(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 114982196 A
(43)申请公布日 2022.08.30

(21)申请号 202080076769.5
(22)申请日 2020.10.05
(30)优先权数据
1915841.9 2019.10.31 GB
(85)PCT国际申请进入国家阶段日
2022.04.29
(86)PCT国际申请的申请数据
PCT/IB2020/059319 2020.10.05
(87)PCT国际申请的公布数据
WO2021/084347 EN 2021.05.06
(71)申请人 区块链许可股份公司
地址 瑞士楚格
(72)发明人 亚历山大·麦凯 克洛伊·塔尔坦 杰德·瓦哈伯 安托阿内塔·尔盖耶娃 克雷格·史蒂芬·怀特
(74)专利代理机构 北京市竞天公诚律师事务所 11770
专利代理师 徐民
(51)Int.Cl.
H04L 9/32(2006.01)

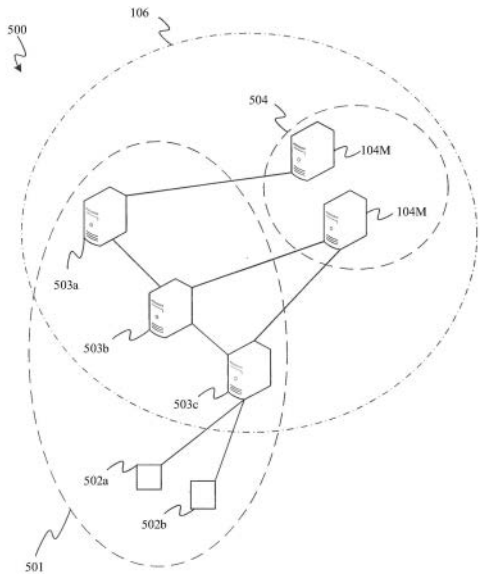
权利要求书3页 说明书31页 附图17页

(54)发明名称

利用区块链事务的通信协议

(57)摘要

一种用于向请求者授予加入第一网络的权限的计算机实现的方法。所述第一网络包括一组桥接节点以及可由所述一组桥接节点中的一个或更多个控制的一组设备。每个桥接节点也是区块链网络中的相应节点。所述方法由注册机构执行，包括生成第一区块链事务。所述第一区块链事务包括输入，所述输入包括链接至所述注册机构的第一公钥的签名。所述第一区块链事务还包括第一输出，所述第一输出包括第一证书，所述第一证书包括分配给所述请求者的标识符。所述方法还包括：将所述第一区块链事务传输至所述区块链网络，以将其包含在所述区块链中。



CN 114982196 A

1.一种用于向请求者授予加入第一网络的权限的计算机实现的方法,其中所述第一网络包括一组桥接节点和一组设备,所述一组设备可由所述一组桥接节点中的一个或更多个桥接节点控制,并且其中每个桥接节点也是区块链网络中的相应节点;所述方法由注册机构执行,包括:

生成第一区块链事务,其中所述第一区块链事务包括输入和第一输出,所述输入包括链接至所述注册机构的第一公钥的签名,所述第一输出包括第一证书,其中所述第一证书包括分配给所述请求者的标识符;以及

将所述第一区块链事务传输至所述区块链网络,以将其包含在所述区块链中。

2.根据权利要求1所述的方法,其中所述第一事务包括第二输出,所述第二输出锁定至所述注册机构的第二公钥。

3.根据权利要求2所述的方法,其中所述第一输出通过时间锁锁定至所述注册机构的所述第二公钥,其中,时间锁防止所述第一输出在预定时间段结束前解锁。

4.根据权利要求2或3所述的方法,其中所述第一输出至少锁定至所述注册机构的所述第二公钥和一不同的公钥。

5.根据权利要求1至4中任一项所述的方法,所述方法包括:将所述第一区块链事务的事务标识符传输给所述权限请求者。

6.根据权利要求1至5中任一项所述的方法,其中所述证书使用加密密钥进行加密,所述加密密钥由所述注册机构生成。

7.根据权利要求1至6中任一项所述的方法,所述方法包括:

接收所述请求者提出的加入所述网络的请求,其中所述请求包括一个或更多个凭证;

根据所述一个或更多个凭证核实所述请求,其中所述生成所述第一区块链事务是以所述请求有效为条件的。

8.根据权利要求1至7中任一项所述的方法,其中所述一组桥接节点包括主节点以及可由所述主节点控制的一组中间节点,其中所述注册机构是所述主节点。

9.根据权利要求1至8中任一项所述的方法,其中所述一组桥接节点包括所述主节点以及可由所述主节点控制的所述一组中间节点,其中所述请求者是所述主节点。

10.根据权利要求1至9中任一项所述的方法,其中所述请求者是所述区块链网络中的相应节点,其中所述证书包括分配给所述权限请求者的公钥。

11.根据权利要求1至8中任一项所述的方法,其中所述请求者是可由所述第一网络中的一个或更多个桥接节点控制的设备,并且其中所述方法包括:

向所述请求者传输一组证书,所述一组证书中的每个证书已经传输至所述一组节点中的相应节点。

12.根据权利要求1至11中任一项所述的方法,所述方法包括:将所述第一证书传输至所述一组桥接节点中的一个或更多个。

13.根据权利要求2或其任何从属权利要求所述的方法,所述方法包括:

生成第二区块链事务,其中所述第二区块链事务包括引用所述第一事务的所述第二输出的输入,还包括链接至所述注册机构的所述第二公钥的签名;

将所述第二区块链事务传输至所述区块链网络,以将其包含在所述区块链中。

14.一种用于请求加入第一网络的权限的计算机实现的方法,其中所述第一网络包括

一组桥接节点和一组设备，所述一组设备可由所述一组桥接节点中的一个或更多个桥接节点控制，并且其中每个桥接节点也是区块链网络中的相应节点；所述方法由请求者执行，包括：

向注册机构发送加入所述第一网络的请求；

获取第一证书，所述证书由所述注册机构颁发并且包括分配给所述请求者的标识符。

15.根据权利要求14所述的方法，其中所述获取包括：

接收包括所述第一证书的第一区块链事务的事务标识符；

使用所述事务标识符从所述区块链中获取所述第一区块链事务。

16.根据权利要求15所述的方法，其中所述第一区块链事务包括第一输入和第二输出，所述第一输入包括所述证书，所述第二输出链接至所述注册机构的公钥，其中所述方法包括：

标识所述注册机构的所述公钥；

从所述注册机构的所述公钥中标识包含在传输至所述区块链的一个或更多个相应事务中的一个或更多个第二证书，每个第二证书被颁发给相应的桥接节点或设备或所述网络。

17.根据权利要求16所述的方法，其中所述第一证书包括所述请求者的公钥，其中颁发给所述第一网络中的所述一组桥接节点中的相应桥接节点的每个第二证书包括所述节点的相应公钥，其中所述方法包括：

将第三区块链事务传输至所述一组桥接节点中的至少一个，其中所述第三区块链事务包括锁定至所述至少一个桥接节点的所述相应公钥的输出。

18.根据权利要求14至17中任一项所述的方法，所述获取第一证书包括：从所述注册机构接收所述第一证书。

19.根据权利要求14至18中任一项所述的方法，所述方法包括：

从所述注册机构接收一个或更多个第二证书，每个第二证书被颁发给所述第一网络中的所述一组桥接节点或设备中的相应桥接节点或设备。

20.根据权利要求19所述的方法，其中所述第一证书包括所述请求者的网络地址，其中颁发给所述第一网络中的所述相应桥接节点的每个第二证书包括所述节点的相应网络地址，其中所述方法包括：

向所述一组桥接节点中的一个或更多个发送消息，所述消息从所述请求者的所述网络地址发送至所述消息所发送到的所述一个或更多个桥接节点的相应网络地址。

21.根据权利要求14或权利要求18至20中任一项所述的方法，其中所述请求者是所述第一网络中的所述一组设备中的一个。

22.根据权利要求14至21中任一项所述的方法，其中所述请求者是所述第一网络中的所述一组节点中的一个。

23.根据权利要求22所述的方法，其中所述一组桥接节点包括主节点以及可由所述主节点控制的一个或更多个中间节点，其中所述请求者是所述主节点。

24.根据权利要求14至23中任一项所述的方法，其中所述请求包括所述请求者的一个或更多个凭证。

25.根据权利要求24所述的方法，其中所述一个或更多个凭证包括所述请求者的IP地

址。

26.一种计算机设备,所述计算机设备包括:

存储器,所述存储器包括一个或更多个存储器单元;

处理装置,所述处理装置包括一个或更多个处理单元,其中所述存储器存储被设置在所述处理装置上运行的代码,所述代码被配置为当在所述处理装置上运行时,执行根据权利要求1至13中任一项所述的方法。

27.一种在计算机可读存储器上实现的计算机程序,所述计算机程序被配置为当在根据权利要求26所述的计算机设备上运行时,执行根据权利要求1至13中任一项所述的方法。

28.一种计算机设备,所述计算机设备包括:

存储器,所述存储器包括一个或更多个存储器单元;

处理装置,所述处理装置包括一个或更多个处理单元,其中所述存储器存储被设置在所述处理装置上运行的代码,所述代码被配置为当在所述处理装置上运行时,执行根据权利要求14至25中任一项所述的方法。

29.一种在计算机可读存储器上实现的计算机程序,所述计算机程序被配置为当在计算机设备上运行时,执行根据权利要求14至25中任一项所述的方法。

利用区块链事务的通信协议

技术领域

[0001] 本公开涉及用于向请求实体授予加入网络的权限的方法，例如，以便所述请求实体访问所述网络。

背景技术

[0002] 区块链是指一种分布式数据结构形式，其中在点对点(P2P)网络中的多个节点中的每个节点处维护区块链副本。区块链包括一系列数据区块，其中每个区块包括一个或更多个事务(transaction)。每个事务都可以回指序列中的先前事务，其可以扩展一个或更多区块。通过称为“挖掘”的过程，事务可以通过提交到网络包括在新区块中，该过程涉及多个挖掘节点中的每个挖掘节点争相执行“工作量证明”，即基于等待包括在区块中的未决事务池解决加密难题。

[0003] 区块链中的事务通常用于传递数字资产，即用作价值储存手段的数据。但是也可利用区块链实现区块链上的分层附加功能。例如，区块链协议可允许在事务输出中存储附加用户数据。现代区块链在单一事务中可储存的最大数据容量在不断增加，从而能够并入更复杂的数据。例如，这可用于在区块链中存储电子文档，甚至音频或视频数据。

[0004] 网络中的每个节点可以具有以下三个角色中的任何一个、两个或全部：转发、挖掘和存储。转发节点在整个网络节点中传播事务。挖掘节点将事务挖掘到区块中。存储节点各自对区块链中的已挖掘区块存储自己的副本。为了将事务记录在区块链中，一方将该事务发送到网络中的节点中的一个节点进行传播。接收该事务的挖掘节点可以争相将该事务挖掘到新区块中。每个节点被配置为遵守相同的节点协议，该协议将包括用于确认事务有效的一个或更多个条件。无效事务将不会传播或挖掘到区块中。假定事务已经核实有效，从而在区块链上被接受，该附加用户数据将因此作为不可改变的公共记录，继续存储在P2P网络中的各个节点处。

发明内容

[0005] 物联网(IoT)技术使物理设备网络能够在无需人工干预的情况下监控事件和交换数据。推动物联网技术发展的原因是需要实时数据采集机制和自动控制机制，以取代各行业的常规监测和控制方法。物联网系统产生大量数据，并且依赖于具有网络可扩展性、强大网络安全性、可靠连接性和最小网络延迟的系统。

[0006] 目前，集中式架构模型广泛应用于认证、授权和连接物联网中的节点。此类模型易受攻击，并且充当单点故障。如果集中式系统受到破坏，可能会向恶意设备授予和/或从现有设备中删除访问所述物联网的权限。例如，如果恶意设备被授予访问所述物联网的权限，该设备可以获取敏感数据或破坏所述网络。

[0007] 与集中式架构相比，点对点(P2P)架构提供了更安全且更高效的解决方案，相邻节点之间无需使用任何集中式节点或智能体即可直接进行交互。区块链技术是安全P2P通信的基础，有望彻底改变物联网系统的发展。然而，如果实现下一代基于区块链的物联网设备

系统，基于区块链的物联网控制方法需要克服开放式系统固有的挑战。这些包括可能不是区块链本身固有的数据隐私和设备保护/控制机制。

[0008] 根据本文公开的一个方面，提供了一种用于向请求者授予加入第一网络的权限的计算机实现的方法，其中所述第一网络包括一组桥接节点以及可由所述一组桥接节点中的一个或更多个控制的一组设备，其中每个桥接节点也是区块链网络中的相应节点；所述方法由注册机构执行，包括：生成第一区块链事务，其中所述第一区块链事务包括输入和第一输出，所述输入包括链接至所述注册机构的第一公钥的签名，所述第一输出包括第一证书，其中所述第一证书包括分配给所述请求者的标识符；将所述第一区块链事务传输至所述区块链网络，以将其包含在所述区块链中。

[0009] 所述第一网络(例如，物联网)包括一个或更多个桥接节点以及可由所述一个或更多个桥接节点控制的一个或更多个设备。所述桥接节点也是区块链网络中的节点。也就是说，所述桥接节点是所述物联网和所述区块链网络的一部分，即，所述桥接节点既可以连接至所述物联网(例如，与其他网络节点和设备通信)，也可以连接至所述区块链网络(例如，将事务传输至所述区块链，并标识和读取所述区块链上记录的事务)。这些节点充当所述第一网络与所述区块链网络之间的网关或网桥。此外，这些节点无需具有所述区块链网络中的挖矿节点、转发节点或存储节点的作用，但也不排除这种情况。在一些示例中，所述第一网络的所述设备中的一个或更多个也可以是所述区块链网络中的节点。

[0010] 所述注册机构(可以是所述物联网的桥接节点，也可以不是)是所述区块链网络中的节点。即，所述注册机构连接至所述区块链，用于将事务传输至所述区块链网络。所述注册机构负责向请求者(请求实体)授予证书，然后，这些证书授予请求者加入所述网络的权限。在本文中，加入所述网络后，请求者便可执行任务，例如，与连接至所述第一网络中的其他节点和/或设备通信。所述注册机构是受信任方，可以验证新设备的凭证并颁发数字证书。所述注册机构充当准入门槛，并且允许或禁止新设备加入所述物联网。

[0011] 在所述事务的输出中，所述第一区块链事务包括颁发给所述请求者的证书。所述证书包括所述请求者的唯一标识符(“设备ID”)。所述设备ID可以是伪随机字节串。所述注册机构将所述事务(以及所述证书)广播到所述区块链网络。所述第一区块链事务通过所述第一输入中的所述签名链接至所述注册机构。所述签名链接至所述注册机构的公钥。换句话说，所述签名是根据与所述注册机构的公钥对应的私钥生成的。只有所述注册机构知道所述私钥，因此只有所述注册机构可以用其签名对所述事务进行签名。换句话说，所述证书无法伪造。因此，证书只能授予真正的实体。当所述请求者在获取证书后与所述第一网络中的其他节点或设备通信时，这些节点和设备可以检查是否已向所述请求者颁发证书，从而检查所述请求者是否是真正的实体。

[0012] 在一些实施例中，所述请求者也是所述区块链网络中的节点。在这种情况下，所述证书可以包括分配给所述请求节点的公钥。换句话说，所述公钥是认证公钥。如果所述请求节点使用其认证公钥或由此派生的密钥与其他节点进行通信(例如，向节点或区块链发送使用所述公钥中的一个签名的事务)，则所述其他节点可以确信发送这些事务的节点是真正的节点。

[0013] 根据本文公开的另一个方面，提供了一种用于请求加入第一网络的权限的计算机实现的方法，其中所述第一网络包括一组桥接节点以及可由所述一组桥接节点中的一个或

更多个控制的一组设备，其中每个桥接节点也是区块链网络中的相应节点；所述方法由请求者执行，包括：向注册机构发送加入所述第一网络的请求；获取第一证书，所述证书由所述注册机构颁发并且包括分配给所述请求者的标识符。

附图说明

[0014] 为了帮助理解本公开的实施例并示出如何实施此类实施例，现将仅通过举例的方式参考附图进行说明，其中：

[0015] 图1是一种用于实现区块链的系统的示意性框图；

[0016] 图2示意性地示出了可记录在区块链中的事务的一些示例；

[0017] 图3是另一种用于实现区块链的系统的示意性框图；

[0018] 图4示出了一种用于根据基于输出的模型的节点协议来处理事务的节点软件的示意性框图；

[0019] 图5示意性地示出了物联网与区块链网络之间的重叠；

[0020] 图6示意性地示出了分层网络拓扑结构；

[0021] 图7a和图7b示意性地示出了部分命令事务和完整命令事务；

[0022] 图8a和图8b示意性地示出了替代的部分事务和完整事务；

[0023] 图9示意性地示出了命令请求和响应周期；

[0024] 图10a和图10b示意性地示出了从服务器节点传输至从节点的部分命令事务和完整命令事务；

[0025] 图11a和图11b示意性地示出了命令请求和批准事务；

[0026] 图12a和图12b示意性地示出了加密的部分命令事务和完整命令事务；

[0027] 图13示出了示例性命令数据格式；

[0028] 图14示意性地示出了示例性点对点打印系统；

[0029] 图15a至图15c示意性地示出了点对点打印系统中使用的示例性事务；

[0030] 图16a和图16b示意性地示出了证书事务和示例性证书格式。

具体实施方式

[0031] 示例性系统概述

[0032] 图1总体地示出了一种用于实现区块链150的示例性系统100。系统100包括分组交换网络101，通常是诸如互联网的广域互联网。分组交换网络101包括多个节点104，该多个节点被设置成在分组交换网络101内形成点对点(P2P)覆盖网络106。每个节点104包括对等体的计算机设备，不同的节点104属于不同的对等体。每个节点104包括含一个或更多个处理器的处理装置，例如一个或更多个中央处理单元(CPU)、加速器处理器、特定应用程序处理器和/或现场可编程门阵列(FPGA)。每个节点还包括存储器，即采用非暂时性计算机可读介质形式的计算机可读存储器。存储器可包括一个或更多个存储器单元，其采用一个或更多个存储器介质，例如诸如硬盘等的磁介质、诸如固态硬盘(SSD)、闪存或电可擦可编程只读存储器(EEPROM)等的电子媒介和/或诸如光盘驱动器等的光学介质。

[0033] 区块链150包括一系列数据区块151，其中在P2P网络160中的多个节点中的每个节点处维护相应的区块链150副本。区块链中的每个区块151均包括一个或多个事务

(transaction)152,其中该上下文中的事务是指一种数据结构。数据结构的性质将取决于用作事务模型或计划的一部分的事务协议类型。给定的区块链通常全程使用一个特定的事务协议。在一种常见的事务协议中,每个事务152的数据结构至少包括一个输入和至少一个输出。每个输出指定一个数额,该数额表示属于输出被加密锁定的用户103的数字资产值(需要该用户的签名进行解锁,从而进行赎回或花费)。每个输入指向先前事务152的输出,从而链接这些事务。

