



(12) 发明专利

(10) 授权公告号 CN 110945831 B

(45) 授权公告日 2021. 04. 27

(21) 申请号 201880048468.4

(22) 申请日 2018.10.09

(65) 同一申请的已公布的文献号
申请公布号 CN 110945831 A

(43) 申请公布日 2020.03.31

(30) 优先权数据
62/668,633 2018.05.08 US

(85) PCT国际申请进入国家阶段日
2020.01.20

(86) PCT国际申请的申请数据
PCT/US2018/055033 2018.10.09

(87) PCT国际申请的公布数据
W02019/216949 EN 2019.11.14

(73) 专利权人 维萨国际服务协会
地址 美国加利福尼亚州

(72) 发明人 M·扎马尼 A·艾加瓦尔

(74) 专利代理机构 上海专利商标事务所有限公司
31100

代理人 钱慰民 张鑫

(51) Int.Cl.
H04L 9/06 (2006.01)
H04L 9/08 (2006.01)

审查员 王朝英

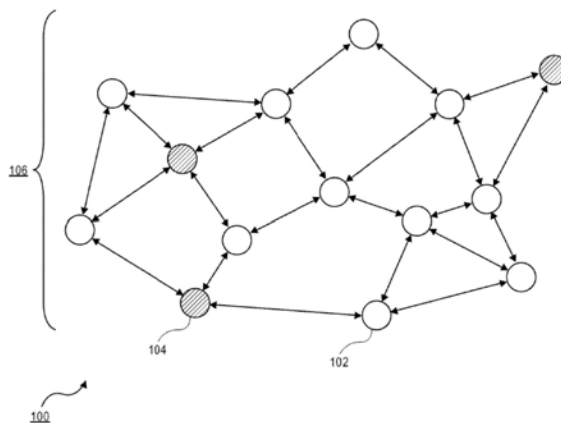
权利要求书3页 说明书24页 附图6页

(54) 发明名称

抗Sybil攻击身份的生成

(57) 摘要

本发明公开了一种方法。多个节点中的一个节点能够执行身份集合生成过程。然后,所述节点能够确定领导节点。所述节点能够将身份集合从所述多个节点中的每个节点扩散到所述多个节点。然后,所述节点能够确定多数集合,所述多数集合包括在所述身份集合的至少一半中出现的身份,其中所述领导节点将所述领导节点的所述多数集合扩散到所述多个节点。所述节点能够验证所述领导节点的所述多数集合。所述节点然后能够基于所述领导节点的所述多数集合来更新所述身份集合。



1. 一种方法,包括:
 - 由多个节点中的一个节点执行身份集合生成过程;
 - 由所述节点确定领导节点;
 - 由所述节点将身份集合扩散到所述多个节点;
 - 由所述节点确定身份集合组;以及
 - 由所述节点确定包括在所述身份集合组的至少一半所述身份集合中出现的身份的多数集合,其中所述领导节点将所述领导节点的所述多数集合扩散到所述多个节点,所述方法还包括:
 - 由所述节点从所述领导节点接收所述多数集合;
 - 由所述节点验证所述领导节点的所述多数集合;以及
 - 由所述节点基于所述领导节点的所述多数集合更新所述身份集合,并且执行所述身份集合生成过程还包括:
 - 由所述节点生成公共密钥、私有密钥和随机质询字符串;
 - 由所述节点将所述随机质询字符串传输到所述多个节点;
 - 由所述节点从所述多个节点接收多个随机质询字符串;
 - 由所述节点生成包括所述多个随机质询字符串和所述随机质询字符串的质询集合;
 - 由所述节点确定求解工作量证明的随机数;
 - 由所述节点将包括所述公共密钥、所述随机数、来自所述工作量证明的哈希值和所述质询集合的元组传输到所述多个节点;
 - 由所述节点从所述多个节点接收多个元组;以及
 - 由所述节点验证所述多个元组,其中如果所述多个元组中的一个元组有效,则将与所述元组相关联的公共密钥存储在身份集合中。
2. 如权利要求1所述的方法,其中确定所述领导节点还包括:
 - 由所述节点确定从所述多个节点接收的多个哈希值中的最小哈希值;
 - 由所述节点确定与所述最小哈希值相关联的所述身份集合的身份;以及
 - 由所述节点选择与所述最小哈希值相关联的第二节点作为所述领导节点。
3. 如权利要求2所述的方法,其中所述节点是所述第二节点。
4. 如权利要求1所述的方法,其中验证所述领导节点的所述多数集合还包括:
 - 由所述节点确定所述多数集合包括在所述身份集合的至少一半中出现的身份。
5. 如权利要求1所述的方法,其中所述随机质询字符串包括 k 位,并且其中所述多个节点中的每个节点生成不同的随机质询字符串。
6. 如权利要求1所述的方法,其中对所述工作量证明的输入包括所述公共密钥、所述质询集合和所述随机数。
7. 如权利要求6所述的方法,其中确定求解所述工作量证明的所述随机数还包括:
 - 由所述节点确定作为所述工作量证明的输出的所述哈希值;以及
 - 由所述节点确定所述哈希值小于预先确定的难度参数。
8. 如权利要求1所述的方法,还包括:
 - 由所述节点将所述身份集合分到存储桶中,其中在轮次中对每个存储桶进行处理。
9. 一种节点,包括:

处理器；

存储器设备；以及

计算机可读介质，所述计算机可读介质耦接到处理器，所述计算机可读介质包括能由所述处理器执行以实现方法的代码，所述方法包括：

由多个节点中的一个节点执行身份生成集合过程；

确定领导节点；

将身份集合扩散到所述多个节点；

由所述节点确定身份集合组；以及

确定包括在所述身份集合组的至少一半所述身份集合中出现的身份的多数集合，其中所述领导节点将所述领导节点的所述多数集合扩散到所述多个节点，

所述方法还包括：

由所述节点从所述领导节点接收所述多数集合；

由所述节点验证所述领导节点的所述多数集合；以及

由所述节点基于所述领导节点的所述多数集合更新所述身份集合，并且

执行所述身份集合生成过程还包括：

由所述节点生成公共密钥、私有密钥和随机质询字符串；

由所述节点将所述随机质询字符串传输到所述多个节点；

由所述节点从所述多个节点接收多个随机质询字符串；

由所述节点生成包括所述多个随机质询字符串和所述随机质询字符串的质询集合；

由所述节点确定求解工作量证明的随机数；

由所述节点将包括所述公共密钥、所述随机数、来自所述工作量证明的哈希值和所述质询集合的元组传输到所述多个节点；

由所述节点从所述多个节点接收多个元组；以及

由所述节点验证所述多个元组，其中如果所述多个元组中的一个元组有效，则将与所述元组相关联的公共密钥存储在身份集合中。

10. 如权利要求9所述的节点，其中确定所述领导节点还包括：

确定从所述多个节点接收的多个哈希值中的最小哈希值；

确定与所述最小哈希值相关联的所述身份集合的身份；以及

选择与所述最小哈希值相关联的第二节点作为所述领导节点。

11. 如权利要求10所述的节点，其中所述节点是所述第二节点。

12. 如权利要求9所述的节点，其中验证所述领导节点的所述多数集合还包括：

确定所述多数集合包括在所述身份集合中的至少一半中出现的身份。

13. 如权利要求9所述的节点，其中所述随机质询字符串包括 k 位，并且其中所述多个节点中的每个节点生成不同的随机质询字符串。

14. 如权利要求9所述的节点，其中对所述工作量证明的输入包括所述公共密钥、所述质询集合和所述随机数。

15. 如权利要求14所述的节点，其中确定求解所述工作量证明的所述随机数还包括：

确定作为所述工作量证明的输出的所述哈希值；以及

确定所述哈希值小于预先确定的难度参数。

16. 如权利要求9所述的节点,其中所述方法还包括:
由所述节点将所述身份集合分到存储桶中,其中在轮次中对每个存储桶进行处理。

抗Sybil攻击身份的生成

[0001] 相关申请的交叉引用

[0002] 本申请是要求2018年5月8日提交的美国临时申请号62/668,633的优先权的PCT申请,该申请以引用方式并入本文。

背景技术

[0003] 许多去中心化系统依赖于可信的第三方来为系统的参与者生成抗Sybil攻击身份的集合。在没有此类参与方的情况下,在保留所需安全属性的同时建立此集合变得极具挑战性。虽然最近有著作提出了解决此问题的方法,但是所有已知方案都具有较大的通信和计算开销。

[0004] 其他去中心化系统依赖于共识协议,该共识协议允许其参与者集体决定协议结果,而无需任何可信的参与方。此类共识协议实质上提供了一种投票机制,在由系统“唯一地标识”为单个实体之后,通过该投票机制为每个参与者分配一个投票。此类标识机制用于阻止双重投票,并且一般来讲用于阻止Sybil攻击,在这种攻击中,敌对方可以通过接管多数身份并且因此获得多数投票来恶意影响系统的集体决策。

[0005] 本发明的实施方案单独地或共同地解决了这些和其他问题。

发明内容

[0006] 本发明的一个实施方案涉及一种方法,包括:由多个节点中的一个节点执行身份集合生成过程;由该节点确定领导节点;由该节点从多个节点中的每个节点向所述多个节点扩散身份集合;并且由该节点确定多数集合,该多数集合包括在该身份集合的至少一半中出现的身份,其中该领导节点将该领导节点的多数集合扩散到所述多个节点。

[0007] 本发明的另一个实施方案涉及身份集合生成过程。该身份集合生成过程包括:由节点生成公共密钥、私有密钥和随机质询字符串;由该节点将该随机质询字符串传输到多个节点;由该节点从所述多个节点接收多个随机质询字符串;由该节点生成包括所述多个随机质询字符串和该随机质询字符串的质询集合;由该节点确定求解工作量证明的随机数;由该节点将包括公共密钥、随机数、来自工作量证明的哈希值和质询集合的元组传输到所述多个节点;由该节点从所述多个节点接收多个元组;并且由该节点验证所述多个元组,其中如果所述多个元组中的一个元组有效,则将与该元组相关联的公共密钥存储在身份集合中。

[0008] 本发明的另一个实施方案涉及节点,该节点包括:处理器;存储器设备;以及耦接到处理器的计算机可读介质,该计算机可读介质包括能由所述处理器执行以实现方法的代码,该方法包括:由多个节点中的一个节点执行身份生成集合过程;确定领导节点;将身份集合从多个节点中的每个节点扩散到所述多个节点;以及确定包括在身份集合的至少一半中出现的身份的多数集合,其中该领导节点将该领导节点的多数集合扩散到所述多个节点。

[0009] 关于本发明的实施方案的其他细节可见于具体实施方式和附图。

附图说明

- [0010] 图1示出根据本发明实施方案的示出节点网络的系统的框图。
- [0011] 图2示出根据本发明实施方案的节点的部件的框图。
- [0012] 图3示出根据本发明实施方案的例示身份集合生成过程的流程图。
- [0013] 图4示出根据本发明实施方案的生成身份集合的方法。
- [0014] 图5示出根据本发明实施方案的例示共识过程的流程图。
- [0015] 图6示出根据本发明实施方案的执行共识过程的方法。

具体实施方式

[0016] 在讨论本发明的实施方案之前,可以进一步详细描述一些术语。

[0017] “密钥对”可以包括一对链接的加密密钥。例如,密钥对可以包括公共密钥和对应的私有密钥。在密钥对中,第一密钥(例如,公共密钥)可以用于加密消息,而第二密钥(例如,私有密钥)可以用于解密加密的消息。另外,公共密钥可能能够认证用对应的私有密钥创建的数字签名。公共密钥可以分布在整个网络中,以便允许对使用对应的私有密钥签名的消息进行认证。公钥和私钥可以是任何适当格式,包括基于RSA或椭圆曲线密码学(ECC)的格式。在一些实施方案中,可以使用非对称密钥对算法来生成密钥对。然而,如本领域的普通技术人员将理解的那样,也可以使用其他方式来生成密钥对。在一些实施方案中,私有密钥可以被称为保密密钥。

[0018] “服务器计算机”可以包括功能强大的计算机或计算机集群。举例来说,服务器计算机可以是大型主机、小型计算机集群或像单元一样工作的一组服务器。在一个实例中,服务器计算机可以是耦合到网络服务器的数据库服务器。服务器计算机可耦合到数据库,且可包含用于服务来自一个或多个客户端计算机的请求的任何硬件、软件、其他逻辑或前述内容的组合。服务器计算机还可以是云中的计算机集合的形式,其服务一个或多个客户端计算机的请求。

[0019] 术语“节点”可以指连接点。在一些实施方案中,节点可以是能够创建、接收或传输数据的物理电子设备。在一些实施方案中,节点可以是网络内的计算设备。在一些实施方案中,节点可以是服务器计算机。在其他实施方案中,节点可以是计算设备上的软件模块,该软件模块是通信网络中的连接点。“诚实节点”可以是诚实地执行操作的节点。诚实节点可以不被敌对方或恶意参与方操作。

[0020] “领导节点”可以包括被其他节点选择和/或确定为领导的节点。在一些实施方案中,领导节点可以是已经从工作量证明确定与其他节点所确定的哈希值相比最低哈希值的节点。领导节点可以说是在轮次中领导其他节点。领导节点能够在轮次中将领导节点的多数集合扩散到多个节点。

[0021] “身份集合”可以包括节点身份的组。每个节点可以存储身份集合。该身份集合可以包括节点当前认为是诚实的身份。在一些实施方案中,一个节点的身份集合可以不同于另一个节点的身份集合。一旦所有诚实节点在相同或基本相同的身份集合上达成一致,诚实节点就可以获知其他每个诚实节点的身份。

[0022] “身份”可以包括标识节点的任何合适的标识符。节点可以在本地创建随机字符串,该随机字符串可以是公共/私有密钥对的随机生成的公共密钥。在一些实施方案中,公

共/私有密钥对可以是节点的身份。在其他实施方案中,节点的身份,也被称为 ID_i ,可以是包含公共密钥、随机数、哈希值和质询集合的元组,如本文所述。

[0023] “多数集合”可以是包括在由节点获得的身份集合的至少一半中出现的节点身份的集合。领导节点可以将领导节点的多数集合扩散到多个节点。

[0024] “质询集合”可以是包括随机质询字符串的集合。节点可以在接收多个随机质询字符串之后创建质询集合。随机质询字符串可以由节点确定的随机值。随机质询字符串可以在任何合适的范围内,例如在 $\{0,1\}^k$ 的范围内,其中 k 可以是安全性参数。

[0025] “工作量证明”和/或“工作过程证明”可以包括比验证解决方案在计算上更难以解决的问题。执行工作过程证明可以是低概率的随机过程,使得在生成有效的工作量证明之前,平均执行多次尝试错误。工作量证明可以通过哈希函数建模。对哈希函数的输入可以包括公共密钥、质询集合、随机数和/或其他合适的数据。哈希函数的输出可以是哈希值。例如,工作过程证明可以是 $h_i = H(pk_i || C_i || x_i) < d$,如本文所述。

