

IZBORNI PROJEKT

Usporedba hash algoritama u kriptovalutama

Petra Avsec



Algoritmi i strukture podataka

Zavod za računarstvo

izv. prof. dr. sc. Kristijan Lenac

Tehnički fakultet Sveučilišta u Rijeci

Kolovoz 2020.

Sadržaj

1	Kriptovaluta	2
1.1	Značajne kriptovalute	2
1.2	Blockchain	3
1.2.1	Proof of Work	4
1.2.2	Hash funkcija	5
1.3	ASIC	6
2	Pregled hash algoritama	7
2.1	SHA-256	7
2.1.1	Opis rada	8
2.1.2	Primjeri	10
2.2	X11	10
2.3	SCRYPT	11
2.4	Dagger Hashimoto	12
2.4.1	Ethash	13
2.5	CryptoNight	14
3	Zaključak	15
	Literatura	16

1 Kriptovaluta

Kriptovaluta je vrsta digitalnog novca, elektronski način razmjene novca i zapisivanja transakcija na računalu. To je sredstvo razmjene koje koristi kriptografiju kako bi stvorilo i osiguralo prijenos novca. Za razliku od banaka, kriptovalute su decentralizirani sustav što znači da nema središnje organizacije koja kontrolira vrijednost, provjerava transakcije, može utjecati na korisnikovo primanje ili slanje novca, ili na bilo koji drugi način utječe na razmjenu novca. Kriptovaluta je peer-to-peer sustav razmjene što znači da za svoje funkcioniranje koristi rad svojih korisnika, oni su zaslužni za stvaranje novih novčanih jedinica i prijenos postojećih.

Sigurnost i integritet transakcija osigurava zajednica rudara (*miners*) koji svojim računalima ovjeravaju transakcije dodavanjem vremenske oznake i ubacivanjem u tzv. glavnu knjigu svih transakcija (*ledger*). Transakcije su računalno zahtjevne i nepraktične za poništiti ili promijeniti što kriptovalutu čini vrlo sigurnim načinom prijenosa novca.[1]

Sustav je siguran dok god većinu čvorova kontroliraju pošteni (*honest*) korisnici, a ne oni zlonamjerni (*attackers*).

Ispravnost svake jedinice kriptovalute osigurava blockchain.

1.1 Značajne kriptovalute

Nakon što je Satoshi Nakamoto 2008. objavio svoj rad u kojem opisuje kriptovalutu Bitcoin, naglo je poraslo zanimanje za navedeno područje.[2]

Koncept digitalnih valuta predstavljen je ranije[3], ali Nakamoto ga je prezentirao na nov, moderan i upotrebljiv način te omogućio stvaranje i razvoj ostalih kriptovaluta. Vrhunac vrijednosti Bitcoin je dosegao krajem 2017. godine, a otad je daleko najvrjednija, svima poznata, digitalna valuta. Na sam spomen naziva kriptovaluta, većina će odmah pomisliti na Bitcoin, ali bez obzira na to, danas postoje i mnoge druge kriptovalute.

Od ostalih kriptovaluta vrijedi izdvojiti open-source platformu Ethereum. Osim što je nakon Bitcoina najvrjednija kriptovaluta, omogućava izradu i pokretanje dis-

tribuiranih aplikacija.[4]

Litecoin se od Bitcoina razlikuje po tome što joj je brzina stvaranja blokova puno veća te je time i potvrda transakcija brža. [5]

Tether je jedna od prvih takozvanih stabilnih valuta. Sve kriptovalute su imale značajne poraste i padove vrijednosti, stoga je Tetheru bio cilj ublažiti nagle promjene vrijednosti te tako privući veći broj korisnika.[6]

Monero je sigurna, privatna kriptovaluta koja onemogućava praćenje platitelja. Za svaku transakciju stvara grupu kriptografskih potpisa među kojima se nalazi i ispravan potpisnik kojeg nije moguće izdvojiti.[7]

Uz navedene, i mnoge druge kriptovalute su ostavile svoj trag te doprinijele popularnosti korištenja digitalnog novca.