[0034] 节点104中的至少一些节点扮演转发节点104F的角色,这些节点转发并因此传播事务152。节点104中的至少一些节点扮演挖掘区块151的矿工104M的角色。节点104中的至少一些节点扮演存储节点104S(有时也称为"完整副本"节点)的角色,每个存储节点均在相应的存储器中存储相同区块链150的相应副本。每个矿工节点104M还维护等待挖掘到区块151中的事务152的池154。给定节点104可以是转发节点104、矿工104M、存储节点104S或其中两个节点或所有节点的任意组合。

[0035] 在给定的当前事务152j中,输入(或每个输入)包括指针,该指针引用事务序列中先前事务152i的输出,指定该输出将在当前事务152j中被赎回或"花费"。通常,当前事务可以是池154或任何区块151中的任何事务。尽管为了确保当前事务有效,将需要存在先前事务152i并核实其有效,但是在创建当前事务152j甚至向网络106发送当前事务152j时,不必存在先前事务152i。因此,在本文中,"先前"是指由指针链接的逻辑序列中的前任,而不一定是时间序列中的创建时间或发送时间,因此,不一定排除无序创建或发送事务152i、152j的情况(参见下面关于孤立事务的讨论)。先前事务152i同样可以称为先行事务或前任事务。

[0036] 当前事务152j的输入还包括先前事务152i的输出被锁定到的用户103a的签名。反过来,当前事务152j的输出可以加密锁定到新用户103b。因此,当前事务152j可将先前事务152i的输入中定义的数额转移到当前事务152j的输出中定义的新用户103b。在某些情况下,事务152可具有多个输出,以在多个用户间分割输入数额(其中一个可以是原始用户103a,以便进行变更)。在某些情况下,一个事务还可以具有多个输入,以将一个或更多个先前事务的多个输出中的数额汇总在一起,并重新分配到当前事务的一个或更多个输出。

[0037] 上述可称为"基于输出的"事务协议,有时也称为未花费的事务输出(UTXO)的协议(其中输出称为UTXO)。用户的总余额不是用区块链中存储的任何一个数字定义的;相反,用户需要特殊"钱包"应用程序105,以整理该用户的所有UTXO值,这些UTXO值分散在区块链151的许多不同事务152中。

[0038] 作为基于账户的事务模型的一部分,另一种类型的事务协议可称为"基于账户的"协议。在基于账户的情况下,每个事务均不通过参考过去事务序列中先前事务的UTXO来定义转移的数额,而是通过参考绝对账户余额进行定义。所有账户的当前状态由矿工单独存储到区块链中,并不断更新。在此类系统中,事务使用账户的运行事务记录(也称为"头寸")进行排序。该值由发送者签名作为其加密签名的一部分,并作为事务引用计算的一部分进行哈希处理。此外,可选的数据字段也可以在事务中签名。例如,如果数据字段中包含先前事务的ID,则该数据字段可指向先前事务。

[0039] 无论采用何种类型的事务协议,当用户103希望执行新事务152j时,其希望将新事务从其计算机终端102发送至P2P网络106的节点104中的一个(现在通常是服务器或数据中

心,但原则上可以是其他用户终端)。此节点104根据在节点104中的每个节点处应用的节点协议检查事务是否有效。节点协议的详细信息将与相关区块链150中使用的事务协议类型相对应,一起形成整个事务模型。节点协议通常要求节点104检查新事务152j中的加密签名是否与预期签名相匹配,这取决于事务152的有序序列中的先前事务152i。在基于输出的情况下,这可包括检查新事务152j的输入中包含的用户加密签名是否与新事务花费的先前事务152i的输出中定义的条件相匹配,其中该条件通常包括至少检查新事务152j的输入中的加密签名是否解锁新事务的输入所指向的先前事务152i的输出。在一些事务协议中,条件可至少部分地由输入和/或输出中包含的自定义脚本定义。或者,这可仅由节点协议单独确定,或可通过其组合确定。无论采用哪种方式,如果新事务152j有效,当前节点会将其转发到P2P网络106中的一个或多个其他节点104。这些节点104中的至少一些节点还作为转发节点104F,根据相同的节点协议应用相同的测试,从而将新事务152j转发到一个或更多个进一步的节点104,依此类推。通过这种方式,新事务在节点104的整个网络中进行传播。

[0040] 在基于输出的模型中,给定输出(例如,UTXO)是否花费的定义是,根据节点协议,其是否通过另一个随后事务152j的输入有效赎回。事务有效的另一个条件是其试图花费或赎回的先前事务152i的输出尚未被另一个有效事务花费/赎回。同样,如果无效,事务152j将不会在区块链中传播或记录。这可防止重复花费,即花费者对同一个事务的输出花费超过一次。另一方面,基于账户的模型通过保持账户余额防止重复花费。因为同样存在定义的事务顺序,账户余额在任何时候均具有单一定义的状态。

[0041] 除核实之外,节点104M中的至少一些节点在称为挖矿的过程中争先创建事务区块,该过程以“工作量证明”为基础。在挖矿节点104M处,将新事务添加到区块中尚未出现的有效事务的池中。然后,矿工争相通过尝试解决加密难题来组装事务池154中事务152的新的有效区块151。通常情况下,这包括搜索“随机数”值,从而当随机数与事务池154并置且进行哈希处理时,哈希值的输出满足预定条件。例如,预定条件可以是哈希值的输出具有某个预定义的前导零数。哈希函数的特性是,相对于其输入,其具有不可预测的输出。因此,该搜索只能通过强力执行,从而在试图解决难题的每个节点104M处消耗大量的处理资源。

[0042] 解决难题的第一矿工节点104M在网络106上宣布难题解决,提供解决方案作为证明,然后网络中的其他节点104则可以轻松检查该解决方案(一旦给出哈希值的解决方案,就可以直接检查该解决方案是否使哈希值的输出满足条件)。基于已在每个此类节点处检查获胜者的已宣布解决方案,获胜者已为其解决该难题的事务池154之后由充当存储节点104S的节点104中的至少一些节点记录在区块链150中作为新区块151。区块指针155还分配给指向区块链中先前创建的区块151n-1的新区块151n。工作量证明有助于降低重复花费的风险,因为创建新区块151需要大量工作,并且由于包含重复花费的任何区块都可能被其他节点104拒绝,因此挖矿节点104M受到激励,不允许在其区块中包含双重花费。一旦创建,则不可修改区块151,因为其根据相同的协议在P2P网络106中的存储节点104S中的每个存储节点进行识别和维护。区块指针155还向区块151施加顺序。由于事务152记录在P2P网络106中每个存储节点104S处的有序区块中,因此提供了事务的不可变公共分类账。

[0043] 应当注意的是,在任何给定时间争相解决难题的不同矿工104M可能会根据任何给定时间的未挖掘事务池154的不同快照执行该操作,具体取决于他们何时开始搜索解决方案。解决相应难题的人员首先定义新区块151n中包含的事务152,并更新当前未挖掘事务池

154。然后，矿工104M继续争相从新定义的未完成池154中创建区块，依此类推。此外，还存在解决可能出现的任何“分叉”的协议，其中两名矿工104M彼此在很短的时间内解决难题，从而传播区块链的冲突视图。简言之，分叉方向最长的成为最终区块链150。

[0044] 在大部分区块链中，获胜矿工104M会自动获得特殊类型的新事务作为奖励，该新事务创建新的数字资产值(与将数字资产数额从一个用户转移至另一个用户的正常事务截然相反)。因此，获胜节点被视为已“挖掘”一定数量的数字资产。这种特殊类型的事务有时称为“生成”事务，其自动形成新区块151n的一部分。该奖励可激励矿工104M争相参与工作量证明。通常情况下，常规(非生成)事务152还将在其输出中的一个输出中指定附加事务费用，以进一步奖励创建其中包含事务的区块151n的获胜矿工104M。

[0045] 由于挖掘中涉及的计算资源，通常至少矿工节点104M中的每个矿工节点采用服务器的形式，该服务器包括一个或更多个物理服务器单元，甚至整个数据中心。每个转发节点104M和/或存储节点104S还可采取服务器或数据中心的形式。但是，原则上来说，任何给定节点104均可采用一个用户终端或联网在一起的一组用户终端的形式。

[0046] 每个节点104的存储器均存储被配置为在节点104的处理装置上运行的软件，以根据节点协议执行其相应的角色并处理事务152。应当理解的是，在本文中归因于节点104的任何动作均可通过在相应计算机设备的处理装置上运行的软件执行。此外，在本文中使用的“区块链”一词是指一般技术类型的通用术语，不限于任何特定专有区块链、协议或服务。

[0047] 扮演消费用户角色的多方103中的每一方的计算机设备102也连接到网络101。他们充当事务中的支付人和收受人，但不一定代表其他方参与挖掘或传播事务。他们不一定运行挖矿协议。出于说明目的，示出了双方103及其相应的设备102：第一方103a及其相应的计算机设备102a，以及第二方103b及其相应的计算机设备102b。应当理解的是，更多此类当事方103及其相应的计算机设备102可能存在并参与系统，但为了方便起见，未进行说明。每一方103均可以是个人或组织。仅出于说明目的，在本文中，第一方103a称为爱丽丝，第二方103b称为鲍勃，但应当理解的是，这并不仅限于爱丽丝或鲍勃，且本文对爱丽丝或鲍勃的任何引用均可分别用“第一方”和“第二方”替换。

[0048] 每一方103的计算机设备102包括相应的处理装置，其包括一个或更多个处理器，例如一个或更多个CPU、图形处理单元(GPU)、其他加速器处理器、特定应用程序处理器和/或FPGA。每一方103的计算机设备102还包括存储器，即采用非暂时性计算机可读介质形式的计算机可读存储器。该存储器可包括一个或更多个存储器单元，其采用一个或更多个存储器介质，例如诸如硬盘等磁介质、诸如SSD、闪存或EEPROM等电子媒介和/或诸如光盘驱动器等的光学介质。每一方103的计算机设备102上的存储器存储软件，其包括被设置为在处理装置上运行的至少一个客户端应用程序105的相应实例。应当理解的是，在本文中归因于给定方103的任何行动均可通过在相应计算机设备102的处理装置上运行的软件执行。每一方103的计算机设备102包括至少一个用户终端，例如台式或笔记本电脑、平板电脑、智能手机或诸如智能手表等的可穿戴设备。给定方103的计算机设备102还可包括一个或更多个其他网络资源，诸如通过用户终端访问的云计算资源。

[0049] 客户端应用程序或软件105最初可通过例如从服务器下载的适当计算机可读存储介质，或通过诸如可移动SSD、闪存密钥、可移动EEPROM、可移动磁盘驱动器、软盘或磁带等的可移动存储设备、诸如CD或DVD ROM等的光盘或可移动光驱等提供至任何给定方103的计

算机设备102。

[0050] 客户端应用程序105至少包括“钱包”功能。这有两个主要功能。其中一个功能是使相应的用户方103创建、签名和发送拟在节点104的整个网络中传播的事务152，并因此包含在区块链150中。另一个功能是向相应方汇报其目前拥有的数字资产数额。在基于输出的系统中，该第二功能包括整理分散在区块链150中属于相关方的各种事务152的输出中定义的数额。

[0051] 每个计算机设备102上的客户端应用程序105的实例可操作地耦合到P2P网络106的转发节点104F中的至少一个转发节点。这可以启用客户端105的钱包功能，以将事务152发送至网络106。客户端105还可联系一个、一些或所有存储节点104，以在区块链150中查询相应方103作为接收者的任何事务(或实际上在区块链150中检查其他方的事务，因为在实施例中，区块链150是在某种程度上通过其公开可见性提供事务信任的公共设施)。每个计算机设备102上的钱包功能被配置为根据事务协议制定和发送事务152。每个节点104运行软件，其被配置为根据节点协议核实事务152有效的软件，并且在转发节点104F的情况下转发事务152，以在整个网络106中传播此类事务。事务协议和节点协议相互对应，给定事务协议和给定节点协议一起实现给定的事务模型。区块链150中的所有事务152均采用相同的事务协议(尽管事务协议可允许其内存在不同的事务子类型)。网络106中的所有节点104采用相同的节点协议(尽管其可根据针对该子类型定义的规则区分处理不同的事务子类型，并且不同的节点还可扮演不同的角色，从而实现协议的不同对应方面)。

[0052] 如上所述，区块链150包括一系列区块151，其中每个区块151包括通过如前所述的工作量证明过程创建的一个或更多个事务152的集合。每个区块151还包括区块指针155，其指向区块链中先前创建的区块151，以定义区块151的顺序。区块链150还包括有效事务池154，其等待通过工作量证明过程包含在新的区块中。每个事务152(除了一生成事务)包括指向先前事务的指针，以定义事务序列的顺序(注:事务152的序列可进行分支)。区块151的区块链一直追溯到创始区块(Gb)153，该创始区块是区块链中的第一区块。区块链150中早期的一个或更多个原始事务152指向创始区块153，而非先前事务。

[0053] 当给定方103(比方说爱丽丝)希望发送拟包含在区块链150中的新事务152j时，她将根据相关事务协议(使用其客户端应用程序105中的钱包功能)制定新事务。然后，她将事务152从客户端应用程序105发送至其连接的一个或更多个转发节点104F中的一个。例如，这可以是与爱丽丝的计算机102最近或最佳连接的转发节点104F。当任何给定节点104接收新事务152j时，其将根据节点协议及其相应的角色进行处理。这包括首先检查新接收的事务152j是否满足变为“有效”的特定条件，具体示例稍后将详细讨论。在一些事务协议中，有效条件可通过事务152中包含的脚本在每个事务的基础上进行配置。或者，条件可仅仅是节点协议的内置功能，或通过组合脚本和节点协议进行定义。

[0054] 如果新接收的事务152j通过有效性测试(即:“有效”的条件下)，接收事务152j的任何存储节点104S将向在该节点104S处维护的区块链150的副本中的池154中添加新有效事务152。进一步地，接收事务152j的任何转发节点104F随后将有效事务152传播至P2P网络106中的一个或更多个其他节点104。由于每个转发节点104F应用相同的协议，因此假定事务152j有效，这意味着事务很快将在整个P2P网络106中传播。

[0055] 一旦进入在一个或更多个存储节点104处维护的区块链150的副本中的池154中，

矿工节点104M将开始竞相解决包括新事务152的池154的最新版本方面的工作量证明难题(其他矿工104M可继续尝试基于池154的旧视角解决难题,但首先解决难题的矿工将定义下一个新区块151的结束位置和新池154的开始位置,最终将有人解决包括爱丽丝的事务152j的池154的一部分的难题)。一旦包括新事务152j的池154完成工作量证明,其将不可变地成为区块链150中区块151中的一个区块的一部分。每个事务152包括指向早前事务的指针,因此事务的顺序也被不可变地记录下来。

[0056] 基于UTXO的模型

[0057] 图2示出了示例性事务协议。这是基于UTXO的协议的示例。事务152(简称“Tx”)是区块链150的基本数据结构(每个区块151包括一个或更多个事务152)。下面将通过参考基于输出或基于“UTXO”的协议进行描述。但这并不限于所有可能的实施例。

[0058] 在基于UTXO的模型中,每个事务(“Tx”)152包括数据结构,其包括一个或更多个输入202和一个或更多个输出203。每个输出203可包括未花费的事务输出(UTXO),其可用作另一新事务的输入202的来源(如果UTXO尚未赎回)。UTXO指定数字资产数额(价值储存手段)。它还可包含其来源事务的事务ID以及其他信息。事务数据结构还可包括标头201,其可包括输入字段202和输出字段203的大小指示符。标头201还可包括事务的ID。在实施例中,事务ID是事务数据(不含事务ID本身)的哈希值,且存储在提交至矿工104M的原始事务152的标头201中。

[0059] 需要注意的是,虽然图2中的每个输出都示为UTXO,但事务可以附加地或替代地包括一个或更多个不可花费的事务输出。

[0060] 比方说爱丽丝103a希望创建转移相关数字资产数额至鲍勃103b的事务152j。在图2中,爱丽丝的新事务152j标记为“Tx_1”。该新事务获取在序列中先前事务152i的输出203中锁定至爱丽丝的数字资产数额,并至少将此类数额中的一部分转移至鲍勃。在图2中,先前事务152i标记为“Tx_0”。Tx_0和Tx_1只是任意的标记,其不一定意味着Tx_0指区块链151中的第一事务且Tx_1指池154中的下一个事务。Tx_1可指向仍具有锁定至爱丽丝的未花费输出203的任何先前(即先行)事务。

[0061] 当爱丽丝创建其新事务Tx_1时,或至少在她将该新事务发送至网络106时,先前事务Tx_0可能已经有效并包括在区块链150中。该事务此时可能已包括在区块151中的一个区块中,或者可能仍在池154中等待,在这种情况下,该事务将很快包括在新区块151中。或者,Tx_0和Tx_1可以创建并一起发送至网络102;或者,如果节点协议允许缓冲“孤立”事务,Tx_0甚至可以在Tx_1之后发送。本文事务序列上下文中使用的“先前”和“后续”一词是指由事务中指定的事务指针定义的序列中的事务顺序(哪个事务指向哪个其他事务等等)。它们同样可以替换为“前任”和“继任”、“先行”和“后代”或“父项”和“子项”等。这不一定指其创建、发送至网络106或到达任何给定节点104的顺序。然而,指向先前事务(先行事务或“父事务”)的后续事务(后代事务或“子事务”)不会有效除非父事务有效。在父事务之前到达节点104的子事务被视为孤立事务。根据节点协议和/或矿工行为,其可被丢弃或缓冲一段时间,以等待父事务。

[0062] 先前事务Tx_0的一个或更多个输出203中的一个包括特定的UTXO,标记为$UTXO_0$。每个UTXO包括指定UTXO表示的数字资产数额的值以及锁定脚本,该锁定脚本定义后续事务的输入202中的解锁脚本必须满足的条件,以使后续事务有效,从而成功赎回UTXO。通常情况

下,锁定脚本将数额锁定至特定方(该数额的事务的受益人)。即,锁定脚本定义解锁条件,该解锁条件通常包括以下条件:后续事务的输入中的解锁脚本包括先前事务被锁定到的一方的加密签名。

[0063] 锁定脚本(亦称scriptPubKey)是节点协议识别的域特定语言中写入的一段代码。此类语言的特定示例称为“脚本(Script)”(S大写)。锁定脚本指定花费事务输出203所需的信息,例如爱丽丝签名的要求。解锁脚本出现在事务的输出中。解锁脚本(亦称scriptSig)是提供满足锁定脚本标准所需信息的域特定语言中写入的一段代码。例如,其可包含鲍勃的签名。解锁脚本出现在事务的输入202中。

