《比特币：点对点的电子货币系统》

摘要：一个真正的p2p电子货币，可以允许在线支付不通过金融机构就可以直接从一方到另一方。数字签名技术虽然提供了部分解决方法，但如果还需要一个受信任的第三方来阻止电子现金“双重支付”，那么这种支付方式最大的好处就没了。我们针对“双重支付”问题，采用点对点的网络技术提出解决方案。该网络通过给交易记录打上时间戳，经过哈希加密，将其并入一个不断增长的哈希数列所组成的链条文件中，以此形成一个新的不能被更改的交易记录，这个长链条是事件结果的证据，由最大的CPU处理器生成。只要计算机节点控制的主要CPU处理器没有被联合去攻击网络，就会生成最长的链，组织攻击。网络本身只需要最简单的结构。方便信息尽可能快的传播给节点，而节点接受信息离开时产生的链条文件就可以加入或离开网络。

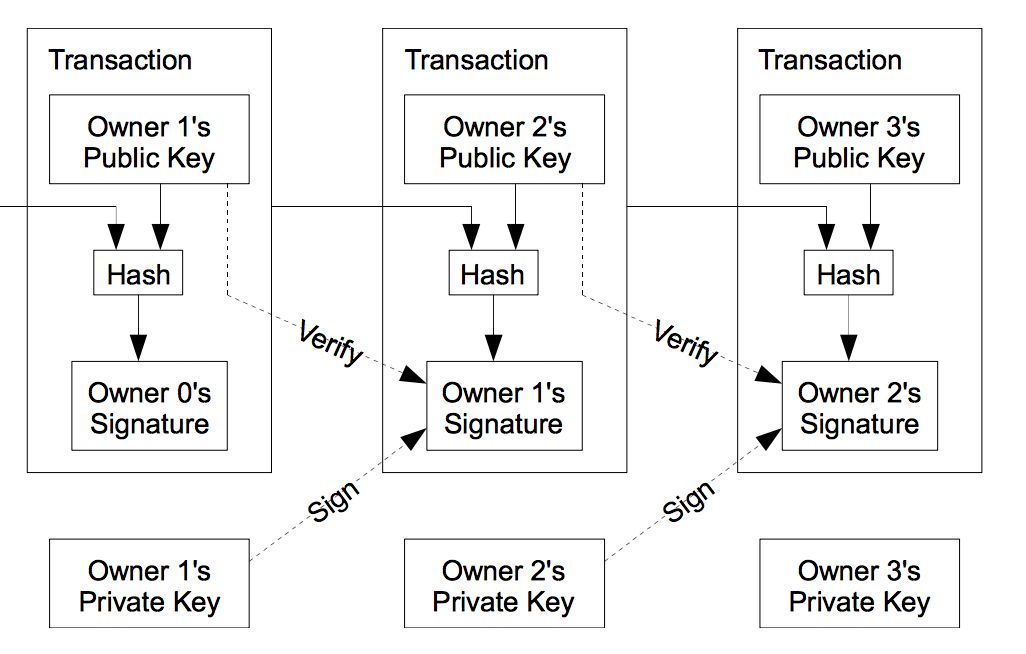
1.简介

网络商务几乎全部都已经发展为依赖受信任的第三方金融机构实现电子支付的阶段。虽然该系统在大多数交易中已经很好的发挥了作用，但是仍继承了基础信用模型的缺憾。自从金融机构参与争议调节，完全不可撤销的交易就不可能实现。调节成本增加了交易成本，限制了实际交易的最小规模，切断了为日常小额交易提供服务的可能性，这样广的成本使不可撤销的服务丧失了不可撤销支付的能力。在可撤销的情况下，信用被广泛需要。商家要警惕他们的客户，骚扰客户以得到更多客户不再需要的信息。一定比例的交易欺诈被接受是不可避免的。使用实体货币虽然可以避免这种成本和支付的不确定性，但不通过可信任的第三方机构沟通，没有商家会提前支付。

