(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 110709874 B
(45)授权公告日 2024.10.29

(21)申请号 201880037501.3
(22)申请日 2018.06.04
(65)同一申请的已公布的文献号
申请公布号 CN 110709874 A
(43)申请公布日 2020.01.17
(30)优先权数据
1709097.8 2017.06.07 GB
1709096.0 2017.06.07 GB
(85)PCT国际申请进入国家阶段日
2019.12.05
(86)PCT国际申请的申请数据
PCT/IB2018/053981 2018.06.04
(87)PCT国际申请的公布数据
WO2018/224945 EN 2018.12.13
(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

(72)发明人 朱塞佩·德泰法尼
西蒙尼·马蒂奥
帕特里克·莫蒂林斯基
史蒂芬·文森特

(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.
G06Q 20/06(2006.01)
H04L 9/08(2006.01)
G06Q 20/40(2006.01)
G06Q 40/04(2006.01)
H04L 9/30(2006.01)

(56)对比文件
CN 101711027 A,2010.05.19
CN 101796860 A,2010.08.04

审查员 王列珂

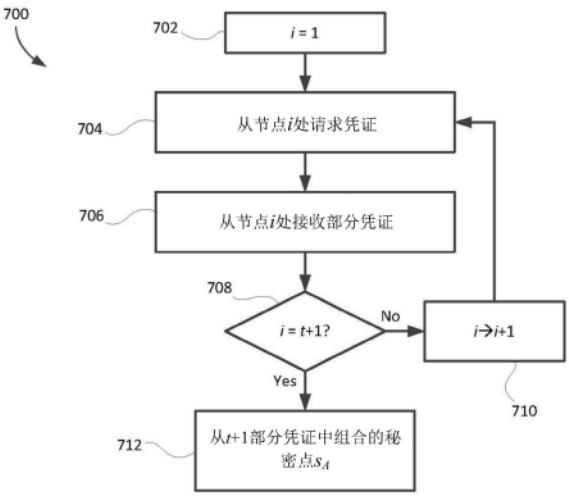
权利要求书2页 说明书10页 附图7页

(54)发明名称

用于区块链网络的凭证生成与分发方法和系统

(57)摘要

用于管理从自治专用节点组到请求节点的凭证的安全分发的方法和设备。凭证的安全分发可以使用秘密份额和任何节点都不重建或拥有的组私钥。该凭证包括用于请求节点的标识符和秘密点,该秘密点是节点从多个专用节点中的每个提供的秘密点的部分中组合而成的,其中秘密点基于组私钥和请求节点的标识符的映射到点散列。



CN 110709874 B

1.一种由计算机实现的方法,所述方法用于节点从专用节点组获得凭证,每个专用节点具有组私钥的私钥份额,其中,所述方法包括:

从所述节点向多个所述专用节点发送对所述凭证的请求,其中,所述节点具有标识符;

从多个所述专用节点中的每个节点处接收秘密点的一部分,所述秘密点的一部分由所述节点从所述标识符和所述节点的私钥份额中生成;和

通过组合接收到的所述秘密点的部分来生成所述秘密点,其中,所述秘密点包括所述组私钥乘以所述标识符的映射到点散列,所述凭证包括所述标识符和所述秘密点,

以及其中,获得所述组私钥的完整副本不需要任何所述专用节点和所述节点。

2.根据权利要求1所述的方法,其中,发送所述请求包括向多个所述专用节点中的每个发送单独的请求。

3.根据权利要求1或权利要求2所述的方法,其中,所述标识符包括标识所述节点的标识符字符串和标识所述专用节点组角色的角色字符串。

4.根据权利要求3中所述的方法,其中,所述标识符还包括所述凭证的到期时间。

5.根据权利要求3所述的方法,其中,发送所述请求包括发送所述标识符字符串,并且其中所述接收包括接收所述标识符。

6.根据权利要求1所述的方法,其中,所述生成所述秘密点包括组合接收到的所述秘密点的部分。

7.根据权利要求1所述的方法,其中,所述生成所述秘密点是使用秘2/3

密份额来生成基于所述组私钥的所述秘密点。

8.根据权利要求7所述的方法,其中,所述秘密份额使用拉格朗日插值法和所述多个专用节点的私钥份额,而不重建所述组私钥。

9.一种节点,包括:

处理器;

存储器;

网络接口;和

应用,所述应用包含处理器可执行指令,用于从专用节点组获得凭证,专用节点中的每一个都具有组私钥的私钥份额,其中,当所述处理器可执行指令被执行时,所述处理器可执行指令使得所述节点:

向多个所述专用节点发送对所述凭证的请求,其中,所述节点具有标识符;

从多个所述专用节点中的每个节点处接收秘密点的一部分,所述秘密点的一部分由所述节点从所述标识符和所述节点的私钥份额中生成;和

通过组合接收到的所述秘密点的部分来生成所述秘密点,其中,所述秘密点包括所述组私钥乘以所述标识符的映射到点散列,所述凭证包括所述标识符和所述秘密点,

以及其中,获得所述组私钥的完整副本不需要任何所述专用节点和所述节点。

10.根据权利要求9所述的节点,其中,当所述处理器可执行指令被执行时,所述处理器可执行指令使得所述节点发送所述请求的方式是:向多个所述专用节点中的每一个发送单独的请求。

11.根据权利要求9或权利要求10所述的节点,其中,所述标识符包括标识所述节点的标识符字符串和标识所述专用节点组角色的角色字符串。

12.根据权利要求11所述的节点,其中,所述标识符还包括所述凭证的到期时间。

13.根据权利要求11所述的节点,其中,当所述处理器可执行指令被执行时,所述处理器可执行指令使得所述节点通过发送所述标识符字符串来发送请求,并通过接收所述标识符来进行接收。

14.根据权利要求9所述的节点,其中,当所述处理器可执行指令被执行时,所述处理器可执行指令使得所述节点通过组合接收到的所述秘密点的部分来生成所述秘密点。

15.根据权利要求9所述的节点,其中,当所述处理器可执行指令被执行时,所述处理器可执行指令使得所述节点生成所述秘密点的方式是:使用秘密份额来生成基于所述组私钥的所述秘密点。

16.根据权利要求15所述的节点,其中,所述秘密份额使用拉格朗日插值法和所述多个专用节点的私钥份额,而不重建所述组私钥。

17.一种非易失性处理器可读介质,存储有处理器可执行指令,所述处理器可执行指令由一个或多个处理器执行时使得所述一个或多个处理器执行根据权利要求1至8中任意一项所述的方法中的操作。

用于区块链网络的凭证生成与分发方法和系统

技术领域

[0001] 本发明总体上涉及区块链交易,具体涉及在节点之间建立可信点对点通信的方法和设备。

背景技术

[0002] 在本文档中,我们使用术语“区块链”来包括所有形式的电子的、基于计算机的分布式账本(distributed ledger),包括但不限于基于共识的区块链和交易链技术、许可和未许可的账本、共享账本及其变形。应指出的是,其他替代性的区块链实现方案和协议也落入本发明的范围内。

[0003] 区块链是基于共识的电子账本,其实现为基于计算机的去中心化、分布式系统,由区块构成,而这些区块进而由交易(transaction)构成。每个交易(Tx)都是对区块链系统中的参与者之间的数字资产的控制的转移进行编码的数据结构,并且该数据结构包括至少一个输入和至少一个输出。每个区块均包含前一区块的散列,这些区块被链接在一起,以创建一个永久的、不可更改的记录,该记录自区块链创建以来写入了该区块链的所有交易。交易包含嵌入其输入和输出中的称为脚本的小程序,这些程序指定了如何以及由谁可以访问交易的输出。这些脚本是使用基于堆栈的脚本语言来编写的。

