Bitcoin-ul – Un sistem monetar electronic peer-to-peer

De Satoshi Nakamoto

satoshin@gmx.com

www.bitcoin.org

Abstract. O versiune în totalitate peer-to-peer de bani electronici ar permite ca plățile online să fie transmise direct de la o parte către alta fără să mai treacă prin instituțiile financiare. Semnăturile digitale oferă o parte din soluție, dar principalele avantaje sunt pierdute dacă încă mai este nevoie de un terț de încredere pentru a preveni cheltuirea aceleiași monezi de două ori. Propunem o soluție pentru problema acestei probleme a monedei cheltuite de două ori folosind o rețea peer-to-peer. Rețeaua pune un timbru de timp tranzacției făcând hashing[metodă de criptare] într-un lanț în desfășurare de dovezi-de-lucru bazate pe hash, și formează un raport care nu poate să fie schimbat fără să reface dovada de lucru. Cel mai lung lanț nu doar că servește ca dovadă a secvenței de evenimente la care a fost martor, ci demonstrează și că a venit de la cel mai mare totalizaotr de putere de procesor. Atâta timp cât majoritatea puterii procesoarelor este controlată de noduri care nu cooperează să atace rețeaua, vor genera cel mai lung lanț și vor întrece atacatorii. Rețeaua în sine cere o structură minimală. Mesajele sunt transmise pe baza celei mai bune încercări iar nodurile pot să părăsească și să se alăture rețelei după cum vor, acceptând cel mai lung lanț de dovadă de muncă ca dovadă a ceea ce s-a întâmplat cât au lipsit.

1. Introducere

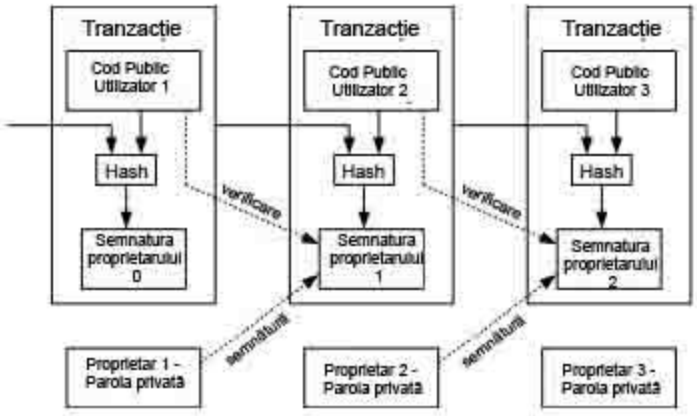
Comerțul pe internet a ajuns să se bazeze aproape exclusiv pe instituții financiare care servesc ca terți de încredere ăentru a procesa plățile electronice. În timp ce sistemul funcționează destul de bine pentru cele mai multe tranzacții, suferă de pe urma slăbiciunii inerente a modelului pe bază de încredere. Tranzacțiile total ireversibile nu sunt cu adevărat posibile, din moment ce instituțiile nu pot să ocolească medierea disputelor. Costurile de mediere cresc și costurile de tranzacționare, limitând dimensiunea tranzacției minime și tăind posibilitatea pentru tranzacții mici, așadar este un cost mult mai mare care survine în urma pierderii abilității de a face plăți ireversibile pentru servicii ireversibile. Cu posibilitatea reversibilității, nevoia de încredere dispare. Comercianții trebuie să fie suspicioși cu toți clienții, storcându-i de informații de care altfel nu ar avea nevoie. Un anumit procent de fraudă este acceptat și de neevitat. Aceste costuri și nesiguranțe ale plății pot fi evitate în persoană folosind moneda fizică, dar nu există un mecanism pentru a face plăți printr-un canal de comunicare fără o altă parte de încredere

Este nevoie de un sistem de plată electronic bazat pe dovezi criptografice în loc de încredere, permițând oricăror două plăți dispuse să tranzacționeze între ele direct fără să aibă nevoie de un terț de încredere. Tranzacțiile care sunt din punct de vedere al calculelor ireversibile ar proteja vânzătorii de fraude, iar mecanisme pentru împuterniciri

legale de rutină ar putea să fie implementate foarte ușor pentru a proteja cumpărătorii. În această lucrare, propunem o soluție pentru problema dublei-cheltuieli folosind un server distribuit de timbre de timp peer-to-peer pentru a genera dovadă computațoinală a ordinii cronologice pentru tranzacții. Sistemul este sigur atâta timp cât un nod fidel controlează colectiv mai multă putere de procesare decât oricare grup de noduri ale atacatorilor.

