比特币：点对点电子现金系统

中本聪

Satoshi Nakamoto

satoshin @ gmx. com

www.bitcoin.org

摘要：一个纯粹点对点版本的电子现金系统将允许在线支付直接从一方发送到另一方，从而无需通过金融机构。数字签名提供了解决方案的一部分，但是它的主要好处会丢失，如果受信任的第三方仍然需要防止双重支出。我们提出了一种使用点对点网络解决双重支出问题的方案。该网络通过随机散列对全部交易加上时间戳，将它们合并入一个不断延伸的基于随机散列的工作量证明的链条作为交易记录，除非重新完成全部的工作量证明，形成的交易记录将不可更改。最长的链不仅可以证明所见证事件的顺序，而且还可以证明它来自最大的 CPU 能力。只要大多数 CPU 能力是由不会合作攻击网络的节点控制的，它们将产生最长的链并超约攻击者。网络本身需要最少的结构。消息以尽力而为的方式传播，节点可以随意离开并重新加入网络，同时会接受最长的工作量证明链作为消息消失时发生的事情的证据。

1. 引言

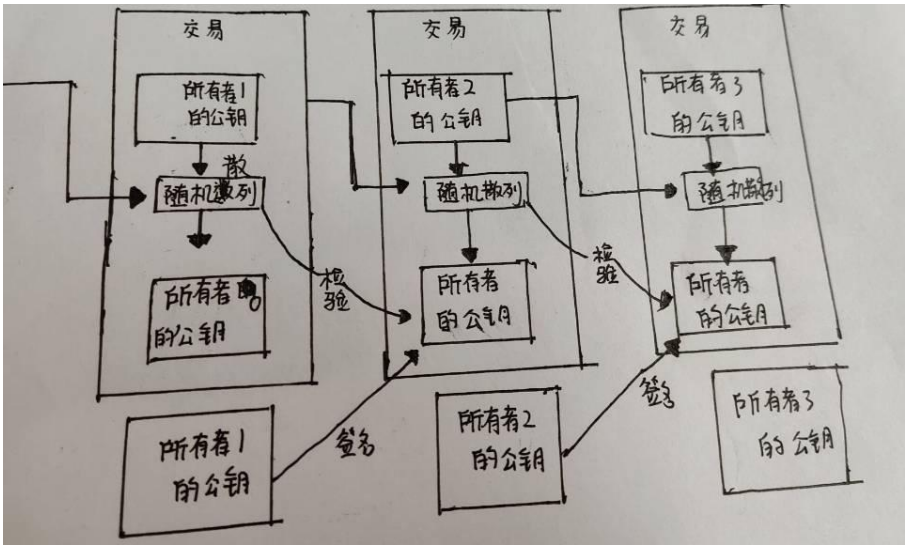
互联网上的贸易几乎都需要借助金融机构作为可资信赖的第三方来处理电子支付信息。虽然在绝大多数情况下这类系统都运作良好，但是它仍然内生性地受制于“基于信用的模式”的弱点。完全不可逆的交易是几乎不可能实现的，因为金融机构不能避免出头调解纠纷。这些调解纠纷的话费增加了交易成本，同时限制了最小实际可行的交易规模，还减少了日常的小额支付交易发生的可能性。同时更大的潜在代价是大多不可逆服务都失去了不可逆付款能力。如果缺乏不可逆的支付手段，互联网的贸易就大大受限。由于有了逆转的可能性，信任的需求就会蔓延开来。此外家也必须提防自己的客户，因此他们会向客户索取完全不必要的个人信息。一定比例的欺诈也是不可避免的。这些费用和付款的不确定性可以通过亲自使用实物货币来避免，但不存在付款机制能在没有受信方的情况下通过通信渠道运行。

所以，我们真的非常需要一种电子支付系统，它基于密码学原理而不基于信用，允许任何达成一致的双方能够直接进行支付，从而不需要第三方中介的参与。在计算上不可行进行逆转的交易将保护卖方避免欺诈，常规的托管机制可以轻松