[0004] 接收新交易的网络节点将迅速尝试将该交易推送到网络中的其他节点。在将新交易传输到其他节点之前,它会被“验证”,这意味着将根据一组标准对新交易进行检查,以确保该交易符合根据适用区块链协议进行适当交易的基本要求。

[0005] 为了将交易写入区块链,节点将该交易合并到一个区块中,该节点旨在收集交易并将这些交易形成区块。节点再尝试完成与该节点相关的“工作证明”。成功的节点将其已确认的区块添加到区块链中,并且该区块通过网络传播,以便维护区块链副本的其他节点可以更新其记录。接收区块的节点也“验证”该区块及其中的所有交易,以确保其符合协议的正式要求。

[0006] 去中心化点对点系统中的挑战之一是确保一个节点可以在可信的基础上与网络中的另一个节点进行通信。随着一些实施方式的网络架构的发展,一些节点可能承担更多的专用任务,而其他节点可能依赖那些专用的节点作为某些数据的来源或某些功能的执行者。如果一个节点要依赖另一个节点获取信息或作为合法来源,则它需要能够建立与该节点进行通信的可信关系。在节点可能具有不同角色的情况下,最好有一种确定和验证节点角色的机制。此外,如果一个节点被证明是非法的或恶意的,则其他节点应该可以记住该节点,从而忽略未来来自这样一个节点的通信。在点对点系统中,所面对的挑战是要在不损害系统的点对点性质的情况下,通过施加中央权限来解决这些问题。

发明内容

[0007] 现在已经设计出了这样的解决方案。

[0008] 因此,根据本发明,提供一种如所附权利要求中限定的方法及系统。

[0009] 本申请可描述和公开管理从自治专用节点组到请求节点的凭证的安全分发的方法和设备。凭证的安全分发可以使用秘密份额和任何节点都不重建或拥有的组私钥。该凭证包括用于请求节点的标识符和秘密点,该秘密点是节点从多个专用节点中的每个提供的秘密点的部分中组合而成的,其中秘密点基于组私钥和请求节点的标识符的映射到点散列(map-to-point hash)。在某些实施方式中,节点可以是区块链节点。

[0010] 本申请可进一步描述和公开两个节点在自治专用节点组的证明下彼此认证,而不涉及该组或不涉及去中心化证书管理器或认证器的方法和设备。该方法可包括第一节点和第二节点使用包含它们各自标识符和秘密点的双线性配对运算来导出相同的会话密钥。假设秘密点和标识符是通过使用组私钥从该组中获得的,那么双线性配对运算会使得在两个节点中的每个节点处生成相同的会话密钥,从而认证它们各自的凭证并实现两个节点之间的可信通信。

[0011] 附加地或替代地,本申请描述了一种计算机实现的方法,该方法用于节点从专用节点组获得凭证,每个专用节点具有组私钥的私钥份额。该方法可包括:从所述节点向多个专用节点发送对所述凭证的请求,其中,所述节点具有标识符;从所述多个专用节点中的每个节点处接收秘密点的一部分,所述秘密点的一部分由所述节点从所述标识符和所述节点的私钥份额中生成;通过组合接收到的所述秘密点的部分来生成所述秘密点,其中,所述秘密点为组私钥乘以标识符的映射到点散列,所述凭证是所述标识符和所述秘密点。要获得组私钥的完整副本可能不需要任何所述专用节点或所述节点。

[0012] 在一些实施方式中,发送请求包括向所述多个专用节点中的每个发送单独的请求。所述标识符可包括标识所述节点的标识符字符串和标识所述专用节点组角色的角色字符串。在一些情况下,所述标识符还包括所述凭证的到期时间,并且在一些其他情况下,发送请求包括发送所述标识符字符串,并且其中接收包括接收所述标识符。

[0013] 在一些实施方式中,生成所述秘密点包括组合接收到的秘密点的部分。在一些示例实施方式中,组合可包括求和。

[0014] 在一些实施方式中,所述秘密点是使用秘密份额并基于组私钥生成的。在某些情况下,秘密份额使用拉格朗日插值法和多个专用节点的私钥份额,而不重建组私钥。

[0015] 附加地或替代地,本申请可描述一种用于第一节点与第二节点建立可信通信的计算机实现的方法,所述第二节点具有第二节点标识符和第二秘密点,所述第二秘密点为组私钥与所述第二节点标识符的映射到点散列相乘,所述组私钥与用于授予凭证的节点组相关联。所述方法可包括:从所述节点组获取第一秘密点,其中,所述第一秘密点为组私钥与第一节点标识符的映射到点散列相乘;将所述第一节点标识符发送给所述第二节点;接收所述第二节点标识符;使用双线性配对运算、所述第二节点标识符的映射到点散列、和所述第一秘密点一起来生成第一会话密钥;确认所述第一会话密钥与第二会话密钥相匹配,所述第二会话密钥由所述第二节点使用双线性配对运算、所述第二秘密点、和所述第一节点标识符的映射到点散列一起生成。

[0016] 在一些实施方式中,所述通过双线性配对运算生成所述第一会话密钥由以下表达式之一来表示:

[0017] $K_A = e(H_1(id_B), s_A)$,和

[0018] $K_A = e(s_B, H_1(id_A))$,

[0019] 通过双线性配对运算生成所述第二会话密钥由所述表达式中的另一个来表示,并且其中e()是双线性配对运算,H_1()是映射到点散列,id_A和id_B分别是所述第一节点标识符和所述第二节点标识符中的一个,s_A和s_B分别是所述第一秘密点和所述第二秘密点中的一个。

[0020] 在一些实施方式中,获得所述第一秘密点包括从所述节点组中的多个节点中的每一个获得所述第一秘密点的各个部分,并且组合所述各个部分来形成所述第一秘密点,而不重组所述组私钥。

[0021] 在一些实施方式中,所述确认包括:将用所述第一会话密钥加密的质询(challenge)从所述第一节点发送到所述第二节点,接收对所述质询的响应,并基于所述响应,确定所述第二节点使用所述第二会话密钥有效地解密了所述质询。

[0022] 在一些实施方式中,所述发送还包括发送第一随机数,并且其中,所述接收还包括接收第二随机数和计算的C_0值,其中,所述C_0值包括所述第二会话密钥、所述第一随机数、和所述第二随机数的级联的散列。在这些情况的某些中,所述级联还包括所述第一节点标识符和所述第二节点标识符。在这些情形下的某些中,所述生成包括生成计算的C_1值,所述C_1值包括所述第一会话密钥、所述第一随机数、和所述第二随机数的级联的散列,并且其中,所述确认包括确认所述计算的C_0值与所述计算的C_1值相匹配。

[0023] 在一些实施方式中,所述第二秘密点为所述组私钥乘以所述第二节点标识符的映射到点散列。

[0024] 在一些实施方式中,所述第一秘密点和所述第二秘密点各自被所述节点组使用秘密共享分别提供给所述第一节点和第二节点。

[0025] 附加地或替代地,本申请可描述一种节点,包括一个或多个处理器,存储器,网络接口以及包括处理器可执行指令的应用,这些指令在被执行时使得所述节点执行本文所述方法中一种或多种方法的操作。

[0026] 附加地或替代地,本申请可描述一种存储处理器可执行指令的非易失性处理器可读介质,所述处理器可执行指令在由一个或多个处理器执行时使得所述一个或多个处理器执行本文所述方法中一种或多种方法的操作。

附图说明

[0027] 参考本文描述的实施例,本发明的这些方面和其他方面将变得清楚并得以阐明。现在将仅通过举例的方式并参考附图来描述本发明的实施例,其中:

[0028] 图1示出了区块链节点的示例网络;