2. Tranzacțiile

Definim moneda electronică ca un lanț de semnături digitale. Fiecare proprietar transferă moneda către următorul semnând digital un hash al tranzacției preecdente și parola publica către următorul proprietar și adăugându-le pe acestea la capătul monezii. Cel plătit poate să verifice semnăturile pentru a verifica lanțul de proprietate.

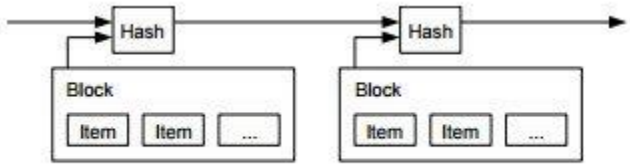


Problema, desigur, este că cel plătit nu poate să verifice că unul dintre proprietari nu a cheltuit de două ori o monedă. O soluție comună este introducerea unei autorități centrale sau a unei monetării care verifică fiecare tranzacție în parte. Dupăfiecare tranzacție, moneda trebuie să fie returnată monetăriei pentru a elibera o nouă monedă și doar monezile emise direct de către monetărie sunt de încredere. Problema cu această soluție este că soarta întrebului sistem monetar depinde de compania care conduce monetăria, iar fiecare tranzacție trebuie să treacă pe la ei, ca printr-o bancă.

Avem nevoie de o soluție pentru ca cel plătit să știe că proprietarul precedent nu a semnat o tranzacție înainte. În acest scop, cea mai timpurie tranzacție este cea care contează, așadar nu ne interesează de încercările de mai târziu de a cheltui de două ori aceeași monedă. Singura metodă de a confirma absența unei tranzacții este să fim conștienți de toate tranzacțiile. În modelul bazat pe monetărie, monetăria era conștientă de toate tranzacțiile și a decis care a intrat prima. Pentru a ajunge la acest rezultat fără un terț de încredere, tranzacțiile trebuie să fie anunțate public și avem nevoie de un sistem de participanți care să cadă de acord asupra unui singur istoric al ordinii în care a fost făcută tranzacția. Cel plătit are nevoie de o dovadă că la momentul fiecărei tranzacții, majoritatea nodurilor care au căzut de acord a fost prima primită.

3. Servere cu timbru de timp

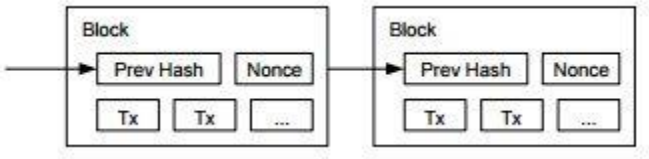
Soluția pe care o propunem începe cu un server de timbre de timp. Un server de timbre de timp funcționează luând hash-ul unui block de elemente la timbrat și publicând hash-ul în masă, de exemplu într-un zir sau într-o postare Usenet. Timbrul de timp demonstrează că data trebuie să fi existat la un moment dat, evident, pentru a primi hash-ul. Fiecare timbru de timp include timbre de timp premergătoare în hash, formând un lanț, cu fiecare timbru de timp incluzându-l pe cel de dinaintea sa.



4. Dovada de lucru

Pentru a implementa un server de timbre de timp distribuit peer-to-peer, vom avea nevoie să folosim un sistem proof-of-work [dovadă de lucru] similar cu Hashcash-ul lui Adam Back, mai degrab[dec\t ziarul sau postările Usenet. Dovada-de-lucru implică scanarea unei valori care atunci când este hashată hash-ul începe cu un număr de 0 biți – ca SHA-256. Munca medie cerută este exponențială în numărul de zero biți cerută și poate fi verificată executând un singur hash.

