Bitcoin: Sistem i Arkës Elektronike Peer-to-Peer

Translated in Albanian from bitcoin.org/bitcoin.pdf

by Fatjon (Tony) Xhufi @TonyXhufi

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

Abstrakt. Një version thjesht peer-to-peer të parave elektronike do të lejonte që pagesat në internet të dërgoheshin direkt nga njëra palë në tjetrën pa kaluar përmes një institucioni financiar. Nënshkrimet dixhitale japin një pjesë të zgjidhjes, por përfitimet kryesore humbasin nëse një palë e tretë e besuar ende kërkohet për të parandaluar shpenzimet e dyfishta. Ne propozojmë një zgjidhje për problemin e shpenzimeve të dyfishta duke përdorur një rrjet peer-to-peer. Transaksionet me vula kohore të rrjetit vendosen në një zinxhir të vazhdueshëm nepermjet procesit të provës së punës të bazuar në hash, duke formuar një rekord që nuk mund të ndryshohet pa ribërë provën e punës. Zinxhiri më i gjatë jo vetëm që shërben si provë e sekuencës së ngjarjeve të dëshmuara, por edhe provë se erdhi nga grupi më i madh i fuqisë së CPU-së. Për sa kohë që një shumicë e energjisë së CPU-së kontrollohet nga nyjet që nuk po bashkëpunojnë për të sulmuar rrjetin, ato do të gjenerojnë zinxhirin me te gjate per te tejkaluar sulmuesit e rrjetit. Vetë rrjeti kërkon strukturë minimale. Mesazhet transmetohen në bazë të përpjekjeve më të mira, dhe nyjet mund të largohen dhe të bashkohen përsëri me rrjetin sipas dëshirës, duke pranuar zinxhirin më të gjatë të provës së punës si provë të asaj që ndodhi ndërsa ishin jo aktive ne rrjet.

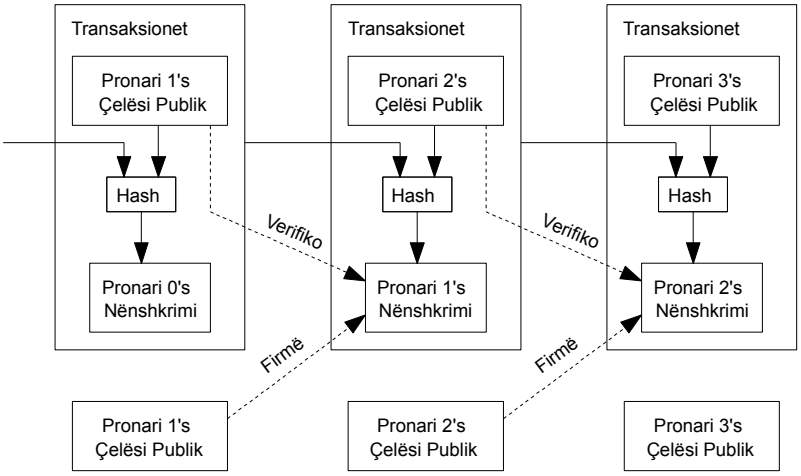
1. Hyrje

Tregtia në internet është mbështetur pothuajse ekskluzivisht në institucionet financiare që shërbejnë si palë të treta të besuara për të përpunuar pagesat elektronike. Ndërsa sistemi funksionon mjaft mirë për shumicën e transaksioneve, ai ende vuan nga dobësitë e qenësishme të modelit të bazuar në besim. Transaksionet plotësisht të pakthyeshme nuk janë vërtet të mundshme, pasi institucionet financiare nuk mund të shmangin mosmarrëveshjet ndërmjetësuese. Kostoja e ndërmjetësimit rrit kostot e transaksionit, duke kufizuar madhësinë minimale të transaksionit praktik dhe duke prerë mundësinë për transaksione të vogla e të rastësishme, dhe ka një kosto më të gjerë në humbjen e aftësisë për të bërë pagesa jo të kthyeshme për shërbime të pakthyeshme. Me mundësinë e përmbysjes, nevoja për besim përhapet. Tregtarët duhet të jenë të kujdesshëm ndaj klientëve të tyre, duke i munduar ata për më shumë informacion sesa do të kishin nevojë përndryshe. Një përqindje e caktuar e mashtrimit pranohet si e pashmangshme. Këto kosto dhe pasiguri pagese mund të shmangen personalisht duke përdorur monedhën fizike, por asnjë mekanizëm nuk ekziston për të kryer pagesa përmes një kanali komunikimi pa një palë të besuar.

