比特币：一种点对点的电子现金系统

**摘要**：一个完全的点对点版本的电子现金将允许一方不通过金融机构直接在线支付给另一方。电子签名提供了部分解决方案，但是如果还需要一个可信任的第三方来防止双花，那么这个最大的好处也就没有意义。我们提出一个用点对点网络来解决双花的方案。这个网络给每笔交易打上时间戳，并进行哈希计算，放进一条基于哈希工作量证明的链，这形成了一个不可改变的记录，除非重做这些工作量。最长的链不仅是见证序列的证明，还证明了它来自最大的CPU算力池。因为大部分的算力由诚实的节点控制，他们将会产生一条比攻击者要长的链。网络本身需要极小化结构。消息被尽力广播，并且节点可以随意离开或重新加入网络，接受最长的工作量证明的链作为它离开这段时间发生事情的证明。

1． 介绍

互联网上的商业几乎完全依赖信任的第三方金融机构来处理电子支付。对于大多数交易来说，这套系统工作的足够好了，但是依然受到了基于信任模型的天然缺点的困扰。完全不能撤销的交易是不可能的，因为第三方金融机构不可避免的要调解纠纷。调解的代价增加了交易的成本，限制了最小实际交易的大小，切断了临时交易的可能性，丧失了对不可撤销服务提供不可撤销支付的可能性，这又是一个广义成本。因为撤销的可能性，信任的需要不断蔓延开来。商户必须堤防他们的客户，越来越多的他们本不该需要的信息困扰着他们。不得不接受一定比例的骗子。这些成本和支付的不确定问题可以用面对面使用现金避免，但是还没有机制存在使得通过通信信道支付而不需要信任的第三方。

需要的是一个电子支付系统，这个系统建立在密码学证明基础上而不是信任，允许任意有这个意愿的双方直接相互转账而不需要一个信任的第三方。交易从计算上不可撤销的，这将保护卖方权益防止被骗，并且常规的托管机制很容易实现来保护买方权益。在本论文里，我们提出了一个防止双花的解决方案，使用点对点分布式时间戳服务器来产生按时间排序的交易的计算证明。只要诚实节点控制的CPU的算力大于攻击者节点的算力，这个系统就是安全的。

2． 交易

我们把一种电子币定义成一条数字签名链。每一个所有者把币转给下一个人的时候，是通过将前一个交易的哈希和下一个所有者的公钥进行数字签名，并把这些追加在币的后面。收款人可以通过验证数字签名来确认链的所有者。

（示意图）

当然，问题是收款人无法验证其中的一个所有者是否同一个币花了两次。一个普遍的做法是引入一个信任的中央机关，或铸币厂，他们可以检查每一笔交易来防止双花问题。每次交易后，这个币必须返回到铸币厂，这样才能发行新币，只有直接从铸币厂发行的币才被相信是没有被双花的。这个方案的问题是，整个金钱系统的命运掌握在经营铸币厂的公司，每一笔交易都要经过他们，就像银行一样。

我们需要一种方法，这种方法让收款人知道上一个所有者没有签署任何以前的交易。我们的目的是，让最早的交易是可信的，我们不关心后面是不是有人企图进行双花。仅有的可以确认某一个交易存在的办法是要知道所有的交易。在基于铸币厂的模型中，铸币厂知道所有的交易，并且可以确定哪个交易先发生。为了在无信任第三方的情况下达到这个目的，交易必须要对公众进行通知，并且我们需要一个系统，这个系统的参与者要达成共识，这个共识就是认同同一个按收到的交易顺序排列的历史记录。收款人需要证据来说明在每一笔交易的时候，大多数节点一致认为这个交易是第一时间到达的。

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。这个服务器工作方式是，对条目所在的区块的哈希加盖时间戳，并且广泛的公布这些哈希，比如通过报纸或新闻组邮件。显然，为了能进入这个哈希序列，时间戳证明的数据在那个时间必须存在。每一个时间戳和以前的时间戳，形成一条链，每一个追加的时间戳都是对前一个时间戳进行加强。

4． 工作量证明

为了基于点对点的基础实现一个分布式的时间戳服务器，我们将需要一个工作量证明的系统，这个系统和亚当贝克的哈希现金类似，而不是报纸或新闻组邮件。这个工作量包含寻找一个哈希值，比如用哈希算法SHA-256，这个哈希值以若干0开头。平均工作量和开头的0的个数是指数关系，并且验证很简单，只需要执行一次单独的哈希计算。

在我们的时间戳网络里，是这样实现工作量证明的，就是不断增加区块里的一个临时的数值，直到找到一个值使得区块的哈希值满足开头0的个数的要求。一旦CPU花费算力计算满足了工作量证明的要求，这个区块链就无法修改，除非重新计算。随着后面的区块不断产生，想要改变这个区块，需要重做它后面所有区块的工作量。

