



(12) 发明专利

(10) 授权公告号 CN 114008969 B

(45) 授权公告日 2025. 07. 11

(21) 申请号 202080038352.X

(22) 申请日 2020.04.22

(65) 同一申请的已公布的文献号
申请公布号 CN 114008969 A

(43) 申请公布日 2022.02.01

(30) 优先权数据
1907339.4 2019.05.24 GB

(85) PCT国际申请进入国家阶段日
2021.11.23

(86) PCT国际申请的申请数据
PCT/IB2020/053815 2020.04.22

(87) PCT国际申请的公布数据
W02020/240297 EN 2020.12.03

(73) 专利权人 区块链控股有限公司

地址 安提瓜和巴布达圣约翰教堂街44号菲
茨杰拉德故居

(72) 发明人 张伟 杰克·戴维斯
克雷格·赖特

(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 陈果

(51) Int.Cl.
H04L 9/32 (2006.01)

(56) 对比文件
CN 109074562 A, 2018.12.21
CN 109155034 A, 2019.01.04

审查员 肖敬伟

权利要求书4页 说明书35页 附图10页

(54) 发明名称

包含在区块链中的交易的延展性

(57) 摘要

本发明公开了一种在区块链中记录目标交易的方法,包括:获取所述目标交易的更新版本,所述更新版本相对于预先存在的第一版本进行更新;发送所述更新版本,而非所述第一版本,以通过节点网络传播并且记录在区块链中。所述目标交易包括输入,所述输入包含解锁脚本和指向第一交易的输出的指针,所述第一交易的输出包含锁定脚本,所述锁定脚本指定用于解锁所述输出的多个备选条件。所述目标交易的第一版本的所述解锁脚本被配置为在满足所述备选条件中的第一条条件的基础上,解锁所述第一交易的所述第一输出,所述更新版本的所述解锁脚本被配置为在满足所述备选条件中的第二条条件的基础上对其进行解锁。



1. 一种在区块链中记录第一方和第二方之间的目标交易的方法,所述方法包括通过所述第一方或第二方的计算机设备:

获取所述目标交易的第二更新版本,相对于所述目标交易预先存在的第一版本进行更新;

发送所述目标交易的所述更新版本,而非所述第一版本,以通过节点网络传播,并且记录在至少部分节点的每个节点维护的区块链副本中;

其中所述目标交易包括输入,所述输入包含解锁脚本和指向第一交易的第一输出的指针,所述第一输出包含锁定脚本,所述锁定脚本指定用于解锁所述第一交易的所述第一输出的多个备选条件;

其中所述目标交易的第一版本的所述解锁脚本被配置为在满足所述备选条件中的第一条条件的基础上,解锁所述第一交易的所述第一输出,所述更新版本的所述解锁脚本被配置为在满足所述备选条件中的第二条条件的基础上,而非所述第一条件,解锁所述第一交易的所述第一输出。

2. 根据权利要求1所述的方法,包括通过所述第二方的计算机设备,提供使所述第二方在满足所述第一条条件的基础上发送所述目标交易的所述第一版本的功能,以通过网络传播并且记录在区块链中。

3. 根据权利要求2所述的方法,其中:

-所述功能支持所述第二方手动选择执行所述第一版本的发送,和/或

-所述功能被配置为在预定时间到期后或在预定事件发生时自动触发所述目标交易的所述第一版本的发送。

4. 根据前述任一项所述的方法,其中所述目标交易的更新版本的获取和发送以传播并且记录在区块链中由所述第二方的计算机设备执行。

5. 根据权利要求4所述的方法,其中所述第二版本的获取包括从第一方接收所述第二版本的至少一部分。

6. 根据前述任一项所述的方法,包括通过所述第二方的计算机设备:

获取所述目标交易的所述第一版本,其中所述目标交易的第一版本的所述解锁脚本被配置为在满足所述第一条条件的基础上,使所述第二方自主地解锁所述第一交易的第一输出。

7. 根据权利要求6所述的方法,其中所述目标交易的第二版本在所述第一版本之后获取。

8. 根据权利要求6或7所述的方法,其中所述第一版本的获取包括从所述第一方接收所述第一版本的至少一部分。

9. 根据权利要求6、7或8所述的方法,其中所述第一版本的获取包括所述第二方生成所述第一交易。

10. 根据前述任一项所述的方法,其中所述第二条条件要求所述解锁脚本包括签署所述目标交易的一部分,不包括所述解锁脚本的所述第一方的加密签名;其中所述目标交易的更新版本的获取包括获取所述第一方的签名,并且发送以传播并且记录在区块链中的所述更新版本在所述解锁脚本中包括所述第一方的签名。

11. 根据权利要求10所述的方法,其中所述第一条条件不需要所述第一方的加密签名。

12. 根据权利要求10或11所述的方法, 其中至少所述第一条件要求所述解锁脚本包括所述第二方的加密签名, 所述加密签名签署所述目标交易的一部分, 不包括所述解锁脚本。

13. 根据权利要求12所述的方法, 其中所述第二条件还要求所述解锁脚本包括签署所述目标交易的一部分, 不包括所述解锁脚本的所述第二方的加密签名; 其中发送以传播并且记录在区块链中的所述更新版本在所述解锁脚本中包括所述第二方的加密签名。

14. 根据从属于权利要求6的权利要求12或13所述的方法, 其中所述更新版本的获取包括:

所述第二方将所述第一版本发送至所述第一方, 以便所述第一方通过添加所述第一方的签名延展为所述更新版本;

其中所述第一和第二条件要求所述第二方采用相同签名签署所述目标交易的相同部分, 从而所述第二方无需重新签署所述更新版本。

15. 根据前述任一项所述的方法, 其中所述第一条件要求在所述解锁脚本中包括数据有效载荷, 但是所述第二条件不要求在所述目标交易中包括所述数据有效载荷; 其中发送以通过网络传播并且记录在区块链中的所述更新版本不包括所述数据有效载荷。

16. 根据权利要求15所述的方法, 其中:

所述第一条件要求所述解锁脚本包括所述数据有效载荷和所述第二方的加密签名, 所述加密签名签署所述目标交易的一部分, 不包括所述解锁脚本, 但不要求在所述目标交易中包括所述第一方的加密签名;

所述第二条件要求所述解锁脚本包括所述第一方和所述第二方的加密签名, 但不要求在所述目标交易中包括所述数据有效载荷;

所述目标交易的更新版本的获取包括获取所述第一方的签名;

发送以传播并且记录在区块链中的所述目标交易的更新版本不包括所述数据有效载荷, 但在所述解锁脚本中包括所述第一方和所述第二方的签名。

17. 根据前述任一项所述的方法, 所述第一交易的第一输出指定向所述第二方支付的款项, 所述款项通过根据所述第一或第二条件解锁所述第一输出进行赎回, 并且其中所述目标交易包括将至少一些所述款项转移至所述第二方的输出。

18. 根据权利要求17所述的方法, 包括通过所述第二方的计算机设备:

将数据部分的序列流式传输到所述第一方, 以所述序列中的最终部分结束;

响应所述数据部分的每一个相应部分, 从所述第一方接收回所述第一交易的相应实例, 每个实例的所述第一输出针对所述相应部分指定向所述第二方支付的款项;

其中所述款项随着每个数据部分的增加而增加;

所述目标交易的更新版本的获取包括在所述序列中的最终部分之后从所述第一方接收所述更新版本。

19. 根据权利要求18所述的方法, 其中所述增加与迄今为止发送的数据部分的数量成线性比例。

20. 根据权利要求18或19所述的方法, 其中每个数据部分是媒体内容项的不同部分。

21. 根据权利要求18或19所述的方法, 其中每个数据部分是用于访问服务单元的不同密钥。

22. 根据权利要求21所述的方法, 其中所述服务包括以下其中一项:

- 提供包括电、气、水公用设施;或
- 租赁房产、车辆或其他实物。

23. 根据从属于权利要求2的权利要求18至22任一项所述的方法,其中:

-所述功能使所述第二方可以选择在所述序列中的任意点手动选择执行所述目标交易的第一版本的发送,以通过网络传播并且记录在区块链中,以赎回迄今为止从所述第一方接收的所述序列中所述第一交易的最新实例;和/或

-所述功能被配置为如果所述第一方停止部分通过所述序列发送所述第一交易的实例,则自动执行通过网络传播并且记录在区块链中的所述第一版本的发送,以赎回所述第一交易的最新实例。

24. 根据权利要求18至23任一项所述的方法,其中:

所述第一交易包括指定输入金额的一个或多个输入,所述第一交易的第一输出指定第一款项,所述第一交易进一步包括一个或多个指定一个或多个进一步款项的进一步输出,使得所述款项的总金额大于所述输入金额,并且所述第二方从所述第一方接收的所述第一交易不包括弥补差额的其他输入,所述网络的节点被配置为如果指定大于总输入金额的总款项,则因为无效而拒绝所述第一交易;

所述方法包括所述第二方向所述第一交易的最终或最新实例添加第二输入以弥补差额,并且发送添加所述第二输入的第一交易,以通过网络传播并且记录在区块链中。

25. 根据权利要求24所述的方法,其中所述进一步输出包括第二输出和第三输出,所述第二输出指定向所述第一方支付的等于所述输入金额减去所述第一款项的款项,所述第三输出指定向所述第二方支付的等于第二款项的第三款项。

26. 根据权利要求17至25任一项所述的方法,其中所述锁定脚本包括所述备选条件中的第三条件,其要求锁定时间已过期以及在所述解锁脚本中包含所述第一方的加密签名,从而如果所述第二方在锁定时间内未赎回,则所述第一方可赎回所述第一交易的第一输出中的款项。

27. 根据权利要求1至9任一项所述的方法,其中:

至少部分所述备选条件中的每一个条件对应不同的投票选择;

所述方法包括在获取所述目标交易的更新版本之前,访问或使第三方可以访问所述第一版本中的解锁脚本,作为投票意图的非约束性指示。

28. 根据权利要求27所述的方法,其中锁定时间包含在所述第一交易和所述目标交易的第一版本的至少一个中,以防止网络在预定时间点之前在区块链中记录所述第一版本。

29. 根据权利要求27或28所述的方法,所述第一交易的输出指定通过根据所述锁定脚本中指定条件的任何一个解锁所述第一输出进行赎回,并且其中所述目标交易包括将至少部分所述款项转移至所述第一方的输出。

30. 根据权利要求27至29任一项所述的方法,其中所述备选条件包括:

- 第一条件,所述第一条件对应于所述第一选择的轮询指示,
- 第二条件,所述第二条件对应于所述第二选择的轮询指示,
- 第三条件,所述第三条件对应于所述第一选择的投票选择,
- 第四条件,所述第四条件对应于所述第二选择的投票选择;

其中所述目标交易的第一版本的所述解锁脚本被配置为在满足所述第一或第二条件

的基础上,解锁所述第一交易的第一输出,

所述目标交易的第二版本的所述解锁脚本被配置为通过被配置为在满足所述第三条件的基础上,而非所述第一条件或第四条件或第二条件,解锁所述第一交易的第一输出来将轮询指示转换为投票。

31.一种包含在计算机可读存储器上的计算机程序,配置为当在所述第一方和/或第二方的计算机设备上运行时执行权利要求1至30其中一项所述的方法。

32.所述第一方和/或第二方的计算机设备,包括:

存储器,所述存储器包括一个或多个存储器单元,

处理装置,所述处理装置包括一个或多个处理单元;

其中所述存储器储存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权利要求1至30其中一项所述的方法。

33.用于记录在区块链中的一组交易,所述组在计算机可读数据介质或媒体上包括:

第一交易,所述第一交易具有包含锁定脚本的输出,所述锁定脚本指定用于解锁所述第一交易的输出的多个备选条件;

所述第二交易的第一版本具有包含解锁脚本的输入和指向所述第一交易的输出的指针,其中所述目标交易的第一版本的所述解锁脚本被配置为在满足所述备选条件中的第一条件的基础上,解锁所述第一交易的输出;

所述第二交易的第二版本具有包含解锁脚本的输入和指向所述第一交易的输出的指针,其中所述解锁脚本被配置为在满足所述备选条件中的第二条件的基础上,而非所述第一条件,解锁所述第一交易的输出。

包含在区块链中的交易的延展性

技术领域

[0001] 本公开涉及在区块链背景中的延展性现象,即在不导致交易无效的情况下更改交易的某些未签署部分的能力。

背景技术

[0002] 区块链指分布式数据结构的一种形式,其中在对等(P2P)网络多个节点中的每个节点维护区块链副本。区块链包括一连串数据块,其中每个区块包括一个或多个交易(transaction)。每个交易均可指向序列中的先前交易。交易可以提交至网络以包含在新的区块中。

[0003] 区块链中的交易通常用于传递数字资产,即作为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交易输出中储存附加用户数据。现代区块链在单一交易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。比如,这可用于在区块链中存储电子文件,或者甚至音频或视频数据。

[0004] 网络中的每个节点都可以具有转发和存储中的任意一种或全部角色。每个转发节点将(有效的)交易传播到一个或多个其他节点,从而在它们之间将交易传播到整个网络节点。为将交易记录在区块链中,一方将交易发送到拟传播网络的一个节点。每个节点被配置为遵守相同的节点协议,该协议将包含一个或多个条件来保证交易有效。无效交易将不会被传播。假定交易已验证并因此在区块链上被接受,附加用户数据将作为不可改变的公共记录,继续储存在P2P网络中的各个节点处。

[0005] 在“基于输出的”模型(有时称为“基于UTXO的模型”)中,给定交易的数据结构包括一个或多个输入和一个或多个输出。任何可花费的输出都包括指定数字资产金额的元素,有时称为UTXO(“未花费的交易输出”)。输出可进一步包括指定赎回输出条件的锁定脚本。每个输入均包括指向先前交易中的此类输出的指针,并且可进一步包括用于解锁所指向输出的锁定脚本的解锁脚本。因此设想一对交易,称为第一交易和第二交易(或“目标”交易)。第一交易包括指定数字资产金额的至少一个输出,包括定义解锁输出的一个或多个条件的锁定脚本。第二目标交易包括至少一个输入,包括指向第一交易的输出的指针,以及用于解锁第一交易的输出的解锁脚本。

[0006] 在此等模型中,当第二目标交易被发送到P2P网络传播并且记录在区块链中时,每个节点应用的有效性条件之一是:解锁脚本满足第一交易的锁定脚本中定义的要求。目标交易有效的另一个条件是:第一交易的输出尚未被另一个有效交易赎回。根据这些条件中的任何一个发现目标交易无效的任何节点都不会传播该交易,也不会包含该交易。

[0007] 假设目标交易是将一定数量的数字资产从第一方(“爱丽丝”)传送给第二方(“鲍勃”)。先前第一交易的锁定脚本中定义的要求之一通常是:目标交易的解锁脚本中包含爱丽丝的加密签名。签名必须由爱丽丝签署目标交易的一部分来产生。至于是哪一部分,可以由解锁脚本灵活定义,或者可以是节点协议的固有特性,具体取决于所使用的协议。然而,拟签署的部分都通常不包含目标交易的一些其他部分,例如解锁脚本本身的部分或全部。

[0008] 这使得“延展性”成为了可能。即,目标交易中未签署的部分可进行修改(“延展”),而不会使交易无效。延展性通常是密码学中的一个已知概念。利用延展性,消息可以被恶意修改但仍被认为是真实的,所以延展性通常被视为存在安全问题。在区块链的背景下,延展性并不一定构成问题,而只不过是一种奇妙的人类产物。利用延展性,可以修改交易的某个部分而不会使其无效。

[0009] 最近,有人提议专门利用延展性,以便将交易用作媒体数据的载体。数据内容可以包含在交易的解锁脚本中,然后该交易通过称作“支付通道”的侧信道在各方之间发送。其中一方延展交易以删除数据,并将延展版本继续发送到P2P网络(然而如果不删除该数据,则交易会使得区块链膨胀)。

发明内容

[0010] 本公开认识到另一种方式,其中延展性或者更通俗地说可更新性可以作为积极特性加以利用。特别地,还知道一对交易中的第一交易可以在其锁定脚本中包括多个不同的解锁条件。通常来说,只会创建一个目标交易版本,以基于一个条件或另一个条件赎回第一交易的输出。但是,本文认为存在两个版本实际上是有用的:具有解锁脚本的目标交易的第一版本基于第一条件赎回第一交易,第二版本基于第二条件赎回第一交易。第二更新版本可以通过延展第一版本来创建,或从头开始创建新版本。两个版本可以并行存在,也可以相继存在。两个版本都不能有效地赎回第一交易,但第一版本的存在可以作为鲍勃的“备份”,以防不能满足必要条件而导致不满足第二条件。

[0011] 根据本文公开的一个方面,提供了一种在区块链中记录第一方和第二方之间的目标交易的方法。该方法包括通过所述第一方或第二方的计算机设备:获取所述目标交易的第二更新版本,相对于所述目标交易预先存在的第一版本进行更新;发送所述目标交易的所述更新版本,而非所述第一版本,以通过节点网络传播,并且记录在至少部分节点的每个节点维护的区块链副本中。所述目标交易包括输入,所述输入包含解锁脚本和指向第一交易的第一输出的指针,所述第一输出包含锁定脚本,所述锁定脚本指定用于解锁所述第一交易的所述第一输出的多个备选条件。所述目标交易的第一版本的所述解锁脚本被配置为在满足所述备选条件中的第一条件的基础上,解锁所述第一交易的所述第一输出,所述更新版本的所述解锁脚本被配置为在满足所述备选条件中的第二条件的基础上,而非所述第一条件,解锁所述第一交易的所述第一输出。

[0012] 对于包括目标交易的多个交易中的每一个交易而言,至少部分网络节点被配置为在交易有效的情况下传播每个交易,并且至少部分节点被配置为在交易有效的情况下在该节点的区块链副本中记录每个交易,其中目标交易的有效性取决于根据所述条件中的任何一个解锁第一交易的输出的解锁脚本,但是一旦在任何给定的节点验证了其中一个版本,则另一个版本将在该节点被视为无效。

[0013] 因此,第二方(“鲍勃”)不会发送两个版本以传播并且记录在区块链中,因为两个版本都无法有效地赎回第一交易。他发送其偏好的拟传播和记录的第二版本,但第一版本存在的事实可以使其在未满足第二版本的必要条件的情况下,转向记录第一版本。

[0014] 比如,在第一示例性用例中,第一交易(Tx_1)包括第一方(“爱丽丝”)针对第二方(“鲍勃”)提供的服务进行支付。第一交易可以由爱丽丝、鲍勃或第三方创建。它可能已经在

链上,或已发送待稍后进行记录。无论采用哪种方式,第一交易的输出中的锁定脚本指定至少两个备选条件(即第一条件和第二条件)用于解锁在该输出中定义的支付。第一条件不需要在目标交易的解锁脚本中包含爱丽丝的签名,但对鲍勃提出了一些其他繁琐的要求,诸如要求解锁脚本中包含大量预定义的数据或者鲍勃的一段敏感或私人数据(鲍勃不希望在区块链上发布该数据)。第二条件要求爱丽丝的签名包含在目标交易的解锁脚本中,但没有强加第一条件的繁琐要求。

[0015] 爱丽丝还向鲍勃提供目标交易的第一版本(Tx_p),或者由鲍勃自己创建(或者由第三方创建)。第一版本被配置为基于第一条件解锁第一交易 Tx_1 的输出,而第二版本(Tx_p')将被配置为基于第二条件解锁。然后,鲍勃提供服务。如果爱丽丝感到满意并且诚实,那么作为回应,她会向鲍勃提供目标交易 Tx_p' 的第二版本,包括她的签名(或至少向鲍勃或第三方提供她的签名以组装到第二版本)。但是如果爱丽丝违背协议,鲍勃仍然可以选择发送现存的、不太优选的第一版本 Tx_p ,以便根据这个不太优选的第一条件赎回第一交易 Tx_1 的输出。

[0016] 在第二示例性用例中,所述方法可包括通过所述第二方的计算机设备:将数据部分的序列流式传输到所述第一方,以所述序列中的最终部分结束;响应所述数据部分的每一个相应部分,从所述第一方接收回所述第一交易的相应实例,每个实例的所述第一输出针对所述相应部分指定向所述第二方支付的款项;其中所述款项随着每个数据部分的增加而增加(例如:所述增加与迄今为止发送的数据部分的数量成线性比例)。在此等实施例中,所述目标交易的更新版本的获取包括在所述序列中的最终部分之后从所述第一方接收所述更新版本。

