Tehnologija Bitcoina i kriptovaluta

Bubalo, Ivana

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku

Ivana Bubalo

Tehnologija Bitcoina i kriptovaluta

Završni rad

Osijek, 2019.

Sveučilište J. J. Strossmayera u Osijeku
Odjel za matematiku

Ivana Bubalo

Tehnologija Bitcoina i kriptovaluta

Završni rad

Mentor: izv. prof. dr. sc. Domagoj Matijević

Osijek, 2019.

Sadržaj

Sažetak

U ovom završnom radu obrađene su kriptografske hash funkcije te neka njihova svojstva i primjeri, a zatim funkcija SHA-256 kojom se koristi Bitcoin. Objašnjeno je kako nam hash-pointeri služe za izgradnju povezanih lista, odnosno blockchaina. Dana je shema digitalnog potpisa i razrađena ideja korištenja njegovog slučajno generiranog javnog ključa kao identiteta, što nam služi za razumijevanje decentraliziranog načina funkcioniranja Bitcoina i privatnosti. Navedene su neke primitivne kriptovalute, njihova svojstva, od kojih će neka nasljediti i Bitcoin, vrste transakcija te problematika napada dvostrukog trošenja. U trećem poglavlju objašnjen je pojam decentralizacije kao vrlo bitnog koncepta kriptovaluta, te što je raspodjeljeni konsenzus. Nadalje, objašnjeno je što je Bitcoin transakcija i uvjeti koji moraju biti ispunjeni ukoliko ju želimo obaviti. Dana je struktura Bitcoin bloka koja sadrži Merkleovo stablo, zbog čega je vrlo korisna rudarima zbog lakog praćenja i potvrđivanja transakcija. Posljednje poglavlje posvećeno je pohranjivanju Bitcoina. Objašnjen je način funkcioniranja wallet-softwarea, kodiranje adresa u svrhu razmjene, te način računanja prioriteta transakcije u ovisnosti o naknadi za rudare. Kratko su objašnjene mogućnosti kupovine ili prodaje kriptovaluta, koji su glavni izvori potražnje za Bitcoinom, dan je pojednostavljeni model posredovanja iz kojeg izvodimo formulu ravnoteže njegove cijene.

Ključne riječi

hash funkcija, kriptovaluta, SHA-256, blokchain, digitalni potpis, double-spending attack, P2P, rudarenje, Bitcoin, novčanik, naknade za transakcije

Abstract

Hash functions and some of their properties, examples, and then one special function SHA-256 used by Bitcoin are discussed in this bachelor's thesis. It is explained how hash-pointers serve us in building linked lists, or blockchain. It is given a digital signature scheme and the idea of using its random generated public key as identity was elaborated, which we use to understand the mode of decentralization in functioning Bitcoin and privacy. Some primitive cryptocurrencies and some of their properties that will be inherited by Bitcoin are listed, types of transactions and problem of double-spending attack. In third chapter it is explained the concept of decentralization as very important concept of cryptocurrencies and what is distributed consensus. Furthermore, there is an explanation of Bitcoin transaction and necessary conditions which

we have to meet if we want to carry it out. There is given the structure of Bitcoin block containing a Merkle tree, which makes it very useful because of easy tracking and verifying transactions. The last chapter is devoted to storing Bitcoin. There is explained the way the wallet software works, the coding of the address for exchanging, and the method of calculating transaction priorities depending on fees for the miners. The possibilities of buying or selling cryptocurrencies, which are the main sources of demand for Bitcoin are briefly explained, and there is a simplified mediation model from which we find the formula of the equilibrium of its price.

Key words

hash function, SHA-256, blockchain, digital signature, double-spending attack, P2P, mining, Bitcoin, wallet, transaction fee

Uvod

U posljednjih su nekoliko godina kriptovalute privukle veliku pažnju industrijske i akademske zajednice. Bitcoin koji se često naziva prvom kriptovalutom uživao je ogroman uspjeh jer je na tržištu kapitala u 2016. dostigao 10 milijardi dolara (*Coindesk, 2016*). *"Kreditna kriza"*, *„propadanje banke"*, *„vladina pomoć"* i slične fraze bile su česta pojava u medijima tijekom jeseni 2008., u razdoblju kada su glavna financijska tržišta izgubila više od 30% svoje vrijednosti. Blokchain je osnovni mehanizam za Bitcoin. Osoba, ili više njih, pod pseudonimom *Satoshi Nakamoto* prvi ga je put najavila u listopadu 2008. godine, a implementirala u siječnju 2009. godine. Prvu zabilježenu transakciju napravio je Laszlo Hanyecz 22. svibnja 2010. kada je kupio dvije pizze za 10,000 Bitcoina, a na taj se dan obilježava *Bitcoin Pizza Day.* U budućnosti će se mnogi ljudi okrenuti Bitcoin-u kao rješenju, najviše zbog mogućnosti njegovog korištenja kao platnog sustava bez uključivanja trećih strana ili vlada. U ovom završnom radu bavit ćemo se nekim od glavnih karakteristika zbog kojih je Bitcoin zadobio povjerenje tržišta.

Poglavlje 1

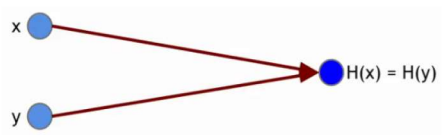
Kriptografija

1.1 Kriptografske hash funkcije

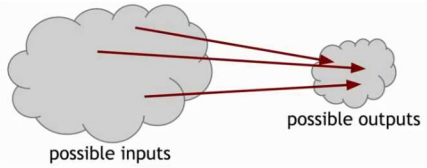
Hash funkcije matematičke su funkcije s ova tri svojstva:
-Input: mogu primiti bilo kakav string bilo koje veličine
-Output je fiksne veličine (Bitcoin ima output veličine 256 bita)
-Vremenska učinkovitost (u razumnom vremenskom roku saznajemo što je output)
Koristit ćemo hash funkcije koje su kriptografski sigurne. Kriptografska svojstva hash funkcija su općenito vrlo opširna tema, pa ćemo se trenutno fokusirati na njihova tri određena svojstva: collision free, hiding property i puzzle-friendly, o kojima će više riječi biti u nastavku.

1.2 Collision free

Prvo svojstvo koje zahtjevamo od kriptografskih funkcija je otpornost na koliziju. Do kolizije dolazi kada dvije različite ulazne vrijednosti daju istu izlaznu vrijednost.



(a) *Kolizija*



(b)

Slika 1.1

Kažemo da je hash funkcija $H(\cdot)$ otporna na koliziju ukoliko je neisplativo tražiti dvije vrijednosti, x i y, td. je $x \neq y$, a da je $H(x) = H(y)$.
Ukoliko funkcija ima ovo svojstvo, "nemoguće" je pronaći različite x, y takve da je

$H(x) = H(y)$ (Slika 1.1(a)). Na prvu bi pomislili da je funkcija s ovim svojstvom injektivna, no ustvari promislimo li malo da je domena ove funkcije puno veća od kodomene (jer je input bilo koje veličine, a output 256 bita), vidimo da kolizija postoji, tj. neke različite točke domene se ipak preslikaju u istu točku kodomene (Slika 1.1(b)). Pitanje je možemo li korištenjem običnog računala pronaći takve točke, odnosno koliziju.

Razmotrimo sljedeću jednostavnu metodu pronalaska kolizije za hash funkciju s 256-bitnom izlaznom vrijednošću: uzmimo $2^{256} + 1$ različitih vrijednosti, izračunajmo vrijednosti hasha u svakoj od njih te provjerimo jesu li neke dvije izlazne vrijednosti jednake. Pošto smo izabrali više ulaznih vrijednosti nego što postoji izlaznih, neke od njih će sigurno dati istu vrijednost kada na njih primjenimo hash funkciju.

Gore navedena metoda garantira pronalazak kolizije. No ukoliko nasumično izaberemo ulazne vrijednosti i izračunamo hash vrijednosti, velika je vjerojatnost da će se kolizija dogoditi puno ranije nego što ispitamo $2^{256}+1$ ulaznih vrijednosti. Zapravo, ukoliko na slučajan način izaberemo samo $2^{130} + 1$ ulaznih vrijednosti, ispostavlja se da će barem dvije od njih dati jednaku izlaznu vrijednost s vjerojatnošću od 99.8%. Činjenica da ispitivanjem približnog kvadratnog korijena broja svih mogućih izlaznih vrijednosti možemo pronaći koliziju proizlazi iz fenomena u vjerojatnosti poznatog kao *problem rođendana*.

Ovaj algoritam za pronalaženje kolizije radi za svaku hash funkciju, no problem je vremenska neučinkovitost. Za hash funkciju s 256-bitnim outputom morali bismo, u najgorem slučaju, računati hash funkciju $2^{256}+1$ puta, a u prosjeku 2^{128} puta. Ukoliko bismo koristili računalo koje računa $10,000$ hasheva po sekundi, trebala bi mu jedna kvadrilijarda (10^{27}) godina kako bi izračunalo 2^{128} hasheva.

