(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119051831 A
(43)申请公布日 2024.11.29

(21)申请号 202411058555.1

(22)申请日 2019.05.08

(30)优先权数据
1807813.9 2018.05.14 GB
1807816.2 2018.05.14 GB
1807807.1 2018.05.14 GB
1807811.3 2018.05.14 GB
PCT/IB2018/053347 2018.05.14 IB
PCT/IB2018/053350 2018.05.14 IB
PCT/IB2018/053346 2018.05.14 IB
PCT/IB2018/053349 2018.05.14 IB

(62)分案原申请数据
201980032652.4 2019.05.08

(71)申请人 区块链控股有限公司
地址 安提瓜和巴布达圣约翰

(72)发明人 C·S·赖特

(74)专利代理机构 隆天知识产权代理有限公司 72003
专利代理师 王翡

(51)Int.Cl.
H04L 9/00(2022.01)
H04L 9/08(2006.01)
H04L 9/32(2006.01)

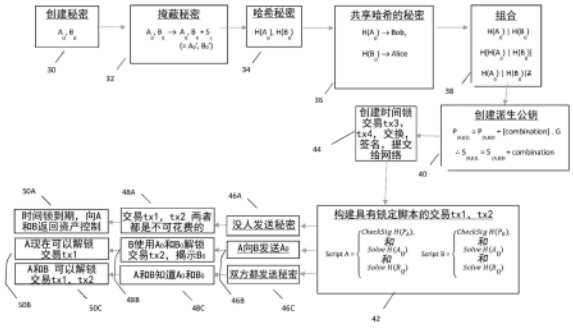
权利要求书2页 说明书17页 附图5页

(54)发明名称

使用区块链执行原子交换的计算机实现的系统和方法

(57)摘要

提供了一种计算机实现的交换方法。该方法可以在一个或多个区块链上进行。该方法包括以下步骤:将第一隐蔽的秘密值(H(A0))从第一用户传达到第二用户,并且将第二隐蔽的秘密值(H(B0))从第二用户传达到第一用户;以及构造第一区块链交易和第二区块链交易(tx1,tx2),其每一个包括第一隐蔽的秘密值和第二隐蔽的秘密值,交易被布置为在向相应的区块链交易提供第一秘密值(A0)和第二秘密值(B0)时是可解锁的,以转移对相应的第一资源或第二资源的控制,其中,第一区块链交易的解锁使第一秘密值被揭示给第二用户,第二区块链交易的解锁使第二秘密值被揭示给第一用户。



CN 119051831 A

1.一种计算机实现的交换方法,所述方法包括以下步骤:

(i)借助于与第一用户相关联的设备,将第一隐蔽秘密值从第一用户传达到第二用户;

(ii)借助于与第一用户相关联的设备,从第二用户接收第二隐蔽秘密值;以及

(iii)构造第一区块链交易,第一区块链交易包括第一隐蔽秘密值和第二隐蔽秘密值,第一区块链交易被设置为在向第一区块链交易提供第一秘密值和第二秘密值时是能解锁的,以转移对第一资源的控制,

其中,第一区块链交易的解锁使第一秘密值被揭示给第二用户,并且第二区块链交易的解锁使第二秘密值被揭示给第一用户,

其中,第二区块链交易包括第一隐蔽秘密值和第二隐蔽秘密值,并且第二区块链交易被设置为在向第二区块链交易提供第一秘密值和第二秘密值时是能解锁的,以转移对第二资源的控制。

2.根据权利要求1所述的方法,其中,第一区块链交易和第二区块链交易中的至少一个被配置为仅在应用相应的第一私钥和第二私钥时是能赎回的。

3.根据权利要求2所述的方法,还包括计算以下中的至少一个的步骤:(a)至少部分地基于第一用户的第一公钥的第一派生公钥;以及(b)至少部分地基于第二用户的第二公钥的第二派生公钥,

其中,第一派生公钥是具有第一私钥的加密密钥对的一部分,并且第二派生公钥是具有第二私钥的加密密钥对的一部分。

4.根据权利要求3所述的方法,其中,计算(a)至少部分地基于第一用户的第一公钥的第一派生公钥以及(b)至少部分地基于第二用户的第二公钥的第二派生公钥中的至少一个的步骤还包括:第一隐蔽秘密值和第二隐蔽秘密值的组合。

5.根据权利要求4所述的方法,其中,第一隐蔽秘密值和第二隐蔽秘密值的组合包括以下中的至少一个:第一隐蔽秘密值和第二隐蔽秘密值的串接、以及至少一个隐蔽秘密值与随机值或伪随机值的串接。

6.根据前述权利要求中的任一项所述的方法,还包括构造以下中的至少一个的步骤:第三区块链交易,第三区块链交易被配置为响应于经过第一区块链交易的非赎回的第一时间段,将对第一资源的控制返回给第一用户;以及第四区块链交易,第四区块链交易被配置为响应于经过第二区块链交易的非赎回的第二时间段,将对第二资源的控制返回给第二用户。

7.根据前述权利要求中的任一项所述的方法,其中,第一隐蔽秘密值和第二隐蔽秘密值中的至少一个包括第一秘密值和第二秘密值中的至少一个与第一用户和第二用户两者都能访问的共享秘密值的组合。

8.根据权利要求7所述的方法,其中,共享秘密值在步骤(i)之前被建立为公共秘密(CS)。

9.根据前述权利要求中的任一项所述的方法,还包括以下步骤:

(iv)从第一秘密值和第二秘密值中的至少一个开始生成至少一个隐蔽秘密值的序列;

(v)使用第一秘密值和第二秘密值中的至少一个来执行任一前述权利要求中的所述方法;

(vi)赎回至少一个区块链交易以揭示第一秘密值和第二秘密值中的至少一个,从而使

得揭示该序列的至少一个隐蔽秘密值。

10.根据权利要求9所述的方法,还包括以下步骤:使用在步骤(vi)中揭示的至少一个隐蔽秘密值来至少执行权利要求1至9中任一项的步骤(iii)。

使用区块链执行原子交换的计算机实现的系统和方法

[0001] 本申请是申请日为2019年5月8日、国家申请号为201980032652.4的PCT国家阶段申请的分案申请。

技术领域

[0002] 本发明大体上涉及计算机实现的安全方法和密码技术。更具体地,本发明涉及一种用于原子交换资源控制的方法。本发明特别适合但不限于在一个或多个区块链和关联的协议上使用。

背景技术

[0003] 在本文献中,我们使用术语“区块链”用来包括所有形式的电子的、基于计算机的、分布式的账本。这些包括基于共识的区块链和交易链技术、被许可的和未被许可的账本、共享账本及其变型。尽管已经提出并开发了其他区块链实现,但是区块链技术最广为人知的应用是加密货币帐本。尽管为了方便和说明的目的在本文中可能提及加密货币,但是应当注意,本发明不限于与加密货币区块链一起使用,并且替代的区块链实现和协议落入本发明的范围内。术语“用户”在本文中可以指基于人类或基于处理器的资源。另外,本文使用的术语“加密货币”旨在包括源自加密货币协议的协议或实现方式的所有版本和变型。

[0004] 区块链是一种点对点的电子帐本,其被实现为基于计算机的去中心化的、分布式系统,该系统由区块组成,该区块又由交易组成。每个交易是数据结构,该数据结构对区块链系统中的参与者之间的数字资产或资源的控制权的转移进行编码,并包括至少一个输入和至少一个输出。每个区块都包含前一个区块的哈希值,以使得区块被链接在一起来创建自区块链建立以来就已经被写入到该区块链的所有交易的永久、不可更改的记录。交易包含被称为嵌入到交易的输入和输出中的脚本的小程序,这些小程序指定如何以及谁可以访问交易的输出。在加密货币平台上,使用基于堆栈的脚本语言编写这些脚本。

[0005] 为了将交易写入区块链,必须对其进行“验证”。网络节点执行工作以确保每笔交易均有效,而无效交易则被网络拒绝。安装在节点上的软件客户端通过执行其锁定和解锁脚本来对未使用的交易(UTXO)执行此验证工作。如果锁定和解锁脚本的执行评估为TRUE,则该交易有效,并将该交易写入区块链。因此,为了将交易写入区块链,必须:i)由接收交易的第一个节点验证-如果交易被验证,则该节点将其中继到网络中的其他节点;ii)添加到网络节点建造的新区块中;iii)添加到过去交易的公共账本中。

[0006] 尽管区块链技术因使用加密货币实现方式而被广泛了解,但数字企业家已经开始探索使用加密货币所基于的加密安全系统以及可以存储在区块链上的数据这两者以实现新系统。如果区块链可以用于不限于加密货币领域的自动化任务和过程,那将是非常有利的。这样的方案将能够利用区块链的好处,例如,事件的永久性、防篡改记录、分布式处理等,同时在其应用中具有更多用途。

[0007] 原子交换的概念先前已在加密货币社区中讨论过。各方之间的交换是“原子性的”,其从某种意义上说,即所有参与者都收到了他们期望的资源(例如,加密货币令牌或硬

币),或者参与者都没有接收到他们期望的资源。在撰写本文档时,维基百科将原子交换描述为“在加密货币中提出的功能,其允许一种加密货币与另一种加密货币进行交换,而无需受信任的第三方。在传统的加密货币中,诸如加密货币交易所之类的受信任的第三方对于进行加密货币的交换是必需的,以防止一方在没有收到返回的货币情况下发送货币。原子交换系统使用哈希时间锁定的智能合约,从而当事方必须在指定时间内交付要交换的货币,否则交易将被取消。这样可以保持原子性,因为要么进行交换,要么没有交换货币”-https://en.wikipedia.org/wiki/Atomic_swap。

[0008] 因此,原子交换在通过区块链进行的转移方面提供了增强的安全性,因为去除对受信任的第三方的需求消除了利用和恶意干预的风险-对于诸如Mount Gox之类的加密货币交换,已经发生了许多安全漏洞或“黑客攻击”。

[0009] 但是,提出的原子交换方案涉及仅使用一个秘密,并且交换是异步执行的。这产生了以下缺点:必须先花费一笔交易,然后才能花费另一笔交易。

发明内容

[0010] 因此,期望提供一种密码增强的资源交换方法,该方法原子地交换具有由区块链技术提供的不信任和不变性的资源或资产,并且增强了在区块链实现的网络上进行转移的安全性。

[0011] 现在已经设计出了这种改进的方案。

[0012] 根据本发明,提供了一种计算机实现的交换、互换或转移方法。根据附加的或替代的限定,本发明提供了一种安全方法,该安全方法控制资源何时可以或不可以跨网络从发送者传输到接收者。附加地或可替代地,本发明提供了一种方法和对应的系统,其被布置为经由区块链执行资源的原子交换或传输。该方法可以包括以下步骤:

[0013] (i)将第一隐蔽的秘密值从第一用户传达到第二用户,并且将第二隐蔽的秘密值从第二用户传达到第一用户;以及

[0014] (ii)构造第一区块链交易和第二区块链交易,其每一个包括第一隐蔽的秘密值和第二隐蔽的秘密值,交易被布置为在向相应的区块链交易提供第一秘密值和第二秘密值时是可解锁的,以转移对相应的第一资源或第二资源的控制,

