比特币：一个点对点电子现金系统

Satoshi Nakamoto

satoshin@gmx.com

[www.bitcoin.org](http://www.bitcoin.org)

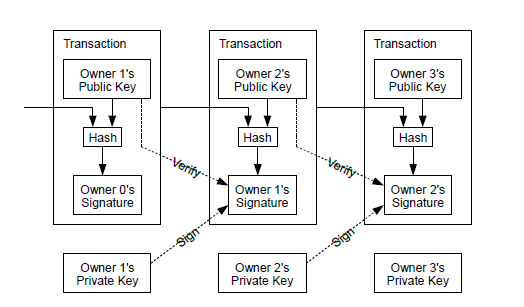
**摘要：**一个真正的点对点(p2p)电子现金系统，一方应当能够直接发起网络支付给另一方，而不需要经过任何金融机构。虽然数字签名技术解决了一部分问题，但是如果仍然需要第三方的支持才能防止双重支付的话，那么这种系统是没有价值的。我们在此对点对点网路下的双重支付问题提出了一种解决办法。该系统会对所有的交易加上时间戳，并通过哈希运算的方式将这些交易并入一个不断延伸的、具有工作量证明的链。该工作量证明也是基于哈希运算。这样以来，除非重新完成全部的工作量证明，否则形成的交易记录将不可更改。最长的链不仅可以视作是交易序列的证明，而且可以被认为是CPU计算能力最大的一方认同的结果。只要大多数的CPU计算力没有打算合作起来对系统进行攻击，那么这些节点生成的链将会是最长的，而且会超过攻击者的链。这个系统只用的很小的基础设施：信息最大化在全网传播；节点可以随时离开或重新加入，只要承认最长的工作量证明链即可。

**1、绪论**

互联网上的贸易，几乎都需要借助金融机构作为可资信赖的第三方来处理电子支付信息。虽然这类系统在绝大多数情况下都运作良好，但是这类系统仍然受限于“基于信用模式”固有的弱点。由于无法实现完全不可逆的交易，因此，金融机构总是不可避免地会协调争端。调停损失会增加交易成本，限制了最小交易规模，进而无法实现日常小额交易，并且，在缺乏信任的环境下实现不可逆交易需要付出更大的成本。由于交易存在取消的可能，实现交易就需要交易双方互相信任。商家会向客户索取完全不必要的个人信息以预防消费者蓄意破坏交易，然而在实际的商业行为中，一定比例的欺诈性客户是不可避免的。这些费用和支付的不确定性在使用物理现金情况下是可以避免的，并且完全无需第三方信用中介的参与。我们需要这样一种电子支付系统，只基于密码学原理而不是信用，就能使得任何达成一致的双方直接进行支付交易，同时不需要第三方中介的参与。杜绝取消交易，就可以保护卖方免于欺诈；也很容易实现常规的托管机制来保护买方的利益。在这篇论文中，我们将提出一种通过点对点分布式的时间戳服务器来生成依照时间前后排列并的交易证明(交易链)，从而解决双重支付问题。只要诚实的节点所控制的计算能力大于任何一个合作攻击者的计算能力，该系统就是安全的。

**2、交易**

我们定义，一枚电子货币是一串数字签名。每一位所有者通过对前一次交易的位置和下一位拥有者的公钥的散列值签署一个数字签名，并将这个签名附加在这枚电子货币的末尾，电子货币就发送给了下一位所有者。收款人通过对签名进行检验，就能够验证该链的所有者。

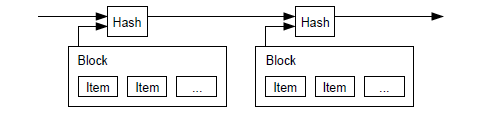


该过程的问题在于，收款人难以检验，付款人是否对这枚电子货币进行了双重支付。通常的解决方案是引入信得过的第三方权威，或者类似于造币厂的机构，来对每一笔交易进行防止双重支付的检验。在每一笔交易结束后，这枚电子货币就要被造币厂回收，同时造币厂将发行一枚新的电子货币；并且只有造币厂直接发行的电子货币，才算作有效，这样就能够有效防止双重支付。可是该解决方案的问题在于，每一笔交易都要经过该造币厂的确认，因此整个货币系统的命运完全依赖于运作造币厂的公司，造币厂就好比是一家银行。

