MOCHIMO

后量子计算时代货币

概述

加密货币的一种创新性的运转方式。这个方式引入抗量子攻击的交易网络、真正的去中心化，超快的交易处理，前所未有的汇聚时间和无于伦比的可扩展性，并且这些全部在消费级设备上无需信任第三方的点对点网络上实现 。

Matt Zweil
mzweil@mochimo.org

目录

1 前言和致谢

2017年夏天,Mochimo开发团队刚开始开发Mochimo时，我们都同意在开发出可用的产品前，应尽量不发布白皮书。 今天,我们不仅仅有了一个加密货币，$MCM，我们还有了一个可向他人授权的一组健全、强大的算法。这些算法允许任何人在可以保证长久运转、永远 不变并且去中心化的高适应力协议上，创建下一代加密货币。

这里包含了对Mochimo加密货币引擎(Mochimo Cryptocurrency Engine)目前情况的描述。 这不是未来计划，而是一个介绍我们是如何已经解决当今困扰加密货币的许多关键问题的文档。 我们已经实施了一系列创新方法，AI和革命性算法，我们将它们一起统称为Mochimo协议。

MCM项目对Andreas Hülsing博士和Daniel Bernstein博士在后量子时代密码学领域的开创性工作充满感激。对Hülsing，具体的说，我们非常感谢您审核我们的加密代码。另外感谢Albert-Laszlo Barabasi对随机网络领域（Field of Random Networks）的贡献。 他的工作也影响了我们的竞争解决和分布网络的基础机制。最后，致比特币之父中本聪：无论你是谁，在哪里：我们都感谢您关于去中心化货币的愿景。我们很自豪能够将您的工作向前推进，以创造真正可扩展且可持续的加密货币。

2 执行摘要

Mochimo [$MCM] 一个第三代加密货币和交易网络。我们从头开始构建，来避免现有区块链系统中的已知问题和缺陷。为了将目前行业的最佳功能都组合到一个加密货币生态系统中，Mochimo由一个空白页面开始，从零编写代码。而且这个加密货币生态系统，能够适应未来，不会过时，因为它有后量子加密算法来保障其长期的安全。作为该协议的一部分，该货币采用了随机化点对点网络、新的共识机制和独特的工作量证明挖矿技术。这些技术组合在一起，成为一个无需信任第三方的分布式帐本。最重要的是，为Mochimo货币开发的各种算法包括多种创新和功能，为当今困扰现有和新型区块链的一些最关键问题提供了已知有效的解决方案。

以下简短地列举了一些我们的创新之处：

1. ChainCrunch™ 技术
   这个专有的技术将区块数据的总大小降低，保证可扩展并处理大量交易的能力（6.75年内从现有的1000 TPS扩展到20000 TPS)；对Mochimo来说，不论是短期内还是长期，扩展(scaling)不是问题。
2. Trigg’s 算法
   专有的工作量证明算法，可以确保FIFO交易以固定的交易费用处理。 将无限期地为所有级别的矿工保持挖矿的可行性。
3. Mochimo 共识机制
   基于随机网络模型构建的新系统，允许高速汇聚，孤立链修剪(orphaned chain pruning)。以及数学上可证明的共识，这个共识优于许多加密货币使用的的谣言共识方式（consensus-by-rumor)。
4. 量子计算方面的安全性
   通过部署由欧盟资助的PQCRYPTO研究组织审查过的WOTS + 来保障Mochimo地址的安全， 并通过让整个MCM协议基于量子安全算法，Mochimo开发团队解决了加密货币存在的一个关键问题。这个问题最终将导致基于ECDSA的协议，如比特币，以太坊和所有ERC-20代币在功能上无法安全运转，不论是用作交易网络还是价值存储。
5. 公平的分发方式
   最少量预挖给开发团队 ，没有ICO，不断自我调节和恒定的挖矿难度， 以及缓慢减少的区块奖励的保护性措施等内置的防护因素，一起保证了MCM币的公平分发，而且可以保证“更迟”接触到MCM 的人更容易加入。

2 执行摘要

Mochimo开发团队由系统架构师Matt Zweil领导，他是一位专家级网络架构师，设计并部署了业内交易网络，数据中心设计和服务提供商网络领域的一些最艰巨的项目。 Mochimo的主开发人是Trigg，他是C语言的大师级程序员和人工智能研究员，自70年代末以来一直致力于开发具有创新性的系统。 他们共同创造了MCM协议和ChainCrunch™技术。 在更大的Mochimo开发团队的协助下，Matt和Trigg已经实现了一个已在运转的协议，这是本文档的主要内容。

3 MOCHIMO 协议设计理念

从2009年起投资了第一波和第二波加密货币，并对其做出贡献后，2017年初，一伙区块链领域的元老聚在一起，启动了Mochimo项目。 作为加密货币纯粹主义者，他们的首要任务之一就是提炼总结出一套决定加密货币设计的原则 。

MCM的起始目标是成为一个适应未来的加密货币。这样的货币本质上是真正去中心化的，无需信任第三方， 永不可变并且可无限扩展没有上限。 我们的协议已经实现了这四项并且之外还实现了更多。

总之，Mochimo加密货币网络是一个点对点、无信任(trustless)的分布式账本，具有高速汇聚和强大的双花保护(Double-spend) 。 Mochimo不是其他区块链的分叉，而且简单地重用市面上已有的代码实现不了Mochimo目前所能做到的。 相反，Mochimo加密货币生态系统是通过新的代码对中本聪式区块链、分布式帐本的重新部署。重新部署的区块链不仅是基于中本聪最初的愿景，还通过多年来的经验性更新得到增强。在这些原则中，最至关重要的是去中心化。

现如今，加密货币基本分为两大类:

1. 真正的去中心化并且因此无信认(Trustless)
2. 半中心化

T所有半中心化加密货币都应该被抛弃，没有例外。任何事物的中心化比如：交易所，信用信息，货币等都会吸引攻击。与区块链的本质： 独立的帐本相反，中心化的权威，不论它们是对是错，都基本上决定着网络上帐本的当前状态。