（示意图）

工作量证明还解决了在多数决策法中决定展示的问题。如果大多数是基于一个IP一票，这可能会被可以支配多个IP的人破坏。工作量证明本质上也是一个IP一票。最长的链代表了大多数人的决定，这个链投入了最大的工作量。如果大多数的CPU算力由诚实节点控制，诚实的链就会增加的很快，超过任何竞争链。为了修改过去的一个块，攻击者需要重新投入算力完成这个区块和它以后所有区块的工作量，然后追上并超越诚实节点的工作量。我们后面讨论落后的攻击者追上的概率，这个概率随着后面区块的增加呈指数级减少。

硬件的速度越来越快，参与运行的节点随时间兴趣也经常变化，为了抵消这些因素的影响，工作量的难度是由每小时区块产生数量的浮动性决定的。如果区块产生的太快，难度就相应的增加。

5． 网络

运行这个网络的步骤如下：

1） 新交易给所有节点广播。

2） 每个节点将新交易放到一个区块。

3） 每个节点开始为这个区块寻找相应难度的工作量证明。

4） 当一个节点找到了这个工作量证明，把这个区块广播给所有节点。

5） 如果区块里所有的交易是有效的并且是没有被花费的，节点就会接受这个区块。

6） 节点把这个区块的哈希作为上一个哈希，并开始进行工作以竞争创建下一个区块。

节点总是认为最长的链是正确的，并且不断的工作去延长它。如果两个节点同时广播下一个区块，一些节点可能接受其中一个，也可能是另一个。这种情况下，他们在最先收到的区块上工作，但是也会保存另外一个分支以防它会变得更长。当下一个工作量被找到并且一个分支变得更长时，这种情况就会被打破，在另外一个分支上工作的节点将会切换到这个长的链上。

新交易广播不一定要广播到所有节点。只要他们能到达很多节点，这个交易很快就会进入下一个块。区块广播也能接受消息丢失。如果一个节点没有收到区块，当它收到下一个块时会发现自己少了一个区块，它就会请求来获得少的这个区块。

6． 激励

按照惯例，区块的第一个交易是一个特别的交易，这个交易会发行新币并且所有者是这个区块的创建者。这为节点支持网络引入了激励机制，并且这提供了一个初始发行货币进入流通的方式，因为没有一个中央机构去发行他们。不断增加新货币的过程类似于黄金矿工消耗资源来增加黄金的流通。在这里，消耗的事CPU的时间和电费。

激励还包括提供交易手续费。如果交易中输出值比输入值小，这个差值就是交易手续费，它被加入到这个区块激励值里。一旦预定数量的币全部进入流通，激励就全部转为交易手续费，完全没有通货膨胀。

激励有助于鼓励节点保持诚实。如果一个贪婪的攻击者掌握了比所有诚实节点还要大的算力，他将面临一个选择，是通过偷回他支付的钱来咋骗别人，还是用这个算力产生新的币。他应该会发现遵守规则更有好处，这个规则可以让他比其他人组合得到更多的新币，比破坏这个系统得到的更多，而且财产合法。

7． 回收磁盘空间

一旦一个币最新的交易被足够多的区块埋没，它之前的花费的交易就可以丢掉来节省空间。为了促成这个而不破坏区块的哈希，用这些交易生成一个默克尔树，仅仅根包含在区块的哈希里。那么旧区块可以通过去除树的一些分支进行压缩。有些内部的哈希就不用保存了。

（示意图）

一个区块头大概80字节。如果我们假设每十分钟产生一个区块，一年就是80字节\*6\*24\*365=4.2兆字节。2008年出售的电脑典型的配置是2GB内存，根据摩尔定律预测，每年增加1.2G，即使区块头全部放在内存里，存储也不是个问题。

8． 简化支付验证

即使不运行全网络节点，验证支付也是可能的。用户仅需要保存最长工作量证明链的区块头的拷贝，他通过查询网络节点直到确信它有最长的链来获取区块头，并且可以得到默克尔分支，分支连接了交易和这个打了时间戳的区块。他本身不能验证交易，但是通过连接到链上的一个地方，他可以看到网络节点已经接受了它，后面的区块进一步确定网络接受了它。

（示意图）

因此，如果诚实节点控制着网络，验证就是可靠的，如果网络被攻击者控制，验证就是很弱的。虽然网络节点本身可以验证交易，但这种简化验证的方法会被攻击者编造的交易欺骗，因为攻击者可能持续控制网络。防止这种情况的一种策略是接收网络节点的告警，当他们检测到无效块的时候，提示用户软件下载整个区块，并且提醒确认交易的一致性。频繁接收支付的企业可能仍然想运行他们自己的节点，为了更独立的安全性和更快的验证。

