SATOSHI NAKAMOTO:
PEER-TO-PEER SYSTÉM
ELEKTRONICKÝCH PENĚZ

BITCOIN WHITEPAPER

www.bitcoin.org
www.obchodovni.com
info@obchodovani.com

Bitcoin: Peer-to-Peer systém elektronických peněz

Satoshi Nakamoto
www.bitcoin.org
www.obchodovani.com
info@obchodovani.com

Abstrakt

Čistě peer-to-peer verze elektronických peněz by umožnila přímé provádění online plateb mezi dvěma stranami, a to bez zprostředkování finanční institucí. Částečným řešením jsou digitální podpisy, ale jejich výhody jsou ztraceny, pokud je stále potřeba důvěryhodné třetí strany k zamezení dvojí útraty. Navrhujeme řešení problému dvojí útraty pomocí peer-to-peer sítě. Tato síť přidělí každé provedené transakci časové razítko a pomocí hashovacích funkcí ji přidá do neustále se aktualizujícího řetězce důkazů o vykonané práci (proof-of-work). Vznikne tak záznam, který nelze změnit bez opětovného provedení důkazů o této již vykonané práci. Nejdelší řetězec slouží nejen jako záznam posloupnosti zápisů a historie událostí, ale je zároveň důkazem, že je potvrzen většinou výpočetního výkonu sítě. Dokud je většina výpočetního výkonu kontrolována nezávislými uzly, které nepodnikají kooperované útoky na síť, tak právě tyto uzly zajistí nejdelší řetězec, a tím předčí případné útočníky. Síť jako taková přitom nevyžaduje speciální strukturu. Zprávy se šíří na principu nevynucování spolehlivosti všech uzlů (princip best-effort), proto se mohou jednotlivé uzly kdykoliv odpojit nebo připojit, přičemž po opětovném připojení akceptují nejdelší řetězec důkazů o vykonané práci jako záznam událostí, ke kterým došlo v jejich nepřítomnosti.

1. Úvod

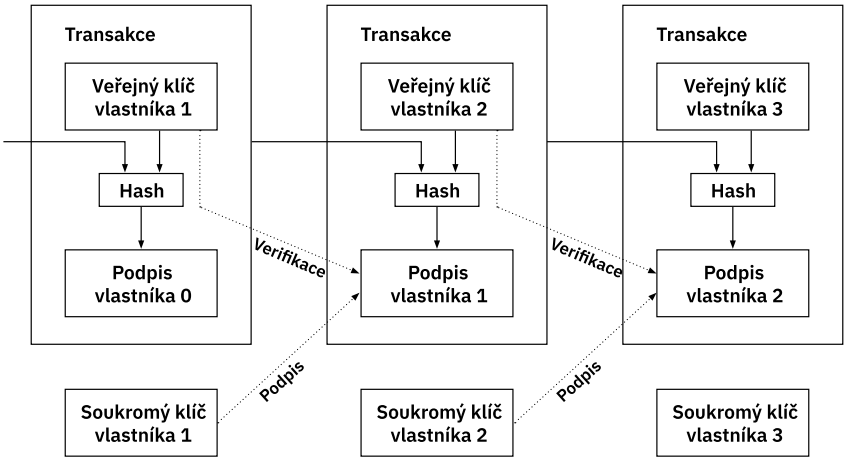
Pokud jde o zpracování elektronických plateb, obchodní styk na internetu dnes spoléhá téměř výhradně na finanční instituce jako na důvěryhodné třetí strany. Tento systém je pro většinu transakcí dostačující, přesto má ale pár slabých míst vyplývajících z modelu založeného na důvěře. Prakticky není možné provést zcela nevratnou transakci, protože k povinnostem finančních institucí patří i zprostředkování řešení případných sporů. Náklady na řešení sporů zvyšují cenu transakcí, čímž limitují minimální praktickou výši transakce a omezují provádění běžných drobných plateb. Je nutno vzít v úvahu i obecnější náklady v souvislosti s tím, že zatímco služby jsou často nevratné, transakce nikoliv. Kvůli tomu, že je možné transakci vrátit, respektive zrušit, vzniká potřeba důvěry. Obchodníci musí ke svým zákazníkům přistupovat s obezřetností a požadovat od nich více informací, než by bylo nutné. Určité procento podvodů je ovšem nevyhnutelné. Těmto nákladům a nejisté povaze plateb je možné se vyhnout v osobním styku při použití fyzických peněz, neexistuje však žádný mechanismus provádění plateb přes komunikační kanál bez důvěryhodné třetí strany.

