Bitcoin: Elektronski gotovinski sistem P2P (P2P - mreže računara istog prioriteta)

Satoši Nakamoto
satoshin@gmx.com
www.bitcoin.org

Rezime: Potpuna verzija P2P elektronskog novca omogućila bi da se plaćanja putem interneta vrše neposredno od jedne strane prema drugoj, odnosno, u takvom aranžmanu elektronska gotovina ne prolazi kroz finansijsku instituciju. Jedan deo rešenja predstavljaju digitalni potpisi, međutim, gube se glavne prednosti ako je i dalje potrebna treća strana od poverenja za sprečavanje duple potrošnje. U ovom radu predlaže se rješenje problema duple potrošnje pomoću mreže P2P. Mreža stavlja vremenske oznake na transakcije tako što ih hešira u tekući lanac dokaza o radu koji se zasniva na hešovima, formirajući zapis koji nije moguće promeniti bez ponavljanja dokaza o radu. Najduži lanac ne samo da služi kao dokaz da je došlo do niza događaja u čije postojanje smo se lično uverili, već i kao dokaz da dolazi iz najveće zbirne procesorske snage. Sve dok računari koji ne sarađuju međusobno u napadu na mrežu kontrolišu većinu procesorske snage, oni će formirati najduži lanac i nadjačati napadače. Minimalna strukturalna rešenja su potrebna za samu mrežu. Poruke se prenose uz pretpostavku da je svaki računar uložio najveće napore da ih prenese, dok računari mogu da napuste i ponovo se pridruže mreži po želji, prihvatajući najduži lanac nastao u toku provere radom kao dokaz onoga što se dogodilo dok su bili odsutni.

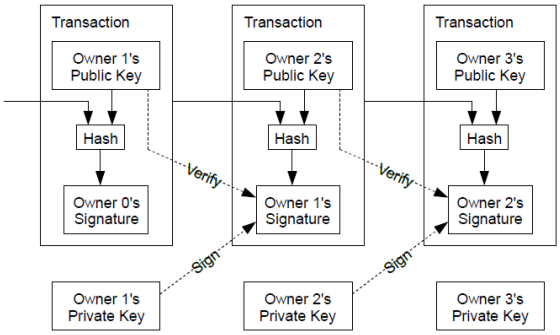
1. Uvod

Trgovina preko Interneta je došla do faze da se gotovo isključivo oslanja na finansijske institucije koje služe kao treća strana od poverenja za obradu elektronskih plaćanja. Iako taj sistem funkcioniše dovoljno dobro za većinu transakcija, još uvek se bori sa svojstvenim slabostima modela zasnovanog na poverenju. Potpuno nepovratne transakcije nisu moguće, jer finansijske institucije ne mogu da izbegnu ulogu posrednika u eventualnim sporovima. Troškovi posredovanja povećavaju troškove u vezi sa transakcijama, ograničavajući najmanju moguću veličinu transakcije i odbacujući mogućnost da dođe do malih, slučajnih transakcija jer su troškovi po osnovu štete za nepovratno plaćanje za nepovratne usluge značajno veliki. Uz mogućnost povraćaja, jača potreba za poverenjem. Trgovci moraju da budu oprezni sa kupcima, tražeći od njih više podataka nego što bi im inače bilo potrebno. Neki procenat prevarnih radnji se prihvata kao neminovnost. Ti troškovi i neizvesnost koja je povezana sa plaćanjima mogu se izbeći korišćenjem fizičke valute, ali ne postoji mehanizam za plaćanje preko kanala za komunikaciju bez potrebe za trećom stranom od poverenja.

Ono što je potrebno jeste sistem za elektronsko plaćanje koji se zasniva na kriptografskom dokazu umesto na poverenju, omogućavajući dvema voljnim stranama da neposredno vrše transakcije između sebe, bez potrebe za trećom stranom od poverenja. Transakcije koje je računski teško poništiti zaštitile bi prodavce od prevarnih radnji, dok se dobro poznati eskrou mehanizmi mogu lako primeniti da bi zaštitili kupce. U ovom radu se predlaže rešenje problema duple potrošnje korišćenjem servera za generisanje vremenskih oznaka mreže računara istog prioriteta (P2P), kako bi se imao računski dokaz hronološkog rasporeda vršenja transakcija. Sistem je siguran sve dok pošteni računari zajedno kontrolišu više procesorske snage nego neka druga udružena grupe napadačkih računara.