所以需要一个基于加密证明的电子支付系统代替信用模型，允许任意自愿的双方在没有受信第三方的情况下交易。计算过地不能实施的交易会撤销，以保护卖方不被欺诈；常规附条件的机械化的合同执行，也很容易的保护了买方。本文提供了一个基于点对点的分布时间戳服务器生成基于时间序列的交易订单的计算证明，用来解决双重支付问题。只要诚实的节点比联合攻击节点控制更多的CPU，系统就是安全的。

2.交易

首先定义一个电子货币作为一个一串数字签名的链。每一个币的拥有者通过哈西技术对上一个交易信息和下一个拥有者的公共密钥进行加密，然后数字签名，最后将这些信息添加到货币末尾。收款人可以用数字签名证明链的所有权。

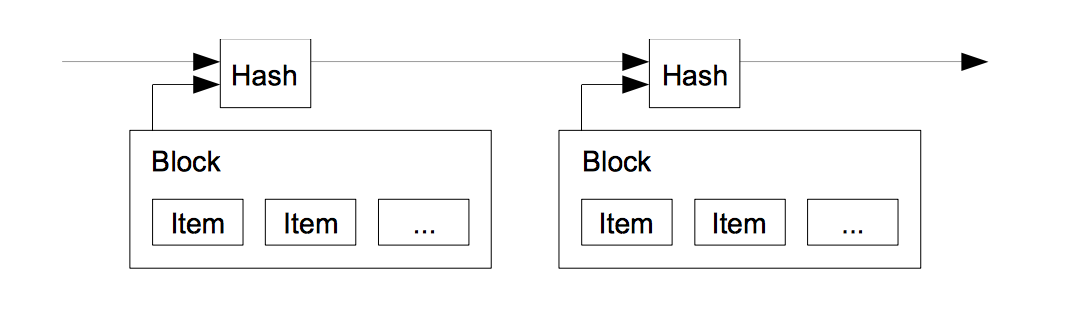


问题在于收款人不能证明币的拥有者是否对该币双重支付。通常引入一个受信权威机构或造币厂检查所有交易。每次交易后，币都必须返回造币厂，然后发行新币，只有直接从造币厂发行的币才会被认为是没有双重支付的。但这样也使得整个货币系统的命运全靠公司运营的造币厂，就像银行一样，每笔交易都不得不经过它。

我们需要一个方案让收款人知道上一个拥有者没有在之前的任何交易中签名以致双重支付。我们意图计算以前的交易，不需要考虑之后的交易是否有双重支付。确认交易存在的唯一方法是知道之前所有的交易。基于造纸厂模型，造纸厂知道所有的交易且能决定第一个到达交易是哪笔。在没有受信机构的情况下完成交易，必须向公众公开，我们需要一个系统的每个参与者都认同曾经经手的单一历史订单。收款人需要通过主要节点认同他们已经第一时间收到了这笔交易，用来证明每一笔交易。

1. 时间戳服务器

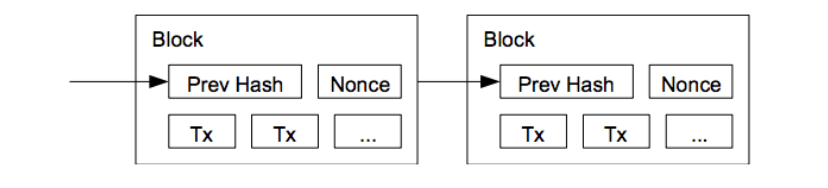
解决方案从时间戳服务器开始，一个时间戳服务器对一组被时间戳标识过的数据块进行哈西加密，然后公开哈西，就像新闻或者论坛发帖一样。显然，时间戳证明的数据在该时间点上必定存在。哈西中的每个时间戳包含上一个时间戳，形成了一个链，以后的每次交易都加强了上一次的时间戳。



1. 工作量证明

我们需要使用工作量证明系统，在在点对点的基础之上构建一个分布式的时间戳服务器，类似于adam back提出的的哈希货币，而不是以前的新闻组及论坛的机制。当数据被哈希加密后，工作量证明系统用安全散列算法sha-256对一个数据的哈希值进行检查。哈希从0字节开始，检查的平均工作量随着0字节的增长而呈指数增长，校验只需要执行一次哈希操作。

