(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110546636 B
(45)授权公告日 2023.08.08

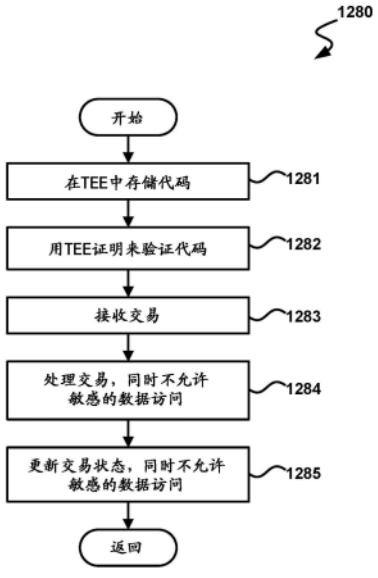
(21)申请号 201880026973.9
(22)申请日 2018.04.07
(65)同一申请的已公布的文献号
申请公布号 CN 110546636 A
(43)申请公布日 2019.12.06
(30)优先权数据
62/489,907 2017.04.25 US
15/715,620 2017.09.26 US
(85)PCT国际申请进入国家阶段日
2019.10.23
(86)PCT国际申请的申请数据
PCT/US2018/026622 2018.04.07
(87)PCT国际申请的公布数据
WO2018/200166 EN 2018.11.01
(73)专利权人 微软技术许可有限责任公司
地址 美国华盛顿州
(72)发明人 I·沃登
(74)专利代理机构 北京世辉律师事务所 16093
专利代理师 王俊
(51)Int.Cl.
G06F 21/12(2013.01)
G06F 21/62(2013.01)
H04L 9/40(2022.01)
H04L 9/32(2006.01)
G06F 21/57(2013.01)
(56)对比文件
US 2016275461 A1,2016.09.22
US 2017011460 A1,2017.01.12
CN 106559211 A,2017.04.05
US 2016261685 A1,2016.09.08
CN 106372868 A,2017.02.01
US 2016379212 A1,2016.12.29
CN 106295401 A,2017.01.04
CN 106296359 A,2017.01.04
审查员 罗思异
权利要求书3页 说明书24页 附图12页

(54)发明名称

联盟区块链网络中的机密性

(57)摘要

所公开的技术通常针对区块链技术。在该技术的一个示例中,预定类型的区块链协议代码被存储在处理器的可信执行环境(TEE)中。TEE证明用于验证在TEE中存储的区块链协议代码是预定类型的区块链协议代码。区块链交易被接收。处理区块链交易,同时不允许访问原始交易数据。基于区块链交易的处理为区块链网络更新已处理区块链的状态,同时不允许访问原始交易数据。



CN 110546636 B

1.一种装置,包括:

设备,包括至少一个存储器和至少一个处理器,所述至少一个存储器适于存储针对所述设备的运行时数据,并且所述至少一个处理器适于执行处理器可执行代码,所述处理器可执行代码响应于执行而使得所述设备能够执行动作,所述动作包括:

在所述至少一个处理器中的处理器的第一可信执行环境TEE中存储预定类型的区块链协议代码;

由所述第一TEE使用TEE证明来验证在所述第一TEE中所存储的所述区块链协议代码是所述预定类型的区块链协议代码;

接收区块链交易;

处理所述区块链交易,同时不允许访问原始交易数据;以及

对于区块链网络,基于所述区块链交易的所述处理来更新经处理的区块链的状态,同时不允许访问原始交易数据和与基于所述区块链交易的所述处理而更新的经处理的所述区块链的所述状态相关联的状态信息,使得所述交易数据的解密不被允许,除非所述解密正在如下设备的TEE中发生,所述设备运行按照基于TEE证明所验证的所述预定类型的区块链协议代码。

2.根据权利要求1所述的装置,其中所述区块链交易被加密并且是机密的使得所述区块链交易的查看被限于授权方。

3.根据权利要求1所述的装置,所述动作还包括:

向与所述区块链交易相关联的用户提供所述区块链交易的同步通知。

4.根据权利要求1所述的装置,所述动作还包括:

接收针对智能合约的代码;

确定所述代码是否符合机密性设计模式;以及

如果所述代码不符合所述机密性设计模式

拒绝所述代码

否则

部署所述代码。

5.根据权利要求4所述的装置,其中针对所述智能合约的所述代码是针对所述智能合约的源代码。

6.根据权利要求4所述的装置,其中确定所述代码是否符合所述机密性设计模式包括静态分析。

7.根据权利要求4所述的装置,其中确定所述代码是否符合所述机密性设计模式经由至少正则表达式而被完成。

8.根据权利要求4所述的装置,其中所述机密性设计模式要求访问控制被内置于所述智能合约中。

9.根据权利要求4所述的装置,其中所述机密性设计模式要求所述区块链网络的事件关于所述事件可用于哪些地址被限制。

10.根据权利要求4所述的装置,其中所述机密性设计模式拒绝除了由所述区块链网络的定义的功能之外的对所述区块链网络的变量的访问。

11.一种方法,包括:

由处理器的第一TEE使用可信执行环境TEE证明来验证在所述处理器的所述第一TEE中存储的安全协议代码是预定类型的安全协议代码；

处理针对联盟网络的区块链交易，同时除了经由智能合约之外不允许访问状态信息；以及

至少使用所述处理器，针对所述联盟网络基于所述区块链交易的所述处理来更新所述区块链交易的状态，同时除了经由智能合约和同步通知之外不允许访问基于所述区块链交易的所述处理的所述状态信息，使得所述区块链交易的解密不被允许，除非所述解密正在如下设备的TEE中发生，所述设备运行按照基于TEE证明所验证的所述预定类型的安全协议代码。

12.根据权利要求11所述的方法，还包括：

向与所述区块链交易相关联的用户提供所述区块链交易的同步通知。

13.根据权利要求11所述的方法，还包括：

接收针对智能合约的源代码；

确定所述源代码是否符合机密性设计模式；以及

如果所述源代码不符合所述机密性设计模式，

拒绝所述源代码

否则

编译和部署所述源代码。

14.根据权利要求13所述的方法，其中确定所述源代码是否符合所述机密性设计模式包括静态分析。

15.根据权利要求13所述的方法，其中所述机密性设计模式要求访问控制被内置到所述智能合约中。

16.根据权利要求13所述的方法，其中所述机密性设计模式要求所述区块链网络的事件关于所述事件可用于哪些地址被限制。

17.根据权利要求13所述的方法，其中所述机密性设计模式拒绝除了由所述区块链网络的定义的功能之外的对所述区块链网络的变量的访问。

18.一种处理器可读存储介质，具有存储在其上的用于计算机网络设计的处理器可执行代码，所述处理器可执行代码在由至少一个处理器执行时能够实现动作，所述动作包括：

在所述至少一个处理器中的处理器的第一可信执行环境TEE中存储预定类型的区块链协议代码；

由所述第一TEE经由TEE证明，验证在所述第一TEE中存储的所述区块链协议代码是所述预定类型的区块链协议代码；以及

处理区块链交易，同时不允许访问原始交易数据；

对于区块链网络，基于所述区块链交易的所述处理来更新经处理的区块链的状态，同时不允许访问原始交易数据和与基于所述区块链交易的所述处理而更新的经处理的所述区块链的所述状态相关联的状态信息，使得所述交易数据的解密不被允许，除非所述解密正在如下设备的TEE中发生，所述设备运行如基于TEE证明所验证的所述预定类型的区块链协议代码。

19.根据权利要求18所述的处理器可读存储介质，所述动作还包括：

接收针对智能合约的源代码；

确定所述代码是否符合机密性设计模式;以及

如果所述源代码不符合所述机密性设计模式，

拒绝所述源代码

否则

编译和部署所述源代码。

20.根据权利要求19所述的处理器可读存储介质,其中所述机密性设计模式要求访问控制被内置在所述智能合约中。

联盟区块链网络中的机密性

背景技术

[0001] 已经提出了用于各种应用场景的区块链系统，包括金融行业，医疗保健，物联网等中的应用。例如开发了一些系统，允许电子现金无需经过金融机构即可直接从一方转移到另一方。新交易可以被生成并被添加到块中的交易堆栈中。包括新所有者的公钥的新交易可以由所有者用所有者的私钥进行数字签名，以将所有权转移给新所有者，如新所有者的公钥所表示。

[0002] 一旦块被填满，就可以用块头部“封闭”该块，该块头部是该块内所有交易标识符的哈希摘要。块头部可以被记录为链中下一个区块中的第一交易，从而创建称为“区块链”的数学层次结构。为了验证当前所有者，可以遵循交易的区块链来验证从第一笔交易到最后一笔交易的每一笔交易。新所有者仅需要具有与交易的公钥匹配的私钥。区块链可以在由安全身份(例如公钥)表示的实体中创建所有权的数学证明，其是伪匿名的。

发明内容

[0003] 提供本发明内容以简化形式介绍概念的选择，这些概念将在下面的具体实施方式中进一步描述。本发明内容既不旨在标识所要求保护的主题的关键特征或必要特征，也不旨在用于限制所要求保护的主题的范围。

[0004] 简要地说，所公开的技术通常针对区块链技术。在该技术的一个示例中，预定类型的区块链协议代码被存储在处理器的可信执行环境(TEE)中。在一些示例中，TEE证明被用于验证在TEE中存储的区块链协议代码是预定类型的区块链协议代码。区块链交易可以被接收。在一些示例中，处理区块链交易，同时不允许访问原始交易数据。在一些示例中，基于区块链交易的处理为区块链网络更新已处理区块链的状态，同时不允许访问原始交易数据。

[0005] 在阅读和理解附图和描述之后，将理解所公开技术的其他方面和应用。

附图说明

[0006] 参考以下附图描述本公开的非限制性和非穷举性示例。在附图中，除非另外指明，否则贯穿各个附图，类似的附图标记表示类似的部分。这些附图不一定按比例绘制。

[0007] 为了更好地理解本公开，将参考下面的具体实施方式，该具体实施方式将结合附图进行阅读，在附图中：

[0008] 图1是示出了可以在其中采用本技术的各方面的合适环境的一个示例的框图；

[0009] 图2是示出了根据所公开的技术的各方面的合适的计算设备的一个示例的框图；

[0010] 图3是示出了系统的示例的框图；

[0011] 图4是示出了验证节点的示例的框图；

[0012] 图5A-5C是示出了用于建立区块链网络的过程的示例数据流的图；

[0013] 图6A-6B是示出了用于区块链网络的交易处理的过程的示例数据流的图；

[0014] 图7是示出了用于区块链网络的交易处理的另一示例过程的图；

[0015] 图8是示出了包括具有三个成员的机密联盟区块链框架(COCO)网络的系统的示例的框图；

[0016] 图9是示出了包括COCO网络的一部分的系统的示例的框图，该COCO网络的一部分包括经历验证节点证明和密钥交换的两个验证节点；

[0017] 图10是图示了用于将新成员添加到COCO网络的示例步骤的图；

[0018] 图11是示出了用于区块链系统的过程的示例数据流的图；以及

[0019] 图12是示出了根据本公开的方面的用于针对区块链网络的具有机密性的交易处理的示例过程的图。

具体实施方式

[0020] 以下描述提供了特定细节，以用于对技术的各种示例的透彻理解和实现描述。本领域技术人员将理解，可以在没有许多这些细节的情况下实践该技术。在某些情况下，没有示出或详细描述众所周知的结构和功能，以避免不必要地使技术示例的描述不清楚。意图以本公开中使用的术语以其最广泛的合理方式来解释，即使其与该技术的某些示例的详细描述结合使用。尽管以下可能会强调某些术语，但旨在以任何受限方式解释的任何术语将如在本具体实施方式部分中的那样被公开且明确地定义。在整个说明书和权利要求书中，除非上下文另外指出，否则以下术语至少具有本文明确关联的含义。以下标识的含义不一定限制这些术语，而仅提供这些术语的示意性示例。例如术语“基于”和“根据”中的每一个都不是排他的，并且等效于术语“至少部分地基于”，并且包括基于附加因素的选项，其中一些本文不做描述。作为另一个示例，术语“经由”不是排他的，并且等效于术语“至少部分地经由”，包括经由附加因素的选项，其中一些可能在本文未描述。“在...中”的含义包括“在...中”和“在...上”。本文使用的短语“在一个实施例中”或“在一个示例中”不一定指代相同的实施例或示例，尽管本文使用的短语“在一个实施例中”或“在一个示例中”可以指代相同的实施例或示例。特定的文本数字标记的使用并不意味着值较小的数字标记的存在。例如引用“从由第三foo和第四bar构成的组中选择的小部件”本身并不意味着存在至少三个foo，也不意味着存在至少四个bar元素。单数形式的引用仅是为了阅读的清楚，并且包括复数引用，除非特别地排除了复数引用。除非另外明确指出，否则术语“或”是包含性的“或”运算符。例如短语“A或B”表示“A、B或A和B”。如本文所使用的，术语“组件”和“系统”旨在涵盖硬件、软件或硬件和软件的各种组合。因此，例如系统或组件可以是过程、在计算设备上执行的过程、计算设备或其一部分。

[0021] 简要地说，公开的技术通常针对区块链技术。在该技术的一个示例中，预定类型的区块链协议代码被存储在处理器的可信执行环境(TEE)中。在一些示例中，TEE证明被用于验证存储在TEE中的区块链协议代码是预定类型的区块链协议代码。可以接收区块链交易。在一些示例中，处理区块链交易，同时不允许访问原始交易数据。在一些示例中，基于区块链交易的处理为区块链网络更新已处理区块链的状态，同时不允许访问原始交易数据。

[0022] 在一些示例中，区块链网络的参与方可以将由区块链主密钥加密的交易提交给区块链网络。可以将交易转发到基于冲突解决协议、共识协议和/或类似物选择的区块链网络中的特定验证节点(VN)。然后，选择的VN可以通过执行交易代码来处理反应，然后可以解决任意冲突。该交易可以是简单交易、智能合约等。

[0023] 在某些示例中，VN根据机密联盟(COCO)区块链框架进行操作，该框架允许使用任意合适的区块链协议，并允许将任意合适的共识协议与COCO框架结合使用。在一些示例中，仅可以使用由成员同意的协议，但是成员可以在任意合适的协议上同意。在某些示例中，COCO框架本身并不定义区块链分类账和实际的区块链交易处理，而是允许使用任意合适的分类账和区块链协议，并能够为交易启用机密属性，管理信任假设，允许网络仅执行一次链码，允许链码是不确定的，并允许链码包括与外部系统的交互。

[0024] 虽然COCO框架可以用于提供某些机密性保证，并且可以与任意合适的区块链协议一起使用，但是诸如以太坊的合适的区块链协议的某些方面可能与COCO的机密性保证不兼容。本公开的一些示例可以使得诸如以太坊的协议能够与COCO框架一起使用，同时仍然支持COCO的机密性并且不需要对协议进行深度修改。

[0025] 在一些示例中，除非通过确保机密性保证被满足的指定手段，否则不允许对原始交易数据的访问，并且不允许对状态信息的访问。例如可以禁用允许访问敏感状态信息的协议的API。

[0026] 在一些示例中，向用户同步地通知用户的交易是否适当地被包括在系统中，而不是向用户提供用户可以用于异步查询交易状态的交易ID。相反，在这些示例中，不允许此类查询。

[0027] 在一些示例中，可以以以下这样的方式在智能合约本身中实现访问控制：只有适当的一方或多方可以查看敏感的状态信息，同时保持COCO机密性保证。在某些示例中，直到已经验证了智能合约源代码以符合机密性设计模式，以便于确保COCO机密性保证被满足，才会部署智能合约。

[0028] 示意性设备/操作环境

[0029] 图1是其中可以实践本技术的各方面的环境100的图。如图所示，环境100包括计算设备110以及经由网络130连接的网络节点120。即使在图1中示出了环境100的特定组件，在其他示例中，环境100也可以包括附加的和/或不同的组件。例如在某些示例中，环境100还可以包括网络存储设备，维护管理器和/或其他合适的组件(未示出)。图1中所示的计算设备110可以在各种位置，包括本地、在云中等。例如计算机设备110可以在客户端侧、在服务器侧等。

