}.Svaki blok poruke $M_{(1)}$, $M_{(2)}$, …, $M_{(N)}$ se obrađuje, jedan po jedan koristeći petlju:

For i=1 to N:
{

1. Priprema za raspored poruka, $\{W_t\}$:

```
For t = 0 to 15
   Wt = M(i)t
For t = 16 to 63
```

$\sigma_1^{\{256\}}(W_{t-2}) + W_{t-7} + \sigma_0^{\{256\}}(W_{t-15}) + W_{t-16}$

gdje vrijedi, $0 \le t < 15$,
i gdje vrijedi, $16 \le t < 63$

Message scheduler je zaslužan za ponovnu podjelu blokova u 64 riječi duljine 32 bita koje se označavaju slovom W i brojevima, odnosno sufiksom od 0 do 63 (W_0, W_1,…, W_{63}). To postiže na način da kreira 16 riječi od inicijalnog bloka riječi te ostalih 48 stvara korištenjem kombinacija operacija navedenih gore (drugi red funkcije).

2. Zatim slijedi inicijalizacija osam radnih varijabli od *a* do *h* uz pomoć (*i*-1)-te vrijednosti hasha za svakog člana:

$a = H_0^{(i-1)}$
$b = H_1^{(i-1)}$
$c = H_2^{(i-1)}$
$d = H_3^{(i-1)}$
$e = H_4^{(i-1)}$
$f = H_5^{(i-1)}$
$g = H_6^{(i-1)}$
$h = H_7^{(i-1)}$

3. Nakon toga primjenjuje se i glavna hash komputacija koja se naziva kompresija na način:

```
For t = 0 to 63
    T1 = h + BSIG1(e) + CH(e,f,g) + Kt + Wt
    T2 = BSIG0(a) + MAJ(a,b,c)
    h = g
    g = f
    f = e
    e = d + T1
    d = c
    c = b
    b = a
    a = T1 + T2
```

- u svakoj petlji algoritma prijašnja vrijednost varijable *a* postaje sljedeća vrijednost varijable *b*, prijašnji *b* postaje *c* i tako dalje

4. Izračun srednjih vrijednosti hashova H(i)

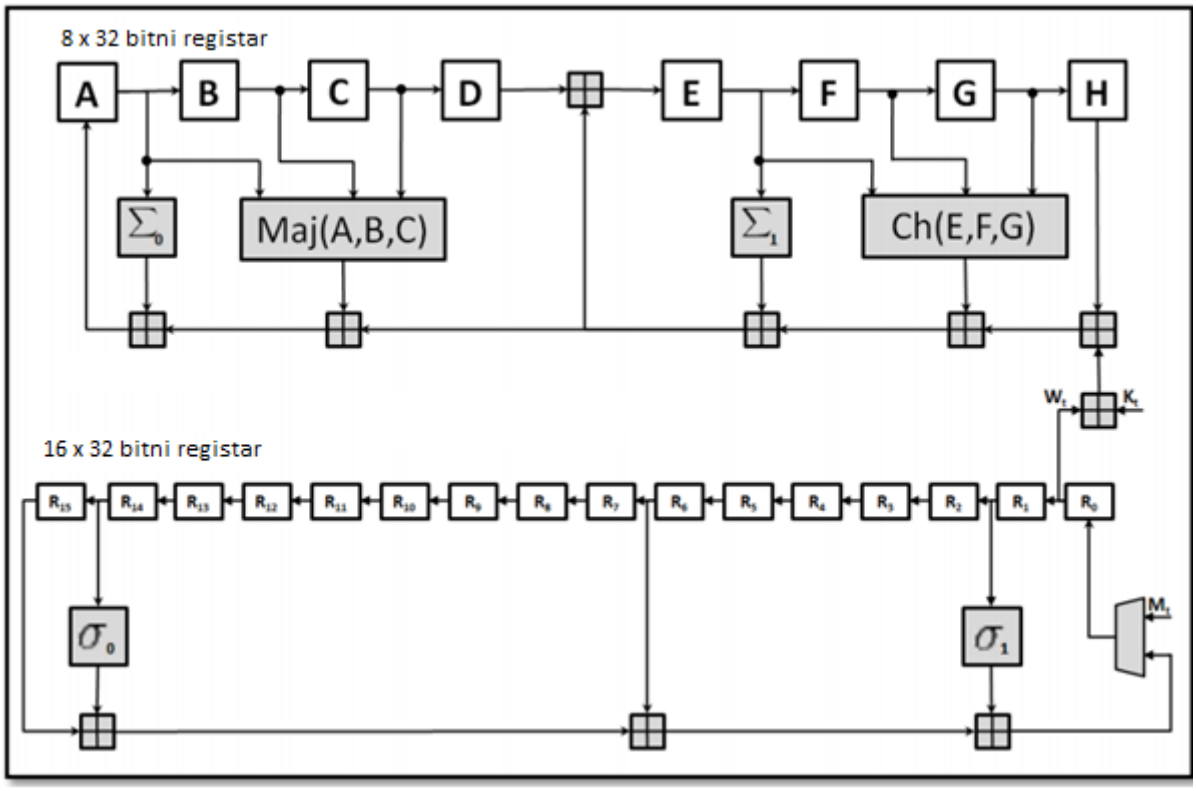
$$H_0^{(i)} = a + H_0^{(i-1)}$$
$$H_1^{(i)} = b + H_1^{(i-1)}$$
$$H_2^{(i)} = c + H_2^{(i-1)}$$
$$H_3^{(i)} = d + H_3^{(i-1)}$$
$$H_4^{(i)} = e + H_4^{(i-1)}$$
$$H_5^{(i)} = f + H_5^{(i-1)}$$
$$H_6^{(i)} = g + H_6^{(i-1)}$$
$$H_7^{(i)} = h + H_7^{(i-1)}$$

Finalni hash je duljine od 256 bitova te konkatenacijom varijabli od H0(N) do H7(N) dolazi se do oblika:

$$H_0^{(N)} || H_1^{(N)} || H_2^{(N)} || H_3^{(N)} || H_4^{(N)} || H_5^{(N)} || H_6^{(N)} || H_7^{(N)}$$

ili konkretno u danom primjeru rješenje iznosi (u heksadecimalnom zapisu):
05bcde980d17ede8dc986f0f98d21b2b482c4e8e34f09b6c67e1edf3bd174897

(Hannesdóttir, 2020; Hansen, 2006; FIPS, 2012)



Slika 1 Funkcija kompresije poruke (gore) i Message scheduler (dolje) (Prema: Naik, 2013)

3.1.2. Keccak256 algoritam

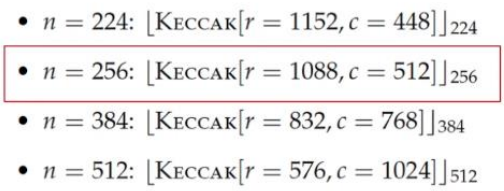
Keccak256 dio je SHA3 obitelji kriptografskih algoritama najčešće korištenih za heširanje. Njegova primjena i sigurnost najviše je očitovana u Ethereum sustavu gdje se ovaj algoritam najviše i koristi. Iako Bitcoin i Ethereum dijele mnoge sličnost što se tiče svojstava, sama srž kriptovalute definirana je algoritmom koji koristi, što u konačnici čini veliku razliku.

Pokrenut lavinom diskusija i špekulacija u prvom desetljeću 21. stoljeća zbog probijanja SHA0 obitelji i MD5 algoritma, osnivač Ethereuma, Vitalik Buterin odlučuje se za korištenje Keccak256 algoritma, umjesto do tada trenutnog, a i sada najraširenijeg algoritma, SHA256. Glavni razlog promjene je taj što je za njezine prethodnike otkriven niz napada i slabosti (SHA1 obitelj), te je strah od daljnjih probijanja novijih algoritama iz SHA2 obitelji rastao. Dakle, kako bi izbjegao istu sudbinu, Buterin se odlučuje za Keccak256 koji je bio potpuno drugačiji od postojećih SHA algoritama i AES algoritma. Prva razlika koja se očituje je sama konstrukcija algoritma. Dok SHA koristi Merkle–Damgård konstrukciju, Keccak koristi spužvastu konstrukciju (eng. *sponge construction*) koja predstavlja klasu algoritama s konačnim unutarnjim stanjem koji uzimaju ulazni bitni tok bilo koje duljine i proizvode izlaz željene duljine. U slučaju Ethereuma, Keccak256, kao što mu i sam naziv kaže, kao svoj izlaz izbacuje hash vrijednosti duljine 256 bitova.

Spužvasta konstrukcija okvir je za određivanje funkcija koje unose podatke u binarnom obliku, a vraćaju izlaze proizvoljne duljine. Za funkcioniranje algoritma potrebne su tri glavne komponente:

- temeljna funkcija koja sadrži podatke fiksne duljine, označena s *f*,
- parametar koji se naziva brzina odnosno eng. *rate*, označena sa slovom *r*, i
- pravilo slaganja/postavljanja odnosno eng. *padding* koje se koristilo i u prijašnjem algoritmu, označeno izrazom „*pad*“.

Funkcija koja je izgrađena od ovih komponenata naziva se spužvastom funkcijom eng. *sponge function*, te se ona označava s izrazom SPONGE [*f*, *pad*, *r*]. Spužvasta funkcija uzima dva ulaza: ulaz tipa *string* u bitovima, označen sa slovom *N*, i duljinu bita, označenu sa slovom *d*, koji ima output, SPONGE [*f*, *pad*, *r*] (*N*, *d*). Pojednostavljena analogija rada ovog algoritma kaže da proizvoljni broj ulaznih bitova „apsorbira” stanje (eng. *state*) funkcije, nakon čega proizvoljan broj izlaznih bitova biva „istisnut” izvan svog stanja. Stanje bi u ovom kontekstu predstavljalo niz bitova (ili bit) koji se u više navrata ažuriraju tijekom izvođenja računalnih procedura. U glavnoj Keccak komputaciji stanje je definirano ili kao trodimenzionalno polje ili kao niz bitova.



Slika 2. Definiranje parametara (Izvor: youtube.com, 2021)

Kao što je vidljivo na slici 2., postoje četiri različite vrste Keccak algoritma, koji se primarno razlikuju po duljini izlaza koji proizvodi algoritam, te time algoritam dobiva definirani tip izlaza. Svi Keccak algoritmi primaju vrijednosti do 1600 bitova što znači da su proizvoljne vrijednosti zbroja parametara brzine *r* i kapaciteta *c* moraju biti u granicama do 1600.

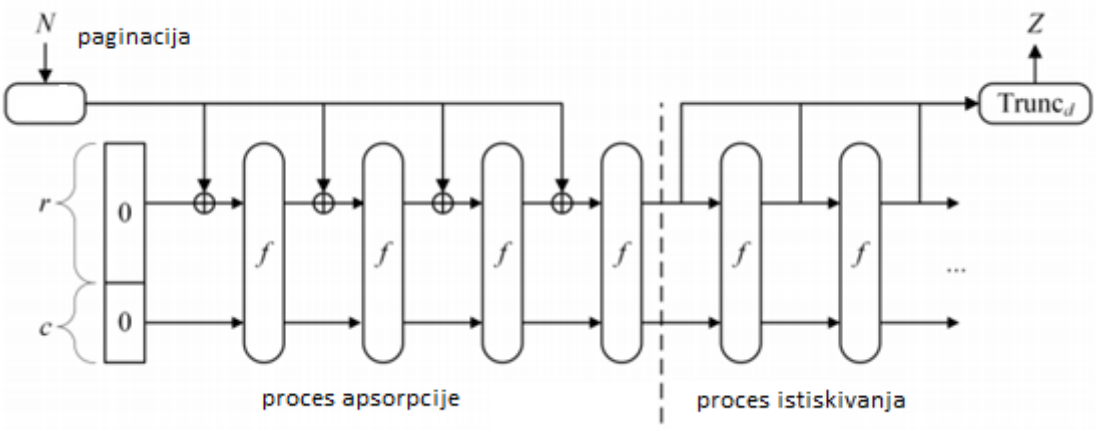
$$\mathrm{r} + \mathrm{c} = \mathrm{b} \in \{25{,}50{,}100{,}200{,}400{,}800{,}1600\}$$

Širina ili eng. *width* funkcije *f* definirana je slovom *b*. *R* predstavlja broj bitova poruke koji će biti procesuiran u svakoj rundi petlje algoritma, *c* je povezan s razinama sigurnosti, a *b* predstavlja širinu glavne funkcije. Još jedno ograničenje koje se mora poštovati kod definiranja parametara je da vrijednost *c* mora biti manji od vrijednosti b $(c < b)$. Konkretno za Keccak256, *r* iznosi 1088, te *c* iznosi 512., a formula izračuna glasi: $\mathrm{c} = 2\mathrm{d}$, gdje *d* predstavlja veličinu izlaza, odnosno formula za brzinu glasi: $\mathrm{r} = \mathrm{b} - 2 * \mathrm{c}$.

Nakon što su neki od parametara definirani, valja proučiti pravila postavljanja. Ova funkcija postavljanja naziva se i eng. *multi-rate padding function* i zaslužna je za generiranje niza odgovarajućih duljina u svrhu dodavanja znakova drugome nizu. Uzevši na primjer pozitivan cijeli broj x i negativan cijeli broj *m*, izlazni blok *(x, m)* predstavlja niz kod kojeg vrijedi da je $\mathrm{m} + \mathrm{len}\,(\mathrm{blok}\,(\mathrm{x}, \mathrm{m}))$ pozitivan višekratnik broja *x*. Unutar konstrukcije spužve, vrijednost x je jednaka vrijednosti *r* i vrijednost *m* je jednaka dužini *len(N)*, tako da se ulaz na kojemu je primijenjeno postavljanje može podijeliti u niz r-bitnih string-ova. Ulaz postavljanja čini pozitivan cijeli broj *x* odnosno poruka transakcije i ne-negativan cijeli broj *m*, a kao izlaz string *P* kod kojeg vrijedi da je $\mathrm{m} + \mathrm{len}\,(\mathrm{P})$ pozitivan višekratnik broja x. Postavljanje poruke *m* (pad 10*1) izvodi se na način $\mathrm{j} = (-\mathrm{m} - 2) \bmod \mathrm{x}$ te izlaz P izgleda ovako $\mathrm{P} = 1||\,0\mathrm{j}\,||\,1$. J predstavlja pomoćnu varijablu koja označava broj nula koji se dodaje kako bi se postigla željena duljina koja iznosi višekratnik od r. To znači da proces postavljanja originalnoj poruci pridodaje minimalno 2 bita, a maksimalno r + 1 bitova.

Zanimljivo je da je National Institute of Standards and Technology izmjenio vrijednosti postavljanja dodajući sufiks veličine dva bita, 0 i 1. Razlog tome bilo je lakše razlikovanje ulaza jedne manje Keccak obitelji, Keccak [c], čiji ulazi proizlaze iz hash funkcija od strane SHA3 obitelji i ulaza koji proizlaze iz XOF operacija u SHA3-u, $\mathrm{SHA3} - 256(\mathrm{M}) = \mathrm{Keccak}\,[512](\mathrm{M}\,||\,01, 256$ iako to nije bio dio originalne dokumentacije Keccak tima. Zbog tog razloga algoritam primijenjen u Ethereumu ne pridodaje već spomenuti sufiks pri kreiranju hash vrijednosti.

Nakon postavljanja slijedi glavna provedba spužvaste funckije koja je podijeljena u dva dijela, fazu „apsorpcije“ i fazu „istiskivanja“. Konstrukcija spuže prikazana je na slici 3.



Slika 3. Konstukcija spužve (FIPS, 2015)

(1) Faza apsorpcije započinje nakon primjene postavljanja na način da se definira inicijalno stanje *S* koje će naravno biti promjenjivo.

Početno stanje S iznositi, $S = 0^b$, gdje je b= 1600, što znači da je prvi spojeni pravokutnik s lijeva na slici gore sačinjen samo od vrijednosti nula.

(2) Zatim se provodi procesiranje svakog bloka riječi koji su prošli postavljanje P_i.

```
For each Pi:
      pad Pi || 0^(b-r)
      S = S ⊗ Pi || 0^(b-r)
      S = f (S)
```

U ovoj petlji na blokove se ponovno primjenjuje postavljanje kako bi ukupna duljina postala duljina b bitova, odnosno 1600.

Nakon toga na stanje S primjenuje se XOR operacija označena znakom „⊗“ te se prosljeđuje funkciji *f*, a komputacija nastavlja sa sljedećim blokom P_1 dok se ne obrade svi blokovi.