当然，有些人可能会把这个缺陷称作好处。为了转移大家对其架构中固有缺陷的注意力，中心化系统经常向大家展示超高的交易吞吐量数据。他们未能提到这样做的代价相当于割掉最核心的无信任(trustless)环境，这像心脏一样关键。实际上，每秒过高的交易数量（“TPS”）的广告经常 初步说明了系统创建者的实际意图有可能是想对投资者的资产有更多的控制。

为什么“快速 ”意味着“控制”？一个中心化的权威机构可以处理数十/百万的TPS，这不是因为他们的共识机制有多高效，而是因为所有的共识都被忽略了。用政治来类比，独裁政权是高

3 MOCHIMO 协议设计理念

效的，但这个政府不再真正代表人民，也不再为人民服务。当前加密货币实现交易速度可扩展的障碍为： 目前的共识机制的速度是瓶颈。 那么我们如何在不将控制交给中心化权威的情况下保持或提高速度呢？ 我们认为，主要的设计挑战如下：

- 确保区块链大小不会大到失去控制
- 维持通信的带宽要求，让普通人容易加入。任何人都应该能够简单轻松地创建节点并加 入网络，完全同步账本，并开始处理交易并挖区块。
- 允许快速传播交易，区块更新和快速汇聚，所有这些都通过有效且可在数学上证明的竞争解决方案实现。

为了解决这些问题，Mochimo生态系统引入了一些创新，其中最主要的是“ChainCrunch™”， 它允许任何单个节点在丢弃旧区块的同时不影响查看整个区块链。Mochimo还具有极快速的汇聚速度和对孤立链的修剪(Pruning)。为了与这一愿景保持一致，Mochimo有加密货币世界中一些说明文档最详尽的代码。

3.1 自主性去中心化

我们的第一信条是，一个加密货币要做到真正的去中心化，在发布后，任何参与者都不应当有能力控制其政策方向，无论矿工还是其开发者。因此，管理系统的法则全部是代码，并且没有人应该控制这个代码。

因为这个原因，Mochimo拒绝目前一些新的加密货币所有尝试的方法：可信节点(trusted nodes)、投票机制、权益证明(POS, Proof of Stake)或代理权益证明(DPOS, Delegated Proof of Stake)。不仅如此，我们完全拒绝挖矿算力的集中，因为这将允许这些参与者通过蛮力控制政策。不论是什么共识机制被用于加密货币中，只要它允许某一个参与者获得比其他参与者更多的影响力，都会导致那些处于权力位置的人的权力得到进一步集中。 最终这将破坏区块链网络的自治。这种操纵已经在几乎所有现有的加密货币中发生，我们可以将此统称为中心化趋势。它有可能导致这些区块链的长期消亡 。

3.2 自我修复式账本

Mochimo团队认为，每个单独的网络节点必须能够在不需要请求权威来源的情况下，确定网络的状态，账本以及任何指定的交易。 需要信任单一权威，这是所有自治系统的短板。

3 MOCHIMO 协议设计理念

出于这个原因，Mochimor的设计拒绝“主节点”，“超级节点”，“可信节点” 和所有其他类似概念。这类方法的设计将让中心化的权威有能力 管理区块链网络的行为。 简要地说： 如果区块链网络需要以任何形式依赖于参与者的互相信任才能运转，那么该账本是可变的。

如果区块链是可变的，那么这个加密货币实际上是一文不值的。

相反，当MCM网络内出现冲突时，节点本身将通过数学确定主链并独立自主地修剪任何孤立链。同样，交易和区块解决方案不通过中心化的节点中继，而是通过多播机制(Multicast mechanisms)点对点传播到整个网络。

Mochimo协议的每一环都拒绝中心化处理，交易分发(transaction distribution)、节点探测（Peer detection)、争用解决(Contention resolution)，投票，Tie-breaking, 代码版本实施(code version enforcement)。 这些机制都是薄弱环节。 它们创造了让账本(Ledger)可变的机会， 其中任何一个都可能会因为某群人为了改变帐本的结果和状态，而被滥用，从而导致其变得可变。 只有通过运行一个无需信任的系统(就是说，每个节点都有做决定并达成共识所需要的所有信息)， 加密货币的终端用户才能真正信任交易网络。

3.3 平等的挖矿门槛

扩展(Scaling)是当今第一代和第二代加密货币面临的最大问题。目前大部分市值前100的加密货币对计算，带宽和存储要求越来越高。挖矿需要的资源需求增长速度大多超过普通人的硬件迭代速度。随着挖矿难度的增加，普通人再也无法参与挖矿。

一旦挖矿被有效地限制于具有足够资源来支撑高性能计算和存储资源的大型组织中， 加密货币的中心化控制就开始了。 因为挖矿是中心化的， 所以节点也被矿工中心化。 通过这，这些矿工们开始控制加密货币的开发方向，并推迟或阻止任何降低其挖矿收益的创新。 在此之后不久，恶意分叉和其他有害行为就随之而来。

类似地，没有人喜欢总是处于没有权利可言的位置 ，而且如果所有决定总是不利于其他参与者，最终他们会抛弃这个加密货币，去支持更平等的选项。

随着中心化的大的控制者不断失去新的和小的参与者。 这个加密货币的目标不再是去创造一个全民日常使用的可持续的价值存储媒介(Store of Value)，而是去代表最大的矿工的利益。

用户将始终能够挖Mochimo，并且不会被不良参与者边缘化。

4 MOCHIMO 的设计解决了许多问题

随着加密货币变得成熟，大规模应用导致各种问题出现。很多第二代加密货币试图解决其中一个或更多问题。与那些只解决一部分问题的协议不同，Mochimo团队开发出了统观整体、未雨绸缪的解决方案。这个解决方案通过结合一系列加密货币设计上的创新得以实现。以下所有问题因而得到解决：

