Bitcoin: Systém peer-to-peer elektronické hotovosti

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

Abstrakt. Peer-to-peer systém elektronické hotovosti umožňuje online platby přímo mezi lidmi bez účasti finančních institucí. Tento problém lze částečně řešit pomocí digitálních podpisů. Výhody tohoto schématu jsou však ztraceny, když k autorizaci transakcí a ochraně před falšováním elektronických peněz je potřeba důvěryhodná finanční autorita. V tomto dokumentu navrhujeme řešení problému s nažádoucím falšováním peněz v systému peer-to-peer. Síť značkuje transakce časovými známkami tím, že je hashuje do do stále rostoucího řetězce, který reprezentuje důkaz-o-práci. Vytváří tím záznam, který nemůže být dále změněn aniž by nebylo potřeba celý důkaz-o-práci přepočítat. Nejdelší řetězec tak nejenom slouží jako důkaz časové posloupnosti finančních transakcí, ale také jako důkaz, že pocházel z největšího zdroje CPU výpočetní síly, který se na těchto událostech společně shodl. Pokud tento většinový zdroj CPU výpočetní síly neprovádí koordinovaný útok na síť, vygeneruje vždy nejdelší řetězec, čímž vytěsní menšinové útočníky. Samotná síť vyžaduje minimální infrastrukturu. Zprávy jsou rozseílány na principu jak-nejlépe-to-jde a jednotlivé uzly se mohou odpojit a připojit dle libosti, akceptujíce vždy nejdelší aktuální řetězec jako zdroj informací o událostech z doby, kdy byly pryč.

1. Úvod

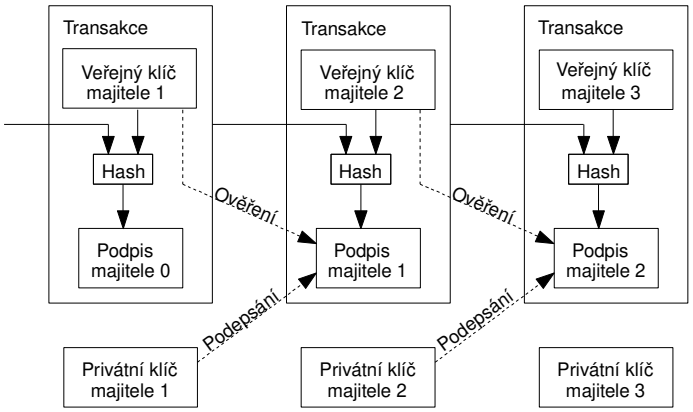
Internetový trh se stal zcela závislý na finančních institucích, které představují důvěryhodné prostředníky pro zprostředkování elektronických plateb. Systém funguje dobře pro většinu transakcí, je však ze své podstaty závislý na vzájemné důvěře mezi interagujícími stranami. Zcela nevratné transakce nejsou možné, neboť finanční instituce se v případě neshod mezi zúčastněnými stranami nemohou zcela vyčlenit z transakčního řetězce, jehož jsou součástí. Náklady na zprostředkování transakce omezují praktičnost malých ojedinělých transakcí a může vést ke ztrátám, pokud byla stornována platba za nevratnou službu. S teoretickou možností zrušení proběhlé platby vzniká potřeba pro důvěryhodné vztahy. Obchodníci potřebují znát identitu svých zákazníků a mohou požadovat více osobních informací než je potřeba pro dokončení samotného obchodu. Určité procento ztrát a podvodů je nevyhnutelné a nutné je akceptovat. Těmto nákladům a nejistotám lze předcházet použitím fyzických peněz, ale zatím neexistuje žádný mechanismus pro provedení platby prostřednictvím veřejných komunikačních kanálů bez účasti důvěryhodného zprostředkovatele.

Co je potřeba je elektronický platební systém založený na kryptografických důkazech místo důvěry, umožňující tak jakýmkoliv dvěma stranám interagovat přímo mezi sebou bez potřeby důvěryhodné třetí strany. Transakce, které jsou z výpočetního hlediska nevratné by ochránily prodávající před krádeží a převodní úschovny by ochránily kupující. V tomto dokumentu představujeme řešení k problému s dvojím utrácením použitím sítě peer-to-peer jak distribuovaný časově značkovací server generující výpočetně náročný důkaz potvrzující chronologičnost

transakcí. Systém je bezpečný pokud čestné síťové uzly kolektivně ovládají více CPU síly než jakákoliv kooperující skupina záškodných síťových uzlů.