Zašto nam je dobro ovo svojstvo?

Pretpostavimo da je funkcija H bez kolizije. Takva funkcija nam može poslužiti u prenošenju poruka. Ako znamo da je $H(x) = H(y)$, sigurno je pretpostaviti da je $x = y$. Pretpostavimo da u lijevom oblaku (slika 1.1(b)) imamo ogromnu datoteku x. Željeli bismo biti u mogućnosti da ubuduće prepoznamo je li neka datoteka jednaka početnoj. Prvi način je da spremimo početnu veliku datoteku i uspoređujemo ju s drugom, y. Drugi, puno učinkovitiji način je zapamtimo $H(x)$, i uspoređujemo ga s ostalima $H(y)$, $H(z)$ itd. Dobijemo li da je $H(x) = H(y)$, možemo zaključiti da su datoteke x i y jednake. Ovaj je način puno učinkovitiji jer uspoređujemo hasheve veličine 256 bita, umjesto velikih datoteka. Zato je hash-funkcija pogodna za brzi pregled poruka, o čemu će biti više riječi u nastavku.

1.3 Jednosmjernost

Ovo je svojstvo koje zahtijevamo od hash-funkcije, a znači: ako nam je dana vrijednost $H(x)$, ne postoji isplativ način pronalaska vrijednosti x. Promotrimo prvo primjer zašto ne možemo u potpunosti vjerovati ovom svojstvu.

Primjer 1.1. Bacanje novčića
Padne li novčić na pismo, rezultat je $H(\text{'}pismo\text{'})$ ili u suprotnom $H(\text{'}glava\text{'})$. U ovoj situaciji, bilo tko, tko nije vidio bacanje novčića, lako će iz vrijednosti $H(\text{'}pismo\text{'})$ ili $H(\text{'}glava\text{'})$ odgonetnuti koja je vrijednost bila početna, odnosno kako je novčić pao. Dakle, kako bi podatak x držali tajnim, morat ćemo malo zakomplicirati stvari. Uzmimo proizvoljan r iz distribucije vjerojatnosti s velikom minimalnom entropijom. Tada je za danu funkciju $H(r|x)$ vrlo neisplativo tražiti x.

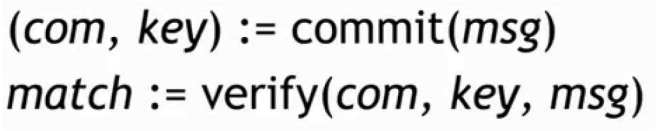
Napomena 1.1. i) $H(r|x)$ znači da su u prvom dijelu inputa bitovi od r-a, zatim oni od x-a. Možemo reći da su r i x ulančani.
ii) *Velika minimalna entropija* - U teoriji informacija, minimalna entropija je mjera koja opisuje koliko je ishod predvidiv, a velika minimalna entropija odnosi se na vrlo široke distribucije (slučajnih varijabli). Pri uzorkovanju iz distribucije, nema određene vrijednosti koja se vjerojatnije pojavljuje. Konkretno, ako je r izabran uniformno iz svih stringova veličine 256 bita, tada je svaki string izabran s vjerojatnošću $1/2^{256}$, što je vrlo neznatna vrijednost. Dakle, birajući r na taj način, $H(r|x)$ će nam vrlo sigurno sakriti x.

Upotreba:
Ovaj postupak je digitalno analogan sa sljedećim:
Uzimamo neku vrijednost, stavimo ju u kovčeg i zaključaljmo ga (tako zapečatimo poruku). Zatim kovčeg stavimo na stol gdje je svima vidljiv. Vrijednost unutar kovčega je nepoznata svim promatračima. Kasnije nekome možemo dozvoliti da provjeri što je u kovčegu na način da mu predamo kovčeg i ključ. Stavljanjem kovčega na stol i objavom ključa, svaki promatrač može provjeriti je li određena poruka ili vrijednost stvarno unutar kovčega.

Obratimo pozornost na dvije sigurnosne karakteristike i sliku 1.2.

1. Promatrajući dani *'com'*, odnosno kovčeg na stolu, nitko ne može zaključiti što je unutar njega, odnosno otkriti poruku.

2. Kad odlučimo što ćemo staviti u kovčeg, unutar njega više ništa ne možemo mijenjati. Na taj način osigurali smo da ne vrijedi $H(x) = H(y) \Rightarrow x \neq y$.



Slika 1.2: *Kako implementirati kovčeg*

Kako implementirati kovčege (slika 1.2):
U namjeri da zapečatimo poruku, generirat ćemo proizvoljnu 256-bitnu vrijednost i nazvati ju *ključ*. Kovčeg će nam "vratiti" vrijednost hasha od ključa povezanog s porukom, odnosno $H(key|msg)$, a vrijednost ključa dobit ćemo s $H(key)$. U namjeri da verificira poruku, osoba će ukucati $H(key|msg)$, te usporediti danu vrijednost s onom u kovčegu.
Dakle, ovo je način zapečaćivanja poruke i ujedno verifikacije njenog sadržaja.

1.4 Puzzle-friendly

Za hash funkciju H kažemo da ima ovo svojstvo ukoliko je za svaku moguću n-bitnu vrijednost, kada je k izabran iz distribucije s velikom minimalnom entropijom, neisplativo tražiti x takav da je $H(k|x) = y$ u vremenu znatno manjem od 2^n.

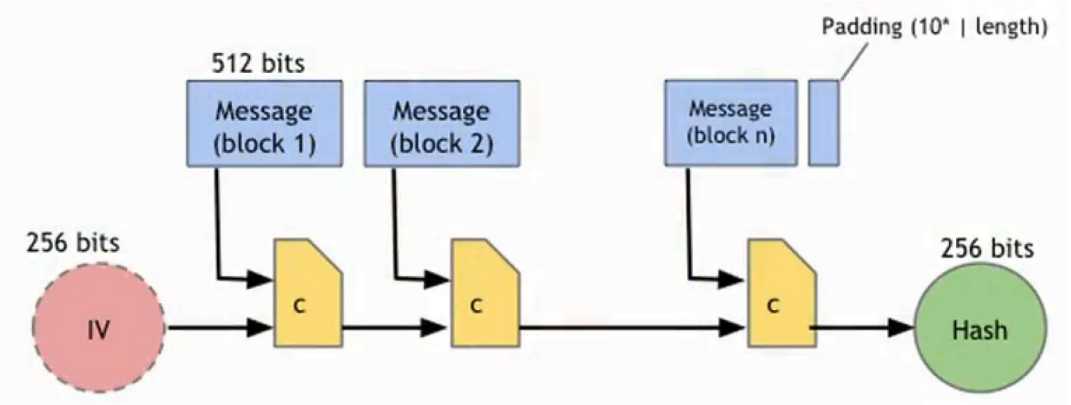
Ovo svojstvo hash-funkcija je vrlo korisno jer ćemo ga koristiti za izradu tzv. *search puzzle*-a, tj. napravit ćemo matematički problem koji zahtjeva pretraživanje ogromnog prostora u potrazi za rješenjem. Konkretnije, ideja je da nam je dan *"puzzle ID"* iz distribucije s velikom minimalnom entropijom i skup ciljanih vrijednosti Y. Želimo pronaći rješenje x, tj. djelujemo li hash-funkcijom na id ulančan s x, dobit ćemo rješenje u skupu Y

$$H(id|x) \in Y.$$

Dakle, Y je ciljano područje (skup rezultata koje želimo dobiti), id karakterizira određenu slagalicu (*puzzle*), a x je rješenje iste. Veličina skupa Y određuje u kojoj će mjeri slagalica biti neprobojna. Ukoliko bi neka vrijednost id-a bila vjerojatnija od drugih, bilo bi moguće računati slagalicu s tim id-om. No, to nije slučaj zbog činjenice da *"puzzle id"* ima veliku minimalnu entropiju. Zbog ovog svojstva nam nikakva posebna strategija neće biti učinkovitija od samo nasumičnog pogađanja x-a.

1.5 Secure Hash Algorithm-256

Postoji mnogo hash funkcija, a SHA-256 je jedna koju koristi Bitcoin.



