(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110730963 B
(45)授权公告日 2023.12.01

(21)申请号 201880005138.7

(22)申请日 2018.11.27

(65)同一申请的已公布的文献号
申请公布号 CN 110730963 A

(43)申请公布日 2020.01.24

(85)PCT国际申请进入国家阶段日
2019.06.18

(86)PCT国际申请的申请数据
PCT/CN2018/117548 2018.11.27

(87)PCT国际申请的公布数据
WO2019/072275 EN 2019.04.18

(73)专利权人 创新先进技术有限公司
地址 开曼群岛大开曼岛乔治镇医院路27号开曼企业中心

(72)发明人 马宝利 张文彬 李漓春 刘正殷山

(74)专利代理机构 北京博思佳知识产权代理有限公司 11415
专利代理师 林祥

(51)Int.Cl.
G06F 21/55(2013.01)
H04L 9/32(2006.01)

(56)对比文件
CN 108021821 A,2018.05.11
US 2018211313 A1,2018.07.26
CN 108667599 A,2018.10.16
US 2016358165 A1,2016.12.08
查选;王旭;倪巍;刘仁平;郭英杰;钮心忻;郑康锋.区块链技术的一致性和容量的研究与发展及在物联网中的应用.物联网学报.2017,(第01期),33-45. (续)

审查员 胡振洲

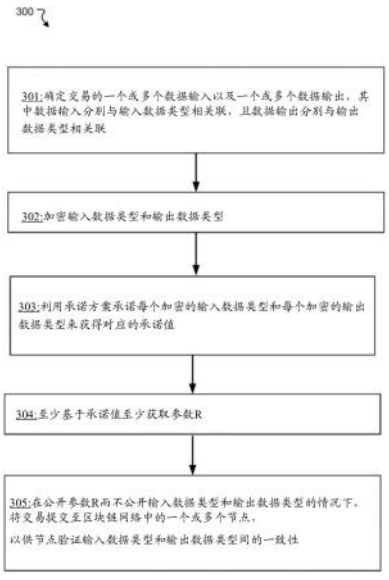
权利要求书5页 说明书14页 附图5页

(54)发明名称

用于信息保护的系统和方法

(57)摘要

一种计算机实现的用于信息保护的方法包括:确定用于交易的一个或多个数据输入及一个或多个数据输出,其中,所述数据输入分别与输入数据类型相关联,且所述数据输出分别与输出数据类型相关联;加密所述输入数据类型和所述输出数据类型;利用承诺方案承诺每个所述加密的输入数据类型和每个所述加密的输出数据类型来获取对应的承诺值;至少基于所述承诺值至少获取参数R;以及在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下,将所述交易提交至区块链网络中的一个或多个节点,以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性。



CN 110730963 B

[转续页]

[接上页]

(56)对比文件

Bin Li et al..RZKPB: A Privacy-Preserving Blockchain-Based Fair Transaction Method for Sharing Economy.2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/ 12th IEEE International Conference On Big Data Science And Engineering (TrustCom/ BigDataSE).2018,全文.

1.一种计算机实现的用于信息保护的方法,包括:

确定用于交易的一个或多个数据输入及一个或多个数据输出,其中,所述数据输入分别与输入数据类型相关联,且所述数据输出分别与输出数据类型相关联;

加密所述输入数据类型和所述输出数据类型;

利用承诺方案承诺每个所述加密的输入数据类型和每个所述加密的输出数据类型来获取对应的承诺值;

至少基于所述承诺值至少获取参数R;以及

在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下,将所述交易提交至区块链网络中的一个或多个节点,以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性;

其中,在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下,将所述交易提交至所述区块链网络中的所述一个或多个节点,以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性,包括:

在公开所述参数R而不公开所述输入数据类型和所述输出数据类型的情况下,将所述交易提交至所述区块链网络中的所述一个或多个节点,以促使所述节点:

获取所述参数R和基点G;

获取所述承诺值的对之间的差值;

级联所述获取的差值;

利用哈希函数加密所述级联的差值以获取加密值x;

至少基于所述获取的差值和所述加密值x获取多项式的总和C;

响应于确定所述总和C等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型一致;以及

响应于确定所述总和C不等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型不一致。

2.如权利要求1所述的方法,其中:加密所述输入数据类型和所述输出数据类型,包括:

利用哈希函数加密所述输入数据类型和所述输出数据类型。

3.如权利要求1所述的方法,其中:所述承诺方案包括佩德森承诺。

4.如权利要求1所述的方法,其中:

所述承诺方案至少包括盲因子;且

所述盲因子随对所述加密的输入数据类型和所述加密的输出数据类型进行承诺的时间而变化。

5.如权利要求1所述的方法,其中:使所述节点在不知道所述输入数据类型和所述输出数据类型的情况下验证所述输入数据类型和所述输出数据类型间的一致性。

6.如权利要求1所述的方法,其中:

所述交易是至少基于未被花费的交易输出UTXO模型的;且

所述数据输入和所述数据输出包括正在进行所述交易的一个或多个资产的类型。

7.如权利要求1所述的方法,其中:

所述承诺方案包括分别对应于所述输入数据类型和所述输出数据类型的多个盲因子;以及

至少基于所述承诺值至少获取所述参数R包括：

获取成对的所述承诺值之间的差值；

级联所述获取的差值；

利用哈希函数加密所述级联的差值以获取加密值x；以及

至少基于所述加密值x以及成对的所述盲因子之间的差值获取所述参数R。

8.一种非暂态计算机可读存储介质，其存储将由处理器执行来促使所述处理器执行操作的指令，所述操作包括：

确定用于交易的一个或多个数据输入及一个或多个数据输出，其中，所述数据输入分别与输入数据类型相关联，且所述数据输出分别与输出数据类型相关联；

加密所述输入数据类型和所述输出数据类型；

利用承诺方案承诺每个所述加密的输入数据类型和每个所述加密的输出数据类型来获取对应的承诺值；

至少基于所述承诺值至少获取参数R；以及

在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下，将所述交易提交至区块链网络中的一个或多个节点，以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性；

其中，在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下，将所述交易提交至所述区块链网络中的所述一个或多个节点，以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性，包括：

在公开所述参数R而不公开所述输入数据类型和所述输出数据类型的情况下，将所述交易提交至所述区块链网络中的所述一个或多个节点，以促使所述节点：

获取所述参数R和基点G；

获取所述承诺值的对之间的差值；

级联所述获取的差值；

利用哈希函数加密所述级联的差值以获取加密值x；

至少基于所述获取的差值和所述加密值x获取多项式的总和C；

响应于确定所述总和C等于所述参数R和所述基点G的积，确定所述输入数据类型和所述输出数据类型一致；以及

响应于确定所述总和C不等于所述参数R和所述基点G的积，确定所述输入数据类型和所述输出数据类型不一致。

9.如权利要求8所述的存储介质，其中：加密所述输入数据类型和所述输出数据类型，包括：

利用哈希函数加密所述输入数据类型和所述输出数据类型。

10.如权利要求8所述的存储介质，其中：所述承诺方案包括佩德森承诺。

11.如权利要求8所述的存储介质，其中：

所述承诺方案至少包括盲因子；且

所述盲因子随对所述加密的输入数据类型和所述加密的输出数据类型进行承诺的时间而变化。

12.如权利要求8所述的存储介质，其中：使所述节点在不知道所述输入数据类型和所

述输出数据类型的情况下验证所述输入数据类型和所述输出数据类型间的一致性。

13.如权利要求8所述的存储介质,其中:

所述交易是至少基于未被花费的交易输出UTXO模型的;且

所述数据输入和所述数据输出包括正在进行所述交易的一个或多个资产的类型。

14.如权利要求8所述的存储介质,其中:

所述承诺方案包括分别对应于所述输入数据类型和所述输出数据类型的多个盲因子;以及

至少基于所述承诺值至少获取所述参数R包括:

获取成对的所述承诺值之间的差值;

级联所述获取的差值;

利用哈希函数加密所述级联的差值以获取加密值x;以及至少基于所述加密值x以及成对的所述盲因子之间的差值获取所述参数R。

15.一种用于信息保护的系统,包括处理器和耦接至所述处理器的非暂态计算机可读存储介质,所述存储介质存储将由所述处理器执行以促使所述系统执行操作的指令,所述操作包括:

确定用于交易的一个或多个数据输入及一个或多个数据输出,其中,所述数据输入分别与输入数据类型相关联,且所述数据输出分别与输出数据类型相关联;

加密所述输入数据类型和所述输出数据类型;

利用承诺方案承诺每个所述加密的输入数据类型和每个所述加密的输出数据类型来获取对应的承诺值;

至少基于所述承诺值至少获取参数R;以及

在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下,将所述交易提交至区块链网络中的一个或多个节点,以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性;

其中,在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下,将所述交易提交至所述区块链网络中的所述一个或多个节点,以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性,包括:

在公开所述参数R而不公开所述输入数据类型和所述输出数据类型的情况下,将所述交易提交至所述区块链网络中的所述一个或多个节点,以促使所述节点:

获取所述参数R和基点G;

获取所述承诺值的对之间的差值;

级联所述获取的差值;

利用哈希函数加密所述级联的差值以获取加密值x;

至少基于所述获取的差值和所述加密值x获取多项式的总和C;

响应于确定所述总和C等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型一致;以及

响应于确定所述总和C不等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型不一致。

16.一种计算机实现的用于信息保护的方法,包括:

区块链网络中的一个或多个节点获取由发起者节点发起的交易,其中:

所述交易与一个或多个数据输入及一个或多个数据输出相关联,

所述数据输入分别与输入数据类型相关联,且所述数据输出分别与输出数据类型相关联,

所述输入数据类型和所述输出数据类型被加密且被交给承诺方案以获取对应的承诺值,以及

所述输入数据类型和所述输出数据类型不对所述一个或多个节点公开;

所述一个或多个节点验证所述输入数据类型和所述输出数据类型间的一致性;