[0026] “随机数”可以包括任何数字、字符串、位序列或其他数据值。随机数可以用于工作过程证明,以改变对哈希函数的输入,以便获得针对特定输入的哈希,其中该哈希符合要求,诸如低于难度值。

[0027] “哈希值”可以包括哈希函数的输出。哈希函数可以是工作过程证明的一部分。哈希值可以是任何合适的范围,例如 $(0,1)$ 的范围内的实数。可以通过加密算法诸如工作过程证明来处理数据,并且生成唯一的数值—哈希值。如果以任何方式修改了输入数据,则哈希值也可以明显地变化。

[0028] “元组”可以包括元素的序列和/或有序列表。元组可以包括任何合适的数据,例如公共密钥、随机数、哈希值、质询集合等。在一些实施方案中,元组可以包括身份,例如 $ID_j = (x_j, h_j, pk_j, C_j)$ 。

[0029] “轮次”可以包括重复发生的事件序列。在一些实施方案中,轮次可以是连续时间步长的序列,其中由诚实节点发送的每个消息可以在序列结束时到达其预期的一个或多个接收方(即,一个或多个其他节点)。时间步长可以包括在任何合适的信道上传输一个位所花费的最短时间。这可类似于同步轮次的标准定义,其中每个节点可以执行以下三个步骤:(1)从其他节点接收消息;(2)必要时执行一些本地计算;以及(3)发送消息至其他节点。对于任何给定的轮次,这三个步骤可以是原子性的。不涉及求解计算谜题的本地计算可以被认为是瞬时的。

[0030] 身份集合可以被分到存储桶中。“存储桶”可以包括数据单元。在一些实施方案中,存储桶可以包括身份集合的分区和/或部分。例如,在一些实施方案中,身份集合 S_i 可以被分到 n_i 存储桶中,其中 $n_i \leftarrow 2^{\lceil \log |S_i| \rceil}$ 。存储桶可以是身份集合 S_i 的分区。每个节点可以确定将本地集合 S_i 分到多少个存储桶中。在由每个节点执行分桶之后,从每个存储桶中,节点可以将PoW值最小的身份确定为该存储桶的领导节点。在轮次中可以对每个存储桶进行处理。

[0031] “处理器”可以指任何合适的一个或多个数据计算设备。处理器可包括一起工作以实现所要功能的一个或多个微处理器。处理器可以包括CPU,该CPU包括至少一个高速数据处理器,该高速数据处理器足以执行用于执行用户和/或系统生成的请求的程序组件。CPU可以是微处理器,诸如AMD的Athlon、Duron和/或Opteron;IBM和/或Motorola的PowerPC;IBM和Sony的Cell处理器;Intel的Celeron、Itanium、Pentium、Xeon和/或XScale;以及/或

者类似的一个或多个处理器。

[0032] “存储器”可以是存储电子数据的任何合适的一个或多个设备。合适的存储器可包括非暂时性计算机可读介质,其存储可由处理器执行以实现所要方法的指令。存储器的示例可以包括一个或多个存储器芯片、磁盘驱动器等。此类存储器可以使用任何合适的电、光和/或磁操作模式来操作。

[0033] I. 引言

[0034] 本发明的实施方案可以允许参与者(即,系统中的节点)在身份集合上达成一致,该身份集合可以在存在计算上受限的拜占庭式敌对方的情况下包含系统中所有诚实节点的身份,而无需对最初可能希望加入系统的节点数量的任何知识。诚实节点可以是行为诚实的非恶意节点。在本发明的一些实施方案中,最终的身份集合可以包括一定分数的敌对身份,其中最终身份集合中敌对身份的分数至多等于敌对方的总计算哈希能力。此外,本发明的实施方案可以处理节点的动态到达和离开,同时,在每个轮次中保留对于高达线性数量的扰动所需的安全性和带宽保证。

[0035] 与先前的著作不同,根据本发明实施方案的方法可以以期望的恒定的轮次数量运行,并且允许诚实节点以高概率(在安全性参数中)就诚实节点的身份集合达成共识。对于可以容忍对数数量的轮次的应用,本发明的实施方案可以以高概率(在节点数量中)允许诚实节点就其身份集合达成共识。在这两种情况下,诚实节点可以每轮发送 $\sim O(n)$ 位,并且可以在此过程中求解一次计算谜题(例如,工作量证明)。

[0036] 去中心化系统通常依赖于某种共识协议,该共识协议允许其参与者集体决定协议结果,而无需任何可信的参与方。此类共识协议实质上提供了一种投票机制,在由系统“唯一地标识”为单个实体之后,通过该投票机制为每个参与者分配一个投票。此类标识机制用于阻止双重投票,并且一般来讲用于阻止Sybil攻击,[John Douceur,“The Sybil attack”,收录在《Proceedings of the Second International Peer-to-Peer Symposium (IPTPS)》中,2002年],在这种攻击中,敌对方可以通过接管多数身份并且因此获得多数投票来恶意影响系统的集体决策。虽然无法完全避免Sybil攻击,但是可以限制其速率,使得每个共识轮次中恶意身份的数量受到限制,从而使得共识问题可以解决,参见[M. Pease, R. Shostak和L. Lamport,“Reaching agreements in the presence of faults”,《Journal of the ACM》,27(2):228-234,1980年4月]。

[0037] 对参与者的抗Sybil攻击标识可能需要花费大量的资源,特别是在缺少可信的权威机构诸如第三方的情况下。每个参与者可以执行一些“艰巨的任务”,诸如求解计算谜题,参见[Cynthia Dwork和Moni Naor,“Pricing via processing or combatting junk mail”,收录在《Advances in Cryptology—CRYPTO’92:12th Annual International Cryptology Conference Santa Barbara, California, USA August 16-20, 1992 Proceedings》中,第139页至第147页, Berlin, Heidelberg, 1993年, Springer Berlin Heidelberg],或CAPTCHA,参见[Luis Von Ahn, Manuel Blum, Nicholas J. Hopper和John Langford,“Captcha: Using hard ai problems for security”,收录在《Proceedings of the 22Nd International Conference on Theory and Applications of Cryptographic Techniques》中, EUROCRYPT’03, 第294页至第311页, Berlin, Heidelberg, 2003年, Springer-Verlag],使得具有实际上有限的资源的敌对方难以进行Sybil攻击。遗憾的是,

当每个节点执行速率限制任务时,此类速率限制技术通常会给系统的节点带来很大的开销。通过为系统中的每个节点分配对于多个会话有效的证书(称为身份),可以在多个共识会话中分摊该成本。拥有证书可以证明完成了速率限制任务。

[0038] 身份生成问题本身可以是一种共识问题,被称为交互一致性(IC),[Marshall Pease、Robert Shostak和Leslie Lamport,“Reaching agreement in the presence of faults”,《Journal of the ACM》(JACM),27(2):228-234,1980年],其中节点在没有任何可信的权威机构的情况下共同地在其输入(例如,身份)的单个向量上达成一致。遗憾的是,现有的IC解决方案[Marshall Pease、Robert Shostak和Leslie Lamport,“Reaching agreement in the presence of faults”,《Journal of the ACM》(JACM),27(2):228-234,1980年],包括可用于解决IC的共识问题的其他变体的解决方案(例如,Paxos[Leslie Lamport,“The part-time parliament”,《ACM Transactions on Computer Systems》,16(2):133-169,1998年5月]和PBFT[Miguel Castro和Barbara Liskov,“Practical byzantine fault tolerance”,收录在《Proceedings of the Third Symposium on Operating Systems Design and Implementation》中,OSDI’99,第173页至第186页,1999年]),不能直接用于解决身份生成问题,因为所有这些解决方案都假定已建立了身份。

[0039] 比特币,参见[Satoshi Nakamoto,“Bitcoin:A peer-to-peer electronic cash system”,2008年,可访问网址<https://bitcoin.org/bitcoin.pdf>获得],是最受欢迎的去中心化系统之一,该系统通过经由计算上困难的谜题(被称为工作量证明(PoW))限制Sybil攻击的速率来避免身份生成问题,参见[Cynthia Dwork和Moni Naor,“Pricing via processing or combatting junk mail”,收录在《Advances in Cryptology—CRYPTO’92:12th Annual International Cryptology Conference Santa Barbara,California,USA August 16-20,1992 Proceedings》中,第139页至第147页,Berlin,Heidelberg,1993年,Springer Berlin Heidelberg]。所有参与者都试图求解由至少一个参与者在每个共识难度调节周期所揭示的谜题。经由工作量证明过程生成的证明可以在短时间内被视为参与者的身份。在比特币中,PoW还用作分布式彩票协议,以在每个轮次中选取无法预测的领导节点,该领导节点帮助完成共识过程。领导者的不可预测性允许比特币可以保护其共识协议免受完全自适应的敌对方的攻击,这些敌对方可以在每一轮次开始时选择已损坏节点的集合,从而提前损坏领导者。然而,实际上,敌对方通常被认为是轻度适应性的,参见[Eleftherios Kokoris-Kogias、Philipp Jovanovic、Linus Gasser、Nicolas Gailly、Ewa Syta和Bryan Ford,“Omniledger:A secure, scale-out, decentralized ledger via sharding”,《Cryptology ePrint Archive, Report 2017/406》,2017年,<https://eprint.iacr.org/2017/406>]。

[0040] 由于随机性问题,即使是现有的区块链协议也不能用于无偏性身份。虽然最近有著作提出了解决此问题的方法,但遗憾的是,所有已知方案都具有大量的通信和计算开销,因此是不切实际的。许多去中心化协议都假定存在可信的设置协议,该设置协议可以生成用于认证其参与者身份的公共密钥基础结构或用于引导协议的公共参考字符串。

[0041] 根据本发明实施方案的方法对诚实节点进行寻址,这些诚实节点在假定不存在可信设置的设定中,有效地生成抗Sybil攻击身份(ID)的集合并且就此达成共识。诚实节点可以在ID的集合上达成一致,使得ID的集合可以包含(1)所有诚实节点的ID,以及(2)系统中

每个不诚实节点的最多一个ID。在一些实施方案中,这些ID可以由单个节点随机生成,这些节点可以使用密钥生成算法来产生密钥对 (pk, sk) , 包括公共密钥和私有密钥,每一者的长度可以为 κ 位。密钥生成阶段可以在每个节点的本地,并且敌对方可以忽略诚实节点的私有随机位。因此,一旦敌对方的公共密钥为所有节点所知,敌对方可能就无法假冒任何诚实节点,除了 κ 中的概率可以忽略不计以外。此外,敌对节点可以自由选择密钥对 (pk, sk) 。因此,敌对节点的密钥对不能被视为无偏随机字符串。

[0042] A. 问题陈述

[0043] 一个示例性系统可以包括多个节点。可以存在 n 个节点, P_1, \dots, P_n ,这些节点可能希望参与在同步对等网络上运行的分布式协议。然而,在过程开始时,可能不存在可信的设置来允许这些节点彼此标识,或者甚至估计节点数量 n 的值。每个节点 P_i 可以在本地创建随机字符串,该随机字符串可以被称为其身份 ID_i 。在一些实施方案中,随机字符串可以是公共/私有密钥对的随机生成的公共密钥。在其他实施方案中,节点的身份可以是包括公共密钥、随机数、哈希值和质询集合的元组,如本文所述。所有诚实节点可能都希望在至少包括所有诚实节点的身份的身份的集合 S 上达成一致。

[0044] 身份生成(IG)协议可以是具有以下属性的共识协议:一致性、有效性、可验证性和可终止性。协议属性可以意指所有诚实节点可以在相同的身份集合 S 上达成一致。有效性属性可以意指如果节点 P_i 为诚实节点,并且其身份为 ID_i ,则所有诚实节点可以在其集合 S 中包括 ID_i 。如果 P_i 是不诚实的,则诚实节点对于集合 S 中的 P_i 可以不具有任何身份或任何任意身份。可验证性属性可以意指所有诚实节点可以验证集合 S 中任何身份的正确性。可终止性属性可以意指每个诚实节点最终可以决定集合 S 。

[0045] IC协议和IG协议之间可存在两个主要区别:(1) IC协议可能不需要提供可验证性属性;以及(2) IC协议可以输出向量而不是集合。在IC协议中,由于可以假定节点已建立了身份,所以可能定义身份的有序集合(即,向量)。

[0046] B. 模型

[0047] 图1示出根据本发明的一些实施方案的包括许多部件的系统100的框图。该系统包括许多诚实节点102和许多恶意节点104。多个节点106可以包括诚实节点102和恶意节点104。多个节点106中的每个节点可以与附近的其他节点进行操作性通信。在一些实施方案中,多个节点106中的每个节点可以与多个节点106中的每个节点进行操作性通信。

[0048] 在不失一般性的情况下,每个诚实节点102和每个恶意节点104可以配备有一个单元的计算哈希能力,这可以允许节点执行等量的工作量证明(PoW)。根据标准随机预言机假设,可以在单个轮次中将PoW建模为计算谜题,参见[Cynthia Dwork和Moni Naor,“Pricing via processing or combatting junk mail”,收录在《Advances in Cryptology—CRYPTO’92:12th Annual International Cryptology Conference Santa Barbara, California, USA August 16-20,1992 Proceedings》中,第139页至第147页,Berlin, Heidelberg,1993年,Springer Berlin Heidelberg]。哈希能力可以是特定节点所具有的计算能力。例如,功能更强大的计算机具有较高的哈希能力。然而,应当理解,每个节点可以具有不同量的计算哈希能力。

[0049] 接下来,将描述恶意节点104的威胁模型。可能存在拜占庭敌对方,该拜占庭敌对方可以任意地偏离协议,但可能不能更改或延迟由诚实节点102发送的消息。该敌对方可以

控制许多恶意节点104。该敌对方可以配备有高达多个节点106的总哈希能力的 $f < 1/3$ 分数的哈希能力。换句话说讲,敌对哈希能力的总量可以为 $\frac{nf}{1-f}$ 。此类敌对方通常被称为计算阈值敌对方,参见[C.Decker and R.Wattenhofer,“Information propagation in the Bitcoin network”,收录在《P2P》中,第1页至第10页,IEEE,2013年],并且通常假定在每个轮次中只能执行多项式量的计算。本发明的实施方案可以允许使用概率多项式时间图灵机(PPT)进行形式化建模。参见[Elaine Shi和Rafael Pass,“Feasibilities and infeasibilities for achieving responsiveness in permissionless consensus”,DISC,2017年],以获得有关此建模的更多详细信息。在一些实施方案中,可以存在任何合适数量的诚实节点102。