[0064] 因此在示出的示例中,Tx_0的输出203中的$UTXO_0$包括锁定脚本[Checksig P_A],该锁定脚本需要爱丽丝的签名Sig P_A,以赎回$UTXO_0$(严格来说,是为了使试图赎回$UTXO_0$的后续事务有效)。[Checksig P_A]包含爱丽丝的公私密钥对中的公钥P_A。Tx_1的输入202包括指向Tx_1的指针(例如,通过其事务ID($TxID_0$),其在实施例中是整个事务Tx_0的哈希值)。Tx_1的输入202包括在Tx_0中标识$UTXO_0$的索引,以在Tx_0的任何其他可能输出中对其进行标识。Tx_1的输入202进一步包括解锁脚本<Sig P_A>,该解锁脚本包括爱丽丝的加密签名,该签名由爱丽丝通过将其密钥对中的私钥应用于预定的部分数据(有时在密码学中称为“消息”)创建。爱丽丝需要签名以提供有效签名的数据(或“消息”)可通过锁定脚本、节点协议或其组合进行定义。

[0065] 当新事务Tx_1到达节点104时,该节点应用节点协议。这包括一起运行锁定脚本和解锁脚本,以检查解锁脚本是否满足锁定脚本中定义的条件(其中该条件可包括一个或更多个标准)。在实施例中,这涉及并置两个脚本:

[0066] <Sig P_A><P_A>||[Checksig P_A]

[0067] 其中“||”表示并置,“<…>”表示将数据放在堆栈上,“[…]”表示由解锁脚本组成的函数(在该示例中指基于堆栈的语言)。同样,脚本可以使用公共堆栈一个接一个地运行,而不是并置脚本。无论采用哪种方式,当一起运行时,脚本使用爱丽丝的公钥P_A(包括在Tx_0的输出的锁定脚本中),以认证Tx_1的输入中的锁定脚本是否包含爱丽丝签名预期部分的数据时的签名。预期的部分数据本身(“消息”)也需要包括在Tx_0中,以便执行此认证。在实施例中,签名的数据包括整个Tx_0(因此不需要包括一个单独的元素来明文指定签名的部分数据,因为其本身便已存在)。

[0068] 本领域技术人员将熟悉通过公私密码进行认证的细节。基本上而言,如果爱丽丝已通过使用其私钥加密签署消息,则给定爱丽丝的公钥和明文中的消息(未加密消息),诸如节点104等其他实体可认证加密版本的消息必须已经由爱丽丝签名。签署通常包括对消息进行散列,签署哈希值和将此标记到消息的明文版本作为签名,从而使公钥的任何持有者能够认证签名。

[0069] 如果Tx_1中的解锁脚本满足Tx_0的锁定脚本中指定的一个或更多个条件(因此,在所示示例中,如果在Tx_1中提供了爱丽丝的签名并进行认证),则节点104认为Tx_1有效。如果是挖矿节点104M,这意味着其将添加至等待工作量证明的事务154池。如果是转发节点104F,则其将事务Tx_1转发到网络106中的一个或更多个其他节点104,从而将在整个网络中传播。一旦Tx_1有效并包括在区块链150中,这将把Tx_0中的$UTXO_0$定义为已花费。请注意,Tx_1仅在花费未花费的事务输出203时才有效。如果试图花费另一事务152已经花费的输出,则即使满

足所有其他条件，Tx_1也将无效。因此，节点104还需要检查先前事务Tx_0中引用的UTXO是否已经花费(已经形成另一有效事务的有效输入)。这是为何区块链150对事务152施加定义的顺序很重要的原因之一。在实践中，给定节点104可维护单独的数据库，标记已花费事务152的UTXO 203，但最终定义UTXO是否已花费取决于是否在区块链150中形成了另一有效事务的有效输入。

[0070] 请注意，在基于UTXO的事务模型中，给定UTXO需要作为一个整体使用。不能“留下”UTXO中定义为已花费的一部分数额，而同时又花费另一部分。但UTXO的数额可以在下一个事务的多个输出之间分割。例如，Tx_0的$UTXO_0$中定义的数额可以在Tx_1中的多个UTXO之间分割。因此，如果爱丽丝不想将$UTXO_0$中定义的所有数额都给鲍勃，她可以使用剩余部分在Tx_1的第二输出中自己找零钱，或者支付给另一方。

[0071] 在实践中，爱丽丝通常还将需要包括获胜矿工的费用，因为现在仅靠生成事务的奖励币通常不足以激励挖掘。如果爱丽丝未包括矿工的费用，Tx_0可能会被矿工节点104M拒绝，因此，尽管技术上有效，但仍然不会传播并包括在区块链150中(如果矿工104M不愿意，矿工协议不会强制他们接受事务152)。在一些协议中，挖掘费不需要其自身的单独输出203(即不需要单独的UTXO)。相反，给定事务152中输入202所指向的总数额与输出203所指定的总数额之间的任何差额都将自动提供给获胜矿工104。例如，假设指向$UTXO_0$的指针是Tx_1的唯一输入，而Tx_1只有一个输出$UTXO_1$。如果$UTXO_0$中指定的数字资产的数额大于$UTXO_1$中指定的数额，则该差额将自动提供给获胜矿工104M。替代地或附加地，这不一定排除可以在其自身事务152的其中一个UTXO 203中明确指定矿工费用。

[0072] 另请注意，如果给定事务152的所有输出203中指定的总数额大于其所有输入202所指向的总数额，则这是大多数事务模型中的另一失效依据。因此，此类事务将不会传播或挖掘到区块151中。

[0073] 爱丽丝和鲍勃的数字资产由区块链150中任何位置的任何事务152中的锁定至他们的未花费UTXO组成。因此，通常情况下，给定方103的资产分散在整个区块链150的各种事务152的UTXO中。区块链150中的任何位置均未存储定义给定方103的总余额的一个数字。客户端应用程序105的钱包功能的作用是将锁定至相应方且在其他随后事务中尚未花费的各种UTXO值整理在一起。通过查询在任何存储节点104S(例如，与相应方的计算机设备102最近或最佳连接的存储节点104S)处存储的区块链150副本，可以实现这一点。

[0074] 请注意，脚本代码通常用示意图表示(即非精确语言)。例如，可写入[Checksig P_A]表示[Checksig P_A]＝OP_DUP OP_HASH160<H(Pa)>OP_EQUALVERIFY OP_CHECKSIG。“OP_...”是指脚本语言的特定操作码。OP_CHECKSIG(又称“Checksig”)是脚本操作码，其取两个输入(签名和公钥)，并使用椭圆曲线数字签名算法(ECDSA)验证签名的有效性。在运行时，移除脚本中任何出现的签名(‘sig’)，但在由‘sig’输入验证的事务中仍保留附加要求，诸如哈希难题。再如，OP_RETURN是脚本语言操作码，用于创建事务的不可花费输出，其可以将元数据储存在事务中，从而将元数据不可变地记录在区块链150中。例如，元数据可包括需存储在区块链中的文件。

[0075] 签名P_A是数字签名。在实施例中，这基于使用椭圆曲线secp256k1的ECDSA。数字签名对特定的数据段进行签名。在实施例中，对于给定事务，签名将对部分事务输入以及全部或部分事务输出进行签名。对输出的特定部分进行签名取决于SIGHASH标志。SIGHASH标志

是包含在签名末尾的4字节代码，用于选择签名的输出(并因此在签名时固定)。

[0076] 锁定脚本有时称为“scriptPubKey”，指其包括相应事务被锁定到的当事方的公钥。解锁脚本有时称为“scriptSig”，指其提供相应的签名。但是更通俗地说，在区块链150的所有应用中，UTXO赎回的条件并不一定包括对签名进行认证。更通俗地说，脚本语言可用于定义任何一个或更多个条件。因此，可以优选更为通用的术语“锁定脚本”和“解锁脚本”。

[0077] 可选的侧信道

[0078] 图3示出了用于实现区块链150的另一系统100。除了附加的通信功能外，该系统100与图1所示的内容基本相同。爱丽丝和鲍勃的每台计算机设备102a,120b上的客户端应用程序分别包括附加通信功能。也就是说，这可使爱丽丝103a建立与鲍勃103b分离的侧信道301(在任何一方或第三方的鼓动下)。侧信道301能够独立于P2P网络实现数据交换。此等通信有时候被称为“链下”通信。比如，当交换爱丽丝与鲍勃之间的事务152时不想将该事务(仍未)发布到P2P网络106或挖掘到区块150，可以采用此等通信，直到其中一方选择将该事务广播到网络106。替代地或附加地，该侧信道301可以用于交换任何其他的事务相关数据，例如密钥、协商的数额或条款、数据内容、等等。

[0079] 通过与P2P覆盖网络106相同的分组交换网络101可建立侧信道301。此外/或者，通过诸如移动蜂窝网络等不同网络、或者诸如本地无线网络等局域网、或者甚至爱丽丝和鲍勃的设备1021,102b之间的直接有线或无线连接可以建立侧信道301。一般而言，本文所指的侧信道301可包括经由一种或更多种联网技术或者通信介质的任何一个或更多个链路，用于“链下”(即独立于P2P覆盖网络106)交换数据。在多个链路被使用的情况下，整个链下链路的捆绑或集合才可以被称为侧信道301。因此，需要注意的是，虽然爱丽丝和鲍勃通过侧信道301对特定的信息或数据片段或者诸如此类进行交换，但这并不一定意味着所有这些数据片段必须通过相同的链路或甚至同一类型网络进行发送。

[0080] 节点软件

[0081] 图4示出了在基于UTXO或基于输出的模型的示例中，在P2P网络106的每个节点104上运行的节点软件400的示例。节点软件400包括协议引擎401、脚本引擎402、堆栈403、应用级决策引擎404以及一个或更多个区块链相关功能模块的集合405。在任何给定节点104处，这些模块可以包括以下三个模块中的任何一个、两个或全部：挖掘模块405M、转发模块405F和存储模块405S(取决于该节点的一个或更多个角色)。协议引擎401被配置为识别事务152的不同字段，并根据节点协议处理此类字段。当接收到具有指向另一先前事务152m-1(Tx_{m-1})的输出(例如，UTXO)的输入的事务152m(Tx_m)时，协议引擎401标识Tx_m中的解锁脚本并将其传递给脚本引擎402。协议引擎401还基于Tx_m的输入中的指针来标识和检索Tx_{m-1}。如果Tx_{m-1}尚未在区块链150上，则可以从相应节点自身的未决事务池154中检索Tx_{m-1}；或者，如果Tx_{m-1}已在区块链150上，则可以从存储在相应节点或另一节点104处的区块链150中的区块151的副本中检索。无论采用哪种方式，脚本引擎401都会标识Tx_{m-1}的指向输出中的锁定脚本，并将其传递给脚本引擎402。

[0082] 因此，脚本引擎402具有Tx_{m-1}的锁定脚本和来自Tx_m的相应输入的解锁脚本。例如，图4中示出的Tx_1和Tx_2，但同样可以应用于任何事务对，诸如Tx_0和Tx_1等。如前所述，脚本引擎402同时运行两个脚本，这将包括根据正在使用的基于堆栈的脚本语言(例如，脚本)将数据放置到堆栈403上并从该堆栈中检索数据。

[0083] 通过同时运行脚本，脚本引擎402确定解锁脚本是否满足锁定脚本中定义的一个或更多个标准，即解锁脚本是否对包括锁定脚本的输出进行解锁?脚本引擎402将该确定的结果返回给协议引擎401。如果脚本引擎402确定解锁脚本确实满足在相应的锁定脚本中指定的一个或更多个标准，则返回结果“TRUE”。否则，返回结果“FALSE”。

[0084] 在基于输出的模型中，来自脚本引擎402的结果“TRUE”是事务有效性的条件之一。通常，还必须满足由协议引擎401评估的一个或更多个进一步协议级条件；例如，Tx_m的输出中所指向的数字资产的总数额不超过(多个)输入指定的总数额，并且Tx_{m-1}的指向输出尚未被另一有效事务花费。协议引擎401评估来自脚本引擎402的结果以及一个或更多个协议级条件，并且只有当它们都为TRUE时，协议引擎401才核实事务Tx_m有效。协议引擎401将事务是否有效的指示输出到应用级决策引擎404。只有在Tx_m确实核实有效的条件下，决策引擎404才可以选择控制挖掘模块405M和转发模块405F中的一个或两个来执行它们涉及Tx_m.的相应区块链相关函数。这可以包括：挖掘模块405M，该挖掘模块将Tx_m添加到节点的相应池154以挖掘到区块151中；和/或转发模块405F，该转发模块将Tx_m转发到P2P网络106中的另一节点104。然而，应当注意的是，在实施例中，虽然决策引擎404不会选择转发或挖掘无效事务，相反，这并不一定意味着，仅因为事务有效，该决策引擎就必须触发该事务的挖掘或转发。可选地，在实施例中，决策引擎404可以在触发这些函数中的一个或两个函数之前应用一个或更多个附加条件。例如，如果节点是挖掘节点104M，则决策引擎可以仅在事务有效且预留足够挖掘费用的条件下选择挖掘该事务。

[0085] 此外，还应当注意的是，在本文中，术语“TRUE”和“FALSE”不一定限于返回仅以单个二进制数(位)形式表示的结果，尽管这确实是一种可能的实现方式。更通俗地说，“TRUE”可以指指示成功或肯定结果的任何状态，而“FALSE”可以指指示不成功或不肯定结果的任何状态。例如，在基于账户的模型(图4中未示出)中，可以通过节点104对签名的隐式协议级核实和智能合约的附加肯定输出的组合来指示结果为“TRUE”(如果两个单独的结果均为TRUE，则认为总体结果为TRUE)。

[0086] 物联网

[0087] 物联网是互联网向日常物理设备和对象的延伸。嵌入计算处理能力和互联网连接的设备可以相互通信和交互，并且可以远程监测和控制。随着时间的推移，由于机器学习、实时分析和多种技术融合，物联网的定义已经发生了变化，尽管人们普遍认为，能够支持无线传感器网络和/或控制系统的设备的系统有可能实现物联网。

[0088] 物联网系统面临着若干挑战。例如，此类系统的可扩展性和成本可能会阻碍物联网系统充分发挥其潜力。当以集中方式连接和控制时，物联网设备需要后端基础设施来传输数据和接收控制命令。这些后端基础设施托管在第三方云服务或本地部署服务器场中。物联网解决方案的可扩展性由后端服务器和数据中心的可扩展性决定，这可能会使物联网服务提供商的运营成本高得令人望而却步。因此，提出的许多物联网解决方案不具有成本效益，不适合在日常场景中使用。网络延迟等性能指标也将成为确定物联网采用率的重要因素。

[0089] 物联网系统面临的另一个挑战是自动化与控制之间的权衡。物联网解决方案旨在实现对日常电子设备的远程访问和控制。大多数物联网解决方案在用户完全控制和设备与其他物联网解决方案组件之间的自动化通信之间取得了平衡。在所述设备或所述物联网系

统发生故障的情况下,需要采取覆盖机制等安全措施。

[0090] 另一个挑战是来自网络攻击的威胁。通过在互联网中实现对设备的自动化控制,用户可能会面临两种形式的潜在安全风险,一种是通过互联网传输物联网设备元数据所产生的隐私风险。例如,如果窃听者从家用电器等设备中获得数据访问权限,窃贼等犯罪分子可能会利用设备使用模式来预测某人何时在家。第二种风险是攻击者或其他第三方有可能获得对物联网设备的控制权。对于性能关键型控制软件,例如用于操作重型机械或危险品的软件,攻击可能会带来灾难性后果。

[0091] 物联网系统可以设计为集中式、分散式和/或混合式。集中式解决方案存在瓶颈,但可以通过物联网系统中的特许组件实现更快、更可靠的控制。分散式状态更新报告使物联网解决方案更具可扩展性。边缘计算有助于缩短关键型应用程序的网络延迟,降低物联网系统对云的依赖性,并对大量物联网数据进行更好的管理。分散式处理的兴起凸显了在系统架构中更好地利用集中式架构和分布式架构的优势的机会。在分层控制结构中结合集中式系统和分布式系统的混合式系统的目的可以在于提高用户安全性和可用性。

[0092] 区块链技术有可能在未来的物联网中发挥主导作用,原因如下:区块链使支付和控制能够集成到一个网络中;现有基础设施可以用于捎带有关设备状态变化的消息;对网络中的数据进行分散控制可以提高用户与设备之间的交互速度。与区块链技术相结合,在现实世界中发挥作用的传统物联网设备能够同时传递消息和交换价值。公共区块链充当全域支付网络和通用商品分类账,其协议内置了强大的加密安全性,可自动解决与物联网相关的多种风险。

[0093] 图5示出了用于实现本公开实施例的示例性系统500。示例性系统500包括由一个或更多个终端设备(即,计算设备)502和一个或更多个桥接节点503(即,运行区块链客户端应用程序并因此充当区块链网络106与第一网络501之间网桥的计算设备)组成的第一网络501。为了清楚起见,第一网络501称为物联网,即通过互联网互连的计算设备的网络。通常,终端设备502和桥接节点503嵌入在日常设备中。终端设备502可以采用多种形式中的一种,例如用户设备(例如,智能电视、智能音箱、玩具、可穿戴设备等)、智能家电(例如,冰箱、洗衣机、烤箱等)、仪表或传感器(例如,智能恒温器、智能照明、安全传感器等)。同样,桥接节点503也可以采用多种形式,可以包括但不限于与终端设备可以采用的相同形式。节点503也可以采用专用服务器设备、基站、接入点、路由器等形式。在一些示例中,每个设备可以具有固定网络(例如,IP)地址。例如,一个、一些或所有终端设备可以是固定设备(例如,智能灯或智能中央加热控制器等),而不是移动设备。

[0094] 物联网是分组交换网络101,通常是互联网等广域互联网。分组交换网络101中的节点503和设备502被布置成在分组交换网络101中形成点对点(P2P)覆盖网络501。每个节点503包括相应的计算机设备,每个计算机设备包括相应的处理装置,所述处理装置包括一个或更多个处理器,例如一个或更多个中央处理单元(CPU)、加速器处理器、专用处理器和/或现场可编程门阵列(FPGA)。每个节点503还包括存储器,即采用非暂时性计算机可读介质形式的计算机可读存储器。该存储器可以包括一个或更多个存储器单元,其采用一个或更多个存储器介质,例如硬盘等磁介质;固态硬盘(SSD)、闪存或EEPROM等电子介质;和/或光盘驱动器等光学介质。

[0095] 物联网中的每个节点503也是区块链节点104。这些节点503被布置成桥接节点(网

关节点),充当第一网络501与区块链网络106之间的网桥(网关)。区块链节点104可以是“侦听节点”。侦听节点运行客户端应用程序,所述应用程序保留区块链的完整副本,核实并传播新的事务和区块,但不主动挖掘或生成新的区块。替代地,节点可以是“简单支付验证节点”(SPV节点)。SPV节点运行轻量级客户端,所述轻量级客户端可以生成和广播比特币事务并间接监控地址,但不会保留区块链的完整副本。