Pentru rețeaua noastră de timestamp, implementăm dovada de lucru prin incrementarea utilizării unice în block până când este găsită o valoare care dă hash-ului de block numărul cerut de zero bits. Odată ce eforturile procesorului au fost folosite la satisfaceera dovezii de lucru, block-ul nu poate să fie schimbat fără să refacă munca. Pe măsură ce block-uri ulterioare sunt legate de acesta, munca pentru a schimba block-ul ar include refacerea block-urilor de după el.



Dovada de lucru rezolvă de asemenea și problema determinării reprezentării în majoritatea deciziilor luate. Dacă majoritatea ar fi bazată pe o adresă de IP/un singur vot, ar putea fi răsturnată de oricine ar putea să repartizeze multe adrese de IP. Dovada de lucru este practic on procesor=un vot. Majoritatea deciziilor este reprezentată de cel mai lung lanț, care are cea mai mare dovadă de lucru investită în el. Dacă o majoritate a puterii de procesoare este concentrată de noduri fidele, lanțul fidel va crește mai repede și va depăși orice lanțuri concurente.

Pentru a modifica un block din trecut, un atacator ar trebui să refacă dovada de lucru al block-ului și al tuturor block-urilor de după el și apoi să prindă din urmă și să întreacă munca nodurilor fidele. Vă vom arăta mai târziu că probabillitatea ca un atacator să prindă din urmă scade exponențial pe măsură ce block-uri ulterioare sunt adăugate.

Pentru a compensa pentru viteza de hardware în creștere și variatele interese în rularea nodurilor cu timpul, dificultatea dovezii de lucru este determinată de o medie de mișcare care țintește numărul mediu de block-uri pe oră. Dacă sunt generate prea repede, dificultatea crește.

5. Rețeaua

Pașii pentru a rula rețeaua sunt după cum urmează:

1. Tranzacțiile noi sunt transmise către toate nodurile
2. Fiecare nod colectează tranzacțiile noi într-un block
3. Fiecare nod lucrează să găsească o dovadă de lucru dificilă pentru block-ul său.
4. Când un nod a găsit dovada de lucru, emite un block către toate nodurile.
5. Nodurile acceptă block-ul doar dacă toate tranzacțiile din el sunt valide și nu au fost deja cheltuite.
6. Nodurile își exprimă acceptarea block-ului lucrând la crearea următorului block din lanț, folosind hash-ul block-ului acceptat ca hash anterior.

Nodurile iau întotdeauna în considerare cel mai lung lanț ca fiind corect și vor continua să lucreze pentru a-l lungi. Dacă două noduri emit diferite versiuni pentru următorul block în același timp, unele noduri ar putea să îl primească pe unul sau pe altul mai întâi. În acest caz, vor lucra la primul pe care îl primesc, dar vor păstra cealaltă ramură în caz că devine mai lungă. Legătura va fi ruptă când următoarea dovadă de lucru este găsită și una dintre ramuri devine mai lungă; nodurile care lucrau la cealaltă ramură se vor muta apoi la cea mai lungă. Noile emiteri de tranzacție nu au neapărat nevoie să ajungă la toate nodurile. Atâta timp cât ajung la cât mai multe noduri, vor intra într-un block mai devreme sau mai târziu. Emisiile de block-uri sunt de asemenea tolerate la mesajele scăpate. Dacă un nod nu primește un block, îl va cere când îl primește pe următorul și își dă seama că i-a scăpat unul.