[0050] 接下来,将描述网络模型。网络中的每一对相邻节点可以经由同步可靠信道连接。为了通过Lamport[Marshall Pease、Robert Shostak和Leslie Lamport,“Reaching agreement in the presence of faults”,《Journal of the ACM》(JACM),27(2):228-234,1980年]和[Michael J Fischer、Nancy A Lynch和Michael Merritt,“Easy impossibility proofs for distributed consensus problems”,《Distributed Computing》,1(1):26-39,1986年]规避下限,这些信道可以提供可靠的消息扩散和达成共识所需的最小认证功能性。由于根据本发明实施方案的方法可以允许节点求解PoW,因此,由于最近发布的不可能性结果,参见[Rafael Pass和Elaine Shi,“Rethinking large-scale consensus”,收录在《Computer Security Foundations Symposium(CSF),2017 IEEE 30th》中,第115页至第129页,IEEE,2017年]和[Juan Garay、Aggelos Kiayias和Nikos Leonardos,“The bitcoin backbone protocol:Analysis and applications”,收录在《Annual International Conference on the Theory and Applications of Cryptographic Techniques》中,第281页至第310页,Springer,2015年],同步通信的假设是不可避免的。可以传递由任何诚实节点102广播的消息。在一些实施方案中,此类广播机制可以通过扩散来实现,例如,参见[R.Karp、C.Schindelhauer、S.Shenker和B.Vocking,“Randomized rumor spreading”,收录在《Proceedings of the 41st Annual Symposium on Foundations of Computer Science》中,FOCS'00,第565页-,Washington,DC,USA,2000年,《IEEE Computer Society》]。

[0051] 图1中所示的实体、提供者、网络和设备之间的消息可以使用安全通信协议进行传输,诸如但不限于文件传输协议(FTP);超文本传输协议(HTTP);安全超文本传输协议(HTTPS)、安全套接字层(SSL)、ISO(例如,ISO 8583),等等。通信网络可以包括任何合适的通信介质。通信网络可以是以下项中的一者和/或它们的组合:直接互连;互联网;局域网(LAN);城域网(MAN);将任务作为互联网上的节点进行操作(OMNI);安全的自定义连接;广域网(WAN);无线网络(例如,采用协议,诸如但不限于无线应用协议(WAP)、I-模式等),等等。

[0052] 根据本发明实施方案的方法可以按轮次进行。轮次可以是连续时间步长的序列,其中由诚实节点102发送的每个消息可以在序列结束时到达其预期的一个或多个接收方(即,一个或多个其他节点)。时间步长可以被定义为在任何合适的信道上传输一个位所花费的最短时间。这可类似于同步轮次的标准定义,其中每个节点可以执行以下三个步骤:

(1) 从其他节点接收消息；(2) 必要时执行一些本地计算；以及(3) 发送消息至其他节点。对于任何给定的轮次，这三个步骤可以是原子性的。可能不涉及求解计算谜题的本地计算可以被认为是瞬时的。

[0053] 图2示出根据本发明实施方案的节点的部件的框图。节点200包括处理器202、计算机可读介质204、输出元件206、安全存储器208、网络接口210，以及输入元件212。计算机可读介质204可以包括许多模块，诸如身份集合生成模块204A和共识模块204B。

[0054] 计算机可读介质204可以包括代码，该代码可由处理器202执行以实现一种方法，该方法包括：由多个节点中的一个节点执行身份集合生成过程；由该节点确定领导节点；由该节点从多个节点中的每个节点向所述多个节点扩散身份集合；以及由该节点确定多数集合，该多数集合包括在身份集合的至少一半中出现的身份，其中领导节点将该领导节点的多数集合扩散到所述多个节点。

[0055] 身份集合生成模块204A可以包括软件代码，该软件代码可以生成公共密钥、私有密钥和随机质询字符串；将该随机质询字符串传输到多个节点；从多个节点接收多个随机质询字符串；生成包括所述多个随机质询字符串和该随机质询字符串的质询集合；确定求解工作量证明的随机数；将包括公共密钥、随机数、来自工作量证明的哈希值和质询集合的元组传输到多个节点；从所述多个节点接收多个元组；并且验证所述多个元组，其中如果所述多个元组中的一个元组是有效的，则将与该元组相关联的公共密钥存储在身份集合中。

[0056] 共识模块204B可以包括软件代码，该软件代码可以确定领导节点；将身份集合从多个节点中的每个节点扩散到所述多个节点；并且确定多数集合，该多数集合包括在身份集合的至少一半中出现的身份，其中领导节点将该领导节点的多数集合扩散到所述多个节点。

[0057] 输出元件206可以包括可以输出数据的任何合适的一个或多个设备。输出元件206的示例可以包括显示屏、扬声器和数据传输设备。

[0058] 安全存储器208可以安全地存储加密的访问数据、密钥标识符、公共密钥和任何其他相关数据。安全存储器208可以是安全元件、硬件安全模块或任何其他合适形式的安全数据存储的形式。

[0059] 网络接口210可以包括可以允许节点200与外部计算机通信的接口。网络接口210可以使节点200能够与另一个设备(例如，其他节点)进行数据通信。网络接口210的一些示例可以包括调制解调器、物理网络接口(诸如以太网卡或其他网络接口卡(NIC))、虚拟网络接口、通信端口、个人计算机存储卡国际协会(PCMCIA)插槽和卡等。由网络接口210启用的无线协议可以包括Wi-Fi™。

[0060] 经由网络接口210传输的数据可以是信号的形式，这些信号可以是能够由外部通信接口接收的电气、电磁、光学或任何其他信号(统称为“电子信号”或“电子消息”)。这些电子消息(其可以包含数据或指令)可以通过通信路径或渠道在网络接口210与其他装置之间提供。如上所述，可以使用任何合适的通信路径或渠道，比方说例如电线或电缆、光纤、电话线、蜂窝链路、射频(RF)链路、WAN或LAN网络、互联网或任何其他合适的介质。

[0061] 输入元件212可以包括能够将数据输入到节点200中的任何合适的一个或多个设备。输入设备的示例包括按钮、触摸屏、触摸板、麦克风等。

[0062] II. 相关著作

[0063] Douceur[John Douceur,“The Sybil attack”,收录在《Proceedings of the Second International Peer-to-Peer Symposium(IPTPS)》中,2002年]相对于需要准入控制来管理某些公共资源使用的应用和最近对无需许可的系统的研究,参见[Elaine Shi和Rafael Pass,“Feasibilities and infeasibilities for achieving responsiveness in permissionless consensus”,《DISC》,2017年],讨论了对抗Sybil身份的问题,在无需许可的系统中参与的用户可以随意加入和离开。一种受欢迎的方法是使用建模为随机预言机或其体现的计算谜题,以防止恶意节点通过利用其有限的计算能力来向系统添加多个身份。这种想法是由Dwork和Naor的开创性著作[Cynthia Dwork和Moni Naor,“Pricing via processing or combatting junk mail”,收录在《Advances in Cryptology—CRYPTO’92: 12th Annual International Cryptology Conference Santa Barbara,California,USA August 16-20,1992 Proceedings》中,第139页至第147页,Berlin,Heidelberg,1993年, Springer Berlin Heidelberg]、[Cynthia Dwork、Moni Naor和Hoeteck Wee,“Pebbling and proofs of work”,收录在《CRYPTO》中,第5卷,第37页至第54页, Springer,2005年]所激发的,即使用计算谜题来对抗垃圾邮件。Aspnes等人[James Aspnes、Collin Jackson和Arvind Krishnamurthy,“Exposing computationally-challenged byzantine impostors”,《Department of Computer Science,Yale University,New Haven,CT, Tech.Rep》,2005年]提出了一种协议,该协议无需假定任何公共随机设置即可创建抗Sybil攻击身份的集合。然而,对于属于不诚实节点的身份,其算法可以导致诚实节点之间相对于其输出集合产生不一致。对它们的方法中存在的问题将在下面进一步详细讨论。有几篇后续论文(参见[Jonathan Katz、Andrew Miller和Elaine Shi,“Pseudonymous broadcast and secure computation from cryptographic puzzles”,《Cryptology ePrint Archive,Report 2014/857,2014》,<https://eprint.iacr.org/2014/857>]、[Marcin Andrychowicz和Stefan Dziembowski,《PoW-Based Distributed Cryptography with No Trusted Setup》,第379页至第399页, Springer Berlin Heidelberg,Berlin,Heidelberg, 2015年]、[Lisa Ekey、Sebastian Faust和Julian Loss,“Efficient algorithms for broadcast and consensus based on proofs of work.Technical report”,《Cryptology ePrint Archive,Report 2017/915,2017》]、[Guido Urdeneta、Guillaume Pierre和Maarten Van Steen,“A survey of dht security techniques”,《ACM Computing Surveys (CSUR)》,43 (2) :8,2011年]和[Diksha Gupta、Jared Saia和Maxwell Young,“Proof of work without all the work”,《arXiv preprint arXiv:1708.01285》,2017年])提出了在各种设定中针对该问题的不同类型的解决方案。

[0064] 针对该问题可存在两种解决方案:一种是假定存在可信设置,诸如PKI(公共密钥基础结构),然后尝试阻止Sybil IP地址访问共享资源,参见[Guido Urdeneta、Guillaume Pierre和Maarten Van Steen,“A survey of dht security techniques”,《ACM Computing Surveys (CSUR)》,43 (2) :8,2011年]、[Diksha Gupta、Jared Saia和Maxwell Young,“Proof of work without all the work”,《arXiv preprint arXiv:1708.01285》, 2017年]、[Diogo Monica,“Thwarting the Sybil attack in wireless ad hoc networks”,《INSTITUTO SUPERIOR TECNICO (IST)》,2009年]、[Satoshi Nakamoto,“Bitcoin:A peer-to-peer electronic cash system”,2008年,可访问网址<https://>

bitcoin.org/bitcoin.pdf获得]、[Nikita Borisov,“Computational puzzles as Sybil defenses”,收录在《Sixth IEEE International Conference on Peer-to-Peer Computing,2006》,P2P 2006,第171页至第176页,IEEE,2006年]、[Frank Li、Prateek Mittal、Matthew Caesar和Nikita Borisov,“Sybilcontrol:practical Sybil defense with computational puzzles”,收录在《Proceedings of the seventh ACM workshop on Scalable trusted computing》中,第67页至第78页,《ACM杂志》,2012年]、[Hosam Rowaihy、William Enck、Patrick Mcdaniel和Thomas La Porta,“Limiting Sybil attacks in structured peer-to-peer networks”,《Technical report,Network and Security Research Center,Department of Computer Science and Engineering, Pennsylvania State University,USA》,2005年],以及[A.Kate、Y.Huang和I.Goldberg,“Distributed key generation in the wild”]。虽然另一种解决方案试图设计技术来提供一种本质上抗Sybil攻击的基础结构,使得可以将其用于引导其他(经过许可的)系统,参见[Jonathan Katz、Andrew Miller和Elaine Shi,“Pseudonymous broadcast and secure computation from cryptographic puzzles”,《Cryptology ePrint Archive,Report 2014/857,2014》,<https://eprint.iacr.org/2014/857>]、[James Aspnes、Collin Jackson和Arvind Krishnamurthy,“Exposing computationally-challenged byzantine impostors”,《Department of Computer Science,Yale University,New Haven,CT, Tech.Rep》,2005年]、[Marcin Andrychowicz和Stefan Dziembowski,“PoW-Based Distributed Cryptography with No Trusted Setup”,第379页至第399页,Springer Berlin Heidelberg,Berlin,Heidelberg,2015年]、[Lisa Eckey、Sebastian Faust和Julian Loss,“Efficient algorithms for broadcast and consensus based on proofs of work.Technical report”,《Cryptology ePrint Archive,Report 2017/915》,2017年]、[Tuyet Duong、Lei Fan、Thomas Veale和Hong-Sheng Zhou,“Securing bitcoin-like backbone protocols against a malicious majority of computing power”,《IACR Cryptology ePrint Archive》,2016:716,2016年]、[Ruomu Hou、Irvan Jahja、Loi Luu、Prateek Saxena和Haifeng Yu,“Randomized view reconciliation in permissionless distributed systems”,收录在《IEEE Conference on Computer Communications (INFOCOM)》中,IEEE,2018年],以及[Juan A Garay、Aggelos Kiayias、Nikos Leonardos和Giorgos Panagiotakos,“Boot-strapping the blockchain-directly”,《IACR Cryptology ePrint Archive》,2016:991,2016年]。本发明的实施方案可涉及第二目标。

[0065] Katz等人[Jonathan Katz、Andrew Miller和Elaine Shi,“Pseudonymous broadcast and secure computation from cryptographic puzzles”,《Cryptology ePrint Archive,Report 2014/857,2014》,<https://eprint.iacr.org/2014/857>]、Andrychowicz和Dziembowski[Marcin Andrychowicz和Stefan Dziembowski,“PoW-Based Distributed Cryptography with No Trusted Setup”,第379页至第399页,Springer Berlin Heidelberg,Berlin,Heidelberg,2015年],以及Hou等人[Ruomu Hou、Irvan Jahja、Loi Luu、Prateek Saxena和Haifeng Yu,“Randomized view reconciliation in permissionless distributed systems”,收录在《IEEE Conference on Computer Communications (INFOCOM)》中,IEEE,2018年]描述了彼此相似的协议。然而,存在一些明显

的差异。Katz等人的著作[Jonathan Katz、Andrew Miller和Elaine Shi,“Pseudonymous broadcast and secure computation from cryptographic puzzles”,《Cryptology ePrint Archive, Report 2014/857, 2014》, <https://eprint.iacr.org/2014/857>]假定存在随机信标,以确保敌对方无法预先计算计算谜题。然后,它描述了一组节点如何根据可公开获得的随机信标共同创建(抗Sybil攻击的)PKI。根据本发明实施方案的方法可以使用基于PoW解决方案的可靠广播,但是,实施方案不像现有技术中那样假定公共可用的随机信标。

[0066] Andrychowicz和Dziembowski的论文[Marcin Andrychowicz和Stefan Dziembowski,“PoW-Based Distributed Cryptography with No Trusted Setup”,第379页至第399页, Springer Berlin Heidelberg, Berlin, Heidelberg, 2015年]通过实现诚实节点的输出之间的一致性,提供了利用随机信标的解决方案。然而,它们的算法相当复杂,并且涉及对来自每个诚实节点的多个计算谜题的解决方案,从而使它们的方法在实践中不可行。本发明的实施方案可以通过每个节点使用一次PoW来改善该结果,因此,实施方案实际上是有效的。