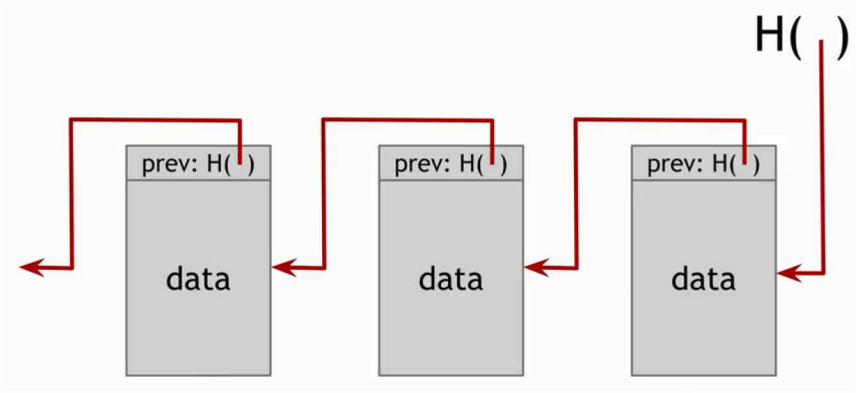
Slika 1.3: *Pojednostavljena SHA-256 hash funkcija*

U osnovi, ona uzima poruku koju želimo "hashati" i izlomi ju u blokove veličine 512 bita. U kriptografiji, inicijalni vektor (IV) ulaz je fiksne veličine od koga se obično zahtjeva da bude slučajan ili pseudoslučajan, što je ključno za postizanje semantičke sigurnosti, kako opetovana upotreba sheme pod istim ključem ne bi omogućila napadaču da otkrije odnose među dijelovima šifrirane poruke.
Veličina poruke neće nužno biti višekratnik broja 512, pa u posljednji blok dodajemo ispunu. Ona sadrži 64-bitni blok koji je jednak duljini poruke u bitovima, zapisanoj u binarnom sustavu. Na kraj poruke, prije 64-bitnog bloka, dodajemo bit "1" popraćen nekim brojem 0-bitova kojeg biramo tako da nam ispuna bude veličine točno jednog bloka ili višekratnik broja 512. Sada je poruka podjeljena na točno $n \in \mathbb{N}$ blokova veličine 512 bita. Počinjemo s 256-bitnom vrijednosti IV i prvim blokom poruke, odnosno ukupno 768 bitova te djelujmo na njih kompresijskom funkcijom koja će ih sažeti u 256 bita. Uzmimo njih i sljedeći blok poruke od 512 bita, djelujmo kompresijskom funkcijom i činimo to sve dok ne kompresiramo i ispunu te kao rezultat dobijemo 256-bitni hash. Također, ukoliko kompresijska funkcija c ima svojstvo da je 'collision-free', tada će i hash-funkcija imati to svojstvo.

1.6 Hash pointeri i strukture podataka

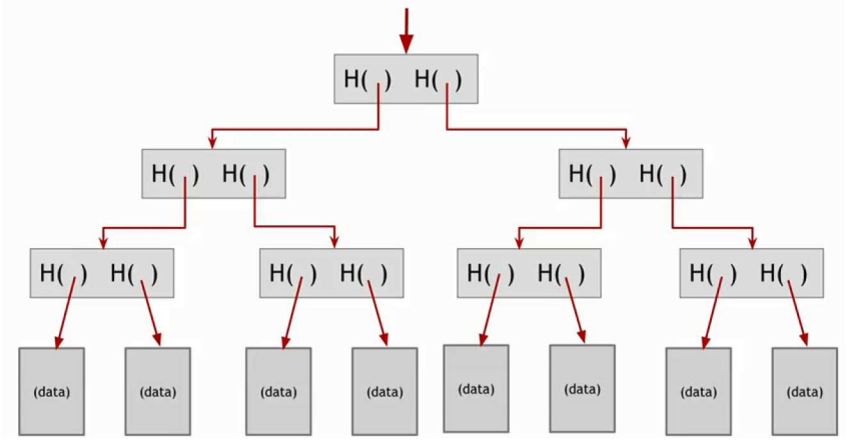
Dok nam regularni pointeri dohvaćaju informacije, hash pointere možemo zatražiti povratne informacije te nam dozvoljavaju da verificiramo da se informacije u međuvremenu nisu mijenjale.

Zašto ih koristimo?
Za izgradnju svih vrsta baza podataka. Poanta je da ne moramo pamtiti podatke koje stavljamo u strukturu, nego samo početni pointer na te podatke.



Slika 1.4: *Block-Chain*

Na slici je prikazana povezana lista sagrađena od hash pointera koju zovemo lanac blokova (eng. *block-chain*). Kada imamo ovakvu strukturu podataka svaki zlonamjerni napad, tj. mijenjanje podataka je vrlo lako uočljivo. Naime, pretpostavimo da netko želi potajno promijeniti sadržaj u bloku A. Tada se podaci iz tog bloka neće poklapati s $H()$ iz prethodnog bloka B. Napadač može mijenjati i $H()$-ove, i tako sve do početka strukture kako bi se pointeri poklapali s podacima, ali ne može promijeniti početni $H()$-glavu liste, što je zapravo i jedina vrijednost strukture koju pamtimo.



Slika 1.5: *Merkleovo stablo*

Merkleovo stablo je još jedna vrlo korisna struktura podataka načinjena od hash pointera. Na dnu sheme nalazi se mnoštvo podataka. Prva dva bloka povezujemo na

način da njihova dva pripadna hash pointera stavimo u jedan blok, a ta dva pointera imaju svoj zajednički pointer u bloku iznad. Lako je za zaključiti da ako napadač promijeni neki podatak na dnu, mora promijeniti i pointer na njega, zatim pointer na taj pointer, i sve tako do vrha stabla,osim početnog pointera kojeg ne može mijenjati, a on se neće poklapati s onima koje je napadač mijenjao i tako detektiramo pokušaj mijenjanja podataka.

1.7 Digitalni potpis i javni ključ kao identitet

Digitalni potpis digitalni je kod koji služi za zaštitu poruka koje se elektronički prenose putem javne mreže. Svrha digitalnog potpisa je omogućiti identifikaciju pošiljaoca te osigurati autentičnost sadržaja poruke.

```
(sk, pk) := generateKeys(keysize)

sig := sign(sk, message)

isValid := verify(pk, message, sig)
```

Slika 1.6: *Shema digitalnog potpisa (API)*

1. Funkciji *generateKeys* prosljedimo duljinu željenih ključeva u bitovima. Ona će stvoriti dva ključa sk (secret signing key) i pk (public verification key);
2. Funkciji *sign* prosljeđujemo sk i poruku koju želimo potpisati. Ona vraća sig, string bitova koji predstavlja potpis;
3. Funkciji *verify* prosljeđujemo javni ključ potpisnika, potpisanu poruku i ono što pretpostavljamo da je potpis. Ona nam vraća *true* ili *false*, ovisno je li potpis valjan za tu poruku.

U ovom dijelu bavit ćemo se jednim korisnim trikom vezanim za digitalni potpis. Ideja je da uzmemo javni ključ (eng. *public key*), *pk*, jedan od javnih verifikacijskih ključeva iz sheme digitalnog potpisa i izjednačimo ga s određenim identitetom (osobom, radnjom ili sustavom).
Prvo, pošiljatelj mora koristiti javni ključ primatelja, a primatelj mora imati vjerodostojnu kopiju javnog ključa pošiljatelja. Ovo je namijenjeno provjeri autentičnosti poruke, nitko ju ne može presresti. Kada napiše poruku, pošiljatelj izračunava hash poruke i šifrira ga svojim tajnim ključem. Kada mu poruka stigne, primatelj će vidjeti

javni ključ pošiljatelja i hash poruke, te će ga dešifrirati pomoću tog javnog ključa. Digitalni potpis omogućuje primatelju provjeru autentičnosti poruke na temelju javnog ključa, poruke i potpisa.
Razmišljamo li o javnom ključu kao o identitetu koji je u mogućnosti davanja tvrdnji, i ako znamo odgovarajući tajni ključ (eng. *secret key*), sk, tada te tvrdnje potpisujemo tajnim ključem, odnosno, jedino osoba koja zna tajni ključ je u mogućnosti davanja korektnih izjava u ime tog identiteta u sustavu. Smatranje javnog ključa identitetom nosi sljedeću posljedicu: možemo napraviti novi identitet (ili više njih) na način da kreiramo novi nasumični uređeni par ključeva (pk, sk), gdje je pk javno ime koje taj identitet koristi. U praksi se većinom za javno ime koristi $H(pk)$ jer su javni ključevi pk obično vrlo veliki.

1.7.1 Decentralizirani menadžment identiteta

Stvaramo li identitete na gore opisan način, ne trebamo se registrirati u nekom centralnom sustavu za registraciju korisnika. Ne moramo odabrati korisničko ime niti ikoga obavijestiti da ćemo djelovati pod tim korisničkim imenom. Svatko može napraviti novi identitet kad god želi i koliko ih god želi. U slučaju da želimo biti prepoznati s određenim imenima, napravili bi nove identitete s tim imenima. Isto tako, želimo li neko vrijeme biti anonimni, napravimo novi identitet s nasumičnim imenom, koristimo ga neko vrijeme i zatim odbacimo. U decentraliziranom sustavu ne postoji nikakva centralna kontrolna točka i upravo se takvim sustavom koristi Bitcoin, gdje su identiteti nazvani *adresama*.

1.7.2 Privatnost

Identiteti, odnosno adrese koje stvaramo nisu direktno povezane sa stvarnim identitetom osobe. Nekim algoritmom generiramo naizgled nasumični javni ključ pk i anonimno izvršavamo određene radnje. Ukoliko se češće koristimo istim imenom, neki bi promatrač mogao s vremenom primjetiti i povezati da su određene radnje rezultat aktivnosti iste adrese, pa potom donositi zaključke o nama na osnovu obrasca ponašanja što smanjuje razinu naše privatnosti.

