1. Introduction

“比特币”[1]已经成功实施了 p2p 电子现金的概念。专业人士和公众都认识到公共交易和工作作为信任模式的方便结合。今天电子现金用户基数稳步增长。客户被低收费吸引，电子现金和商家提供的匿名性重视其预测和分散排放。比特币有效证明，电子现金可以像纸币、信用卡一样简单方便。 不幸的是，比特币遭遇了几个缺陷。例如，系统的分布式属性是不灵活的，阻止新功能的实现，直到几乎所有的网络用户更新其客户端。一些不能快速解决的关键缺陷阻止了比特币广泛传播。在这样不灵活的模式中，推出新项目更有效率，而不是永久地修复原始项目。 在本文中，我们研究并提出了比特币主要缺陷的解决方案。我们认为，考虑到我们提出的解决方案的系统将导致不同电子现金系统之间的健康竞争。我们还提出自己的电子现金“CryptoNote”，这个名字强调电子现金的下一个突破。

2. 比特币的缺点和一些可能的解决方案

2.1 交易追溯

隐私和匿名是电子现金最重要的方面。对等支付寻求从第三方的角度来看，与传统银行业相比有明显差异。特别是 T. Okamoto 和 K. Ohta 描述了理想电子现金的六个标准，其中包括“隐私：用户和他的购买之间的关系必须由任何人不可追究”[30]。从他们的描述中，我们得出了两个完全匿名的电子现金模型必须满足的属性，以符合冈本和 T. Okamoto 和 K. Ohta 大田概述的要求： 不可追踪性：对于每个进入的交易，所有可能的发件人从概率上是相等的。 无关联性：对于任何两个外部交易，不可能证明将其发送给同一个人。 不幸的是，比特币不符合非追溯要求。由于网络参与者之间发生的所有交易都是公开的，因此任何交易都可以明确地追溯到独特的起源和最终的收件人。即使两个参与者以间接的方式交换资金，一个适当设计的路径寻找方法将揭示起源和最终的收件人。 也有人怀疑比特币不符合第二财产。一些研究人员表示（[33,35,29,31]）仔细的块链分析可能揭示了比特币网络用户与其交易之间的联系。虽然有一些方法有争议[25]，但怀疑可以从公共数据库中提取很多隐藏的个人信息。 比特币未能满足上述两个特性，导致我们得出结论，它不是一个匿名的，而是一个伪匿名的电子现金系统。用户很快开发解决方案来规避这个缺点。两个直接的解决办法是“洗钱服务”[2]和分布式方法的发展[3,4]。这两

个解决方案都是基于混合几个公共交易并通过一些中间地址发送他们的想法;这反过来又具有要求可靠的第三方的缺点。 最近，Miers 等人提出了一个更具创意的方案。 [28]：“Zerocoin”。 Zerocoin 利用加密单向累加器和零知识证明，允许用户将比特币“转换”为零竞争，并使用匿名的所有权证明，而不是基于公开密码的基于数字签名。然而，这种知识证明有一个不变但不方便的大小 - 约 30kb（基于今天的比特币限制），这使得该提案不切实际。作者承认，该协议不太可能被大多数 Bitcoin 用户所接受[5]。

2.2 工作量证明

Bitcoin 创作者 Satoshi Nakamoto 将多数决策算法描述为“一 CPU 一票”，并使用 CPU 限价定价功能（双 SHA-256）作为其工作量证明方案。由于用户投票单个交易历史顺序[1]，这个过程的合理性和一致性是整个系统的关键条件。 这种模式的安全性有两个缺点。首先，它需要 51％的网络采矿权力由诚实的用户控制。其次，系统的进度（bug 修复，安全修复等）要求绝大多数用户支持并同意更改（这在用户更新其钱包软件时发生）[6]。最后这个相同的投票机制也用于集体民意调查实施某些特征[7]。 这样我们可以推测工作定价功能必须满足的属性。这种功能不能使网络参与者具有比另一参与者更大的优势;它需要通用硬件和定制设备的高成本之间的平衡。从最近的例子[8]可以看出，Bitcoin 架构中使用的 SHA-256 功能并不具备这一特性，因为与高端 CPU 相比，采集在 GPU 和 ASIC 器件上的效率更高。 因此，与 CPU 占有者相比，GPU 和 ASIC 所有者拥有更大的投票权，因此比特币为参与者的投票权力之间的巨大差距创造了有利条件，因为它违反了“一 CPU 一票”原则。这是帕累托原则的典型例子，其中 20％的系统参与者控制超过 80％的选票。人们可以认为，这种不平等与网络的安全性无关，因为控制大多数投票的参与人数不多，但这些参与者的诚实是重要的。然而，这样的论据有些缺陷，因为它可能是出现廉价的专业硬件，而不是参与者的诚实造成威胁。为了说明这一点，我们来看一下例子。假设一个恶毒的个体通过廉价创建自己的采矿场，获得了巨大的矿业权力以前描述的硬件。假设全球哈希值显着下降，即使是一段时间，他现在也可以利用他的挖掘能力来分支链和双重支出。正如我们在本文后面将会看到的，前面描述的事件不可能发生。