[0067] 最后,Hou等人的著作[Ruomu Hou、Irvan Jahja、Loi Luu、Prateek Saxena和Haifeng Yu,“Randomized view reconciliation in permissionless distributed systems”,收录在《IEEE Conference on Computer Communications (INFOCOM)》中, IEEE, 2018年]将该问题描述为观点分歧问题,并且提供了一种随机化的解决方案,适用于敌对哈希能力的分数高达总哈希能力的四分之一、总轮次复杂度为 $\Theta(\log N / \log \log N)$ 的情况。本发明的实施方案可以通过提供可以以恒定的轮次数量运行的算法来改善该结果,该算法可以以安全性参数中的高概率确保共识,并且可以将对敌对哈希能力的弹性提高到总哈希能力的三分之一。

[0068] III. 阶段I

[0069] 根据本发明实施方案的方法可以分两个阶段进行。在高级别处,本发明的实施方案可以首先允许诚实节点为所有节点本地计算身份集合,使得与任意两个诚实节点相对应的集合在不诚实节点的身份上可以不同。接下来,根据本发明实施方案的方法可以使用可以为随后的轮次选择领导者的分桶算法,每个轮次可以使诚实节点更接近在其身份集合上达成一致。

[0070] 该过程的第一阶段可以被称为几乎处处身份一致阶段,其灵感来自几乎处处拜占庭一致的概念,参见[C. Dwork、D. Peleg、N. Pippenger和E. Upfal,“Fault tolerance in networks of bounded degree”,收录在《Proceedings of the eighteenth annual ACM symposium on Theory of computing》中,第379页,《ACM杂志》,1986年],其中并非所有但几乎所有诚实参与方都必须达成一致。每个节点可以创建ID的集合, S_i ,该ID的集合起初可以为空。第一阶段可以如下进行:每个诚实节点可以在本地生成随机密钥对,包括公共密钥和私有密钥,以及随机质询字符串。每个节点可以将其质询字符串扩散到网络(即,多个节点中的其他节点),并且可以接收从其他节点生成的质询字符串。这些质询字符串可以一起被收集在集合中,并且在一些实施方案中,可以以节点的公共密钥为后缀。例如,这可以用字符串的级联来表示,并且该集合中的成员资格可以由每个节点轻松测试。然后,节点可以求解PoW,该PoW可以由哈希函数H建模。节点可以计算随机数x,使得当随机数x被附加到由

公共密钥和质询集合的级联形成的字符串时,哈希函数H在输入上的结果可以是哈希值。该哈希值可以是(0,1)范围内的实数。

[0071] 在一些实施方案中,如果哈希函数H的输出小于某个已知难度参数d,则PoW可以成功。可以根据网络延迟和安全性参数的边界来设置难度参数d。一旦确定对PoW的成功的解决方案,每个诚实节点都可以将其解决方案及其公共密钥扩散到多个节点中的其他节点。在从某个节点 P_j 接收解决方案时,接收方诚实节点 P_i 可以检查PoW是否相对于随机数和难度参数d正确计算。然后,如果 P_j 在质询字符串的集合中包括 P_i 的质询字符串以求解PoW,则接收方诚实节点 P_i 可以在其本地ID集合中包括 P_j 的公共密钥。这样可以防止敌对方使不同的诚实节点为相同的恶意节点添加不同的公共密钥。 $S_i S_i$ 可以是第一阶段结束时诚实节点 P_i 所接受的ID集合。

[0072] 图3示出根据本发明实施方案的例示身份集合生成过程的流程图。图3包括第一节点302、第二节点304和第三节点306。虽然以特定顺序示出了步骤,但是应当理解,本发明的实施方案可以包括具有以不同顺序进行的步骤的方法。另外,可以省略或添加步骤,并且这些步骤仍然可以在本发明的实施方案内。在图3中,多个节点中的一个节点可以执行身份集合生成过程。

[0073] 在步骤1之前,多个节点中的每个节点可以生成随机的公共密钥/私有密钥对 (pk_i, sk_i) ,并且对随机质询字符串 $c_i \in \{0, 1\}^k$ 进行采样。例如,第一节点302可以生成第一公共/私有密钥对 (pk_1, sk_1) 和第一质询字符串 c_1 。

[0074] 在步骤1处,每个节点可以在身份集合 S_i 中包括其自身的公共密钥 pk_i ,并且可以将随机质询字符串 c_i 扩散到多个节点中的每个节点。例如,第一节点302可以在第一身份集合 $S_1 308$ 中包括第一公共密钥 pk_1 ,然后将第一随机质询字符串 c_1 传输到多个节点,例如,传输到第二节点304和第三节点306。

[0075] 在步骤2处,在每个节点从多个节点中的每个其他节点接收随机质询字符串 c_i 之后,每个节点可以确定质询集合 C_i ,该质询集合可以包括从每个其他节点接收的随机质询字符串 c_i 。然后,每个节点可以确定随机数 $x_i \in \{0, 1\}^k$,使得 $h_i = H(pk_i || C_i || x_i) < d$,其中d可以是难度参数d。每个节点可以将元组 $ID_i = (x_i, h_i, pk_i, C_i)$ 传输到多个节点中的每个节点。例如,第一节点302可以从第二节点304接收第二质询字符串 c_2 ,并且可以从第三节点306接收第三质询字符串 c_3 。第一节点302可以基于 c_1, c_2 和 c_3 来确定第一质询集合 $C_1 310$ 。第一质询集合 $C_1 310$ 可以包括任何合适数量的质询字符串,诸如节点所接收的所有质询字符串。

[0076] 在确定第一质询集合 $C_1 310$ 之后,第一节点302然后可以确定求解哈希函数H的第一随机数 x_1 。第一节点302可以生成包括第一随机数 x_1 、第一哈希值 h_1 、第一公共密钥 pk_1 和第一质询集合 C_1 的第一元组 ID_1 。在生成第一元组 ID_1 之后,第一节点302可以将该第一元组 ID_1 传输到多个节点中的每个其他节点。每个节点可以将包括公共密钥、随机数、来自工作量证明的哈希值和质询集合的元组传输到多个节点。

[0077] 在步骤3处,在从多个节点 P_j 中的每个其他节点接收元组即 $ID_j = (x_j, h_j, pk_j, C_j)$ 之后,如果工作证明 $h_j = H(pk_j || C_j || x_j)$ 是可接受的,并且节点自身的质询字符串在所接收的质询集合,即, $c_i \in C_j$ 中,则每个节点可以在身份集合 S_i 中包括元组的所接收的公共密钥 pk_j 。节点可以验证多个元组,其中如果多个元组中的一个元组是有效的,则节点可以将与

该元组相关联的公共密钥存储在身份集合中。有效元组可以与可接受的工作量证明相关联。另外,有效元组可以包括质询集合,该质询集合包括节点(即,验证元组的节点)自身的质询字符串 c_i 。包括质询集合可以允许第一节点确定在不同节点的工作量证明过程的计算中,该不同节点是否已包括第一节点的质询字符串。如果不使用适当质询集合的情况下完成了恶意节点的工作量证明过程,则节点可以确定不同节点为恶意节点。

[0078] 下面的协议1描述了上面图3中所示的身份集合生成过程的示例性伪代码。

协议 1: 几乎处处身份生成

诚实节点 P_i 遵循的步骤顺序。

1. **轮次 1:** 创建随机的公共密钥/私有密钥对 (pk_i, sk_i) , 并且对随机质询字符串 $c_i \in \{0,1\}^k$ 进行采样。将 pk_i 添加到 S_i , 并且将 c_i 扩散到网络。
 - [0079] 2. **轮次 2:** 令 C_i 为从上一步骤接收的质询集合。找到随机数 $x_i \in \{0,1\}^k$, 使得 $h_i = H(pk_i || C_i || x_i) < d$, 其中 d 为难度参数。将元组 $ID_i = (x_i, h_i, pk_i, C_i)$ 扩散到网络。
 3. **轮次 3:** 在从 P_j 接收元组 $ID_j = (x_j, h_j, pk_j, C_j)$ 时, 如果 $h_j = H(pk_j || C_j || x_j)$ 并且 $c_i \in C_j$, 则将 pk_j 添加到 S_i 。
-

[0080] 接下来,将讨论该阶段的属性。首先,每个公共密钥的长度可以为 $\Theta(\log n)$ 位,因此,由每个诚实节点发送的位的总数可以为 $\Theta(kn + n \log n) = \tilde{O}(n)$,其中 \tilde{O} 符号隐藏了对数因子。其次,身份集合 S_1, \dots, S_n 在它们所包含的公共密钥中可能不一致。这可能是因诚实为敌对方可以选择将其公共密钥(以及对PoW的解决方案)选择性地发送到诚实节点的子集,而不是其他节点,因此,将诚实节点在其各自的身份集合中对该公共密钥的成员身份的观点进行分区。以下引理可以在身份生成阶段结束时建立集合 S_i 的属性。

[0081] 引理1:令 $S_h = \{pk_i | P_i \text{为诚实}\}$ 。则在身份生成阶段结束时,对于每个诚实节点 P_i , $S_h \subseteq S_i$ 。这可能是由于以下事实:由诚实节点扩散的每个消息可以被所有其他诚实节点接收,而在发送该消息的轮次结束之前不会发生任何损坏。推论1:令 P_i 和 P_j 为两个诚实节点,在算法结束时分别具有集合 S_i 和 S_j 。令 $pk \in S_i \setminus S_j$ 。那么 pk 可以属于不诚实节点。

[0082] 引理2:令 P_i 和 P_j 为两个诚实节点,在算法结束时分别具有集合 S_i 和 S_j 。那么对于所有 $pk_1 \in S_i \setminus S_j$ 和 $pk_2 \in S_j \setminus S_i$,可以是 pk_1 和 pk_2 可以属于系统中的两个不同节点。由于每个敌对方受控节点在每个轮次中只能计算一个成功的PoW,因此该节点不可能在同轮次中同时利用 pk_1 和 pk_2 来计算PoW。因此,这两个公共密钥可以是来自系统中的两个不同节点。推论2:对于每个不诚实节点 P_i ,诚实节点可以在其身份集合中包括至多一个公共密钥 pk_i 。

[0083] 推论3:令 $S_n = \bigcap_{P_i:\text{诚实}} S_i$ 。则 S_n 可以包含所有诚实节点的公共密钥,并且每个不诚实节点包含至多一个公共密钥。推论4:令 $S_u = \bigcup_{P_i:\text{诚实}} S_i$ 。则 S_u 可以包含所有诚实节点的公共密钥,并且每个不诚实节点包含至多一个公共密钥。

[0084] 建立了有关身份集合 S_i 的这些属性后,根据推论3和推论4中使用的符号,问题陈述可以是在满足 $S_n \subseteq S \subseteq S_u$ 的公共密钥的集合 S 上达成共识。在下面的阶段II部分中讨论了在此类集合 S 上达成共识。

[0085] 图4示出根据本发明实施方案的生成身份集合的方法。将在包括多个节点的系统的上下文中描述图4中所示的方法。然而,应当理解,本发明可以应用于其他情况(例如,在

希望防止Sybil攻击的系统或网络中,等等)。虽然以特定顺序示出了步骤,但是应当理解,本发明的实施方案可以包括具有以不同顺序进行的步骤的方法。另外,可以省略或添加步骤,并且这些步骤仍然可以在本发明的实施方案内。

[0086] 当初始化节点的网络时,可以发生步骤S402。每个节点可能不会存储与将在网络中的节点的总数相关的数据。在步骤S402处,多个节点中的每个节点可以生成公共密钥、私有密钥和随机质询字符串。本领域的技术人员将理解,可以使用适用于本发明的任何方法来生成公共密钥/私有密钥对。如本文所述,随机质询字符串可以随机生成,并且可以具有集合 $\{0,1\}^k$ 中的值,其中 k 是预先确定的安全性参数。安全性参数 k 可以指示值的强度,即,较大的安全性参数 k 可以导致较大的随机质询字符串。随机质询字符串可以包括 k 位。多个节点中的每个节点可以生成不同的随机质询字符串。

[0087] 在步骤S404处,在生成公共密钥、私有密钥和随机质询字符串之后,多个节点中的每个节点可以将该随机质询字符串传输到多个节点中的其他节点。

[0088] 在步骤S406处,多个节点中的每个节点可以从多个节点中的其他节点接收多个随机质询字符串。每个节点可以从其他每个节点接收一个随机质询字符串。然而,在一些情况下,恶意节点可以将随机质询字符串传输到一些节点,而不传输到其他节点。由于这种恶意活动,每个节点可以接收不同数量的随机质询字符串。

[0089] 在步骤S408处,多个节点中的每个节点可以生成包括多个随机质询字符串的质询集合。例如,在一些实施方案中,每个节点可以通过级联每个所接收的质询字符串来生成质询集合。

[0090] 在步骤S410处,在生成质询集合之后,多个节点中的每个节点可以确定求解工作量证明的随机数。对工作量证明的输入可以包括公共密钥、质询集合和随机数,如本文所述,例如, $h_i = H(pk_i || C_i || x_i) < d$ 。在一些实施方案中,确定求解工作量证明的随机数 x_i 还可以包括确定哈希值 h_i ,该哈希值是工作量证明的输出。每个节点可以确定通过计算哈希函数 H 得出的哈希值 h_i 。然后,每个节点可以确定哈希值 h_i 是否小于预先确定的难度参数 d 。如果哈希值 h_i 小于预先确定的难度参数 d ,则节点已确定可接受的随机数 x_i 。如果哈希值 h_i 大于预先确定的难度参数 d ,则节点可以用不同的随机数 x_i 重新计算哈希函数 H 。在一些实施方案中,节点可以以任何合适的次数重新计算哈希函数 H 。例如,节点可以计算哈希函数 H 20次、100次、500次或1000次。

[0091] 每个节点可以通过确定正确的随机数 x_i 来确定小于预先确定的难度参数 d 的哈希值 h_i 。恶意节点将无法确定求解多个可能不同的哈希函数的许多随机数 x_i 。由于求解哈希函数 H 的随机数 x_i 的计算,恶意节点无法创建许多身份,因为其计算能力有限并且无法计算许多不同的工作过程证明,从而防止Sybil攻击。在恶意节点伪造随机数 x_i 并且试图使用伪造的随机数获得附加身份的情况下,诚实节点可以确定伪造的随机数不利用其他合适的输入正确地求解哈希函数,以确定小于难度级别的哈希值。诚实节点可以验证随机数是伪造的随机数还是正确的随机数,如以下步骤S416中所述。

[0092] 在步骤S412处,在确定随机数之后,每个节点可以将元组传输到多个节点中的其他节点。元组可以包括公共密钥、随机数、质询集合和哈希值。这样,元组可以包括对哈希函数的输入(即,公共密钥、随机数和质询集合),以及哈希函数的输出(即,哈希值)。接收元组的第二节点可以验证对哈希函数的输入是否导致哈希函数的输出,因此,第二节点可以验