[0015] 其中,第一区块链交易的解锁使第一秘密值被揭示给第二用户,第二区块链交易的解锁使第二秘密值被揭示给第一用户。

[0016] 该方法可以通过包括至少两个交易(Tx1和Tx2)来提供原子交换机制,每个交易具有至少一个未花费的输出(UTXO),该至少一个未花费的输出仅在向与相应的输出相关联的多个难题或脚本提供所需的标准时才可以被解锁。换句话说,第一交易中未花费的输出的锁定和解锁标准可以与该交易或另一交易中未花费的输出的锁定标准相同并通过其被镜像。

[0017] 向第一交易中未花费的输出提供所需的解锁标准可以揭示解锁该交易或另一交易中未花费的输出所需的一个或多个秘密值或使其是可访问的。

[0018] 包含秘密值的解锁脚本可以在后续交易的输入中提供,该后续交易在第一交易或第二交易中花费输出。一旦后续交易的解锁脚本与第一交易或第二交易的锁定脚本一起被执行,那么后续交易就可以被验证并随后被发布在区块链上,这使在后续交易的输入中提

供的一个或多个秘密值是从区块链可访问的或可读的。

[0019] 该方法提供了一种确保在不信任的环境中原子交换秘密值的安全方法,在该环境中,该方法的任何用户都没有比其他用户拥有对该方法的更多的控制。

[0020] 秘密值与对应的隐蔽的秘密值有关,因为秘密值不能从隐蔽的秘密值中确定,但是隐蔽的秘密值可以从秘密值中确定。这种关系的示例是将单向函数(例如,哈希或模算术)应用于秘密值提供隐蔽的秘密值。因此,根据一种限定,隐蔽(秘密)值可以是可以从原始(秘密)值导出或已经从原始(秘密)值导出的值,但不能用于确定原始(秘密)值。可能无法对隐蔽值进行逆向工程以提供原始值。

[0021] 短语“解锁交易”可以包括解锁或花费交易中提供的至少一个未花费的输出(UTXO)的含义。这可以通过提供满足与未花费的输出关联的锁定脚本所必需的所需数据/解锁脚本来实现。

[0022] 第一交易和第二交易中的至少一个可以被配置为仅在应用或提供相应的第一私钥和第二私钥时才是可赎回的(可花费的)。

[0023] 这样做的优点在于,只有预定的接收者(由其私钥指示)才能解锁交易。

[0024] 该方法还可包括计算以下中的至少一个的步骤:(a)至少部分地基于第一用户的第一公钥的第一派生公钥;以及(b)至少部分地基于第二用户的第二公钥的第二派生公钥,其中,第一派生公钥是具有第一私钥的加密密钥对的一部分,第二派生公钥是具有第二私钥的加密密钥对的一部分。

[0025] 这使得资产或资源能够存储在派生而不是公开已知的地址处,从而为该方法的用户提供了额外的隐私性和安全性。应当注意,术语“资产”和“资源”在本文中可以互换使用。术语“资产”不应仅被解释为具有财务背景或用途。资产可以是例如代币,其表示区块链上或区块链外的其他实体。

[0026] 计算以下中的至少一个的步骤:(a)至少部分地基于第一用户的第一公钥的第一派生公钥;以及(b)至少部分地基于第二用户的第二公钥的第二派生公钥,还可以包括:第一隐蔽秘密值和第二隐蔽秘密值的组合。

[0027] 这提供了以下优点:提供进行的原子交换和交易之间记录的和不可磨灭的链接。

[0028] 第一隐蔽的秘密值和第二隐蔽的秘密值的组合可以包括第一隐蔽的秘密值和第二隐蔽的秘密值的串接,以及至少一个隐蔽的秘密值与随机或伪随机值的串接中的至少一个。

[0029] 这提供了以下优点:经由附加的确定性混淆进一步提高交易的安全性。

[0030] 该方法还可以包括以下步骤:构造以下中的至少一个:第三区块链交易,其被配置为响应于经过第一交易的非赎回的第一时间段,将对第一资源的控制返回给第一用户;以及第四区块链交易,其被配置为响应于经过第二交易的非赎回的第二时间段,将对第二资源的控制返回给第二用户。

[0031] 这使得该方法的至少一个用户能够在另一用户没有完全参与交换的情况下将对相应的资源的控制返回给他们,从而增加了该方法的多功能性。

[0032] 第一隐蔽的秘密值和第二隐蔽的秘密值中的至少一个可以包括第一秘密值和第二秘密值中的至少一个与第一用户和第二用户两者都可访问的共享秘密值的组合。

[0033] 这提供了增加该方法提供的隐私性和安全性的优点。

[0034] 可以在步骤(i)之前将共享秘密值建立为公共秘密。

[0035] 这提供了进一步增加该方法的安全性的优点。

[0036] 该方法还可以包括以下步骤:

[0037] (iii)从第一秘密值和第二秘密值中的至少一个开始生成至少一个隐蔽的秘密值的序列;

[0038] (iv)使用第一秘密值和第二秘密值中的至少一个来执行任一前述权利要求中的方法;

[0039] (v)赎回至少一个区块链交易以揭示第一秘密值和第二秘密值中的至少一个,从而使该序列的至少一个隐蔽的秘密值被揭示。

[0040] 这使得以比方法的简单重复更高的效率来执行安全原子交换链,因为存储秘密所需的存储空间更少。此外,所需的通信回合更少。这样可以节省时间并提高安全性。

[0041] 执行该方法的至少步骤(ii)的步骤可以使用在该方法的步骤(v)中揭示的至少一个隐蔽的秘密值。

[0042] 这提供了进一步提高该方法的效率的优点。

附图说明

[0043] 通过参考本文描述的实施例,本发明的这些和其他方面将变得显而易见和被阐明。现在将参考附图仅以举例的方式而不是任何限制的方式描述本发明的实施例,其中:

[0044] 图1示出了说明实施本发明的方法中采取的步骤的流程图;

[0045] 图2是用于确定第一节点和第二节点的公共秘密的示例系统的示意图,该系统可以根据本发明用于高度敏感信息的安全传输;

[0046] 图3是用于确定公共秘密的计算机实现的方法的流程图,该方法可以根据本发明用于高度敏感信息的安全传输;

[0047] 图4是用于注册第一节点和第二节点的计算机实现方法的流程图;以及

[0048] 图5是用于确定公共秘密的计算机实现的方法的另一流程图,该方法可以根据本发明用于高度敏感信息的安全传输。

具体实施方式

[0049] 区块链上的原子交易交换意味着对于两个交易,一个交易从第一用户爱丽丝到第二用户鲍勃,而另一个交易从鲍勃到爱丽丝,要么两个交易都完成,要么两个交易都没完成。

[0050] 参照图1,本发明涉及使爱丽丝和鲍勃均能够创建秘密(分别表示为A和B_0)30。如果爱丽丝和鲍勃是可信任的,则他们可以使用通信信道来交换包括这些秘密的信息,该通信信道不是区块链协议的一部分。他们可以使用下面在副标题“确定公共秘密”下描述的安全秘密交换。

[0051] 假设一方不可信任,并且没有分享他们的秘密。本发明提供了该方花费其资金的唯一方法是在区块链上揭示其秘密,从而使该秘密成为公共知识并可供其他用户使用。这是由于交换中使用的交易的配置。因此,该方法不需要任何一方信任另一方。

[0052] 在本发明的实施例中,存在两个秘密:一个秘密由爱丽丝生成并可由爱丽丝访问,

另一个秘密由鲍勃生成并可由鲍勃生访问。这些通过链下信道(off-blockchain channel)进行传达。

[0053] 单原子交换

[0054] 令P_{A_0}表示具有对应的私钥S_{A_0}的爱丽丝的椭圆曲线数字签名算法(ECDSA)公钥,而令P_{B_0}表示具有私钥S_{B_0}的鲍勃的公钥。

[0055] 1.在30处,爱丽丝选择了只有她自己知道的秘密$A_0 \in \mathbb{Z}_n^*$,而鲍勃选择了只有他自己知道的秘密$B_0 \in \mathbb{Z}_n^*$。这些秘密与爱丽丝和鲍勃的公钥或私钥无关。在此,n是椭圆曲线生成器点G的阶数。秘密可以是已通过SHA256(mod n)算法的通用数据结构的形式。

[0056] 2.爱丽丝和鲍勃打开他们之间的通信信道。这可以是使用下面在副标题“确定公共秘密”下描述的方法创建的安全通信信道。然后,他们对各自的秘密进行哈希处理(步骤34),并共享其公钥和各自秘密的哈希值(步骤36)。A_0和B_0的哈希值分别表示为$H(A_0)$和$H(B_0)$,其中,可以使用诸如SHA-256等标准哈希函数。也可以公开地共享值$H(A_0)$和$H(B_0)$。爱丽丝和鲍勃现在都知道

[0057] $P_{A_0}, P_{B_0}, H(A_0), H(B_0)$。

[0058] 3.在38处,爱丽丝和鲍勃计算确定性密钥

[0059] $H(A_0)|H(B_0)$,

[0060] 其中,“|”表示操作OP_CAT,或可替代地,派生的哈希值,例如$H(H(A_0)|H(B_0))$。

[0061] 4.在40处,现在,爱丽丝和鲍勃创建派生的公钥

[0062] 爱丽丝:$P_{A_1} = P_{A_0} + (H(A_0)|H(B_0)) \cdot G$

[0063] 鲍勃:$P_{B_1} = P_{B_0} + (H(A_0)|H(B_0)) \cdot G$。其具有对应的私钥

[0064] 爱丽丝:$S_{A_1} = S_{A_0} + H(A_0)|H(B_0)$

[0065] 鲍勃:$S_{B_1} = S_{B_0} + H(A_0)|H(B_0)$。爱丽丝和鲍勃将使用派生的公钥$P_{A_1}$、$P_{B_1}$执行原子交换。原则上,他们可以使用其原始的公钥$P_{A_0}$、$P_{B_0}$,但是派生的公钥具有与原子交换绑定的优点,并且可以很容易地由爱丽丝和鲍勃而不由其他任何人计算得出(除非$H(A_0)$和$H(B_0)$已公开)。

[0066] 如果在38处还合并确定性伪随机表象值(例如,如下所示的),则可以实现增加的隐私:

[0067] $H(A_0)|H(B_0)|Z$

[0068] 其中,Z是双方都可以计算的值,例如Zeta函数,它是基于共享的起始值预先约定的。

[0069] 5.在42处,爱丽丝和鲍勃构造以下锁定脚本。在此,通过稍后示出的加密货币脚本中的示例性实现示意性地描述了脚本。

[0070]
$$LockingScript(A) = CheckSig\ H(P_{A_1})\ 与\ Solve\ H(A_0)\ 与\ Solve\ H(B_0)$$
$$LockingScript(B) = CheckSig\ H(P_{B_1})\ 与\ Solve\ H(A_0)\ 与\ Solve\ H(B_0)$$