Je potřeba zavést systém elektronických plateb založený na kryptografických záznamech, nikoliv na důvěře, aby umožnil kterýmkoliv dvěma stranám provádět transakce přímo mezi sebou bez potřeby důvěryhodné třetí strany. Transakce, které je z výpočetního hlediska obtížné zvrátit, by chránily prodejce před podvodníky a na ochranu kupujících by mohly být snadno zavedeny rutinní escrow mechanismy. V tomto dokumentu navrhujeme řešení problému dvojí útraty pomocí peer-to-peer distribuovaného serveru přidělujícího časová razítka (timestamp

server), který vytváří výpočetní důkaz chronologického pořadí transakcí. Tento systém je bezpečný, dokud poctivé uzly kolektivně ovládají více výpočetní síly než jakákoliv spolupracující skupina útočících uzlů.

2. Transakce

Elektronickou minci definujeme jako řetězec digitálních podpisů. Každý vlastník převádí minci na následujícího tak, že digitálně podepíše hash předchozí transakce, veřejný klíč následujícího vlastníka a obojí přidá na konec elektronické mince. Příjemce pak může podpisy verifikovat, a tak řetězec vlastnictví potvrdit.

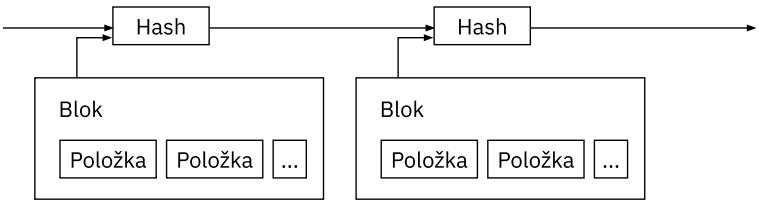


Příjemce nemůže potvrdit, že nedošlo ke dvojí útratě mince jedním z vlastníků, což samozřejmě představuje problém. Obvyklým řešením je zavedení důvěryhodné centrální autority či „mincovny"*, která všechny transakce ověřuje, nedošlo-li ke dvojí útratě. Po každé transakci musí být mince vrácena do mincovny a ta pak vydá minci novou. Jen u mincí vydaných přímo mincovnou je zaručeno, že u nich nedošlo ke dvojí útratě. Problém tohoto řešení spočívá v tom, že osud celého peněžního systému závisí na společnosti provozující mincovnu, přes kterou musí každá transakce projít obdobně jako přes banku.

Potřebujeme zajistit, aby příjemce věděl, že předchozí vlastníci nepodepsali žádné dřívější transakce. Pro naše účely se počítá jen transakce, která proběhla jako první, žádné následné pokusy o dvojí útratu nás tedy nezajímají. Jediným způsobem, jak potvrdit, že se nějaká transakce neuskutečnila, je o veškerých transakcích vědět. V modelu založeném na mincovně věděla o veškerých transakcích mincovna a ta také rozhodovala, ke kterým transakcím došlo dřív. Abychom toho dosáhli bez důvěryhodné třetí strany, transakce musí být oznamovány veřejně [1] a potřebujeme systém, který by umožnil shodu účastníků na tom, v jakém pořadí byly transakce přijaty. Příjemce potřebuje důkaz, že v okamžiku každé transakce většina uzlů souhlasila, že tato transakce byla právě tou první přijatou.

3. Server přidělující časová razítka (timestamp server)

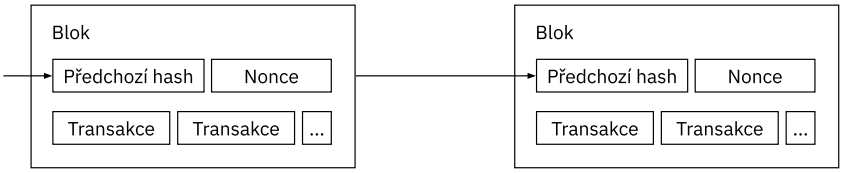
Námi navrhované řešení začíná serverem přidělujícím časová razítka. Tento server funguje tak, že vezme hash bloku položek, kterým mají být přidělena časová razítka, a zveřejní ho např. v novinách nebo v příspěvku na Usenetu.[2, 3, 4, 5] Časové razítko dokazuje, že daná data jednoznačně musela v příslušném okamžiku existovat, aby mohla být zahashována. Každé časové razítko do svého hashe začlení předchozí časové razítko, a vzniká tak řetězec, ve kterém každé další časové razítko posiluje všechna předchozí.



4. Důkaz o vykonané práci (proof-of-work)

K implementaci distribuovaného serveru přidělujícího časová razítka na peer-to-peer bázi budeme spíše než příspěvky v novinách či na Usenetu potřebovat systém založený na tzv. důkazu o provedené práci podobný Hashcashi Adama Backa.[6] Důkaz o provedené práci obnáší hledání takové hodnoty, která po zahashování, např. pomocí funkce SHA-256, začíná jedním nebo několika nulovými bity. Průměrná potřebná práce exponenciálně roste spolu s počtem potřebných nulových bitů a může být ověřena provedením jediného hashe.