1.2 Blockchain

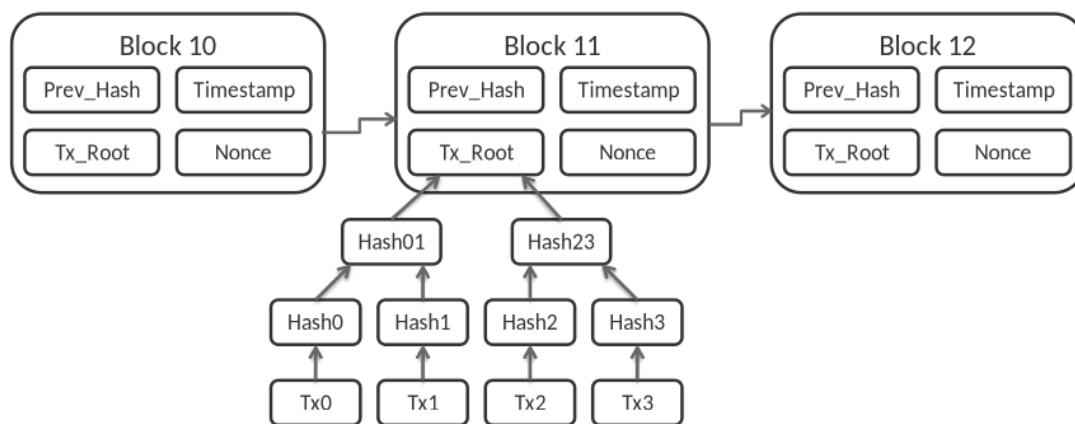
Sustav koji kriptovalute koriste za spremanje podataka o transakcijama je blockchain, decentralizirani mehanizam pohranjivanja informacija o transakcijama. Blockchain je distribuirana glavna knjiga u kojoj se spremaju i čuvaju podaci o transakcijama tako da svi korisnici imaju svoju kopiju, nema osobe ili organizacije koja ima popis transakcija te ih može mijenjati ili brisati. Sve informacije su vidljive svima koji koriste tu valutu te nije moguća manipulacija podataka.

Blockchain je realiziran kao rastuća lista podataka raspoređenih u blokove. Svaki blok je povezan s prethodnim tako što sadrži hash vrijednost tog bloka. Osim hash vrijednosti, blokovi sadrže i: index bloka, hash prethodnog bloka, vremenske oznaku, podatke, koji su u slučaju kriptovaluta, transakcije.

Transakcije su strukturirane u obliku Merkle stabla koje se stvara hashiranjem podataka u više navrata. Raspršeno adresiranje (*hashing*) je pretvorba niza znakova u vrijednost određene duljine koja predstavlja prvotni string. Na početku se hashiraju same transakcije, zatim upare dva hasha koji se ponovno hashiraju. Postupak se ponavlja dok ne dobijemo samo jedan hash, hash korijen (*root hash*) ili Merkle korijen. Svaki list je hash transakcije, a svaki čvor hash prethodnih hasheva. Kada

su podaci spremljeni u ovakvoj strukturi podataka, lako se može provjeriti da li je transakcija spremljena u tom setu podataka.

Potvrda da su transakcije određenog bloka prihvaćeni od strane ostalih čvorova i time ispravni, dobije se kada novi čvorovi počnu koristiti hash tog bloka u potrazi za novim.