[0017] 比如,每个数据部分可以是媒体内容项(例如音频和/或视频内容)的不同部分。或者,每个数据部分是用于访问服务单元(诸如气、水、电等公用设施;或租赁房产、车辆或其他实物)的不同密钥。

[0018] 第一交易(Tx_1 、 Tx_2 、 Tx_3 ……)的实例将被网络的每个节点识别为基本相同的交易的实例,例如因为第一输出在每个实例中都具有相同的输出标识符(例如UTXO标识符)。因此,只有一个实例可以记录在区块链中。一旦一个实例被任一给定节点接受为有效,那么任何尝试将其记录为下一版本的行为都将被视为无效而因此被该节点拒绝。此外,一旦发现第一交易的一个实例在任一给定节点上被目标交易(Tx_p 或 Tx_p')的其中一个版本有效赎回,那么试图赎回第一交易的任何实例的任何其他目标交易都将被该节点视为无效,并且因此不会由该节点传播或记录在区块链中。

[0019] 因为目标交易(Tx_p 或 Tx_p')的任何版本都将指向基本相同的第一交易的实例的相同输出ID(只是定义不同的数额),只有一个目标交易可以有效地赎回该输出。一旦发现第一交易(Tx_1 、 Tx_2 、 Tx_3 ……)的一个实例在任一给定节点上被目标交易(Tx_p 或 Tx_p')的一个版本有效赎回,那么试图赎回第一交易的任何实例的任何其他目标交易都将被视为无效,并且因此不会由该节点传播或记录在区块链中。尽管如此,因为每个实例中的支付还随着每一部分数据的交换而增加,所有第二方(“鲍勃”)必须要做的是发送目标交易 Tx_p' 的一个版本,以赎回拟传播并且记录在区块链中的第一交易 Tx_n 的最后或最近的实例。然后,他将收到基于单对交易发送至该点的全部数据部分的全额付款。相反,如果第一方(“爱丽丝”)为每个数据部分发送单独的款项,这会使得区块链膨胀并且导致更多的网络流量,因为鲍勃

将不得不为每个数据部分发送单独的交易(例如,音频或视频内容的每个数据包或数据块)。

[0020] 如果爱丽丝在序列结束之前在任意点停止发送第一交易(Tx_1 、 Tx_2 、 Tx_3 ……)的实例,鲍勃仍然可以选择停止向爱丽丝发送更多数据部分,并将目标交易 Tx_p 的第一版本发送到网络,以赎回在该点之前发送的数据部分的款项(因此仅丢失发送的最新数据部分)。相反,如果鲍勃在任意点停止发送数据部分,爱丽丝则可以选择停止发送第一交易 Tx_1 的更多实例,并且鲍勃将只能赎回爱丽丝迄今为止收到的数据部分的款项。

[0021] 在特定的可选实施方式中,第一交易可包括指定输入金额的一个或多个第一输入,第一交易的第一输出可指定第一款项,第一交易可进一步包括一个或多个指定一个或多个进一步款项的进一步输出,使得款项的总金额大于输入金额,其中第二方从第一方接收的第一交易可不包括弥补差额的其他输入。如果指定付款总额大于总输入金额,网络的节点将因为无效而拒绝第一交易。在此等实施例中,该方法包括第二方向第一交易的最终或最新实例添加第二输入以弥补差额,并且发送添加第二输入的第一交易,以通过网络传播并且记录在区块链中。

[0022] 作为该方法的示例性实施方式,进一步输出包括第二输出和第三输出,第二输出指定向所述第一方支付的等于输入金额减去第一款项的第二款项,第三输出指定向第二方支付的等于第二款项的第三款项。

[0023] 此等实施例可以防止第一方(“爱丽丝”)通过发送她自己的目标交易来赎回一个早期实例来欺骗系统,从而防止鲍勃赎回一个稍后实例(或者实际上是任一实例),因为这样做时爱丽丝不得不添加额外输入,从而使其产生更多的数字资产。因此,不值得爱丽丝欺骗系统。

[0024] 在另一示例性用例中,至少部分备选条件中的每一个条件可对应于不同的投票选择,该方法可包括在获取目标交易的更新版本之前:访问或使第三方可以访问第一版本中的解锁脚本,作为投票意图的非约束性指示。如果轮询指示从未真正转换为投票,则第一版本可以保留作为轮询指示,以体现轮询与最终投票结果有何不同。

[0025] 根据本文公开的又一方面,提供了一种执行该方法的程序,和/或被编程用于执行该方法的第二方的计算机设备。程序或计算机设备可以被配置为使第二方能够选择性地发送拟通过网络传播并且记录在区块链中的第一或第二版本,要么通过手动选项来选择任一版本,和/或通过自动功能配置为在第一事件时自动发送第一版本,并进一步配置为在第二事件时自动发送拟传播和记录的第二版本,而不是第二版本。

附图说明

[0026] 为了帮助理解本公开的实施例并显示如何实施此等实施例,现将仅通过举例的方式参考附图进行说明,其中:

[0027] 图1是实现区块链的系统的示意性框图,

[0028] 图2示意性地示出了可记录在区块链中的交易的一些示例,

[0029] 图3是实现区块链的另一系统的示意性框图,

[0030] 图4是客户端应用程序的示意性框图,

[0031] 图5是可由图4的客户端应用程序表示的示例性用户界面的示意性模型,

- [0032] 图6是一组交易的示意图，
- [0033] 图7是示出一种流式传输数据的方法的信令图，
- [0034] 图8是示出在图7的方法的示例性实施方式中第一交易的实例的输入和输出值的图，
- [0035] 图9是在图7的方法中第一交易的示例性交易格式，
- [0036] 图10是示出投票方法的信令图，
- [0037] 图11示出了用于实现图10的方法的一组示例性交易。

具体实施方式

[0038] 延展性是密码学中的一个现有概念，其涉及到安全问题。利用延展性，消息可以被恶意修改但仍被接受为真实的。数字签名方案旨在解决这一问题。但是，在区块链中，延展性系指在不导致整个交易无效的情况下修改交易的至少一部分的能力。通过相关形式的加密签名（例如ECDSA签名）签署的交易中的任何信息都不存在延展的可能性。与延展性有关的任何安全问题均由于不恰当的实施方式引起，而不是协议本身。

[0039] 下面将探讨延展性作为一种有用的特性如何推进快速、安全且免信任的支付通道。具体的思路是确定交易中哪个或哪些部分没有或不必要签名（例如，通过ECDSA签名）。公开的方案将利用以下事实：交易的每个输入的解锁脚本（例如‘scriptSig’字段）中的任何内容都没有任何签名。实施例还可以利用SIGHASH标志，从而使修改交易更灵活且不会使交易无效。

[0040] 下面举个例子。当在区块链上进行数据交换时，常见的做法是通过哈希难题来促使数据显示和同意付款同时发生。为了避免这种情况的发生，可以构建交易来要求付款，使得只需满足下列两种条件的其中之一即可：i) 提供“数据+鲍勃的签名”；或者ii) 提供“爱丽丝的签名+鲍勃的签名”。

[0041] 鲍勃将通过提供数据和他的签名构建交易来要求付款，并且将该交易发送至爱丽丝。然后，爱丽丝用她的签名替代该数据，并将该交易广播到网络。或者，鲍勃获得爱丽丝的签名，用它替代该数据，并且广播到网络。无论采用哪种方式，由于该数据并不构成经鲍勃签名的消息的一部分，因此用爱丽丝的签名替代该数据将不会使该交易无效。此外，因为输入满足条件ii)，所以该交易仍然有效。如果爱丽丝未将该交易广播到网络或者未提供她的签名，那么鲍勃仍然可以按照条件i) 将原始交易进行广播来要求付款（因为鲍勃必须上传相当大的数据和/或专有数据，所以这并不是优选的）。可通过请求确认、折扣激励的方式鼓励爱丽丝提供她的签名，或者鼓励爱丽丝对鲍勃提供的良好服务进行奖励。

[0042] 系统概述

[0043] 图1示出了用于实现区块链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网。分组交换网络101包括多个节点104，该多个节点被设置在分组交换网络101内形成对等（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包括含一个或多个处理器的处理装置，例如一个或多个中央处理单元（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多个存储器单元，其采用一个或多个存储器介质，例如诸

如硬盘等磁介质、诸如固态硬盘(SSD)等电子媒介、闪存或电可擦只读存储器和/或诸如光盘驱动器等光学介质。

[0044] 区块链150包括一系列数据区块151,其中P2P网络160多个节点中的每个节点维护区块链150的相应副本。链中的每个区块151均包括一个或多个交易152,其中该上下文中的交易指一种数据结构。数据结构的性质将取决于用作交易模型或计划的一部分的交易协议类型。给定的区块链通常全程使用一个特定的交易协议。在一种常见的交易协议中,每个交易152的数据结构至少包括一个输入和至少一个输出。每个输出指定一个金额,该金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)。每个输入指向先前交易152的输出,从而链接这些交易。

[0045] 至少一些节点104扮演转发节点104F的角色,这些节点转发并因此传播交易152。至少一些节点104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。给定节点104可以是转发节点104、存储节点104S或其中节点的任意组合。

[0046] 在给定的当前交易152j中,输入(或每个输入)包括指针,该指针引用交易序列中先前交易152i的输出,指定该输出将在当前交易152j中被赎回或“花费”。通常而言,先前交易可以是池154或任何区块151中的任何交易。为了保证当前交易有效,当创建当前交易152j或甚至将当前交易152j发送到网络106时,先前交易152i不一定要存在,但是先前交易152i必须存在并且有效。因此,本文中的“先前”系指通过指针连接的逻辑序列中的前期,不一定是时间序列中创建或发送的时间。因此,并不一定排除交易152i,152j是无序创建或发送的可能(详见下文中有孤立交易的描述)。先前交易152i同样可以被称为先行或前期交易。

[0047] 当前交易152j的输入还包括先前交易152i输出被锁定的用户103a的签名。反过来,当前交易152j的输出可以加密锁定至新用户103b。因此,当前交易152j可将先前交易152i输入中定义的金额转移到当前交易152j输出中定义的新用户103b。在某些情况下,交易152可具有多个输出,以在多个用户间分割输入金额(其中一个可以是原始用户103a,以便进行变更)。在某些情况下,交易还可以具有多个输入,将一个或多个先前交易的多个输出的金额汇总在一起,并重新分配到当前交易的一个或多个输出。

[0048] 上述可称为“基于输出型”交易协议,有时也称为未花费的交易输出(UTX0)型协议(其中输出被称为UTX0)。用户的总余额不是用区块链中储存的任何一个数字定义;相反,用户需要特殊“钱包”应用程序105,以整理该用户的所有UTX0值,这些UTX0值分散在区块链151的许多不同交易152中。

[0049] 作为基于账户型交易模型的一部分,另一种类型的交易协议可称为“基于账户型”协议。在基于账户的情况下,每个交易均不通过参考过去交易序列中先前交易的UTX0来定义转移的金额,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单独存储到区块链中,并不断更新。本公开涉及一种基于输出的模型,而不是基于账户的模型。

[0050] 无论采用何种类型的交易协议,当用户103希望执行新交易152j时,其希望将新交易从其计算机终端102发送至P2P网络106的一个节点104(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根据应用于每个节点104的节点协议检查交易是否有效。节点协议的详细信息将与相关区块链150中使用的交易协议类型相对应,一起形

成整个交易模型。节点协议通常要求节点104检查新交易152j中的加密签名是否与预期签名相匹配,这取决于交易152的有序序列中的上一个交易152i。在基于输出型案例中,这可包括检查新交易152j输入中包含的用户加密签名是否与新交易花费的先前交易152i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交易152j输入中的加密签名是否解锁新交易输入所指向的上一个交易152i的输出。在一些交易协议中,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固定,或可通过其组合固定。无论采用哪种方式,如果新交易152j有效,则当前节点将新交易转发至P2P网络106中一个或多个其他节点104。这些节点104中的至少一些还作为转发节点104F,根据相同的节点协议应用相同的测试,从而将新交易152j转发至一个或多个进一步的节点104,依此类推。通过这种方式,新交易在整个节点104网络中进行传播。

[0051] 在基于输出型模型中,给定输出(例如:UTX0)是否花费的定义是,根据节点协议,其是否通过另一个随后交易152j的输入有效赎回。交易有效的另一个条件是其试图花费或赎回的先前交易152i的输出尚未被另一个有效交易花费/赎回。同样,如果无效,交易152j将不会在区块链中传播或记录。这可防止重复花费,即花费者对同一个交易的输出花费超过一次。

[0052] 一旦创建后,则不可修改区块151,因为其根据相同的协议在P2P网络106中的每个存储节点104S进行识别和维护。区块指针155还向区块151施加顺序。因为交易152记录在P2P网络106各存储节点104S的有序区块中,因此提供了不可改变的交易公共分类账。

[0053] 每个转发节点104M和/或存储节点104S可以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均可采用用户终端或一组联网的用户终端的形式。

[0054] 每个节点104的存储器均储存被配置为在节点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和处理交易152。应当理解的是,本文节点104的任何行动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,本文的“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。

[0055] 同时连接至网络101的还包括作为消费用户角色的多方103中的每一方的计算机设备102。他们作为交易中的付款人和收款人,但不一定代表其他方参与传播交易。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,第二方103b及其相应的计算机设备102b。应当理解的是,更多这样的当事方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。为进行说明,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。

[0056] 每一方103的计算机设备102包括相应的处理装置,其包含一个或多个处理器,例如一个或多个中央处理单元、中央处理单元、其他加速器处理器、特定应用程序处理器和/或现场可编程门阵列。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多个存储器单元,其采用一个或多个存储器介质,例如诸如硬盘等磁介质、诸如固态硬盘等电子媒介、闪存或电可擦只读存储器和/或诸如光盘驱动器等光学介质。每一方103计算机设备102上的存储器储存包含被设置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,本文

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可穿戴设备。给定方103的计算机设备102还可包括一个或多个其他网络资源,诸如通过用户终端访问的云计算资源。

[0057] 客户端应用程序或软件105最初可通过适当的计算机可读存储介质提供至任何给定方103的计算机设备102,例如从服务器下载,或提供在移动式存储设备上,诸如可移动固态硬盘、闪存密钥、可移动电可擦只读存储器、可移动磁盘驱动器、软盘或磁带、光盘(诸如CD或DVD ROM)或可移动光驱等。

[0058] 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、签署和发送拟在整个节点网络104中传播的交易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金额。在基于输出型系统中,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152交易的输出中定义的金額。

[0059] 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的至少一个转发节点104F。这可以启用客户端105的钱包功能,将交易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查询相应方103为接收方的任何交易(或实际上检查区块链150中的其他方交易,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交易信任的公共设施)。每个计算机设备102上的钱包功能被配置为根据交易协议制定和发送交易152。每个节点104在转发节点104F转发交易152的情况下运行被配置成根据节点协议验证交易152的软件,以在整个网络106中进行传播。交易协议和节点协议相互对应,给定交易协议和给定节点协议一起实现给定的交易模型。区块链150中的所有交易152均采用相同的交易协议(尽管交易协议可允许其内存在不同的交易子类型)。网络106中的所有节点104采用相同的节点协议(尽管其可根据子类型的规则区分处理不同的交易子类型,且不同的节点还可扮演不同的角色,因此实现协议的不同对应方面)。

[0060] 如前所述,区块链150包括一些列区块151,其中每个区块151包括一组已创建的一个或多个交易152。每个区块151还包括区块指针155,其指向链中先前创建的区块151,以定义区块151的顺序。区块链150还包括有效交易池154,其等待包含在新的区块中。每个交易152包括指向上一个交易的指针,以定义交易序列的顺序(注:交易152的序列可进行分支)。区块151的链一直追溯到创始区块(Gb) 153,该创始区块是链中的第一个区块。早期链150中的一个或多个原始交易152指向创始区块153,而非先前交易。

[0061] 当给定方103(比方说爱丽丝)希望发送拟包含在区块链150的新交易152j时,她根据相关的交易协议(使用其客户端应用程序105中的钱包功能)制定新的交易。然后,她将交易152从客户端应用程序105发送至其连接的一个或多个转发节点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接收新交易152j时,其根据节点协议及其相应的角色进行处理。这包括首先检查是否新接收的交易152j符合“有效”的特定条件,具体示例稍后将详细描述。在一些交易协议中,验证条件可通过交易152中包含的脚本在每个交易的基础上进行配置。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。

[0062] 如果新接收的交易152j通过有效性测试(即:经过“验证”的条件下),任何接收交易152j的存储节点104S将把新验证交易152添加至节点104S维护的区块链150副本的池154中。进一步地,任何接收交易152j的转发节点104F随后将传播验证交易152至P2P网络106中的一个或多个其他节点104。由于每个转发节点104F应用相同的协议,因此假定交易152j有效,这意味着交易很快将通过整个P2P网络106进行传播。

[0063] 图2示出了示例性交易协议。这是基于UTX0协议的示例。交易152(简称“Tx”)是区块链150的基本数据结构(每个区块151包括一个或多个交易152)。下文的描述参考基于输出或基于“UTX0”的协议。但这并不限于所有可能的实施例。

[0064] 在基于UTX0的模型中,每个交易(“Tx”)152包括数据结构,其包括一个或多个输入202和一个或多个输出203。每个输出203可包括未花费的交易输出(UTX0),其可用作另一个新交易的输入202来源(如果UTX0未被赎回)。UTX0指定了数字资产金额(价值存储手段)。其还可包含其来源交易的交易ID以及其他信息。交易数据结构可包括标题201,其可包括输入字段202和输出字段203的大小指示符。标题201还可包括交易的ID。在实施例中,交易ID是交易数据(不含交易ID本身)的哈希值,且储存在原始交易152的标题201中。

[0065] 即爱丽丝103a希望创建转移相关数字资产金额至鲍勃103b的交易152j。在图2中,爱丽丝的新交易152j标记为“Tx₁”。其获取在序列中先前交易152j输出203锁定至爱丽丝的数字资产金额,并至少转移一部分此等金额至鲍勃。图2中的先前交易152j标记为“Tx₀”。Tx₀和Tx₁只是一个任意的标签。其不一定意味着Tx₀指区块链151中的第一交易和Tx₁指池154中的下一个交易。Tx₁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)交易。

[0066] 当爱丽丝创建其新交易Tx₁时,或至少在她将其发送至网络106时,先前交易Tx₀可能已经验证并包含在区块链150中。此时可能已包含在一个区块151中,或可仍在池154中等待,在这种情况下,其将很快被包括在新块151中。或者,Tx₀和Tx₁可以创建并一起发送至网络102,或如果节点协议允许缓冲“孤立”交易,Tx₀甚至可以在Tx₁之后发送。本文交易顺序上下文中使用的“先前”和“后续”一词指交易中指定的交易指针所定义的序列中的交易顺序(哪个交易指向哪个其他交易等等)。它们同样可以用“前期”和“后续”、“先行”和“后继”、“上代”和“子代”或者诸如此类的词替代。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,指向先前交易(先行交易或“父交易”)的后续交易(后继交易或“子交易”)在父交易进行验证之前不会进行验证。在父交易之前到达节点104的子交易被视为孤立交易。根据节点协议,其可被丢弃或缓冲一段时间,以等待父交易。

[0067] 先前交易Tx₀的一个或多个输出203中的一个包括特定的UTX0,标记为UTX0₀。每个UTX0包括指定UTX0表示的数字资产金额的值以及锁定脚本,该锁定脚本定义后续交易输入202中解锁脚本必须满足的条件,以供后续交易进行验证,从而成功赎回UTX0。通常情况下,锁定脚本将金额锁定至特定方(该金额的交易受益人)。即,锁定脚本定义解锁条件,通常包括以下条件:后续交易输入中的解锁脚本包括先前交易被锁定一方的加密签名。

[0068] 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。此等语言的特定示例称为“脚本”(S大写)。锁定脚本指定花费交易输出203所需的信息,例如爱丽丝签名的要求。解锁脚本出现在交易的输出中。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

锁脚本出现在交易的输入202中。

[0069] 因此在示出的示例中, Tx_0 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 $UTXO_0$ 的后续交易有效)。 $[Checksig P_A]$ 包括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, 通过其交易ID ($TxID_0$), 其在实施例中的是整个交易 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 $\langle Sig P_A \rangle$, 其包括爱丽丝的加密签名, 该签名由爱丽丝将其密钥对中的私钥应用到预定的部分数据进行创建 (有时在密码学中称为“消息”)。 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。