- 量子计算机的威胁
- 网络可扩展性的长期解决方案
- 通过转账费用方面的创新以确保交易处理先进先出(F.I.F.O.)
- 交易吞吐量和安全性

4.1 量子计算机的威胁

Mochimo解决的第一个也是最值得注意的问题是迅速发展的量子计算机对加密货币生态系统的威胁。 目前大部分区块链系统和加密货币的钱包地址和余额是通过量子不安全的数据签名算法保护 。 目前最常被使用的是ECDSA(即椭圆曲线数字签名算法，比特币、以太坊和所有ERC20代币都使用此算法)， 量子计算攻击ECDSA像戳纸一样简单。

尽管对传统计算机来说破解ECDSA是计算死结（Computationally intractable)。 ECDSA的破解是量子计算机的最先的发展目标之一。 鉴于此，这些使用ECDSA的加密货币是暴露于风险中的，尽管它们本身已是了不起的成就 。 因而他们实际上不可被视作长期的价值存储媒介。 而且，也没有什么神奇的补丁可以加到他们的代码中，使它们变得可防量子攻击。

这是防量子攻击能力需要从零开发的原因，也解释了我们为什么不以任何现有的区块链项目为基础借鉴、分叉或设计。 一些意识到ECDSA量子方面缺陷的组织和个人已经开始编写 “后量子时代安全”的新加密协议标准。 其中最值得一提的是欧盟资助的PQCRYPTO工作组。 他们的”ICT-645622”文档是基于量子计算机目前和预期能力的量子安全加密算法参考标准。

该标准中，他们推荐了一些抗量子数字签名算法。量子领域计算能力的提高并不影响这些算法的计算不可解性(Computationally intractable)2。 Mochimo开发团队这些算法中选择了使用Winternitz一次性签名的WOTS+版本的XMSS+(由Andreas Hulsing于2017年9月提出并证明)3。

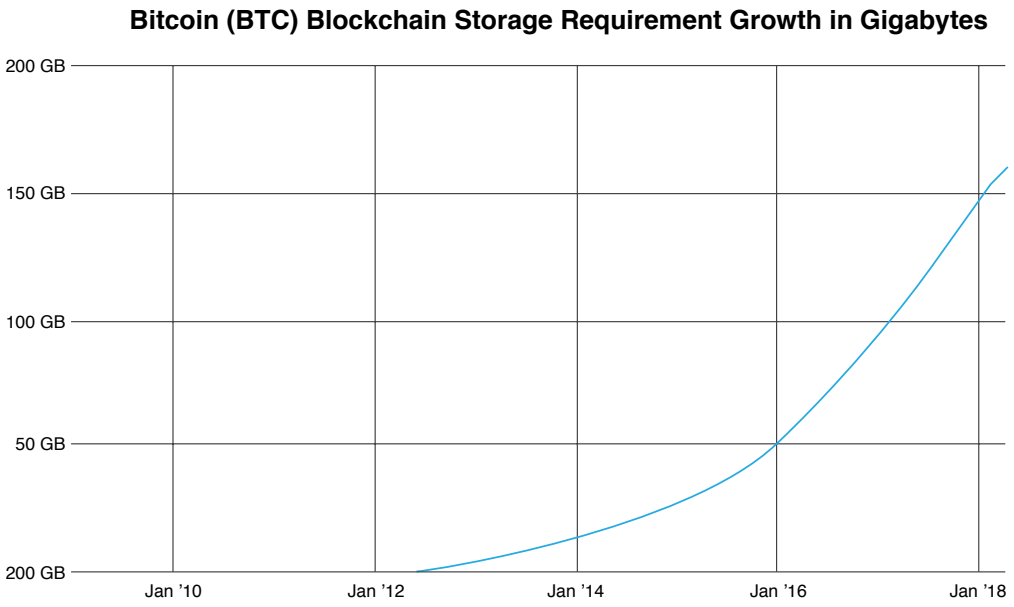
> Mochimo开发团队在2018年2月与Hulsing签定合同，让他对我们的加密代码全面审核、评价。Mochimo项目中WOTS+数字签名算法的应用，以及针对协议和钱包地址抗量子计算算法的C语言实现代码（C语言版本ANSI-C,1989），已被该算法的创造者本人Andreas Hulsing全部审核、评价过。我们的代码没有被发现缺陷，并且Hulsing’s最后对代码的评价将和白皮书一起在Mochimo网站上发布。

4 MOCHIMO 的设计解决了许多问题

4.2 网络可扩展性的长期解决方案

Mochimo是一个交易网络；因此，交易网络的可扩展性(Scalability)是一个首要考虑因素。 许多协议都不计后果地允许其区块链区块大小无限地增长，通常增到数百GB，甚至TB。

以比特币4为例，我们可以清楚地看到其区块大小每12-16个月就翻一倍。地址和签名数据相对较小的比特币区块，一直以指数级曲线增长。截至2018年3月，原始数据已超过180GB。众所周知经常网络拥堵的以太坊有过的多次网络暂停，都是由更快得多的区块大小增加速度导致，它的区块大小（包括EVM trace数据）现在必需存储空间已经超过1TB。



其他加密货币在网络需求和区块大小增加的情况下，这方面不会更好。鉴于这种交易网络拥堵问题，很容易理解为什么可能存在实施量子安全措施的阻力。地址和签名数据大小比比特币或以太坊等协议的 大一个数量级。

尽管存在着这种拥堵问题，这些加密货币开发者总是声称摩尔定律预测的未来技术发展速度跟得上区块链区块大小的增加速度。诉诸摩尔定律根本上是错误的，因为加密货币批评者们凭经验实际观察到，几乎所有区块链的存储消耗方面增长值都超过了摩尔定律18个月周期的技术发展目标值 。

4 MOCHIMO 的设计解决了许多问题

不过Mochimo已经通过名为ChainCrunch™的创新性区块链处理算法解决了区块大小增长过大并失控的问题。ChainCrunch™是Mochimo专有技术。它可以让用户运行完整的节点，但只保留历史区块数据的一小部分。ChainCrunch™受HASH256保护 ，并对量子计算攻击免疫。因为这项创新(这个文档后面将详细介绍) -Mochimo区块链大小不会增长。 相反，无论区块链网络运行了多少年或我们处理多少交易，它的大小都完美稳定于一个很小的值 。

革命性的ChainCrunch™ 方案不止以上面的方式解决扩展性(Scaling)问题。 它还允许系统节省存储空间，并极大地改善数据查询速度 。 使用ChainCrunch™，新节点可以加入网络并在几分钟内完全同步， 而不是几天或几周。该技术无需大量存储空间即可实现前所未有的交易速度扩展级别。此外，MCM协议中有一系列创新可以几乎实时地查询地址余额和交易数据。

4.3 通过转账费用方面的创新以确保交易处理先进先出(F.I.F.O.)