Poglavlje 2

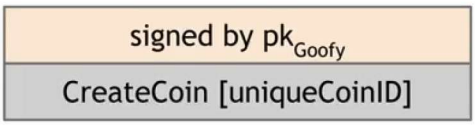
Jednostavne kriptovalute

U prvom poglavlju bilo je puno riječi o kriptografiji. To je podloga bez koje se ne može ako želimo govoriti o kriptovalutama i u nastavku će biti pokazano kako su svi ti dijelovi povezani i zašto su neke kriptografske operacije, poput hash funkcija i digitalnog potpisa, vrlo korisne.

2.1 GoofyCoin

Najjednostavnija fiktivna kriptovaluta je GoofyCoin. Način njegovog rada zasniva se na sljedećim pravilima:

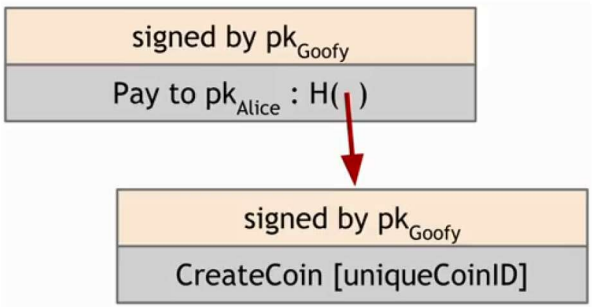
1. Goofy sam može stvarati nove novčiće. Može ih napraviti kad god želi, koliko god ih želi i svi novi novčići pripadaju njemu.



Slika 2.1: *Stvaranje GoofyCoina*

Kada Goody stvori novčić, on izgleda kao struktura podataka na slici 2.1. Imamo operaciju CreateCoin s jedinstvenim id-om kojeg je Goofy generirao, te Goofyjev digitalni potpis kojeg svatko može verificirati.

2. Svatko tko posjeduje GoofyCoin, može ga proslijediti nekome drugome, odnosno trošiti ga.

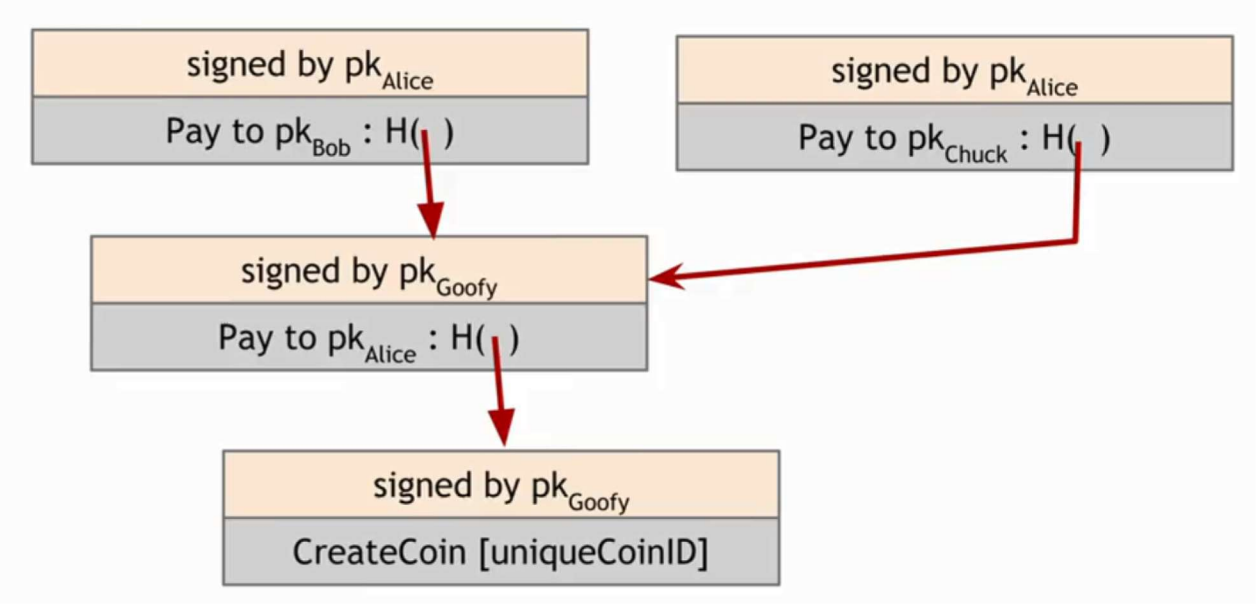


Slika 2.2: *Transakcija GoofyCoina*

Goofy može izjaviti da će novčić dati Alice (tj. javnom ključu s tim imenom). Gornja shema doslovno znači: *"Javnom ključu Alice isplati novčić reprezentiran hash pointerom"*. Primjetimo da se Goofy potpisao i na novčić kada ga je izradio i na transakciju koju je izvršio. Goofy mora potpisati svaku transakciju GoofyCoina. Alice može novčić predati drugoj osobi, a dokaz validnosti novčića je Goofyjev potpis.

2.2 Napad dvostrukim trošenjem

Obzirom na njegovu jednostavnost i vrlo jasna pravila, na primjeru GoofyCoina je najlakše objasniti što je dvostruko trošenje novčića (eng. *double-spending attack*).



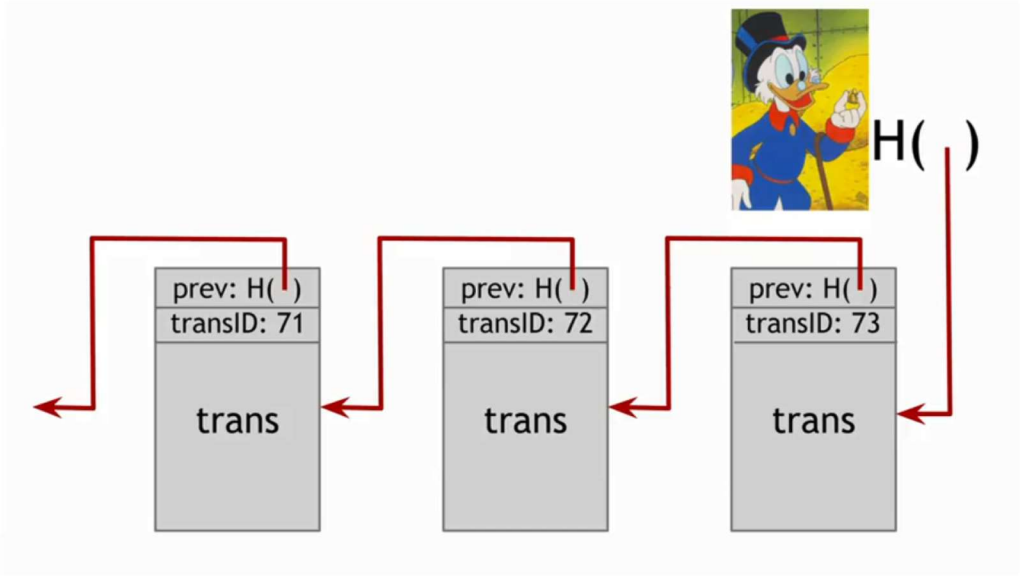
Slika 2.3: *Double-spending attack*

Pogledajmo sliku 2.3 i novčić lijevo u sredini, to je novčić koji je Goofy napravio i

predao ga Alice. Alice kao novi vlasnik novčića može novčić predati Bobu, ali može napraviti i novu strukturu (desno) i isti novčić predati Chucku. Pretpostavimo li da Chuck ne zna što se događa na lijevoj strani sheme, pogledat će da je taj novčić s Alicinim potpisom došao od Goofyja i zaključiti da je transakcija sasvim valjana te da je on novi vlasnik novčića. Također, ni Bob ne zna da Alice jedan novčić pokušava predati dvjema osobama, pa vodeći se istom logikom kao i Chuck smatra da je njegova transakcija s Alice valjana. Borba protiv ovakvih napada je zapravo najveći izazov pri dizajniranju održivih kriptovaluta.

2.3 ScroogeCoin

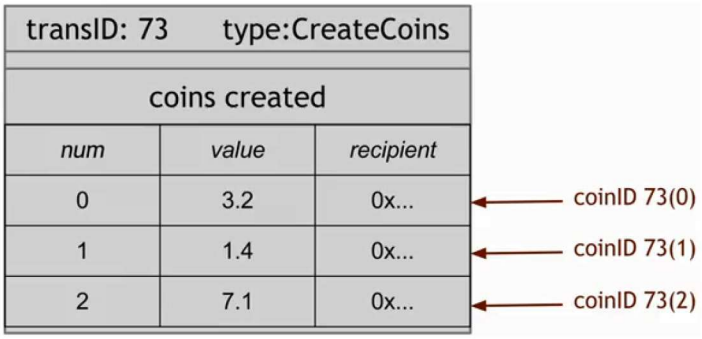
ScroogeCoin je kriptovaluta vrlo slična GoofyCoinu, a osnovna im je razlika to što je Scrooge na neki način spriječio napade dvostrukog trošenja. Scrooge je tvorac svakog novčića, potpisuje ga, te objavljuje povijest svih transakcija nekog novčića. Tako možemo provjeriti je li u nizu transakcija bilo napada dvostrukog trošenja.