2.3 不平缓的释放曲线

比特币具有预定的排放率：每个解决的块产生固定量的硬币。大约每四年一次，这个报酬减半。最初的目的是创造一个具有指数衰减的有限的平滑发射，

但实际上我们有一个分段线性发射功能，其断点可能会导致比特币基础设施的问题。 当断点发生时，矿工开始只收到他们以前的报酬的一半。 BTC（预计在 2020 年）之间的绝对差异似乎是可以忍受的。然而，在审查 2012 年 11 月 28 日发生的 50 至 25 点 BTC 跌幅时，对于大量的采矿社区成员来说，这是不合适的。图 1 显示了 11 月底网络哈希率的急剧下降，恰恰在减半的时候。这个事件本来可能是工作功能部分描述的恶意个人进行双重支出攻击的完美时刻[36]。

2.4 硬编码常数

比特币具有许多硬编码限制，其中一些是原始设计的自然元素（例如块频率，最大货币供应量，确认数量），而其他似乎是人为约束。这不是极限，因为无法快速改变。他们如果有必要，造成主要的缺点。不幸的是，很难预测常数可能需要改变的时间，并且替换它们可能会导致可怕的后果。 硬编码极限变化导致灾难性后果的一个很好的例子是块大小限制设置为 250kb1。这个限制足以容纳约 10000 个标准交易。在 2013 年初，这个限制几乎已经达到了，并且达成了增加限制的协议。这个变化是在钱包版本 0.8 中实现的，结束了 24 块链分裂和成功的双重花费攻击[9]。虽然该错误不在 Bitcoin 协议中，而是在数据库引擎中，如果没有人为引入的块大小限制，它可能很容易被简单的压力测试所捕获。 常数也是集中点的形式。尽管比特币的 P2P 对等性质，绝大多数节点使用了一小群人开发的官方参考客户端[10]。该组决定实施协议的更改，大多数人接受这些更改，而不管其“正确性”如何。一些决定引起了热烈的讨论，甚至呼吁抵制[11]，这表明社区和开发商可能会对某些重要点不同意。因此，具有用户可配置和自调整变量的协议似乎是合乎逻辑的，作为避免这些问题的可能方法。

2.5 庞大的脚本

比特币的脚本系统是一个沉重而复杂的功能。它可能允许创建复杂的事务[12]，但是由于安全性问题，其某些功能被禁用，有些甚至没有被使用[13]。Bitcoin 最流行的交易的脚本（包括发件人和接收者的两个部分）看起来像这样： OP DUP OP HASH160 OP EQUALVERIFY OP CHECKSIG。 该脚本长度为 164 字节，而其唯一目的是检查接收者是否具有验证其签名所需的秘钥。

3. CryptoNote 技术

现在我们已经涵盖了比特币技术的局限性，我们将集中介绍 CryptoNote 的功能。

4. 不可追溯的交易

在本节中，我们提出了一个完全匿名交易的方案，满足不合规范和不相关的条件。 我们的解决方案的一个重要特征是它的自主性：发件人不需要与其他用户或受信任的第三方合作进行交易; 因此每个参与者独立地产生覆盖流量。

4.1 文献评论

我们的方案依赖于称为组签名的加密原语。首先由 D. Chaum 和 E. van Heyst 提出[19]，它允许用户代表组签署他的消息。用户签署消息后（为了验证目的）不是他自己的单个公共消息 1 这是所谓的“软限制” - 创建新块的参考客户限制。硬度最大值可能的 blockize 是 1 MB 关键，但他的组的所有用户的关键。验证者相信真正的签名者是该组的成员，但不能专门识别签名者。原始协议需要可靠的第三方（称为集团经理），他是唯一可以跟踪签名者的协议。 Rivest 等人介绍的下一个版本称为环形签名。在[34]中，是一个没有组经理和匿名撤销的自治计划。该方案的各种修改稍后出现：可链接环签名[26,27,17]允许确定两个签名是否由同一组成员生成，可追溯环签名[24,23]通过提供跟踪签名者的可能性来限制过度匿名关于相同的元信息（或[24]的“标签”）的两条消息）。 类似的加密结构也被称为特设组签名[16,38]。它强调任意组合，而组/环签名方案则意味着一组固定成员。 在大多数情况下，我们的解决方案是基于 E. Fujisaki 和 K. Suzuki 的“可溯源环签名”[24]。为了区分原始算法和我们的修改，我们将后者称为一次性环签名，强调用户在其私钥下只生成一个有效签名的能力。我们削弱了可追溯性，并保持了链接性，只能提供一致性：公钥可能出现在许多外国验证集中，私钥可用于生成唯一的匿名签名。如果双重花费尝试，这两个签名将被链接在一起，但是为了我们的目的，揭示签名者是不必要的。

4.2 定义

4.2.1 Elliptic curve parameters 作为我们的基本签名算法，我们选择使用由 D.J.开发和实现的快速方案 EdDSA。伯恩斯坦等人[18]。像比特币的 ECDSA 一样，它是基于椭圆曲线离散对数问题，所以我们的方案将来也可以应用于比特币。 常用参数有： q：素数; q = 2255 - 19; d：Fq 的元素; d = -121665/121666; E：椭圆曲线方程; -x2 + y2 = 1 + dx2y2; tt：一个基点 tt =（x，-4/5）; l：基点的主要顺序; l = 2252 + 27742317777372353535851937790883648493; Hs：加密哈希函数{0,1} *→Fq; Hp：确定性散列函数 E（Fq）→E（Fq）。 4.2.2 术语 增强的隐私需要一个不应与比特币实体混淆的新术语。 私钥 ec-key：是一个标准的椭圆曲线私钥：a a a∈[1，l - 1]; 公共 ec-key：是一个标准的椭圆曲线公钥：点 A = att; 一次性关键词：是一对私人和公共电子钥匙;

