



(12) 发明专利申请

(10) 申请公布号 CN 116235465 A

(43) 申请公布日 2023. 06. 06

(21) 申请号 202180066566.2

(22) 申请日 2021.08.31

(30) 优先权数据

2015477.9 2020.09.30 GB

(85) PCT国际申请进入国家阶段日

2023.03.28

(86) PCT国际申请的申请数据

PCT/EP2021/073986 2021.08.31

(87) PCT国际申请的公布数据

W02022/069135 EN 2022.04.07

(71) 申请人 区块链许可股份公司

地址 瑞士楚格

(72) 发明人 杰克·欧文·戴维斯

克雷格·史蒂文·赖特

(74) 专利代理机构 北京市竞天公诚律师事务所

11770

专利代理师 孙磊 徐民

(51) Int.Cl.

H04L 9/32 (2006.01)

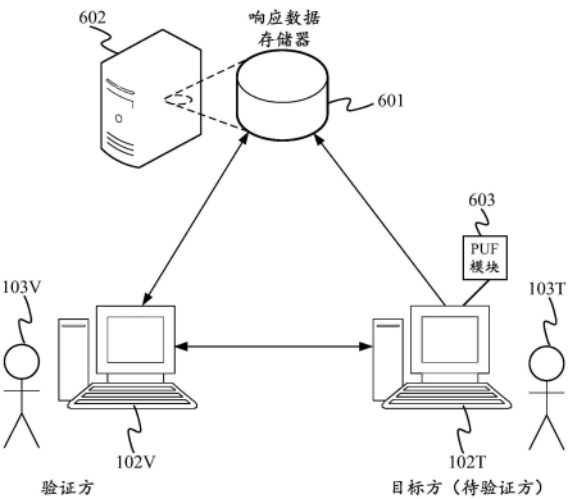
权利要求书4页 说明书46页 附图8页

(54) 发明名称

在数据存储器中存储响应值的物理不可克隆函数

(57) 摘要

一种用于使验证方能够验证目标的身份的方法,所述目标包括目标方或设备。所述方法包括,在建立阶段,由除所述验证方之外的一方:将一个或多个质询的集输入到包括物理不可克隆函数PUF的PUF模块中,以基于所述PUF生成一个或多个响应的相应集;以及,将所述响应集中的每个响应的相应响应数据记录存储在所述目标方或所述验证方的任何设备外部的数据存储器中,所述数据存储器是第三方计算机设备的一部分、或是公共对等发布介质。从而所述响应数据记录被提供给所述验证方,以在后续验证阶段验证所述目标的所述身份。



1. 一种计算机实现的方法,用于使一个或多个验证方能够验证目标的身份,所述目标包括目标方或所述目标方的设备;所述方法包括,在建立阶段,由除所述一个或多个验证方中的任一验证方之外的建立方:

将一个或多个质询的集输入到包括物理不可克隆函数PUF的PUF模块中,以基于所述PUF生成一个或多个响应的相应集;以及

将所述响应集中的每个响应的相应响应数据记录存储在所述目标方或所述一个或多个验证方的任何设备外部的数据存储器中,所述数据存储器是第三方计算机设备的一部分、或是公共对等发布介质,从而向所述一个或多个验证方提供所述响应数据记录,以在后续验证阶段验证所述目标的所述身份,其中每个响应的所述响应数据记录包括相应的响应或从中导出的数据。

2. 根据权利要求1所述的方法,其中所述建立方是所述目标方,所述方法由所述目标方使用所述目标方的计算机设备执行。

3. 根据权利要求2所述的方法,其中所述PUF模块集成在所述目标方的所述计算机设备中。

4. 根据权利要求2所述的方法,其中所述PUF模块是所述目标方的所述计算机设备的外部的外围设备。

5. 根据权利要求2至4中任一项所述的方法,其中所述数据存储器是所述第三方计算机设备的一部分,所述目标方从所述第三方接收所述质询集,输入到所述PUF模块的所述质询集是从所述第三方接收的所述质询集。

6. 根据权利要求1所述的方法,其中所述建立方是可信第三方。

7. 根据权利要求6所述的方法,其中所述数据存储器是所述第三方计算机设备的一部分,所述方法由所述可信第三方使用所述第三方计算机设备执行。

8. 根据权利要求7所述的方法,其中所述第三方计算机设备包括服务器设备,所述服务器设备包括位于一个或多个地理位置的一个或多个服务器单元。

9. 根据权利要求7或8所述的方法,其中所述PUF模块是所述第三方计算机设备的外部外围设备或可移动组件,所述方法包括:在所述建立阶段之后以物理方式将所述PUF模块发送给所述目标方,以使所述目标方能够在所述验证阶段响应来自所述验证方的质询。

10. 根据权利要求1至6中任一项所述的方法,其中所述数据存储器是对等发布介质。

11. 根据权利要求10所述的方法,其中所述对等发布介质是区块链。

12. 根据权利要求11所述的方法,其中所述区块链采用基于输出的事务模型,所述区块链包括多个事务,每个事务具有至少一个输入和至少一个输出,其中每个事务的所述输入指向另一个事务的所述输出,其中所述响应数据记录存储在所述区块链上记录的一个或多个事务的一个或多个输出中。

13. 根据权利要求12所述的方法,其中所述方法还包括:通过在区块链上记录新事务来更新或撤销所述响应数据记录中的过期响应数据记录,所述新事务具有指向存储所述过期响应数据记录的所述事务的所述输出的输入。

14. 根据前述任一项权利要求所述的方法,其中所述存储步骤包括:通过网络访问所述数据存储器,并通过所述网络发送数据,以在所述数据存储器中创建所述响应数据记录。

15. 根据前述任一项权利要求所述的方法,其中所述一个或多个验证方是多个验证方。

16. 根据权利要求15所述的方法, 其中所述存储仅向所述验证方中的每个验证方提供所述响应数据记录中的一个或多个响应数据记录的相应子集, 不同子集向所述验证方中的不同验证方提供记录。

17. 根据权利要求16所述的方法, 其中所述子集相互排斥。

18. 根据前述任一项权利要求所述的方法, 其中所述响应数据记录中的每个响应数据记录包括相应响应的显式值。

19. 根据权利要求18所述的方法, 其中每个响应数据记录的值仅以加密形式存储在所述数据存储器中。

20. 根据权利要求19所述的方法, 其中每个响应数据记录需要不同的相应解密密钥来解密。

21. 根据从属于权利要求16的权利要求20所述的方法, 其中, 通过将所述解密密钥中的不同解密密钥分发给不同的验证方, 向所述验证方中的不同验证方提供所述不同子集。

22. 根据权利要求1至17中任一项所述的方法, 其中, 所述响应数据记录中的每个响应数据记录仅包括相应响应的证明, 而不包括所述响应的显式值, 所述证明包括相应响应的变换, 所述变换不公开所述响应, 但使验证方能够通过由所述目标返回的相应响应的另一实例执行相同的变换并与所述证明进行比较来验证所述目标的所述身份。

23. 根据权利要求22所述的方法, 其中所述变换包括哈希或双重哈希。

24. 根据前述任一项权利要求所述的方法, 所述方法包括: 从所述验证方中的一个验证方接收请求, 所述请求包括所述质询集中的一个质询; 以及, 在响应中, 将所述质询的所述另一实例输入到所述PUF模块中, 以生成对应响应的另一实例, 并将所述响应的所述另一实例返回给所述验证方, 以使所述验证方能够基于与所述数据存储器中存储的所述响应的比较来验证所述目标的所述身份。

25. 根据权利要求1至17中任一项所述的方法, 其中所述响应数据记录中的每个响应数据记录包括从所述相应响应推导出的相应公钥-私钥对的相应公钥。

26. 根据权利要求25所述的方法, 其中所述推导包括使用所述相应响应作为生成所述相应公钥-私钥对的算法中的种子。

27. 根据权利要求25或26所述的方法, 所述方法包括: 向所述验证方中的一个验证方, 发送、或提供由所述目标使用所述公钥中的一个公钥进行签名的消息。

28. 根据前述任一项权利要求所述的方法, 其中所述PUF模块包括PUF、接口逻辑、和确定性变换函数; 其中所述接口逻辑通过以下方式引起所述响应集的生成:

- 接收所述质询集;
- 将基本质询输入到所述PUF中, 以生成对应的主要响应; 以及
- 将所述接收的质询集中的每个质询与所生成的基本响应一起输入到所述变换函数中, 以生成所述响应集中的相应响应, 所述变换函数是来自所接收的质询集的相应质询和所生成的基本响应的函数。

29. 根据前述任一项权利要求所述的方法, 其中所述响应数据记录中的每个响应数据记录与所述相应质询的指示相关联地存储在所述数据存储器中, 从而使所述验证方能够确定所述相应质询、或能够基于所述相应质询查找所述相应响应, 所述相应质询将用于在验证中对所述目标提出质询。

30. 根据权利要求29所述的方法, 其中每个相应质询的所述指示包括与所述相应响应数据相关联地存储的所述相应质询的显式值。

31. 根据权利要求29所述的方法, 其中所述质询的所述指示包括主质询, 从所述主质询可以导出所述质询集。

32. 根据前述任一项权利要求所述的方法, 其中所述目标方是多个目标方中的一个目标方, 对于所述多个目标方, 所述数据存储器记录了对所述多个目标方的对应PUF的质询所产生的响应数据集。

33. 一种计算机设备, 所述计算机设备包括:

存储器, 所述存储器包括一个或多个存储器单元; 以及

处理装置, 所述处理装置包括一个或多个处理单元, 其中所述存储器存储被设置在所述处理装置上运行的代码, 所述代码被配置为当在所述处理装置上运行时, 执行根据前述任一项权利要求所述的方法。

34. 一种计算机程序, 所述计算机程序包含在非暂时性计算机可读介质上并且被配置为当在一个或多个处理器上运行时, 执行根据权利要求1至32中任一项所述的方法。

35. 一种计算机实现的方法, 用于使一个或多个验证方能够验证目标的身份, 所述目标包括目标方或所述目标方的设备; 所述方法包括, 在建立阶段, 由可信第三方:

从所述目标方接收响应数据, 所述响应数据是由所述目标方将相应的质询输入到包括物理不可克隆函数PUF的PUF模块中以生成相应的响应而产生的;

将所述一个或多个响应中的每个响应的相应响应数据记录存储在所述可信第三方的数据存储器中, 每个响应数据记录包括所述相应响应或从中导出的数据; 以及

向所述一个或多个验证方提供所述响应数据记录, 以在后续验证阶段验证所述目标的所述身份, 其中每个响应的所述响应数据记录包括所述相应响应或从中导出的数据。

36. 一种计算机设备, 所述计算机设备包括:

存储器, 所述存储器包括一个或多个存储器单元; 以及

处理装置, 所述处理装置包括一个或多个处理单元, 其中所述存储器存储被设置在所述处理装置上运行的代码, 所述代码被配置为当在所述处理装置上运行时, 执行根据权利要求35所述的方法。

37. 一种计算机程序, 所述计算机程序包含在非暂时性计算机可读介质上并且被配置为当在一个或多个处理器上运行时, 执行根据权利要求35所述的方法。

38. 一种计算机实现的方法, 其中验证方验证目标的身份, 所述目标包括目标方或所述目标方的设备; 所述方法包括, 由所述验证方:

访问所述目标方或所述验证方的任何设备外部的数据存储器中的响应数据记录, 所述数据存储器是可信第三方的第三方计算机设备的一部分、或是公共对等发布介质, 其中所述响应数据记录包括: a) 响应值, 所述响应值是通过将质询输入到包括与目标相关联的物理不可克隆函数PUF的PUF模块中而得到的, 或b) 证明, 所述证明包括所述响应的变换;

向所述目标发送包括所述质询的请求, 并在响应中接收回所述响应的另一实例;

执行比较, 并在所述比较结果匹配的条件下验证所述目标的所述身份, 其中所述比较包括: 在a) 所存储的所述响应的值与所述另一实例匹配的情况下, 或在b) 所述证明与应用于所述另一实例的相同变换匹配的情况下。

39. 一种计算机设备,所述计算机设备包括:
存储器,所述存储器包括一个或多个存储器单元;以及
处理装置,所述处理装置包括一个或多个处理单元,其中所述存储器存储被设置在所述处理装置上运行的代码,所述代码被配置为当在所述处理装置上运行时,执行根据权利要求38所述的方法。

40. 一种计算机程序,所述计算机程序包含在非暂时性计算机可读介质上并且被配置为当在一个或多个处理器上运行时,执行根据权利要求38所述的方法。

在数据存储器中存储响应值的物理不可克隆函数

技术领域

[0001] 本公开涉及物理不可克隆函数PUF领域。

背景技术

[0002] 物理不可克隆函数(physically unclonable function, PUF)是一个专门术语,指包括确定性但不可预测的物理现象的函数。PUF有时也被称为物理随机函数。PUF接收称为“质询challenge”的输入,并根据质询和PUF采用的物理现象生成称为对应“响应”的输出。PUF有时分为强PUF和弱PUF。强PUF能够为大量不同的质询生成相应的响应,通常能够取质询的任意值。弱PUF只能为单个响应或少量响应生成响应(通常质询不能取任意值)。换句话说,强PUF具有大量的质询-响应对(具有大的质询-响应空间),而弱PUF具有单个质询-响应对或数量有限的质询-响应对(质询-响应空间小或有限)。根据一种定义,弱PUF的响应数量与质询比特数成线性关系,或者更一般地,其增长不超过其他参数的线性增长。

[0003] 强PUF的一个已知示例是光学PUF。例如,光学PUF可以包括激光、光学传感器和固体光学介质,在该介质中设置有气泡或其他此类伪影。激光以可控的角度透过光学介质,产生衍射或散射图案(这是介质中气泡或伪影的效果)。传感器被布置成感测该图案。质询在于激光的角度,而响应是基于感测的图案生成的。

[0004] 弱PUF的一个示例是SRAM PUF。在这种情况下,质询在于打开静态随机存取存储器(SRAM)。由于一个SRAM与另一个SRAM之间的制造差异很小,因此SRAM单元在通电时会恰好进入0/1状态的独特模式,从而形成单个SRAM的特征指纹。PUF被配置为在通电时将其输出作为响应。

