Bitcoin: Peer-to-Peer systém elektronickej hotovosti

Satoshi Nakamoto
satoshin@gmx.com
www.bitcoin.org
SK preklad: www.dvadsatjeden.org

Abstrakt. Čisto peer-to-peer verzia elektronickej hotovosti by umožnila posielať online platby priamo od jednej strany k druhej bez sprostredkovania finančnou inštitúciou. Digitálne podpisy síce tvoria časť riešenia, avšak hlavné výhody sa vytrácajú, ak je na zabránenie dvojitému míňaniu (double-spending) stále potrebná dôveryhodná tretia strana.
Navrhujeme riešenie problému dvojitého míňania prostredníctvom peer-to-peer siete. Sieť časovo označuje transakcie tak, že ich zahashuje do prebiehajúceho reťazca dôkazu práce (proof-of-work) založeného na hashi, čím vytvára záznam, ktorý nie je možné zmeniť bez opätovného vykonania dôkazu práce. Najdlhší reťazec neslúži len ako dôkaz chronologického poradia zaznamenaných udalostí, ale aj ako dôkaz, že pochádza z najväčšieho poolu výpočtového výkonu. Pokiaľ väčšinu výpočtového výkonu kontrolujú uzly, ktoré nespolupracujú pri útoku na sieť, vytvoria najdlhší reťazec a predbehnú útočníkov. Sieť sama osebe vyžaduje len minimálnu štruktúru. Správy sú šírené na princípe najlepšieho úsilia (princíp best effort) a uzly môžu kedykoľvek sieť opustiť a znova sa k nej pripojiť, pričom ako dôkaz o tom, čo sa počas ich neprítomnosti stalo, prijmú najdlhší reťazec dôkazu práce.

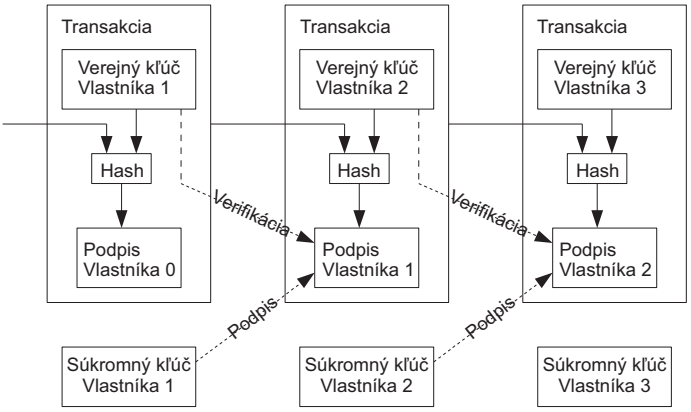
1. Úvod

Obchodovanie na internete sa takmer výlučne spolieha na finančné inštitúcie, ktoré slúžia ako dôveryhodné tretie strany pri spracovaní elektronických platieb. Hoci tento systém funguje dostatočne dobre pre väčšinu transakcií, stále trpí vrodenými slabinami modelu založeného na dôvere. Úplne nevratné transakcie nie sú skutočne možné, pretože finančné inštitúcie sa nemôžu vyhnúť sprostredkovaniu sporov. Náklady na sprostredkovanie zvyšujú celkové transakčné náklady, čím obmedzujú minimálnu praktickú veľkosť transakcie a znemožňujú malé príležitostné transakcie. Navyše existuje širší problém v podobe straty možnosti realizovať nevratné platby za nevratné služby. Keďže existuje možnosť zvrátenia transakcií, potreba dôvery sa rozširuje. Predajcovia musia byť opatrní voči svojim zákazníkom a vyžadovať od nich viac informácií, než by inak potrebovali. Určité percento podvodov je považované za nevyhnutné. Týmto nákladom a neistotám pri platbách sa dá vyhnúť pri osobných transakciách s použitím fyzickej meny, avšak neexistuje žiadny mechanizmus, ktorý by umožňoval platby cez komunikačný kanál bez dôveryhodnej tretej strany.

