Digitalni Denar: Uvod v Bitcoin

Tilen Kavčič

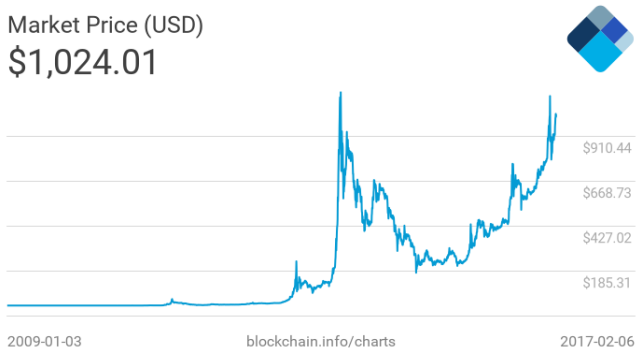
***Povzetek*— Digitalni denar se je v zadnjih letih izkazal za stabilno in verodostojno valuto za trgovanje. Med najuspešnejšimi je BitCoin s katerimi poslujejo največja podjetja in organizacije (Wikipedia, Tesla, Microsoft, Dell, itd.) Članek se posveča delovanju BitCoina vendar se lahko glavne ideje prenesejo tudi na drugi digitalni denar. Najprej je predstavljena kratka zgodovina digitalnega denarja in motivacija za pomembnost decentralizacije digitalne denarja na primeru. Nato sledi predstavitev podatkovnih struktur, kriptografskih rešitev, P2P omrežja, napadov na omrežje in rudarjenje. Na koncu je predstavitev lastnega digitalnega denarja (FRIKOIN) in prihodnost digitalnega denarja.**

I. UVOD

S porastom spletnih trgovin je uporaba digitalnega denarja postala vedno bolj priljubljena. Digitalni denar je zelo podoben fizičnemu denarju, vendar se razlikuje v takojšnem prenosu lastništva sredstev. Digitalni denar delimo na kripto valute in virtualne valute. Virtualna valuta je zaprtega tipa, omejena na določeno tržišče, ki ga ima pod nadzorom podjetje. Z virtualno valuto se lahko kupuje le na zaprtih tržiščih. Kripto valute so zelo podobne pravim valutam. Močno se upirajo na kriptografijo, "peer-to-peer"omrežja in decentralizacijo. V članku se bomo posvečali Bitcoinu saj je bila prva kripto valute, ki je rešila težavo z decentralizacijo valute in s tem postavlja temelje za vse ostale kripto valute.

Leta 2008 se je nekdo pod imenom Satoshi Nakamoto objavil članek z naslovom: "Bitcoin: A Peer-to-Peer Electronic Cash System"[4]. Članek opisuje način, kako poslovati z digitalnem denarjem brez predhodnega zaupanja. Rešil je tudi problem dvojnega zapravljanja z verigo blokov (eng.: "blockchain"). Kupna vrednost 1 BTC, leta 2009, ko so

Fig. 1. Kupna vrednost 1 BTC [9]



objavili prvo odprto kodno implementacijo Bitcoina, je bila zelo nizka. Prvih par blokov je ustvaril Satoshi Nakamoto na katerih temeljijo vse transakcije. S polaganjem temeljev je Satoshi zaslužil okoli 1 milijona BTC. Cena BTC je z leti počasi naraščala. Kljub temu, da se Bitcoin izkazal za resno valuto, ki jo podpirajo velika podjetja (npr.: Tesla, Microsoft, Valve, itd.) so nekatere države in institucije še vedno skeptične zaradi anonimnosti valute. Centralizirane valute so pod nadzorom inštitucij, ki nadzorujejo njen obtok. Dobri primer tega sistema so centralne banke, ki imajo pod nadzorom Euro. Težave centralizacije je anonimnost uporabnikov. Zaupanje in količina valute v obtoku da valuti vrednost. Centralna inštitucija preverja in izvaja transakcije. V primeru, da inštitucija propade valute postane neuporabna. Poleg tega pa mora uporabnik slepo verjeti, da bo inštitucija poštena. Pri decentralizaciji je drugače. V takem okolju nimamo inštitucije, ki bi igrala glavne vloge pri preverjanju transakcij. To nalogo zaupamo tretji osebi. Poleg tega pa noben nima pod nadzorom izdajo novega denarja, ki onemogoči socio-ekonomske nepoštenosti. Bitcoin naj bi zaradi svojih lastnosti nasledil zlato.