响应于确定所述输入数据类型和所述输出数据类型一致,所述一个或多个节点将所述交易添加至所述区块链网络中;以及

响应于确定所述输入数据类型和所述输出数据类型不一致,所述一个或多个节点拒绝将所述交易添加至所述区块链网络中;

其中,验证所述输入数据类型和所述输出数据类型间的一致性,包括:获取参数R和基点G;

获取成对的所述承诺值之间的差值;

级联所述获取的差值;

利用哈希函数加密所述级联的差值以获取加密值x;

至少基于所述获取的差值和所述加密值x获取多项式的总和C;

确定所述总和C是否等于所述参数R和所述基点G的积。

17.如权利要求16所述的方法,还包括:

响应于确定所述总和C等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型一致;以及

响应于确定所述总和C不等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型不一致。

18.一种非暂态计算机可读存储介质,其存储将由处理器执行来促使所述处理器执行操作的指令,所述操作包括:

区块链网络中的一个或多个节点获取由发起者节点发起的交易,其中:

所述交易与一个或多个数据输入及一个或多个数据输出相关联,

所述数据输入分别与输入数据类型相关联,且所述数据输出分别与输出数据类型相关联,

所述输入数据类型和所述输出数据类型被加密且被交给承诺方案以获取对应的承诺值,以及

所述输入数据类型和所述输出数据类型不对所述一个或多个节点公开;

所述一个或多个节点验证所述输入数据类型和所述输出数据类型间的一致性;

响应于确定所述输入数据类型和所述输出数据类型一致,所述一个或多个节点将所述交易添加至所述区块链网络中;以及

响应于确定所述输入数据类型和所述输出数据类型不一致,所述一个或多个节点拒绝将所述交易添加至所述区块链网络中;

其中,验证所述输入数据类型和所述输出数据类型间的一致性,包括:获取参数R和基

点G；

获取成对的所述承诺值之间的差值；

级联所述获取的差值；

利用哈希函数加密所述级联的差值以获取加密值x；

至少基于所述获取的差值和所述加密值x获取多项式的总和C；

确定所述总和C是否等于所述参数R和所述基点G的积。

19.一种用于信息保护的系统，包括处理器和耦接至所述处理器的非暂态计算机可读存储介质，所述存储介质存储将由所述处理器执行的指令以促使所述系统执行操作，所述操作包括：

区块链网络中的一个或多个节点获取由发起者节点发起的交易，其中：

所述交易与一个或多个数据输入及一个或多个数据输出相关联，

所述数据输入分别与输入数据类型相关联，且所述数据输出分别与输出数据类型相关联，

所述输入数据类型和所述输出数据类型被加密且被交给承诺方案以获取对应的承诺值，以及

所述输入数据类型和所述输出数据类型不对所述一个或多个节点公开；

所述一个或多个节点验证所述输入数据类型和所述输出数据类型间的一致性；

响应于确定所述输入数据类型和所述输出数据类型一致，所述一个或多个节点将所述交易添加至所述区块链网络中；以及

响应于确定所述输入数据类型和所述输出数据类型不一致，所述一个或多个节点拒绝将所述交易添加至所述区块链网络中；

其中，验证所述输入数据类型和所述输出数据类型间的一致性，包括：获取参数R和基点G；

获取成对的所述承诺值之间的差值；

级联所述获取的差值；

利用哈希函数加密所述级联的差值以获取加密值x；

至少基于所述获取的差值和所述加密值x获取多项式的总和C；

确定所述总和C是否等于所述参数R和所述基点G的积。

用于信息保护的系统和方法

技术领域

[0001] 本发明总体上涉及用于信息保护的方法和设备。

背景技术

[0002] 隐私对于各种用户之间的通信和数据传送是重要的。在没有保护的情况下，用户暴露于身份盗窃、非法转账或其它潜在损失的风险中。当在线实现通信和传送时，由于对在线信息的自由访问，风险甚至变得更大。

发明内容

[0003] 本公开的各种实施例包括用于信息保护的系统、方法和非暂态计算机可读介质。

[0004] 根据一方面，一种计算机实现的用于信息保护的方法包括：确定用于交易的一个或多个数据输入及一个或多个数据输出，其中，所述数据输入分别与输入数据类型相关联，且所述数据输出分别与输出数据类型相关联；加密所述输入数据类型和所述输出数据类型；利用承诺方案承诺每个所述加密的输入数据类型和每个所述加密的输出数据类型来获取对应的承诺值；至少基于所述承诺值至少获取参数R；以及在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下，将所述交易提交至区块链网络中的一个或多个节点，以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性。

[0005] 在一些实施例中，加密所述输入数据类型和所述输出数据类型，包括：利用哈希函数加密所述输入数据类型和所述输出数据类型。

[0006] 在一些实施例中，所述承诺方案包括佩德森(Pedersen)承诺。

[0007] 在一些实施例中，所述承诺方案至少包括盲因子；且所述盲因子随对所述加密的输入数据类型和所述加密的输出数据类型进行承诺的时间而变化。

[0008] 在一些实施例中，使所述节点在不知道所述输入数据类型和所述输出数据类型的情况下验证所述输入数据类型和所述输出数据类型间的一致性。

[0009] 在一些实施例中，所述交易是至少基于未被花费的交易输出(UTXO)模型的；且所述数据输入和所述数据输出包括正在进行所述交易的一个或多个资产的类型。

[0010] 在一些实施例中，所述承诺方案包括分别对应于所述输入数据类型和所述输出数据类型的多个盲因子；以及至少基于所述承诺值至少获取所述参数R包括：获取成对的所述承诺值之间的差值；级联(concatenation)所述获取的差值；利用哈希函数加密所述级联的差值以获取加密值x；以及至少基于加密值x以及成对的所述盲因子之间的差值获取所述参数R。

[0011] 在一些实施例中，在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下，将所述交易提交至所述区块链网络中的所述一个或多个节点，以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性，包括在公开所述参数R而不公开所述输入数据类型和所述输出数据类型的情况下，将所述交易提交至所述区块链网络中的所述一个或多个节点以促使所述节点：获取所述参数R和基点G；获取成对的所述承诺值

之间的差值;级联所述获取的差值;利用哈希函数加密所述级联的差值以获取加密值x;至少基于所述获取的差值和所述加密值x获取多项式的总和C;响应于确定所述总和C等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型一致;以及响应于确定所述总和C不等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型不一致。

[0012] 根据另一方面,一种非暂态计算机可读存储介质,其存储将由处理器执行来促使所述处理器执行操作的指令,所述操作包括:确定用于交易的一个或多个数据输入及一个或多个数据输出,其中,所述数据输入分别与输入数据类型相关联,且所述数据输出分别与输出数据类型相关联;加密所述输入数据类型和所述输出数据类型;利用承诺方案承诺每个所述加密的输入数据类型和每个所述加密的输出数据类型来获取对应的承诺值;至少基于所述承诺值至少获取参数R;以及在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下,将所述交易提交至区块链网络中的一个或多个节点以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性。

[0013] 根据另一方面,一种用于信息保护的系统,包括处理器和耦接至所述处理器的非暂态计算机可读存储介质,所述存储介质存储将由所述处理器执行以促使所述系统执行操作的指令,所述操作包括:确定用于交易的一个或多个数据输入及一个或多个数据输出,其中,所述数据输入分别与输入数据类型相关联,且所述数据输出分别与输出数据类型相关联;加密所述输入数据类型和所述输出数据类型;利用承诺方案承诺每个所述加密的输入数据类型和每个所述加密的输出数据类型来获取对应的承诺值;至少基于所述承诺值至少获取参数R;以及在公开所述参数R且不公开所述输入数据类型和所述输出数据类型的情况下,将所述交易提交至区块链网络中的一个或多个节点以供所述节点验证所述输入数据类型和所述输出数据类型间的一致性。

[0014] 根据另一方面,一种计算机实现的用于信息保护的方法,包括:由区块链网络中的一个或多个节点获取由发起者节点发起的交易,其中:所述交易与一个或多个数据输入及一个或多个数据输出相关联,所述数据输入分别与输入数据类型相关联,且所述数据输出分别与输出数据类型相关联,所述输入数据类型和所述输出数据类型被加密且被交给承诺方案以获取对应的承诺值,以及所述输入数据类型和所述输出数据类型不对所述一个或多个节点公开;由所述一个或多个节点验证所述输入数据类型和所述输出数据类型间的一致性;响应于确定所述输入数据类型和所述输出数据类型一致,由所述一个或多个节点将所述交易添加至所述区块链网络中;以及响应于确定所述输入数据类型和所述输出数据类型不一致,由所述一个或多个节点拒绝将所述交易添加至所述区块链网络中。

[0015] 在一些实施例中,验证所述输入数据类型和所述输出数据类型间的一致性包括:获取参数R和基点G;获取成对的所述承诺值之间的差值;级联所述获取的差值;利用哈希函数加密所述级联的差值以获取加密值x;至少基于所述获取的差值和所述加密值x获取多项式的总和C;确定所述总和C是否等于所述参数R和所述基点G的积。

[0016] 在一些实施例中,所述方法还包括:响应于确定所述总和C等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型一致;以及响应于确定所述总和C不等于所述参数R和所述基点G的积,确定所述输入数据类型和所述输出数据类型不一致。

[0017] 在一些实施例中,所述一个或多个节点包括共识节点。

[0018] 根据另一方面，一种非暂态计算机可读存储介质，其存储将由处理器执行来促使所述处理器执行操作的指令，所述操作包括：由区块链网络中的一个或多个节点获取由发起者节点发起的交易，其中：所述交易与一个或多个数据输入及一个或多个数据输出相关联，所述数据输入分别与输入数据类型相关联，且所述数据输出分别与输出数据类型相关联，所述输入数据类型和所述输出数据类型被加密且被交给承诺方案以获取对应的承诺值，以及所述输入数据类型和所述输出数据类型不对所述一个或多个节点公开；由所述一个或多个节点验证所述输入数据类型和所述输出数据类型间的一致性；响应于确定所述输入数据类型和所述输出数据类型一致，由所述一个或多个节点将所述交易添加至所述区块链网络中；以及响应于确定所述输入数据类型和所述输出数据类型不一致，由所述一个或多个节点拒绝将所述交易添加至所述区块链网络中。