Slika 1: Prikaz blockchaina

1.2.1 Proof of Work

Proof of work je mehanizam postizanja distribuiranog konsenzusa. Ključni je dio blockchaina jer ima veliki utjecaj na njegovu sigurnost i performanse. Pruža način na koji se korisnici mogu složiti oko prihvaćanja svih prijašnjih transakcija i potvrde

nadolazećih, a osigurava i cijeli blockchain od malicioznih čvorova, ako je većina sudionika dobronamjerna.[8]

Koristimo kada želimo ograničiti, ali ne i zabraniti pristup resursu. Ovaj mehanizam od korisnika zahtjeva neku vrstu rada, najčešće procesorsko vrijeme, i tako odvrća *denial-of-service* napade i druge vrste iskorištavanja usluga kao što je spam. Proof of work traži od korisnika da izračuna neku funkciju, tzv. *pricing function*. [9]

Kriptovalute primjenjuju proof of work tako što miner mora pronaći nonce vrijednost (proizvoljan broj koji se koristi može koristiti samo jednom), koja hashirana zajedno sa ostalim parametrima koji ulaze u blok, zadovoljava neke uvjete. Kao primjer se može uzeti kriptovaluta Bitcoin, kod koje hashirana vrijednost mora imati određen broj bitova na početku hash-a nula. Što je više nula u tom zahtjevu to je teže naći pripadajući hash.

Proof of work se koristi u blockchainu kako bi se stvorio zapis transakcija koji se ne može lako promijeniti, tj. trebalo bi ponoviti potreban rad za sve transakcije koje su bile nakon te koju bi htjeli promijeniti. Najdulji lanac služi kao dokaz svih događaja u lancu i dokaz najveće utrošene procesorske snage.

1.2.2 Hash funkcija

Hash funkcije su funkcije koje ulazne podatke proizvoljne dužine sažimaju u izlaz određenog formata i veličine.

Hash funkcije koje se koriste u blockchainu, kriptovalutama i općenito kriptografiji moraju biti jednostrane što znači da ne možemo lako iz izlaza dobiti odgovarajući ulaz.

Svojstva optimalnih hash funkcija:

- determinističke su - ista poruka (ulazna vrijednost) uvijek rezultira istim hashom
- velikom brzinom računaju hash vrijednost bilo koje poruke
- teško je i nepraktično generirati poruku koja daje određeni hash
- teško je pronaći dvije različite poruke s istom hash vrijednosti

- mala promjena poruke treba imati veliki utjecaj na izlaznu hash vrijednost kako se te dvije poruke ne bi mogle povezati na temelju sličnih izlaznih vrijednosti (učinak lavine, *avalanche effect*)[10]

Sigurne hash funkcije su otporne na sve vrste kriptanalitičkih napada:

- *preimage attack* - napad u kojem se nastoji pronaći ulaz određenog hasha ako znamo duljinu ulaza. *Brute force* napadom (napad koristeći grubu silu, u ovom kontekstu slanje svih mogućih ulaza u hash funkciju dok se ne dobije željeni hash) se ulaz može pronaći u 2^N evaluacija, ako je N duljina ulaza
- *collision attack* - napad čiji se uspjeh temelji na pronalasku dva različita ulaza čiji su izlazi identični
- *birthday attack* - pronalazak dva različita ulaza hash funkcije koji rezultiraju istim izlazom (kolizija) slučajno, nasumičnim procesom

1.3 ASIC

ASIC (application-specific integrated circuit) je IC čip specifično izrađen za neki zadatak, svrhu. Kod kriptovaluta njihova uloga je računanje hash algoritama kriptovaluta (kao primjer možemo uzeti Bitcoin ASIC miner koji je dizajniran za računanje SHA-256).

ASIC mineri se oduvijek smatraju prijetnjom sustava rudarenja i kriptovaluta. Glavni su uzrok centralizacije hashing snage. GPU i CPU su u nepovoljnoj poziciji kraj ASIC-a ako se uzme u obzir rudarenje blokova i dobivanje nagrada za to, te tako ograničavaju većinu potencijalnih korisnika od rudarenja kriptovalute. ASIC-i su superiorniji CPU i GPU zato što mogu izračunati više hasheva po sekundi, tako da su rudari koji koriste ASIC u prednosti nad ostalima.[11]

2 Pregled hash algoritama

S obzirom na trenutani broj kriptovaluta, relativno je malen broj različitih algoritama koje one koriste. Većina ih je implementirana s već testiranim i često upotrebljavanim algoritmima kao što su SHA-256 i Scrypt, dok su određene kriptovalute razvile vlastite funkcije i algoritme: Ethereum (Ethash).