[0030] 如图1所示，网络130可以包括一个或多个网络节点120，一个或多个网络节点120互连多个计算设备110，并将计算设备110连接到外部网络140，例如因特网或内联网。例如网络节点120可以包括交换机、路由器、集线器、网络控制器或其他网络元件。在某些示例中，计算设备110可以被组织成机架、动作区域、组、集合或其他合适的分区。例如在所示的示例中，计算设备110被分组为单独标识为第一、第二和第三主机集112a-112c的三个主机集。在所示示例中，主机集112a-112c中的每一个分别可操作地耦合到对应的网络节点120a-120c，其通常被称为“机架顶部”或“TOR”网络节点。TOR网络节点120a-120c然后可以可操作地耦合到附加网络节点120，以形成分层、平面、网状或其他合适类型的拓扑中的计算机网络，该计算机网络允许计算设备110与外部网络140之间的通信。在其他示例中，多个主机集112a-112c可以共享单个网络节点120。计算设备110实际上可以是任意类型的通用或专用计算设备。例如这些计算设备可以是诸如台式计算机、膝上型计算机、平板计算机、显示设备、照相机、打印机或智能电话的用户设备。然而，在数据中心环境中，这些计算设备

可以是服务器设备，诸如应用服务器计算机、虚拟计算主机计算机或文件服务器计算机。此外，计算设备110可以被单独地配置为提供计算，存储和/或其他合适的计算服务。

[0031] 示意性计算设备

[0032] 图2是说明其中可实践本技术的各方面的计算设备200的一个实例的图。计算设备200实际上可以是任意类型的通用或专用计算设备。例如计算设备200可以是诸如台式计算机、膝上型计算机、平板计算机、显示设备、照相机、打印机、嵌入式设备、可编程逻辑控制器(PLC)或智能手机的用户设备。同样地，计算设备200也可以是服务器设备，诸如应用服务器计算机、虚拟计算主机计算机或文件服务器计算机，例如计算设备200可以是图1的计算设备110或网络节点120的示例。计算设备200也可以是连接到网络以接收IoT服务的IoT设备。同样，计算机设备200可以是在各个附图中示出或引用的任意设备、节点、成员或其他实体的示例，如下文更详细地讨论。如图2所示，计算设备200包括处理电路210、操作存储器220、存储器控制器230、数据存储存储器250、输入接口260、输出接口270和网络适配器280。计算设备200的这些先前列出的组件中的每一个包括至少一个硬件元件。

[0033] 计算设备200包括至少一个处理电路210，该至少一个处理电路210被配置为执行指令，诸如用于实现本文描述的工作负载、过程或技术的指令。处理电路210可以包括微处理器、微控制器、图形处理器、协处理器、现场可编程门阵列、可编程逻辑器件、信号处理器或适合于处理数据的任意其他电路。前述指令以及其他数据(例如数据集、元数据、操作系统指令等)可以在计算设备200的运行时间期间存储在操作存储器220中。操作存储器220还可以包括多种数据存储设备/组件中的任意一种，诸如易失性存储器、半易失性存储器、随机存取存储器、静态存储器、高速缓存、缓冲器或用于存储运行时信息的其他介质。在一示例中，当计算设备200断电时，操作存储器220不保留信息。相反，计算设备200可以被配置为作为引导或其他加载过程的一部分，将指令从非易失性数据存储组件(例如数据存储组件250)转移到操作存储器220。

[0034] 操作存储器220可以包括第四代双倍数据速率(DDR4)存储器、第三代双倍数据速率(DDR3)存储器、其他动态随机存取存储器(DRAM)、高带宽存储器(HBM)、混合存储器多维数据集存储器、3D-堆叠存储器、静态随机存取存储器(SRAM)或其他存储器，并且这种存储器可以包括集成在DIMM、SIMM、SODIMM或其他封装上的一个或多个存储器电路。可以根据通道、排(“rank”)和存储体(“bank”)来组织这样的操作存储模块或设备。例如操作存储器设备可以经由存储器控制器230在通道中耦合到处理电路210。计算设备200的一个示例可包括每个通道一个或两个DIMM，每个通道具有一个或两个排。排内的操作存储器可以使用共享时钟、共享地址和命令总线进行操作。此外，操作存储设备可以被组织为几个存储体，其中存储体可以被认为是由行和列寻址的阵列。基于操作存储器的这种组织，操作存储器内的物理地址可以由通道、排、存储体、行和列的元组来引用。

[0035] 尽管进行了上述讨论，但是操作存储器220特别地不包括或不涵盖通信介质，任意通信介质或任意信号本身。

[0036] 存储器控制器230被配置为将处理电路210对接到操作存储器220。例如存储器控制器230可以被配置为在操作存储器220和处理电路210之间对接命令、地址和数据。存储器控制器230也可以是配置为从处理电路210或为处理电路210抽象化或以其他方式管理存储器管理的某些方面。尽管将存储器控制器230示为与处理电路210分离的单个存储器控制

器,但在其他示例中,可以采用多个存储器控制器,一个或多个存储器控制器可以与操作存储器220集成等。此外,可以将存储器控制器集成到处理电路210中。这些和其他变体是可能的。

[0037] 在计算设备200中,数据存储存储器250、输入接口260、输出接口270和网络适配器280通过总线240对接到处理电路210。尽管图2将总线240示为单个无源总线,但其他配置诸如总线的集合、点对点链接的集合、输入/输出控制器、桥、其他接口电路或其任意集合等也可以适当地用于将数据存储存储器250、输入接口260、输出接口270或网络适配器280对接到处理电路210。

[0038] 在计算设备200中,数据存储存储器250用于长期非易失性数据存储。数据存储存储器250可以包括各种非易失性数据存储设备/组件中的任意一个,诸如非易失性存储器、磁盘、磁盘驱动器、硬盘驱动器、固态驱动器或可以用于信息的非易失性存储的任意其他介质。然而,数据存储存储器250具体地不包括或不涵盖通信介质、任意通信介质或任意信号本身。与操作存储器220相反,计算设备200将数据存储存储器250用于非易失性长期数据存储,而不是用于运行时数据存储。

[0039] 此外,计算设备200可以包括或耦合到任意类型的处理器可读介质,诸如处理器可读存储介质(例如操作存储器220和数据存储存储器250)和通信介质(例如通信信号和无线电波)。尽管术语处理器可读存储介质包括操作存储器220和数据存储存储器250,但是在整个说明书和权利要求中,术语“处理器可读存储介质”无论是单数还是复数使用都在本文被定义,使得该术语“处理器可读存储介质”具体排除并且不涵盖通信介质、任意通信介质或任意信号本身。然而,术语“处理器可读存储介质”的确涵盖处理器高速缓存、随机存取存储器(RAM)、寄存器存储器和/或类似物。

[0040] 计算设备200还包括输入接口260,其可以被配置为使得计算设备200能够从用户或其他设备接收输入。另外,计算设备200包括输出接口270,其可以被配置为提供来自计算设备200的输出。在一个示例中,输出接口270包括帧缓冲器、图形处理器、图形处理器或加速器,并且被配置为绘制显示以用于在分离的视觉显示设备(诸如监视器、投影仪、虚拟计算客户端计算机等)上进行呈现。在另一个示例中,输出接口270包括视觉显示设备,并且被配置为绘制和呈现用于观看的显示。

[0041] 在所示的示例中,计算设备200被配置为经由网络适配器280与其他计算设备或实体进行通信。网络适配器280可包括有线网络适配器,例如以太网适配器、令牌环适配器或数字订户线(DSL)适配器。网络适配器280还可包括无线网络适配器,例如Wi-Fi适配器、蓝牙适配器、ZigBee适配器、长期演进(LTE)适配器或5G适配器。

[0042] 尽管示出了具有以特定布置配置的某些组件的计算设备200,但是这些组件和布置仅仅是可以在其中采用该技术的计算设备的一个示例。在其他示例中,数据存储存储器250、输入接口260、输出接口270或网络适配器280可以直接耦合到处理电路210,或者可以经由输入/输出控制器、桥或其他接口电路耦合到处理电路210。该技术的其他变体是可能的。

[0043] 计算设备200的一些示例包括适于存储运行时数据的至少一个存储器(例如操作存储器220)和分别适于执行处理器可执行代码的至少一个处理器(例如处理单元210),该处理器可执行代码响应于执行,使得计算设备200能够执行动作。

[0044] 示意性系统

[0045] 图3是示出了用于区块链联盟的系统(300)的示例的框图。系统300可以包括网络330、VN 351-353、成员设备341-343和参与方设备311-313。每个VN 351-353均包括对应的TEE 361-363。

[0046] 成员设备341-343、参与方设备311-313和/或VN 351-353中的每一个可以包括图2的计算设备200的示例。图3和本说明书中图3的对应描述示出了示例系统，其出于不限制本公开的范围的示意性目的。

[0047] 网络330可以包括一个或多个计算机网络，包括有线和/或无线网络，其中每个网络可以是例如无线网络、局域网(LAN)、广域网(WAN)、和/或全球网络，诸如因特网。在包括基于不同体系结构和协议的LAN的互连的LAN集合上，路由器充当LAN之间的链接，从而使消息能够从一个发送到另一个。此外，LAN内的通信链路通常包括双绞线或同轴电缆，而网络之间的通信链路可能利用模拟电话线，包括T1、T2、T3和T4的完整或部分专用数字线、综合业务数字网(ISDN)、数字用户线(DSL)、包括卫星链路的无线链路或本领域技术人员已知的其他通信链路。此外，远程计算机和其他相关电子设备可以经由调制解调器和临时电话链路来远程连接到LAN或WAN。网络330可以包括各种其他网络，诸如使用诸如6LoWPAN、ZigBee等的本地网络协议的一个或多个网络。某些IoT设备可以经由网络330中的不同网络而不是其他IoT设备来连接到用户设备。本质上，网络330包括信息可以在VN 351-353、成员设备341-343和参与方设备311-313之间传播的任意通信方法。尽管每个设备或服务显示为连接到网络330，但这并不意味着每个设备都与所示的每个其他设备通信。在一些示例中，所示的某些设备/服务仅经由一个或多个中间设备与所示的某些其他设备/服务通信。而且，尽管网络330被示为一个网络，但是在一些示例中，网络330可以替代地包括可以彼此连接或可以不彼此连接的多个网络，其中示出的设备中的一些通过多个网络中的一个网络彼此通信，并且所示的设备中的其他设备与多个网络中的不同网络彼此通信。

[0048] 在一些示例中，成员设备341-343是成员用来在网络330上进行通信的设备，诸如用于成员与其对应的VN之间的通信，例如以认可VN。在一些示例中，参与方设备311-313是成员用来在网络330上进行通信诸如以请求交易的设备。

[0049] 在一些示例中，VN 351-353是在正常操作期间验证和处理提交的区块链交易并执行链码的设备。如上所述，在一些示例中，每个VN 351-353包括TEE。在一些示例中，TEE能够实现在处理器内部受保护区域的创建，使得保护区域中的存储器被加密，并且仅在TEE内部被解密。TEE可以互换地称为飞地(“enclave”)，或者在某些示例中，TEE可以看作是飞地的子集。TEE证明可用于验证TEE中运行的代码。在某些示例中，TEE允许TEE内部发生的计算的完整性以及TEE内部发生的事情的机密性。

[0050] TEE的一些示例基于硬件，而其他示例基于软件。基于硬件的TEE的示例使用对定义TEE地址范围的存储器范围的基于硬件的保护。在一些示例中，对TEE的写被加密。在某些示例中，CPU不允许TEE地址范围之外的任意内容看到以明文形式的该地址范围，并且在TEE地址范围中不许可从地址范围区域之外进行写入。

[0051] 在一些示例中，在建立联盟区块链网络之前，某些细节由区块链网络的预期成员同意，包括哪些区块链协议代码和哪些共识协议代码将在联盟区块链网络的每个验证节点的TEE中执行。在联盟区块链网络的建立中，一个或多个验证节点得到联盟区块链网络的更

多成员之一的认可。而且,TEE证明可用于验证每个验证节点是否正在执行同意的区块链协议代码和同意的共识协议代码。在建立联盟区块链网络之后,在某些示例中,能够改变最初同意的参数,诸如要在TEE中运行的同意的区块链协议代码--在某些示例中,此类改变是基于如下面更详细地解释的共识协议,N票中M票赞成的投票和/或类似物确定的。

[0052] 系统300的示例可以使联盟区块链网络能够使用任意合适的协议,同时仍然能够实现COCO的机密性保证,如下面更详细地讨论的。

[0053] 系统300可以包括比图3所示更多或更少的设备,其仅以示例的方式示出。

[0054] 图4是示出了可以与例如参与方411和/或认可方441通信的VN 451的示例的框图。VN451可以包括TEE 461、区块链服务471和磁盘存储装置472。TEE461可以包括区块链协议465、机密联盟(COCO)区块链应用编程接口(API)466和COCO核心467。

[0055] 区块链服务471可以包括执行用于维持区块链系统的可用性和耐久性的功能的软件。当TEE中没有足够的可用存储装置来存储所有数据时,或在其他合适的情况下,磁盘存储装置472可以例如包括加密的区块链数据和元数据。区块链协议465可以包括同意的区块链协议代码。在某些示例中,COCO核心是核心COCO框架代码,下面将对其进行详细说明。在一些示例中,COCO API 466是用于COCO框架的API。

[0056] VN 451可以通过网络与其他VN、一个或多个参与方和/或一个或多个认可方通信,包括例如参与方411和认可方441,其中认可方441是认可VN 451的成员。

[0057] 在一些示例中,TEE之外的VN中的所有内容都被建模为不受信任。在一些示例中,TEE外部的VN的部分起到维持系统可用性和耐用性的作用。在某些示例中,参与方或认可方与TEE之间存在安全套接字层(SSL)、传输层安全性(TLS)或其他安全通道,以用于安全通信,其中TEE外部的VN中的元素是通过进行安全通信。

[0058] 在一些示例中,TEE中的区块链协议的生命周期由TEE内部运行的软件基于成员的指令、并且随后基于作为整体的联盟(例如法定人数)的决定来控制,如下文更详细讨论。

[0059] 如以下更详细地讨论的,尽管在图4中未示出,但是在一些示例中,TEE被分成两个单独的受保护区域,一个具有区块链API,并且另一个运行区块链协议,它们之间具有安全连接。

[0060] 示意性过程

[0061] 为了清楚起见,根据由系统的特定设备或组件以特定顺序执行的操作来描述本文描述的过程。然而,应注意,其他过程不限于所述的序列、设备或组件。例如某些动作可以以不同的顺序执行、并行地执行、被省略、或者可以由附加的动作或特征来补充,无论本文中是否描述了这样的顺序、并行性、动作或特征。同样,本公开中描述的任意技术都可以结合到所描述的过程或其他过程中,无论该技术是否与过程结合进行具体描述。所公开的过程还可以在其他设备、组件或系统上或由其他设备、组件或系统执行,不管是否本文描述了这样的设备、组件或系统。这些过程也可以以多种方式体现。例如它们可以体现在制品上,例如作为存储在处理器可读存储介质中的处理器可读指令、或作为计算机实现的过程来执行。作为替代示例,这些过程可以被编码为处理器可执行指令,并经由通信介质发送。

[0062] 图5A-5C是示出用于建立联盟区块链网络的过程(520)的示例数据流的图。

[0063] 在所示的示例中,步骤521首先发生。在一些示例中,在步骤521,区块链联盟的预期成员在建立网络之前同意联盟区块链网络的某些方面。在一些示例中,同意的方面可以

包括以下一项或多项:初始成员将是谁、被批准在网络的VN的TEE中执行什么代码(包括区块链协议代码和共识协议代码)、网络的VN中可接受什么处理器、什么构成可接受的TEE、区块链代码的哪些软件版本将被执行等。

[0064] 如所示,在一些示例中,接下来发生步骤522。在一些示例中,在步骤522,每个预期成员认可至少一个单独的VN。在某些示例中,作为验证的一部分,预期成员将同意的代码存储在TEE中,并将公钥/私钥对(分别为PBK和KBK)存储在TEE中。某些同意的方面也可以存储在VN中,诸如同意的成员资格列表和要在TEE中执行的同意的代码的标识。

[0065] 在一些示例中,可以以与上述不同的方式执行步骤522。例如在某些示例中,并非每个预期成员都认可单独的VN。例如在某些示例中,仅认可一个VN。

[0066] 如所示,在一些示例中,接下来发生步骤523。在步骤523,每个VN发现网络中的其他VN。在一些示例中,仅存在一个VN,并且不执行步骤523。

[0067] 如所示,在一些示例中,步骤524接下来发生。在步骤524,在一些示例中,从联盟的预期成员中的每一个接收以下各项:多个成员列表、以及来自联盟的多个预期成员的多个授权。在一些示例中,授权是与区块链协议代码的预定类型和共识协议代码的预定类型相关联的指示。在一些示例中,成员资格列表和授权由VN中的每个VN发送,并且每个VN接收由每个其他VN发送的成员资格列表和授权。在其他示例中,只有一个VN,并且从预期成员的成员设备接收成员资格列表和授权。