现有区块链交易费系统存在的问题是它们激励矿工优先处理愿意支付更高手续费的用户的交易，本该是平等、去中心化的网络因此受到了破坏。矿工选择高手续费的交易，会导致支付了低手续费或零手续的交易在繁忙的内存池中等待数小时。(内存池即Mempool，或 Memory Pool， 用于临时存储未确认的交易)。比如以太坊上的ICO进行时，用户会为了插队抢先完成交易，付夸张、惊人的高额手续费。这种用户行为 不利于可靠、可扩展的区块链系统的发展。 只有改变挖矿激励方式才能让其停止 。

Mochimo 开发团队认为激励矿工优先处理某类交易起的作用相反，无益于健康的网络。因此，Mochimo协议采用了一种新的费用不变的方法处理交易 。Mochimo网络上发送交易的手续费始终是: 0.000005 $MCM.这样描述可更直观地理解交易手续费有多小： 假设Mochimo的市值超过了所有替代货币(Altcoin)总市值，1个MCM便值25000美元，这时的交易手续费仍然小于 0.13美元。因而，我们可以放心地说，MCM的交易成本在很多年内都会小到微不足道。这有利于MCM的日常使用。

那么如何激励矿工？ Mochimo小额固定的交易费会鼓励矿工尽可能多地将交易堆到他们的候选区块中，这将确保交易以先进先出(F.I.F.O.) 的方式完成 ，并防止其他加密货币中会看到的内存池 (MEMPOOL) 排队现象 。

4 MOCHIMO 的设计解决了许多问题

4.4 交易吞吐量和安全性

Mochimo协议上的每笔交易都必须有以下6个基本元素：源地址（Source address），目标地址（Destination address），变更后地址（Change Address)， 已发送金额(Amount sent)，矿工费（固定不变）和余额变更后金额 (Balance Change Amount)。

它还有以下重要限制：源地址在使用后总是被清空和销毁。系统会检查确认已发送余额、余额变更后金额、矿工费之和等于源地址余额。这样，每笔交易的所占字节大小固定 ，输入输出简单到可忽略不计，对丢失的币的保护也非常简单。

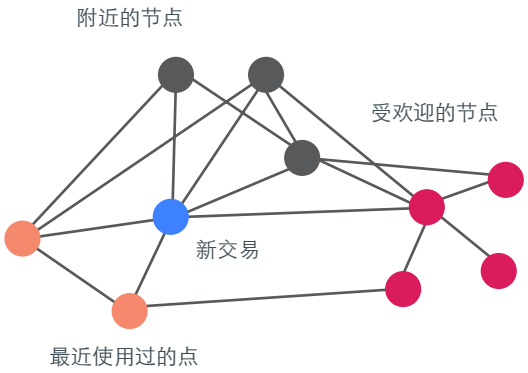
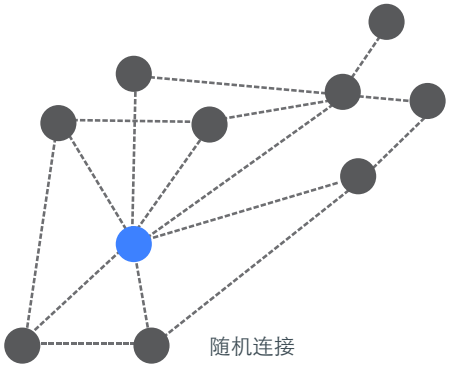
此外，如果不对现有源地址中的所有货币进行完整记帐，则无法发送交易。也不能汇总多个输入和输出。地址的所有钱包和余额管理都是钱包软件的客户端功能。

与Mochimo的ChainCrunch™技术实现的非凡速度增长相结合，我们网络上交易的查询， 验证和执行速度是业内最快的。 更重要的是，因为有ChainCrunch™，随着网络规模或交易吞吐量的增长，速度不会减慢或停滞。相反，交易速度会一直保持很快，并且随着普通节点的处理硬件的增加，它将超出其初始能力。

所以Mochimo速度很快，并且只会随着时间推移变得更快。

5 主要技术概述

5.1 使用随机网络模型的连接管理

普通 区块链 整个区块链网络上非随机散播信息	Mochimo 区块链 整个区块链网络上随机散播信息
	
大多数区块链仅选择附近的，受欢迎的或最近使用的节点(Peer或Node)。从更小的类似的节点组中重复选择允许各种方式来攻击网络	Mochimo网络上的每个节点在整个网络上随机打开和关闭连接。服务器随机让目前的16节点清单开始第二度连接， 每个节点都尝试连接，并且失败的连接总是被修剪掉(Pruning)。对于每个失败连接，服务器向一个成功的二度连接发送、执行GETPEER，随机化获得的结果，并继续让目前节点列表开始唯一的第三度连接

MCM连接管理的设计目标不是仅连接到最近的，知名的的或地理上相近的点(Peers，或Nodes)，而是保证一个无信任的环境， 以此让Mochimo网络上的每个节点在整个网络上随机打开和关闭连接。随机网络协议(Random Networks)旨在快速和异步地将所有收到的交易信息传播到网络的其余部分。Albert-Laszlo Barabasi的在他的网络科学出版物：Random Networks5中正式分析了随机网络理论 (Random Network)5。