To, čo je potrebné, je elektronický platobný systém založený na kryptografickom dôkaze namiesto dôvery, ktorý umožní dvom stranám priamo obchodovať bez potreby dôveryhodnej tretej strany. Transakcie, ktoré by boli výpočtovo neuskutočniteľné na zvrátenie, by chránili predajcov pred podvodmi, pričom bežné mechanizmy úschovy by mohli jednoducho chrániť kupujúcich. V tomto dokumente navrhujeme riešenie problému dvojitého míňania pomocou distribuovaného časového servera fungujúceho na princípe peer-to-peer, ktorý generuje výpočtový dôkaz o chronologickom poradí transakcií. Systém zostáva bezpečný, pokiaľ čestné uzly spoločne ovládajú väčší výpočtový výkon ako akákoľvek koordinovaná skupina útočníkov.

2. Transakcie

Elektronickú mincu definujeme ako reťazec digitálnych podpisov. Každý vlastník prevádza mincu na ďalšieho tým, že digitálne podpíše hash predchádzajúcej transakcie a verejný kľúč ďalšieho vlastníka, pričom tieto údaje pridá na koniec mince. Príjemca môže overením podpisov skontrolovať reťazec vlastníctva.



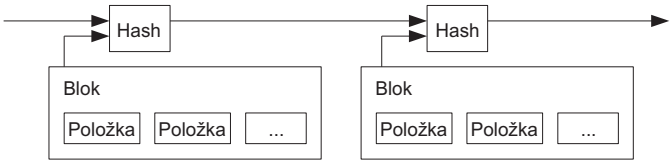
Problém samozrejme spočíva v tom, že príjemca nemôže overiť, či niektorý z predchádzajúcich vlastníkov mincu neutratil dvakrát. Bežným riešením je zavedenie dôveryhodnej centrálnej autority, alebo „mincovne“, ktorá kontroluje každú transakciu, aby zabránila dvojitému míňaniu. Po každej transakcii sa musí minca vrátiť do mincovne, ktorá vydá novú mincu, pričom len mince vydané priamo mincovňou sú považované za dôveryhodné a nezneužité. Problém tohto riešenia spočíva v tom, že osud celého peňažného systému závisí od spoločnosti, ktorá prevádzkuje mincovňu, pričom každá transakcia musí prechádzať cez ňu – podobne ako v prípade bánk.

Potrebujeme spôsob, aby príjemca vedel, že predchádzajúci vlastníci nepodpísali žiadnu skoršiu transakciu. Pre naše účely sa berie do úvahy najskoršia transakcia, takže neskoršie pokusy o dvojité míňanie nás nezaujímajú. Jediný spôsob, ako potvrdiť neexistenciu transakcie, je mať prehľad o všetkých transakciách. V modeli založenom na mincovni mala mincovňa prehľad o všetkých transakciách a rozhodovala, ktorá prišla ako prvá. Aby sme to dosiahli bez dôveryhodnej strany, transakcie musia byť oznámené verejne [1] a potrebujeme systém, v ktorom sa účastníci dohodnú na jedinečnej histórii poradia prijatých transakcií. Príjemca potrebuje dôkaz, že v čase každej transakcie väčšina uzlov súhlasila s tým, že bola prijatá ako prvá.

3. Server Časových razítok (timestamp server)

Riešenie, ktoré navrhujeme, začína časovým serverom. Časový server funguje tak, že vytvorí hash bloku položiek, ktoré majú byť časovo označené, a tento hash následne verejne zverejní, napríklad v novinách alebo na Usenet fóre[2-5].

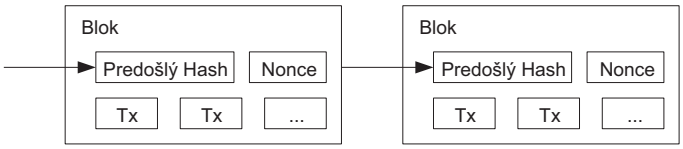
Časové razítko dokazuje, že údaje museli existovať v danom čase, aby sa mohli stať súčasťou hashu. Každé časové razítko obsahuje hash predchádzajúceho razítka čím sa vytvára reťazec, pričom každé ďalšie časové razítko posilňuje platnosť tých predchádzajúcich. Tento mechanizmus zabezpečuje nemennosť údajov a umožňuje dôveryhodné určenie poradia transakcií bez potreby centrálnej autority.