2. Transakce

Elektronickou minci definujeme jako řetězec digitálních podpisů. Každý majitel převede minci dalšímu tím, že digitálně podepíše hash předchozí transakce, veřejný klíč následujícího držitele a tuto zprávu připojí na konec této elektronické mince. Příjemce může ověřit podpisy, aby ověřil řetězec jejího vlastnictví.

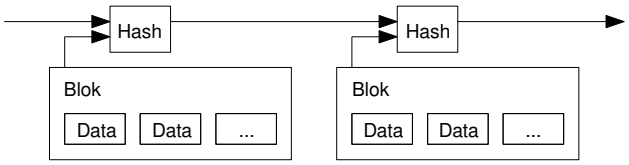


Problém je však, že příjemce nemůže ověřit, že některý z předchozích vlastníků nepoužil minci dvakrát. Toto se běžně řeší zavedením centrální důvěryhodné autority či mincovny, která každou transakci ověří a schválí pokud se jedná o první použití. Po každé transakci by se mince vrátila do mincovny a byla vyměněna za novou, přičemž pouze mince vydané touto mincovnou by byly důvěryhodné a autentické. Problém tohoto řešení je, že osud celého systému je závislý na společnosti řídící mincovnu, neboť každá transakce musí jít skrze ní – tak jako u banky.

Potřebujeme najít způsob, jak si příjemce může být jistý, že předchozí majitel nepodepsal jinou transakci již dříve. Pro naše účely uznáváme jako platnou transakci tu, která nastane jako první a o pozdější pokusy o dvojí utracení se nestaráme. Jediný způsob jak si být jistý, že nějaká transakce neproběhla, je mít záznam všech transakcí. V modelu centrální mincovny to byla ona, kdo měl seznam všech transakcí a rozhodoval která byla dřív. Aby stejného efektu bylo dosaženo bez důvěryhodné třetí strany, transakce musí být zveřejněny [1] a musí být zaveden sytstém zajišťující shodu mezi účastníky na společné historii transakcí. Příjemce potřebuje důkaz, že každá příchozí transakce bude většinovou skupinou uzlů rozeznaná jako první transakce dané mince.

3. Časově-značkovací server

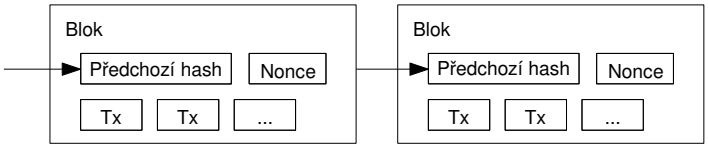
Řešení, které navrhujeme začíná časově-značkovacím serverem. Server funguje tak, že vezme kryptografickou hash bloku značených dat a tu zveřejní – např. v novinách nebo Usenetovém příspěvku [2-5]. Časová známka dokazuje, že data musela očividně v danou dobu existovat, aby mohla být součástí veřejné hashe. Každá časová známka obsahuje předchozí časovou známku ve své hashi, čímž formuje řetězec, kde každá další iterace posílí značky předchozí.



4. Důkaz-o-práci

Aby časově-značkovací server fungoval distribuovaně na bázi sítě peer-to-peer, použijeme namísto příspěvků do novin či Usenetu systém důkazu-o-práci podobný tomu, který Adam Back vytvořil pro Hashcash [6]. Důkaz-o-práci zahrnuje proces hledání hodnoty, na kterou je-li aplikována kryptografická hash funkce, jako například SHA-256, výsledný hashe začne určitým počten nulových bitů. Potřebné množství práce pro nalezení takové hashe roste exponenciálně s počtem vyžadovaných nul přičemž správnost řešení lze ověřit vyhodnocením jediné hash funkce.

V naší časově-značkovací síti implementujeme důkaz-o-práci tím, že budeme inkrementovat číslo nonce, jež je součástí bloku, dokud nalezená hodnota hashe celého bloku nebude začínat požadovaným počtem nul. Jakmile je číslo nonce nalezeno po vynaložení určitého CPU výpočetního výkonu, blok nemůže být změněn aniž by bylo potřeba číslo nonce nalézt znovu. Tím, že bloky jsou dále zřetězeny, změna jednoho bloku v minulosti vyžaduje přepočítání hodnot všech následujících bloků.