(3) Odmah nakon faze apsorpcije počinje faza istiskivanja, koja počinje s definiranjem praznog stringa Z_0 kojem se pridružuje vrijednost prvih r bitova stanja. Ako je duljina Z_0 veća od duljine željenog outputa, algoritam će Z_0 pridružiti varijabli Z te će ispisati prvih n bitova stringa. Ako je duljina pak premala, algoritam vrti stanje kroz još jednu petlju koja prosljeđuje stanje ponovno funkciji *f* i ovaj proces se ponavlja sve dok se ne dođe do željenog broja bitova.

```
Z0 =⌊S⌋r
      if |Z0| ≥ n:
            Z= Z0
            ispiši  |Z|n  kao
vrijednost hasha
      if |Z0|< n:
            S = f(S)
            Z1 = ⌊S⌋r
            Z = Z0 || Z1
```

Nakon opisivanja rada algoritma treba opisati glavnu funkciju komputaciju *f*, gdje *f* označava komputacijsku funkciju broja n_r -a (eng. *number of rounds*) ovisnu o vrijednosti R.

Kao što je spomenuto, stanje S može biti prikazano/konvertirano u trodimenzionalnom obliku kao što je prikazano na slici dolje. Stanje permutacije sastoji se od *b* broja bitova, što je u ovom slučaju 1600. Znajući taj broj mogu se izračunati i dvije preostale varijable w i l. W se izračunava uz pomoć formule $b/25$ dok se l izračunava formulom $\log2(b/25)$. To bi značilo da je u ovom slučaju vrijednost w = 64, a vrijednost l = 6. Uz pomoć tih brojki i ograničenja, izgrađuje se trodimenzionalno tijelo za sve trojce za koje vrijedi:

$$0 \leq x < 5,\ 0 \leq y < 5 \text{ i } 0 \leq z < w$$

$S[w(5y + x) + z] = a[x][y][z]$, gdje varijabla x sadrži vrijednosti od 0 do 4, a *y* također sadrži vrijednosti od 0 do 4, dok *z* sadrži vrijednosti $\text{od } 0 \text{ do } 2l - 1 (z = 0, \dots, 63)$.

Sukladno tome, nastalo geometrijsko tijelo ima vrijednosti $1600 = 5 * 5 * w \text{ i } nr = 12 + 2l = 24$, te se sastoji od mnogo malih dijelova (Slika 10.). Najmanji dio (jedna kocka tijela) predstavlja jedan bit, zatim slijede redovi, stupci i staze. Nakon toga podjela se nastavlja na „stranice“ koje se dijele na okomite, vodoravne i položene stranice (eng. *plane, slice, sheet*).

Iz toga se može zaključiti da će algoritmu biti potrebno 24 runde funkcije R koja se sastoji od pet manjih funkcija i definira kao:

$$R = \iota(\chi(\pi(\rho(\theta(A)))))$$

Svaki algoritam funkcija kao ulaz unosi stanje područja A (eng. *state array*), a kao izlaz vraća novo stanje područja, *A'*.

(1) Prva funkcija R-a, „*θ*“, primjenjuje XOR operator na svaki bit u stanju s paritetom dvije kolone. Konkretno, kod bita A [x_0, y_0, z_0], x i z koordinate u jednom od stupaca iznose $(x0 - 1) \bmod 5$, dok x koordinate drugog stupca iznose $(x0 + 1) \bmod 5$, a *z* koordinate $(z0 - 1) \bmod w$. Ova funkcija provodi se u svrhu smanjenja difuzije.

Za sve parove *(x, z)* za koje vrijedi $0 \leq x < 5$ i $0 \leq z < w$ primjenjuje

```
C [x, z] = A[x, 0, z] ⊗ A[x, 1, z] ⊗ A[x, 2, z] ⊗ A[x, 3, z] ⊗ A[x, 4, z]
```

Za sve parove (x, z) za koje vrijedi $0 \leq x < 5$ i $0 \leq z < w$ primjenjuje

```
D [x, z] = C [(x -1) mod 5, z] ⊗ C [(x+1) mod 5, (z - 1) mod w]
```

I za sve trojce (x, y, z) za koje vrijedi $0 \leq x < 5, 0 \leq y < 5$ i $0 \leq z < w$ primjenjuje

```
A' [x, y, z] = A [x, y, z] ⊗ D [x, z]
```

(2) Druga funkcija „ρ“ rotira bitove svake staze (eng. *lane*) po dužini, što se još naziva i pomak eng. *offset*. Pomak ovisi o fiksnim koordinatama *x* i *y* staza. Sukladno tome, za svaki bit u stazi, z koordinata se modificira na način da pridodaje pomak, ali se na tu vrijednost primjenjuje mod dužine te staze.

Za sve z-ove za koje vrijedi $0 \leq z < w$,

neka `A' [0, 0, z] = A [0, 0, z]` i `(x,y) = (1,0)`

```
For t from 0 to 23:
```

Za sve z-ove za koje vrijedi $0 \leq z < w$,

neka `A'[x, y, z] = A[x, y, (z-(t+1)(t+2)/2)mod w]`

```
(x+y) = (y,(2x+3y) mod 5)

return A'
```

(3) Treća funkcija „π“ zaslužna je za transpoziciju staza što rezultira disperzijom u svrhu dugoročne difuzije.

```
Za sve trojce (x, y, z)
      za koje vrijedi 0 ≤ x < 5, 0 ≤ y < 5 i 0 ≤ z < w
      neka
        A' [x, y, z] = A[(x + 3y) mod 5, x, z]

      return A'
```

(4) Četvrta funkcija „χ“ je jedina ne-linearna funkcija u ovom algoritmu te je ona zaslužna za cikličnost funkcije R. Bez nje, R bi bila linearna funkcija. Na svaki bit u području stanja primjenjuje se XOR operator sa ne-linearnom funkcijom druga dva bita iz reda.

```
Za sve trojce (x, y, z) za koje vrijedi 0 ≤ x < 5, 0 ≤ y < 5 i 0 ≤ z < w
neka
A' [x, y, z] ⊗ ((A [(x+1) mod 5, y, z] ⊗ 1) · A [(x+2) mod 5, y, z])

      return A'
```

(5) Posljednja funkcija ι zaslužna je za zbrajanje cjelobrojnih konstanti s ciljem ravnomjerne distribucije simetrije. Bez nje, ciklične funkcije bile bi translacijski invarijantne u smjeru z osi i sve runde algoritma bile bi iste, što bi značilo da je algoritam jako podložan za napad. Bitovi cikličnih konstanti se razlikuju od runde do runde a definiraju se izlazom maksimalne dužine eng. *Linear Feedback Shift Register*-a (LFSR). LFSR se u računalstvu često koristi za ispis slučajnih brojeva, odnosno u ovom slučaju nula i jedinica u savršenom omjeru. S obzirom na to da na to se maksimalna duljina LFSR-a ne može predvidjeti, to ga čini idealnim za primjenu u kriptografiji. (Texas instruments Incorporated, 1996)

Ova funkcija za razliku od ostalih prima dva ulaza, te se zato dijeli na dva dijela. Prvi ulaz definiran je vrijednosti cijelog broja *t*, a prvi izlaz koji nastaje je u bitnom zapisu rc*(t)* koja je definirana dolje:

```
if t mod 255 = 0, return 1
      let R = 10000000
      for i from 1 to t mod 255, let:
            R = 0 || R;
            R[0] = R[0] ⊗ R[8];
            R[4] = R[4] ⊗ R[8];
            R[5] = R[5] ⊗ R[8];
            R[6] = R[6] ⊗ R[8];
            R = Trunc8[R]
      return R[0]
```

Trunc funkcija zaokružuje izlaz na željenu duljinu.

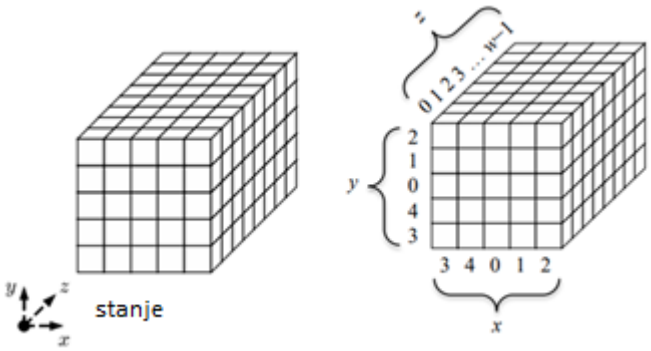
Funkcija ι (A, i_r) definirana je ulazom stanja područja A i indeksom rundi i_r (eng. *round index*), te izlazom novog stanja područja A' :

Za sve trojce (x, y, z) za koje vrijedi $0 \leq x < 5$, $0 \leq y < 5$ i $0 \leq z < w$,
neka $A'[x, y, z] = A[x, y, z]$
$RC = 0^w$
for j from 0 to l, let $RC[2^j - 1] = rc(j + 7i_r)$
za sve z-ove za koje vrijedi $0 \leq z < w$, let $A'[0, 0, z] \otimes RC[z]$

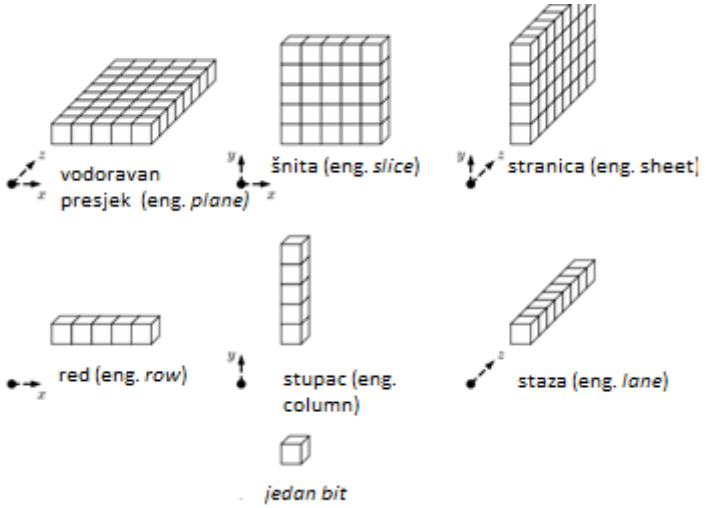
return A'

RC predstavlja konstantu runde ili eng. *round constant* koja određuje l + 1 broj bitova vrijednosti staze.

rc predstavlja linearnu funkciju temeljenu na linearnom povratnom registru pomaka ili eng. *linear feedback shift register* te on generira spomenute l + 1 bitove. (FIPS, 2015; Dinur, Dunkelman, Shamir, 2012; Bertoni, Daemen, Peeters, Assche, 2010)



Slika 4 Trodimenzionalni prikaz vrijednosti Keccak algoritma (Prema: FIPS, 2015)



Slika 5 Dijelovi područja stanja (eng. state array) (Prema: FIPS, 2015)

3.2. Elektronički potpis

Revolucijom digitalnog pohranjivanja podataka kao i komunikacijske tehnologije, digitalne informacije danas mogu se lako pohranjivati, kopirati ili biti promijenjene u svrhu zlouporabe. Sva ova svojstva vrlo su korisna kada se govori o sigurnosti informacija i transakcija unutar sustava lanca vrijednosti, stoga je od velike važnosti pripaziti na digitalizaciju kada se govori o privatnosti, autentičnosti i cjelovitosti podataka koji u slučaju kriptovaluta najčešće imaju veliki novčani značaj. Sprječavanje neovlaštenog pristup i krađa podataka najbitnija je u sferama računa i dokumenata bitnog značaja. Konvencionalni potpisi su zastarjeli i predstavljaju sigurnosni problem iz razloga što su uključeni u dokument kao dio samog dokumenta. Rješenje tih sigurnosnih problema je elektronički potpis. Potpis se mora biti dokaziv, to jest, ako je njegova valjanost upitna i ne zna se tko je potpisao dokument, treća stranka mora moći riješiti spor bez zahtijevanja tajne informacije, odnosno privatnog ključa potpisnika. Korištenjem elektroničkog potpisa u bilo kojem trenutku može se provjeriti autentičnost sadržaja, kao i identitet samog potpisnika, te se time osigurava valjanost upravo te transakcije.

Sigurnosni zahtjevi koje elektronički potpis treba poštovati su:

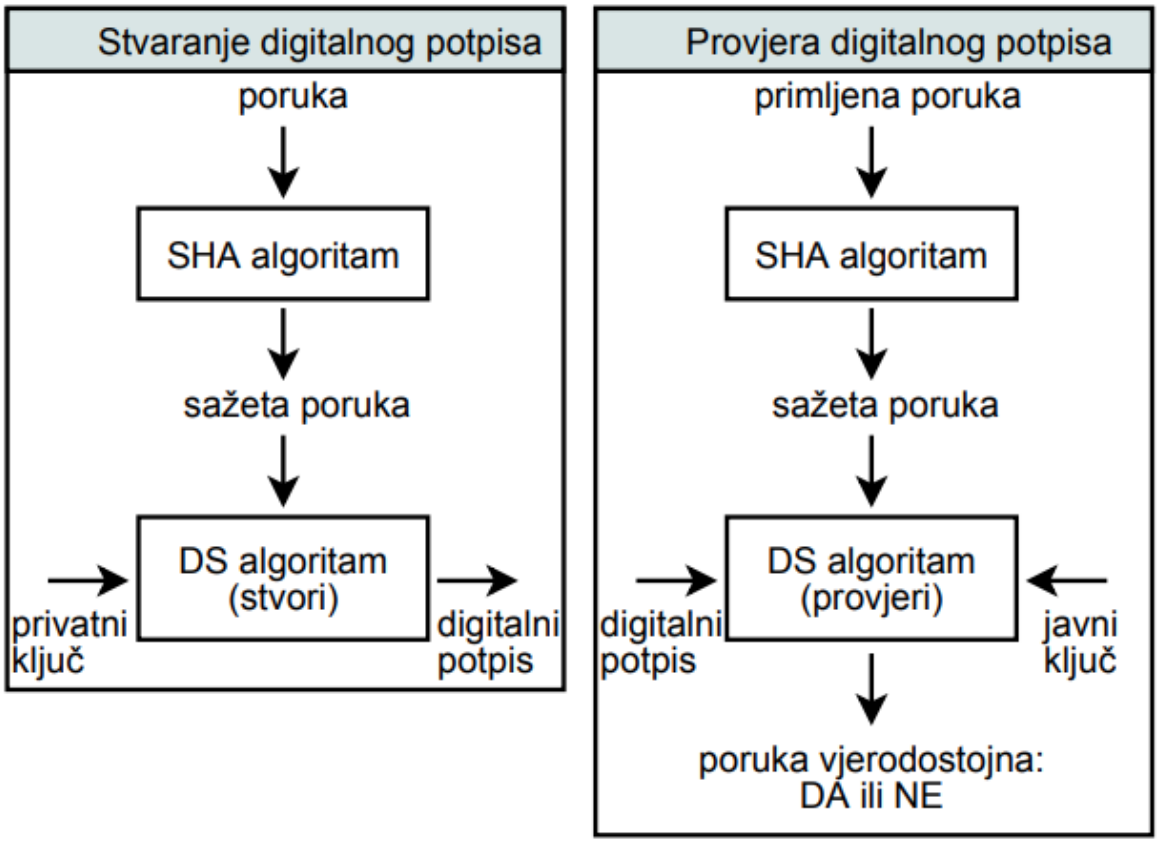
• dokaz o podrijetlu (eng. *Proof of Origin*) - primatelj poruke mora biti siguran u podrijetlo datoteke.

• integritet sadržaja poruke (eng. *Message Integrity*) - primatelj mora biti siguran da datoteka nije presretana i izmijenjena u svrhu prevare tijekom procesa slanja i/ili primanja.

• neporecivost (eng. *Non-Repudiation*) - niti jedna stranka uključena u transakciju ne može u nekom kasnijem vremenu zanijekati uključenost u tu transakciju. (DocuSign, bez dat.)