过程：服务器系统上准备好候选的知名节点列表。这些只是该版本系统自带的标准节点。通过命令行下使用 -S开关，该节点列表可被用户提供的节点覆盖。用户可以提供一个用于初始连接的特定节点，或是包含一系列节点的文件。虽然Mochimo开发团队设置了那些知名的节点，它们与其他节点没有不同。

5 主要技术概述

> 轮换连接的随机化的网络意味着开放网络上两个节点(Node)之间的分隔度是16^D = N， 其中D是分隔度，N是网络上的节点数。例如，有恰好4096个节点的网络中任何两个节点之间的平均分隔度将是16^3 = 4096或3个跳转(Hop)。对于65536个节点的网络，平均增加到4个跳转。

初始化时，服务器加载知名节点或用户提供的列表，将其随机化，并随机选择一个准备连接的点。连接时，服务器执行OP_CODE：GET_PEER，如果成功，则在第一度连接中获得一个复制的最近节点列表。由此列表，服务器随机让目前的16节点清单开始第二度连接， 每个节点都尝试连接，并且失败的连接总是被修剪掉(Pruning)。对于每个失败连接，服务器向一个成功的二度连接发送、执行GETPEER，随机化获得的结果 ，并继续让目前节点列表开始唯一的第三度连接 。

连接并更新节点列表的过程不断重复，直到当前节点列表中包含16个与初始知名节点至少2度或3度相隔的活动节点。当前节点超时并无法连接时，四度连接会以同样的方式循环接替当前节点列表，即通过从上一次添加的节点中获取节点列表。这样，每个Mochimo服务器会永远不断地循环连接全世界新的和不同的节点。

此外，此随机网络理论可用于数学上确定Mochimo网络上任意两个节点之间的平均分隔度。我们有意这样设计：随机在网络中抓取新节点，并且每个Mochimo服务器动态地确保交易镜像和区块分配的快速传播。

那么如何扩展？使用这个网络类型，假设节点数量N 等于目前比特币大概的节点数，循环、随机选择节点数 k=16，那么Mochimo网络上任意两节点间平均分隔度会在3到4之间。这意味着，交易信息扩散，区块发现消息传递，全网范围汇聚需要最多3到4次跳转来获得相关数据6。

5.2 交易镜像

交易镜像的设计目标是利用随机网络模型(Random Networks model)。

要理解镜像在哪里起作用，首先要理解主服务器进程中维持一组循环更新的当前节点，接收多个交易请求，让其排队，并验证签名，然后生成子进程以执行交易镜像。

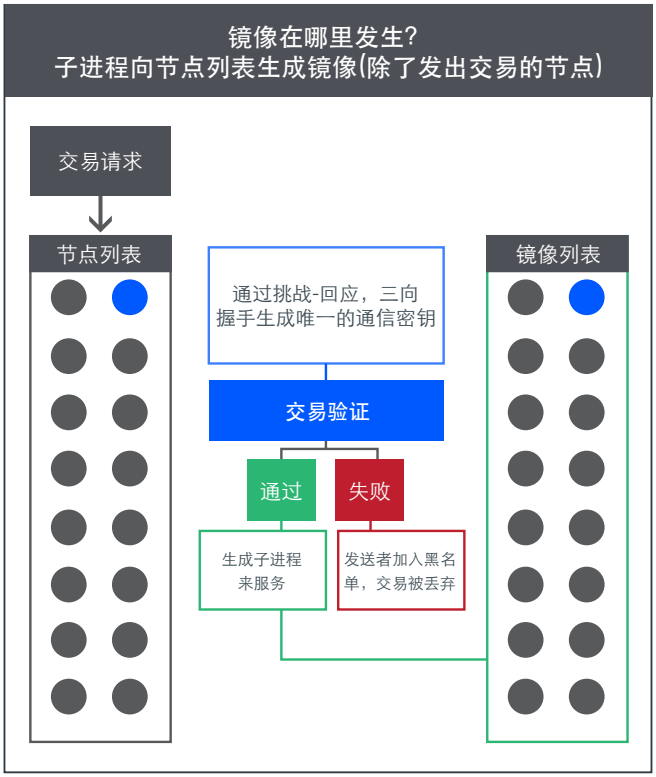
5 主要技术概述

过程：服务器对16个最近的节点(Peers)维持一组节点连接,(又称：当前节点表。英文：CURRENT-PEER table)，并查看每个节点的是否有外来请求。通过挑战-回应(Challenge-response)，三向握手(three-way handshake)，服务器为每个外来请求生成唯一的通信密钥。

通过OP_CODE: OP_TX，从一个节点接收到一个交易信息。服务器解析这个交易信息，并将其传递给到交易验证函数 (Transaction Validator function) (TX_VAL)。

TX_VAL可验证一些参数L，包括如下：节点传播的当前区块是否与本地系统相同？ 根据本地区块链帐本(Ledger)交易参数是否有效？然后它执行重复检测；如果所有交易参数没问题，则执行签名验证。

如果通过验证，TX将被移至CLEAN_TXQUEUE。如果TX未通过验证，则会被丢弃，发送者也将被加入黑名单。 对于移至CLEANTXQUEUE的交易，服务器异步生成子进程来服务CLEAN_TXQUEUE。子进程通过以下方式为其服务： 在当前节点列表(CURRENT_PEERLIST)中为最高到16个节点形成交易镜像（除了发出交易的节点），然后退出。与此同时，服务器继续其正常的处理循环。TX镜像通过外来‘源-哈希’(inbound source-hash)验证来保证不向交易发起者生成镜像。