2. Transakcije

Elektronski novčić definišemo kao lanac digitalnih potpisa. Vlasnici prenose novčiće između sebe tako što digitalno potpišu heš prethodne transakcije i javni ključ sljedećeg vlasnika, dodajući ih zatim na kraj novčića. Primalac plaćanja može da proveri potpise kako bi potvrdio lanac vlasništva.

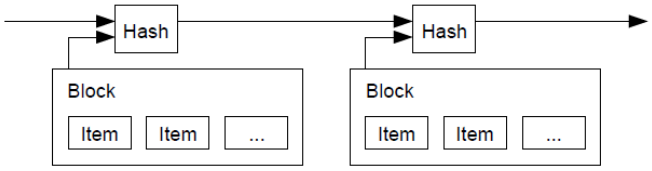


Međutim, problem je u tome što primalac plaćanja ne može da proveri da li je neki od vlasnika dva puta potrošio isti novčić. Uobičajeno rešenje podrazumeva uvođenje pouzdanog centralnog tela, ili kreatora novčića, koji proverava svaku transakciju kako bi utvrdio postojanje eventualne duple potrošnje. Nakon svake transakcije, novčić se mora vratiti kreatoru novčića kako bi se izdao novi novčić, a smatra se da samo oni novčići koje kreatori izdaju niesu ranije upotrebljeni za neko plaćanje. Problem sa ovim rešenjem je u tome što sudbina celokupnog novčanog sistema zavisi od kompanije koja kreira novčiće, jer svaka transakcija mora da prođe preko njih, baš kao što je slučaj kod banke.

Potreban nam je način da primalac plaćanja bude siguran da prethodni vlasnici nisu potpisali neku raniju transakciju. Za naše potrebe, uzimamo u obzir transakciju koja se prva dogodila, tako da nas ne zanimaju naredni pokušaji duple potrošnje. Jedini način da potvrdimo da do transakcije nije došlo jeste da imamo informacije o svim transakcijama koje su se desile. U modelu koji se zasniva na kreatoru novčića, kreator ima informacije o svim transakcijama i odlučuje koja transakcija je prva stigla. Da bismo to postigli bez poverljivog posrednika, transakcije se moraju javno objaviti [1], a potreban nam je i sistem u kojem se učesnici mogu dogovoriti o jedinstvenoj istoriji redosleda po kojem su transakcije primljene. Primaocu plaćanja je potreban dokaz da se u trenutku transakcije većina računara složila oko toga da je ta transakcija prva primljena.

3. Server vremenskih oznaka

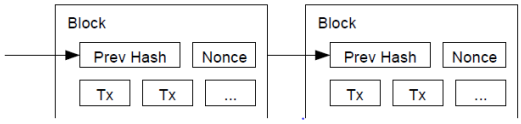
Rešenje koje predlažemo počinje sa serverom vremenskih oznaka. Server vremenskih oznaka funkcioniše tako što uzima heš bloka podataka kojima će se dodeliti vremenska oznaka i objaviti taj heš na mreži, slično kao što se to radi u novinama ili kao post na Usenet-u [2-5]. Vremenska oznaka dokazuje da su podaci morali postojati u to vreme da bi ušli u heš. Svaka vremenska oznaka sadrži prethodnu vremensku oznaku u svom hešu, formirajući tako lanac, pri čemu svaka dodatna vremenska oznaka potvrđuje one pre nje.



4. Dokaz o radu

Da bismo implementirali distribuirani server vremenskih oznaka na ravnopravnoj osnovi (P2P), moramo da koristimo sistem dokazivanja rada sličan Heškešu Adama Beka [6], umesto objava u novinama ili na mreži Usenet. Dokaz o radu uključuje traženje vrednosti koja, kada se hešuje, npr. kao kod SHA-256, stvara heš koji počinje sa određenim brojem nula. Prosečna količina potrebnog rada eksponencijalno raste sa brojem potrebnih nula i može se proveriti izvršavanjem jednog heširanja.