Ajo që është e nevojshme është një sistem elektronik pagese i bazuar në prova kriptografike në vend të besimit, duke lejuar çdo dy palë të gatshme të bëjnë transaksione direkt me njëra-tjetrën pa pasur nevojë për një palë të tretë të besuar. Transaksionet që janë llogaritëse jopraktike për t'u kthyer në të, do të mbronin shitësit nga mashtrimi dhe mekanizmat rutinë të ruajtjes mund të zbatoheshin lehtësisht për të mbrojtur blerësit. Në këtë publikim, ne propozojmë një zgjidhje për problemin e shpenzimeve të dyfishta duke përdorur një server të decentralizuar peer-to-peer me vula kohore për të gjeneruar prova llogaritëse të rendit kronologjik të transaksioneve. Sistemi është i sigurt për sa kohë që nyjet e ndershme kolektivisht kontrollojnë më shumë fuqi CPU sesa çdo grup bashkëpunues i nyjeve sulmuese.

2. Transaksionet

Ne përcaktojmë një monedhë elektronike si një zinxhir të nënshkrimeve dixhitale. Secili pronar transferon monedhën tek tjetri duke nënshkruar dixhitalisht një hash të transaksionit të mëparshëm dhe çelësin publik të pronarit tjetër dhe duke i shtuar këto në fund të monedhës. Një i paguar mund të verifikojë nënshkrimet për të verifikuar zinxhirin e pronësisë.

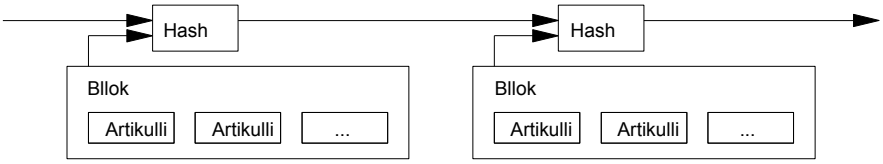


Problemi natyrisht është që i paguari nuk mund të verifikojë që një nga pronarët nuk e ka shpenzuar dy herë monedhën. Një zgjidhje e zakonshme është prezantimi i një autoriteti qendror të besuar, ose pres monedhe, që kontrollon çdo transaksion për shpenzime të dyfishta. Pas çdo transaksioni, monedha duhet të kthehet në pres (krijim monedhe) për të lëshuar një monedhë të re, dhe vetëm monedhat e lëshuara drejtpërdrejt nga presi besohet se nuk do të shpenzohen dy herë. Problemi me këtë zgjidhje është se fati i të gjithë sistemit të parave varet nga kompania që drejton presin e monedhave, me çdo transaksion që duhet të kalojë nëpër to, ashtu si një bankë.

Na duhet një mënyrë që i paguari të dijë që pronarët e mëparshëm nuk kanë nënshkruar ndonjë transaksion të mëparshëm. Për qëllimet tona, transaksioni më i hershëm është ai që vlen, kështu që nuk na interesojnë përpjekjet e mëvonshme për të shpenzuar dy herë. Mënyra e vetme për të konfirmuar mungesën e një transaksioni është të jesh i vetëdijshëm për të gjitha transaksionet. Në modelin e bazuar në presin e monedhes, presi ishte në dijeni të të gjitha transaksioneve dhe vendosi se cilat mbërritën të parat. Për ta arritur këtë pa një palë të besuar, transaksionet duhet të shpallen publikisht [1], dhe na duhet një sistem që pjesëmarrësit të bien dakord për një histori të vetme të rendit në të cilin u morën. Paguesit i duhet prova se në kohën e çdo transaksioni, shumica e nyjeve ranë dakord se ishte e para e marrë.