[0096] 物联网中的每个节点503用于直接或间接控制终端设备502。直接连接至终端设备502的节点503可以直接控制该设备。未直接连接至终端设备502的节点503只能间接控制该设备,例如,通过一个或更多个中间节点将控制消息转发给终端节点。每个节点503连接至一个或更多个挖矿节点104M。

[0097] 图5还示出了挖矿节点104M的网络504,该网络是区块链网络106的子集。上文参考图1至图3对挖矿节点进行了讨论。挖矿节点104M用于挖掘传输至区块链150的有效事务(例如,从物联网节点传输的事务)。

[0098] 如图5所示,节点503构成P2P网络501和区块链P2P网络106的一部分,而挖矿节点104M仅构成区块链P2P网络106的一部分。虽然图5中示出终端设备502仅构成P2P物联网501的一部分,但不排除终端设备502也可以是区块链节点104。

[0099] 图6示出了物联网501的示例性拓扑结构。物联网501可以控制主节点503a、一个或更多个中间节点(503b、503c)中的一组或更多组(601)以及一组终端设备502。主节点502a用于控制一个或更多个中间节点(503b、503c)。如果物联网501包括多组(例如,多层)(601a、601b)中间节点,则主节点503a用于直接控制第一组(层)(601a)中间节点(“服务器节点”503b),并间接控制一组或更多组(层)(601b)其他中间节点(例如,一层“从节点”503c)。主节点503a是能够覆盖和控制服务器节点和从节点的控制节点。每个服务器节点503b是能够控制从节点503c的节点。每个从节点503c是受控于服务器节点503b和主节点503a的节点。例如,为了指示终端设备502a,主节点503a通过服务端节点503b向从节点503c发出命令。

[0100] 虽然图6所示的示例性物联网仅示出了两层中间节点(服务器节点和从节点),但其他示例可以包括一组或更多组其他中间节点,例如主节点503a与服务器节点503b之间的中间节点和/或服务器节点503b与从节点503c之间的中间节点。如图所示,每个节点通过相应的连接602连接至一个或更多个其他节点,每个终端设备502通过相应的连接602连接至一个或更多个从节点。一个或更多个节点(例如,主节点)在下文中称为控制节点。每个控制节点是可以通过发出命令来指示其他节点执行操作的节点503。

[0101] 物联网节点503可以对应于功能范围、指令/特权优先级和/或访问范围方面的层次结构。在一些实现方式中,一组分层SPV节点实现了具有三个层级的“物联网控制器”,对应于图5和图6所示的主节点503a、服务器节点503b和从节点503c。主节点503a指示一个或更多个服务器节点503b,每个服务器节点指示一个或更多个从节点503c。每个从节点503c接收来自一个或更多个服务器节点503b的指令。每个从节点503c与一个或更多个物联网终端设备502进行通信,这是物联网控制器503与物联网终端设备502之间的直接通信信道。物联网控制器503的执行状态记录在区块链事务Tx中。每个物联网节点(即,主节点、服务器节点或从节点)都有能力创建对应的事务Tx并将其广播到区块链网络106。每个从节点监控来自终端设备502的触发和/或确认信号,每个物联网节点503具有与任何其他物联网节点交

互的能力,以执行物联网控制器的整体逻辑。

[0102] 主节点、服务器节点和从节点可以分别单独连接到区块链网络106中的节点104,操作区块链钱包(例如,监视区块链地址),并且可能运行全节点(尽管这不是必需的)。主节点503a用于监控直接和间接受其控制的其他物联网节点的活动,以区块链事务Tx的形式向这些节点发出命令,并对警报做出响应。服务器节点503b用于监视多个地址,包括不受服务器节点503b直接控制的地址。主节点503a可以命令服务器节点503b执行操作。从节点503c用于监控直接受其控制的终端设备502的活动。从节点503c受服务器节点503b的直接命令,也可以按主节点503a的命令执行操作。从节点503c充当终端设备502的网关节点(即,终端设备与区块链网络106之间的网关)。终端设备502用于连接至附近的从设备。所述从设备采用链下消息传递协议报告终端设备状态。

[0103] 需要注意的是,虽然在物联网节点503与终端设备502之间进行了区分(即,终端设备502由物联网节点503控制,但其本身不控制物联网节点503),但终端设备502也可以是区块链网络106中的节点104。也就是说,在一些示例中,终端设备502可以操作区块链协议客户端或钱包应用程序。

[0104] 物联网501通过将命令和控制层次结构与区块链网络基础设施的使用相结合,在集中化与分散化之间取得了平衡。网络501的用户可以创建自己的多级控制层次结构,所述层次结构包括客户端-服务器以及设备之间的点对点关系。网络架构包括三层:物联网501;区块链P2P网络104(即,完整区块链客户端和轻量级区块链客户端,例如,所述主节点、所述服务器节点和所述从节点是操作SPV钱包的轻量级客户端);区块链挖矿网络504(核实、传播和存储由物联网节点传播的事务的区块链P2P网络的子集)。区块链网络106充当后端基础设施,物联网501与区块链P2P网络106之间存在重叠。

[0105] 许可协议

[0106] 本公开实施例提供了一种用于向请求访问网络501的节点503或设备502授予加入网络501的权限的协议。在物联网环境中,允许新节点503使用注册机构(例如,网络中的可信实体)提供的链上防伪数字证书加入物联网501。通过确保只有真正的节点才能访问网络和/或控制网络中的其他节点或设备,该协议可解决与网络攻击相关的问题。

[0107] 加入物联网501的权限由注册机构授予(注册机构registration authority也可以称为“权限授予机构”或“认证机构”)。所述注册机构负责向请求实体(例如,请求节点或请求设备)颁发数字证书。具有有效证书的实体可以访问物联网501。所述注册机构包括相应的计算机设备,每个计算机设备包括相应的处理装置,所述处理装置包括一个或更多个处理器,例如一个或更多个中央处理单元(CPU)、加速器处理器、专用处理器和/或现场可编程门阵列(FPGA)。所述注册机构的所述计算设备还包括存储器,即采用非暂时性计算机可读介质形式的计算机可读存储器。该存储器可以包括一个或更多个存储器单元,其采用一个或更多个存储器介质,例如硬盘等磁介质;固态硬盘(SSD)、闪存或EEPROM等电子介质;和/或光盘驱动器等光学介质。

[0108] 为了授予请求实体加入网络501的权限,所述注册机构生成区块链事务Tx,在下文中称为“证书事务”。示例性证书事务如图16a所示。证书事务Tx包括一个或更多个输入以及一个或更多个输出。至少一个输入1501a包括所述注册机构的数字签名。也就是说,所述注册机构具有可从中生成数字签名的第一私钥(例如,第一私钥-公钥对),所述注册机构使用

该数字签名对所述事务进行签名。示例性证书格式如图16b所示。通过对所述证书事务进行签名，所述注册机构证实了事务输出中所包含的数据。所述数字签名只能由知道所述第一私钥的所述注册机构生成。所述事务还具有第一输出1502a(例如，不可花费的输出)，所述输出包括由所述注册机构颁发给所述请求者的数字证书。所述数字证书包含分配给所述请求者的标识符。在物联网501中，对所述请求者来说，所述标识符是唯一的。为所述请求者分配标识符，该标识符一经发出就必须保持不变，并且会出现在颁发给所述设备的任何证书中。优选地，在生成证书时分配设备标识符。然而，不排除所述请求者已经拥有设备标识符的情况，然后，所述设备标识符通过包含在证书中的方式进行认证。

[0109] 所述证书事务生成后，所述注册机构将其传输至区块链网络106中的一个或更多个节点104，以将其记录在区块链150中。所述证书事务记录在区块链150中后，所述请求者可以使用证书向网络501中的其他节点或设备证明所述请求者已经被授予加入网络501的权限。例如，当与网络501中的其他节点503进行通信时，所述请求者可以包含标识所述证书事务的信息，并且因此包含证书。

[0110] 参考图1至图3，在这些示例中，第一节点可以是爱丽丝103a的计算机设备102a，第二节点可以是鲍勃103b的计算机设备102b。

[0111] 如果所述请求者是网络501中的节点503(或者正在请求作为节点加入网络501的权限)，则证书可以包括分配给该节点的唯一公钥。请求节点503加入网络501后，公钥允许请求节点503传输和接收区块链事务。

[0112] 私钥是只有私钥所有者知道的秘密数值。例如，所述私钥可以是256位的字符串。公钥是从所述私钥派生的相关公共值，可以共享。例如，所述公钥可以通过所述私钥与secp256k1椭圆曲线生成点的椭圆曲线乘法计算得出。签名可以是加密签名，例如使用椭圆曲线数字签名算法(ECDSA)生成的签名。可以使用替代的签名方案，例如Rabin签名。

[0113] 所述证书事务可以包括锁定至所述注册机构的第二公钥的第二输出1502b。所述第二公钥可以与用于生成对所述证书事务进行签名的所述签名的所述公钥相同，也可以是不同的公钥。第二输出1502b锁定至所述第二公钥，即，需要知道所述第二公钥才能解锁该输出。例如，所述第二输出可以包括所述第二公钥的哈希，为了由后续事务的输入解锁，该输入必须包括所述第二公钥。当第二输出1502b与所述第二事务的输入一起执行时，对所述输入中提供的所述第二公钥进行哈希处理，并与第二输出1502b中所包含的哈希进行比较。如果这两个哈希匹配，则可以解锁第二输出1502b(前提是满足任何附加约束条件)。

[0114] 可以通过支付到公钥哈希(P2PKH)将输出锁定至公钥。P2PKH是一种将输出锁定至公钥哈希的脚本模式。如果接收者提供了相对于与所述公钥哈希匹配的公钥有效的签名，则可以花费P2PKH输出。也就是说，P2PKH输出要求花费者提供两个项目：公钥，使得所述公钥的哈希与所述P2PKH输出中的地址相匹配；签名，对所述公钥和事务消息有效，但不一定按照该顺序。

[0115] 由于第二输出1502b锁定至所述注册机构的公钥，因此只有所述注册机构可以撤销证书。这样可以防止证书被恶意方撤销。

[0116] 第二输出1502b可以被时间锁锁定至所述第二公钥。通过时间锁锁定的输出只有在预定时间段结束后才能解锁。例如，所述注册机构可以将锁定时间包含在所述证书事务中。锁定时间可以防止所述证书事务的第二输出1502b在某个时间段(该时间段可以由Unix

时间或区块高度等指定)结束前被后续事务成功花费。锁定时间可以通过事务的“nLocktime”字段实现。nLocktime是一个事务参数,规定了在此之前可以花费输出的最短时间。与nLockTime相结合,操作码(例如,OP_CHECKLOCKTIMEVERIFY(CLTV))将防止后续事务花费第二输出(通过导致脚本执行失败),除非第一事务的nLocktime等于或大于提供给操作码的时间参数。由于后续事务只有在其nLockTime发生在过去的情况下才能包含在有效区块中,这就确保了基于CLTV的时间锁在后续事务可以包含在有效区块中之前已经过期。

[0117] 附加地或替代地,第二输出1502b可以是“多重签名”输出。多重签名输出锁定至多个公钥,即所述注册机构的所述第二公钥和至少一个其他公钥。所述其他公钥可以是网络501中的另一个节点的公钥,也可以是在物联网外但在区块链网络106内的第三方节点的公钥。试图花费第二输出1502b的后续事务的输入必须包含多个签名,第二输出锁定至其上的每个公钥一个签名。

[0118] 时间锁可以防止所述注册机构在商定的时间之前或者在未经不同节点(例如,所述主节点)许可的情况下撤销证书。多重签名输出可以防止证书在未经不同节点(例如,所述主节点)许可的情况下被撤销。这两种技术都强制要求最短的证书持续时间。

[0119] 当记录在区块链150中时,每个事务可以通过唯一的事务标识符TxID进行标识。可以通过计算序列化事务字节的(双重)SHA256哈希来生成事务标识符。可以使用其他哈希函数来代替SHA256。所述注册机构可以将所述证书事务的事务标识符传输给所述请求者。这使得所述请求者能够标识所述证书事务,从而获取所述证书事务中的证书。替代地,所述请求者可以侦听从所述注册机构的地址传输至区块链150的事务。

[0120] 如果所述请求者作为节点(例如,服务端节点)加入网络501,则请求节点可以使用事务标识符来获取所述注册机构的第一公钥,并标识从该第一公钥发送的一个或更多个其他事务(即,其他证书事务)。其他事务中的每一个可以包括网络501中的一个或更多个其他节点或设备的相应证书。然后,所述请求者可以获取(例如,下载和保存)这些证书。证书中的信息(例如,设备标识符和/或公钥)可以用于与网络501中的其他节点503和/或设备502通信。例如,所述请求者可以使用另一个节点503的认证公钥将区块链事务传输至该节点,例如通过在锁定至所述认证公钥的事务中包含输出(例如,P2PKH输出)。在接收命令时,所述请求者可以使用证书来检查该命令是否从授权节点503或设备502发出。

[0121] 如果所述请求者作为不能访问区块链的终端设备502加入网络501,所述注册机构可以通过有线连接或无线连接(例如,蓝牙、Wi-Fi等)等方式将证书传输给终端设备。所述注册机构还可以将由一个或更多个第二证书组成的一组第二证书传输给请求终端设备502。颁发给网络501中的相应节点或终端设备的每个第二证书可以用于确保所述请求终端设备与授权节点503和设备502之间的通信。

[0122] 每个证书(第一证书和第二证书)可以包括向其颁发证书的节点503或终端设备502的网络地址(例如,IP地址)。所述请求者可以使用授权(即,认证)节点的网络地址与该节点通信,例如发送传感器读数或命令确认。

[0123] 所述注册机构可以将颁发给所述请求者的证书传输至网络501中的一个或更多个节点和/或终端设备。这些终端设备可以使用该证书与所述请求者进行通信,并验证所述请求者是否已经被授予加入网络501的权限。

[0124] 为了防止第三方查看证书的内容(可能包含敏感信息),所述注册机构可以对证书进行加密。例如,可以使用基于所述注册机构的公钥(可以与所述注册机构的所述第一公钥和/或所述第二公钥相同,也可以不同)的加密密钥对证书进行加密。替代地,所述加密密钥可以是由所述注册机构生成的随机数。

[0125] 在一些示例中,所述注册机构可以仅向所述请求者颁发证书,即,不需要接收明确的请求。在其他示例中,所述请求者可以首先向所述注册机构发送请求。所述请求可以包含所述请求者的一个或更多个凭证。例如,所述凭证可以包含以下一项或更多项:设备类型(例如,笔记本电脑、电话、烤箱、冰箱等)、节点类型(例如,主节点、服务端节点、从节点、终端设备)、网络地址(例如,IP地址,可以是IPv6地址)等。所述注册机构或不同的节点(例如,所述主节点)可以核实所述请求。如果所述请求得到核实,所述注册机构可以生成第一事务,并将其传输至区块链网络106。如果所述请求未获批准,所述注册机构可能不会生成事务。

[0126] 在某些情况下,可能需要撤销颁发给请求者的证书。例如,所述请求者可能受到了破坏,或者可能出现了故障。为了撤销证书,所述注册机构会生成第二区块链事务(“撤销事务”)。所述撤销事务具有引用所述证书事务的所述第二输出(即,锁定至所述注册机构的所述第二公钥的输出)的输入。所述输入包括链接至所述第二公钥的签名。如果所述证书事务的所述第二输出是P2PKH输出,所述撤销事务的所述输入必须包括公钥,使得所述公钥的哈希(例如,OP_HASH160)与所述P2PKH输出中的所述公钥哈希相匹配。P2PKH输出要求花费者提供两个项目:公钥,使得所述公钥的哈希与所述P2PKH输出中的地址相匹配;签名,对所述公钥和事务消息有效,但不一定按照该顺序。

[0127] 所述撤销事务可以包括一个或更多个输出,例如锁定至所述注册机构的第三公钥(可以与所述注册机构的所述第一公钥和/或所述第二公钥相同,也可以不同)的输出。然后,所述注册机构将所述撤销事务传输至区块链网络106,以将其记录在区块链150中。将所述撤销事务记录在区块链150中后,从未花费的事务输出(UTXO)集中删除所述证书事务。UTXO是区块链事务未被另一个区块链事务花费的输出。当网络501中的不同节点试图标识颁发给请求节点的证书时,该节点就会发现包括该证书的所述证书事务已被花费,并将此解释为证书被撤销。网络501中的各节点能够通过观察从发证地址(即,所述第二公钥)生成的事务以及传输至发证地址的事务,动态更新其对等节点列表(即,授权节点/认证节点列表)。

[0128] 节点/设备证书的有效性可能取决于三个标准:发行密钥是公认的发行密钥(包含在由主密钥签名的证书中);证书根据预定协议正确格式化;所述证书事务中可花费的输出未花费。证书撤销后便可进行更新。所述注册机构花费旧证书中的UTXO,然后使用更新后的信息创建新证书Tx。然后,所述注册机构可以将新的证书输出点位置索引广播到物联网501中的设备。这也适用于所述注册机构自己的(自签名)证书。

[0129] 如上文所述,证书包括所述请求者的唯一标识符。证书还可以包括所述请求者的唯一公钥。一般而言,证书可以包括一个或更多个以下字段:

字段大小（字节）	字段名	数据类型	描述
4	物联网协议标识符	uint32_t	指示协议的前缀
1	有效负载类型	uint16_t	指示消息是常规消息还是证书的单字节标识符
32	新设备 ID	char[32]	新设备的唯一设备 ID
33	新设备公钥	char[32]	用于与节点（如果在区块链 P2P 网络上）进行通信的 *secp256k1*（压缩）公钥
4	设备类型	uint16_t	设备类型（例如，笔记本电脑、电话、烤箱、冰箱、灯柱等）
4	物联网节点类型	uint16_t	节点类型，例如主节点、服务器节点、从节点、终端设备
16 + 2	IPv6 地址+端口	Char[16]	IPv6 网络地址网络字节顺序
4	UNIX 时间创建日期	uint16_t	设备创建日期
4	UNIX 时间证书到期日	uint16_t	证书到期并且必须颁发新证书的 UNIX 时间
0-80	附加设备信息	char[]	附加设备信息（包括制造商信息）

[0130] 这种格式的证书需要104到184字节的数据，例如编码在区块链事务的不可花费的(OP_RETURN)输出中。

[0131] 如上文所述，网络501可以包括主节点503。在一些示例中，所述注册机构可以包括主节点503a。所述请求者可以是节点503或终端设备502。因此，主节点503a可以自己向节点503或设备502颁发证书。在附加或替代示例中，所述请求者可以是主节点503a。如果主节点503a既是所述注册机构又是所述请求者，则主节点503a自签自己的证书。

[0132] 物联网501访问权限已获授权，并使用许可(或自举)算法来授权新的实体(节点或设备)。下面提供了两种示例性算法。主节点503a可以由注册机构控制，所述注册机构可以(直接或间接)验证请求加入网络501的权限的任何新设备的凭证。然后，主节点503s可以颁发链上证书，该证书会被广播到网络501中的所有其他节点503。不操作区块链钱包的设备的自举算法不同，因为该算法依赖于IP地址之间的通信，而不是通过区块链事务。