Cilj članka je seznaniti bralca s podatkovnimi strukturami, kriptografskimi algoritmi in protokoli v P2P omrežju, ki postavljajo temelje vsem kripto valutam.

Članek je razdeljena na štiri dele. V prvem delu je opis podatkovnih struktur, ki jih uporablja Bitcoin. Nato sledi opis kriptografskih algoritmov, delovanje P2P omrežja in naloge rudarjev. Na koncu je na primerih predstavljena lastna kripto valute FRIKOIN. Cilj članka je usmeriti bralca v svet kriptografskih valut.

II. PODATKOVNE STRUKTURE

V naslednjem razdelku sta opisi podatkovni strukturi, ki se uporabljata pri Bitcoinu. Glavna naloga obeh je preverjanje integritete transakcij.

A. *Hash kazalci*

Temelje Bitcoina sestavljajo bloki v katerem se nahajajo posamezne transakcije. Blok poleg transakcije vsebuje tudi zgoščeno vrednost prejšnjega bloka. Vrednost le te kaže na prejšnji blok v verigi. V primeru spremembe le enega bloka v verigi povzroči spremembo hash vrednosti v vseh nadaljnjih blokih.

B. *Merklee drevesa*

Merkleevo drevo ali binarno hash drevo je podatkovno struktura, ki zmanjša velikost posameznega bloka. Namesto, da bi v glavi bloka imeli vse transakcije se izračuna samo njihova hash vrednost. S tem se optimizira poraba prostora na pomnilniškem mediju in hitrost preverjanja posamezne transakcije. Listi v drevesu predstavljajo posamezne transakcije. Drevo zgradimo tako, da rekurzivno zgoščujemo vrednosti posameznega vozlišča. Pri tem se uporablja SHA256

Fig. 2. Grafični prikaz hash kazalca. Vsak blok vsebuje zgoščeno vrednost prejšnega bloka. [4]

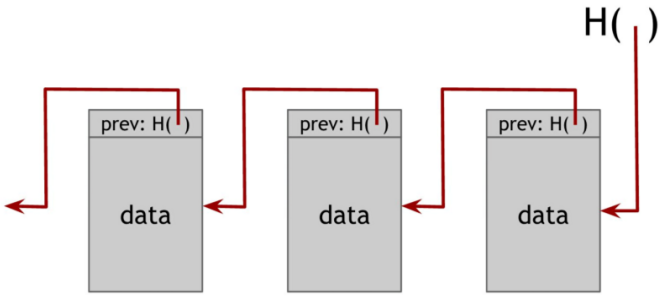
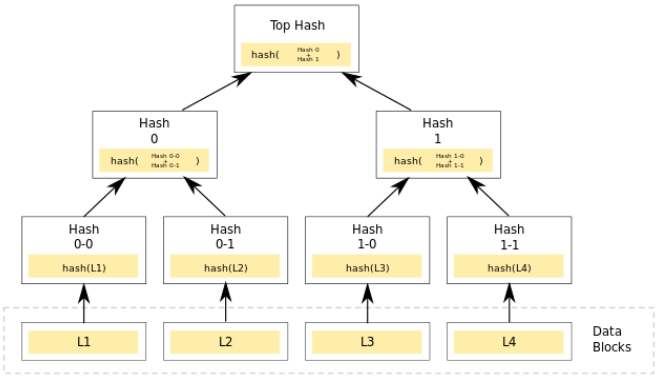


Fig. 3. Grafični prikaz Merklee-ovega drevesa



zgoščevalna funkcija. Ko je drevo zgrajeno, lahko z največ $2 * \log(N)$ operacijami, kjer je N število listov, preverimo ali se iskana transakcija nahaja v drevesu.