Slika 2.4: *Transakcije ScroogeCoina*

Dvije su vrste transakcija ScroogeCoina:

1. *CreateCoins* transakcija koja stvara nove novčiće i sasvim je valjana jer je prvo Scroogeovo pravilo, kao i Goofyjevo, da može sam stvarati novčiće kad god i koliko god želi.



Slika 2.5: *CreateCoins transakcija*

Pogledajmo sliku 2.5. Dozvolili smo da se u jednoj transakciji stvori više novčića. Svaka ovakva transakcija ima svoj ID (u ovom primjeru je to 73), *num* predstavlja serijski broj proizvedenog novčića, *value* je vrijednost pojedinog novčića u ScroogeCoinima, a *recipient* predstavlja javni ključ primatelja novčića nakon njegovog stvaranja. Dakle, svakom novonastalom novčiću biti će pridružen broj transakcije u kojoj je nastao i serijski broj u toj transakciji.

2. *PayCoins* transakcija troši novčiće, uništava ih i zatim stvara nove novčiće u istoj vrijednosti.
Pravila koja transakcija mora zadovoljavati da bi bila važeća su:

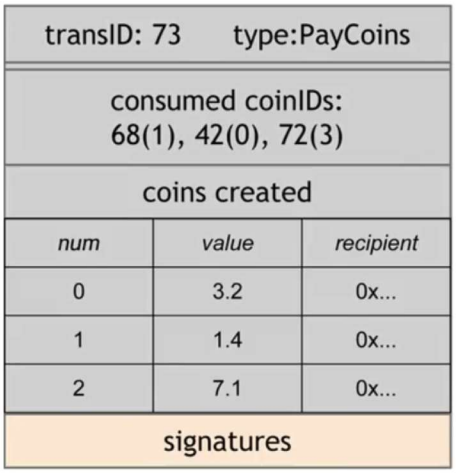
-Trošeni novčići moraju biti važeći

-Isti novčići ne smiju biti prethodno trošeni (sprječava dvostruko trošenje)

-Ukupna vrijednost novčića koji ulaze u transakciju jednaka je vrijednosti novčića koji izlaze iz transakcije

-Transakcija je potpisana od strane svih stranaka koje zajedno troše novčiće

Ako je transakcija valjana, Scrooge će ju prihvatiti, potpisati i upisati ju u povijest transakcija u blockchainu. Tako će svi moći vidjeti da se ova transakcija dogodila.



Slika 2.6: *PayCoins transakcija*

Consumed coinIDs predstavlja listu novčića koje ćemo potrošiti.
Bitno je napomenuti da je svaki novčić sam po sebi nepromjenjiv, tj. ne možemo mijenjati njihovu vrijednost, jedan novčić ne može biti podijeljen na više dijelova, i ne možemo ih kombinirati. Navedene stvari možemo učiniti samo nizom transakcija. Npr ukoliko želimo podijeliti jedan novčić na dva, možemo kreirati transakciju koja će od jednog napraviti dva nova novčića iste vrijednosti, a ukupne vrijednosti kao početni novčić, te zatim isplatimo ta dva novčića sami sebi.

Jedini problem na koji nailazimo kod ScroogeCoina jest Scrooge, odnosno centralni autoritet koji se uvijek može početi ponašati onako kako to korisnicima ne odgovara. Tako nailazimo na tehnički izazov, stvaranje decentraliziranog sustava koji će se djelomično ponašati kao Scrooge, odnosno centralizirani sustav.

Poglavlje 3

Kako je Bitcoin postigao decentralizaciju

3.1 Centralizacija i decentralizacija

Decentralizacija je vrlo bitan koncept kriptovaluta, ne samo Bitcoina. Pojam konkurentnih modela centralizacije naspram decentraliziranih počiva na raznim vrstama digitalne tehnologije. Prvo je potrebno spomenuti da skoro ne postoji sustav koji je u potpunosti decentraliziran ili centraliziran. Kao primjer takvoga je e-mail, u osnovi decentralizirani sustav, u kojem se u zadnjem desetljeću pojavilo nekoliko dominantnih web-mail poslužitelja, koji su jedna vrsta centraliziranog sustava.

Promotrimo tri različita aspekta Bitcoina i pogledajmo kada bi ih svrstali u centralizirani ili decentralizirani spektar:

1. *Peer-to-peer network (P2P)*

Najviše decentraliziran aspekt. Svatko može pokrenuti Bitcoin čvor (eng. node) s eventualno vrlo malim tehničkim preprekama. U P2P sustavima komunikacija između čvorova u mreži se obavlja bez poslužitelja, a svi čvorovi su neovisni i ravnopravni. Čvorovi u P2P sustavima dijele raspoložive resurse, procesorsko vrijeme, korisničke podatke, mrežnu propusnost, što je ujedno i svrha ovakvog sustava.

2. *Mining* (rudarenje)

Također aspekt otvoren svima, ali podrazumijeva neizbježne i velike potrošnje električne energije, odnosno, skupo je i time nije u potpunosti dostupno svima, odnosno decentralizirano.

3. *Updates to the software*

Zajednica mora vjerovati glavnim programerima (*core developers*), koji zato imaju veliku moć. Svatko tko posjeduje Bitcoin može kreirati vlastiti software, što bi značilo

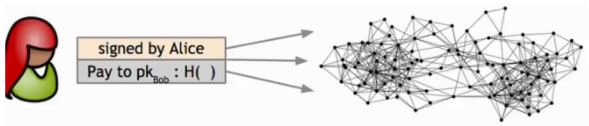
da je ovaj aspekt decentraliziran, ali to u praksi nije tako, jer se zajednica uglavnom oslanja na glavne programere koji odlučuju kakav će software pokrenuti.

3.2 Raspodjeljeni konsenzus (eng. distributed consensus)

Na tehničkoj razini, glavni izazov koji moramo proći da bi napravili rapodjeljeni, npr. e-cash sustav je raspodjeljeni konsenzus. To je zapravo klasa protokola na koje se korisnici mogu osloniti kao sigurne.
Kompanije poput Googla, Facebooka i sl. imaju milijune servera koji čine ogromnu raspodjeljenu bazu podataka sa svim aktivnostima korisnika. Recimo da želimo ostaviti novi komentar na Facebooku. On će biti zapamćen u 10-ak različitih čvorova. Sustav se mora uvjeriti da je komentar zapamćen u svim kopijama te baze podataka, ili u nijednoj. Ako nije uopće zapamćen, korisnika će se zatražiti da ponovo napiše komentar.

Neka je $n \in \mathbb{N}$ fiksan broj čvorova ili procesa u obradi i neka svi čvorovi imaju istu vrijednost inputa. Konsenzusni protokol završava u nekom trenutku i ocjenjuje neke čvorove ispravnima (i/ili nezlonamjernima), koji će imati zajedničku vrijednost (eng. *consensus value*). Ta vrijednost nije proizvoljan broj, nego vrijednost inputa jednog od čvorova.

3.2.1 Raspodjeljeni konsenzus i Bitcoin



Slika 3.1: *Struktura transakcije*

Prisjetimo se da je Bitcoin P2P sustav, što znači da kada Alice želi platiti Bobu, objavit će transakciju svim čvorovima uključenima u P2P mrežu. Alice potpisuje transakciju kako bi svi čvorovi znali odakle je došla, a ona će imati Bobov javni ključ, tj. adresu koja treba primiti Bitcoine. Hash u ovoj transakciji je poveznica s prijašnjom transakcijom tog novčića, onom gdje je novčić pripao Alice.

U mreži se nalazi mnoštvo korisnika koji mreži objavljuju transakcije. Korisnici teže postizanju konsenzusa vezanog za: koje će transakcije biti objavljene i kojim redoslijedom će se odvijati.

Za postizanje konsenzusa potrebno je u prosjeku sat vremena, ali i po završetku tog vremena ne možemo biti sasvim sigurni u njegovo postizanje. Umjesto toga, kako vrijeme prolazi, vjerojatnost da je postignut je sve veća i veća, točnije, vjerojatnost da nije postignut pada eksponencijalno.

3.3 Konsenzus bez identiteta

U decentraliziranom P2P sustavu ne postoji centralni autoritet koji bi čvorovima pridruživao identitete. Anonimnost je sama po sebi glavna poanta Bitcoina. Za sudjelovanje u P2P mreži u blockchainu, korisnik nije obvezan koristiti se svojim stvarnim identitetom, IP adresom, i sličnim, pa se većina koristi pseudonimima.

Kada govorimo o Bitcoin konsenzusu, on podrazumijeva sljedeće:

-*Append-only ledger* - struktura podataka u koju je moguće samo upisivanje podataka, a sve što je zapisano ostaje zauvijek u, šturo rečeno, glavnoj knjizi transakcija;

-Decentralizirani konsenzus;

-Rudari (eng. *miners*) provjeravaju valjanost transakcija, jesu li dobro formirane, postoji li mogućnost napada dvostrukog trošenja, te za to dobivaju određene naknade.