[0019] 根据另一方面，一种用于信息保护的系统，包括处理器和耦接至所述处理器的非暂态计算机可读存储介质，所述存储介质存储将由所述处理器执行的指令以促使所述系统执行操作，所述操作包括：由区块链网络中的一个或多个节点获取由发起者节点发起的交易，其中：所述交易与一个或多个数据输入及一个或多个数据输出相关联，所述数据输入分别与输入数据类型相关联，且所述数据输出分别与输出数据类型相关联，所述输入数据类型和所述输出数据类型被加密且被交给承诺方案以获取对应的承诺值，以及所述输入数据类型和所述输出数据类型不对所述一个或多个节点公开；由所述一个或多个节点验证所述输入数据类型和所述输出数据类型间的一致性；响应于确定所述输入数据类型和所述输出数据类型一致，由所述一个或多个节点将所述交易添加至所述区块链网络中；以及响应于确定所述输入数据类型和所述输出数据类型不一致，由所述一个或多个节点拒绝将所述交易添加至所述区块链网络中。

[0020] 在参考附图考虑以下描述和所附权利要求书之后，本文公开的系统、方法和非暂态计算机可读介质的这些和其它特征，以及操作方法和结构的相关元件的功能以及部件的组合和制造的经济性将变得更加明显，所有附图形成本说明书的一部分，其中，在各个附图中，同一的附图标记表示对应的部件。然而，应当清楚地理解，附图仅用于说明和描述的目的，而不意图作为对本发明的限制的定义。

附图说明

[0021] 本技术的各种实施例的某些特征在所附权利要求中具体阐述。通过参考以下阐述了利用本发明原理的示例性实施例的详细说明以及附图，将获得对本技术的特征和优点的更好理解，其中，

[0022] 图1示出了根据各种实施例的用于信息保护的示例性系统。

[0023] 图2示出了根据各种实施例的交易发起和验证的示例性步骤。

[0024] 图3示出了根据各种实施例的用于信息保护的示例性方法的流程图。

[0025] 图4示出了根据各种实施例的用于信息保护的示例性方法的流程图。

[0026] 图5示出了可实现本文所描述的任何实施例的示例性计算机系统的框图。

具体实施方式

[0027] 因为操作由网络中的各个节点(例如，计算设备)执行，区块链可以被认为是去中

心化的数据库，通常被称为分布式账本。任何信息都可以被写入并被保存至区块链中或从区块链被读取。任何人都可以建立服务器并作为节点加入区块链网络。任何节点都可以通过执行诸如哈希计算的复杂计算以向当前区块链添加区块，从而对维持区块链贡献计算能力，并且所添加的区块可以包含各种类型的数据或信息。可以用代币(例如，数字货币单位)对为所添加的区块贡献计算能力的节点进行奖励。由于区块链没有中心节点，所以每个节点是等同的，并且保存整个区块链数据库。

[0028] 节点是例如支持区块链网络并保持其平稳运行的计算设备或大型计算机系统。节点可以由贡献金钱购买被称为挖矿装备的强力计算机系统的个人或一群人运行。有两种类型的节点，全量节点和轻量节点。全量节点保存区块链的完整副本。区块链网络上的全量节点验证它们接收的交易和区块，并将这些交易和区块中继到连接的对等点，以提供交易的共识验证。另一方面，轻量节点仅下载区块链的一小部分。例如，轻量节点用于数字货币交易。当轻量节点想要进行交易时，轻量节点将与全量节点通信。

[0029] 这种去中心化属性可以帮助防止处于控制位置的管理中心出现。例如，比特币区块链的维护是由运行区域中(装有)比特币软件的通信节点组成的网络执行。也就是说，取代传统意义上的银行、机构或管理员，多个中间角色以执行比特币软件的计算机服务器的形式存在。这些计算机服务器形成经由因特网连接的网络，其中，任何人都可能加入该网络。网络所容纳的交易可以具有以下形式：“用户A想要将Z个比特币发送给用户B”，其中，使用容易获得的软件应用程序将交易广播至网络。计算机服务器用作比特币服务器，该比特币服务器可操作以验证这些金融交易，并将这些金融交易的记录添加至其账本的副本中，接着向网络的其它服务器广播这些账本添加操作。

[0030] 维护区块链被称为“挖矿”，并且进行这种维护的人被用如上所述的新创造的比特币和交易费进行奖励。例如，节点可以基于区块链网络已经同意的一组规则确定交易是否有效。矿工可位于任何大陆上，并通过验证每一交易是有效的以及将该交易添加至区块链中来处理支付。通过由多个矿工提供的共识实现这种验证，并且假定不存在系统串通。最后，所有数据将是一致的，因为计算必须满足某些要求才是有效的，并且所有节点将被同步以确保区块链是一致的。

[0031] 通过挖矿过程，交易诸如资产转账被网络节点验证并被添加至区块链的区块的增长链中。通过遍历整个区块链，验证可包括例如付款方是否有权访问转账资产、资产是否以前已经被花费、转账金额是否正确等。例如，在由发送方签发的假设交易(例如，根据UTXO(未被花费的交易输出)模型的比特币交易)中，提出的交易可被广播至区块链网络以便挖矿。矿工需要根据区块链历史检查交易是否有资格被执行。如果根据现有的区块链历史，发送方的钱包余额具有足够的资金，则认为交易是有效的，并且可以将该交易添加至区块中。资产转账一旦被验证，其可以被包含于下一个待被添加至区块链的区块中。

[0032] 区块非常像数据库记录。每次写入数据时创建一个区块。这些区块被链接并使用密码术被保护以成为互连的网络。每个区块都连接至前一区块，这也是名称“区块链”的起源。每个区块通常包含前一区块的加密哈希值、生成时间和实际数据。例如，每个区块包含两个部分：记录当前区块的特征值的区块头，以及记录实际数据(例如，交易数据)的主体。区块的链通过区块头链接。每个区块头可以包含多个特征值，例如版本、前一区块哈希值、默克尔(Merkle)根、时间戳、难度目标和随机数。前一区块哈希值不仅包含前一区块的地

址，而且包含前一区块内部的数据的哈希值，从而使得区块链不可变。随机数是当被包括时产生具有指定数量的前导零位的哈希值的数。

[0033] 为了挖矿，节点获取新区块的内容的哈希值。随机数(例如，随机字符串)被附加到该哈希值，从而获得新字符串。新字符串被再次进行哈希处理。然后，将最终的哈希值与难度目标(例如，级别)进行比较，并确定最终的哈希值是否实际小于难度目标。如果最终的哈希值不小于难度目标，则改变随机数，并且再次重复该过程。如果最终的哈希值小于难度目标，则将所述区块添加至链中，更新公开账本并提醒所述添加。负责成功添加的节点获得比特币奖励，例如，通过向新区块添加对自身的奖励交易(称为造币生成)。

[0034] 即，对于每个输出“Y”，如果k是从最小熵(min-entropy)高的分布中选择的，则不可能找到输入x以使得H(k|x)＝Y，其中K是随机数，x是区块的哈希值，Y是难度目标，并且“|”表示级联(concatenation)。由于加密哈希值基本上是随机的，由于加密哈希值的输出不能从其输入预测，仅存在一种找到随机数的已知方法：一个接一个地尝试整数，例如1、然后2、然后3、等等，这可以被称为蛮力。前导零的数量越大，找到必需的随机数Y所花费的平均时间就越长。在一个示例中，比特币系统不断地调整前导零的数量，使得找到随机数的平均时间约为十分钟。这样，由于计算硬件的处理能力随着时间提高，接下来几年，比特币协议将只需要更多的前导零位，以使得实现挖矿总是花费大约十分钟的持续时间。

[0035] 如上所述，哈希处理是区块链的重要基础。可以将哈希算法理解为将任何长度的消息压缩成固定长度消息摘要的函数。MD5和SHA是更常使用的。在一些实施例中，区块链的哈希值长度是256位，这意味着无论原始内容是什么，最终都计算256位的二进制数。并且只要原始内容不同，就可以保证相应的哈希值是唯一的。例如，字符串“123”的哈希值是a8fdc205a9f19cc1c7507a60c4f01b13d11d7fd0(十六进制)，其当被转换为二进制时具有256位，并且仅“123”具有这个哈希值。区块链中的哈希算法是不可逆的，即，正向计算是容易的(从“123”至a8fdc205a9f19cc1c7507a60c4f01b1c7507a60c4f01b13d11d7fd0)，并且即使耗尽所有计算资源，也不能完成逆向计算。因此，区块链的每个区块的哈希值是唯一的。

[0036] 此外，如果区块的内容改变，则其哈希值也将改变。区块和哈希值一一对应，并且针对区块头具体计算每个区块的哈希值。即，连接区块头的特征值以形成长字符串，然后为所述字符串计算哈希值。例如，“哈希值＝SHA256(区块头)”是区块哈希值计算公式，SHA256是应用于区块头的区块链哈希算法。哈希值由区块头而不是区块主体唯一地确定。如上所述，区块头包含许多内容，包括当前区块的哈希值和前一区块的哈希值。这意味着如果当前区块的内容改变，或者如果前一区块的哈希值改变，则将导致当前区块中的哈希值改变。如果黑客修改了区块，则所述区块的哈希值改变。由于下一区块必须包含前一区块的哈希值，为了使后面的区块连接至被修改的区块，黑客必须依次修改所有随后的区块。否则，被修改的区块将脱离区块链。由于设计原因，哈希值计算是耗时的，几乎不可能在短时间内修改多个区块，除非黑客已经掌握了整个网络51％以上的计算能力。因此，区块链保证了其自身的可靠性，并且一旦数据被写入，数据就不能被篡改。