III. KRIPTOGRAFIJA V BITCOIN-U

V naslednjem razdelku so opisani kriptografski algoritmi, ki omogočajo preverjanje integritete, verodostojnosti transakcij in dokaz lastništva posameznega kovanca. Opisa je algoritem za podpisovanje, ki temelji na eliptičnih krivuljah in zgoščevalna funkcija SHA256. Na koncu je opisana tudi uporaba le te v algoritmu Hashcash.

A. ECDSA in Secp256k1

ECDSA je algoritem, ki temelji na eliptičnih krivuljah in je namenjen izključno le digitalnemu podpisovanju. Posamezne bloke se pošilja v čistopisu zaradi politike valute. Vse transakcije morajo biti javno dostopne in preverljive tistemu, ki želi to storiti. S tako politiko dosežejo zaupanje v valuto in posledično povečajo njeno vrednost. Digitalni podpisi so namenjeni preverjanju lastništva Bitcoinov in zahtevka za prenos sredstev (z drugimi besedami ali je pošiljatelj zahtevka pooblaščen za rokovanje z Bitcoini, ki jih želi porabiti).
Eliptična krivulja E je množica rešitev enačbe $y^2 = x^3 + Ax + B$. Pri tem je E definirana na končnem polju $\mathbb{F}_q$, kjer je $q = p^n, p \in \mathbb{P}$. Imamo dve točki $P, Q \in E(\mathbb{F}_q)$ na krivulji E. Potrebno je najdi število a, da bo enačba $Q = aP$ držala. To enačbo prevedemo v obliko $a = \log_P Q$ in dobimo problem diskretnega logaritma. Pri tem je število a za zasebni ključ, točka P je javni ključ in točka Q je baza. Varnost izhaja iz težavnosti reševanje diskretnega logaritma. Pri uporabi Secp256k1 je enačba eliptične krivulje oblike $y^2 = x^3 + 7$.

B. SHA256

SHA256 je zgoščevalna funkcija iz družine SHA-2. Za vhod dobi poljubno velik niz, ki ga zgosti v 256 bitni niz. Pri tem ima zgoščevalna funkcija naslednje lastnosti:

- Težko je najdi vrednost x in y tako, da sta $x \neq y$ in $H(x) = H(y)$ kjer je H zgoščevalna funkcija. Trki pri omejenem izhodu zgoščevalne funkcije (256 bitov) obstajajo. Ampak tako iskanje je zamudno in posledično nepraktično.
- Zgoščevalna funkcija mora dobro skriti prvotni vhod. Če imamo vrednost $H(x)$, ne moremo najdi x. Praktična uporaba te lastnosti je, da se nekdo predhodno odloči za neko vrednost, razglasi njeno zgoščeno vrednost in šele nato razglasi vrednost vhoda. Pri tem lahko vsak preveri ali je razglašena vrednost res tista za katero se je predhodno odločil.

C. Hashcash

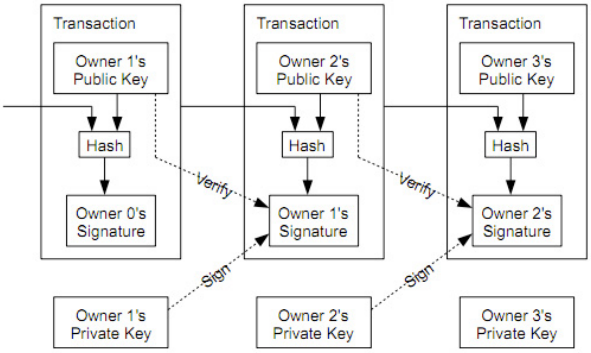
Leta 1997 je Adam Back izdelal algoritem z imenom Hashcash. Algoritem temelji na ideji, da se mora uporabnik z določeno količino dela pokazati, da res želi uporabljati določeno storitev. Za primer lahko uporabimo neželeno elektronsko pošto. Tisti, ki pošilja neželeno pošto jo pošilja v veliki količini. Kot protiukrep lahko končni prejemnik pošte zahteva od pošiljatelja naj vloži več truda v pošiljanje. V primeru, da pošiljatelj pošilja samo eni osebi časovni vložek ni velik. Če pa pošilja v veliki količini pošto, je časovni vložek zelo velik. Gre za zaviranje napadov na sistem. Algoritem delo ustvari tako, da od računalnika zahteva naj najde s SHA256 zgoščeno vrednost, ki ima na začetku določeno število ničel. Sporočilo (fiksni del) združimo z naključno vrednostjo in izračunamo zgoščeno vrednost. S povečanjem začetnih ničel v zgoščeni vrednosti povečamo tudi delo. V Bitcoinu se algoritem uporablja za izbiranje rudarjev v omrežju in kot protiukrep za napade na verigo blokov. Če napadalec želi spremeniti blok potem moral poiskati tako vrednost, da dobi točno določeno število ničel v zgoščeni vrednosti. Ker posamezni blok vsebuje tudi zgoščeno vrednost prejšnjega bloka mora podobno narediti tudi za vse sledeče bloke.