私人用户密钥：是两个不同的私人密钥的一对（a，b）; 跟踪密钥：是私人和公共密钥对（a，B）（其中 B = btt 和 a f = b）; 公共用户密钥：是从（a，b）派生的两个公共 ec 密钥的一对（A，B）; 标准地址：是给出具有纠错的人性化字符串的公共用户密钥的表示; 截断地址：是给出具有错误校正的人性化字符串的公共用户密钥的后半部分（点 B）的表示。 交易结构与 Bitcoin 结构保持类似：每个用户都可以选择几个独立的收款（交易输出），并用相应的私钥签名并将其发送到不同的目的地。 与比特币模型相反，用户拥有唯一的私有和公钥，在提出的模型中，发送者基于收件人的地址和一些随机数据生成一次性公开密钥。在这个意义上，同一收件人的传入交易被发送到一次性公开密钥（而不是直接到唯一地址），只有收件人可以恢复相应的私人部分来兑换他的资金（使用他的唯一私钥）。收件人可以使用戒指签名来支付资金，将其所有权和实际支出保持匿名。协议的细节将在下一小节中解释。

4.3 无法支付的款项

经典比特币地址一旦被发布，就成为收入付款的明确标识符，将它们连接在一起并绑定到收件人的假名。如果有人想收到一个“无约束”的交易，他应该通过一个私人渠道将他的地址传达给发件人。如果他想收到不能被证明属于同一所有者的不同交易，他应该生成所有不同的地址，而不是以自己的化名发表。

我们提出一个解决方案，允许用户发布单个地址并接收无条件的不可链接的付款。每个 CryptoNote 输出的目的地（默认情况下）是一个公钥，从收件人的地址和发件人的随机数据派生。对比特币的主要优点是默认情况下，每个目标

密钥都是唯一的（除非发件人使用相同的数据为每个交易到相同的收件人）。因此，不存在通过设计的“地址重用”这样的问题，并且没有观察者可以确定是否有任何事务被发送到特定的地址或将两个地址链接在一起。

首先，发件人执行 Diffie-Hellman 交换，从他的数据和收件人地址的一半获取共享密钥。然后，他使用共享密钥和地址的下半部分计算一次性目的地密钥。从这两个步骤的收件人需要两个不同的 ec 密钥，因此标准的 CryptoNote 地址是比特币钱包地址的两倍。接收机还执行 Diffie-Hellman 交换以恢复相应的秘密密钥。 标准交易序列如下：

爱丽丝想向已经发布他的标准地址的鲍勃发送付款。她打开地址并获取 Bob 的公钥（A，B）。 Alice 生成随机 r∈[1，1 - 1]，并计算一次性公钥 P = Hs（rA）tt + B. Alice 使用 P 作为输出的目的地密钥，并将值 R = rtt（作为 Diffie-Hellman 交换的一部分）包装到事务的某处。请注意，她可以使用独特的公钥创建其他输出：不同的收件人的键（Ai，Bi）意味着不同的 Pi，即使使用相同的 r。

爱丽丝发送交易。 Bob 用他的私钥（a，b）检查每个过程的交易，并计算 P r = Hs（aR）tt + B。如果 Alice 与 Bob 作为收件人的交易在其中，则 aR = artt = rA 和 P r = P。 Bob 可以恢复相应的一次性私钥：x = Hs（aR）+ b，因此 P = xtt。他可以随时通过与 x 签署交易来支出此输出。

因此，Bob 获得了与一次性公钥相关的收款无法连接观众。一些附加说明：当鲍勃“认出”他的交易（见步骤 5）时，他实际上只使用了他的私人信息的一半：（a，B）。这对，也称为跟踪键，可以传递给第三方（Carol）。 Bob 可以委托她处理新的交易。 Bob 不需要明确地信任 Carol，因为她无法恢复一次性秘密密钥 p 没有 Bob 的完整私钥（a，b）。当 Bob 缺乏带宽或计算能力（智能手机，硬件钱包等）时，此方法非常有用。 如果爱丽丝想证明她向 Bob 的地址发送了一个交易，她可以透露 r 或使用任何一种零知识协议来证明她知道 r（例如通过与 r 签署交易）。 如果 Bob 希望拥有所有传入交易可链接的审计兼容地址，他可以发布其跟踪密钥或使用截断的地址。该地址仅代表一个公共密钥 B，协议要求的其余部分是得出如下：a = Hs（B）和 A = Hs（B）tt。在这两种情况下，每个人都能够“认出”所有 Bob 的进入交易，但是当然，没有一个可以在没有秘密密钥 b 的情况下花费在其中的资金。