[0037] 一旦矿工找到了新区块的哈希值(即，合格的签名或解决方案)，矿工就将该签名广播至所有其它矿工(区块链中的节点)。现在，其它矿工依次验证所述解决方案是否与发送方的区块的问题相对应(即，确定哈希值输入是否实际上导致所述签名)。如果所述解决方案是有效的，则其它矿工将确认该解决方案，并同意可将新区块添加至区块链。因此，达

成了新区块的共识。这也被称为“工作量证明”。已经达成共识的区块现在可以被添加至区块链中，并且与其签名一起被广播至网络上的所有节点。只要区块内的交易在所述时间点正确地对应于当前钱包余额(交易历史)，节点就将接受该区块并将保存该区块于节点的交易数据。每当在所述区块的顶部添加新的区块时，所述添加还可算作为对其之前的区块的另一“确认”。例如，如果交易被包括在区块502中，并且区块链有507个区块，这意味着交易具有五个确认(对应于区块507至区块502)。交易具有越多的确认，攻击者就越难改变。

[0038] 在一些实施例中，示例性区块链资产系统利用公钥密码术，其中生成两个密钥，一个公钥和一个私钥。可认为公钥是账号，而可认为私钥是所有权凭证。例如，比特币钱包是公钥和私钥的集合。可以用属于资产地址的私钥的知识来证明与该地址相关联的资产(例如，数字货币、现金资产、股票、股权、债券)的所有权。例如，比特币钱包软件，有时称为“比特币客户软件”，允许给定用户交易比特币。钱包程序生成并存储私钥，并与比特币网络上的对等点通信。

[0039] 在区块链交易中，通过付款人和收款人的公钥在区块链中识别付款人和收款人。例如，大多数当代比特币转账是从一个公钥到不同的公钥。实际上，这些密钥的哈希值被用于区块链中，并且被称为“比特币地址”。原则上，如果使用用户的比特币地址而不是他们的名字，假想的攻击者个人S可以通过简单地向区块链账本添加像“个人A向个人S支付100个比特币”一样的交易从个人A偷钱。比特币协议通过要求每次转账都用付款人的私钥进行数字签名防止这种盗窃，并且只有经签名的转账可以被添加至区块链账本中。由于个人S不能伪造个人A的签名，因此个人S不能通过向区块链添加相当于“个人A向个人S支付200个比特币”的条目来欺骗个人A。同时，任何人都可以使用他/她的公钥来验证个人A的签名，并因此在他/她是付款人的情况下验证他/她已经授权了区块链中的任何交易。

[0040] 在比特币交易的情况下，为了向用户B转账一些比特币，用户A可以构造包含关于通过节点的交易的信息的记录。该记录可以签有用户A的签名密钥(私钥)，并且包含用户A的公共验证密钥和用户B的公共验证密钥。签名用于确认交易来自用户A，并且一旦发布交易，还防止该交易被任何人更改。该记录与在新区块中的同一时间窗口中发生的其它记录一起可以被广播至全量节点。在接收到记录时，全量节点可以将记录合并到区块链系统中曾经发生的所有交易的账本中，通过上述挖矿过程将新区块添加至先前接受的区块链，并且针对网络的共识规则验证所添加的区块。

[0041] 用户A的待转账资产可以是UTXO(未被花费的交易输出)形式。UTXO是区块链对象模型。根据UTXO，资产由尚未被花费的区块链交易的输出表示，所述输出可以用作新交易中的输入。为了花费(交易)资产，用户必须用私钥签名。比特币是使用UTXO模型的数字货币的示例。在有效的区块链交易的情况下，未被花费的输出可用于实现进一步的交易。在一些实施例中，在进一步的交易中可以仅使用未被花费的输出，以防止双花和欺诈。为此，区块链上的输入在交易发生时被删除，同时创建UTXO形式的输出。这些未被花费的交易输出可以(由私钥持有者，例如，具有数字货币钱包的人)用于未来交易。

[0042] 由于区块链和其它类似的账本是完全公开的，因此区块链本身没有隐私保护。P2P网络的公共特性意味着尽管使用它的人不是通过名字来标识的，但是将交易链接至个人和公司是可行的。例如，在跨境汇款中或在供应链中，资产类型具有极高级别的隐私保护值，因为使用资产类型信息，可以推断交易方的特定位置和身份。资产类型可以包括例如金钱、

数字货币、合同、契据、医疗记录、客户明细、股票、债券、债权或可以以数字形式描述的任何其他资产类型。尽管UTXO模型为身份和交易金额提供匿名，并已经应用于门罗币(Monero)和大零币(Zcash)，但是交易资产类型仍然是不受保护的。因此，本公开所解决的技术问题是如何保护诸如交易中的资产类型的在线信息的隐私。所公开的系统和方法可以被集成至UTXO模型以为各种交易内容提供隐私保护。

[0043] 在交易期间，信息保护对于保护用户隐私是重要的，且交易资产类型是一种缺乏保护的信息类型。图1示出了根据各种实施例的用于信息保护的示例性系统100。如所示出的，区块链网络可以包括多个节点(例如，在服务器、计算机等中实现的全量节点)。对于某些区块链平台(例如，NEO)，具有某级别投票权的全量节点可被称为共识节点，共识节点承担交易验证的责任。在本公开中，全量节点、共识节点或其它等效节点可以验证交易。

[0044] 此外，如图1所示，用户A和用户B可以使用作为轻量节点的相应设备，诸如膝上型计算机和移动电话等执行交易。例如，用户A可能想要通过将用户A的账户中的某些资产转账到用户B的账户来与用户B交易。用户A和用户B可以使用安装有适当的交易用区块链软件的相应设备。用户A的设备可以称为发起方节点A，发起方节点A发起与被称为接收方节点B的用户B的设备的交易。节点A可以通过与节点1的通信访问区块链，而节点B可以通过与节点2的通信访问区块链。例如，节点A和节点B可以通过节点1和节点2向区块链提交交易，以请求向区块链添加交易。除了区块链，节点A和节点B可以具有其它通信信道。例如，节点A和节点B可以通过常规因特网通信获取彼此的公钥。

[0045] 图1中的每个节点可以包括处理器和非暂态计算机可读存储介质，所述非暂态计算机可读存储介质用于存储待由处理器执行以使节点(例如，节点的处理器)执行用于本文所述的信息保护的各个步骤的指令。每个节点可以安装有软件(例如，交易程序)和/或硬件(例如，有线、无线连接)以与其它节点和/或其它设备通信。稍后参考图5描述节点硬件和软件的进一步细节。

[0046] 图2示出了根据各种实施例的交易发起和验证的示例性步骤。

[0047] 交易发起可以由发起者节点实施。在一些实施例中，每种类型的资产类型可以被映射或分配至唯一身份。例如，该唯一身份可以是用以下方式计算出的序列号sn：

[0048] 步骤1.2 sn＝Hash(资产类型)

[0049] 其中Hash()是哈希函数。此外，可以通过如下承诺方案(例如，佩德森(Pedersen)承诺)对资产类型进行加密：

[0050] 步骤1.3 C(sn)＝r×G+sn×H

[0051] 其中，r是提供隐藏的随机盲因子(或者称为盲因子)，G和H是公开同意的椭圆曲线的生成元/基点，并且可以是随机选择的，sn是承诺的值，C(sn)是用作承诺并被给予对方的曲线点，并且H是另一曲线点。也就是说，G和H可以是对于节点已知的参数。通过利用从一个点映射到另一个点的哈希函数H＝Hash(G)对基点G哈希处理，生成H的“空袖数(nothing up my sleeve)”生成。H和G是给定系统的公共参数(例如，椭圆曲线上的随机生成的点)。发送者节点可能已将H和G发布至所有节点。尽管以上提供了椭圆曲线形式的Pedersen承诺的示例，但是可以替代地使用各种其它形式的Pedersen承诺或其它承诺方案。

[0052] 承诺方案保持数据保密但承诺数据，使得数据的发送者稍后不能改变数据。如果一方仅知道承诺值(例如，C(sn))，它们不能确定哪些底层数据值(例如，sn)已经被承诺。随

后数据(例如,sn)以及盲因子(例如,r)可以被显露(例如,由发起者节点),承诺的接收方(例如,共识节点)可以运行该承诺,并验证该承诺的数据与所显露的数据相匹配。之所以存在这种盲因子,是因为如果没有这种盲因子,有人可能试图猜测数据。

[0053] 承诺方案是发送方(承诺方)承诺一值(例如,sn)使得承诺的值保持隐私,但是可以在当承诺方泄露承诺过程的必要参数的稍后时间显示该承诺的值的一种方式。强承诺方案可以既是信息隐藏的又是计算绑定的。隐藏是指给定值sn和该值的承诺C(sn)应该是不相关的这一概念。即,C(sn)不应显露关于sn的信息。在已知C(sn)、G和H的情况下,因为随机数r,几乎不可能知道sn。如果几乎不存在使得两个不同的值可以导致相同承诺的方式,则承诺方案是绑定的。Pedersen承诺在离散对数假设下是完全隐藏和计算绑定的。

[0054] Pedersen承诺具有加法性质:可以将承诺相加,并且一直承诺的总和与对数据总和的承诺相同(其中盲键设置为盲键的总和):

[0055] C(BF1,data1)+C(BF2,data2)＝＝C(BF1+BF2,data1+data2);

[0056] C(BF1,data1)-C(BF1,data1)＝＝0。

[0057] 换句话说,承诺保留了加法并且交换属性适用,即,Pedersen承诺是相加同态的,因为底层数据可以进行数学运算,就像它没有被加密一样。

[0058] 在一个实施例中,用于加密输入值的Pedersen承诺可以使用椭圆曲线点来构造。传统上,椭圆曲线密码(ECC)公钥是通过将用于群(G)的生成元与密钥(r)相乘来创建的:Pub＝rG。结果可以被序列化为33字节阵列。ECC公钥可以服从前面关于Pedersen承诺所提到的相加同态属性。即:Pub1+Pub2＝(r1+r2(mod n))G。