6. Motivarea

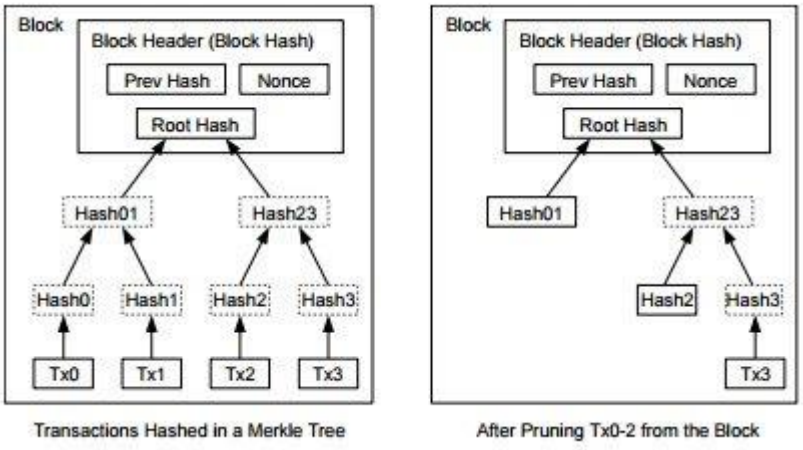
Prin convenție, prima tranzacție din block este o tranzacție specială care începe o nouă monedă deținută de creatorul unui block. Aceasta adaugă o motivație pentru ca nodurile să susțină rețeaua, și oferă o metodă de a distribui inițial monede în circulație, din moment ce nu există o autoritate centrală care să le emită. Adăugarea constantă a unei sume constante de noi monede este analogă cu minerii de aur care cheltuie resurse pentru a aduce aur în circulație, în cazul nostru fiind vorba despre timp de procesare și curent cheltuit.

Motivația poate de asemenea să fie finanțată cu taxe de tranzacționare. Dacă valoarea producției unei tranzacții este mai mică decât valoarea intrărilor, diferența este o taxă de tranzacție care este adăugată valorii stimulului block-ului care conține tranzacția. Odată ce un număr predeterminat de monede a intrat în circulație, stimului poate să tranziteze în totalitate către taxe de tranzacție și să fie în totalitate ferit de inflatie.

Motivația poate contribui să încurajeze nodurile să rămână fidele. Dacă un atacator avid reușește să facă rost de mai mulltă putere de procesare decât toate nodurile fidele, ar avea de ales între a folosi acest lucru pentru a frauda oamenii furându-le plățile sau ar putea să folosească sistemul pentru a genera noi monezi. Ar trebui să găsească că e mai profitabil să joace după reguli, reguli care îl favorizează pe el cu mai multe monezi decât toți ceilalți laolaltă, decât să submineze sistemul și validitatea propriei averi.

7. Reclamarea spațiului de pe disc

Odată ce ultima tranzacție este îngropată sub destule block-uri, tranzacția cheltuitää înaintea sa poate să fie abandonată pentru a economisi spațiu pe disc. Pentru a simplifica acest lucru fără să distrugă hash-ul block-ului, tranzacția este hash-ată într-un Merkle Tree, care are doar rădăcinile incluse în hash-ul block-ului. Block-urile vechi pot fi apoi compactate tăind ramurile copacului. Hash-urile interioare nu trebuie să fie stocate.