Za našu mrežu vremenskih oznaka implementiramo dokaz radom tako što povećavamo nonce (jednokratnu vrednost) u bloku dok se ne pronađe vrednost koja hešu bloka daje potreban broj nula. Kada se utroši procesorska snaga kako bi se zadovoljio dokaz o radu, blok se ne može promeniti bez ponavljanja rada. Kako se kasniji blokovi vežu na taj blok, rad na promeni bloka podrazumevao bi ponovno obrađivanje svih blokova nakon njega.



Dokaz o radu rešava i problem utvrđivanja većine pri odlučivanju. Ako bi se većina zasnivala na principu jedna IP adresa-jedan glas, mogao bi je poništiti svako ko može da dodeli mnogo IP adresa. Dokaz o radu predstavlja u suštini jedan glas po procesorskoj jedinici. Većinsku odluku predstavlja najduži lanac, u čije stvaranje je uloženo najviše rada tokom dokazivanja. Ako većinu procesorske snage kontrolišu pošteni računari, pošteni lanac najbrže raste i nadjačava sve konkurentske lance. Da bi izmenio prethodni blok, napadač bi morao da ponovi dokaz o radu bloka i svih blokova nakon njega, a zatim da sustigne i nadmaši količinu rada poštenih računara. Kasnije ćemo pokazati da se verovatnoća da će sporiji napadač uspeti da sustigne eksponencijalno smanjuje dodavanjem narednih blokova.

Da bi se kompenzovalo povećanje brzine hardvera i različito interesovanje ljudi za upravljanjem računarima tokom vremena, težina postizanja dokaza o radu određuje se prema prosečnom broju blokova kreiranih u toku jednog sata. Ako se blokovi prebrzo generišu, povećava se težina.

5. Mreža

Koraci za vođenje mreže su kao što sledi:

1) Nove transakcije se emituju ka svim računarima.
2) Svaki računar prikuplja nove transakcije u blok.
3) Svaki računar radi na pronalaženju rešenja putem dokaza o radu za svoj blok.
4) Kada računar pronađe rešenje putem dokaza o radu, emituje taj blok svim računarima.
5) Računari prihvataju blok samo ako su sve transakcije koje se nalaze u njemu važeće i niesu već potrošene.
6) Računari izražavaju svoje prihvatanje bloka radeći na stvaranju sledećeg bloka u lancu, koristeći heš prihvaćenog bloka kao prethodni heš.

Računari uvek smatraju da je najduži lanac ispravan i nastaviće da ga produžavaju. Ako dva računara emituju različite verzije sledećeg bloka istovremeno, neki računari mogu da prime prvo jedan ili drugi blok. U tom slučaju svaki računar radi na prvom bloku koji je primio, ali čuvaju drugu kariku lanca u slučaju da ona postane duža. Svaka dilema prestaje kada se pronađe sledeći dokaz o radu i jedna karika postane duža; računari koji su radili na drugoj karici lanca se zatim prebacaju na onu koja je duža.

Nova emitovanja transakcija ne moraju neophodno da stignu do svih računara. Sve dok stižu do većeg broja računara, te transakcije će brzo ući u blok. Emitovanja blokova tolerišu i ispuštene poruke. Ako računar ne primi blok, zatražiće ga kada primi sledeći blok i shvatiti da je propustio prethodni blok.