5 主要技术概述

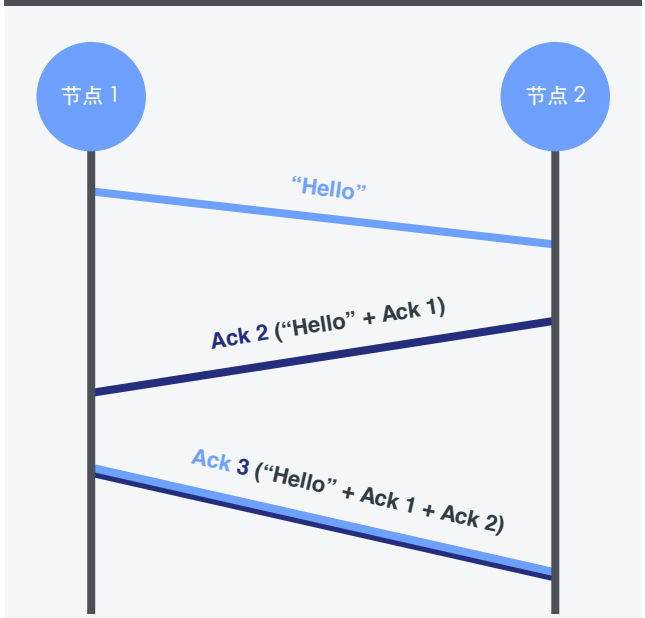
5.3 三向握手

所有点对点的通信中，Mochimo都会部署一个防止拒绝服务攻击(英语：denial-of-service attack，缩写：DoS)、幌骗攻击(Spoofing) 和中间人攻击(man-in-the-middle attack)的安全功能。 这种三向握手可以在许多网络协议中找到; 以下是它在Mochimo中的工作原理。

过程： 对于每个新的外出连接(Outbound connection)，Mochimo服务器会在一个Hello消息中生成随机的16比特(bit)的标识符。该标识符 称为ID1。接收的节点(Peer)响应的信息是包含ID1的Hello-ack和随机生成的识别唯一识别符ID2。原来发送ID1的节点发送 包含ID1和ID2的Hello-ACK-ACK,完成三向握手，这样，两个节点就“完全相邻”(Fully Adjacent)。通过添加标识符，它们现在可以这一会话期可以互相发送和接收信息。

每次标识符是唯一的，只在那个短暂的会话期有效，这样可以防止通过伪造节点(node)进行的拒绝服务攻击(DoS)和中间人攻击。 因为每次交易时会重新生成唯一的ID1和ID2，Mochimo的三向握手提供快速、简单和一次性的安全防护。

三向握手
任何不含Ack1+Ack2的信息都会被忽略



任何不含Ack1+Ack2的信息都会被忽略

任何节点接收的信息如果：

1. 不是三向握手状态机的一部分，或
2. 没有正确的ID，就都会被直接忽略。

5.4 快速服务器初始化

Mochimo服务器刚上线时会处于两种状态: 纯净启动(Clean Boot) 或优雅重启(Graceful Restart)。只有终端用户从监视器关闭系统时，服务器才会进行优雅重启。所有其他情况下，如果系统下线(比如因为解决争用,剪除孤立链,或因某种致命错误下线)，服务器的所有状态信息都会被清除。这包括节点列表,本地区块链帐本,全部区块数据,磁盘上的任何候选块等。

5 主要技术概述

过程：t任何时候确认有损坏(Corruption)或争用(Contention)问题，Mochimo系统这样维持稳定性：先纯净启动，然后利用Mochimo快速汇聚的特点重建区块文件和本地帐本。因为Mochimo上争用极少见，因此软件强制的纯净启动是不常有的。

纯净启动时，系统初始化其主要IP列表，并创建最小2度和3度的随机节点(Peer)列表。

然后服务器用MCM专有的Quorum函数在节点列表里存在的节点中识别最长的区块链(即主链)。最长的链执行的工作量最多。为了识别它，Mochimo节点会维护一个历史区块尾方件(Trail File)，上面有一直到最初的第一个区块(即创世区块，Genesis Block)的区块链信息。这个尾文件里包含了所有已解决的区块的条目(每个100字节) ，它们组成了一个可验证的区块解决过程记录的链表。(区块解决过程记录了，包括Nonce、Hash和用于获得每个区块难度要求的时间签名)。

这个文件会被从头到尾遍历。对于每个已解决的区块，其解决难度会被加入计数器(Counter)。T-File链中已解决区块的权重等于2^(难度 - 1)，意味着难度35的已解决区块的权重是难度34区块权重的2倍。因为T-File链是连接着的(可证明)，并且包含X难度的已解决区块信息，通过请求节点(Peer)的T-File，我们可以计算这个节点的从第一个区块到最新区块所执行的总工作量。如果这条链的总权重是网络上可见的最高值，那么这个节点(node)就在主链上，并且是可临时信任的。

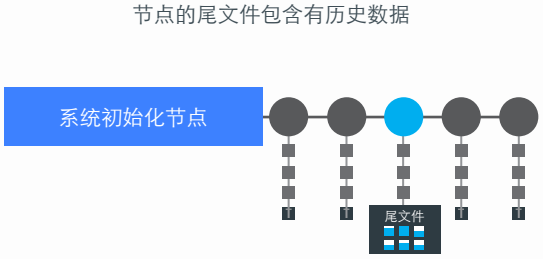
5.5 CHAINCRUNCH™背后的技术

Mochimo最重要的专有创新之一，ChainCrunch™，可以让区块链网络维持一个可完整查看的帐本，以完美的准确度验证、确认区块，并在短短几分钟(而不是几周)内重建一个新或现有节点的本地数据库。此外，ChainCrunch™ 允许极快查找交易确认信息的同时，保持很小的区块大小和存储要求(与其他加密货币所需存储要求相比，只是一点点)。

5 主要技术概述

过程： 一个Mochimo “Aeon” 是256个区块，每个区块的解决时间间隔为337.5秒，所以Aeon的平均生命周期为 86,400秒，即一天。Mochimo服务器在磁盘上维护着一个本地数据库，它是网络上所有有余额地址的排序列表(按地址索引)。具体情境中描述 ：系统通过处理有效交易并对每个成功交易后的变化进行排序，来对该数据库进行添加和增量(根据从区块1到区块255)。 到达区块255时，系统将数据库作为区块256写入磁盘，给它哈希值 ，并附加尾部信息(Trailer)。 这个特殊区块称 为“新生区块(或Neogenesis 区块，英文：Neogenesis block)”。从新生区块开始，系统继续解决区块257。