2.1 SHA-256

Secure Hash Algorithm 2 je set kriptografskih funkcija koje je 2001. objavila NSA. Broj 256 u nazivu označava veličinu izlazne vrijednosti funkcije koja iznosi 256 bitova. Postoji nekoliko različitih verzija ovog algoritma s različitim veličinama izlaza od 224, 384, 512, 512/224 i 512/256 bitova.

SHA je jedan od najkorištenijih algoritama u svijetu, a koriste ga među ostalim kriptovalute Bitcoin, Namecoin, Peercoin, Nxt, MazaCoin, te se koristi i u autentifikacijskom procesu Debian softver paketa.

SHA-256 je razvijen koristeći Merkle-Damgård strukturu koja je pak napravljena pomoću Davies–Meyer jednostrane kompresijske funkcije. Temeljna ideja iza Davies–Meyer konstrukcije je kompresija bloka teksta u n bitova koristeći enkripcijski algoritam. To se postiže stavljanjem nasumične početne vrijednosti od n bitova kao poruke i korištenjem teksta kao ključa. Time se kao produkt enkripcije dobije blok od n bitova. Na rezultat se zatim još primjeni operacija XOR kako bi se smanjila vjerojatnost stvaranja kolizije.

Merkle–Damgård hash funkcija je način stvaranja kriptografske hash funkcije otporne na kolizije iz jednostranih kompresijskih funkcija. Koristi se u SHA algoritmu kod rastavljanja ulaza na blokove duljine 512 bitova, koji se zatim obrađuju jedan po jedan koristeći jednostrane kompresijske funkcije bazirane na Davis-Meyeru.

2.1.1 Opis rada

Poruka se dijeli u blokove duljine 512 bitova, a ako je blok manji, proširimo ga do 512 bitova. Blokovi se zatim obrađuju jedan po jedan.

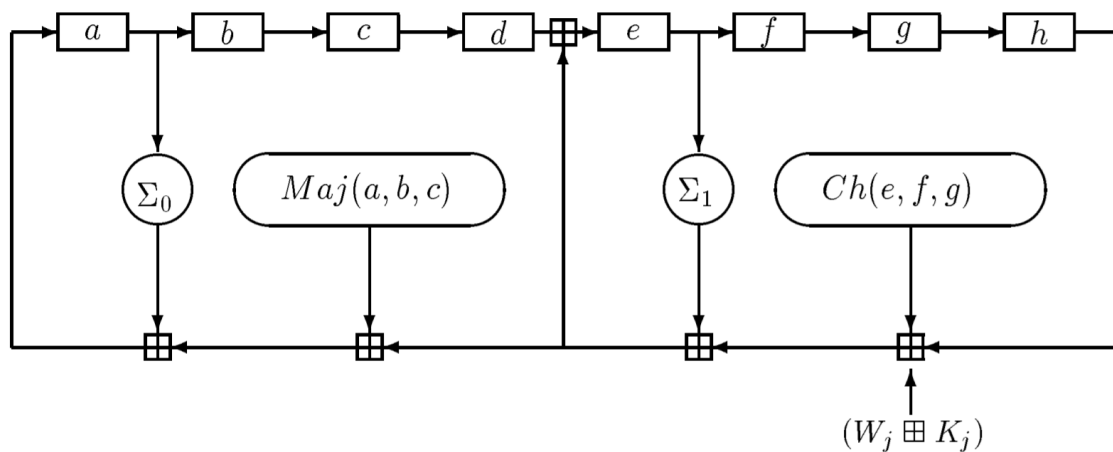
Poruka se proširuje tako da se duljini poruke nadoda jedan bit na kraj čime se dobije N bitova. Na kraj dobivenog dodaje se $512 - 64 - N$ nula, a u zadnja 64 bita se zapiše duljinu originalne poruke. Blokovi od 512 bitova dijele se na 16 dijelova po 32 bita ($m_0 - m_{15}$).