3. Serveri i Vulës Kohore

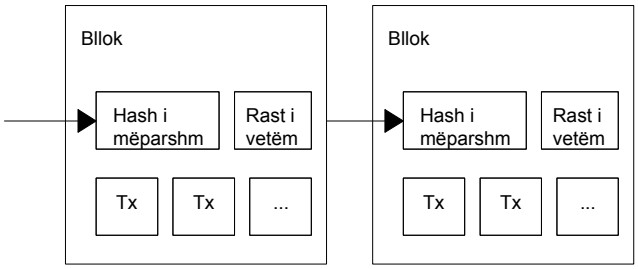
Zgjidhja që ne propozojmë fillon me një server i vulës kohore. Një server i vulës kohore punon duke marrë një hash të një blloku të artikujve për tu vulosur në kohë dhe duke botuar gjerësisht hashin, të tilla si në një gazetë ose postim në Usenet [2-5]. Vula kohore provon që të dhënat duhet të kenë ekzistuar në atë kohë, padyshim, në mënyrë që të futen në hash. Secila vulë kohore përfshin vulën e mëparshme kohore në hash-in e saj, duke formuar një zinxhir, me secilën vulë shtesë kohore që përforcon ato para tij.



4. Prova e Punës

Për të implementuar një server të decentralizuar të vulës kohore në bazë peer-to-peer, ne do të duhet të përdorim një sistem provë-pune të ngjashëm me Hashcash të Adam Back [6], në vend të postimeve në gazeta ose Usenet. Prova e punës përfshin skanimin për një vlerë që kur hashed, të tilla si me SHA-256, hashi fillon me një numër zero bitësh. Puna mesatare e kërkuar është eksponenciale në numrin e bitëve zero të kërkuara dhe mund të verifikohet duke ekzekutuar një hash të vetëm.

Për rrjetin tonë të vulës kohore, ne implementojmë provën e punës duke shtuar një rast të vetem në bllok derisa të gjendet një vlerë që i jep bllokut të hashed bitët e kërkuara zero. Sapo të jetë harxhuar përpjekja e CPU -së për ta bërë atë të kënaqur provën e punës, blloku nuk mund të ndryshohet pa e bërë përsëri punën. Ndërsa blloqet e mëvonshëm janë lidhur me zinxhirë pas tij, puna për të ndryshuar bllokun do të përfshijë ribërjen e të gjitha blloqeve pas tij.



Prova e punës gjithashtu zgjidh problemin e përcaktimit të përfaqësimit në vendimmarrjen e shumicës. Nëse shumica do të bazohej në një-adresë IP-një-votë, ajo mund të përmbyset nga çdokush që mund të caktojë shumë IP. Prova e punës është në thelb një-CPU-një votë. Vendimi i shumicës përfaqësohet nga zinxhiri më i gjatë, i cili ka përpjekjen më të madhe për punën e investuar në të. Nëse një shumicë e fuqisë së CPU-së kontrollohet nga nyjet e ndershme, zinxhiri i ndershëm do të rritet më shpejt dhe do të tejkalojë çdo zinxhir konkurrues. Për të modifikuar një bllok të kaluar, një sulmues do të duhet të ribëjë provën e punës së bllokut dhe të gjitha blloqeve pas tij dhe pastaj të arrijë dhe të tejkalojë punën e nyjeve të ndershme. Ne do të tregojmë më vonë se probabiliteti që një sulmues më i ngadaltë të arrijë zvogëlohet në mënyrë eksponenciale kur shtohen blloqet pasuese.