实施以保护购买者。在时间戳服务器中生成事务按时间顺序的计算证明。只要诚实节点共同控制的 CPU 能力超过其他任何 CPU 能力，系统便是安全的。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加以记录的电子交易证明，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的总和，大于有合作关系的攻击者的计算能力的总和，该系统就是安全的。

2.交易

我们这样定义，一个电子货币是这种一串电子签名，每一名拥有者将电子货币发给下一位所有者要通过对前一次交易和下一位拥有者的公钥签署一个随机散列的数字签名，并把他们附加在这枚电子货币的尾部。然后收款人可以对签名进行检验，进而能检验出这个链条的所有者。

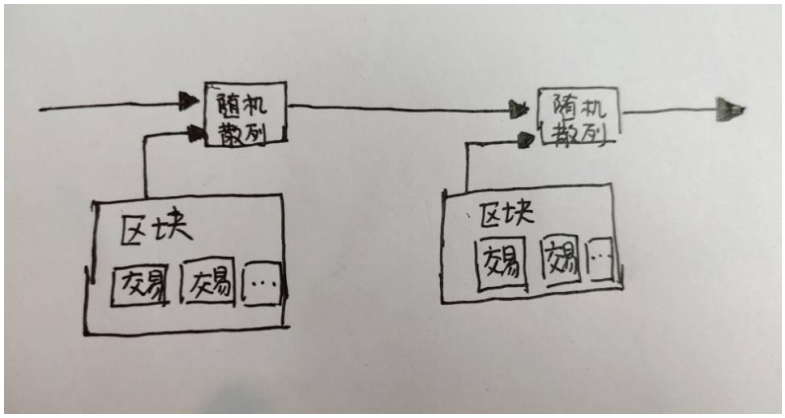


但是这个过程的问题在于收款人不能检验之前是否有一位所有者对这枚电子货币进行了双重支付。一个平常的解决方案是请一位值得信赖的第三方权威，或者功能像造币厂的机构，从而检验每一次进行的交易，目的是防止双重支付的发生。每次交易结束后，电子货币将被造币厂回收，造币厂将发行新的电子货币；只有由铸币局直接发行的电子货币才被认为有效。可以防止重复付款。但这个解决方案的问题是整个货币系统的命运完全取决于经营造币厂的公司，因为每笔交易都必须经过造币厂的确认，而造币厂就像银行一样。

我们需要一种方式为收款人确保之前的所有者没有签署的交易发生在更早。从逻辑的角度来看，为了达到这个目标，其实我们需要注意的只是在交易之前发生的交易，而不需要注意交易之后是否会有双重支付的尝试。为了确保某个事务不存在，唯一的方法是了解之前发生的所有事务。在铸币厂模型中，铸币厂学习所有交易并决定订单并决定了交易完成的顺序。如果你想把第三方中介排除在电子系统之外，那么交易信息应该公开公布。我们需要整个系统中的所有参与者都具有唯一可识别的历史事务序列。收款人需要确保在交易期间，绝大多数节点同意交易是第一次发生。

3.时间戳服务器

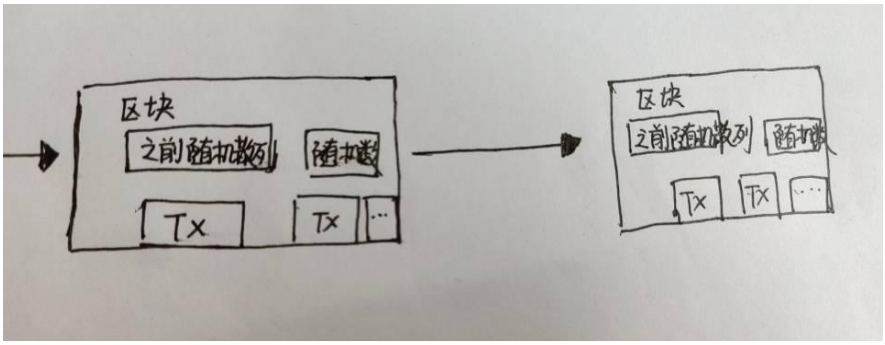
该解决方案首先提出了“时间戳服务器”。 时间戳服务器以块的形式对一组数据的随机散列进行时间戳记，并广播该随机散列，就好像它是新闻或全球新闻组网络上的帖子一样[2-5]。 显然，这个时间戳可以确认某个数据在某个时间一定存在，因为只有那个时候才能得到对应的随机哈希值。每个时间戳应该在其随机哈希值中包含前一个时间戳，每个后续时间戳都增强前一个时间戳，从而形成一个链。