Kao što je već rečeno, prilikom stvaranja elektroničkog potpisa koristi se par ključeva koji se sastoji od privatnog i javnog ključa. Privatni ključ zna samo onaj korisnik koji ga posjeduje, što onemogućava krivotvorenje potpisa, dok je javni ključ, kao što mu i sam naziv kaže, javan i svima dostupan. Svrha transparentnosti javnog ključa je ta da svi ostali korisnici mogu provjeriti valjanost para ključeva uz pomoć javnog ključa te time osigurati da je osoba s kojom provode transakciju također poštena. Podaci, odnosno same datoteke koje se obilježavaju ili „potpisuju“ elektroničkim potpisom ostaju nepromijenjene, a pokušaj izmjene sadržaja ili samog potpisa rezultira vidljivom promjenom u izgledu hash vrijednosti. U samom postupku stvaranja elektroničkog potpisa za dobivanje sažetka poruke (eng. *message digest*) odnosno provođenja procesa haširanja koristi se sigurna jednosmjerna funkcija spomenuta u prethodnom poglavlju, SHA256 (eng. *Secure Hash Algorithm 256*) algoritam. Iz dobivenog sažetka DS algoritmom stvara digitalni potpis, te se poruka zajedno s pripadajućim potpisom, šalje primaocu koji uz pomoć pošiljateljevog javnog ključa utvrđuje vjerodostojnost poruke i

samog digitalnog postupka. U postupku provjere važno je koristi identičan algoritam jednak onom koji se koristio prilikom stvaranja elektroničkog potpisa, što bi u slučaju provođenja Bitcoin transakcija bio SHA256. Na slici 6. prikazani su postupci stvaranja i provjere elektroničkog potpisa te je važno napomenuti da autori slike koriste drugi naziv za elektronički potpis, a to je digitalni potpis. (CARNet, 2007.)



Slika 6. Stvaranje i provjera elektroničkog potpisa (Izvor: CARNet, 2007)

3.3. Rudarenje kriptovaluta

Nakon što je transakcija poslana od strane dogovora između platitelja i primaoca, te kriptirana spomenutim algoritmima kriptiranja, ulazi u mrežu lanca vrijednosti. Ona još ne postaje dio zajedničke „knjige“ (eng. *blockchain*) sve dok se ne potvrdi njena valjanost i ne uključi u sam blok transakcija. To transakcija ostvaruje nakon procesa rudarenja ili kopanja kriptovaluta odnosno eng. *cryptocurrency mining*. Sustav kriptovaluta temelji se na povjerenju, stoga je lako zaključiti da rudarenje predstavlja jedan od najvažnijih procesa u cijeloj mreži. Niz transakcija spaja se u blokove transakcija koji zahtijevaju ogromnu količinu računanja kako bi se dokazale i potvrdile, ali samo će mala količina izračuna biti uspješno potvrđena i dodana u lanac vrijednosti. Rudarenje ima dvije glavne svrhe kod kriptovaluta. Prva svrha je kreiranje novih novčića kriptovalute, što je nešto kao ispisivanje novca. Većina kriptovaluta ima ograničenu količinu koja se može proizvesti (ili je već proizvedena), što znači da im se ponuda svakog dana smanjuje, kako bi se povećala potražnja i spriječila moguća inflacija. Druga svrha rudarenja kriptovaluta je kreiranje već spomenutog povjerenja na način da mreža osigurava potvrđivanje transakcije samo ako je dovoljno računalne moći posvećeno jednom bloku. Povećanje u broju kreiranih blokova znači više računanja što u totalu znači više povjerenja. (Antonopoulos, 2016.)

S obzirom na to da se težina rudarenja eksponencijalno povećava s vremenom, ljudi su od korištenja svog starog laptopa prešli na nešto veće i zahtjevnije, te će vrste i načini rudarenje biti objašnjeni u nastavku.

Prvi i najjednostavniji način rudarenje kriptovaluta je pojedinačno rudarenje ili samostalno rudarenje (eng. *solo mining*) koje se prvi puta pojavilo 2009. sa samom pojavom prve kriptovalute Bitcoin. Radi se o neovisnom rudarenju temeljenom na vlastitoj opremi, koje ne zahtijeva kolektivno sudjelovanje u bazenima/grupama (eng. *mining pools*). Prvi rudari koristili su procesorsku moć (*Central processing unit – CPU*) za svoja prva iskapanja, te se zatim prešlo na korištenje grafičkih komponenti računala (eng. *Graphics processing unit – GPU)* kako bi se rudarile nove kriptovalute. Uz to, svoju primjenu našao je i novi hardver odnosno programirljivi logički sklopovi (eng. *Field-programmable gate array – FPGA)* koji nisu uvijek postizali mnogo veće brzine rudarenje, ali su postizali znatno smanjenu potrošnju električne energije, koja je iznosila do čak 100 puta veću uštedu u usporedbi s rudarenjem uz pomoć CPU-a. Velika snaga opreme važna je za samostalno rudarenje jer je potrebna velika količina pokušaja kako bi se riješila računalna složenost zadatka. Samostalno rudarenje je korisno za iskapanje novih kriptovaluta, čija vrijednost nije velika u trenutku iskapanja, ali to

naknadno može rezultirati popularizacijom iste. (polygant.net, bez.dat.; Courtois, Grajek, Naik, 2014.)

Drugi način rudarenja kriptovaluta je međusobno udruživanje i suradnja u takozvanim bazenima rudara (eng. *mining pools*). Bazen predstavlja server koji kombinira računalne snage svakog od pojedinih rudara, kako bi stvorio veliku i efikasnu mrežu. Zajedničkim grupiranjem šansa da će baš njihov bazen iskopati novi blok eksponencijalno raste što je više članova bazena. Ovakav način rudarenja je najpopularniji u današnje vrijeme, dok je pojedinačno rudarenje prije bilo popularnije. Na taj način nastaje zajednička računalna mreža koja se bavi stvaranjem novih blokova, naravno u svrhu zarade od nagrada iskapanja. Rudarski novci, odnosno nagrada, podijeljeni su međusobno među rudarima, prema vlastitom udjelu u procesu, odnosno pruženoj računalnoj moći tijekom procesa iskapanja. Udruživanje je postalo potrebno zbog sve veće složenosti mreže popularnih kriptovaluta, što pokazuju i brojke koje kažu da je na vrhu popularnosti 2010. godine u mreži sudjelovalo 67 911 aktivnih rudara, dok je 2019. godine zabilježen podatak od samo 144 aktivnih rudara. (Han, Foutris, Kotselidis, 2019.) Ta činjenica je vjerojatno i usko povezana sa smanjenjem količina nagrade koje se dobivaju prilikom iskapanja jednog bloka transakcija.

Već 2013. godine, opća populacija dolaze na ideju stvaranja super računala čija će svrha isključivo biti rudarenje kriptovaluta, te tako nastaje pojam ASIC rudarenje (eng. *ASIC mining*) koji koristi integrirane sklopove za specifičnu primjenu (eng. *Application-specific integrated circuit)*. ASIC rudarenje je u osnovi proces rudarenja kriptovalutama poput Bitcoina koristeći ASIC platforme. ASIC rudari ulaze u zajedničke bazene te time stvaraju još veću konkurenciju ostalim rudarima. Loše karakteristike udruživanja mogu biti neisplativost zbog prevelikog broja rudara koji dijele istu nagradu ili pak velika ulaganja u neuspješna i neefikasna iskapanja. (polygant.net, bez.dat.; coinmarketcap.com, 2021; Xu, Bai, Hu, Tian, Wu, 2021; Courtois, Grajek, Naik, 2014.)

Posljednji način iskapanja kriptovaluta je iskapanje u oblaku (eng. *cloud mining*). Kod ovog načina iskapanja unajmljuje se računalna moć od usluge koja nudi svoje resurse. Ovaj način je u suprotnosti s prijašnjim dvjema metodama rudarenja koje koriste vlastitu opremu. U ovom slučaju više nije potrebno pokretati vlastite projekte i održavati ih, već potencijalni rudar plaća usluge rudarenja onoliko vremena koliko on hoće, te pri tome snosi puno manje rizika od ostala dva načina rudarenja kriptovaluta. Mogući rizici koje korisnici iskapanja u oblaku izbjegavaju su rizici od pregrijavanja i propadanja hardvera korištenog za rudarenje, te ušteda na prostoru i električnoj energiji. Iskapanje u oblaku omogućuje uključivanje u proces rudarenja bez velikih ulaganja, ali sa sobom ipak nosi i loše strane. Smanjena profitabilnost po iskapanju prvi je problem s kojim se korisnik susreće, iz razloga što skupo plaća sve prednosti iskapanje u oblaku, što rezultira naravno manjim profitom nego kao na primjer kod samostalnog rudarenja. Manjak kontrole nad akcijama koji se odvijaju u procesu

rudarenja rezultira time da je tržište rudarenja u oblaku prepuno prijevara zbog manjka regulacija od strane država te nesigurnih kompanija koje žele doći do brze zarade nudeći nikakvu sigurnost ako stvari pođu po krivu. (polygant.net, bez.dat.; Schoeman, 2021; Courtois, Grajek, Naik, 2014.)

3.4. Dokaz o radu

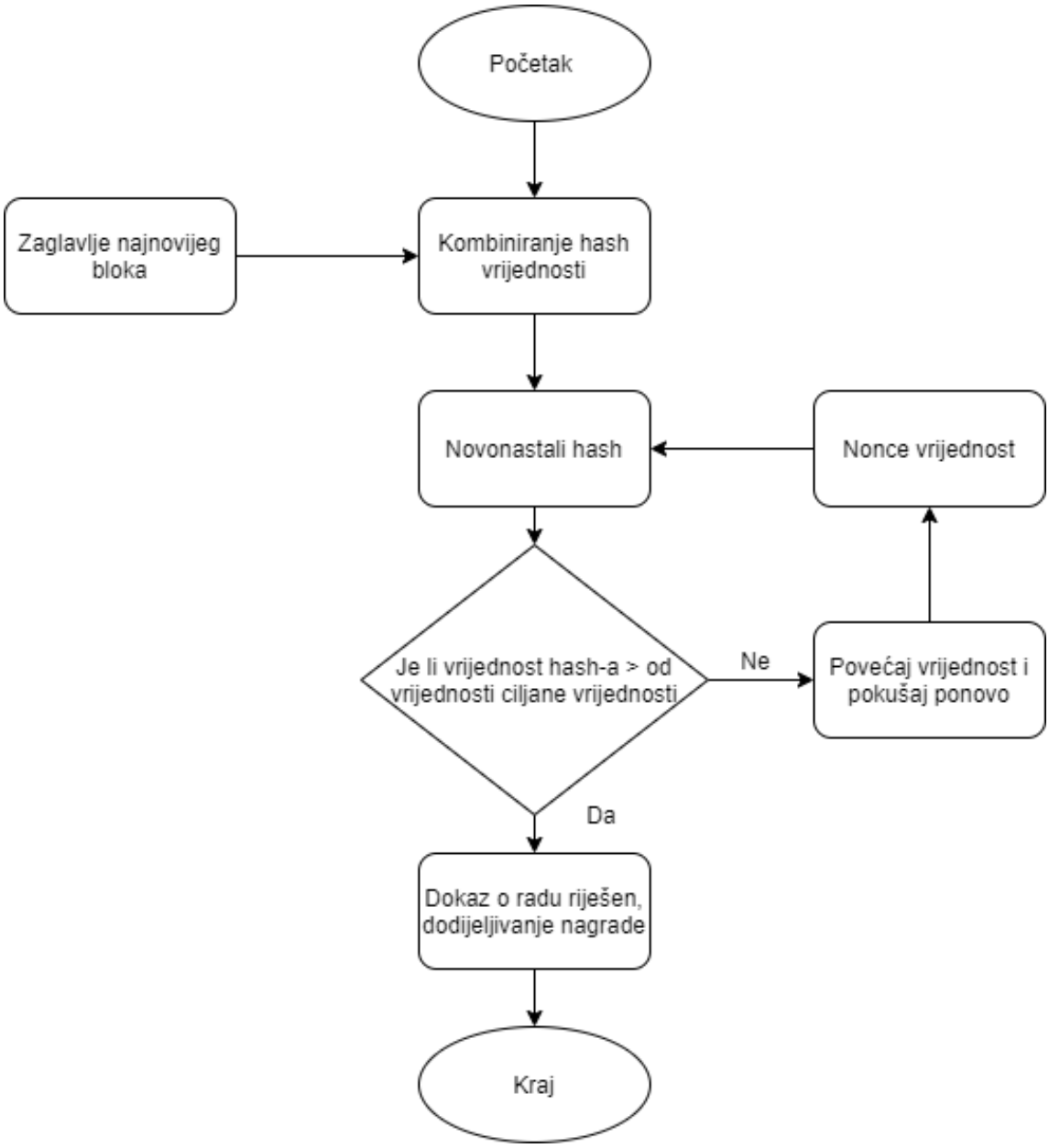
Sada kada je definirana podjela i načini rudarenja kriptovaluta potrebno je i definirati kako kriptovalute uopće bivaju rudarene.

Svi entiteti u decentraliziranoj mreži lanca vrijednosti moraju se složiti oko svih transakcija u mreži, što znači provjeriti valjanost lanca vrijednosti i utvrditi hoće li se transakcija dodati u lanac ili ne, te koji blok transakcija dodati sljedeći. Kada je riječ o Bitcoinu, svi se entiteti moraju dogovoriti oko povijesti transakcija jer decentralizirane mreže nemaju centralno tijelo za rješavanje takvih sporova, stoga niti sudionici mreže nemaju niti mogu imati povjerenja među sobom. Izazov je ovdje kako postići da se svi ti entiteti mogu dogovoriti o ispravnosti evidencije podataka i kako postići dogovor i zadovoljstvo sviju, a da sama sigurnost nije narušena.

Rješenje ovog problema nastalo je 1993. godine idejom Dwork-a, Back-a i Finney-a koji su svoju ideju prvobitno smislili kao protumjeru za napad od uskraćivanja usluge (eng. *denial of service* - DoS). (Judmayer, Aljosha, 2017) Nadalje, svatko tko želi sudjelovati u dodavanju novih blokova u mrežu, odnosno provjeravati valjanost drugih blokova dodanih u mrežu, treba dati nekakav ulog. Ulog je neka vrsta vrijednosti koju validator odnosno „potvrđivač“ mora iznijeti, što ih obeshrabruje od neiskrenog postupanja i pokušaja manipulacije mreže. U nastavku teksta umjesto izraza „potvrđivač“ bit će korišten izraz „validator“. Ako se validatori ipak odluče pokušati varati, izgubit će svoj ulog. Općenito govoreći, primjeri uloga uključuju računalnu snagu, kriptovalutu ili čak ugled koji će im ograničiti daljnje radnje na mreži. S druge strane, validatorima se treba ponuditi neka nagrada ako uspiju u svojem poslu. Ta nagrada očituje se naravno u obliku naknade koju plaća korisnik čije su transakcije potvrđene, te ta nagrada odlazi uspješnom validatoru. Taj proces opisan je u prethodnom poglavlju. (Al-Saqqa, 2020.)

Dokaz o radu ili eng. *Proof of Work* je algoritam koji koriste mnogi sustavi lanca vrijednosti u svrhu potvrđivanja svih transakcija u mreži i uvođenje relevantnih blokova transakcija u lanac vrijednosti. Najpoznatije kriptovalute koje koriste Dokaz o radu su Bitcoin, Ethereum, Dogecoin i Bitcoin Cash. Fokus ovog dijela bit će na Bitcoin sustavu jer je on najstariji te iz razloga što ima najviše korisnika. Bitcoin mreža redovito troši mnogo energije zbog potrebe za rješavanjem intenzivnih matematičkih komputacija kako bi pronašla nove

blokove i time potvrdila transakcije koje se žele provesti. Te komputacije za pronalaženje blokova su u osnovi matematičke „zagonetke“ koje samo rudar može pogoditi, a „zadatak“ sudionicima mreže postavio je mehanizam Dokaza o radu. Dokaz u Dokazu o radu predstavlja računalna moć, električna energija, vrijeme i novac uložen u hardver koji provodi te komputacije. Da bi uspješno iskopali blok transakcija, rudar mora hešírati zaglavlje bloka tako da je taj hash manji ili jednak „cilju”. Cilj zahtijeva, da SHA256 hash vrijednost zaglavlja bloka mora biti 256-bitni alfanumerički niz. Cilj rudara je pronaći jednu vrijednost tako da $\mathrm{H}\,(B.N) < \mathrm{T}$, gdje je *B* string koji predstavlja najnovije transakcije, *N* predstavlja nasumičnu vrijednost korištenu samo jednom koja se naziva eng. *nonce.* Znak točke („ . “) predstavlja operator konkatenacije dok *H* predstavlja Bitcoin hash funkciju $\mathrm{H}\,(S) = \mathrm{SHA256}(\mathrm{SHA256}(\mathrm{S}))$. Stoga rudar mora izvršiti velik broj pokušaja mijenjajući vrijednost nasumične vrijednosti (eng. *nonce*-a). Broj neuspjelih pokušaja u sekundi se zove hash stopa ili hash snaga (eng. *hash rate/hash power*). Što se tiče rudara, najvažniji pojam za njih predstavlja pojam brzine rudarenja odnosno eng. *hash rate*. Brzina rudarenja svakog rudara varira ovisno o jačini hardvera koji posjeduje, kvaliteti softvera koji je napravio ili koristi te efikasnosti obaju. Zbog toga je jedino moguće analizirati cjelokupnu „grešnost“ mreže. Sudeći prema bitinfocharts.com, brzina rudarenja cjelokupne mreže trenutno iznosi 91 702 EH/s. „E“ predstavlja mjernu jedinicu za jedan kvintilion hasheva po sekundi, što bi značilo da cijela Bitcoin mreža zajedno izračuna $91\,702 * 10^{22}$ hash vrijednosti po sekundi. Ako je složenost SHA256 algoritma 2^{256}, odnosno $1.16 * 10^{77}$, to bi teoretski značilo da bi za samo eng. *brute force* odnosno slijedno probijanje svih mogućnosti rješenja algoritma trebalo jako puno vremena, točnije $3.92\mathrm{x}10^{57}$ minuta. Postoji i još jedan način napada na cjelokupnu mrežu, ali on zahtjeva da napadač ili organizirana skupina napadača posjeduje barem 51% računalne moći cijele mreže. S obzirom na to da je pojam kriptovaluta sve popularniji i broj samih korisnika sve veći, ostvarivanje tolike moći je gotovo nemoguće, jer se potrebna količina napadača proporcionalno povećava rastom broja korisnika. Dodatno, resursi koji bi bili potrošeni u pokušaju takvog napada, u trenutku pisanja ovoga rada, bili bi mnogostruko veći od resursa koji bi napadači dobili zauzvrat, stoga je lako zaključiti zašto se takav napad do sada još nije dogodio. (coinsutra.com, 2019; Karl, O'Dwyer, Malone, 2014.; stackexchange.com, 2016.)