D. Potek plačila

V tem razdelku je opisano kako se kriptografske algoritme, ki so opisani zgoraj uporabi v Bitcoin sistemu. Predpostavljeno je, da ima uporabnik v lasti vsaj en kovanec. Bitcoin kovanec je povezan s svojim lastnikom preko njegovega javnega ključa. Tisti, ki si lasti njegov par, zasebni ključ, si lasti tudi kovanec. Recimo, da želi Ana poslati Bojanu kovanec. Vrednost kovanca za naš primer ni pomemben. Še predeno Ana pošlje kovanec moram Bojan ustvari z ECDSA algoritmom zasebni in javni ključ. Javni ključ služi kot naslov na katerega bo prejel Anin kovanec. Nato Ana

ustvari zahtevek za transakcijo kjer napiše, da je nov lastnik kovanca Bojan. Zahtevek podpiše s svojim zasebnim ključem in ga pošlje v Bitcoin omrežje. S tem se razglasi novi lastnik kovanca.

Fig. 4. Grafični prikaz uporabe kriptografije pri transakcijah [4]



IV. P2P OMREŽJE IN RUDARJENJE

Bitcoin omrežje je zasnovano na ideji decentralizacije. To dosežejo s tako imenovanim Peer-to-peer omrežjem. Omrežje nima fiksnih vozlišč in vsa vozlišča hranijo iste informacije. Torej, če kakšno vozlišče izgubi stik z ostalimi vozlišči omrežje še vedno nemoteno deluje. Vendar mora vpeljati protokol po katerem se bodo vozlišča ravnala. Protokol se imenuje protokol za soglasje. Vendar pri izdelavi protokola je potrebno upoštevati določene prepreke, kot so na primer: vozlišča lahko nenadoma izginejo ali pa so zlonamerne. Tukaj se Bitcoin omrežje razlikuje od ostalih P2P omrežjih v dveh stvareh:

- Uvaja spodbude (vsako pošteno vozlišče po za svoje delo dobi plačilo v obliki kovancev)
- Vozlišča so popolnoma samostojna in delujejo na principi kdo "prvi pride prvi melje".

A. Protokol za soglasje

Protokol za soglasje omogoča, da P2P omrežje kjer lahko vozlišče nenadoma izgine ali pa se izkaže za zlonamerno, deluje nemoteno. Protokol za soglasje teče tako:

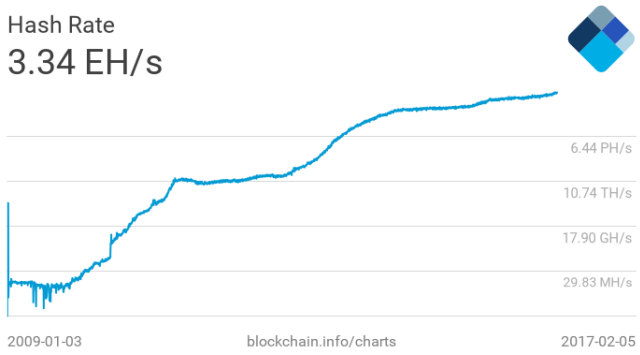
1) Nov zahtevek za transakcijo se objavi na omrežju.
2) Vsako vozlišče izbere poljubno mnogo zahtevkov.
3) Vsak krog vozlišča predlagajo novi blok v verigi.
4) Ostala vozlišča sprejmejo blok, če je transakcija veljavna (pravilen podpis, neporabljena sredstva).
5) Na koncu vozlišča vključijo sprejeti blok v naslednji blok.

