Bitcoin: Peer-to-peer електронна парична система

Сатоши Накамото
satoshin@gmx.com
www.bitcoin.org

Резюме: Една чиста peer-to-peer (архитектура с равнопоставени възли) версия на електронните пари би позволила онлайн плащанията да бъдат изпращани директно от една страна до друга без да минават през финансова институция. Електронните подписи осигуряват една част от решението, но основните ползи се губят, тъй като все още е необходимо сигурна трета страна да предотврати двойното харчене. Ние предлагаме решение на проблема с двойното харчене, използвайки peer-to-peer (P2P) мрежа. Мрежата поставя времеви маркери на трансакциите като ги хешира в онлайн верига от хешове, базирани на доказателство за работа (Proof-of-Work / PoW), формирайки запис, който не може да бъде променян без повторно изработване на доказателството за работа. Най-дългата верига служи не само като доказателство за последователността на наблюдаваните събития, но и като доказателство, че те идват от най-големия басейн от процесорна (изчислителна) мощ. Докато по-голямата част от изчислителната мощност се контролира от основни устройства за данни (възли), които не си сътрудничат, за да атакуват мрежата, те ще генерират най-дългата верига по-бързо от потенциалните нападатели (на мрежата). Самата мрежа изисква минимална структура. Съобщенията се предават на принципа на най-добрия опит, а възлите могат да напускат и отново да се включват към мрежата по желание, приемайки най-дългата PoW верига за доказателство какво се е случило по време на отсъствието им.

1. Въведение

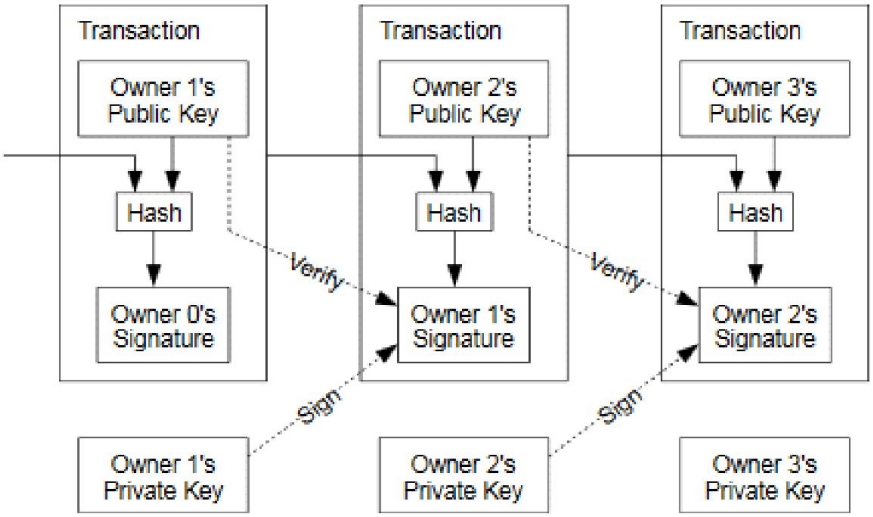
Интернет търговията е приела да разчита почти изцяло на финансови институции, които да изпълняват ролята на доверителни трети страни при обработката на електронни плащания. Въпреки че системата работи достатъчно добре за повечето трансакции, тя все пак страда от вродените слабости на базирания на доверие модел. Напълно необратими трансакции наистина не са възможни, тъй като

финансовите институции не могат да избегнат посредническите спорове. Цената на посредничеството увеличава трансакционните разходи. Тя ограничава минималния размер на практическата трансакция и намалява възможността за малки случайни трансакции. Така се губи способността за извършване на необратими плащания за необратими услуги. При наличието на възможност за връщане, необходимостта от доверие се увеличава. На търговците им се налага да са постоянно нащрек по отношение на клиентите и да изискват от тях повече информация, отколкото би им била необходима иначе. Определен процент от измами се приемат като неизбежни. Тези разходи и разплащателни несигурности могат ба бъдат избегнати в личен план чрез използването на физическа валута, но няма механизъм за извършване на плащания по комуникационен канал, без доверена страна.