4. 工作证明

为了在点对点的基础上构建一组去中心化的时间戳服务器，仅仅像报纸或全球新闻网络组一样工作是不够的。 我们还需要类似于亚当·柏克提出的哈希现金[6]。在进行随机散列运算时，工作量证明机制引入了对某一个特定值的扫描工作，比方说 SHA-256 下，随机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工作量将呈指数增长，但是检验结果仅需要一次随机散列运算。

我们向区块添加一个随机数，以便给定区块的随机哈希值具有所需数量的零。我们通过反复试验找到这个随机数，直到找到为止。所以我们建立了一个工作量证明机制。只要 CPU 消耗的工作量足以满足工作量证明机制，除非重做大量工作，否则无法更改区块信息。由于后续块链接在该块之后，因此要更改该块中的信息需要为所有后续块重做全部工作。



同时，工作量证明机制还解决了在小组投票中谁占多数的问题。如果大多数决策都是基于 IP 地址做出的，一个 IP 地址一票，那么如果有人有权分配大量 IP 地址，这个机制就会被打破。工作量证明的本质是一个 CPU，一票。“大多数”

决策表示为最长的链，因为最长的链包含最多的工作量。如果大多数 CPU 由诚实节点控制，那么诚实链将增长最快并超过其他竞争链。如果想要对业已出现的区块进行修改，攻击者必须重新完成该区块的工作量外加该区块之后所有区块的工作量，并最终赶上和超越诚实节点的工作量。我们将在后文证明，设想一个较慢的攻击者试图赶上随后的区块，那么其成功概率将呈指数化递减。

另一个问题是，硬件的运算速度在高速增长，且节点参与网络的程度会有所起伏。为了解决这个问题，工作量证明的难度将采用移动平均目标的方法来确定，即令难度指向令每小时生成区块的速度为某一预设的平均数。如果区块生成的速度过快，那么难度就会提高。

5. 网络

运行网络的步骤如下：

1) 全网广播新交易；
2）每个节点将接收到的交易信息放入一个区块中；
3）每个节点都试图在自己的区块中找到一个具有足够难度的工作量证明；
4）当一个节点找到工作量证明时，向全网广播；
5) 只有当且仅当该区块中包含的所有交易均有效且之前不存在时，其他节点才会承认该区块的有效性；
6) 其他节点通过跟随块的末尾，创建新块以扩展链，并将接受块的随机散列视为比新块的随机散列更快来表示它们接受该块。

节点总是认为最长的链是正确的链，并继续工作和扩展它。 如果两个节点同时广播一个新区块的不同版本，其他节点将在不同时间收到该区块。 发生这种情况时，他们会处理收到的第一个区块，但保留另一条链，以防它成为最长的链。 僵局被打破，直到找到下一个工作证明，并且当一个链条被证明是更长的链条时，在另一个分支链上工作的节点切换到集中营并开始在更长的链上工作。

所谓“新的交易要广播”，实际上不需要抵达全部的节点。只要交易信息能够抵达足够多的节点，那么他们将很快被整合进一个区块中。而区块的广播对被丢弃的信息是具有容错能力的。如果一个节点没有收到某特定区块，那么该节点将会发现自己缺失了某个区块，也就可以提出自己下载该区块的请求。