Poglavlje 4

Transakcija Bitcoina

4.1 Novčanik

Sama činjenica da postoje transakcije kriptovaluta povlači egzistenciju nečega što bi čuvalo i brojalo ono što posjedujemo. Upravo to je novčanik (eng. *wallet*), odnosno računalni program zadužen za slanje, primanje te skladištenje kriptovaluta, pa tako i Bitcoina. Novčanici funkcioniraju vrlo slično kao i bankovni računi. Prikazuju stanje računa, popis obavljenih transakcija te daju mogućnost stvaranja drugih transakcija. Karakterizira ih laka dostupnost, tj. moguće ih je instalirati u obliku aplikacija na mobilne uređaje, računala ili tablete, ili ih možemo koristiti putem određenih web servisa. Svatko može imati više novčanika, a svaki novčanik može imati više neovisnih podračuna, odnosno adresa sa svojim zasebnim stanjem. Stanje novčanika čini zbroj svih stanja svih pripadnih adresa. Zbog mogućih tehničkih kvarova ili pokušaja krađe uređaja na kojima su instalirani novčanici, ne preporuča se držanje veće svote Bitcoina u jednom novčaniku. U slučaju kvara ili krađe novčanika, korisnik bespovratno gubi sve što se u njemu nalazilo.

4.2 Bitcoin transakcija

Bitcoin transakcija je transakcija u Bitcoin mreži koji znači da se određeni iznos Bitcoina prenosi s jedne adrese na drugu, a ona je javna, transparentna i anonimna. Kada korisnik, odnosno novčanik, kreira novu transakciju i pošalje ju u mrežu, ona je je nevažeća i nepotvrđena sve dok rudar blok s tom transakcijom ne doda u blockchain. To je vrsta potvrde da je s transakcijom sve u redu. Većina sustava smatra transakciju valjanom nakon što dobije barem 3 potvrde. Kako bi osigurali da naša transakcija čim prije bude dodana u blockchain, korisnici rudarima daju naknade (eng. *transaction fee*) u vidu male količine Bitcoina. Veća naknada privući će više rudara, transakcija će biti

brže dodana u blockchain i imat će više potvrda valjanosti, koje ostalim korisnicima daju određeno povjerenje u nečije pošteno trgovanje. Transakciju možemo obaviti s jedne ili više adresa na jednu ili više adresa.

Nakon što korisnik transakciju pošalje u mrežu, prvi čvor koji ju primi potvrđuje njenu valjanost, a onda ju prosljeđuje drugim čvorovima. Pri tome, prvi čvor mora provjeriti vrijedi li sljedeće:

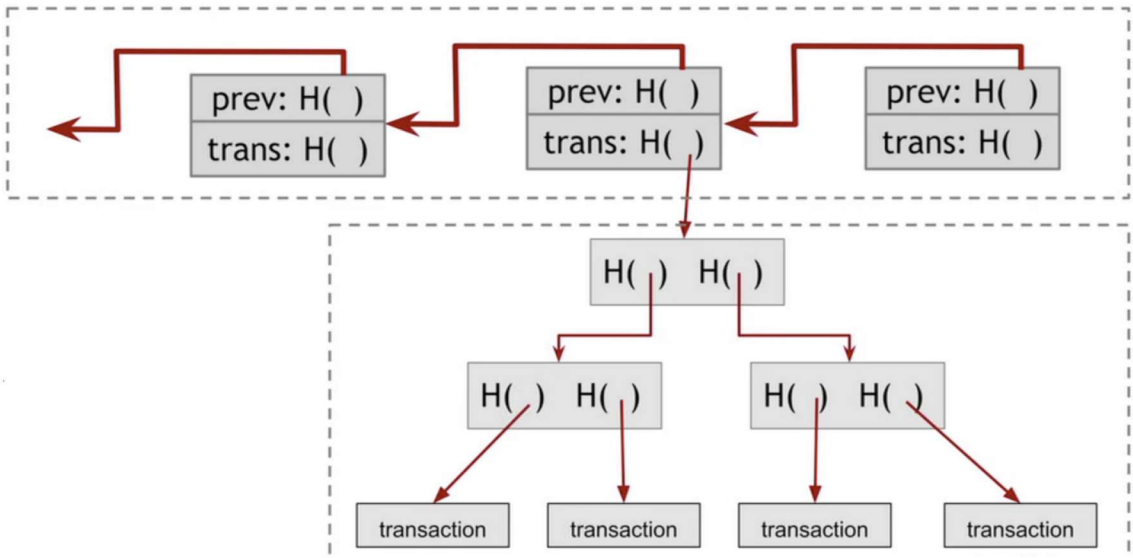
-Izlazne transakcije s adrese slanja moraju biti u potpunosti utrošene (radi računalne učinkovitosti);

-Je li suma vrijednosti ulaza veća ili jednaka sumi vrijednosti izlaza, jer korisnik ne može nekome predati više Bitcoina nego što ima na toj adresi. Razlika između te dvije sume čini naknadu za rudara;

-Valjanost potpisa za svaki ulaz.

4.3 Bitcoin blokovi

Obzirom da svaka transakcija u blockchainu mora imati potvrdu valjanosti, možemo i sami zaključiti da je rudarima isplativije potvrđivati valjanost jedne cijele grupe transakcija, tj. bloka, nego pojedinačno. Također, jedan blok s velikim brojem transakcija zahtjeva samo jedan hash, umjesto njih isto toliko mnogo. Ovakva struktura podataka omogućava puno lakše praćenje i potvrđivanje valjanosti transakcija.



Slika 4.1: *Struktura Bitcoin bloka*

Na vrhu ove strukture vidimo lanac hasheva pripradnih blokova. Svaki od njih ima zaglavlje i pokazivač na prijašnji blok u nizu, te pokazivač na podatke o nekoj transakciji. Transakcije koje pripadaju istom bloku povezane su već spomenutim Merkleovim stablom. U zaglavlju se nalaze još neke bitne informacije, npr.: vremenske oznake, naznake koliko je bilo teško pronaći taj blok, itd. Jedini podatak sadržan u zaglavlju je jedna grana Merkleovog stabla, koja nam je dovoljna kako bismo utvrdili valjanost tog bloka transakcija. Dakle, ukoliko želimo provjeriti valjanost bloka, lako ćemo provjeriti postoji li hash u zaglavlju. Najvažnija stvar u ovoj strukturi je da hash u zaglavlju bloka mora počinjati velikim brojem nula kako bi blok bio valjan. To naznačava stvaranje novih novčića, tj. da novčići nisu stvoreni prethodnim transakcijama (već trošeni).

Poglavlje 5

Pohrana i korištenje bitcoina

5.1 Jednostavno lokalno pohranjivanje i wallet software

Ukoliko namjeravamo trošiti svoje novčiće, moramo znati sljedeće:
- Neke informacije iz javnog lanca blokova (blockchaina), tj. identitet novčića i npr. njegovu vrijednost
- Tajni potpisni ključ vlasnika Bitcoina (pretpostavimo da smo to mi).

Kada bolje razmislimo, najviše računa moramo voditi o drugonavedenoj stvari. U praksi, kada se govori o načinu skladištenju Bitcoina, zapravo se govori o načinu skladištenja i organiziranja ključeva.
Primarni ciljevi pri takvom organiziranju su:
- Dostupnost - možemo potrošiti novčiće kad god to želimo
- Sigurnost - nitko drugi ne može trošiti naše novčiće
- Udobnost - što god da radimo je relativno jednostavno za korištenje

Najjednostavniji pristup organiziranju ključeva je spremanje ključeva u mapu i stavljanje mape na lokalni uređaj (bilo koji gadget). Ovakav pristup nam daje maksimalnu udobnost, tj. pritiskom nekoliko tipki ili jednostavnim prelaskom prsta preko ekrana možemo trošiti Bitcoine. Isto tako vrlo lako i jednostavno uređaj se može pokvariti ili biti ukraden, što rezultira bespovratnim gubitkom svega što smo na njemu imali. Ključevi, odnosno vaši novčići su sigurni onoliko koliko je siguran uređaj. Ako se uređaj zarazi nekim zlonamjernim softwareom, on novčiće može prebaciti s jednog računa na drugi.

Korisnici se najčešće koriste *wallet softwareima*, softwareima koji organiziraju sve detalje vezane za ključeve i tako cijeli proces čine vrlo udobnim za korisnika. Oni prate

što se zbiva s novčićima i obično pružaju pregledna korisnička sučelja. Također, wallet software nam dozvoljava korištenje velikog broja različitih adresa, pa umjesto da držimo sve novčiće na jednom mjestu koje kontroliramo samo jednim ključem, možemo ih rasporediti tako da imamo različitu adresu i različiti ključ za svaki novčić. Time smo vlastitu anonimnost i privatnost podigli na najviši stupanj.