[0059] 可以通过为所述群挑选附加的生成元(H,在下面的等式中)来创建用于输入值的Pedersen承诺,使得没有人知道第二生成元H相对于第一生成元G的离散对数(反之亦然),这意味着没有人知道使得xG＝H的x。这可以例如通过使用G的加密哈希挑选H来完成:

[0060] H＝to_point(SHA256(ENCODE(G)))。

[0061] 给定两个生成元G和H,加密输入值的示例承诺方案可以被定义为:承诺＝rG+aH。这里,r可以是秘密盲因子,a可以是被承诺的输入值。因此,如果sn被承诺,则可以获得上述承诺方案C(sn)＝r×G+sn×H。Pedersen承诺在信息理论上是隐私的:对于任何承诺,存在使一数值与该承诺匹配的盲因子。因为任意映射是不可计算的,Pedersen承诺可以从计算上对抗虚假承诺。

[0062] 承诺该值的一方(节点)可以通过公开原始值sn和完成承诺等式的因子r开放该承诺。然后,希望公开值C(sn)的一方将再次计算承诺,以验证共享的原始值确实与最初接收的承诺C(sn)相匹配。因此,可以通过将资产类型信息映射至唯一的序列号,然后通过Pedersen承诺对其加密而保护资产类型信息。当生成承诺时选择的随机数r使得任何人几乎不可能根据承诺值C(sn)推断出承诺的资产类型的类型。

[0063] 在一些实施例中,当在UTXO模型下包含资产类型信息保护方法时,可以验证输入的资产类型(sn_in)和输出的资产类型(sn_out)的一致性,以确定交易的有效性。例如,区块链节点可以拒绝一致性测试sn_in＝＝sn_out失败的交易或区块。因为资产类型sn是(例如,通过Pedersen承诺)被加密的,所以该一致性测试是验证C(sn_in)＝＝C(sn_out)是否成立。

[0064] 在一些实施例中,如图2中所示的,步骤1,UTXO型交易可以包括m个输入(例如,可

用资产)和n个输出(例如,转账的资产和剩余资产)。输入可以被标记为sn_in_k,其中1≤k≤m,且输出可以被标记为sn_out_k,其中1≤k≤n。一些输出可以被转发至接收方节点B,而剩余的输出可以返回发起者节点A。例如,在假想交易中,用户A可以在其钱包中拥有共计5个比特币和10支股票,且对于交易输入,sn_in_1=Hash(比特币),sn_in_2=Hash(股票)。如果用户A想要向用户B转账3个比特币,对于交易输出,sn_out_1=Hash(比特币),sn_out_2=Hash(比特币)且sn_out_3=Hash(股票),由此一个比特币输出(3个比特币)被送至用户B,而另一比特币输出(2个比特币)和股票输出被送回用户A。

[0065] 因此,在一些实施例中,输入对应资产类型可以以这样的形式被加密:

[0066] C_in_k=r_in_k×G+sn_in_k×H,其中1≤k≤m

[0067] 输出资产类型对应于以下加密形式:

[0068] C_out_k=r_out_k×G+sn_out_k×H,其中1≤k≤n

[0069] 在资产类型被隐藏的情况下,交易发起者需要向节点(例如,全量节点、共识节点)证明交易的输入资产类型分别与输出资产类型一致。因此,全量节点可以验证交易是否有效。

[0070] 在一些实施例中,为了发起资产类型被Pedersen承诺隐藏的UTXO型交易,交易发起者可以选择合适的输入和输出以执行以下步骤2.1至2.5(对应于图2中的步骤2):

[0071] 步骤2.1计算

[0072] C_1=C_in_1-C_in_2,

[0073] C_2=C_in_2-C_in_3,

[0074] …

[0075] C_(m-1)=C_in_(m-1)-C_in_m,

[0076] C_m=C_out_1-C_out_2,

[0077] C_(m+1)=C_out_2-C_out_3,

[0078] …

[0079] C_(m+n-2)=C_out_(n-1)-C_out_n,

[0080] C_(m+n-1)=C_in_1-C_out_1;

[0081] 步骤2.2计算x=Hash(C_1||C_2||C_3||…||C_(m+n-1)),其中“||”代表级联;

[0082] 步骤2.3计算$C=C_1+x\times C_2+x^2\times C_3+\cdots+x^{(m+n-2)}\times C_{(m+n-1)}$。注意到多项式的项可与步骤2.1中的项对应;

[0083] 步骤2.4计算$R=(r_{in_1}-r_{in_2})+x\times(r_{in_2}-r_{in_3})+x^2\times(r_{in_3}-r_{in_4})+\cdots+x^{(m+n-2)}\times(r_{in_1}-r_{out_1})$。注意到多项式中的项与步骤2.1中的项对应,例如(r_in_1-r_in_2)对应于C_in_1-C_in_2;

[0084] 步骤2.5将R发布至节点,例如,在交易信息广播中。

[0085] 在一些实施例中,为验证输入资产类型和输出资产类型一致,C=R×G必须成立。例如,在交易验证期间,节点执行以下步骤3.1至3.3(对应于图2,步骤3.1-3.3)以验证交易资产类型是否一致。

[0086] 步骤3.1计算x=Hash(C_1||C_2||C_3||…||C_(m+n-1));

[0087] 步骤3.2计算$C=C_1+x\times C_2+x^2\times C_3+\cdots+x^{(m+n-2)}\times C_{(m+n-1)}$;

[0088] 步骤3.3验证C=R×G是否成立。如果C=R×G成立,则资产类型一致,否则,资产类

型不一致，且交易被拒绝。在一些实施例中，可以将C(sn)发布至节点，且步骤2.1至2.3的算法被节点(例如，包括提交交易的节点以及验证交易的节点)获知。因此，验证交易的节点可以对应地进行步骤3.1至3.3以执行该验证。因此，被拒绝的交易将不会被添加至区块链中。如图2中的步骤4所示出的，基于一致性确认，节点可以确定是否将交易添加至区块链或拒绝添加交易。

[0089] 照此，交易发起者可以为区块链节点提交信息以基于输入到交易和从交易输出的资产类型的一致性来验证交易，而不会公开实际的资产类型，也无法更改所提交的信息。为每种资产类型分配序列号(例如，哈希值)扩展并随机化每种资产类型的表示，使得交易发起者难以伪造资产类型以通过验证。此外，由于随机数r的存在，不同时刻加密的相同资产类型是不同的。将Pedersen承诺应用于对资产类型哈希值进行加密，使得资产类型的隐私保护提高到了更高水平。因此，通过步骤2.1至2.5，交易发起者可以在不泄露资产类型的情况下向其他节点证明交易的资产类型有效。例如，获得输入资产类型和输出资产类型之间的差值，并基于所述差值构造多项式，以便交易发起者可以将转换后的资产类型传递给其他节点以证明资产类型的一致性和交易的有效性。同时，因为x是通过哈希处理计算得出以作为多项式中各个指数的底数，所以可以忽略交易发起者或其他节点能够伪造资产类型的可能性。此外，R的公开允许其他节点在不知道资产类型的情况下通过步骤3.1至3.3来验证交易中的资产类型是一致的。因此，利用所公开的系统和方法，可以在保持出色的隐私保护的同时由第三方验证数据信息。

[0090] 图3示出了根据本公开的各种实施例的用于信息保护的示例方法300的流程图。方法300可由图1的系统100的一个或多个组件(例如，节点A)实现。方法300可由包括处理器和非暂态计算机可读存储介质(例如，存储器)的系统或设备(例如，计算机)实现，所述存储介质用于存储将由处理器执行的、促使该系统或设备(例如，处理器)执行方法300的指令。下面给出的方法300的操作旨在是说明性的。取决于实施方式，示例方法300可包括以各种顺序或并行执行的附加的、较少的或替代的步骤。

[0091] 方框301包括：确定交易的一个或多个数据输入以及一个或多个数据输出，其中数据输入分别与输入数据类型相关联，且数据输出分别与输出数据类型相关联。参见例如图2中的步骤1。在一些实施例中，交易是至少基于未被花费的交易输出(UTXO)模型的；且数据输入和数据输出包括发送者(发起者节点)和接收方(接收方节点)之间正在进行所述交易的一个或多个资产的类型。资产类型可以包括例如金钱、数字货币、合同、契据、医疗记录、客户明细、股票、债券、债权或可以以数字形式描述的任何其他资产的类型。

[0092] 方框302包括：加密输入数据类型和输出数据类型。参见例如上述的步骤1.2。在一些实施例中，加密输入数据类型和输出数据类型包括利用哈希函数或另一单向函数加密每个输入数据类型和每个输出数据类型。

[0093] 方框303包括：利用承诺方案承诺每个加密的输入数据类型和每个加密的输出数据类型来获得对应的承诺值。参见例如上述的步骤1.3。在一些实施例中，承诺方案包括Pedersen承诺。在一些实施例中，承诺方案至少包括盲因子；且该盲因子随对加密的输入数据类型和加密的输出数据类型进行承诺的时间而变化。也即，由于变化的盲因子，即使在不同时间加密的相同的数据(例如，相同的数据类型)也会是不同的承诺值。

[0094] 方框304包括：至少基于承诺值至少获取参数R。参见例如上述步骤2.1至2.4。在一

些实施例中，承诺方案包括分别对应于输入数据类型和输出数据类型(参见例如，r_in_k和r_out_k)的多个盲因子；且至少基于承诺值至少获取参数R包括：获取成对的承诺值之间的差值(参见例如步骤2.1，针对输入资产类型和输出资产类型中各成对的承诺值，可获得各成对的承诺值之间的差值)；级联所获取的差值(参见例如，步骤2.2)；利用哈希函数加密所级联的差值以获取加密值x(参见例如，步骤2.2)；并至少基于加密值x以及成对的盲因子之间的差值获取参数R(参见例如，步骤2.4)。