[0029] 图2示意性地示出了从专用区块链节点组或专用区块链节点网络处请求凭证的区块链节点;

[0030] 图3示意性地示出了区块链节点从所述专用区块链节点组中获得部分凭证的示例实施方式;

[0031] 图4以流程图形式示出了从区块链节点组获得凭证的示例过程;

[0032] 图5示出了两个区块链节点彼此认证从专用区块链节点组获得的凭证的示例过程的示意图;

[0033] 图6示出了两个区块链节点彼此认证从专用区块链节点组获得的凭证的另一示例

过程的示意图；

[0034] 图7示出了示例区块链节点的简化框图。

具体实施方式

[0035] 在本申请中，术语“和/或”旨在涵盖所列元素的所有可能的组合和子组合，包括任何一个单独列出的元素、任何子组合或所有元素，并且不一定排除附加元素。

[0036] 在本申请中，短语“……或……中的至少一个”旨在覆盖任何一个或多个所列元素，包括任何一个单独列出的元素、任何子组合、或所有元素，不一定排除任何其他元素，并且不一定要求具有所有元素。

[0037] 尽管以下描述的示例参考了在区块链网络的上下文中的实施方式，但是应当理解的是，本申请不必限于区块链节点或在区块链网络中的实施方式。

[0038] 首先参考图1，图1以框图形式示出了与区块链相关联的示例网络，在此可以称为区块链网络100。区块链网络100是一个点对点的开放式成员网络，任何人都可以加入，无需邀请或无需经其他成员同意。运行区块链协议(区块链网络100在区块链协议下运行)实例的分布式电子设备可以参与区块链网络100。这种分布式电子设备可以称为节点102。

[0039] 运行区块链协议并形成区块链网络100的节点102的电子设备可以是各种类型的，包括例如计算机(如台式计算机、笔记本电脑、平板电脑、服务器)、移动设备(如智能手机)、可穿戴计算机(如智能手表)、或其他电子设备。

[0040] 区块链网络100的节点102使用合适的通信技术彼此连接，该通信技术可以包括有线和无线通信技术。在许多情况下，区块链网络100至少部分地由互联网实现，并且一些节点102可以位于地理上分散的位置。

[0041] 节点102维护区块链上所有交易的全局账本，按组分为不同的区块，每个区块包含链中先前区块的散列。全局账本是分布式账本，每个节点102可以存储全局账本的完整副本或部分副本。通过节点102影响全局账本的交易由其他节点102验证，从而保持全局账本的有效性。本领域的普通技术人员将会理解实现和操作区块链网络的细节。

[0042] 每个交易通常具有一个或多个输入和一个或多个输出。嵌入到输入和输出中的脚本指定了如何以及由谁可以访问交易的输出。交易的输出可以是作为交易结果的值被转移到的地址。然后，该值与该输出地址相关联，作为未花费的交易输出。随后的交易可以将该地址作为输入，以花费或分散该值。

[0043] 节点102可以根据其功能而具有不同的类型或类别。

[0044] 在区块链的当前实施方式中，每个接收区块的节点首先验证该区块，然后将该区块发送到其他节点。验证区块所需的时间减慢了区块在网络中的传播速度。注意，区块链的一些实施方式，包括现有协议的演进，可以仅由节点的子集而不是网络中的每个节点来提供区块验证。然而，大多数节点上的区块验证仍然可能是任何区块链实施方式的特征，以防止无效区块通过网络传播。

[0045] 验证区块涉及确认该区块符合适用区块链协议设定的规定标准。除了确认区块本身符合规定的标准外，还可以评估区块内的每个交易以符合交易级标准。

[0046] 通常，交易级验证标准是交易在适用的区块链协议下被视为有效必须具备的规定特征。同样地，区块级验证标准是区块在适用的区块链协议下被视为有效必须具备的规定

特征。

[0047] 区块链的某些演变可以为更多的专用节点提供服务。例如,可能存在节点,其被构造为具有大量快速访问存储器和大量网络通信资源的存储节点。存储节点可以实现存储未确认交易的分布式交易数据库(“内存池”)。在一些实施方式中,存储节点可以为未花费的交易输出实现分布式未花费的交易输出数据库。在某些情况下,可以通过分布式散列表的方式来实现这些分布式数据库。另一个示例是,一些节点可以被构造为确认器节点,其被设计为着重于交易的快速验证和传播。确认器节点和存储节点可以通过在区块链网络100内形成快速交易商网络,来促进与大规模支付处理相关的交易量的增加,该快速交易商网络被设计用于分布式大容量存储以及快速验证和传播交易。

[0048] 在支付处理中使用的交易的数量的显着增长,将增加存储在每个内存池中交易的量(volume)。假设交易(TX)的平均大小为500字节,交易速率为~104TX/s,则该内存池可接收~400GB的每日传入数据。所有这些数据都需要在未确认交易的内存池中存储不同的时间。因此,在一些实施方式中,节点网络可以使用通过分布式散列表(DHT)实现的共享内存池,以便存储未确认交易池,而不是在每个节点处存储完整的内存池。在下面的一些示例中,术语“专用节点”可用于指代在区块链网络中具有特定角色的节点,例如确认器节点,存储节点等。

[0049] 在专用节点自身中,它们可能需要认证其中一个专用节点的身份以及该节点参与协作工作的权限。在许多网络架构中,此功能是通过实施中央认证服务(例如证书颁发机构等)来实现的,该中央认证服务对凭证进行集中管理并对参与节点进行认证。但是这种架构在分布式点对点架构中是不受欢迎的,因为在分布式对等架构中应避免中央控制或中央机构。

[0050] 随着区块链网络架构发展以适应具有特殊角色的节点(无论是用于验证、存储还是其他功能),在区块链网络的其他部分也可能出现相同的问题。这些节点充当具有特定角色的组的成员,并且具有该角色的组的其他成员希望能够标识和认证该组中的成员身份。同样,不是该组成员但与该组专用节点交互并需要“信任”该组专用节点的节点希望能够对该节点在组中的成员身份进行认证。因此将期望使节点能够认证身份和成员身份,无需在去中心化的区块链网络上施加中心化证书颁发机构。

[0051] 一方面,本申请提供一种用于依赖于节点组之间的分布式私钥份额来认证节点的方法和设备,该节点组为专用节点组的成员。因此,通过组中的专用节点的协作生成凭证并将其提供给请求节点,这样,任何一个专用节点都不能单独向请求节点提供有效凭证。

[0052] 另一方面,本申请提供一种通过确认每个节点都拥有由同一组节点授予的凭证来在两个节点之间建立可信通信信道的方法。可信通信通道中涉及的节点可以都是授予凭证的组的成员,但不一定是成员。在一个示例中,一个节点是专用节点组的成员,而另一个节点不是。对两个节点的凭证的有效性进行的确认,可以基于每个节点使用双线性配对运算生成密钥,确保密钥匹配来进行。用于该个节点的双线性配对运算可包括另一节点的标识符的映射到点散列和由该节点组协作地生成并提供给该个节点的秘密点,该映射到点散列和该秘密点作为自变量。

[0053] 现参考图2,图2示意性地示出了专用节点组502。请求节点504希望从该专用节点组502获得凭证,以便能够证明其作为专用节点之一或作为与专用节点之一通信的节点的