[0070] 当新交易 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多个标准)。在实施例中, 这涉及并置两个脚本:

[0071] $\langle Sig P_A \rangle \langle P_A \rangle || [Checksig P_A]$

[0072] 其中“||”表示并置, “ $\langle \dots \rangle$ ”表示将数据放在堆栈上, “[$\dots$]”表示由解锁脚本组成的函数 (在此示例中指基于堆栈的语言)。同样地, 使用普通堆栈, 脚本就可以连续运行, 而不是连接脚本。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 (包含在 Tx_0 输出的锁定脚本中), 以验证 Tx_1 输入中的锁定脚本是否包含爱丽丝签署预期部分数据时的签名。预期部分数据本身 (“消息”) 也需要包含在 Tx_0 中, 以便执行此验证。在实施例中, 签署的数据包含整个 Tx_0 (因此需要包含一个单独的元素, 明文指定签署的部分数据, 因为其本身便存在)。

[0073] 本领域技术人员将熟悉通过公私密码进行验证的细节。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等其他实体可验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哈希处理, 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验证签名。因此, 需要注意的是, 本文中提及特定数据片段、部分交易内容或诸如此类的签名可表示对该数据片段或部分交易内容的哈希值进行签名。

[0074] 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验证), 则节点104认为 Tx_1 有效。如果是存储节点104S, 这意味着其将添加至交易池154。如果是转发节点104F, 则其将交易 Tx_1 转发到网络106中的一个或多个其他节点104, 从而将在整个网络中传播。一旦 Tx_1 被验证并包含在区块链150中, 这将把 Tx_0 中的 $UTXO_0$ 定义为已花费。请注意, Tx_1 仅在花费未花费的交易输出203时才有效。如果试图花费另一交易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还需要检查先前交易 Tx_0 中引用的 $UTXO$ 是否已经花费 (已经形成另一有效交易的有效输入)。这是为何区块链150对交易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, 标记已花费交易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中形成了另一有效交易的有效输入。

[0075] 请注意, 在基于 $UTXO$ 的交易模型中, 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金额, 而同时又花费另一部分。但 $UTXO$ 的金额可以在下一个交易的多个输出之间分割。例如, Tx_0 的 $UTXO_0$ 中定义的金额可以在 Tx_1 中的多个 $UTXO$ 之间分

割。因此,如果爱丽丝不想将UTXO₀中定义的所有金额都给鲍勃,她可以使用剩余部分在Tx₁的第二输出中自己找零钱,或者支付给另一方。

[0076] 同时,还应当注意,如果给定交易152的全部输出203中指定的总金额大于其全部输入202所指向的总金额,则说明大多数交易模型中的交易是无效的。因此,此等交易将不会被传播。

[0077] 爱丽丝和鲍勃的数字资产由区块链150中其任何交易152锁定的未花费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的各种交易152的UTXO中。区块链150中均不定义给定方103的总余额。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交易中尚未花费的不同UTXO值整理在一起。通过查询任何存储节点104S储存的区块链150副本可实现这一点,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。

[0078] 请注意,脚本代码通常用示意图表示(即,非精确的语言)。例如,写入[Checksig P_A]意味着[Checksig P_A]=OP_DUP OP_HASH160<H(P_A)>OP_EQUALVERIFY OP_CHECKSIG。“OP...”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’输入验证的交易中仍保留附加要求,诸如哈希难题。再如,OP_RETURN是脚本语言操作码,用于创建交易的不可花费输出,其可以将元数据储存在交易中,从而将元数据不可改变地记录在区块链150中。例如,元数据可包括需储存在区块链中的文件。

[0079] 签名P_A是数字签名。在实施例,这基于使用椭圆曲线secp256k1的椭圆曲线数字签名算法。数字签名对特定的数据段进行签名。在实施例,对于给定交易,签名将签署部分交易输入、全部或部分交易输出。其签署输出的特定部分取决于SIGHASH标志。SIGHASH标志是包含在签名末尾的4字节代码,用于选择签署的输出(并因此在签名时固定)。

[0080] 锁定脚本有时称为“scriptPubKey”,指其包括相应交易锁定方的公钥。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验证签名。更通俗地说,脚本语言可用于定义任何一个或多个条件。因此,可以优选更为通用的术语“锁定脚本”和“解锁脚本”。

[0081] 图3示出了用于实现区块链150的系统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交易152时不想将该交易(仍未)发布到P2P网络106或使其进入区块链150,可以采用此等通信,直到其中一方选择将该交易广播到网络106。此等侧信道301有时候被用作例如“支付通道”。

[0082] 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102a,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可包括经由一种或多种联网技术或者通信介质的任何一个或多个链路,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。

[0083] 示例性定义

[0084] 下面是一些实施方式中可能采用的示例性定义。需注意,这些示例性定义并不会完全限制所有可能的实施方式,而只是为了帮助理解特定的可能实施方式,诸如可能用在下文所述的示例性用例的一些可能实施方式中的定义。

[0085] 定义1:交易。交易系指包含输入和输出的消息。它还可以包括协议版本号和/或锁定时间。协议版本是指交易协议的版本。锁定时间将会稍后单独进行解释。

[0086] 定义2:输入。交易的输入构成有序列表。该列表中的每一个条目包含输出点(未花费的交易输出的标识符)和scriptSig(解锁脚本)。它还可以包括序列号。

[0087] 定义3:输出。交易的输出构成有序列表。该列表中的每一个条目包含数值(以基本单位表示的数字资产金额)和scriptPubKey(锁定脚本)。

[0088] 定义4:输出点。输出点通过交易IDTxID和索引号i进行唯一定义。它是指交易TxID输出中的第i个条目,表示未花费的交易输出(UTXO)的唯一位置。此处“未花费的”一词指输出点从未在任何有效的后续交易中出现过。

[0089] 定义5:scriptSig。这是解锁或花费与给定输出点对应的UTXO所需的信息。在标准的交易中,这种信息通常是指ECDSA签名。因此,该脚本被称为’scriptSig’。但是,解锁输出点所需的信息可以是满足UTXO锁定条件的任何数据。

[0090] 定义6:scriptPubKey。这是指锁定特定UTXO相关资金的脚本。当且仅当scriptSig被添加到scriptPubKey中,并且组合脚本的执行显示为TRUE时,该资金才能被解锁并且被花费。如果不是这种情况,那么交易无效并且将会被拒绝。由于该脚本通常包含用于标准交易的ECDSA公钥的哈希值,所以它被称为“scriptPubKey”。

[0091] 在下一个定义中,如果引用签署输入,则意味着签署不包括scriptSig部分(见定义2)的输入。

[0092] 定义7:SIGHASH标志。当提供ECDSA签名时,还需要添加下列SIGHASH标志之一。

[0093]	标志	功能含义
	SIGHASH_ALL	签署所有输入和输出
	SIGHASH_SINGLE	使用相同的索引签署输出和所有输入
	SIGHASH_NONE	签署所有输入, 不含输出
	SIGHASH_ALL ANYONECANPAY	签署其自身的输入和所有输出
	SIGHASH_SINGLE ANYONECANPAY	使用相同的索引签署其自身的输入和输出
	SIGHASH_NONE ANYONECANPAY	签署其自身的输入, 不含输出

[0094] 当把延展性作为特性讨论时,要寻找未通过ECDSA签名签署的交易中的信息。除了可以从拟签署的消息中排除的输入和输出之外,scriptSig的内容始终被排除在外。这是因为scriptSig被设计用作签名的占位符。

[0095] 定义8:区块链时间锁。通常而言,交易中可使用两种时间锁:绝对时间锁和相对时

间锁。绝对时间锁指定具体的时间点,发生在这个时间点之后的一些信息会被认为是有效的;而相对时间锁指定时间段,发生在这个时间段之后的一些信息才会被认为有效。在这两种情况下,当使用区块链时间锁时,可以用区块高度(挖到的区块数量)或经过时间(例如UNIX时间)来代理时间。

[0096] 区块链时间锁的另一个属性在于它们出现的位置以及它们所应用的交易方面。在这种意义上,又可以划分为两个类别时间锁:交易级时间锁,用于锁定整个交易;和脚本级时间锁,用于解锁特定的输出。这两个类别的时间锁均可用于实现绝对或相对时间锁。下表汇总了四种实现可基于上述属性创建的时间锁的机制。

		类型	
		绝对时间锁	相对时间锁
[0097]	交易级	nLocktime	nSequence
	脚本级	OP_CLTV	OP_CSV

[0098] 定义9:nLocktime。锁定时间(nLocktime)是指非负整数,表示区块高度或特定的Unix时间。如果交易只能在指定的区块或指定的时间后才能被添加到区块链中,那么从这个意义上讲,则是交易级时间锁。如果nLocktime被设置为小于500,000,000,则视为区块高度。如果其被设置为等于或大于500,000,000,则视为表示Unix时间。也就是说,1970年1月1日00:00:00之后的秒数。

[0099] 定义10:nSequence。序列号(nSequence)作为消息,表示交易版本。修改交易会使得序列号增大。nSequence的最大值是 $2^{32}-1$,并且通常而言,序列号将默认设置为该最大值,以表示交易已完成。交易的每个输入都定义nSequence值,该值指定输入所引用的UTXO被包含在区块中之后到该输入可用作有效输入之前所需的时间段。但是,这个特性通常已被禁用。

[0100] 定义11:CheckLockTimeVerify(OP_CLTV)。OP_CHECKLOCKTIMEVERIFY(OP_CLTV)操作码是脚本级绝对时间锁,可用于锁定未来某个特定时间或区块高度的交易的特定输出。如果UTXO被引用到交易中的当前Unix时间或区块高度小于UTXO被创建的Unix时间或区块高度以及在OP_CLTV操作码之前指定的参数,那么花费交易的脚本执行将失败。

[0101] 定义12:CheckSequenceVerify(OP_CSV)。OP_CHECKSEQUENCEVERIFY(OP_CSV)操作码是脚本级相对时间锁,可用于锁定未来某个特定时间段或区块数量的交易的特定输出。该与OP_CLTV的操作类似,不同之处在于提供给OP_CSV的参数表示相对时间。如果UTXO被引用到交易中的当前Unix时间或区块高度小于UTXO被创建的Unix时间或区块高度以及在OP_CSV操作码之前指定的参数,那么花费交易的脚本执行将失败。

[0102] 定义13:延展性。通常而言,区块链交易可能具有两大类别的延展性,两者均允许修改交易内容,且不导致输入中提供的签名无效。

[0103] 为了说明这两种类别,以首次交易Tx为例,该交易包含一个输入、该输入中的一个签名和一个输出。

[0104] 第1类:脚本级延展性。这种延展性利用了拟使用脚本操作码OP_CHECKSIG检查的签名不会签署交易中任何输入的脚本字段的事实。这一事实允许我们在交易Tx上生成签名,修改输入脚本,使得交易Tx'不等同于Tx,并且仍然可以视Tx和Tx'为区块链共识规则下由相同签名签署的有效交易消息。

[0105] 第2类:输入输出级延展性。这种延展性依赖于SIGHASH标志的使用,而不是交易中采用的SIGHASH ALL。如果交易Tx有输入签名且该签名使用其他五个SIGHASH标志组合中任意一个,那么可以添加输入或输出来创建不等同的交易Tx',使得两者在共识规则下将被视为有效的交易消息,而不需要修改签名。

[0106] 延展性特性

[0107] 图4示出了用于实现本公开方案的实施例的客户端应用程序105的示例性实施方式。客户端应用程序105包括交易引擎401和用户界面(UI)层402。根据上文讨论的方案以及稍后将进一步详细讨论的内容,交易引擎401被配置为实现客户端105的基础交易相关功能,诸如制定交易152,通过侧信道301接收和/或发送交易和/或其他数据,和/或发送交易以通过P2P网络106传播。按照本文公开的实施例,至少鲍勃的客户端105b的交易引擎401包括选择功能形式的应用功能403,其能够选择从鲍勃的相应计算机设备102发送两个或多个不同版本的目标交易(“Tx_p”和“Tx_p'”)中的哪一个,以通过P2P网络106传播进行验证,并且因此记录在区块链150中(传播和记录本身通过前面讨论的机制来实现)。需再次注意,该发送可包括直接从鲍勃的计算机设备102b发送目标交易到网络106的转发节点104F之一,或者发送目标交易到爱丽丝的设备102b或第三方的设备,以转发到网络106的节点104F之一。

[0108] UI层402被配置为通过相应用户的计算机设备102的用户输入/输出(I/O)方式呈现用户界面,包括通过设备102的用户输出方式向相应用户103输出信息,和通过设备102的用户输入方式接收来自相应用户103的输入。例如,用户输出方式可包括提供视觉输出的一个或多个显示屏(触摸或非触摸屏)、提供音频输出的一个或多个扬声器、和/或提供触觉输出的一个或多个触觉输出设备等。用户输入方式可包括例如一个或多个触摸屏的输入阵列(与用于输出方式的相同或不同);一个或多个基于光标的设备,诸如鼠标、轨迹板或轨迹球;一个或多个麦克风和语音或声音识别算法,用于接收语音或声音输入;一个或多个基于手势的输入设备,用于接收手动或身体手势形式的输入;或者一个或多个机械按钮、开关或控制杆等。

[0109] 注:虽然本文中的各种功能可以被描述为集成到同一客户端应用程序105中,但这并不一定构成限制,相反,它们可以在两个或多个不同应用程序组成的一套程序中实现,例如一个是另一个的插件。比如,交易引擎401的功能可以在单独的应用程序中实现,而不是在UI层402中实现,或者诸如交易引擎401的给定模块的功能可以在多个应用程序之间分割。同时,也不排除假设某些或所有描述的功能可以在比如操作系统层实现。在本文任何位置参考单个或给定应用程序105或诸如此类的情况下,应当理解的是这只是作为示例,并且更通俗地说,所描述的功能可以在任何形式的软件中实现。

[0110] 图5给出了用户界面(UI)500的示例的模型,其可由鲍勃的设备102b上的客户端应用程序105b的UI层402呈现。用户界面500包括至少两个用户可选择的选项501,502,这些选项可以通过用户输出方式(诸如两个屏幕上的按钮)或者菜单中的两个不同选项呈现为两个不同的UI元素。用户输入方式被设置成使用户103b(在这种情况下为鲍勃)能够选择其中一个选项,诸如通过点击或触摸屏幕上的UI元素,或者说出所需选项的名称(注:本文使用的“手动”一词仅用于与自动进行对比,而不仅限于手的使用)。应当理解的是,呈现和选择选项的特定方式并不重要。

[0111] 不管使用什么方式,每个选项对应于第一和第二目标交易(Tx_p和Tx_p')中的不同交

易。选择功能403被配置为与UI层402接口以启用下述功能。也就是说,如果鲍勃103b选择第一选项501,则这将导致交易引擎403发送目标交易 Tx_p 的第一版本,以通过网络106传播并且记录在区块链150中;但是,如果鲍勃103b选择第二选项502,则这将导致交易引擎403发送目标交易 Tx_p 的第一版本,以通过网络106传播并且记录在区块链150中。

[0112] 应当理解的是,图5中示出的UI 500只是一个图示模型,在实践中,它可包括一个或多个进一步的UI元素,为了简洁起见,未对其进行说明。

[0113] 作为替代方案或除了UI选项501,502之外,选择功能403可以被配置为在发送目标交易的第一和第二版本(Tx_p 或 Tx_p')之间执行自动选择,用于记录在区块链150中。这可以由预定事件或在预定超时后自动触发。比如,如果事件Y发生在超时之前,则功能403自动发送第二版本 Tx_p' ,以通过网络150传播,但是如果超时在事件Y之前发生,则功能403自动发送第一版本 Tx_p 。或者如果发生事件X,则功能403自动发送第一版本 Tx_p ,以通过网络150传播,但是如果发生事件Y,则功能403自动发送第二版本 Tx_p' 。原则上,几乎任何情况均可以被系统设计者配置为自动将第一和第二版本发送到网络150的情况。

[0114] 图6示出了根据本文公开的实施例使用的一组交易152。该组包括第零交易 Tx_0 、第一交易 Tx_1 和第二交易 Tx_p/Tx_p' 。需注意,这些名称只是便利标签。它们并不一定意味着这些交易将立即相继放置在区块151或区块链150中,也不意味着第零交易是区块151或区块链150中的初始交易。这些标签不一定意味着任何关于他们的交易被发送到网络106的顺序的信息。他们仅指逻辑序列,其中下一交易的输入指向一个交易的输出。需记住的是,在一些系统中,可以在其子交易之后将父交易发送至网络106(在这种情况下,“孤立”子交易将在一个或多个节点104处缓冲一段时间,同时等待父交易到达)。

[0115] 如果两个交易都包含引用第一交易的相同输出(例如相同UTXO)的输入,则可以说两个交易 Tx_p 或 Tx_p' 是(基本上)相同交易的多个版本。不同版本可通过满足该输出的不同解锁条件来提供不同的功能。不同版本还可都包含相同的输入签名(即,任一实例中的已签名消息都相同)。稍后讨论的一些实施例还可涉及第一交易 Tx_1 的不同实例(其中 $i=1,2,3,\dots$)。如果两个交易都包含引用同源交易(或“第零”交易) Tx_0 的相同输出(例如UTXO)的输入,则可以说本文的两个(或多个)交易是(基本上)相同第一交易的多个实例。他们可在满足相同解锁条件的基础上赎回该输入。但是,他们可包含不同的输入签名(即,任一实例中的已签名消息都不相同)。不同的实例可以基本上用作相同的功能,但是作为对不同的相应数据部分的支付,并且指定数字资产的增加金额。稍后将参考图7-图9和示例性用例2对此进行详细讨论。

[0116] 第零交易 Tx_0 还可称为就本发明而言的源交易,因为其充当锁定至爱丽丝103a的数字资产金额的来源。第一交易 Tx_1 还可称为就本发明而言的中间交易或条件交易,因为其充当有条件地从源交易 Tx_0 转移数字资产金额的中介。第二交易 Tx_p/Tx_p' 还可称为目标交易或者支付交易(因此下标为“P”),因为该交易将解锁其中一个条件并且为鲍勃(或者鲍勃代表的潜在受益人)支付款项。如前所述,目标交易有两个版本,即第一版本 Tx_p 和第二版本 Tx_p' 。这些交易将至少在一些时间点存在,在爱丽丝的计算机设备102a(第一方)、或鲍勃(第二方)的计算机设备102b、或第三方的计算机设备(未示出)或者这些的任何组合上显示。两个版本 Tx_p 和 Tx_p' 可同时存在一段时间,或者相继存在,或者在时间上部分重叠。

[0117] 如图6所示,源交易 Tx_0 包括至少一个输出203₀(例如 Tx_0 的输出0),其指定数字资产

的金额,并且进一步包含将该输出锁定至爱丽丝103a的锁定脚本。这意味着源交易 Tx_0 的锁定脚本要求至少满足一种条件,即试图解锁输出(并且因此赎回数字资产的金额)的任何交易的输入必须在其解锁脚本中包含爱丽丝的加密签名(即使用爱丽丝的公钥)。在这种意义上, Tx_0 的输出中定义的金额可以说是由爱丽丝拥有。该输出可以称为UTXO。就本发明而言, Tx_0 的输入指向的先前交易的输出并不特别重要(只要足以涵盖 Tx_0 的总输出)。

[0118] 在这种情况下,锁定源交易 Tx_0 的输出的交易是第一或中间交易 Tx_1 。因此, Tx_1 具有至少一个输入202₁(例如 Tx_1 的输入0),该输入包括指向 Tx_0 的相关输出的指针(所示示例中 Tx_0 的输出0),并且进一步包括被配置为根据该输出的锁定脚本中定义的条件解锁 Tx_0 所指向输出的解锁脚本,其至少要求爱丽丝的签名。 Tx_0 的锁定脚本所需的爱丽丝的签名需要签署 Tx_1 的一部分。在一些协议中, Tx_1 需要签署的部分可以是 Tx_1 的解锁脚本中定义的设置。例如,这可以通过SIGHASH标志设置,该标志附加至签名,为一个字节,因此就数据而言,解锁脚本显示为: $\langle \text{Sig } P_A \rangle \langle \text{sighashflag} \rangle \langle P_A \rangle$ 。或者,需要签署的部分可以仅仅是 Tx_1 的固定部分。无论采用哪种方式,拟签署的部分都通常不包含解锁脚本本身,并且可能不包括 Tx_1 的部分或全部输入。这意味着 Tx_1 的输入是可延展的。