Důkaz-o-práci také řeší problém ve většinovém rozhodování, kde není jasné kdo reprezentuje jakou identitu. Kdyby se hlasovalo systémem jedna-IP-adresa-jeden-hlas, systém by mohl být napadnut kýmkoliv kdo by byl schopen alokovat velké množství IP adres. Důkaz-o-práci je v podstatě jedno-CPU-jeden-hlas a většinové rozhodnutí je reprezentováno jako nejdelší řetězec s nejvyšší naakumulovanou prací. Pokud je většina CPU výpočetního výkonu v kontrole čestných síťových uzlů, čestně vytvořený řetězec bude akumulovat vykonanou práci nejrychleji a předežene konkurenční řetězce. Ke změně bloku v minulosti musí útočník znovu vytvořit důkaz-o-práci daného bloku, poté všech následujících a nakonec dohnat a překonat množství práce v čestném řetězci, který byl mezitím vytvářen čestnými uzly. Později ukážeme, že pravděpodobnost slabšího útočníka předehnat zbytek sítě klesá exponenciálně s množstvím nově přidaných bloků.

Jelikož se množství uzlů v síti mění a rychlost hardwaru také, obtížnost důkazu-o-práci bude určena pomocí klouzavého průměr dob nalezení posledních několika bloků, nastavujícího očekávané množství bloků za hodinu. Pokud jsou bloky generovány příliš rychle, obtížnost vzroste.

5. Síť

Síť funguje podle následujících kroků:

1) Nové transakce jsou rozesílány všem síťovým uzlům
2) Každý uzel shromažďuje transakce do bloku.
3) Každý uzel pracuje na nalezení obtížného důkazu-o-práci pro svůj blok.
4) Pokud uzel nalezne důkaz-o-práci, rozešle blok ostatním uzlům.
5) Ostatní uzly nový blok přijmou pouze pokud všechny obsažené transakce jsou platné a neobsahuje již použitou minci.
6) Uzly vyjádří přijetí bloku tím, že začnou pracovat na novém bloku a použijí hash přijatého bloku jako referenci na předchozí blok.

Uzly vždy považují nejdelší řetězec za správný a budou pokračovat na rozšiřování. Pokud dva uzly rozešlou nový blok současně, dostanou se k různým uzlům jako první různé verze bloků. Tento konflikt rozhodne až důkaz-o-práci dalšího bloku, který jednu alternativu řetězce prodlouží o jeden, čímž ho ostatní uzly akceptují jako delší.

Nově rozesílané transakce nemusí být nutně doručeny všem síťovým uzlům. Pokud jsou doručeny dostatečně mnoha uzlům, budou zařazeny do bloku relativně brzy. Přijímání nových bloků také toleruje výpadky. Pokud uzel nezíská nový blok, může si jej vyžádat po obdržení následujícího, kdy zjistí, že jeden chybí.

6. Odměny

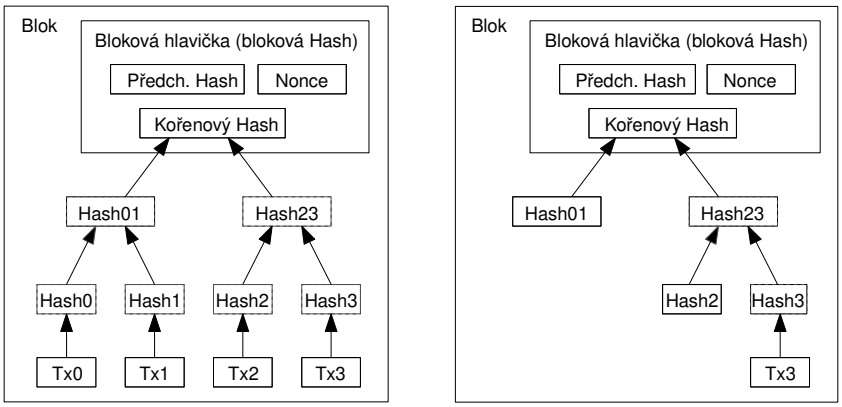
Zavedeme pravidlo, že první transakce v bloku vždy vytvoří novou minci kterou autor bloku může přidělit sobě. Tento mechanizmus motivuje provozovatele uzlů, aby aktivně podporovali síť, a také zajišťuje mechanismus počáteční distribuce nových mincí do oběhu. Není zde žádná centrální autorita, která by tuto úlohu zajistila. Stálý přítok nových mincí do oběhu je analogický k těžbě zlata, která využívá fyzických prostředků pro získání a zavedení nového zlata do oběhu. V našem případě jsou těmi prostředky CPU čas a elektřina.