V naší síti s časovými razítky důkaz o vykonané práci spočívá v postupném zvyšování hodnoty nonce v příslušném bloku až do chvíle, kdy je nalezena hodnota, při které hash bloku obsahuje potřebné nulové bity. Jakmile bylo vynaloženo výpočetní úsilí k uspokojení důkazu o vykonané práci, blok již nemůže být změněn bez opětovného provedení práce. Protože se za blokem řetězí další bloky, práce potřebná k jeho změně by vyžadovala přepracování všech bloků po něm.



Důkaz o vykonané práci řeší také problém, jakým způsobem je při rozhodování reprezentovaná většina. Kdyby byla většina založena na principu jeden hlas = jedna IP adresa, systém by mohl být rozvrácen kýmkoliv, kdo by byl schopen získat velké množství IP adres. Důkaz o vykonané práci v podstatě znamená, že jeden hlas = jeden procesor. Většinové rozhodnutí je reprezentováno nejdelším řetězcem, do něhož bylo investováno nejvíce úsilí důkazem o vykonané práci. Je-li většina výpočetní síly ovládána poctivými uzly, poctivý řetězec

poroste nejrychleji a předstihne veškeré konkurenční řetězce. Ke změně některého z předchozích bloků by útočník musel opětovně provést důkaz o vykonané práci v příslušném bloku i ve všech následujících blocích a potom dohnat a předstihnout práci poctivých uzlů. Jak ukážeme dále, pravděpodobnost, že pomalejší útočník dožene poctivé uzly, se exponenciálně snižuje s rostoucím počtem bloků v řetězci.

Aby byla vyvážena rostoucí rychlost hardwaru a proměnlivý zájem o provozování uzlů v průběhu času, obtížnost důkazu o vykonané práci je stanovena na základě klouzavého průměru cíleného na určitý průměrný počet bloků za hodinu. Jsou-li bloky generovány příliš rychle, obtížnost se zvyšuje.

5. Síť

Kroky k provozu sítě jsou následující:

1) Nové transakce jsou vysílány všem uzlům.
2) Každý uzel shromažďuje nové transakce do bloku.
3) Každý uzel pro svůj blok hledá obtížně nalezitelný důkaz o vykonané práci.
4) Když některý uzel najde důkaz o vykonané práci, vyšle blok všem ostatním uzlům.
5) Uzly blok přijmou pouze tehdy, jsou-li všechny transakce, které jsou v něm obsažené, platné a nedošlo u nich k dvojí útratě.
6) Uzly příslušný blok přijmou tak, že začnou pracovat na dalším bloku v řetězci s použitím hashe přijatého bloku coby reference na předchozí blok.

Uzly vždy považují za správný nejdelší řetězec a navazují vždy na něj. Pokud vyšlou dva uzly současně dvě různé verze následujícího bloku, může se stát, že některé další uzly obdrží jednu z nich jako první. V takovém případě pracují na první verzi, kterou obdržely, ale uloží si i druhou větev, pokud by se právě ona měla stát tou delší. Vazba mezi oběma verzemi je zrušena v momentě, kdy je nalezen následující důkaz o vykonané práci a jedna z větví se stane tou delší; uzly, které pracovaly na druhé větvi, ji opustí a přejdou na delší větev.

Vysílání nových transakcí nemusí nutně dorazit ke všem uzlům. Pokud dorazí k většímu množství uzlů, co nevidět jsou zařazeny do bloku. Vysílání bloků je také odolné vůči ztrátě zpráv. Když některý uzel blok neobdrží, vyžádá si ho po přijetí bloku následujícího, protože tím zjistí, že mu předchozí blok chybí.

6. Motivační odměna

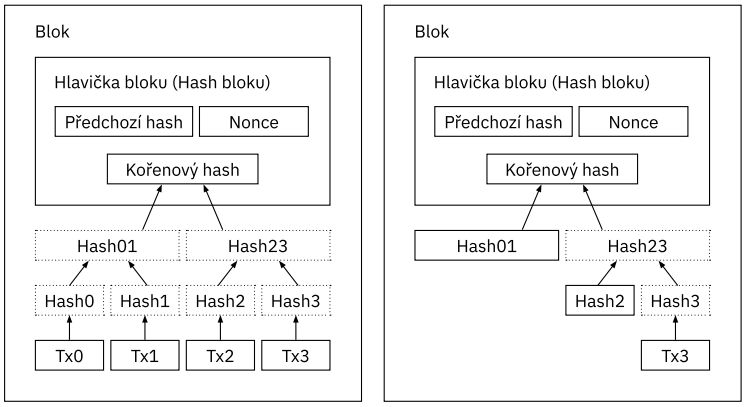
První transakce v bloku je podle dohody transakcí speciální, kterou začíná nová mince vlastněná tvůrcem bloku. To funguje pro uzly jako odměna, která je motivuje síť podporovat, a vzhledem k absenci centrálního orgánu, který by mince vydával, to zároveň představuje způsob, kterým jsou mince zpočátku uváděny do oběhu. Stabilní přírůst konstantního množství nových mincí lze připodobnit k tomu, když zlatokopové vynakládají úsilí, aby přidávali do oběhu zlato. V našem případě se vynakládá výpočetní čas a elektřina.