Необходима е електронна система за плащания, базирана на криптографско доказателство, вместо на доверие, което да позволи на двете страни да сключат сделка, без да се нуждаят от трета доверена страна. Трансакции, които са математически неизгодни да бъдат обратими, биха предпазили продавачите от измами, а лесно могат да бъдат приложени и познатите механизми на набирателните сметки за да защитят купувачите. В този документ предлагаме решение на проблема с двойното харчене, като се използва сървър за разпределение на времевите печати от типа peer-to-peer, за да се генерира математическо потвърждение за хронологичния ред на трансакциите. Докато колективната изчислителна мощност на честните възли в мрежата е по-голяма от изчислителната мощност на потенциалните атакуващи възли, системата може да бъде считана за сигурна.

2. Трансакции

Ние определяме електронната монета като верига от цифрови подписи. Всеки собственик прехвърля монетата на следващия с цифров подпис на хеш от предишната трансакция и публичния ключ на следващия собственик и ги добавя в края на монетата. Получателят може да провери подписите, за да провери веригата на собственост.



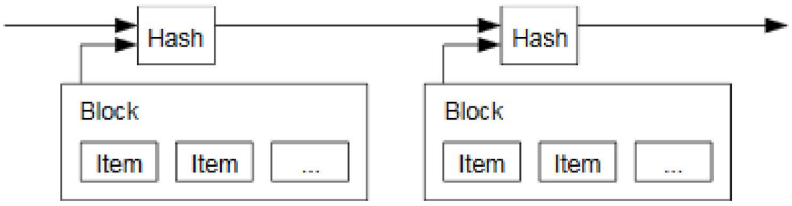
Проблемът, разбира се, е че получателят не може да провери дали някой от собствениците не е похарчил два пъти монетата. Обичайно решение е да се въведе доверен централен орган или монетен двор, който да проверява всяка трансакция за двойно харчене. След всяка трансакция монетата трябва да бъде върната на централния орган, за да бъде издадена нова монета, и само за монетите, издавани директно от монетния двор е сигурно, че не са похарчени двойно. Проблемът с това решение е, че съдбата на цялата парична система зависи от компанията, управляваща монетния двор, като всяка трансакция трябва да мине през тях, точно както през банка.

Нуждаем се от начин, по който получателят да знае, че предишните собственици не са подписали по-ранни трансакции. За нашите цели, най-ранната сделка е тази, която се взема под внимание и не ни

интересуват опитите за двойно харчене след нея. Единственият начин да потвърдите липсата на такива трансакции е да сте наясно с всички трансакции. В модела с монетния двор, дворът е наясно с всички трансакции и е решил коя от тях е първата. За да се постигне това без трета доверена страна, трансакциите трябва да са публично оповестени. Нуждаем се от система, която позволява на участниците да се споразумеят за реда, по който трансакциите са получени. Получателят се нуждае от доказателство, че по времето на всяка една от трансакциите, болшинството от възлите са съгласни, че тя е първата получена.

3. Сървър с времеви печати

Решението, което предлагаме, започва със сървър за времеви печати. Той работи така: взема хеш от блок от елементи, маркира го с време и разпространява информацията за хеша - по подобие на датирането на публикациите във вестниците или форумите. Печатът показва, че данните трябва да са съществували по онова време, за да влязат в хеша. Всеки времеви печат включва предишните печати в техните хешове и образува верига, като всяко добавено време подсилва тези преди него.

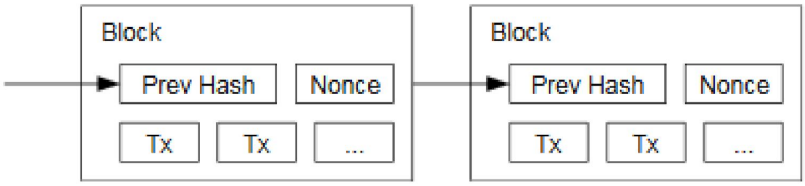


4. Доказателство за работа (Proof-of-Work / PoW)

За да внедрим сървър за времеви печати на принципа "peer-to-peer", ще трябва да използваме система за доказателство на работата, подобна на Hashcash на Adam Back и доста по-добра от публикациите във вестниците или в Usenet. Доказателството за работа включва сканиране

за стойност, която при хеширане, например при SHA-256, хешът започва с определен брой нула бита. Средната необходима работа е експоненциална на броя на необходимите нулеви битове и може да бъде потвърдена чрез изпълнение на единичен хеш.