[0068] 如所示,在一些示例中,接下来发生判定框525。在判定框525,在一些示例中,做出关于来自预期成员中的每一个的成员列表是否彼此匹配的确定。如果不是,则在一些示例中,过程进行到返回框,在该处恢复其他过程。然而,如果在判定框525处的确定是肯定的,则在一些示例中,过程移至判定框526。

[0069] 在一些示例中,在判定框526处,做出关于来自预期成员中的每一个的区块链协议代码和共识协议代码的类型的授权是否彼此匹配的确定。如果不是,则在一些示例中,过程进行到返回框,在该处恢复其他处理。然而,如果在一些示例中,判定框526处的确定为肯定,则处理移至判定框527。

[0070] 在判定框527处,TEE证明可用于验证与联盟的预期成员相关联的节点是否存储了要在TEE上运行的同意的授权区块链协议代码和共识协议代码。在某些示例中,如果TEE证明为否定的,则过程移至返回块。否则,在一些示例中,处理前进到框528。

[0071] 在框528,可以在VN之间交换私钥。如图所示,在一些示例中,接下来发生框529。在框529处,可以从私钥中的每一个生成区块链主密钥(BMK)。如图所示,在一些示例中,接下来发生判定框530。在框530,在一些示例中,参与方被成员批准加入。在一些示例中,允许参与方请求交易并查看他们被授权查看的交易,但不具有成员的其他权利,诸如对改变成员资格进行投票、改变所使用的协议区块链代码的投票权等。在某些示例中,网络现在准备处理来自参与方的交易。然后该过程可以进行到返回框。

[0072] 在一些示例中,公共网络可以允许任意参与方提交交易,但是非公共网络要求参与方要被供应,这是VN认可方的另一特权。参与方可以由被允许提交交易以供网络执行的公钥/私钥对表示。参与方集合不一定包括成员,尽管在某些示例中参与方集合包括成员。成员可以通过向他们的VN提交参与方的公共交易密钥(PTK)来授权参与方,并且VN可以与网络共享他们供应的参与方的列表。如果网络要求参与方由一个以上的成员批准,则该网

络可以使用类似于批准新成员的协议，如下文更详细地讨论。

[0073] 一旦建立了联盟区块链网络，联盟区块链网络就可以开始接收和处理交易。

[0074] 图6A-6B是示出了用于联盟区块链网络的交易处理的过程(620)的示例数据流的图。

[0075] 在所示的示例中，步骤621首先发生。在一些示例中，在步骤621，参与方使用区块链主密钥加密区块链交易。如图所示，接下来发生步骤622。在一些示例中，在步骤622，参与方发送加密的区块链交易。如图所示，在一些示例中，接下来发生步骤623。在步骤623，可以将加密的区块链交易转发到基于同意的共识协议代码选择的特定VN。

[0076] 例如在一些示例中，可以使用共识算法来选举VN领导者，并且VN领导者接收交易、提交交易并将交易广播到所有其他成员。(在本文中，VN领导者可互换地称为主VN。)在这些示例中，在步骤623，使用共识算法选举VN领导者，并将加密的区块链交易转发给VN领导者。

[0077] 如所示，在一些示例中，接下来发生步骤624。在一些示例中，在步骤624，加密交易被转发到的VN接收加密区块链交易。如图所示，在一些示例中，接下来发生步骤625。在框625，在一些示例中，VN通过执行交易的代码来处理区块链交易。

[0078] 如所示，在一些示例中，接下来发生步骤626。在框626，在一些示例中，VN基于区块链交易的处理来直接更新已处理区块链的官方状态。“直接”更新状态表示VN更新状态而无需任意其他实体进行任意操作或确认即可更新状态。如图所示，在一些示例中，接下来发生步骤627。在步骤627，可以将已处理的区块链的更新后的官方状态广播到区块链网络。

[0079] 在一些示例中，由于信任，VN不进行重新计算以进行验证，这使得可以在步骤626直接更新区块链状态。在一些示例中，因为区块链联盟网络中的VN是完全信任的，网络上的其他VN可以隐式接受从受信任的VN接收到的任意区块链状态更新，因为它们符合区块链的协议，并且被保护不受TEE的外部篡改。此外，在一些示例中，不需要交易的副本来确认阻止。

[0080] 然后，该过程可以进行到返回框，其中恢复其他处理。

[0081] 在一些示例中，链码不需要由网络执行一次以上，链码可以是不确定的，并且链码可以包括与外部系统的交互。在一些示例中，区块链还可以构建为具有不同级别的防御能力，以抵御试图危害完整性和机密性的敌对参与方，从防止单个流氓成员到N票中M票赞成的投票，到要求所有成员同意状态变化。网络可以容纳任意的区块链抽象、任意的区块链协议和任意的分类帐，并且可以能够集成任意类型的现有区块链技术。本文，关于N票中M票赞成的投票，N是指成员总数，并且M是指用于建立表决的法定人数的成员数。

[0082] 网络的示例可以能够实现在不改变成员之间的信任假设的情况下改善区块链的性能特征。该网络的示例可以允许将数据放到区块链上时保持私有，以便只有相关方才能看到交易。网络的示例可以使在区块链内部运行的代码能够具有不确定性，使得例如每次都可以产生不同的结果。网络示例可以在不引入不必要的复杂性、引入额外的显著性能开销或引入不自然的信任假设的情况下实现此类结果。

[0083] 网络的示例在网络的VN中进行TEE证明，以便TEE内部的代码可以具有信任，其接受在TEE的操作环境中TEE周围的操作系统和其他用户模式代码是完全不受信任，而TEE仍然可以通过依靠硬件强制的隐私(例如通过利用或以其他方式采用软件防护扩展、平台安全处理器、安全执行环境等)来以受信任的方式操作，并且TEE可以通过建立到TEE外部某个

端点安全连接来在外部投射信任。在一些示例中，这允许TEE内部发生的计算的完整性以及TEE内部发生的事情的机密性。网络示例使用与TEE相关的信任作为区块链系统中信任假设的构造块。

[0084] 在一些示例中，由于其他VN可以验证其认可成员对VN的认可的事实，并使用TEE证明VN正在执行由成员资格批准的代码，因此网络中的每个VN都被网络完全信任。如前所述，在VN证明VN的代码是在受信任的TEE中运行的受信任版本的事实，并且其具有相同的成员资格列表的事实之后，VN可以通过类似于认可的过程来建立相互之间的信任连接，其他VN释放了其认可成员的KBK。在一些示例中，作为来自另一节点的输入而提供的一个节点的输出可以被信任，因为生成该输出的代码被证明是先前由成员资格批准的信任代码。在一些示例中，区块链的所有内容均被加密，并且区块链的内容仅在具有有效证明的TEE内部被解密，从而能够实现机密性。另外，具有有效证明的TEE中的代码可以是不确定的-由于TEE是受信任的，因此这种示例中，不需要其他节点来重现其他节点的结果。

[0085] 在一些示例中，VN根据机密联盟区块链框架(COCO)框架进行操作，该框架允许任意适当的区块链协议(例如由成员同意)被使用，并允许任意适当的共识协议(例如成员同意)与COCO框架一起被使用。尽管COCO中使用了“机密”一词，但COCO框架的某些示例包括机密性，而COCO框架的某些示例不包括机密性。在一些示例中，COCO框架不被自身使用，而是例如与由成员资格选择的区块链协议和共识协议结合使用。在一些示例中，COCO框架是在VN中运行的代码，其用于建立区块链网络，支持任意区块链和共识协议，任意区块链分类帐，任意区块链抽象以及任意合适的区块链技术的使用。在一些示例中，大多数COCO代码在TEE中被执行，但是COCO的某些外围方面可以在TEE之外执行，如下面更详细地讨论的。在一些示例中，COCO框架本身并没有定义区块链分类帐和实际的区块链交易处理，而是允许使用任意合适的分类帐和区块链协议，例如由成员同意要使用的那些协议，并为交易启用机密属性，管理信任假设，允许链码仅由网络执行一次，允许链码不确定，并允许链码包括与外部系统的交互。COCO框架还可能包括针对试图破坏完整性和机密性的敌对参与方的各种防御级别，从针对单个流氓成员的保护，到N票中M票赞成的投票，到要求所有成员同意状态变化。可以将根据在VN中的每一个上执行的COCO框架进行操作的VN的网络称为COCO网络。

[0086] 如上所述，在一些示例中，成员最初同意某些批准，包括例如成员将是谁，被批准在TEE中执行什么代码，可接受的处理器是什么，什么构成接受的TEE，要使用什么COCO框架，将执行什么软件版本的区块链代码，并且每个成员至少启动一个VN。在某些示例中，每个成员维护至少一个VN，每个VN都参与区块链网络的交易处理和共识协议。在其他示例中，每个成员不一定维护其自己的VN。

[0087] 在一些示例中，某些初始批准需要一致同意以便建立网络，而其他批准可能仅需要在预期成员的法定人数之间达成协议，或者初始批准可以基于在共识协议代码中的共识协议来被确定。

[0088] 在某些示例中，在正常操作期间，VN验证并处理提交的交易，并执行链码。成员可以向可信执行环境(TEE)部署他们信任的区块链协议实现的版本，并且一旦区块链代码证明其是成员信任的代码并且它在也受到成员信任的TEE中执行的事实，该成员通过为其供应他们的公共和私有区块链密钥(分别为PBK和KBK)来认可VN。在某些示例中，VN的所有者被视为其认可方，并且认可方可以通过提交成员的公共区块链签名密钥来更新VN信任的成

员资格列表。

[0089] 在一些示例中，成员资格和协议更新以及对区块链和链码状态的更新需要共识。在一些示例中，如上所述，使用的共识协议是用于建立网络的同意参数之一。在某些示例中，每个VN都完全信任与之建立信任连接的每个其他VN。因此，在某些示例中，只要区块链更新与VN维护的现有状态没有冲突，其就可以被明确地接受。在这些示例中，仅当可能存在冲突更新时才需要明确的共识协议。在各种示例中，网络可以通过多种方式实现共识，但是在某些示例中，网络不需要浪费的工作量证明、造成延迟的时限或潜在的不公平的利益证明算法。

[0090] 在一些示例中，共识依赖于Paxos或许多类似Paxos的共识算法之一，当网络由相对较少数量的共识参与方组成时，这是可行的。在一些示例中，在TEE中执行的区块链代码实现共识，并且由于该代码是受信任的，因此无需防御拜占庭式错误，诸如恶意消息。在一些示例中，Paxos被允许设置为增长以容纳每个成员的至少一个VN，但是在某些示例中，不包括多于一个的VN，以使共识参与方集合保持较小，以实现最大效率。在某些示例中，Paxos类型的共识用于实现集中式数据库的高可用性，因此示例小型网络可以使用它来实现数据库级的吞吐量和延迟。这种方案可以确保存在网络的大多数人都同意的一个版本的区块链状态。

[0091] 图7是示出用于区块链交易处理的过程(760)的示例数据流的图。

[0092] 首先，参与方可以提交交易。在一些示例中，交易用BMK加密。

[0093] 在一些示例中，网络中的任意VN都可以接受和处理交易，并且使用同意的共识协议。在一些示例中，基于共识协议，将接收到的交易转发给充当主机的VN。本文示例中，主VN随后接收交易。

[0094] 接下来，根据同意的共识协议，要处理交易的VN可以通过执行交易代码来处理交易。在某些示例中，主VN然后解决所有冲突。在某些示例中，TEE内部的所有数据都是纯文本格式。在某些示例中，TEE(磁盘/网络)外部的数据由BSK签名并由BMK加密。在某些示例中，每个VN都存储分类帐的完整副本，TEE内部的数据为纯文本格式，TEE外部的数据，诸如磁盘上的数据或通过网络传送的数据，由BSK签名并由BMK加密。

[0095] 在其他示例中，每个VN不需要存储分类帐的副本。在其他示例中，分类帐可以存储在外部存储云服务中或本地存储阵列中。

[0096] 接下来，VN可以向其他VN广播交易和交易状态。接下来，VN可以直接更新区块链状态。在某些示例中，由于信任，VN不需要重新计算以进行验证，并且区块链状态被直接更新了。

[0097] 图8是示出了包括具有三个成员和三个对应的VN的示例COCO网络(800)的系统的示例的框图。图8示出了示例COCO网络，其中已供应了三个VN，每个VN由不同的成员供应。

[0098] 将成员i的公钥描述为PBK_{mi}，并将其私钥描述为KBK_{mi}。图8的示例中的VN已建立信任连接并共享了这些密钥。

[0099] 在一些示例中，一旦建立了具有以VN代表的大多数成员的受信网络，基础在可以实现采用高效共识算法的区块链协议的位置处，在该基础上，单个VN可以代表整个网络进行操作，以便其他节点接受知道它们符合区块链的规则的交易，包括链码交易。

[0100] 在一些示例中，VN还可以单方面或通过表决方案批准参与方，包括他们自己。在一

些示例中，参与方没有网络投票特权，但是由于网络被供应有其公共交易密钥(PTK)，因此可以提交交易。

[0101] 通过利用网络的信任，VN的区块链协议代码可以接受来自其他VN的、知道它们遵守区块链协议规则的交易，并且可以进行与任意已经提交的交易不冲突的交易。在一些示例中，如果一个交易与另一个未提交的交易发生冲突，则VN使用共识协议来确定哪个交易获胜，并且大多数VN将迅速收敛于获胜的交易。此外，由于VN的执行可能受到TEE的保护而不受外部检查，因此它可以实现任意机密性模型，包括仅允许参与交易的成员查看该交易。

[0102] 在一些示例中，VN通过使用一对区块链公钥和私钥(BPK和BSK)来保护外部存储的区块链状态的完整性，并可选地通过使用从成员的KBK的集合导出的区块链主密钥(BMK)来保护其机密性。在某些示例中，由于这个原因，KBK在网络的VN之间共享。在一些示例中，BMK特定于成员资格集合，并且灵活的N个中的M个方案通过少于M个共谋成员来保护BMK免受破坏。在一些示例中，当成员资格集合更新时，网络将生成新的BMK，以保护所有随后附加的状态。

[0103] 上面的各种示例讨论了联盟区块链网络的建立和联盟区块链网络中的交易处理。在一些示例中，可以基于共识协议代码来改变诸如成员资格、区块链协议代码、共识协议代码、COCO框架、处理器、TEE和其他参数的初始同意的参数中的一个、一些或全部。例如在一些示例中，可以向验证节点发出请求以更改这些参数之一，诸如成员资格添加、成员资格移除、包括TEE的处理器的处理器类型、TEE类型、在TEE中执行的代码版本、或在TEE中执行的机密联盟区块链框架(COCO)版本。接收到请求的验证节点可以基于共识协议代码来确定是否更改参数。

[0104] 如上所述，在一些示例中，仅在N个成员中M个成员的同意下才能恢复BMK。类似地，可以通过N个投票中的M个来改变对上述某些或所有参数的改变，其中N是当前成员总数，M是从1到N的数字，即改变参数的法定人数，其中M是最初建立联盟区块链网络时同意的参数之一。如上所述，在一些示例中，成员具有投票特权，而不是成员的参与方没有投票特权。

[0105] 在一些示例中，新参与方的添加需要基于同意的共识协议达成共识。在其他示例中，成员可以单方面添加新参与方。是否可以单方面添加新参与方可能是同意的初始参数之一。

[0106] 在一些示例中，可以基于同意的共识协议来批准对网络的成员列表的添加和从网络的成员列表的移除，以及对网络允许参与的经批准的区块链实现的集合进行改变。此外，在一些示例中，对于要求参与方由成员授权的网络，成员还可以批准有权向网络提交交易的参与方集合。成员之间达成一致的协议可以通过在每个验证节点的TEE中存储的同意的代码来实现。

[0107] 在一些示例中，每个成员用公钥/私钥对(PBK和KBK)表示，该成员通过认可的过程使用公钥/私钥对来引导其在COCO中的成员资格。在某些示例中，认可表示成员完全信任网络区块链协议的特定实现的事实。在某些示例中，VN在受信任的执行环境(TEE)中执行认可的区块链代码，并且每个VN均恰由一个成员认可。例如存储在区块链上的代码可以作为交易的一部分执行。在某些示例中，作为认可操作的一部分，成员与VN共享成员的KBK。在这些示例中，由于成员出于安全原因与VN共享成员的KBK，因此对于成员来说，信任区块链协议实现和TEE不泄露密钥可能很有用。可以通过对区块链协议的仔细审查及其实现以及基于