6.激励机制

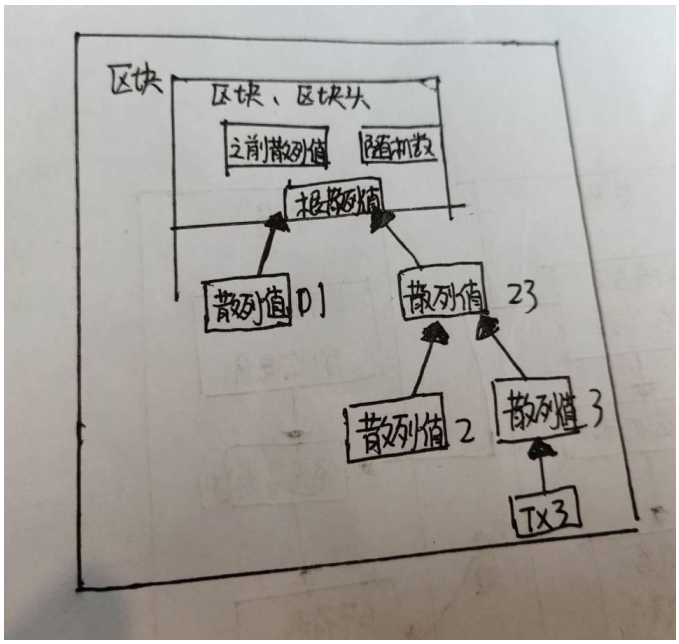
我们同意每个区块中的第一笔交易将是专门的，从而产生由该区块的创建者拥有的新数字硬币。这增加了节点支持网络的动力，并提供了一种向流通领域分发电子货币的手段，而无需集中发行货币的权限。这种向货币体系中不断增加一定数量的新货币的方法与利用资源开采黄金并将其注入流通非常相似。在这种情况下，CPU 时间和功耗是消耗的资源。

另一个激励来源是交易费。如果特定交易的输出值小于输入值，则差额为交易费用，该费用将添加到该区块的激励中。只要一定数量的电子货币在流通，激励机制就可以逐渐转变为完全依赖交易费用，货币体系可以免受通货膨胀的影响。

激励系统也有助于鼓励节点诚实。如果一个贪婪的攻击者可以召集比所有诚实节点加起来更多的 CPU 能力，他有一个选择：要么用它来诚实地工作以生成新的数字货币，要么用它来进行二次支付攻击。 然后他会发现遵守规则和诚实工作更有利可图。这样的规则允许他拥有更多的电子货币，而不是破坏系统和他自己财富的有效性。

7. 回收硬盘空间

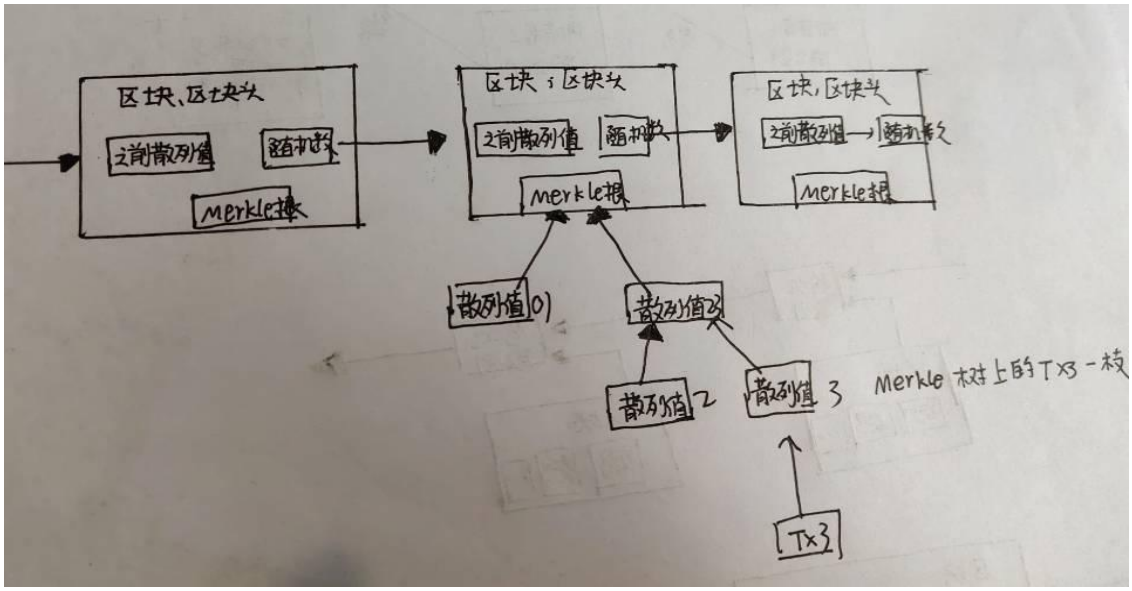
如果最近的事务已包含在足够多的块中，则可以丢弃该事务之前的数据以回收磁盘空间。 为保证区块的随机哈希值不会同时被泄露，当交易信息被随机哈希时，将其构造成一棵默克尔树[7]，使得该区块的随机哈希值中只包含根堵塞。通过拔出树枝，可以压缩旧块。内部随机哈希值不需要保存