身份。请求节点504可以是或可以不是该专用节点组502的一部分。

[0054] 请求节点504将对于凭证的请求506发送到节点组502。该请求506与请求节点504的标识符相关联。该标识符可以由请求节点504生成,并且在一些实施方式中与请求506一起发送。例如,请求节点504可以生成其标识符“Alice”,并告诉节点组502其身份(id)为Alice。在一些实施方式中,标识符可以由节点组502来分配,或更具体地的说,由节点组502中的节点之一来分配,并且传达给请求节点504。在一些实施方式中,请求节点504向节点组502提供其“名称”或标识符字符串例如“Alice”,该名称或标识符字符串可以是任何字母数字字符串或其他唯一标识符。节点组502可以将附加信息级联到标识符字符串,以生成请求节点504的标识符。例如,节点组502可以将标识符字符串与节点组502的名称或标识符相结合。在一些实施方式中,名称是节点组502的“角色”或功能,例如“确认器”等。在一些实施方式中,节点组502可以将标识符字符串与凭证的到期时间相结合,这样评估请求节点504凭证的任何其他节点都可以根据其标识符评估凭证是否仍然未到期。在下文示出的一个示例实施方式中,请求节点504的标识符由标识符字符串(例如,Alice)与凭证的到期时间的级联形成,该标识符字符串为识别节点组502或节点的功能角色的角色字符串。标识符的其他可能的实施方式将被本领域普通技术人员理解。

[0055] 节点组502中的每个节点包括私钥k的份额k_i。秘密共享用在阈值密码系统中,在该阈值密码系统中,秘密k在n个用户之间分配,因此至少需要t+1个参与者进行协作才能重建k。已知秘密k中的任何t个都无法对剩下的一个进行确定。

[0056] 秘密共享是基于多项式插值的,并且假定秘密是有限域F的元素。该方案包括一个交易者dealer(也存在无交易者dealerless版本),一组n个参与者$U_1,\ldots,U_n$。在该协议中,任意随机秘密作为f(0)存储在t次多项式f(x)中,只有节点i才能计算其份额$f(x_i)$。如果n个节点中的t+1个协作,则它们可以重构f(x)上的任何点,密钥k的份额$k_1,k_2,\ldots,k_n$对应于使用拉格朗日多项式插值法的$f(x_1),f(x_2),\ldots,f(x_n)$。拉格朗日多项式插值法提供了可以用t+1个点重建t次函数f(x):

[0057] $p=\{(x_1,f(x_1)),(x_2,f(x_2)),\ldots,(x_{t+1},f(x_{t+1}))\}$

[0058] $$f(x)=\sum_{i\in p}f(x_i)\prod_{j\in p,j\neq i}\frac{x-x_j}{x_i-x_j}=\sum_{i\in p}f(x_i)b_{i,p}(x)$$

[0059] 其中,

[0060] $$b_{i,p}(x)=\prod_{j\in p,j\neq i}\frac{x-x_j}{x_i-x_j}$$

[0061] 注意:$b_{i,p}(x_i)=1$和$b_{i,p}(x_j)=0$。在此,b_i是插值系数。

[0062] 在涉及无交易者份额分配的实施方式中:

[0063] 1.为每个节点i分配所有人都知道的x_i。每个x_i都须是唯一的。

[0064] 2.每个节点i生成t次随机多项式$f_i(x)$。

[0065] 3.每个节点i向每隔一个节点秘密发送它们各自在多项式$f_i(x_j)$mod n上的点。

[0066] 4.每个节点i将所有接收到的$f_1(x_i),f_2(x_i),\ldots f_p(x_i)$以及所有mod n相加(其中n是基域$F_n$的特性($F_n$也可以表示为GF(n),Z/nZ))以形成$k_i=f(x_i)$mod n,其是多项式f(x) mod n的份额。

[0067] 再参考图2的示例,节点组502中的节点协作来为请求节点504生成秘密点s_A,如附图标记508所示。秘密点s_A在循环群G_1中,由以下有关握手和配对的解释中可以清楚看出其重要性。秘密点与标识符相关,符合以下关系式:

[0068] $s_A = k \cdot H_1(id_A)$

[0069] 其中,id_A是请求节点504的标识符。在某些实施方式中,可以将凭证当做(id_A, s_A)。H_1是一个映射到点的散列函数,下文将对此进行更详细的说明。

[0070] 为了在不重建组私钥k的情况下生成秘密点,节点组502使用秘密共享加入(Secret Share Joining)的形式协作生成s_A。图3示出了节点组502向请求节点504授予凭证的另一示例。在该图示中,请求节点504将凭证请求发送到节点组502的至少t+1个节点。这些节点中的每个节点向请求节点504提供s_A的一部分。特别地,秘密点s_A的每个份额s_{Ai}由节点i确定并发送到请求节点504,然后请求节点504将它们组装(即,将它们组合)到秘密点s_A中。在一些示例中,份额s_{Ai}的组合是通过将份额s_{Ai}相加以获得秘密点s_A来实现的。

[0071] 现参考图4,其以流程图形式示出了用于基于组的分布式凭证生成的一个示例过程700。过程700由请求节点执行。首先,在操作702中请求节点将索引i设置为1。然后,请求节点从节点i处请求凭证。节点i是节点组中的专用节点,该节点组用于执行本文描述的凭证授予过程的实现。节点i使用组私钥份额k_i的一部分来确定秘密点s_A的第i部分。该部分可以称为s_{Ai}。

[0072] 在操作706中,请求节点从第i个节点处接收该部分s_{Ai},即,接收部分凭证。在操作708中,接收节点评估索引i是否为t+1。如果是,则接收节点已接收到t+1部分凭证,但是如果没有,则仍需要其他部分凭证来重建秘密点s_A。如果i还不等于t+1,则在操作710中将其增加1,并且过程700返回到操作704,以从节点组中的另一节点处请求另一部分凭证。如果i等于t+1,则在操作712中,请求节点组装秘密点s_A。

[0073] 通过节点组中的节点的共同努力获得凭证后,最好有一个机制来检查或确认凭证而不依赖于中央机构或返回到该组进行确认/认证。这样,两个声称拥有来自同一节点组的有效凭证的节点可以确认彼此的凭证,而无需承担时间延迟和通信开销的负担,这些通信开销与使节点组协作验证和确认所发出的凭证相关。

[0074] 作为概述,可以认为组私钥k是有限域的一个成员,即$k \in Z_q$。凭证在获取它们的请求节点中以其标识符id_A和循环群G_1中的秘密点s_A的形式存在。在该示例实施例中,标识符(或"假名")id_A是(Alice||role||expirationtime)其中||表示两个字符串的级联,而role是与节点组和/或其功能或角色相关联的字符串或其他字母数字标识符。请求节点可以向任何人透露其标识符,但会将s_A保密。

[0075] 另一请求节点将标识符id_B获得为(Bob||role||expirationtime)和秘密点s_B。

[0076] 注意,秘密点s_A由$s_A = k \cdot H_1(id_A)$获得,且秘密点s_B由$s_B = k \cdot H_1(id_B)$获得。

[0077] 当Alice和Bob希望彼此建立可信通信时,即他们中的一个或两个都希望验证彼此的凭证时,两个节点交换各自的标识符/假名而不公开各自的秘密点。

[0078] 然后,节点Alice通过以下计算生成会话密钥K_A:

[0079] $K_A = e(H_1(Bob||role||expirationtime), s_A)$

[0080] 其中,e()是双线性映射,即配对运算,节点Bob通过以下计算来计算会话密钥K_B:

[0081] $K_B = e(s_B, H_1(Alice||role||expirationtime))$

[0082] 运算$H_1()$是映射到点函数,下文将进一步说明。由于配对的性质,上述两个密钥是相同的,即$K_A=K_B$。这两个节点可以通过发送质询值并获得对质询值的响应来验证协议的成功,即这两个节点各自持有由同一节点组发出的指示相同角色的凭证。或者,一方可以向另一方发送一些内容,当且仅当握手成功时,该内容才会被成功解密。