我们需要收款人有某种方法，能够确保付款人在本次交易之前没有对该电子货币进行交易签名。为了实现这个目的，我们只需要关注本交易之前是否进行过支付，而不需要关注本次交易之后有没有进行双重支付。为了确认一次交易是不存在的，那么唯一的方法就是查看之前发生过的所有交易。在造币厂模型里面，造币厂获悉所有的交易，并且决定交易完成的先后顺序。如果想要在电子现金系统中排除第三方信任机构，那么交易信息就应当被公开，并且我们需要一个包含所有参与者都承认的历史交易序列的系统。收款人在交易期间需要验证是否绝大多数的节点都认同该交易是首次出现。

**3、时间戳服务器**

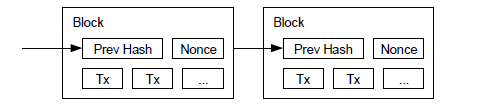
我们提出的解决方案从一个“时间戳服务器”开始的。时间戳服务器会对由一组数据和时间戳组成的区块进行散列，并将该散列进行广播，就像报纸或全球新闻网的传播一样。显然，该时间戳能够证明该组数据在这个时间是确实存在的，因为只有加入了该时间戳才能得到相应的散列值。每个时间戳应当将前一个时间戳纳入其散列值中，像这样，每一个后面的时间戳都对之前的一个时间戳进行增强，这样就形成了一个链。



**4、工作量证明**

为了在点对点的基础上实现一组分布式的时间戳服务器，仅仅像报纸或全球新闻网一样是不够的，我们还需要使用一个类似于亚当•柏克提出的哈希现金的工作量证明系统。在进行散列运算时，工作量证明机制会对某一个特定值进行检测，比方说采用SHA-256算法进行散列，要求散列值以一个或多个0开始。那么随着0的数目的上升, 找到这个解所需要的平均工作量将呈指数增长，但是对结果进行检验则仅需要一次散列运算。

在我们的时间戳服务器组成的网络中，区块中会放置一个随机数，通过不断地调整这个随机数来使得该区块的散列值满足所要求个数的0。一旦耗费的CPU计算力满足了所要求的工作量，那么除非重新完成相同的工作量，否则该区块的信息就是不可更改的。由于之后的区块是链接在该区块之后的，所以想要更改该区块中的信息，就必须重新完成之后所有区块要求的全部工作量。



同时，该工作量证明机制还解决了在集体投票表决时，谁是多数的问题。如果决定多数的方式是基于一个IP地址为一票，那么一旦有人能够分配大量IP地址，该表决机制就会被破坏。工作量证明机制的本质则是一CPU一票。“多数”的决定表现为最长的链，因为最长的链包含了最大的工作量。如果多数的CPU为诚实的节点控制，那么诚实的链将以最快的速度延长，并超越其他的竞争链。如果想要对早已生成的区块进行修改，攻击者必须付出完成该区块及其之后所有区块的工作量，并最终赶上并超越诚实节点提供的工作量。我们将在后文证明，一个较慢的攻击者试图超过随后所有的区块，其成功概率将呈指数化减小。

为了解决高速增长的硬件运算速度和参与节点数量的变动带来的实际工作量提供能力的差异，工作量证明的难度将采用移动平均方法来确定，即每小时生成区块的速度为某一个预定的平均数的难度大小。如果区块生成的速度过快，那么难度就会提高。