Pseudokod:

- 8 početnih hash vrijednosti duljine 32 bita (ostatci korijena prvih osam prim brojeva):
 - $H(0)_1 = 6a09e667$
 - $H(0)_2 = bb67ae85$
 - $H(0)_3 = 3c6ef372$
 - $H(0)_4 = a54ff53a$
 - $H(0)_5 = 510e527f$
 - $H(0)_6 = 9b05688c$
 - $H(0)_7 = 1f83d9ab$
 - $H(0)_8 = 5be0cd19$
- Glavna petlja (N iteracija gdje je N broj blokova poruke):
 1. inicijalizacija a, b, c, d, e, f, g, h s vrijednostima h_1 do h_8
 2. primjena SHA-256 kompresijske funkcije čime se dobiju nove vrijednosti varijabli $a - h$
 3. računanje novih $h_1 - h_8$, tako da $h_1 = a + h_1 \dots h_8 = h + h_8$

Rezultat izvršavanja petlje je hash poruke s početka.

- Kompresijska funkcija:
 - 64 iteracije, 6 logičkih funkcija koje se koriste za dobivanje novih vrijednosti $a - h$. Sve funkcije rade s 32-bitnim riječima i izlaz im je veličine 32 bita. U računanju se koriste i 64 konstante, 32-bitne riječi $K_0 - K_{63}$ (ostatci kubnih korijena prvih 64 prostih brojeva)
 - u računanju se za kriptiranje koristi: bitwise XOR, bitwise AND, bitwise OR, bitwise komplement, zbrajanje mod 2^{32} , posmak u desno za n bitova, rotacija u desno za n bitova[12]



Slika 2: Kompresijska funkcija SHA-256 algoritma. Slika prikazuje j -ti korak funkcije, a \boxplus označava mod 2^{32} zbrajanje

Razina zaštite, sigurnosti, u kriptografiji je mjera snage hash funkcije. Izražena je u bitovima, n - *bit security*, što znači da napadač treba izvesti 2^n operacija da bi razbio funkciju.

Broken hash funkcija - uspjeli smo izvesti barem jedan od napada: collision attack ili preimage attack.

SHA-256 ima 128 bitnu razinu zaštite.

2.1.2 Primjeri

Za niz znakova 'abc', hash algoritma SHA-256 je:

```
ba7816bf8f01cfea414140de5dae2223b00361a396177a9cb410ff61f20015ad
```

Za niz znakova 'aBc', hash algoritma SHA-256 je:

```
516dd854ec42b5b992888cfa87ae16e260864f5e051e045cd7d7c0b45eacbeb2
```

Iz navedenog primjera lako se može uočiti učinak lavine, svojstvo optimalnih hash funkcija, koje kaže da male promjene u ulaznoj poruci trebaju imati značajan utjecaj na izlaz.

2.2 X11

X11 algoritam je, isto kao i SHA-256, proof of work algoritam. Za razliku od ostalih algoritama koristi drugačiji pristup hashiranju, ulančavanje algoritama. Razvio ga je Evan Duffield te je algoritam 2014. implementiran u protokol Darkcoin kriptovalute, kasnije preimenovane u DASH.

X11 algoritam je kombinacija 11 hash funkcija: BLAKE, BLUE MIDNIGHT WISH (BMW), Grøstl, JH, Keccak, Skein, Luffa, CubeHash, SHAvite-3, SIMD, ECHO. Poruka se predaje prvoj hash funkciji koja ga obradi i proslijedi svoj izlaz sljedećoj funkciji kao poruku.