为了我们的时间戳网络可行，我们增加一个不会重复的随机数到数据块内并执行一定的工作量来找到它，这个数据块的的哈希已经包含所需数量的0字节。若CPU处理能力经证明满足了所需的工作量，那么不重做所有的工作，这个数据块将不能被修改。随后的数据块被链接在末尾，修改数据块的信息需要把其后所有的数据块的工作量重做。



工作量系统解决了集体决策谁代表多数的问题。如果多数是基于一个IP地址一票的机制，会被能分配大量IP地址的人所破坏，工作量证明是一个CPU一票。多数决策被最长的链代表，也代表了最大工作量效果的投入。如果大部分CPU被诚实节点控制，诚实链将最快速的增长，超过任何竞争链。 修改一个过去的数据块，攻击者将不得不把块和之后所有的块总体的工作量重做，然后赶超诚实节点的工作量。我们将展示随数据块的增加，一个慢速攻击者追上随后的数据块的可能性会呈指数增加。

为了平衡硬件增加的速度和节点运行时的收益变化，会通过一个每小时平均生成数据块的移动平均值来确定工作量难度。如果它们生成的太快，难度也增加。

1. 网络

网络运行步骤：

1. 新的交易被传送到所有节点。
2. 每个节点把新交易编入一个数据块。
3. 每个节点找出数据块的工作量难度。
4. 当一个节点证明了数据块的工作量，会把它传到所有节点。
5. 节点接收数据块，只有块中的交易发生但未支付，节点才会接受该数据块。
6. 节点在数据链上创建下一个数据块，同时把发送节点数据块的哈希作为创建数据块的上一个哈希，表示它们接受了这个数据块。

节点始终认为最长的数据链是正确的，会一直在上面延展。如果两个节点一起传播不同版本的数据块，一些节点先接收其中的一个。案例中，它们会在第一个接收的数据块上工作，但是储存另一个，防止它变长。当工作量证明网络发现其中一个分支变的更长，在短链上工作的节点会切换到更长的链上，其所属关系也会打破。

新的交易不需要传到所有节点，只需要尽可能到达多的节点，然后整合进数据块中。数据块也可以在传播中丢弃信息。如果一个节点没有接收到数据块，它将持续请求，直到接收到下一个数据块，并认为它是丢失的那个。

1. 激励

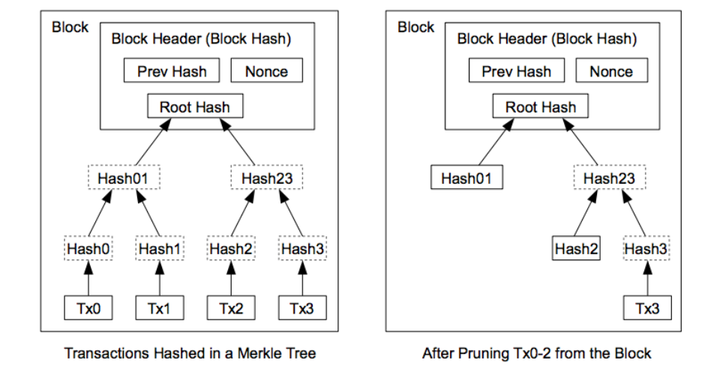
按照习惯，数据块里的第一个交易是特定的,它创建了一个新货币，由这个数据块的创建者拥有。这给节点一个支持网络的激励，同时提供一个在没有中心权威去影响的情况下，分布发行货币但的方法。稳定增加的新货币和挖金人花费资源增加黄金进入循环系统一样。在案例中，处理器要花费时间和电力。

交易费用也可以产生激励。如果交易的输出值小于输入值，差值就是交易费，它作为激励被编入数据块。一旦预先设定了整个循环的货币量，交易费完全可以作为激励，避免通货膨胀。

激励也有助于节点保持诚实。如果一个贪婪的攻击者有能力集合很多CPU处理器能力且超过诚实节点，他要么选择从自己的交易里欺诈他人，要么生成新货币。他会发现遵守规则更有利，这样的规则使他更倾向于赚取新货币，而不是联合更多去削弱系统损害自身。