Për të kompensuar rritjen e shpejtësisë së pajisjes dhe interesin e ndryshëm për nyjet drejtuese me kalimin e kohës, vështirësia e provës së punës përcaktohet nga një mesatare lëvizëse që synon një numër mesatar të blloqeve në orë. Nëse ato gjenerohen shumë shpejt, vështirësia rritet.

5. Rrjeti

Hapat për të drejtuar rrjetin janë si më poshtë:

1) Transaksionet e reja transmetohen në të gjitha nyjet.
2) Çdo nyje mbledh transaksione të reja në një bllok.
3) Secila nyje punon për gjetjen e një prove të vështirë të punës për bllokun e saj.
4) Kur një nyje gjen një provë të punës, ajo transmeton bllokun në të gjitha nyjet.
5) Nyjet e pranojnë bllokun vetëm nëse të gjitha transaksionet në të janë të vlefshme dhe nuk janë shpenzuar tashmë.
6) Nyjet shprehin pranimin e tyre të bllokut duke punuar në krijimin e bllokut tjetër në zinxhir, duke përdorur hashin e bllokut të pranuar si hash i mëparshëm.

Nyjet gjithmonë e konsiderojnë zinxhirin më të gjatë si të duhur dhe do të vazhdojnë të punojnë në zgjatjen e tij. Nëse dy nyje transmetojnë versione të ndryshme të bllokut tjetër njëkohësisht, disa nyje mund të marrin njërën ose tjetrën së pari. Në atë rast, ata punojnë në të parën që morën, por ruajnë degën tjetër në rast se zgjatet më shumë. Lidhja do të prishet kur të gjendet prova tjetër e punës dhe një degë të bëhet më e gjatë; nyjet që ishin duke punuar në degën tjetër do të kalojnë në atë më të gjatë.

Transmetimet e reja të transaksioneve nuk kanë nevojë domosdoshmërisht për të arritur të gjitha nyjet. Për sa kohë që ata arrijnë shumë nyje, ata do të futen në një bllok para shumë kohësh. Transmetimet e blloqeve janë gjithashtu tolerante ndaj mesazheve të rëna. Nëse një nyje nuk merr një bllok, ajo do ta kërkojë atë kur të marrë bllokun tjetër dhe të kuptojë se e ka humbur një.

6. Nxitjet

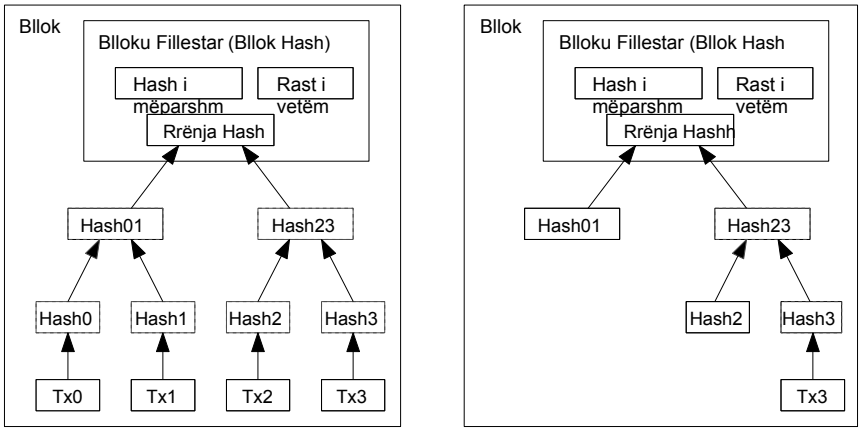
Sipas konventës, transaksioni i parë në një bllok është një transaksion i veçantë që fillon një monedhë të re në pronësi të krijuesit të bllokut. Kjo shton një nxitje për nyjet për të mbështetur rrjetin dhe ofron një mënyrë për të shpërndarë fillimisht monedhat në qarkullim, pasi nuk ka autoritet qendror për t'i lëshuar ato. Shtimi i qëndrueshëm i një konstante të shumës së monedhave të reja është analoge me minatorët e arit që shpenzojnë burime për të shtuar ar në qarkullim. Në rastin tonë, është koha e procesorit dhe energjia elektrike që harxhohen.