[0133] 示例性许可算法(主端/服务端/从端)

[0134] 步骤1：运行节点的计算设备向注册机构注册其凭证，包含所有相关制造信息。所述注册机构可以是操作主节点503a的同一实体。然而，优选地，用于对常规命令或消息进行签名的公钥不同于用于对所述证书事务进行签名的公钥。

[0135] 步骤2：对密钥PK_{Issue}有控制权的所述注册机构核实凭证的真实性，并确定所述设备是否应被授权加入物联网501。

[0136] 步骤3：如果所述设备得到授权，所述注册机构为该节点创建唯一的证书事务。该事务从PK_{Issue}广播到PK_{Issue}地址。所述注册机构向新节点发送TxID。

[0137] 步骤4:所述新节点找到事务并向所述注册机构标识PK_{Issue}。所述新节点下载并验证由PK_{Issue}颁发且当前处于生效状态的所有证书。

[0138] 步骤5:网络501上已经存在的服务器节点和从节点用于侦听广播到PK_{Issue}的事务以及从PK_{Issue}广播的事务。在看到新的事务、下载并评估新的设备证书后,上述节点可以配置其钱包,与新节点进行通信。

[0139] 示例性许可算法(终端设备)

[0140] 步骤1:终端设备502向所述注册机构注册其凭证,包含所有相关制造信息和IP地址。同样,所述注册机构可以是操作主节点503a的同一实体。然而,优选地,用于对常规命令或消息进行签名的公钥不同于用于对所述证书事务进行签名的公钥。

[0141] 步骤2:对密钥PK_{Issue}有控制权的所述注册机构核实凭证的真实性,并确定设备502是否应被授权加入物联网501。

[0142] 步骤3:如果设备502得到授权,所述注册机构为该设备创建唯一的设备证书事务。该事务从PK_{Issue}控制的地址广播,或广播到PK_{Issue}控制的地址。证书数据经过加密,因此,只有物联网节点可以看到详细信息。

[0143] 步骤4:所述注册机构发送使终端设备502能够创建对等节点列表的证书列表。该列表使终端设备502能够连接至网络501中的其他节点503c,并解释公钥层次结构。

[0144] 步骤5:网络501中的节点503能够看到设备证书,并且能够与终端设备进行IP到IP(TLS)通信。

[0145] 请求与响应协议

[0146] 本公开还提供了一种用于网络(例如,物联网)501中的节点使用区块链事务Tx发出命令请求、根据这些命令请求指示设备并发出命令确认的协议。虽然将参照物联网501对实施例进行描述,但一般而言,本公开的教导可以应用于任何网络,该网络包括操作区块链协议客户端应用程序105的节点,以及至少可由这些节点的子集控制的终端设备。

[0147] 网络501中的第一桥接节点503(例如,主节点503a或服务器节点503b)生成第一事务Tx_1,该事务包括输入和输出,所述输入由所述第一节点签名,所述输出包括命令数据。所述命令数据包括要控制的终端设备502的标识符以及用于控制终端设备502的命令消息。所述第一节点可以是命令的发起者。也就是说,所述第一节点可以生成所述命令数据。

[0148] 所述第一节点可以将第一事务Tx_1传输至控制终端设备502的第一网络501中的第二桥接节点503(例如,从节点503c)。第一事务Tx_1可以在链下传输,即,无需传输至区块链。例如,第一事务Tx_1可以通过互联网等方式直接从所述第一节点发送至所述第二节点。例如,所述第一节点可以是服务器节点503b,所述第二节点可以是从节点503c。替代地,第一事务Tx_1可以通过一个或更多个中间节点等间接发送。例如,第一事务Tx_1可以通过服务器节点503b从主节点503a发送至从节点503c。所述第二节点可以通过有线连接或无线连接(例如,通过以太网或Wi-Fi连接)连接至终端设备502。

[0149] 附加地或替代地,所述第一节点可以将第一事务Tx_1传输至区块链网络106,以将其记录在区块链150中。这依赖于第一事务Tx_1是有效的事务。如下文所述,在一些情况下,最好不要将第一事务Tx_1传输至区块链。

[0150] 参考图1至图3,在这些示例中,所述第一节点可以由爱丽丝103a的计算机设备102a组成,所述第二节点可以由鲍勃103b的计算机设备102b组成。如上文所述,爱丽丝和鲍

勃可以使用侧信道(例如,侧信道301)来交换事务,而无需(尚未)将该事务发布到区块链网络106上或进入链150,直到其中一方选择将其广播到网络106。

[0151] 所述第二节点可以直接或间接地从所述第一节点获取第一事务Tx_1,例如,第一事务Tx_1可以通过一个或更多个中间节点转发给所述第二节点。所述第二节点使用所述命令数据将控制指令传输至由所述命令数据中的设备标识符(“设备ID”)标识的终端设备502。所述命令数据中的控制消息可以定义终端设备502的所需操作。所述控制消息可以用于使所述第二节点将多个可能的指令中的一个特定指令传输至终端设备502。替代地,所述第二节点可以用于向终端设备502发送单个指令,即,所述第二节点仅向所述终端设备发送相同的指令。例如,如果终端设备502是像传感器一样的简单设备,并且指令是对传感器读数的请求,则可能会出现这种情况。

[0152] 命令(即,对终端设备的指令)可以通过有线连接或无线连接(例如,使用Wi-Fi)在链下传输至设备。替代地,如果该设备也是网络中的节点,则该命令可以通过区块链事务Tx传输。

[0153] 在一些实施例中,设备与控制器通信的请求和响应周期可以由所述第一节点和所述第二节点实现。所述请求(命令)作为包含输出的部分完成的事务发出,所述输出包括所述命令数据(例如,OP_RETURN有效负载)。所述响应(命令确认)是对包含请求者节点和响应者节点的签名的已完成事务进行广播。事务延展性可以实现这种通信方法,因为消息接收方可以添加输入和输出,但不能更改所述命令数据(例如,OP_RETURN有效负载)。

[0154] 操作码是在脚本引擎402中使用的指令字节或指令包,用于指示矿工104M对数据执行基于堆栈的操作和加密操作。在本文中,脚本引擎402是执行环境,用于在区块链事务Tx中核实脚本;堆栈403是数据结构(元素集合),涉及以下两种主要操作:“入栈push”,向集合中添加元素;“出栈pop”,移除最近添加的元素。操作码被设计用于对堆栈元素执行操作。在核实事务Tx时,脚本引擎402不会在OP_RETURN操作码之后执行输出脚本(ScriptPubkey)中的任何数据。在实践中,这意味着剩余的脚本数据可以是任意数据,而输出本身是不可花费的(在一个区块链协议中,OP_RETURN之前需要有OP_FALSE操作码,以确保输出的不可花费性)。

[0155] 从所述第一节点传输至所述第二节点的第一事务Tx_1可以在没有第二输出的情况下传输。即,事务包括单个输出(所述输出包括所述命令数据)。为了完成部分完成的事务,该第二者可以通过向所述第一事务添加输入和输出来更新事务。所述输入包括所述第二节点的签名,即,使用所述第二节点的私钥生成的签名。所述输出是锁定至所述第二节点的公钥的输出,例如P2PKH输出。为了花费P2PKH输出,花费事务的输入必须包括公钥,使得所述公钥的哈希(例如,OP_HASH160)与所述P2PKH输出中的所述公钥哈希相匹配。P2PKH输出要求花费者提供两个项目:公钥,使得所述公钥的哈希与所述P2PKH输出中的地址相匹配;签名,对所述公钥和事务消息有效,但不一定按照该顺序。所述公钥可以对应于用于生成所述签名的所述私钥。替代地,所述签名可以链接至第一公钥,所述输出可以锁定至不同的公钥。然后,所述第二节点可以将完成的事务传输至区块链网络106。所述完成的事务(在这些实施例中称为命令事务)在区块链150中可供其他节点(例如,所述第一节点)查看,并充当设备所执行命令的记录。也就是说,广播事务后,独立观察者可以看到哪个公钥发出了命令/消息,哪个公钥对其做出了响应。

[0156] 图7a和图7b示出了示例性部分第一事务Tx_1(部分)和示例性更新后的第一事务Tx_1(完整)。所述部分第一事务包括单个输入701a和单个输出702a。所述更新后的第一事务包括由所述第二节点添加的输入701b和输出702b。SIGHASH_SINGLE签名类型可用于实现所需级别的事务延展性。例如,具有公钥PK_0的节点向具有公钥PK_1的节点发送指令。该指令被编码在使用SIGHASH_SINGLE签名类型签名的事务的不可花费的输出(例如,OP_RETURN输出)中(图10a)。部分完成的事务是有效的。指令完成后,具有PK_1的所述第二节点会添加锁定至其地址的输出。然后,具有PK_1的所述第二节点通过使用SIGHASH_ALL签名类型对整个事务进行签名来完成事务(见图10b)。

[0157] 在替代实施例中,从所述第一节点传输至所述第二节点的第一事务Tx_1可以与第二输出一起传输。所述第二输出锁定至所述第二节点的公钥。例如,所述第二输出可以是锁定至所述第二节点的公钥的P2PKH。

[0158] 为了完成第一事务Tx_1,所述第二节点通过向所述第一事务添加输入来更新所述第一事务。第一事务Tx_1现在包括两个输入和两个输出。所述第二输入包括所述第二节点的公钥。所述第二输入中的所述公钥可以与所述第二输出锁定至其上的所述公钥相同,也可以不同。完成后,所述更新后的第一事务(在这些实施例中称为命令事务)被发送至区块链网络106,以将其包含在区块链150中。广播命令事务后,独立观察者可以看到哪个公钥发出了命令/消息,哪个公钥对其做出了响应。

[0159] 与所述第一事务的所述第一输入所引用的数字资产数额相比,锁定至所述第二节点的所述公钥的所述第二输出可以转移更大的数字资产数额。在这种情况下,第一事务Tx_1是部分完成的事务,不会被区块链网络106的其他节点视为有效。也就是说,第一事务Tx_1不会满足区块链节点所遵循的共识规则,因此不会被挖掘到区块链150的区块152中。在更新第一事务Tx_1时,所述第二节点必须确保所述第一输入和所述第二输入所引用的数字资产的总数额大于锁定至所述第二输出的数字资产的数额。

[0160] 图8a和图8b示出了示例性部分第一事务Tx_1(部分)和示例性更新后的第一事务Tx_1(完整)。所述第一事务包括锁定至所述第二节点的所述公钥的第一输出802a和第二输出802b中的命令数据。所述更新后的第一事务包括由所述第二节点添加的附加输入801b。如果具有PK_0的所述第一节点向具有PK_1的所述第二节点发送希望仅由具有PK_1的所述第二节点执行的指令,则可以发送锁定两个输出(802a、802b)的部分完成的事务,但不支付费用(因此,该事务不会被挖掘或传播)。为了赎回锁定至PK_1的数字资产,具有PK_1的所述第二节点需要提供支付费用的输入801b。为了使用部分完成的事务发出命令,$< \mathrm{Sig}_{\mathrm{PK}_0} >$的SIGHASH标志设置为SIGHASH_ANYONECANPAY,并包含带有所述命令数据的OP_RETURN输出。这意味着,虽然第一输出802a中包含的命令数据是固定的,但任何人都可以添加附加输入。接收到该命令的公钥可以添加附加输入801b来赎回输入801a中的资金。为了确保新输入801b的安全并防止进一步的事务延展性,资金接收者添加最小值(dust)输入,并使用SIGHASH_ALL对事务输出进行签名。

[0161] 需要注意的是,SIGHASH标志是添加到事务输入的签名中的标志,用于指示所述签名对事务的哪一部分进行签名。默认为SIGHASH_ALL(对除ScriptSig以外的所有事务部分进行签名)。可以修改事务的未签名部分。

[0162] 下面将参考图9、图10a和图10b提供一种示例性请求和响应算法。控制设备503b用

于与网络501中的其他节点进行通信，并且可以计算出与网络上任何其他节点的最短通信路线。例如，PK_{serv}标识出PK_{slave}是距离具有device_ID的设备最近的控制器。

[0163] 步骤1：具有公钥PK_{serv}的控制设备503b向具有公钥PK_{slave}的第二控制设备503c发送部分命令Tx_1(见图10a)。事务中包含的物联网消息指定了命令以及具有device_ID的目标设备。

[0164] 步骤2：第二控制设备(PK_{slave})根据网络501的规则检查事务的签名是否有效，并检查物联网消息有效负载中包含的消息是否有效。

[0165] 步骤3：第二控制设备(PK_{slave})通过链下通信(例如，有线连接、蓝牙、IP到IP)向设备(device_ID)发送命令消息(“Msg”)。

[0166] 步骤4：通过命令请求的操作完成后，设备(device_ID)将命令完成或确认消息(“ack”)发送回第二控制设备(PK_{slave})。

[0167] 步骤5：第二控制器(PK_{slave})添加第二输入和签名，并完成事务(见图10b)。这表示第二控制器确认命令完成。

[0168] 步骤6：第二控制器(PK_{slave})将完成的事务广播到区块链(挖矿)网络504。

[0169] 在一些实施例中，第一事务Tx_1可以是命令请求事务Tx_1(请求)，而不是事务。也就是说，所述第一节点可以向网络中的两个或更多个节点发送请求，请求获得控制终端设备502的批准。例如，所述第一节点可以是请求获得主节点503a和不同桥接节点503(例如，另一个服务器节点503b)的批准的服务器节点503b。第一事务Tx_1包括所述命令数据，其中包括用于控制终端设备502的命令消息。然而，所述第一节点将第一事务Tx_1传输至可以批准命令请求的两个或更多个节点，而不是将第一事务Tx_1传输至控制终端设备502的节点。也就是说，所述第一节点包括锁定至网络501中的两个或更多个节点的输出。为了解锁输出，必须在后续事务Tx_2的输入中提供每个节点的签名。第一事务Tx_1被广播到区块链网络160，以将其包含在区块链150中。当两个或更多个节点看到事务Tx_2时(通过侦听支付给其公钥或公钥地址的事务)，如果所述两个或更多个节点想要批准命令请求，则所述两个或更多个节点中的每个节点都要对命令批准事务Tx_2(批准)的输入进行签名，并将该事务Tx_2广播到区块链网络106。命令批准事务Tx_2包括与命令请求事务Tx_1相同的命令数据，以及锁定至控制终端设备502的所述第二节点的所述公钥的附加输出。

[0170] 图11a和图11b示出了示例性第一事务和第二事务，第一事务Tx_1是命令请求事务，第二事务Tx_2是命令批准事务。第一事务Tx_1包括由所述第一节点签名的输入1101a、包含命令数据的第一输出1102a以及锁定至两个不同公钥的第二输出1102b：一个是第三节点的公钥，另一个是第四节点(也是物联网501的桥接节点503)的公钥。第二事务Tx_2包括由第三节点和第四节点签名的输入1101b、第一输出1102a以及锁定至所述第二节点的所述公钥的第二输出1102c。多重签名脚本可以实现命令的多因素批准。在某些情况下，命令的解释或证书的核实可能需要两个或更多个节点(例如，物联网中的节点)的签名。此外，还可能需要检查可花费的输出是否在UTXO集合中。例如，具有公钥PK_0的第一节点可能想要指示具有公钥PK_3的第二节点执行某项操作。作为安全要求，该命令要求具有公钥PK_1的第三节点和具有公钥PK_2的第四节点同时批准。具有公钥PK_0的所述第一节点创建第一事务，所述第一事务将资金发送至多重签名地址。所述第一事务还包含需要批准的命令数据(将命令编码到PK_3中)。作为响应，具有公钥PK_1和PK_2的节点503可以选择从多重签名地址支出资金，但条件是双方

都提供签名。批准可以理解为命令请求事务中输出的花费。虽然这些附图示出了2-2支付到多重签名(P2MS)输出,但一般而言,P2MS输出可以是n-n输出,其中n是任意整数,P2MS是一种允许支付人将输出锁定至多个地址的脚本模式。为了被花费,输出可能需要一组指定公钥的一个或更多个签名。

[0171] 在一些实施例中,如上文所述,所述第一节点可以生成命令请求事务Tx_1。附加地或替代地,所述第一节点可以响应于由物联网501的不同节点生成的命令请求事务,来生成命令批准事务Tx_2。例如,所述第一节点可以是能够批准请求的主节点503a。在这种情况下,区块链150包括所述命令请求事务,其包括锁定至所述第一节点的公钥和物联网501的一个或更多个其他节点503的一个或更多个相应公钥的输出。例如,所述第一节点可以是批准向所述第二节点控制的终端设备502发出的命令的节点。如果所述第一节点批准该命令,则所述第一节点对引用所述命令请求事务的输出的命令批准事务进行签名。如果所述第一节点是对所述命令批准事务进行签名的最后一个节点,则所述第一节点将所述事务传输至区块链网络106。广播事务Tx_2可以理解为是对命令请求的批准。

[0172] 当所述第二节点(例如,从节点)获得命令事务或命令-批准事务时,所述第二节点向由命令数据中的设备标识符(Device_ID)标识的设备发送命令(Msg)。在一些示例中,设备502可以向所述第二节点传送确认消息(Ack),以表明其已经接收到所述命令和/或已经执行了所述命令。在这些示例中,所述第二节点只能在接收到所述设备的确认信息的情况下更新所述第一事务(然后广播更新后的事务)。这可为终端设备已经执行命令提供进一步的支持证据。

[0173] 由于大多数日常小型电子设备的资源限制,可能无法轻松监控区块链150和/或甚至无法与其即时位置以外的物联网组件进行通信,因此对终端设备502的控制是在本地(第二节点到设备)和链下执行的。终端设备收发的消息可以采用原始命令数据(例如,OP_RETURN有效负载)的形式,而不需要附加事务元数据。这确保了包含消息的数据包保持较小,并且不需要计算密集型操作(例如,椭圆曲线数学)。

[0174] 在一些实施例中,对事务中包含的命令数据进行加密。所述命令数据可以使用基于随机数的加密密钥进行加密,尽管优选地,所述加密密钥是根据所述第一节点的公钥和所述第二节点的公钥生成的。所述第一节点和/或所述第二节点的所述公钥可以是认证公钥(认证公钥将在下文中进行讨论)。认证公钥包含在颁发给所述第一节点和/或所述第二节点的相应证书中。这些公钥可以不同于所述第一节点和所述第二节点用于生成和更新所述第一事务的公钥。也就是说,用于生成签名或锁定事务输出的公钥(“事务公钥”)可以不同于用于生成所述加密密钥的公钥。在其他示例中,所述事务公钥可以用于生成所述加密密钥。