Odměna za vytvoření bloku může být také tvořena transakčními poplatky. Pokud výstupní hodnota transakce je nižší než vstupní, rozdíl je transakční poplatek, který je přidán k blokové odměně bloku obsahujícího transakci. Až předem dané množství mincí vstoupí do oběhu, motivace pro provozovatele uzlů může být kompletně přeorientována na transakční poplatky, a měna bude zbavena veškeré inflace.

Bloková odměna motivuje uzly, aby zůstaly čestné. Pokud hamižný útočník nahromadí větší výpočetní výkon než zbytek čestných uzlů, může si zvolit, zda ho použije k útoku na síť tím, že zpětně bude krást své vlastní platby, nebo zda ho využije pro generování nových mincí. Měl by zjistit, že výhodnější je řídit se standardními pravidly, ktrá mu přinesou více nových mincí než všem ostatním dohromady, než stejnými prostředky útočit na síť jíž je také součástí.

7. Hospodaření s úložným prostorem

Transakce, které tvoří součást historie mince a jsou obsaženy ve starších blocích, mohou být smazány pro efektivnější využití úložné kapacity. Aby bylo nadále možné ověřit platnost blokové hashe a zachovat možnost plného ověření ostatních transakcí v bloku, jsou transakce hashovány v Merklově stromu [7][2][5], přičemž pouze kořen stromu je součástí hlavičky bloku. Všechny listy a větve stromu nemusí být nutně uloženy.



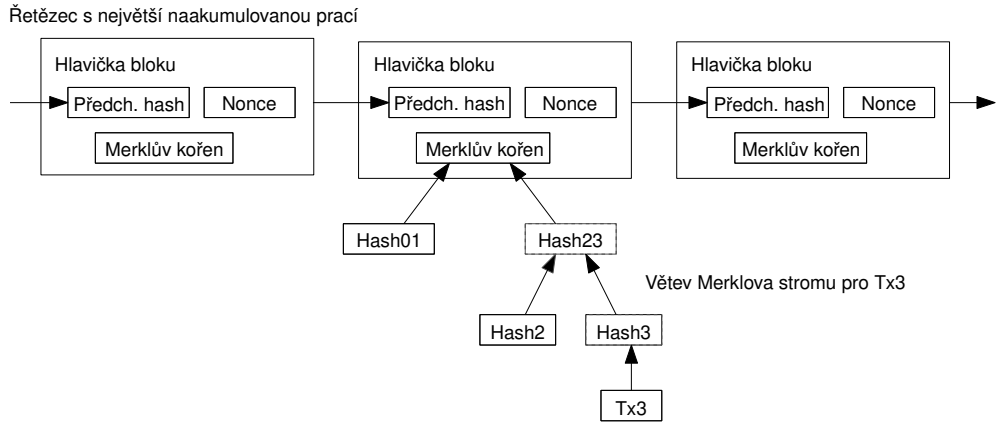
Transakce hashované v Merklově stromě

Po vyčištění bloku od starých transakcí

Hlavička bloku bez transakcí zabírá zhruba 80 Bytů. Pokud předpokládáme nový blok každých 10 minut, znamená to 80 Bytů * 6 * 24 * 365 = 4.2 MB ročně. S počítačovými systémy prodávanými v roce 2008 typicky s 2 GB RAM, Moorovým zákonem předpovídajícím současný růst 1.2 GB ročně, by s úložištěm neměl být problém i kdyby hlavičky bloků byly uloženy v operační paměti.

8. Zjednodušená kontrola plateb

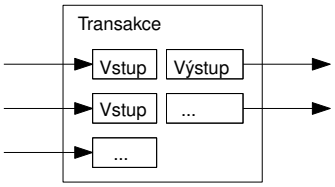
Ověřit platby je možné i bez provozu úplného uzlu. Uživatel pouze potřebuje kopii hlaviček nejdelšího řetězce s platným důkazem-o-práci, který může získat dotazováním ostatních uzlů v síti po dobu než se přesvědčí, že se skutečně jedná o nejdelší. Vyžádá si také větev Merklova stromu bloku, která dokazuje, že hash transakce je součástí blokové hashe. Sámotnou platnost transakce sice ověřit nelze, ale fakt, že transakce byla zařazena do bloku, jehož hlavička je součástí aktuálního řetězce hlaviček signalizuje, že jí síť přijala.