Veriga z vsako sprejeto transakcijo narašča. Pri tem poštena vozlišča dodajajo nove bloke k največji veljavni verigi.

B. Rudarjenje

Vozlišča ali rudarji se potegujejo za možnost predlaganja novega bloka v verigi. Tekmovanje poteka tako, da za posamezni zahtevek za transakcijo opravijo delo, ki ga da Hashcash algoritem. Težavnost iskanja teh zgoščenih

Fig. 5. Grafični prikaz naraščanja hash moči Bitcoin omrežja [9]



vrednosti je odvisno od število ničel na začetku. Bitcoin omrežje je narejeno tako, da na vsaka dva tedna prilagodi zahtevnost glede na hash moč omrežja. Vsak rudar, ki najde pravilno zgoščeno vrednost dobi pravico, da zaprosi za nov kovanec in se odloči na kateri naslov ga bo prejel. Kovanec prejme samo takrat, ko je njegov blok dodan verigi. Omrežje samo na tak način izdeluje nove kovance.

Tekmovalnost med rudarji spodbuja le zaslužek za opravljeno storitev. Zaslužek prinese manj zlonamernih rudarje saj tisti, ki delajo škodo ne zaslužijo nič. Ko se je Bitcoin prvič pojavil, leta 2009, je vsak rudar za storitev zaslužil 50BTC. Zaslužek se na vsake 4 leta razpolovi. Trenutni zaslužek je 12.5BTC. Zadnji kovanec naj bi se proizvedel leta 2140. V ta namen so avtorji vključili možnost, da plačnik v zahtevek za transakcijo vključi tudi napitnino za rudarja ali pa rudar zahteva od plačnika plačilo za storitev.

C. Napadi na BitCoin

V sledečem razdelku sta izpostavljena dva napada, ki sta pomembna za razumevanje Bitcoin sistema.

1) Dvojno zapravljanje: Pri dvojnem zapravljanju nekdo poskuša zapraviti en kovanec večkrat. Sistem je zgrajen tako, da se kovanci ne nahajajo na enem koncu omrežja. Kovanci živijo v verigi. Vsak v omrežju ve komu pripada posamezen kovanec in kdaj se je kovanec pojavil v omrežju. Če rudar najde ponovitev kovanca v eni izmed verig, zahtevek za transakcijo zavrne.

2) 51% napadalec: Gre za napad na omrežje kjer ima napadalec več kot 50% hash moči omrežja. Napadalec želi izvesti napad dvojnega zapravljanja. Trgovcu pošlje kovanec. Še preden mu trgovec pošlje kupljeno dobrino trgovec počaka, da se na verigo takoj za njegovim blokom dodajo poljubno mnogo drugi blokov. Pred vsakim novim blokom se preverijo vsi prejšnji. Torej več kot je blokov večja je verjetnost, da gre za pošteno transakcijo. Vendar, če napadalcu uspe najdi dovolj blokov na katere čaka trgovec in jih dodati na verigo lahko izvede napada dvojnega zapravljanja. Verjetnost za uspeh napada je odvisna koliko moči ima napadalec. Če si lasti 10% moči in trgovec čaka na 6 novih blokom, potem ima napadalec 0.1% možnosti za uspeh. V primeru 100% verjetnosti za uspeh mora imeti več kot 50% moči celotnega omrežja.