6. Podsticaj

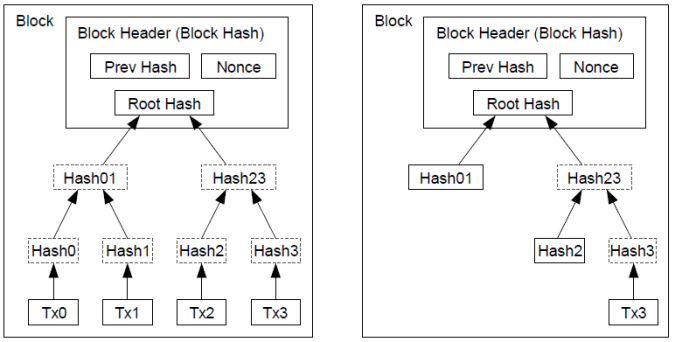
Po pravilu, prva transakcija u bloku obično je posebna transakcija koja kreira novi novčić koji je u vlasništvu kreatora bloka. Time se daje podsticaj računarima da podrže mrežu i omogućava da se pokrene inicijalno ubacanje novčića u opticaj, budući da ne postoji centralno telo koje ih izdaje. Stalno dodavanje konstantne količine novih novčića može se uporediti sa rudarima zlata koji ulažu resurse kako bi izvadili nove količine zlata i dodali ih u opticaj. U našem slučaju troše se procesorsko vreme i električna energija.

Naknade za transakcije se mogu koristiti za finansiranje podsticaja. Ako je izlazna vrednost transakcije manja od njene ulazne vreednosti, razliku čini naknada za transakciju koja se dodaje podsticajnoj vrednosti bloka koji sadrži transakciju. Kada unapred određeni broj novčića uđe u opticaj, podsticaj mogu u potpunosti da sačinjavaju naknade za transakcije, čime se isključuje inflacija.

Podsticaji podstiču računare da ostanu pošteni. U slučaju da je napadač pohlepan i može da angažuje više procesorske snage od svih poštenih računara, morao bi da bira između toga da tu snagu koristi za prevaru tako što krade plaćanja ili da je koristi za kreiranje novih novčića. Napadač bi trebalo da shvati da se više isplati igrati po pravilima, odnosno pravilima koja ga nagrađuju tako da dobija više novčića od svih ostalih zajedno, nego da potkopava sistem i vrednost sopstvenog bogatstva.

7. Oslobađanje prostora na disku

Kada se najnovija transakcija u novčiću nalazi pod dovoljno blokova, mogu se odbaciti prethodne transakcije kako bi se oslobodio prostor na disku. Da bi se to sve sprovelo bez razbijanja heš bloka, transakcije se hešuju u Marklovo stablo [7][2][5], pri čemu je samo koren uključen u heš bloka. Stari blokovi se mogu zatim sabiti uklanjanjem nepotrebnih lanaca. Unutrašnje hešove nije potrebno skladištiti.



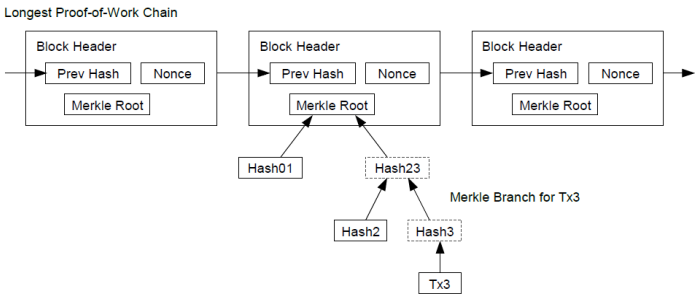
Transactions Hashed in a Merkle Tree

After Pruning Tx0-2 from the Block

Zaglavlje bloka bez transakcija iznosilo bi oko 80 bajtova. Ako pretpostavimo da se blokovi generišu na svakih 10 minuta, 80 bajtova * 6 * 24 * 365 = 4.2 MB godišnje. Uzimajući u obzir činjenicu da se u 2008. godini prodaju računarski sistemi sa obično 2 GB RAM-a i Murov zakon koji predviđa trenutni rast od 1,2 GB na godišnjem nivou, skladištenje ne bi trebalo da predstavlja problem čak i ako se zaglavlja bloka moraju čuvati u memoriji.

8. Pojednostavljena verifikacija plaćanja

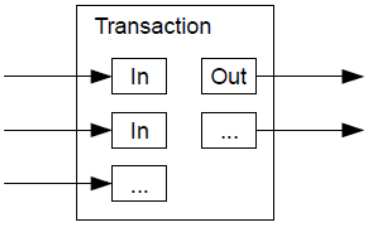
Moguće je verifikovati plaćanja bez korišćenja mrežnog računara. Korisnik samo treba da sačuva kopiju zaglavlja blokova najdužeg lanca dokaza o radu, koje može dobiti upitom ka mrežnim računarima dok se ne uveri da ima najduži lanac i dobije Marklovu granu koja povezuje transakciju sa blokom na koji je utisnuta vremenska oznaka. On ne može samostalno da proveri transakciju, ali povezivanjem sa mestom u lancu može da vidi da je mrežni računar prihvatio, i da se blokovi dodaju nakon što dalje potvrđuje da ju je mreža prihvatila.