Motivační odměna může být také financována z transakčních poplatků. Je-li výstupní hodnota transakce nižší než její vstupní hodnota, rozdíl je v podstatě transakčním poplatkem, který se přičítá k hodnotě motivační odměny bloku obsahujícího příslušnou transakci. Když do oběhu vstoupí předem stanovené množství mincí, motivační odměna může plně přejít na transakční poplatky a být zcela bez inflace.

Motivační odměna napomáhá zachovávat poctivost uzlů. Pokud se chamtivému útočníkovi podaří shromáždit více výpočetní síly, než kolik mají všechny poctivé uzly, musí se rozhodnout, zda tuto sílu využije k obírání lidí vracením svých plateb, nebo ke generování nových mincí. Mělo by mu dojít, že dodržování pravidel, která mu umožní získat více mincí než všem ostatním dohromady, je pro něj výhodnější než podrývání systému a tím i hodnoty vlastního bohatství.

7. Hospodaření s úložným prostorem

Jakmile je poslední transakce v minci překryta dostatečným množstvím bloků, je možné předchozí utracené transakce smazat, a ušetřit tak paměť. Aby to bylo možné bez porušení hashe příslušného bloku, transakce jsou hashovány ve struktuře Merkleova stromu.[7, 2, 5] Hash bloku obsahuje vždy jen kořen (Merkleova) stromu. Staré bloky tak mohou být komprimovány odříznutím větví stromu. Vnitřní hashe není nutné archivovat.



Transakce zahashované ve struktuře Merkleova stromu **Blok po ořezání Tx0-2**

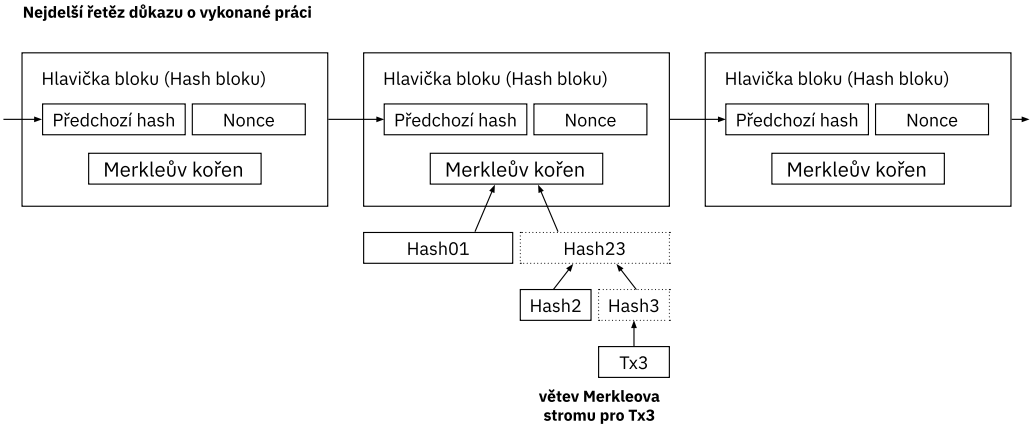
Hlavička bloku by bez transakcí měla velikost asi 80 bytů. Předpokládáme-li, že bloky jsou generovány každých 10 minut, dělá to 80 bytů × 6 × 24 × 365 = 4,2 MB za rok. Při typické kapacitě počítačových systémů 2GB RAM v roce 2008 a s přihlédnutím k Moorovu zákonu, který předpovídá růst paměti o 1,2 GB ročně, by ani nutnost načítání hlaviček do paměti neměla představovat problém.

8. Zjednodušené ověřování plateb

Uživatel může platby ověřovat i v případě, že neprovozuje plnohodnotný síťový uzel. Stačí, když uchovává kopie hlaviček bloků z nejdelšího řetězce důkazu o vykonané práci. Tuto informaci získá ptaním se okolních uzlů, dokud si není dostatečně jistý, že má nejdelší řetězec, a pak odvodí kořen Merkleova stromu spojením hledané transakce s blokem, v němž se nachází její časové razítko. Uživatel si nemůže transakci ověřit sám, ale tím, že dokáže ověřit její pozici v řetězci transakcí, se ujistí, že ji nějaký uzel přijal, a později přidané bloky dále potvrzují, že transakci akceptovala celá síť.