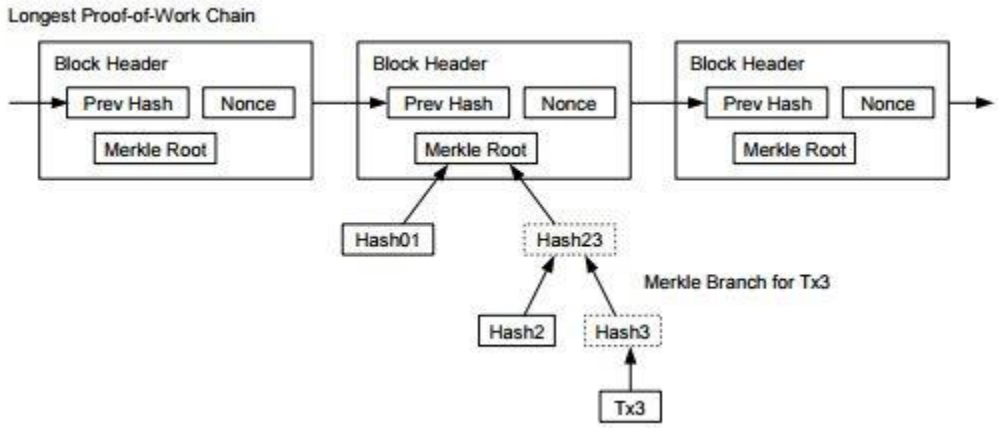


Un header de block fără tranzzacții ar avea cam 80 de biți. Dacă presupunem că bock-urile sunt generate la fiecare 10 minute, 80 biți x 6 x24 x 365 =4.2MB pe an. Ținând cont că sistemele de calculatoare se vând în mod normal 2GB de RAM începând cu 2008, și Legea lui Moore care anticipează o creștere de 1.2GB pe an, stocarea nu ar trebui să fie o problemă nici dacă headerele de block-uri ar trebui să fie păstrate pe memorie.

8. Verificarea simplificată a plăților

Este posibil să verifici plățile fără să rulezi un nod de rețea în întregime. Un utilizator are nevoie doar să păstreze o copie a header-ului de block pentru cel mai lung lanț de dovadă de lucru, pe care o poate obține interogând nodurile rețelei până când este sigur că are cel mai lung lanț și să obțină ramura Merkle care face legătura dintre tranzacție și block-ul cu timbur de timp. Nu poate să verifice singur tranzacția, dar făcând legătura cu un loc din lanț poate să vadă că un nod de rețea l-a acceptat, iar

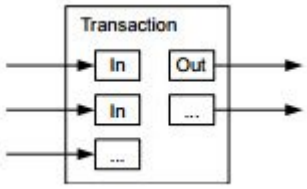
block-urile adăugate după confirmă în continuare că rețeaua l-a acceptat.



Astfel, verificarea este de încredere atâta timp cât nodurile fidele controlează rețeaua, dar este mai vulnerabilă dacă rețeaua este înfrântă de un atacator. Atâta timp cât nodurile de rețea pot să verifice singure tranzacțiile, metoda simplificată poate fi păcălită de tranzacțiile inventate de un atacator atâta timp cât acesta poate să continue să depășească rețeaua. O strategie pentru a te proteja de acest lucru ar fi să accepți alerte de la nodurile de rețea când detectează un block invalid, determinând programul utilizatorului să descarce întregul block și alertând tranzacția pentru a confirma inconsecvența. Afacerile care primesc frecvent plăți vor vrea probabil în continuare să ruleze propriile noduri pentru siguranță independentă și verificare mai rapidă.

9. Combinarea și disocierea valorii

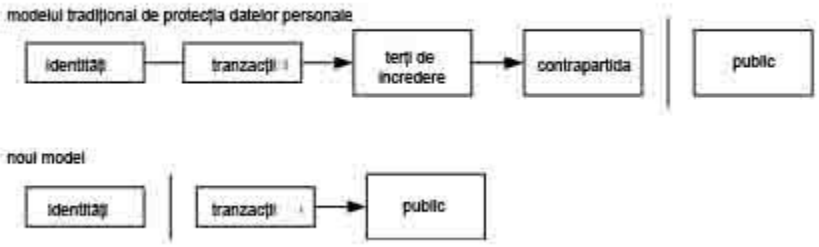
Deși ar părea posibil să manevrezi monedele individual, ar fi greoi să faci o tranzacție separată pentru fiecare cent dintr-un transfer. Pentru a permite valorii să fie împărțite și combinate, tranzacțiile conțin mai multe inputuri și outputuri. În mod normal va fi fie un singur input dintr-o tranzacție precedentă mai mare sau mai multe inputuri combinând sume mai mici, ori cel mult două outputuri: unul pentru plăți și unul returnând restul, dacă este cazul, emițătorului.



Este de notat că fan-out-ul [numărul de inputuri care poate fi conectat la un output specificat], în care o tranzacție depinde de mai multe tranzacții și acele tranzacții depind de mai multe, nu ridică aici o problemă. Nu este niciodată nevoie să extragi o copie completă independentă a istoricului tranzacției.