4. Dôkaz o práci (Proof of Work)

Na implementáciu distribuovaného časového servera na báze peer-to-peer použijeme systém proof-of-work, podobný Hashcashu od Adama Backa [6], namiesto publikovania v novinách alebo na Usenet fórach. Proof-of-work spočíva v hľadaní hodnoty, ktorá pri zahashovaní (napríklad pomocou SHA-256) vytvorí hash začínajúci určitým počtom úvodných núl. Priemerné množstvo výpočtovej práce potrebné na nájdenie takejto hodnoty rastie exponenciálne s počtom požadovaných úvodných núl, pričom správnosť riešenia sa dá jednoducho overiť jedným výpočtom hashu.

V našej časovej sieti implementujeme proof-of-work inkrementáciou nonce v bloku dovtedy, kým nenájdeme hodnotu, ktorá spôsobí, že hash bloku spĺňa požadovaný počet úvodných núl. Keď je výpočtová práca vynaložená na splnenie proof-of-work, blok už nemožno zmeniť bez opätovného vykonania tejto práce. Keďže neskoršie bloky sú reťazovo naviazané na ten predchádzajúci, ak by niekto chcel upraviť starší blok, musel by prepočítať všetky nasledujúce bloky, čo by bolo extrémne náročné.



Proof-of-work zároveň rieši problém určovania reprezentácie pri rozhodovaní väčšiny. Ak by väčšina bola založená na princípe jedna IP adresa = jeden hlas, mohol by systém ľahko zmanipulovať ktokoľvek, kto by si vedel prideliť veľké množstvo IP adries. Proof-of-work funguje ako „jeden CPU = jeden hlas“. Väčšinové rozhodnutie sa reprezentuje najdlhším reťazcom, ktorý obsahuje najviac investovanej výpočtovej práce. Ak väčšinu výpočtového výkonu ovládajú čestné uzly, čestný reťazec bude rásť najrýchlejšie a prekoná akékoľvek konkurenčné reťazce. Ak by chcel útočník upraviť minulý blok, musel by znovu vypočítať proof-of-work pre tento blok a aj pre všetky nasledujúce bloky, a zároveň dobehnúť a predbehnúť prácu vykonanú čestnými uzlami. Neskôr ukážeme, že pravdepodobnosť, že pomalší útočník dokáže dobehnúť čestný reťazec, exponenciálne klesá s pribúdajúcimi blokmi.

Aby sa prispôsobil rastu výpočtového výkonu hardvéru a zmenám v záujme o prevádzkovanie uzlov, obtiažnosť proof-of-work sa dynamicky upravuje na základe kĺzavého priemeru, pričom cieľom je udržať stabilný priemerný počet blokov za hodinu. Ak sa bloky generujú príliš rýchlo, obtiažnosť sa zvýši.

5. Sieť

Kroky na prevádzku siete sú nasledovné:

1) Nové transakcie sa vysielajú do všetkých uzlov.
2) Každý uzol zhromažďuje nové transakcie do bloku.
3) Každý uzol pracuje na nájdení proof-of-work pre svoj blok.
4) Keď uzol nájde proof-of-work, vysiela blok do všetkých uzlov.
5) Uzly akceptujú blok, iba ak sú všetky transakcie v ňom platné a neboli už minuté.
6) Uzly vyjadrujú akceptovanie bloku tým, že začnú pracovať na vytvorení ďalšieho bloku v reťazci, pričom použijú hash prijatého bloku ako predchádzajúci hash.

Uzly vždy považujú najdlhší reťazec za správny a budú na ňom ďalej pracovať. Ak dva uzly súčasne vysielajú rôzne verzie nasledujúceho bloku, niektoré uzly môžu prijať jednu verziu skôr než druhú. V takom prípade uzly pracujú na tej verzii, ktorú dostali ako prvú, ale zároveň uložia druhú vetvu v prípade, že by sa neskôr predĺžila. Tento konflikt sa vyrieši, keď sa nájde ďalší proof-of-work a jedna vetva sa stane dlhšou. V tom momente uzly, ktoré pracovali na kratšej vetve, presunú svoju prácu na dlhšiu vetvu, pretože práve tá sa považuje za správnu.