威胁模型和TEE破坏的风险来实现这种信任。在某些示例中，每个成员都认可至少一个VN，但是认可多个节点可以提供高可用性。

[0108] 认可可以向认可成员提供排他的能力，以授权VN将参与的其他成员以及区块链协议实现的其他实现。在某些示例中，VN将接受成员资格更新和批准，以信任区块链协议实现，而不是仅来自VN的认可成员的其自身。在某些示例中，为了保护VN的所有外部通信，VN使用TEE来生成公钥/私钥对，并在外部共享其公钥。成员可以使用VN的公钥与VN建立安全通道，包括针对其VN不是认可方的潜在通道。然后，VN可以使用VN的KBK的所有权证明来认证VN请求。

[0109] 一种类型的VN请求是授权网络成员，如上所述，在某些示例中，只有节点的认可方才能执行。在某些示例中，要授权成员，认可方会提交该成员的PBK。在某些示例中，VN一旦填充了认可方的密钥和由其PBK标识的网络成员列表，便可以加入网络。在某些示例中，加入是VN与网络的其他VN建立受信任的连接的动作。网络可能会使用符合网络要求的发现协议，以向VN提供用于查找网络的其他VN并与网络的其他VN连接的能力。例如认可方可以通过提交节点的DNS名称、IP地址或特定于区块链网络的发现和连接系统的标识符来提供节点列表。

[0110] 如上所述，在一些示例中，建立与另一个VN的受信任的连接始于在传送VN的公用密钥上建立的相互认证的安全通道的创建。在某些示例中，作为信任建立的一部分，VN证明其拥有其认可方的KBK。在某些示例中，这可以确保远程节点已获得授权成员的认可，因为通道证明该节点可以提供引用VN公钥的认可，VN公钥由与成员PBK匹配的成员KBK签名。

[0111] 图9是示出了系统(900)的示例的框图，该系统包括COCO网络的一部分，该COCO网络的一部分包括经历验证节点证明和密钥交换的两个验证节点，作为加入该网络的新成员的最终部分。

[0112] 图9示出了VN证明和密钥交换的示例，作为经由节点正在运行的区块链协议实现的确切版本的交换以及成员资格列表的比较在节点之间建立信任的最后步骤。节点可以使用TEE证明来执行此操作，其中包括该节点正在特定TEE中运行特定实现的证明。

[0113] 在一些示例中，一旦VN确定远程节点正在VN也信任的TEE中执行受信任的代码，则VN会检查以确保远程认可方在本地成员身份列表中，并且如果是，则通道被认为是受信任的，并且VN将接受来自远程节点的区块链更新。在某些示例中，这意味着VN确信远程节点不会暴露或泄漏区块链协议不允许的数据。在某些示例中，同意成员资格集合本身可以通过让VN通过其PBK的散列(例如盐化散列(“salted hash”))而不是PBK本身来引用成员来保持保密。在某些示例中，一旦受信任的连接被建立，VN就会共享该VN的认可方的KBK。

[0114] 在一些示例中，区块链发展有两种方式：如上所述的成员资格更新和区块链代码更新。当网络同意协议更新时，这可能会导致网络希望接受区块链代码的一个或多个新实现。希望升级区块链代码的成员可以关闭其现有的VN并启动新的VN来代替现有的VN。成员可以选择允许新的VN信任以前的版本，或仅信任新版本。信任现有版本可以确保在新版本没有N个中的M个多数时可以继续处理交易，但是如果由于某种原因认为先前代码不可信，或者如果维持协议兼容性的复杂性太大了，则相反可以信任新版本而不是现有版本。然而，升级期间的跨版本不兼容可能会中断交易处理。

[0115] 在某些示例中，代表每个成员的VN之间的信任的建立导致每个节点最终获得所有

网络成员的KBK副本。在某些示例中，无论节点是直接从成员的VN还是从另一个受信任的VN接收交易，节点都将接受使用节点被供应参与的成员的私有签名密钥签名的区块链更新。

[0116] 向网络中添加新成员可能需要VN的认可方通过供应新成员的PBK来将其提议给网络。然后，认可方可以让其他VN知道该提案，以及认可方已经投票赞成新成员进入网络的事实。当其他成员提议同一成员时，其VN可能会记下投票并让网络知道。在某些示例中，VN不会接受成员提出的区块链更新，直到基于已达同意的共识协议达成共识为止。

[0117] 图10是示出用于将新成员添加到联盟区块链网络的过程(1080)的示例的图。图10示出了具有两个成员(m1和m2)加上第三成员m3的网络所遵循的步骤的示例。图10的步骤的示例如下进行。在(a)中，成员m1提出了新成员m3。成员的VN记录该提案并且成员的认可方投票，然后将该提案传递给其他VN。m1的VN记录了提案和m1的进入投票。然后，在(b)中，成员m2提议新成员，并且两个VN都记录了投票，其现在是一致的，因此在(c)中，每个VN都添加了新成员，一旦加入，VN就会考虑新成员的投票并将接受来自新成员的区块链更新。

[0118] 在一些示例中，成员可以随时请求区块链网络(即VN)将其从成员资格集中移除。然而，在某些示例中，取决于网络的成员资格规则，当代表除了被移除的成员之外的成员的多数或所有VN同意移除该成员时，该VN才会移除该成员。在某些示例中，用移除成员不具有投票(因此有效地≥1/2N)的警告，大多数(≥1/2N+1，其中N是初始成员资格计数)需要移除成员，以便防止流氓成员指示其VN移除所有其他成员，这将使其具有解密所有区块链状态的能力(以下将详细讨论区块链加密方案的示例)。在某些示例中，当成员被移除后，VN会破坏VN的成员的私有对称加密密钥的临时副本，并触发新成员资格元数据块的生成。

[0119] 成员还可以请求密钥翻转操作，由此，成员资格元数据被更新以反映新的公共密钥，并且成员的旧KBK用新的密钥进行加密并存储在成员资格元数据中。在某些示例中，一旦滚动，网络将不接受任意引用旧KBK的交易。然而，当成员证明他们具有对新密钥的访问时，网络可以允许该成员访问与旧密钥相关联的任意交易，这可以使成员能够完全访问其数据，同时阻止旧密钥访问。

[0120] 在某些示例中，除了针对成员资格、受信任的连接建立和共识的COCO的选项之外，COCO还支持任意的区块链协议。例如区块链代码可以实现以太坊、Corda、Chain Core或Hyperledger系统。然而，现有系统并未被设计为利用某些COCO示例的信任和机密性。

[0121] 在一些示例中，由于COCO VN是完全受信任的，因此网络上的其他VN可以隐式接受从受信任的VN接收到的任意区块链状态更新，因为这些更新符合区块链协议，并且被保护不受TEE的外部篡改。这就允许链码的一次性和不确定执行的可能性。链码可能会使用TEE的不确定性，例如生成用于彩票的随机数或解决冲突。链码还可以向外部系统发出呼叫，这可以用链码所信任的外部Oracle来替换对链上Oracle的需求，其优点是外部系统可以在VN接受链码交易时被引用。如果这种具有副作用的外部操作有可能由于取代冲突而过时，或者在大多数VN提交之前交易被丢失了，只要补偿动作可以撤消该操作，链码就也可以执行具有副作用的外部操作。

[0122] 利用COCO信任基础，可以建立任意机密性模型。

[0123] 例如首先，参与方可以通过使用成员的KBK的PBK对交易进行加密来确保其提交交易的网络或更具体的VN将是机密的。在这些示例中，只有VN的代表成员认可的网络中的VN才能访问交易内容，其中可能包含敏感信息。对于其中参与方仅希望向作为具有附加成员

的网络中的一部分的VN透露交易，可以使用唯一密钥对交易进行加密，而该唯一密钥本身使用这些成员的PBK的N个中的M个编码进行加密。

[0124] 在一些示例中，当成员将自己与另一成员或多个成员之间的交易指定为机密时，VN全部以明文方式处理交易，但是交易本身被存储在用BMK加密的区块链中。在某些示例中，VN仅允许交易中涉及的成员查看交易。可以将相同的模型应用于链码和链码状态。

[0125] 替代地，在其他示例中，大多数成员可以请求披露其他方式的机密交易，这对于联盟随后需要全面了解历史活动的情况可能是理想的。其他示例允许审核成员的供应。就像新的参与成员的N个中的M个接受一样，N个中的M个方案可以使成员同意承认具有特殊特权的成员，例如能够读取所有交易，甚至标记为机密的交易。

[0126] 示例性区块链网络的一方面是VN如何达成共识。可以使用多播，广播树或通过针对成员资格的规模的成员资格和VN通信网络的拓扑高效地认为的任意其他协议来执行在整个网络中分发交易。在某些示例中，VN的可信任性质意味着可以像将消息分发到大多数节点一样快地达成共识。

[0127] 在一些示例中，在任意VN都可以接受交易的模型中，每个VN可以构建不同版本的区块链。在任意时候，VN都可能具有由VN的多数子集N提交的一些交易，以及由不同子集N提交的其他交易。在某些示例中，由于最终所有节点都收敛到已提交交易的公共视图，即使块的顺序是唯一的，每个VN的已提交区块链状态是区块链状态的正确表示。

[0128] 可以用其他共识协议来构建COCO网络，但是一些示例使用共识协议，该协议可以利用以下事实：从可信VN接收到的消息本身就是可信的，以实现高效的协议和最大的吞吐量。

[0129] 在一些示例中，交易处理如下进行。首先，参与方提交交易。参与方提交的交易可以使用BMK进行加密，并基于共识协议被转发给充当主VN的VN。(主VN可互换地称为引导方VN。)本文示例中，主VN然后接收交易并通过执行交易代码来处理交易。本文示例中，主VN然后解决任意冲突，并将交易和交易的状态广播到其他VN。在一个示例中，每个VN都存储分类帐的完整副本，TEE内部的数据为纯文本格式，并且诸如磁盘上的数据或通过网络通信的数据的TEE外部的数据由BSK签名并由BMK加密。然后，VN可以直接更新交易状态。

[0130] 在其他示例中，每个VN不需要存储分类帐的副本。在其他示例中，分类帐可以被存储在外部存储云服务或内部存储阵列中。

[0131] COCO区块链的持久状态，即写入持久存储装置的状态，可能包括三种类型的数据：元数据、交易和链码状态。链码代码可以被认为是链码状态的一部分。一对区块链公钥(BPK)和区块链私钥(BSK)可用于保护持久状态的完整性。BPK可以用于验证由对应BSK签名的块状态摘要。在一些示例中，该链是仅附加有到链的任意附加，该任意附加使用BSK签名的摘要被绑定到链的先前部分。如果BSK泄漏，流氓成员或合谋成员可能会破坏区块链的完整性，这就是在某些示例中BSK在TEE中生成并完全密封在TEE中的原因。

[0132] 在一些示例中，元数据包括用于进行完整性验证的纯文本的BPK。在一些示例中，使用对称密码术用区块链主密钥(BMK)来保护不以明文暴露的任意区块链状态的机密性。BMK也可能以受保护的方式成为区块链元数据的一部分，因为在紧急情况下(例如在所有TEE销毁后进行灾难恢复)，法定人数的成员可能需要或希望绕开TEE来解密和恢复区块链数据。

[0133] 在一些示例中，区块链包括随着时间增长的分类帐。在某些示例中，区块链分类帐可能会变得太大而无法被存储在TEE中，而是代之以被存储在磁盘中。在某些示例中，当私有数据被存储到磁盘时，BPK和BSK允许将数据存储在磁盘上，并在读回数据后验证数据是否遭到篡改。在某些示例中，BMK可以用于加密被存储到磁盘的私有数据，以便即使将数据被存储在磁盘上，也无法看到数据。

[0134] 如在某些示例中使用的保护BMK的一种方案是，将针对每个成员的BMK副本存储在区块链元数据中，该BMK的副本被加密到该成员的KBK。这可以防止BMK泄漏到成员之外。然而，在某些示例中，它无法防止流氓或破坏的成员泄漏它。

[0135] 为了防御流氓节点或合谋成员，基于COCO的区块链可以使用N个中的M个的加密方案，以要求必须知道一定阈值数量的成员的KBK才能获得BMK。然后，只要VN与该数量的成员认可的足够的其他VN一起加入，VN即可获得访问，但是在某些示例中，这种访问不会很快。在某些示例中，由于成员只能在TEE外部访问其自己的密钥，因此至少需要M个成员泄漏或共享其KBK，以用于在VN的TEE外部解密链。

[0136] 虽然M可以等于N，或者甚至可以是1，但是COCO实现的一些示例要求M至少为1/2N+1，并且匹配成员资格集合协议共识模型，这可以确保只要有多数成员集合不合谋，区块链状态的机密性由在TEE中执行的相互信任的代码保护。还可以确保只要网络可以与代表至少M+1个成员的VN建立连接，它就可以运行，从而即使在成员撤出其VN或其VN被从网络中划分的情况下，也可以确保弹性。

[0137] 在一些示例中，可以采用如下方式实现N个中的M个的方案。区块链成员资格元数据块中列出的每个成员均由成员的公钥以及加密到该公钥的BMK的片段表示，如表1所示。在某些示例中，第一个元数据成员资格块在网络引导过程期间被创建，将BMK加密为初始成员集合。当将附加成员添加到网络时，COCO的示例通过以下各项采取步骤来调整M和N：创建新的BMK，跨更新的N个成员集合对其进行分段，然后附加新的元数据块，新的元数据块包括由新的BMK已加密的先前BMK以及更新的成员资格列表两者。

[0138]

PBK_{m1}	PBK_{m2}	PBK_{m3}	BMK	E_{BMK}(交易)
$E_{KBKm1}(BMK_{m1})$	$E_{KBKm2}(BMK_{m2})$	$E_{KBKm3}(BMK_{m3})$		

[0139] 表1.持久性区块链中BMK的N个中的M个的加密

[0140] 与之前一样，根据区块链的N个中的M个的实现，直到成员资格更新的点的区块链状态可能会受到先前成员资格集合的合集的破坏。但是通过创建新的BMK，该区块链时间轴上从该点起的所有的区块链状态都可能受到更新的成员集的保护，包括更新的N个中的M个的方案，如果该区块链实现了一个。

[0141] 为了防止区块链回滚到先前状态，可以将签名、认证代码或最后添加到链中的唯一标识符存储在仅VN可以访问的受信任的非易失性存储装置中。类似地，最后添加到区块链的签名可以被包括在与成员的所有交互中，以允许成员检测将区块链回滚到先前状态的任意尝试。在某些示例中，在成员改变时成员资格后会在KBK上回滚，以防止旧代码版本中的错误。在某些示例中，在诸如从代码批准列表中移除代码版本的某些情况下，在每个成员资格版本之后重新运行证明、密钥交换或两者。

[0142] 在一些示例中，COCO适应于不需要每个成员具有专用VN的模型。共享的VN可以用于开发和/或测试、成本敏感的成员或作为云提供商或充当其他成员托管代理的成员的多租户托管解决方案的基础。在一个示例中，所有成员都依赖于单个集中方托管的单个VN(或一组高可用性的VN)。在集中模型的此示例中，所有成员可能都需要依赖集中方来处理交易和处理成员资格更新，并且除非区块链状态被流式传输或导出，否则集中式基础架构可能会成为区块链状态的单点故障。

[0143] 在一些示例中，为了容纳多租户VN，认可处理程序接受上述成员输入，在TEE内部为每个成员存储单独的认可状态。在某些示例中，认可处理程序仅在参与共享VN的成员之间达到法定人数后才提交改变。

[0144] 与传统的区块链实现不同，COCO VN的一些示例信任其对等方，从而能够跨VN实现高效、可扩展和高效的交易处理。

[0145] 几种措施可以实质上减轻与TEE破坏相关的风险。TEE破坏的可能性可以通过多种方式降低。降低TEE破坏的可能性的一种方式可以如下实现。每个VN分为两个部分。管理者TEE可能负责私钥管理、密码处理和成员资格更新逻辑，使得实现管理者TEE所需的代码量受到限制，并且该代码不必随每个区块链集成而变化。然后该代码可以被正式验证并被大量审核，以进一步最小化遭受破坏的风险。

[0146] 在这样的示例中，管理器TEE本质上是负责VN最受信任操作的基于TEE的硬件安全模块(HSM)，而工作者TEE是托管区块链协议代码的独立飞地。(在某些示例中，工作者TEE可能包括比管理者TEE大得多的代码库，并且工作者TEE的代码库随与COCO集成的每个分类帐而变化。)在这样的示例中，工作者TEE向管理者TEE发出请求用于使用VN的私钥对数据加密、解密和签名，使得如果工作者TEE被破坏，攻击者就可以访问由工作者TEE加载到存储器中的任意纯文本数据-但它无法直接访问密钥，独立解密磁盘上的块，或者提议或批准成员资格改变。此两个TEE分解的示例减少了管理者TEE破坏的可能性，并减少了工人TEE破坏的影响。