[0071] 过程$CheckSig\ H(P_{A_1})$是针对公钥/私钥对P_{A_1}、S_{A_1}的标准ECDSA签名验证操

作。取而代之的是,可以使用$CheckSig\ H(P_{A_0})$,这是具有公钥/私钥对P_{A_0}、S_{A_0}的标准ECDSA签名验证。过程Solve H(A_0)是具有解A_0的哈希难题(hash puzzle),这意味着解锁脚本必须包含有效值A_0,该值在被哈希处理时等于锁定脚本中提供的H(A_0)。解锁脚本由下式给出

[0072]
$$UnlockingScript(A)=[B_0][A_0][Sig\ P_{A_1}][P_{A_1}]$$
$$UnlockingScript(B)=[B_0][A_0][Sig\ P_{B_1}][P_{B_1}]\quad 。$$

[0073] 此处,可以看出,如果爱丽丝或鲍勃解锁他们的资金,则他们必须在区块链上暴露值A_0和B_0。

[0074] 6.在42处,爱丽丝通过锁定脚本LockingScript(B)创建到P_{B_1}的交易tx_1,而鲍勃使用锁定脚本LockingScript(A)创建到P_{A_1}的交易tx_2。在该阶段,由于爱丽丝和鲍勃都不知道A_0和B_0,因此双方都不能在P_{A_1}和P_{B_1}处花费资金。这些交易被发送到网络,随后出现在区块链上。

[0075] 7.在46C处,爱丽丝向鲍勃发送她的秘密A_0,鲍勃向爱丽丝发送他的秘密B_0。这是使用上面建立的爱丽丝和鲍勃之间的通信信道执行的。爱丽丝和鲍勃可以通过确认其哈希值等于H(A_0)和H(B_0)来检查这些值是正确的。

[0076] 8.假设爱丽丝和鲍勃都是诚实的并且共享其正确的秘密,则双方都知道这两个秘密(步骤48C),并且都可以花费锁定在P_{A_1}和P_{B_1}中的资金(步骤50C),完成原子交换。

[0077] 9.例如,假设鲍勃没有将其正确的秘密B_0发送给爱丽丝。即,假设只有爱丽丝发送她的秘密,发生步骤46B而不是46C。由于锁定脚本LockingScript(B)的形式,为了使鲍勃花费锁定的资金P_{B_1},他必须在解锁脚本中公开地暴露他的秘密B_0。因此,只要鲍勃花了他的资金,爱丽丝就得知鲍勃的秘密(步骤48B),因此能够花费她在P_{A_1}中的资金(步骤50B)。这样可以确保要么爱丽丝和鲍勃都可以花费他们的资金,要么都不能花费他们的资金。

[0078] 以下是上述步骤4中与加密货币区块链兼容的爱丽丝的示例性锁定和解锁脚本。

[0079] 爱丽丝的锁定脚本:

[0080] OP_DUP OP_HASH160<Hash160 P_{A_1} >OP_EQUALVERI FY OP_CHECKS IG OP_HASH256<Hash256 A_0>OP_EQUALVERIFY OP_HASH256<Hash256 B_0>OP_EQUALVERIFY

[0081] 爱丽丝的解锁脚本:

[0082] <B_0> <A_0> <Sig P_{A_1}> <P_{A_1}>

[0083] 请注意,到支付到公钥哈希(P2PKH)地址和支付到脚本哈希(P2SH)地址的交易都允许锁定和解锁上述类型的脚本。对于P2SH地址,锁定脚本被呈现为包含相同信息的赎回脚本的哈希。

[0084] 以上方法是参照区块链进行描述的,该区块链使用类似于加密货币区块链上使用的ECSDA的公钥/私钥加密系统。但是,该方法可以推广到通用加密机制,该机制要求在解锁脚本中暴露的秘密的通用形式,其可以是任意数据结构。所需的是锁定脚本、交易和区块

链,这是安全、可验证的通信信道。

[0085] 时间锁退款交易

[0086] 如果鲍勃拒绝给爱丽丝其正确的秘密B_0,并且也没有解锁其存储在地址P_{B_1}中的资金,那么鲍勃的秘密将不会被揭示给爱丽丝,她也永远无法解锁她存储在P_{A_1}中的资金。此外,爱丽丝也永远无法收回她发送给鲍勃的存储在P_{B_1}中的资金。

[0087] 可以通过引入从鲍勃到爱丽丝的新交易来解决该问题,该新交易被配置为如果资金未被花费,那么一定量的时间后将资金发送回去。这还需要稍微修改LockingScript(A)和LockingScript(B),下面将对修改进行描述。

[0088] 该新交易使用锁定脚本中依赖于时间的操作,该操作仅在经过特定的预定时间后才允许区块接受交易。例如,在加密货币脚本中,这可以是自指定值以来的相对时间量的操作“检查序列验证”(CSV)或者固定时间值的“检查锁定时间验证”(CLTV)。

[0089] 修改了上面步骤4中的锁定脚本,以包括如果爱丽丝和鲍勃都同意签字则进行花费的选项,如下所示:

$$LockingScript'(A)=\begin{cases}CheckSig\ H(P_{A_1})\ 与\ Solve\ H(A_0)\ 与\ Solve\ H(B_0)\\ 或\\ CheckSig\ H(P_{A_1})\ 与\ CheckSig\ H(P_{B_1})\end{cases}$$

[0090]

$$LockingScript'(B)=\begin{cases}CheckSig\ H(P_{B_1})\ 与\ Solve\ H(A_0)\ 与\ Solve\ H(B_0)\\ 或\\ CheckSig\ H(P_{A_1})\ 与\ CheckSig\ H(P_{B_1})\end{cases}$$

[0091] 在44处,然后在上述方法中的步骤4之后和步骤5之前创建两个新交易。爱丽丝创建了从P_{A_1}到鲍勃的交易tx_4,该交易返回了鲍勃所有的资金。该交易是时间锁定的,使得该交易仅可以在一定量的时间(例如,24小时)之后在区块中被接受。鲍勃创建了从P_{B_1}到爱丽丝的类似的交易tx_3。交易tx和tx_4具有相应的锁定脚本

$$LockingScript2(A)=CheckSig\ H(P_{A_1})\ 与\ CSV(24\ hours)$$

[0092]

$$LockingScript2(B)=CheckSig\ H(P_{B_1})\ 与\ CSV(24\ hours)$$

[0093] 爱丽丝签署tx_4并将其发送给鲍勃,鲍勃则对其进行签名并将其发送给网络。类似地,鲍勃签署tx_3并将其发送给爱丽丝,爱丽丝则对其进行签名并将其发送给网络。

[0094] 在此阶段,如果任何一方都不遵守,则该过程将被放弃,并且资金不会被转移。如果双方都遵守,则执行以上方法的步骤5(42)。现在,如果任何一方都没有花费在原子交换中交换的资金(46A),则这些资金将在24小时后退还给原始所有者(48A,50A)。

[0095] 请注意,此处以24小时的CSV相对时间为例,但是有可能使用将来的任何相对时间或将来的任何特定时间(例如,使用CLTV运算符)。

[0096] 使用加密货币区块链在24小时后将资金返还给爱丽丝的tx_3的锁定脚本的示例是

[0097] “24h”OP_CHECKSEQUENCEVERIFY OP_DROP OP_DUP OP_HASH160

<Hash160 P_{A_1}>OP_EQUALVERIFY OP_CHECKS IG

[0098] 对应的解锁脚本由<Sig P_{A_1}> <P_{A_1}>给出

[0099] 秘密值的掩蔽

[0100] 另一替代实施例包括掩蔽步骤32,使得值A_0和B_0仅对爱丽丝和鲍勃是已知的,并且从不公开。

[0101] 最初,爱丽丝和鲍伯都同意只有他们知道的共享秘密S_c。这可以通过使用下面描述的标题为“确定公共秘密”的方法安全地交换秘密来实现。

[0102] 然后爱丽丝和鲍勃定义新的秘密

[0103] $A'_0 = A_0 + S_c$

[0104] $B'_0 = B_0 + S_c$。

[0105] 然后,他们通过掩蔽的秘密A'_0、B'_0而不是原始秘密按照上面概述的方法进行操作。在原子交换期间,只有掩蔽的秘密才在区块链上被揭示给公众。

[0106] 如果秘密值A_0和B_0也将在其他背景下(例如,在下面描述的其他实施例中)使用,则这是有用的。

[0107] 另一替代实施例使爱丽丝和鲍勃能够进行一系列的n个原子交换。各方从随机秘密开始,并创建该秘密的哈希值的序列,这称为访问链。当执行原子交换时,它将暴露要在下一个原子交换中使用的下一个秘密的哈希值。迭代地重复该过程,直到n次的最大值。

[0108] 该方法在一系列单独的交换上可高效节约,因为爱丽丝和鲍勃一次只需要存储一个秘密,秘密所需的存储空间更少。他们可以从哈希先前的秘密来计算下一个秘密。由于他们不需要每次都传达他们秘密的哈希,因此他们彼此之间需要更少的通信回合。

[0109] 该方法如下:

[0110] 爱丽丝和鲍勃同意重复交换的次数n。他们分别创建随机值A_n和B_n。爱丽丝计算以下访问链:

[0111]
$$A_n = \text{随机}$$
$$A_{n-1} = \text{哈希}(A_n)$$
$$A_{n-2} = \text{哈希}\ (A_{n-1})$$
$$\vdots$$
$$A_{i-1} = \text{哈希}\ (A_i)$$
$$\vdots$$
$$A_0 = \text{哈希}\ (A_1)$$

[0112] 鲍勃从B_n开始计算等效链。这些链对应于将在一系列交换中使用的秘密值。可能的交换的数量将按顺序{0,1,...,n}。也就是说,各方可以在需要重新初始化新链之前将这些值用于0和n个交易之间的交换。

[0113] 下面概述了实现交换的方法。应当理解,鲍勃遵循等效的过程。

[0114] 1.爱丽丝从她的链$A_0, A_1, \ldots, A_n$、鲍勃的公钥P_{B_0}以及鲍勃的秘密的哈希值$H(B_0)$开始。和以前一样,鲍勃可以公开地分享$H(B_0)$。

[0115] 2.爱丽丝计算派生的公钥

[0116] 爱丽丝:$P_{A_1} = P_{A_0} + (H(A_0) | H(B_0)) \cdot G$

[0117] 鲍勃:$P_{B_1} = P_{B_0} + (H(A_0) | H(B_0)) \cdot G$,然后计算锁定脚本

$$LockingScript(A)_0 = CheckSig\ H(P_{A_1})\ 与\ Solve\ H(A_0)\ \ Solve\ H(B_0)$$

[0118]

$$LockingScript(B)_0 = CheckSig\ H(P_{B_1})\ 与\ Solve\ H(A_0)\ \ Solve\ H(B_0).$$

[0119] 注意,在较早的实施例中描述的依赖于时间的退款可以被包括在上述锁定脚本中,而无需对逻辑进行任何实质性改变。

[0120] 3.爱丽丝和鲍勃执行第一交换。如上所述,这涉及在爱丽丝和鲍勃之间交换A_0和B_0。这意味着在交换之后,爱丽丝现在知道$H(B_1) = B_0$。