主链的发现
第二部分: 系统在对比尾文件(Trailer File)后确定工作量最多的区块链



服务器利用MCM的Quorum函数遍历尾文件来获得每个区块的难度。拥有最多工作量的区块链在主链发现过程获胜。

从这个新生区块写入磁盘的那一刻起，系统就不再需要来自区块链的所有历史数据来运行。

新生区块可以让系统丢弃链上的几乎所有区块，但仍然以完美的准确度验证交易。使用ChainCrunch™和新生区块，系统将永远不必在磁盘上存储超过512个块，因为重新启动新系统需要从第一个先前的新生区块重建(就是说，不是当前Aeon的新生区块)。

当新系统上线或由于未通过争用检查而重新启动时，系统将使用随机网络模型查找主链上的选定的一组(Quorum)节点(Peers)。从那时起，Mochimo服务器向Quorum中的一个随机成员发出请求，并下载先前的第一个新生成区块。

由于每256个区块产生一个新的新生区块，所请求的区块序号将是网络上的当前区块减去(最多)511个区块。因此，完全同步Mochimo节点所需的区块数将始终保持在257和511区块之间。如第16 页的图中所示，无论您在第二个Aeon中的哪个位置，您与先前的新生区块之间的距离总是小于512个块。

5 主要技术概述

服务器下载的未受信任的新生成块包含原始帐本(ledger)文件，列出Mochimo网络中每个有余额的地址的余额。Mochimo服务器以未受信任的状态导入此帐本(Ledger)，并使用它来验证链中随后的256个区块(每次从 Quorum 群中提取一个区块，以此帐本对其进行验证，同时根据需要进行调整，并在本地尾文件中进行构建)。

到达第256个区块时，服务器生成其自己的新生区块（该区块以临时可信的状态存在），并开始验证 当前Aeon中的所有现有区块，直到到达与Quorum 相同的当前区块，并与其验证区块的哈希值。从那时起，服务器等待由非Quorum成员解决新块，如果该块可以针对服务器的本地状态进行验证， 则称服务器为“已同步”，并且本地状态信息现在被信任。

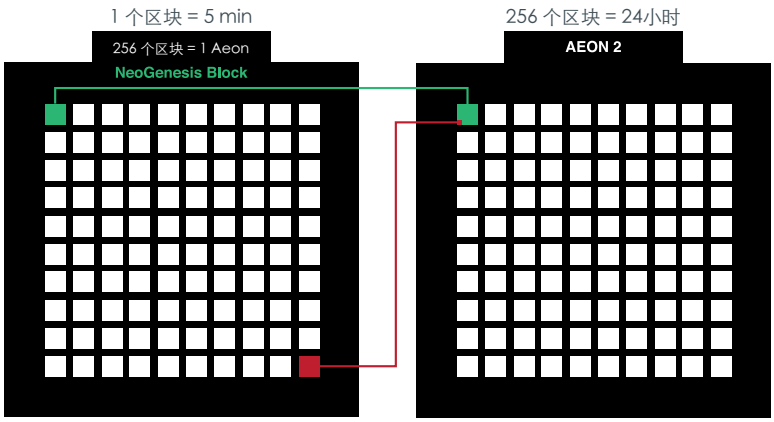
服务器这时进入“在线”状态并像往常一样发送和接收交易，尝试解决区块等。整个过程从开始到结束 通常需要几分钟，具体取决于多长时间才能解决下一个区块。

5.6 共识算法

共识算法回答了一个问题：“节点如何知道它是在主链还是孤立链(Orphaned Chain)上？”Mochimo根据广播节点所在区块链执行的工作量，通过验证“BLOCK FOUND”消息来回答这个问题。要得出这个值，我们必须首先介绍尾文件(Trailer File)和链权重(Chain Weight)的概念。

过程： 共识算法回答了一个问题：“节点如何知道它是在主链还是孤立链(Orphaned Chain)上？”Mochimo根据广播节点所在区块链执行的工作量，通过验证“BLOCK FOUND”消

区块 & Aeon 细节



5 主要技术概述

息来回答这个问题。要得出这个值，我们必须首先介绍尾文件(Trailer File)和链权重(Chain Weight)的概念。过程：Mochimo尾文件(Trailer File)是自自创世区块以来每个区块的尾部信息的链表(Linked list)，不论系统处于哪个Aeon还是已经有了数千或数百万个区块。每个尾文件都是100字节的 数据，因此即使Mochimo运行了数千年，这样大小的数据也不会大到无法储存。尾文件是不可变的，它包含：每个区块的开始和解决时间，区块哈希(Block Hash)，解决它的Nonce/Hash(哈希)以及解决时所需的难度。尾文件的链表用于计算声称已找到区块的链的工作量之和。该值被称为链的“权重”。

ChainCrunch™的底层机制让我们可以丢弃区块数据的同时，保留从第一个区块到当前区块解决记录的链表。尾文件是ChainCrunch™这个底层机制的一部分。

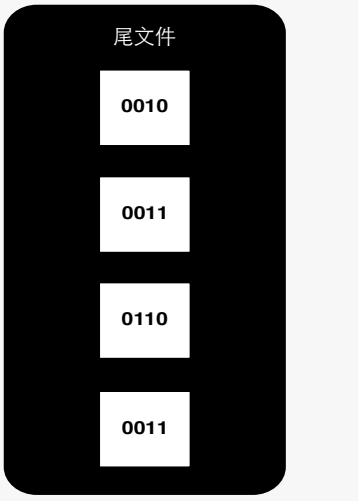
为了确定所执行的工作量，我们只需将每个块的难度值以二进制形式加在一起。例如，以34的难度解决的区块需要有34个前导零(Leading Zero)的哈希(Hash)输出值。计算权重时，我们将2^34加到解决该区块的链的总权重上。如果下一个区块以35的难度被解决，则该区块的权重是之前的两倍。