**5、网络**

运行该网络的步骤如下：

1) 新的交易向全网进行散布

2) 每一个节点都将收到的交易信息纳入一个区块中

3) 每个节点都尝试在自己的区块中找到一个具有足够难度的工作量证明

4) 当一个节点找到了一个工作量证明，它就向全网进行传送

5) 当且仅当包含在该区块中的所有交易都是有效的且之前并未消费过，其他节点才认同该区块的有效性

6) 若其他节点接受该区块，则会在该区块的末尾，使用该区块的散列值制造新的区块延长该链

节点始终都将最长的链视为正确的链，并持续工作和延长它。如果有两个节点同时传送不同版本的新区块，那么其他节点在接收到两个区块的时间上将存在先后差别。在此情形下，他们将在率先收到区块的基础上进行工作，但也会保留另外一个区块，以防后者变成最长的链。该僵局会在下一个工作量证明时打破，而其中的一条链会被证明是较长的一条，那么在另一条分支链上工作的节点将转换阵营，开始在这条较长的链上进行工作。

新的交易并不需要发送至全部的节点。只要交易信息能够撒布至足够多的节点，这些交易信息就会很快被整合进一个区块中。区块的撒布对被丢弃的信息是有容错能力的。如果一个节点没有收到某特定区块，那么该节点将会发现缺失了某个区块，就会向其他节点提出下载该区块的请求。

**6、激励**

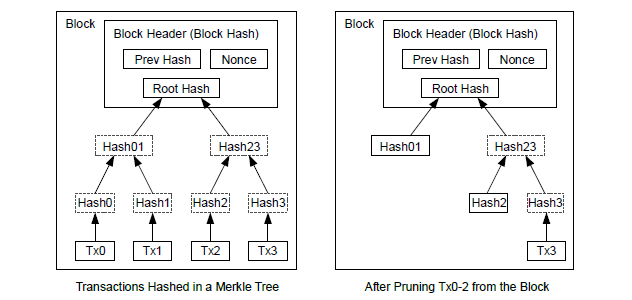
我们约定：每个区块的第一个交易都是一笔特殊的交易，该交易产生一枚由该区块创造者拥有的新的电子货币。这样就为了节点支持该网络的进行了激励，在没有中央集权机构发行货币的情况下，这种方式提供了一种将电子货币分配到流通领域的方法。这种将一定数量新货币持续增添到货币系统中的方法，非常类似于耗费资源去挖掘金矿并将黄金注入到流通领域。在我们的系统中，CPU的运算时间和电力就是消耗的资源。

另外一个激励的来源则是交易费。如果某笔交易的输出值小于输入值，那么差额就是交易费，该交易费将被增加到该区块的激励中。只要既定数量的电子货币已经进入流通，那么激励机制就可以逐渐转换为完全依靠交易费，并且该货币系统能够免于通货膨胀。

激励系统也有助于鼓励节点保持诚实。如果有一个贪婪的攻击者能够调集比所有诚实节点加起来还要多的CPU计算力，那么他就面临一个选择：要么将其用于诚实工作产生新的电子货币，或者将其用于进行双重支付攻击。那么他就会发现，按照规则行事、诚实工作是更有利可图的。因为该规则使得他能够拥有更多的电子货币，而不是破坏这个系统使得其自身财富的有效性受损。

**7、回收硬盘空间**

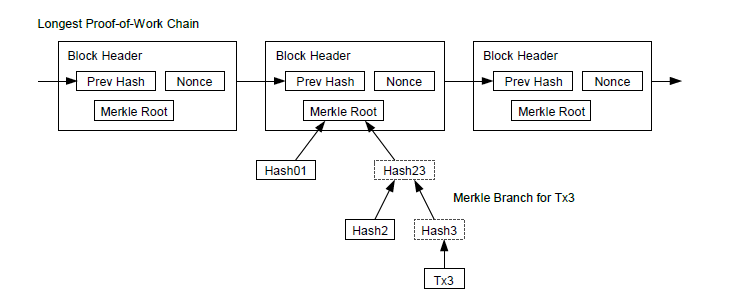
如果最近的交易已经被纳入了足够多的区块之中，那么就可以丢弃该交易之前的数据，以回收硬盘空间。为了确保不破坏区块的散列值，交易信息在被散列时，会构建成一种默克尔树的形态，使得只有根(root)被纳入了区块。通过将该树的分支拔除的方法，老区块就能被压缩。区块内部的散列值是不必保存的。