X11 je sigurniji od SHA-256. Sve hash funkcije koje se koriste kod X11 algoritma su bile kandidati kod traženja novog standarda, boljeg sigurnijeg algoritma SHA3 koji je temeljen na Keccak funkciji. Razvijeno je još algoritama koji se temelje na istoj ideji kao i X11: X13, X14, X15 i X17 koji koriste više hash funkcija.

Kriptovalute koje koriste X11 su Dash, Hatch, Pura, SmartCoin, CannabisCoin, Influxcoin, StartCoin, Onix, i mnoge druge.[\[13\]](#)

Napravljen je kako bi se otežala izrada ASIC-a koji će efikasno rudariti X11 algoritam, iako danas više nije ASIC otporan.

S obzirom na navedene poteškoće s kojima se susreću kriptovalute te njihovi rudari, X11 algoritam je dizajniran kako bi bio efikasan i na CPU i GPU. Osmišljen je da bi

bio čim duže ASIC otporan (ili duže nego ostale konkurentne kriptovalute) kako bi "obični" korisnici mogli dulje sudjelovati u rudarenju, što je vrlo povoljno za Dash kriptovalutu, njeno širenje i dobru distribuciju.

2.3 SCRYPT

Još jedan u nizu hash algoritama koji su razvijeni s namjerom da se izbjegne stvaranje ASIC-ova te maksimalno oteža njihovo rudarenje kriptovaluta. Scrypt algoritam je vrlo memorijski zahtjevan - osim što zahtjeva od rudara da brzo stvaraju brojeve, ti brojevi se spremaju u RAM i treba im se pristupiti prije dobivanja konačnog hash-a. Ovim pristupom se drastično smanjuje učinkovitost integriranih krugova specifične namjene.

Scrypt je osmislio Colin Percival za spremanje online sigurnosnih kopija UNIX operativnih sustava. Algoritam dodatno otežava rješavanje dodajući šum (noise) - nasumično generirane brojeve te tako povećava vrijeme potrebno za dobivanje hash-a. Koriste ga kriptovalute Litecoin, Dogecoin, ProsperCoin, MonaCoin...

Algoritam je sačinjen od nekoliko parametara:

- N - parametar koji označava koliko CPU/memorije traži algoritam
- p - paralelizacijski parametar, pozitivni cijeli broj
- r - veličina bloka
- S - "sol", nasumična vrijednost koja se često koristi kao dodatni ulaz u kriptografskim funkcijama. Služi kao zaštita od *Rainbow table* napada
- P - ulazna vrijednost, niz znakova koji želimo hashirati
- *dkLen* - željena duljina izlazne vrijednosti u oktetima

Navedene podatke prosljedimo *key derivation* funkciji PBKDF2 (funkcija kojom izvlačimo tajne ključeve iz vrijednosti kao što su master key ili neke vrste lozinke). PBKDF2 je akronim za Password-Based Key Derivation Function 2, kojom smanjujemo ranjivost na napade grubom silom (brute force) tako da dobijemo izvedeni ključ.[\[14\]](#)

2.4 Dagger Hashimoto

Dagger Hashimoto je prethodnik, istraživački algoritam za kriptovalutu Ethereum 1.0, kasnije zamijenjen za algoritam Ethash. Razvijen je s namjerom da ispuni sljedeće zahtjeve:

- ASIC otpornost - mala mogućnost izrade ASIC-a koji će moći rudariti Ethereum ili kad se on uspije izraditi, minimalna isplativost ASIC-a naspram CPU
- širok spektar različitih jednostavnih uređaja, uređaja male snage, koji će moći potvrditi blok
- spremanje cijelog blockchain-a

Temelj algoritma Dagger Hashimoto činili su već postojeći algoritmi:

- Hashimoto - autor algoritma je Thaddeus Dryja, a ASIC otpornost nastoji postići tako što je vezan za IO, ograničava pristupe memoriji tijekom rudarenja
- Dagger - razvio ga je Vitalik Buterin, ujedno i jedan od autora Ethereuma. Koristi acikličke grafove kako bi računanje bilo memorijski zahtjevno, ali i da bi validacija blokova istovremeno koristila relativno malo memorije. Alternativa isto tako memorijski zahtjevnom algoritmu, Scryptu. Odustalo se od njegova korištenja zbog otkrivenog manjka sigurnosti [15]

2.4.1 Ethash

Ethash je ažurirana verzija Dagger Hashimota, iako se više ne može nazvati tako zato što je algoritam značajno promijenjen kako bi se ispravili svi nedostaci njegova prethodnika u zadnjim fazama razvoja.

Okvirni proces algoritma je sljedeći:

- postoji nasumični broj koji se može izračunati prolaskom kroz zaglavlja postojećih blokova
- iz tog broja se računa pseudo nasumični cache od 16 MB koji spremaju lagani klijenti
- na temelju brze memorije se stvori set podataka veličine 1GB kojeg spremaju klijenti i rudari. Set podataka raste s vremenom, a svaki element ovisi samo o malom broju elemenata iz brze memorije
- rudarenjem se uzimaju nasumični dijelovi seta podataka koji se zatim zajedno hashiraju. Verifikacija se može napraviti vrlo brzo uz korištenje malo memorije, upotrebom brze memorije jednostavno se stvore potrebni dijelovi seta podataka[16]

2.5 CryptoNight

Slično ostalim algoritmima, CryptoNight-u je glavni cilj otpornost na ASIC, a uz to, efikasniji je za korištenje na CPU nego na GPU. Izvršavanje algoritma je vrlo osjetljivo na latenciju memorije zato što se kod implementacije koristi petlja u kojoj se neprestano čitaju i pišu podaci u memoriju.

CryptoNight algoritam se odvija u tri koraka:

1. inicijalizacija memorije s pseudo nasumičnim podacima. Ulaz se hashira Keccak algoritmom te se nad dobivenim rezultatom provodi AES enkripcija čime se inicijalizira prvih 128 bitova memorije. Zatim se dobiveni blok podataka ponovno enkriptira te zapiše u sljedećih 128 bitova. Proces se ponavlja dok enkriptirani podaci ne zauzmu 2MiB memorije
2. petlja u kojoj se ponavlja čitanje i pisanje u memoriju. Petlja se ponavlja 524288 puta
3. hashiranje cijele memorije u kojoj se nalaze podaci nad kojima su se odvijale navedene operacije. U zadnjem koraku algoritma, izlaz se proslijedi jednoj od sljedećih funkcija na temelju posljednja dva, najmanje značajna bita dobivenog rezultata: Skein, JH, Groestl ili BLAKE.[17]

3 Zaključak

Osim navedenih algoritama, u svijetu se u različite svrhe koristi nebrojeno mnogo drugih algoritama i hash funkcija.

Pojava kriptovaluta, prvenstveno Bitcoina, svakako je povoljno utjecala na osmišljavanje novih algoritama i motivaciju za njihovo istraživanje i testiranje.

Algoritmi su ključni dio kriptovaluta i blockchaina na temelju kojeg su one implementirane. Utječu na njihovu sigurnost i ispravnost.

Decentralizaciji kriptovaluta prijete ASIC-i koji svojim vlasnicima omogućavaju vrlo brzo i efikasno rudarenje naspram ostalih "običnih" korisnika koji koriste njima dostupne CPU i GPU. Pojavom ASIC-a za pojedini algoritam većini korisnika prestaje biti isplativo rudarenje, te većina novih novčanih jedinica pripadne manjem broju korisnika. Cilj svih navedenih algoritama je, relativno neuspješno, bio onemogućiti izradu ASIC-a za taj specifični algoritam. U većini slučajeva, uspješno je samo odgođen i produljen njihov razvoj. S druge strane, prebacivanje cijele mreže na rudarenje pomoću ASIC-a bi uvelike doprinjelo sigurnosti kriptovalute koja bi koristila takav pristup.[\[19\]](#) Stoga se isplativost njihove upotrebe mora dodatno istražiti.