[0121] 4.爱丽丝重复该方法的步骤2,但是使用了链中的鲍勃的第二秘密的哈希值$H(B_1)$。明确地,她计算派生的公钥

[0122] 爱丽丝:$P_{A_2} = P_{A_1} + (H(A_1) | H(B_1)) \cdot G$

[0123] 鲍勃:$P_{B_2} = P_{B_1} + (H(A_1) | H(B_1)) \cdot G$,和锁定脚本

$$LockingScript(A)_1 = CheckSig\ H(P_{A_2})\ 与\ Solve\ H(A_1)\ \ Solve\ H(B_1)$$

[0124]

$$LockingScript(B)_1 = CheckSig\ H(P_{B_2})\ 与\ Solve\ H(A_1)\ \ Solve\ H(B_1).$$

[0125] 5.一旦已经完成了第二交换,爱丽丝就知道了$H(B_2) = B_1$。她利用鲍勃的第三秘密的哈希值$H(B_2)$再次重复步骤2。

[0126] 6.迭代地重复该过程,直到任一交换未完成或已达到n个交换的最大数量。

[0127] 如较早的实施例中所述,可以通过将伪随机值Z_i引入到操作$H(A_i) | H(B_i) | Z_i$来合并进一步的安全性。在这种情况下,该函数应通过使用哈希函数$Z_{i-1} = H(Z_i)$变换每次迭代。

[0128] 上面概述的原子交换方法不限于加密货币区块链。上述原子交换方法中的重要组成部分是,当一方在步骤7中花费其资金时,他们会在区块链上揭示其秘密。这意味着上述方法可用于在任何允许锁定和解锁步骤4中给出的形式的脚本的区块链上执行原子交换。

[0129] 此外,原子交换方法可以用于交换加密货币。例如,它可以用于爱丽丝在一种加密货币区块链上将一种加密货币发送给鲍勃,以及用于鲍勃在另一种加密货币区块链上将另一种加密货币发送给爱丽丝。

[0130]

	发送	接收
爱丽丝	BCH	Eth
鲍勃	Eth	BCH

[0131] 对两个不同区块链之间的原子交换的唯一限制是,这两个区块链允许在锁定脚本(或等效脚本)的哈希难题中使用相同的哈希函数。原因如下:假设爱丽丝的区块链仅允许

使用SHA-256哈希算法,而鲍勃的区块链仅允许使用SHA-384算法。鲍勃向爱丽丝发送一个秘密的SHA-256哈希,但是在鲍勃的锁定脚本中,他为不同的秘密设置了SHA-384哈希难题。当鲍勃花费其资金时,解锁脚本将向爱丽丝揭示无用的秘密,直到鲍勃已经花光他的资金,爱丽丝才知道该秘密。

[0132] 根据另一实施例,提供了一种方法,该方法使双方均能够创建公钥,对于该公钥,仅使得双方都可访问对应的私钥或都不可访问对应的私钥。该方法利用上述原子交换方法,以便在双方之间交换两个秘密值。这些秘密值用于计算私钥。

[0133] 该方法的一种应用是,它允许两方交换由单个公钥/私钥对控制的多种类型的加密货币。

[0134] 该方法使爱丽丝和鲍勃均能够创建公钥,对于该公钥,直到发生原子交换,才知道私钥。原子交换确保爱丽丝和鲍勃都可以计算其对应的私钥,或者都不可以计算其私钥。

[0135] 下面使用ECSDA私钥和公钥对对该方法进行描述。然而,该方法不是严格地依赖于ECDSA协议,并且可以容易地适用于任何基于公钥/私钥的密码系统,对于任何基于公钥/私钥的密码系统,可以从现有的私钥和公共已知的确定性密钥确定性地创建新的安全公钥。

[0136] 该方法是匿名的,在某种意义上来说,关于新私钥的部分信息存储在为开放式账本的一个或多个区块链上。但是,只有该过程中涉及的各方才能解码此信息,因此安全性永远不会受到损害。

[0137] 1.爱丽丝以私钥S_A和对应的公钥$P_A=S_A \cdot G$以及只有她自己知道的秘密S_2开始。鲍勃以私钥S_B和对应的公钥$P_B=S_B \cdot G$以及只有他自己知道的秘密S_1开始。

[0138] 2.爱丽丝向鲍勃发送$P_2=S_2 \cdot G$,鲍勃向爱丽丝发送$P_1=S_1 \cdot G$。由于秘密被乘以椭圆曲线基点,因此在此过程中它们不会被暴露,P_2和P_1可能是公开已知的。

[0139] 3.爱丽丝创建新的公钥$P_{AE}=P_A+P_1$,其可以用作接收加密货币交易的地址,或类似替代币的地址。鲍勃创建新的公钥$P_{BE}=P_B+P_2$。

[0140] 根据椭圆曲线密码学的特性,P_{AE}的对应私钥为$S_{AE}=S_A+S_1$,这意味着$P_{AE}=S_{AE} \cdot G$。P_{BE}的对应私钥为$S_{BE}=S_B+S_2$。

[0141] 在此阶段,爱丽丝不知道S_1,因此不知道P_{AE}的私钥。尽管鲍勃知道S_1,但他不知道S_A,因此也不知道P_{AE}的私钥。按照同样的逻辑,爱丽丝和鲍勃都不知道P_{BE}的私钥。

[0142] 4.爱丽丝向鲍勃的地址P_{BE}进行交易,而鲍勃向爱丽丝的地址P_{AE}进行交易。这些交易可以是使用公钥/私钥系统的任何加密货币的交换,或者它们可以将令牌甚至物理资产转移给公钥P_{AE}和P_{BE}的所有者。它也可以是上述的组合。

[0143] 5.如上所述,爱丽丝和鲍勃现在使用任何区块链通过S_2和S_1作为各自的秘密来初始化原子交换。

[0144] 6.爱丽丝和鲍勃交换秘密。这意味着:

[0145]

	发送	接收
爱丽丝	S_2	S_1
鲍勃	S_1	S_2

[0146] 爱丽丝和鲍勃可以使用公式$P_1=S_1 \cdot G$和$P_2=S_2 \cdot G$来检查他们已经接收到了正确的秘密。如果他们没有交换正确的值,那么他们将不能花费原子交换的输出。

[0147] 7.现在,爱丽丝拥有S_1,她可以计算与P_{AE}相对应的私钥。由于除了爱丽丝之外,没有其他人知道她的私钥S_A,因此即使S_1是公开已知的,也没有其他人可以计算出与P_{AE}相对应的私钥。类似地,既然鲍勃拥有秘密S_2,则他可以计算对应于P_{BE}的私钥,并且除了鲍勃之外,没有其他人可以这样做。

[0148] 如果爱丽丝和鲍勃都不花费原子交换的交易输出,则爱丽丝的秘密S_2不会暴露给鲍勃,鲍勃的秘密S_1也不会暴露给爱丽丝。在这种情况下,爱丽丝和鲍勃都无法计算与P_{AE}和P_{BE}对应的私钥。

[0149] 区块链使用公钥/私钥加密系统签署交易并证明对交易输出的所有权。这使得能够使用以上实施例的方法来以几种加密货币同时向P_{AE}和P_{BE}发送交易。例如,在上述步骤3中建立P_{AE}和P_{BE}之后:

[0150] 爱丽丝将BCH和ETH中的资金转移到P_{BE}。

[0151] 鲍勃将BCH和DASH中的资金转移到P_{AE}。

[0152] 一旦已经执行了原子交换,就解锁了P_{BE}和P_{AE}的私钥。这些解锁了爱丽丝持有的一种加密货币公钥以及鲍勃持有的另一种加密货币公钥中的资金。因此,从爱丽丝到鲍勃的以下交易可以安全地完成

[0153]

	发送	接收
爱丽丝	BCH, Eth	BCH,DASH

[0154]

鲍勃	BCH,DASH	BCH, Eth

[0155] 请注意,这些区块链不必在其锁定脚本中允许相同的哈希函数。

[0156] 上面提供了两方通过使用原子交换进行的秘密交换来解锁公钥的通用方法。这具有超越加密货币交换的应用,并且与使用类似于ECDSA的公钥/私钥加密方案的任何系统有关。例如,其他用例包括但不限于:

[0157] 1.提供对分布式哈希表(DHT)的访问;

[0158] 2.加密计算;

[0159] 3.私人电子邮件客户;

[0160] 4.获取物流数据和交换;

[0161] 5.商品和服务的交换;

[0162] 6.值的私有交换;以及

[0163] 7.密钥层次结构。

[0164] 确定公共秘密

[0165] 在适当的情况下,可以通过以下方法来提高安全性:使用例如如下所述的公钥/私钥系统在两方之间交换信息的安全方法。

[0166] 公共秘密(CS)可以在两方之间建立,然后用于生成用于传输一个或多个份额的安全的加密密钥。公共秘密(CS)被生成并用于使任何秘密($S_{A,B,1,2}$)(例如,秘密值、密钥或其份额)能够安全交换。

[0167] 在下文中,为了方便起见,爱丽丝和鲍勃将被称为第一节点(C)和第二节点(S)。目的是生成两个节点都知道的公共秘密(CS),但是未经由通信信道发送该公共秘密,从而消除了未经授权的发现的可能性。

[0168] 安全传输技术涉及在传输的每一端以独立的方式生成CS,因此,尽管两个节点都知道CS,但是CS不必在可能不安全的通信信道上传输。一旦已经在两端处建立了CS,就可以使用它来生成安全的加密密钥,此后两个节点都可以使用该安全的加密密钥进行通信。

[0169] 图2示出了系统1,该系统包括第一节点3,该第一节点3通过通信网络5与第二节点7通信。第一节点3具有关联的第一处理装置23,第二节点5具有关联的第二处理装置27。第一节点3和第二节点7可以包括电子装置,例如计算机、电话、平板计算机、移动通信装置、计算机服务器等。在一个示例中,第一节点3可以是客户端(用户)装置,第二节点7可以是服务器。该服务器可以是数字钱包提供商的服务器。

[0170] 第一节点3与具有第一节点主私钥(V_{1C})和第一节点主公钥(P_{1C})的第一非对称密码对相关联。第二节点(7)与具有第二节点主私钥(V_{1S})和第二节点主公钥(P_{1S})的第二非对称密码对相关联。换句话说,第一节点和第二节点均具有相应的公钥-私钥对。

[0171] 相应的第一节点3和第二节点7的第一非对称密码对和第二非对称密码对可以在诸如钱包的注册之类的注册过程期间生成。每个节点的公钥可以例如通过通信网络5被公开地共享。

[0172] 为了确定第一节点3和第二节点7两者处的公共秘密(CS),节点3、7执行相应的方法300、400的步骤,而无需通过通信网络5传达私钥。

[0173] 由第一节点3执行的方法300包括:至少基于第一节点主私钥(V_{1C})和生成器值(GV)来确定330第一节点第二私钥(V_{2C})。生成器值可以基于在第一节点和第二节点之间共享的消息(M),其可以包括如下文更详细描述的在通信网络5上共享消息。方法300还包括至少基于第二节点主公钥(P_{1S})和生成器值(GV)来确定370第二节点第二公钥(P_{2S})。方法300包括基于第一节点第二私钥(V_{2C})和第二节点第二公钥(P_{2S})确定380公共秘密(CS)。