[0119] 第一或中间交易 Tx_1 具有至少一个输出203₁(例如, Tx_1 的输出0,输出也可称为UTXO)。中间交易 Tx_1 的输出未对任何一方无条件锁定。就像 Tx_0 ,其具有至少一个输出(例如 Tx_1 的输出0),指定随后拟转移的数字资产金额,并且进一步包括锁定脚本,该锁定脚本定义解锁输出并且因此赎回该金额的所需内容。但是,该锁定脚本允许基于多个不同可能条件的任何一个解锁其输出,至少包括:i)第一条条件(“条件1”)和ii)第二条条件(“条件2”)。

[0120] 第二目标交易 Tx_p/Tx_p' 具有至少一个输入202_p(例如 Tx_p/Tx_p' 的输出0),该输出包括指向 Tx_1 的前述输出的指针(Tx_1 的输出0,如示例所示),该输出还包括解锁脚本,该解锁脚本被配置为基于满足 Tx_1 的锁定脚本中定义的多条备选条件中的一个条件解锁 Tx_1 的所述输出。在目标交易 Tx_p 的第一版本中,解锁脚本被配置为满足第一条条件,即条件1。在某一时刻,还将创建目标交易的第二版本 Tx_p' 。在第二版本中,解锁脚本被配置为满足第二条条件,即条件2。

[0121] 第二目标交易 Tx_p/Tx_p' 具有至少一个输出202_p(例如 Tx_p/Tx_p' 的输出0),在任何一个版本中,该输出指定转移给鲍勃的数字资产金额和将此锁定至鲍勃的锁定脚本(即,要求进一步随后交易在花费的解锁脚本中包含鲍勃的签名)。在这种意义上,目标交易 Tx_p/Tx_p' 的输出可以说是由鲍勃拥有。该输出同样可以称为UTXO。

[0122] 在实施例中,第一条条件要求试图解锁 Tx_1 的交易的解锁脚本(在这种情况下,指目标交易的第一版本 Tx_p)在其解锁脚本中包含鲍勃的加密签名和/或数据有效载荷,该数据有效载荷可能是鲍勃必须提供或包含的鲍勃的数据。包括数据有效载荷的要求可以通过 Tx_1 的锁定脚本中包含的哈希质询施加。该质询包括数据的哈希值(不是数据本身)以及配置为(当与解锁脚本一起在节点104上运行时)测试在对应解锁脚本中提供的数据哈希值是否等于锁定脚本中提供的哈希值的一段脚本。签名的要求可通过例如之前讨论的CheckSig施加。在实施例中,第一条条件不要求爱丽丝的签名包含在 Tx_p 的解锁脚本中。 Tx_p 需要由鲍勃签署的部分可以是 Tx_p 的解锁脚本的设置(例如通过SIGHASH标志指定),或者可以固定。无论采用哪种方式,其至少不包含解锁脚本。因此, Tx_p 的解锁脚本是可延展的。

[0123] 在实施例中,第二条条件要求试图解锁 Tx_1 的交易的解锁脚本(在这种情况下,指目

标交易的第二版本 Tx_p')在其解锁脚本中包括鲍勃的加密签名和爱丽丝的加密签名。同样,这可以通过例如CheckSig施加。在实施例中,第一条件不要求数据有效载荷包含在 Tx_p' 的解锁脚本中。 Tx_p' 需要由爱丽丝和鲍勃签署的部分可以是 Tx_p' 的解锁脚本的设置(例如通过SIGHASH标志指定),或者可以固定。

[0124] 第零交易(即源交易) Tx_0 可以由爱丽丝、鲍勃或第三方生成。其通常需要先前方的签名,爱丽丝从该先前方获得 Tx_0 的输入中定义的金额。其可由爱丽丝、鲍勃、先前方或其他第三方发送至网络106。

[0125] 第一交易(即中间交易和条件交易) Tx_1 还可以由爱丽丝、鲍勃或第三方生成。因为在实施例中需要爱丽丝的签名,所以其可以由爱丽丝生成。或者,其可以由鲍勃或第三方生成作为模板,然后发送至爱丽丝进行签名,例如通过侧信道301发送。然后,爱丽丝可自己将已签署交易发送至网络106,或者将其发送至鲍勃或第三方以供他们转发至网络106,或者仅发送她的签名供鲍勃或第三方组装到已签署 Tx_1 并转发至网络106。同样,在发送 Tx_1 至网络106之前的任何链下交换均可通过侧信道301执行。

[0126] 第二交易(即目标交易或支付交易) Tx_p/Tx_p' 的任何版本都可由爱丽丝、鲍勃或第三方生成。由于第一版本需要鲍勃的签名和/或数据,其可由鲍勃生成。或者,其可以由爱丽丝或第三方生成作为模板,然后发送至鲍勃进行签名和添加数据,例如通过侧信道301发送至鲍勃。然后,鲍勃可自己将已签署交易发送至网络106,或者将其发送至爱丽丝或第三方以供他们转发至网络106,或者仅发送他的签名和数据供爱丽丝或第三方组装到已签署 Tx_p 并转发至网络。在实施例中,第二版本要求鲍勃和爱丽丝两者的签名。因此,其可以由爱丽丝或鲍勃生成作为模板,并且发送至其他方作为模板以添加他们的签名,例如同样通过侧信道301。或者,其可以由第三方生成作为模板,然后发送至爱丽丝,其中爱丽丝添加她的签名,并且转发至鲍勃以添加他的签名。然后,鲍勃将已签署交易转发至网络106,或者发送回爱丽丝或第三方以供他们转发至网络106。或者, Tx_p' 可以由第三方生成作为模板,然后发送至鲍勃,其中鲍勃添加他的签名,然后转发至爱丽丝以添加她的签名。然后,爱丽丝将已签署交易转发至网络106,或者发送回鲍勃或第三方以供他们转发至网络106。在进一步的变体中,爱丽丝和/或鲍勃签署接收的交易模板,并且仅将其签名返回至其他方中的一方,以供该方组装到 Tx_p' 并转发至网络106。同样,在发送 Tx_p 和/或 Tx_p' 至网络106之前的任何链下交换均可通过侧信道301执行。

[0127] 应当理解的是,存在可以生成和组装交易的不同元素的多个位置,以及用于随后将其直接或间接地发送至P2P网络106的最终目的地的各种方法。所公开技术的实施方式的范围不限于这些方面中的任何一个。

[0128] 还应当理解的是,本文中诸如“由爱丽丝”、“由鲍勃”和“由第三方”等词组可分别用作“由爱丽丝103a的计算机设备102a”、“由鲍勃103b的计算机设备102b”和“由第三方的计算机设备”的简略语。此外,需再次注意,给定方的设备可包括由该方使用的一个或多个用户设备、或者诸如该方使用的云资源等服务器资源,或这些的任何组合。这不一定限制在单个用户设备上执行的行为。

[0129] 因为目标交易 Tx_p 的解锁脚本是可延展的,在实施例中,目标交易的第二版本 Tx_p' 可通过延展第一版本 Tx_p 生成,即采用 Tx_p 的现有数据结构并且对其进行修改形成第二版本 Tx_p' (在这种情况下,通过延展锁定脚本)。这是脚本级延展性的示例。但是,在等效变体中,

T_{x_p}' 可以通过创建具有相同结构的目标交易的新版本生成,不同的解锁脚本除外。本文可使用“更新”作为一般术语,以描述延展现有结构或创建新的替换版本的可能性。可通过示例的方式涉及本文的多个实施例来提及延展性,但应当理解的是,这可以由从头开始创建目标交易的新版本所代替。无论采用哪种方式,新版本的延展或创建均可鉴于爱丽丝和/或鲍勃的签名,由爱丽丝和/或鲍勃和/或第三方执行。

[0130] 在一些实施例中, T_{x_1} 的锁定脚本可包括第三解锁条件,作为第一和第二条件的替代。第三条件可要求锁定时间已过期,并且爱丽丝的签名包含在目标交易的第三版本 T_{x_p}'' 的解锁脚本中。如果鲍勃未基于第一和第二条件中的任何一个条件提出索赔,则这使爱丽丝能够从 T_{x_1} 的输出中收回其支付款项(例如:因为他根本不参与该过程或者未在规定时间内参与该过程)。锁定时间可定义为绝对时间点、或者拟过去的时间段,例如以秒进行测量。

[0131] 示例性用例1-针对鲍勃的安全性

[0132] 如前所述,在实施例中,在 T_{x_1} 中,第一条条件i)要求将鲍勃的签名以及数据有效载荷包含在 T_{x_p} 的解锁脚本中,但不是爱丽丝的签名;第二条条件ii)要求爱丽丝和鲍勃两者的签名,但不要求数据有效载荷包含在 T_{x_p}' 的解锁脚本中。

[0133] 假设爱丽丝想向鲍勃付费使他提供一些服务,诸如为她做一些DIY,提供一些商品,提供一些咨询服务等等。爱丽丝创建第一(中间)交易 T_{x_1} ,其输出中包括服务支付款项(T_{x_1} 的输出0参考上述示例)。她通过侧信道301将此发送给鲍勃。或者,第一交易可以由鲍勃或第三方创建。在这阶段,其可以由爱丽丝、鲍勃或第三方中的任何一方发送以在网络106中传播用于记录在区块链150中;或者其可在稍后由这些方的任何一方在与第二目标交易 T_{x_p}/T_{x_p}' 相同的时间或者甚至在目标交易之后发送至网络106。

[0134] 爱丽丝还通过侧信道301向鲍勃发送目标交易的第一版本 T_{x_p} 作为模板,以供鲍勃签署和添加数据有效载荷(或者可由爱丽丝包括有效载荷)。或者,目标交易的第一版本可以由鲍勃或第三方创建作为模板,以供鲍勃签署和添加其数据有效载荷(或者第三方可包括有效载荷)。鲍勃保留目标交易的第一版本 T_{x_p} 的副本,至少直至获得第二版本 T_{x_p}' 为止,或者鉴于鲍勃的签名和数据,第一版本 T_{x_p} 可以由第三方代表鲍勃保留。

[0135] 目标交易 T_{x_p} 最初包含数据有效载荷,因此可由鲍勃(或者代表鲍勃)单方面地赎回,无需爱丽丝的签名,但前提是数据有效载荷包含在解锁脚本中。这将使得可能破坏鲍勃的私人或专有数据。因此,鲍勃优选让爱丽丝欣然提供目标交易的第二更新版本 T_{x_p}' ,包括她的签名(或者发送鲍勃或第三方组装进目标交易 T_{x_p}' 中的她的签名)。这可使鲍勃赎回 T_{x_1} 的输出,并且不必在目标交易 T_{x_p}' 中包括数据有效载荷。但是,如果鲍勃提供了服务,但爱丽丝违背协议,或者对服务的满意提供存在争议,鲍勃仍然可基于不太优选的第一条件转向赎回 T_{x_1} 的输出,要求数据有效载荷包含在目标交易 T_{x_p} 中。

[0136] 当鲍勃为爱丽丝完成服务时,如果爱丽丝诚实且对服务满意,则她将提供她的签名。目标交易的第二版本 T_{x_p}' 要求爱丽丝和鲍勃两者的签名。鲍勃可通过侧信道301向爱丽丝发送带或者不带数据有效载荷第一版本 T_{x_p} ,以供爱丽丝通过签署她的签名(如果仍然存在,则删除数据)进行延展。或者,鲍勃可以只通过侧信道301向爱丽丝发送他的签名,以供爱丽丝组装进第二版本 T_{x_p}' 。然后,爱丽丝可发送该版本,以通过P2P网络106传播并且记录在区块链150中(有时称“广播”或“发布”交易至网络106)。或者,爱丽丝可通过侧信道301返回目标交易 T_{x_p}' 的完整第二版本至鲍勃,以供鲍勃广播至网络106,或者通过侧信道将其发

送至第三方以供第三方广播。再如,爱丽丝根据第一或第二版本生成她的签名(需要记住的是,他们之间的唯一区别在于可延展部分),并且通过侧信道301仅发送她的签名至鲍勃。然后,鲍勃将爱丽丝的签名连同他的签名一起组装进目标交易的第二版本 Tx_p' ,并且广播至网络106或者通过侧信道发送至第三方以供第三方广播。或者在另一个替代方案中,爱丽丝和鲍勃都通过侧信道发送他们的签名至第三方,以供第三方组装进目标交易的完整第二版本 Tx_p' 并且广播至网络106。

[0137] 第一交易 Tx_1 和源交易也需要广播至网络106,用于记录在区块链150中。只要他们最终都在某个阶段得到验证,则这可以由任何一方在任意点完成。

[0138] 第一条件不要求爱丽丝的签名的事实意味着鲍勃可以基于第一条件使用目标交易的第一版本 Tx_p 自主赎回 Tx_1 的输出中定义的金额,即使爱丽丝没有通过她的签名提供她的授权。但是,包括数据有效载荷的要求是繁琐的。此外/或者,相关数据可能是鲍勃的私有、敏感或专有数据,使得必须将数据发布至区块链150将代表对鲍勃的另一种形式的惩罚。因此,鲍勃被激励提供良好的服务,以尝试获得爱丽丝的签名并且基于优选的第二条件使用目标交易的第二版本 Tx_p' 赎回 Tx_1 。但是,如果爱丽丝违约,鲍勃仍然可基于不太优选的第一条件使用目标交易的第一版本 Tx_p 赎回 Tx_1 。因此,第一版本 Tx_p 作为鲍勃的一种担保或保证。

[0139] 需注意, Tx_1 的锁定脚本的第一条件中的要求并未要求在 Tx_1 的锁定脚本中包括数据有效载荷,而且也不要求爱丽丝知道数据有效载荷,即使在 Tx_1 由爱丽丝制定的情况下也同样如此。相反,只要求将数据有效载荷的哈希值包含在 Tx_1 的锁定脚本中(连同质询 Tx_p 的解锁脚本的脚本,以提供当在节点104进行哈希处理时将解锁脚本中的哈希值相匹配的数据)。因此,即使爱丽丝或第三方制定 Tx_1 ,鲍勃仅需提供其数据的哈希值,而不是数据本身。只有在他必须发布目标交易的第一版本 Tx_p 至链的情况下,他才必须发布数据。

[0140] 示例性用例2-流式传输

[0141] 参考图7,假设爱丽丝希望针对来自鲍勃的一些数据的流式传输进行付费。数据将从鲍勃“逐块”传送到爱丽丝,即在 D_0 、 D_1 、 D_2 等部分的序列中。这些可以例如是由鲍勃向爱丽丝流式传输的媒体内容项的部分,例如包括视频轨(诸如电影)和/或音频轨(诸如一段音乐)。视频可包括时变图像或图形,例如电影、电视节目、幻灯片或其他此等序列或静态图像、动态矢量图形和/或游戏内容。音频可以包含采样音频和/或合成音频,包括语音、音乐、噪音和/或特效或诸如此类。在另一个示例中,要让爱丽丝即时支付服务可使用下列技术:例如提供诸如气、电、水等公用设施;或租赁车辆、房产或其他实物。在支付服务款项时,每个数据部分 D_0 、 D_1 、 D_2 等包括解锁服务单元所需的不同相应密钥,而不是本身是所需内容的一部分的数据部分。比如,爱丽丝的水电气供应通过连接至她的计算机设备102a的智能仪表来管理。她将每个收到的密钥从其计算机设备102a提供给她仪表,仪表验证相应密钥后解锁另一个公用设施单元。

[0142] 采用这样的方式流式传输数据部分是可取的,即鲍勃的支付款项与迄今为止收到的数据部分的数量成比例。为此,爱丽丝可响应于从鲍勃收到的每个数据部分 D_0 、 D_1 、 D_2 ……,通过侧信道301返回相应已签署交易 Tx_1 、 Tx_2 、 Tx_3 ……至鲍勃。这将意味着如果鲍勃停止发送数据,则爱丽丝可以仅停止发送支付款项;并且如果爱丽丝停止发送支付款项,鲍勃可以仅停止发送数据并且不多发送爱丽丝未支付的数据D。

[0143] 但是,还需要以这样的方式实现,即对于流式传输的每个单独数据部分 D_0 、 D_1 、 D_2 等不需要将单独交易广播至网络106并且记录在区块链150中,因为这会增加网络拥塞和导致区块链150膨胀。

[0144] 为解决这一问题,响应于她从鲍勃分别接收的每个数据部分 D_0 、 D_1 、 D_2 ……,爱丽丝发送回鲍勃的每个交易 T_{x_1} 、 T_{x_2} 、 T_{x_3} ……是指向相同源交易 T_{x_0} 的相同输出(例如相同UTXO)的第一交易的不同实例。因为第一交易的金额每次都会增加,所以鲍勃只需要在某个已定义的序列(例如电影等音频轨或视频轨结束时,或者指定的服务期,例如每小时、每天、每周或每月)结束时要求最后交易的输出即可。稍后将进一步参考图7进行更详细地解释。

[0145] 同时,还优选采用这样的方式流式传输该部分,即首先,鲍勃不能通过不发送数据但仍然收取爱丽丝的款项进行欺骗;其次,爱丽丝不能通过收到数据但不向鲍勃付款进行欺骗。

[0146] 在实施例中,第一交易的每个实例都包含多个输出,该输出的数字资产总金额大于其输入指向的金额。这意味着只有在某人(在实践中是鲍勃)添加他自身的另一个输入来弥补差异(输入级延展性的示例)之后交易才有效。这会阻止爱丽丝在序列中发布早期交易,进而阻止鲍勃发布后期交易。因此,这使得没有初始资金交易的流式传输充当整部电影或诸如此类的存款。稍后将参考图8对此进行更详细地讨论。

[0147] 为实现流式传输方法,爱丽丝和鲍勃彼此间可以建立链下侧信道301。即通过该信道发送的交易将不会(尚未)被发布到P2P网络以用于记录在区块链150中。这将作为支付通道的修改形式,在本文中也称为“小额支付通道”。而且,鲍勃向爱丽丝提供序列中数据部分 D_0 、 D_1 、 D_2 ……的哈希集。例如,鲍勃可以通过支付通道301向爱丽丝发送哈希集,或者可以将其公开,以便在互联网101或诸如此类上通过服务器访问该集合。哈希集包含一组哈希值,使爱丽丝能够针对数据创建哈希质询,而不必提前知道数据本身。例如,哈希集可包括哈希树(也称为“默克尔树”) (注意:从最广泛的意义上讲,本文使用的默克尔树是指任意哈希树,不一定限于例如二进制分支)。或者,哈希集合还包含哈希链或哈希表。

[0148] 鲍勃刚开始通过支付通道301向爱丽丝发送第一数据部分 D_0 。该第一部分被免费发送或在双方互信的基础上发送。如果爱丽丝不付款,那么鲍勃的损失不超过第一数据部分的价值。假设爱丽丝想继续,则响应于接收 D_0 ,她通过支付通道301向鲍勃发送第一交易 T_{x_1} 的第一实例。作为回应,鲍勃向爱丽丝发送序列中的下一数据部分 D_1 ,然后作为回应,爱丽丝向鲍勃发送第一交易的第二实例 T_{x_2} ,之后鲍勃向爱丽丝发送 D_2 ,爱丽丝向鲍勃发送 T_{x_2} 等等,整个过程均通过支付通道301进行。第一交易的每一实例 T_{x_1} 、 T_{x_2} 、 T_{x_3} ……指定向鲍勃支付的递增款项,例如随着迄今为止收到的数据部分D的数量而呈线性增加。但是,第一交易的每个实例 T_{x_1} 、 T_{x_2} 、 T_{x_3} ……指向爱丽丝的相同UTXO。因此,鲍勃只能构建第二更新交易 T_{x_p}/T_{x_p}' 的有效实例,向他们中的其中一人要求付款(任何试图赎回相同UTXO两次的行为都将因为无效而被网络106拒绝)。假设一切正常,因此鲍勃将创建目标交易版本,要求从序列中第一交易的最后实例支付款项。

[0149] 在实施例中,如前所述(例如,参考图5),第一交易的每个实例 T_{x_1} 、 T_{x_2} 、 T_{x_3} ……或至少是最终实例 T_{x_n} 定义该交易的输出中对赎回爱丽丝支付款项的多个备选条件。在这种情况下,爱丽丝除了要确认序列中最后数据部分 D_n 以外,还需要提供目标交易的第二版本 T_{x_p}' ,或者至少提供她的签名使鲍勃能够组装 T_{x_p}' 。这使鲍勃能够根据优选的第二条件要求