[0147] 在某些示例中，网络还可以强制执行以下规则:参与网络的VN授权是必须定期更新的租约。续订过程可能需要将TEE重置为原始状态，并且可能消除攻击者可能获得的任意立足点。不更新其租约的VN可以从网络中隔离，并且因此可以防止隔离生效后添加到区块链的新数据的暴露。

[0148] KBK的加密和认证功能可以被分成两个单独的密钥。这种划分可以消除VN复制认证私钥的需要。在某些示例中，仅加密密钥需要跨多个VN复制-这可以消除在管理者TEE受到破坏时成员认证私钥的全局暴露。

[0149] VN可以对从其他VN发送的交易进行采样，并在提交之前同步地完全验证采样。这可以确保可以快速检测到产生虚假结果的VN，以在不大幅降低性能的情况下发出警报。同样，分离的节点可以异步验证所有交易，并且如果检测到任意问题，也可以发出警报。

[0150] 区块链系统的各种示例利用TEE的属性，并且可以用作支持数据库级交易吞吐量和延迟以及灵活的机密性模型的区块链实现的基础。在一些示例中，许多不同的区块链协议都可以利用COCO框架的示例来提供具有区块链和交易机密性的高效交易处理。

[0151] 通过说明书，已经给出了其中成员具有其自己的VN的示例。然而，在一些示例中，如上面更详细地讨论的，多个成员可以共享相同的VN，或者甚至所有成员都可以共享由所

有成员认可的单个节点。

[0152] 并非所有示例都包括本文讨论的所有特征。例如某些示例不使用加密。

[0153] 图11是示出了用于区块链系统的过程(1190)的示例数据流的图。在一些示例中，图11的过程由验证节点执行。在所示的示例中，步骤1191首先发生。在一些示例中，在步骤1191，将预定类型的区块链协议代码和预定类型的共识代码存储在处理器的可信执行环境(TEE)中。

[0154] 如图所示，在一些示例中，接下来发生步骤1192。在一些示例中，在步骤1192，使用TEE证明来验证存储在TEE中的区块链协议代码是预定类型的区块链协议代码，并验证存储在TEE中的共识代码是预定类型的共识代码。如图所示，在一些示例中，接下来发生步骤1193。在步骤1193，可以接收改变预定类型的区块链协议代码的请求。如图所示，在一些示例中，接下来发生步骤1194。在一些示例中，在步骤1194，基于预定的共识代码来确定是否改变预定类型的区块链协议代码。然后处理进行到返回框，在该处恢复其他处理。

[0155] COCO中的以太坊机密性

[0156] 在一些示例中，在COCO中，平台的机密性保证与区块链运行时集成在一起。在一些示例中，以最小的开发人员修改核心协议的方式来处理机密性。

[0157] 控制数据流

[0158] 在一些示例中，在安全区域内，存在对什么数据流入和流出的完全控制。在某些示例中，所有链状态都存储在外部磁盘上，但使用仅驻留在飞地内的区块链主密钥进行加密。在这些示例中，这意味着链状态对每个指示的参数是可用的。

[0159] 以太坊中的交易与智能合约

[0160] 在一些示例中，以太坊交易具有两种格式：

[0161] • 将X以太币从地址A发送到地址B

[0162] • 与地址C的智能合约进行交互，并可能与某些提供的参数进行交互

[0163] 在一些示例中，在联盟的上下文中，第一点不相关-例如以太币是以太坊公共网络之外的无价值标记，并且COCO共识机制可以完全避免用于支付天然气开销的其主要用途。

[0164] 在一些示例中，在联盟上下文中，交易本身并不重要，但是重要的是给定交易具有智能合约状态的结果。

[0165] 从机密性的角度来看，如果关于原始交易的信息可用，则保持机密性可能意味着要清理交易-例如移除“发件人地址”和“收件人地址”字段中的信息。即使这样，也可以从统计分析中获得有关交易数量和时间戳的洞察力。例如在运营单个COCO网络的银行联盟中，该网络的参与方可以对交易数据进行分析，并了解到，每当宣布两家银行之间的合作伙伴关系时，公告前几天的交易总数激增。如果参与方注意到进行中的交易数量激增，则该参与方可能会拥有其他联盟参与方可能不希望该参与方拥有的机密信息。

[0166] 在一些示例中，鉴于对飞地外的数据流的总体控制，并且鉴于允许用户查询原始交易数据变得难以维护机密性，在某些示例中，未向用户提供有关交易的数据。以太坊RPC规范具有诸如“getTransaction”和“getBlock”的功能，这些功能可能尚未实现。仍可能支持允许用户与智能合约进行交互的全套功能。

[0167] 然而，在一些示例中，通知用户关于他们的交易是否被适当地包括在系统中。在一些示例中，通知是同步完成的，而不是向用户提供用户可以用来异步查询所述交易的状态

的交易ID。

[0168] COCO中的智能合约机密性

[0169] 在一些示例中，向用户提供了API端点以查询智能合约状态，但是通过将访问控制构建到智能合约中，确保了智能合约的状态只能由预期的参与方读取。

[0170] 在一些示例中，对于每个成员和参与方私有密钥，使用任意合适的确定性密钥生成算法来确定性地生成以太坊公钥/私钥对。例如在一个示例中，挖掘节点的私钥等于所供应的用户密码的SHA256哈希。本文示例中，以太坊地址进而从公共以太坊密钥确定性地生成的，因此存在一个成员/参与方到以太坊地址的映射。

[0171] 在一些示例中，由于该地址生成处理是确定性的，因此任意特定成员将知道其以太坊地址，并将其分发给其他成员。

[0172] 在一些示例中，COCO区块链运行时被配置为使得当交易由成员/参与方密钥签名时，运行时验证交易，查找(或即时生成)以太坊地址，使用恰当的密钥对以太坊交易进行签名，并将其转发给要评估的EVM。

[0173] 以下是智能合约的示例：

```
pragma solidity ^0.4.0;
contract Bank {
    event transferLog(address sender, address recipient, uint amount);
    mapping (address => uint) public accountBalances;
    function Bank() {
        accountBalances[msg.sender] = 1000000;
    }
```

[0174]

```
    function transfer(address recipient, uint amount) {
        if (accountBalances[msg.sender] < amount)
            throw;
        accountBalances[msg.sender] -= amount;
        accountBalances[recipient] += amount;
        transferLog(msg.sender, recipient, amount);
    }
}
```

[0175] 该合约可以充当银行。在某些示例中，部署合约的人的帐户开始为一百万，并且传递功能允许有足够余额的任何人将其资金传递到收款人账户。当发生传递时，“transferLog”事件可能会被记录到区块链中。

[0176] 在一些示例中，事件是具有可定义参数的区块链上的永久日志。用户还可以“订阅”特定事件-本文示例中，用户可能希望订阅转移事件，使得他们在收到钱时被通知。在某

些示例中,事件还为用户提供了了解其余额更新的方式。

[0177] 以上合约没有任意形式的访问控制。在某些示例中,accountBalances变量上的“public”修饰符会自动为帐户余额设置getter函数-可以访问区块链的任何人都可以检查任何人的余额。此外,在某些示例中,任何人都可以看到传递事件的完整日志。

[0178] 相反,与访问控制有关的以下合约已改变:

[0179]
```
pragma solidity ^0.4.0;
contract Bank {
  event transferLog(address availableTo, address sender, uint amount);
    mapping (address => uint) accountBalances;
    function Bank() {
      accountBalances[msg.sender] = 1000000;
    }
    function transfer(address recipient, uint amount) {
      if (accountBalances[msg.sender] < amount)
        throw;
      accountBalances[msg.sender] -= amount;
      accountBalances[recipient] += amount;
      transferLog(recipient, sender, amount);
    }
    function getBalance() returns(uint) {
      return accountBalances[msg.sender];
    }
  }
```

[0180] 该合约具有与访问控制有关的一些改变,并且“public”关键字已从accountBalances变量中移除。取而代之的是定义了“getBalance”功能,该功能返回当前用户的余额。在某些示例中,如果没有能力查看原始交易,则任意给定的用户将只能确定自己的余额。

[0181] 还对事件进行了轻微修改,引入了“availableTo”变量作为第一个参数。在某些示例中,如果区块链中的每个事件都实现了此变量,则当用户查询事件时,可以仅向客户显示“availableTo”参数对应于其地址的事件。

[0182] 地址数组可以存储在事件变量中，允许可以通过在“availableTo”参数中指定多个地址来进行机密性分组。

[0183] 作为智能合约的另一示例：

[0184]

```
pragma solidity ^0.4.0;
contract SecretStore {
    event UserAdded(address[] availableTo);
    string secret;
    address[] allowedUserArray;
    function SecretStore(address[] initialUsers, string newSecret) {
        secret = newSecret;

        allowedUserArray = initialUsers;
    }
    function addressInArray(address input) private returns (bool) {
        bool addressInArray = false;
        for (uint i = 0; i < allowedUserArray.length; i++) {
            if (allowedUserArray[i] == input) {
                addressInArray = true;
                break;
            }
        }
        return addressInArray;
    }
    modifier onlyAllowedUsers {
        if (!addressInArray(msg.sender))
            throw;
        _;
    }
```

```
        function addUser(address newUser) onlyAllowedUsers {
            if (!addressInArray(newUser)) {
                allowedUserArray.push(newUser);
                UserAdded(allowedUserArray);
            }
        }
        function getSecret() onlyAllowedUsers returns(string) {
            return secret;
        }
    }
```

[0185]

[0186] 这个更复杂的示例可以将对若干功能的访问限制为包含地址列表的“allowedUserArray”变量。构造函数中提供了此数组，以及可用于数组中地址的秘密字符串。在某些示例中，将新用户添加到该数组时，将创建一个事件，该事件可用于成员资格列表中当前的每个地址。

[0187] 在一些示例中，这种类型的智能合约允许将功能和状态限制为定义的地址集合，并且可以记录那些地址可用的事件。在某些示例中，以太坊地址是确定性的，并且可以在节点设置期间与PBK一起分发。在某些示例中，部署上述合约的COCO成员将知道他希望限制其访问的成员的以太坊地址，并可以据此创作其合约。

[0188] 实施机密性设计模式

[0189] 在一些示例中，为了实现机密性，该系统还包括一种用于强制要求用户在其智能合约代码中实现这些设计模式的约束的方式。

[0190] 在一些示例中，甚至可以使用正则表达式来强制执行以下设计要求。例如可将正则表达式用于：

[0191] • 对于智能合约变量，不允许使用“public”关键字。例如用户将为智能合约状态编写自己的getter函数，希望考虑到机密性问题。

[0192] • 需要事件将“availableTo”变量实现为第一个参数。例如这可以用于将事件条目锁定到特定的以太坊地址或地址组。

[0193] COCO的一般机密性

[0194] 上面讨论的是在以太坊中实现COCO机密性保证的特定示例。然而，本公开不限于以太坊，而是可以通常应用于任意合适的协议。本公开的示例可以使得当COCO框架与诸如以太坊的现有协议或其他合适的协议一起使用时，可以使用COCO机密性保证，并且可以在不需要对协议进行深度修改的情况下启用COCO机密性保证。

[0195] 在一些示例中，在包括区块链交易处理和状态更新的区块链操作期间，不允许访问原始交易数据。同样，在某些示例中，类似地，在包括区块链交易的处理和状态更新的区块链操作期间，除经由智能合约和同步通知外，不允许访问状态信息。

[0196] 关于同步通知，在一些示例中，当发生区块链交易时，与区块链交易相关联的用户

被同步地通知交易。

[0197] 在一些示例中，不允许访问原始交易数据和/或不允许访问状态信息的一部分可以包括禁用允许访问原始交易数据或状态信息的API。可以通过以下各项来禁用API：例如现在允许API，该API允许访问要被调用的敏感信息；或者完全不实现API。

[0198] 上面更详细地讨论了禁用API，该API将允许访问以太坊的敏感数据。在某些示例中，不同的分析用于不同的协议。在一些示例中，列举了协议的API功能，标识了揭示敏感状态信息的协议的API功能，然后禁用了所识别的API功能。

[0199] 可通过要求智能合约的机密性设计模式来实现对授权方的状态信息的访问，其中该机密性设计模式要求将访问控制构建到智能合约中。在某些示例中，内置在智能合约中的访问控制仅允许适当的用户接收状态信息，使得不会违反COCO机密性保证。

[0200] 在一些示例中，机密性设计模式要求事件被限制关于所述事件可用于哪些地址。在一些示例中，机密性设计模式要求将变量限制为由定义的功能访问。上面更详细地描述了以太坊的特定机密性设计模式。

[0201] 在一些示例中，在任意智能合约代码被编译或部署在区块链上之前，经由加密通道接收智能合约源代码，并且确定智能合约源代码是否符合机密性设计模式。在某些示例中，用静态分析来完成确定智能合约源代码是否符合机密性设计模式。在某些示例中，使用正则表达式完成确定智能合约源代码是否符合机密性设计模式。

[0202] 在一些示例中，在确定智能合约源代码不符合机密性设计模式时，该代码被拒绝。在一些示例中，可以向智能合约源代码的发送方发送指示该源代码不符合机密性设计模式的消息，并请求用户发送符合机密性设计模式的源代码。

[0203] 在一些示例中，在确定智能合约源代码符合机密性设计模式时，智能合约源代码被编译并部署，并且如果部署成功，则经由加密的通道来返回部署智能合约的地址。

[0204] 图12是示出了用于在COCO网络中实现机密性的过程(1280)的示例数据流的图。

[0205] 在所示的示例中，步骤1281首先发生。在一些示例中，在步骤1281，将预定类型的区块链协议代码存储在处理器的可信执行环境(TEE)中。如图所示，在一些示例中，接下来发生步骤1282。在一些示例中，在步骤1282，使用TEE证明来验证存储在TEE中的区块链协议代码是预定类型的区块链协议代码。如图所示，在一些示例中，接下来发生步骤1283。在步骤1283，接收区块链交易。

[0206] 如图所示，在一些示例中，接下来发生步骤1284。在一些示例中，在步骤1284，处理区块链交易，同时不允许访问原始交易数据。如图所示，在一些示例中，接下来发生步骤1285。在一些示例中，在步骤1285，基于区块链交易的处理来为区块链网络更新已处理区块链的状态，同时不允许访问原始交易数据。然后，该处理可以进行到返回框，在该框恢复其他处理。

[0207] 结论

[0208] 尽管以上具体实施方式描述了该技术的某些示例，并且描述了预期的最佳模式，但是无论上述内容在文本中出现的多么详细，都可以以多种方式实践该技术。细节可以在实现中变化，同时仍然被本文描述的技术所涵盖。如上所述，当描述技术的某些特征或方面时使用的特定术语不应被理解为暗示该术语在本文中被重新定义为限于与该术语相关联的任意特定的特征、特性征或方面。通常，除非具体实施方式中明确定义了这些术语，否则