[0174] 也可以通过方法400在第二节点7处确定相同的公共秘密(CS)。方法400包括:基于第一节点主公钥(P_{1C})和生成器值(GV)来确定430第一节点第二公钥(P_{2C})。方法400还包括基于第二节点主私钥(V_{1S})和生成器值(GV)来确定470第二节点第二私钥(V_{2S})。方法400包括基于第二节点第二私钥(V_{2S})和第一节点第二公钥(P_{2C})确定480公共秘密(CS)。

[0175] 通信网络5可以包括局域网、广域网、蜂窝网络、无线电通信网络、互联网等。可以经由诸如电线、光纤之类的通信介质来传输数据或者无线地传输数据的这些网络可能例如通过窃听者11容易被窃听。方法300、400可以允许第一节点3和第二节点7两者独立地确定公共秘密,而无需在通信网络5上传输公共秘密。

[0176] 因此,一个优点是,可以由每个节点安全且独立地确定公用秘密(CS),而不必通过可能不安全的通信网络5传输私钥。反过来,公用秘密可以用作秘密密钥(或作为秘密密钥的基础),用于在第一节点3和第二节点7之间通过通信网络5进行加密通信。

[0177] 方法300、400可以包括附加步骤。方法300可以包括在第一节点3处基于消息(M)和第一节点第二私钥(V_{2C})来生成签名的消息(SM1)。方法300还包括通过通信网络向第二节点7发送360第一签名的消息(SM1)。反过来,第二节点7可以执行接收440第一签名的消息(SM1)的步骤。方法400还包括以下步骤:利用第一节点第二公钥(P_{2C})来验证450第一签名的消息(SM2),以及基于验证第一签名的消息(SM1)的结果来认证460第一节点3。有利地,这允许第二节点7认证声称的第一节点(其中生成了第一签名的消息)是第一节点3。这基于这样的假设:仅第一节点3可以访问第一节点主私钥(V_{1C}),因此只有第一节点3可以确定第一节

点第二私钥(V_{2C}),以生成第一签名的消息(SM1)。应当理解,类似地,第二签名的消息(SM2)可以在第二节点7处生成并被发送到第一节点3,使得第一节点3可以例如在对等场景(peer-to-peer scenario)中认证第二节点7。

[0178] 可以以各种方式来实现在第一节点与第二节点之间共享消息(M)。在一个示例中,消息可以在第一节点3处生成,然后通过通信网络5被发送到第二节点7。或者,消息可以在第二节点7处生成然后通过通信网络5被发送到第二节点7。在又一示例中,该消息可在第三节点9处生成,并且该消息被发送到第一节点3和第二节点7。在又一替代方案中,用户可以通过用户界面15输入要由第一节点3和第二节点7接收的消息。在又一示例中,消息(M)可以从数据存储19中检索并发送到第一节点3和第二节点7。在一些示例中,消息(M)可以是公开的,因此可以在不安全的网络5上传输。

[0179] 在其他示例中,一个或多个消息(M)可以存储在数据存储13、17、19中,其中,该消息可以与某个实体相关联,例如,数字钱包或在第一节点3和第二节点7之间建立的通信会话。因此,消息(M)可被检索并用于在相应的第一节点3和第二节点7处重新创建与该钱包或会话相关联的公共秘密(CS)。

[0180] 有利地,可以保留允许重建公共秘密(CS)的记录,而记录本身不必私下地存储或安全地传输。如果在第一节点3和第二节点7处执行大量交易并且将所有消息(M)都存储在节点自身处是不切实际的时候,这可能是有利的。

[0181] 将参考图4描述注册方法100、200的示例,其中,方法100由第一节点3执行,方法200由第二节点7执行。这包括建立相应的第一节点3和第二节点7的第一非对称密码对和第二非对称密码对。

[0182] 非对称密码对包括关联的私钥和公钥,例如在公钥加密中使用的私钥和公钥。在此示例中,使用椭圆曲线密码术(ECC)和椭圆曲线操作的特性生成非对称密码对。

[0183] ECC的标准可以包括已知的标准,例如高效密码技术标准组(www.sceg.org)描述的那些标准。椭圆曲线密码术还在以下文献中进行描述:US 5,600,725、US 5,761,305、US 5889,865、US 5,896,455、US 5,933,504、US 6,122,736、US6,141,420、US 6,618,483、US 6,704,870、US 6,785,813、US 6,078,667、US 6,792,530。

[0184] 在方法100、200中,这包括第一节点和第二节点在公共ECC系统上达成协议110、210并且使用基点(G)(请注意:基点可以称为公共生成器,但术语“基点”用于避免与生成器值GV混淆)。在一个示例中,公共ECC系统可以基于secp256K1,secp256K1是加密货币使用的ECC系统。可以选择、随机生成或分配基点(G)。

[0185] 现在转到第一节点3,方法100包括选定110公共ECC系统和基点(G)。这可以包括从第二节点7或第三节点9接收公共ECC系统和基点。可替代地,用户界面15可以与第一节点3相关联,由此用户可以选择性地提供公共ECC系统和/或基点(G)。在又一替代方案中,公共ECC系统和/或基点(G)之一或二者可以由第一节点3随机地选择。第一节点3可以通过通信网络5向第二节点7发送指示使用公共ECC系统以及基点(G)的通知。反过来,第二节点7可以通过发送指示使用公共ECC系统和基点(G)的确认的通知来进行选定210。

[0186] 方法100还包括第一节点3生成120第一非对称密码对,该第一非对称密码对包括第一节点主私钥(V_{1C})和第一节点主公钥(P_{1C})。这包括至少部分地基于在公共ECC系统中指定的可允许范围内的随机整数来生成第一主私钥(V_{1C})。这还包括根据以下公式基于第一节

点主私钥(P_{1C})和基点(G)的椭圆曲线点乘法来确定第一节点主公钥(P_{1C}):

[0187] $P_{1C}=V_{1C}$ x G(等式1)

[0188] 因此,第一非对称密码对包括:

[0189] V_{1C}:第一节点秘密保留的第一节点主私钥。

[0190] P_{1C}:公开已知的第一节点主公钥。

[0191] 第一节点3可以将第一节点主私钥(V_{1C})和第一节点主公钥(P_{1C})存储在与第一节点3相关联的第一数据存储13中。为了安全起见,第一节点主私钥(V_{1C})可以存储在第一数据存储13的安全部分中,以确保密钥保持私有。

[0192] 方法100还包括通过通信网络5向第二节点7发送130第一节点主公钥(P_{1C})。第二节点7在接收到220第一节点主公钥(P_{1C})时,可以在与第二节点7相关联的第二数据存储17中存储230第一节点主公钥$_{(P1C}$。

[0193] 类似于第一节点3,第二节点7的方法200包括生成240第二非对称密码对,该第二非对称密码对包括第二节点主私钥(V_{1S})和第二节点主公钥(P_{1S})。第二节点主私钥(V_{1S})也是可允许范围内的随机整数。反过来,第二节点主公钥(P_{1S})由以下公式确定:

[0194] $P_{1S}=V_{1S}$ x G(等式2)

[0195] 因此,第二非对称密码对包括:

[0196] V_{1S}:第二节点将其保密的第二节点主私钥。

[0197] P_{1S}:公开已知的第二节点主公钥。

[0198] 第二节点7可以将第二非对称密码对存储在第二数据存储17中。方法200还包括将第二节点主公钥(P_{1S})发送250到第一节点3。反过来,第一节点3可以接收140并存储150第二节点主公钥(P_{1S})。

[0199] 应当理解,在一些替代方案中,可以接收相应的公共主密钥并将其存储在与第三节点9(例如,受信任的第三方)相关联的第三数据存储19处。这可以包括充当共用目录(public directory)的第三方,例如认证机构。因此,在一些示例中,仅当需要确定公共秘密(CS)时,第二节点7才可以请求和接收第一节点主公钥(P_{1C}),反之亦然。

[0200] 注册步骤可能只需要发生一次,作为例如数字钱包的初始设置。

[0201] 现在将参考图5描述确定公共秘密(CS)的示例。公共秘密(CS)可以用于第一节点3和第二节点7之间的特定会话、时间、交易或其他目的,因此使用同一公共秘密(CS)可能并不是期望的,或也是不安全的。因此,可以在不同的会话、时间、交易等之间改变公共秘密(CS)。

[0202] 提供以下内容以说明上述安全传输技术。

[0203] 在该示例中,由第一节点3执行的方法300包括生成310消息(M)。消息(M)可以是随机的、伪随机的或用户限定的。在一个示例中,消息(M)基于Unix时间和随机数(和任意值)。例如,消息(M)可以被提供为:

[0204] 消息(M)=Unix时间+随机数(等式3)

[0205] 在一些示例中,消息(M)是任意的。然而,应当理解,消息(M)可以具有在某些应用中可能有用的选择值(例如,Unix时间等)。

[0206] 方法300包括通过通信网络3向第二节点7发送315消息(M)。由于消息(M)不包括关于私钥的信息,所以消息(M)可以通过不安全的网络发送。

[0207] 方法300还包括基于消息(M)确定320生成器值(GV)的步骤。在该示例中,这包括确定消息的密码哈希。密码哈希算法的示例包括SHA-256,用于创建256位生成器值(GV)。即:

[0208] GV＝SHA-256(M)(等式4)

[0209] 应当理解,可以使用其他哈希算法。这可以包括安全哈希算法(SHA)族中的其他算法。一些特定示例包括SHA-3子集中的实例,包括SHA3-224、SHA3-256、SHA3-384、SHA3-512、SHAKE128、SHAKE256。其他哈希算法可以包括RACE完整性基元评估消息摘要(RIPEMD)族中的算法。特定示例可以包括RIPEMD-160。其他哈希函数可以包括基于Zémor-Tillich哈希函数的族和基于背包(knapsack)的哈希函数。

[0210] 然后方法300包括基于第二节点主私钥(V_{1C})和生成器值(GV)来确定330第一节点第二私钥(V_{2C})的步骤330。这可以根据以下公式基于第一节点主私钥(V_{1C})和生成器值(GV)的标量加法:

[0211] $V_{2C}＝V_{1C}+GV$(等式5)

[0212] 因此,第一节点第二私钥(V_{2C})不是随机值,而是确定性地从第一节点主私钥导出的。密码对中对应的公钥(即第一节点第二公钥(P_{2C}))具有以下关系:

[0213] $P_{2C}＝V_{2C}\ x\ G$(等式6)

[0214] 将来自等式5的V_{2C}代入公式6提供:

[0215] $P_{2C}＝(V_{1C}+GV)x\ G$(等式7)

[0216] 其中,“+”运算符表示椭圆曲线点加法。注意,椭圆曲线密码代数是分布式的,等式7可以表示为:

[0217] $P_{2C}＝V_{1C}\ x\ G+GV\ x\ G$(等式8)

[0218] 最后,等式1可以被代入等式7以提供

[0219] $P_{2C}＝P_{1C}+GV\ x\ G$ (等式9.1)