Nxitja gjithashtu mund të financohet me tarifa transaksioni. Nëse vlera e prodhimit të një transaksioni është më e vogël se vlera e saj e hyrjes, diferenca është një tarifë transaksioni që i shtohet vlerës stimuluese të bllokut që përmban transaksionin. Sapo një numër i paracaktuar monedhash të ketë hyrë në qarkullim, stimulimi mund të kalojë tërësisht në tarifat e transaksioneve dhe të jetë plotësisht pa inflacion.

Nxitja mund të ndihmojë në inkurajimin e nyjeve për të qëndruar të ndershëm. Nëse një sulmues lakmitar është në gjendje të mbledhë më shumë fuqi CPU sesa të gjitha nyjet e ndershme, ai do të duhet të zgjedhë midis përdorimit të tij për të mashtruar njerëzit duke i vjedhur përsëri pagesat e tij, ose duke e përdorur atë për të gjeneruar monedha të reja. Ai duhet ta konsiderojë më fitimprurëse të veprojë sipas rregullave, rregulla të tilla që e favorizojnë atë me më shumë monedha të reja se të gjithë të tjerët së bashku, sesa të minojë sistemin dhe vlefshmërinë e pasurisë së tij.

7. Rikuperim i Hapësirës së Diskut

Sapo transaksioni i fundit në një monedhë të vendoset nën blloqe të mjaftueshme, transaksionet e shpenzuara para tij mund të hidhen poshtë për të kursyer hapësirë në disk. Për ta lehtësuar këtë pa prishur hashin e bllokut, transaksionet hashen në një Merkle Tree [7] [2] [5], me vetëm rrënjën të përfshirë në hashin e bllokut. Blloqet e vjetra mund të kompaktohen duke shkëputur degët e pemës. Hashet e brendshme nuk kanë nevojë të ruhen.



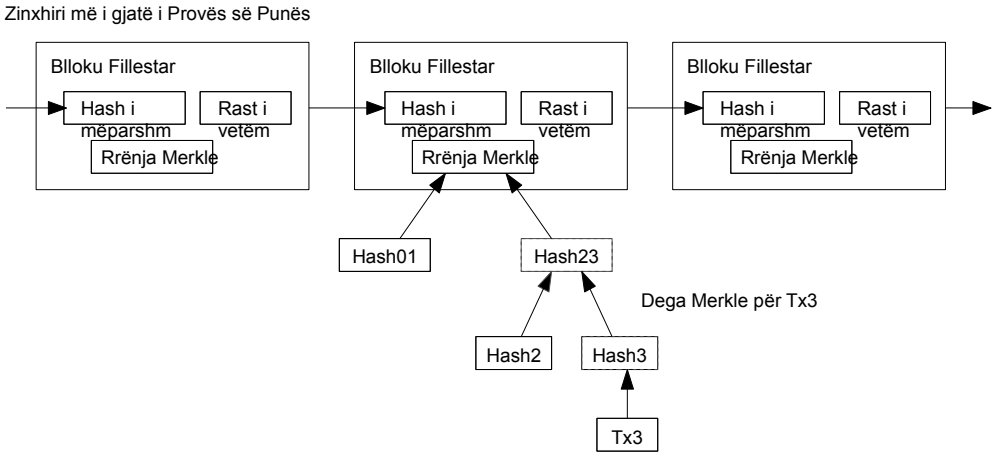
Transaksionet Hashed në Merkle Tree

Mbas krasitjes Tx0-2 nga Blloku

Fillim i bllokut pa transaksione do të ishte rreth 80 bajt. Nëse supozojmë se blloqet gjenerohen çdo 10 minuta, 80 bajte * 6 * 24 * 365 = 4.2 MB në vit. Me sistemet kompjuterike që shiten zakonisht me 2 GB RAM që nga viti 2008, dhe Ligji i Moore që parashikon rritjen aktuale prej 1.2 GB në vit, hapësira e ruajtjes nuk duhet të jetë problem edhe nëse fillimet e bllokut duhet të mbahen në kujtesë.