[0095] 方框305包括：在公开参数R而不公开输入数据类型和输出数据类型的情况下，将交易提交至区块链网络中的一个或多个节点以供节点验证输入数据类型和输出数据类型间的一致性。在一些实施例中，使所述节点在不知道输入数据类型和输出数据类型的情况下验证输入数据类型和输出数据类型间的一致性。

[0096] 在一些实施例中，在公开参数R而不公开输入数据类型和输出数据类型的情况下，将交易提交至区块链网络中的一个或多个节点以供节点验证输入数据类型和输出数据类型间的一致性，包括在公开参数R而不公开输入数据类型和输出数据类型的情况下，将交易提交至区块链网络中的一个或多个节点以促使节点：获取参数R和基点G(参见例如步骤3.1中的G。H和G可以是所有节点可用的公开参数)；获取成对的输入资产类型和输出资产类型的承诺值之间的差值(参见例如与步骤2.1相似的步骤)；级联所获取的差值(参见例如步骤3.1)；利用哈希函数加密所级联的差值以获取加密值x(参见例如步骤3.1)；至少基于所获取的差值和加密值x获取多项式的总和C(参见例如步骤3.2)；响应于确定该总和C等于参数R和基点G的积，确定输入数据类型和输出数据类型一致并将交易添加至区块链中(参见例如步骤3.3)；以及响应于确定该总和C不等于参数R和基点G的积，确定输入数据类型和输出数据类型不一致并拒绝将交易添加至区块链中(参见例如步骤3.3)。

[0097] 图4示出了根据本公开的各种实施例的用于信息保护的示例方法400的流程图。方法400可由图1的系统100的一个或多个组件(例如，节点i)实现。节点i可以包括在服务器上实现的全量节点。方法400可以由包括处理器(例如，计算机)和非暂态计算机可读存储介质(例如，存储器)的系统或设备实现，所述存储介质用于存储将由处理器执行的、使系统或设备(例如，处理器)执行方法400的指令。下面给出的方法400的操作旨在是说明性的。取决于实施方式，示例方法400可以包括以各种顺序或并行执行的附加的、较少的或替代的步骤。

[0098] 方框401包括：由区块链网络中的一个或多个节点(例如，共识节点)获取由发起者节点发起的交易。交易与一个或多个数据输入及一个或多个数据输出相关联。数据输入分别与输入数据类型相关联，且数据输出分别与输出数据类型相关联。输入数据类型和输出数据类型被加密且被交给承诺方案以获取对应的承诺值。输入数据类型和输出数据类型不对所述一个或多个节点公开。

[0099] 方框402包括：由所述一个或多个节点验证输入数据类型和输出数据类型间的一致性。在一些实施例中，验证输入数据类型和输出数据类型间的一致性包括：获取参数R和基点G(参见例如步骤2.4和2.5中的R，步骤3.1中的G)；获取成对的输入资产类型和输出资产类型的承诺值之间的差值(参见例如与步骤2.1相似的步骤)；级联所获取的差值(参见例如步骤3.1)；利用哈希函数加密所级联的差值以获取加密值x(参见例如步骤3.1)；至少基于所获取的差值和加密值x获取多项式的总和C(参见例如步骤3.2)；以及确定总和C是否等于参数R和基点G的积(参见例如步骤3.3)。

[0100] 方框403包括:响应于确定输入数据类型和输出数据类型一致,所述一个或多个节点将交易添加至区块链网络中。

[0101] 方框404包括:响应于确定输入数据类型和输出数据类型不一致,所述一个或多个节点拒绝将交易添加至区块链网络中。

[0102] 在一些实施例中,方法还包括:响应于确定该总和C等于参数R和基点G的积,确定输入数据类型和输出数据类型一致;以及响应于确定该总和C不等于参数R和基点G的积,确定输入数据类型和输出数据类型不一致。

[0103] 照此,交易发起者可以提交信息以供区块链节点基于输入到交易和从交易输出的资产类型之间的一致性来验证交易,而不会公开实际的资产类型,也无法更改所提交的信息。为每种资产类型分配序列号(例如,哈希值)扩展并随机化每种资产类型的表示,使得交易发起者难以伪造资产类型以通过验证。此外,由于随机数r的存在,不同时刻加密的相同资产类型是不同的。将Pedersen承诺应用于对资产类型哈希值进行加密,使得资产类型的隐私保护提高到更高水平。因此,通过步骤2.1至2.5,交易发起者可以在不公开资产类型的情况下向其他节点证明交易的资产类型有效。例如,获得输入资产类型和输出资产类型之间的差值,并基于所述差值构造多项式,以便交易发起者可以将转换后的资产类型传递给其他节点以证明资产类型的一致性和交易的有效性。同时,因为x是通过哈希处理计算得出以作为多项式中各个指数的底数,所以可以忽略交易发起者或其他节点能够伪造资产类型的可能性。此外,R的公开允许其他节点在不知道资产类型的情况下通过步骤3.1至3.3来验证交易中的资产类型是一致的。因此,利用所公开的系统和方法,可以在保持出色的隐私保护的同时由第三方验证数据信息。

[0104] 本文描述的技术由一个或多个专用计算设备实现。专用计算设备可以是台式计算机系统、服务器计算机系统、便携式计算机系统、手持设备、联网设备或任何其他设备,或包括硬连线和/或程序逻辑以实现所述技术的设备的组合。计算设备通常由操作系统软件控制和协调。传统的操作系统控制和调度用于执行的计算机进程,执行存储器管理,提供文件系统、联网、I/O服务,并提供用户界面功能、例如图形用户界面(GUI)等等。

[0105] 图5是示出了可以在其上实现本文描述的任何实施例的计算机系统500的框图。系统500可以在本文描述的任何节点中实现,并且被配置为执行用于信息保护方法的相应步骤。计算机系统500包括总线502或用于传送信息的其它通信机制、与总线502耦合的用于处理信息的一个或多个硬件处理器504。硬件处理器504可以是例如一个或多个通用微处理器。

[0106] 计算机系统500还包括耦合至总线502的用于存储信息和要由处理器504执行的指令的主存储器506,诸如随机存取存储器(RAM)、高速缓存和/或其他动态存储设备。主存储器506还可以用于存储在执行指令期间要由(一个或多个)处理器504执行的临时变量或其它中间信息。当这些指令被存储在处理器504可访问的存储介质中时,这些指令使计算机系统500呈现为专用机器,所述专用机器被定制为执行指令中所指定的操作。计算机系统500还包括耦合至总线502的用于存储用于处理器504的静态信息和指令的只读存储器(ROM)508或其他静态存储设备。诸如磁盘、光盘或USB拇指驱动器(闪存驱动器)等的存储设备510被提供并耦合至总线502,用于存储信息和指令。

[0107] 计算机系统500可以使用定制的硬连线逻辑、一个或多个ASIC或FPGA、固件和/或

程序逻辑实现本文描述的技术，所述固件和/或程序逻辑与计算机系统相结合使得计算机系统500成为专用机器或将计算机系统500编程为专用机器。根据一个实施例，计算机系统500响应于处理器504执行包含在主存储器506中的一个或多个指令的一个或多个序列，执行本文描述的操作、方法和过程。这些指令可以从另一存储介质诸如存储设备510读入主存储器506。执行包含在主存储器506中的指令序列使得处理器504执行本文描述的处理步骤。在替代实施例中，可以使用硬连线电路代替软件指令，或将硬连线电路与软件指令组合使用。

[0108] 主存储器506、ROM 508和/或存储设备510可以包括非暂时性存储介质。如本文所使用的术语“非暂时性介质”和类似术语指的是存储用以使机器以特定方式操作的数据和/或指令的介质，所述介质不包括瞬态信号。这种非暂时性介质可包括非易失性介质和/或易失性介质。非易失性介质包括例如光盘或磁盘，诸如存储设备510。易失性介质包括动态存储器，例如主存储器506。非暂时性介质的常见形式包括例如软盘、柔性盘、硬盘、固态驱动器、磁带或任何其它磁性数据存储介质、CD-ROM、任何其它光学数据存储介质、具有孔图案的任何物理介质、RAM、PROM和EPROM、FLASH-EPROM、NVRAM、任何其它存储芯片或存储盒、以及它们的联网版本。

[0109] 计算机系统500还包括耦合至总线502的网络接口518。网络接口518提供耦合至一个或多个网络链路的双向数据通信，所述一个或多个网络链路连接至一个或多个本地网络。例如，网络接口518可以是综合业务数字网络(ISDN)卡、电缆调制解调器、卫星调制解调器或提供至相应类型的电话线的数据通信连接的调制解调器。作为另一示例，网络接口518可以是提供至兼容局域网(LAN)(或与WAN通信的WAN组件)的数据通信连接的LAN卡。无线链路也可以被实现。在任何这样的实现方式中，网络接口518发送和接收承载表示各种类型的信息的数字数据流的电信号、电磁信号或光信号。

[0110] 计算机系统500可以通过网络、网络链路和网络接口518发送消息并接收数据，包括程序代码。在因特网示例中，服务器可以通过因特网、ISP、本地网络和网络接口518传输所请求的用于应用程序的代码。

[0111] 所接收的代码可以在被接收时由处理器504执行，和/或存储在存储设备510或其他非易失性存储设备中以供稍后执行。

[0112] 在前述部分中描述的每个过程、方法和算法可以在由一个或多个计算机系统或包括计算机硬件的计算机处理器执行的代码模块中实现，并且是完全自动化或部分自动化的。过程和算法可以在专用电路中部分或全部地被实现。