За нашата мрежа за времеви печати ние прилагаме доказателството за работа, като увеличаваме с единица еднократното случайно число (nonce) в блока, докато не бъде намерена стойност, която дава на блока необходимите нулеви бита. След като усилията на процесора са били изразходвани, за да може да бъде удовлетворено доказателството за работа, блокът не може да бъде променян без повторно извършване на работата. Когато следващите блокът се наредят във веригата, работата за промяната му ще включва повторно изчисляване на всички блокове след него.



Доказателството за работа решава и проблема с определянето на представителството при взимане на мнозинство. Ако мнозинството се изразяваше с един IP-адрес – един глас, то това може да бъде саботирано от всеки, който разполага с много IP-адреси. По същество, доказателството за работа е едно CPU – един глас. Решението на мнозинството е представено от най-дългата верига, в която са инвестирани усилията на най-голямото доказателство за работа. Ако по-голямата част от мощността на процесора се контролира от честни възли, честната верига ще се разраства бързо и ще изпревари конкурентните вериги. За да направи промени в един минал блок, нападателят трябва отново да докаже работата на блока, както и на всички блокове след него, а след това да настигне и надмине работата на честните възли. По-късно ще покажем, че вероятността за по-бавна

атака от страна на нападателя прогресивно намалява , тъй като се добавят последващите блокове.

За да компенсира увеличаването на скоростта на хардуера и променливия интерес към поддържане на възли (нодове), трудността на доказателството за работа се определя от пълзящата средна стойност на блокове за час. Ако блоковете се генерират твърде бързо, трудността се увеличава.

5. Мрежа

Стъпките за функционирането на мрежата са следните:

1) Новите трансакции се предават към всички възли.
2) Всеки възел събира новите трансакции в блок.
3) Всеки възел работи за намиране на доказателство за работа на неговия блок.
4) Когато възелът намери доказателство за работа, той разпространява блока до всички възли.
5) Възлите приемат блока само, ако всички трансакции в него са валидни и вече не са били похарчени.
6) Възлите показват, че приемат блока, като използват хеша му за предхождащ хеш и работят по създаване на следващия блок във веригата.

Възлите винаги смятат, че най-дългата верига е най-правилната и продължават работата си по продължаването ѝ. Ако два възела едновременно излъчват различни версии на следващ блок, някои от възлите могат да ги получат в различна последователност. В този случай, те работят върху първия, който са получили, но запазват и другия клон, в случай че той стане по-дълъг. Връзката ще бъде прекъсната когато се намери следващо доказателство за работа и

единият клон стане по-дълъг. Така възлите, които са работили по другия, веднага ще се насочат към по-дългия клон.

Не е задължително излъчванията на новите трансакции да достигнат до всички възли. През времето, необходимо да стигнат до много възли, те могат да влязат в блок и да отпаднат като съобщения. Ако даден възел не получи информацията за блока, може да я поиска при получаването на следващ блок, когато става ясно, че предишният блок е пропуснат.

6. Възнаграждение

Според споразумението, първата трансакция в блока е специална и е начало на нова монета, собственост на създателя на блока. Това е допълнително стимулиране за възлите да подкрепят мрежата и дава възможност за първоначално разпространения на монетите в обращение, понеже няма централен орган за издаването им. Непрестанното добавяне на постоянни суми от нови монети е аналогично на работата на миньорите, които копаят злато и харчат ресурси, за да пускат златото в обращение. В нашия случай се изразходва електроенергия и времето за работа на процесора.