8. Verifikim i Thjeshtuar i Pagesës

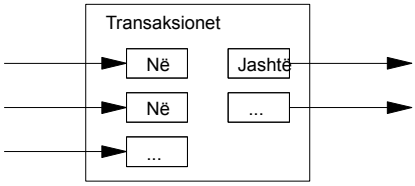
Është e mundshme të verifikoni pagesat pa ekzekutuar një nyje të plotë të rrjetit. Një përdorues duhet të mbajë vetëm një kopje të fillimit të bllokut të zinxhirit më të gjatë të provës së punës, të cilën ai mund të marrë duke kërkuar nyjet e rrjetit derisa të bindet se ka zinxhirin më të gjatë dhe të marrë degën Merkle që lidh transaksionin me bllokun që është i vulosur në kohë. Ai nuk mund ta kontrollojë vetë transaksionin, por duke e lidhur atë me një vend në zinxhir, ai mund të shohë që një nyje rrjeti e ka pranuar atë, dhe blloqet e shtuara pasi të konfirmojnë më tej se rrjeti e ka pranuar atë.



Si i tillë, verifikimi është i besueshëm për sa kohë që nyjet e ndershme kontrollojnë rrjetin, por është më e prekshme nëse rrjeti mbizotëron nga një sulmues. Ndërsa nyjet e rrjetit mund të verifikojnë vetë transaksionet, metoda e thjeshtuar mund të mashtrohet nga transaksionet e fabrikuara të një sulmuesi për aq kohë sa sulmuesi mund të vazhdojë të mbizotërojë rrjetin. Një strategji për tu mbrojtur nga kjo do të ishte pranimi i sinjalizimeve nga nyjet e rrjetit kur zbulojnë një bllok të pavlefshëm, duke nxitur softuerin e përdoruesit të shkarkojë bllokun e plotë dhe transaksionet e paralajmëruara për të konfirmuar mospërputhjen. Bizneset që marrin pagesa të shpeshta ndoshta do të dëshirojnë ende të drejtojnë nyjet e tyre për një siguri më të pavarur dhe verifikim më të shpejtë.

9. Kombinimi dhe Ndarj*a e Vleres*

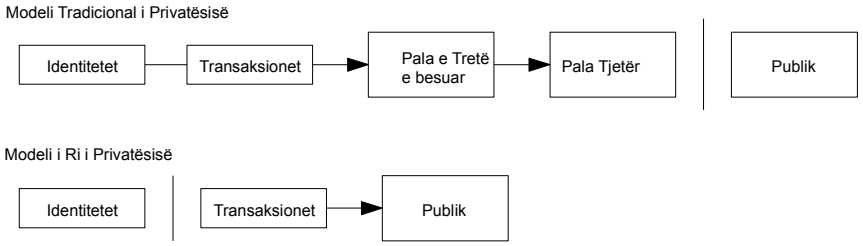
Megjithëse do të ishte e mundur të merresh me monedha individualisht, do të ishte e palodhur të bësh një transaksion të veçantë për çdo cent në një transferim. Për të lejuar ndarjen dhe kombinimin e vlerës, transaksionet përmbajnë hyrje dhe dalje të shumta. Normalisht do të ketë ose një hyrje të vetme nga një transaksion i mëparshëm më i madhë ose hyrje të shumëfishta që kombinojnë shuma më të vogla, dhe më së shumti dy rezultate: një për pagesën dhe një që i kthen ndryshimin, nëse ka, përsëri te dërguesi.