Slika 7. Proof of Work flow dijagram (Prema: Ghimire, Selvaraj, 2018)

3.5. Dokaz o ulogu

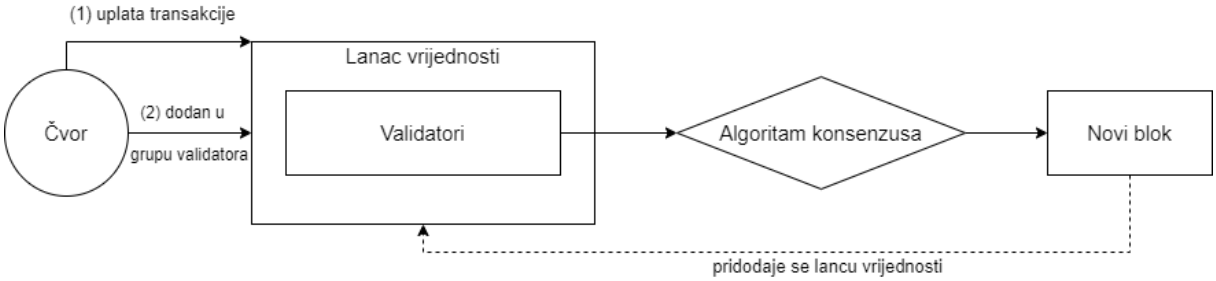
Dokaz o ulogu ili eng. *Proof of Stake* predstavlja još jedan algoritam s ciljem postizanja konsenzusa/dogovora između sudionika decentralizirane mreže. Dokaz o ulogu uveden je 2011. godine nakon objave ideje na Bitcointalk forumu kako bi se riješili glavni problemi najpoznatijeg algoritma „Dokaz o radu“. (bitcointalk.org, 2011.) Iako ova dva algoritma dijele isti cilj postizanja konsenzusa u lancu vrijednosti, procesi za njegovo postizanje prilično su različiti. Dokaz o ulogu koristi pseudoslučajni (eng. *pseudorandom*) proces odlučivanja koji odabire sljedećeg validatora novog bloka koji je temeljen nizom faktora. Sudionik mreže odnosno čvor (eng. *node*) sudjeluje u postupku ostvarivanja konsenzusa (generiranjem odnosno potvrđivanjem sljedećeg bloka) proporcionalno ulogom koji unosi u mrežu. Što je ulog veći, to je šansa veća da će taj čvor postati sljedeći validator. Čvorovi se ne moraju međusobno natjecati u rješavanju kompliciranih matematičkih problema kako bi transakcije bile potvrđene, niti rudariti nove novčiće, što automatski rezultira štednjom električne energije. S obzirom na to da se novčići ne rudare tradicionalnim načinom kao što je slučaj kod kriptovaluta koje koriste Dokaz o radu, kriptovalute koje se temelje na Dokazu o ulogu započinju svoj rad prodavanjem inicijalno odnosno prethodno iskopanih novčića kriptovalute ili pak krenu s radom koristeći Dokaz o radu, te se kasnije prebace na Dokaz o ulogu. To znači da u čvorovima ne postoji uloga rudara ili kopača, već samo validatora. Kao poticaj da čvorovi/validatori ostanu lojalni, validator dobiva nagradu za dodavanje bloka u lanac vrijednosti. Ta nagrada proizlazi „iz džepa“ platitelja koji želi da mu transakcija bude potvrđena unutar mreže, te zbog toga plaća naknadu (eng. *transaction fee*) pri provođenju svake transakcije. Takav pristup plaćanja naknada može se vidjeti i u online platnom sistemu kao što je na primjer Paypal. Sam proces potvrđivanja transakcija ne uključuje „rudarenje“ stoga se naziva „kovanje“ (eng. *forging*) ili eng. *minting*, što reprezentira imaginarni proces kovanja stvarnih novčića.

Kao što je već rečeno, da bi čvor postao dio mreže odlučivanja, mora u mrežu staviti ulog u obliku traženu količine novčića kriptovalute, koje onda ne može koristiti u daljim aktivnostima. No, tu se postavlja pitanje, kako osigurati raznolikost u biranju validatora, u tom smislu da validatori ne budu samo bogate jedinke koje validiranjem dobivaju nagrade i time postaju još bogatiji i ostvaruju još veću šansu da budu izabrani kao sljedeći validatori blokova.

Zanimljiva činjenica je da je Ethereum 2.0 u izradi već godinama gdje će u uporabi biti prilagođeni oblik Dokaza o ulogu. Tako na primjer Peercoin, prva kriptovaluta koja je uopće uvela pojam „Dokaz o ulogu“ 2012. godine, i Blackcoin, koriste malo sofisticiranije načine određivanja sljedećih validatora. Naime, oni koriste starost uloga koji ulazi u jednadžbu za

izračun sljedećeg validatora. Starost uloga se izračunava umnoškom broja dana kojih je ulog bio uložen i brojem novčića koji su uloženi. Nakon što čvor izvrši kovanje novog bloka, njegova starost uloga vraća se na nulu. Na taj način, stalni aktivni korisnici mogu biti nagrađeni za svoju vjernost, a velikim i bogatim čvorovima se ograničava dominacija. Još jedan od postojećih poštenijih načina odabira validatora naziva se nasumični odabir blokova ili eng. *random block selection* koji je temeljen kombinacijom najvećeg uloga i najniže hash vrijednosti. S obzirom na to da su ulozi javni, mreža u nekim slučajevima može predvidjeti sljedećeg validatora. Od novijih prijedloga poboljšanja tu je i prijedlog da se pri određivanju validatora odrede i zamjenici validatora koji mogu „uskočiti“ ako je originalni validator napravio grešku ili uopće nije obavio svoj „posao“. Bilo kako bilo, svaka kriptovaluta ima svoj vlastiti skup pravila za koji oni misle da su poštena i efikasna.

Nakon što čvor bude odabran za kovanje novog bloka, njegova zadaća je da provjeri i validira transakcije unutar bloka, potpisuje blok svojim javnim ključem te pridodaje potvrđeni blok u lanac vrijednosti. Kao nagradu, validator dobiva već spomenutu nagradu. Ako iz nekog razloga čvor želi prestati biti validator/kovač, njegov ulog i sve povezane nagrade bit će zaključane te će sredstva kroz neko vrijeme polako biti otključana i vraćena čvoru. Na taj način mreža ima vremena provjeriti prijašnje potvrđene blokova. Ovaj mehanizam kreiran je sa svrhom sprječavanja napadača validatora koji su odobrili lažne transakcije s ciljem ostvarenja brzog profita i „bježanja“ s ukradenim sredstvima. U slučaju da mreža otkrije nelegalne radnje, čvor koji je zaslužan za te radnje gubi dio ili sav svoj ulog kao i pravo da bude ponovno izabran kao validator. To znači da dok je god ulog veći od potencijalne nagrade, napadačima neće biti isplativo pokušati bilo kakve radnje. Da bi čvor napadač uspio potvrđivati lažne transakcije, potrebno mu je posjedovanje od 51% sveukupne količine cirkulirajućih novčića određene kriptovalute. Odnosno, za posjedovanje 51% cirkulirajuće količine na primjer PeerCoin-a, u vrijeme pisanja ovoga rada, bilo bi potrebno uložiti 12.734.478 milijuna dolara, što je neisplativo vrlo skupo samo po sebi. Glavne prednosti korištenja algoritma Dokaza o radu su energetska učinkovitost i sigurnost što rezultira u većoj decentraliziranosti jer je samo sudjelovanje u mreži lakše i pristupačnije. Veća stopa decentraliziranosti očituje se i u činjenici da je zastupljenost kreiranja bazena (eng. *mining pools*) smanjena ili je nema, dok je u Bitcoin mreži to postala neka vrsta normale. Uz to, potvrđivanje transakcija nije usko povezano s kreiranjem novih novčića kriptovalute što znači da fluidnost vrijednosti određene kriptovalute može biti manja. (Lepore, Ceria, Visconti, Rao, Shah, Zanolini, 2020.; Wang, Huang, Wang, Chen, Zhang, He, 2020.; Proof of Stake, 2020.; bitcointalk.org, 2011.; peercoin.net, 2012.)



Slika 8. Grafički prikaz kreiranja blokova kod Proof of Stake (Prema: Bez, Fornari, Vardaneg, 2019)

3.6. Parametri rudarenja kriptovaluta na primjeru Bitcoina

Kao što je već spomenuto, Bitcoin rudar dio je mreže vjernosti (eng. *peer-to-peer*) koja prikuplja sve podatke provedenih transakcija i ima cilj upotpuniti i izvršiti Dokaz o radu (eng. *Proof of Work*). Dokaz o radu zahtjeva od rudara da ispuni uvjete mreže. Njihov prvobitni cilj je trenutna ciljna vrijednost *T* (eng. *current target value*), koja predstavlja ciljanu vrijednost koju sudionici mreže, u ovom slučaju rudari, žele dobiti. Ako je vrijednost hash-a za koju rudar izjavi da je rješenje veća od hash-a ciljane vrijednosti, njegov pokušaj se odbija. Najvažniji dio cijele formule je eng. *nonce* koja predstavlja nasumični broj koji se koristi samo jednom te će se radi jednostavnosti u nastavku teksta koristiti engleski izraz. To bi značilo da je proces rudarenja ustvari pogađanje broja sve dok se ne dođe do rješenja. Takav sličan oblik „pogađanja“ odnosno „napada“ koristi se i u hakerskim napadima probijanja odnosno slijednog pogađanja (eng. *brute force attack*). Pronalazak nonce-a zapravo znači pronalazak/rudarenje bloka. Vrijednost nonce-a uvijek započinje s nizom nula, a u trenutku pisanja ovog rada, njegova vrijednost započinje s 19 nula, što znači da se te vrijednosti ne moraju izračunavati, jer će taj broj ostati nepromijenjen, sve do sljedeće promjene težine rudarenja. Nadalje preostaje pokušati pronaći vrijednosti ostalih znamenaka hash-a. Za izračun preostalog broja znakova hash-a treba znati da je output SHA256 algoritma 256 bitni zapis koji se najčešće zapisuje u heksadecimalnom obliku od 64 znaka. Stoga je broj preostalih znakova iznosi $64 - 19 = 46$, a iz toga složenost pogađanja ostalih vrijednosti iznosi 16^{46} ili $2.451992865 * 10^{55}$. Postotak šanse za pogađanje svih vrijednosti iz prvog pokušaja provođenjem jedne komputacije iznosi malih 0.00000000000000000002% (Eremenko, 2018.).

Kao što je već rečeno, težina rudarenja Bitcoina definirana je još kod samog kreiranja kriptovalute i ona ovisi o ograničenjima vrijednost *T*-a. Dobra strana je ta, što sveukupna težina rudarenja ovisi i o trenutnom broju rudara u mreži i samoj brzini po kojoj oni rudare, te se za nju zna reći samo „težina“ (eng. *difficulty*), a označava se slovom *D*. Međuodnos cilja *T* i težine *D* definiran je formulom $\mathrm{D} = T_{max} / \mathrm{T}$. (Karl, O'Dwyer, Malone, 2014.) U ovoj formuli, T_{max} označava „prvu“ i najveću težinu, koja predstavlja težinu prvog provođenja Bitcoin rudarenja, te je ta vrijednost iznosila „1“, što znači da je njezin heksadecimalni oblik izgleda ovako „0x1d00ffff“ , a veličina joj je 4 bajta. Proces rudarenja mora zadovoljavati uvjete u kojima je pronalazak rješenja moguć, što znači da hash vrijednost mora biti niža ili jednaka od ciljane vrijednosti (eng. *target value*). Ciljana vrijednost u mreži lanca vrijednosti označava se kao „Bits“, ali je vrijednost tog broja kreirana uz pomoć koeficijenta i indeksa. Uzevši na primjer blok broj 690695 koji je u vrijeme pisanja rada trenutni najnoviji blok pridodan lancu vrijednosti, moguće je analizirati njegovu težinu rudarenja. Njegova ciljana vrijednost, odnosno „Bits“ iznosi 387160270. Ako se taj broj pretvori u heksadecimalni, dobiva se vrijednost 0x171398ce. Početni dio, 0x17 označava indeks te je njegova veličina 1 bajt, ostatak izraza, 0x1398ce predstavlja koeficijent veličine ostalih 3 bajta. Važno je naglasiti da 0x dio označava heksadecimalni dio. Nadalje, cilj *T* računa se po formuli $\mathrm{T} = \mathrm{koeficijent} * 2\text{^}(8 * (\mathrm{indeks} - 3))$ te bi izračun u ovom slučaju izgledao ovako: prvo je potrebno heksadecimalne vrijednosti pretvoriti ponovo u decimalni zapis, tada bi indeks bio broj 23, a koeficijent 1284302. Primjenjujući formulu, dobiveni zapis se pretvara u heksadecimalni broj i uspoređuje s hash-om promatranog bloka. Najčešći faktor koji se uspoređuje kod određivanja težine je niz nula s kojim započinje hash bloka, te što je veći broj nula, to je teže iskopati taj blok (što je usko povezano sa svojstvom nonce-a koji kaže da mora počinjati s nizom nula). Drugim riječima, što je manji broj hash-a to je težina veća. (Dong-ha, 2019)

Težina rudarenja *D* za Bitcoin se ponovno računa svakih 2016 iskopanih blokova kako bi se održalo prosječno vrijeme iskapanja od 10 minuta. Kako bi se izračunala nova težina, prvo se uzima ukupno vrijeme potrebno za iskapanje zadnjih 2016 blokova. To bi značilo da je pojam težine rudarenja u Bitcoinu usko povezan s brojem raspoloživih/preostalih novčića kriptovalute, odnosno te su dvije vrijednosti obrnuto proporcionalne. Nova težina odnosi se na cijelu Bitcoin mrežu i traje naravno dok se sljedećih 2016 blokova ne izrudari. Ako je dobivena težina više od četiri puta veća (ili četiri puta manja) od trenutne težine, rezultat se ograničava na četiri puta teže (ili manje). Uzevši to u obzir, lako je izračunati da se podešavanje težine dešava svakih 2 tjedna ili 14 dana, naravno u idealnom slučaju. Trenutno vrijeme iskapanja iznosi 11 minuta i 37 sekundi što je pad od otprilike 14% u brzini, odnosno porast od 14% u težini. S obzirom na to da je u zadnjih 24 sata od pisanja ovog poglavlja izrudareno 122 blokova i ako brzina rudarenja ostane otprilike ista, za novo podešavanje težine bit će potrebno 16.26 dana od dana kada je

težina promijenjena. Ako je trenutni sveukupan broj blokova u mreži 690695 to znači da se od postojanja Bitcoina desilo oko 342 podešavanja težine. (bitinfocharts.com, bez dat.)