1. 回收磁盘空间

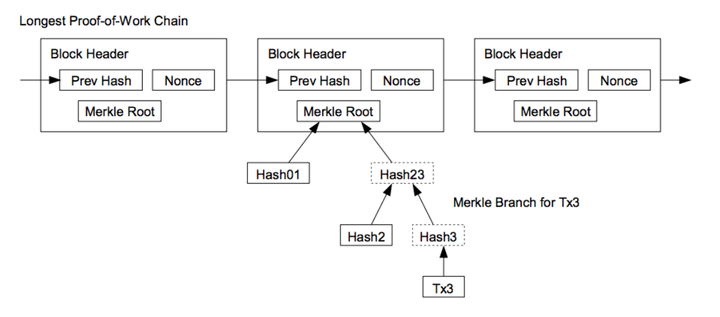
一个货币里最后的交易已经被足够多的数据块覆盖，那么支付交易之前的数据可以不再使用以节省磁盘空间。为了在不打断数据块哈希的前提下加快，交易被哈希到默克尔树（Merkle Tree），只有这个数据块哈希的根包含在内，老的数据块可以被压缩进树的枝干而拔除。内部的哈希不需要存储。



一个不含交易信息的数据块头部大概有80字节。如果我们支持每十分钟生成一个数据块。80字节\*6\*24\*365=4.2M/每年。2008年，每个通用计算机有2G内存。根据摩尔定律预测，每年增长1.2GB,数据头部也可以存在内存里。

1. 简化的支付验证

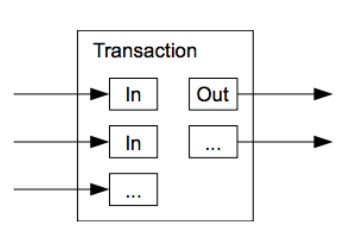
认证支付不需要运行一个完整的网络节点。用户只需要复制工作量网络的最长数据链的数据块头部，可以通过在网络节点上排队等待直到相信自己已经得到了最长的链确认，交易数据块就可以连接到默克尔分支。他不能验证自己的交易，但是通过连接到链的某个位置，能看到网络节点已经接受了这个数据，并且其后增加的数据块也会证明网络节点接受了它。



这样，这个支付验证依靠尽可能诚实的节点控制网络，但是如果网络被一个拥有大量算力的攻击者控制，会更容易受到攻击。当网络节点可以确认自身交易，只要攻击者拥有超过网络的算力，简化的模型就会被攻击者利用。一个保护策略是，当网络节点侦测到一个无效数据块，会收到警告，促使软件用户去下载整个数据块，以确认被警告交易是否一致。支付频繁的商家还会运行自己的节点，获得更独立的安全性，更快的确认交易。

1. 合并、拆分数据

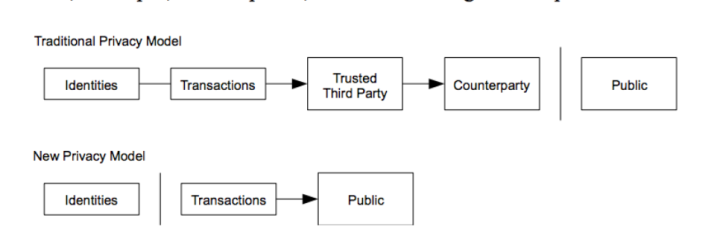
尽管它可以控制单个货币的交易，但不擅长分开处理每一分钱的交易。交易包含多个输入和输出，应该允许数值的拆分和合并。通常，要么就是从上一个更大的交易里单一输入，要么就是把多个输入合并成更小的数值，并且最多只有两个输出：一个负责支付，一个负责找零(如果有，则返回给发送者)。



需要注意的是输出端。一个交易依赖于多个交易，同时这些交易依赖于更多交易，这并不是问题。永远不需要去展开一个交易的历史记录。

1. 隐私

传统银行的模式给合作伙伴提供受限的信息访问权，同时还涉及受信第三方。向公众公布交易信息的必要性虽然排除了传统方法，但维护隐私还可以通过匿名的公共密钥，打断其他方面的信息流，以公开交易信息。公众可以看到某人发送给其他人金额，但没有交易人的信息，这个很像股票交易所的信息发布级别，公众记录了单比交易的时间和规模，但是不知道交易者。