V. PRIMER LASTNEGA DIGITALNEGA DENARJA: FRIKOIN

V naslednjem razdelku je opisana lastna kripto valuta z imenom FRIKOIN. Valuta je napisana v Javi in uporablja kritografsko knjižnico Bouncycastle 1.56. Valuta služi kot primer uporabe kriptografskih rešitev, ki jih uporabljajo ostale kripto valute. Na primeru je tudi pokazan napad dvojnega zapravljanja. Koda se nahaja na naslovu: https://gitlab.lem.im/tilen.kavcic/frikoin

A. Rudar

Naloge rudarja so:

- Ko dobi zahtevek za transakcijo naprej preveri ali je kovanec veljaven.
- Nato preveri ali je pošiljatelj zahtevka upravičen do pripetega kovanca.
- Sledi izračun zgoščene vrednosti.
- Novo ustvarjen blok doda verigi.

B. Uporabnik

Naloge uporabnika so:

- Vsakič, ko prejme kovanec mora pridobiti najnovejšo verigo in pogledati, koliko kovancev mu pripada.
- Ko želi prejeti nov kovanec, mora ustvariti z ECDSA nov javni in zasebni ključ.
- Ko želi poslati kovanec, mora ustvariti zahtevek za transakcijo in ga podpisati z zasebnim ključem. Podpis se bo preveril z javnim ključem, ki je del kovanca.

C. Primer transakcije

V tem primeru je prikazan zahtevek za transakcijo med dvema uporabnikoma. Pri tem kovanec nima vrednost. Postopek primera:

1) Najprej se ustvari začetnik blok in z njim veljavni kovanec.
2) Bojanu ustvari javni in zasebni ključ in se pripravi na sprejem kovanca.
3) Kovanec dobi novega lastnika tako, da pripnemo njegov javni ključ. Ker ima Bojan zasebni ključ lahko pošlje kovanec drugemu uporabniku.
4) Bojan nato pošlje zahtevek za transakcijo novo pridobljenega kovanca Ani.
5) Ana v ta namen ustvari javni in zasebni ključ.
6) Rudar nato sprejme Bojanov zahtevek za transakcijo.
7) Najprej preveri ali je pripeti kovanec legitimno ustvarjen in ali je še vedno veljaven v verigi. V primeru, da je Bojan dobil legitimen kovanec in ga ni zapravil lahko uporabi kovanec za plačilo.
8) Nato preveri ali je Bojan res lastnik tega kovanca. To se preveri tako, da se pogleda podpis na zahtevku. Če je podpis ustreza javnemu ključu, ki je pripet kovancu potem se transakcija lahko izvede.
9) Izračuna se zgoščena vrednost, ki ima 3 zaporedne ničle.
10) Blok se doda verigi.
11) Nato Ana pošlje Bojanu z istim postopkom nazaj kovanec.

```
Coin is valid!
Signature is valid!
Last valid block: 0
Last valid hash: 0003712
    d5e17cc7b9b16926fcd29a5f17c5d7
da57b5b559e79db0cf6b6a956e1
Found nonce: f4c3797e
Hash value: 000
    f7fec98df24acd5c7eced89c208ba03efad3aa3
422b7d48e96dbd8929296c
Block added!
Coin transferred successfully!
================================================

Coin is valid!
Signature is valid!
Last valid block: 1
Last valid hash: 000
    f7fec98df24acd5c7eced89c208ba03efa
d3aa3422b7d48e96dbd8929296c
Found nonce: 5fb7f493
Hash value: 000
    f647aa59793c701a97380618c2689f7123a1c1e
5e034e0fac521c04e3ac47
Block added!
Coin transferred successfully!
```

D. Primer napada z ponarejenim kovancem

V tem primeru je prikazan zahtevek za transakcijo med dvema uporabnikoma s ponarejenim kovancem. Pri tem kovanec nima vrednost. Postopek primera:

1) Bojan naprej sprazni svojo denarnico tako, da pošlje svoj edini kovanec Ani.
2) Nato Bojan ustvari nov kovanec in nov javni in zasebni ključ. Kovanec pripiše samemu sebi.
3) Kovanec nato pošlje Ani
4) Podpis je veljaven, vendar v verigi je razvidno, da kovanec ni bil legitimno ustvarjen.
5) Transakcija je zavrnjena.

```
FAKE COIN ATTACK
EMPTYING WALLET
Coin is valid!
Signature is valid!
Last valid block: 2
Last valid hash:
000f647aa59793c701a97380618c2689f7123a1c1
e5e034e0fac521c04e3ac47
Found nonce: f66abf15   Hash value:
00020f0ff98325600b94f3dea96b104797c4695655
a120902d694f31ecf2a83f
Block added!
Coin transferred successfully!
================================================
CREATING FAKE COIN
Coin is not valid!
```