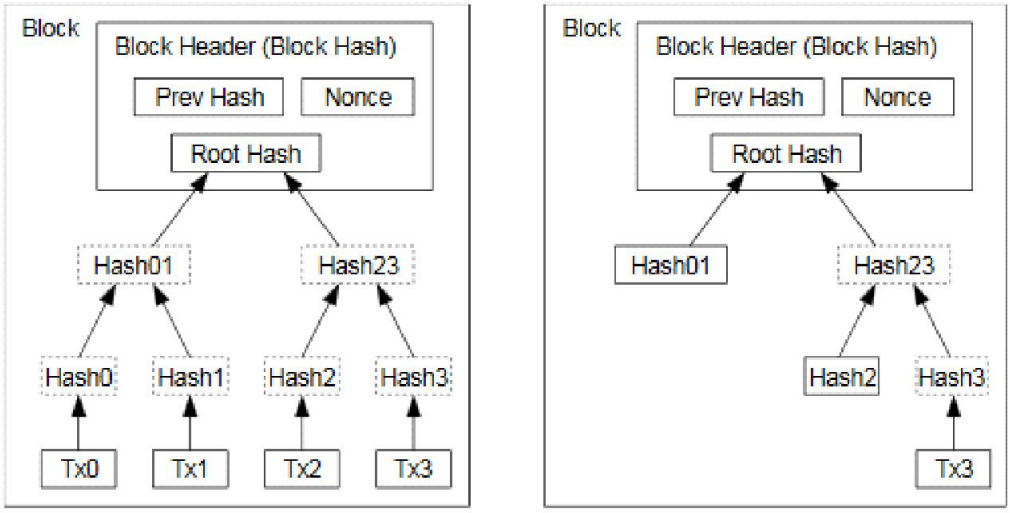
Възнаграждението може да бъде финансирано и с трансакционни такси. Ако изходящата стойност на дадена трансакция е по-малка от нейната входяща стойност, разликата се нарича такса за трансакцията, която се добавя към възнаграждението за блока, съдържащ трансакцията. След като в обращение бъдат пуснати предварително определения брой монети, стимулът може изцяло да се превърне в такси за трансакции и да не подлежи на инфлация.

Възнаграждението може да помогне на възлите да останат честни. Ако се появи нападател, който е в състояние да събере повече мощност от всички честни възли, той ще трябва да избира между използването на тази сила за да измами хората или за генериране на нови монети. Така се получава, че е по-изгодно да се играе по правилата и да се спечелят

нови монети, вместо да се подкопава системата и да се дискредитира валидността и на собственото богатство.

7. Възстановяване на дисковото пространство

След като последната трансакция в една монета е затрупана под достатъчно блокове, всички трансакции преди нея могат да бъдат съкратени, за да се спести място на диска. За да се улесни това, без да се прекъсва хешът на блока, трансакциите се преобразуват в Merkle Tree, като само коренът е включен в хеша на блока. След това старите блокове могат да бъдат сбити като се окастрят клоните на дървото. Не е необходимо да се съхраняват вътрешните хешове.



Transactions Hashed in a Merkle Tree

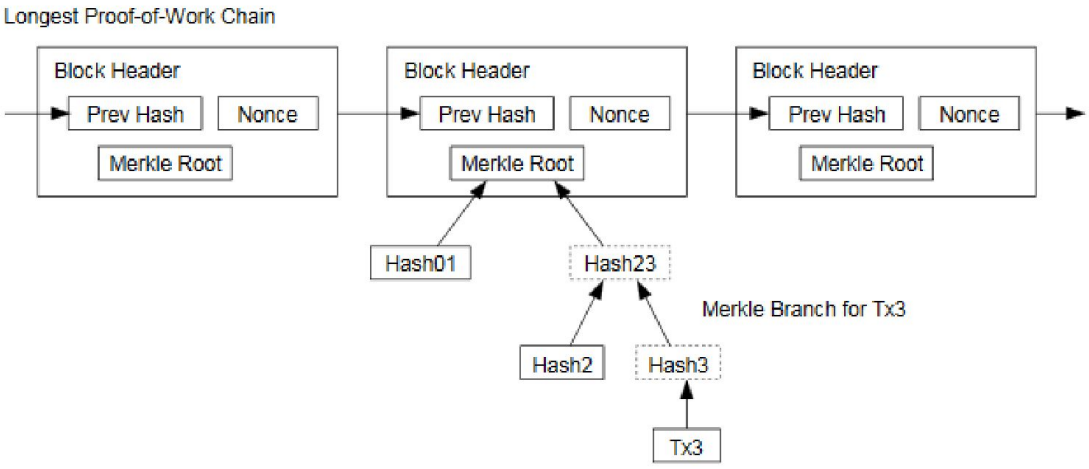
After Pruning Tx0-2 from the Block

Заглавната част (хедър) на блока, без трансакции, би била около 80 байта. Ако предположим, че блоковете се генерират на всеки 10 минути, 80 байта x 6 x 24 x 365 = 4.2 MB годишно. От 2008 г., компютрите обикновено се продават с 2 GB RAM, а според Закона на Мур прогнозите са за ръст от 1.2 GB на година. Така складирането не

би трябвало да е проблем, дори ако хедърите на блоковете се съхраняват в паметта.