4.4 一次性环签名

基于一次性环签名的协议允许用户实现无条件的无连接性。不幸的是，普通类型的加密签名允许跟踪它们各自的发送者和接收者的交易。我们解决这个不足之处在于使用与目前在电子现金系统中使用的不同的签名类型。 我们将首先提供我们的算法的一般描述，没有明确提及电子现金。 一次性环形签名包含四种算法：（GEN，SIG，VER，LNK）： GEN：采用公共参数并输出 ec 对（P，x）和公钥 I. SIG：获取消息 m，公钥 Sr Pi，i f = s，一对（Ps，xs），并输出签名 σ 和一组 S = Sr∪{Ps}。 VER：取消消息 m，集合 S，签名 σ，并输出“true”或“false”。 LNK：取集合 I = {Ii}，签名 σ 并输出“链接”或“独立”。 协议背后的想法相当简单：用户产生签名，可以通过一组公钥而不是唯一的公钥进行检查。签名人的身份与其公共密钥在其中的用户是不可区分的，直到所有者使用相同的关键字生成第二个签名。

GEN：签名者选择随机密钥 x∈[1，1 - 1]，并计算相应的公钥 P = xtt。此外，他计算另一个公钥 I = xHp（P），我们称之为“关键图像”。 SIG：签名者使用[21]中的技术生成具有非交互式零知识证明的一次性环形签名。他从其他用户的公钥 Pi，他自己的密钥对（x，P）和密钥图像 I 中选择 n 的随机子集 Sr.令 0≤s≤n 是 S 中的签名者的秘密索引（使得他的公钥是 Ps ）。 他选择一个随机的{qi | i = 0。 。 。 n}和{wi | i = 0。 。 。 n，i*f* = s}从（1。。 I）并应用 以下变换： 李= Ri = .qitt，如果 i = s qitt + wiPi，如果 i*f* = s 如果 i = s qiHp（Pi）+ wiI，则如果 i f = s 下一步是获得非互动的挑战： c = Hs（m，L1，...，Ln，R1，...，Rn） 最后，签名者计算响应： wi，如果我 *f* = s ci = n ri = c - 。 ci mod I，如果 i = s i= 0 如果我 *f* = s qs - csx mod I，如果 i = s 得到的签名是 σ=（I，c1，...，cn，r1，...，rn）。 VER：验证者通过应用逆变换来检查签名： i = ritt + ciPi Ri = riHp（Pi）+ ciI N + 最后，验证者检查是否。 ci = Hs（m，Lr，...，Lr，Rr，...，Rr）mod I I = 0 0 N 0 N 如果这个相等性是正确的，验证者运行算法 LNK。否则验证者拒绝签名。 LNK：验证者检查我是否在过去的签名中被使用（这些值存储在集合 I 中）。多重用途意味着在相同密钥下生成了两个签名。 协议的含义：通过应用 L 转换，签名者证明他知道这样 x 至少有一个 Pi = xtt。为了使此证明不可重复，我们将关键图像引入为 I = xHp（P）。签名者使用相同的系数（ri，ci）来证明几乎相同的陈述：他知道这样的 x，至少有一个 Hp（Pi）= I·x-1。如果映射 x→I 是一个注入： 没有人可以从关键图像中恢复公钥，并识别签名者; 签名者不能用不同的我和同一个 x 进行两个签名。附录 A 提供完整的安全性分析。

4.5 标准 CryptoNote 交易

通过组合两种方法（不可链接的公开密钥和不可追踪的环形签名），Bob 与原始的比特币方案相比，达到了新的隐私级别。它要求他仅存储一个私钥（a，b）和发布（A，B）以开始接收和发送匿名事务。在验证每个事务时，Bob 另外只执行两个椭圆曲线多项式，每次输出一次，以检查一个事务是否属于他。对于他的每一个输出，Bob 都会恢复一次性关键词（pi，Pi）并将其存储在他的钱包中。任何投入都可以被证明是具有相同的所有者只有当它们出现在一个单一的事务。在事实上，这种关系由于一次性签名而难以建立。 通过一个环形签名，Bob 可以有效地隐藏别人的每一个输入;所有可能的消费者将是平等的，即使以前的所有者（爱丽丝）没有任何观察者的信息。 在签署他的交易时，Bob 指定了与他的输出相同数量的外部输出，混合所有输出，而没有其他用户的参与。鲍勃本人（以及其他人）不知道是否有任何这些付款已经花费了：输出可以在成千上万的签名中用作歧义因素，而不是隐藏的目标。在检查所使用的密钥图像集时，在 LNK 阶段发生双重支出检查。 鲍勃可以自己选择模糊度：n = 1 意味着他花费的概率是 50％的概率，n = 99 给出 1％。所得到的签名的大小随着 O（n + 1）线性增加，所以对 Bob 的额外交易费用的匿名性提高了。他也可以设置 n = 0，并使他的戒指签名只包含一个元素，但是这将立即显示他作为一个花费。

5. 平等工作量证明

在本节中，我们提出并研究了新的工作验证算法。我们的主要目标是缩小 CPU（大多数）和 GPU / FPGA / ASIC（少数）矿工之间的差距。一些用户可以比其他用户具有一定的优势是适当的，但是他们的投资应该随着电力线性增长而增长。更广泛地说，生产专用设备必须尽可能少的获利。