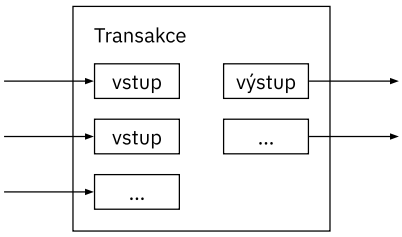
Toto ověřování je zcela spolehlivé, pokud je většina uzlů v síti poctivá, ale v případě, že síť ovládne útočník, je tento způsob zranitelnější. Jednotlivé uzly sice mohou ověřovat transakce samy, ale dokud útočník ovládá většinu sítě, může výše popsaný systém zjednodušeného ověřování klamat falešnými transakcemi. Jistou ochrannou strategii by mohlo představovat

přijímání upozornění od síťových uzlů, které identifikovaly neplatný blok. Software uživatele by reagoval stažením celého bloku a podezřelých transakcí, aby si potvrdil případné nesrovnalosti. Počítá se nicméně s tím, že uživatelé sítě, kteří často přijímají platby, budou chtít provozovat plnohodnotný uzel, což jim umožní nezávislejší bezpečnost a rychlejší ověřování.



9. Slučování a rozdělování částek

Ačkoliv by bylo technicky možné evidovat mince jednotlivě, bylo by nepraktické při převodu částky provádět oddělenou transakci pro každý převáděný cent. Aby mohly být částky sloučeny nebo rozděleny, transakce obsahují více vstupů a výstupů. Obvykle jde buď o jediný vstup z předchozí větší transakce, nebo o několik vstupů složených z menších částek a maximálně dva výstupy – jeden pro platbu a případně druhý, kterým se odesílateli vrací nazpět.

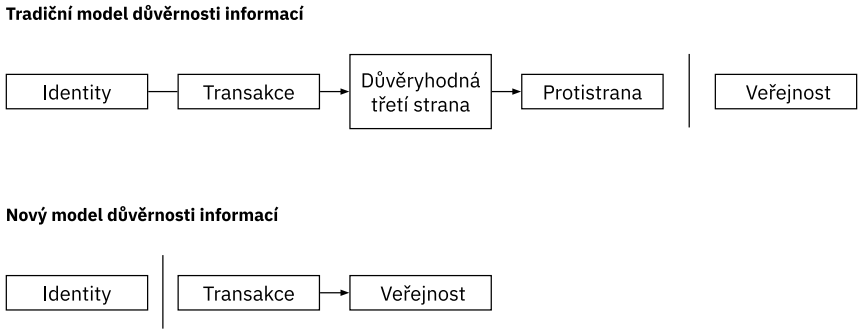


Je třeba zmínit, že vzniklá rozvětvená struktura, kdy jedna transakce vychází z několika dalších transakcí a ty zase z mnoha dalších transakcí, nepředstavuje z praktického hlediska žádný problém. Nikdy totiž není nutné odděleně extrahovat kompletní historii jedné konkrétní platby.

10. Ochrana osobních údajů

Tradiční bankovní model zajišťuje jistou míru důvěrnosti informací tím, že umožňuje přístup k informacím pouze zúčastněným stranám a důvěryhodné třetí straně. To, že je třeba zveřejňovat všechny transakce, tomuto modelu vlastně brání. Důvěrnost informací lze však zachovat přerušením toku informací na jiném místě – tak, že veřejné klíče ponecháme anonymní. Každý

vidí, že jeden uživatel posílá druhému určitou částku, ale není zde žádná informace, která by umožnila transakci spojit s konkrétními osobami. Na podobné úrovni zveřejňování informací fungují burzy cenných papírů, které prostřednictvím tzv. „pásky"** (tape) zveřejňují objem a čas prováděných transakcí, aniž jakkoli informují o zúčastněných stranách.



Jako dodatečná ochrana se pro každou transakci doporučuje používat novou dvojici klíčů, aby je nebylo možné spojit s jediným vlastníkem. Odhalení určité spojitosti je přesto nevyhnutelné v případě transakcí s více vstupy, u kterých je zřejmé, že pocházejí od jednoho vlastníka. Nebezpečí zde spočívá v tom, že pokud je odhalen vlastník určitého klíče, spojení by mohlo vést k odhalení jeho dalších transakcí.

11. Výpočty

Zvažme situaci, kdy se útočník pokouší vytvářet alternativní řetězec rychleji, než vzniká řetězec poctivý. I kdyby se mu to podařilo, systém nezačne jen tak přijímat libovolné změny: nelze například vytvářet hodnotu z ničeho a útočník si nemůže přivlastňovat peníze, které mu nikdy nepatřily. Uzly nikdy nepřijmou jako platbu neplatnou transakci a poctivé uzly nikdy nepřijmou blok, který by takovou transakci obsahoval. Případný útočník se může pouze pokusit změnit některou ze svých vlastních transakcí, a získat tak zpět své nedávno utracené peníze.