对全序列(例如整部电影)付款,而不是处罚鲍勃的第一条件。如果鲍勃停止对部分D,进行流式传输并且爱丽丝不满意,那么她可能不会根据需要提供满足第二条件的签名,因此鲍勃只能根据第一条件(即不太优选的条件)要求付款。在另一方面,如果爱丽丝中途停止请求进一步的部分但是没有不满意(例如,她只是选择停止观看电影),并且假设迄今为止第一交易的每个实例 TX_1 、 TX_2 、 TX_3 ……都包含多个备选条件,则爱丽丝可能会根据优选的第二条件提供 TX_p 或她的签名,以使鲍勃能够要求对该序列支付款项。

[0150] 第一交易 TX_i 的实例 $i=1,2,3$ ……使用通用UTX0,但是在不同的消息上使用签名。因此,在该上下文中的实例指爱丽丝的相应数据请求,也就是交易的实例(稍后讨论)。这是因为修改数值和请求数据会改变已签名消息。

[0151] 第二交易 TX_p/TX_p' 的版本使用通用UTX0,但是在相同的消息上使用签名。因此,在该上下文中,版本指交易的未延展和延展形式,也就是交易的相应版本。这是因为脚本级延展不会改变已签名消息。

[0152] 注意:之前就哪一方生成和/或广播第一交易 TX_1 ……以及目标交易的第一和第二版本 TX_p/TX_p' 而讨论的任何变体在这里也同样适用。例如,第三方可以代表爱丽丝或鲍勃生成和/或广播部分或全部交易;或者鲍勃也可以自己发送 TX_p/TX_p' 至网络或者发送至爱丽丝去广播,或者发送他的签名给爱丽丝让爱丽丝组装目标交易 TX_p/TX_p' 等。为简洁起见,这些不同的选项在此不再一一重复赘述。

[0153] 这里举一个电影行业的示例。编写脚本时,脚本大小限制在10千字节。因此,对于每部电影,可以分割成许多个8千字节的部分。如果存在其他限制条件,一个部分的大小可能更小;或者如果脚本大小限制增加,一个部分的大小可能更大。一旦部分被定义,就可以创建默克尔树,并且根哈希会连同电影标题一起被公开显示。

[0154] 如果显式输入不能包含显式输出和隐含的交易费用,则假设存在另一个隐式输入。

[0155] 爱丽丝将向鲍勃购买一部电影。该电影是通过 $n+1$ 小数据包 $D_0, \dots, D_n$ 及其带根哈希 H_{root} 的默克尔树 T 定义的。这种方法将构建从爱丽丝至鲍勃的一系列交易 $TX_1, TX_2, \dots, TX_n$ 。每个交易 TX_i 对应于 D_i 的请求和接收 D_{i-1} 的确认。理想状态下,当支付通道301被正确关闭时,只有两个交易(TX'_n 和 TX_p)将发布以完成爱丽丝向鲍勃的付款。这种场景在图7中进行了描述,该图为用例2中爱丽丝与鲍勃之间支付通道301的序列图。需注意,鲍勃先给爱丽丝发送一条消息,接着是每个数据包的 n 个消息对和关闭通道的最后两条消息。

[0156] 第一轮——爱丽丝:最初,鲍勃向爱丽丝发送 D_0 和包含所需数据包的完整默克尔树。爱丽丝检查根哈希是否确实属于她选择的电影标题,并且验证 D_0 的默克尔路径。一旦爱丽丝对收到的数据感到满意,则她会构建 TX_1 确认她已经收到 D_0 ,并且想要请求 D_1 。该交易可采取以下形式。

[0157] TX_1

[0158] 锁定时间:0

[0159] 输入0:

[0160] • 爱丽丝的未花费输出点($TxID_0, vout=0$)

[0161] • 爱丽丝的签名和SIGHASH_ALL|ANYONECANPAY

[0162] 输出0:

[0163] • 锁定条件:

[0164] (i) 如果鲍勃提供 D_1 和他的签名,则他可要求输出。

[0165] (ii) 否则,如果爱丽丝和鲍勃提供了他们的签名,鲍勃可要求输出。

[0166] (iii) 否则,在720个区块之后,爱丽丝可要求输出。

[0167] • 数值:500个单位的数字资产

[0168] 输出1:

[0169] • 爱丽丝的零钱

[0170] 输出2:

[0171] • 向鲍勃支付与输出1中相同的金额

[0172] 图9示出了这方面的示例性例示。这种单一的交易设计具有三个预期功能。通过向交易中的某些字段指定额外的隐含意义,可以只用一个单一的交易模板替换数据交易场景下所需要的多个消息。 $\text{Sig}(P_A, T_{x_1})$ 是确认之前的数据包已被接收且符合要求的签名。 $OP_DUP\ OP_SHA256\langle H(D_1) \rangle OP_EQUAL$ 是下一数据包的请求。500个单位是下一数据包的支付款项。

[0173] T_{x_1} 的输入至少包括输入0。这包括指向锁定至爱丽丝的上一个交易 T_{x_0} 的UTX0的指针,其(例如2000个单位)金额大于 T_{x_1} 的输出0(见下文)。 T_{x_1} 的输出0还包括输入的解锁脚本中爱丽丝的签名,以及能够使其他方添加输入(“ANYONECANPAY”)的标志。

[0174] T_{x_1} 的输出至少包括输出0。这指定了(最初的)少量数字资产(例如,500个单位),小于 T_{x_1} 的输入0。 T_{x_1} 的输出0还包括锁定脚本,使其能够在以下任何条件下解锁:

[0175] i) 后续交易 T_{x_p} 的输入中的解锁脚本包含 D_1 和鲍勃的签名;

[0176] ii) 后续交易 T_{x_p} 的输入中的解锁脚本包含爱丽丝的签名和鲍勃的签名;或

[0177] iii) 超时限制已过,并且后续交易的输入中的解锁脚本包含爱丽丝的签名。

[0178] 关于条件i),因为鲍勃已经向爱丽丝发送过默克尔树(也称为哈希树),所以爱丽丝知道预期的数据部分。从而,她可以确定 D_1 的哈希值,这足以让她通过哈希质询包括这个条件(T_{x_1} 的锁定脚本包含 D_1 的哈希值以及用于检查 T_{x_p} 输入的解锁脚本中出现的值被进行哈希处理时是否会匹配锁定脚本中的值的部分代码)。这个条件意味着如果鲍勃在未获得爱丽丝的签名时想要求付款,则他必须上传 D_1 到区块链150,但因为 D_1 是鲍勃的专有数据,所以他不愿意这样做。这种技术也可以同样用于后续的数据部分 D_2 、 D_3 等。

[0179] 条件ii)将使鲍勃能够要求付款,而不必上传数据,前提是他获得爱丽丝的签名。但是,假设他还不想这样做。

[0180] 条件iii)是可选的。如果鲍勃在某个指定的超时期限结束之后没有以任何理由要求付款(例如,鲍勃从未参与该过程),爱丽丝则可以要求收回 T_{x_1} 的输出0中的金额。应当理解的是,720个区块的特定超时值只是一个示例。更通俗地说,可以根据区块的数量或者人工时间(诸如秒)定义超时期限,也可以将超时期限设为任意值。它可以定义为在绝对时间点或经过的一段时间内到期。

[0181] T_{x_1} 还可以可选地包括一个或多个进一步输出。在实施例中,这些包括输出1和输出2。输出1包括定义与输入金额相等的数字资产金额脚本,并且将此锁定至爱丽丝(“爱丽丝的零钱”)。例如,2000-500个单位=1500个单位。

[0182] 输出2包括定义与输出1相等的数字资产金额脚本,并且将此锁定至鲍勃(“向鲍

勃支付与输出1中爱丽丝的零钱相同的金额”)。这样做的结果是,总输出(本示例中,500+1500+1500个单位=3500个单位)始终大于输入,除非其他人(在实践中只能是鲍勃)为了弥补差异而向Tx₁中添加他自身的另一个输入1。

[0183] 输出2是为了阻止爱丽丝将未经鲍勃确认的交易进行发布而设计的一种技巧。作为付款人,爱丽丝起初并未被激励去广播TX₁。但是几轮之后,当在其他交易中爱丽丝向鲍勃支付更多费用时,在鲍勃使用稍后的实例来要求付款之前,爱丽丝可以使用TX₁通过将其广播到网络106来使这些交易无效。通过包含输出2,鲍勃将他自己的输入(输入1)添加到TX₁后,TX₁才会生效,以弥补输出和输入之间的差额。鲍勃将可以添加额外的输入,因为爱丽丝使用SIGHASH标志“ALL|ANYONECANPAY”。因此,TX₁将可能仅由鲍勃广播。对于Tx₁有效所需的额外输入,爱丽丝不想添加,因为这比不欺骗系统要花费得更多。

[0184] 爱丽丝的零钱定义为爱丽丝的输入值(输入0)减去鲍勃的款项值(输出0)。如图8所示,鲍勃的保险(输出2)将确保总输出(虚线)始终大于发送最后数据部分前的输入。

[0185] 更通俗地说,输出的其他组合起来可以用于创建一种情况,即Tx₁的总输出值大于总输入值,因此需要鲍勃添加他自己的输入,以申请Tx₁的输出0并阻止爱丽丝广播Tx₁。

[0186] 作为在脚本语言的输出0中实现i)、ii)和iii)三个条件的示例,可以使用哈希难题和条件操作码,例如如下所示。

```
[0187]  OP_DUP OP_HASH256 < hash of D1 > OP_EQUAL
        OP_IF
            OP_DROP OP_DUP OP_SHA256 < hash of Bob's public key >
            OP_EQUALVERIFY
            OP_CHECKSIGVERIFY
        OP_ELSE
            2 < hash of Alice's public key > < hash of Bob's public key >
            2OP_CHECKMULTISIG
        OP_IF
[0188]      OP_VERIFY
        OP_ELSE
            < 720 blocks > OP_CHECKSEQUENCEVERIFY
            OP_DUP OP_SHA256 < hash of Alice's public key >
            OP_EQUALVERIFY OP_CHECKSIGVERIFY
        OP_ENDIF
    OP_ENDIF
```

[0189] 第一轮——轮到鲍勃:当鲍勃收到交易TX₁时,他只发送D₁至爱丽丝。需注意,鲍勃这样做很安全,因为他可以在TX₁中通过下述操作要求付款,而无需爱丽丝的任何协助。首先,鲍勃使用输出点(TxID_B, vout=0)通过添加其自身的输入来创建TX'₁,以涵盖TX₁中输出2的值。其次,鲍勃创建另一个交易TX_p,以要求付款:

[0190] TX_p

[0191] 锁定时间:0

[0192] 输入0:

[0193] • 来自 TX'_1 输出0的输出点。

[0194] • 解锁数据

[0195] oD_1

[0196] o 鲍勃的签名

[0197] 输出0:

[0198] • 向鲍勃付款

[0199] • 数值:500个单位

[0200] 然后,鲍勃将两个交易广播到网络106。但是,对于鲍勃而言,这不是理想的情况,因为他必须在交易中透露 D_1 。这被视为通道的提前关闭。但是,如果爱丽丝遵循正确的程序关闭支付通道(如下所述),则鲍勃无需这样做。

[0201] 第二轮——轮到爱丽丝:当爱丽丝收到 D_1 并且对内容满意时,她为下一个数据包构建以下交易。该交易还将被视为接收 D_1 的确认。

[0202] TX_2

[0203] 锁定时间:0

[0204] 输入0:

[0205] • 爱丽丝的未花费输出点(假定是与 TX_1 中相同的输出点)

[0206] • 爱丽丝的签名和SIGHASH_ALL | ANYONECANPAY

[0207] 输出0:

[0208] • 锁定条件:

[0209] o 如果鲍勃提供 D_2 和他的签名,则他可要求输出。

[0210] o 否则,如果爱丽丝和鲍勃提供了他们的签名,鲍勃可要求输出。

[0211] o 否则,在720个区块之后,爱丽丝可要求输出。

[0212] • 数值:1000个单位

[0213] 输出1:

[0214] • 爱丽丝的零钱

[0215] 输出2:

[0216] • 向鲍勃支付与输出1中相同的金额

[0217] 比较 TX_1 和 TX_2 ,需注意, D_1 更改为 D_2 ,并且输出0的值从500个单位的数字资产增加至1000个单位。由于这两点更改,其他输出将具有不同的值(假设爱丽丝使用相同的未花费的输出点)。而且,由于这些更改不是在交易的可延展部分上进行,爱丽丝必须为 TX_2 生成新的签名。

[0218] 第二轮——轮到鲍勃:当鲍勃收到交易 TX_2 时,他只发送 D_2 至爱丽丝。

[0219] 与之前一样,鲍勃这样做是安全的,因为他可以在没有爱丽丝帮助的情况下像第一轮一样要求付款:鲍勃将同样从 $(TxID_B, vout=0)$ 添加其自身的输入,以涵盖 TX_2 中的输出2,并且创建 TX'_2 ;鲍勃还将创建另一个交易 TX_p ,以要求付款:

[0220] TX_p

[0221] 锁定时间:0

[0222] 输入0:

- [0223] • 来自TX'₂输出0的输出点。
- [0224] • 解锁数据
- [0225] oD₂
- [0226] o鲍勃的签名
- [0227] 输出0:
- [0228] • 向鲍勃付款
- [0229] • 数值:1000个单位
- [0230] 然后,鲍勃将两个交易广播到网络。
- [0231] 如果爱丽丝和鲍勃合作关闭通道,鲍勃可以避免第一轮中提到的操作。
- [0232] 最后一轮——轮到爱丽丝:几轮后,爱丽丝构建TX_n请求最终的数据包D_n。
- [0233] TX_n
- [0234] 锁定时间:0
- [0235] 输入0:
- [0236] • 爱丽丝的未花费输出点(假定是与TX₁中相同的输出点)
- [0237] • 爱丽丝的签名和SIGHASH_ALL|ANYONECANPAY
- [0238] 输出0:
- [0239] • 锁定条件:
- [0240] o如果鲍勃提供D_n和他的签名,则他可要求输出。
- [0241] o否则,如果爱丽丝和鲍勃提供了他们的签名,鲍勃可要求输出。
- [0242] o否则,在720个区块之后,爱丽丝可要求输出。
- [0243] • 数值:500n个单位
- [0244] 输出1:
- [0245] • 爱丽丝的零钱
- [0246] 输出2:
- [0247] • 向鲍勃支付与输出1中相同的金额
- [0248] 最后一轮——轮到鲍勃:鲍勃用最终数据包D_n进行响应。
- [0249] 关闭通道:为了关闭支付通道,爱丽丝和鲍勃之间有一些交互。爱丽丝或鲍勃都可以向对方发出信号,表明打算关闭通道301的意愿。在不失一般性的前提下,假设从鲍勃发送给爱丽丝的最后数据包是D_n。鲍勃发现TX_n,即请求D_n的交易,并且添加其自身的输入,以涵盖输出2来创建TX'_n。鲍勃创建TX_p,如下所示:
- [0250] TX_p
- [0251] 锁定时间:0
- [0252] 输入0:
- [0253] • 来自TX'_n输出0的输出点。
- [0254] • 解锁数据
- [0255] oD_n
- [0256] o鲍勃的签名
- [0257] 输出0:
- [0258] • 向鲍勃付款

[0259] • 数值:500n个单位

[0260] 鲍勃通过支付通道301直接向爱丽丝发送两个交易(TX'_n 和 TX_p)。爱丽丝检查 TX'_n 和 TX_p 的输入是否确实如她所期望的那样相关。爱丽丝签署 TX_p ,并且将 D_n 替换为她的签名以创建 TX'_p 。

[0261] TX'_p

[0262] 锁定时间:0

[0263] 输入0:

[0264] • 来自 TX'_n 输出0的输出点。

[0265] • 解锁数据

[0266] o爱丽丝的签名

[0267] o鲍勃的签名

[0268] 输出0:

[0269] • 向鲍勃付款

[0270] • 数值:500n个单位

[0271] 爱丽丝将 TX'_p 发送给鲍勃。鲍勃广播 TX'_n 和 TX'_p 至网络106。或者,爱丽丝将广播 TX'_n 和/或 TX'_p ,或者可发送他们中的一个或两个至第三方,以代表爱丽丝和鲍勃进行广播。需注意,爱丽丝可以随时选择关闭通道。

[0272] 注意图7:(A)鲍勃可以通过广播一对交易而单方面关闭通道;(B)形成交易对的两个交易都在通道关闭时被广播,显示通道在链下有效“打开”而不与网络106通信。

[0273] 为概括此序列,鲍勃发送 D_2 给爱丽丝,作为对 Tx_1 的响应。爱丽丝向鲍勃发送 Tx_2 进行确认,然后鲍勃发送 D_3 等。 Tx_2 与 Tx_1 相同,但 D_1 替换为 D_2 ,输出0的金额增加。在 Tx_3 中, D_2 被替换为 D_3 ,输出0的金额再次增加。在实施例,输出0中的金额随着 i 呈线性增加,即随着每个数据块和在确认中发送的Tx增加。或者,不排除使用了另一种递增关系,例如在序列结束时给予更高的权重,以进一步激励完成序列。

[0274] 鲍勃可以基于标准ii)单方面要求 $Tx_1 \cdots Tx_i$ 中的任何一个。为此,鲍勃会进行延展以创建 Tx'_i 。延展包括添加鲍勃部分数字资产的输入以弥补差额,然后创建在其输入中包含 D_i 的另一个交易 Tx_p ,以花费 Tx_i 。 Tx_p 具有无条件地锁定至鲍勃的输出。鲍勃可以添加他的输入,并且花费早期交易 Tx_1 或 Tx_2 等中的一个,但并不值得。他更愿意继续发送电影并在最后获得全部金额。此外,鲍勃更倾向于不依赖标准ii),因为他必须在区块链上发布电影的其中一个数据块。

[0275] 需注意,每个 Tx_1 、 Tx_2 、 $Tx_3 \cdots$ 的输入指定 Tx_0 的相同UTX0。因此,如果其中任何一个广播至网络并且在任意给定节点104验证,则他们中的任何其他将在该节点不再被视为有效(有效的条件是Tx不试图花费已经被另一个交易有效花费的UTX0)。不同节点104可能首先接收不同的实例,并且因此在挖掘一个实例之前,对于哪个实例‘有效’具有冲突的观点,在这一点上,所有节点104都同意所挖掘的实例是唯一有效的实例。如果节点104接受一个实例为有效,然后发现第二实例已被记录在区块链中150,则该节点104(必须)接受这一点并且将丢弃最初接受的未挖掘实例(即视为无效)。

[0276] 如果鲍勃提前兑现并且停止发送电影,那么爱丽丝只多支付了一个数据块,因此只损失了500。爱丽丝可以在任意点保释,但如果她这样做,鲍勃则不会向爱丽丝发送其未

支付的任何一个多余的电影数据块(例如价值500个单位)。

[0277] 该机制之所以起作用,是因为金额每次都会从小到大增加,直到电影结束;并且所有交易都试图在 Tx_0 中花费相同的UTX0,因此兑现任何一个交易都会使得其他交易无效。此外,在实施例1中,输出总和必须大于输入,直到鲍勃添加他自己的输入。这是输入级延展性的示例。

[0278] 如果鲍勃和爱丽丝都等到电影结束,鲍勃将发送 Tx_n 至爱丽丝供爱丽丝签署,使得 D_n 的哈希值替换为她的签名。这使鲍勃能够在不发布任何数据块D的情况下要求全额付款。这是脚本级延展性的示例。

[0279] 为了避免区块链上的数据拥塞,该过程使用锁定脚本,该锁定脚本可以通过使用某些数据包或数据接收方的签名来解锁。通过使用交易的延展性,数据接收方可以用她或他的签名替换解锁脚本中的数据。这一行为不仅能确认数据已收到或确认支付通道关闭,还可以从交易中修剪数据以节省空间。

[0280] 考虑到交易价值增量的增加,除了从爱丽丝那里收到的最后一个交易之外,鲍勃没有发布任何交易的动机。如果爱丽丝过早离开通道,那么鲍勃只需发布他从爱丽丝那里收到的最后一个交易,以及要求付款的交易。如果他没有收到爱丽丝的延展交易,则必须公开相关的数据包才能要求付款。在实施例1中,如果对数据机密性有很高的要求,鲍勃可以对发送给爱丽丝的数据进行加密,并在交易中透露解密密钥(见下文)。