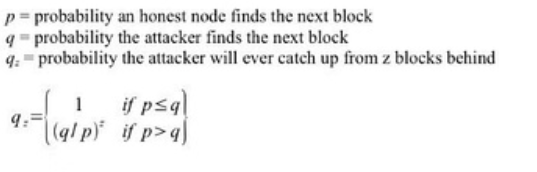
作为一个额外的防火墙，同一个拥有者的每一比新交易要连接一对新的配对密钥。一些连接还是不可避免的对应多个交易，这必须暴露同一个拥有者过去的其他输入，这存在风险，如果拥有者的密钥暴露了，也会暴露属于同一拥有者的其他交易。

1. 计算

我们假设一个场景，攻击者试图生成一个更快的链以替代诚实链。假如它成功了，抛开这个系统随意改变，例如，凭空创造一个值或者拿走从不属于他的钱。节点不会接受一个无效的支付，诚实节点将永远不会接受它的数据块。攻击者能做的仅仅是努力去改变自己的交易，从上一次支付里把钱拿回来。

诚实链和攻击链之间的比赛是一个二项分布的随机漫步。成功事件是诚实节点被一个数据块扩展，领先增加一个点；失败事件是攻击者的链扩展一个数据块，差距减少一个点。

攻击者从一个短缺链赶超的可能性类似赌徒破产问题。假设一个赌徒拥有无限的信用，从赤字开始，有无限次赌的机会，直到盈亏平衡。计算他达到盈亏平衡的可能性，那也就是攻击链追上诚实链的可能性，如下所示：

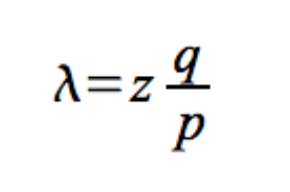


假设p>q,可能性随着攻击者追上数据块的增加呈指数下降，如果他不能在早期幸运的赶上，越往后，他的机会更渺茫。

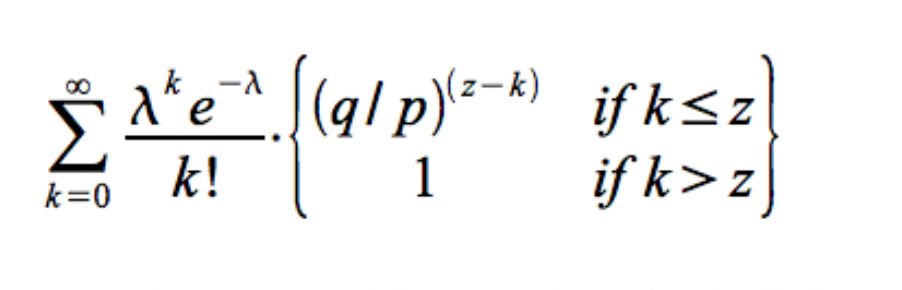
现在思考需要等待多久才能确认发送者不能改变交易。我们假设发送者是个攻击者，他想让接收者相信他已经付钱，之后，他把钱再付给他自己。当发生时，接受者会收到警告，但是发送者希望警告迟一些。

接收者在签名前生成一个新的配对密钥然后很快把公钥给了发送者。此时执行这个交易，可以在他幸运的跑到前面之前，防止发送者在这个时间点前准备一个数据块链并开始后续工作。交易一经发出，不诚实的发送者就开始在一个并行包含了他交易的替代版本的链上秘密工作。

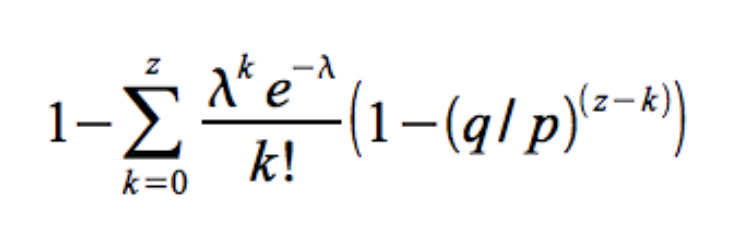
接收者等待交易添加进一个数据块的同时Z数据块已经链接在它的后面。他不知道攻击者制造的数据块进展，但是，假设诚实的数据块按每个数据块的期望值生成，攻击者进展的期望值呈柏松分布。