[0220] $P_{2C}＝P_{1C}+SHA\text{-}256(M)x\ G$ (等式9.2)

[0221] 因此,给定第一节点主公钥(P_{1C})和消息(M)的知识,对应的第一节点第二公钥(P_{2C})可以是可导出的。第二节点7可以具有这样的知识,以独立地确定第一节点第二公钥(P_{2C}),如下面将相关于方法400更详细地讨论的。

[0222] 方法300还包括基于消息(M)和确定的第一节点第二私钥(V_{2C})来生成350第一签名的消息(SM1)。生成签名的消息包括应用数字签名算法对消息(M)进行数字签名。在一个示例中,这包括在椭圆曲线数字签名算法(ECDSA)中将第一节点第二私钥(V_{2C})应用于消息,以获得第一签名的消息(SM1)。

[0223] ECDSA的示例包括基于具有secp256k1、secp256r1、secp384r1、se3cp521r1的ECC系统的ECDSA。

[0224] 可以在第二节点7处利用对应的第一节点第二公钥(P_{2C})验证第一签名的消息(SM1)。第二节点7可以使用对第一签名的消息(SM1)的该验证来认证第一节点3,这将在下面的方法400中讨论。

[0225] 然后,第一节点3可以确定370第二节点第二公钥(P_{2S})。如上所述,第二节点第二公钥(P_{2S})可以至少基于第二节点主公钥(P_{1S})和生成器值(GV)。在该示例中,由于公钥通过基点(G)和椭圆曲线点乘法被确定370’为私钥,因此可以以类似于等式6的方式表示第二节点第二公钥(P_{2S}),如下:

[0226] $P_{2S}=V_{2S}$ x G (等式10.1)

[0227] $P_{2S}=P_{1S}+GV$ x G(等式10.2)

[0228] 等式10.2的数学证明与上述推导第一节点第二公钥(P_{2C})的等式9.1所描述的相同。应当理解,第一节点3可以独立于第二节点7确定370第二节点第二公钥。

[0229] 然后,第一节点3可以基于确定的第一节点第二私钥(V_{2C})和确定的第二节点第二公钥(P_{2S})来确定380公共秘密(CS)。可以由第一节点3通过以下公式确定公共秘密(CS):

[0230] $S=V_{2C}$ x P_{2S}(等式11)

[0231] 在第二节点7处执行的方法400

[0232] 现在将描述在第二节点7处执行的对应的方法400。应当理解,这些步骤中的一些类似于第一节点3执行的上述步骤。

[0233] 方法400包括通过通信网络5从第一节点3接收410消息(M)。这可以包括在步骤315处由第一节点3发送的消息(M)。然后第二节点7基于消息(M)确定420a生成器值(GV)。第二节点7确定420生成器值(GV)的步骤与上述第一节点执行的步骤320相似。在该示例中,第二节点7独立于第一节点3执行该确定步骤420。

[0234] 下一步骤包括基于第一节点主公钥(P_{1C})和生成器值(GV)来确定430第一节点第二公钥(P_{2C})。在此示例中,由于公钥通过基点(G)和椭圆曲线点乘法被确定430'为私钥,因此可以以类似于等式9的方式表示第一节点第二公钥(P_{2c}),如下:

[0235] $P_{2C}=V_{2C}$ x G(等式12.1)

[0236] $P_{2C}=P_{1C}+GV$ x G(等式12.2)

[0237] 等式12.1和12.2的数学证明与上面针对公式10.1和10.2讨论的证明相同。

[0238] 方法400可以包括由第二节点7执行的步骤,用于认证声称的第一节点3是第一节点3。如先前所讨论的,这包括从第一节点3接收440第一签名的消息(SM1)。然后第二节点7可以利用第一节点第二公钥(P_{2C})来验证450第一签名的消息(SM1)上的签名,该第一节点第二公钥(P_{2C})是在步骤430处确定的。

[0239] 可以根据如上所述的椭圆曲线数字签名算法(ECDSA)来完成对数字签名的验证。重要的是,由于V_{2C}和P_{2C}形成了密码对,所以利用第一节点第二私钥(V_{2C})签名的第一签名的消息(SM1)应该仅通过对应的第一节点第二公钥(P_{2C})进行正确地验证。由于在第一节点3的注册时生成的第一节点主私钥(V_{1C})和第一节点主公钥(P_{1C})上这些密钥是确定的,因此,验证第一签名的消息(SM1)可以被用作以下验证的基础:验证在注册期间发送第一签名的消息(SM1)的声称的第一节点为同一第一节点3。因此,第二节点7还可以执行基于验证(450)第一签名的消息的结果,认证(460)第一节点3的步骤。

[0240] 以上认证可以适用于以下情景:两个节点之一是受信任节点,并且仅一个节点需要被认证。例如,第一节点3可以是客户端,而第二节点7可以是客户端信任的服务器,例如钱包提供商。因此,服务器(第二节点7)可能需要认证客户端(第一节点3)的凭证(credential),以允许客户端访问服务器系统。服务器可能没有必要向客户端认证服务器的凭证。然而,在某些情景下,例如在对等场景下,可能期望两个节点都彼此认证。

[0241] 方法400还可以包括第二节点7基于第二节点主私钥(V_{1S})和生成器值(GV)来确定470第二节点第二私钥(V_{2S})。类似于第一节点3执行的步骤330,第二节点第二私钥(V_{2S})可以根据以下公式基于第二节点主私钥(V_{1S})和生成器值(GV)的标量加法:

[0242] $V_{2S}=V_{1S}+GV$(等式13.1)

[0243] $V_{2S}=V_{1S}+SHA\text{-}256(M)$(等式13.2)

[0244] 然后,第二节点7可以独立于第一节点3,基于以下公式,基于第二节点第二私钥(V_{2S})和第一节点第二公钥(P_{2C})确定480公共秘密(CS):

[0245] $S=V_{2S} \times P_{2C}$(等式14)

[0246] 由第一节点3确定的公共秘密(CS)与在第二节点7处确定的公共秘密(CS)相同。现在将描述等式11和等式14提供相同的公共秘密(CS)的数学证明。

[0247] 转向由第一节点3确定的公共秘密(CS),可以将等式10.1代入等式11,如下所示:

[0248] $S=V_{2C} \times P_{2S}$(等式11)

[0249] $S=V_{2C} \times (V_{2S} \times G))$

[0250] $S=(V_{2C} \times V_{2S}) \times G$(等式15)

[0251] 转向由第二节点7确定的公共秘密(CS),可以将等式12.1代入等式14,如下所示:

[0252] $S=V_{2S} \times P_{2C}$(等式14)

[0253] $S=V_{2S} \times (V_{2C} \times G)$

[0254] $S=(V_{2S} \times V_{2C}) \times G$(等式16)

[0255] 由于ECC代数是可交换的,因此等式15和等式16是等效的,因为:

[0256] $S=(V_{2C} \times V_{2S}) \times G=(V_{2S} \times V_{2C}) \times G$(等式17)

[0257] 现在,公共秘密(CS)可以用作秘密密钥,或者用作对称密钥算法中秘密密钥的基础,以用于第一节点3和第二节点7之间的安全通信。该通信可以用于传达私钥的一部分、私钥的表示形式或标识符,或私钥的助记符。因此,一旦已经在例如数字钱包或其他受控资源的建立期间使用了本发明,则此后可以在各方之间进行安全通信。

[0258] 共同秘密(CS)可以是椭圆曲线点的形式(x_S,y_S)。可以使用节点3、7同意的公开已知的标准操作将其转换为标准密钥格式。例如,x_S值可以是256位整数,可以用作AES_{256}加密的密钥。对于需要该长度密钥的任何应用,也可以使用RIPEMD160将其转换为160位整数。

[0259] 可以根据需要确定公共秘密(CS)。重要的是,第一节点3不需要存储公共秘密(CS),因为可以根据消息(M)重新确定公共秘密。在一些示例中,使用的消息(M)可以被存储在数据存储装13、17、19(或其他数据存储)中,而无需与主私钥所需的相同级别的安全性。在一些示例中,消息(M)可以是公开可用的。

[0260] 然而,根据某些应用,公共秘密(CS)可以存储在与第一节点相关联的第一数据存储(X)中,只要公共秘密(CS)被保持与第一节点主私钥(V_{1C})一样安全。

[0261] 应当注意,上述实施例说明而不是限制本发明,并且本领域技术人员将能够设计许多替代实施例而不脱离由所附权利要求书限定的本发明的范围。在权利要求书中,括号中的任何附图标记都不应解释为对权利要求的限制。单词“包括”和“包含”等不排除任何权利要求或整个说明书中列出的元素或步骤之外的元素或步骤的存在。在本说明书中,“包括”是指“包含或由……组成”。元素的单数形式并不排除此类元素的复数形式,反之亦然。本发明可以通过包括几个不同元件的硬件以及通过适当编程的计算机来实现。在列举几个器件的装置权利要求中,这些装置中的几个可以由一个且相同的硬件来实施。在互不相同的从属权利要求中记载某些手段的事实并不表示不能有利地使用这些手段的组合。