Nové transakcie nemusia nevyhnutne dosiahnuť všetky uzly. Stačí, ak sa dostanú k dostatočne veľkému počtu uzlov, pretože skôr či neskôr budú zahrnuté do bloku. Prenos blokov je takisto odolný voči stratám správ. Ak nejaký uzol neprijme blok, automaticky si ho vyžiada v momente, keď dostane nasledujúci blok a uvedomí si, že mu predchádzajúci chýba.

6. Motivácia

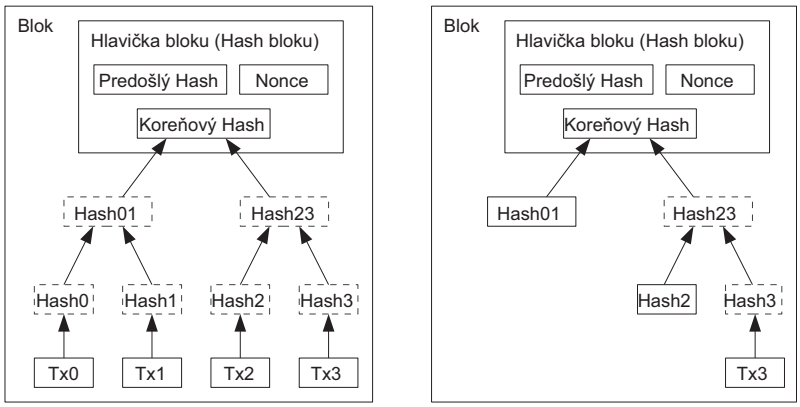
Podľa konvencie je prvou transakciou v každom bloku špeciálna transakcia, ktorá vytvára novú mincu vlastnenú tvorcom bloku. Táto mechanika poskytuje motiváciu pre uzly podporovať sieť a spôsob distribúcie nových mincí do obehu, keďže neexistuje žiadna centrálna autorita na ich vydávanie. Postupné pridávanie pevného množstva nových mincí je analogické ťažbe zlata, kde baníci vynakladajú zdroje na pridávanie zlata do obehu. V našom prípade sa týmto zdrojom stáva výpočtový výkon a elektrina.

Okrem nových mincí môže byť motivácia financovaná aj transakčnými poplatkami. Ak je výstupná hodnota transakcie nižšia ako vstupná, rozdiel predstavuje transakčný poplatok, ktorý sa pridáva k odmenám za vyťažený blok. Keď do obehu vstúpi predurčený počet mincí, systém sa môže postupne prepnúť na model založený čisto na transakčných poplatkoch, čím sa odstráni inflačný efekt nových mincí.

Táto motivácia zároveň pomáha udržať uzly čestné. Ak by chamtivý útočník získal viac výpočtového výkonu než všetky čestné uzly dohromady, musel by sa rozhodnúť medzi zneužitím svojej sily a pokúsil by sa podviesť ľudí (napr. vrátiť si už minuté platby), alebo hraním podľa pravidiel a ťažením nových mincí, pričom by získaval viac odmien ako ktokoľvek iný. Pre útočníka je výhodnejšie zostať čestný, pretože tým maximalizuje vlastný zisk, než aby poškodil systém a tým aj hodnotu vlastného majetku.

7. Uvoľnenie miesta na disku

Keď je najnovšia transakcia danej mince dostatočne hlboko zakopaná pod ďalšími blokmi, staršie transakcie, ktoré boli už minuté, môžu byť odstránené, aby sa šetrilo miesto na disku. Aby bolo možné toto dosiahnuť bez narušenia hashu bloku, transakcie sú organizované v Merkleovom strome (Merkle Tree) [7][2][5], pričom iba koreňový hash stromu je zahrnutý do hashu bloku. Vďaka tomu môžu byť staré bloky komprimované tak, že sa odstránia nepotrebné vetvy stromu, pričom stačí uchovať len koreňové hashe. Vnútorné hashe nemusia byť uložené, čím sa výrazne šetrí úložný priestor.