[0113] 上述各种特征和过程可彼此独立地使用，或者可以各种方式组合。所有可能的组合和子组合都将落入本公开的范围内。另外，在一些实现方式中可以省略某些方法或过程框。本文描述的方法和过程也不限于任何特定次序，且与其相关的框或状态可以以其它适当的次序来执行。例如，所描述的框或状态可以以不同于具体公开的顺序来执行，或者多个框或状态可以被组合在单个框或状态中。示例性框或状态可以串行、并行或以一些其它方式执行。可以向所公开的示例性实施例添加框或状态或从所公开的示例性实施例中移除框或状态。本文所述的示例性系统和组件可与所述的不同地构造。例如，与所公开的示例性实施例相比，可以添加、去除或重新布置元件。

[0114] 本文描述的示例性方法的各种操作可以至少部分地由算法执行。所述算法可以包

括在存储在存储器(例如,上述的非暂时性计算机可读存储介质)中的程序代码或指令中。这种算法可以包括机器学习算法。在一些实施例中,机器学习算法可以不显式地编程计算机以执行功能,但可以从训练数据中学习以建立执行所述功能的预测模型。

[0115] 本文描述的示例性方法的各种操作可以至少部分地由(例如,通过软件)被临时配置或永久地配置为执行相关操作的一个或多个处理器执行。无论是临时配置还是永久配置,这样的处理器可以构成操作以执行本文描述的一个或多个操作或功能的处理器实现的引擎。

[0116] 类似地,本文描述的方法可以至少部分地由处理器实现,其中特定的一个或多个处理器是硬件的示例。例如,方法的至少一些操作可以由一个或多个处理器或者处理器实现的引擎来执行。此外,一个或多个处理器还可操作以支持“云计算”环境中的相关操作的性能,或作为“软件即服务”(SaaS)操作。例如,至少一些操作可以由一组计算机(例如,包括处理器的机器)执行,这些操作可以经由网络(例如,因特网)并经由一个或多个适当的接口(例如,应用程序接口(API))被访问。

[0117] 某些操作性能可以分布在处理器之间,不仅驻留在单个机器内,而且跨多个机器部署。在一些示例性实施例中,处理器或处理器实现的引擎可以位于单个地理位置(例如,在家庭环境、办公室环境或服务器群内)。在其它示例性实施例中,处理器或处理器实现的引擎可以分布在多个地理位置上。

[0118] 在整个说明书中,多个实例可以实现被描述为单个实例的组件、操作或结构。尽管将一个或多个方法的各个操作示出并描述为单独的操作,但是这些单个操作中的一个或多个可以同时执行,并且不要求这些操作按所示顺序执行。在示例性配置中作为单独组件呈现的结构和功能可以被实现为组合结构或组件。类似地,作为单个组件呈现的结构和功能可以被实现为单独的组件。这些和其它变化、修改、添加和改进都落入本文的主题的范围内。

[0119] 尽管已经参考具体的示例性实施例描述了主题的概述,但是在不脱离本公开的实施例的更广范围的情况下,可以对这些实施例进行各种修改和改变。如果实际上公开了不只一个公开内容或概念,主题的这些实施例在本文中可单独或共同地由术语“发明”指代,这仅仅是为了方便,而不是旨在主动将本申请的范围限制到任何单个公开内容或概念。

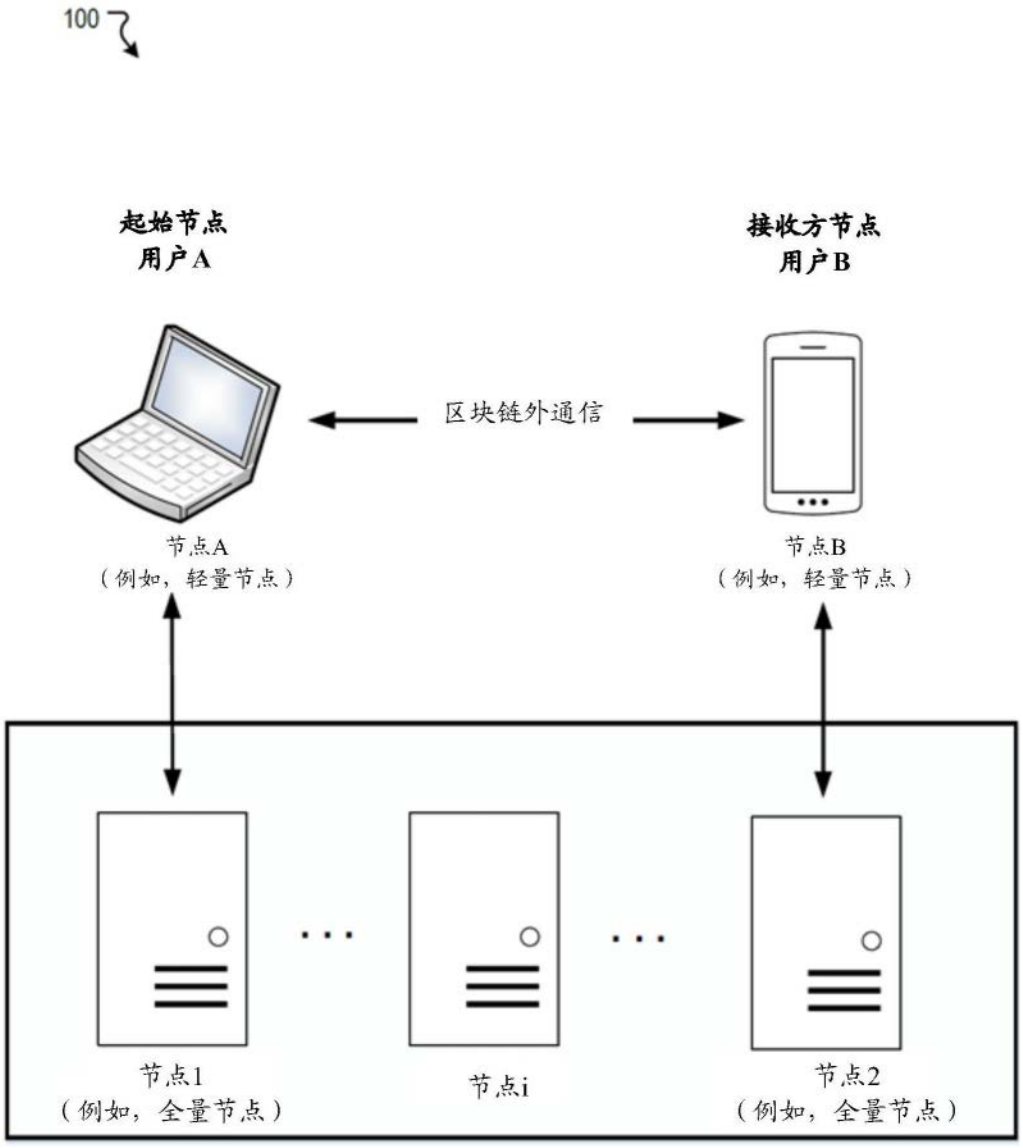
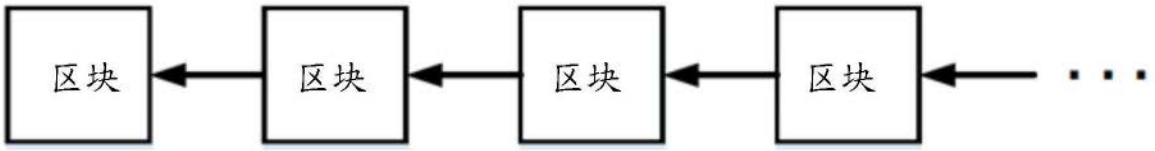

100
起始节点
用户A
接收方节点
用户B
区块链外通信
节点A
（例如，轻量节点）
节点B
（例如，轻量节点）
节点1
（例如，全量节点）
节点i
节点2
（例如，全量节点）