没有交易信息的区块头大小只有 80 字节。如果将块生成速率设置为每 10 分钟 1 个，则每年将生成 4.2MB 数据位。（80 字节 ＊ 6 ＊ 24 ＊ 365 = 4.2MB）。2008 年，PC 系统通常有 2GB 的内存，摩尔定律预测将所有块头存储在内存中不会成为问题。

8. 简化的支付确认

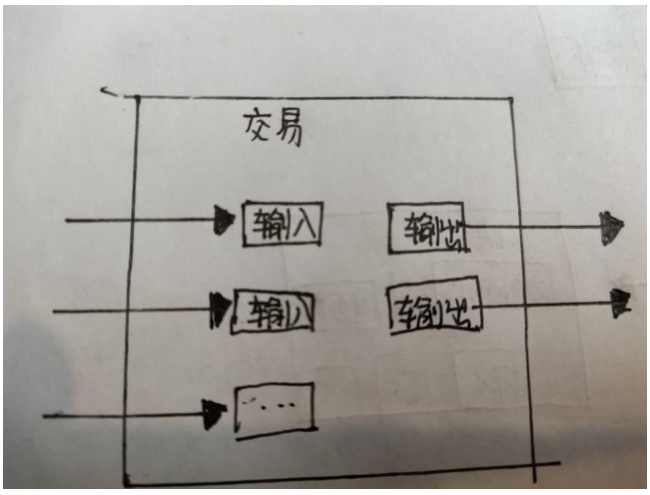
在不运行完整网络节点的情况下，也能够对支付进行检验。用户需要保留最长工作量证明链的块头副本，并且可以不断询问网络，直到确信它拥有最长的链并且可以通过默克尔分支到达它所在的交易 带有时间戳并包含在块中。 节点本身无法验证交易，但是通过追溯到链中的某个点，它可以看到一个节点已经接受了它，随后的区块进一步证明了整个网络已经接受了它。



在这种情况下，只要诚实节点控制网络，验证机制就是可靠的。 然而，当整个网络受到计算上占优势的攻击者的攻击时，它变得更加脆弱。 由于网络节点可以自行确认交易的有效性，只要攻击者能够持续保持算力优势，简化的机制就可以被攻击者的焊接交易所愚弄。 一种可能的策略是一旦发现无效区块就发送警报，收到警报的用户将立即开始下载他们收到警报的区块或交易的完整信息，以确定不一致之处。对于每天有大量支出的企业，他们可能仍然希望运行自己的全节点，以保持更大的独立完整性和验证速度。