Závod mezi poctivým řetězcem a útočícím řetězcem můžeme popsat s využitím matematického modelu náhodné procházky (Random Walk). V úspěšném případě dojde k prodloužení poctivého řetězce o jeden blok a ke zvýšení jeho náskoku o +1, v neúspěšném případě dojde k prodloužení útočníkova řetězu o jeden blok, a snížení rozdílu o −1.

Výpočet pravděpodobnosti, že útočník dožene poctivý řetězec s určitým předem daným náskokem, odpovídá problému zruinování hráče (Gambler's Ruin Problem). Předpokládejme, že hráč s neomezeným kontem začíná v minusu a má potenciálně neomezený počet pokusů, aby dosáhl hranice rentability. Pravděpodobnost, že útočník hranice rentability dosáhne, tedy že dožene poctivý řetězec, lze spočítat následujícím způsobem: [8]

p = pravděpodobnost, že nový blok bude nalezen poctivým uzlem
q = pravděpodobnost, že nový blok bude nalezen útočníkem
q_z = pravděpodobnost, že útočník dožene náskok z bloků

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

Předpokládáme-li, že $p > q$, pravděpodobnost se exponenciálně snižuje se zvyšujícím se počtem bloků, které útočník musí dostihnout. Nepodaří-li se útočníkovi získat výhodu hned zpočátku, pak jeho šance na úspěch spolu s tím, jak stále víc a víc zaostává, vzhledem k už tak nízké pravděpodobnosti klesá až téměř na nulu.

Zvažme nyní, jak dlouhou dobu příjemce nové transakce potřebuje vyčkat, aby si mohl být dostatečně jistý, že odesílatel již transakci nemůže změnit. Předpokládáme, že odesílatelem je útočník, který se příjemce snaží na chvíli přesvědčit, že mu zaplatil, a po uplynutí nějaké doby platbu sám sobě vrátit. Příjemce o tom sice bude notifikován, odesílatel však doufá, že už bude příliš pozdě.

Příjemce generuje novou dvojici klíčů a veřejný klíč předává odesílateli těsně před podpisem transakce. Tak odesílateli znemožňuje, aby si nejprve soustavnou prací připravil alternativní řetězec bloků a transakci provedl až ve chvíli, kdy se mu podaří získat dostatečný náskok. Jakmile je transakce odeslána, nepoctivý odesílatel začíná tajně pracovat na paralelním řetězci s alternativní verzí své transakce.

Příjemce počká, až bude transakce přidána do bloku a za blok bude připojeno z dalších bloků. Příjemce nezná přesný rozsah útočníkova pokroku, ale počítáme-li s průměrnou dobou vytváření nových bloků, útočníkův potenciální pokrok lze vyjádřit jako Poissonovo rozdělení s očekávanou hodnotou:

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

Pravděpodobnost, že útočník poctivý řetězec v tuto chvíli ještě dostihne, vypočítáme tak, že pro každý možný objem útočníkova postupu vynásobíme Poissonovu hustotu pravděpodobností, že by z daného stavu ještě mohl poctivý řetězec dostihnout:

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

Abychom se vyhnuli součtu nekonečné řady rozdělení, zapíšeme to následovně:

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

Po přepisu 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;
}
```

Když zkusíme několik výpočtů, zjistíme, že pravděpodobnost exponenciálně 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
```

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

Předložili jsme systém elektronických transakcí, který není závislý na důvěře. Vyšli jsme z obvyklého systému mincí založeného na digitálních podpisech, který sice poskytuje silnou kontrolu vlastnických práv, je však neúplný, protože nenabízí způsob, jak zabránit dvojím útratám. Tento problém jsme navrhli vyřešit pomocí peer-to-peer sítě využívající důkaz o vykonané práci (proof-of-work) k vytvoření veřejného záznamu historie transakcí, který se v případě, že většinu výpočetní síly kontrolují poctivé uzly, pro případného útočníka rychle stává z výpočetního hlediska prakticky nezměnitelným.

Robustnost této sítě spočívá v její nestrukturovanosti a v jednoduchosti. Všechny uzly pracují zároveň s minimální mírou koordinace. Identifikace uzlů není nutná, neboť zprávy nejsou určeny žádnému konkrétnímu příjemci a šíří se pouze na principu best effort. Uzly se mohou kdykoliv odpojit nebo opět připojit, přičemž akceptují nejdelší řetězec důkazu o vykonané práci jako záznam událostí, ke kterým došlo v jejich nepřítomnosti. Hlasují svou výpočetní silou, přičemž platné bloky přijímají tím, že na nich pracují a prodlužují je, a neplatné bloky zamítají tak, že na nich pracovat odmítají. Prostřednictvím tohoto mechanismu vzájemné shody lze prosazovat jakákoliv případná pravidla a motivační odměny.