[0005] PUF可以用作生成密钥的手段,例如,用于加密算法(例如,对文档进行签名或加密)。PUF的另一种应用是标识包含PUF的计算机设备等设备。如果之前已经确定了对给定质询的预期响应,则验证方可以稍后用该质询来质询目标设备,并检查其是否给出预期响应,从而检查目标设备是否是预期响应相关联的设备。

[0006] 由于质询响应空间有限,弱PUF的输入-输出(i/o)接口往往仅限于一方或有限数量的多方(例如,只有一个可信方或有限数量的可信方可能被实际或合法授予访问PUF的权限,或者PUF的接口可能受到密码保护等)。即,只有有关的一方或多方才能访问提交质询所需的PUF的输入以及从中接收响应的输出。另一方面,对于强PUF,强PUF的i/o接口可以被大量或无限数量的多方广泛使用,这些有关方不一定是已知或可信的有关方。原因是质询响应空间足够大,攻击者不可能列举所有质询-响应对,因此攻击者自由访问PUF的能力不应像弱PUF那样允许列举和欺骗PUF,从而危及其安全性。

[0007] 在不同的技术领域,区块链是指一种分布式数据结构,其中在分布式对等(P2P)网络(以下称为“区块链网络”)中的多个节点中的每个节点处维护区块链的副本,并且广泛公开该副本。区块链包括一系列数据区块,其中每个区块包括一个或多个事务(transaction)。除所谓的“coinbase事务”外,每个事务都指向序列中的先前事务,该序列可以跨越一个或多个区块,回到一个或多个coinbase事务。coinbase事务将在下文进

一步讨论。提交给区块链网络的事务包括在新区块中。新区块的创建过程通常称为“挖掘”，该过程涉及多个节点中的每个节点争相执行“工作量证明”，即，基于等待被包括在区块链的新区块中的一组定义的有序且核实有效的未决事务的表示解决加密难题。应当注意的是，区块链可以在一些节点处被修剪(prune)，并且区块的发布可以通过仅发布区块头来实现。

[0008] 区块链中的事务可用于以下目的中的一个或多个：传送数字资产(即，一定数量的数字通证)；对虚拟化分类账或注册表中的一组条目进行排序；接收和处理时间戳条目；和/或对索引指针按时间排序。也可利用区块链实现区块链上的层级附加功能。例如，区块链协议可允许在事务中存储附加的用户数据或数据索引。能够存储在单个事务中的最大数据容量没有预先指定的限制，因此可以并入越来越复杂的数据。例如，这可用于在区块链中存储电子文档、音频或视频数据。

[0009] 区块链网络的节点(通常称为“矿工”)执行分布式事务注册和验证过程，这将后续更详细地描述。总之，在该过程中，节点核实事务并将这些事务插入到区块模板中，这些事务尝试为该区块模板标识有效的工作量证明解。一旦找到有效的解，新区块便会被传播到网络的其它节点，从而使得每个节点能够在区块链上记录新区块。为了将事务记录在区块链中，用户(例如，区块链客户端应用程序)将该事务发送到网络中的节点中的一个节点进行传播。接收该事务的节点可以争相寻找将核实有效的事务并入新区块的工作量证明解。每个节点被配置为执行相同的节点协议，该协议将包括用于确认事务有效的一个或多个条件。无效事务将不会传播或并入到区块中。假定事务已经核实有效，从而在区块链上被接受，则该事务(包括任何用户数据)将因此在区块链网络中的每个节点上作为不可改变的公共记录进行注册和索引。

[0010] 成功解决工作量证明难题可创建最新区块的节点通常被奖励一个称为“coinbase事务”的新事务，该事务分发数字资产数额，即通证数量。无效事务的检测和拒绝是通过竞争节点的行动来执行的，这些竞争节点充当网络的代理并且通过激励报告和阻止不正当行为。信息的广泛发布使得用户可以连续地审计节点的性能。仅发布区块头使得参与者可以确保区块链具有持续完整性。

[0011] 在“基于输出的”模型(有时称为基于UTX0的模型)中，给定事务的数据结构包括一个或多个输入和一个或多个输出。任何可花费输出包括指定数字资产数额的元素，该元素可从进行中的事务序列导出。可花费输出有时称为UTX0(“未花费事务输出”)。输出还可以包括锁定脚本，该锁定脚本指定输出的未来赎回条件。锁定脚本是限定核实和传送数字通证或资产所必需的条件的谓词。事务(除coinbase事务之外)的每个输入包括指向先前事务中的此类输出的指针(即引用)，并且还可以包括解锁脚本，用于解锁指向输出的锁定脚本。因此，考虑一对事务，将其称为第一事务和第二事务(或“目标”事务)。第一事务包括指定数字资产数额的至少一个输出，并且包括定义解锁该输出的一个或多个条件的锁定脚本。第二目标事务包括至少一个输入和解锁脚本，该至少一个输入包括指向第一事务的输出的指针；该解锁脚本用于解锁第一事务的输出。

[0012] 在此类模型中，当第二目标事务被发送到区块链网络以在区块链中传播和记录时，在每个节点处应用的有效性条件之一将是解锁脚本满足在第一事务的锁定脚本中定义的一个或多个条件中的所有条件。另一条件将是第一事务的输出尚未被另一早期有效事

务赎回。根据这些条件中的任何一个条件发现目标事务无效的任何节点都不会传播该事务(作为有效事务,但可能注册无效事务),也不将该事务包括在要记录在区块链中的新区块中。

[0013] 另一种事务模型是基于账户的模型。在这种情况下,每个事务均不通过参考过去事务序列中先前事务的UTX0来定义转移的数额,而是通过参考绝对账户余额进行定义。所有账户的当前状态由节点单独存储到区块链中,并不断更新。

发明内容

[0014] 通常,目标方(待验证方,即,证明者)必须直接与验证方(质询者)建立一个或多个CR对,所述验证方必须将所述CR对存储在本地以备将来参考。另一方面,本公开提供了一种方案,其中验证方不必参与建立,也不必存储自己的CR对。

[0015] 根据本文公开的一个方面,提供了一种计算机实现的方法,用于使一个或多个验证方能够验证目标的身份,所述目标包括目标方或所述目标方的设备。所述方法包括,在建立阶段,由除所述一个或多个验证方中的任一验证方之外的建立方:将一个或多个质询的集合输入到包括物理不可克隆函数PUF的PUF模块中,以基于所述PUF生成一个或多个响应的相应的集合;以及,将响应的所述集合中的每个响应的相应响应数据记录存储在所述目标方或所述一个或多个验证方的任何设备外部的数据存储器中。所述数据存储器是第三方计算机设备的一部分、或是公共对等发布介质(例如区块链),从而向所述一个或多个验证方提供所述响应数据记录,以在后续验证阶段验证所述目标的所述身份。每个响应的所述响应数据记录可以包括相应的响应自身、或从中导出的数据。

[0016] 在实施例中,所述PUF模块在所述目标方侧实现。这是一种特别轻量级的布置,因为这使第三方不必在建立时为所述目标方持有PUF模块,尤其是当第三方是存储多个目标方的记录的服务时。在一些实施例中,可以完全取消第三方,并且可以将响应数据记录(例如,响应的证明)完全存储在区块链等公共介质上。

附图说明

[0017] 为了帮助理解本公开的实施例并示出如何实施此类实施例,现将仅通过举例的方式参考附图进行说明,其中:

[0018] 图1是一种用于实现区块链的系统的示意性框图;

[0019] 图2示意性地示出了可记录在区块链中的事务的一些示例;

[0020] 图3示意性地示出了PUF的质询和响应;

[0021] 图4是包括PUF的系统的示意性框图;

[0022] 图5A是根据本文公开的实施例的扩展PUF的示意性框图;

[0023] 图5B是处于非扩展运算模式的扩展PUF的示意性框图;

[0024] 图6是在质询-响应对的分发中涉及可信第三方或发布介质的系统的示意图;

[0025] 图7是根据本文公开的实施例的验证过程的示意性流程图;

[0026] 图8A至图8C示意性地示出了根据本文公开的实施例的从主质询生成质询集的方法;以及

[0027] 图9示意性地示出了在链上记录响应数据的方法。

具体实施方式

[0028] 通过引入物理不可克隆函数 (PUF), 可以提高用于人和机器的密钥生成系统和隐私保护身份系统等系统的鲁棒性。这些可以是相互交互或与区块链等公共系统交互的各方和/或自主机器。

[0029] 这些函数基于物理系统, 是通过假设物理设备制造过程中存在随机、不可确定且不可重复的偏差来保证的, 可以用于加强人类身份与其设备之间建立的联系, 或进一步为设备本身建立不可伪造的唯一身份。

[0030] 在文献中, PUF根据其不同的性质进行分类, 分为弱PUF和强PUF。根据下文中的一个方面, 提供了一种通用扩展PUF (ePUF) 框架, 用于描述具有这两种类型的PUF的优点的实用PUF设备; 即, ePUF可以生成在各种应用程序中使用的各种各样的质询-应答对, 同时保持实施的实用性和成本效益。

[0031] 更一般地, 本文公开了与PUF和质询-应答对管理相关的各个方面。这些不同的方面可以单独使用, 也可以任意组合使用, 例如, 包括:

[0032] I. 用于扩展PUF的质询-响应空间的扩展PUF;

[0033] II. 用于通过使用ePUF设备来建立人类身份和/或设备身份的一组隐式区块链协议;

[0034] III. 通过利用区块链来改进这些身份协议的框架;

[0035] IV. 质询-响应对的轻量级存储技术; 以及

[0036] V. ePUF设备在各种问题上的一系列新应用, 例如, 用于简化支付验证 (SPV) 流程以及设备的可验证计算的KYC的实现。

[0037] 1. 物理不可克隆函数 (PUF) ——预备知识

[0038] 术语物理不可克隆函数 (PUF) 是指一类充当通用随机函数的物理系统和设备。这些PUF具有独特的物理特性, 通常在亚微米级, 这意味着每种PUF都可以通过用物理刺激探查这些特性进行唯一识别和验证。

[0039] 在高层次上, 可以将PUF视为将质询映射到响应的函数; 其对通常称为质询-响应对 (challenge-response pair, CRP)。可以使用以下符号来描述此类映射F:

[0040] $F: C \rightarrow R \quad \forall (C, R) \in \Phi_F,$

[0041] 其中C, R分别表示质询和响应, Φ_F 是PUF可以生成的 (C, R) 形式的所有质询-响应对的集。

[0042] PUF独特的物理特性通常是硅片等物理设备制造过程中固有的随机工艺偏差的结果。通常对PUF所作的假设是:

[0043] 1. 难以通过任何形式的分析来完全确定物理系统的参数; 以及

[0044] 2. 包括用作PUF的设备的原始制造者在内的任何一方都不知道物理系统的参数。这种假设通常被称为制造者抗性。

[0045] 这些假设允许使用PUF对任意质询生成不可预测但具有确定性的响应。该质询-响应过程将PUF视为物理黑盒子, 如图3所示。

[0046] 图3示出了PUF 302的物理黑盒子模型。提交方103S将质询C作为输入提交给PUF 302, 作为响应, PUF 302生成对应的响应R。所述提交方从所述提交方的计算机设备 (未示出) 等设备提交质询, 所述设备可以与实现PUF 302本身的设备相同或不同。

[0047] 提交方103S可以是生成质询-响应(CR)对的一方,作为建立阶段(稍后讨论的示例)的一部分,以建立链接到目标方或设备的身份的预期响应集。或者,提交方103S可以是在稍后的验证阶段提交质询的验证方,以验证生成的响应是否与预期响应匹配,从而验证包括PUF 302的目标设备或拥有PUF的目标方的身份。

[0048] 在另一示例性场景中,提交方103S可以是希望使用生成的响应作为密钥或种子来生成密钥的一方,以便在区块链应用程序等加密应用程序中使用(例如,对区块链事务进行签名)。

[0049] 图4示出了包括PUF 302的接口的示例的系统。所述系统包括处理器402和PUF 302。所述接口包括接口逻辑404,该接口逻辑存储在存储器中并且被设置为在处理器402上运行。在其上存储接口逻辑404的存储器可以包括采用一种或多种存储介质(例如,磁盘或磁带等磁介质,或ROM、EPROM、EEPROM、闪存、SRAM、DRAM等电子媒介)的一个或多个存储器单元。处理器402可以包括一个或多个处理单元(例如,CPU等通用处理器,或GPU、DSP或加密处理器等专用或加速器处理器)。也不排除接口逻辑404可以部分或全部在专用硬件电路中实现,或在PGA或FPGA等可配置或可重构电路中实现。

[0050] 提交方103S使用设备(未示出)通过接口逻辑404将质询C提交给PUF 302。提交方103S使用的设备可以是计算机设备(外部计算机设备或在其上实现处理器402的同一计算机设备)等。PUF 302随后通过接口逻辑404将对应的响应R返回给提交方302的设备。在一些实施例中,稍后将更详细地讨论,接口逻辑404可以包括访问控制逻辑406,该访问控制逻辑将对PUF 302的访问仅限于某些方,例如,可以出示密码、PIN或生物特征信息等认可凭证的相关方。和/或,包括处理器402的设备的物理接口可能受到限制,例如,放置在只有授权人员才能进入的房间或综合大楼中,或保存在上锁的盒子或柜子中。然而,在替代系统中,可以将接口逻辑404提供给任何一方进行质询查询。

[0051] PUF的质询-响应过程允许通过从选定的响应中提取这些质询来生成伪随机数据值。例如,PUF可以用作密钥生成器来提取要在密码学中使用的随机可重复数据。应当注意的是,PUF 302以确定性和可重复的方式起作用,使得在多个不同的场合给出相同的质询时,PUF将产生相同的响应。

[0052] 有许多不同的物理系统可以用作PUF,并且使用这些系统的PUF有许多不同的实现方式。PUF的一个说明性示例是含有气泡的光学介质,当用激光探查时,气泡会产生响应衍射或‘散斑’图案,该图案是由(i)激光的位置和(ii)光学介质的小尺度参数确定的。