8. Опростено удостоверяване на плащането

Възможно да се верифицират плащания и без да се поддържа пълен възел (нод) от мрежата. За целта, потребителят трябва да съхранява копие на блоковите хедъри на най-дългата верига на доказателство за работа. Това той може да получи чрез запитване към мрежовите възли и когато се убеди, че има най-дългата верига, да получи клон от дървото на Merkle, свързващ трансакцията с блока, за който се отнасят същите времеви печати. Така той не може да провери сделката за себе си, но като я свърже с място във веригата, може да види, че мрежовият възел я е приел, блокът я е добавил и потвърдил, а мрежата я е приела.

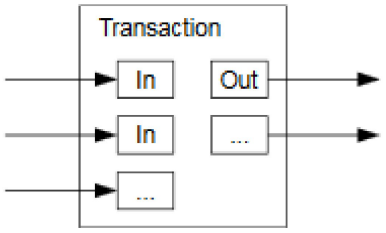


Проверката е надеждна, когато честните възли контролират мрежата, но е по-уязвима, ако мрежата бъде доминирана от нападател. Докато мрежовите възли могат да проверяват трансакциите за себе си, опростеният метод може да бъде заблуден от измислените трансакции на атакуващия, докато той продължава да доминира мрежата. Добра стратегия за защита срещу това е приемането на сигнали от мрежовите възли, когато те открият невалиден блок, принуждавайки софтуера на потребителя да изтегли пълния блок и съмнителните транзакции, за да

потвърди несъответствието. Фирмите, които получават чести плащания, вероятно биха искали да управляват собствените си възли за по-голяма сигурност и по-бърза проверка.

9. Комбиниране и разделяне на стойността

Независимо от това, че е възможно да се обработват монети поотделно, би било трудно да се прави отделна трансакция за всеки цент в трансфера. За да се позволи разделянето и комбинирането на стойността, трансакциите съдържат множество входове и изходи (входяща и изходяща информация). Обикновено има или единичен вход от по-голяма предишна трансакция или множество входове, комбиниращи по-малки суми и най-много два изхода – един за плащане и един за връщане на промяната обратно към подателя.

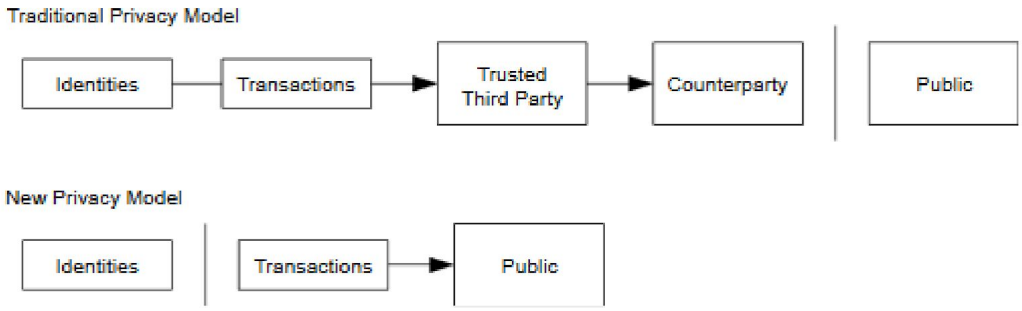


Трябва да се отбележи, че не е проблем фактът, че трансакцията зависи от няколко трансакции, а те съответно зависят от много повече. Никога не е необходимо извличането на напълно самостоятелно копие от историята на трансакцията.

10. Поверителност

Традиционният банков модел постига ниво на поверителност, като ограничава достъпа до информация на участващите страни и доверената трета страна. Необходимостта от публично обявяване на всички трансакции изключва този метод, но поверителността все още

може да бъде поддържана чрез прекъсване на потока от информация на друго място: чрез запазване анонимността на публичните ключове. Публично може да види, че някой изпраща сума на някой друг, но без информация, свързваща трансакцията с някого. Това е подобно на нивото на информация, публикувана от фондовите борси, където времето и размерът на отделните сделки се оповестяват публично, но без да се казва кои са страните.



Като допълнителна защитна стена трябва да се използва нова двойка ключове за всяка трансакция, за да не бъдат свързани с общ собственик. При трансакции с няколко входа, някои от взаимовръзките остават неизбежни. Риск има, ако собственикът на ключ се разкрие и по този начин се заплаши поверителността и на други трансакции.