Za spomenuto smanjenje i pad u brojkama rudara u trećem poglavlju moguć je i još jedan pokazatelj, a to je smanjenje u dobivenim nagradama nakon uspješno iskopanog bloka. Na početku rada Bitcoin mreže, nagrada je iznosila 50 Bitcoina za svaki iskopani blok, te je tada rudarenje bilo jako popularno, dok u 2021. godini ta ista nagrada iznosi samo 6,25 BTC-a, što je 8 puta manja količina. Satoshi Nakamoto, tvorac Bitcoina, još je na početku kreiranja svoje ideje odredio da će se dobivena nagrada smanjivati za čak pola svoje vrijednosti svakih 210 000 iskopanih blokova. Uzevši u obzir prosječnu brzinu rudarenja od 10 minuta, prepolovljene nagrade bi se trebalo dogoditi otprilike svakih 1459 dana, odnosno svake 4 godine. Zadnje prepolovljene nagrade dogodilo se 11.05.2020. godine što bi značilo da se sljedeće očekuje negdje nakon 09.05.2024. To bi značilo da će do 2040. godine vrijednost Bitcoina vrijediti manje od jednog Satoshi-ja, najmanjeg dijela Bitcoina, koja ima skoro zanemarivu vrijednost. Ta činjenica također sa sobom povlači trend opadanja rudara kriptovaluta, što znači da će njihov broj vrlo vjerojatno nastaviti padati, što zbog povećanja težine, što zbog smanjenja u nagradi, osim ako sama vrijednost Bitcoina ne poraste na brojku koja je za njih isplativa. (bitinfocharts.com, bez dat.; stormgain.com, 2020.)

Block 690695

This block was mined on July 12, 2021 at 1:31 PM GMT+2 by AntPool. It currently has 1 confirmations on the Bitcoin blockchain.

The miner(s) of this block earned a total reward of 6.25000000 BTC ($210,373.69). The reward consisted of a base reward of 6.25000000 BTC ($210,373.69) with an ad·
2338 transactions which were included in the block. The Block rewards, also known as the Coinbase reward, were sent to this address.

A total of 3,911.82442568 BTC ($131,671,188.69) were sent in the block with the average transaction being 1.67314988 BTC ($56,317.87). Learn more about how blocks w

Hash	000000000000000000063ca4c8679e2693ac791dbad75307027fccc2f41d16ff
Confirmations	1
Timestamp	2021-07-12 13:31
Height	690695
Miner	AntPool
Number of Transactions	2,338
Difficulty	14,363,025,673,659.97
Merkle root	9651b23e0fa68c562136dae47e273e97e8d22990de93f032ed983d13b4d0c60e
Version	0x20000004
Bits	387,160,270
Weight	3,993,597 WU
Size	1,521,237 bytes
Nonce	617,829,142
Transaction Volume	3911.82442568 BTC
Block Reward	6.25000000 BTC
Fee Reward	0.15595627 BTC

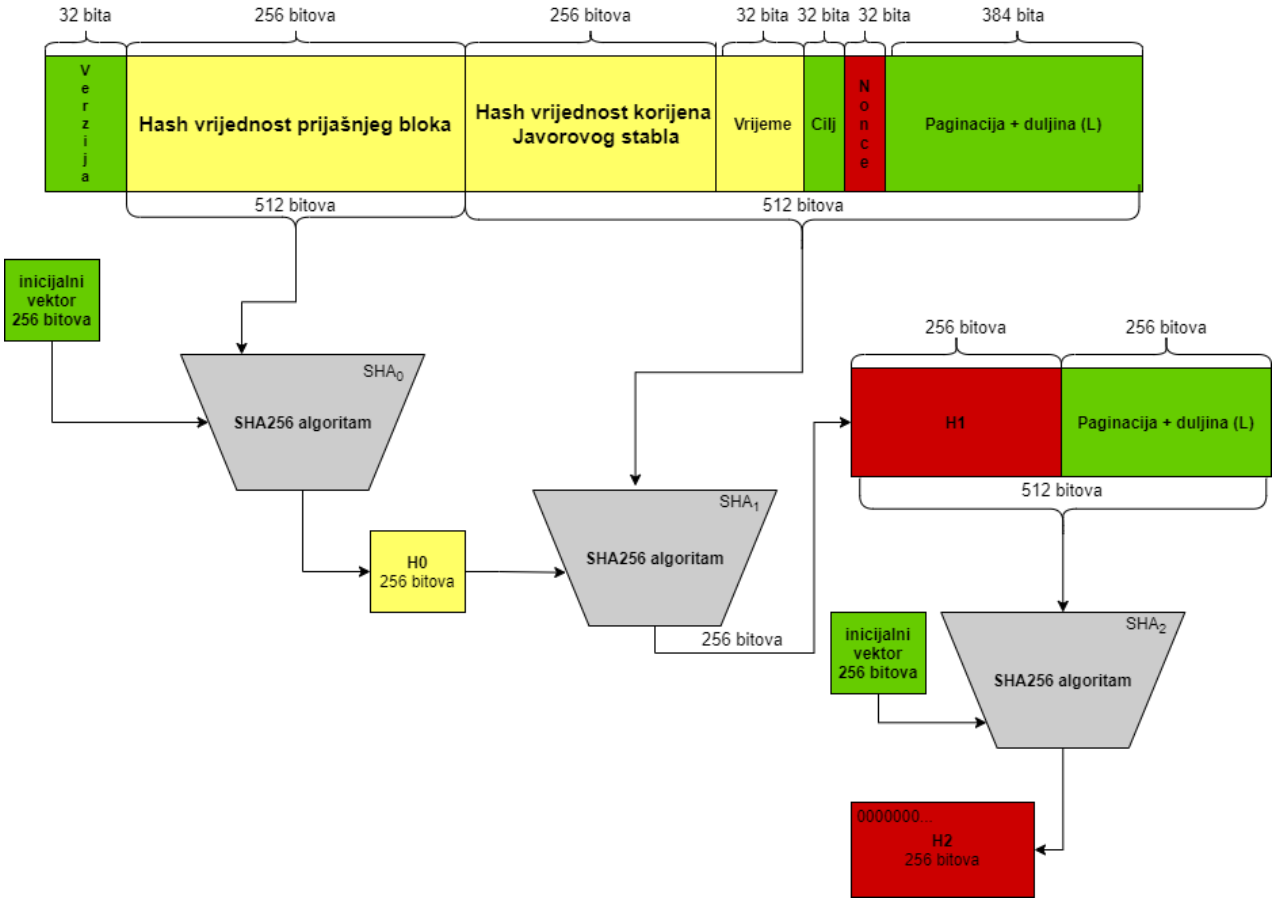
Slika 9. Prikaz informacija o najnovijem generiranom bloku Bitcoina (Izor: blockchain.com, 2021.)

	Bitcoin (Explorer, top100)	Ethereum
Total	18,754,199 BTC	116,663,385 ETH
Price	1 BTC = $ 33,567.17 USD hitbtc: 33,538.63 USD coinbasepro: 33,540.01 USD simex: 33,531.48 USD p2pb2b: 33,543.22 USD bitfinex: 33,528 USD 1 USD = 0.00003 BTC	1 ETH = $ 2,106.23 USD hitbtc: 2,102.55 USD coinbasepro: 2,099.33 USD p2pb2b: 2,103.22 USD kraken: 2,103.57 USD bitfinex: 2,101.6 USD 1 USD = 0.00047 ETH
Market Capitalization	$629,525,332,894 USD	$245,720,258,312 USD
Transactions last 24h	189,087	1,165,912
Transactions avg. per hour	7,879	48,580
Sent last 24h	247,173 BTC ($8,296,909,419 USD) 1.32% market cap	1,687,832 ETH ($3,554,968,039 USD) 1.45% market cap
Sent avg. per hour	10,299 BTC ($345,704,559 USD)	70,326 ETH ($148,123,668 USD)
Avg. Transaction Value	1.31 BTC ($43,879 USD)	1.45 ETH ($3,049 USD)
Median Transaction Value	0.018 BTC ($620.82 USD)	0 ETH ($0 USD)
Block Time	11m 43s	13.4s
Blocks Count	690,709	12,812,904
Blocks last 24h	123	6,430
Blocks avg. per hour	5	268
Reward Per Block	6.25+0.1913 BTC ($216,215.1 USD)	2+0.2675+0.00278+0.07632 ETH ($4,942.4 USD)
Reward last 24h	768.75+23.53 BTC ($26,594,457.53 USD)	12,860+1720+17.88+490.75 ETH ($31,779,647.94 USD)
Difficulty	14.363 T	6.781 P +0.95% in 24 hours
Hashrate	91.702 Ehash/s -15.52% in 24 hours	527.224 Thash/s +1.95% in 24 hours
Mining Profitability	0.29 USD/Day for 1 THash/s	0.0603 USD/Day for 1 MHash/s
Top 100 Richest	2,873,829 BTC ($96,466,307,457 USD) 15.32% Total	

Slika 10. Prikaz podataka Bitcoin i Ethereum mreže (Izvor: bitinfocharts.com, 2021.)

3.7. Primjena SHA256 algoritma u rudarenju Bitcoina

Kao što je već rečeno, rudarski uređaji primjenjuju hešing algoritam na zaglavlje Bitcoin bloka u svrhu pronalaženja novih blokova, a time i za rudarenje i kreiranje novih Bitcoina. Gledajući iz čisto tehničke perspektive, proces rudarenja Bitcoina u osnovi uključuje rudarske uređaje koji kontinuirano računaju trostruki SHA256 hash zaglavlja bloka kako bi dobili izlaz koji će biti prihvaćen od strane Bitcoin mreže. Radi lakšeg reprezentiranja i shvaćanja bit će korištene tri boje: zelena, žuta i crvena čija je svrha objasniti brzinu fluktuacije tih vrijednosti u odnosu na proces rudarenja Bitcoina. Zelena boja označava da će vrijednost ili ostati nepromijenjena do kraja procesa ili barem neko duže vremensko razdoblje. Žuta boja označava da će se vrijednost promijeniti, ali prilično rijetko, to jest relativno nakon nekog vremena. Crvena boja označava da će se ta vrijednost podataka promijeniti najbrže, odnosno nakon svake hash komputacije. Sljedeća slika prikazuje strukturu zaglavlja bloka Bitcoina i način na koji se na njemu primjenjuje trostruki SHA256 algoritam radi dobivanja vrijednosti hash-a koju prihvaća Bitcoin mreža.

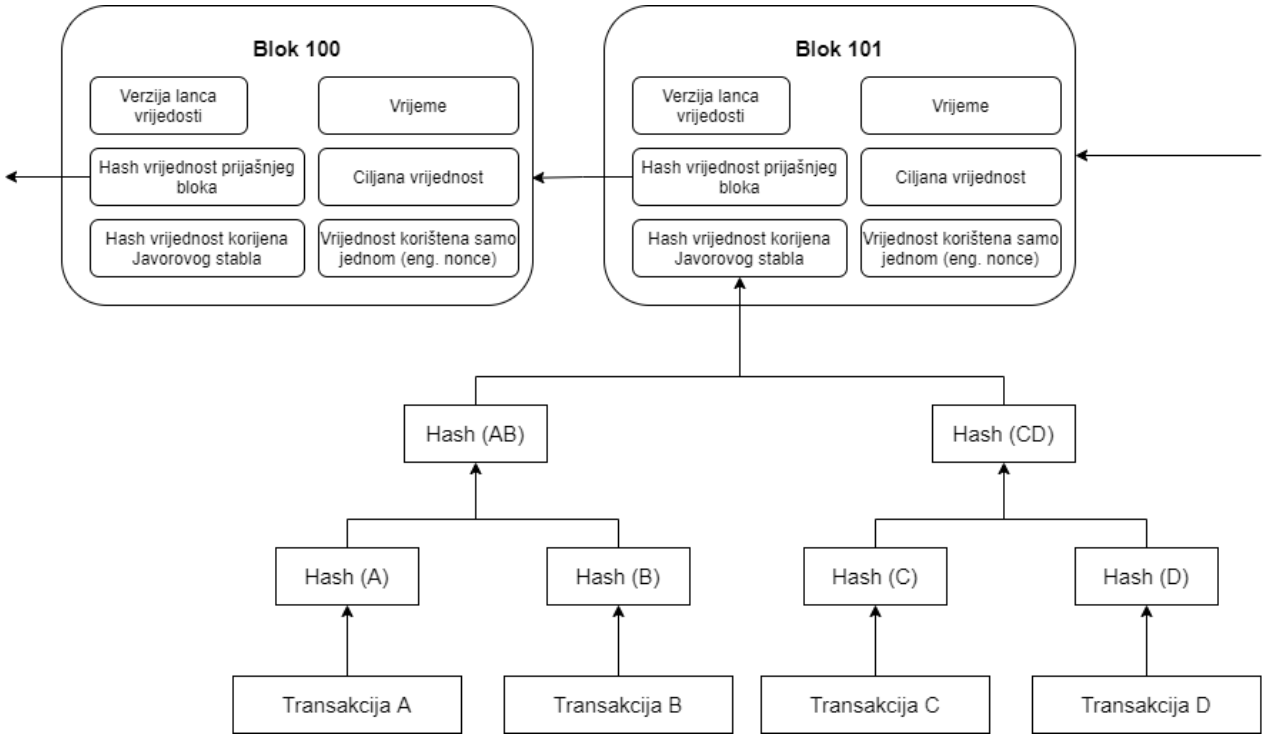


Slika 11. Zaglavlje lanca vrijednosti i proces obrade (Prema: Naik, 2013)

Prva stvar koja je vidljiva iz slike je da zaglavlje lanca vrijednosti mora proći kroz tri različite runde SHA256 algoritma. Zbog toga će svaka od rundi dobiti sufiks, odnosno zvat će se $SHA256_0$, $SHA256_1$ i $SHA256_2$. Radi praktičnosti ulaz pokaznog slučaja veći je od 512 bitova, stoga prolazi kroz dvije primjene ($SHA256_0$ i $SHA256_1$) SHA-a gdje se naravno poruka dijeli na blokove koji se obrađuju jedan po jedan. $SHA256_0$ uzima prvi blok duljine 512 bitova kao svoj ulaz te nakon 64 runde obrade proizvodi prvi dio obrađene poruke koja se naziva H_0 (pogledati poglavlje 3.1.1). Zatim algoritam uzima zadane inicijalne vektore *„IV“* koji su označeni zelenom bojom. Vrijednost H_0 označena je žutom bojom zbog toga što njena vrijednost ovisi o vrijednostima iz samog zaglavlja koje su također označene žutom bojom. Nakon toga $SHA256_1$ uzima H_0 kao svoj inicijalni vektor te uzima sljedećih 512 bitova kao sljedeći ulazni blok. U drugu rundu algoritma ulazi i spomenuti *nonce* koji je označen crvenom bojom stoga se izlaz provođenja $SHA256_1$ također označava crvenom bojom. Taj izlaz H_1 prolazi kroz još jedan krug koji je imenovan SHA_2. SHA_2 uzima 256 bitni blok od H_1 kao svoj ulaz te primjenjuje već spomenuto postavljanje (eng. *padding*) kako bi blok bio željene veličine od 512 bitova. Za razliku od SHA_1, SHA_2 ponovno koristi originalne inicijalne vektore „*IV“* te su zato oni označeni zelenom bojom. Ponovno se nakon 64 runde funkcije kompresije, SHA_2 generira finalni hash H_2 koji se označava crvenom bojom. Zatim slijedi spomenuta provjera zadovoljavanja uvjeta mreže i H_2 vrijednosti te ako je odgovor potvrdan, blok se zajedno sa pripadajućim *nonce*-om dodaje u lanac vrijednosti. (Naik, 2013.)