[0053] 1.1.PUF的类别

[0054] 1.1.1弱PUF:弱PUF的特点是质询-响应空间小,并且许多PUF只有单个质询,使得CRP空间的大小为 $|\Phi_F|=1$ 。通常认为弱PUF的质询-响应空间达到 $O(n)$ 量级,其中 n 是PUF中受不可控制制造偏差影响的组件的数量。

[0055] 在使用弱PUF的情况下,通常还假设对PUF的响应的访问受到限制。这是因为,由于弱PUF对其提供服务的CRP数量少,攻击者可以在合理的时间内列举所有此类对,因此可以模仿或“欺骗”PUF的行为。在讨论弱PUF的行为时,该限制有时称为受限质询-响应接口。

[0056] 这些特性使得弱PUF天生最适合在加密应用程序中用作密钥生成器,其中PUF生成的一个(或几个)CRP可以用作加密操作的密钥,例如用于加密设备上的非易失性存储器(NVM)或用作HMAC对称密钥。在此类情况下,从PUF的响应中导出的密钥必须保密,并且只有

设备的拥有者知道,以确保所执行的加密过程的安全性以及PUF本身的安全性。

[0057] 弱PUF的一个突出且广泛实现的示例是SRAM PUF,其中术语‘SRAM’是指“静态随机存取存储器”。SRAM PUF的设计利用了SRAM芯片“通电”状态的变化,由于在芯片通电时芯片中的SRAM单元处于“0”或“1”状态的变化,每个SRAM芯片都具有唯一的指纹。

[0058] 在这种情况下,认为PUF构造较弱,因为有一个固定模式来探查PUF(即,通过给SRAM芯片通电),因此只有单个CRP。在这种情况下,唯一的“质询”是为SRAM芯片供电,而响应是从其通电状态获得的唯一指纹。为了确保响应的保密性,还可以使用在使用SRAM PUF的设备上现有的存储器访问控制策略或机制,或在设备上采用的替代机制来实现访问控制。

[0059] 一些PUF实现方式的特征(例如,在SRAM PUF的情况下)是在PUF生成的响应中使用纠错,以确保相同的质询将以条件不变和时间不变的方式产生相同的响应。此类纠错技术的细节对于本领域技术人员来说是已知的。在某些情况下,纠错过程可能要求PUF设备最初‘注册’,以提供辅助数据源,该辅助数据源与稍后根据需要生成的响应相结合,以便于纠错。

[0060] 1.1.2.强PUF:与弱PUF相比,强PUF的特点是可能的质询-响应对(CRP对,或CRP)的可利用空间大。CRP的空间大意味着认为攻击者不可能在多项式时间内列举强PUF域内的所有质询-响应对。这一特性意味着强PUF通常可能具有不受保护的质询-响应接口,因为攻击者自由访问PUF的能力不会像弱PUF那样允许列举和欺骗PUF,从而危及其安全性。据说这类PUF还会产生不可预测的响应,即使是从知道 Φ_F 的大子集的攻击者的角度来看也是如此,这意味着强PUF更像具有大域的加密哈希函数。

[0061] 然而,对强PUF有一个限制,即,在面对质询C时,PUF只应给出响应R,并且在此过程中不得泄露关于PUF的内部工作或运算的任何其他信息。这种限制是为了减轻各种分析攻击,攻击者可能借此试图表征支撑PUF行为的物理系统。在文献中,这些通常被称为模型攻击。

[0062] 与弱PUF类似,一些强PUF构造可以依赖于纠错技术来确保设备生成的响应的准确性。

[0063] 强PUF现有的主要应用是使用固有的质询-响应机制促进系统身份验证和标识。这些机制依赖于涉及在双方之间直接创建作为共享秘密的CRP的协议,并且通常需要至少一方提前生成CRP表(初始设置),以用作另一方的身份验证通行证。

[0064] 强PUF实现方式的最早示例中的一个示例是光学PUF系统。在这种构造中,PUF包括光学介质,其中包含随机分布的物理缺陷,这些物理缺陷是制造偏差引起的,会散射入射光。

[0065] 这种PUF构造能够被定向光学散射介质的激光束探查到。在这种情况下,入射光束的方向和偏振形成质询,将观测到的散射图案作为PUF响应。

[0066] 然而,由于测量设备与PUF设备的其余部分分离,并且也很难与半导体组件直接集成,因此这种强PUF构造实现起来很复杂。除此之外,与装置本身相关的成本以及缺乏便携性的布置降低了其在日常应用中的实用性。

[0067] 此后,提出了一种称为仲裁器PUF(APUF)的电气集成强PUF,该PUF克服了这些问题中的一些问题。这种构造利用了信号多路复用,并且在电子组件中利用了运行时延迟。并行

提出了许多其他强PUF构造,尽管许多构造缺乏广泛使用的实用性,并且许多构造在安全性和潜在攻击向量方面存在相关弱点。例如,一种问题非常严重的潜在攻击是中间人攻击,攻击者可以通过这种攻击拦截以明文形式提交的质询,并欺骗认证计算。

[0068] 1.1.3.受控PUF:第三类PUF,称为受控PUF(CPUF),改进了现有的强PUF构造,但将其用作构建块。这些PUF采用强PUF并应用限制对PUF进行访问的附加控制逻辑,使它们区别于非受控强PUF,否则可能具有不受保护的质询-响应接口。

[0069] 如图4所示,应用于PUF的控制逻辑406(现在是较大PUF设备的一部分)可以协调对PUF 302本身的访问。这意味着控制逻辑组件406可以限制向PUF呈现哪些质询,以及控制如何向用户显示后续响应。

[0070] 在CPUF构造中,优选地,控制逻辑组件406应当嵌入强PUF组件内或被强PUF组件封装。根据CPUF的一种定义,如果PUF只能通过以不可分割的方式物理链接到PUF的算法进行访问(即,试图绕过该算法将导致PUF毁损),则认为PUF受控。这种嵌入会使控制逻辑的探查变得相当困难。

[0071] 这将在PUF组件和控制逻辑组件之间建立一种互利的关系,使得每个组件减轻对另一个组件的一种攻击。即,在PUF设备内封装控制逻辑可以保护控制逻辑不受物理攻击或侵入性攻击,因为这会不可修复地损坏PUF组件并改变其响应,而控制逻辑自然地保护PUF组件不受协议级攻击,以提取CRP或关于PUF本身底层的内部物理系统的其他信息。

[0072] CPUF的应用与强PUF大致相同,但可以通过更稳健的方式实现。具体地,使用上述协议可以很容易地实现认证计算和执行证明。

[0073] CPUF的一个早期示例扩展了强仲裁器PUF(APUF)的设计,要求控制逻辑以已经描述的方式与APUF本身交织在一起,以便控制逻辑和APUF保护彼此不受不同类型的攻击。受控APUF设计通过结合系统的瞬态响应,从集成电路(IC)的单个静态响应生成大CRP集。

[0074] 受控PUF的另一个已知示例是PUF-FSM构造。该PUF-FSM构造包括强PUF(实际上是APUF)和有限状态机(FSM),后者充当控制逻辑,限制对APUF组件本身的质询-响应接口的访问。

[0075] 1.2.讨论

[0076] 1.2.1.实用性:文献中公认,生成既实用又轻量化,同时还可与标准互补金属氧化物半导体(CMOS)组件集成的强PUF是非常具有挑战性的。相比之下,SRAM PUF等弱PUF生成成本较低,并且可以与集成电路架构轻松地结合。

[0077] 1.2.2.对PUF的攻击:已经提出并研究了许多不同的攻击,其中不同的攻击可能针对特定PUF构造或类别。下面列出了一些最为人知的攻击类型。

[0078] ●MITM攻击——这些攻击的目标是PUF不受控制的强PUF,攻击者可以拦截明文发出的质询,以模仿或欺骗PUF的响应,特别是在用于认证计算时。

[0079] ●模型攻击——已经证明这些攻击是APUF等许多强PUF构造的漏洞。

[0080] ●选择质询攻击——这些攻击也会影响强PUF,在一定程度上是转向CPUF架构的动机。

[0081] 各种PUF设计还存在其他问题,例如,在某些情况下缺乏唯一性,从而产生破坏有关PUF系统的安全性的漏洞。

[0082] 1.2.3安全模型:PUF构造的安全模型往往有一些相似之处,例如,假设产生其CRP

的随机工艺或制造偏差是制造者抗拒的,并且难以通过分析手段来表征PUF的物理系统。然而,这三个主要PUF类别的安全模型也存在一些差异。

[0083] • 弱PUF——弱PUF的安全性依赖于其CRP是保密的假设,否则可以列举并模拟设备。这意味着弱PUF可以用于为加密操作提供熵源并安全存储熵,但实际的CRP响应数据本身不会在该过程中公开。

[0084] • 强PUF——强PUF的安全性取决于其CRP空间在质询比特数上往往呈指数增长,因此在合理的时间范围内列举整个空间是不可行的。这意味着,与弱PUF的情况不同,强PUF的CRP响应可以通过设备显示。

[0085] • 受控PUF——受控PUF的安全性由控制逻辑和PUF本身共同决定,控制逻辑可防止协议级攻击,而PUF本身则可防止物理攻击。

[0086] 强PUF区别于弱PUF的两个特性如下。首先,强PUF具有大CRP集。这意味着强PUF具有较大的质询空间 Φ_F ,而弱PUF通常只有一个(或少许)质询可用。此外,相对于任何和所有已知的CRP,强PUF被认为是不可预测的。换句话说,知道任意数量的CRP在预测新质询的响应方面没有优势。

[0087] 其次,强PUF可以具有不受保护的质询-响应接口。假设给定的强PUF不需要访问控制逻辑来限制对质询-响应接口的访问。这意味着对PUF具有物理访问权限的任何一方都可以任意提出质询并获得响应,而无需透露有关PUF或其物理特性的任何其他信息。

[0088] 受控PUF具有受保护的质询-响应接口,但也具有像强PUF一样的大质询-响应空间。

[0089] 2. 扩展PUF (ePUF)

[0090] 下面公开了一种通过从基本PUF 302的给定CR对生成多个次要CR对来扩展PUF的质询-响应(CR)空间的系统和方法。这在本文中可以称为“扩展PUF”或“ePUF”。例如,该思路可以用于扩展仅具有一个或有限数量的固有CR对,而不具有典型强PUF机制的复杂性或不切实际性的弱PUF(例如,需要激光、光学介质和传感器的光学PUF)的质询-响应空间。然而,原则上,所公开的技术可以更普遍地用于扩展任何基本PUF的CR对的数量,无论是弱PUF、强PUF、受控PUF还是其他PUF;或出于混淆或重用等其他目的而变换任何PUF的CR对。

[0091] 图5A示出了根据本文公开的实施例的扩展PUF (ePUF) 500。ePUF 500包括组成性基本PUF 302,该组成性基本PUF可以是常规弱PUF。ePUF 500还包括变换函数502,例如,加密哈希函数(例如,SHA256等)等哈希函数。ePUF 500还包括接口逻辑404',该接口逻辑可以与关于图4所讨论的接口逻辑404相似,但具有附加接口功能。接口逻辑404'和变换函数502可以在嵌入式固件等软件中实现,存储在存储器中并被设置为在处理器402上运行(如图4所示,但运行接口404'和变换函数502的附加功能)。存储接口函数404'和变换逻辑504的存储器可以包括一个或多个存储器单元,该存储器单元采用一个或多个存储介质(例如,磁盘或磁带等磁介质,或ROM、EPROM、EEPROM、闪存、SRAM、DRAM、熔丝锁存器等电子媒介)。运行接口逻辑404'和变换函数502的处理器可以包括一个或多个处理单元(例如,CPU等通用处理器,或GPU、DSP或加密处理器等专用或加速器处理器)。也不排除接口逻辑404'和/或变换函数502可以部分或全部在专用硬件电路中实现,或在PGA或FPGA等可配置或可重构电路中实现。

[0092] 接口逻辑404'可操作地耦合到变换函数502,并且可选地还耦合到基本PUF 302。

基本PUF 302可操作地耦合到变换函数。接口逻辑404' 被设置为从提交方103S(图5A中未示出)的设备(例如,计算机设备)接收输入并向其提供输出,该设备可以是在其上实现ePUF 500的同一设备或外部设备。提交方103S可以是使用ePUF 500来执行建立、生成链接到身份以供将来参考的质询和预期响应集的一方;也可以是稍后使用PUF来验证生成的响应是否与先前建立的预期响应匹配的验证方(或生成要提供给验证方的响应的质询者)。在另一示例性应用中,提交方103S可以使用ePUF 500生成用作密钥或用作生成密钥的种子的响应。例如,该响应可以用作对消息进行加密或签名的加密密钥,例如,对区块链事务的一部分进行签名。

[0093] 基本PUF 302可运算,以对应于接收作为输入的“主要”质询 C_w ,生成作为输出的“主要”响应 R_w 。本文中的“主要”质询-响应(CR)对是指基本组成性PUF 302的基本或“原生”(即,固有)CR对。在一些实施例中,基本PUF 302可以像弱PUF一样能够响应于单个质询 C_w 仅生成单个基本(即,主要)响应 C_w 。

[0094] 在操作中,接口逻辑404' 从提交方103S的设备接收包括至少一个“次要secondary”质询 C_i 的质询数据(质询输入)。此外,将主要primary(基本base)质询 C_w 输入基本PUF 302,以生成主要(基本)响应 R_w 。在实施例中,提交方103S需要将基本质询 C_w 包括在输入到ePUF 500的质询数据中,然后接口逻辑404' 将其路由到基本PUF 302,以生成主要响应 R_w 。然而,在其他实施例中,不排除将主要质询 C_w 从存储器、熔丝锁存器或专用电路等内源输入到基本PUF 302。无论采用哪种方式,变换函数502被设置为接收作为输入的:a)从提交方的输入质询数据中接收的次要质询 C_i ;以及b)由基本PUF 302生成的主要响应 R_w 。变换函数502是被配置为将这些的组合确定性地映射到与输入到变换函数502的 C_i 和 R_w 的特定组合相对应的唯一的相应“次要”响应 R_i 上的函数。次要质询响应对在本文中可称为“次要”,因为它们在主要(基本)CR对之上分层,部分是基于主要响应 R_w 生成的。它们也可以称为“扩展层expanded layer”或“补充supplementary”质询和响应。