11. Изчисления

Да разгледаме сценарият за нападател, който се опитва да генерира алтернативна верига, която е по-бърза от честната верига. Дори това да се осъществи, то не оставя системата отворена за произволни промени, като например създаване на измислени стойности или прибиране на чужди пари. Възлите няма да приемат невалидна трансакция като плащане и честните възли никога няма да приемат блок, който ги съдържа. Нападателят може само да се опита да промени някоя от

собствените си трансакции за да вземе обратно пари, които наскоро е похарчил.

Състезанието между честната верига и тази на нападателя може да се характеризира като биномно случайно обхождане (Binomial Random Walk). Състезанието ще е успешно за честната верига, ако тя се увеличи с един блок и ще е провал, ако веригата на нападателя се увеличи с един блок.

Вероятността нападателят да навакса след като е загубил преднината си е аналогична с проблема на хазартния играч, който увеличава залаганията си. Да предположим, че играч с неограничен кредит стартира с дефицит и прави потенциално безкраен брой опити да достигне до печалба. Можем да изчислим вероятността той някога да вземе тази печалба или да надмине честната верига както следва:

p = вероятност частен възел да намери следващия блок;

q = вероятност нападателят да открие следващия блок;

q_z = вероятност нападателят някога да се измъкне и да навакса от блок z

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

Като се има предвид нашето предположение, че $p > q$, вероятността пада прогресивно, тъй като броят на блоковете, които нападателят трябва да навакса се увеличават. Ако той няма късмета да се промъкне напред, шансовете му стават прекалено малки.

Сега разглеждаме колко дълго трябва да изчака получателят на нова трансакция, преди да е достатъчно сигурен, че изпращачът не може да промени сделката. Предполагаме, че подателят е нападател, който иска да накара получателя да повярва, че е получил плащането но това е само за известно време, след което да го прехвърли и да го върне на

самия себе. Получателят ще бъде предупреден за този ход, но подателят се надява, че ще бъде твърде късно за реакция.

Получателят генерира нова двойка ключове и дава публичния ключ на подателя малко преди подписването. Това предотвратява риска подателят да подготви верига от блокове напред, като работи непрекъснато, за да се измъкне напред и да извърши трансакцията в подходящия момент. След изпращането на трансакцията, нечестният подател започва да работи тайно по паралелна верига, съдържаща алтернативна версия на трансакцията си.

Получателят чака, докато трансакцията бива добавена към блок, а z –блоковете се свързват след това. Той не подозира напредъка на нападателя, но, ако се приеме, че честните блокове са взели средно очакваното време за блок, потенциалният прогрес на атакуващия ще бъде Поасоново разпределение с очаквана стойност :

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

За да разберем каква е вероятността нападателят да успее, умножаваме гъстотата на Поасон по всеки прогрес, който той би могъл да направи до момента:

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

Пренарежда се така, че да се избегне сумиране на безкрайна опашка:

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

Преобразуване в C код:

```
#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;

}
```

Изпълнявайки някои резултати, можем да видим, че вероятността пада прогресивно със *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

Решението за P е по-малко от 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. Заключение

Ние предложихме система за електронни трансакции, без да разчитаме на доверие. Започнахме с обичайната рамка от монети, направени от цифрови подписи, която осигурява силен контрол върху собствеността, но не е пълна, без да се намери начин за избягване на двойното харчене. За да предотвратим това, ние предложихме peer-to-peer мрежа, използвайки доказателство за работа, за да запишем публично историята на трансакциите. Така те стават неприложими за нападателите на мрежата, в случай че честните възли контролират повече изчислителна мощност. Мрежата е силна с неструктурираната си простота. Възлите работят едновременно с малка координация. Не е необходимо да бъдат идентифицирани, понеже съобщенията не се насочват към определено място и трябва да се доставят само на базата на доброволно усилие. Възлите могат да напускат и да се връщат в мрежата по желание, като приемат доказателствата за работа като доказателство за това, което се е случило, докато ги е нямало. Те гласуват чрез изчислителната си мощност, като по този начин

изразяват приемането на валидни блокове и работят по разширяването им или отхвърлят невалидните блокове, като откажат да работят по тях. С този консенсусен механизъм могат да се въведат и наложат всички необходими правила и стимули.

Препратки към:

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.

Благодарности на Владислав Драмалиев, Стоян Пепеланов и Пламен Петров за усилията по превода на текста.