Kao takva, verifikacija je pouzdana sve dok pošteni računari kontrolišu mrežu, ali je ranjiva ako napadači nadjačaju mrežu. Mrežni računari mogu sami da verifikuju transakcije, ali fabrikovane transakcije napadača mogu da prevare pojednostavljeni metod dok god je napadač u stanju da nadjača mrežu. Jedna od strategija zaštite bila bi da se prihvati upozorenje od mrežnih računara kada otkriju nevažeći blok, čime se podstiče softver korisnika da preuzme celi blok i sporne transakcije da bi se potvrdila nepravilnost. Privredna društva koja primaju česte uplate će verovatno želeti da pokrenu i vode sopstvene računare radi potrebe da im sigurnost ne zavisi od drugih i zbog brže verifikacije.

9. Kombinovanje i deljenje vrednosti

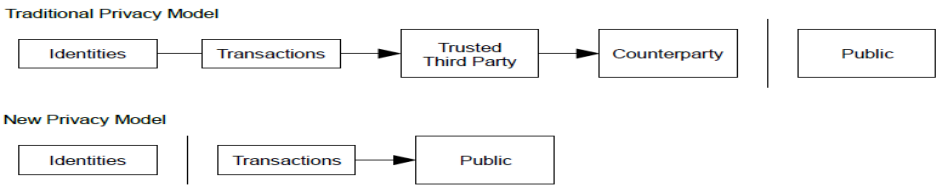
Iako bi bilo moguće rukovati novčićima pojedinačno, nije zgodno kreirati pojedinačne transakcije za svaki cent u tom transferu. Da bi se omogućilo da se vrednost podeli i kombinuje, transakcije sadrže višestruke ulaze i izlaze. Obično postoji ili jedan ulaz iz veće prethodne transakcije ili više ulaza koji kombinuju manje iznose, i najviše dva izlaza: jedan za plaćanje i jedan koji vraća kusur pošiljaocu, ako kusura ima.



Treba napomenuti da situacija gde transakcija zavisi od nekoliko transakcija, a te transakcije zavise od mnogo više drugih transakcija, ovde ne predstavlja problem. Nikada nema potrebe za izdvajanjem kompletne istorije transakcija.

10. Privatnost

Tradicionalni bankarski model postiže određeni nivo privatnosti tako što ograničava pristup informacijama stranama uključenim u transakcije i trećoj strani od poverenja. Potreba da sve transakcije budu javno objavljene isključuje mogućnost primene tog modela, ali privatnost se i dalje može sačuvati prekidanjem protoka informacija na drugom mestu: čuvanjem javnih ključeva anonimnim. Javnost može da vidi da neko šalje iznos nekom drugom, ali ne i informacije koje bi mogle povezati takvu transakciju sa nekim licem. Ovo je slično količini informacija koje objavljuju berze, gde se javno objavljuje vreme i veličina pojedinačnih trgovina, ali se ne navodi ko su strane u transakciji.



Kao dodatni zaštitni zid, preporučuje se korišćenje novog para ključeva za svaku transakciju, kako bi se sprečilo da se one dovedu u vezu sa zajedničkim vlasnikom. Neka povezivanja se ne mogu izbeći kada je reč o transakcijama sa više ulaza, koje otkrivaju da su njihovi ulazi bili u vlasništvu istog vlasnika. Rizik je da se u slučaju otkrivanja identiteta vlasnika ključa može povezati koje su druge transakcije pripadale tom istom vlasniku.