不含交易信息的区块头（Block header）仅有80字节。如果我们设定区块生成的速率为每10分钟一个，那么每一年产生数据的大小为4.2MB(80 bytes \* 6 \* 24 \* 365 = 4.2MB)。2008年，PC系统通常的内存容量为2GB，按照摩尔定律的预言，即使将全部的区块头存储于内存之中都不是问题。

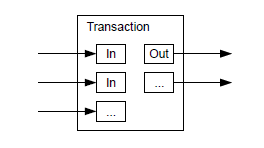
**8、简化的支付确认**

在不运行完整网络节点的情况下，也能够对支付进行检验。一个用户需要保留最长工作量证明链区块头的拷贝(通过不断向网络发起询问，直到能确认拥有最长的链)，得到能连接交易及包含该交易的块的默克尔分支。节点想要自行检验交易的有效性是不可能的，但通过检查该交易在链上的位置，就能确认某个节点已经接受了交易，并且于其后追加的区块都会进一步证明全网已经接受了该交易。



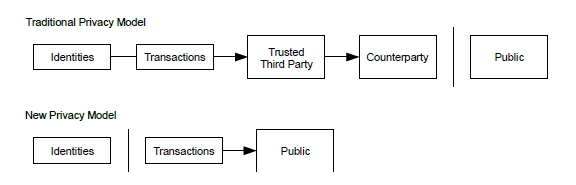
只要诚实的节点控制了网络，检验机制就是可靠的。但是，当攻击者的计算力占优时，该验证机制会变得较为脆弱。在网络节点能够自行确认交易的有效性时，只要攻击者能够持续地保持计算力优势，简化的验证机制会被攻击者伪造的交易欺骗。一个可行的策略是，一旦发现了一个无效的区块，就立刻发出警报，收到警报的用户将立刻开始下载被警告有问题的区块，以便确认信息的不一致性。对于会发生大量日常交易的商业机构，可能会希望运行他们自己的完整节点，以保持较大的独立安全和更快的校验速度。

**9、价值的合并与分割**

虽然可以单独的对电子货币进行交易，但是将交易分割然后一枚一枚进行是笨拙的办法。为了使得货币易于合并或分割，交易被设计为允许多个输入和输出。一般而言之前某次金额较大的交易构成的单一输入，或者由几次金额较小的交易共同构成的并行输入，但是输出最多只有两个：一个用于支付，另一个用于找零（如有）。

需要指出的是，当一笔交易依赖于之前的多笔交易时，这些交易又各自依赖于多笔交易，这是正常的。这些交易依赖并不意味着需要从交易记录中抽取个副本出来。

**10、隐私**

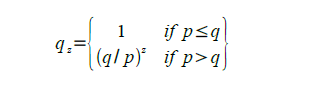
传统的银行模型为交易的参与者提供了一定程度的隐私保护，因为试图向可信任的第三方索取交易信息是受限制的。如果将交易信息向全网进行广播，那么就无法通过这个方法来保护隐私。但是通过隐藏交易信息的某个环节——公钥匿名化，隐私就仍然是可以得到保护的。公众得知的信息仅仅是有某个人将一定数量的货币支付给了另外一个人，但是难以将该交易同特定的人联系在一起。这和股票交易发布的信息是类似的，股票交易的时间和数量是可供查询的，但是无法获知具体是“谁”在进行交易。

作为额外的保护措施，使用者可以让每次交易都生成一个新的地址，以确保这些交易不被追溯到一个共同的所有者。由于并行输入的存在，一定程度上的追溯还是不可避免的，因为并行输入表明这些货币属于同一个所有者。此时的风险在于，如果某人的公钥暴露了，那么就可以借此公钥追溯出此人的其它交易。