Duhet të theksohet se shpërndarja, ku një transaksion varet nga disa transaksione, dhe ato transaksione varen nga shumë më tepër, nuk është problem këtu. Asnjëherë nuk ka nevojë për të nxjerrë një kopje të plotë të pavarur të historisë së një transaksioni.

10. Privatësia

Modeli tradicional bankar arrin një nivel të privatësisë duke kufizuar aksesin në informacion për palët e përfshira dhe palën e tretë të besuar. Nevoja për të njoftuar të gjitha transaksionet përjashton publikisht këtë metodë, por privatësia ende mund të ruhet duke thyer rrjedhën e informacionit në një vend tjetër: duke mbajtur anonim çelësat publikë. Publiku mund të shohë se dikush po i dërgon një shumë dikujt tjetër, por pa informacion që lidh transaksionin me dikë. Kjo është e ngjashme me nivelin e informacionit të lëshuar nga bursat, ku koha dhe madhësia e tregtive individuale, "shiriti", bëhen publike, por pa treguar se kush ishin palët.



Si një firewall shtesë, një çift i ri çelësash duhet të përdoret për çdo transaksion për t'i mbajtur ata të mos lidhen me një pronar të përbashkët. Disa lidhje janë ende të pashmangshme me transaksione me shumë hyrje, të cilat zbulojnë domosdoshmërisht që inputet e tyre ishin në pronësi të të njëjtit pronar. Rreziku është që nëse zbulohet pronari i një çelësi, lidhja mund të zbulojë transaksione të tjera që i përkisnin të njëjtit pronar.

11. Llogaritjet

Ne e konsiderojmë skenarin e një sulmuesi që përpiqet të gjenerojë një zinxhir alternativ më shpejt sesa zinxhiri i sinqert. Edhe nëse kjo është arritur, ajo nuk e hedh sistemin të hapur ndaj ndryshimeve arbitrare, të tilla si krijimi i vlerës nga asnjë gjëja ose marrja e parave që kurrë nuk i përkisnin sulmuesit. Nyjet nuk do të pranojnë një transaksion të pavlefshëm si pagesë dhe nyjet e ndershme nuk do të pranojnë kurrë një bllok që i përmban ato. Një sulmues mund të përpiqet të ndryshojë vetëm një nga transaksionet e tij për të marrë para që ka shpenzuar së fundmi.

Gara midis zinxhirit të ndershëm dhe një zinxhiri sulmues mund të karakterizohet si një Shëtitje e Rastit Binomiale. Ngjarja e suksesit është zinxhiri i ndershëm që zgjatet nga një bllok, duke rritur epërsinë e tij me +1, dhe ngjarja e dështimit është zinxhiri i sulmuesit që zgjatet nga një bllok, duke zvogëluar boshllëkun me -1.

Mundësia që një sulmues të arrijë një deficit të caktuar është analoge me problemin e Gambler's Ruin. Supozoni se një lojtar me kredi të pakufizuar fillon me një deficit dhe luan potencialisht një numër të pafund provash për t'u përpjekur të arrijë rentabilitetin. Ne mund të llogarisim mundësinë që ai të arrijë ndonjëherë rentabilitetin, ose që një sulmues të kapë ndonjëherë zinxhirin e ndershëm, si më poshtë [8]:

p = probabiliteti që një nyje e sinqertë gjen bllokun tjetër
q = probabiliteti që sulmuesi të gjejë bllokun tjetër
qz = probabiliteti që sulmuesi do të arrijë ndonjëherë nga z blloqet prapa

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

Duke pasur parasysh supozimin tonë se p> q, probabiliteti bie në mënyrë eksponenciale pasi numri i blloqeve që sulmuesi duhet të arrijë rritet. Me shanset kundër tij, nëse ai nuk bën një lëvizje herët përpara, shanset e tij bëhen jashtëzakonisht të vogla ndërsa ai mbetet prapa.