11. Obračuni

Razmatramo scenario u kojem napadač pokušava da generiše alternativni lanac brže od poštenog lanca. Čak i da se to postigne, time se onemogućavaju proizvoljne promene, kao što je stvaranje vrednosti ni iz čega ili uzimanje novca koji nikada nije pripadao napadaču. Računari neće prihvatiti nevažeću transakciju kao plaćanje, a pošteni računari nikada neće prihvatiti blok koji sadrži takve transakcije. Napadač može jedino da pokuša da promenu neku od svojih transakcija kako bi vratio novac koji je nedavno potrošio.

Trka između poštenog lanca i lanca napadača može se okarakterisati kao binomna distribucija slučajne varijable (*Binomial Random Walk*). Uspešni ishod je da se pošteni lanac produži za jedan blok, povećavajući svoje vođstvo za +1, a neuspešni ishod je da se lanac napadača produži za jedan blok, smanjujući zaostatak na -1.

Verovatnoća da će napadač nadoknaditi neki nedostatak analogna je problemu propasti kockara (“Gamblers Ruin”). Pretpostavimo da kockar sa neograničenim iznosom novca počinje sa zaostatkom i igra potencijalno beskonačan broj pokušaja kako bi nadoknadio zaostatak. Možemo izračunati verovatnoću da stigne do nule ili do poštenog lanca, na sledeći način [8]:

str = verovatnoća da će pošteni računar pronaći sledeći blok
q = verovatnoća da će napadač pronaći sledeći blok
q_z = verovatnoća da će napadač ikada sustići z blokova zaostatka

$$q_z = \begin{cases} 1 & if\ p \le q \\ (q/p)^z & if\ p > q \end{cases}$$

S obzirom na našu pretpostavku da je $p>q$, verovatnoća eksponencijalno opada kako raste broj blokova koje napadač mora da nadoknadi. Sa šansama protiv njega, ako mu se na samom početku ne posreći, šanse mu se smanjuju sa povećanjem zaostatka.

Sada ćemo razmotriti koliko dugo primalac nove transakcije treba da čeka pre nego što bude dovoljno siguran da pošiljalac ne može da promeni transakciju. Pretpostavljamo da je pošiljalac napadač koji želi da natera primaoca da veruje neko vreme da mu je platio, a zatim da tu transakciju preusmeri sebi. Primalac će dobiti upozorenje kada se to dogodi, ali pošiljalac se nada da će tada već biti prekasno.

Primalac generiše novi par ključeva i daje javni ključ pošiljaocu neposredno pre potpisivanja. Ovim se pošiljaoc sprečava da unapred pripremi lanac blokova radeći na njemu neprekidno sve dok uz pomoć sreće ne odmakne dovoljno napred, a zatim da izvrši transakciju u tom trenutku. Kada se transakcija pošalje, nepošteni pošiljalac počinje u tajnosti da radi na paralelnom lancu koji sadrži alternativnu verziju njegove transakcije.

Primalac čeka dok se transakcija ne doda u blok i dok se z blokova povežu nakon toga. On ne zna koliko je napadač napredovao, ali pod pretpostavkom da su pošteni blokovi kreirani očekivanom dinamikom, potencijalni napredak napadača će biti prikazan Poasonovom distribucijom sa očekivanom vrednošću:

$$\lambda = z\frac{q}{p}$$

Da bismo izračunali verovatnoću da bi napadač mogao da nadoknadi zaostatak, množimo Poasonovu gustinu za svaki nivo napretka koji je mogao da postigne sa verovatnoćom da od tog trenutka može da potpuno nadoknadi zaostatak:

$$\sum_{k=0}^{\infty}\frac{\lambda^k e^{-\lambda}}{k!}\cdot\begin{cases}(q/p)^{(z-k)} & \text{if } k\le z\\ 1 & \text{if } k>z\end{cases}$$

Preuređujemo formulu da bismo izbegli sabiranje beskonačnog broja sabiraka zbog repa distribucije...

$$1-\sum_{k=0}^{z}\frac{\lambda^k e^{-\lambda}}{k!}\left(1-(q/p)^{(z-k)}\right)$$