不应将所附权利要求中使用的术语解释为将本技术限制于本文公开的特定示例。因此，该技术的实际范围不仅涵盖所公开的示例，而且涵盖实践或实现该技术的所有等效方式。

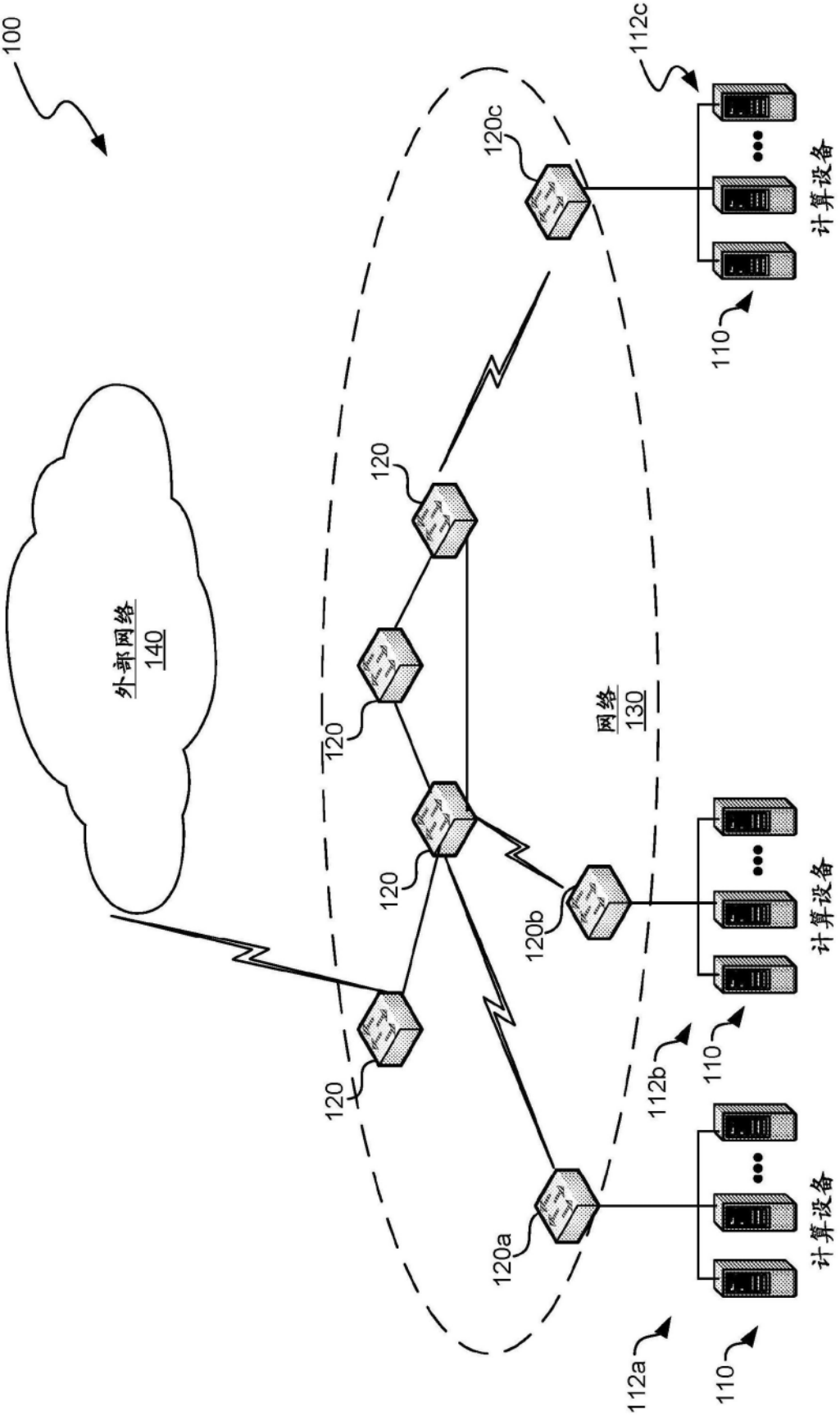

100
外部网络
140
网络
130
120
120
120
120
120a
120b
120c
110
110
110
112a
112b
112c
计算设备
计算设备
计算设备

图1

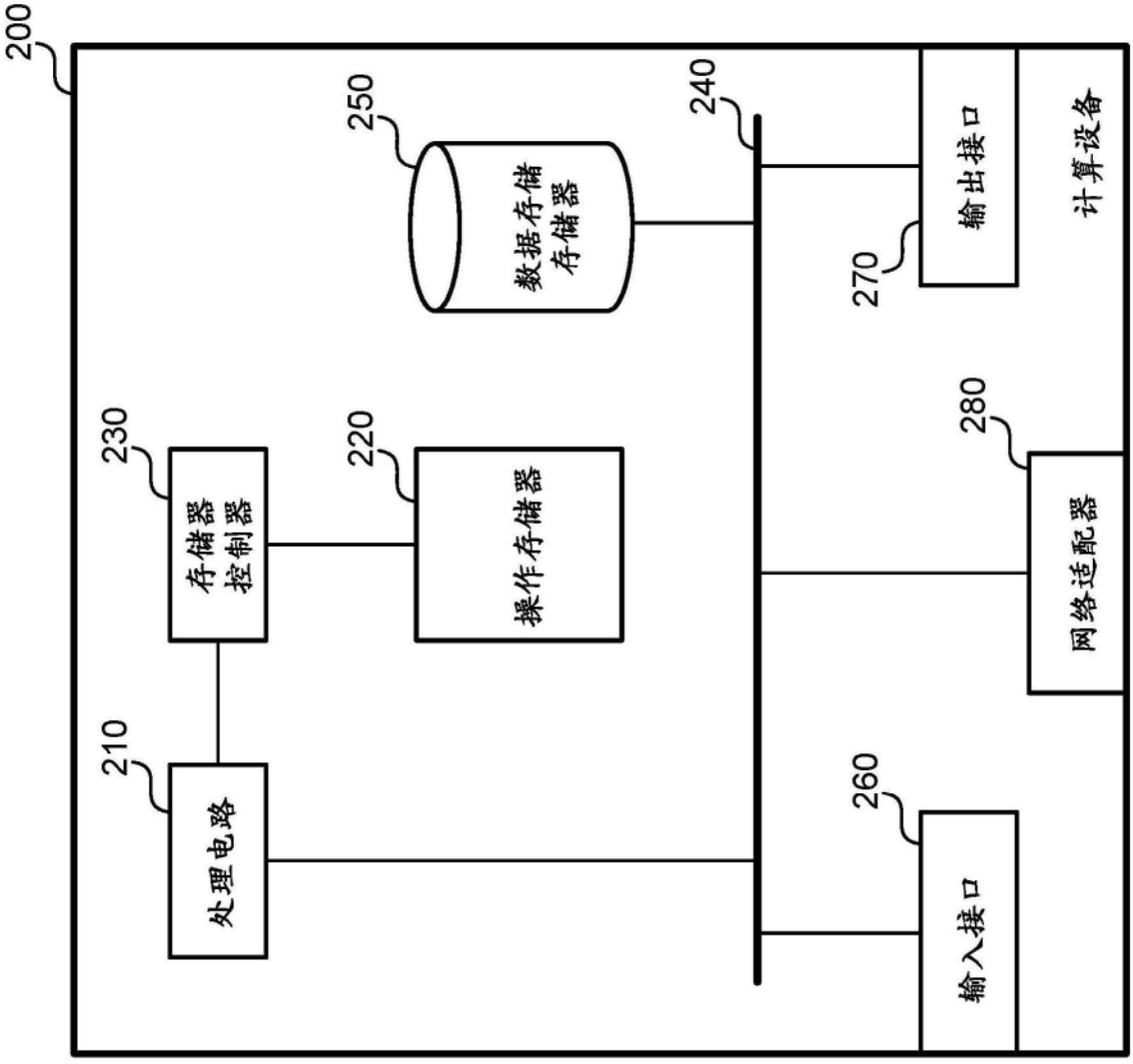

200
计算设备
250
数据存储存储器
240
输出接口
270
230
存储器控制器
220
操作存储器
280
网络适配器
210
处理电路
260
输入接口

图2

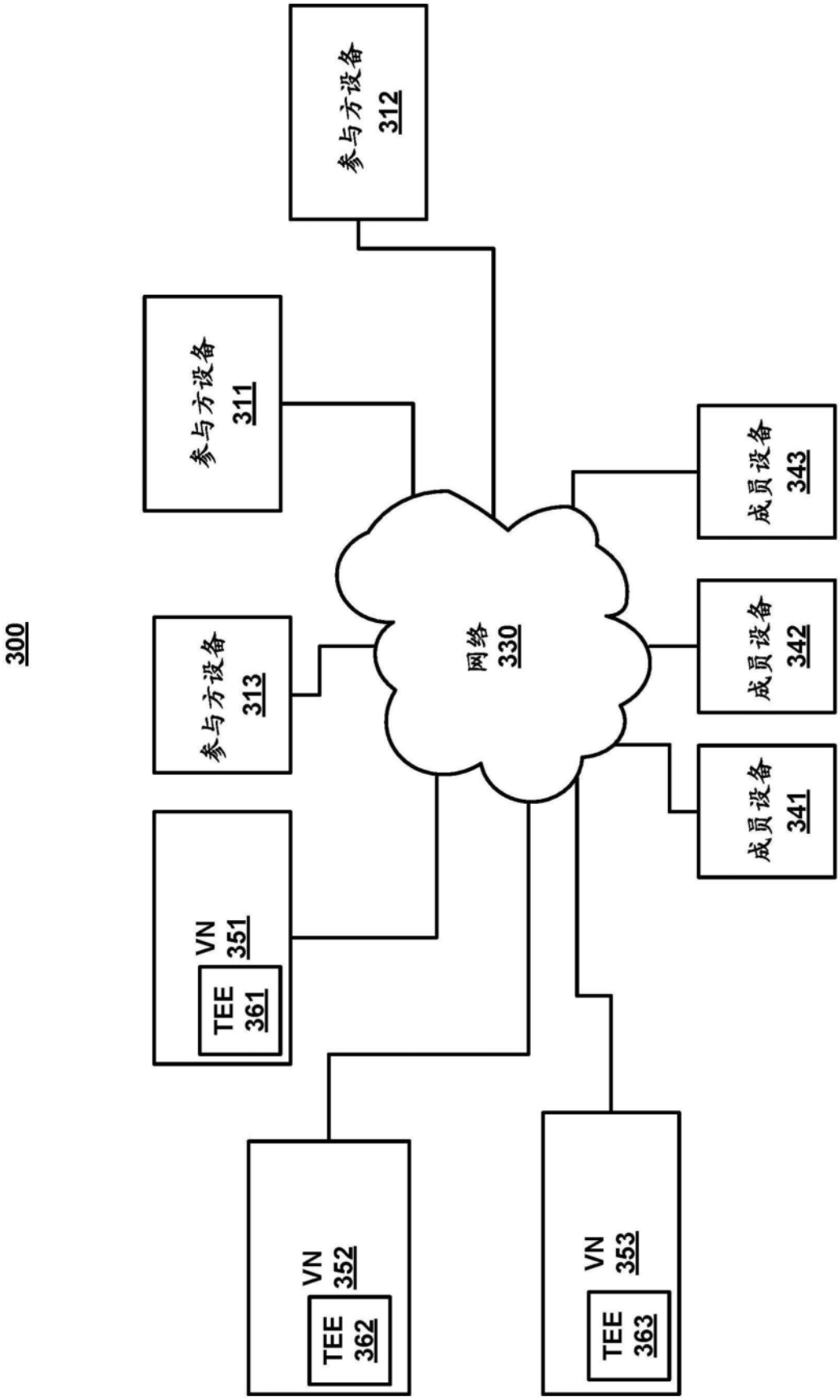

300
参与方设备 312
参与方设备 311
参与方设备 313
网络 330
成员设备 343
成员设备 342
成员设备 341
VN 351
TEE 361
VN 352
TEE 362
VN 353
TEE 363

图3

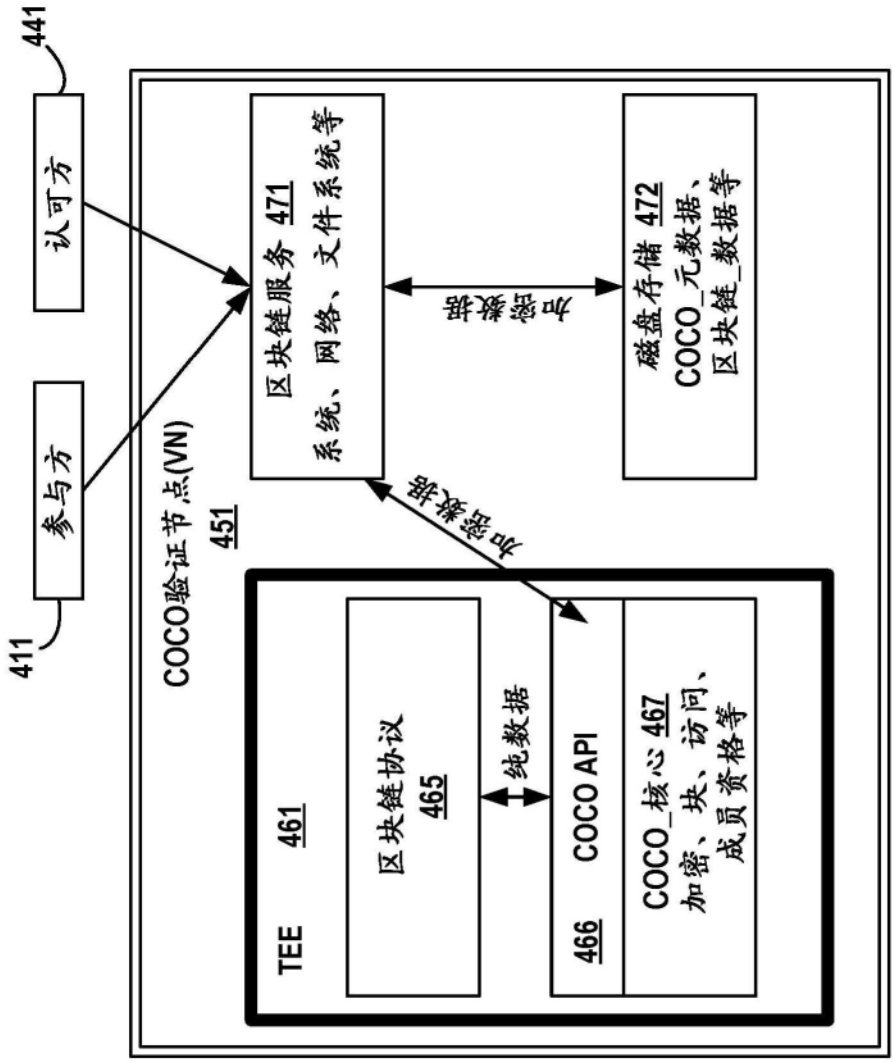

441
认可方
411
参与方
COCO验证节点(VN)
451
区块链服务 471
系统、网络、文件系统等
磁盘存储 472
COCO_元数据、
区块链_数据等
TEE 461
区块链协议
465
纯数据
466 COCO API
COCO_核心 467
加密、块、访问、
成员资格等