3.8. Lanac vrijednosti

Lanac vrijednosti ili eng. *blockchain* je nepromjenjiv i distribuiran sustav koji drži evidenciju svih provedenih transakcija od početka njegovog rada. Evidencija svih tih transakcija čuva se na jednom mjestu pod nazivom glavna knjiga ili eng. *ledger*. Glavna knjiga također predstavlja zajedničku knjigu uz pomoć koje se izvršava razmjena podataka ili transakcije među svim sudionicima mreža. Skup transakcija međusobno povezanih na poseban način naziva se blok (eng. *block*). Blok je povezan na njegov prethodni blok pomoću hash vrijednosti te tako više povezanih blokova čini veliku mrežu odnosno lanac blokova. Transakcije iz tih blokova provjeravaju se od strane takozvanih "čvorova" (eng. *nodes*) u mreži, a aktivne čvorove čine već spomenuti rudari. Nakon provjere i potpisivanja od strane čvorova, taj se blok dodaje u lanac i ne može se više mijenjati (zbog svojstva hash funkcija). Lanac vrijednosti osigurava pouzdanost prijenosa podataka na decentralizirani način bez ikakvih uključenja treće strane. Blokovi su raspoređeni u kronološkom redoslijedu, te se prvi blok u lancu spominje kao blok postanka ili eng. *Genesis block* i on nema nadređeni blok (eng. *parent block*). U osnovi, blok se sastoji od zaglavlja bloka, hash vrijednosti njegovog prethodnika i hash-a trenutnog bloka, broja transakcije, vremenske oznake (eng. *timestamp*) i korijena Javorovog stabla (eng. *Merkle root*) kao što je i prikazano na slici 12. Transakcija je pokrenuta čvorom i elektronički potpisanim privatnim ključem. Nakon toga stvara se blok koji predstavlja skupinu transakcija. Taj se blok onda dijeli sa svim čvorovima u mreži kako bi bila dodijeljena grupi čvorova dodjeljuje na provjeru valjanosti, a zauzvrat, uspješni čvorovi primaju nagradu za njihov obavljeni posao, te se provjereni blok dodaje se postojećem lancu. Svi ovi procesi se rade na temelju niza pravila koji su poznati kao protokol konsenzusa (eng. *consensus protocol*) koji je također spomenut. (Sharma, Jain, 2019.)



Slika 12. Grafički prikaz pojednostavljenog lanca vrijednosti (Prema: Xiaomeng, Fan, Shenwen, Jinglin, 2020)

Lanac vrijednosti ima različita obilježja i značajke, zbog čega mu popularnost korištenja naglo raste, a glavne značajke su mu:

• decentralizacija: lanac vrijednosti stvara novi način sustava koja ne zahtijeva središnji čvor ili treću stranku koja pohranjuje ili provjerava bilo koju transakciju zahtijevajući da to učini skup čvorova. Pohrana i ažuriranje podataka događa se na distribuiran i svima vidljivi način.

• nepromjenjivost: transakcija u lancu vrijednosti evidentirana je u decentraliziranom nepromjenjivom zapisu što znači da se ne može promijeniti niti nakon dodavanja. Nakon potvrđivanja transakcije, ona se pohranjuje u već spomenutoj distribuiranoj bazi podataka, što donosi veće povjerenje u transakcije zapise i buduće transakcije.

• transparentnost: korisnici lanca vrijednosti mogu vidjeti svoje podatke i transakcije na transparentan način gdje sve inkriminirajuće informacije ostaju kriptirane. Transparentnost je ključni element lanca vrijednosti koji može biti primjenjiv u mnogo aplikacija kao što su one za opskrbni lanac ili financijske usluge, te se svakodnevno pronalaze nova primjenjiva rješenja.

• anonimnost: omogućuje korisnicima obavljanje anonimnih transakcija, korisnik je povezan s javnom adresom i nitko neće znati njegovo pravo ime ili adresu. Već spomenute kriptografske metode se koriste za prikrivanje korisničkog identiteta.

• na temelju konsenzusa: provjera i dodavanje svakog bloka u lanac vrijednosti postignut je postizanjem sporazuma među svim čvorovima lanca. Ovaj sporazum događa se pomoću konsenzusnog algoritma (npr. Dokaz o radu, Dokaz o ulogu…) koji obuhvaća pravila za validaciju bloka te svaka kriptovaluta definira svoj sporazum po kojem funkcionira.

Sustav lanca vrijednosti može postojati u više različitih oblika gdje se ovisno o načinu implementacije lanca mogu razlikovati različita svojstva i osobine. Postoje tri vrste prema vidljivosti lanca vrijednosti: (1) javni, (2) korporativni i (3) potpuno privatni lanac. Te se vrste dalje mogu svrstati u dvije kategorije ovisno o tome tko ima pristup lancu vrijednosti. To su lanac vrijednosti bez dozvole (eng. *permissionless*) i lanac vrijednosti temeljen na dozvolama (eng. *permission-based blockchain*). U lancu vrijednosti bez dozvole, dozvole pristupa nisu kontrolirane kao što je slučaj u javnom lancu vrijednosti, dok se u lancu vrijednosti baziranom na dozvolama, dozvole pristupa su strogo kontrolirane, što je slučaj kod korporativnih i potpuno privatnih lanaca vrijednosti. U nastavku će biti opisani tipovi lanca vrijednosti prema vidljivosti:

Javni lanac vrijednosti je tip lanca vrijednosti koji je otvoren za cijeli svijet, odnosno svatko može biti dio lanca vrijednosti što znači da može čitati, pisati, slati transakcije i sudjelovati u procesu odlučivanja u kojem svi sudionici imaju istu moć. Iako je javni lanac vrijednosti otvoren, osiguran je kriptografskim algoritmima sličnim drugim vrstama lanca vrijednosti. Dva glavna primjera javnog lanca vrijednosti su Bitcoin i Ethereum.

Korporativni lanac vrijednosti predstavlja tip lanca vrijednosti kod kojeg svi sudionici nemaju dozvolu za pisanje niti istu moć validacija transakcija. Ovaj tip je kontroliran od unaprijed odabranog broja čvorova, te oni mogu kontrolirati broj novih čvorova koji se mogu pridružiti u lanac vrijednosti ili koji mogu potvrditi transakcije. Dozvole za čitanje informacija mogu biti javne ili ograničene. Primjeri ovakvih lanca vrijednosti bili bi Quorum i Corda.

Kod potpuno privatnog lanca vrijednosti broj čvorova u mreži jako je ograničen. To znači da ne može svatko biti dio ove mreže, zbog toga što ima centraliziranu bazu podataka i strukturu. Dodatno, samo jedan subjekt ili organizacija ima dozvolu za pisanje i samo on može donositi odluke i potvrditi transakcije. Dozvola za čitanje može biti javna ili ograničena, ovisno o tome kako je odredilo glavno tijelo. (Al-Saqqa, 2020.)

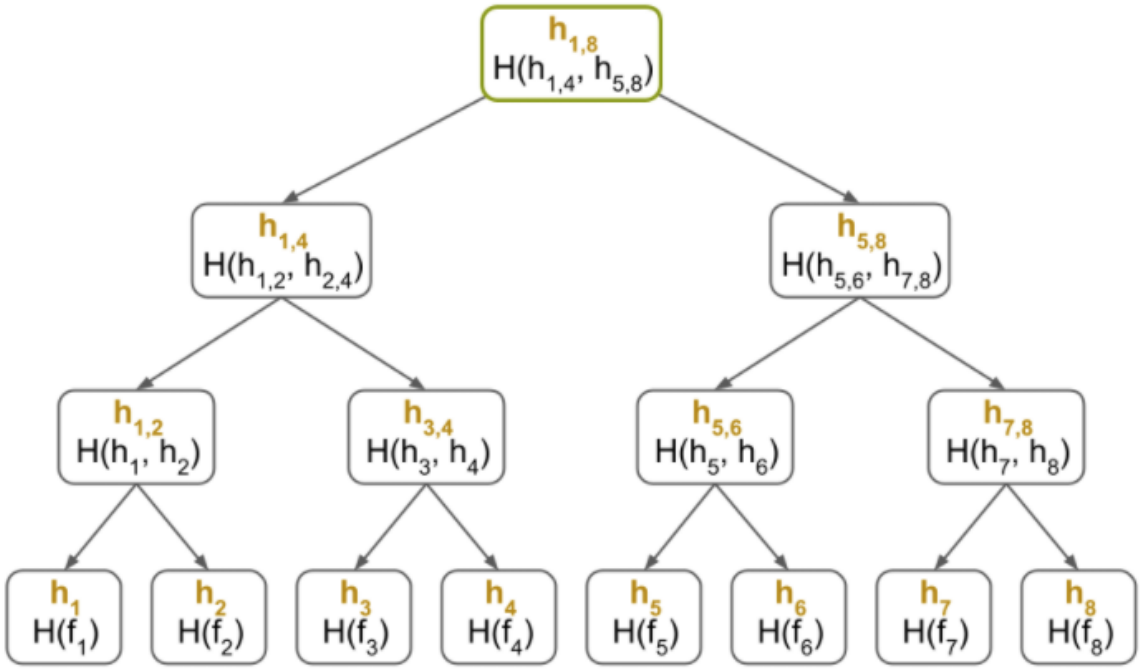
3.8.1. Blok i njegova struktura

Blok je osnovna građevna struktura lanca vrijednosti te nije određeno koliko blokova sačinjava lanac vrijednosti. Prvi nastali blok kreiran je 03.01.2009. kada je Satoshi Nakamoto osmislio Bitcoin sustav, a taj se blok nazivao eng. *Genesis block* što bi u prijevodu značilo nešto od čega je sve poteklo ili kraće, izvor svih blokova. Svaki blok veličine je do 1 MB, a u sebi sadrži hash adresu prijašnje kreiranog bloka veličine 256 bitova, te se ona sprema u zaglavlje bloka (eng. *block header*) veličine oko 80 bajtova. Zaglavlje bloka sadržava još dvije bitne komponente podjele, a to su: osnovne informacije o bloku, vidljive svima te dijelovi korijen Javorovog stabla (eng. *Merkel root hash*). Konkretno, zaglavlje bloka čini broj verzije (eng. *version number*) što je broj koji indicira verziju protokola lanca vrijednosti. Bitno je da svi blokovi imaju definiranu verziju kako ne bi došlo do pokušaja uključivanja blokova koji rade na starijim verzijama. Nadalje, zaglavlje bloka sadrži točno vrijeme generiranja bloka (eng. *timestamp*) što je jako važna stavka kod procesa rudarenja jer se bez te vrijednosti ne bi znalo tko je prvi došao do rješenja. Zaglavlje sadrži i vrijednost težine rudarenja *D* (eng. *difficulty*) te vrijednost nonce-a. Od novijih pojmova u zaglavlju se javljaju i hash vrijednost prijašnjeg bloka u lancu koji služi kao provjera u slučaju da netko pokuša promijeniti neke informacije u novom bloku. Zaglavlje bloka također sadrži hash vrijednost korijena Javorovog stabla. Korijen Javorovog stabla izračunava se na način da se heširaju transakcije unutar bloka, te će taj proces biti detaljnije obrađen u narednom poglavlju. Same transakcije nisu dio zaglavlja bloka, nego se samo pridodaju vrijednosti jednog hasha kako bi se uvijek mogao provjeriti integritet transakcija. Važno je naglasiti da svaki blok sadrži od različiti broj transakcija te se zato blokovi razlikuju i po veličini. (Naik, 2013.; Al-Saqqa, 2020.)

3.8.2. Javorovo stablo

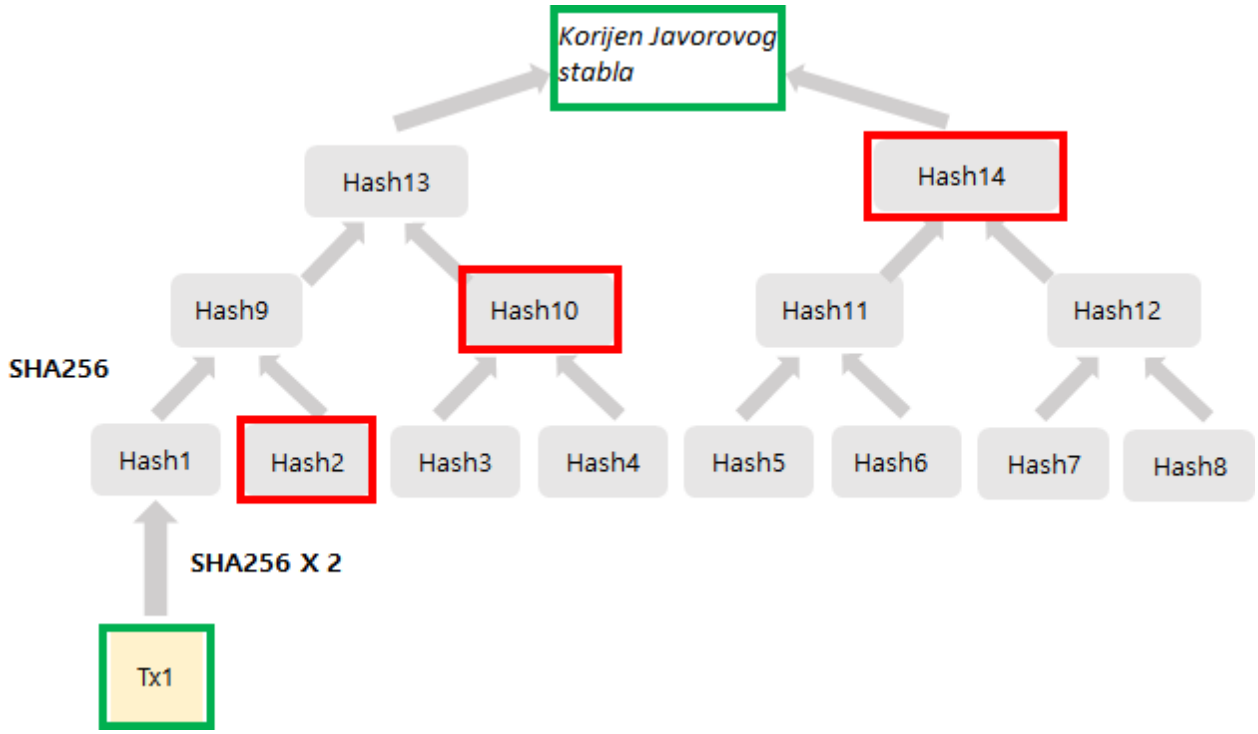
Pojam Javorovo stablo (eng. *Merkle tree*) predstavlja pojam za računalnu znanstvenu strukturu dizajniranu za efikasno dokazivanje o uključivanju. Preduvjeti su složena lista stavki i kriptografska hash funkcija. U slučaju Bitcoina, stavke složene liste su transakcije u bloku, a hash funkcija je SHA256. Za izgradnju Javorovog stabla potrebno je slijediti korake:

1. Heširati sve blokove liste.
2. Ako postoji točno 1 hash u listi, time završava proces.
3. U suprotnom, ako postoji neparan broj hash-eva, treba duplicirati zadnji hash na popisu i dodati ga kraju tako da je sada u listi paran broj hash-eva.
4. Nadalje treba upariti hash-eve po redu i te konkatenirane hash-eve ponovo heširati kako bi se dobili hash-evi roditeljske razine, te bi tada broj hash-eva roditelja trebao biti upola manji nego djece.
5. Nakon toga proces se ponavlja dok se ne dođe do samo jednog finalnog hash-a koji predstavlja cijelu listu. (Song, 2019.)



Slika 13. Grafički prikaz Javorovog stabla (Izvor: Tomescu, 2020.)

Na slici 13. prikazan je grafički prikaz jednog Javorovog stabla, te njegov *n* iznosi 8 što označava broj listova, (eng. *leaves)* odnosno početni broj članova liste. Lista u ovom slučaju označava skup transakcija. Zadnja i najviša vrijednost na vrhu stabla naziva se korijen ili eng. *Merkle root hash* te se baš taj hash sprema u zaglavlje lanca vrijednosti, a ima veličinu od 256 bitova. Kreiranje konstrukcije Javorovog stabla za 2^n objekata kreira $2^n + 2^n - 1 + \cdots + 2 + 1 = 2^{n+1} - 1$ hashev-a što je po vrijednosti vrlo blizu 2n. Sama svrha izvođenja konkatenacije nad dva hash-a i spajanje istih te ponovno heširanje kako bi se dobili roditelji je ta što je onda moguće pružiti dokaz o uključenosti. To znači da se može dokazati da je vrijednost *L* nastala od roditelja *P*, otkrivanjem vrijednosti *R*. Zatim moguće je kombinirati vrijednost *L* i vrijednost *R* kako bi nastao *P* i svejedno imati dokaz o tome da je vrijednost *L* korištena za stvaranje *P*. Ako vrijednost *L* nije nastala od vrijednosti *P*, tvorac stabla mora biti u mogućnosti pružiti vrijednost *R* koja će biti ekvivalent pružanju identične hash vrijednosti (eng. *preimage*), što je vrlo vrlo teško. Identična hash vrijednost je pojam koji se u kriptografiji koristi kao niz svih mogućih ulaza *x* koji daju isti izlaz *y* primjenjujući istu hash funkciju H *(H(x) = y)*. Još jedna od prednosti Javorovog stabla je i mogućnost brzog pretraživanja koja uz pomoć korištenja Javorovog puta ili *eng. Merkle Path*, što označava postupak potvrđivanja je li transakcija sadržana u stablu ili je pak nastala od neke druge transakcije. Sve što je potrebno jest znati korijen Javorovog stabla, odnosno posjedovati informacije o zaglavlju bloka. Na primjer ako osoba A želi provjeriti je li njihova transakcija *Tx1* dio nekog određenog Javorovog drva. Taj se postupak može provjeriti na način da se vlasnikova transakciju *Tx1* dva puta hešira uz pomoć SHA256 algoritma, te se zatim pronalazi vrijednost susjednog hash-a koji se naziva i desni hash (uzevši u obzir da je *Tx1* odnosno Hash1 lijevi hash) koji se nalazi u Javorovom putu. Nadalje, osoba A hešira vrijednosti hash-eva hash-a 1 i hash-a 2 kako bi dobila vrijednost hash-a roditelja. Na slici 14. to bi bili hash-evi 1 i 2 odnosno roditelj bi bio hash broj 9 (uzevši u obzir ako su hashevi/listovi prve razine brojevi od 1 do 8). S obzirom na to da Javorov put sadrži i vrijednost Hash-a 10, susjednog Hash-a 9, osoba A ponovno može dobiti Hash 13. Hash 13 se zatim kombinira s Hash-em 14 koji se također nalazi u Javorovom putu te se tako dobiva Javorov korijen. Na posljetku, osoba A bez problema može usporediti vrijednost dobivenog hash-a i vrijednost očekivanog hash-a korijena kako bi još jednom provjerila valjanost. (Dong-ha, 2019.) (Paris, Schwartz, 2020.)