[0175] 用于对所述命令数据进行加密的所述加密密钥可以由所述第一节点和所述第二节点独立生成。例如,所述第一节点可以根据所述第一节点已知的私钥和所述第二节点的公钥来生成所述加密密钥。所述第二节点可以根据与所述第一节点的私钥对应的公钥和与所述第二节点的公钥对应的私钥来生成所述加密密钥。这确保了所述第一节点和所述第二节点都可以解密所述命令数据,例如,这样所述第二节点就可以访问命令消息以指示所述设备。

[0176] 可选地,包括所述命令数据的所述第一事务的输出可以包含用于对所述命令数据

进行加密的所述加密密钥。所述加密密钥可以使用不同的第二加密密钥进行加密。知道所述第二加密密钥的一方可以解密所述加密的加密密钥，然后解密所述加密的命令数据。例如，对于主节点等实体来说，能够查看网络节点收发的所有命令消息可能是有益的。可以根据所述第一节点的公钥和不同节点(例如，所述主节点)的公钥生成所述第二加密密钥。所述第一节点的所述公钥可以是所述第一节点的认证密钥或事务公钥。同样，所述主节点的所述公钥可以是所述主节点的认证密钥或事务公钥。

[0177] 加密可以是对称的。对称加密为链上数据提供了与HTTPS为互联网通信提供的相同级别的隐私。为此，所述主节点创建了加密密钥，用于对本地物联网节点之间的所有常规消息进行加密。每个物联网事务OP_RETURN有效负载(即，包括所述命令数据的输出)可以有两个数据块。一个是BIE1ECIES加密的物联网消息，其中所述加密密钥是使用请求和响应设备(端到端)的(认证)公钥派生的。另一个是BIE1 ECIES加密的加密/解密密钥，用于物联网消息。椭圆曲线集成加密方案(ECIES)是一种基于Diffie-Hellman交换的加密方案。所述第二加密密钥(用于对该数据推送进行加密)是使用主密钥和请求者公钥派生的。将附加字节推送添加到包含物联网消息解密密钥的加密有效负载的末尾，该字节推送本身在所述请求节点与所述主节点之间使用BIE1进行加密。这可确保所述主节点能够查看网络上设备之间发送的解密数据。可以采用其他加密技术，例如美国加密标准(AES)加密技术。

[0178] 图12a和图12b示出了包含加密有效负载数据的所述命令和响应事务。

[0179] 如上文所述，在生成和更新事务时，所述第一节点和所述第二节点可以使用事务公钥。虽然公钥用于标识物联网中的节点，但优选地，这些公钥不应该用于对事务进行签名。例如，每个节点可以具有包含在由注册机构颁发的证书中的公钥。认证密钥(例如，具有由所述主节点签名的对应证书的公钥)的所有者可以派生出共享密钥，该共享密钥对第三方屏蔽其身份，并且可以用于对事务进行签名。

[0180] 如上文所述，物联网501可以包括主节点503a。主节点503a可以获取(例如，生成)种子密钥，然后生成一组(例如，多个)私钥，每个私钥都基于所述种子密钥。然后，主节点503a可以将所述一组私钥(以下简称联合私钥)传输至网络中的节点(例如，服务器节点和从节点)。每个节点接收一组相同的联合私钥，但不接收种子私钥。

[0181] 每个节点都有相应的主私钥，例如与该节点的认证公钥对应的私钥。包括主节点503a在内的每个节点使用所述一组联合私钥来生成一组对应的二级(或事务)私钥。通过将所述相应节点的主私钥添加到每个联合密钥来生成事务私钥。对于每个节点，可以根据所述一组事务私钥生成一组对应的事务公钥。

[0182] 所述第一节点可以是主节点503a，即生成所述一组联合私钥的节点。替代地，所述第一节点可以是从所述主节点接收所述一组联合私钥的中间节点(例如，服务器节点或从节点)。在一些示例中，每个节点可以仅使用一次事务公钥。

[0183] 下面提供了一种用于根据认证密钥生成事务密钥的示例性密钥屏蔽算法。已向本地物联网501中的所有节点503颁发注册公钥的证书。具体而言，服务器节点具有认证密钥PK_{Serv}，主节点具有认证密钥PK_{Master}，所述认证密钥具有私钥sk_{Master}。

[0184] 步骤1：主节点503a根据种子密钥生成主扩展私钥m^{joint}。m用于生成密钥钱包，这些密钥将在物联网501中共享，并用于屏蔽每个节点的地址。联合钱包具有索引密钥：

[0185] $sk_{i,j}^{joint}$, $PK_{i,j}^{joint} = sk_{i,j}^{joint} \cdot G$。

[0186] 步骤2:主节点503a采用链下端到端加密方案(例如,BIE1 ECIES)在链下消息中与物联网中的其他节点共享m。

[0187] 步骤3:服务器节点503b在获得m后,可以派生出钱包中的一组分层确定性密钥对。

[0188] 步骤4:网络501中的每个节点503通过将其认证密钥对中的私钥添加到联合钱包生成的私钥中,来生成其钱包私钥。例如,所述主节点生成事务签名密钥,$s_{i,j}^{M}$,其中

[0189] $sk_{i,j}^{M} = sk_{i,j}^{joint} + sk_{Master}$。

[0190] 步骤5:每个节点503可以通过将其他节点的公钥添加到联合钱包中的公钥中,来标识物联网501中其他节点503的所有支付端点(地址)。例如,服务器节点503b可以派生出所述主节点的支付地址公钥,$PK_{i,j}^{M}$,其中

[0191] $PK_{i,j}^{M} = PK_{i,j}^{joint} + PK_{Master}$。

[0192] 物联网501中的每个节点503可以派生出其自己的钱包,并监控其他设备的地址,但条件是知道m以及相关物联网证书的位置。

[0193] 加密和密钥屏蔽都能保护设备活动数据不会泄露给第三方,同时仍能确保本地物联网501中所有节点503的设备可见性。

[0194] 由物联网501中的节点503传输的每个事务包括输出,所述输出包括命令数据。所述输出和/或所述命令数据可以包括协议标志,用以指示所述输出包括所述命令数据。这使得物联网设备和独立第三方能够标识链上命令、操作或状态更新何时发生。

[0195] 图13示出了第一事务的示例性命令数据输出。所述第一事务包括输入(未示出)和输出1301,所述输入包括第一节点的签名,输出1301包括所述命令数据。所述第一事务还可以包括第二输出(未示出),将在下文中进行讨论。在该示例中,协议标识符(4字节)之后是93字节有效负载,其中包含物联网通信信息。所述通信信息包括命令指令的预期接收者的32字节设备ID、设备证书的位置、命令和设备状态。在一些示例中,发出新命令或状态更新的每个事务必须遵循这种格式,否则将被视为无效命令。如果某个字段对于任何链上消息都不是必需的,则可以将其字节设置为0x00000000。如下文所述,优选地对有效负载数据本身进行加密。然后,只有持有解密密钥的各方才能访问有效负载数据。下表描述了示例性物联网消息有效负载的字段。

[0196]

字段大小（字节）	描述		数据类型	注释
4	物联网协议标识符		uint32_t	指示物联网协议的前缀
1	有效负载类型		uint16_t	指示消息是常规物联网消息还是证书的单字节标识符
4	软件版本号		uint32_t	物联网版本号（协议更新/升级所需）
32	设备 ID		char[32]	服从命令/消息的设备的唯一设备 ID
40 (32 + 4 + 4)	设备证书位置	TXID	char[32]	包含设备证书的事务的事务 ID
		VOUT	uint16_t	证书 TX 中的撤消 UTXO 位置
		VOUT	uint16_t	证书有效负载的输出编号
4	命令/消息		uint32_t	对发送到带有设备 ID 的设备的命令或消息进行编码的字符串
4	状态		uint16_t	当前设备状态
4	先前状态		uint16_t	设备的最新先前状态

[0197] 设备状态副本是设备报告状态或期望状态的逻辑表示。在物联网消息中，设备状态信息编码在设备ID、状态和先前状态中。与设备ID相关的最新事务表示当前设备状态。包含与设备状态相关的命令、响应和数据的消息包含在区块链上带有时间戳的区块中，使用公钥加密和工作量证明进行保护。

[0198] 总之，物联网501中的节点503通过直接使用包含物联网命令数据的事务以及连接至区块链网络106进行通信，来广播事务。区块链150用作永久数据存储器，用于记录来自物联网组件的命令和状态更新，以及发出与物联网设备502相关的报告和警报。该协议可以利用以下一个或更多个特征。

[0199] 请求和响应消息传递系统——使用用于接收和确认命令的请求和响应系统。请求是对可由终端设备解释的物联网逻辑进行编码的线下(点对点)事务。根据区块链网络106中的事务可见性来解释响应或确认。

[0200] 线下事务传播——直接(点对点)发送编码到事务中的指令。主节点、服务器节点和从节点可以通过验证事务签名来独立验证事务的来源。这也是控制器的一种支付方式。

[0201] 节点与终端设备之间的直接通信——如果编码到事务命令有效负载中的指令是针对终端设备的，服务器或从节点可以从事务中提取所述指令，并将其直接传送给终端设备。

[0202] 广播事务作为操作确认——将事务广播到区块链网络表示命令中编码的操作已由具有设备ID的设备执行。

[0203] 编码设备状态和历史记录的挖矿事务——区块链充当存储完整设备状态和历史记录的(逻辑)集中式和物理分布式数据库。

[0204] 实施例提供了以下一种或更多种有利特征。

[0205] 底层区块链基础设施的安全性——使用公钥加密和工作量证明对编码价值转移和记录物联网交互的所有事务进行保护。基于secp256k1参数的椭圆曲线密码(ECC)为用于标识物联网节点的公钥/私钥提供保护,工作量证明为记录物联网状态和历史记录的区块链提供保护。

[0206] 安全密钥管理和混淆——密钥混淆技术用于确保敏感公钥不会因过度使用其对应的私钥而易受攻击。密钥混淆还使物联网解决方案组件能够通过屏蔽其公共地址来提高隐私性。

[0207] 加密——所包含的特定设备数据经过端到端加密(例如,BIE1或AES),因此,只有具有解密密钥的物联网节点才能获得访问权限。

[0208] 示例性用例

[0209] 示例性用例包括公共图书馆的打印机服务。在大多数公立图书馆或大学图书馆,打印费用是通过对每张纸收取最低数额(例如,6-10p)进行支付的。在当前(集中式)模式中,用户开设由图书馆管理部门管理的账户。账户需要提前记入,事务必须由图书馆操作的软件管理,这给图书馆带来了巨大的管理负担。本公开通过结合使用许可协议和点对点控制协议解决了这个问题。图14示出了用于P2P打印的示例性物联网501。

[0210] 1)图书馆管理员建立主节点503a(由图书馆管理员控制),并配置直接控制打印机(终端设备)502的从节点503c。

[0211] 2)管理员配置规则引擎,从节点503c和终端设备502将使用该引擎解释消息。规则引擎是执行一个或更多个规则的系统。

[0212] 3)图书馆管理员配置从节点503c。从节点503c是可以直接指示打印机执行物理操作的支付接收者。

[0213] 4)对主节点503a通过标准注册方法/登录方法(例如,用户名和密码等一个或更多个凭证)授权的图书馆新用户进行验证。物联网许可算法在后端执行。图15a示出了图书馆管理员用于向图书馆新用户颁发证书的示例性事务。在这种情况下,图书馆管理员是主节点503a,也是具有证书撤销权的注册机构。

[0214] 5)当用户想要打印文件时,打印的文件可以在图书馆内联网系统内发送。命令事务与文件一起发送。该事务包含支付到协调打印机的从节点、打印机设备ID以及SIGHASH_SINGLE事务签名。图15b示出了由笔记本电脑(服务器节点)发送至控制器(从节点)的示例性命令事务。

[0215] 6)控制器(从节点)将认证事务是否是有效的区块链事务,已授权事务源(公钥)进入系统,并且事务值足以支付命令中所包含指令的费用。

[0216] 7)如果所有检查都通过,从节点会指示打印机执行用户请求的操作。

[0217] 8)从节点添加将支付锁定至其地址的输出,并添加具有SIGHASH_ALL的签名,然后将事务广播到区块链网络106。图15c示出了由控制器广播到区块链网络106的示例性命令-确认事务。

[0218] 结论

[0219] 应当理解,上述实施例仅通过示例的方式进行描述。更通俗地说,可根据下述任何一个或更多个语句提供一种方法、装置或程序。

[0220] 语句1、一种用于向请求者授予加入第一网络的权限的计算机实现的方法，其中所述第一网络包括一组桥接节点和一组设备，所述一组设备可由所述一组桥接节点中的一个或更多个桥接节点控制，并且其中每个桥接节点也是区块链网络中的相应节点；所述方法由注册机构执行，包括：

[0221] 生成第一区块链事务，其中所述第一区块链事务包括输入和第一输出，所述输入包括链接至所述注册机构的第一公钥的签名，所述第一输出包括第一证书，其中所述第一证书包括分配给所述请求者的标识符；以及

[0222] 将所述第一区块链事务传输至所述区块链网络，以将其包含在所述区块链中。

[0223] 语句2、如语句1所述的方法，其中所述第一事务包括第二输出，所述第二输出锁定至所述注册机构的第二公钥。

[0224] 语句3、如语句2所述的方法，其中所述第一输出通过时间锁锁定至所述注册机构的所述第二公钥，其中，时间锁防止所述第一输出在预定时间段结束前解锁。

[0225] 语句4、如语句2或3所述的方法，其中所述第一输出至少锁定至所述注册机构的所述第二公钥和一不同的公钥。

[0226] 不同的公钥可以是网络中节点的公钥，也可以是网络外第三方的公钥。这意味着证书撤销需要多重签名。

[0227] 语句5、如语句1至4中任一项所述的方法，所述方法包括：将所述第一区块链事务的事务标识符传输给所述权限请求者。

[0228] 语句6、如语句1至5中任一项所述的方法，其中所述证书使用加密密钥进行加密，所述加密密钥由所述注册机构生成。

[0229] 例如，该加密密钥可以是由该注册机构生成的随机数。

[0230] 语句7、如语句1至6中任一项所述的方法，所述方法包括：

[0231] 接收所述请求者提出的加入所述网络的请求，其中所述请求包括一个或更多个凭证；

[0232] 根据所述一个或更多个凭证核实所述请求，其中所述生成所述第一区块链事务是以所述请求有效为条件的。

[0233] 例如，该一个或更多个凭证可以包括该请求者的IP地址和/或制造信息。

[0234] 语句8、如语句1至7中任一项所述的方法，其中所述一组桥接节点包括主节点以及可由所述主节点控制的一组中间节点，其中所述注册机构是所述主节点。

[0235] 语句9、如语句1至8中任一项所述的方法，其中所述一组桥接节点包括所述主节点以及可由所述主节点控制的所述一组中间节点，其中所述请求者是所述主节点。

[0236] 语句10、如语句1至9中任一项所述的方法，其中所述请求者是所述区块链网络中的相应节点，其中所述证书包括分配给所述权限请求者的公钥。

[0237] 语句11、如语句1至8中任一项所述的方法，其中所述请求者是可由所述第一网络中的一个或更多个桥接节点控制的设备，并且其中所述方法包括：

[0238] 向所述请求者传输一组证书，所述一组证书中的每个证书已经传输至所述一组节点中的相应节点。

[0239] 语句12、如语句1至11中任一项所述的方法，所述方法包括：将所述第一证书传输至所述一组桥接节点中的一个或更多个。

[0240] 语句13、如语句2或其任何从属语句所述的方法，所述方法包括：

[0241] 生成第二区块链事务，其中所述第二区块链事务包括引用所述第一事务的所述第二输出的输入，还包括链接至所述注册机构的所述第二公钥的签名；

[0242] 将所述第二区块链事务传输至所述区块链网络，以将其包含在所述区块链中。

[0243] 语句14、一种用于请求加入第一网络的权限的计算机实现的方法，其中所述第一网络包括一组桥接节点和一组设备，所述一组设备可由所述一组桥接节点中的一个或更多个桥接节点控制，并且其中每个桥接节点也是区块链网络中的相应节点；所述方法由请求者执行，包括：

[0244] 向注册机构发送加入所述第一网络的请求；

[0245] 获取第一证书，所述证书由所述注册机构颁发并且包括分配给所述请求者的标识符。

[0246] 语句15、如语句14所述的方法，其中所述获取包括：

[0247] 接收包括所述第一证书的第一区块链事务的事务标识符；

[0248] 使用所述事务标识符从所述区块链中获取所述第一区块链事务。

[0249] 语句16、如语句15所述的方法，其中所述第一区块链事务包括第一输入和第二输出，所述第一输入包括所述证书，所述第二输出链接至所述注册机构的公钥，其中所述方法包括：

[0250] 标识所述注册机构的所述公钥；

[0251] 从所述注册机构的所述公钥中标识包含在传输至所述区块链的一个或更多个相应事务中的一个或更多个第二证书，每个第二证书被颁发给相应的桥接节点或设备或所述网络。

[0252] 语句17、如语句16所述的方法，其中所述第一证书包括所述请求者的公钥，其中颁发给所述第一网络中的所述一组桥接节点中的相应桥接节点的每个第二证书包括所述节点的相应公钥，其中所述方法包括：

[0253] 将第三区块链事务传输至所述一组桥接节点中的至少一个，其中所述第三区块链事务包括锁定至所述至少一个桥接节点的所述相应公钥的输出。

[0254] 语句18、如语句14至17中任一项所述的方法，所述获取第一证书包括：从所述注册机构接收所述第一证书。

[0255] 语句19、如语句14至18中任一项所述的方法，所述方法包括：

[0256] 从所述注册机构接收一个或更多个第二证书，每个第二证书被颁发给所述第一网络中的所述一组桥接节点或设备中的相应桥接节点或设备。

[0257] 语句20、如语句19所述的方法，其中所述第一证书包括所述请求者的网络地址，其中颁发给所述第一网络中的所述相应桥接节点的每个第二证书包括所述节点的相应网络地址，其中所述方法包括：

[0258] 向所述一组桥接节点中的一个或更多个发送消息，所述消息从所述请求者的所述网络地址发送至所述消息所发送到的所述一个或更多个桥接节点的相应网络地址。

[0259] 该网络地址可以是IP地址。

[0260] 语句21、如语句14或语句18至20中任一项所述的方法，其中所述请求者是所述第一网络中的所述一组设备中的一个。

[0261] 语句22、如语句14至21中任一项所述的方法，其中所述请求者是所述第一网络中的所述一组节点中的一个。

[0262] 语句23、如语句22所述的方法，其中所述一组桥接节点包括主节点以及可由所述主节点控制的一个或更多个中间节点，其中所述请求者是所述主节点。

[0263] 语句24、如语句14至23中任一项所述的方法，其中所述请求包括所述请求者的一个或更多个凭证。

[0264] 语句25、如语句24所述的方法，其中所述一个或更多个凭证包括所述请求者的IP地址。

[0265] 语句26、一种计算机设备，所述计算机设备包括：

[0266] 存储器，所述存储器包括一个或更多个存储器单元；

[0267] 处理装置，所述处理装置包括一个或更多个处理单元，其中所述存储器存储被设置在所述处理装置上运行的代码，所述代码被配置为当在所述处理装置上运行时，执行如语句1至13中任一项所述的方法。