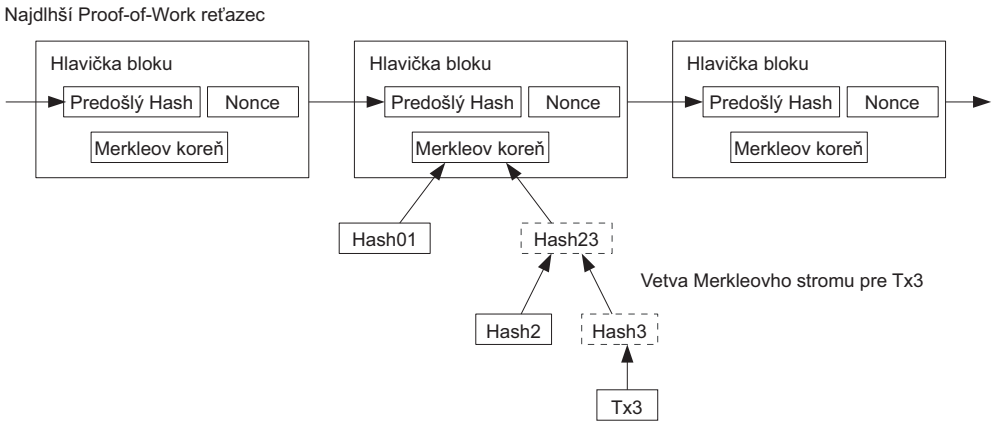
Transakcie hashované v Merkleovom strome

Po odstránení (pruning) transakcií Tx0-2 z bloku.

Hlavička bloku bez transakcií by mala približne 80 bajtov. Ak predpokladáme, že bloky sa generujú každých 10 minút, výpočet je nasledovný: 80 bajtov * 6 * 24 * 365 = 4,2 MB za rok. Keďže v roku 2008 sa bežne predávali počítačové systémy s 2 GB RAM, a podľa Mooreovho zákona sa očakával rast o 1,2 GB ročne, ukladanie blokových hlavičiek v pamäti by nemalo byť problémom.

8. Zjednodušené overovanie platieb

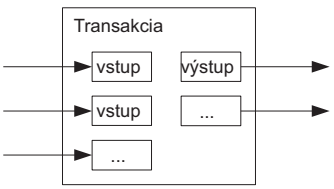
Platby je možné overovať bez spustenia plnohodnotného uzla siete. Používateľ si musí uchovávať iba hlavičky blokov z najdlhšieho reťazca proof-of-work, ktoré získa dotazovaním sieťových uzlov, kým si nie je istý, že má najdlhší reťazec. Následne získa Merkleovu vetvu, ktorá spája transakciu s blokom, v ktorom bola časovo označená. Používateľ nemôže sám overiť transakciu, ale tým, že je prepojená s konkrétnym blokom v reťazci, vidí, že ju sieťový uzol prijal. Bloky pridané po nej navyše potvrdzujú, že ju sieť akceptovala.



Overovanie je spoľahlivé, pokiaľ čestné uzly kontrolujú sieť, no je zraniteľnejšie, ak sieť ovládne útočník. Zatiaľ čo sieťové uzly si môžu transakcie overovať samy, zjednodušená metóda môže byť oklamaná falošnými transakciami, pokiaľ má útočník dostatočný výpočtový výkon na prekonanie siete. Jednou zo stratégií ochrany je umožniť prijímanie varovaní od sieťových uzlov, keď zistia neplatný blok. To by podnietilo softvér používateľa k stiahnutiu celého bloku a upozornených transakcií na potvrdenie nezrovnalosti. Podniky, ktoré prijímajú časté platby, budú pravdepodobne chcieť prevádzkovať vlastné uzly pre vyššiu bezpečnosť a rýchlejšiu verifikáciu.