[0281] 但是对于爱丽丝来说,当她收到足够大的数据包时,可能有发布第一交易的动机。由于第一交易和最新交易均有效,因此不确定她是否会成功地使最新的通信交易无效。为了完全避免这种情况,实施例1中包含一个使交易本身无效的额外输出,除非有人弥补了输出和输入之间的差额。对于爱丽丝而言,为使交易有效,她必须提供额外的输入,但这违背了其广播交易的目的。

[0282] 在鲍勃下线的情况下,爱丽丝将无法继续观看电影。但是,她也不会为观看过的内容支付多余的费用。她可以等待鲍勃回来,或者也可以转到另一个服务提供商并从她停下的地方开始。

[0283] 需注意,这种形式的支付通道不需要资金交易。此外,这非常灵活,爱丽丝可以在任意点重新连接以恢复流式传输服务。也就是说,没有建立支付通道的开销。

[0284] 实施例1解决了支付通道内的所有风险。但爱丽丝仍有可能在支付通道外双重支付UTX0。实施例1可以防止以下三个选项中的任何一个或多个,以防止这种情况发生。第一种是采用技术强制爱丽丝在尝试双重支付UTX0时向存款账户透露密钥,在这种情况下,鲍勃将能够领取所有存款。第二选项是使爱丽丝的确认也具有法律约束力。即爱丽丝在鲍勃的支付请求交易上的签名可以被视为是具有约束力的身份证明。爱丽丝的任何不当行为都将受到法律实施的约束。第三选项是鲍勃不时终止并重新启动支付通道,例如每5分钟。鲍勃可以根据自己对风险的评估来调整频率。需注意,因为不需要资金交易,所以重新启动支付通道不会产生开销。

[0285] 数据加密:当数据通过公共网络交换时,通常对数据机密性有要求。在上一节中,假设数据接收方确切地知道预期接收的内容。但是,当数据被加密时,在没有任何通信的情况下,很难提前知道密文或期望的哈希值。这就需要在锁定脚本中构建哈希难题。为了缓解这种情况,在实施例1中,数据销售方可以在传输之前将密文的哈希值传送给数据接收方。数

据接收方使用给定的哈希值构建支付交易。当接收到加密数据时,数据接收方可以对数据进行解密,并且验证数据是否符合预期。如果符合,则一切顺利。如果不符合,那么最坏的情况是接收方赔钱。但是,接收方可能损失的金额受每个数据包的价格约束。如果是电影,可能大约500个单位,例如加起来大约5美元一部电影。考虑到经济价值小和声誉影响大,数据销售方没有欺骗的动机。

[0286] 一些实施例可以实现一种机制,在数据销售方和每个数据购买方之间通过对称加密建立共享密钥。因此,传输中的所有数据都可以加密。

[0287] 示例性用例3-轮询和投票

[0288] 在另一个示例性用例中,例如图5所示,第一交易 Tx_1 的不同备选条件可用于表示不同的投票选择,以使用区块链150作为记录投票的方式。在 Tx_p 被提交至网络以包含进区块151之前,爱丽丝在 Tx_p 中的选择可作为其投票意向的指示性轮询。在下述示例中,第一交易将称为 Tx_{S1ip} ,第二(目标)交易的第一版本将称为 Tx_{Poll} ,目标交易的第二版本将称为 Tx_{Vote} 。

[0289] 设想以下示例性场景。即将举行一场选举。政府机构(“鲍勃”103b)希望通过使用基于区块链的投票系统为参与的合格投票者提供报酬,以激励民众参与轮询和投票。为了举例说明,假设这是一个两党制国家:“工党”和“保守党”。政府机构 P_{Gov} 负责(a)合格投票者的登记,(b)签发投票单,以及(c)统计轮询和选票。在这个场景中,政府机构是第二方103b(也称为“鲍勃”)。选民包括N合格投票者。每个投票者(“爱丽丝”103a)在登记时选择私人的“投票令牌” T_v ,该令牌仅投票人自己知道。鲍勃的政府机构知道每个登记的投票者的哈希值 $H(T_v)$ 。

[0290] 以下定义可用于轮询和投票场景。

[0291] • 登记:=投票者通过某些区块外或区块内进程登记为合格投票者。投票者私下选择 T_v 并提供给鲍勃的政府机构 $H(T_v)$ 。

[0292] • 投票单(Ts_{S1ip}):=交易,包括锁定脚本,用于支付参与轮询和/或投票的投票者。

[0293] • 轮询(Tx_{Poll}):=使用解锁条件(I)或(II)来解锁投票单输出的已签署交易,其中条件(I)和(II)代表两个相应选项的轮询选择,例如公投回应或政党(参见下面的示例)。

[0294] • 投票(Tx_{Vote}):=使用解锁条件(III)或(IV)解锁投票单输出的已签署交易,其中条件(III)和(IV)表示两个相应选项的投票选择(同样参见下面的示例性条件)。

[0295] 每个投票者也将拥有自己的公钥 P_v 。为了参与轮询或投票给他们选择的政党,每个人都必须使用代表他们选择意愿的公钥进行签名。

[0296] 在说明性的两党制示例中,只能选择其中一个政党,投票者用来表达其选择意愿的两个可能的公钥可以如下所示:

[0297] $P_{v,L} = P_v + \text{SHA256}(\text{"Labour"}) \cdot G,$

[0298] $P_{v,C} = P_v + \text{SHA256}(\text{"Conservative"}) \cdot G,$

[0299] 其中“ $\cdot$ ”定义为“EC点乘以标量”的运算符。

[0300] 关于解锁条件,锁定脚本将被构建为:当满足四个不同解锁条件中的任何一个时,即可解锁。条件(I)和(III)均与“工党”有关,分别代表轮询和投票。条件(II)和(IV)均与“保守党”有关,分别代表轮询和投票。条件全文如下:

[0301] (I) $\langle \text{SigP}_{V,L} \rangle \langle P_{V,L} \rangle \langle \text{"工党"} \rangle$

[0302] (II) $\langle \text{SigP}_{V,C} \rangle \langle P_{V,C} \rangle \langle \text{"保守党"} \rangle$

[0303] (III) $\langle \text{SigP}_{\text{Gov}} \rangle \langle P_{\text{Gov}} \rangle \langle T_V \rangle \langle \text{SigP}_{V,L} \rangle \langle P_{V,L} \rangle \langle \text{"工党"} \rangle$

[0304] (IV) $\langle \text{SigP}_{\text{Gov}} \rangle \langle P_{\text{Gov}} \rangle \langle T_V \rangle \langle \text{SigP}_{V,C} \rangle \langle P_{V,C} \rangle \langle \text{"保守党"} \rangle$

[0305] 实施例将通过脚本级延展性切换条件的概念进行切换：

[0306] • (I) $\rightarrow$ (III) 将工党轮询转换为工党投票，或者

[0307] • (II) $\rightarrow$ (IV) 将保守党轮询转换为保守党投票。

[0308] 这些切换表示将轮询转换为相同方的 (“ $\rightarrow$ ”) 投票。这些转换可以通过延展性来实现，因为使用的公钥和签名在整个交换中是一致的——只有输入脚本数据的其余部分被延展，因此不需要更改投票者的签名。

[0309] 但是，如果合格的投票者希望进行下述切换：

[0310] • (I) $\rightarrow$ (II) [将在轮询中投给工党的投票转给保守党]；或者

[0311] • (III) $\rightarrow$ (IV) [将给工党的投票转给保守党]。

[0312] 在实施例中，这不能使用延展性和切换条件来实现。但是，或者这也可以通过分别增加轮询或投票交易的序列号来实现，其中由爱丽丝（投票者）签名的消息包含序列号。这意味着爱丽丝轮询选择的签名不能被第三方恶意地用来伪造她随后的投票选择。

[0313] 解锁条件用于解锁特定的锁定脚本，由此每个解锁条件满足锁定脚本中的不同锁定条件。相关锁定脚本（本文可称为“投票脚本”）如下所示。

[0314] `OP_DUP OP_HASH256 <H("Labour")> OP_EQUAL`

```

        OP_IF
            OP_DROP OP_DUP OP_HASH160 <H(PV,L)> OP_EQUALVERIFY
OP_CHECKSIGVERIFY
        OP_HASH256 <H(TV)> OP_EQUAL
        OP_IF
            OP_DUP OP_HASH160 < H(PGov) > OP_EQUALVERIFY
OP_CHECKSIGVERIFY
        OP_ELSE
            OP_1
        OP_ENDIF
        OP_ELSE
        OP_HASH256 <H("Conservative")> OP_EQUAL
        OP_IF
[0315]     OP_DROP OP_DUP OP_HASH160 <H(PV,C)> OP_EQUALVERIFY
OP_CHECKSIGVERIFY
        OP_HASH256 <H(TV)> OP_EQUAL
        OP_IF
            OP_DUP OP_HASH160 < H(PGov) > OP_EQUALVERIFY
OP_CHECKSIGVERIFY
        OP_ELSE
            OP_1
        OP_ENDIF
        OP_ELSE
        OP_0
        OP_ENDIF
    OP_ENDIF

```

[0316] 轮询和投票程序如图10所示。这是序列图，示出了从在 t_0 宣布选举至在 t_3 投票期结束的完整轮询和投票过程。

[0317] 此程序分为下述步骤。

[0318] 步骤1：在 t_0 时间，鲍勃的政府机构 P_{Gov} 针对在稍后的时间 t_2 进行的投票开放投票者登记。投票者现在可以自由登记并且提供其哈希令牌 $H(T_V)$ 作为登记过程的一部分。

[0319] 步骤2：鲍勃的政府机构 P_{Gov} 发布N投票单交易 $T_{x_{Poll}}$ 。每张投票单均付费，并且投票者的公钥 P_V 适用于单个投票者（带锁定脚本，如前所示）。每个交易均锁定到稍后的时间 t_2 。

[0320] 步骤3：投票者爱丽丝（带公钥 P_V ）希望参与轮询阶段。轮询阶段定义为所有时间 t ： $t_0 < t < t_2$ 。投票者通过选择“工党”和构建轮询交易 $T_{x_{Pool}}$ （图11）和使用解锁脚本条件（I）参与轮询。

[0321] 此交易可通过投票者、政府或第三方公开，但直至 t_2 之前不得被提交至网络以包含在新的区块中。

[0322] 如果投票者希望将轮询选择改为“保守党”，则她必须使用脚本条件（II）生成新的轮询交易，并且增加序列号。

[0323] 步骤4：投票者爱丽丝在某些时间 t_1 决定，她希望将其“工党”的轮询选择转换为投

票选择。这涉及延展轮询交易的解锁脚本,以生成投票交易。该延展将解锁条件更改为脚本条件(III) [(I) $\rightarrow$ (III)]。

[0324] 为执行延展,在步骤4中,投票者爱丽丝将她的投票令牌 T_v 发送至鲍勃的政府机构。在步骤5中,鲍勃的政府机构将轮询令牌切换为投票令牌,以将轮询交易变为投票交易,并且用 $\text{Sig}(P_{\text{Gov}}, T_{x_{\text{Vote}}})$ 签署投票交易,以确认其为有效投票。作为变体,投票者可以将投票令牌放入交易中,并将其发送给政府签字。

[0325] 需注意,由于两个交易施加的锁定时间 t_2 ,轮询交易和随后创建的对应投票交易在 $t < t_2$ 的情况下始终无效。

[0326] 步骤6:鲍勃的政府机构在 t_2 广播所有有效投票交易。其继续广播在 $t: t_2 < t < t_3$ 的时间敲定的任何其他投票交易,其中 t_2 定义投票期的开始时间, t_3 定义投票期的结束时间。

[0327] 在实施例中,政府机构从不广播轮询交易。被轮询者(爱丽丝)有权自行决定是否对比进行广播并要求获得奖励。

[0328] 需注意, $T_{x_{\text{slip}}}$ 还需在流程中的某个点进行提交至网络以包含在新的区块中,使得 $T_{x_{\text{poll}}}/T_{x_{\text{vote}}}$ 中花费的UTXO真实存在。为简单起见,此步骤未在图10中示出。用于锁定交易时间的真实方法也会略微影响图10的信令图中 $T_{x_{\text{slip}}}$ 发送至网络106的点。如果每个 $T_{x_{\text{slip}}}$ 设置'Locktime'(定义8), $T_{x_{\text{slip}}}$ 还将需要广播以在图中的阶段6(和/或最终可选阶段)进行提交至网络以包含在新的区块中。如果一旦选择在 $T_{x_{\text{slip}}}$ 的锁定脚本中设置OP_CLTV or OP_CSV(定义10和11),则 $T_{x_{\text{slip}}}$ 可按照上述步骤2所述同时进行提交至网络以包含在新的区块中。

[0329] 图11示出了可用于实现流程的一些示例性交易。顶部交易是投票单交易 $T_{x_{\text{slip}}}$,其允许全体选民通过轮询和/或投票申请资金。只有在 t_2 时投票期已开始后,才能申请输出。中间交易是轮询交易 $T_{x_{\text{poll}}}$,其表示投票者已选择一个选项作为轮询的一部分。举例来说,假设锁定时间使用OP_CSV or OP_CLTV实现,并且 $T_{x_{\text{slip}}}$ 在发送至投票者的同时发送至网络(步骤2),则该交易在 t_2 前将无效,直至输入中引用的 $T_{x_{\text{slip}}}$ 输出上设置时间锁为止。底部交易为投票交易 $T_{x_{\text{vote}}}$,其表示投票者已选择一个选项作为投票的一部分。该交易与 $T_{x_{\text{poll}}}$ 相同,但已使用脚本级延展性切换了解锁条件。类似地,该交易将在 t_2 之前无效。

[0330] 在实施例中,时间锁包含为第一交易($T_{x_{\text{slip}}}$)的输出的一部分。但是,在替代机制中会存在时间锁 $T_{x_{\text{poll}}}$,而不是达到大致相同效果的 $T_{x_{\text{slip}}}$ 。

[0331] 结论

[0332] 应当理解的是,上述实施例仅通过示例的方式进行描述。

[0333] 更通俗地说,根据本文公开的一方面,提供了一种在区块链中记录第一方和第二方之间的目标交易的方法;该方法包括通过第一方或第二方的计算机设备:获取目标交易的第二更新版本,相对于目标交易预先存在的第一版本进行更新;发送目标交易的更新版本,而非第一版本,以通过节点网络传播,并且记录在至少部分节点的每个节点维护的区块链副本中;其中目标交易包括输入,该输入包含解锁脚本和指向第一交易的第一输出的指针,第一输出包含锁定脚本,该锁定脚本指定用于解锁第一交易的第一输出的多个备选条件;其中目标交易的第一版本的解锁脚本被配置为在满足所述备选条件中的第一条件的基础上,解锁第一交易的第一输出,更新版本的解锁脚本被配置为在满足备选条件中的第二条件的基础上,而非第一条件,解锁第一交易的第一输出。

[0334] 在实施例中,该方法可包括通过第二方的计算机设备:提供使第二方在满足第一条件的基础上发送目标交易的第一版本的功能,以通过网络传播并且记录在区块链中。

[0335] 该方法可使第二方在分别满足所述第一或第二条件的基础上选择性地发送目标交易的第一或更新版本,以通过网络传播并且记录在区块链中;该方法包括第二方使用所述功能执行所述更新版本的发送,以记录在区块链中,而非第一版本。

[0336] 在实施例中,所述功能可供第二方选择手动选择执行所述第一版本的发送,和/或所述功能可被配置为在预定时间到期后或在预定事件发生时自动触发所述目标交易的第一版本的发送。

[0337] 在实施例中,目标交易的更新版本的获取和发送以传播并且记录在区块链中可由第二方的计算机设备执行。

[0338] 在实施例中,第二版本的获取可包括第二方从第一方接收第二版本的至少一部分。

[0339] 在实施例中,该方法可包括通过第二方的计算机设备:获取目标交易的第一版本,其中目标交易的第一版本的解锁脚本可被配置为在满足第一条件的基础上,使第二方自主地解锁第一交易的第一输出。

[0340] 在实施例中,目标交易的第二版本可在第一版本之后获取。

[0341] 在实施例中,第一版本的获取可包括第二方从第一方接收第一版本的至少一部分。

[0342] 在实施例中,第一版本的获取可包括第二方生成第一交易。

[0343] 在实施例中,第二条件可要求解锁脚本包括第一方的加密签名,该加密签名签署目标交易的一部分,不包括解锁脚本。在这种情况下,所述目标交易的更新版本的获取包括获取第一方的签名,并且发送以传播并且记录在区块链中的更新版本在解锁脚本中包括第一方的签名。

[0344] 在实施例中,第一条件可不要求第一方的加密签名。

[0345] 在实施例中,至少第一条件可要求解锁脚本包括第二方的加密签名,该加密签名签署目标交易的一部分,不包括解锁脚本。

[0346] 在实施例中,第二条件还可要求解锁脚本包括第二方的加密签名,该加密签名签署目标交易的一部分,不包括解锁脚本。在这种情况下,发送以传播并且记录在区块链中的更新版本在解锁脚本中包括第二方的加密签名。

[0347] 在实施例中,所述更新版本的获取可包括:第二方将第一版本发送至第一方,以便第一方通过添加第一方的签名延展为更新版本。第一和第二条件可要求第二方采用相同签名签署目标交易的相同部分,从而第二方无需重新签署更新版本。

[0348] 在实施例中,第一条件可要求在解锁脚本中包括数据有效载荷,但是第二条件可不要求在目标交易中包括数据有效载荷。在这种情况下,发送以通过网络传播并且记录在区块链中的更新版本不需要包括数据有效载荷。

[0349] 在实施例中,第一条件要求解锁脚本包括数据有效载荷和第二方的加密签名,该加密签名签署目标交易的一部分,不包括解锁脚本,但不要求在目标交易中包括第一方的加密签名;第二条件可要求解锁脚本包括第一方和第二方的加密签名,但不要求在目标交易中包括数据有效载荷。在此等实施例中,所述目标交易的更新版本的获取包括获取第一方

的签名;发送以传播并且记录在区块链中的目标交易的更新版本不需要包括数据有效载荷,但在解锁脚本中包含第一方和第二方的签名。

[0350] 在实施例中,包括相应数据部分的要求通过锁定脚本中包含的哈希质询创建,该哈希质询包括相应部分数据的哈希值和哈希函数,以检查解锁脚本中相应数据部分的哈希值与锁定脚本中包含的哈希值相匹配。该方法可包括,独立于网络并且在接收第一交易的实例之前,使哈希集(例如哈希树)对第一方可用,该哈希集包括部分数据的每个部分的哈希值,以使第一方可以为相应数据部分生成哈希质询。

[0351] 在实施例中,第一交易的第一输出可指定向第二方支付的款项,该款项通过根据第一或第二条件解锁第一输出进行赎回,其中目标交易可包括将至少一些款项转移至第二方的输出。

[0352] 在实施例中,该方法可包括通过第二方的计算机设备:将数据部分的序列流式传输到第一方,以序列中的最终部分结束;响应数据部分的每一个相应部分,从第一方接收回第一交易的相应实例,每个实例的第一输出针对相应部分指定向第二方支付的款项;其中该款项随着每个数据部分的增加而增加;其中所述目标交易的更新版本的获取包括在序列中的最后部分之后从第一方接收更新版本。

[0353] 在实施例中,该增加可以与迄今为止发送的数据部分的数量成线性比例。

[0354] 在实施例中,每个数据部分可以是媒体内容项(例如音频和/或视频内容)的不同部分。

[0355] 或者,每个数据部分可以是用于访问服务单元(例如提供气、水、电等公用设施;或租赁房产、车辆或其他实物)的不同密钥。

[0356] 在实施例中,所述功能使第二方可以选择在序列中的任意点手动选择执行目标交易的第一版本的发送,以通过网络传播并且记录在区块链中,以赎回迄今为止从第一方接收的序列中第一交易的最新实例。此外/或者,所述功能可被配置为如果第一方停止部分通过序列发送第一交易的实例,则自动执行通过网络传播并且记录在区块链中的第一版本的发送,以赎回第一交易的最新实例。

[0357] 在实施例中,第一交易可包括指定输入金额的一个或多个第一输入,第一交易的第一输出可指定第一款项,第一交易可进一步包括一个或多个指定一个或多个进一步款项的进一步输出,使得款项的总金额大于输入金额,并且第二方从第一方接收的第一交易可不包括弥补差额的其他输入。如果指定付款总额大于总输入金额,网络的节点可被配置为因为无效而拒绝诸如第一交易等交易。在此等实施例中,该方法可包括第二方向第一交易的最终或最新实例添加第二输入以弥补差额,并且发送添加第二输入的第一交易,以通过网络传播并且记录在区块链中。