9. 价值的组合与分割

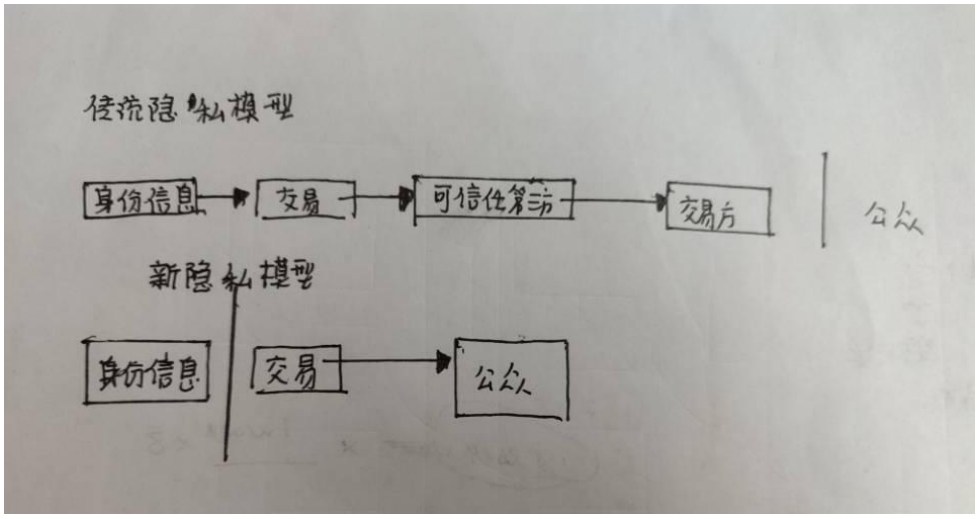
尽管可以单独处理电子货币，但单独为每个电子货币启动交易将是一种笨拙的方式。 为了使价值易于组合和拆分，交易被设计为包含多个输入和输出。 一般来说，它是由一个较大价值的先前交易组成的单个输入，或者由先前具有较小价值的几个交易组成的并行输入，但最多只有两个输出：一个用于支付，另一个用于找零（如果有的话，送还给交易人）。



需要指出的是，虽然一个事务依赖于多个先前的事务，而这些事务每个又依赖于多个事务，但没有问题。 因为这个工作机制不需要开始检查之前发生的所有交易历史。

10. 隐私

因为从受信任的第三方获取交易信息的尝试受到严格限制。但是如果交易信息被广播到全网，就意味着这样的方法是无效的。但是隐私仍然可以得到保护：保持公钥匿名。公众知道的信息只是某人向另一个人发送了一定数量的货币，但很难将交易与特定的人联系起来。换句话说，公众很难确定这些人是谁。这与证券交易所发布的信息类似。记录了每次股票交易的时间和数量，可供查询，但未披露交易各方的身份信息。



作为额外的预防措施，用户可以让每笔交易生成一个新地址，以确保这些交易不会追溯到共同所有者。但是，由于平行输入，一定程度的溯源仍然是不可避免的，因为平行输入意味着这些货币都属于同一个所有者。此时的风险在于，如果某个人的某一个公钥被确认属于他，那么就可以追溯出此人的其它很多交易。

11. 计算

设想如下场景：一个攻击者试图比诚实节点产生链条更快地制造替代性区块链。即便它达到了这一目的，但是整个系统也并非就此完全受制于攻击者的独断意志了，比方说凭空创造价值，或者掠夺本不属于攻击者的货币。这是因为节点将不会接受无效的交易，而诚实的节点永远不会接受一个包含了无效信息的区块。一个攻击者能做的，最多是更改他自己的交易信息，并试图拿回他刚刚付给别人的钱。

诚实链条和攻击者链条之间的竞赛，可以用二叉树随机漫步来描述。成功事件定义为诚实链条延长了一个区块，使其领先性+1，而失败事件则是攻击者的链条被延长了一个区块，使得差距-1。

攻击者成功填补某一既定差距的可能性，可以近似地看做赌徒破产问题。假定一个赌徒拥有无限的透支信用，然后开始进行潜在次数为无穷的赌博，试图填补上自己的亏空。那么我们可以计算他填补上亏空的概率，也就是该攻击者赶上诚实链条，如下所示[8]：

p=诚实节点制造出下一个节点的概率

q=攻击者制造出下一个节点的概率

qz=攻击者最终消弭了 z 个区块的落后差

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

假定 p>q，那么攻击成功的概率就因为区块数的增长而呈现指数化下降。由于概率是攻击者的敌人，如果他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愈发渺茫。那么我们考虑一个收款人需要等待多长时间，才能足够确信付款人已经难以更改交易了。我们假设付款人是一个支付攻击者，希望让收款人在一段时间内相信他已经付过款了，然后立即将支付的款项重新支付给自己。虽然收款人届时会发现这一点，但为时已晚。