E. Primer dvojnega zapravljena

V tem primeru je prikazan zahtevek za transakcijo med dvema uporabnikoma pri katerem prvi uporabnik želi dvakrat zapraviti isti kovanec. Pri tem kovanec nima vrednost. Postopek primera:

1) Ana pošlje Bojanu nazaj kovanec (kovanec iz prejšnjega primer).
2) Bojan naredi kopijo dobljenega kovanca in pošlje en kovanec Ani.
3) Nato Bojan ponovno pošlje kopijo kovanca Ani.

4) Ker je v verigi zabeleženo, da je bil ta kovanec že porabljen, je transakcija zavrnjena.

```
DOUBLE SPEND ATTACK
Coin is valid!
Signature is valid!
Last valid block: 3
Last valid hash:
 00020f0ff98325600b94f3dea96b104797c4695655
 a120902d694f31ecf2a83f
Found nonce: c1ddadea   Hash value:
 000675b9b61263c9751072278788cf93e7188f6eae
 964783965f84c95e833866
Block added!
Coin transferred successfully!
================================================
Coin is valid!
Signature is valid!
Last valid block: 4
Last valid hash:
000675b9b61263c9751072278788cf93e7188f6eae
964783965f84c95e833866
Found nonce: a9b5ad14   Hash value:
0005ba247cf85ae8e219a8046415f6b9678ad7d29b
1d540ab9f227ccfa453a9e
Block added!
Coin transferred successfully!
================================================
Coin is not valid!
```

VI. Zaključek

V članku je predstavljena kripto valute Bitcoin na katerih temeljijo vse obstoječe kripto valute. Predstavljena je pomembnost decentralizacije valut in rešitev problema dvojnega zapravljanja. Bralec je seznanjen s podatkovnimi strukturami (binarna hash drevesa in hash kazalci), kriptografskimi algoritmi (ECDSA in SHA256) in protokol P2P omrežja, ki jih uporablja Bitcoin. Na koncu je predstavljen lastna kripto valute FRIKOIN. Na primerih je bralec seznanjen z praktično uporabo kriptografskih algoritmov.

Reference

[1] Narayanan, Arvind, et al. Bitcoin and cryptocurrency technologies. Princeton University Pres, 2016.

[2] Maurer, Bill, Taylor C. Nelms, and Lana Swartz. ""When perhaps the real problem is money itself!": the practical materiality of Bitcoin.Šocial Semiotics 23.2 (2013): 261-277.

[3] Antonopoulos, Andreas M. Mastering Bitcoin: unlocking digital cryptocurrencies. "O'Reilly Media, Inc.", 2014.

[4] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system."(2008).

[5] Barski, Conrad, and Chris Wilmer. Bitcoin for the Befuddled. No Starch Press, 2014.

[6] Secg.org. (2017). [online] Available at: http://www.secg.org/sec2-v2.pdf [Accessed 6 Feb. 2017].

[7] En.bitcoin.it. (2017). Secp256k1 - Bitcoin Wiki. [online] Available at: https://en.bitcoin.it/wiki/Secp256k1 [Accessed 6 Feb. 2017].

[8] What does the curve used in Bitcoin, l. (2017). What does the curve used in Bitcoin, secp256k1, look like?. [online] Bitcoin.stackexchange.com. Available at: https://bitcoin.stackexchange.com/questions/21907/what-does-the-curve-used-in-bitcoin-secp256k1-look-like [Accessed 6 Feb. 2017].

[9] Blockchain.info. (2017). Bitcoin Block Explorer - Blockchain. [online] Available at: https://blockchain.info/ [Accessed 7 Feb. 2017].