9. Kombinovanie a rozdeľovanie hodnoty

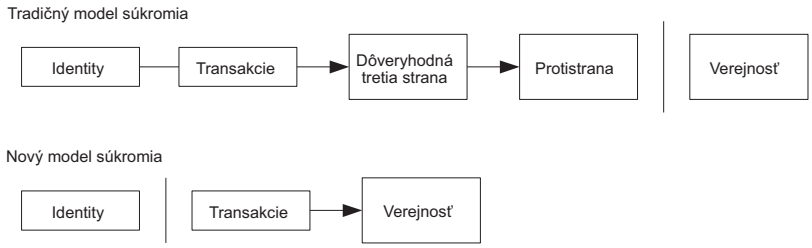
Hoci by bolo možné spracovávať mince jednotlivo, bolo by nepraktické vytvárať samostatnú transakciu pre každý cent v platbe. Aby bolo možné hodnotu kombinovať a rozdeľovať, transakcie obsahujú viacero vstupov a výstupov. Zvyčajne existuje jeden vstup z predchádzajúcej transakcie s vyššou hodnotou, alebo viacero vstupov, ktoré kombinujú menšie sumy. Vo väčšine prípadov sú maximálne dva výstupy: Jeden pre platbu príjemcovi, Druhý na vrátenie zvyšnej sumy odosielateľovi, ak nejaká ostane.



Je potrebné poznamenať, že rozvetvenie, pri ktorom transakcia závisí od viacerých predchádzajúcich transakcií a tie zase od ďalších, tu nepredstavuje problém. Nikdy nie je potrebné extrahovať kompletnú samostatnú kópiu histórie transakcie.

10. Súkromie

Tradičný bankový model dosahuje istú úroveň súkromia tým, že obmedzuje prístup k informáciám iba na zúčastnené strany a dôveryhodnú tretiu stranu. Nutnosť oznamovať všetky transakcie verejne vylučuje túto metódu, ale súkromie môže byť stále zachované tým, že sa preruší tok informácií na inom mieste: tým, že sa udržujú verejné kľúče anonymné. Verejnosť môže vidieť, že niekto posiela istú sumu niekomu inému, ale bez informácií, ktoré by transakciu spájali s konkrétnou osobou. Je to podobné úrovni informácií, ktoré uverejňujú burzy, kde sa čas a veľkosť jednotlivých obchodov, tzv. "tape", uverejňuje, ale bez uvedenia toho, kto boli strany.



Ako dodatočná ochrana by mal byť použitý nový pár kľúčov pre každú transakciu, aby sa zabránilo ich spájaniu s jediným vlastníkom. Niektoré prepojenia sú však stále neodvratné pri transakciách s viacerými vstupmi, ktoré nevyhnutne odhalia, že ich vstupmi boli vlastníctva tej istej osoby. Rizikom je, že ak je odhalený vlastník kľúča, prepojenie môže odhaliť aj iné transakcie, ktoré patrili tomuto vlastníkovi.

11. Výpočty

Zvažujeme scenár, v ktorom sa útočník pokúša vygenerovať alternatívny reťazec rýchlejšie ako čestný reťazec. Aj keby sa mu to podarilo, nemohol by svojvoľne meniť systém, napríklad vytvárať nové mince „z ničoho“, či privlastniť si prostriedky, ktoré mu nikdy nepatrili. Uzly neprijmú neplatnú transakciu ako platbu a čestné uzly nikdy neakceptujú blok, ktorý ju obsahuje. Jediné, čo útočník môže skúsiť, je zmeniť jednu zo svojich vlastných transakcií, aby si vrátil už minuté prostriedky.

Súboj medzi čestným reťazcom a útočníkovým reťazcom možno opísať ako binomickú náhodnú prechádzku (Binomial Random Walk). Úspešná udalosť: Čestný reťazec sa predĺži o jeden blok (+1 náskok). Neúspešná udalosť: Útočníkov reťazec sa predĺži o jeden blok (-1 náskok).