图4

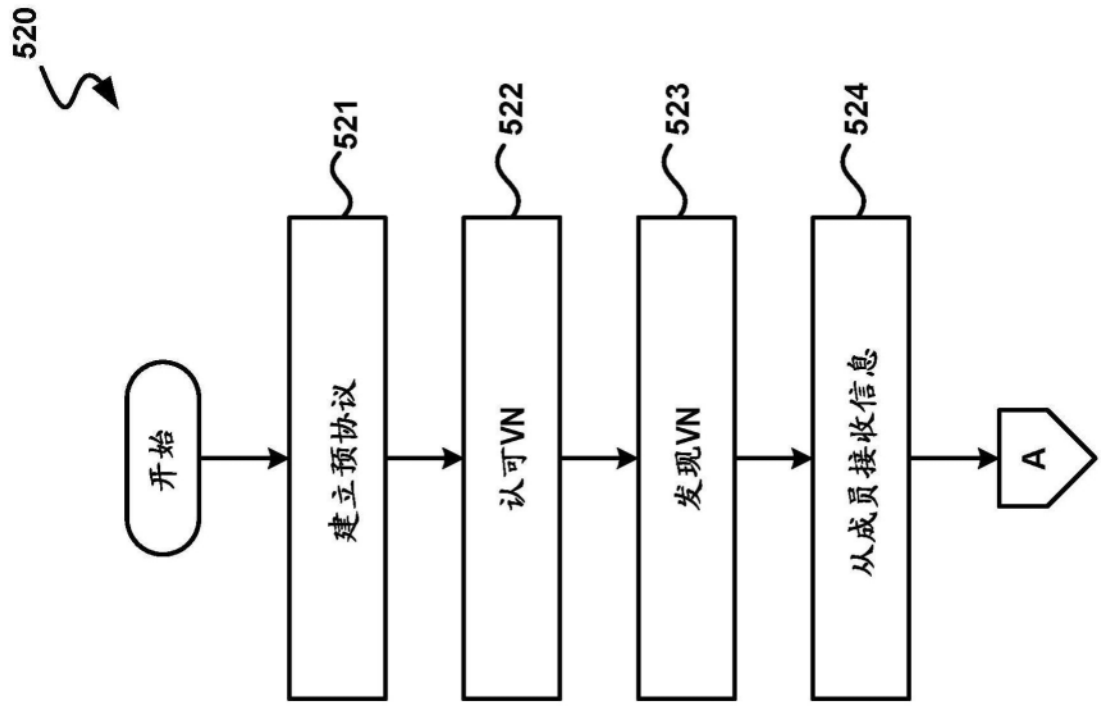

520
521
522
523
524
开始
建立预协议
认可VN
发现VN
从成员接收信息
A

图5A

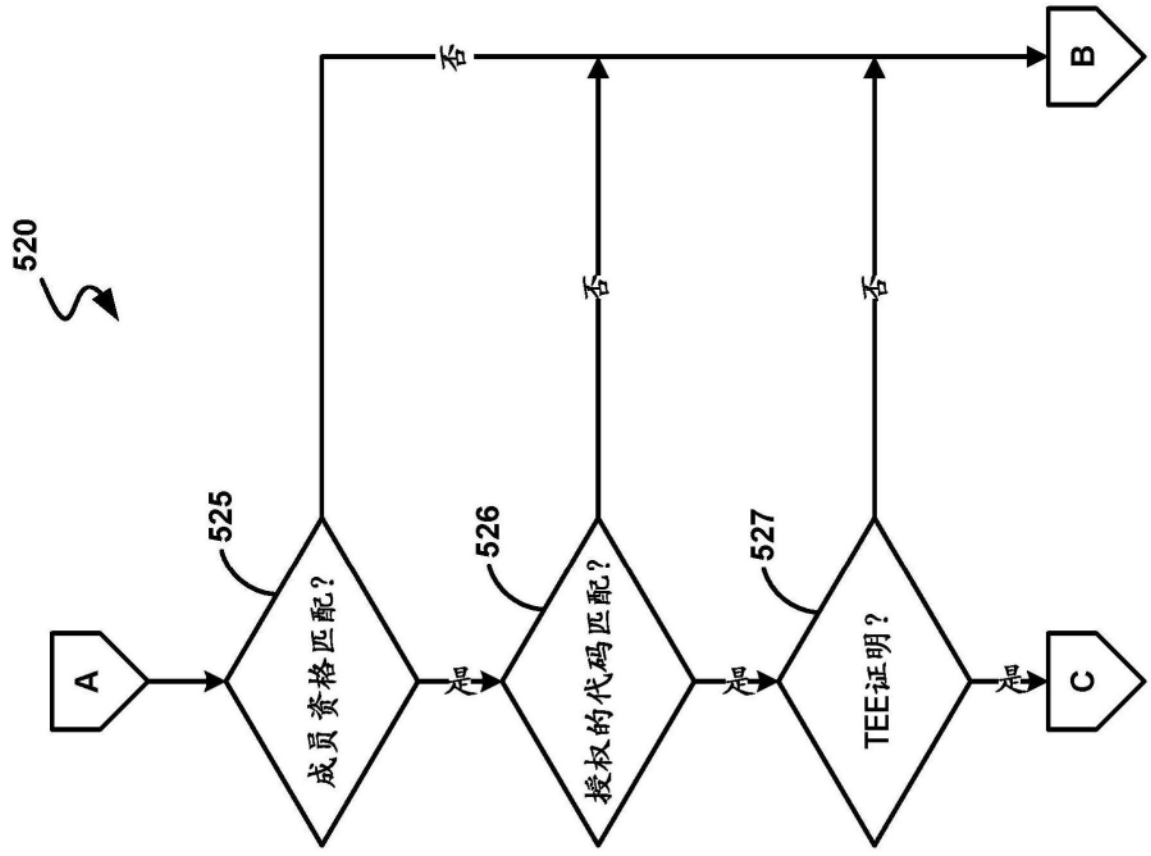

520
525
526
527
A
成员资格匹配?
授权的代码匹配?
TEE证明?
否
否
否
是
是
是
B
C

图5B

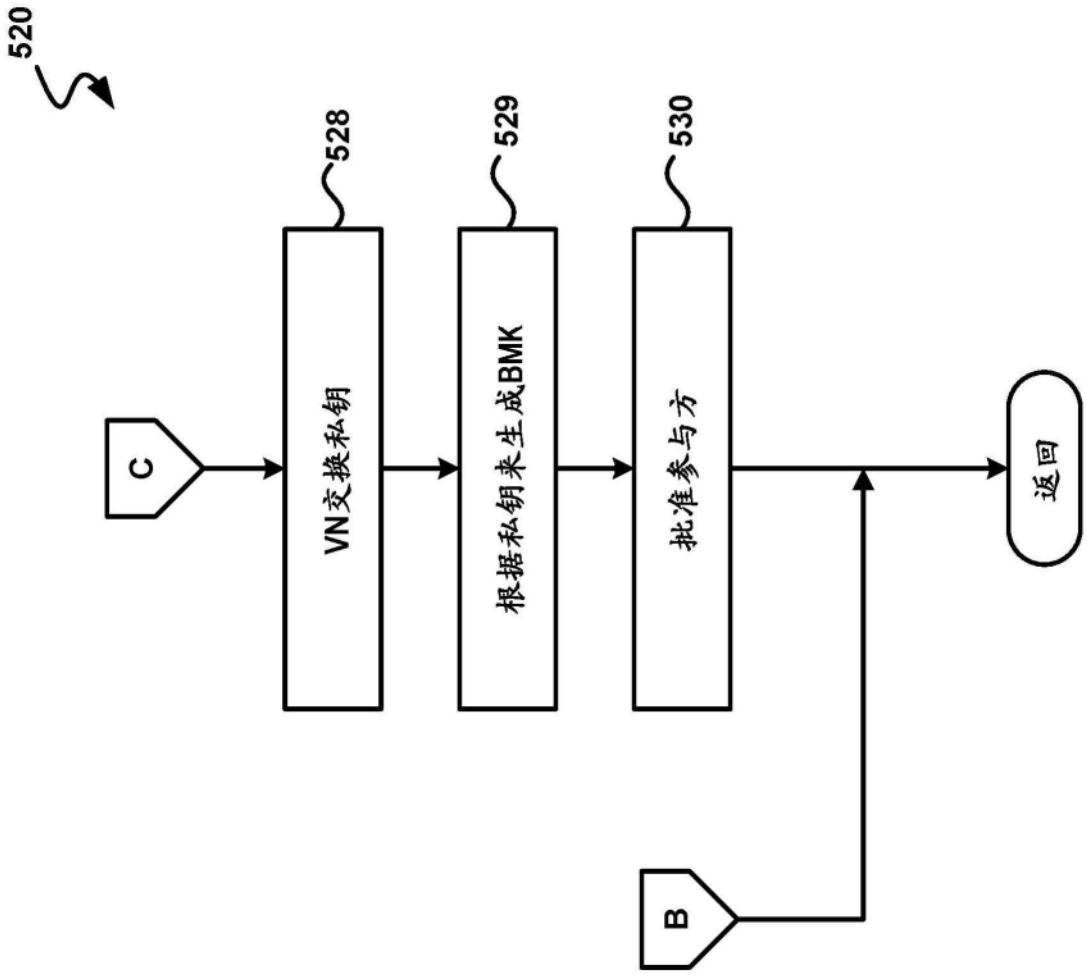


图5C

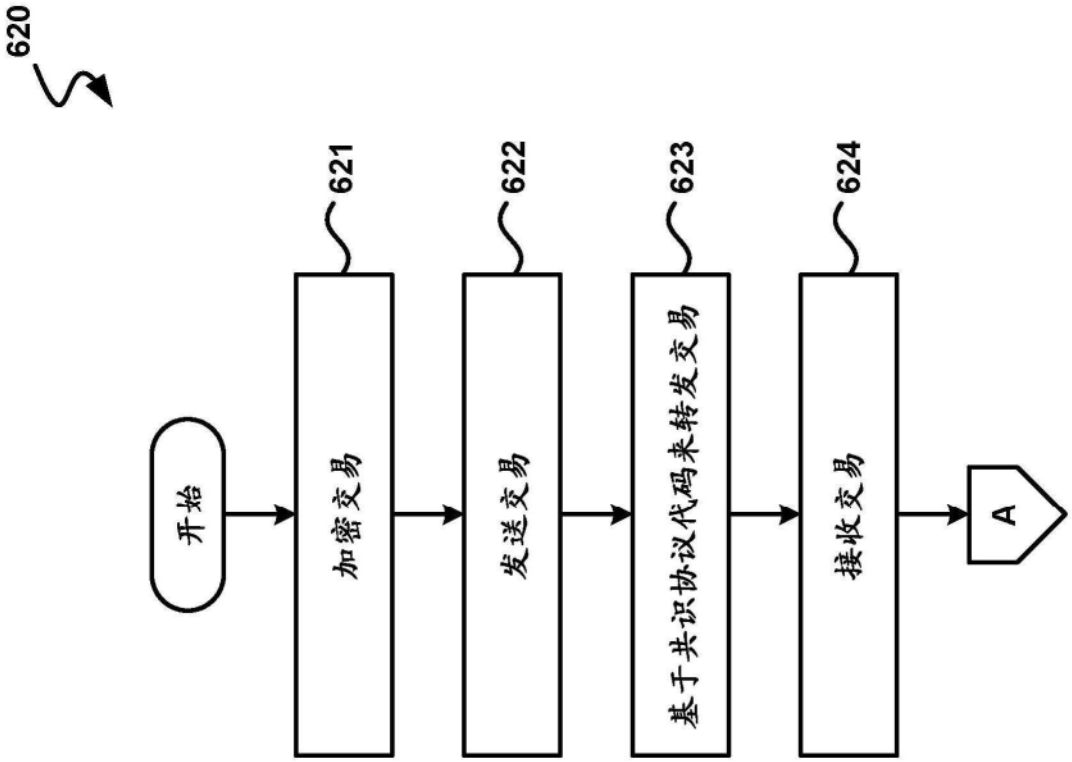

620
621
622
623
624
开始
加密交易
发送交易
基于共识协议代码来转发交易
接收交易
A

图6A

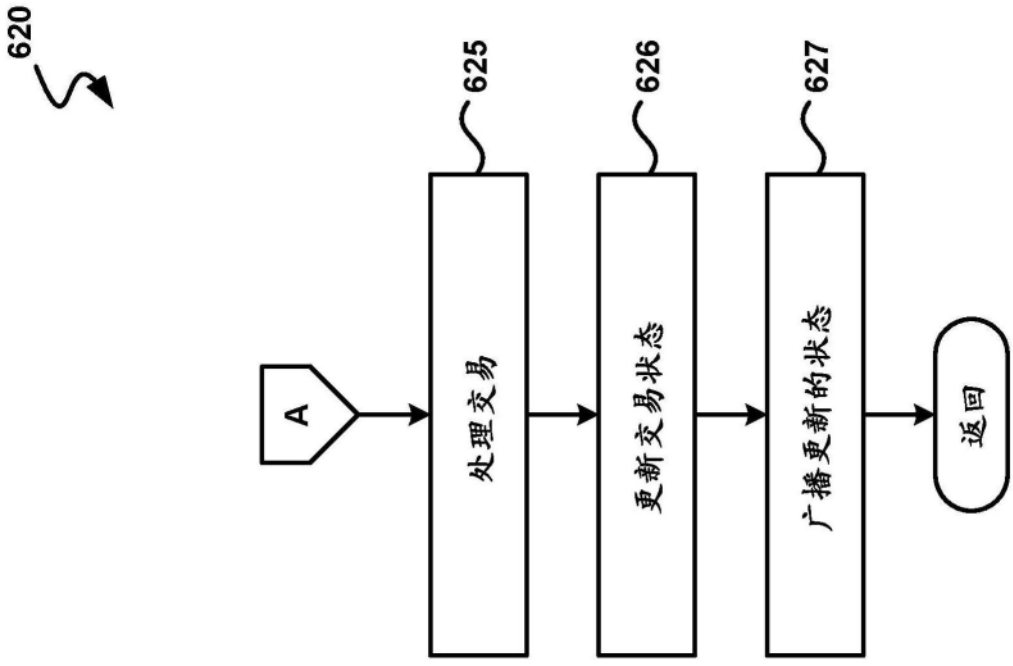

620
625
626
627
A
处理交易
更新交易状态
广播更新的状态
返回

图6B

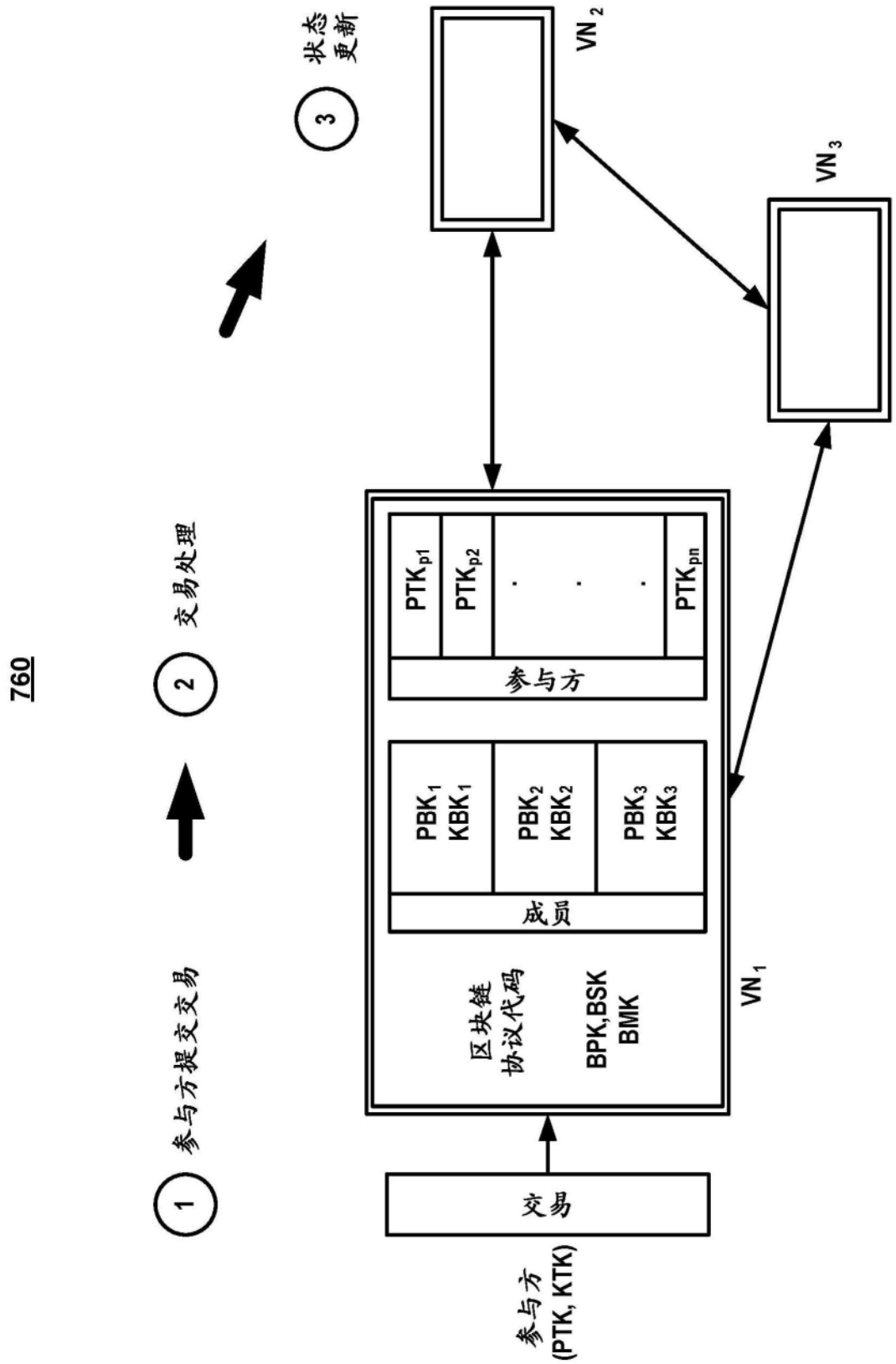

760
1 参与方提交交易
2 交易处理
3 状态更新
交易
参与方
(PTK, KTK)
区块链
协议代码
BPK,BSK
BMK
VN1
PBK1 KBK1
PBK2 KBK2
PBK3 KBK3
成员
PTKp1
PTKp2
PTKpn
参与方
VN2
VN3