[0083] 该示例中的配对运算在这种情况下依赖于s_A和s_B来自相同循环群G_1。在其他情况下,s_A和s_B可能来自不同的循环群G_1和G_2,G_1和G_2作为两个阿贝尔群相互关联,其在另一个阿贝尔群的取值为:

[0084] $e:G_1\times G_2\rightarrow G_T$

[0085] 其中G_1、G_2、和G_T是顺序相同的循环群。

[0086] 凭证的生成中有一些参数是公共的,这些参数可以由该过程中的任何节点生成,并与其他节点共享。假设E是在域F_q上包含n个点的椭圆曲线,其中q是相对于2和3互质的质数的幂,则公共参数可包括以下内容:

[0087] $params=(p,G_1,G_2,G_T,g_1,g_2,e,H_1,H_2)$

[0088] 其中,$g_1(resp.g_2)$是具有双线性映射e(.,.)的质数阶p的群$G_1(resp.G_2)$的生成元。G_1是阶为r的$E(F_q)$的循环子群,其中r,q互质。并且,$e(g_1,g_2)$生成G_T,G_T的阶也为p。H_1和H_2分别是相对于G_1和G_2的映射到点散列。如上所述,在一些实施方式中,并且在本文描述的多个示例中,可能仅使用G_1,这意味着对秘密点s_A和s_B使用相同的映射到点的散列函数H_1。

[0089] 在某些椭圆曲线密码系统中,散列算法用于将密码或其他字符串(其为有限字段的元素)映射到给定椭圆曲线的点上,这些都是映射到点的散列。更确切地说,映射到点的散列函数H_1是一种转换,该转换接受消息m作为输入并返回点$P\in E(K)$。更具体地说,H_1返回点$P\in E(K)[r]$,其中$E(K)[r]$是由G_1生成的E(K)的子群。在该方案中,存在从基域$\mathbb{F}_q$到曲线的一对一映射f。这使得可以使用f(H(m))进行散列,其中H是经典散列函数,且$H(m)\in F_q$。

[0090] 现参考图5,图5示出了两个节点A和B之间通过验证彼此的凭证来建立可信通信通道时的消息流。在该示例实施方式中,节点A将其标识符id_A提供给节点B。标识符id_A是公共可用的,并且在某些情况下可以由节点B从另一个来源获得。据此,节点B能够使用节点A的标识符、节点B所持有的秘密点s_B、以及映射到点散列函数H_1(映射到点散列函数H_1由发布凭证的节点组规定)来生成会话密钥K_B。会话密钥K_B是使用配对运算e()生成的,配对运算e()也由节点组规定:

[0091] $K_B=e(s_B,H_1(Alice||role||expirationtime))$

[0092] 其中,在此示例中,节点B的标识符id_B为(Alice||role||expirationtime)。

[0093] 节点B将其标识符id_B提供给节点A,然后节点A能够使用节点B的标识符、其秘密点s_A、以及以下同样的配对运算和映射到点函数,类似地生成会话密钥K_A:

[0094] $K_A=e(H_1(Bob||role||expirationtime),s_A)$

[0095] 如果秘密点是由节点组使用相同的组私钥k以及节点A和B各自的标识符合法协作生成的,则配对运算应得出$K_A=K_B$。这可以是以任意多种方式进行的测试。在此示例实施方式中,节点A向通过会话密钥K_A加密的节点B发送质询。节点B尝试使用其会话密钥K_B解密质询并发送对该质询的响应。该响应可以由会话密钥K_B加密。在此基础上,两个节点都可以确保具有相同的会话密钥。该会话密钥可以用于加密两个节点之间的通信。在另一种实施方式中,会话密钥可用于加密致使建立另一密钥的通信。在又一实施方式中,两个节点之间的

通信仅依赖于上述过程来验证/认证彼此的凭证,并且两个节点之间的通信要么未加密,要么使用节点通常的公钥-私钥对加密。

[0096] 图6的信号示意图示出了另一示例实施方式。在此示例中,节点A首先生成任意随机数$nonce_A$。节点A将其标识符id_A及其随机数发送到节点B。节点B生成自己的随机数$nonce_B$。然后,在此示例中,节点B生成值C_0,该值是配对运算和其他数据(包括随机数)的结果的散列。在此示例中,值C_0可由下式给出:

[0097] $C_0=H(e(s_B,H_1(id_A))||id_A||id_B||nonce_A||nonce_B)$

[0098] 其中H是从字符串到字符串的防冲突散列函数。在一示例中,H为SHA-256。

[0099] 然后节点B答复节点A,向节点A提供其标识符id_B、其随机数$nonce_B$以及值C_0。然后,节点A执行类似的计算以获得值C_1,在此示例中,该值表示为:

[0100] $C_1=H(e(H_1(id_B),s_A)||id_A||id_B||nonce_A||nonce_B)$

[0101] 可以注意到,C_1和C_0的表达式中自变量的唯一区别在于配对运算。因此,如果双线性配对运算确认节点A和节点B的凭证是从同一节点组获得的,则两个值应匹配。节点A可以验证它们是否匹配,如果匹配,则将C_1发送到节点B,节点B也要验证它们是否匹配。在一个实施方式中,值$C_0=C_1$被用作会话密钥,用以加密节点A和节点B之间的通信。在一些实施方式中,节点仅依赖于它们通常的公钥-私钥对来加密通信,以上所描述的操作用于认证而不是用于建立会话密钥。

[0102] 在一些示例实施方式中,被级联并散列以形成值C_0和C_1的数据或字符串可以包括其他数据或字符串,或者可以不包括上述示例字符串中的一些。例如,在一个实施方式中,值C_0和C_1可以不包括标识符id_A和id_B的级联。虽然如此,但是可以理解的是,散列和随机数有助于保护通信免受中间人攻击。

[0103] 现参考图7,图7以框图形式示出了区块链节点1000的简化示例。该示例中的区块链节点1000包括处理器1002、网络接口1004和存储器1006。区块链节点1000可以通过使用任何合适的计算硬件来实现,这些计算硬件具有网络连接性以及足够的处理和存储资源来执行本文描述的功能。区块链节点1000可包括处理器可执行指令,以实现本文描述的功能。在一些情况下,处理器可执行指令可以被称为区块链应用1020,但是应当理解的是,根据硬件和操作,可以在一个或多个模块、应用、脚本、或其他编程结构中实现指令。处理器1002可包括多核处理器和/或多个处理器。

[0104] 应当理解的是,本文描述的设备和过程以及实现所描述的用于配置视频特征提取器的方法/过程的任何模块、例程、过程、线程、应用或其他软件组件可使用标准计算机编程技术和语言来实现。本申请不限于特定处理器、计算机语言、计算机编程约定、数据结构、或其他此类实施细节。

[0105] 应当说明,上述实施例说明而非限制本发明,在不脱离本发明的由所附权利要求限定的范围的情况下,本领域技术人员将能够设计出许多替代性实施例。在权利要求中,括号中的任何附图标记不应解释为对权利要求的限制。词语“包括(comprising)”和“包括(comprises)”等并非在整体上排除其他元件和步骤的存在,尽管这些元件和步骤并没有在任何权利要求或说明书中列出。在本说明书中,“包括(comprises)”意指“包括(includes)或由.....组成(consists of)”,“包括(comprises)”意指“包括(including)或由.....组成(consisting of)”。元件的单数引用不意味着排除这些元件的复数引用,反之亦然。本

发明可以借助包括若干不同元件的硬件,以及借助适当编程的计算机来实施。在列举了若干方式的设备权利要求中,这些方式中的若干个可以由硬件的同一个部件来体现。在相互不同的从属权利要求中列举某些措施的这一事实,并不指示这些措施的组合不能有利利用。