Reference

1 Wei Dai, “b-money”, http://www.weidai.com bmoney.txt, 1998.

2 Henri Massias, Xavier Serret-Avila, and Jean-Jacques Quisquater, “ Design of a secure timestamping service with minimal trust requirements,” In 20th Symposium on Information Theory in the Benelux, květen 1999.

3 Stuart Haber, W. Scott Stornetta, “How to time-stamp a digital document,” In Journal of Cryptology, ročník 3, číslo 2, strany 99-111, 1991.

4 Dave Bayer, Stuart Haber, W. Scott Stornetta, “Improving the efficiency and reliability of digital time-stamping,” In Sequences II: Methods in Communication, Security and Computer Science, strany 329-334, 1993.

5 Stuart Haber, W. Scott Stornetta, “Secure names for bit-strings,” In Proceedings of the 4th ACM Conference on Computer and Communications Security, s. 28-35, duben 1997.

6 Adam Back, “Hashcash - a denial of service counter-measure,” http://www.hashcash.org/papers/hashcash.pdf, 2002.

7 Ralph C. Merkle, “Protocols for public key cryptosystems,” In Proc. 1980 Symposium on Security and Privacy, IEEE Computer Society, s. 122-133, duben 1980.

8 William Feller, “An introduction to probability theory and its applications,” 1957.

* Mincovnu je třeba v tomto kontextu chápat jako abstraktní instituci, která by mohla podobnou dozorčí činnost vykonávat. Mincovny historicky neměly kontrolní úlohu nad transakcemi, pouze razily nové fyzické mince. (Pozn. překl.)

** Telegramové pásky představovaly nejstarší nástroj, kterým se na burzách komunikovaly ceny komodit. Ačkoliv byla technologie v 70. letech nahrazena počítači, princip zůstává stejný: na obrazovce podobně jako na pásku běží přehled objemu a cen obchodovaných komodit. (Pozn. překl.)

Glosář

peer-to-peer

Typ počítačové sítě, kde spolu přímo komunikují jednotliví uživatelé jako peer-to-peer, tedy rovný s rovným, bez přítomnosti centrálního serveru. Uživatelé či uzly na síti jsou si rovnocenné, není tu přítomna žádná centrální autorita. Bitcoin je od počátku koncipován jako peer-to-peer síť.

dvojí útrata (double spend)

Situace, kdy se ty samé bitcoiny uživatel pokusí utratit vícekrát. Vyřešení problému dvojité útraty tím, že jsou transakce zapisovány do řetězce a chráněny důkazem o vykonané práci, je hlavní technologický přínos Satoshiho Nakamota.

důkaz o vykonané práci (proof-of-work, POW)

Data, pro které je možné snadno ověřit, že splňují nějakou specifickou vlastnost, ale pro jejichž vytvoření není známý žádný efektivní postup. Data naopak musíme vytvořit „neefektivním“ způsobem, při kterém je vynakládána faktická práce (v podobě času, výpočetního výkonu či elektřiny k němu potřebné). Výsledná data pak lze chápat jako důkaz, že někdo vykonal práci pro nalezení takových dat. Kritickou vlastností mechanismu proof of work je pak právě neschopnost “zjednodušit” si práci například nějakým chytrým trikem nebo zrychleným výpočtem. V případě Bitcoinu se jedná o počítání hashe hlavičky bloku (náhodné číslo), který musí mít menší než stanovenou hodnotu. Je triviální ověřit, že pro konkrétní hlavičku je podmínka splněna – stačí porovnat spočítaný hash s daným číslem. Ale není jednoduché najít takovou hlavičku, pro kterou po zahashování dostaneme očekávaný výsledek, protože výsledek hashovaci funkce se chová zcela náhodně. Při hledání vhodného hashe tak nezbývá nic jiného, než opakovaně zkoušet hashovat různé vstupy.

Hashovací funkce (hash function)

Matematický algoritmus převodu libovolně dlouhých vstupních dat na číslo o omezené velikosti. Výstup takové funkce se označuje jako hash a funguje jako „otisk” vstupních dat. Mezi přednosti hashovací funkce patří především to, že sebemenší změna vstupu vede k zásadně odlišnému výsledku, který vypadá nahodile. Druhou předností je, že vstupní data se nedají prakticky rekonstruovat na základě znalosti výsledného hashe.

uzel (node)

V systému Bitcoin je uzlem každý počítač či zařízení, které se k síti připojí. Uzel si vyměnuje data s okolními zuly a typicky uchovává kopii celého řetězce, validuje nové bloky a transakce a je skrze něj možné vyslat transakci k ostatním uzlům v síti. Vzhledem k peer-to-peer povaze bitcoinové sítě jsou si všechny uzly rovny.

escrow mechanismus

Je forma kontraktu, ve které figuruje prostředník (escrow agent) zprostředkovávající transakci mezi dvěma stranami, které by si případně nedůvěřovaly. Escrow agent může držet prostředky, dokud se obě strany nedohodnou. V technologii Bitcoinu je tohoto mechanismu použito při ověřování podpisů u multisig adres.

veřejný klíč (public key)

Jeden z dvojice klíčů v asymetrické kryptografii (spolu se soukromým klíčem). V Bitcoinu slouží veřejný klíč pro identifikaci příjemce peněz v transakci. Z veřejného klíče se generuje textová adresa, se kterou se uživatelům lépe pracuje.