Kontrola jako taková je spolehlivá pokud je síť v rukou čestných uzlů. Zatímco plně validované uzly si mohou transakce ověřit samy, lehká kontrola může být manipulována uměle vytvořenými transakcemi po dobu kdy útočník přemůže zbytek sítě. Jedna z možností, jak tuto situaci řešit, je zavést systém varování, kterým čestné plné uzly upozorní lehký uzel v případě, že detekují v síti neplatný blok, a přimějí lehký uzel ke stažení všech pochybných transakcí pro jejich ověření. Podniky, které přijímají mnoho plateb však pravděpodobně budou chtít stále provozovat plně validovaný uzel, který nabízí nezávislost a vyšší rychlost ověření.

9. Spojování a rozdělování hodnot mincí

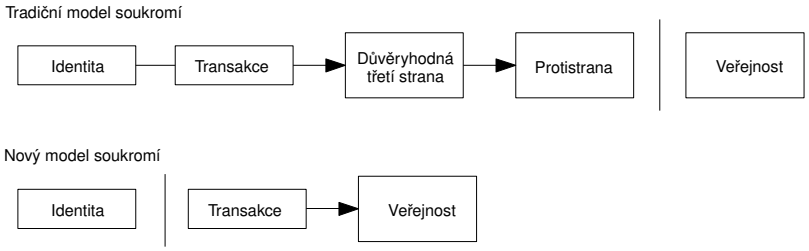
Zpracovávat mince po jedné je sice možné, je ale nepraktické převádět každou jednotku ve zvláštní transakci. Aby bylo možné hodnotu dělit či spojovat, transakce budou obsahovat vícero vstupů a výstupů. Typicky bude jeden vstup z větší předchozí transakce nebo vícero vstupů z menších transakcí a nanejvýš dva výstupy: jeden reprezentující platbu a druhý jako přebytek vracející se zpět k odesílateli.



Stojí za připomínku, že transakce, která závisí na několika dalších, které záleží na mnoha dalších, v tomto případě není problém. V žádném okamžiku není potřeba extrahovat kompletní samostatnou historii všech předchozích transakcí.

10. Soukromí

V modelu tradičního bankovnictví se určitá úroven soukromí zajišťuje omezením přístupu k soukromým údajům zúčastněných a důvěryhodných třetích stran. Potřeba zveřejňovat všechny transakce tento způsob znemožňuje. Soukromí je však stále možné do jisté míry zajistit, pokud veřejné spojitosti od transakcí až k fyzickým entitám přerušíme na úrovni veřejných klíčů, které zůstanou anonymní. Veřejně je vidět, že někdo posílá nějaké množství mincí někomu jinému, ale již nelze tuto transakci spojit s konkrétními lidmi. Je to podobný model jako na akciových trzích, kde jsou zveřejněny časy a objemy obchodů, ale ne jména nakupujících či prodávajících stran.



Pro dodatečnou ochranu by měl být použit nový klíč pro každou transakci, aby je nebylo možné spojit s jedním majitelem. Vzniku určitých spojitostí nelze zabránit u transakcí s vícero vstupy, které odhalí, že více vstupů kontroluje jediný majitel. Takto lze odhalit další transakce jedince, jehož identita jednou byla odhalena.

11. Výpočty

Uvažujme scénář, kde se útočník snaží generovat alternativní řetězec rychleji než čestný řetězec. I když tato situace nastane, neotevírá to systém libovolným útokům a změnám jako je vytváření nové hodnoty z ničeho nebo přivlastňování si cizích peněz. Síťové uzly nepřijmou neplatnou transakci jako platbu a čestné uzly nepřijmou nový blok, který takovou transakci obsahuje. Útočník pouze může zkusit změnit svou vlastní transakci a vrátit si peníze, které odeslal pryč.

Závod mezi čestným řetězcem a konkurenčním může být popsán jako binomiální náhodná procházka. Úspěšný jev je ten, kde čestný řetězec přidá platný blok k řetězci, čímž zvýší svůj náskok o 1. Neúspěšný jev je ten, kde úřočník prodlouží svůj řetězec o platný blok, čímž sníží svojí ztrátu o 1.