[0268] 语句27、一种在计算机可读存储器上实现的计算机程序，所述计算机程序被配置为当在如语句26所述的计算机设备上运行时，执行如语句1至13中任一项所述的方法。

[0269] 语句28、一种计算机设备，所述计算机设备包括：

[0270] 存储器，所述存储器包括一个或更多个存储器单元；

[0271] 处理装置，所述处理装置包括一个或更多个处理单元，其中所述存储器存储被设置在所述处理装置上运行的代码，所述代码被配置为当在所述处理装置上运行时，执行如语句14至25中任一项所述的方法。

[0272] 语句29、一种在计算机可读存储器上实现的计算机程序，所述计算机程序被配置为当在如语句28所述的计算机设备上运行时，执行如语句14至25中任一项所述的方法。

[0273] 根据本文公开的教导的另一个方面，可提供一种方法，所述方法包括所述注册机构和所述权限请求者的操作。

[0274] 根据本文公开的教导的另一个方面，可提供一种系统，所述系统包括所述注册机构和所述权限请求者的计算机设备。

[0275] 根据本文公开的教导的另一个方面，可提供一组事务，所述一组事务包括第一区块链事务和/或第二区块链事务。

[0276] 一旦给出本文的公开内容，所公开技术的其他变体或用例对于本领域技术人员可能变得显而易见。本公开的范围不受所描述的实施例限制，而仅受随附语句限制。

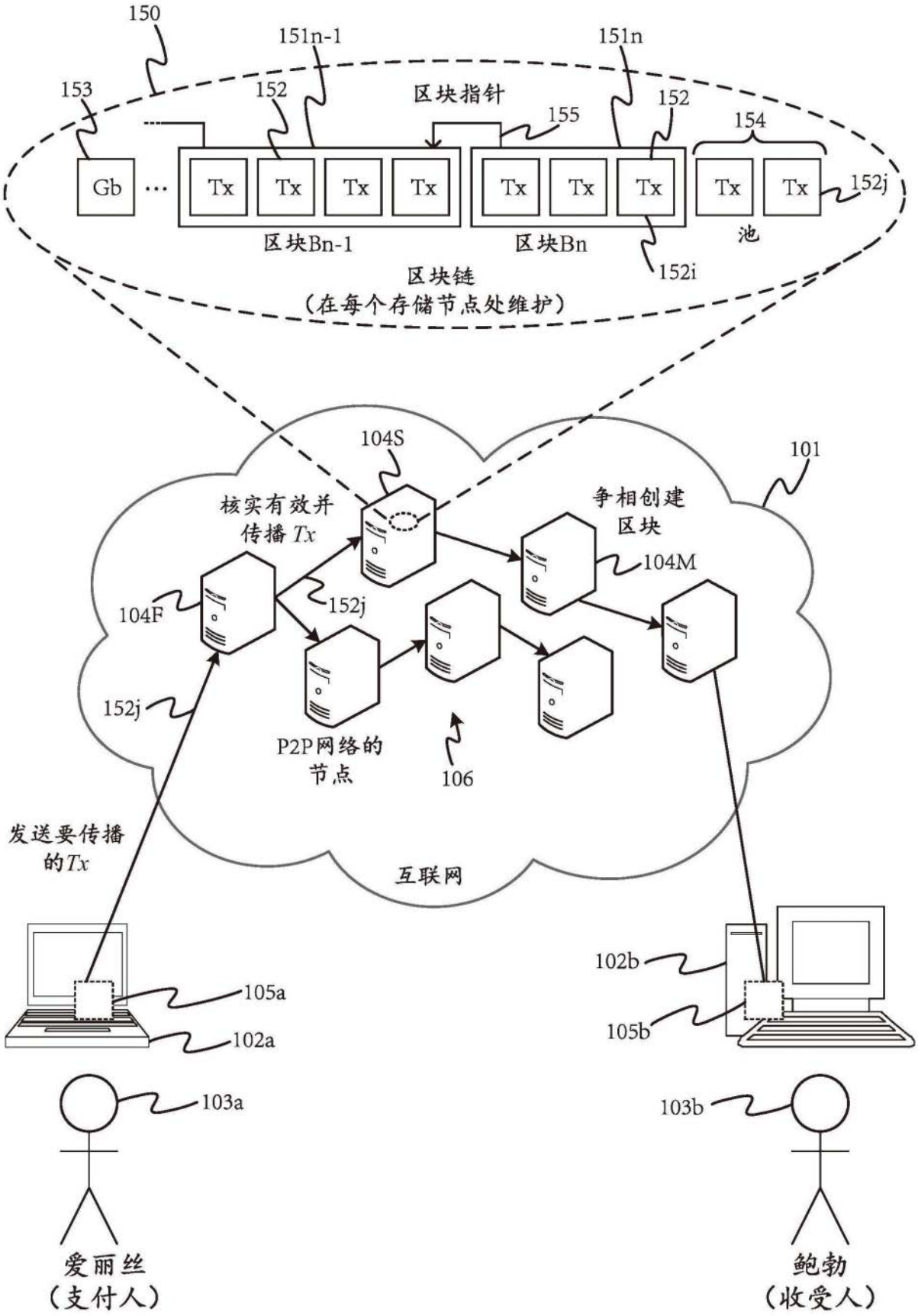

150
151n-1
151n
153
152
区块指针
155
152
154
Gb
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
152j
区块Bn-1
区块Bn
池
152i
区块链
（在每个存储节点处维护）
104S
101
核实有效并
传播 Tx
争相创建
区块
104M
104F
152j
152j
P2P网络的
节点
106
发送要传播
的Tx
互联网
105a
102a
102b
105b
103a
103b
爱丽丝
（支付人）
鲍勃
（收受人）

图1

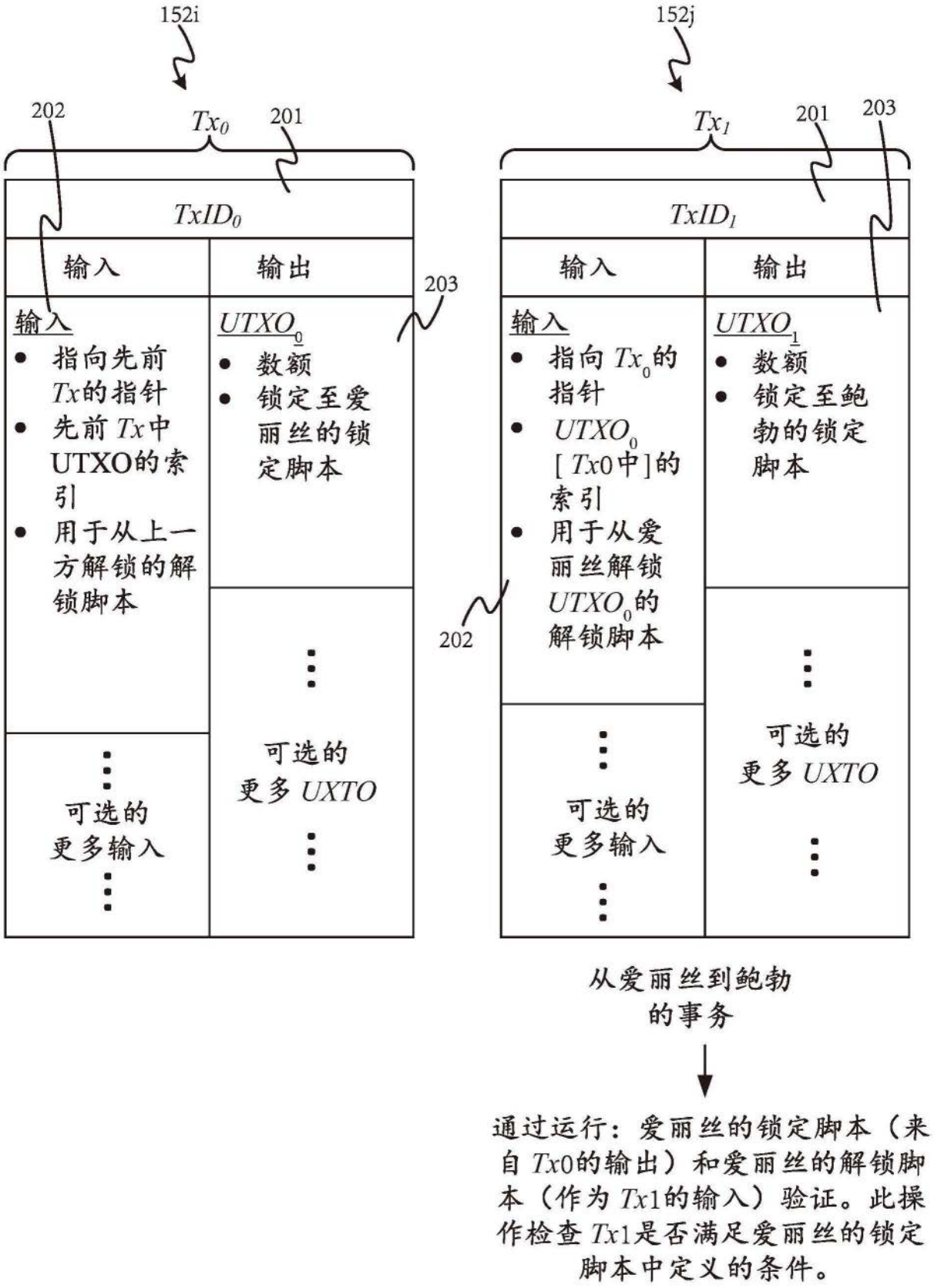

152i
202
Tx_0
201
$TxID_0$
输入
输出
203
输入
• 指向先前Tx的指针
• 先前Tx中UTXO的索引
• 用于从上一方解锁的解锁脚本
$UTXO_0$
• 数额
• 锁定至爱丽丝的锁定脚本
可选的更多输入
可选的更多 UXTO
152j
Tx_1
201
203
$TxID_1$
输入
输出
202
输入
• 指向 Tx_0 的指针
• $UTXO_0$ [Tx0中]的索引
• 用于从爱丽丝解锁 $UTXO_0$ 的解锁脚本
$UTXO_1$
• 数额
• 锁定至鲍勃的锁定脚本
可选的更多输入
可选的更多 UXTO
从爱丽丝到鲍勃的事务
通过运行：爱丽丝的锁定脚本（来自Tx0的输出）和爱丽丝的解锁脚本（作为Tx1的输入）验证。此操作检查Tx1是否满足爱丽丝的锁定脚本中定义的条件。

图2

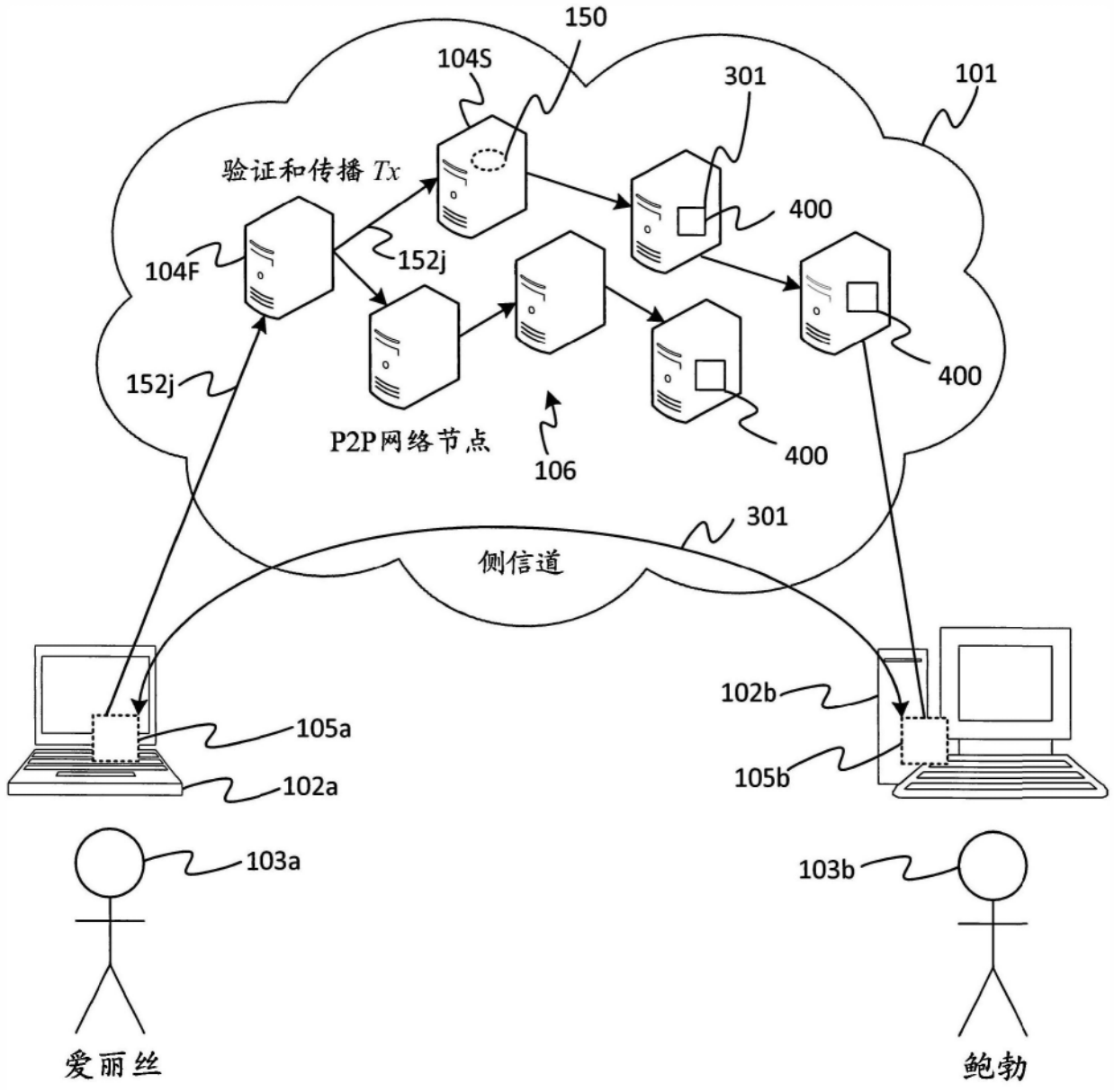

150
104S
301
101
验证和传播 Tx
400
104F
152j
152j
400
P2P网络节点
106
400
301
侧信道
102b
105a
105b
102a
103a
103b
爱丽丝
鲍勃

图3

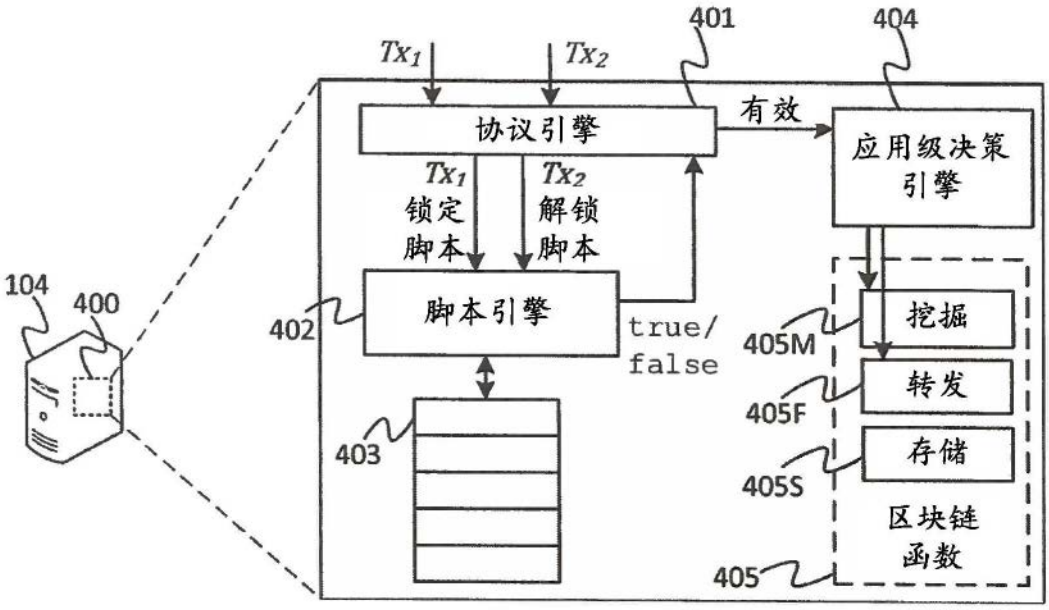

401
404
Tx_1
Tx_2
协议引擎
有效
应用级决策引擎
Tx_1 锁定脚本
Tx_2 解锁脚本
104
400
402
脚本引擎
true/false
405M
挖掘
转发
405F
存储
405S
403
区块链函数
405