证该节点正确执行了工作量证明。在步骤S414处,多个节点中的每个节点可以接收多个元组。给定节点可以从多个节点中的不同节点接收多个元组中的每个元组。

[0093] 在接收多个元组之后,在步骤S416处,多个节点中的每个节点可以验证多个元组。节点可以通过确定与元组相关联的公共密钥、随机数和质询集合是否求解哈希函数,使得与该元组相关联的所得哈希值小于预先确定的难度参数 d 来验证该元组。如果节点验证元组,则该节点可以将与该元组相关联的公共密钥存储在身份集合中。如果未验证元组,例如,如果所得哈希值大于预先确定的难度参数 d ,则节点可以确定不在身份集合中存储与该元组相关联的公共密钥。每个节点都可以验证其已接收的每个元组。在每个节点验证每个元组之后,每个节点处的身份集合可以针对每个诚实节点包含一个身份,然而每个节点可以存储不同的身份集合。下面的阶段II描述了节点如何在相同的身份集合上达成共识。

[0094] IV. 阶段II

[0095] 在该部分中,将描述第二阶段。首先,将描述一些用于在节点的身份集合 S 及其问题上达成共识的稻草人方法。然后,将描述根据本发明实施方案的方法。

[0096] A. 稻草人解决方案

[0097] 第一稻草人解决方案可以包括本地身份生成。第一稻草人解决方案根据基于剪切-选择技术达成共识的想法。该稻草人解决方案的主要思想可以如下:(1) 每个节点 P_i 可以从其身份集合 R_i 中随机选择公共密钥集合 S_i ;(2) 每个节点 P_i 可以查询其公共密钥为 $pk_j \in R_i$ (通过用这些密钥对消息进行签名,使得只有接收方可以恢复内容并且知道该消息是针对它们的)的每个节点 P_j ,并且可以请求获得 S_j ;(3) 一旦获得 S_j ,节点 P_i 就可以计算出到目前为止 P_j 尚未看到(但 P_i 已经看到)的公共密钥的集合 $T_{i,j} = S_i \setminus S_j$;(4) P_i 可以从 S_i 中移除 $T_{i,j}$ 中的每个ID。需注意,由于上述推论1,(4)可能是有效的。

[0098] 虽然此方法从 S_i 中移除了一些属于不诚实节点的公共密钥,但有两个重要的点值得注意。(1) 由于诚实节点仅从ID的集合中对其子集进行采样来验证,因此仍然有可能未从最终集合中移除一些不诚实的ID。此外,在此过程之后获得的最终集合可能仍然不一致,因此,该方法无法解决问题,除非是以一定概率。一种解决方案应该以高概率起作用,但在此处并非如此。(2) 即使通过该方法靠运气达成了共识,达成一致的集合(例如, $S' \subseteq S_n \subseteq S' \subseteq S$),因此该方法与所需方法相比解决更强的共识问题。最多可以容忍每个不诚实节点的一个ID,而如果敌对方选择将其传递保留给一些诚实节点,则 S' 不太可能包含此类ID。

[0099] 接下来,将讨论另一种稻草人解决方案。该稻草人解决方案包括在每个节点上的Bracha广播。每个诚实节点 P_i 可以针对其集合 S_i 运行可靠的广播,参见[Gabriel Bracha, "Asynchronous Byzantine agreement protocols",《Information and Computation》,75(2):130-143,1987年11月]。然后,每个节点可以通过采用作为此广播的一部分接收的所有集合的并集来构造解决方案。 P_i 在此过程开始时, P_i 只能通过属于 S_i 的ID参与广播实例。此过程在计算上可能非常昂贵,每个节点每轮发送 $\tilde{O}(n^2)$ 位。此外,由不诚实节点运行的广播实例无法保证对其输出的共识,因此,即使这种方法也无法提供对问题的解决方案。

[0100] 另一种稻草人解决方案可以包括集合的并集。该稻草人解决方案可以依据推论4,并且诚实节点会仔细计算其集合的并集以在 S_U 上达成一致。然而,为了防止不诚实节点将SybilID包含在其集合中,可以运行某种形式的共识算法,以在待采用并集的集合上达成一

致。此类共识本质上需要准确标识由敌对方控制的节点(至少知道哪些节点在模棱两可并且将SybilID添加到其本地集合中)。因此,每个节点都需要对其他每个节点的集合的知识,然后对于这些集合的并集中的每个ID,将需要分别运行共识以检查多数节点是否在其集合中包含该ID。此类算法的计算量非常大,要求每个参与方在每个轮次中发送 $O(n^4)$ 位,这使其在实践和预期应的用中完全不可行。然而,根据本发明实施方案的方法可以通过某种领导者选举机制针对共识问题仔细选举领导者。然后,可以减少此带宽,并且系统可以在更少的轮次中达成共识。下一部分中将详细说明根据本发明实施方案的方法。

[0101] B.阶段II(完全一致)

[0102] 该过程的第二阶段可以被称为处处身份一致。获得本地集合 S_i 后,将通过根据Ren等人,参见[Ling Ren、Kartik Nayak、Ittai Abraham和Srinivas Devadas,“Practical synchronous byzantine consensus”,CoRR,abs/1704.02397,2017年],采用的协议寻求一致,该协议是用于同步通信网络中多值输入的期望的恒定轮次共识算法。该协议假定随机的领导者选举阶段作为前导码,其结果的正确性可取决于该阶段。在本发明的一些实施方案中,由于可以不假定节点共享任何随机位或信标,因此执行该随机领导者选择是新的质询。如上所述,根据本发明实施方案的方法可以通过使用和建立在第一阶段中建立的部分一致来克服这一点。

[0103] 在该部分中,将讨论共识阶段。共识阶段可以允许诚实节点在ID的集合 S 上达成一致。在第一个阶段中获得本地集合 S_i 之后,如本文所述,节点可以寻求一致。类似的算法可以来自Ren等人的最新著作[Ling Ren、Kartik Nayak、Ittai Abraham和Srinivas Devadas,“Practical synchronous byzantine consensus”,CoRR,abs/1704.02397,2017年]。他们论文中的算法是针对同步通信网络上多值输入的期望的恒定轮次共识算法;这不是本发明的实施方案所执行的操作,因为他们的算法假定随机的领导者选举阶段作为前导码,并且结果的正确性严重依赖于该阶段。而在根据本发明实施方案的方法中,由于不假定节点共享任何随机位或信标,因此执行该随机领导者选择是质询。如上所述,本发明的实施方案可以通过巧妙地使用由第一阶段中的节点计算出的针对PoW的解决方案来克服该质询。

[0104] 图5示出根据本发明实施方案的例示共识过程的流程图。虽然以特定顺序示出了步骤,但是应当理解,本发明的实施方案可以包括具有以不同顺序进行的步骤的方法。另外,可以省略或添加步骤,并且这些步骤仍然可以在本发明的实施方案内。

[0105] 在步骤1之前,节点可以将将在第一阶段中确定的本地集合 S_i 分到多个存储桶中。例如,在一些实施方案中,本地集合 S_i 可以被分到 n_i 个存储桶中,其中 $n_i \leftarrow 2^{\lceil \log |S_i| \rceil}$ 。存储桶可以是身份集合 S_i 的分区。每个节点可以确定将本地集合 S_i 分到多少个存储桶。可以为 n_i 选择此类值,而不仅仅是 $|S_i|$,因为此类值可以允许所有诚实节点为它们对系统中节点总数的估计计算约相同的值。与轮次数量相比,其估计中的误差可能较小,因此可以容忍(参考引理4)。以下引理帮助确立这一点。

[0106] 引理3:对于分别具有本地集合 S_i 和 S_j 的任何两个诚实节点 P_i 和 P_j ,令

$$n_i = 2^{\lceil \log |S_i| \rceil} \text{ 和 } n_j = 2^{\lceil \log |S_j| \rceil}。 \text{ 假定 } f < n/3, \text{ 则 } \left| \frac{1}{n_i} - \frac{1}{n_j} \right| \leq \frac{3}{n}。$$

[0107] 在不失一般性的情况下,假定 $|S_i| \leq |S_j|$,使得 $n_i \leq n_j$ 。则

$$\begin{aligned}
\frac{1}{n_i} - \frac{1}{n_j} &= \frac{1}{2^{\lceil \log |S_i| \rceil}} - \frac{1}{2^{\lceil \log |S_j| \rceil}} \\
&= \frac{1}{2^{\lceil \log |S_j| \rceil}} \left(\frac{1}{2^{\lceil \log |S_i| \rceil - \lceil \log |S_j| \rceil}} - 1 \right) \\
[0108] \quad &\leq \frac{1}{|S_j|} (2^{\lceil \log |S_j| \rceil - \lceil \log |S_i| \rceil} - 1) \\
&\leq \frac{1}{|S_j|} (2^{(\log |S_j| + 1 - \log |S_i|)} - 1) \\
&= \frac{1}{|S_j|} \left(\frac{2|S_j|}{|S_i|} - 1 \right)
\end{aligned}$$

[0109] 现在,根据引理1, $|S_j| \geq n-f = 2n/3$ 。这还可以适用于 $|S_i|$ 。此外,根据推论2, $|S_j| \leq n$ 。因此,替换这些值:

$$[0110] \quad \frac{1}{n_i} - \frac{1}{n_j} \leq \frac{3}{2n} \left(\frac{6n}{2n} - 1 \right) = \frac{3}{n}.$$

[0111] 在由每个节点执行分桶之后,从每个存储桶中,节点可以将PoW值最小的身份确定为该存储桶的领导节点。例如,节点可以基于哈希值来确定领导节点。在轮次中可以对每个存储桶进行处理。因此,从第一存储桶开始,可以运行与存储桶的领导节点的同步共识协议的多轮,以在S上达成一致。可以使用所有 S_i 的多数并集来计算该 $|S|$,这实际上意味着在S中添加属于多数 S_i 的公共密钥。在一些实施方案中,如果协议在轮次中没有成功,则该共识协议可以与下一个存储桶的领导节点重复。可以针对每个存储桶重复执行此操作,直到共识协议成功为止,这可以以 κ 中的高概率在期望的恒定轮次数量中发生,并且以 n 中的高概率在期望的对数(在 n 中)的许多轮次中发生。

[0112] 为了确定领导节点,每个节点可以将哈希值传输到多个节点中的每个其他节点。在一些实施方案中,每个节点可能已经具有来自阶段I中所述的每个其他节点的(在元组中接收的)哈希值。然后,每个节点可以确定从多个节点接收的多个哈希值中的最小哈希值。

[0113] 由于元组包含哈希值和公共密钥两者,因此每个哈希值与一个公共密钥相关联。公共密钥可以是节点的身份。每个节点可以确定与最小哈希值相关联的身份,并且可以选择与最小哈希值和身份相关联的节点(例如,第二节点)作为该轮的领导节点。在一些实施方案中,节点可以确定它是领导节点。

[0114] 在图5的步骤1处,在节点确定领导节点之后,节点可以将其自身的本地身份集合 S_i 散布到网络。每个节点都可以向网络中的每个其他节点传输或广播其本地身份集合 S_i 。例如,第一节点502可以向第二节点504和第三节点506传输第一本地身份集合 S_1 。第二节点504可以向第一节点502和第三节点506传输第二本地身份集合 S_2 。第三节点506可以向第一节点502和第二节点504传输第三本地身份集合 S_3 。

[0115] 在步骤2处,在从多个节点中的每个节点接收身份集合之后,每个节点可以创建身份集合的集合 \mathcal{S}_i 508。身份集合的集合 \mathcal{S}_i 508可以包括节点从其他节点接收的身份集合 S_i 。在一些实施方案中,每个节点可以在身份集合的集合 \mathcal{S}_i 508中包括它们自身的身份集合 S_i 。

例如,第一节点502可以创建身份集合 \mathcal{S}_1 的第一集合。身份集合 \mathcal{S}_1 的第一集合可以包括第一身份集合 S_1 、第二身份集合 S_2 和第三身份集合 S_3 。

[0116] 在创建身份集合的集合 \mathcal{S}_i 508之后,每个节点可以确定多数集合 T_i 510。多数集合 T_i 510可以是包括公共密钥的集合,这些公共密钥出现在身份集合的集合 \mathcal{S}_i 508中的身份集合 S_i 的至少一半(即, $n/2$)中。每个节点可以确定在身份集合 \mathcal{S}_i 的集合中的本地身份集合 S_i 的至少一半中出现的公共密钥。例如,第一节点502可以确定多数集合,该多数集合包括在至少一半的身份集合中出现的身份。如果这三个公共密钥中的每一个出现在身份集合 \mathcal{S}_1 的第一集合中的身份集合 S_i 的至少一半中,则第一节点502可以创建包括 Pk_1 、 Pk_2 和 Pk_3 的多数集合。

[0117] 在步骤3处,在确定多数集合 T_i 510之后,领导节点可以确定将其本地身份集合 S_i 设置为等于其多数集合 T_i 510。在其他实施方案中,领导节点可以将多数集合 T_i 510分配给其本地身份集合 S_i 508。例如,如果领导节点是第二节点504,则第二节点504可以将 T_2 分配给 S_2 (例如, $S_i \leftarrow T_i$)。领导节点然后将多数集合 T_i 510散布到网络。例如,第二节点504(领导节点)可以将多数集合 T_2 传输或以其他方式广播到第一节点502和第三节点506。如果节点不是领导节点,则该节点可以从该轮次的领导节点接收多数集合 T_j 。

[0118] 在步骤3之后,不是领导节点的每个节点可以验证所接收的多数集合 T_j 中的每个公共密钥是否出现在身份集合的集合 \mathcal{S}_i 508中的身份集合 S_i 的至少一半中。如果节点验证所接收的多数集合 T_j ,则每个节点可以确定将其本地身份集合 S_i 设置为等于所接收的多数集合 T_j 。节点可以基于领导节点的多数集合来更新其身份集合。这可以被称为最终身份集合。

[0119] 在节点确定最终身份集合之后,该节点可以知道哪些其他节点是诚实节点。该节点然后可以与其他诚实节点进行交互。例如,在一些实施方案中,该节点可以与身份在最终身份集合中的节点通信。这两个节点可以经由任何合适的加密协议安全地通信,并且知道接收方节点是诚实的。

[0120] 由每个诚实节点所存储的最终身份集合确定的诚实节点可以执行由多个诚实节点执行的任何合适的过程。在一些实施方案中,均存储包括另一方的身份的最终身份集合的两个节点可以执行交互,诸如交易。例如,两个节点可以在知道它们正在与诚实节点执行交易的情况下执行交易。第一节点可以联系第二节点以执行交易。第一节点和第二节点可以在节点以及任何其他商品和/或服务之间待转移的金额上达成一致。