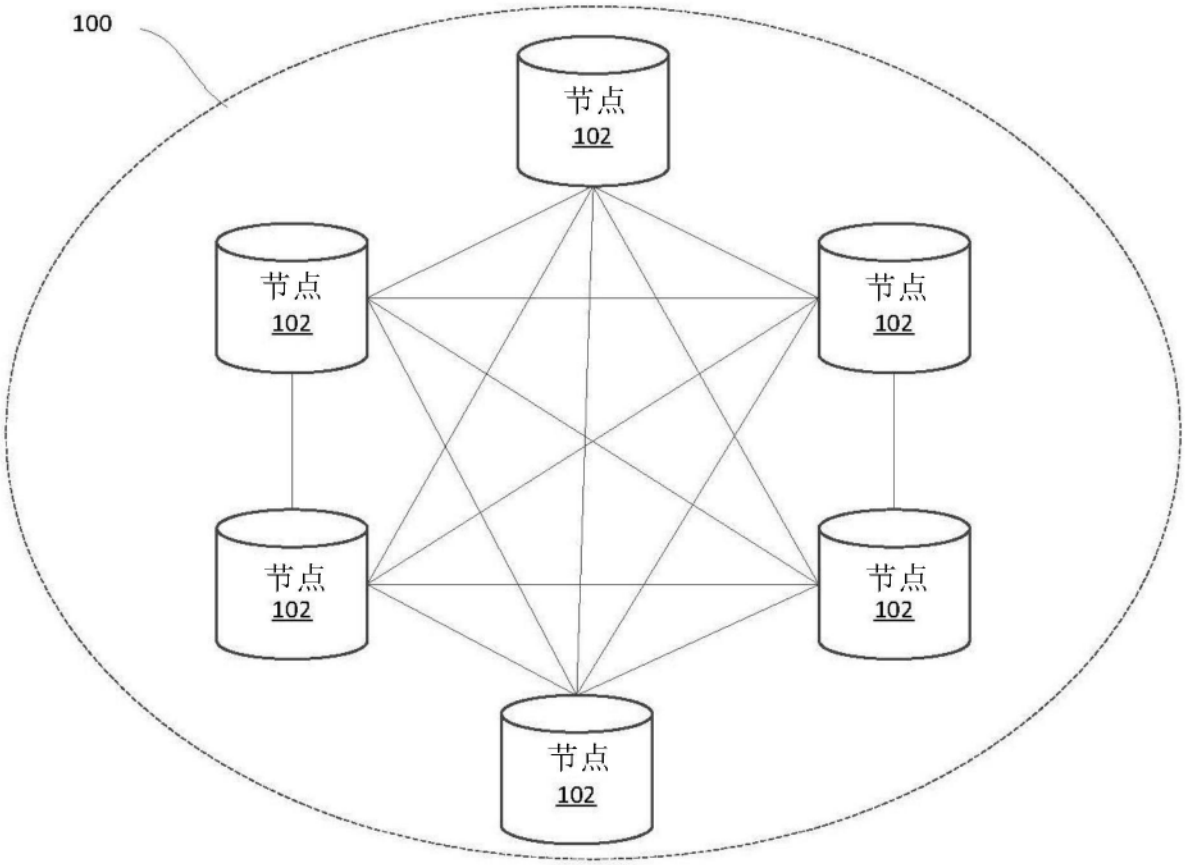


图1

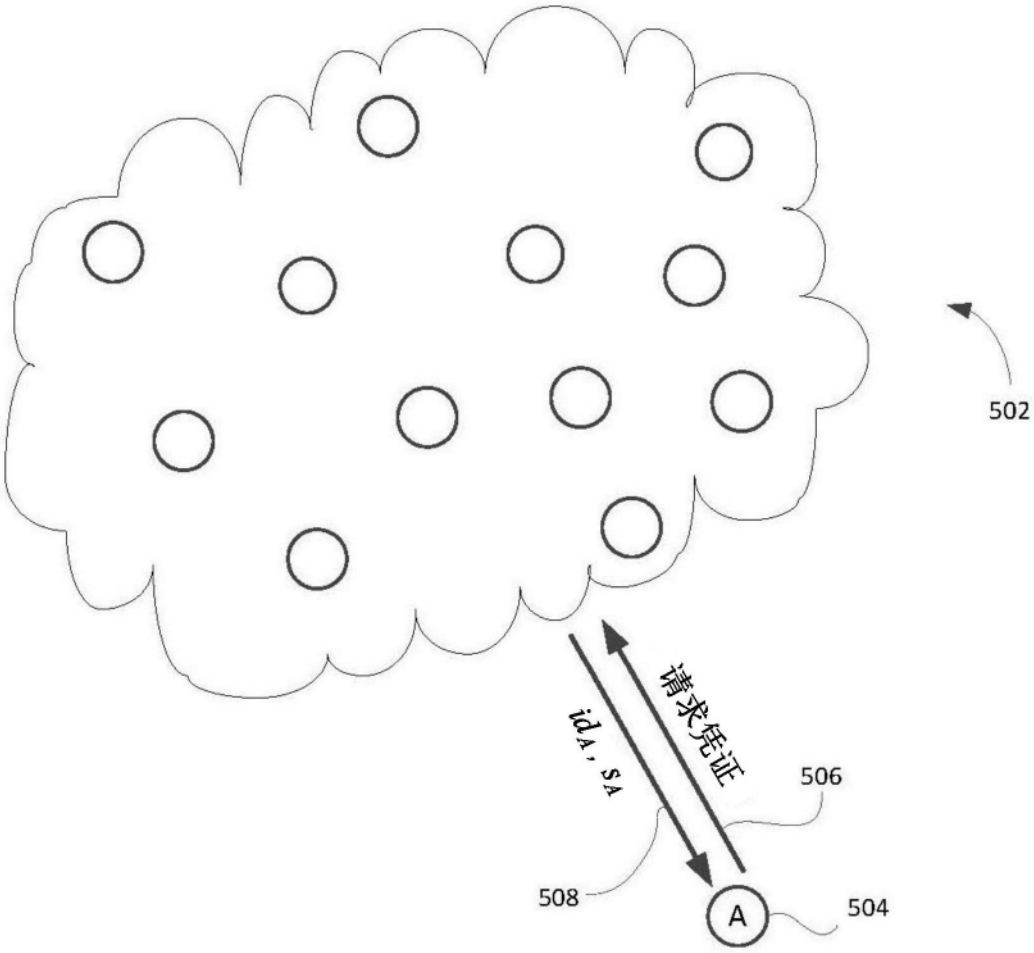


图2

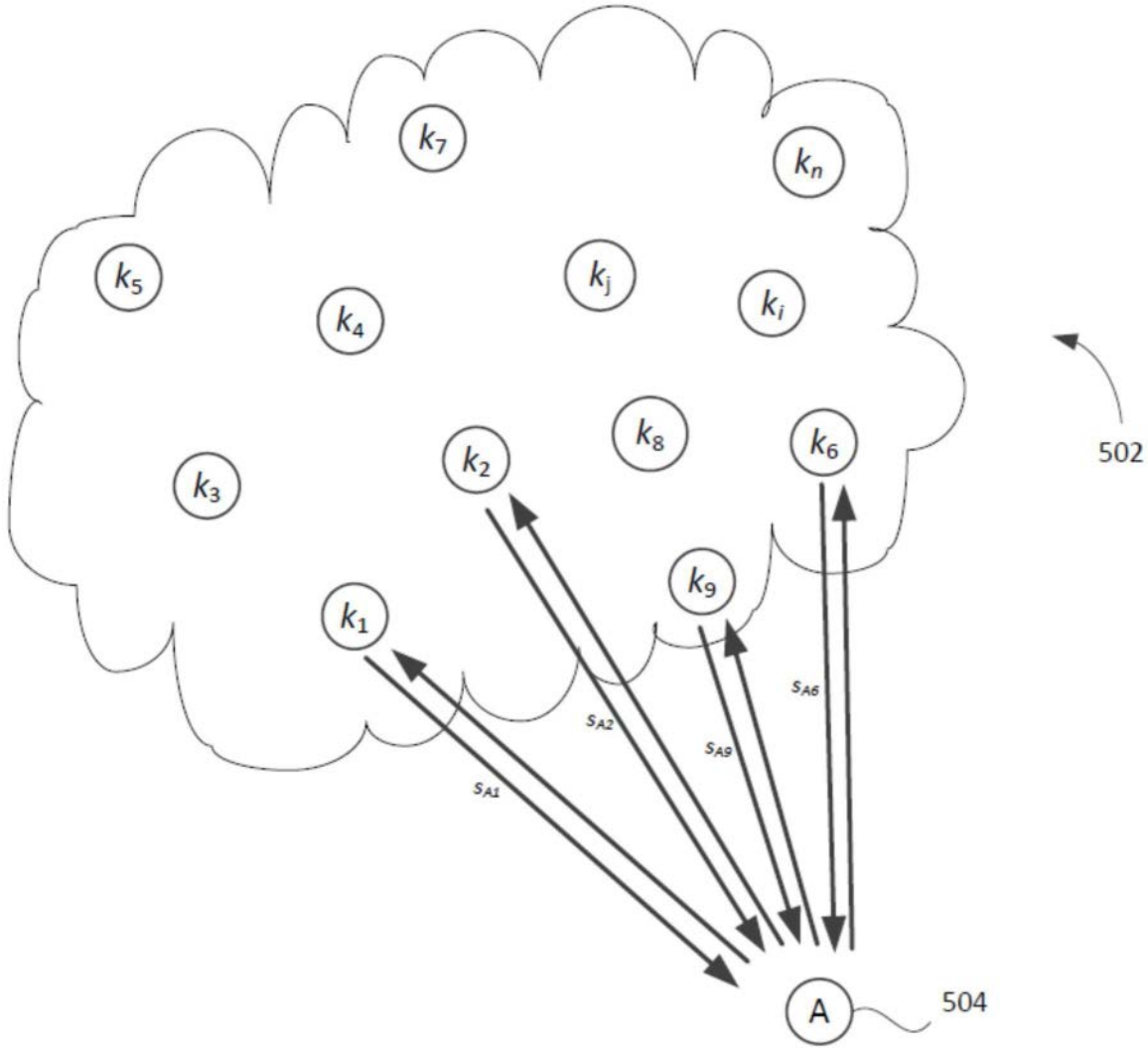


图3