5.2 Kodiranje adresa

Ukoliko želimo primiti Bitcoine kao način plaćanja ili ih nekome isplatiti, moramo nekako razmijeniti adresu s drugom stranom. Dakle, moramo dati ili primiti adresu, a za to postoje dva glavna načina:

1. Kodiranje adrese u obliku tekstualnog stringa: uzimamo bitove ključa i kodiramo ih u notaciji s 58 znakova. Uključeni su brojevi od 1 do 9 te velika i mala slova, s tim da se ne koristimo slovima i brojevima koji su slični (npr. nula i veliko slovo "O", broj jedan i malo slovo "l").
2. Korištenje QR koda: mobilni uređaji skeniraju ovaj jednostavni 2D barkod i odvedu nas na adresu koju bitovi barkoda predstavljaju. Ovakav način plaćanja ima sve širu upotrebu u uslužnim djelatnostima, pa čak i u nekim bolnicama.

5.3 Naknade za transakcije

Prisjetimo se, naknada za transakciju definirana je kao razlika vrijednosti inputa i outputa, tj. razlika ukupne vrijednosti novčića koji ulaze u transakciju i ukupne vrijednosti isporučenih novčića. Vrijednost inputa trebala bi biti barem onolika kolika je vrijednost outputa jer regularnim transakcijama ne možemo stvarati dodatne novčiće. Razlika vrijednosti inputa i outputa pripada rudaru koji je napravio blok u kojem je ta transakcija zabilježena, jer on za to mora uložiti određene resurse. Naravno, korisnici nisu prisiljeni davati nikakve naknade za transakcije, no takve transakcije neće privući rudare. U pravilu, ako plaćamo veću naknadu za transakciju ona će brže biti prenesena i zabilježena te pouzdanija.

U slučaju da je ukupna vrijednost transakcije veličine manje od 1000 bytea, svi outputi su barem 0.01 BTC (što na dan 14. rujna 2019. vrijedi 103.57 USD), i ukoliko je *prioritet* transakcije (koji se računa po određenoj formuli) dovoljno velik, tada se ne plaća naknada za transakciju.
Prioritetna formula glasi ovako:
Pomnožimo svaki input transakcije s njegovom vrijednošću u BTC, sve te produkte

zbrojimo, pa podijelimo s veličinom transakcije:

$$\frac{sum(input \cdot value)}{trans\ size}.$$

Naknada iznosi približno 0.0001 BTC po 1000 bytea.
Većina rudara daje prvenstvo transakcijama s plaćenom naknadom ili onima s većim prioritetom prema navedenoj formuli. Ukoliko bismo željeli napraviti transakciju koja ne uključuje naknadu, ona će vjerojatno naći svoj put do blockchaina, iako puno sporije. Iz tog razloga većina wallet softwarea automatski uključuje naknadu u transakciju.

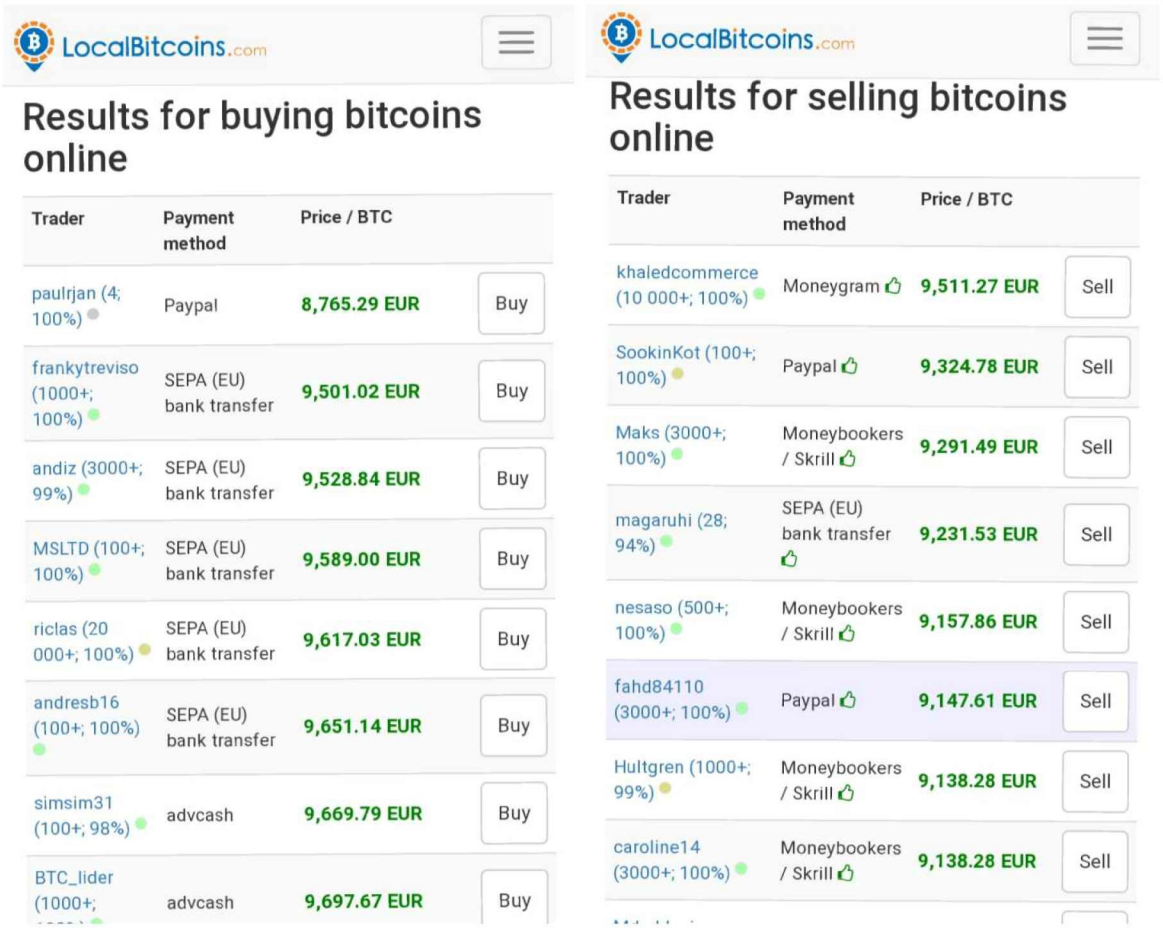
5.4 Tržišta razmjene

Tržišta razmjene kriptovaluta tržišta su na kojima se valute poput Bitcoina razmjenjuju s "tradicionalnim" valutama poput dolara, eura i sl.

Symbol	Latest Price	30 days	Average	Volume	Low/High	Bid	Ask	24h Avg.	Volume	Low/High
BitStamp USD	10350.01 0 min ago		10221.91 122.10 1.26%	190,909.90 1,951,463,287.17 USD	9320 10955.48	10351.46	10360.37	10305.22 44.79 0.44%	2,950.13 30,401,712.48 USD	10153.87 10422.79
Kraken USD	10355.1 ju ctno::		10203.67 151.43 1.48%	156,178.32 1,593,693,616.7 1 USD	8000 12000	10355	10356.3	10289.43 86.87 0.84%	3,334.95 34,214,747.00 USD	8000 12000
Coinsbit USD	10422.81407036 ju ctno::		10218.02 204.79 2.00%	139,969.22 1,430,203,934.62 USD	9154 14266.26626627	10350	10436	10364.79 58.02 0.56%	6,980.90 72,366,661.28 USD	10284.63 10454.02010051
Kraken EUR	9347.6 0 min ago		9219.80 127.80 1.39%	121,092.99 1,116,463,429.62 EUR	8407.5 9927.6	9343.3	9344.7	9301.33 46.27 0.50%	1,953.63 18,171,382.77 EUR	9198 9393.1
CoinsBank GBP	8231.24 0 min ago		8378.18 -146.94 -1.76%	115,067.65 964,067,269.37 GBP	7572.61 9120.42	7932.29	8560.14	8238.25 -7.01 -0.08%	3,741.45 30,822,001.42 GBP	8170.18 8560.14
coincheck JPY	1118891 0 min ago		1090486.70 28404.30 2.61%	56,201.49 61,226,978,000.65 JPY	990342 1171175	1118244	1119188	1114740.37 4160.63 0.37%	1,177.70 1,312,224,852.27 JPY	1100000 1128786
BitBay PLN	40650 0 min ago		40403.75 246.26 0.61%	50,718.21 2,048,206,332.30 PLN	37399.99 43120.78	40600.01	40650	40389.16 260.34 0.65%	1,009.06 40,764,921.66 PLN	39900.19 40650
CoinsBank USD	10358.2 0 min ago		10203.23 164.97 1.62%	39,745.55 405,633,040.62 USD	9342.06 11340.58	10236.57	10468.21	10303.54 54.69 0.53%	1,296.52 13,355,756.17 USD	10167.86 10438.99
Bitstamp EUR	9361.03 0 min ago		9213.96 147.07 1.60%	37,320.94 343,873,897.67 EUR	8421 9928.65	9343.23	9358.99	9288.39 72.64 0.78%	473.28 4,396,899.27 EUR	9163.65 9405.52
BTCBOX JPY	1119058 0 min ago		1085301.97 33768.03 3.11%	33,035.68 36,563,837,366.97 JPY	926104 1170000	1118600	1120163	1114009.56 5048.44 0.46%	985.78 1,095,163,446.22 JPY	1100000 1125904
LakeBTC.com USD	10694.49 1 min ago		10563.41 131.08 1.24%	31,861.06 336,661,287.78 USD	9668.84 11285.44	10696.7	10696.77	10634.40 60.09 0.57%	665.35 7,076,670.29 USD	10504.07 10736.66
Zaif JPY	1118035 0 min ago		1090402.85 27632.16 2.53%	29,294.92 31,940,261,169.47 JPY	991900 1170000	1119275	1119675	1114000.30 4054.70 0.36%	735.65 818,810,799.33 JPY	1099700 1125505
CoinsBank EUR	9375.91 22 min ago		9223.46 152.46 1.66%	22,322.17 206,327,726.61 EUR	8391.9 10130.73	9241.25	9466.11	9306.96 68.96 0.74%	680.33 6,331,790.31 EUR	9153.33 9471.06

Slika 5.1: `bitcoincharts.com/markets`; screenshot na dan 14. rujna 2019.