Pravdepodobnosť, že útočník dobehne čestný reťazec, je podobná problému zruinovania hráča (Gambler's Ruin problem). Predstavme si hráča s neobmedzeným kreditom, ktorý začína v strate a môže hrať nekonečný počet hier, aby sa pokúsil vyrovnať svoju bilanciu. Pravdepodobnosť, že sa mu to niekedy podarí – teda že útočník dobehne čestný reťazec – môžeme vypočítať podľa vzorca [8]:

p = pravdepodobnosť, že čestný uzol nájde ďalší blok
q = pravdepodobnosť, že útočník nájde ďalší blok
q_z = pravdepodobnosť, že útočník niekedy dobehne čestný reťazec zo z blokov pozadu

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

Za predpokladu, že $p > q$, pravdepodobnosť dobehnutia čestného reťazca klesá exponenciálne s počtom blokov, ktoré musí útočník dohnať. Keďže šance sú proti nemu, ak sa mu nepodarí čoskoro dosiahnuť šťastný náskok, jeho pravdepodobnosť úspechu sa s narastajúcim odstupom stáva takmer nulovou.

Teraz zvažujeme, ako dlho musí príjemca novej transakcie čakať, aby si mohol byť dostatočne istý, že odosielateľ ju už nemôže zmeniť. Predpokladáme, že odosielateľ je útočník, ktorý chce nechať príjemcu chvíľu veriť, že platba prebehla, a potom ju zmeniť tak, aby si ju vrátil späť. Príjemca síce dostane upozornenie, keď sa tak stane, ale útočník dúfa, že to už bude príliš neskoro.

Príjemca pred transakciou vygeneruje nový pár kľúčov a odosielateľovi poskytne len verejný kľúč. Tým zabráni útočníkovi predpripraviť si vlastný reťazec blokov vopred tým, že by na ňom tajne pracoval, kým sa mu nepodarí dosiahnuť dostatočný náskok, a až potom vykonať podvodnú transakciu. Keď je transakcia odoslaná, nečestný odosielateľ začne v tajnosti pracovať na paralelnom reťazci, ktorý obsahuje alternatívnu verziu jeho transakcie.

Príjemca čaká, kým transakcia nebude pridaná do bloku a na ňu sa nenaviaže ďalších z blokov. Nevie presne, ako ďaleko sa útočník dostal, ale ak predpokladáme, že čestné bloky sa ťažia priemerným očakávaným časom na blok, potenciálny pokrok útočníka bude nasledovať Poissonovo rozdelenie s očakávanou hodnotou:

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

Aby sme získali pravdepodobnosť, že útočník ešte stále môže dohnať čestný reťazec, vynásobíme hustotu Poissonovho rozdelenia pre každý možný pokrok, ktorý mohol dosiahnuť, pravdepodobnosťou, že by z tohto bodu mohol dohnať reťazec:

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

Aby sme sa vyhli súčtu nekonečného radu rozdelení, zapíšeme rovnicu takto:

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

Konverzia do jazyka 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;
}
```

Vykonaním niekoľkých výpočtov zistíme, že pravdepodobnosť exponenciálne klesá s hodnotou *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
```

Riešenie pre P menšie ako 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. Záver

Navrhli sme systém pre elektronické transakcie bez potreby dôvery. Začali sme tradičným rámcom mincí vytvorených z digitálnych podpisov, ktorý poskytuje silnú kontrolu vlastníctva, ale je nekompletný bez mechanizmu na zabránenie dvojitému míňaniu. Na vyriešenie tohto problému sme navrhli peer-to-peer sieť využívajúcu proof-of-work na zaznamenanie verejnej histórie transakcií, ktorá sa rýchlo stáva výpočtovo neuskutočniteľnou na zmenu, pokiaľ väčšinu výpočtového výkonu kontrolujú čestné uzly. Sieť je robustná vo svojej neštruktúrovanej jednoduchosti. Uzly pracujú súčasne s minimálnou koordináciou. Nemusia byť identifikované, keďže správy nie sú smerované na konkrétne miesto a stačí, aby boli doručené najlepším možným spôsobom. Uzly môžu sieť kedykoľvek opustiť a znovu sa pripojiť, pričom ako dôkaz o histórii transakcií prijmú existujúci reťazec proof-of-work. Hlasujú svojím výpočtovým výkonom, čím prijímajú platné bloky tým, že na nich ďalej pracujú, a odmietajú neplatné bloky tým, že na nich nepracujú. Všetky potrebné pravidlá a motivácie môžu byť vynútené týmto konsenzuálnym mechanizmom.

Referencie

[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 omputer 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.