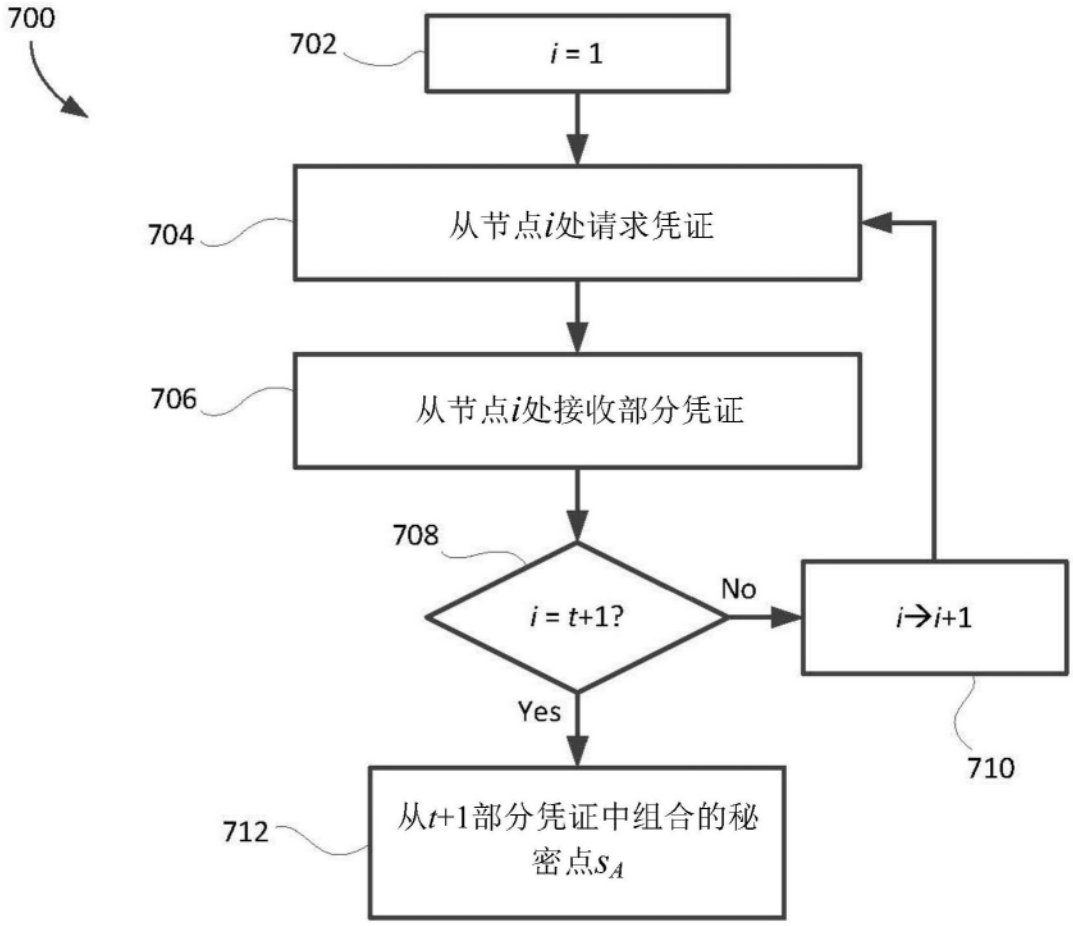

700
702
i = 1
704
从节点i处请求凭证
706
从节点i处接收部分凭证
708
i = t+1?
No
i→i+1
710
Yes
712
从t+1部分凭证中组合的秘密点s_A