[0095] 在实施例中,变换函数502包括哈希函数,例如,SHA或DSA哈希函数等加密哈希函数。哈希函数至少有两种不同的使用方式。在第一种使用方式中,变换函数502包括原像的哈希,其中所述原像包括接收的次要质询 C_i 和生成的主要响应的组合(例如,级联),即, $R_i = H(C_i || R_w)$ 。或者更一般地,所述原像也可以包括其他元素,和/或除级联之外的另一种形式的组合。

[0096] 在第二种替代方法中,变换函数502包括原像的哈希,其中所述原像包括接收的次要质询,并且哈希函数使用生成的主要响应进行初始化。即, $R_i = H(C_i)$,其中H由 R_w 初始化。或者更一般地,H的原像也可以包括其他元素,只要该原像至少包括 C_i 。由 R_w 初始化意味着原像到由哈希函数H定义的输出的映射本身将取决于 R_w 。而在前一种情况下,由H引起的原像到输出的映射不取决于 R_w ,而是原像取决于 R_w 。即,在上一段中,原像取决于 R_w ,而在本段中,只有H取决于 R_w 。

[0097] 更一般地,对于ePUF 500要容纳的域中的每个可能的 C_i ,原则上可以使用任何函数,只要该函数确定性地并且唯一地将 C_i 和 R_w 的组合映射到 R_i 的相应值上。

[0098] 次要质询 C_i 可以取许多不同的可能值中的任意值,变换函数502将基于接收的特定次要质询 C_i 的值以及主要响应 R_w 的值将这些任意值映射到次要响应 R_i 的相应值。因此,ePUF 502能够将给定的主要(基本)CR对的CR空间扩展到多个次要CR对。在实施例中, C_i 可

以取所使用的变量支持的值范围内的任意值(例如,如果是32位整数,则可以取 2^{32} 中的任意值)。

[0099] 在一些实施例中,ePUF 500能够在另一种运算模式下运算,如图5B所示。在这种情况下,接口逻辑404'检测到输入质询数据仅包括主要质询 C_w 。作为响应,接口逻辑将接收的 C_w 值路由到基本PUF 302,并将得到的主要响应 R_w 路由回提交方103S的设备。换句话说,在该实施例中,ePUF 500也能够“传统”或“非扩展”模式下运算。

[0100] 可选地,根据应用程序,接口逻辑404'可以包括访问控制逻辑406,该访问控制逻辑将访问仅限于有限数量的可能的提交方103S,例如通过仅向能够出示其识别为映射到被授权方的凭证(例如,密码、PIN或生物特征输入)的一方授予访问权限。在这种情况下,ePUF 500可以被视为CPUF的一种形式。替代地,ePUF 500的物理接口可以受到法律或物理保护,例如通过将包括ePUF 500的设备保存在只允许有限的一组有关方进入的房间或场所,或将其保存在上锁的盒子、柜子或房间中。在这种情况下,ePUF 500可以被视为一种扩展弱PUF。

[0101] 作为对PUF接口的此类物理限制的替代或补充,还可以通过限制对主要质询的访问来限制访问。例如,目标方103T(“爱丽丝”,稍后讨论)可能是唯一知道 C_w 的一方。

[0102] 然而,作为另一替代方案,对接口逻辑404'的访问可以不受限制,例如,任何一方都可以通过互联网自由查询。在这种情况下,ePUF 500可以被视为一种通过扩展弱基本PUF机制而创建的强PUF 502。

[0103] 图5A所示的布置提供了一种新的混合类PUF设备,本文中称为扩展PUF(ePUF),其通常可以用作许多应用程序的框架,如稍后部分所给出的。

[0104] ePUF可以定义为如图5A所示的物理设备或系统,包括以下三个模块的组合:基本PUF 302,例如固有弱PUF;变换函数502,例如加密哈希函数;以及接口逻辑模块404'。如所讨论的,通过引入加密哈希函数等变换函数404',ePUF 500可以相对于常规PUF 302进行“扩展”,因为它将唯一质询空间 Φ_F 的大小从基本弱PUF 302的 $|\Phi_F| \sim 1$ 增加到由哈希函数选择而不是弱PUF的物理系统所限定的 $|\Phi_F| \gg 1$ 。

[0105] 之前已经探讨了实现将强PUF的大CRP空间与弱PUF本身的实用性相结合的系统想法。众所周知,在组合运算中使用多个基于FPGA的弱PUF来生成具有强PUF特征的系统。这里的目的是“扩展”基本弱PUF的CRP空间。然而,这种性质的现有构造在实际应用时有局限性。在上述FPGA设计的情况下,系统必须构建在FPGA上,仍然受到相对较小的CRP空间($\sim 2^{10}$)的影响。

[0106] 本文公开的ePUF设计是非常轻量级的设计,因为只需要向现有的弱PUF 302添加接口逻辑组件404'和加密哈希函数(或其他此类变换函数)502。例如,如果选择SRAM PUF作为广泛使用的弱PUF 302,则添加其余两个模块404'、502应该不会产生显著的开销,例如,作为软件(例如,固件)中的小算法或相对简单的硬件电路片段来实现。此外,将ePUF 500的可能输出的空间扩展到选定哈希或变换函数502的范围,该范围比上述范围大得多。例如,如果选择SHA-256哈希函数,则可能的输出(以及因此生成的CRP)的空间立即增加到 $2^{256}-1$,除了嵌入哈希函数模块本身外,不需要进一步扩展硬件开销。

[0107] 图5A示出了扩展PUF(ePUF) 500的示意性设计。使用加密哈希函数的实施例还意味着ePUF 500具有其CRP不可预测的特性,这对于强PUF系统也是如此。

[0108] ePUF设备的控制逻辑元件406也可以概括为这种构造。例如,与SRAM PUF类似,如

果这适用于应用程序,控制逻辑406可以简单地实现为物理安全。

[0109] 或者,控制逻辑模块406可以实现为与CPUF一起使用的软件控制模块类似的软件控制模块,其中所述控制逻辑模块实际上嵌入到PUF设备本身内,以提供前面讨论的封装的安全互利。然而,ePUF设计与CPUF设计的区别在于,对以这种方式实现的控制逻辑没有严格要求。

[0110] 不必假设对控制模块406的侵入性攻击必然会改变ePUF设计中弱PUF组件302的行为。相反,可以根据具体情况选择该元素的实现方式。

[0111] 2.1.ePUF的质询和响应

[0112] 可以按照以下方式定义与ePUF相对应的质询-响应对 $(C, R) \in \Phi_F$ 的集:

[0113] $\Phi_F = \{(C_w, R_w), (C_1, R_1), (C_2, R_2), \dots, (C_N, R_N)\},$

[0114] $F: C_i \rightarrow R_i, \quad \forall i \in (1, N)$

[0115] $F_w: C_w \rightarrow R_w$

[0116] 其中 (C_w, R_w) 是与弱PUF 302的基本质询-响应相对应的特许CRP,其中映射 F_w 由弱PUF的独特物理特性定义。对 (C_w, R_w) 在本文可以称为ePUF的基本对或主要对。相反,映射 F 由ePUF选择的加密哈希函数定义。图5A和图5B示出了从ePUF 500中提取响应,其中(图5B)质询仅为 C_w , (图5A)质询还包括 C_i 。

[0117] 在扩展PUF的一些实施例中,所有质询 $C_i, i \in \{1, 2, \dots, N\}$ 必须伴有基本质询 C_w , 并且基本响应 R_w 被纳入生成所有其他响应 R_i 的过程中,如图5A所示。

[0118] 图5A所示的使用ePUF生成通用CRP的过程旨在通过将基本秘密对应用于任何其他任意质询 C_i 来扩展该基本秘密对,从而使用基本质询-响应对 (C_w, R_w) 。用于从ePUF生成CRP的算法可以针对特定用途进行定制,前提是该算法以确定性的方式使用基本对 (C_w, R_w) 。此类算法的简单示例(表示为 `getResponse()`)可以写为如下形式。

[0119] `getResponse()`:

[0120] 输入:质询

[0121] 1.从用户/客户端获得质询。

[0122] 2.检查质询 $= C_w$?

[0123] i.如果是:

[0124] 1.使用 C_w 探查弱PUF模块,以获得 R_w ,

[0125] 2.设置响应 $\leftarrow R_w$ 。

[0126] ii.如果不是:

[0127] 1.将质询分为 C_w 和 C_i 组件,

[0128] 2.使用 C_w 探查弱PUF模块,以获得 R_w ,

[0129] 3.将 C_i 和 R_w 发送到哈希函数模块,

[0130] 4.计算 $\text{hash}(C_i, R_w, H)$,

[0131] 5.设置响应 $\leftarrow \text{hash}(C_i, R_w, H)$ 。

[0132] 3.返回响应

[0133] 输出:响应

[0134] 函数 $\text{hash}(C_i, R_w, H)$ 是通用函数,用于使用加密哈希函数 H 计算哈希摘要。函数 $\text{hash}()$ 可以通过多种方式实现,例如,在简单情况下通过简单地计算 $H(C_i || R_w)$ 来实现,或者可

以通过繁琐地计算 $H(C_i)_{R_w}$ 来实现,其中值 R_w 已用作哈希函数 H 的初始向量。无论采用哪种方式,hash()的输出都取决于 C_i 和 R_w 。

[0135] 图5A和图5B中的图示出ePUF 500可以配备接口逻辑404',可选地包括控制逻辑模块406。在实施例中,在生成响应时有两种可能的路径,其中当质询只是 C_w 时使用图5B所示的路径,而当质询是伴有 C_w 的新值 C_i 时使用图5A所示的路径。这是确定性的。

[0136] 所公开的ePUF设计可以用于提供以下优点和/或其他优点中的任何一个优点。

[0137] • 大CRP空间,由选定哈希函数的域和范围限定。

[0138] • 将控制逻辑与PUF本身分开的灵活性。

[0139] • 弱PUF的安全原语。

[0140] 这意味着用户可以像使用CPUF设备一样使用ePUF设备,但其中对PUF的受控访问包括(I)安全地存储弱PUF(C_w, R_w)的基本CRP,以及(II)将对PUF设备的物理访问仅限于预期用户。

[0141] 在该模型中,基本对(C_w, R_w)就像主密钥,从中可以导出(C_i, R_i)形式的大量其他CRP,并且可以由外方或第三方提交 C_i 。

[0142] 2.2.ePUF的应用

[0143] ePUF设备的可能应用(用例)可以大致分为至少两大类:

[0144] 1.将身份链接到活动或计算操作;以及

[0145] 2.充当加密操作的密钥生成器。

[0146] 应用(1)通常由现有的强PUF实现,而应用(2)通常由现有的弱PUF实现。ePUF构造结合了每一种的特性,这意味着ePUF同样适用于任何一种应用。在应用(1)中,一个优点是在实践中使用ePUF通常比大多数强PUF或受控PUF更容易实现此类应用。

[0147] 3.身份链接系统

[0148] 在本节中,公开了一种用于将人类身份或机器身份链接到PUF设备的通用框架。

[0149] 实施例可以使用扩展PUF(ePUF)。这里的目的是制定PUF体系结构,该体系结构提供了健壮、高度通用且灵活的身份系统,可以重新用于许多不同的用例。打算在该构造中获得以下特性:

[0150] • 与强PUF相当的大CRP空间;

[0151] • 与弱PUF相当的实用性;以及

[0152] • 比CPUF更灵活的控制逻辑。

[0153] ePUF设计可以用作在一系列身份建立协议中使用的PUF的基础模型。实施例可以允许终端用户或机器在该过程中的独立性。如果现有方案(也可以重新用于使用ePUF)依赖于可信第三方在建立期间直接访问PUF设备,则基于ePUF提出的系统可以允许PUF设备的终端用户转而建立身份并参与后续身份认证,而无需第三方在建立期间在本地或直接访问设备。

[0154] 一些实现方式可以通过引入公共区块链来提高这些身份链接协议的鲁棒性并进一步扩展这些协议。这里可以采用的两个概念是:(A)使用区块链作为防篡改CRP管理系统;以及(B)使用区块链网络作为时间戳服务,用于协调身份链接协议中使用的请求-响应消息,并提供高效的撤销系统。

[0155] 图6示出了根据本文公开的实施例的用于身份链接和验证的示例性系统。图7示出

了对应的方法。

[0156] 所述系统包括PUF模块603、目标方103T的计算机设备102T以及响应数据存储器601。PUF模块603包括如先前关于图5A和图5B所述的ePUF 500,或替代地,可以仅包括如先前关于图3和图4所述的常规PUF 302或PUF和常规接口逻辑404。响应数据存储器601可以是第三方计算机设备602的一部分,并且由可信第三方管理,或者可以是区块链等分布式对等存储介质。第三方设备602可以包括服务器设备等,该服务器设备包括位于一个或多个地理位置的一个或多个服务器单元(云存储技术本身是本领域已知的)。所述系统还可以包括验证方103V的计算机设备102V,或在一些替代情况下,验证方可以直接与PUF模块603、目标方的计算机设备102T或第三方计算机设备602交互。

[0157] 本文中对用户或一方103等(无论是验证方103V、目标方103T还是第三方)的动作的任何引用都涵盖了该方通过该方的计算机设备102行动的可能性。为了简洁起见,这不一定每次都会明确说明,但可以理解为隐含地涵盖。这涵盖了以下两种可能性:A)该动作由该方对计算机设备的手动用户输入触发或在其控制下执行,或B)该动作由代表该方的计算机设备自动执行(说一方执行动作不一定意味着该方的人类用户手动发起该动作,而是可能意味着该方的设备代表他/她执行该自主动作)。为免存疑,还应当注意的是,一方可以指单个个体、团体、个人或组织,例如公司、慈善机构、政府机构或市政或学术机构。