图4

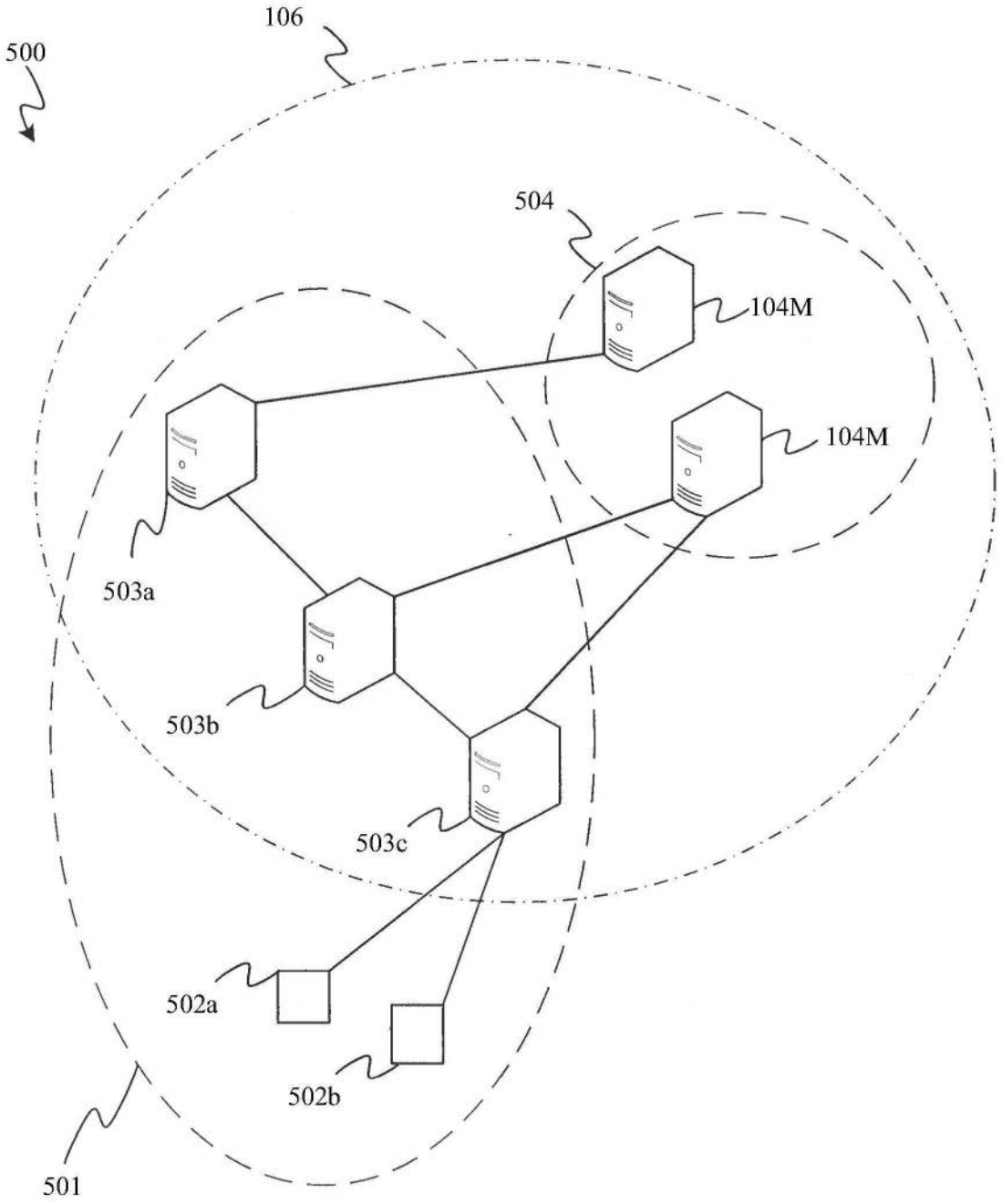


图5

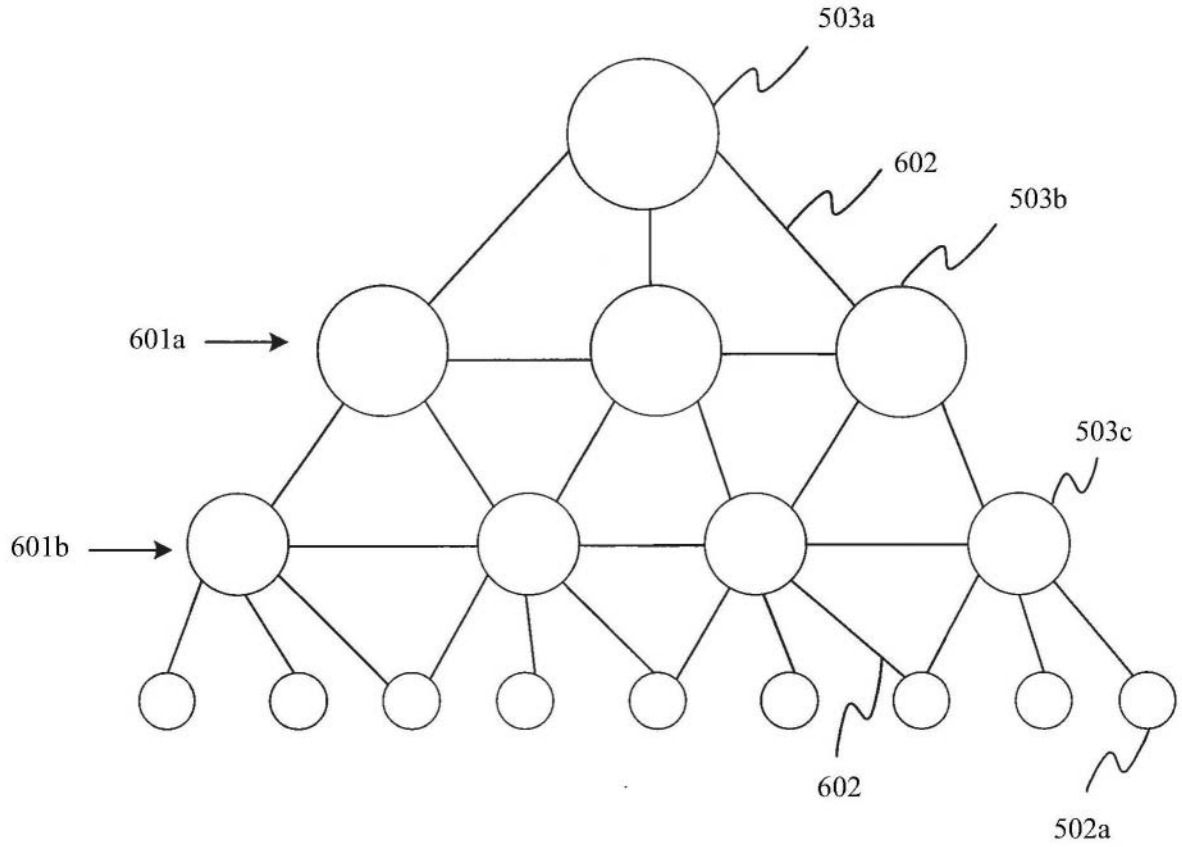


图6

Tx_1（部分）			
输入		输出	
值		值	
x	$<Sig_{PK0}>$ $<PK_0>$	0	OP_FALSE OP_RETURN 0x4d494f54<命令数据>

701a 702a

图7a

Tx_1（完整）			
输入		输出	
值		值	
x	$<Sig_{PK0}> <PK_0>$	0	OP_FALSE OP_RETURN 0x4d494f54<命令数据>
>0	$<Sig_{PK1}> <PK_1>$	x	OP_DUP OP_HASH160 $<H_{160}(PK_1)>$ OP_EQUALVERIFY OP_CHECKSIG

701b

702b

图7b

Tx_1（部分）			
输入		输出	
值		值	
x	$<Sig_{PK0}> <PK_0>$	0	OP_FALSE OP_RETURN 0x4d494f54<命令数据>
		$x+\delta$	OP_DUP OP_HASH160 $<H_{160}(PK_1)>$ OP_EQUALVERIFY OP_CHECKSIG

801a

802b

802a

图8a

Tx_1（完整）			
输入		输出	
值		值	
x	$<Sig_{PK0}><PK_0>$	0	OP_FALSE OP_RETURN 0x4d494f54<命令数据>
$>\delta$	$<Sig_{PK1}><PK_1>$	x	OP_DUP OP_HASH160 $<H_{160}(PK_1)>$ OP_EQUALVERIFY OP_CHECKSIG

801b

图8b

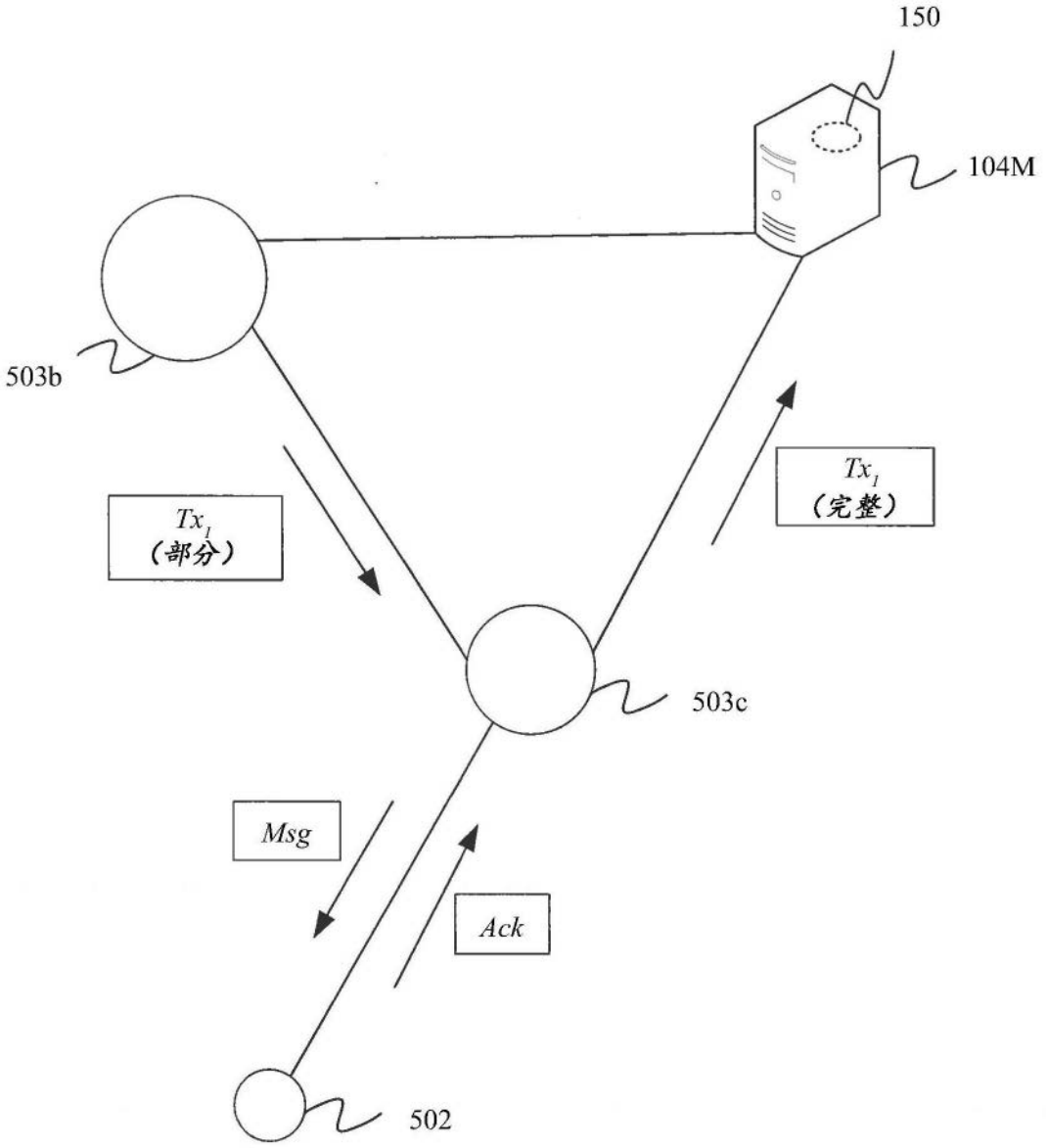


图9

Tx_1（部分）			
输入		输出	
值		值	
x	$<Sig_{PKserv}><PK_{serv}>$	0	OP_FALSE OP_RETURN 0x4d494f54<命令数据>
		$x+\delta$	OP_DUP OP_HASH160 $<H_{160}(PK_{slave})>$ OP_EQUALVERIFY OP_CHECKSIG

图10a

Tx_1（完整）			
输入		输出	
值		值	
x	$<Sig_{PKserv}><PK_{serv}>$	0	OP_FALSE OP_RETURN 0x4d494f54<命令数据>
$>\delta$	$<Sig_{PKslave}><PK_{slave}>$	x	OP_DUP OP_HASH160 $<H_{160}(PK_{slave})>$ OP_EQUALVERIFY OP_CHECKSIG

图10b

Tx_1（请求）			
输入		输出	
值		值	
x_1	$<Sig_{PK0}><PK_0>$	0	OP_FALSE OP_RETURN 0x4d494f54<命令数据>
		y_1	OP_2 OP$<PK_1><PK_2>$ OP_CHECKMULTISIG

1101a 1102b 1102a

图11a

Tx_2（批准）			
输入		输出	
值		值	
x_2	$<Sig_{PK1}><PK_1>$ $<Sig_{PK2}><PK_2>$	0	OP_FALSE OP_RETURN 0x4d494f54<命令数据>
		y_2	OP_DUP OP_HASH160 $<H_{160}(PK_3)>$ OP_EQUALVERIFY OP_CHECKSIG

1101b 1102c

图11b

Tx_1（部分）			
输入		输出	
值		值	
x	$<Sig_{PKserv\_0}><PK_{serv\_0}>$	0	OP_FALSE OP_RETURN 0x4d494f54 OP_PUSHDATA1 [有效负载长度]<BIE1加密的命令数据> OP_PUSHDATA1 [有效负载长度] <BIE1加密的加密密钥>
		$x+\delta$	OP_DUP OP_HASH160 $<H_{160}(PK_{slave\_0})>$ OP_EQUALVERIFY OP_CHECKSIG

图12a

Tx1（完整）			
输入		输出	
值		值	
x	$<Sig_{PKserv\_0}><PK_{serv\_0}>$	0	OP_FALSE OP_RETURN 0x4d494f54 OP_PUSHDATA1 [有效负载长度]<BIE1加密的命令数据> OP_PUSHDATA1 [有效负载长度] <BIE1加密的加密密钥>
$>\delta$	$<Sig_{PKslave\_1}><PK_{slave\_1}>$	$x+\delta$	OP_DUP OP_HASH160 $<H_{160}(PK_{slave\_0})>$ OP_EQUALVERIFY OP_CHECKSIG

图12b

输出	
值	
0	OP_FALSE OP_RETURN OP_PUSHDATA1 0x4d494f54-*物联网协议标识符* 0x01-*有效负载类型* 0x00000001-*物联网软件版本号* 0x3dd5dfac...32-*设备ID* 0x234a3789...22-*设备公钥* 0x4d348912...87-*设备证书位置数据* 0x3ad21fac-*命令* 0x5665b456-*状态* 0x5665b456-*先前状态*

图13

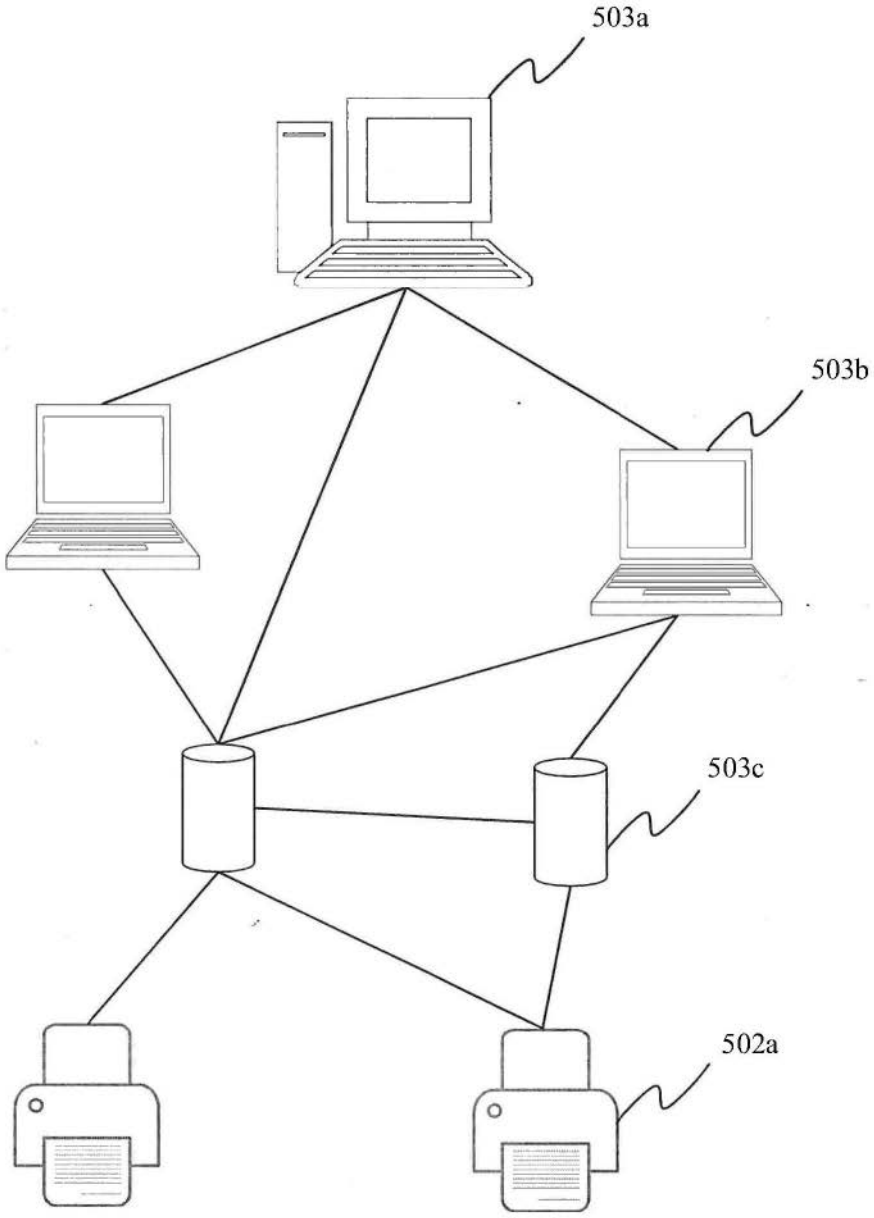


图14

Tx_1（证书）			
输入		输出	
值		值	
x_1	$<Sig_{PKadmin}><PK_{admin}>$	0	OP_FALSE OP_RETURN 0x4d494f54<证书数据>
		y_1	OP_DUP OP_HASH160 $<H_{160}(PK_{admin})>$ OP_EQUALVERIFY OP_CHECKSIG

图15a

Tx_1（部分）			
输入		输出	
值		值	
x_2	$<Sig_{PKlaptop\_0}><PK_{laptop\_0}>$	0	OP_FALSE OP_RETURN 0x4d494f54<加密命令数据>
		y_2	OP_DUP OP_HASH160 $<H_{160}(PK_{controller\_0})>$ OP_EQUALVERIFY OP_CHECKSIG

图15b

Tx1（控制器）			
输入		输出	
值		值	
x_2	$<Sig_{PKlaptop\_0}> <PK_{laptop\_0}>$	0	OP_FALSE OP_RETURN 0x4d494f54<加密命令数据>
x_3	$<Sig_{PKcontroller\_0}>$ $<PK_{controller\_0}>$	*y2*	OP_DUP OP_HASH160 $<H_{160}(PK_{controller\_0})>$ OP_EQUALVERIFY OP_CHECKSIG

图15c

Tx_1（证书）			
输入		输出	
值		值	
x_1	$<Sig_{PKreg}> <PK_{reg}>$	0	OP_FALSE OP_RETURN 0x4d494f54<证书数据>
		y_1	OP_DUP OP_HASH160 $<H_{160}(PK_{reg})>$ OP_EQUALVERIFY OP_CHECKSIG

1501a 1502b 1502a

图16a

输出	
值	
0	OP_FALSE OP_RETURN OP_PUSHDATA1 〈有效负载长度〉 0x4d494f54-物联网协议标识符 0x02-有效负载类型 0x00000001-物联网软件版本号 0x3dd5dfac...32-新设备ID 0x234a3789...22-新设备压缩公钥 0x3ad21fac-设备类型 0x00000004-设备物联网节点类型 0x0000...06f-IPv6地址和端口号 0x5664b456-UNIX时间创建日期 0x5665b456-UNIX时间证书到期日 0x76d5335c.....45-附加信息

图16b