图4

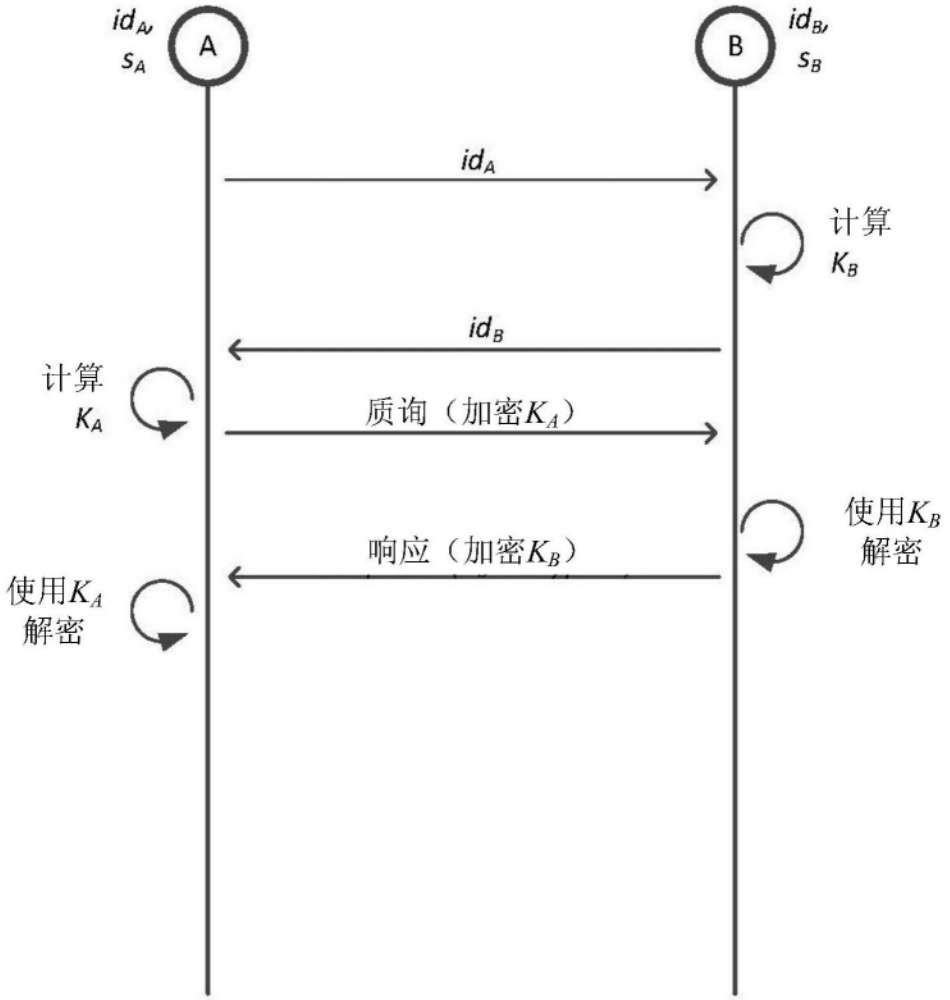

id_A, s_A
A
id_B, s_B
B
id_A
计算 K_B
id_B
计算 K_A
质询（加密K_A）
使用K_B 解密
响应（加密K_B）
使用K_A 解密

图5

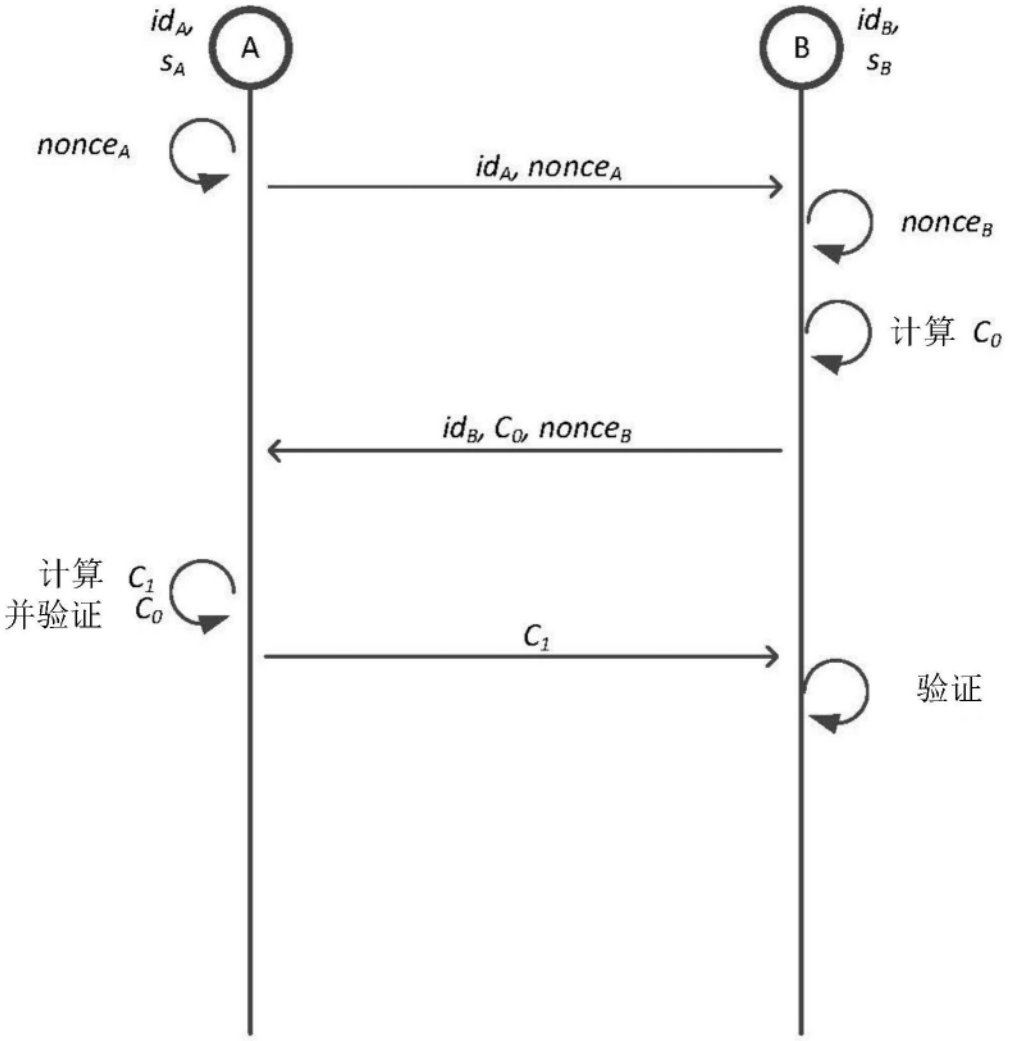

id_A, s_A
A
id_B, s_B
B
$nonce_A$
id_A, $nonce_A$
$nonce_B$
计算 C_0
id_B, C_0, $nonce_B$
计算 C_1
并验证 C_0
C_1
验证

图6

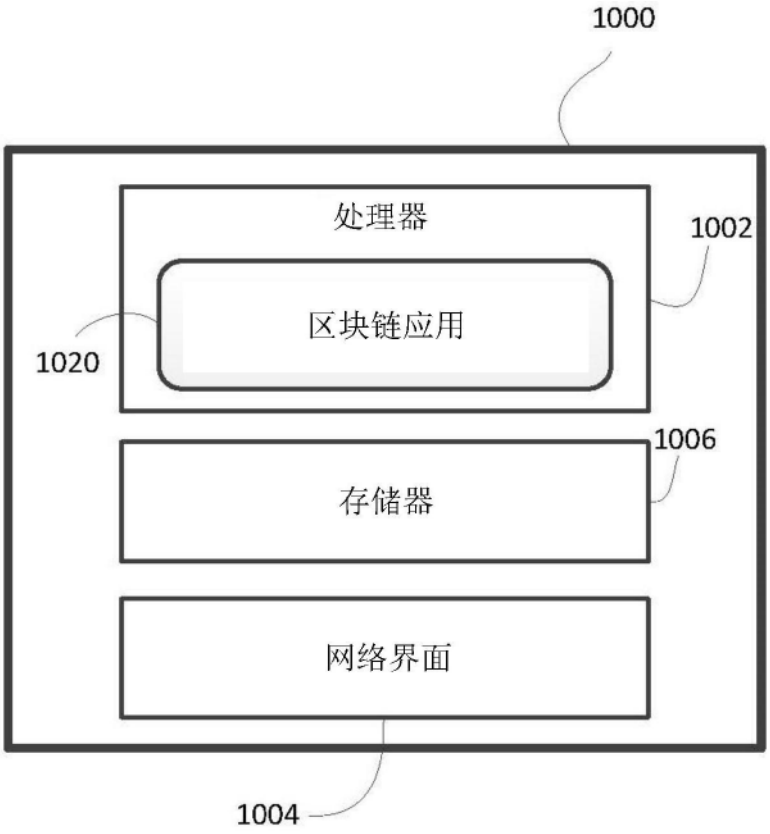


图7