图7

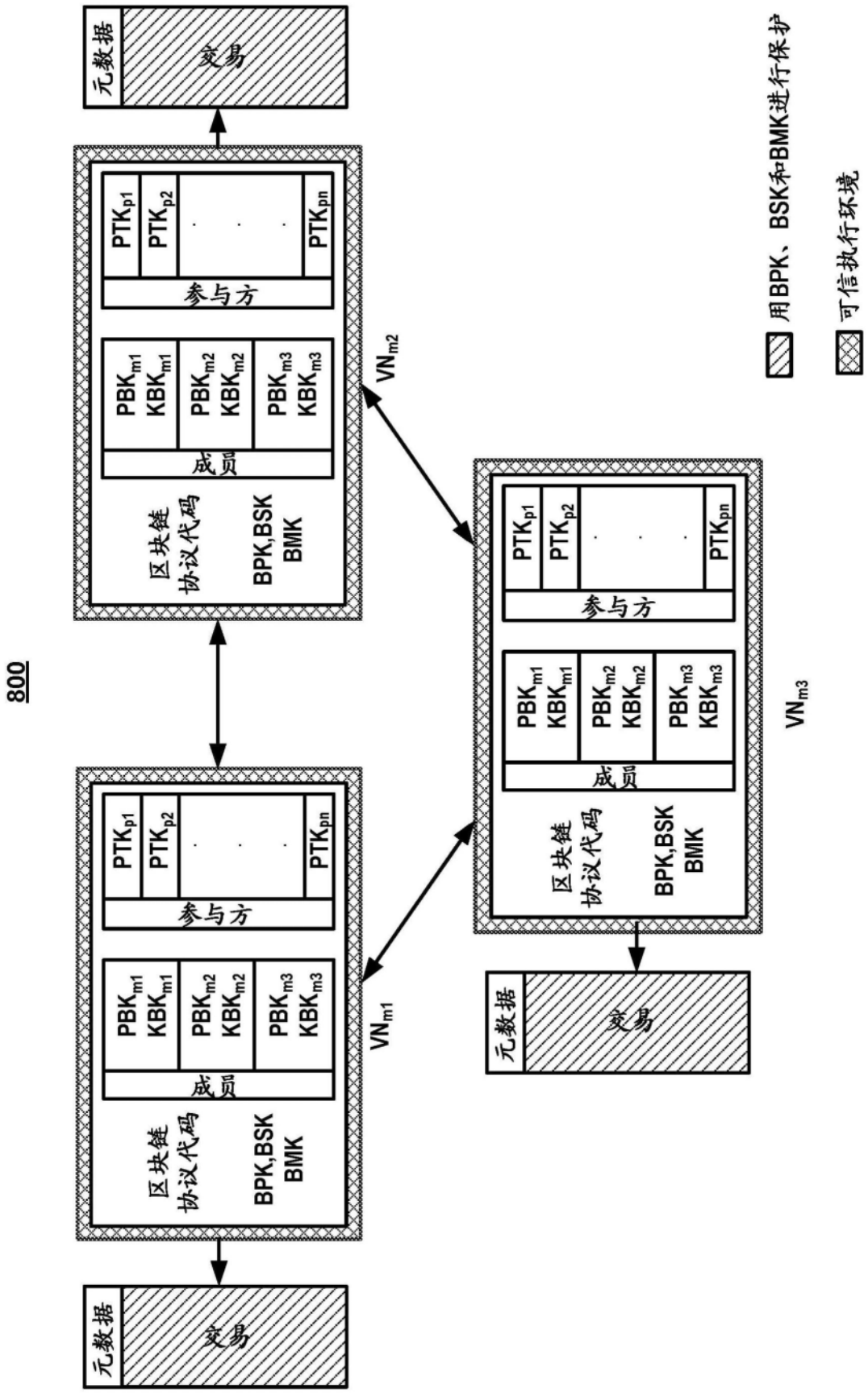

800
元数据
交易
参与方
PTK$_{p1}$
PTK$_{p2}$
PTK$_{pn}$
成员
PBK$_{m1}$ KBK$_{m1}$
PBK$_{m2}$ KBK$_{m2}$
PBK$_{m3}$ KBK$_{m3}$
区块链协议代码
BPK,BSK BMK
VN$_{m1}$
VN$_{m2}$
VN$_{m3}$
用BPK、BSK和BMK进行保护
可信执行环境

图8

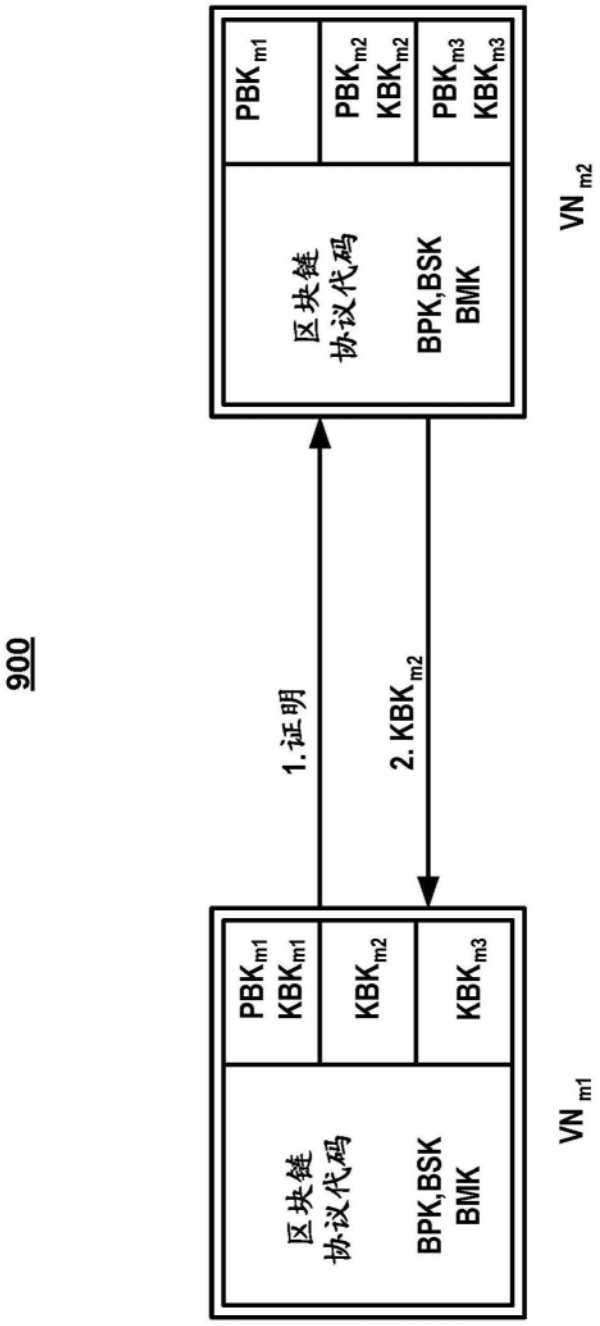

900
1.证明
2. KBK$_{m2}$
PBK$_{m1}$
PBK$_{m2}$ KBK$_{m2}$
PBK$_{m3}$ KBK$_{m3}$
区块链协议代码
BPK,BSK BMK
VN $_{m2}$
PBK$_{m1}$ KBK$_{m1}$
KBK$_{m2}$
KBK$_{m3}$
区块链协议代码
BPK,BSK BMK
VN $_{m1}$

图9

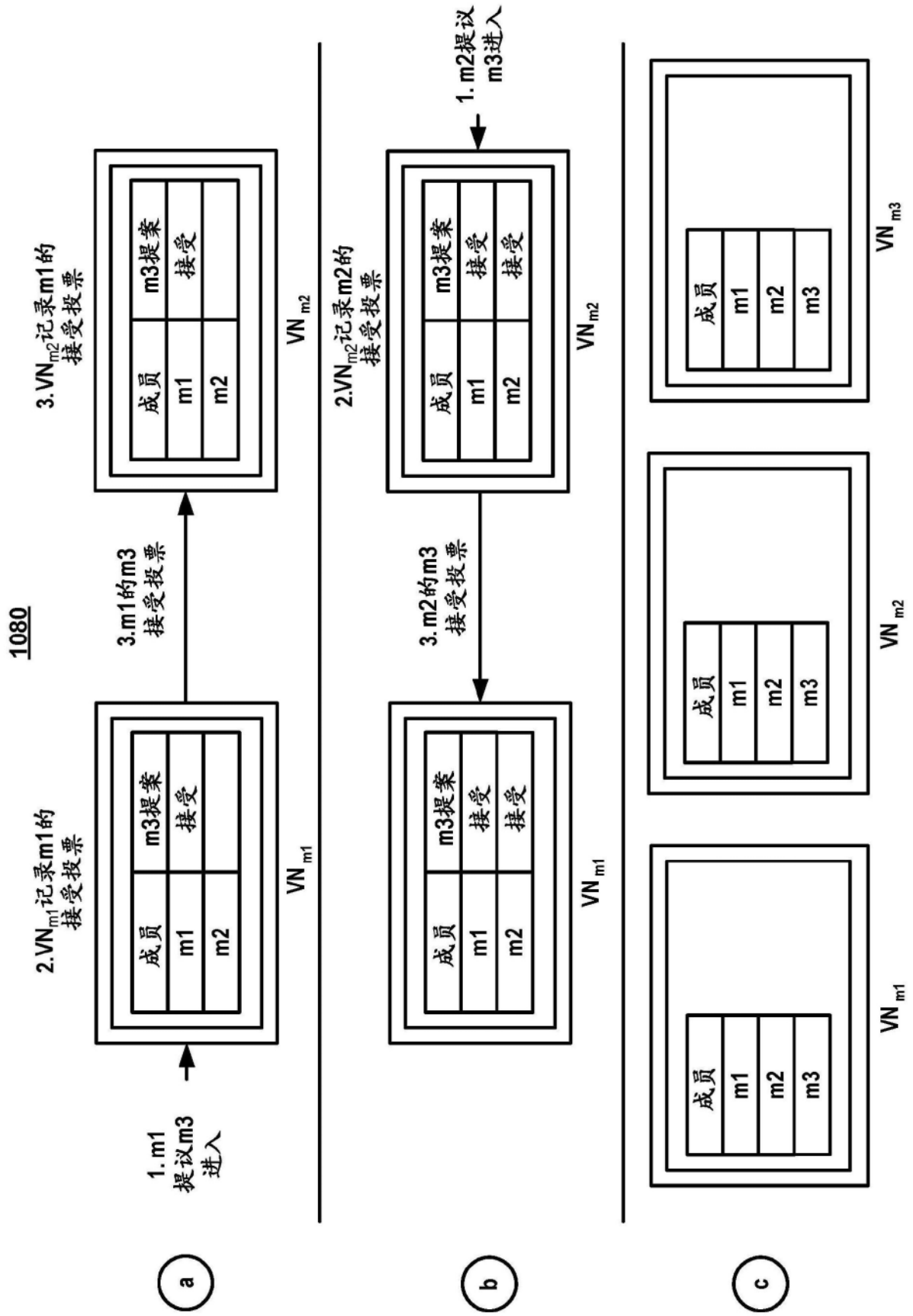

1080
1. m1 提议m3 进入
2. VN$_{m1}$记录m1的 接受投票
3. m1的m3 接受投票
3. VN$_{m2}$记录m1的 接受投票
VN$_{m1}$
VN$_{m2}$
成员
m1
m2
m3提案
接受
a
1. m2提议 m3进入
2. VN$_{m2}$记录m2的 接受投票
3. m2的m3 接受投票
b
VN$_{m3}$
m3
c

图10

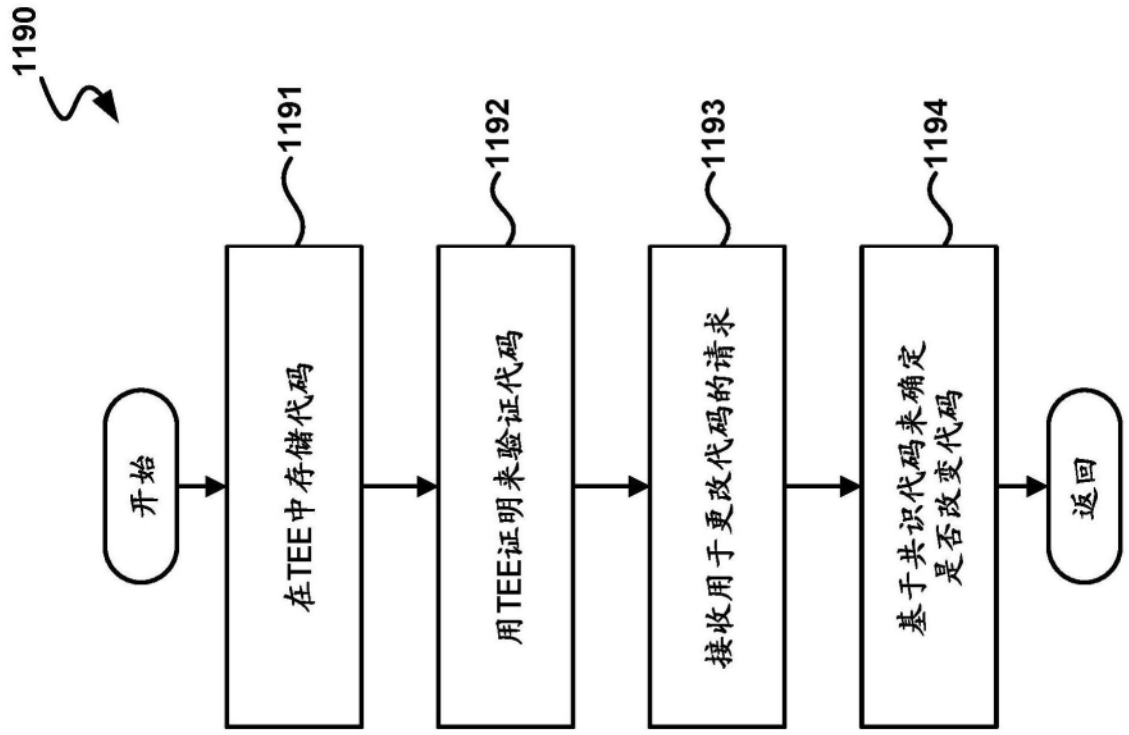

1190
1191
1192
1193
1194
开始
在TEE中存储代码
用TEE证明来验证代码
接收用于更改代码的请求
基于共识代码来确定是否改变代码
返回

图11

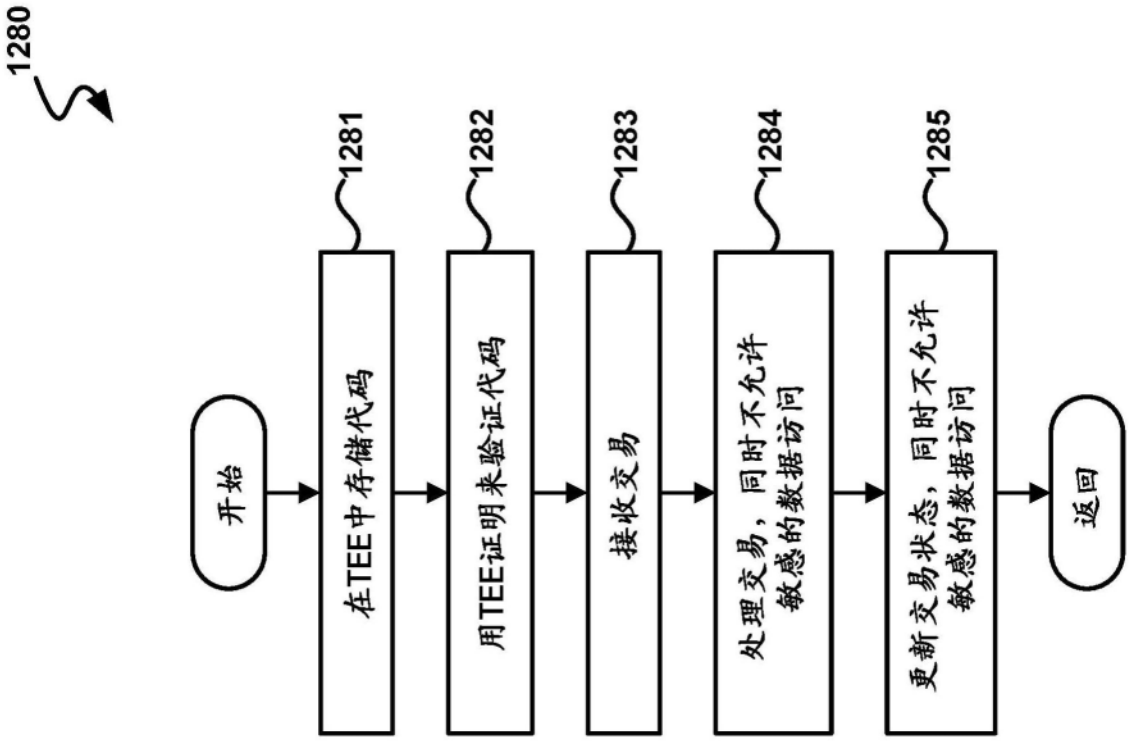

1280
1281
1282
1283
1284
1285
开始
在TEE中存储代码
用TEE证明来验证代码
接收交易
处理交易，同时不允许敏感的数据访问
更新交易状态，同时不允许敏感的数据访问
返回

图12