Web stranice za trgovanje većinom pružaju praćenje vrijednosti valuta uživo, odnosno promjenu cijene u realnom vremenu. U slučaju da korisnik ne želi trgovati online, postoje stranice koje mu pomažu da se fizički sastane s osobama koje prodaju ili kupuju Bitcoine.



Slika 5.2: `localbitcoins.com`; Kupci i prodavači u Hrvatskoj; screenshot na dan 14. rujna 2019.

Tržište ove vrste relativno je veliko, dnevno se proda ili kupi kriptovaluta u prosječnoj vrijednosti od 1 000 000 USD. Cijene se formiraju kao i na svakom tekućem tržištu, na osnovi ponude i potražnje.

Ponuda Bitcoina uključuje Bitcoine u optjecaju i tražene depozite. Trenutno je u optjecaju 17.9 milijuna BTC-a (stanje 14. rujna 2019.), a taj broj će polako rasti i s vremenom dosegnuti limit od 21 milijun BTC-a.

Dva su glavna izvora potražnje Bitcoina:

- *Potražnja koja zahtjeva fiat novac*[1]: Zamislimo da Luka želi kupiti neki proizvod od Marka koji živi jako daleko i mora mu za to prebaciti određenu svotu novca, npr. dolara. Oni smatraju da je prikladnije koristiti Bitcoin za takav transfer jer će, npr. naknada za transakciju biti puno manja nego što zahtjevaju neki drugi servisi. Luka prvo mora kupiti Bitcoine za dolare, zatim ih šalje Marku Bitcoin transakcijom, a onoga trenutka kada transakcija bude upisana u blockchain i potvrđena, Marko će moći prodati Bitcoine za dolare. Ova transakcija počinje i završava s dolarima, a njena ključna stvar je (Lukina) potražnja za BTC-om. Ovakve transakcije stvaraju protok BTC-a.

- *Potražnja u obliku investicije*: Luka želi kupiti Bitcoine i zadržati ih u nadi da će im cijena u budućnosti rasti, te će ih tada preprodati i zaraditi. Bitcoine u ovoj svrsi ne smatramo dijelom protoka.

Ovako izgleda pojednostavljeni model posredovanja, gdje smo pretpostavili sljedeće parametre:
T = ukupna vrijednost transakcije određene BTC-om [$/sec];
D = vrijeme koje ta količina nije u optjecaju, tj. vrijeme koje mora proći od trenutka kada kupimo BTC do trenutka kada ga primimo i budemo u mogućnosti preprodati ga, što obično traje nekoliko sekundi [sec];
S = broj svih BTC-a u opskrbnom lancu, ne računajući one u dugoročnim investicijama.
Tada je

$$ponuda = \frac{S}{D}[n/sec],$$

što je broj dostupnih BTC-a po sekundi, odnosno s koliko njih se posreduje u jednoj sekundi;

$$potražnja = \frac{T}{P}[BTC/sec],$$

što je broj BTC-a za kojima se potražuje u jednoj sekundi, gdje je P = cijena dolara po BTC-u.

Promotrimo li jednu sekundu te ponudu i potražnju u toj sekundi, vidimo da je tržište ove vrste vrlo dinamično kao i puno drugih tržišta, tj. ponuda oscilira u skladu

[1] Fiat novac je novac koji nema (ili ima vrlo malu, u suštini zanemarivu) unutarnju vrijednost, a ipak obnaša sve funkcije novca. Riječ "fiat" označava obavljanje te funkcije zahvaljujući državnoj odredbi. Iako fiat novac nema pokriće u zlatu, niti pokriće u nekim drugim vrijednostima, prima se s povjerenjem da će se s njime i dalje moći služiti kao novcem. U suvremenim uvjetima, primjeri fiat novca su papirnati novac i sitni kovani novac.

s potražnjom. Ukoliko je ponuda veća od potražnje, tada postoje neprodani Bitcoini, pa će osobe koje ih namjeravaju prodati spustiti cijenu. Isto tako, poveća li se broj ljudi koji žele izvršavati transakcije koristeći BTC, njega neće biti dovoljno u optjecaju za sve korisnike, koji će za njega početi nuditi više novca. Kada izjednačimo ponudu i potražnju, dobivamo *formulu ravnoteže*

$$P = \frac{T \cdot D}{S}.$$

D je vrijeme za kojega moramo zadržati BTC prije nego što ga iskoristimo u transakciji, dakle ono se ne može mijenjati, kao što se ni broj svih BTC-a u opskrbnom lancu ne može mijenjati (ako zanemarimo vrlo sporo stvaranje novih BTC-a). Možemo zaključiti da će tada cijena biti proporcionalna potrebi za stvaranjem transakcija. U slučaju da sve više ljudi počne kupovati BTC-ove radi investiranja, smanjit će se broj BTC-a dostupnih za transakcije, S, te će mu cijena rasti.

Zaključak

Kriptovalute koriste šifriranje za generiranje novca i omogućavanje sigurnih transakcija. Omogućuju elektroničko trgovanje izvan regulatornog okvira treće strane. Implementacija takvog sustava smatrala se teškim zadatkom zbog problema s dvostrukom potrošnjom, koji je od početka stvarao napore da se elektronički novac stvori na webu. Dakle, to je uglavnom bio teorijski koncept s ograničenom praktičnom primjenom sve do revolucionarne *Bijele knjige* takozvanog *Satoshija Nakamotoa* iz 2009. godine.
Obećanje o nižim naknadama za transakcije nego na standardnim mehanizamima za internetsko plaćanje i činjenica da njime upravlja decentralizirani sustav, tj. izvan vlade ili regulatornog tijela osiguralo je da Bitcoin vrlo brzo pronađe put do velikog broja korisnika. Ali što je još važnije, opskrba Bitcoinom kontrolira se na način koji osigurava ograničenu opskrbu na tržištu, što je jedan od glavnih uzroka trenutno visokih cijena. Danas je tržišna kapitalizacija kriptovaluta veća od 180 milijardi USD, a dnevno se obavi približno 300,000 transakcija BTC-a. U petak 6. rujna 2019. zabilježen je najveći BTC transfer ikad vrijedan 94,504 BTC-a, odnosno 1,018,147,900 USD, potpuno anoniman i bez ikakve naknade. Prema *Coindesk*-u, ova je transakcija podigla cijenu BTC-a s 10,569 USB na 10,790 USD. Uz ovakva saznanja možemo zaključiti koliko će ovakva tehnologija biti razarajuća u smislu međunarodnih platnih sustava.

Literatura

[1] A.M. Antonopoulos, *The Internet of Money*, 4^{th} print, 2017.

[2] A.M. Antonopoulos, *Mastering Bitcoin Unlocking digital cryptocurrencies*, California O'Reilly Media, Sebastopol, 2014.

[3] Đ. Borozan, *Makroekonomija* , Ekonomski fakultet u Osijeku (3. izmijenjeno izdanje), Osijek, 2012.

[4] J.G. Brookshear, D. Brylow, *Computer Science: An Overview*, Pearson Education Limited (12. izdanje), England, 2015.

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

[6] S. Nakamoto, *Bitcoin: A peer-to-peer Electronic Cash System*, 2008.

[7] A. Narayanan, J. Bonneau, E. Felten, A. Miller, S. Goldfeder , *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*, Princeton University Press, 2016.

[8] ISO/IEC 10116:2017, *Information technology — Security techniques — Modes of operation for an n-bit block cipher*, Switzerland, 2017.

[9] THE INFORMATION WARFARE SITE
`http://www.iwar.org.uk/comsec/resources/cipher/sha256-384-512.pdf`

[10] WHALE ALERT
`https://whale-alert.io/transaction/bitcoin/`
`4410c8d14ff9f87ceeed1d65cb58e7c7b2422b2d7529afc675208ce2ce09ed7d`