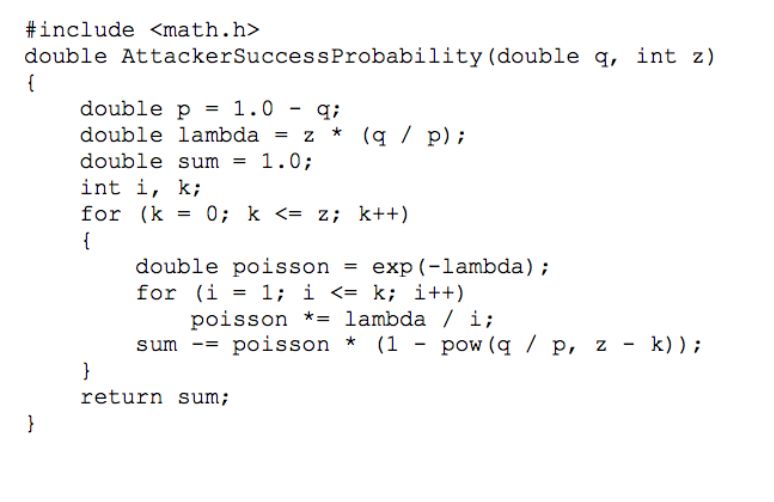
为了得到攻击者在这时追上的概率，我们用柏松分布密度乘以他可能在这个点上追上的可能进展数：



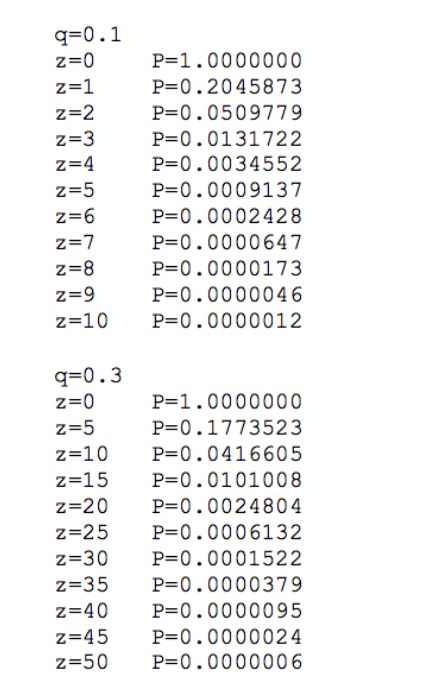
重新整理避免分布无限循环的尾部求和



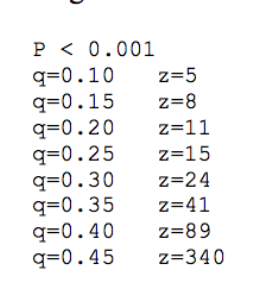
转换到C代码……



运行一些结果，我们可以看到随着z值的增加，概率呈指数下降。



求解 P 小于 0.1%……



1. 总结

我们已经提出了一个不需要受信第三方的电子交易系统。我们从常用的基于数字签名的货币模型开始，虽然它提供了强大的控制力，但是在防止双重支付方面做的不完整。为了解决这个问题，我们提出了一个依靠工作量证明的点对点网络，用其记录一个公共的交易历史，如果诚实节点控制了主要的处理能力，那么经过计算，攻击者希望修改记录的意图将变的不切实际。这个网络简单且稳健。所有网络上的节点仅需要一点点的协调。它们不需要被认证，信息不需要路由到任何特别的地方，只需要尽可能传播。接受它们离开时工作量网络产生的数据链,计算节点就可以随时加入或离开网络。用CPU处理器能力投票，通过在数据块上扩展新的数据，表示认同数据块有效；拒绝在数据块上扩展以拒绝无效数据块。任何需要的规则和奖励都整合进统一机制且强制执行。

参考书目