[0121] 在其他实施方案中,具有包括在最终身份集合中的身份的节点可以投票。例如,每个诚实节点可以给新领导节点投票以使其领导进一步的处理。

[0122] 在一些实施方案中,在确定最终身份集合之后,节点可以创建包括交易、合同、事件和/或其他合适数据的区块链。节点可以读取和/或写入区块链。每个节点可能能够维护区块链的副本,并且可以验证存储在区块链上的先前区块。例如,多个节点可以接收交易,并且可以验证该交易是否有效(例如,节点可以验证是否存在足够的资金用于该交易)。多个节点能够对是否接受交易进行投票。如果多个节点投票接受交易,则多个节点可以将交易写入区块链上的区块。

[0123] 下面的协议2描述了如上面在图5中所述的抗Sybil攻击身份的生成协议的阶段II

的示例性伪代码。

协议 2: 抗 Sybil 攻击身份的生成

参与方 P_i 执行以下步骤:

阶段 II: 处处身份一致

- [0124] 1. $n_i \leftarrow 2^{\lceil \log |S_{i1}| \rceil}$
2. **领导者选举。** 对于每个 $k \in \{0, \dots, n_i - 1\}$, 从 S_i 中选取具有最小 h_j 的身份 $ID_j = (x_j, h_j, pk_j, c_j)$, 使得 $\frac{h_j}{d} \in \left[\frac{k}{n_i}, \frac{k+1}{n_i} \right)$ 。将 ID_j 作为 $k + 1$ 轮次的领导节点。
3. **共识。** 针对每一轮次 $r \in \{1, \dots, \lceil \log n_i \rceil\}$,
- (a) 将 S_i 散布到网络。
- (b) 令 S_i 为节点 P_i 在运行上一步之后从网络接收的身份集合的收集。也将 S_i 添加到 S_i 。同样, 令 T_i 为在 S_i 中的超过 $n/2$ 的集合中出现的所有公共密钥的集合。
- [0125] (c) 如果 P_i 是领导节点, 则 $S_i \leftarrow T_i$, 并且散布 T_i 。否则, 它从领导节点接收 T_j 。
- (d) 如果 T_i (或 T_j) 中的每个公共密钥在 S_i 中的至少 n 集合中出现, 则 $S_i \leftarrow T_i$ (或 T_j)。
4. 输出 S_i 。
-

[0126] 在图6中进一步描述了协议2, 其中示出了根据本发明实施方案的执行共识过程的方法。如本文所述, 将在已经执行阶段I的系统的背景中描述图6中所示的方法。然而, 应当理解, 本发明可以应用于其他情况 (例如, 确定对数据集达成共识的系统等)。虽然以特定顺序示出了步骤, 但是应当理解, 本发明的实施方案可以包括具有以不同顺序进行的步骤的方法。另外, 可以省略或添加步骤, 并且这些步骤仍然可以在本发明的实施方案内。

[0127] 在步骤S602处, 多个节点中的每个节点可以执行身份集合生成过程, 如本文所述。在步骤S604处, 在执行身份集合生成过程之后, 每个节点可以存储身份集合。多个节点中的每个节点可以确定领导节点。在一些实施方案中, 如本文所述, 节点可以通过首先确定身份集合要被分到的存储桶的数量来确定领导节点。在将身份集合分到存储桶中之后, 每个节点可以确定该存储桶中的身份集合的多个哈希值中的最小哈希值。然后, 每个节点可以确定与最小哈希值相关联的身份集合的身份。在一些实施方案中, 节点可以确定它是领导节点。在其他实施方案中, 一个节点可以确定多个节点中的另一个节点是领导节点。

[0128] 节点可以使用最小哈希值来确定领导节点, 因为该最小哈希值是每个节点可以确定的方便的值。然而, 应当理解, 节点可以使用不同的规则和/或逻辑来选择与领导节点相关联的特定哈希值, 诸如但不限于最大哈希值、中值哈希值, 等等。

[0129] 在步骤S606处, 在确定领导节点之后, 每个节点可以将身份集合扩散到多个节点。每个节点可以从多个节点中的每个其他节点接收一个身份集合。然而, 在一些情况下, 恶意节点可以将身份集合传输到一些节点, 而不是传输到其他节点。在这种情况下, 节点可以接收的身份集合的数量少于多个节点中的节点的数量。每个节点可以创建包括每个所接收的身份集合的身份集合的集合。可以使用本文所述的任何合适的方法来构造身份集合的集合。

[0130] 在步骤S608处, 多个节点中的每个节点可以确定多数集合, 该多数集合包括在身

份集合的集合中的至少一半的身份集合中出现的身份。通过接收多个身份集合并且确定多数集合,节点可以确定多个节点中至少1/2的哪些身份被认为是诚实节点。

[0131] 在一些实施方案中,在步骤S608之后,领导节点可以将领导节点的多数集合扩散到多个节点。例如,一个节点(例如,领导节点)可以接收三个身份集合。第一身份集合可以包括来自节点1、2和3的身份。第二身份集合可以包括来自节点1和2的身份。第三身份集可以包括来自节点1、2和4的身份。节点可以确定包括节点1和节点2的身的多数集合,因为这些身份出现在所接收的身份集合中的至少一半中。然后,领导节点可以将多数集合传输到多个节点。多个节点中的每个节点可以接收领导节点的多数集合。

[0132] 在从领导节点接收多数集合之后,每个节点可以验证领导节点的多数集合。每个节点可以验证所接收的多数集合中的每个身份(即,公共密钥)是否在由该节点存储的身份集合的集合中的身份集合的至少一半中出现。如果节点验证领导节点的多数集合,则节点可以基于领导节点的多数集合更新其自身的身份集合。

[0133] 现在,每个节点可以存储轮次的多数集合,其中包括诚实节点的身份。每个节点可以针对先前通过将身份集合分到存储桶而确定的每个轮次重复先前的步骤。

[0134] C. 安全性分析

[0135] 接下来,将讨论根据本发明实施方案的方法的安全性。在一些实施方案中,敌对方可能使不同的诚实节点相信给定轮次中谁是领导者的不同顺序。以下引理可以帮助确立诚实节点领导能力的相对顺序在所有诚实节点之间可以是一致的。

[0136] 引理4:令 P_i 是诚实节点,并且 $\sigma_i: \cup_{i:\text{诚实}} P_i \rightarrow \{0, \dots, n_i-1\}$ 是函数,鉴于 P_i ,如果哈希值为 $h_j \in \left[\frac{r}{n_i}, \frac{r+1}{n_i}\right)$,则该函数可以将每个诚实节点 P_j 分配给轮次 $r = \sigma_i(P_j)$ 。然后,在概率为 $1-o(1)$ 的情况下,对于每个其他诚实节点 P_j ,对于每个诚实公共密钥 P_k ,可以是 $\sigma_j(P_k) = \sigma_i(P_k)$ 。回想一下,对于所有诚实公共密钥 P_k ,扩散协议可以允许每个诚实节点接收相同的 h_k 值。因此,如果 $n_i = n_j$,则可以存在唯一的整数 m ,使得 $h_k \in \left[\frac{m}{n_i}, \frac{m+1}{n_i}\right]$ 。这可以暗示 $\sigma_j(P_k) = \sigma_i(P_k)$ 。

[0137] 现在,在不失一般性的情况下, $n_i < n_j$,这可以暗示 $\sigma_j(P_k) \leq \sigma_i(P_k)$ (因为对应于 n_i 的间隔可以更大)。令 $\varepsilon_{i,j,k}$ 对于诚实节点 P_i, P_j 是 $\sigma_j(P_k) \neq \sigma_i(P_k)$,并且 P_k ,使得 $P_i \neq P_j$ 。则

$$\begin{aligned} \Pr(\varepsilon_{i,j,k} | n_i < n_j) &= \sum_{m=0}^{n_i-1} \Pr(\varepsilon_{i,j,k} | n_i = m, n_i < n_j) \Pr(n_i = m) \\ [0138] \quad &= \frac{1}{n_i} \sum_{m=0}^{n_i-1} \Pr(\sigma_j(P_k) < \sigma_i(P_k) | n_i = m, n_i < n_j) \end{aligned}$$

[0139] 现在,因 $\sigma_j(P_k) < \sigma_i(P_k)$ 为真,因此可以是 $\frac{\sigma_j(P_k)+1}{n_j} > \frac{\sigma_i(P_k)}{m}$,这可以暗示

$$\frac{n_j}{m} < \frac{(\sigma_j(P_k)+1)}{\sigma_i(P_k)}。现在, \frac{n_j}{m} = \frac{2^{\lfloor \log |S_j| \rfloor}}{2^{\lfloor \log |S_i| \rfloor}} \geq \frac{|S_j|}{2|S_i|} \geq \frac{1}{3}, (因为(根据推论2) |S_j| \geq 2n/3 并且 |S_i|$$

$\leq n$ 。因此,因 $\varepsilon_{i,j,k}$ 为真, $\frac{(\sigma_j(P_k)+1)}{\sigma_i(P_k)} \geq \frac{1}{3}$ 可以为真,这可以暗示 $\sigma_j(P_k) \geq \frac{\sigma_i(P_k)}{3} - 1$ 。由于 $\sigma_j(P_k) \leq \sigma_i(P_k)$,因此这可以暗示 $\sigma_i(P_k) \geq \frac{3}{2}$,这可以与 $\sigma_i(P_k) \geq 2$ 的情况相同(因为 $\sigma_i(P_k)$ 可以是整数)。

[0140] (在领导节点可能是敌对的情况下)为了处理哪个节点是领导节点的观点上的不一致问题,PoW谜题的解决方案可以在其中具有足够的熵。在一些实施方案中,以上由每个节点计算的PoW的谜题可涉及由诚实节点新近生成的质询。

[0141] 在下面的引理中,将描述一个类比,以得出到达第一轮次所需的轮次数的边界,在该第一轮次中,所有诚实节点可以在哪个节点是领导节点上达成一致。此类轮次(被称为良好轮次)可以始终具有对领导节点的一致观点。良好轮次的特征可以表述为与该轮次相对应的间隔只可以具有诚实节点对PoW谜题的解决方案。所有诚实节点可以就唯一领导节点达成一致,并且由Ren等人提出的协议,参见[Ling Ren、Kartik Nayak、Ittai Abraham和Srinivas Devadas,“Practical synchronous byzantine consensus”,CoRR,abs/1704.02397,2017年],可以在这一轮次结束之前终止。因此,系统对ID的集合达成了共识。

[0142] 引理5:对于给定的常数 $\varepsilon \in (0, 1)$,考虑将 $n\varepsilon$ 个红球和 $n(1-\varepsilon)$ 个蓝球分别随机且均匀地丢入(标号为1到 n 的) n 个箱中。然后,在具有 n 中的高概率的情况下,在前 $O(\log n)$ 箱中可以存在只有红球的箱。 X_i 对于箱 i 仅包含红球的情况, X_i 可以是指标随机变量。则 $\mathbb{E}(X_i) = \left(1 - \frac{1}{n}\right)^{n(1-\varepsilon)} \left(1 - \left(1 - \frac{1}{n}\right)^{n\varepsilon}\right)$ 。现在,利用对于所有实数 x 为 $1+x \leq e^x$ 和(通过泰勒展开式) $\left(1 - \frac{1}{n}\right)^{n(1-\varepsilon)} \geq \varepsilon$ 的事实, $\mathbb{E}(X_i) \geq \varepsilon(1 - e^{-\varepsilon})$,这可以为常数。由于 X_i 可以是负相关的,所以标准的切尔诺夫界,参见[Devdatt Dubhashi和Desh Ranjan,“Balls and bins:A study in negative dependence”,《Random Structures and Algorithms》,13(2):99-124,1998年],可以围绕此平均值建立严格的集中度。

[0143] 对于 $\varepsilon > 2/3$ ($f < n/3$ 时也是如此),引理5指出,给定的箱仅包含红球的概率为约0.342。因此,期望每3个箱中的一个可以只包含红球,因此,可以期望前四轮包含算法的良好轮次,并且在具有 n 中的高概率的情况下,前 $\log(n)$ 轮也可以包括该良好轮次。对于上面描述的协议的任何给定轮次 r , $S_i^{(r)}$ 可以表示轮次 r 结束时带有诚实节点 P_i 的公共密钥集合。

[0144] 引理6:令 $S_h = \{pk_i | P_i \text{为诚实}\}$ 。在任何轮次 r 中,对于每个诚实节点 P_i ,可以是 $S_h \subseteq S_i^{(r)}$ 。 $pk \in S_h$ $pk \in S_h$ 可以是某个诚实节点(即, P_j)的公共密钥。根据引理1,对于所有诚实节点 $P_i, P_j \in S_i$ 。因此,当任何诚实节点 P_i 散布身份集合 S_i 时,该集合可以包含公共密钥 pk 。由于诚实节点的数量可以严格大于 $n/2$,因此,身份集合 S_i 的集合包含每个诚实节点 P_i 的公共密钥 pk 。因此,与轮回 r 的领导节点是谁无关,对于所有诚实节点 $P_i, pk \in S_i^{(r)}$ 。这可以暗示 $S_h \subseteq S_i^{(r)}$ 。

[0145] 引理7:(安全性)在任何轮次 r 中,由所有这些身份集合的并集形成的

$S^{(r)}$ $\stackrel{[Opt]}{def} = \bigcup_{i:诚实} S_i^{(r)}$ 可以满足 $S^{(r)} \subseteq \bigcup_{i:诚实} S_i$, 即, 它可以不包含新的ID (协议结束时任何诚实节点集合中都可以不包含该ID)。从引理6开始, 集合 $S^{(r)}$ 和 $\bigcup_{i:诚实} S_i$ 可以具有非空交集。在一些实施方案中, 当发生这种情况时, 可以存在公共密钥 $pk \in S^{(r)} \setminus \bigcup_{i:诚实} S_i$, 并且 r 可以是协议中最小的此类轮次索引。根据引理1, 该公共密钥 pk 可以属于敌对方。由于 $pk \in S^{(r)}$, 因此某些诚实节点 P_i 可以包含 $pk \in S_i^{(r)}$ 。然而, 由于 $pk \in \bigcup_{i:诚实} S_i$, 则 P_i 可以不包含 $pk \in S_i$ 。现在, 可以出现以下两种情况。

[0146] 在第一种情况下, P_i 可以是轮次 r 中的领导节点, 并且可能已将公共密钥 pk 添加到 $S_i^{(r)}$ 。为此, 公共密钥 pk 可能已经在轮次 r 中的身份集合 S_i 的集合中的超过 $n/2$ 的身份集合中出现。然而, 由于身份集合 S_i 的集合包含所有诚实节点的身份集合, 并且敌对身份集合的数量可以小于 $n/3$, 因此至少一个诚实节点 P_j 可以具有 $pk \in S_j^{(r-1)}$ 。 r 可以是最小的轮次索引, 其中公共密钥 pk 被添加到某个诚实节点的身份集合中, 这可能是矛盾的。