收款人生成了新的一对密钥组合，然后只预留一个较短的时间将公钥发送给付款人。这将可以防止以下情况：付款人预先准备好一个区块链然后持续地对此区块进行运算，直到运气让他的区块链超越了诚实链条，方才立即执行支付。当此情形，只要交易一旦发出，攻击者就开始秘密地准备一条包含了该交易替代版本的平行链条。

然后收款人将等待交易出现在首个区块中，然后在等到 z 个区块链接其后。此时，他仍然不能确切知道攻击者已经进展了多少个区块，但是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就是一个泊松分布，

分布的期望值为：$\lambda=z\frac{q}{p}$

当此情形，为了计算攻击者追赶上的概率，我们将攻击者取得进展区块数量的泊松分布的概率密度，乘以在该数量下攻击者依然能够追赶上的概率。

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

化为如下形式，避免对无限数列求和：

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

写为如下 C 语言代码：

```
include<math.h>
doubleAttackerSuccessProbability(doubleq,intz)
```

```
{
doublep=1.0-q;
doublelambda=z*(q/p);
doublesum=1.0;
inti,k;
for(k=0;k<=z;k++)
{
doublepoisson=exp(-lambda);
for(i=1;i<=k;i++)
poisson*=lambda/i;
sum-=poisson*(1-pow(q/p,z-k));
}
returnsum;
}
```

对其进行运算，我们可以得到如下的概率结果，发现概率对 z 值呈指数下降。

当 q=0.1 时

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

当 q=0.3 时

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%的 z 值：

为使 P<0.001，则
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

本文提出一种不需要信用中介的电子支付系统。首先，讨论从数字签名的一般框架开始。 尽管该系统对电子货币的所有权提供了强有力的控制，但不足以防止双重支付。

为了解决这个问题，我们提出了一种点对点网络，该网络使用工作量证明机制来记录交易的公共信息。只要诚实节点可以控制大部分 CPU 算力，就可以使攻击者难以更改交易记录。网络的优势在于其结构简单。节点之间的大部分工作是相互独立的。在 p2p 网络中，每个节点不需要验证自己的身份。由于对交易信息的流动路径没有要求，所以只需要尽力传播即可。节点可以随时离开网络，重新加入网络非常容易，因为它只需要在离开期间补充工作量证明链节点利用自己的 CPU 算力进行投票，投票支持其对有效区块的确认，并继续扩展有效区块链以表达其确认。任何所需的规则和激励措施都可以通过这种共识机制来实施。

参考文献

[1] 戴维，“b-money”，http://www.weidai.com/bmoney.txt，1998

[2] H. Massias, X.S.Av 和 J.-J. Quisquater，“设计一个安全的时间戳服务，使用最少的信任要求”，在比荷卢经济联盟第 20 届信息论研讨会上，1999 年 5 月。

[3] S. Haber, W.S. Stornetta，“如何为数字文档添加时间戳”，《密码学杂志》，第 3 卷，第 2 期，第 99-111 页，1991 年。

[4] D.拜耳，S.哈伯，W.S. Stornetta，“提高数字时间戳的效率和可靠性”，序列 II：通信、安全和计算机科学方法，第 329-334 页，1993 年。

[5] S. Haber, W.S. Stornetta，“位串的安全名称”，第 4 届 ACM 计算机和通信安全会议论文集，第 28-35 页，1997 年 4 月。

[6]A.Back,“哈希现金-拒绝服务对策”
http://www.hashcash.org/papers/hashcash.pdf，2002 年。

[7] R.C. Merkle，“公钥密码系统协议”， 1980 年安全与隐私研讨会会刊，IEEE 计算机协会，第 122-133 页，1980 年 4 月。

[8] W. Feller，“概率论及其应用简介”，1957 年。