5.1 相关工作

原始的比特币工作协议使用了 CPU 密集型定价功能 SHA-256。它主要由基本逻辑运算符组成，仅依赖于处理器的计算速度，因此完全适用于多核/传输器实现。 然而，现代计算机并不受每秒操作次数的限制，也不受内存大小的限制。虽然一些处理器可以比其他处理器快得多[8]，但存储器大小在机器之间不太可能发生变化。 内存限价功能首先由 Abadi 等人引入，并被定义为“计算时间由访问内存所花费的时间主导”的功能[15]。主要思想是构建一种在内存中

分配大量数据（“暂存器”）的算法，该数据块可以相对缓慢地访问（例如RAM）和“访问不可预测的位置序列”。一个块应该足够大，以便保留数据比对每个访问重新计算更有优势。该算法还应该防止内部并行，因此 N 个并发线程应该一次需要 N 倍的内存。 Dwork 等人[22]调查并正式化了这种方法，导致他们提出定价功能的另一个变体：“Mbound”。另外一件工作属于 F. Coelho [20]，谁提出最有效的解决方案：“北海道”。 根据我们的知识，在大阵列中基于伪随机搜索的思想的最后一个工作是由 C. Percival [32]称为“scrypt”的算法。与以前的功能不同，它专注于关键推导，而不是工作证明系统。尽管这个事实，scrypt 可以为我们的目的服务：它在部分散列转换问题（如 Bitcoin 中的 SHA-256）中的定价功能很好。 现在，scrypt 已经应用于 Litecoin [14]和其他一些 Bitcoin 叉。然而，它的实现并不是真正的记忆限制：比率“内存访问时间/总时间”不够大，因为每个实例只使用 128 KB。这样可以让 GPU 采矿者的效率大概高出 10 倍，并且继续留下创造相对便宜但高效的采矿设备的可能性。 此外，scrypt 构造本身允许在存储器大小和 CPU 速度之间进行线性折衷，这是由于暂存器中的每个块仅来自前一个块的事实。例如，您可以存储每隔一个块，并以懒惰的方式重新计算其他块，即只有当它变得必要时。伪随机索引被假定为均匀分布，因此附加块重新计算的期望值为 1·N，其中 N 为迭代次数。总体计算时间增加不到一半，因为也有时间独立（恒定时间）操作，如准备暂存器和散列每次迭代节省 2/3 的内存费用 1·N + 1·2·N = N 个额外的重新计算;3 3 9/10 结果在 1·N +。 。 。 + 1·9·N = 4.5N。很容易显示只存储所有块中的 1 个 10 10 秒 增加小于 s-1 因子的时间。这反过来又意味着一台带有 CPU 的机器比现代芯片快 200 倍可以只存储 320 字节的暂存器。

5.2 提出的算法

我们提出了一个新的内存限制算法，用于工作定价功能。它依赖于随机访问缓慢的内存并强调延迟依赖性。与每个新的块（长度为 64 个字节）相比，都取决于所有先前的块。因此，假设的“记忆保护”应该以指数级增加他的计算速度。 我们的算法需要大约 2 Mb 每个实例，原因如下： 它适用于现代处理器的 L3 缓存（每核心），这几年应成为主流; 一兆字节的内部存储器是现代 ASIC 管道几乎不可接受的尺寸; GPU 可能会运行数百个并发实例，但是以其他方式受到限制：GDDR5 内存比 CPU L3 缓存慢，并且其带宽显着，而不是随机访问速度。 暂存器的显着扩展将需要迭代次数的增加，这又意味着总体时间增加。在无信任的 p2p 网络中的“重”呼叫可能会导致严重的漏洞，因为节点有义务检查每个新的块的工作证明。如果一个节点在每个哈希评估中花费相

当多的时间，那么可以通过大量具有任意工作数据（随机数值）的假对象来轻易地 DDoS。

6. 更多的优势

6.1 平滑的释放

CryptoNote 数字硬币总量的上限为：MSupply = 264-1 原子单位。这仅仅是基于实施限制的自然限制，而不是像“N 币对任何人都应该足够”这样的直觉。为了确保排放过程的平稳性，我们使用以下公式来获得积分奖励：BaseReward =（MSupply - A）18， 其中 A 是先前产生的硬币量。

6.2 可调参数

6.2.1 困难

CryptoNote 包含一个改变每个块的难度的定位算法。当网络哈希值急剧增长或缩小时，这会降低系统的反应时间，保持恒定的阻塞率。原始比特币方法计算了上一个 2016 块之间实际和目标时间跨度的关系，并将其用作当前难度的乘数。显然这不适合快速重新计算（因为惯性大）并导致振荡。 我们的算法背后的一般思想是将节点完成的所有工作归纳到所花费的时间。工作量是每个块中相应的难度值。但是由于不准确和不受信任的时间戳，我们无法确定块之间的确切时间间隔。用户可以将他的时间戳转移到将来，并且下一个时间间隔可能不太可能甚至是负的。大概会有这样的事件很少，所以我们可以排序时间戳和截断异常值（即 20％）。其余值的范围是 80％的相应块花费的时间。