[0147] 在第二种情况下, 节点 P_i 可以不是轮次 r 中的领导节点, 并且该节点可能已经将公共密钥 pk 添加到 $S_i^{(r)}$ 。在这种情况下, P_i 在此轮次 r 中可能已经从领导者 P_j (在某些情况下可能是敌对的) 中接收多数集合 T_j , 使得 $pk \in T_j$ 和 pk 可以出现在身份集合 S_i 的集合中的至少 $3n/4$ 个身份集合中。根据与上述相同的推理, 这可以暗示至少一个诚实节点可以在 $r-1$ 轮次结束时包含公共密钥 pk , 这可能是矛盾的。

[0148] 引理8: 令 r 为第一良好轮次的索引。则对于每对诚实节点 P_i 和 P_j , 可以是 $S_i^{(r)} = S_j^{(r)}$ 。在一些实施方案中, $S_i^{(r)} \neq S_j^{(r)}$ 在第一轮次中 r 。 P_k 可以是诚实节点, 该诚实节点可以是轮次 r 的领导者。然后, 在不失一般性的情况下, 可以出现以下两种情况。

[0149] 在第一种情况下, $P_i = P_k$ 。在这种情况下, 为了使 $S_i^{(r)} \neq S_j^{(r)}$, 节点 P_j 可能已经从 P_i 接收了多数集合 T_i , 使得某些公共密钥 $pk \in T_i$ 可能不包含在身份集合 S_j 的集合中的 $n/2$ 的身份集合中。这可以暗示可能存在某些诚实节点 P_u , 该诚实节点可能在其集合 $S_u^{(r-1)}$ 中不包含公共密钥 pk , 这暗示公共密钥 pk 可能属于敌对方。然而, 由于 $pk \in T_i$, 节点 P_i 可以在身份集合 S_i 的集合中的至少 $n/2$ 的身份集合中包含公共密钥 pk 。这可以暗示至少一个诚实节点 P_v 可以在其集合 $S_v^{(r-1)}$ 中包含公共密钥 pk 。或者另选地, 在第二种情况下, $P_i, P_j \neq P_k$ 。

[0150] 引理9: (活性) 如果存储桶为空或包含至少一个不诚实节点, 则协议的共识部分可进行下一轮次 (即, 下一个存储桶)。针对存储桶可以存在以下四种情况: (1) 一个恶意领导者: 该恶意领导者可以遵循 Srini 的证明; (2) 一些诚实领导者和一些恶意领导者: 在最坏的情况下, 所有诚实节点可以继续到下一个存储桶; (3) 全部为恶意领导者: 因为某些恶意领导者可以诚实地执行动作, 因此可以从情况2中减少这种情况; (4) 空存储桶: 此情况可以被减少到情况1, 因为恶意领导者可以在该轮次中保持沉默, 也可以遵循 Srini 的证明。

[0151] 引理10: (完整性) 对于所有轮次 $r' > r$ 和每个诚实参与方 P_i , 可以是 $S_i^{(r')} = S_i^{(r)}$ 。这可以在 [Ling Ren, Kartik Nayak, Ittai Abraham 和 Srinivas Devadas, "Practical

synchronous byzantine consensus”, CoRR, abs/1704.02397, 2017年]中示出。

[0152] 定理1: (一致性) 在具有n中的高概率的情况下, 所有诚实节点在 $\lceil \log n \rceil$ 轮次后可能会输出相同的集合, 其中n是节点数。每个诚实节点都可以根据引理5随机生成质询字符串, 在具有n中的高概率的情况下, 可以存在只包含诚实节点的ID的存储桶。此类轮次可以始终具有对领导节点的一致观点。一旦系统开始这一轮次, 所有诚实节点可以就唯一领导节点达成一致, 在一些实施方案中, Renet等人提出的协议, 参见[Ling Ren、Kartik Nayak、Ittai Abraham和Srinivas Devadas, “Practical synchronous byzantine consensus”, CoRR, abs/1704.02397, 2017年], 可以允许所有诚实节点在此轮结束时在其本地身份集合上达成一致。最后, 根据引理7, 当协议终止时, 可以对节点之间的身份集合达成共识。

[0153] 定理2: (带宽) 由每个诚实节点发送的位总数在第一阶段为 $O(n^2)$, 在第二阶段为 $O(nd \log^2 n)$, 其中d可以是网络中任何节点的最大邻居(即, 周围节点)数量。定理的第一部分可以直接从上面的分析得出。对于第二部分, 需注意, 在每个轮次中, 每个节点可以向其他每个节点散布其节点的集合两次。此集合可以包含至少所有诚实节点的公共密钥列表, 因此, 大小可以为 $O(n \log)$ 。此外, 由于散布协议可以将消息转发给紧邻的邻居, 因此对于每个节点每轮总共发送 $O(nd \log^2 n)$ 位, 总共可以发送d个此类消息。此外, 根据本发明实施方案的一些方法可以处理扰动, 增加弹性, 处理自适应敌对方, 并且减少带宽。

[0154] V. 结论

[0155] 根据本发明实施方案的方法可以允许节点的集合在其他节点的身份集合上达成一致, 使得Sybil身份的数量被最小化。本发明的实施方案在简单性、带宽, 以及由节点执行的针对PoW谜题的解决方案的数量方面可以是有效的。终止可以以高概率发生在对数数量的轮次中。

[0156] 本发明的实施方案具有许多优点。例如, 节点可以确定所有诚实节点的身份集合, 而无需对最初可能想要加入系统的节点数量的知识, 也不需要初始可信设置。另外, 本发明的实施方案允许节点的动态到达和离开, 同时, 在每个轮次中保留所需的安全性和带宽。另一个优点是, 本发明的实施方案可以以恒定的轮次数执行。

[0157] 本申请中描述的任何软件组件或功能可被实现为要使用例如Java、C、C++、C#、Objective-C、Swift的任何合适的计算机语言或例如Perl或Python的脚本语言, 使用例如常规的或面向对象的技术由处理器执行的软件代码。软件代码可作为一系列指令或命令存储在计算机可读介质上以供存储和/或传递, 合适的介质包含随机存取存储器(RAM)、只读存储器(ROM)、例如硬盘驱动器或软盘的磁性介质, 或例如光盘(CD)或数字通用盘(DVD)的光学介质、闪存存储器等等。计算机可读介质可以是此类存储或传输设备的任何组合。

[0158] 此类程序还可以使用适应于经由包括互联网的符合多种协议的有线、光学和/或无线网络进行传输的载波信号来编码和传输。因此, 根据本发明的实施方案的计算机可读介质可以使用以此类程序编码的数据信号来创建。以程序代码编码的计算机可读介质可与兼容设备一起封装或与其他设备分开地提供(例如, 经由因特网下载)。任何此类计算机可读介质可以驻留于单个计算机产品(例如, 硬盘驱动器、CD或整个计算机系统)上或内, 且可存在于系统或网络内的不同计算机产品上或内。计算机系统可以包含用于将本文中所提及的任何结果提供给用户的监视器、打印机或其他合适的显示器。