Tani konsiderojmë se për sa kohë duhet të presë marrësi i një transaksioni të ri para se të jemi mjaft të sigurt që dërguesi nuk mund ta ndryshojë transaksionin. Ne supozojmë se dërguesi është një sulmues që dëshiron ta bëjë marrësin të besojë se ai e pagoi atë për një kohë, dhe më pas ta kalojë atë për t'i paguar vetvetes pasi të ketë kaluar ca kohë. Marrësi do të lajmërohet kur kjo të ndodhë, por dërguesi shpreson se do të jetë vonë.

Marrësi gjeneron një çift të ri çelësash dhe i jep çelësin publik dërguesit pak para nënshkrimit. Kjo e ndalon dërguesin të përgatisë një zinxhir blloqesh para kohe duke punuar në të vazhdimisht derisa të jetë me fat që të dalë mjaft përpara, pastaj të ekzekutojë transaksionin në atë moment. Sapo të dërgohet transaksioni, dërguesi i pandershëm fillon të punojë në fshehtësi në një zinxhir paralel që përmban një version alternativ të transaksionit të tij.

Marrësi pret derisa transaksioni të shtohet në një bllok dhe blloqet z të jenë lidhur pas tij. Ai nuk e di sasinë e saktë të progresit që sulmuesi ka bërë, por duke supozuar se blloqet e ndershme kanë marrë kohën mesatare të pritur për bllok, progresi i mundshëm i sulmuesit do të jetë një shpërndarje e teoris Poisson me vlerën e pritur:

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

Për të marrë probabilitetin që sulmuesi mund të arrijë tani, ne shumëzojmë dendësinë e Poisson për secilën sasi të progresit që ai mund të ketë bërë me probabilitetin që ai të mund të arrijë nga ajo pikë:

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

Riorganizimi për të shmangur mbledhjen e bishtit infinit të shpërndarjes...

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

Konvertimi në kodin C...

```
#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;
}
```

Duke ekzekutuar disa rezultate, mund të shohim që probabiliteti bie në mënyrë eksponenciale me 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
```

Zgjidhja për P më pak se 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. Përfundim

Ne kemi propozuar një sistem për transaksione elektronike pa u mbështetur në besimin. Ne filluam me kornizën e zakonshme të monedhave të bëra nga nënshkrimet dixhitale, e cila siguron kontroll të fortë të pronësisë, por është e paplotë pa ndonjë mënyrë për të parandaluar shpenzimet e dyfishta. Për të zgjidhur këtë, ne propozuam një rrjet peer-to-peer duke përdorur provën e punës për të regjistruar një histori publike të transaksioneve që shpejt bëhet kompjuterike jopraktike për një sulmues të ndryshojë nëse nyjet e ndershme kontrollojnë shumicën e fuqisë së CPU.
Rrjeti është i fortë në thjeshtësinë e tij të pastrukturuar. Nyjet funksionojnë njëkohësisht me pak koordinim. Ato nuk kanë nevojë të identifikohen, pasi mesazhet nuk drejtohen në ndonjë vend të veçantë dhe duhet të dorëzohen vetëm në bazë të përpjekjeve më të mira. Nyjet mund të largohen dhe të bashkohen përsëri me rrjetin sipas dëshirës, duke pranuar zinxhirin e provës së punës si provë të asaj që ndodhi ndërsa ishin zhdukur. Ata votojnë me fuqinë e tyre të CPU-së, duke shprehur pranimin e tyre të blloqeve të vlefshme duke punuar në zgjatjen e tyre dhe duke refuzuar blloqet e pavlefshme duke refuzuar të punojnë në to. Rregullat dhe stimujt e nevojshëm mund të zbatohen me këtë mekanizëm konsensusi.

Referencat

[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.