10. Dreptul la intimitate

Modelul bancar tradițional ajunge la un nivel de intimitate limitând accesul la informație doar părților implicate și terților de încredere. Necesitatea de a anunța toate tranzacțiile public exclude această metodă, dar dreptul la intimitate poate fi menținut întrerupând cursul informațiilor în alte locuri: păstrând codurile publice anonime. Publicul poate să vadă că cineva trimite o sumă către altcineva, dar fără informații care să lege tranzacțiile de o persoană. Acest sistem este similar nivelului de informație emis de bursă, unde momentul și dimensiunea tranzacțiilor individuale, "caseta", este făcută publică, fără însă a spune care sunt părțile implicate.



Ca metodă de siguranță suplimentară, o nouă pereche de coduri ar trebui să fie folosită pentru fiecare tranzacție pentru ca tranzacția să nu poate fi legată de un anumit proprietar. Anumite legături sunt de neevitat cu tranzacțiile cu mai multe inputuri, care implicit dezvăluie că inputul lor a fost folosit de un anumit proprietar. Riscul este ca dacă utilizatorul unui cod este dezvăluit, legăturile ar putea să se facă și cu alte tranzacții care aparțin aceluiași proprietar.

11. Estimate

Considerăm ipoteza unui atacator care încearcă să genereze un lanț alternativ mai rapid decât un lanț fidel. Dacă reușește acest lucru, nu deschide sistemul către schimbări arbitrare, cum ar fi crearea de valoare din piatră seacă sau posibilitatea de a lua bani care nu i-au aparținut niciodată atacatorului. Nodurile nu vor accepta o tranzacție invalidă ca plată, iar nodurile fidele nu vor accepta niciodată un block care conține astfel de tranzacții. Un atacator poate doar să încerce să schimbe uan din propriile tranzacții pentru a lua înapoi bani pe care i-a cheltuit de curând.

Cursa dintre lanțul onest și lanțul atacator poate să fie caracterizată ca un traseu binominal random. Succesul evenimentului este ideea că lanțul fidel este mărit cu un block, crescând avansul cu +1, iar nereușita este ca lanțul atacatorului să fie extins cu un block reducând diferența cu -1.

Probabilitatea ca un atacator care are un deficit să prindă din urmă un lanț fidel, este analogă cu problema cunoscută ca Gambler's Ruin. Să presupunem că un jucător de pariuri cu credit nelimitat începe cu un deficit și joacă un număr ipotetic infinit de jocuri pentru a încerca să ajungă la un prag de rentabilitate. Putem să calculăm probabilitatea de a ajunge la pragul de rentabilitate, sau felul în care un atacator prinde din urmă un lanț fidel, după cum urmează:

p= probabilitatea ca un nod fidel să găsească următorul block

q=probabilitatea ca atacatorul să găsească următorul block

q2=probabilitatea ca atacatorul să nu prindă niciodată din urmă numărul de z blocuri avans al nodului fidel

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

Ținând cont de presupunerea noastră că p>q, probabilitatea scade exponențial pe măsură ce numărul de block-uri pe care atacatorul trebuie să le prindă din urmă crește. Ținând cont că șansele sunt împotriva lui, dacă nu face un salt norocos înainte încă de la început, șansele sale devin din ce în ce mai mici pe măsură ce rămâne din ce în ce mai mult în urmă.

Acum trebuie să luăm în calcul cât timp îi ia receptorului unei noi tranzacții să aștepte înainte de a fi destul de sigur că un anumit emițător nu poate să schimbe tranzacția. Presupunem că cel care trimite este un atacator care vrea să îl facă pe receptor să creadă o perioadă de timp că a fost plătit, iar apoi să comuteplata către el după ce acea perioadă de timp a trecut. Receptorul va fi atenționat când acest lucru se întâmplă, dar cel care trimite speră că acest lucru se va întâmpla cât mai târziu. Receptorul generează o nouă pereche de cod și dă codul public celui care trimite la scurt timp înainte să semneze. Acest lucru previne problema ca cel care trimite să pregătească un lanț de block-uri dinainte lucrând la el în continuu până când are norocul să prindă avans, apoi excută tranzacția în acel moment. Odată ce tranzacția a fost trimisă, cel care trimite începe să lucreze în secret la un lanț paralel care să conțină o versiune alternativă pentru tranzacția sa.