Slika 14. Grafički prikaz pronalaska korijena Javorovog stabla transakcije Tx1 uz pomoć Javorovog puta (Prema: Dong-ha, 2019.)

4. Usporedba algoritama

Smisao provođenja konsenzusa je postizanje općeg sporazuma između svih uključenih sudionika odnosno čvorova ili blokova u mreži lanca vrijednosti. Kao što je već spomenuto, ono predstavlja samo srce lanca vrijednosti. Primjenom algoritma za konsenzus, lanac vrijednosti pruža pouzdanost i povjerenje u mrežu između anonimnih čvorova u distribuiranom računalnom kontekstu. Naglim razvojem kriptovaluta i sve većom primjenom u svijetu, do sada se razvilo mnogo različitih algoritama za postizanje konsenzusa koji se koriste u sustavima lanca vrijednosti kriptovaluta te će oni u ovom poglavlju biti međusobno uspoređeni. U svrhu komparativne usporedbe bit će korištena neka od obilježja koja najbolje očituju međusobne razlike algoritama te će se ti podaci prikazati kroz povezane tablice koje su podijeljene radi preglednosti.

Tablica 2. Osnovne značajke

Algoritam	***Osnovne značajke***			
	Tip lanca vrijednosti	**Odabir rudara temeljem**	**Ušteda energije**	**Tolerantnost**
Dokaz o radu	Javni	Rješavanje kompleksnog hash-a	Ne	<25% računalne moći
Dokaz o ulogu	Javni	Količina uloga, starost novčića	Djelomična	<51% cirkulirajuće količine kriptovalute
Delegirani dokaz o ulogu	Javni	Količina uloga		<51% validatora
Dokaz o aktivnosti	Javni	Rješavanje kompleksnog hash-a		50% online uloga
Ripple	Javni	/	Da	<20% neispravnosti čvorova
Dokaz o važnosti	Korporativni	Visoki prioritet		<50% važnosti
Dokaz o sreći	Korporativni	Vrijednost sreće		<50% procesorske moći
Praktični problem Bizantskog generala	Privatni	Matematičke operacije		<33.3% replika

(Izvor: Alsunaidi, Alhaidari, 2019.)

Osnovne značajke koje su bile promatrane kod usporedbe ovih algoritama bile su: tip lanca vrijednosti, temelj odabira rudara, ušteda energije algoritama te tolerantnost.

Stupac „Tip lanca vrijednosti“ tablice „Osnovne značajke“ definiran je iz razloga što neki od navedenih algoritama konsenzusa kao što su Dokaz o važnosti, Dokaz o sreći te Praktični problem Bizantskog generala zahtijevaju provjeru identiteta čvorova koji će sudjelovati u procesu rudarenja. Ti su algoritmi prikladni za privatne ili korporativne sustave lanca vrijednosti zbog poteškoća u prepoznavanju čvorova u javnim mrežama. Algoritmi kao što su Dokaz o radu, Dokaz o ulogu ili Dokaz o aktivnosti pogodniji su za javne tipove mreže lanca vrijednosti, gdje se čvorovi mogu slobodno priključiti mreži kada god žele.

Stupac „Odabir rudara temeljem“ tablice „Osnovne značajke“ određuje na temelju kojih svojstava će sljedeći rudar odnosno validator biti izabran. Tako primjerice algoritam Dokaza o radu zahtijeva da rudar uspješno riješi kompleksnu hash funkciju kako bi postao rudar i stvorio novi blok. S druge strane, kod Praktičnog problema Bizantskog generala, proces rudarenja ovisi o razmjeni mnogih elektronskih poruka među čvorovima koji uzrokuju i sudjeluju u općem proračunu kod mrežnih komunikacija.

Stupac „Ušteda energije“ tablice „Osnovne značajke“ predstavlja omjer potrošnje energije ili sredstava pri provođenju i izvršavanju algoritama. Dokaz o radu jedini je algoritam u tablici koji uopće ne štedi na energiji jer se u njegovom radu sudionici međusobno natječu te pri tome koriste ogromne količine električne energije koje pokreću hardver potreban za izvršavanje algoritma. Neefikasna potrošnja energije leži u tome što je samo jedan od rudara mreže lanca vrijednosti koja koristi dokaz od radu nagrađen, odnosno, ostali rudari su električnu energiju zapravo „potrošili bez razloga“. Međutim, količina potrošnje energije kod algoritama Dokaza o ulogu, Dokaza o aktivnosti te Delegiranom dokazu o ulogu je mnogostruko manja nego kod Dokaza o radu, stoga je njihova vrijednost označena kao djelomična. Sama potrošnja kod Praktičnog problema Bizantskog generala, Dokaza o sreći i Dokaza o aktivnosti mnogostruko je manja od potrošnje prije spomenutih algoritama te su sukladno tome ti algoritmi definirani kao algoritmi koji štede energiju.

Stupac „Tolerantnost“ tablice „Osnovne značajke“ definira postotke sredstava koje bi morala posjedovati napadačka strana koja želi manipulirati sustavnom lanca vrijednosti određene kriptovalute. Tako se iz tablice može zaključiti da napadač mora posjedovati više od 25% računalne moći kako bi manipulirao na primjer Bitcoin sustavom ili pak kod Delegiranog dokaza o ulogu više od 51% validatora moraju biti napadači kako bi „preuzeli“ mrežu kriptovalute kao što je na primjer BitShares. (Alsunaidi, Alhaidari, 2019.)

Tablica 3. Poticaj za rudare

Algoritam	*Poticaj za rudare*	
	Transakcijske naknade	**Nagrada za iskapanje**
Dokaz o radu	Da, za sve rudare	Da, za prvog rudara
Dokaz o ulogu	Da, za sve rudare	Ne
Delegirani dokaz o ulogu	Da, za sve sudionike	Da, za izabrane "svjedoke"
Dokaz o aktivnosti	Da, za sve rudare i sudionike	Ne
Ripple	Da	/
Dokaz o važnosti	Da, za partnere u transakciji	Ne
Dokaz o sreći	Ne	Da, za pobjednika
Praktični problem Bizantskog generala	Ne	Ne

(Izvor: Alsunaidi, Alhaidari, 2019.)

Sljedeća tablica „Poticaj za rudare“ sadrži evaluacije koje odgovaraju na glavno pitanje „Koju korist imaju sami rudari i sudionici mreže lanca vrijednosti?“. Dva pojma koja su bila obrađena su transakcijske naknade te nagrade za iskapanje odnosno potvrđivanje transakcije.

Iz stupca „Transakcijske naknade“ tablice „Poticaj za rudare“ vidljivo je da algoritam Dokaza o sreći kao i Praktični problem Bizantskog generala nemaju transakcijske naknade što za korisnike mreže predstavlja vrlo pozitivu i korisnu značajku. S druge strane, Dokaz o radu i Dokaz o ulogu sadrže transakcijske naknade samo za rudare, dok Delegirani dokaz o ulogu i Dokaz o važnosti zahtijevaju od svih sudionika mreže transakcijske naknade. Ripple mreža uopće nema mogućnost rudarenja XRP kriptovalute, ali zahtijeva od korisnika koji vrše transakciju plaćanje minimalnih naknada. (ripple.net, bez dat.; Alsunaidi, Alhaidari, 2019.)

Stupac „Nagrada za iskapanje“ pokazuje da algoritmi Dokaza o ulogu, Dokaza o važnosti i Praktičnog problema Bizantskog cara nemaju nagrade za iskapanje. Razlog tome je što ovi algoritmi određuju sljedećeg rudara ili validatora na potpuno drugačiji način od ostalih algoritama. Na primjer, algoritam Dokaza o važnosti temelji se na kombinaciji Dokaza o radu i Dokaza o ulogu te tako određuje sljedećeg validatora na temeljem broja transakcija odgovarajuće kriptovalute koje obavljaju. (moneyland.ch, bez dat. ; Alsunaidi, Alhaidari,

2019.) S druge strane, Dokaz o radu, Delegirani dokaz o ulogu i Dokaz o sreći nagrađuju posebno izabrane sudionike mreže kako bi potaknuli međusobnu suradnju sudionika. Tako je u slučaju Dokaza o radu, nagrađen rudar koji je prvi pronašao nonce vrijednost novokreiranog bloka u lancu vrijednosti. Kod Dokaza o sreći, pobjednik se dobiva generiranjem nasumične vrijednosti L, ograničenja $\in$ [0, 1), iz ujednačene raspodjele koja se koristi za određivanje pobjedničkog bloka svih iskopanih blokova sudionika. (Milutinovic, He, 2016; Alsunaidi, Alhaidari, 2019.)

Uz to, vidljiva je i poveznica između tipa lanca vrijednosti i smanjenja prisutnosti nagrada i naknada za rudare jer u privatnim mrežama lanca vrijednosti operacije rudarenja ovise o resursima poduzeća, stoga nije potrebno nagrađivati rudare. U suprotnosti s time, u javnim mrežama lanca vrijednosti ova pojava je neophodna kako bi se osiguralo sudjelovanje u mreži te kako bi se financijski nadomjestila ulaganja u opremu koju koriste rudari te električnu energiju koju ta oprema troši. (Alsunaidi, Alhaidari, 2019.)

Tablica 4. Opseg i rad algoritma

<table>
<tr><th rowspan="2">Algoritam</th><th colspan="5">Opseg i rad algoritma</th></tr>
<tr><th>Brzina potvrđivanja</th><th>Propusnost (transakcije/sekunda)</th><th>Brzina kreiranja bloka</th><th>Skalabilnost</th><th>Proširivost</th></tr>
<tr><td>Dokaz o radu</td><td>>100s</td><td><100</td><td>Spora</td><td rowspan="7">Jaka</td><td rowspan="2">Ne</td></tr>
<tr><td>Dokaz o ulogu</td><td><100s</td><td><1000</td><td rowspan="8">Brza</td></tr>
<tr><td>Delegirani dokaz o ulogu</td><td><100s</td><td><1000</td><td rowspan="5">Da</td></tr>
<tr><td>Dokaz o aktivnosti</td><td>nepoznato</td><td>nepoznato</td></tr>
<tr><td>Ripple</td><td><10s</td><td><1000</td></tr>
<tr><td>Dokaz o važnosti</td><td>nepoznato</td><td>nepoznato</td></tr>
<tr><td>Dokaz o sreći</td><td>>15s</td><td>nepoznato</td></tr>
<tr><td>Praktični problem Bizantskog generala</td><td><10s</td><td><2000</td><td>Slaba</td><td>Ne</td></tr>
</table>

(Izvor: Alsunaidi, Alhaidari, 2019.)

Učinkovitost je jedna od najvažnijih aspekata kod biranja algoritma, iz razloga što se učinkovitost često smanjuje kada se veličina mreže povećava. Faktori koji će ulaziti u ovu analizu su brzina potvrđivanja, propusnost, brzina kreiranja bloka, skalabilnost te proširivost.

Brzina potvrđivanja transakcija označava vrijeme izraženo u sekundama potrebno za izvršavanje postupka provjere valjanosti transakcije. Na ovaj će čimbenik uglavnom utjecati tip mreže lanca vrijednosti i brzina računalne jedinice koja izvršava algoritam. S obzirom na to da različite kriptovalute koriste isti algoritam, ali nemaju jednako vrijeme potvrđivanja transakcija, lako je zaključiti da je za vrijeme prikazano u tablici „Opseg i rad algoritma“ uzet prosjek. Sukladno tome, najveće vrijeme potrebno za potvrđivanje transakcija imaju Dokaz o radu i Dokaz o aktivnosti, dok je brzina potvrđivanja najmanja kod Praktičnog problema bizantskog generala te Ripple sustav. Što manje vrijeme potvrđivanja je naravno najidealnije, iako ta činjenica uvelike ovisi i o namjeni određene kriptovalute, što može značiti da velika brzina potvrđivanja transakcija može biti i nepoželjna osobina.

Stupac „Propusnost“ tablice „Opseg i rad algoritma“ pokazuje procjenu broja transakcija koju mreža dopušta, odnosno s kojom nesmetano funkcionira. To bi značilo da algoritam Praktičnog problema bizantskog generala ima najveću propusnost koja iznosi do 2000 transakcija po sekundi, što je dvostruko više od drugog algoritma po redu, čije mjesto dijele Ripple i Delegirani dokaz o radu. Najmanju propusnost ponovo ima algoritam Dokaza o radu, dok je podataka za Dokaz o aktivnosti, Dokaz o važnosti i Dokaz o sreći nepoznat.

Već spomenuta brzina kreiranja bloka, koja se nalazi u trećem stupcu tablice „Opseg i rad algoritma“ usko ovisi o brzini provjere i metodama koje se koriste za odabir kreatora novog bloka u lancu vrijednosti. Iz ovog stupca može se primijetiti da je jedino Dokaz o radu klasificiran kao spor iz razloga što se novi blokovi u mreži lanca vrijednosti kao što je na primjer Bitcoin kreiraju prosječno svakih 10 minuta, što predstavlja dugo vrijeme u sustavima kriptovaluta.

„Skalabilnost“ predstavlja jedan od najvećih izazova u mreži lanca vrijednosti iz razloga što identificira količinu transakcija koje bi se mogle obraditi istovremeno kao i veličinu bloka koju može stvoriti jedan čvor. U ovom slučaju jedino je Praktični problem Bizantskog generala označen kao algoritam niske skalabilnosti iz razloga što ovaj algoritam obrađuje zahtjeve za transakcijom pri velikoj brzini, ali infrastruktura iz koje proizlaze komunikacija unutar mreže ograničava skalabilnost.

Stupac „Proširivost“ tablice „Opseg i rad algoritma“ određuje u kojoj se mjeri mreža može proširiti samim povećanjem broja čvorova. Algoritmi Dokaza o radu i Dokaza o ulogu ne mogu se više proširiti zbog načina na koji se odabire rudar koji s vremenom postaje polu-

pristran određenom rudaru koji ima snažnu računalnu jedinicu ili ima velik udio. Također, algoritmi koji se koriste u privatnom lancu vrijednosti ne mogu se proširiti zbog potrebe identificiranja čvorova prije početka rada u mreži. Međutim, povećanje veličine mreže neizbježna je činjenica, s obzirom na to da se broj korisnika svakoga dana povećava. (Alsunaidi, Alhaidari, 2019.)

Tablica 5. Vjerojatnost izloženosti

Algoritam	***Vjerojatnost izloženosti***		**Primjena**	**Primjer kriptovalute**
	51% napad	**Dvostruko trošenje**		
Dokaz o radu	Da	Da	Pametni ugovori, kriptovalute	Bitcoin, Litecoin, ZCash, Ethereum
Dokaz o ulogu	Ne	Ne	Pametni ugovori, kriptovalute	Peercoin, Tezos, Tendermint
Delegirani dokaz o ulogu			Kriptovalute, BitShares	Steemit, EOS, BitShares
Dokaz o aktivnosti			Kriptovalute	Parity
Ripple			Bankarstvo	XRP
Dokaz o važnosti			Internet stvari	XEM
Dokaz o sreći			Kriptovalute	Luckychain
Praktični problem Bizantskog generala			Pametni ugovori	Hyperledger

(Izvor: Alsunaidi, Alhaidari, 2019.)

Posljednja tablica „Vjerojatnost izloženosti“ proizlazi iz decentralizirane arhitekture mreža lanca vrijednosti kojoj je zadaća osiguravanje dobre razine sigurnosti te je podijeljena u dva stupca koja sadrži dvije vrste napada te dodatne primjere kriptovaluta koje koriste algoritme kao i njihove primjene.