**11、相关计算**

设想如下场景：一个攻击者拥有更快的计算力并且试图生成另一条链来代替诚实节点生成的链。即便它达到了这一目的，也不意味着整个系统就能被攻击者完全控制，比方说凭空创造价值(货币)，或者使用任何不属于攻击者的货币。这是因为节点不会接受无效的交易，并且诚实的节点也永远不会接受一个包含无效信息的区块。一个攻击者能做的，最多是更改他自己的交易信息，例如拿回他刚刚付给别人的钱。诚实链和攻击链之间的竞赛，可以用二叉随机漫步来描述。成功事件定义为诚实的链延长了一个区块，其领先性+1，而失败事件则是攻击者的链被延长了一个区块，使得差距-1。攻击者弥补某一既定差距的概率，可以近似地看作赌徒破产问题。假定一个赌徒拥有无限的透支信用，然后开始进行可能无数次的赌博，直到填补上自己的亏损。那么我们可以计算赌徒填补上亏损的概率，也就是攻击者赶上诚实节点链的概率，如下所示：

**p** = **probability an honest node finds the next block 诚实节点发现下一个块的概率**  
**q** = **probability the attacker finds the next block 攻击者发现下一个块的概率**  
**qz**= **probability the attacker will ever catch up from z blocks behind 攻击者消除z个区块差距的概率**



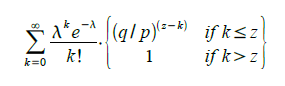
假定p>q，那么攻击成功的概率就因为区块数z的增长而呈现指数化下降。由于攻击者找到下一个块的概率小，倘若攻击者不能幸运且快速地获得成功，那么他获得成功的机会随着时间的流逝就变得越来越小。

现在讨论一个收款人究竟需要等待多长时间，才能确信付款人已经难以改变交易。我们假设付款人是一个攻击者，他希望能够让收款人在一段时间内相信他已经付过款了，然后建立新的分支块将支付的款项重新支付给自己。虽然收款人之后会发现这一点，但是等到发现时收款人已经什么都做不了了。

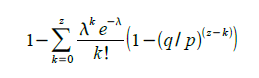
收款人生成了一个新的密钥对，然后在签名之前快速地将公钥发给付款人。这样可以防止以下情况：付款人预先准备好一个区块然后不断地对此区块进行运算，直到运气让他的区块链超越了诚实的链，然后付款人才执行支付。在此情形下，只要交易一旦发出，攻击者就开始秘密地准备一条包含了他自己交易的平行链来替代刚刚支付出去的链。

收款人会等待交易出现在一个区块中，并且紧随其后有z个区块。此时，收款人仍然不能确切知道攻击者已经进展了多少个区块，假设诚实区块将耗费平均预期时间以产生一个区块，那么攻击者的潜在进展服从泊松分布，期期望值为：

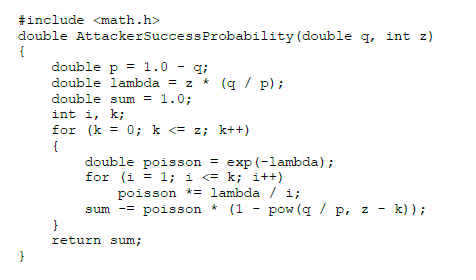
攻击者追赶上的概率为，一定进展区块数量下的泊松概率密度，乘以在该数量下依然能够追赶上的概率。



化为如下形式，避免对无限数列求和：



写为如下C语言代码：



对其进行运算，我们可以得到如下的结果，发现概率对z值呈指数下降。

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

求解令P<0.1%时，z的值：

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、结论**

我们提出了一种不需要信用中介的电子支付系统。首先讨论了通常的电子货币的电子签名原理，虽然这种系统为所有权提供了强有力的控制，但是不足以防止双重支付。为了解决这个问题，我们提出了一种采用工作量证明机制的点对点网络来记录公开的交易信息，只要诚实的节点能够控制绝大多数的CPU计算力，就能使得攻击者难以改变交易记录。该网络的强健之处在于它非结构化简洁性。节点的工作大部分是彼此独立的，只需要很少的协同。每个节点都不需要明确自己的身份，由于交易信息的流动路径并无任何要求，所以只需要尽其最大努力传播即可。节点可以随时离开网络，而想重新加入网络也非常容易，只需要接受离开期间的工作量证明链即可。节点通过自己的CPU计算力进行投票，它们不断延长有效的区块链来表达自己的确认，并拒绝在无效的区块之后延长区块以表示拒绝。这套交易系统共识机制包含所有需要的规则和激励措施。

**参考文献**

[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.