图1

发起者节点发起将被添加至区块链的交易

1.发起交易

2.计算x、C和R，并将R发布至节点



共识节点验证交易是否有效

3.1. 计算x

3.2. 计算C

3.3. 验证C=RG是否成立，以确认交易有效

4. 基于有效性将交易添加至区块链或拒绝

图2

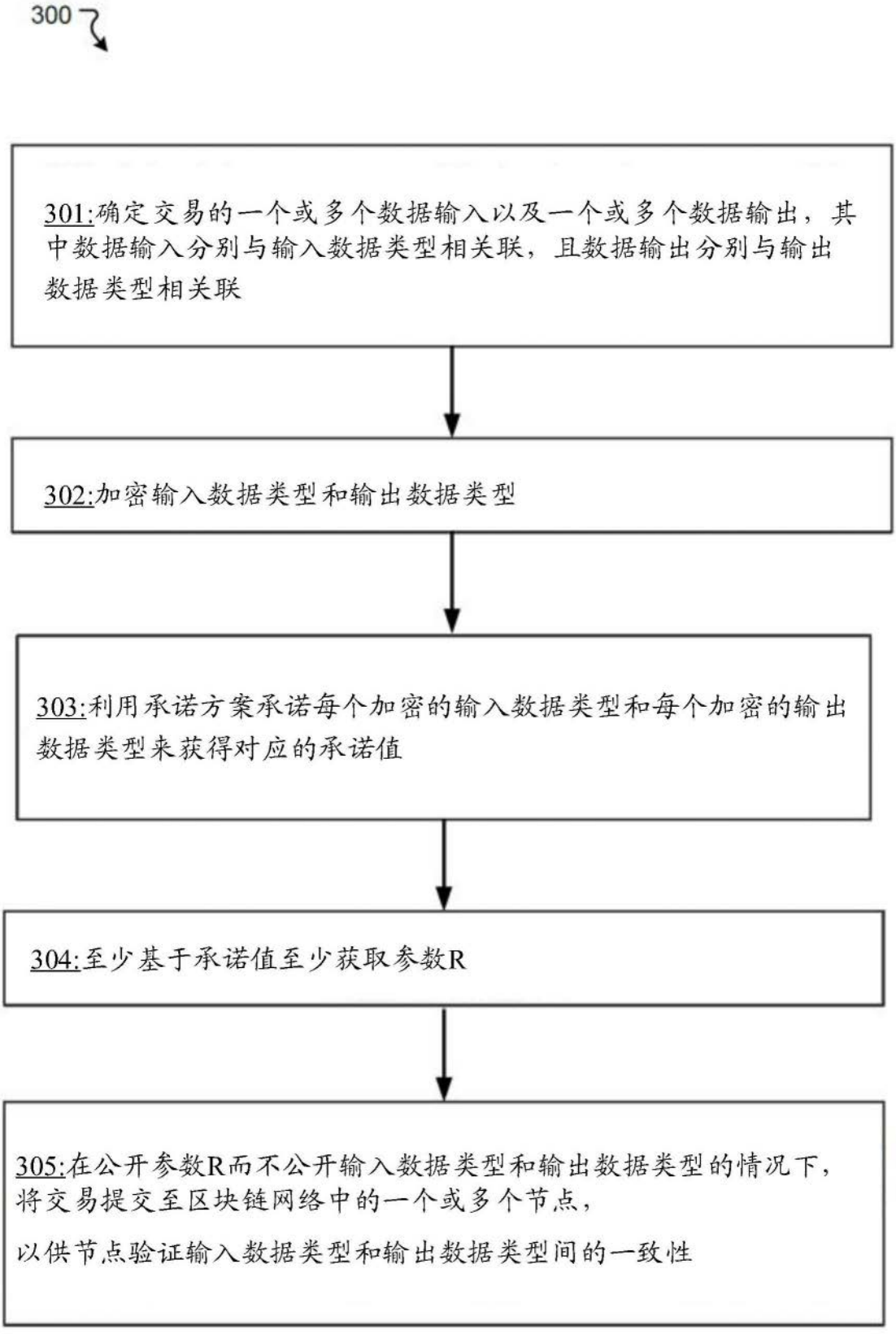

300
301:确定交易的一个或多个数据输入以及一个或多个数据输出，其中数据输入分别与输入数据类型相关联，且数据输出分别与输出数据类型相关联
302:加密输入数据类型和输出数据类型
303:利用承诺方案承诺每个加密的输入数据类型和每个加密的输出数据类型来获得对应的承诺值
304:至少基于承诺值至少获取参数R
305:在公开参数R而不公开输入数据类型和输出数据类型的情况下，将交易提交至区块链网络中的一个或多个节点，
以供节点验证输入数据类型和输出数据类型间的一致性

图3

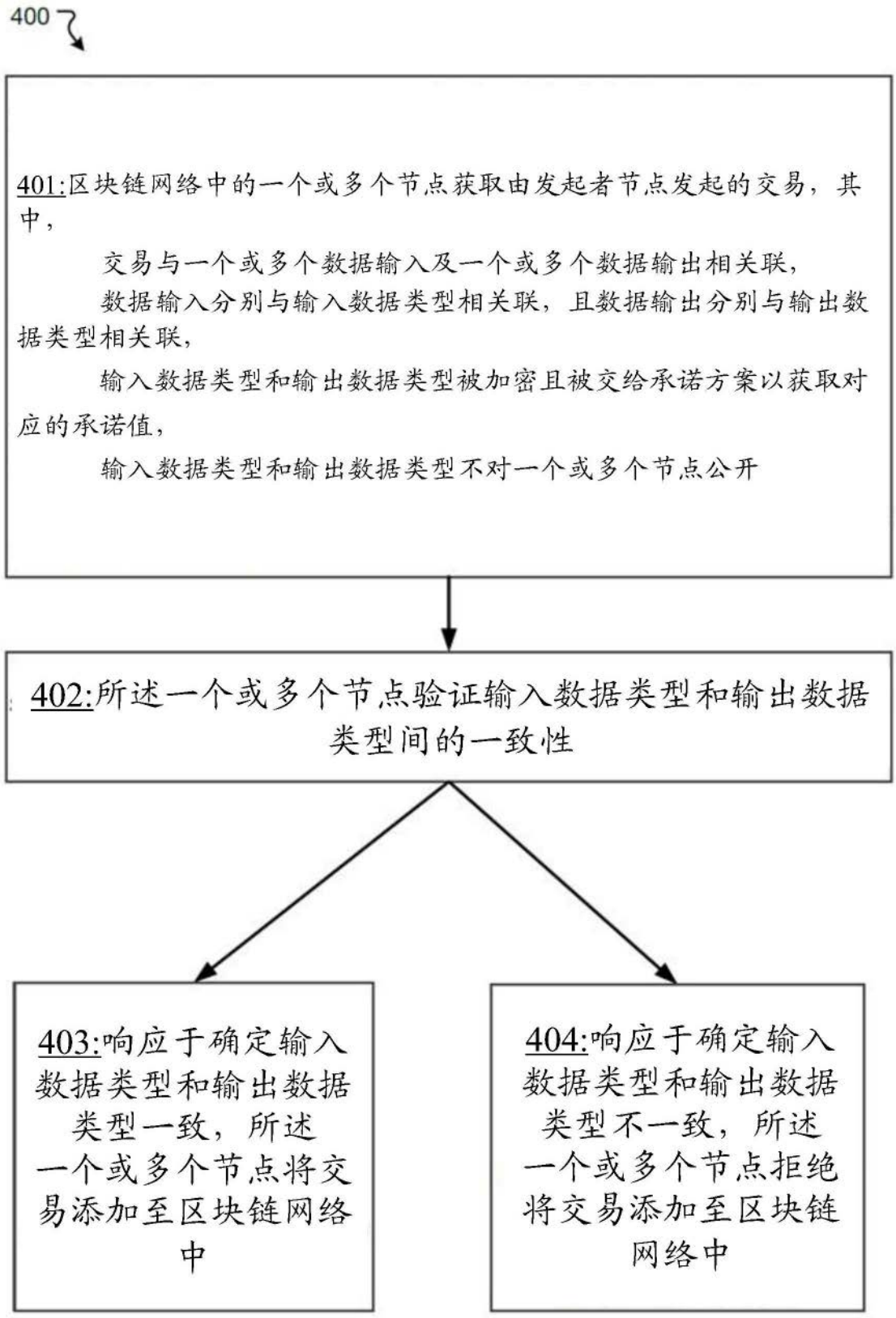

400
401:区块链网络中的一个或多个节点获取由发起者节点发起的交易，其中，
交易与一个或多个数据输入及一个或多个数据输出相关联，
数据输入分别与输入数据类型相关联，且数据输出分别与输出数据类型相关联，
输入数据类型和输出数据类型被加密且被交给承诺方案以获取对应的承诺值，
输入数据类型和输出数据类型不对一个或多个节点公开
402:所述一个或多个节点验证输入数据类型和输出数据类型间的一致性
403:响应于确定输入数据类型和输出数据类型一致，所述一个或多个节点将交易添加至区块链网络中
404:响应于确定输入数据类型和输出数据类型不一致，所述一个或多个节点拒绝将交易添加至区块链网络中

图4

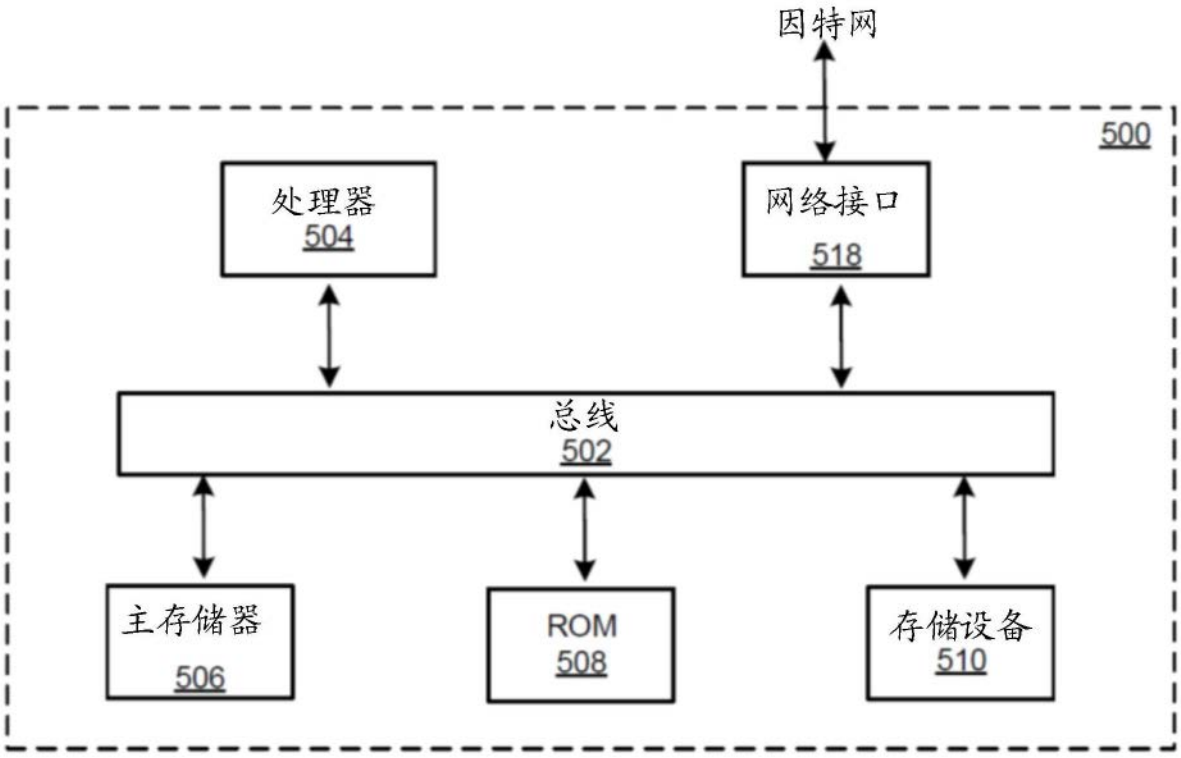


图5