9． 组合和分割价值

虽然可以单独处理币，为转账的每一分都单独交易是很不方便的。为了能是价值分割和组合，交易包含多个输入和多个输出。正常的会有一个单独从以前交易来的大额输入或多个小额输入组合在一起，最多两个输出：一个用来支付，一个用来找零，有零钱的话会返还给发送者。

（示意图）

应该注意的是，一个交易依赖几个交易，这些交易依赖更多的交易，看起来很分散，但在这里不是一个问题。从不需要提取一个交易全部历史的独立的拷贝。

10． 隐私

传统的银行通过限制向有关方和信任的第三方提供信息来达到一个保护隐私的目的。向公众广播所有交易的必须性将这个方法排除了。通过打破信息在其他地方的流动性仍然可以保护隐私：通过保持公钥匿名性。公众可以看到一个人给其他人转钱了，但是没有信息可以把交易和某人联系起来。这类似于证券交易所公布的信息水平，交易时间和个人的交易规模是公开的，但不告诉当事人是谁。

（示意图）

作为一个附加的防火墙，每次交易都使用一个新的密钥对，防止和一个共同的所有者联系起来。对于多输入交易来说，这个联系无法避免，所有的输入必须表明由同一个人所有。风险是如果表明了某一个密钥的所有者，这种联系将表明其他的交易也属于同一个人。

11． 计算

我们想象一个这样的场景，攻击者想用比诚实节点更快的速度产生一个替代链。即使成功了，也不会让系统能任意被修改，比如凭空产生价值或拿走不属于攻击者的钱。节点将不会接受一个无效的支付，并且诚实节点绝不会接受包含这种支付的区块。攻击者只能试着改变自己的交易来拿回本该花出去的钱。

诚实的链和攻击链的竞争可以说是二项式随机走动。成功事件是诚实链延长一个区块，领先优势加一，失败事件是攻击链延长一个区块，缩小一个差距。

一个攻击者从一个给定的赤字中追上的概率类似于一个赌徒破产问题。假设一个信用无限的赌徒从赤字开始，开始进行潜在次数无数的赌博，试图达到盈亏平衡。我们可以计算他达到盈亏平衡的概率，或者说是攻击链赶上诚实链的概率，如下：

p=诚实节点发现下一个区块的概率

q=攻击者找到下一个区块的概率

qz =攻击者在落后z个区块的情况下，追上的概率

（公式）

假设p>q，随着落后区块数量（z）增加，攻击者追上的概率呈指数下降。这个概率情况对攻击者不利，如果他没有幸运的提前向前冲刺，落后越多希望就越渺茫。

我们现在考虑接收者在收到新的交易的时候，需要等待多长时间才能完全确定交易不能被发送者修改。我们假设发送者是攻击者，他想让接收者暂时相信他已经付款了，然后过了一段时间又换成是支付给自己。这事发生的时候接收者会收到告警，但是发送者希望一切都晚了。

接收者创建了一个新的密钥对，签名之前很短的时间把公钥给了发送者。这防止发送者提前准备一条链，持续在上面工作，直到他足够幸运达到了领先的程度，正好执行刚才这条交易。一旦这个交易发送了，不诚实的发送者开始在一个并行的链上秘密工作，这条链包含他的交易的另一个版本。

接收者一直等待直到交易被添加到一个区块中，并且后面已经追加了z个区块了。他并不知道攻击者的准确进展，但是可以假设诚实区块每个区块花费的时间是平均期望时间，攻击者潜在的进展将服从泊松分布，期望值：

（公式）

为了得到目前攻击者仍能追上的概率，我们将他所取得的每一步进展的泊松密度乘以他可能从那一点赶上的概率。

（公式）

变换一下避免对分布的无穷尾部求和…

（公式）

转换成C代码…

（代码）

运行结果，我们可以看到概率随z的增加呈指数下降。

（运行结果）

12． 结论

我们为无信任电子交易提出了一个系统。我们从数字签名币的常用框架开始，它对所有者有很强的控制，但是因为不能避免双花，所以还不完整。为了解决双花，我们提出了一个点对点的网络，这个网络使用工作量证明记录一个公共的交易历史，只要诚实节点控制大部分CPU算力，很快使得攻击者无法通过计算来改变交易历史。该网络的非结构化简单性使得它很稳健。节点同时工作，很少需要相互协调。他们不需要被识别，因为消息不需要路由到任何特定的位置，只需尽力传递就好。节点可以离开网络，也可以需要的时候重新加入网络，接受工作量链作为他离开的时候发生了什么的证据。他们用CPU算力投票，通过在有效区块上工作并延续它来表达对区块的接受，通过不在新区块上工作表示拒绝无效区块。任何需要的规则和激励都可以在这种共识机制下进行。