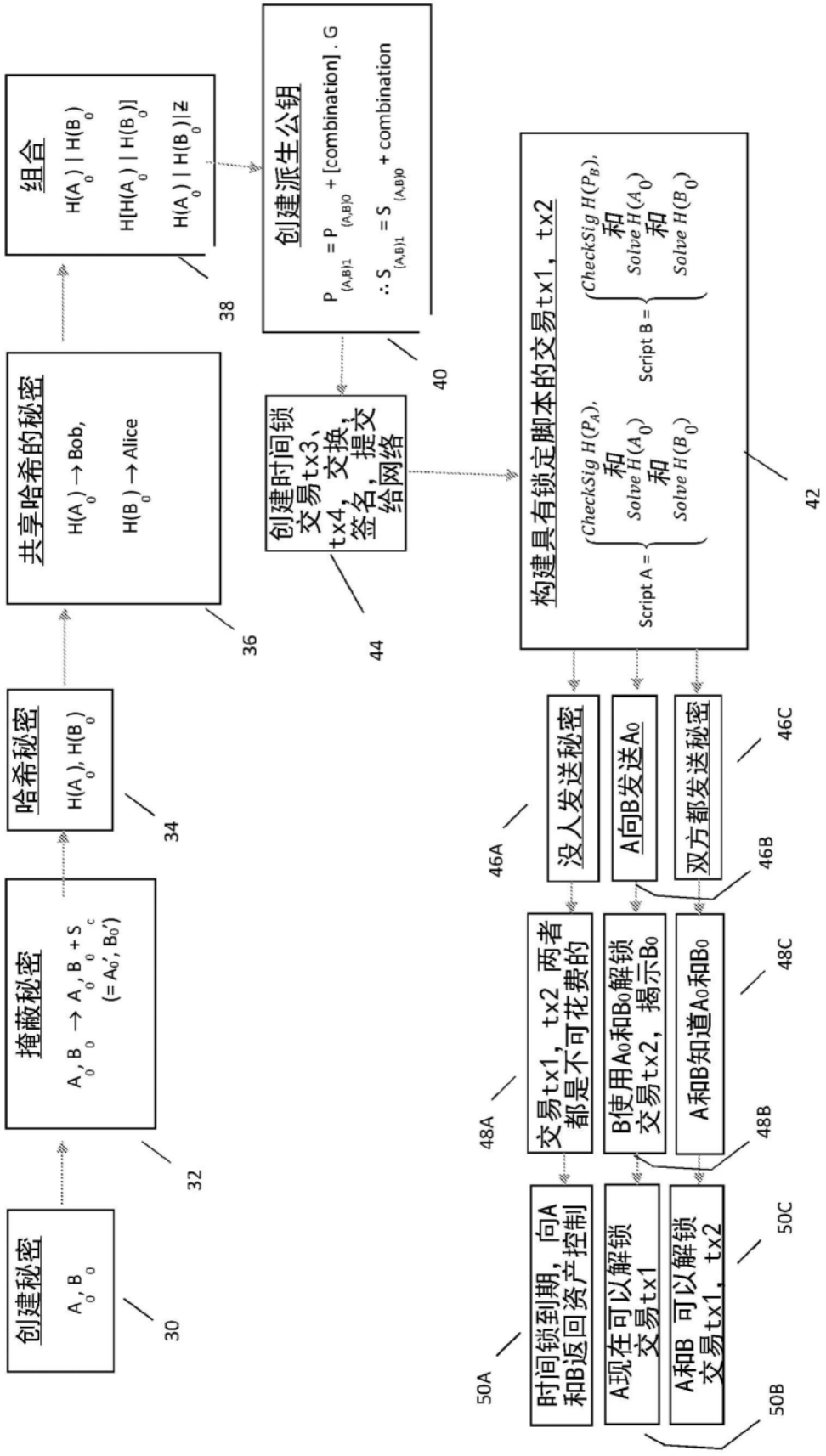

创建秘密
A_0, B_0
30
掩蔽秘密
$A_0, B_0 \rightarrow A_0, B_0 + S_c$
$(= A_0', B_0')$
32
哈希秘密
$H(A_0), H(B_0)$
34
共享哈希的秘密
$H(A_0) \rightarrow$ Bob,
$H(B_0) \rightarrow$ Alice
36
组合
$H(A_0) \mid H(B_0)$
$H[H(A_0) \mid H(B_0)]$
$H(A_0) \mid H(B_0) \mid z$
38
创建派生公钥
$P_{(A,B)1} = P_{(A,B)0} + [combination] \cdot G$
$\therefore S_{(A,B)1} = S_{(A,B)0} + combination$
40
创建时间锁交易tx3、tx4，交换，签名，提交给网络
44
构建具有锁定脚本的交易tx1，tx2
Script A = $\begin{cases} CheckSig\ H(P_A), \\ 和 \\ Solve\ H(A_0) \\ 和 \\ Solve\ H(B_0) \end{cases}$
Script B = $\begin{cases} CheckSig\ H(P_B), \\ 和 \\ Solve\ H(A_0) \\ 和 \\ Solve\ H(B_0) \end{cases}$
42
没人发送秘密
A向B发送A_0
双方都发送秘密
46A
46B
46C
交易tx1，tx2 两者都是不可花费的
B使用A_0和B_0解锁交易tx2，揭示B_0
A和B知道A_0和B_0
48A
48B
48C
时间锁到期，向A和B返回资产控制
A现在可以解锁交易tx1
A和B 可以解锁交易tx1，tx2
50A
50B
50C

图1

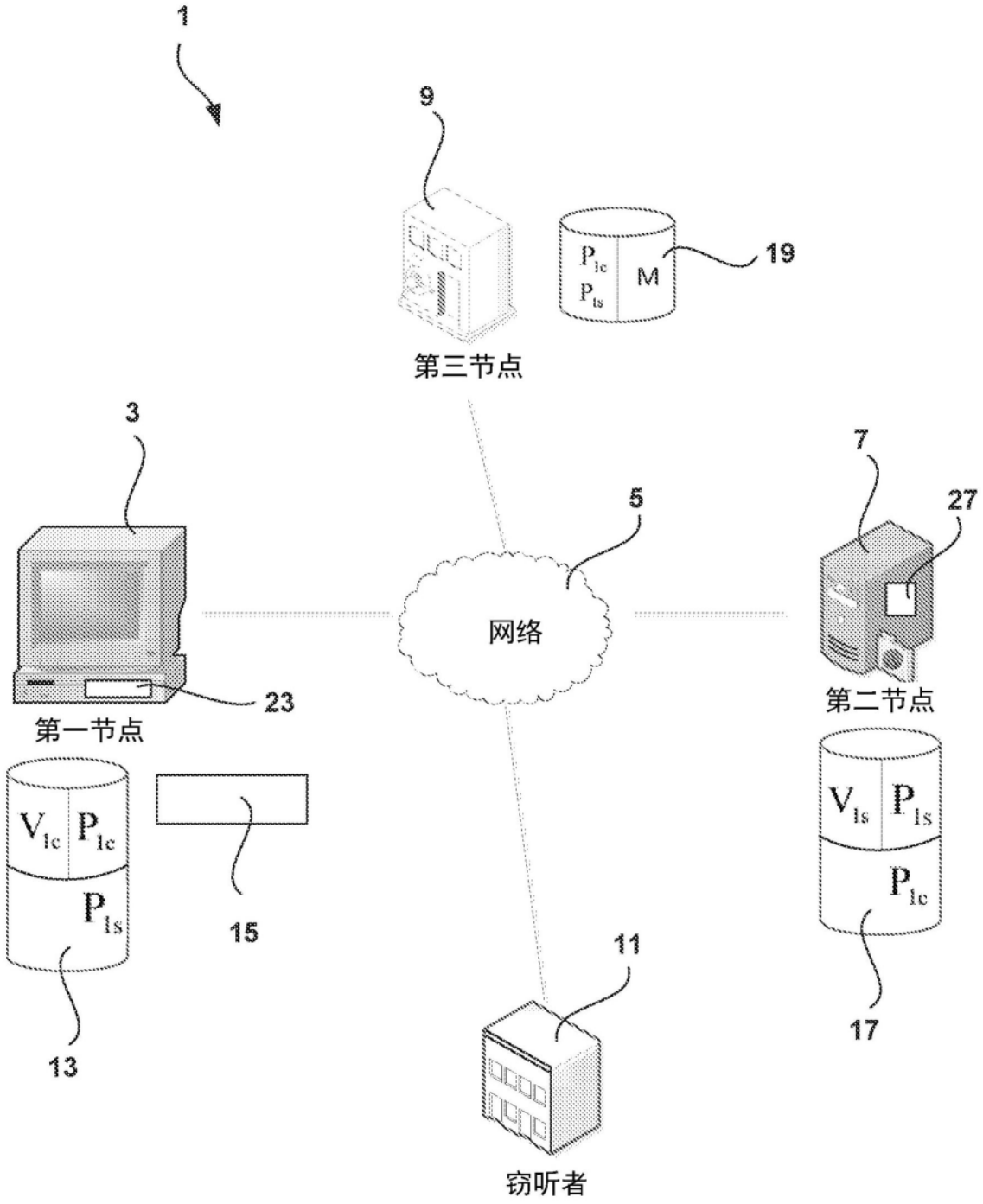

1
9
P_{1c}
P_{1s}
M
19
第三节点
3
5
7
27
网络
第二节点
23
第一节点
V_{1c}
P_{1c}
P_{1s}
15
V_{1s}
P_{1s}
P_{1c}
13
17
11
窃听者

图2

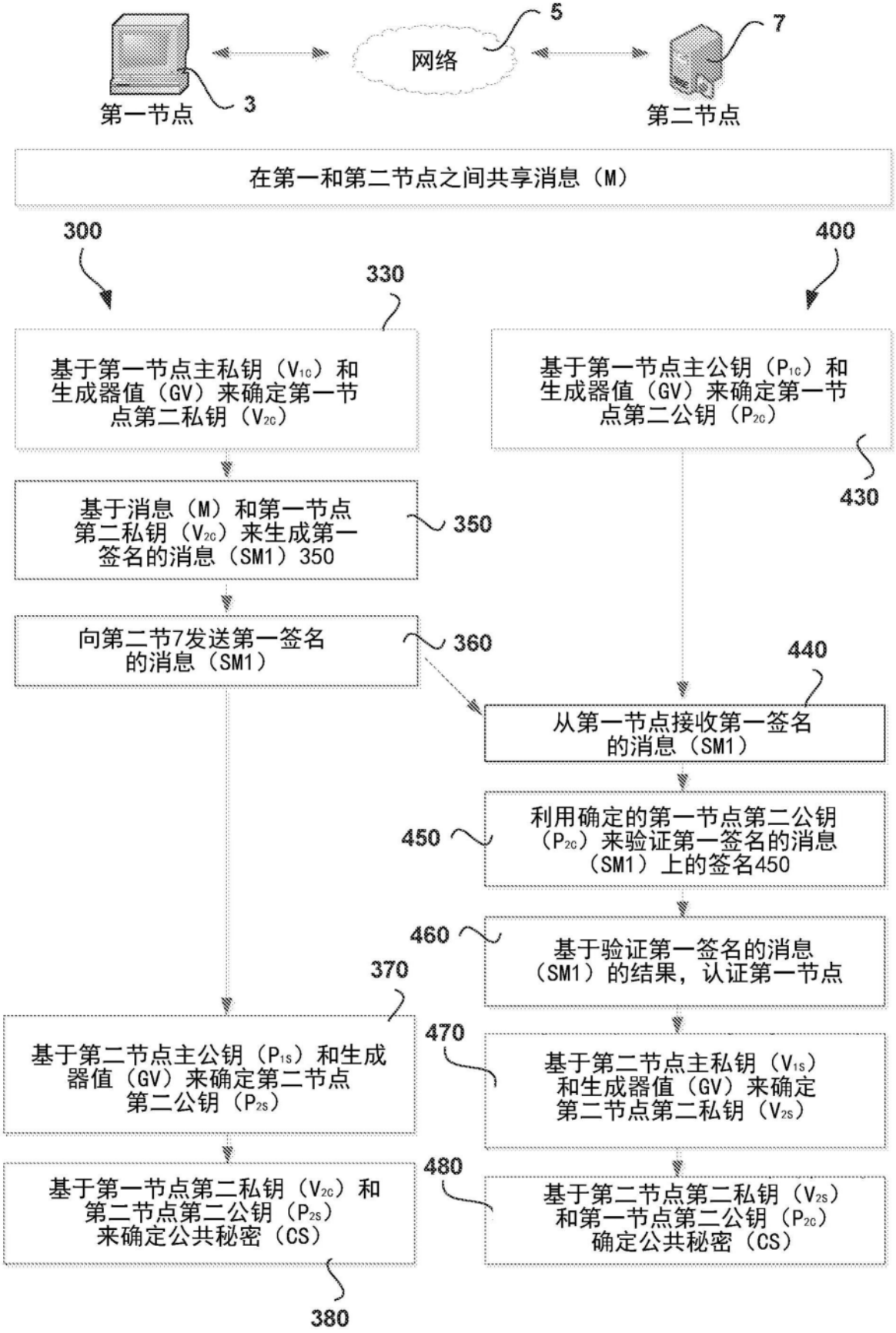

网络
5
7
第一节点
3
第二节点
在第一和第二节点之间共享消息（M）
300
400
330
基于第一节点主私钥（V1C）和生成器值（GV）来确定第一节点第二私钥（V2C）
基于第一节点主公钥（P1C）和生成器值（GV）来确定第一节点第二公钥（P2C）
430
基于消息（M）和第一节点第二私钥（V2C）来生成第一签名的消息（SM1）350
350
向第二节7发送第一签名的消息（SM1）
360
440
从第一节点接收第一签名的消息（SM1）
利用确定的第一节点第二公钥（P2C）来验证第一签名的消息（SM1）上的签名450
450
460
基于验证第一签名的消息（SM1）的结果，认证第一节点
370
基于第二节点主公钥（P1S）和生成器值（GV）来确定第二节点第二公钥（P2S）
470
基于第二节点主私钥（V1S）和生成器值（GV）来确定第二节点第二私钥（V2S）
基于第一节点第二私钥（V2C）和第二节点第二公钥（P2S）来确定公共秘密（CS）
480
基于第二节点第二私钥（V2S）和第一节点第二公钥（P2C）确定公共秘密（CS）
380