Hashcash

Systém založený na důkazu o vykonané práci, který brání proti e-mailovému spamu a DoS útokům. Systém, který v roce 1997 představil Adam Back, připojoval před odesláním ke každému e-mailu časové razítko, jehož výpočet počítači zabere malé množství času. Jelikož ale útočníci

posílají velké množství nevyžádaných zpráv najednou, výpočet časového razítka je zbrzdí v masovému odesílání zpráv.

Usenet (User's Network)

Představuje jeden z nejstarších webových komunikačních systémů. Vznikl na začátku 80. let ve Spojených státech. Systém tvořily vzájemně propojené uzly, které si mezi sebou předávaly zprávy.

SHA-256

Zkratka z anglického Secure Hash Algorithm, tedy Bezpečný hashovací algoritmus je kryptografická funkce, která z libovolně dlouhého vstupu vytvoří výstup fixní délky. Z výstupu je proto prakticky nemožné rekonstruovat původní vstup, stejně jako narazit na dvě rozdílné zprávy s totožným výstupem. Číslo 256 značí délku výstupu v bitech.

nonce

Značí v kryptografii číslo, které se používá jako jednorázová hodnota přinášející náhodný element. Takové hodnotě není přisuzován žádný specifický význam, její role spočívá pouze v její libovolnosti a nemožnosti ji odhadnout. Nonce tvoří součást bitcoinového bloku proto, aby bylo možné tuto hodnotu libovolně měnit a zkoušet tak hledat hash hlavičky splňující podmínku pro platný blok.

Merkleův strom (Merkle tree)

Někdy též zvaný jako binární hashovací strom je druh datové struktury používané v kryptografii. Slouží k efektivnímu zakódování dat v řetězci důkazů o vykonané práci. Umožňuje totiž ověření konkrétní transakce z bloku bez nutnosti načtení celého řetězce.

hlavička bloku (Block header)

Datová struktura obsahující klíčové hodnoty v bitcoinovém bloku: hash hlavičky předchozího bloku, čas a obtížnost, kdy byl blok vytěžen, kořen Merkleova stromu transakcí obsažených v daném bloku, verzi bloku a nonci. Právě hashováním hlavičky bloku tvoří těžaři důkaz o vykonané práci, aby daný blok vytěžili. Těžaři tedy nemusí pracovat při hashování s celým blokem tvořeným tisíci transakcí.

náhodná procházka (Random Walk)

Koncept, který popisuje, s jakou pravděpodobností se lze dostat v systému do určitých stavů po provedení daného množství náhodných kroků různými směry s danými pravděpodobnostmi. V kontextu Bitcoinu, kdy mezi sebou soutěží dva řetězce, znamená prodloužení prvního z nich o jeden blok krok jedním směrem a prodloužení druhého z nich krok druhým směrem. Pravděpodobnosti jednotlivých kroků jsou známy. Lze tedy vyvodit, kterým směrem se systém jako celek bude pohybovat, i například s jakou směrodatnou odchylkou.

problém zruinování hráče (Gambler's Ruin problem)

Matematický model pravděpodobnostní hry na tahy, který předpovídá, s jakou pravděpodobností hráč prohraje svoje peníze po odehrání určitého množství kol. V našem kontextu hráči soupeří o délku zapsaného řetězce na počet bloků a v sázce je množství vynaložené práce. Hráč, který začíná se zpožděním a snaží se dohnat ztrátu a vyprodukovat delší vítězný řetězec s menším vynaloženým úsilím než protihráč, má určitou pravděpodobnost, že se mu to podaří. Tato pravděpodobnost však exponenciálně klesá s tím, jak velkou ztrátu se snaží dohnat. To ve výsledku znamená, že pokus o nalezení a prosazení alternativní historie minoritními hráči statisticky povede k neúspěchu a ke ztrátě jimi vynaložené práce.

Poissonovo rozdělení (Poisson distribution)

Poissonovo rozdělení charakterizuje pravděpodobnosti výskytu určitého množství jevů, pokud se dějí nezávisle na sobě s určitou průměrnou frekvencí. Určuje například, jaká je pravděpodobnost, že do domu uhodí blesk v daném roce, když víme, že v průměru k tomu dojde jednou za 10 let. Podrobné vysvětlení Nakamotovy analýzy problému dvojí útraty naleznete zde.