Recipientul așteaptă până când tranzacția a fost adăugată la un block și z block-uri au fost aliniate după acesta. Nu știe exact numărul de acțiuni pe care atacatorul l-a făcut, dar presupunând că block-ul fidel a așteptat un număr mediu de timp pentru un block, progresul potențial al atacatorului va fi o distribuție Poisson cu valoarea:

λ=zqp

Pentru a obține probabilitatea ca atacatorul să prindă din urmă acum, înmulțim densitatea Poisson pentru fiecare sumă de progrez pe care l-ar fi putut face cu probabilitatea că ar fi putut să prindă din urmă din acel punct.

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

Rearanjând pentru a evita să adunăm o coadă infinită de distribuție…

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

Și convertind în cod C

```
#include

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;

}
```

Rulând câteva rezultate, putem vedea probabilitatea care scade exponențial cu z

q=0.1

z=0 P=1.0000000

z=1 P=0.2045873

z=2 P=0.0509779

z=3 P=0.0131722

z=4 P=0.0034552

z=5 P=0.0009137

z=6 P=0.0002428

z=7 P=0.0000647

z=8 P=0.0000173

z=9 P=0.0000046

z=10 P=0.0000012

q=0.3

z=0 P=1.0000000

z=5 P=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

Rezolvând pentru P mai mic de 0.1%…

P < 0.001

q=0.10 z=5

q=0.15 z=8

q=0.20 z=11

q=0.25 z=15

q=0.30 z=24

q=0.35 z=41

q=0.40 z=89

q=0.45 z=340

12. Concluzii

Am propus un sistem pentru tranzacții electronice fără să ne bazăm pe încredere. Am început cu cadrul de lucru obișnuit de monede făcute din semnatura digitală, care oferă control forte asupra proprietății, dar este incomplet fără o metodă de a preveni dubla cheltuire. Pentru a rezolva această problemă, am propus o rețea peer-to-peer folosind dovada de lucru care înregistrează un istoric public al tranzacțiilor. Astfel, devine rapid impractic din punct de vedere computațional, pentru un atacator să schimbe ceva dacă noduri fidele controlează o mare parte din puterea computațională. Rețeaua este robustă în simplitatea sa nestructurată. Nodurile lucrează toate în același timp cu un minim de coordonare. Nu au nevoie să fie identificate, din moment ce mesajele nu sunt routate către vreun loc în particular. Nodurile pot să părăsească și să se alăture rețelei după bunul plac, acceptând lanț cu dovadă de lucru ca dovadă a evenimentelor care au avut loc cât au lipsit. Votează cu puterea de procesare, își exprimă acceptul block-urilor valide lucrând la expandarea lor și resping block-urile nevalide refuzând să lucreze cu ele. Orice reguli de care este nevoie și motivații pot fi impuse cu acest mecanism de aprobare unanimă.

Referințe

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

[2] H. Massias, X.S. Avila, and J.-J. Quisquater, "Design of a secure timestamping service with minimal trust requirements," In *20th Symposium on Information Theory in the Benelux*, May 1999.

[3] S. Haber, W.S. Stornetta, "How to time-stamp a digital document," In *Journal of Cryptology*, vol 3, no 2, pages 99-111, 1991.

[4] D. Bayer, S. Haber, W.S. Stornetta, "Improving the efficiency and reliability of digital time-stamping," In *Sequences II: Methods in Communication, Security and Computer Science*, pages 329-334, 1993.

[5] S. Haber, W.S. Stornetta, "Secure names for bit-strings," In *Proceedings of the 4th ACM Conference on Computer and Communications Security*, pages 28-35, April 1997.

[6] A. Back, "Hashcash - a denial of service counter-measure," http://www.hashcash.org/papers/hashcash.pdf, 2002.

[7] R.C. Merkle, "Protocols for public key cryptosystems," In *Proc. 1980 Symposium on Security and Privacy*, IEEE Computer Society, pages 122-133, April 1980.

[8] W. Feller, "An introduction to probability theory and its applications," 1957.