图3

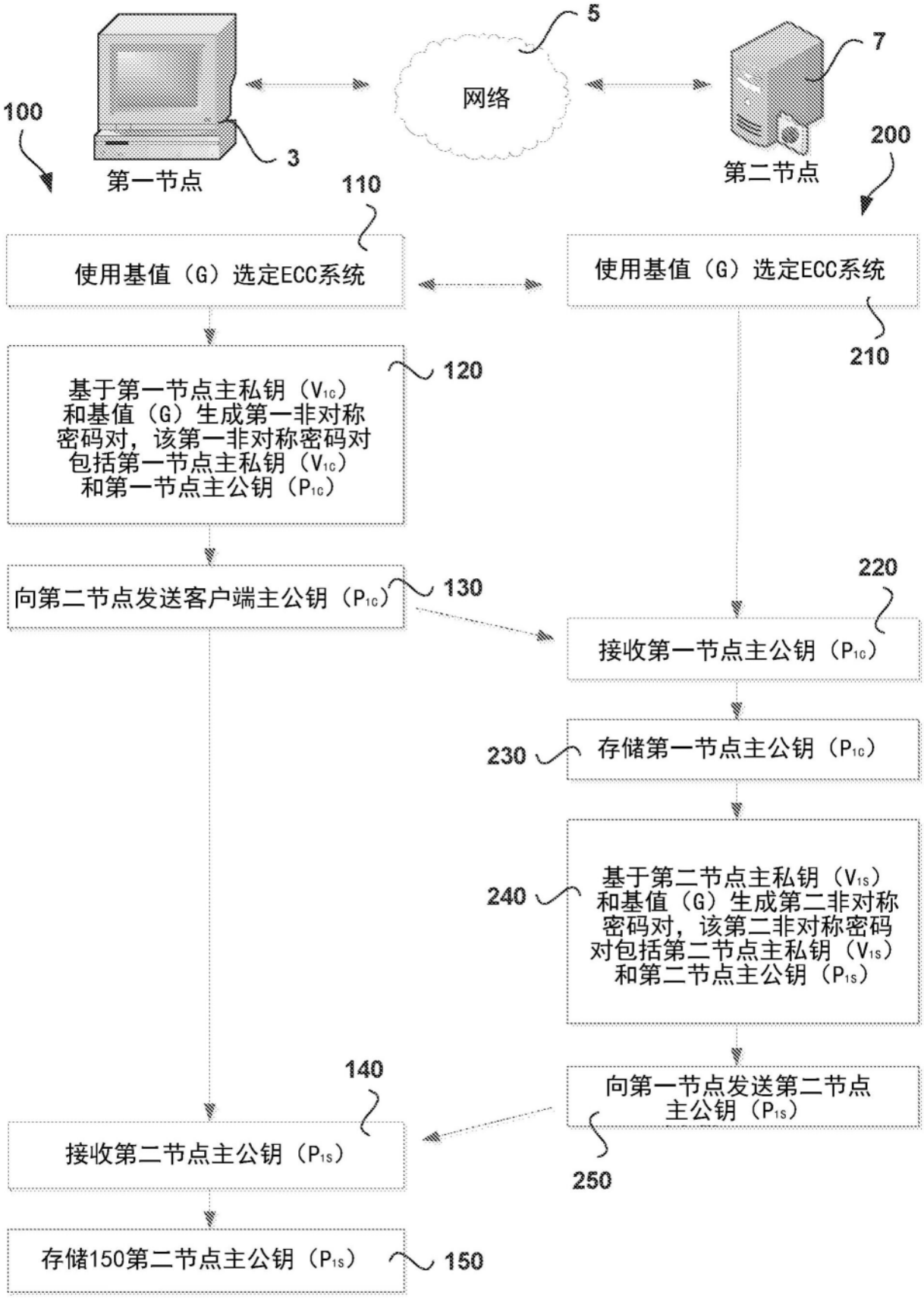

100
第一节点
3
5
网络
7
第二节点
200
110
使用基值（G）选定ECC系统
使用基值（G）选定ECC系统
210
120
基于第一节点主私钥（V1C）和基值（G）生成第一非对称密码对，该第一非对称密码对包括第一节点主私钥（V1C）和第一节点主公钥（P1C）
向第二节点发送客户端主公钥（P1C）
130
220
接收第一节点主公钥（P1C）
230
存储第一节点主公钥（P1C）
240
基于第二节点主私钥（V1S）和基值（G）生成第二非对称密码对，该第二非对称密码对包括第二节点主私钥（V1S）和第二节点主公钥（P1S）
140
向第一节点发送第二节点主公钥（P1S）
250
接收第二节点主公钥（P1S）
存储150第二节点主公钥（P1S）
150

图4

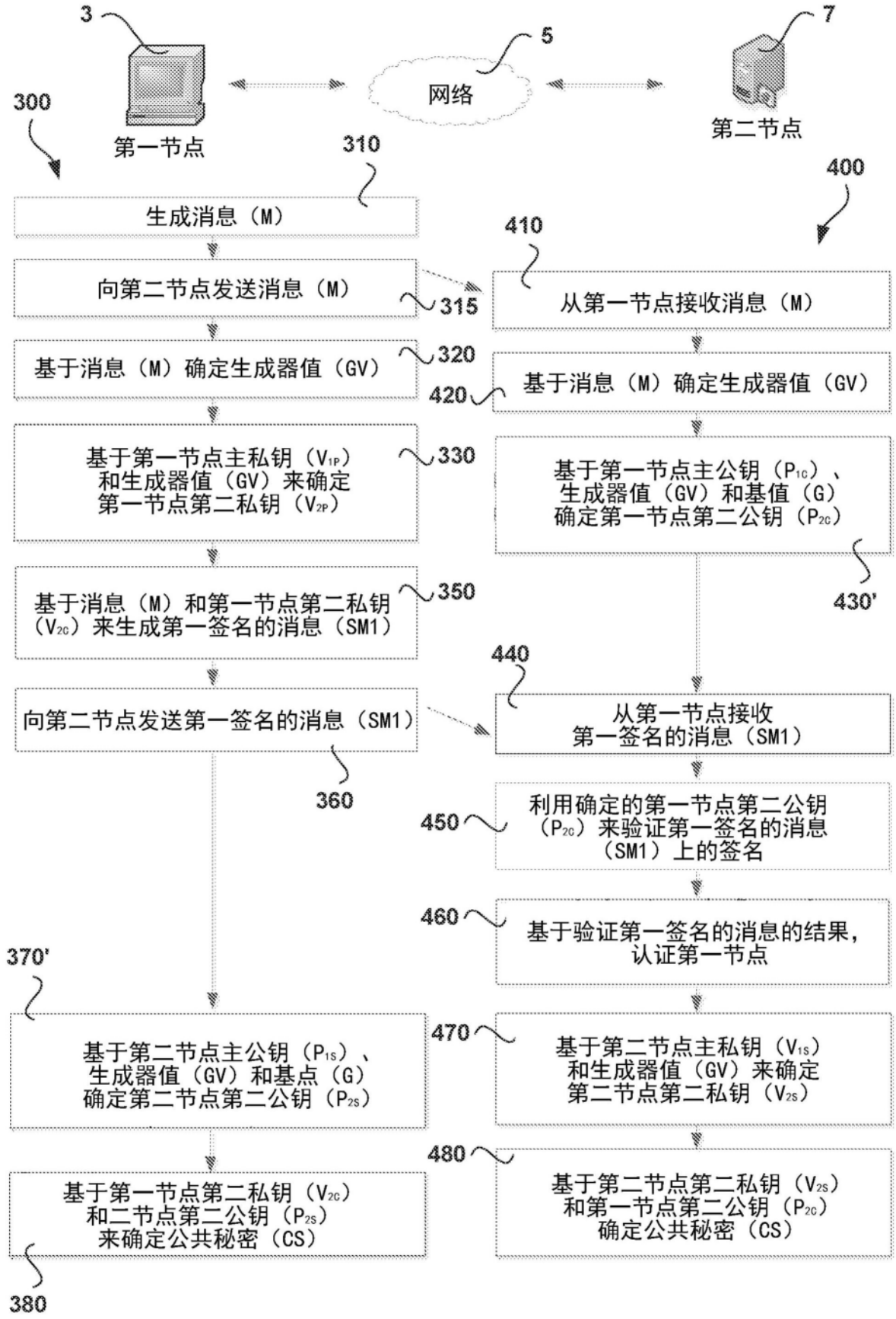

3
5
7
网络
第一节点
第二节点
300
310
400
生成消息（M）
410
向第二节点发送消息（M）
315
从第一节点接收消息（M）
基于消息（M）确定生成器值（GV）
320
420
基于消息（M）确定生成器值（GV）
基于第一节点主私钥（V1P）和生成器值（GV）来确定第一节点第二私钥（V2P）
330
基于第一节点主公钥（P1C）、生成器值（GV）和基值（G）确定第一节点第二公钥（P2C）
430'
基于消息（M）和第一节点第二私钥（V2C）来生成第一签名的消息（SM1）
350
440
向第二节点发送第一签名的消息（SM1）
从第一节点接收第一签名的消息（SM1）
360
利用确定的第一节点第二公钥（P2C）来验证第一签名的消息（SM1）上的签名
450
460
基于验证第一签名的消息的结果，认证第一节点
370'
基于第二节点主公钥（P1S）、生成器值（GV）和基点（G）确定第二节点第二公钥（P2S）
470
基于第二节点主私钥（V1S）和生成器值（GV）来确定第二节点第二私钥（V2S）
480
基于第一节点第二私钥（V2C）和二节点第二公钥（P2S）来确定公共秘密（CS）
基于第二节点第二私钥（V2S）和第一节点第二公钥（P2C）确定公共秘密（CS）
380

图5