Daljnji razvoj i širenje kriptovaluta zasigurno će pozitivno utjecati na razvijanje novih, boljih, sigurnijih algoritama te kriptanalize općenito.

Literatura

- [1] U. Mukhopadhyay, A. Skjellum, O. Hambolu, J. Oakley, L. Yu, and R. Brooks. A brief survey of cryptocurrency systems. In *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pages 745–752, 2016.
- [2] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Cryptography Mailing list at <https://metzdowd.com>*, 2009.
- [3] W. Dai. b-money, 1998. <http://www.weidai.com/bmoney.txt>.
- [4] Vitalik Buterin. A next generation smart contract and decentralized application platform, 2014. https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf.
- [5] Jeff Reed. *Litecoin: An Introduction to Litecoin Cryptocurrency and Litecoin Mining*. CreateSpace Independent Publishing Platform, North Charleston, SC, USA, 2017. ISBN 1974692949.
- [6] J.R. Willett. Tether: Fiat currencies on the bitcoin blockchain, 2012. <https://tether.to/wp-content/uploads/2016/06/TetherWhitePaper.pdf>.
- [7] Nicolas van Saberhagen. Cryptonote v 2.0, 2013. <https://cryptonote.org/whitepaper.pdf>.
- [8] Y. Xiao, N. Zhang, W. Lou, and Y. T. Hou. A survey of distributed consensus protocols for blockchain networks. *IEEE Communications Surveys Tutorials*, 22(2):1432–1465, 2020.
- [9] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In *Advances in Cryptology - CRYPTO '92, 12th Annual International Cryptology Conference, Santa Barbara, California, USA, August 16-20, 1992, Proceedings*, volume 740 of *Lecture Notes in Computer Science*, pages 139–147. Springer, 1992. doi: 10.1007/3-540-48071-4_10.

- [10] Saif Al-Kuwari, James H. Davenport, and Russell J. Bradford. Cryptographic hash functions: Recent design trends and security notions. Cryptology ePrint Archive, Report 2011/565, 2011. <https://eprint.iacr.org/2011/565>.
- [11] How application-specific integrated circuits are powering the future of it today, . <https://www.cio.com/article/3261425/how-application-specific-integrated-circuits-are-powering-the-future-of-it-today.html>, 22.08.2020.
- [12] NIST. Descriptions of sha-256, sha-384, and sha-512. <https://web.archive.org/web/20130526224224/http://csrc.nist.gov/groups/STM/cavp/documents/shs/sha256-384-512.pdf>.
- [13] Evan Duffield and Daniel Diaz. Dash: A payments-focused cryptocurrency, 2014. <https://github.com/dashpay/dash/wiki/Whitepaper>.
- [14] Colin Percival. Stronger key derivation via sequential memory-hard functions. 2009.
- [15] Dagger hashimoto. <https://github.com/ethereum/wiki/wiki/Dagger-Hashimoto>, 15.08.2020.
- [16] Ethash. <https://eth.wiki/en/concepts/ethash/ethash>, 15.08.2020.
- [17] anonymous. Cryptonight hash function. 2013. <https://cryptonote.org/cns/cns008.txt>, 21.08.2020.
- [18] John Tromp. Cuckoo cycle: a memory bound graph-theoretic proof-of-work. 2019.
- [19] Switching bitcoin to 100 <https://cointelegraph.com/news/switching-bitcoin-to-100-asic-mining-could-increase-security-2000x>, 31.08.2020.