[0158] 目标方103T的计算机设备102T可以可操作地连接到响应数据存储器601(例如,通过连接到第三方设备602)。验证方103V的计算机设备102V可以可操作地连接到响应数据存储器601(例如,通过连接到第三方设备602)。目标方103T的计算机设备102T可以可操作地连接到验证方103V的计算机设备102V。这些连接中的任何连接都可以经由一个或多个网络(例如,因特网或移动蜂窝网络等一个或多个广域网)形成。在实施例中,这些连接中的任何连接都可以经由相应的安全信道形成,例如基于有关双方之间共享的共享秘密来建立。在本文中,当双方以任何方式(例如,通过发送质询或接收响应)进行通信时,应当理解的是,这涵盖了可以通过其相应的计算机设备(102V、102T;102T、602;或102V、602)之间的任何合适的直接连接或网络连接来进行这些通信的可能性。为了简洁起见,这不一定每次都会明确说明,但可以理解为隐含地涵盖。

[0159] 目标方103T是指其身份将基于PUF模块603进行验证的一方,或者拥有需要基于PUF模块603进行验证的设备的一方,或以其他方式对该设备负责的一方,或与该设备相关联的一方。验证方103V是要执行验证的一方。可能有多个验证方103V(每个验证方可以通过相应的计算机设备102V行事),但为了便于说明,图6中仅示出了一个验证方。PUF模块603可以由目标方103T拥有,可以并入或连接到其计算机设备103T,例如,作为外围设备或通过本地网络或组合(例如,接口逻辑404/404'可以在计算机设备103T上实现,PUF 302可以是外部外围设备)。或者,PUF模块603可以由可信第三方拥有,可以并入或连接到第三方计算机设备602,例如,作为外围设备或通过本地网络或组合(例如,接口逻辑404/404'可以在第三方设备602上实现,PUF 302可以是外部外围设备)。

[0160] 通常,目标方103T、验证方103V或第三方中的任何一方都可以担当先前关于图3、图4和图5所讨论的提交方的角色。目标方103T、验证方103V或第三方中的任何一方可以担当提交方的角色,或者可以担当建立方的角色,使用PUF模块603建立一个或多个CR对的集,并将它们链接到目标方103T的身份,以便在稍后的验证阶段使用。稍后将更详细地讨论一

些具体示例性场景。

[0161] 响应数据存储器601存储PUF模块603在建立阶段生成的响应数据。数据存储器601将该响应数据与目标的身份证据相关联地存储,所述目标可以是目标方103T或目标方103T的设备。验证方103V可以访问响应数据存储器601,并且可以在验证阶段的稍后时间使用该数据存储器来验证所述目标的所述身份。为此,验证方103V向所述目标提出质询,以生成对质询 C_i 的响应 R_i ,该质询 C_i 先前包含在建立阶段使用的质询集中。如果所述目标能够根据存储在响应数据存储器601中的内容生成预期响应,则这证明所述目标拥有或控制PUF模块603,因此可以假设所述目标是在建立阶段捕获其身份的一方。

[0162] 在替代变体中,响应数据存储器601可以存储基于在建立阶段生成的一个或多个响应(例如,使用响应作为种子)生成的一个或多个相应公钥-私钥对的一个或多个公钥。如果所述目标稍后使用私钥中的一个私钥对消息(例如,文档或区块链事务)进行签名,则验证方可以使用来自响应数据存储器601的对应公钥来验证签名。应当注意的是,在此类变体中,术语“响应数据”在更广泛的意义上用于涵盖从响应 R_i 导出的数据,而不一定是响应 R_i 的显式值或证明。

[0163] 响应数据存储器601可以是可公开访问的,或访问可以仅限于包括至少一个验证方103V的一方或多方的有限集合。该响应数据存储器可以托管在第三方系统602上或以点对点方式托管,或替代地,可以在目标方103T的计算机设备102T或验证方103V的计算机设备102V中实现。

[0164] 参考图7,所述方法包括两个阶段:建立阶段702和验证阶段704。在建立阶段,在步骤710中,充当建立方的目标方103T或第三方中的一方将一个或多个质询 C_i ($i=1\cdots n$,其中 $n\geq 1$)的集提交给PUF模块603。在使用ePUF 500的情况下,这些质询是次要质询。在目标方103T拥有PUF模块603并且正在执行建立的情况下,质询 C_i 可以由目标方103T生成或从第三方系统602或验证方103V接收。在第三方拥有PUF模块603并且正在执行建立的情况下,质询可以由第三方系统602生成或从目标方103T或验证方103V接收。无论采用哪种方式,作为响应,PUF模块603基于PUF 302/500生成响应 R_i 的对应集。在使用ePUF 500的情况下,这些响应是次要响应。因此,所述方法生成CR对 $\{C_i, R_i\}$ 的集。

[0165] 在实施例中,对PUF模块603的访问受到限制,使得只有目标方103T(以及建立方,如果是不同方)可以获得对响应 R_i 的访问权限。这可以通过访问控制逻辑404或404'来实现,该访问控制逻辑可以仅向能够出示密码、PIN、生物特征数据等认可凭证的一方授予访问权限。和/或对PUF模块603的物理接口的访问可以受到物理保护,例如通过将其保存在上锁的容器、柜子或房间中;或者可以受到法律保护,例如通过将PUF模块603存储在仅允许特定人员进入的房间或综合大楼中。作为另一替代或附加限制,在使用ePUF 501的情况下,可以限制对主要质询 C_w 的了解,使得只有目标方103T(以及在实施例中充当单独建立方的可信第三方)知道 C_w 。

[0166] 在步骤720中,所述方法包括将响应数据存储在响应数据存储器601中。在实施例中,存储的响应数据包括生成的CR对 $\{C_i, R_i\}$ 的记录。每个CR对的记录包括以指示该对的对应质询 C_i 的方式存储的相应响应 R_i 的记录。在实施例中,每个响应 R_i 的存储记录包括响应的显式值(即, R_i 的实际值),该显式值显式地向可以读取记录的验证方103V公开。该值可以明文存储,或者如果验证方具有解密该值的解密密钥,则可以对其进行加密,但尽管如此,

就本文的目的而言,存储值仍然被称为显式值,因为该值显式地向验证方103V公开。或者,响应的记录可以包括响应 R_i 的“证明”,包括 R_i 的确定性变换。一个示例是存储哈希 $H(R_i)$ 或双重哈希 $H^2(R_i)$ 的值。这使得验证方能够通过检查应用于 R'_i (例如, $H(R'_i)$ 或 $H^2(R'_i)$)的相同变换是否与证明匹配来检查响应 R'_i 的值是否与存储器中记录的值相同。这样做的好处是,没有公开响应 R_i 的实际值。因此,在存储器601是区块链等公共介质的情况下,该方法变体可能特别有用。然而,加密是另一种可能性。

[0167] 在响应数据以加密形式存储的情况下,可以单独加密每个响应数据片段(例如,每个CR对),每个响应数据片段需要不同的相应解密密钥来解密。或者,响应数据的子集或整个集(例如,给定目标方103T的所有CR对)可以一起加密,所有响应数据可以作为一组用同一密钥一起解密。

[0168] CR对等响应数据与目标的身份证据相关联地存储在响应数据存储器601中。例如,目标方103T可能需要生成标识信息的一个或多个片段(例如,护照),作为建立的一部分。与响应数据相关联地保存在响应数据存储器601中的证据可以包括与响应数据(以明文形式或验证方103V可访问的加密形式)相关联地显式存储的该信息本身的副本。或者,如果响应数据存储器601由可信第三方或验证方103V自己管理,那么仅凭响应数据与特定身份相关联地注册在响应数据存储器601中的事实就可以被认为是充分的证据(假设验证方103V信任建立方和管理响应数据存储器601的一方(例如,可信第三方)在建立时已经适当地检查了目标方的标识信息)。

[0169] 在验证阶段704,在步骤730中,验证方103V访问响应数据存储器,以确定要在验证操作中使用的响应数据。在实施例,有多个潜在的验证方103V,为每个验证方分配一个或多个CR对的不同子集。即,响应数据存储器601仅向给定验证方103V公开分配给该方的一个或多个CR对的一个或多个预期响应 R_i 。例如,该方案可以由可信第三方系统602管理。此类方案有利于保持CR对分离,使得一个验证方103V不能假装另一验证方是目标。然而,如果有权访问存储器601的所有验证方103V都是可信的,那么这就不是必需的。

[0170] 在实施例,验证方103V最初并不知道他/她将要使用的质询,并通过从数据存储器601访问该质询以及对应的响应数据(例如,响应或证明)来确定这一点。或者,验证方103V确实预先知道他/她打算使用哪个质询,并使用该质询在数据存储器601中查找映射到该质询的响应数据。

[0171] 在验证方103V(或者实际上任何一方)访问来自区块链的数据(例如,确定响应数据和/或质询)的场景中,访问区块链可以通过直接查询区块链网络的节点来执行,或通过查询缓存区块链数据或代表寻求访问区块链数据的各方协调查询的中间服务来间接执行。例如,验证者103V可以访问来自另一服务提供者的数据,该服务提供者未直接连接到区块链网络106,但可能只提供响应相关数据,可能还会提供默克尔证明。

[0172] 在步骤740中,验证方103V向拥有或控制PUF模块603的目标方103T提交质询 C_i 。该质询是与验证方103V在步骤730中从响应数据存储器601访问的记录中的一条记录相对应的质询。应当注意的是,在可信第三方在建立时拥有PUF模块603的场景中,PUF模块603可以在建立阶段702和验证阶段704之间从可信第三方以物理方式传递给目标方103T。

[0173] 响应于提交的质询 C_i ,PUF模块603生成对应的响应 R_i ,目标方103V将该响应返回给验证方。在步骤750中,验证方检查接收的响应 R_i 是否与根据在步骤730中从响应数据存

储器601访问的响应数据所预期的响应一致。

[0174] 如上所述,执行建立步骤702的一方可以是目标方103T或存储响应数据(例如,CR对)的可信第三方。在进一步的变体中,这些步骤可以由另一协调方(例如,可信的预言机(在实施例,运行包括数据存储器610的第三方计算机设备602的一方之外的另一第三方))来执行。在此类实施例中,数据存储器601可以是(不同的第三方的)第三方系统602或区块链等公共对等介质。和/或在进一步的变体中,可以在对PUF模块603执行输入的一方和接收输出的一方之间进行分离。

[0175] 同样如上所述,对于在响应数据存储器601中记录响应 R_i 的方式,至少存在两种可能性。第一种方式只是显式地存储 R_i 本身的实际值。在这种情况下,步骤750只包括将存储的值(在建立702时确定)与现在响应于所提交的质询 C_i (在验证阶段704)接收的值 R'_i (响应 R_i 的声称值)进行比较。如果匹配,则所述方法转到步骤760,在该步骤中声明目标方103T的身份得到验证。否则,所述方法转到步骤770,在该步骤中声明目标方103T的身份未得到验证。

[0176] 第二种可能性是只有 R_i 的证明(例如,哈希或双重哈希)存储在响应数据存储器601中。在这种情况下,验证方103V将用于生成证明的相同变换应用于他/她在验证阶段704从目标方103T接收的响应 R'_i 。如果这与存储的证明匹配,则所述方法转到步骤760,在该步骤中声明目标方103T的身份得到验证。否则,所述方法转到步骤770,在该步骤中声明目标方103T的身份未得到验证。

[0177] 在响应数据存储器601中,对于将对应的质询 C_i 指示为与每个记录的响应 R_i 相关联的方式,存在至少两种可能性。第一种方式只是存储每个CR对 $\{C_i, R_i\}$ 的显式值,即,存储 R_i 和 C_i 的实际值(明文或加密)。或者,根据本文公开的实施例,第二种更轻量级的方式是存储主质询 C_m ,根据预定的确定性质询推导函数 f ,可以从该主质询推导出质询 C_i 。

[0178] 如图8A所示。每个响应 R_i 与相应的索引相关联地存储。函数 f 要么存储在响应数据存储器601中,要么是验证方103V预先知道的。无论采用哪种方式,验证方103V将主质询 C_m 输入到函数 f 中,以确定与响应 R_i 中的至少一个响应 R_i 的索引 i 相对应的质询 C_i 。然后,验证方103V使用该质询 C_i 来验证目标。

[0179] 在一些此类实施例中,函数 f 也可以是标识信息806的函数,其可以是标识信息的单个片段或标识信息802的多个片段(例如,护照信息、母亲的娘家姓和指纹信息)的组合804(例如,级联)。这可以包括目标方103T的标识信息。这使得能够实现特定目标方103T特有的质询 C_i 集,由于唯一性可能很重要,因此这出于安全原因是有利的,例如,如果使用同一第三方系统602为不同的目标方生成质询集。使用目标方103T的护照信息或母亲的娘家姓等个人身份信息是很好的选择,因为这是他/她已经知道的,并且往往是保密的。

[0180] 替代地或附加地,标识信息806可以包括验证方103V的标识信息,使得 f 是特定验证方103V的身份的函数。这可以用于将一个或多个特定质询的特定子集分配给特定验证方103V,从而为不同的验证方103V提供在验证704中使用的不同质询 C_i 。

[0181] 在一些实施例中,无论主质询 C_m 是如何形成的,质询 C_i 都可以通过链式方式映射到主质询 C_m ,使得 $C_1 = f(C_m)$ 、 $C_2 = f(C_1)$ 等,如图8B所示。换句话说,通过将函数 f 应用于主质询 C_m 来确定第一质询 C_1 ,然后通过将同一函数 f 应用于第一质询来确定第二质询 C_2 ,以此类推。例如, f 可以包括哈希函数。

[0182] 在另一变体中,质询 C_i 可以通过分层的方式映射到主质询 C_m ,如图8C所示。稍后将对此进行更详细地讨论。

[0183] 链式方法更加轻量级,如果 $f()$ 不需要根密钥之外的任何数据,则链式方法也更容易从根信息中恢复。在分层推导的情况下,将添加树中的索引,这对于像这样的简单链是不需要的: C_m 、 $H(C_m)$ 、 $H(H(C_m))$ ……,例如,其中 $f()$ 只是哈希函数。