[0358] 在实施例中,该进一步输出可包括第二输出和第三输出,第二输出指定向第一方支付的等于输入金额减去第一款项的第二款项,第三输出指定向第二方支付的等于第二款项的第三款项。

[0359] 在实施例中,锁定脚本包括所述备选条件中的第三条件,其要求锁定时间已过期以及在解锁脚本中包含第一方的加密签名,从而如果第二方在锁定时间内未赎回,则第一方可赎回第一交易的第一输出中的款项。

[0360] 在实施例中,至少部分备选条件中的每一个条件可对应于不同的投票选择;该方

法可包括:在获取目标交易的更新版本之前,访问或使第三方可以访问第一版本中的解锁脚本,作为投票意图的非约束性指示。

[0361] 在实施例中,锁定时间可包含在第一交易和目标交易的第一版本的至少一个中,以防止网络在预定时间点之前在区块链中记录第一版本。

[0362] 在实施例中,第一交易的输出可指定通过根据锁定脚本中指定条件的任何一个解锁第一输出进行赎回,并且目标交易可包括将至少部分款项转移至第一方的输出。

[0363] 在实施例中,备选条件可包括:第一条件、第二条件、第三条件和第四条件,该第一条件对应于第一选择的轮询指示,该第二条件对应于第二选择的轮询指示,该第三条件对应于第一选择的投票选择,该第四条件对应于第二选择的投票选择;其中目标交易的第一版本的解锁脚本可被配置为在满足第一或第二条件的基础上,解锁第一交易的第一输出;目标交易的第二版本的解锁脚本被配置为通过被配置为在满足第三条件的基础上,而非第一条件或第四条件或第二条件,解锁第一交易的第一输出来将轮询指示转换为投票。

[0364] 在投票情况下,第一条件可要求目标交易的解锁脚本包括第一方的加密签名,该加密签名签署目标交易的一部分(不包括锁定脚本)。第一条件可要求第一方的签名签署私人投票令牌。第二条件可要求目标交易的解锁脚本包括第一方的加密签名和第二方的加密签名,每个签名都签署目标交易的一部分(不包括锁定脚本)。

[0365] 根据本文公开的另一方面,提供了一种包含在计算机可读存储器上的计算机程序,配置为当在第一方和/或第二方的计算机设备上运行时执行本文公开的任何实施例所述的方法。

[0366] 根据另一方面,提供了第一方和/或第二方的计算机设备,包括:存储器,该存储器包括一个或多个存储单元,和处理装置,该处理装置包括一个或多个处理单元;其中该存储器储存被设置在处理装置上运行的代码,该代码被配置为当在处理装置上运行时执行本文公开的任何实施例所述的方法。

[0367] 根据本文公开的另一方面,提供了一种包含在计算机可读存储器上的计算机程序,配置为当在第二方的计算机设备上运行时执行下述操作:获取目标交易的第一版本;获取目标交易的第二版本;使第二方选择性地发送第一和第二版本中的任何一个,以通过节点网络传播,并且记录在至少部分节点维护的区块链副本中;其中目标交易包括输入,该输入包含解锁脚本和指向第一交易的第一输出的指针,第一输出指定第一方的数字资产金额,包括锁定脚本,该锁定脚本指定用于解锁第一交易的第一输出的多个备选条件;其中目标交易的第一版本的解锁脚本被配置为在满足所述备选条件中的第一条件的基础上,解锁第一交易的第一输出,解锁脚本被配置为在满足所述备选条件中的第二条件的基础上,而非第一条件,解锁第一交易的第一输出。

[0368] 根据本文公开的另一方面,提供了一种包含在计算机可读存储器上的计算机程序,配置为当在第二方的计算机设备上运行时执行下述操作:获取目标交易的第一版本;获取目标交易的第二版本;使第二方选择性地发送第一和第二版本中的任何一个,以通过节点网络传播,并且记录在至少部分节点维护的区块链副本中;其中目标交易包括输入,该输入包含解锁脚本和指向第一交易的第一输出的指针,第一输出指定第一方的数字资产金额,包括锁定脚本,该锁定脚本指定用于解锁第一交易的第一输出的多个备选条件;其中目标交易的第一版本的解锁脚本被配置为在满足所述备选条件中的第一条件的基础上,解锁

第一交易的第一输出,解锁脚本被配置为在满足所述备选条件中的第二条件的基础上,而非第一条件,解锁第一交易的第一输出。

[0369] 根据本文公开的另一方面,提供了第二方的计算机设备,包括:存储器,该存储器包括一个或多个存储单元,和处理装置,该处理装置包括一个或多个处理单元;其中该存储器储存被设置在处理装置上运行的代码,该代码被配置为当在处理装置上运行时执行下述操作:获取目标交易的第一版本;获取目标交易的第二版本;使第二方选择性地发送第一和第二版本中的任何一个,以通过节点网络传播,并且记录在至少部分节点维护的区块链副本中;其中目标交易包括输入,该输入包含解锁脚本和指向第一交易的第一输出的指针,第一输出指定第一方的数字资产金额,包括锁定脚本,该锁定脚本指定用于解锁第一交易的第一输出的多个备选条件;其中目标交易的第一版本的解锁脚本被配置为在满足所述备选条件中的第一条件的基础上,解锁第一交易的第一输出,解锁脚本被配置为在满足所述备选条件中的第二条件的基础上,而非第一条件,解锁第一交易的第一输出。

[0370] 根据本公开的另一方面,提供了一种使第二方在区块链中记录目标交易的方法;该方法包括通过第一方的计算机设备:向第二方发送目标交易的更新版本,相对于目标交易预先存在的第一版本进行更新,从而使第二方发送更新版本,以通过节点网络传播,并且记录在至少部分节点维护的区块链副本中;其中目标交易包括输入,该输入包含解锁脚本和指向第一交易的第一输出的指针,第一输出包含锁定脚本,该锁定脚本指定用于解锁第一交易的第一输出的多个备选条件;其中目标交易的第一版本的解锁脚本被配置为在满足所述备选条件中的第一条件的基础上,解锁第一交易的第一输出,并且解锁脚本被配置为在满足备选条件中的第二条件的基础上,而非第一条件,解锁第一交易的第一输出。

[0371] 根据本文公开的另一方面,提供了用于记录在区块链中的一组交易,该组在计算机可读数据介质或媒体上包括:第一交易,该第一交易具有包含锁定脚本的输出,该锁定脚本指定用于解锁第一交易的输出的多个备选条件;第二交易的第一版本具有包含解锁脚本的输入和指向第一交易的输出的指针,其中目标交易的第一版本的解锁脚本被配置为在满足所述备选条件中的第一条件的基础上,解锁第一交易的输出;第二交易的第二版本具有包含解锁脚本的输入和指向第一交易的输出的指针,其中解锁脚本被配置为在满足所述备选条件中的第二条件的基础上,而非所述第一条件,解锁第一交易的输出。