6.2.2 尺寸限制

用户支付存储块，并有权投票。每个矿工处理平衡费用和收益利润之间的平衡，并设定自己的“软限制”来创建块。另外，最大块大小的核心规则对于防止块链被虚假事务充斥是必要的，但是该值不应该被硬编码。 令 MN 为最后 N 个块大小的中值。那么接受块大小的“限制”是 2·MN。它阻止块状物膨胀，但如果需要，仍然允许极限随时间缓慢增长。 事务大小不需要明确限制。它被块的大小限制;如果有人想通过数百个输入/输出创建一个巨大的交易（或者环形签名的模糊度很高），他可以通过支付足够的费用来做到这一点。 6.2.3 超大尺寸的罚款 矿工仍然有能力填补自己的零收费交易，最大限度为

2·Mb。 尽管只有大多数矿工能够调整中位值，但还是有一个可能使块链膨胀并在节点上产生额外的负载。 为了阻止恶意参与者创建大块，我们引入惩罚函数： NewReward = BaseReward· 。BLKSIZE0.2 MN - 只有当 BlkSize 大于最小可用块大小（10kb，MN·110％）时，才应用此规则。 当整体收费超过罚款时，矿工被允许创造“通常规模”的块，甚至超过利润。 但费用不可能增长二次不同于惩罚值，所以会有一个平衡。

6.3 交易脚本

CryptoNote 具有非常简约的脚本子系统。发送方指定一个表达式 Φ= f（x1，x2，...，xn），其中 n 是目的地公钥数目{Pi n。只支持五个二进制运算符：min，max，sum，mul 和 cmp。当接收方支付这笔款项时，他产生 0≤k≤n 个签名，并将其传递给交易输入。验证过程简单地用 xi = 1 评估 Φ，以检查公钥 Pi 的有效签名，xi = 0。如果 Φ> 0，验证者接受证明。 尽管简单，这种方法涵盖了所有可能的情况： 多/门槛签名。对于比特币风格的“M-out-of-N”多签名（即接收方应提供至少 0≤M≤N 的有效签名）Φ= x1 + x2 +。 。 +xN≥M（为了清楚起见，我们使用通用代数符号）。加权阈值签名（一些键可能比其他键更重要）可以表示为 Φ= w1·x1 + w2·x2 +。 。 。 + wN·xN≥wM。以及主键对应于 Φ= max（M·x，x1 + x2 + ... + xN）≥M 的场景。很容易表明，任何复杂的情况都可以用这些操作者表达，即它们形成基础。 密码保护。拥有秘密密码 s 等同于私钥的知识，确定性地从密码导出：k = KDF（s）。因此，接收者可以通过在密钥 k 下提供另一个签名来证明他知道密码。发件人只需将相应的公钥添加到自己的输出中。请注意，该方法比 Bitcoin [13]中使用的“交易拼图”更加安全密码在输入中被明确传递。 退化病例。Φ= 1 表示任何人都可以花钱; Φ= 0 表示输出永远不会消耗。 在输出脚本与公钥相结合的情况下，对发件人来说太大，他可以使用特殊的输出类型，这表示收件人将这些数据放在他的输入中，而发送者只提供一个散列。这种方法与 Bitcoin 的“付费 - 散列”功能类似，但是不是添加新的脚本命令，我们在数据结构级别处理这种情况。

7. 结论

我们调查了比特币的主要缺陷，并提出了一些可能的解决方案。这些优势和我们正在进行的开发使得新的电子现金系统 CryptoNote 成为比特币的一个严重的对手，超越了所有的叉子。 诺贝尔奖得主弗里德里希·哈耶克在他着名的作

品中证明，存在独立的货币具有巨大的积极作用。每个货币发行商（或我们案例中的开发商）正在通过改进他的产品来吸引用户。货币就像一种商品：它可以有独特的利益和缺点，最方便和值得信赖的货币有最大的需求。假设我们有一个比较优势的货币：这意味着比特币会发展得更快，变得更好。作为开源项目的最大支持来自于对其感兴趣的用户。 我们不认为 CryptoNote 是 Bitcoin 的完全替代品。相反，拥有两种（或多种）强大和便利的货币比只有一种更好。并行运行两个以上不同的项目是电子现金经济的自然流动。 安全

我们将给予我们一次性签名计划的证明。在某种程度上，它与[24]中证明的部分是一致的，但我们决定用参考来重写它们，而不是强迫读者从一张纸到另一张纸。 这些是要建立的属性： Linkability。给定集合 S 的所有秘密密钥{xi} i = 1，不可能产生 n + 1 个有效签名 σ1，σ2。 。 。 ，σn+ 1，使得它们都通过 LNK 相位（即，具有 n + 1 个不同的关键图像 Ii）。该属性意味着在 CryptoNote 的上下文中双重支出保护。 辩解能力。给定集 S，最多 n-1 个对应的私钥 xi（不包括 i = j）和密钥 xj 的图像 Ij 不可能产生具有 Ij 的有效签名 σ。该属性意味着在 CryptoNote 的上下文中的盗窃保护。 不可伪造性。只给出一个公钥集 S，就不可能产生有效的签名 σ。 Anonymity。给定签名 σ 和相应的集合 S，不可能以概率 p> 1 确定签名者的秘密索引 j。 可连接性

定理 1.我们的一次性环签名方案是可随机的 oracle 模型链接的。 证明。假设对手可以为任何 i，j∈[1]产生带有关键图像的 n + 1 个有效签名 σiii*f*=