[0184] 无论 $f()$ 的形式如何,或者主质询是否包括标识信息和/或其他信息,在实施例中,第三方系统602可以在建立702期间从目标方103T接收主质询 C_m 。然后,第三方将接收的主质询存储在数据存储器601中(例如,本地或链上),以供将来在验证704中使用。或者,第三方系统602从目标方103T接收质询 C_i 的集,并通过应用函数 $f()$ 的逆函数等方式从中导出主质询 C_m 。在这些方法的变体中,第三方系统602可以从目标方103T之外的别处(例如,从预言机或协调方(未示出))接收标识信息、主质询或质询集。也可以使用此类方法的组合(例如,从目标方接收标识信息的一个片段,从别处获得标识信息的一个片段)。或在进一步的替代方案中,不涉及第三方,并且目标方103T将主质询存储在他/她自己的链上(或一些其他对等发布介质中)。

[0185] 在图7所示方法的进一步变体中,存储在响应数据存储器601中的响应数据可以不包括在建立时生成的一个或多个CR对的记录。相反,响应数据可以包括公钥-私钥对的公钥或此类公钥的集,其中一个或多个密钥对中的每个密钥对是基于来自建立阶段702的相应PUF响应 R_i 生成的。例如,响应 R_i 可以用作公钥-私钥对生成算法中的种子。在此类实施例中,除了在步骤730中,验证方访问存储的公钥中的一个公钥,以及在步骤740中,验证方103V不提交要输入到目标的PUF模块603的质询 C_i 之外,所述方法按照图7所示进行。相反,验证方103V获得由目标(据称)签名的消息(例如,文档、文件或区块链事务的一部分)。该消息可以由目标方103T发送给他/她,或者验证方103V可以从区块链或网站等发布介质自主地访问该消息。无论采用哪种方式,在步骤750中,检查包括使用从存储器601访问的公钥来验证应用于消息的签名(基于本身在本领域众所周知的已知的公钥-私钥签名验证技术)。

[0186] 下面根据本文公开的实施例更概括地描述用于ePUF或PUF的一些示例性身份建立和验证协议。考虑证明者爱丽丝Alice(目标方103T)和验证者鲍勃Bob(验证方103V)。PUF身份系统中至少有三种不同的质询类型。下面将通过示例的方式用ePUF进行描述,但更一般地,可以使用任何PUF设备(包括PUF模块603的任何设备)。

[0187] 1. 远程PUF质询——验证者通过请求爱丽丝对鲍勃提交的质询做出响应远程对证明者提出质询。该模式假设验证者知道来自证明者的PUF的预期响应,并且PUF由合法所有者拥有。

[0188] 2. 本地PUF质询——验证者通过与爱丽丝控制的PUF设备交互,在本地向证明者提出质询。该模式假设验证者知道一些关于证明者身份的信息,但对其PUF的行为一无所知。

[0189] 3. 加密质询——验证者向证明者提出质询,以满足与其身份相关的一些加密要求,例如,通过使用可证明链接到认证公钥的密钥对消息进行签名。

[0190] 在类型1和类型2的情况下,从证明者和验证者的角度来看,质询都明显依赖于PUF模块603。在这些情况下,质询以及对应的验证过程本质上与PUF设备(包括PUF模块603的设备,例如,爱丽丝的计算机设备102T)的操作相关联。在这些情况下,使用PUF设备的特性,即,其物理状态可以唯一地绑定到身份,因此PUF在使用的身份系统中起着核心作用。

[0191] 应当注意的是,术语“远程”和“本地”特指在提出质询时验证者和证明者的PUF之间的交互。这并不排除远程质询协议具有建立阶段,该建立阶段涉及提前在证明者和验证者之间进行本地交互。

[0192] 然而,在情况3中,从证明者的角度来看,质询和验证过程只需要与PUF设备相关。验证不依赖于验证者知道证明者在生成对其质询的响应时是否使用了PUF。在这种情况下,所述方法只需使用PUF的实用程序作为爱丽丝的密钥生成器,而不是将其实用程序用于将身份链接到设备本身。

[0193] 在下文中,为上述三种操作模式中的每种操作模式下的身份系统的建立和验证、可选更新以及撤销过程提供了示例性实现方式。在实施例中,通用可信第三方参与到与基于PUF的身份系统相关的过程中。这是因为此类身份系统往往需要此类第三方,以便有意义地确保身份和相关凭证的完整性和信任。在此类系统中建立和使用个人身份的情况下,有关可信第三方可以是认证中心、政府代理或诸如银行之类的金融服务提供者。

[0194] 在为机器或非人类实体建立身份的情况下,第三方可以是设备制造者、发行者、监管者或一些其他相关参与者。这种情况特别适合物联网 (IoT) 或物联网区块链 (BoT) 范例,其中将身份分配给设备网络的不同成员,这些成员可以协作执行任务或计算以实现某个目标。

[0195] 3.1. 远程PUF系统

[0196] 3.1.1. 建立:在远程PUF质询的情况下,假设向证明者提交质询C的验证者提前知道预期响应R。这意味着在这种情况下,建立过程必须在爱丽丝和另一方之间建立CRP集(即,至少一个),该CRP集可用于导出爱丽丝和另一方之间的共享秘密,该共享秘密可用于在以后验证爱丽丝的身份。

[0197] 如前所述,假设爱丽丝与配备用于建立身份的通用第三方建立该共享秘密,并且该第三方可以是稍后与爱丽丝一起参与验证过程的验证方,也可以不是。在验证方与建立身份的第三方不同的情况下,假设验证方可以从第三方处获得用于一个或多个共享秘密的相关CRP信息。

[0198] 这里的建立阶段有两个不同的选项,根据爱丽丝是否是始终可以访问PUF设备的唯一一方,或者可信第三方是否也可以仅在建立阶段访问PUF设备进行分类。

[0199] 案例1:爱丽丝具有对PUF的唯一访问权限

[0200] 1. 制造ePUF设备并将其分发给爱丽丝。

[0201] 2. 爱丽丝通过联系可信第三方申请将其身份链接到其ePUF设备。

[0202] i. 第三方为爱丽丝创建身份帐户,并要求爱丽丝提供身份证明。

[0203] ii. 爱丽丝向第三方提供相关身份证明文件或凭证。

[0204] iii. 第三方验证爱丽丝的身份。

[0205] 3. 爱丽丝和第三方为建立过程的其余部分建立安全通信信道(例如,通过标准Diffie-Hellman密钥交换):

[0206] i. 爱丽丝和第三方分别交换公钥 P_A, P_T 。

[0207] ii. 爱丽丝和第三方独立地为其余建立通信建立临时秘密,即, $S = S_A \cdot P_T = P_A \cdot S_T$ 。

[0208] iii. 爱丽丝和第三方开始通过由S保护的信道(例如,AES加密信道)

- [0209] 进行通信。
- [0210] 4. 第三方通过安全信道向爱丽丝发送质询 $C_1, C_2, \dots, C_n$ 的集。
- [0211] 5. 爱丽丝从ePUF设备获得响应 $R_1, R_2, \dots, R_n$ 。
- [0212] 6. 爱丽丝通过安全信道向第三方发送响应 $R_1, R_2, \dots, R_n$ 。
- [0213] 7. 第三方根据爱丽丝的身份帐户存储响应CRP集 $\{(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)\}$ 。
- [0214] 案例2: 第三方在建立期间访问PUF
- [0215] 1. 第三方知道基本对和哈希函数。例如, 制造ePUF设备并将其分发给可信第三方*。
- [0216] 2. 第三方从设备获得基本CRP (C_w, R_w) 。
- [0217] 3. 爱丽丝通过联系第三方申请与身份链接的ePUF设备。这可以通过未经保护的通信信道来完成。
- [0218] i. 第三方为爱丽丝创建身份帐户, 并要求爱丽丝提供身份证明。
- [0219] ii. 爱丽丝向第三方提供相关身份证明文件或凭证。
- [0220] iii. 第三方验证爱丽丝的身份, 并将ePUF设备及其基本对 (C_w, R_w) 分配给爱丽丝的帐户。共享秘密是该CRP, 或该CRP的派生物。
- [0221] 4. 第三方将ePUF设备发送给爱丽丝。
- [0222] (*设备可以先分发给爱丽丝, 然后由爱丽丝发送。然而, 在大多数情况下, 将设备直接分发给第三方更有意义。例如, 如果设备是智能借记卡, 则可以根据PUF建立将卡从制造者发送到发卡行, 然后从发卡行发送到客户爱丽丝。)
- [0223] 建立协议在爱丽丝和可信第三方之间建立一个或多个共享秘密, 以便在稍后的验证过程中用于验证爱丽丝的身份(或包含PUF的设备)。这两个案例的相似之处在于, 两个案例都优选地涉及爱丽丝和可信第三方之间的安全通信。
- [0224] 然而, 这两个案例之间的区别在于, 案例1通过建立安全通信信道来实现安全通信, 而案例2通过物理安全来实现安全通信。
- [0225] 分别在案例1和案例2中的两个协议之间要注意的另一个区别是, 在案例2中, 可信第三方可以在没有PUF的情况下导出与爱丽丝一样多的CRP, 而在案例1中, 该方必须存储固定数量的对。
- [0226] 这是案例2优于用于为用户建立PUF设备的现有协议的优点, 因为在案例2中, 允许可信第三方远程生成任意数量的CRP, 而在现有协议中, 可信第三方可能需要与终端用户或设备制造者合作才能做到这一点。在案例1中, 如果增加爱丽丝通过安全信道向鲍勃发送基本对 (C_w, R_w) 的步骤(相信第三方不会以恶意方式使用基本对), 则可以实现相同的技术优势。
- [0227] 应当注意的是, 在建立阶段使用安全通信允许未来的通信(例如, 验证过程)通过未经保护的信道进行传输。这样做的好处是, 允许在较少的技术限制下进行验证, 例如, 验证时双方都需要在线, 并且在这种一次性建立过程中仅需要额外的安全通信开销。
- [0228] 3.1.2. 验证: 在远程PUF验证模式下, 需要记住的是, 在建立阶段有两个不同的案例, 这反映在略有不同的远程验证协议中, 如下所述。
- [0229] 案例1: 爱丽丝具有对PUF的唯一访问权限

[0230] 1. 鲍勃从爱丽丝和第三方在建立期间建立的集 $\{(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)\}$ 中获得未使用的CRP, 例如 (C_1, R_1) 。

[0231] i. 如果鲍勃也是可信第三方, 他只需从该集中检索一个元素。

[0232] ii. 如果鲍勃不是可信第三方, 他通过为爱丽丝请求未使用的CRP来与第三方通信。

[0233] 2. 鲍勃向爱丽丝发送质询 C_1 。

[0234] 3. 爱丽丝从她的ePUF设备获得候选响应 R'_1 , 并将其发送给鲍勃。

[0235] 4. 鲍勃验证是否 $R'_1 = R_1$:

[0236] i. 如果是, 则验证通过。

[0237] ii. 如果否, 则验证失败。

[0238] 5. 可信第三方随后删除对 (C_1, R_1) , 留下剩余质询-响应对 $\{(C_2, R_2), (C_3, R_3), \dots, (C_n, R_n)\}$ 的集。

[0239] 应当注意的是, 在步骤1.iii.中, CRP的一次性使用性质确保了任意鲍勃不可能使用特定CRP来‘模仿’爱丽丝, 因为可信第三方只需监控每种给定情况下每个对的使用情况, 并且应该在每次身份验证尝试中使用新的CRP。

[0240] 案例2: 第三方在建立期间访问PUF

[0241] 1. 鲍勃为验证生成新的质询 C 。这可以随机进行, 也可以根据一些其他数据(例如, 已知的KYC数据、生物特征、图像)确定性地确定。

[0242] 2. 鲍勃向爱丽丝发送质询 C 。

[0243] 3. 爱丽丝从她的ePUF设备获得候选响应 R' , 并将其发送给鲍勃。

[0244] 4. 鲍勃获得预期响应 R 。

[0245] i. 如果鲍勃是可信第三方, 他能够通过计算 $R = \text{hash}(C, R_w, H)$ 直接计算响应*。

[0246] ii. 如果鲍勃不是可信第三方, 他会向第三方发送 C 并请求响应 R 。

[0247] 5. 鲍勃验证是否 $R' = R$:

[0248] i. 如果是, 则验证通过。

[0249] ii. 如果否, 则验证失败。

[0250] (*这是因为第三方在建立协议(案例2)期间获得了基本对 (C_w, R_w) , 这意味着他们知道 R_w 。还假设哈希函数 H 至少为第三方(如果不是所有人)所知, 即, 该哈希函数是SHA-256等公用标准。)

[0251] 3.1.3. 更新: 考虑到爱丽丝和第三方在验证中的一次性使用性质(以及登录等其他有用协议), 可能还需要为爱丽丝和第三方指定建立新的CRP的流程。

[0252] 案例1: 爱丽丝具有对PUF的唯一访问权限。在该案例中, 就像在建立中一样, 建立另一个安全信道来在爱丽丝和第三方之间传输质询和响应。假设爱丽丝至少具有一个 (C_i, R_i) 形式的剩余CRP来建立 $S = H(R_i)$ 形式或类似形式的共享秘密, 或者可以通过DH密钥交换访问先前的共享秘密 $S = S_A \cdot P_T = P_A \cdot S_T$ 。

[0253] 1. 爱丽丝和第三方使用共享秘密 S 建立安全通信信道。这可以通过许多方式导出, 协议对此是不可知的。

[0254] 2. 第三方通过安全信道向爱丽丝发送质询 $C_1, C_2, \dots, C_n$ 的集。

[0255] 3. 爱丽丝从ePUF设备获得响应 $R_1, R_2, \dots, R_n$ 。

[0256] 4. 爱丽丝通过安全信道向第三方发送响应 $R_1, R_2, \dots, R_n$ 。

[0257] 5. 第三方根据爱丽丝的身份帐户存储响应CRP集 $\{(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)\}$ 。

[0258] 应当注意的是,步骤2至步骤5至少与建立步骤4至步骤7相同。

[0259] 另请参见之前关于爱丽丝通过信道告知第三方 (C_w, R_w) 的评论。

[0260] 案例2:第三方在建立期间访问PUF。在该案例中,第三方可以间接生成任意数量的CRP,因为他们知道基本对 (C_w, R_w) 和哈希函数 $H()$ 。这意味着在该案例中不需要交互式更新。