[0372] 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,而仅受随附权利要求限制。

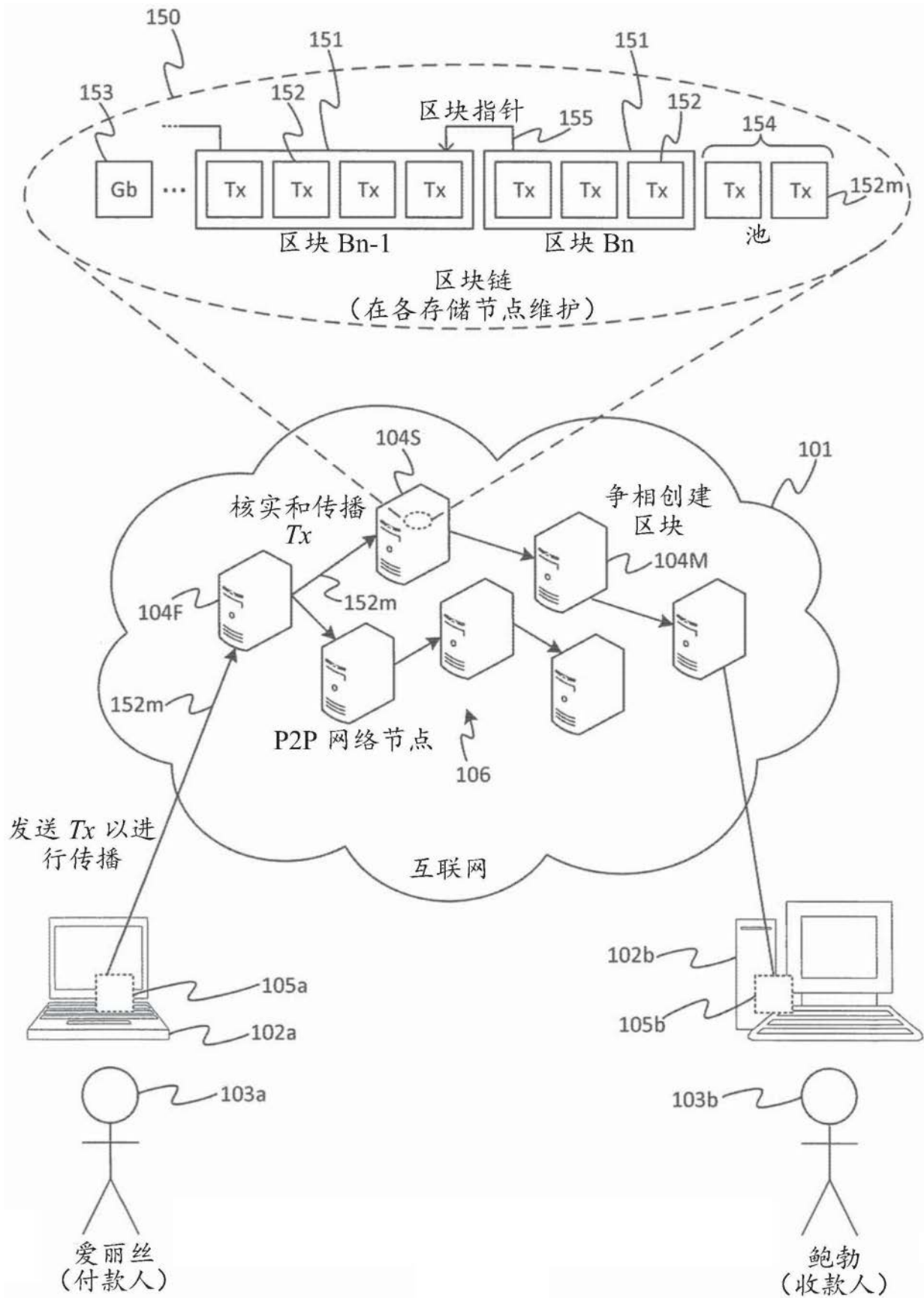


图1

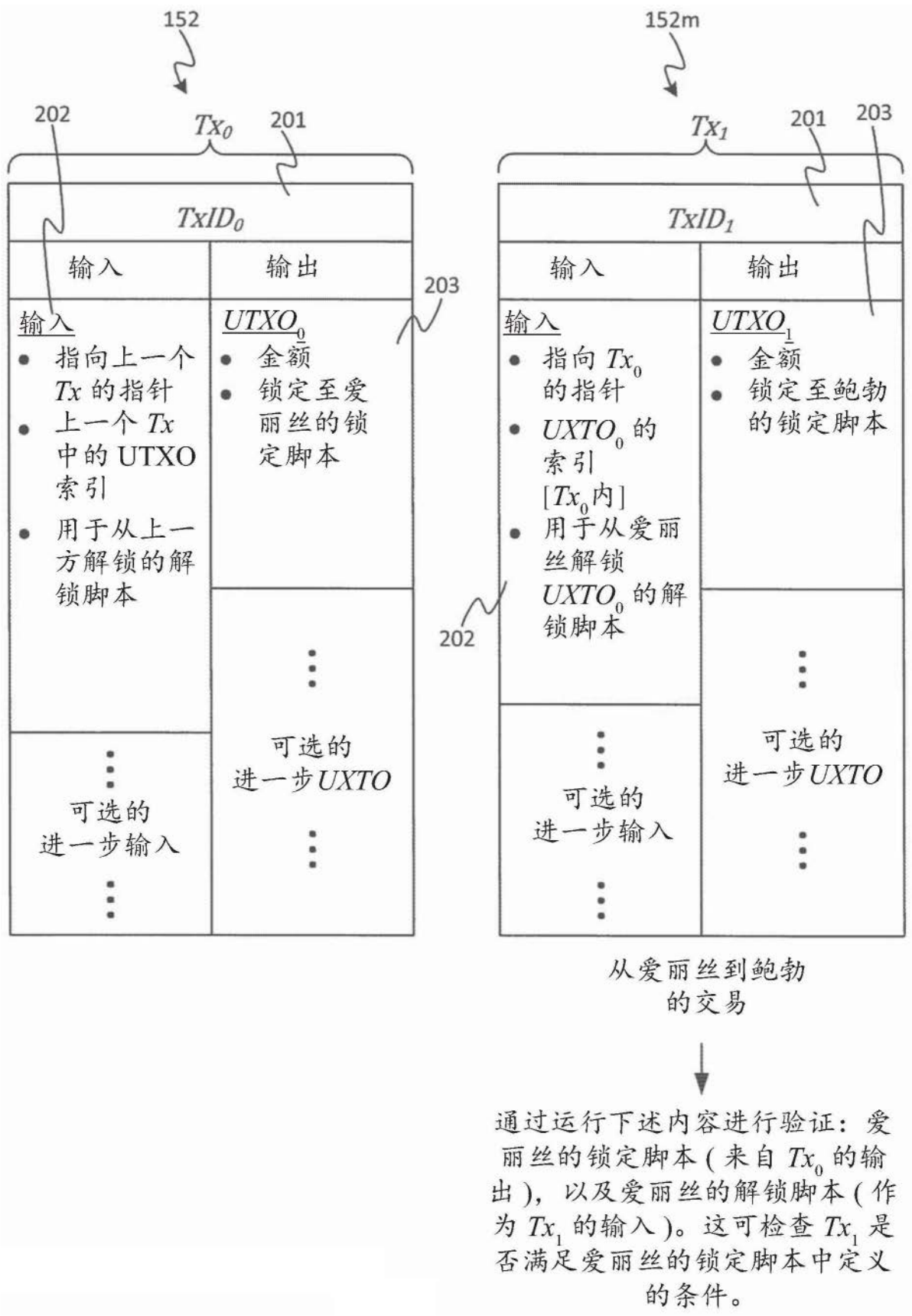


图2

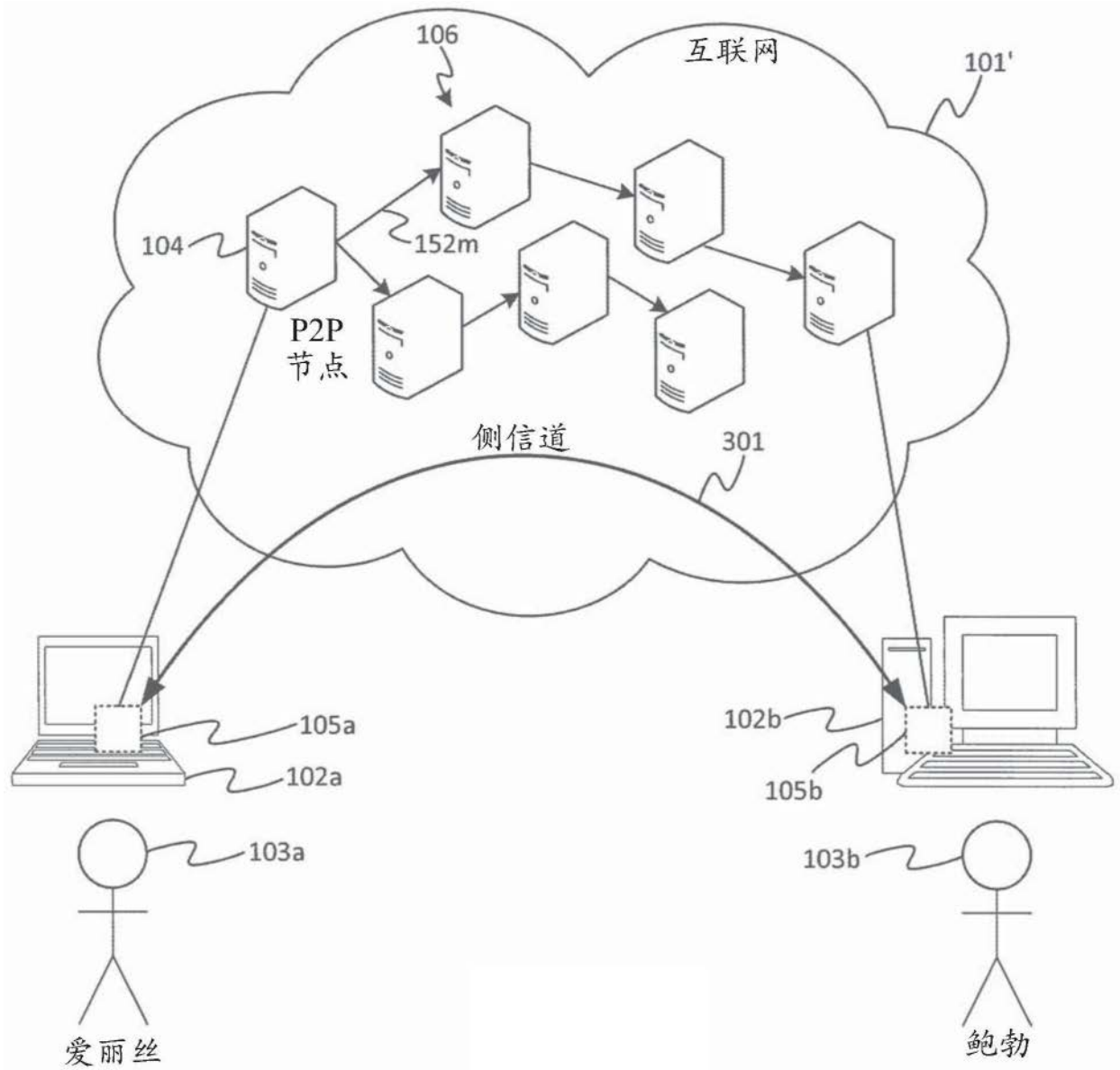


图3

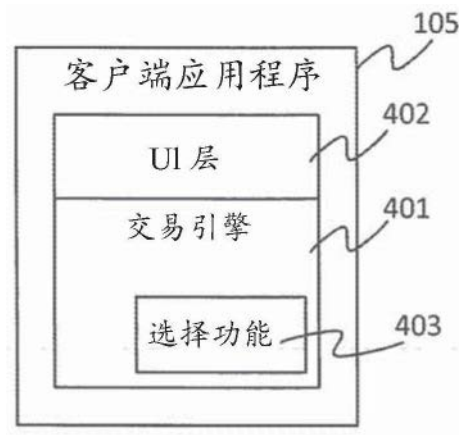


图4

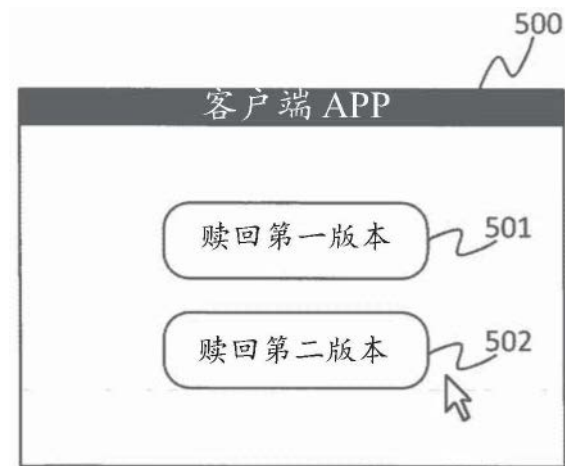


图5

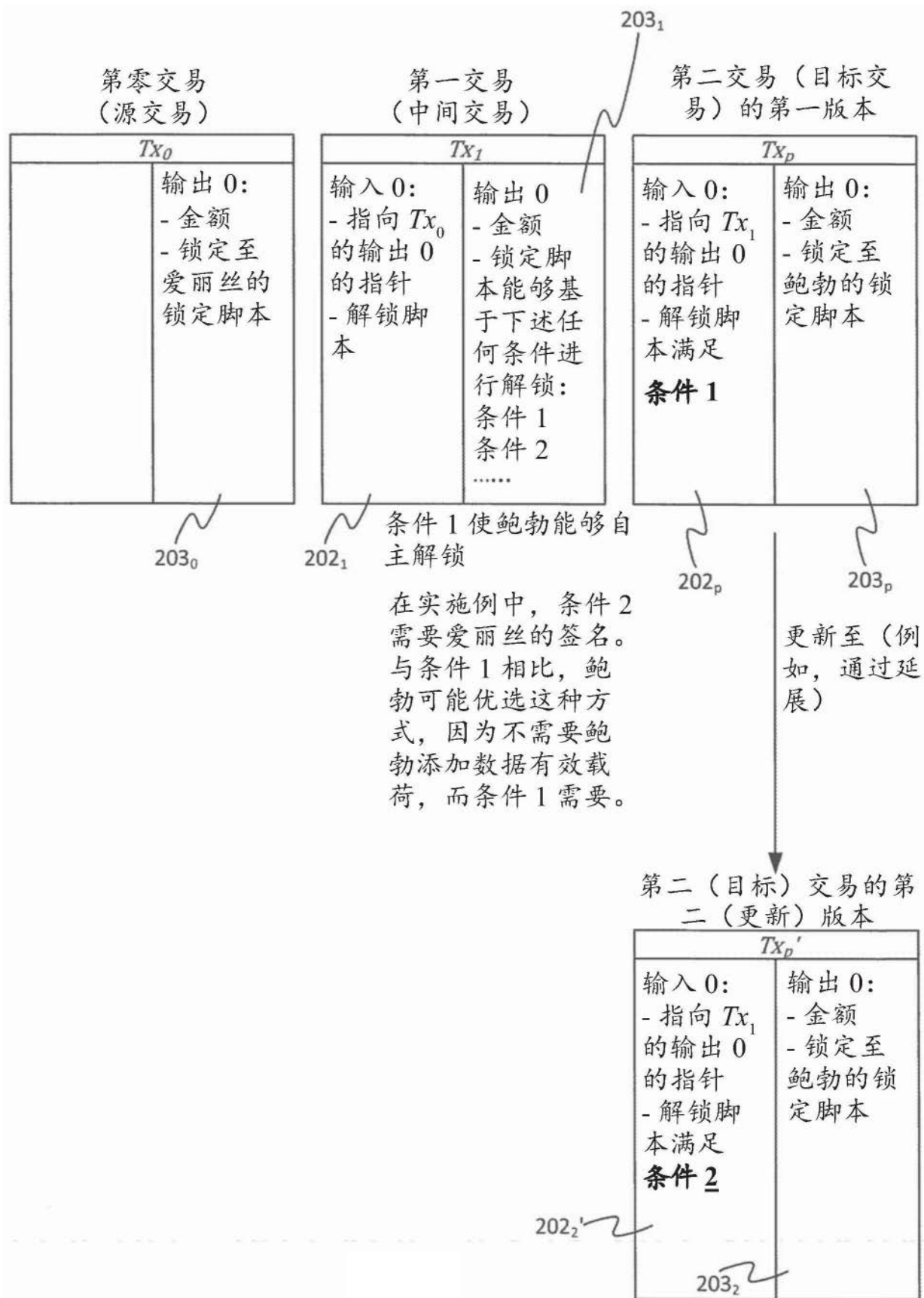


图6

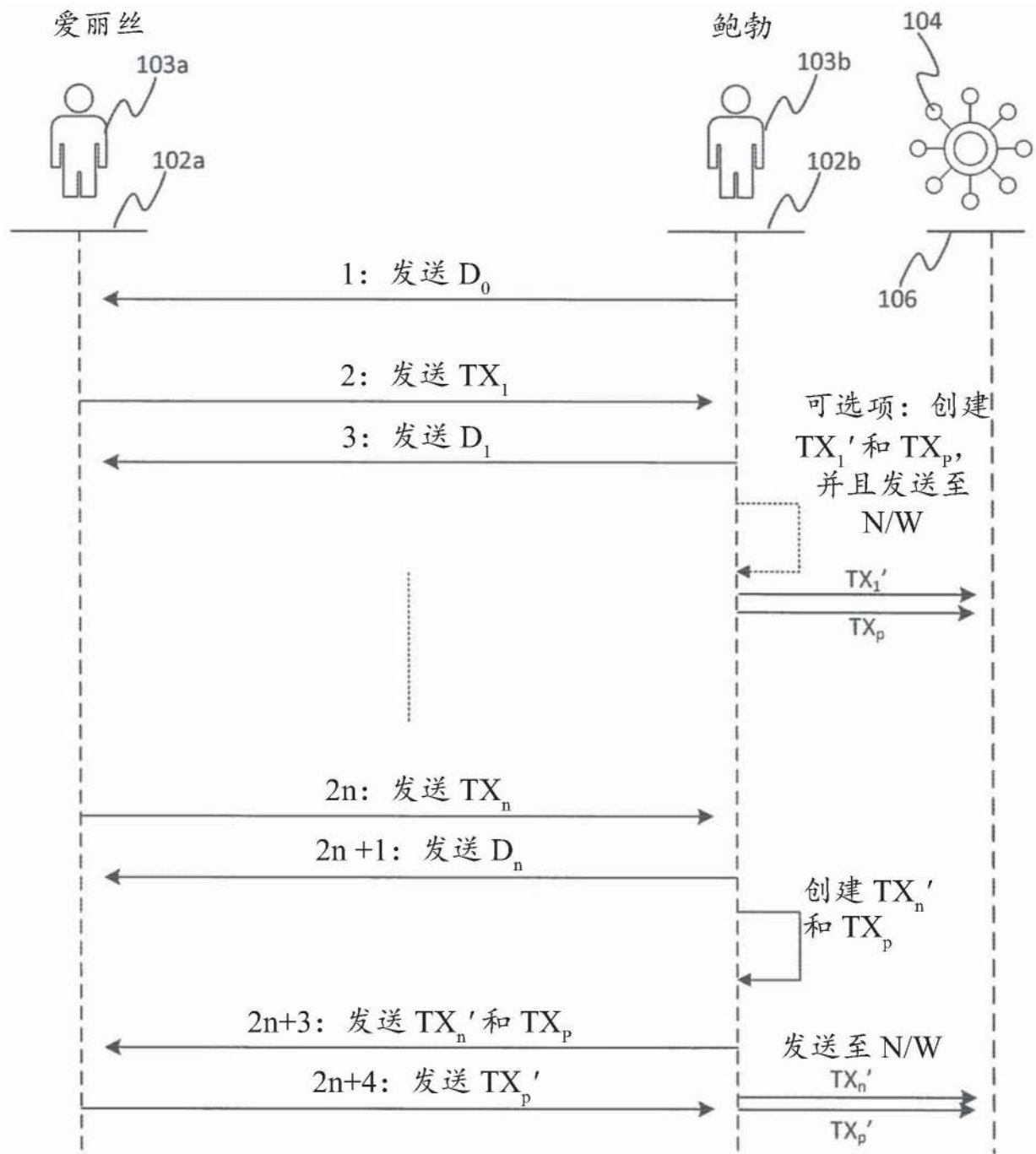


图7

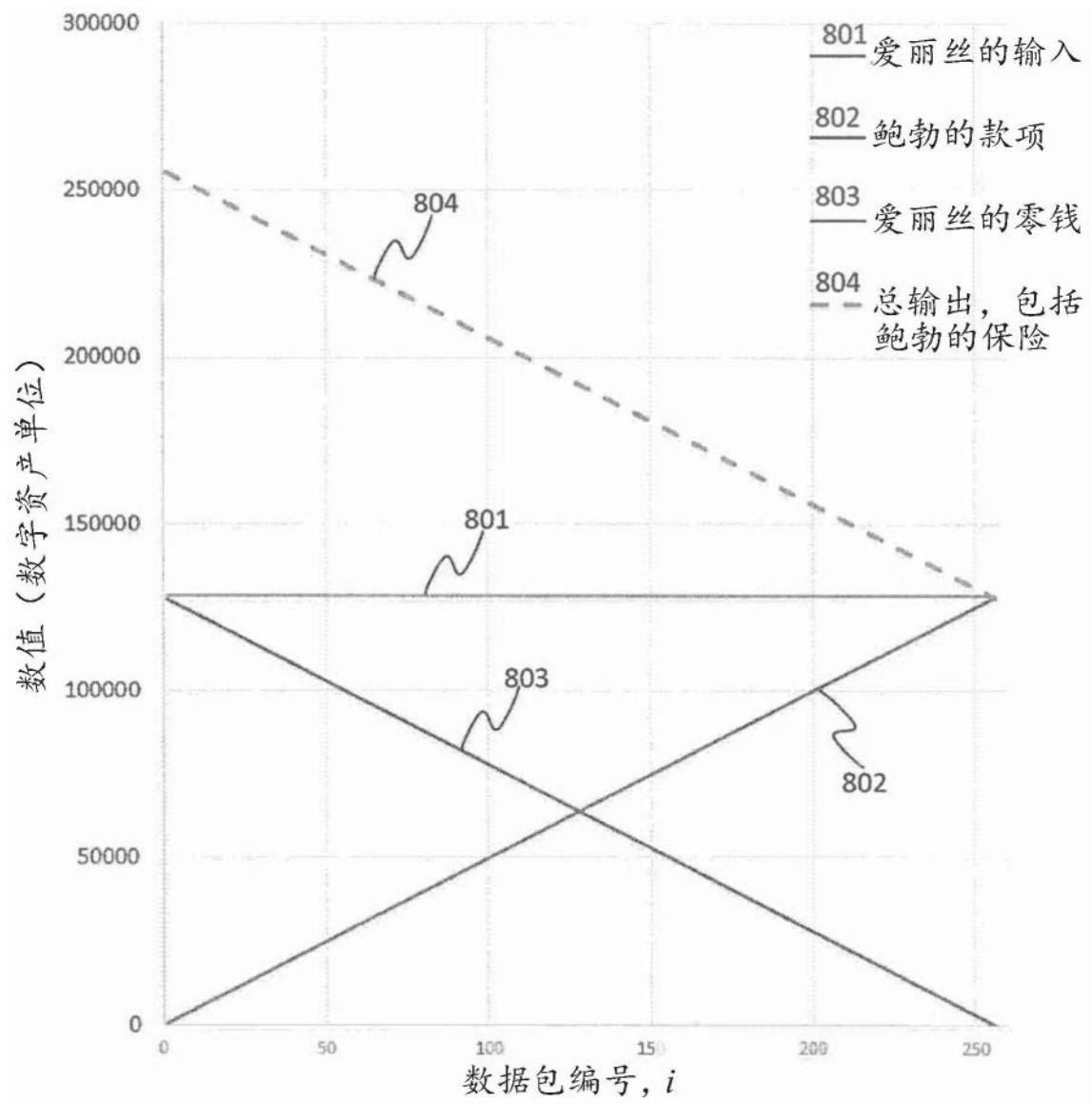


图8

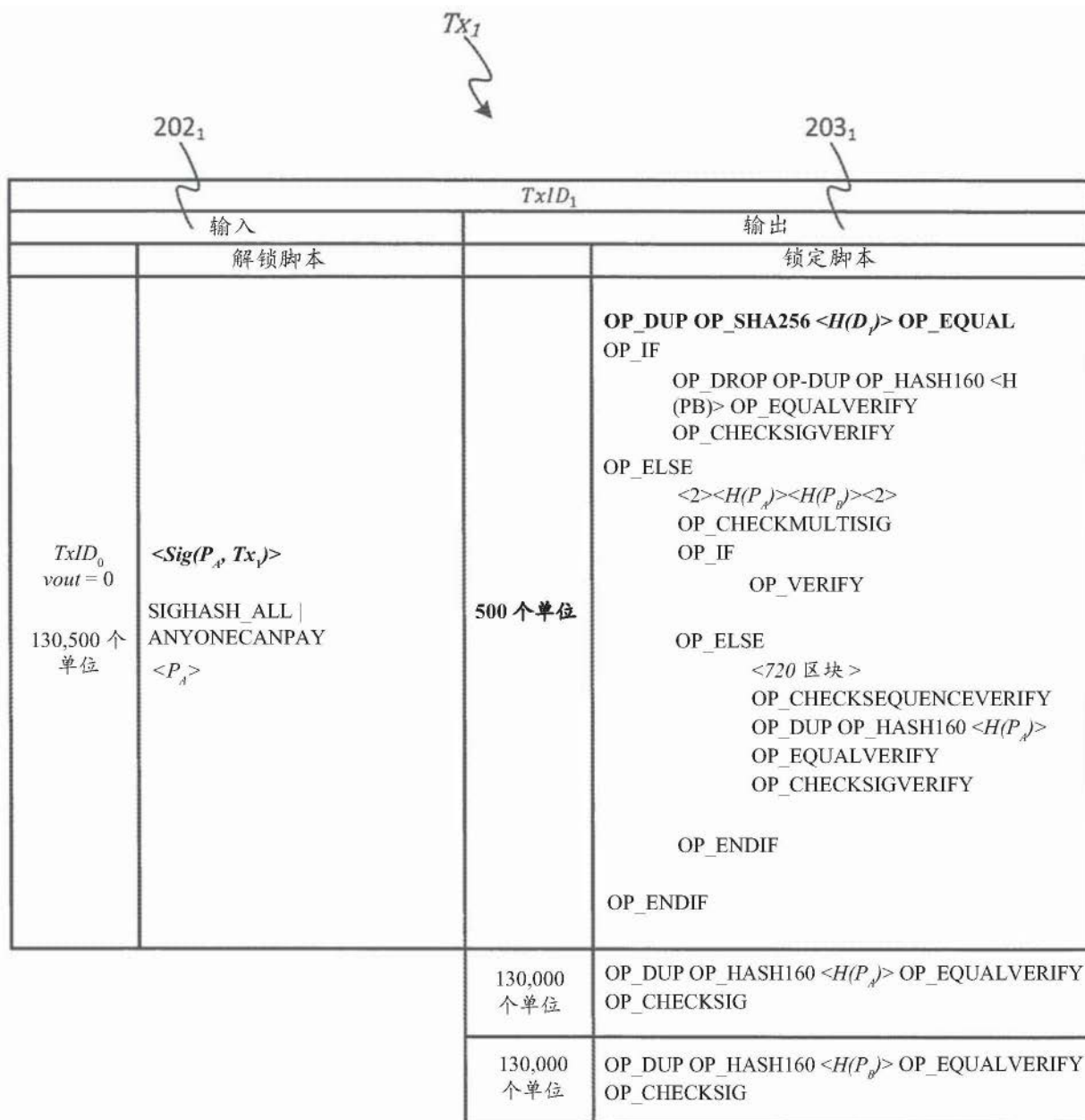


图9

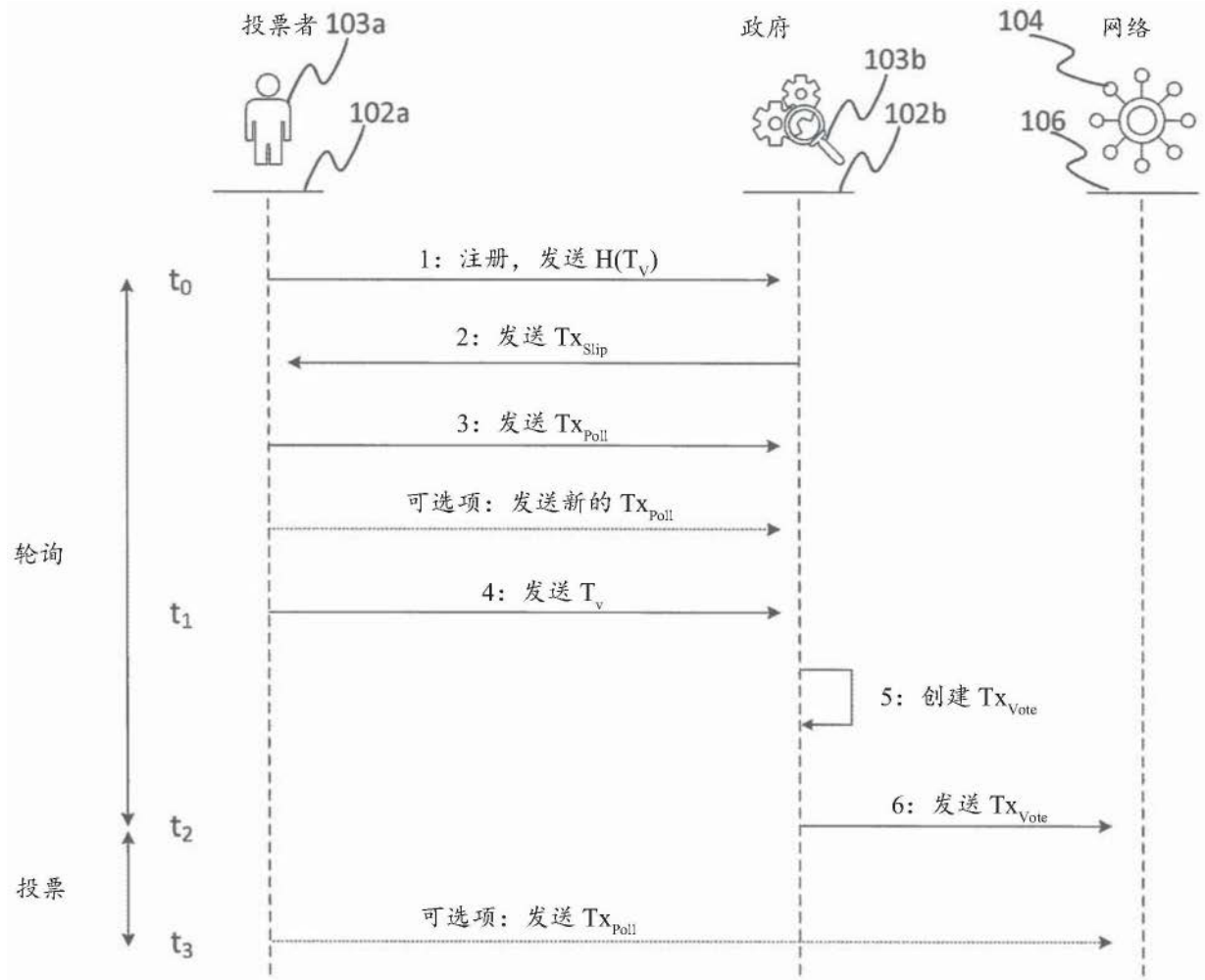


图10

$TxID_{slip}$			
输入		输出	
	解锁脚本		锁定脚本
x BSV	$\langle Sig(P_{Gov}, Tx_{slip}) \rangle \langle P_{Gov} \rangle$	x BSV	$\langle I_2 \rangle$ OP_CHECKSEQUENCEVERIFY [投票脚本]

$TxID_{poll}$			
输入		输出	
	解锁脚本		锁定脚本
$TxID_{slip}$ $vout = 0$	$\langle Sig(P_{V,L}, Tx_{poll}) \rangle \langle P_{V,L} \rangle \langle \text{"Lab"} \rangle$	x BSV	OP_DUP OP_HASH160 $\langle H(P_v) \rangle$ OP_EQUALVERIFY OP_CHECKSIG
x BSV	或 $\langle Sig(P_{V,C}, Tx_{poll}) \rangle \langle P_{V,C} \rangle \langle \text{"Cons"} \rangle$		

$TxID_{vote}$			
输入		输出	
	解锁脚本		锁定脚本
$TxID_{slip}$ $vout = 0$	$\langle Sig(P_{Gov}, Tx_{vote}) \rangle \langle P_{Gov} \rangle \langle T_v \rangle$ $\langle Sig(P_{V,L}, Tx_{poll}) \rangle \langle P_{V,L} \rangle \langle \text{"Lab"} \rangle$	x BSV	OP_DUP OP_HASH160 $\langle H(P_v) \rangle$ OP_EQUALVERIFY OP_CHECKSIG
x BSV	或 $\langle Sig(P_{Gov}, Tx_{vote}) \rangle \langle P_{Gov} \rangle \langle T_v \rangle$ $\langle Sig(P_{V,C}, Tx_{poll}) \rangle \langle P_{V,C} \rangle \langle \text{"Cons"} \rangle$		

图11