Ij。 。 。 N]。由于#S = n，对于每个 i，至少有一个 Ii*f*= xiHp（Pi）。考虑相应的签名 σ=（I，c1，...，cn，r1，...，rn）。 VER（σ）=“true”，这意味着 Lr i = ritt + ciPi Rr i = riHp（Pi）+ cil ñ 。 ci = Hs（m，Lr，...，Lr，Rr，...，Rr）mod l l = 1 1 N 1Ñ 前两个平等意味着 一世 i.logG Lr = ri + cixi 日志 HP（PI） Rr = ri + ci log Hp（Pi）I 其中 logA B 非正式地表示 B 到基本 A 的离散对数。 如[24]所示，我们注意到$ i：xi = logHp（Pi）I 意味着所有 ci 都是唯一确定的。第三次平等迫使对手找到 Hs 的前一个形象来成功攻击 其概率被认为可忽略的事件。 辩解能力

定理 2.我们的一次性环签名方案在随机 oracle 模型中的离散对数假设下是可以忽略的。 证明。假设对手可以产生一个有效的签名 σ=（I，c1，...，cn，r1，...，rn），其中 I = xj HP（Pj）给定{xi | i = 1， 。 。 ，j-1，j + 1，... 。 。 ，n}。然后，我们可以构建一个解决 E（Fq）中的离散对数问题的算法 A. 假设 inst =（tt，P）∈E（Fq）是 DLP 的给定实例，并且目标是获

得 s，使得 P = stt。使用标准技术（如[24]），A 模拟随机和签名词，并使对手在集合 S 中产生 Pj = P 的两个有效签名： σ=（I，c1，...，cn，r1，...，rn）和 σr=（I，cr，...，cr，rr，...，rr）。 1 n 1 n rj -rt 由于 I = xj Hp（Pj）在两个签名中，我们计算 xj = logHp（Pj）I = ct 莫德 A 输出 xj，因为 Lj = rjtt + cj Pj = rr tt + cr Pj 和 Pj = P。 ĴĴ 不可伪造性

在[24]中已经表明，不可伪造只是可链接性和可排除性的一个含义。 定理 3. 如果一次性戒指签名方案是可链接的并且可以被取消，则是不可伪造的。 证明。假设对手可以伪造给定集合 S 的签名：σ0=（I0，...）。考虑所有有效的签名（由诚实的签名人生成）为同一个消息 m 和集合 S：σ1，σ2，。 。 。 ，σn。有两种可能的情况： I0∈{Ii n。哪些是违规的。 I0ƒ∈{Ii n。哪些与可链接性相矛盾。 匿名

定理 4.我们的一次性环签名方案在随机 oracle 模型中的决定性 Diffie-Hellman 假设下是匿名的。 证明。假设对手可以以概率 p = 1 + s 确定签名者的秘密索引 j。然后，我们可以构造算法 A，其解决了 E（Fq）中的决定性 Diffie-Hellman 问题，其概率为 1 + s。 2 2 令 inst =（tt1，tt2，Q1，Q2）∈E（Fq）是 DDH 的实例，并且确定是否 logG1 Q1 = logG2 Q2。 A 用有效的签名 σ0=（I，...）向对手提供，其中 Pj = xjtt1 = Q1 和 I = Q2 并模拟 oracle Hp，返回 tt2 查询 Hp（Pj）。 对手返回 k 作为他对索引 i 的猜测：I = xiHP（Pi）。如果 k = j，则 A 返回 1（对于“是”），否则为随机 r∈{1,0}。正确选择的概率是 如[24]所示：1 + Pr（1 | inst∈DDH）-Pr（1 | inst∈/ DDH）= 1 + Pr（k = j | inst∈DDH）+ 2 2 Pr（kƒ= j | inst∈DDH）·Pr（r = 1）-Pr（k = j | inst∈/ DDH）-Pr（kƒ= j | inst∈/ DDH） = 1 1 n-1 1 1 n-1 1 1 s 2 + n + s +（ n - s）·2 - n - n·2 = 2 + 2 事实上，结果应该由 Hs 中碰撞的概率来降低，但是这个值是 被认为是可以忽略的。 关于散列函数 Hp 的注释

我们将 Hp 定义为确定性散列函数 E（Fq）→E（Fq）。没有一个证明要求 Hp 成为理想的加密散列函数。主要目的是以某种确定的方式获得图像密钥 I = xHp（xtt）的伪随机数。 使用固定基数（I = xtt2），可能有以下情况： 爱丽丝向鲍勃发送两个标准交易，产生一次性 tx 键：P2 = Hs（r1A）tt + B 和 P1 = Hs（r2A）tt + B Bob 恢复对应的一次私人 tx 键 x1 和 x2，并花费具有有效

签名和图像密钥 I1 = x1tt2 和 I2 = x2tt2 的输出。 现在，爱丽丝可以链接这些签名，检查等式 I1 -I2 =（Hs（r1A）-Hs（r2A））tt2。 问题是 Alice 知道公钥 P1 和 P2 之间的线性相关，而在固定基地 tt2 的情况下，她也获得了关键图像 I1 和 I2 之间的相同关系。用 hp（xtt2）替换 tt2，它不保留线性，修复了这个缺陷。 为了构建确定性 Hp，我们使用[37]中提出的算法。