[0261] 3.1.4.撤销:身份系统的另一部分可能是撤销特定的ePUF设备,使其不再用于身份目的。撤销过程很简单,可以通过以下两种方式执行:(i)由独立于用户爱丽丝的第三方执行撤销,或(ii)作为传达的撤销请求由爱丽丝执行的撤销。

[0262] 第一种情况不需要任何涉及ePUF或其他方面的技术手段。第二种情况不需要特定于ePUF的协议或解决方案,因为在第一种情况下需要撤销的一个很好的示例是爱丽丝是否丢失了包含ePUF的物理设备,或者是否以某种方式受到损害。

[0263] 然而,如果希望在撤销过程中可选地利用ePUF,其中爱丽丝仍然具有对设备的物理控制权,则可以规定使用爱丽丝和第三方建立(或导出其共享秘密)的CRP中的一个CRP对爱丽丝的请求进行身份验证,例如,通过HMAC或在每种情况下使用CRP响应或秘密作为密钥的加密消息进行身份验证。然而,由于上述原因,这无论如何都不被视为系统的严格要求。

[0264] 3.2.本地PUF系统

[0265] 3.2.1.建立:可用于本地PUF的建立与用于远程PUF的建立完全相同,但本地和远程案例之间的区别在于如何执行下面的验证步骤。

[0266] 3.2.2.验证:在该场景中,验证在本地执行。这意味着验证过程要求证明者(爱丽丝)和验证者(鲍勃)在同一物理位置。

[0267] 例如,该场景可能与法院诉讼程序(针对人类身份)有关,其中法律要求爱丽丝使用其ePUF设备在本地配合调查,或者要执行物联网系统的分析(针对设备身份),其中系统管理员可能希望在本地显式地检查特定设备的响应。也可能与支付场景相关。

[0268] 该过程适用的其他场景可能包括撞车后的车辆诊断,当局希望准确地确定是哪个数字组件发出了指令。在这种情况下,输入 C 可能是一些环境或动力学条件,而响应 R 将是设备给出的指令的一部分。

[0269] 下面概述的本地PUF验证协议与先前的远程PUF验证协议之间的区别在于,该本地协议没有假设验证者提前知道ePUF的响应。换句话说,在本地验证过程中生成的响应不能提前提供给验证者。

[0270] 然而,在该场景中,验证过程中使用的质询很可能在某种程度上是有意义的。例如,考虑一台机器,其身份可以被视为其嵌入式ePUF组件的基本对 (C_w, R_w) 。可以执行验证过程来验证之前从给定输入 C 产生输出 R 的正是该特定设备。

[0271] 1. 鲍勃基于有关CRP (C, R) 获得要提交给ePUF设备的相关质询 C 。

[0272] 2. 鲍勃可以访问ePUF设备。

[0273] 3. 鲍勃使用ePUF设备生成候选响应 $R' = \text{hash}(C, R_w, H)$ 。

[0274] 4. 鲍勃验证是否 $R' = R$:

[0275] i. 如果是, 则验证通过。

[0276] ii. 如果否, 则验证失败。

[0277] 在这些场景中, 鲍勃事先不知道候选响应 R' , 而是验证他现在从PUF设备接收的响应是否与先前生成的响应匹配。例如, 这可以用于验证(例如, 在法庭上)生成响应的胜诉人(爱丽丝)或设备是否与现在在场(例如, 在法庭上)的人或设备相同。例如, 在数字组件的示例中, 这将被配置为在基于一些输入质询 C 生成 R 时发出指令。例如, 如果设备是自动驾驶汽车, 组件接收源自或包含“前方车辆太近”数据的质询, 则会生成响应 R , R 会触发组件发出刹车指令。因此, 在回顾性诊断验证中, 验证者认为汽车减速了, 并希望验证情况实际上是“前方车辆太近”而触发了该响应。

[0278] 3.2.3. 更新: 生成更新的CRP的流程可以遵循为远程案例提出的相同逻辑, 因为该场景中的主要差异仅适用于验证。

[0279] 3.2.4 撤销: 描述的用于远程撤销的相同技术在这里也适用。

[0280] 3.3. 加密PUF系统

[0281] 3.3.1 建立: 在这种情况下, 爱丽丝使用标准加密方法与第三方建立身份, 但在该过程中使用ePUF设备。

[0282] 在该场景中, 第三方可以可选地知道在该过程中使用了ePUF。类似地, 对于以这种方式建立的身份, 身份验证者可能知道身份验证过程涉及到ePUF设备, 也可能不知道。简而言之, 以下协议仅规定设备的所有者爱丽丝知道身份系统中涉及ePUF设备。

[0283] 1. 制造ePUF设备并将其分发给爱丽丝。

[0284] 2. 爱丽丝通过联系可信第三方申请建立加密身份。

[0285] i. 第三方为爱丽丝创建身份帐户, 并要求爱丽丝提供身份证明。

[0286] ii. 爱丽丝向第三方提供相关身份证明文件或凭证。

[0287] iii. 第三方验证爱丽丝的身份。

[0288] 3. 爱丽丝选择一种加密方法来建立与其身份的加密链接, 例如, 使用其CRP建立认证的非对称密钥对。

[0289] i. 第三方从爱丽丝获得公钥 P_A , 其中 $P_A = s_A \cdot G$ 是EC密钥对。

[0290] ii. 第三方请求爱丽丝使用私钥 s_A 对消息 m 进行签名(例如, 通过ECDSA)。

[0291] iii. 爱丽丝生成ECDSA签名 $\text{Sig}(P_A, m)$ 并发送给第三方。

[0292] iv. 第三方验证签名。

[0293] 4. 如果签名有效, 第三方将根据爱丽丝的身份对密钥 P_A 进行认证。

[0294] 步骤3涉及使用用户选择的加密方案, 但假设该过程中涉及的相关密钥将是只有爱丽丝知道的CRP响应的派生物。在上面选择的示例中, 这意味着私钥 s_A 将从特定的ePUF响应 R 中导出, 例如 $s_A = H(R)$ 。

[0295] 3.3.2 验证: 在加密情况下, 使用在前面详细介绍的加密建立阶段建立的加密信息来执行身份验证。在这种情况下, 以在建立期间根据爱丽丝的身份建立认证的EC非对称密钥对为例, 现在使用该密钥进行验证。

[0296] 然而, 下面的协议可以简单地适用于任何其他加密方案, 只要在适当的情况下用现有的建立和验证协议来代替这些方案。这里的区别是使用ePUF设备作为建立和验证过程的安全密钥生成器, 这降低了对持有者爱丽丝的恶意损害的风险。

- [0297] 1. 鲍勃获得身份链接信息 P_A , 例如, 认证密钥。
- [0298] i. 如果鲍勃是可信第三方, 他只需从爱丽丝的帐户中检索 P_A 。
- [0299] ii. 如果鲍勃不是可信第三方, 他与第三方通信并为爱丽丝请求认证公钥。
- [0300] 2. 鲍勃选择消息 m 让爱丽丝签名, 并发送给爱丽丝。
- [0301] 3. 爱丽丝对消息 m 生成签名。
- [0302] i. 如果爱丽丝希望使用其认证密钥进行签名, 她将生成签名 $\text{Sig}(P_A, m)$ 。
- [0303] ii. 如果爱丽丝希望使用一次性派生密钥进行签名, 她将生成签名 $\text{Sig}(P_a, m)$, 其中 $P_a = P_A + H(d) \cdot G$ 和 d 是一些一次性数据*。
- [0304] 4. 爱丽丝将签名发送给鲍勃。此时, 如果鲍勃还不知道数据 d , 爱丽丝也可以发送数据 d 。
- [0305] 5. 鲍勃使用 P_A (以及 d , 如果适用) 根据公钥对签名进行验证。
- [0306] i. 如果签名验证通过, 则身份验证通过。
- [0307] ii. 如果签名验证失败, 则身份验证失败。
- [0308] (*该数据可能与验证相关, 例如发票信息或生物特征模糊匹配数据。数据 d 可以由鲍勃或爱丽丝选择。或者, d 可以是爱丽丝和鲍勃已知的共享秘密, 例如, 通过Diffie-Hellman密钥交换和/或HMAC导出的共享秘密。)
- [0309] 如果身份是使用EC或PGP密钥等类似的加密原语建立的, 则如上一节所述, 上述加密验证过程也可以应用于独立建立的身份。
- [0310] 3.3.3. 更新: 这里更新爱丽丝身份的过程并不依赖于在密钥生成中使用ePUF设备, 因此这里没有必要规定任何特定的方法。相反, 可以使用更新 P_A 等认证密钥的标准方法。
- [0311] 可以简单地假设ePUF将参与为任何必需的签名或一个或多个现有进程所需的其他加密过程生成密钥。
- [0312] 3.3.4. 撤销: 类似地, 这里没有必要规定特定的撤销协议, 而是遵循标准机制。可以再次假设ePUF将作为相关加密操作的密钥生成器参与后台工作。
- [0313] 3.4. 独立PUF机制
- [0314] 3.4.1 建立: 在使用ePUF设备建立身份的独立情况下, 考虑实体希望独立于任何第三方建立人类身份或者在封闭系统内建立设备身份的场景。参与该过程的唯一一方是爱丽丝, 她是ePUF设备的“所有者”, 也是稍后验证过程中的最终证明者。
- [0315] 案例1: 爱丽丝建立人类身份
- [0316] 1. 爱丽丝获得ePUF设备。
- [0317] 2. 爱丽丝通过质询 C 探查ePUF。
- [0318] 3. 爱丽丝从ePUF获得响应 R 。
- [0319] 4. 爱丽丝使用对 (C, R) 为自己建立身份:
- [0320] i. 爱丽丝可以通过加密设置来建立未经认证的身份密钥 P_A 。
- [0321] ii. 爱丽丝根据其身份发布其身份密钥。
- [0322] 5. 爱丽丝可能希望发布对其CRP的证明, 例如响应的双重哈希 $H^2(R)$ 。
- [0323] 爱丽丝为自己建立“自主”身份的案例在一定程度上有助于为只有她能控制的设备提供唯一且可复制的设备标识符。然而, 此类身份系统中缺少可信第三方, 这意味着验证

者稍后必须信任证明者的身份和证明者的设备之间的链接。这在现实世界中的应用可能非常有限。

[0324] 案例2:爱丽丝为设备建立了身份

[0325] 1. 爱丽丝获得ePUF设备。

[0326] 2. 爱丽丝通过质询C探查ePUF。

[0327] 3. 爱丽丝从ePUF获得响应R。

[0328] 4. 爱丽丝使用对(C,R)在其系统中为设备建立身份:

[0329] i. 爱丽丝将对(C,R)映射到其设备。

[0330] ii. 爱丽丝保存了数据库,其中包含其所有设备和CRP映射。

[0331] 5. 爱丽丝可能希望发布对其CRP的证明,例如响应的双重哈希 $H^2(R)$ 。

[0332] 在上述为设备创建“自主”身份的案例中,可以看出,该设计在封闭系统中可能非常有用,其中管理员只需查找该系统中的不同设备。这也可能有助于以后向其他人证明。然而,在建立期间缺少可信第三方仍然限制了证明者说服外部验证者设备没有更改,具体取决于场景。

[0333] 应当注意的是,案例1和案例2可以被视为相同的过程,但具有不同的预期目的。因此,案例1和案例2可以一起作为一种用于为人类或机器生成“自主”身份的方法,其中在后一个案例中,系统管理员(例如,IoT系统中的爱丽丝)本身是可信实体。在这两个案例中,爱丽丝都是可信实体。

[0334] 3.4.2验证:这种情况下的验证过程很简单,只需使用给定质询探查ePUF设备并检查其响应即可。可能需要在此基础上为外部各方构建更复杂的证明或证据,以向他们证明身份。

[0335] 3.4.3更新:这种情况下的更新过程只是重复建立过程,其中管理员(在这种情况下是爱丽丝)列举额外的CRP以供将来使用。

[0336] 3.4.4.撤销:在该场景中,唯一一种身份撤销是管理员(爱丽丝)希望独立撤销身份的情况,因为该过程中没有第三方参与。这意味着撤销可能就像爱丽丝停止使用ePUF设备并清除其CRP数据库一样简单。

[0337] 在后面一节中,将公开通过区块链证明和证据使这种自主撤销更加稳健的方法,以便可以在以后说服外部方。

[0338] 3.5.基于身份的CRP管理

[0339] 在上述情况下,尤其是基于远程PUF的身份系统,在建立和验证协议中用于验证身份的CRP的一次性使用性质对有关各方提出了CRP管理质询。

[0340] 例如,在可信第三方在建立期间不访问PUF设备的情况下,可能需要列举由第三方存储以供将来验证的许多CRP $\{(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)\}$ 。此外,由于ePUF本身充当质询到响应的确定性伪随机映射,因此响应看起来互不相关。因此,如果可信第三方必须为大量用户提供服务,那么为其用户或客户端制表和存储CRP集的负担将很快带来扩展问题。

[0341] 图8A示出了根据本文公开的实施例的从标识数据中确定性地导出质询。

[0342] 根据此类实施例,为了解决可信第三方的负担问题,CRP管理主要在生成质询 $C_1, C_2, \dots, C_n$ 时进行处理。这里的思路是,质询应该从单个主质询或从中导出主质询的主数据中确定性地(也可能是分层地)导出。这一概念类似于使用分层确定性(HD)钱包来管理一

次性比特币密钥,因为其旨在允许可信第三方(或另一相关方)仅使用主数据来恢复所有相关质询,在比特币场景中,主数据称为“钱包种子”。

[0343] 在一些此类实施例中,爱丽丝(目标方103T)的标识数据806作为用于生成大量质询的主数据,以确定在诸如前几节中提出的那些身份系统中使用哪些CRP。标识数据本身可以包括不同数据元素802的组合804,但在组合时,这些数据元素优选地具有以下特性:

[0344] • 唯一性——标识数据对其所属实体是唯一的;以及

[0345] • 保密性——标识数据仅为其所属实体(或所有者)所知。

[0346] 标识数据组成的简单示例可以包括护照号、国民保险号、姓名、出生日期或安全问题的答案(例如,母亲的娘家姓),或者设备标识情况下的序列号和制造信息。然而,认为也可以使用通过更先进的技术手段获得的数据,例如指纹或面部识别数据,这些数据可以使用模糊技术进行提取,以保持唯一性。