Napad na mrežu moguć je na više načina, primjerice, kada se transakcije potvrđuju na temelju dogovora većine čvorova u mreži, napadač može upravljati dijelom tih čvorova, te tada imati značajan utjecaj na odluku o procesu rudarenja (izvršavanja transakcije).

Međutim, jedan od glavnih ciljeva mreže lanca vrijednosti je eliminacija takvih problema kao i centralizacije te stoga svaki algoritam nastoji spriječiti napad uz pomoć povećanja troškova za napadača. Primjerice, u Dokazu o radu napadač mora imati više od 51 posto računalne moći, kako bi uspio osigurati više od polovice sveukupne računalne moći mreže, ističući kako se ti zahtjevi s vremenom povećavaju. S druge strane, algoritam Dokaza o radu i ostali algoritmi onemogućuju spomenuti način napada već napadač mora posjedovati više od 51 posto ukupnog udjela u mreži kako bi izvršio uspješan napad. Algoritam Dokaza o radu također ne uspijeva savršeno otkloniti problem dvostrukog trošenja gdje je dolazilo do slučajeva kod kojih je ista kriptovaluta potrošena dva puta. (Alsunaidi, Alhaidari, 2019.)

5. Zaključak

U ovom radu objašnjeni su svi osnovni koncepti sustava kriptovaluta te terminologija potrebna za shvaćanje iste. Početak rada sadržavao je informacije o kratkoj povijesti nastanka kriptovaluta, njihovim vrstama te mjestima i načinima na koje se one čuvaju. Također su bile opisane transakcije uz pomoć kriptovalute te njihova primjena u radnim sektorima. Nakon toga detaljno je obrađena tehnološka osnovica kriptovaluta gdje je opisana kompleksnost i faze algoritama SHA256 te Keccak256. U nastavku je opisan elektronički potpis i njegova primjena nakon čega slijedi opisivanje procesa rudarenja kriptovaluta. U tom poglavlju je zaključeno da je proces rudarenja kriptovaluta usko povezan s procesom potvrđivanja, odnosno verificiranja transakcija nakon čega su opisani parametri tog procesa. Tu su bili spomenuti pojmovi kao što su načini rudarenja, težina rudarenja, ciljana vrijednost rudarenja, nasumična vrijednost koja se koristi jednom (eng. *nonce*), Dokaz o radu, Dokaz o ulogu, kao i njihova primjena. U nastavku je obrađen i lanac vrijednosti te njegovi sastavni dijelovi koji sadrže skupove transakcija koji se još nazivaju i blokovi. Pojam blokova još je detaljnije opisan u nastavku poglavlja gdje je bio opisan pojam Javorovih stabala i njihova primjena. Na kraju rada međusobno su uspoređeni algoritmi konsenzusa bez kojih proces stvaranja novih kriptovaluta, a niti proces potvrđivanja transakcija unutar mreže ne bi bio moguć. Za to poglavlje iskorišteni su algoritmi Dokaz o radu, Dokaz o ulogu, Delegirani dokaz o ulogu, Dokaz o aktivnosti, Ripple, Dokaz o važnosti, Dokaz o sreći te Praktični problem Bizantskog generala. Za kraj treba reći da je sustav lanca vrijednosti vrlo kompliciran slijed pažljivo isplaniranih i sigurnih događaja koji će u budućnosti, ako ne već i sada, pridonijeti veliki napredak u većini aspekata ljudskog života.

6. Literatura

Agrawal, H. (2019) *Explaining Hash Rate Or Hash Power In Cryptocurrencies* . pristupljeno 07.07.2021. na https://coinsutra.com/hash-rate-or-hash-power/

Alam, A., Zia Ur Rashid, S. M., Abdus Salam, Md., & Islam, A. (2018). *Towards blockchain-based e-voting system.* 2018 International Conference on Innovations in Science, Engineering and Technology (ICISET), 351–354. Chittagong, Bangladesh: IEEE. *https://doi.org/10.1109/ICISET.2018.8745613*

Al-Saqqa, S. (2020) . Blockchain Technology Consensus Algorithms and Applications: A Survey . The University of Jordan, Amman https://doi.org/10.3991/ijim.v14i15.15893

Alsunaidi, Shikah J., and Fahd A. Alhaidari. (2019) . *A Survey of Consensus Algorithms for Blockchain Technology*. 2019 International Conference on Computer and Information Sciences (ICCIS), IEEE, 2019, pp. 1–6. DOI.org (Crossref), https://doi.org/10.1109/ICCISci.2019.8716424.

Antonopoulos, A. M. (2016). *Mastering Bitcoin: Unlocking Digital Crypto-Currencies* . OReilly
Behind Bitcoin Mining . University College London

Bertoni, G., Daemen, J., Peeters, M., Van Assche, G. (2010) . *Keccak sponge function family*
bez autora (2014). *What are the equations to convert between bits and difficulty?.* pristupljeno 04.07.2021. na https://bitcoin.stackexchange.com/questions/30467/what-are-the-equations-to-convert-between-bits-and-difficulty

bez autora, (2016). *Bit Shifting* . pristupljeno 26.06.2021. na https://www.interviewcake.com/concept/java/bit-shift

Bodkhe, U., Tanwar, S., Parekh, K., Khanpara, P., Tyagi, S., Kumar, N., & Alazab, M. (2020). *Blockchain for industry 4. 0: A comprehensive review.* IEEE Access, 8, 79764–79800. https://doi.org/10.1109/ACCESS.2020.2988579

Caro, M. P., Ali, M. S., Vecchio, M., Giaffreda, R. (2018) . *Blockchainbased traceability in agri-food supply chain management: A practical implementation* . in Proc. IoT Vertical Topical Summit Agricult.-Tuscany (IOT Tuscany)

bez autora. bez. dat. *Cryptocurrency Mining Types*. pristupljeno 2.07.2021. na https://polygant.net/blog/cryptocurrency-mining-types/

bez autora, bez datuma . *Cryptocurrency statistics* . pristupljeno 07.07.2021. na https://bitinfocharts.com
bez autora . (2014) . *CRYPTO-VOTING The e-voting system based on Blockchain technology* . pristupljeno 28.7. 2021. na https://www.netservice.eu/en/research-and-development/crypto-voting

Courtois, N. T., Grajek, M., Naik, R. (2014) . *The Unreasonable Fundamental Incertitudes*

de Pedro Crespo, A., S., Garcia, L. I. C. (2016) . *Stampery Blockchain Timestamping Architecture* (BTA) .

Dinur, I., Dunkelman, O., Shamir, A. (2012). *New attacks on Keccak-224 and Keccak-256* . The Weizmann Institute, Rehovot . Israel . *main document* . https://eprint.iacr.org/2011/624.pdf

DocuSign, (bez dat.), *Understanding digital signatures* preuzeto 20.06.2021. s https://www.docusign.com/how-it-works/electronic-signature/digital-signature/digital-signature-faq

Dong-ha S., (2019) . *Bitcoin #3: Transaction .* pristupljeno 19.06.2021. na https://medium.com/@dongha.sohn/bitcoin-3-transaction-fa43a9151b7c

Dong-ha S., (2019) . *Bitcoin#5: Pool & Merkle Root .* pristupljeno 19.06.2021. na https://medium.com/@dongha.sohn/bitcoin-5-pool-merkle-root-272a9c83dec7

Dong-ha S., (2019) . *Bitcoin#6: Target and Difficulty.* pristupljeno 06.07.2021. na https://medium.com/@dongha.sohn/bitcoin-6-target-and-difficulty-ee3bc9cc5962

Entity , (University of Science and Technology Beijing).

Eremenko, K. (2018) . *How does Bitcoin / Blockchain Mining work?* . pristupljeno 07.07.2021. na https://medium.com/swlh/how-does-bitcoin-blockchain-mining-work-36db1c5cb55d

Federal Information Processing Standards Publication (FIPS) . (2012). *Secure Hash Standard (SHS)*. Information Technology Laboratory National Institute of Standards and Technology. Gaithersburg

Federal Information Processing Standards Publication (FIPS), (2015). *SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions* . National Institute of Standards and Technology . Gaithersburg

Fornari, M., Vardaneg, T. (2019) . *The Scalability Challenge of Ethereum: An Initial Quantitative Analysis* . University of Padova, Italy

Frankenfield J., (2021), *Altcoin* preuzeto 20.06.2021. s https://www.investopedia.com/terms/a/altcoin.asp

Ghimire, S., Selvaraj, H. (2018) . *A Survey on Bitcoin Cryptocurrency and its Mining .* University of Nevada, Las Vegas

Halaburda, H., & Sarvary, M. (2016). Beyond bitcoin: The economics of digital currencies. New York City, NY: Palgrave Macmillan.

Han, R., Foutris, N., & Kotselidis, C. (2019). *Demystifying crypto-mining: Analysis and optimizations of memory-hard pow algorithms*. 2019 IEEE International Symposium on Performance

Analysis of Systems and Software (ISPASS), 22–33. Madison, WI, USA: IEEE. https://doi.org/10.1109/ISPASS.2019.00011

Hannesdóttir, H. (2020). *An Analysis of the Bitcoin Blockchain Cryptography*. Reykjavik University

Hansen, T. 2006. *RFC 4634 - US Secure Hash Algorithms (SHA and HMAC-SHA)*. Dostupnu na poveznici: https://tools.ietf.org/html/rfc4634

He , X., Zhang, F., & Lin, S (2020). *A Review on Data Analysis of Bitcoin Transaction*

bez autora . (2021) . *Historical Snapshot - 21 March 2021* . pristupljeno 07.07.2021. na https://coinmarketcap.com/historical/20210321/

Hrvatska akademska i istraživačka mreža – CARNet, CCERT, LS & S . (2007) . *Digitalni potpis*

Judmayer, A., Stifter, N., Krombholz, K., & Weippl, E. (2017). *Blocks and chains: Introduction to bitcoin, cryptocurrencies, and their consensus mechanisms*. Synthesis Lectures on Information Security, Privacy, and Trust, 9(1), 1–123. https://doi.org/10.2200/S00773ED1V01Y201704SPT020

Judmayer, Aljosha, (2017). *Blocks and Chains: Introduction to Bitcoin, Cryptocurrencies, and Their Consensus Mechanisms*. 2017. Open WorldCat, IEEE.

Kohno, T., Ferguson, N., Schneier, B., (2010). *Cryptography Engineering: Design Principles and Practical Applications*. Wiley Pub., inc, 2010.

Kohno, Tadayoshi, et al. *Cryptography Engineering: Design Principles and Practical Applications*. Wiley Pub., inc, 2010.

Lee, D. (Ed.). (2015). *Handbook of digital currency: Bitcoin, innovation, financial instruments, and big data*. Amsterdam: Elsevier/ AP.

Lepore, C., Ceria, M., Visconti, A., Rao, U. P., Shah, K. A., & Zanolini, L. (2020). A survey on blockchain consensus with a performance comparison of pow, pos and pure pos. Mathematics, 8(10), 1782. https://doi.org/10.3390/math8101782

McEvoy, R. P., et al. "*Optimisation of the SHA-2 Family of Hash Functions on FPGAs*." IEEE Computer Society Annual Symposium on Emerging VLSI Technologies and Architectures (ISVLSI'06), vol. 00, IEEE, 2006, pp. 317–22. DOI.org (Crossref), doi:10.1109/ISVLSI.2006.70

Milutinovic, M., He, W. (2016) . *Proof of Luck: An Efficient Blockchain Consensus Protocol*. Proceedings of the 1st Workshop on System Software for Trusted Execution, ACM, 2016, pp. 1–6. DOI.org (Crossref), https://doi.org/10.1145/3007788.3007790

Naik, R. P. (2013). *Optimising the SHA256 Hashing Algorithm for Faster and More Efficient Bitcoin Mining* . University College London

Oakley, J., Worley, C., Yu, L., Brooks, R. , Skjellum, A. (2018) . *Unmasking Criminal Enterprises: An Analysis of Bitcoin Transactions* . Clemson University Clemson, SC., USA

O'Dwyer, K. J., Malone, D. (2014). *Bitcoin Mining and its Energy Footprint* . Hamilton Institute . National University of Ireland Maynooth

Paris, J.-F., & Schwarz, T. (2020). *Merkle hash grids instead of merkle trees*. 2020 28th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 1–8. Nice, France: IEEE. https://doi.org/10.1109/MASCOTS50786.2020.9285942

PeerCoin (2012) . *Pioneer of Proof of Stake* . pristupljeno 11.07.2021. na https://www.peercoin.net

bez autora . (bez datuma) . Proof of importance pristupljeno 27.7.2021. na https://www.moneyland.ch/en/proof-of-importance-definition

Proof of Stake (2018) . *Proof of Stake Explained* . pristupljeno 10.07.2021. na https://academy.binance.com/en/articles/proof-of-stake-explained

Proof of Work ideja . (2011) . *Proof of stake instead of proof of work* . pristupljeno 10.07.2021. na https://bitcointalk.org/index.php?topic=27787.0

Ricou, E. (2020). *Bitcoin halving dates history*. pristupljeno 07.07.2021. na https://stormgain.com/blog/bitcoin-halving-dates-history

Rohit, S. Kamra, M. Sharma and A. Leekha, "*Secure Hashing Algorithms and Their Comparison*," 2019 6th International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 2019, pp. 788-792.

Schoeman, L. (2021). *Cloud Mining – The Ultimate Beginners Step by Step Guide* . pristupljeno 31.06.2021. na https://sashares.co.za/cloud-mining/#gs.5ngb7w

Sharma, K., Jain, D. (2019) . *Consensus Algorithms in Blockchain Technology: A Survey* . Delhi Technological University New Delhi

Song, J. (2019). *Programming bitcoin: Learn how to program bitcoin from scratch (First edition)*. Beijing . O'Reilly.

Suratkar, S., Shirole, M., & Bhirud, S. (2020). *Cryptocurrency wallet: A review*. 2020 4th International Conference on Computer, Communication and Signal Processing (ICCCSP), 1–7. Chennai, India: IEEE. https://doi.org/10.1109/ICCCSP49186.2020.9315193

Texas Instruments Incorporated, (1996). *What's an LFSR?* . Texas

bez autora . (bez datuma) . *The Financial Network of the Future pristupuljeno* 20.7.2021. na https://ripple.com/ripplenet

Tomescu, A. (2020) . *What is a Merkle Tree?* . pristupljeno 10.07.2021. na https://decentralizedthoughts.github.io/2020-12-22-what-is-a-merkle-tree/

Vermak, W. (2021). *What Is ASIC Mining?*. pristupljeno 31.06.2021. na https://coinmarketcap.com/alexandria/article/what-is-asic-mining

Wang, Q, Huang, J., Wang, S., Chen, Y., Zhang, P., He, L. (2020) . *A Comparative Study of Blockchain Consensus Algorithms* . 2nd International Symposium on Big Data and Applied Statistics . Beijing

bez autora, (2016). *What is translation invariance in computer vision and convolutional neural network?*. pristupljeno 26.06.2021. na https://stats.stackexchange.com/questions/208936/what-is-translation-invariance-in-computer-vision-and-convolutional-neural-netwo

bez autora, (2016). *Which cryptographic hash function does Ethereum use?* . pristupljeno 26.06.2021. na https://ethereum.stackexchange.com/questions/550/which-cryptographic-hash-function-does-ethereum-use

bez autora (2016) . *Won't ASIC miners eventually break SHA-256 encryption?* . pristupljeno 07.07.2021. na https://bitcoin.stackexchange.com/questions/41829/wont-asic-miners-eventually-break-sha-256-encryption/41842

Xiaomeng, J., Fan, Z., Shenwen, L., Jinglin, Y., Ketai, H. (2020) . *Data Analysis of Bitcoin Blockchain Network Nodes* . University of Science and Technology Beijing

Xu, J., Bai, W., Hu, M., Tian, H., & Wu, D. (2021). *Bitcoin miners: Exploring a covert community in the Bitcoin ecosystem*. Peer-to-Peer Networking and Applications, 14(2), 644–654. https://doi.org/10.1007/s12083-020-01021-1

Yue, X., Wang, H., Jin, D., Li, M., & Jiang, W. (2016). Healthcare data gateways: Found healthcare intelligence on blockchain with novel privacy risk control. Journal of Medical Systems, 40(10), 218. https://doi.org/10.1007/s10916-016-0574-6

ZPR-FER-UNIZG: *Algoritmi i složenosti - predavanja*, 2019. Fakultet elektronike i računarstva

7. Popis slika

8. Popis tablica