随着网络规模和矿工数量的增加，难度也相应增加，使得每个解决的区块随着时间的推移比以前的区块的权重值更高。 当我们验证T-File时，我们验证每个Hash(哈希)和nonce，开始和解决时间，重新创建难度(从区块0计算），然后附加到候选链的权重上，这很重要。

使用链权重，而不是以区块数量衡量的区块链长度，使我们能够快速且确定地证明一条链的总工作量多于另一条链。它还可以防止恶意行为者试图将假链或错误区块/账本

什么是区块权重？ 它的作用是什么？
所含区块总权重决定哪条链是主链。

我们计算每个区块上的工作量(值为二进制)



并相加得到总权重值

5 主要技术概述

加入网络，因为为了伪造一个区块，他们必须执行网络所执行的所有工作量的总和。当节点(Peers)发出“BLOCK FOUND”声明时，附加到该声明是四个项目：区块的hash(哈希值)，前一个区块的hash，nonce和难度。

在第一次传递时，接收的节点检查以确定BLOCK FOUND消息指示的先前区块哈希值是否与其当前区块匹配。如果它找到该消息，则这个节点对所广播的难度进行合理性检查，确认其与预期的难度相匹配。如果该检查失败，广播的节点将被列入黑名单。确认节点都在正确的链上并且难度匹配时，本地节点向这个节点请求该区块。

在接收到区块时，节点调用区块验证器，区块验证器验证区块中包含的每个交易，构造候选块， 并通过挖矿算法传递它和提供的nonce。如果区块有效，则本地节点暂停挖矿，刷新其交易队列， 并调用区块更新程序。区块更新器执行所有相关的全局变量增量，更新本地帐本数据库， 如果我们处于Aeon边界，则调用新生区块例程。当区块更新程序完成执行后，系统状态将返回到“已同步”，也就是我们已与网络“达成共识”。

5 主要技术概述

5.7 争用算法

仅当网络中的两个或更多节点在大致相同的时间解决区块时才会发生争用(Contention)。由于采用随机网络模型、以及我们将平均解决时间设置为337.5秒，争用在Mochimo网络中将相对发生较少，但是当这种情况出现时，共识算法将按照以下列方式非常快速地汇聚网络。

过程： 收到“BLOCK FOUND(中文:发现区块)”消息后，如共识算法部分代码所示，节点进行更新。当网络进入争用时，意味着在同一最大长度的网络上存在多个链，我们可以预期该节点将接收 另一个指示“BLOCK FOUND”的消息，但发现的是该节点已经在处理的区块。此消息还将具有 不同的哈希值，这表示节点存在争用。该节点将忽略这个BLOCK FOUND消息，因为所显示的区块序号不高于节点的当前区块，虽然权重是相同的。

在这种情况下，我们可以说网络上存在两个链，每个链具有相同的权重。由于区块的传播可以在 0.5秒内跨越网络，因此我们可以预期将在N*(337.5/0.5)个区块后看到网络上的第二个链，其中N是网络上的节点数。然而，在大多数情况下，第二链的传播程度将受到严重限制。

解决这个问题的方法非常简单：当下一个区块被挖出时，无论同时已经挖出了多少个链， 所有节点都将收到一个具有更高权重的更高区块序号的BLOCK FOUND消息。通过该消息， 广播的节点不在同一链上的事实将显而易见，因为每个广播的BLOCK FOUND也显示先前区块的哈希值。由于前一个区块的哈希值与我们的哈希值不匹配，我们知道与我们不同的链在声称是自己主链。

此时，为了解决争用，本地节点从广播的节点获取最近的哈希表。节点通过提供从新生区块 到当前区块的每个区块的哈希值来响应。本地节点将此列表与其自己的哈希列表进行比较，并向后搜索以查找匹配项。如果找到匹配项，则该节点位于孤立链上。执行一些额外的验证检查以确保块没有被幌骗(Spoofing)，并且在验证时，接收节点刷新所有状态并进行重启。

注，Mochimo网络的超快速汇聚能力让我们可以通过这样解决争用：允许发现自己在孤立链上的节点离开网络并重新同步。此外，链不仅可以分裂一次还可以分裂第二次。然而，发生这种情况的 可能性是1比(N*(337.5 / .5)^2)区块。

理解这一点很重要：网络上活动链的数量永远不会超过2个，并且在发生争用后的一个BLOCK FOUND消息不会增加网络上链的数量，而是增加刷新节点状态并进行重启的节点数量。

6 需要注意的货币数据

以下是一些关键的Mochimo数据:

最大供应量: 76,533,882
可挖掘的币量: 71,776,816 (93.8%)
挖矿算法: Trigg’s 算法– PoW
难度调整: 每个块
目标出块时间: 337.5 秒
创世区块(即第一个区块): 2018年6月25日
网络交易手续费(Network TX Fee): 0.0000005 MCM (固定)
起始奖励: 5.0 MCM / 块
每次区块奖励增加值(直到区块区块 373,760 4年): .00015 MCM
最大奖励 (区块 373,760): 59.17 MCM
每次区块奖励减少值（直到区块 2,097,152 22 年): .000028488 MCM
最终区块奖励 (区块 2,097,152): 5 MCM
总挖矿时间: ~22 年

预挖细节:
合计预挖比例: 6.34% (476万 MCM)
用于开发团队工作报酬的预挖比例: 4.18% (320万 MCM)
其他预挖币(由Mochimo基金会管理): 2.16% (156万 MCM)
第一个区块的生成日期: 2018年6月25日 23:40 UTC

7 关键引用文献

1. Bernstein, et al., https://eprint.iacr.org/2017/314.pdf
2. PQCRYPTO, https://pqcrypto.eu.org/docs/initial-recommendations.pdf
3. Hülsing, https://eprint.iacr.org/2017/965.pdf
4. https://blockchain.info/charts/blocks-size?timespan=all
5. Barabasi, Albert-Laszlo, http://barabasi.com/f/624.pdf
6. BITNODES, https://bitnodes.earn.com/