如果你觉得我的文章对你有用，或者想和我取得联系，你可以：
加微信好友：zzkklee
捐献门罗币：
43NAEMGwYAG3g8aEvLuhVtUrvMoWpCiWdQgykFsex2w6fa8N2PtW6o2U
DSpBuetfEZatYGo5wjLz47hX4LsuPbDeNb2h6uM

References

[1] http://bitcoin.org.
[2] https://en.bitcoin.it/wiki/Category:Mixing Services.
[3] http://blog.ezyang.com/2012/07/secure-multiparty-bitcoin-anonymization.
[4] https://bitcointalk.org/index.php?topic=279249.0.
[5] http://msrvideo.vo.msecnd.net/rmcvideos/192058/dl/192058.pdf.
[6] https://github.com/bitcoin/bips/blob/master/bip-0034.mediawiki#Specification.
[7] https://github.com/bitcoin/bips/blob/master/bip-0016.mediawiki#Backwards Compatibility.
[8] https://en.bitcoin.it/wiki/Mining hardware comparison.
[9] https://github.com/bitcoin/bips/blob/master/bip-0050.mediawiki.
[10] http://luke.dashjr.org/programs/bitcoin/files/charts/branches.html.
[11] https://bitcointalk.org/index.php?topic=196259.0.
[12] https://en.bitcoin.it/wiki/Contracts.
[13] https://en.bitcoin.it/wiki/Script.
[14] http://litecoin.org.
[15] Mart´ın Abadi, Michael Burrows, and Ted Wobber. Moderately hard, memory-bound func- tions. In NDSS, 2003.
[16] Ben Adida, Susan Hohenberger, and Ronald L. Rivest. Ad-hoc-group signatures from hi- jacked keypairs. In in DIMACS Workshop on Theft in E-Commerce, 2005.

[17] Man Ho Au, Sherman S. M. Chow, Willy Susilo, and Patrick P. Tsang. Short linkable ring signatures revisited. In EuroPKI, pages 101–115, 2006.
[18] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. J. Cryptographic Engineering, 2(2):77–89, 2012.
[19] David Chaum and Eug`ene van Heyst. Group signatures. In EUROCRYPT, pages 257–265, 1991.
[20] Fabien Coelho. Exponential memory-bound functions for proof of work protocols. IACR Cryptology ePrint Archive, 2005:356, 2005.
[21] Ronald Cramer, Ivan Damg˚ard, and Berry Schoenmakers. Proofs of partial knowledge and simplified design of witness hiding protocols. In CRYPTO, pages 174–187, 1994.
[22] Cynthia Dwork, Andrew Goldberg, and Moni Naor. On memory-bound functions for fighting spam. In CRYPTO, pages 426–444, 2003.
[23] Eiichiro Fujisaki. Sub-linear size traceable ring signatures without random oracles. In CT- RSA, pages 393–415, 2011.
[24] Eiichiro Fujisaki and Koutarou Suzuki. Traceable ring signature. In Public Key Cryptogra- phy, pages 181–200, 2007.
[25] Jezz Garzik. Peer review of “quantitative analysis of the full bitcoin transaction graph”. https://gist.github.com/3901921, 2012.
[26] Joseph K. Liu, Victor K. Wei, and Duncan S. Wong. Linkable spontaneous anonymous group signature for ad hoc groups (extended abstract). In ACISP, pages 325–335, 2004.
[27] Joseph K. Liu and Duncan S. Wong. Linkable ring signatures: Security models and new schemes. In ICCSA (2), pages 614–623, 2005.
[28] Ian Miers, Christina Garman, Matthew Green, and Aviel D. Rubin. Zerocoin: Anonymous distributed e-cash from bitcoin. In IEEE Symposium on Security and Privacy, pages 397– 411, 2013.
[29] Micha Ober, Stefan Katzenbeisser, and Kay Hamacher. Structure and anonymity of the bitcoin transaction graph. Future internet, 5(2):237–250, 2013.
[30] Tatsuaki Okamoto and Kazuo Ohta. Universal electronic cash. In CRYPTO, pages 324–337, 1991.
[31] Marc Santamaria Ortega. The bitcoin transaction graph — anonymity. Master’s thesis, Universitat Oberta de Catalunya, June 2013.

[32] Colin Percival. Stronger key derivation via sequential memory-hard functions. Presented at BSDCan’09, May 2009.
[33] Fergal Reid and Martin Harrigan. An analysis of anonymity in the bitcoin system. CoRR, abs/1107.4524, 2011.
[34] Ronald L. Rivest, Adi Shamir, and Yael Tauman. How to leak a secret. In ASIACRYPT, pages 552–565, 2001.
[35] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph.
IACR Cryptology ePrint Archive, 2012:584, 2012.
[36] Meni Rosenfeld. Analysis of hashrate-based double-spending. 2012.
[37] Maciej Ulas. Rational points on certain hyperelliptic curves over finite fields. Bulletin of the Polish Academy of Sciences. Mathematics, 55(2):97–104, 2007.
[38] Qianhong Wu, Willy Susilo, Yi Mu, and Fangguo Zhang. Ad hoc group signatures. In IWSEC, pages 120–135, 2006.