Pretvaranje u C kod...

```
#include <math.h>
double AttackerSuccessProbability(double q, int z)
{
        double p = 1.0 - q;
        double lambda = z * (q / p);
        double sum = 1.0;
        int i, k;
        for (k = 0; k <= z; k++)
        {
                double poisson = exp(-lambda);
                for (i = 1; i <= k; i++)
                poisson *= lambda / i;
                sum -= poisson * (1 - pow(q / p, z - k));
                }
                return sum;
                }
```

Dovoljno je nekoliko primera da vidimo da verovatnoća opada eksponencijalno sa z.

```
q=0,1
z=0P=1,0000000
z=1P=0,2045873
z=2P=0,0509779
z=3P=0,0131722
z=4P=0,0034552
z=5P=0,0009137
z=6P=0,0002428
z=7P=0,0000647
z=8P=0,0000173
z=9P=0,0000046
z=10 P=0,0000012

q=0,3
z=0P=1,0000000
z=5P=0,1773523
z=10 P=0,0416605
z=15 P=0,0101008
z=20 P=0,0024804
z=25 P=0,0006132
z=30 P=0,0001522
z=35 P=0,0000379
z=40 P=0,0000095
z=45 P=0,0000024
z=50 P=0,0000006
```

Rešavanje za P manje od 0,1%...

```
P < 0,001
q=0,10z=5
q=0,15z=8
q=0,20z=11
q=0,25z=15
q=0,30z=24
q=0,35z=41
q=0,40z=89
q=0,45z=340
```

12. Zaključak

U ovom radu je predložen sistem elektronskih transakcija bez oslanjanja na poverenje. Počeli smo sa uobičajenim šablonom novčića kreiranih iz digitalnih potpisa, koji pruža kontrolu nad vlasništvom, ali je nepotpun ako ne postoji način da se spreči dupla potrošnja. Da bismo ovo rešili, predložili smo mrežu P2P (peer to peer) koja koristi dokaz o radu za čuvanje javne istorije transakcija usled čega brzo postaje računski nepraktično za napadača da krene sa izmenama ako pošteni računari kontrolišu većinu procesorske snage. Mreža je robusna u svojoj nestrukturiranoj jednostavnosti. Računari rade istovremeno uz malo koordinacije. Nije potrebno identifikovati ih, budući da se poruke ne usmeravaju na neko određeno mesto, već ih samo treba isporučiti uz najveći trud. Računari mogu da napuste i ponovo se pridruže mreži po želji, prihvatajući lanac dokaza o radu kao dokaz onoga što se dogodilo dok ih nije bilo. Oni glasaju svojom procesorskom snagom, izražavajući prihvatanje valjanih blokova tako što rade na njihovom proširenju i odbacujući nevažeće blokove odbijanjem da rade na njima. Sva potrebna pravila i podsticaji mogu se nametnuti ovim mehanizmom postizanja konsenzusa.

Izvori:

[1] W. Dai, "b-money", http://www.weidai.com/bmoney.txt, 1998.

[2] H. Mesijas, XS Avila i J.-J. Kvikvoter, "Dizajn sigurne usluge vremenskih oznaka sa minimalnim zahtjevima povjerenja", iz *20. simpozijum o teoriji informacija u zemljama Beneluksa*, maj 1999.

[3] S. Haber, VS Storneta, "Kako vremenski označiti digitalni dokument," u *Časopis za kriptologiju*, tom 3, br. 2, stranice 99-111, 1991.

[4] D. Bajer, S. Haber, VS Storneta, "Poboljšanje efikasnosti i pouzdanosti digitalnih vremenskih oznaka," U *Sekvence II: Metode u komunikaciji, sigurnosti i računarstvu*, strane 329-334, 1993.

[5] S. Haber, VS Storneta, "Sigurna imena za bit-stringove", iz *Zbornik radova 4. ACM konferencije o sigurnosti računara i komunikacija*, strane 28-35, april 1997.

[6] A. Nazad, "Heškeš - protivmeera uskraćivanja usluge", http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] RC Markl, "Protokoli za kriptosisteme javnog ključa", u *Proc. 1980 Simpozijum o bezbednosti i Privatnost*, IEEE Computer Society, stranice 122-133, april 1980.

[8] V. Feler, "Uvod u teoriju vjerovatnoće i njene primeene", 1957.