Pravděpodobnost, že útočník dožene určitou ztrátu je analogický Problému zruinovaného hráče. Předpokládejme hráče s neomezeným kreditem začínajícím s určitou zrátou. Snaží se ztrátu dohnat a má k dispozici neomezenou sekvenci pokusů. Můžeme spočítat pravděpodobnost, že se mu to podaří, v našem případě že se mu podaří dohnat čestný řetězec [8]:

p=pravděpodobnost, že čestný uzel najde další blok
q=pravděpodobnost, že útočník najde další blok
q_z=pravděpodobnost, že útočník dožene ztrátu z bloků

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

Předpokládáme-li, že $p > q$, tak pravděpodobnost, že útočník dožene ztrátu určitého množství bloků, klesá exponenciálně s velikostí ztráty. Pro útočníka je pak navíc během uplynulé doby čím dál těžší ztrátu dohnat, neboť jeho ztráta se statisticky stále zvětšuje.

Nyní rozvedeme jak dlouho musí příjemce platby počkat než může s dostatečnou jistotou vyloučit, že odesílající transakci nezmění. Předpokládáme, že odesílající je útočník, který se snaží přesvědčit příjemce, že platba proběhla, a po určité době se pokusí vrátit platbu zpět. To provede tak, že změní příjemce, doufaje, že v době kdy si toho příjemce všimne již bude pozdě.

Příjemce vygeneruje nový pár klíčů a sdělí veřejný klíč odesílateli krátce před podepsáním. To odesílateli zabrání, aby si předpřipravil řetězec bloků tím, že bude zkoušet dokud nebude mít štěstí, aby si vytvořil dostatečný náskok a vzápětí provedl platbu. Poté, co je transakce odeslána, útočník začne pracovat tajně na alternativním řetězci s jinou verzí odeslané transakce.

Příjemce čeká dokud transakce nebude přidána do platného bloku, který bude následován alespoň z dalšími bloky. Netuší, jak úspěšný útočník byl, ale předpokládv že zkonstruovat čestný řetězec zabralo průměrný čas vzniku bloku, útočníkův potenciální úspěch je řízen Poissonovým rozdělením se střední hodnotou:

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

Abychom získali pravděpodobnost, že útočník dokáže dohnat ztrátu, vynásobíme Poissonovu hustotu každého možného pokroku pravděpodobností, že by mohl ztrátu z daného pokroku dohnat.

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

Což, přeuspořádáno, aby bylo zabráněno sčítání nekonečných asymptotických hodnot distribuce, dává...

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

V jazyce C vyjádřeno...

```
#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;
}
```

Výsledky výpočtu pro určité hodnoty ukazují, že pravděpodobnost klesá exponenciálně se *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
```

Řešení pro P menší než 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. Závěr

Navrhli jsme systém pro elektronické transakce, který nevyžaduje vzájemnou důvěru interagujících stran. Začali jsme s obvyklým rámcem mince tvořené digitálními podpisy, které poskytují vysoký stupeň kontroly vlastnictví, který však není kompletní bez možnosti efektivně zabránit opakovanému utrácení mincí. K řešení tohoto problému jsme navrhli systém peer-to-peer sítě využívající důkaz-o-práci k vytváření veřejného záznamu transakcí, jež se stávají velmi brzy prakticky nezměnitelné je-li síť tvořena většinově čestnými uzly. Síť je robustní díky své nestrukturované jednoduchosti. Všechny uzly pracují naráz s velmi nízkou úrovní vzájemné koordinace. Nepotřebují být identifikovány, neboť zprávy nejsou směrovány do specifických míst a stačí je pouze do sítě rozesílat stylem jak-nejlépe-to-jde. Uzly mohou libovolně přicházet a odcházet, akceptujíce vždy řetězec důkazu-o-práci jako důkaz o sledu událostí z doby, kdy byly pryč. Pomocí svého CPU hlasují, přičemž souhlas s aktuálním řetězcem dávají tím, že pracují na rozšíření řetězce, nesouhlas s řetězcem tím, že začnou pracovat na rozšíření alternativního řetězce. Jakákoliv potřebná pravidla a motivace mohou být vynuceny tímto konsenzuálním mechanismem.

Reference

[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.