[0159] 以上描述是示意性的不是限制性的。在所属领域的技术人员阅读了本公开后, 本

发明的许多变化将变得显而易见。因此,本发明的范围不应参考以上描述来确定,而是应参考待决的权利要求以及其完整范围或等效物来确定。

[0160] 在不脱离本发明的范围的情况下,任何实施方案的一个或多个特征可与任何其他实施方案的一个或多个特征组合。

[0161] 如本文所用,除非明确指示有相反的意思,否则“一个”、“一种”或“该”的使用旨在表示“至少一个”。

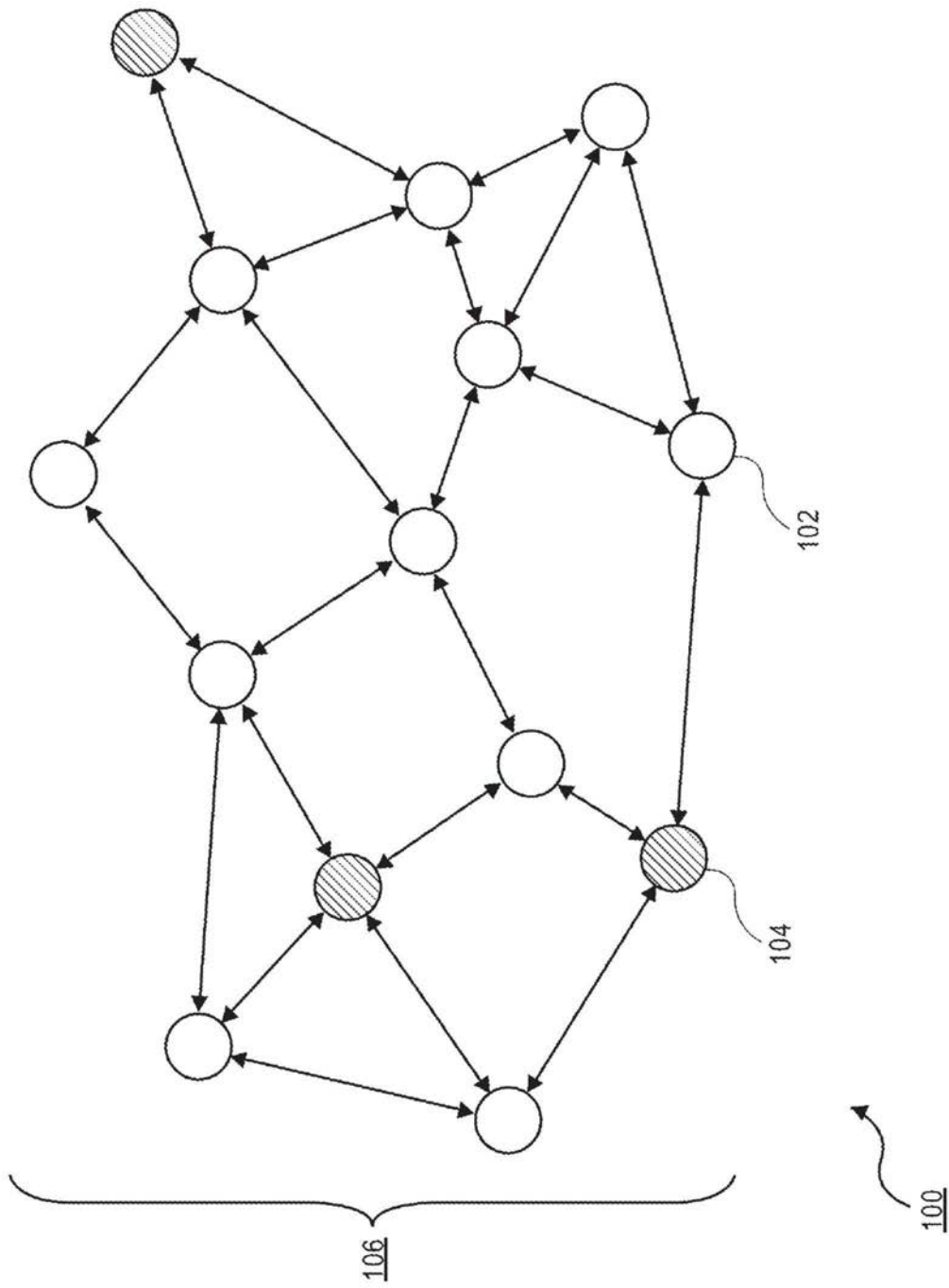


图1

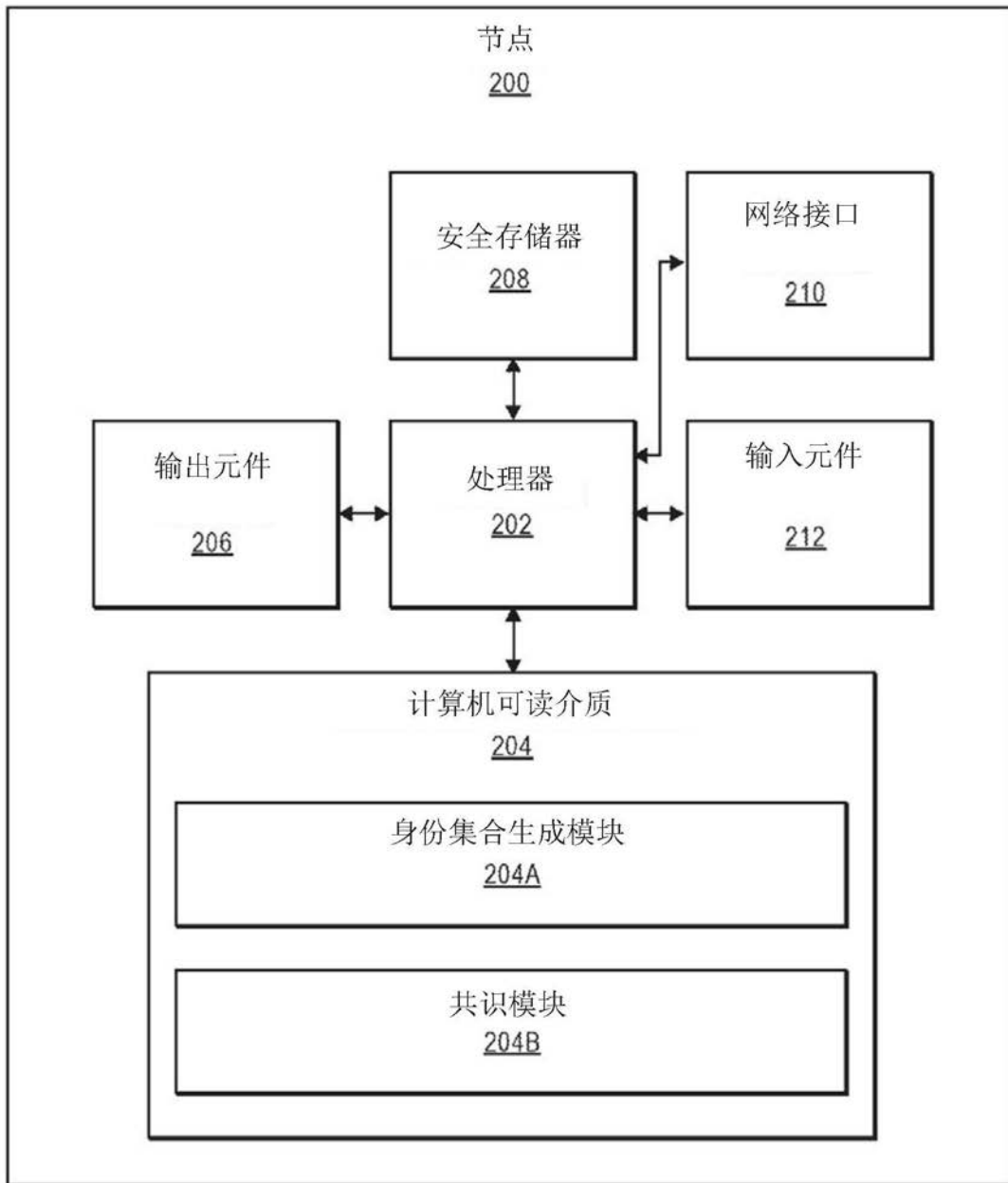


图2

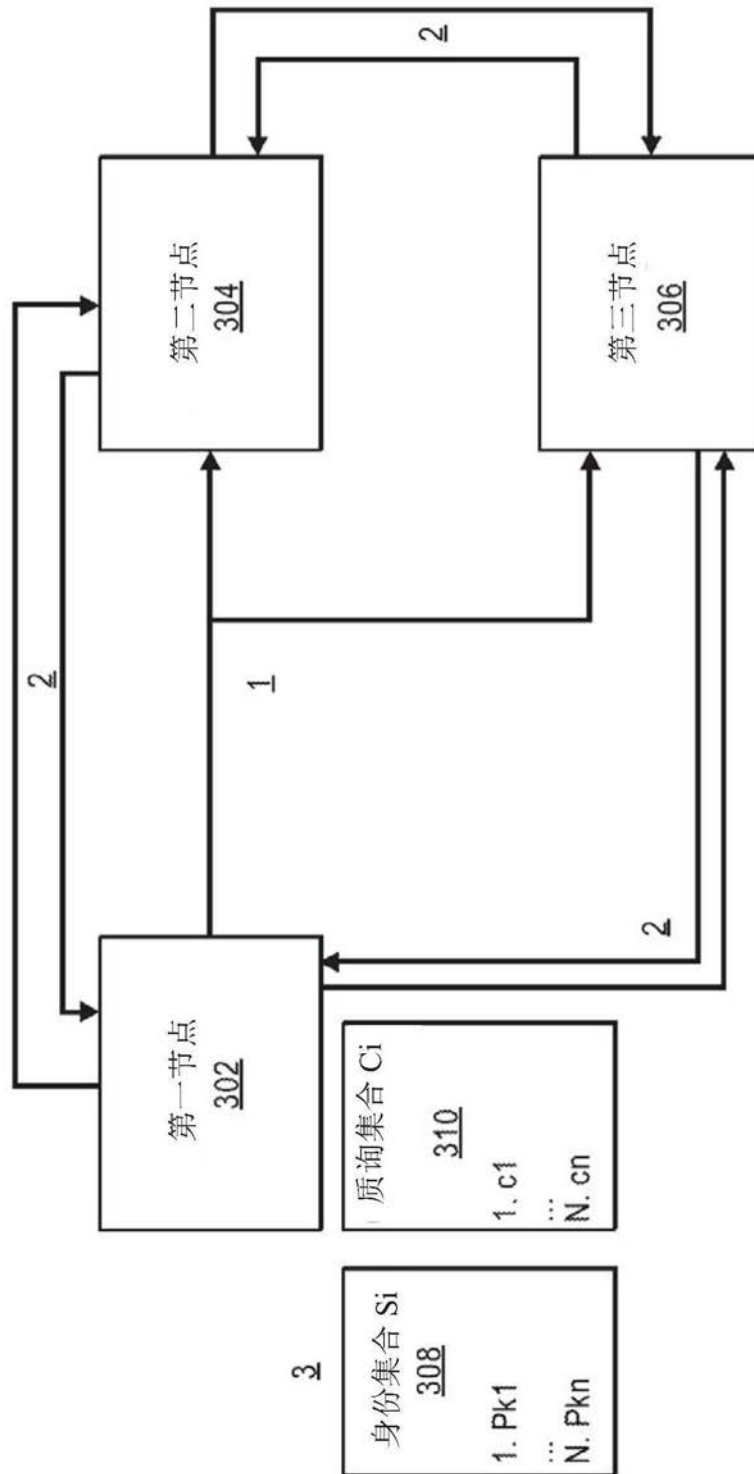


图3

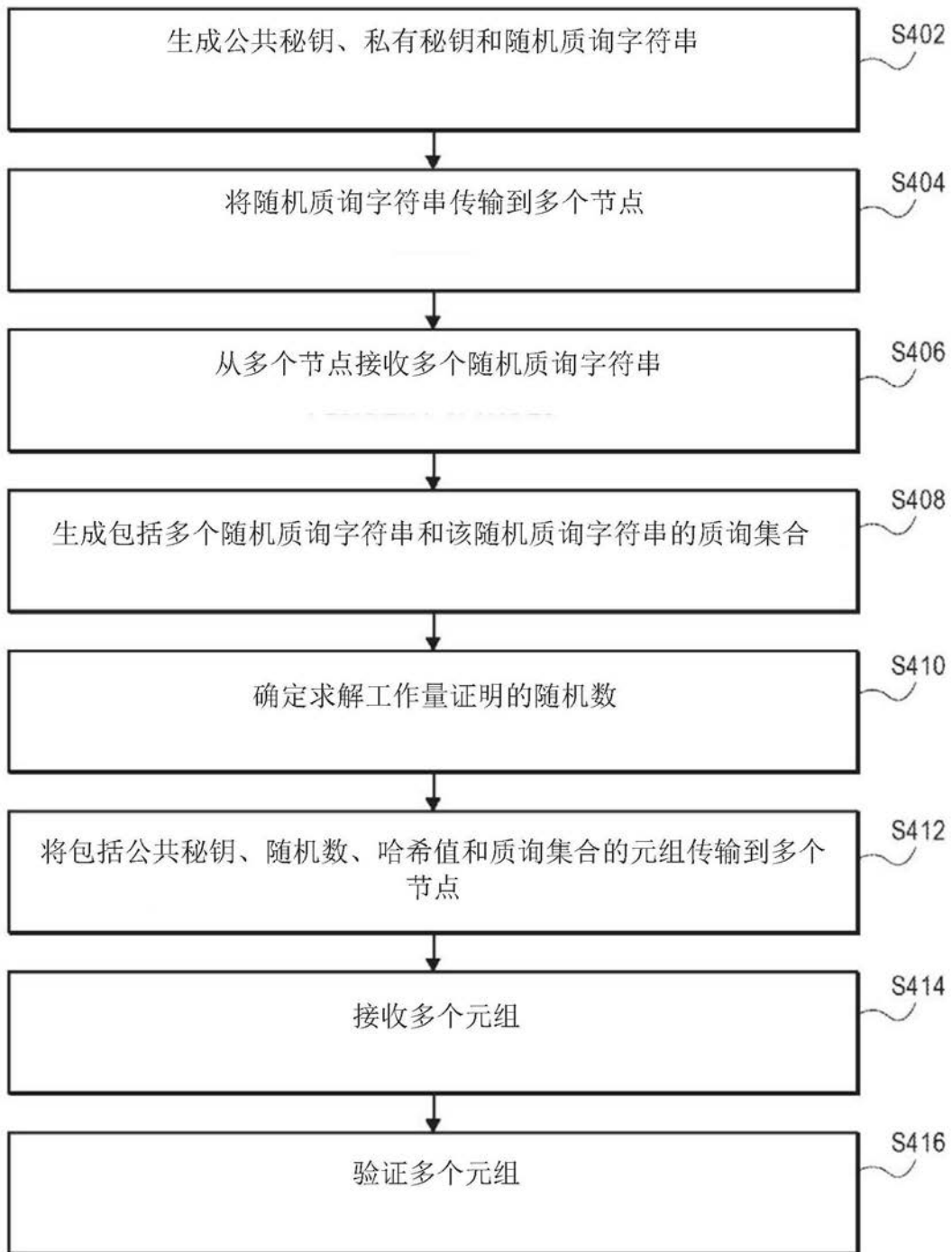


图4

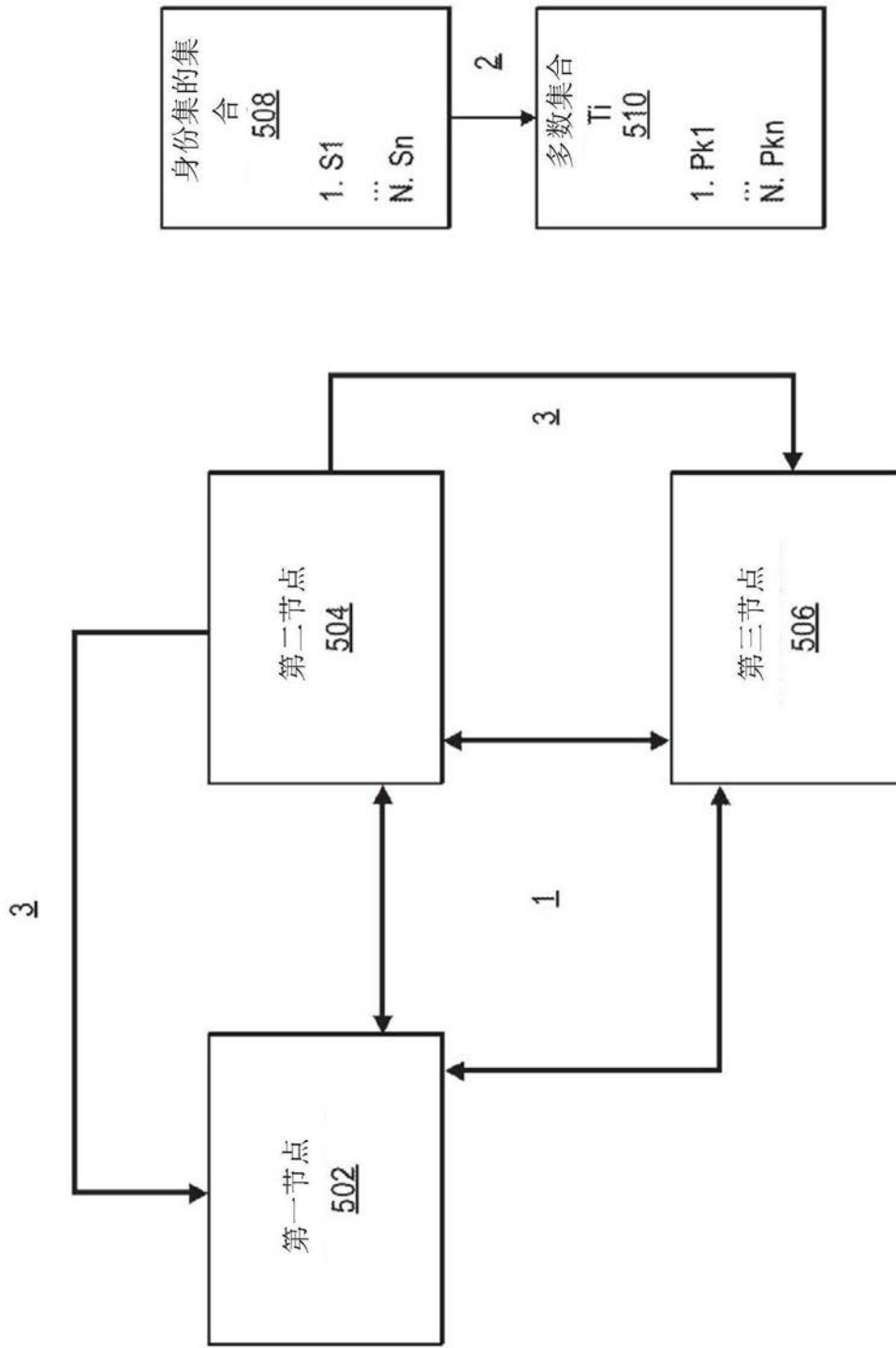


图5

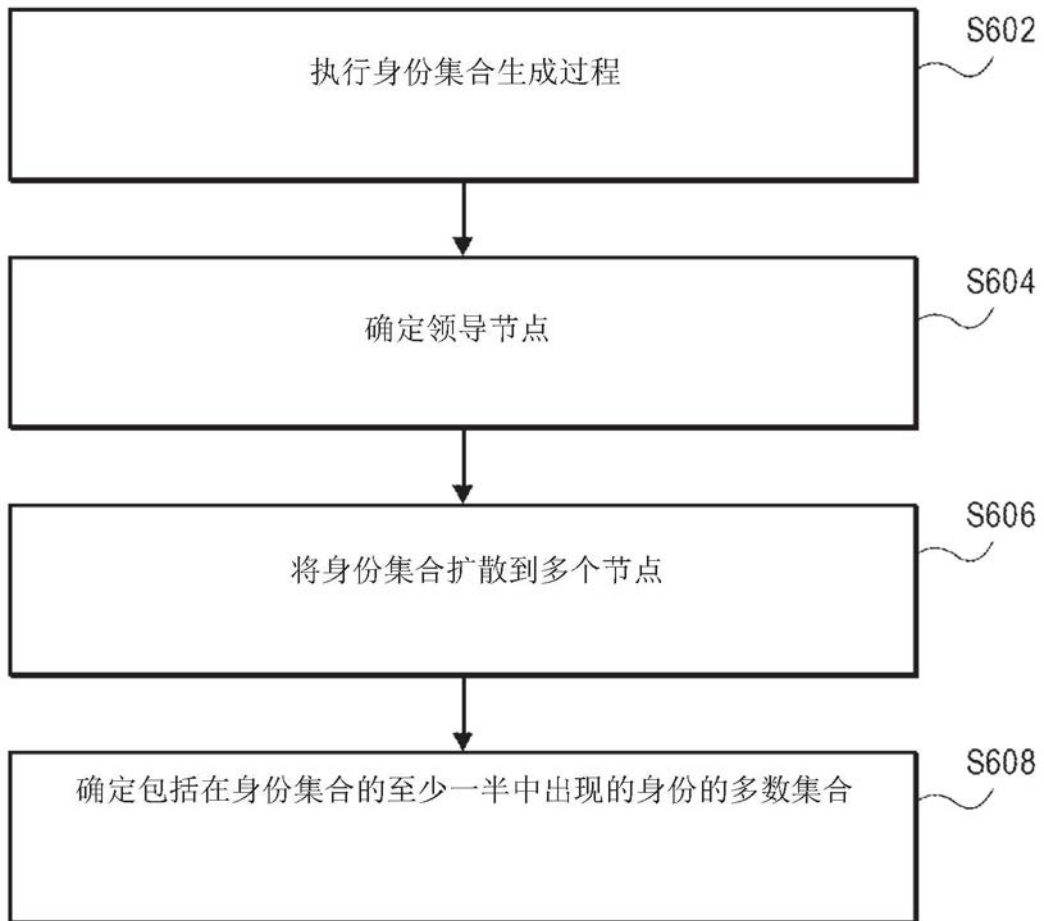


图6