[0347] 在实施例中,用作主输入的‘标识数据’(从中导出质询集)可以包括多个上述数据。其中一个原因是确保信息对尽可能多的可信第三方保密,因为前几节中的一些协议依赖于与第三方和/或外部验证方共享质询。未经证明方爱丽丝同意,任何第三方都难以完全复制包括多个组成部分的标识数据。

[0348] 使用标识数据确定性地生成CRP的机制如图8A所示。标识数据的组成部分首先由过程‘A’(804)进行组合,该过程可以是级联、逐位运算(例如,XOR)或任何其他相关的组合运算,应当注意的是,该运算可以通过将原始数据转换为模糊形式来寻求保护隐私。

[0349] 然后,通过哈希函数或类似的过程将标识数据转换为主质询 C_m 。最后,使用主质询通过推导函数 $f()$ 确定性地推导出一系列一次性质询 $C_1, C_2, \dots, C_n$ 。在实施例中,如图8B所示,推导函数 $f()$ 可以包括哈希函数和随机数的注入,使得每个连续的质询生成成为 $C_i = \text{SHA256}(C_{i-1}, i)$,其中 i 用作所述随机数。

[0350] 过程A、从标识数据生成质询 C_m 以及推导函数 $f()$ 都可以根据特定实现方式的需要进行配置。

[0351] 图8C示出了另一个特定示例,即质询的分层和确定性推导(响应未示出)。如图8B所示,可能希望以分层方式从主质询 C_m 中导出一次性质询 C_i 。在这种情况下,CRP管理得到进一步改善,因为特定质询的生成不需要像之前的情况中那样依赖于之前的所有质询。

[0352] 使用基于身份数据的确定性质询推导减少了身份协议中证明者爱丽丝和可信第三方的存储开销。任何一方都可以只存储标识数据(或其子集),并在需要时重新计算必要的质询。

[0353] 此外,爱丽丝还可以根据需要进行选择保留或与每个身份识别服务共享尽可能多的信息,从而调整自己的隐私,但代价是她自己可以存储更多数据。

[0354] 4. 示例性区块链系统

[0355] 下面描述了可以在本公开的某些实施例中使用的示例性区块链系统。应当注意的是,“爱丽丝”和“鲍勃”只是双方的任意名称,爱丽丝和鲍勃在本节中的角色不一定与上一节或下面几节中的角色相同。

[0356] 在一些实施例中,例如,如在上一节中所讨论的,基于PUF的输出的响应数据可以存储在链上。存储在链上的响应数据可以采取实际响应本身的形式,或采取哈希或双重哈希(所谓的证明或哈希提交)等其变换的形式,或采取从PUF响应导出的公钥-私钥对的公钥

的形式。无论链上响应数据采取何种形式,都可以使另一验证方能够检查作为身份证据的目标响应或签名是否如预期的那样。在其他实施例中,区块链可以用作管理质询-响应对的手段,例如更新或撤销质询-响应对。

[0357] 下面描述了可以用于实现此类特征的区块链系统的示例。

[0358] 4.1. 示例性系统概述

[0359] 图1示出了一种用于实现区块链150的示例性系统100。系统100可以包括分组交换网络101,通常是诸如互联网的广域网。分组交换网络101包括多个区块链节点104,该多个区块链节点可以被设置成在分组交换网络101内形成对等(P2P)网络106。虽然未示出,但是区块链节点104可以被设置为近完全图。因此,每个区块链节点104高度连接到其它区块链节点104。

[0360] 每个区块链节点104包括对等体的计算机设备,不同的节点104属于不同的对等体。每个区块链节点104包括处理装置,该处理装置包括一个或更多个处理器,例如一个或更多个中央处理单元(CPU)、加速器处理器、专用处理器和/或现场可编程门阵列(FPGA),以及其它设备,例如专用集成电路(ASIC)。每个节点还包括存储器,即采用非暂时性计算机可读介质形式的计算机可读存储器。存储器可包括一个或更多个存储器单元,其采用一个或更多个存储器介质,例如诸如硬盘等磁介质、诸如固态硬盘(SSD)、闪存或电可擦可编程只读存储器(EEPROM)等电子媒介和/或诸如光盘驱动器等光学介质。

[0361] 区块链150包括一系列数据区块151,其中在分布式或区块链网络106中的多个区块链节点104中的每个节点处维护区块链150的相应副本。如上所述,维护区块链150的副本不一定意味着完全存储区块链150。相反,只要每个区块链节点150存储每个区块151的区块头(下面讨论),区块链150就可以进行数据修剪。区块链中的每个区块151均包括一个或更多个事务152,其中该上下文中的事务是指一种数据结构。数据结构的性质将取决于用作事务模型或计划的一部分的事务协议类型。给定的区块链全程使用一个特定的事务协议。在一种常见的事务协议中,每个事务152的数据结构至少包括一个输入和至少一个输出。每个输出指定将数字资产的数量表示为财产的数额,其一个示例是输出被密码锁定到的用户103(需要该用户的签名或其它解进行解锁,从而进行赎回或花费)。每个输入指向先前事务152的输出,从而链接这些事务。

[0362] 每个区块151还包括区块指针155,其指向区块链中先前创建的区块151,以定义区块151的顺序。每个事务152(除coinbase事务之外)包括指向上一事务的指针,以定义事务序列的顺序(注:事务152的序列可进行分支)。区块151的区块链一直追溯到创始区块(Gb)153,该创始区块是区块链中的第一区块。区块链150中早期的一个或更多个原始事务152指向创始区块153,而非先前事务。

[0363] 每个区块链节点104被配置为将事务152转发到其它区块链节点104,从而使得事务152在整个网络106中传播。每个区块链节点104被配置为创建区块151,并将相同区块链150的相应副本存储在其相应的存储器中。每个区块链节点104还维护等待并入到区块151中的事务152的有序集(或“池”)154。有序池154通常称为“内存池”。在本文中,该术语并不意在限制于任何特定的区块链、协议或模型。该术语是指节点104已接受为有效的有序事务集,并且对于该有序事务集,强制节点104不接受试图花费相同输出的任何其它事务。

[0364] 在给定的当前事务152_j中,输入(或每个输入)包括指针,该指针引用事务序列中

先前事务152i的输出,指定该输出将在当前事务152j中被赎回或“花费”。通常,先前事务可以是有序集154或任何区块151中的任何事务。尽管为了确保当前事务有效,将需要存在先前事务152i并核实其有效,但是在创建当前事务152j甚至向网络106发送当前事务152j时,不必存在先前事务152i。因此,在本文中,“先前”是指由指针链接的逻辑序列中的前任,而不一定是时间序列中的创建时间或发送时间,因此,不一定排除无序创建或发送事务152i、152j的情况(参见下面关于孤立事务的讨论)。先前事务152i同样可以称为先行事务或前任事务。

[0365] 当前事务152j的输入还包括输入授权,例如先前事务152i的输出被锁定到的用户103a的签名。反过来,当前事务152j的输出可以加密锁定到新用户或实体103b。因此,当前事务152j可将先前事务152i的输入中定义的数额转移到当前事务152j的输出中定义的新用户或实体103b。在某些情况下,事务152j可具有多个输出,以在多个用户或实体间分割输入数额(其中一个可以是原始用户或实体103a,以便进行变更)。在某些情况下,事务还可以具有多个输入,将一个或更多个先前事务的多个输出中的数额汇总在一起,并重新分配到当前事务的一个或更多个输出。

[0366] 根据基于输出的事务协议,例如比特币,当诸如个体用户或组织这类的一方103希望颁布新的事务152j时(由该方采用的自动程序或人为地),该颁布方将该新事务从其计算机终端102发送到接收者。颁布方或接收者将最终向网络106的一个或更多个区块链节点104(现在通常是服务器或数据中心,但原则上也可以是其它用户终端)发送该事务。另外还不排除颁布新事务152j的一方103可以将事务直接发送到一个或更多个区块链节点104,并且在一些示例中,可以不将事务发送到接收者。接收事务的区块链节点104根据在每个区块链节点104处应用的区块链节点协议来检查事务是否有效。区块链节点协议通常要求区块链节点104检查新事务152j中的加密签名是否与预期签名相匹配,这取决于事务152j的有序序列中的上一事务152i。在这种基于输出的事务协议中,这可以包括检查新事务152j的输入中包括的一方103的密码签名或其它授权是否与新事务分配的先前事务152i的输出中定义的条件匹配,其中该条件通常包括至少检查新事务152j的输入中的密码签名或其它授权是否解锁新事务的输入所链接到的上一事务152i的输出。条件可以至少部分地由包括在先前事务152i的输出中的脚本来定义。或者,这可仅由区块链节点协议单独确定,或可通过其组合确定。无论采用哪种方式,如果新事务152j有效,区块链节点104会将其转发到区块链网络106中的一个或更多个其它区块链节点104。这些其它区块链节点104根据相同的区块链节点协议应用相同的测试,并因此将新事务152j转发到一个或更多个其它节点104等等。通过这种方式,新事务在区块链节点104的整个网络中进行传播。

[0367] 在基于输出的模型中,给定输出(例如,UTXO)是否分配(例如,花费)的定义是,根据区块链节点协议,其是否通过另一个随后事务152j的输入有效赎回。事务有效的另一个条件是其试图赎回的先前事务152i的输出尚未被另一个事务赎回。同样,如果无效,则事务152j将不会在区块链150中传播(除非被标记为无效并且被传播用于提醒)或记录。这可防止重复花费,即事务处理者对同一个事务的输出分配超过一次。另一方面,基于账户的模型通过保持账户余额防止重复花费。因为同样存在定义的事务顺序,账户余额在任何时候均具有单一定义的状态。

[0368] 除了核实事务有效之外,区块链节点104还争相成为在通常称为挖矿的过程中创

建事务区块的第一个节点,而该过程由“工作量证明”支持。在区块链节点104处,新事务被添加到尚未出现在记录在区块链150上的区块151中的有效事务的有序池154。然后,区块链节点争相通过尝试解决加密难题以组装有序事务集154中事务152的新有效事务区块151。通常情况下,这包括搜索“随机数”值,从而当随机数与未决事务有序池154的表示并置且进行哈希处理时,哈希值的输出满足预定条件。例如,预定条件可以是哈希值的输出具有某个预定义的前导零数。注意,这仅仅是一种特定类型的工作量证明难题,并且不排除其它类型。哈希函数的特性是,相对于其输入,其具有不可预测的输出。因此,该搜索只能通过强力执行,从而在试图解决难题的每个区块链节点104处消耗大量的处理资源。

[0369] 解决难题的第一区块链节点104在网络106上宣布难题解决,提供解决方案作为证明,然后网络中的其它区块链节点104则可以轻松检查该解决方案(一旦给出哈希值的解决方案,就可以直接检查该解决方案是否使哈希值的输出满足条件)。第一区块链节点104将一个区块传播到接受该区块的其它节点达成阈值共识,从而执行协议规则。然后,有序事务集154被每个区块链节点104记录为区块链150中的新区块151。区块指针155还分配给指向该区块链中先前创建的区块151_{n-1}的新区块151_n。创建工作量证明解所需的大量工作(例如采用哈希的形式)发出信号通知第一节点104的意图以遵循区块链协议。这些规则包括如果它分配与先前核实有效的事务相同的输出,则不接受事务为有效,否则称之为重复花费。一旦创建,区块151就不能修改,因为它在区块链网络106中的每个区块链节点104处进行标识和维护。区块指针155还向区块151施加顺序。由于事务152记录在网络106中每个区块链节点104处的有序区块中,因此提供了事务的不可改变公共分类账。

[0370] 应当注意的是,在任何给定时间争相解决难题的不同区块链节点104可以基于在任何给定时间尚未发布的事务的池154的不同快照来这样做,具体取决于它们何时开始搜索解或接收事务的顺序。解决相应难题的人员首先定义新区块151_n中包括的事务152及其顺序,并且更新当前的未发布事务池154。然后,区块链节点104继续争相从新定义的未发布事务有序池154中创建区块,等等。此外,还存在解决可能出现的任何“分叉”的协议,其中两个区块链节点104彼此在很短的时间内解决难题,从而在节点104之间传播区块链的冲突视图。简言之,分叉方向最长的成为最终区块链150。应当注意的是,这不会影响网络的用户或代理,因为同一事务将出现在两个分叉中。

[0371] 根据比特币区块链(和大多数其它区块链),成功构造新区块104的节点被授予在分配附加限定数量数字资产的新特殊类型事务中新分配附加的、接受的数额的数字资产的能力(与代理间或用户间事务相反,该事务将一定数量的数字资产从一个代理或用户转移到另一个代理或用户)。这种特殊类型的事务通常称为“coinbase事务”,但是也可以称为“启动事务”或“产生事务”。它通常形成新区块151_n的第一事务。工作量证明发出信号通知构造新区块的节点的意图以遵循协议规则,从而允许稍后赎回该特定事务。在可以赎回该特殊事务之前,区块链协议规则可能需要成熟期,例如100个区块。通常,常规(非生成)事务152还将在其输出中的一个输出中指定附加事务费用,以进一步奖励创建其中发布该事务的区块151_n的区块链节点104。该费用通常称为“事务费用”,并在下文中讨论。

[0372] 由于事务核实和发布中涉及的资源,通常至少每个区块链节点104采用包括一个或多个物理服务器单元的服务器形式,或者甚至整个数据中心。但是,原则上来说,任何给定区块链节点104均可采用一个用户终端或联网在一起的一组用户终端的形式。

[0373] 每个区块链节点104的存储器均存储被配置为在区块链节点104的处理装置上运行的软件,以根据区块链节点协议执行其相应的角色并处理事务152。应当理解的是,在本文中归因于区块链节点104的任何动作均可通过在相应计算机设备的处理装置上运行的软件执行。节点软件可以在应用层或诸如操作系统层或协议层的较低层或这些层任意组合的一个或更多个应用中实现。

[0374] 扮演消费用户角色的多方103中的每一方的计算机设备102也连接到网络101。这些用户可以与区块链网络106交互,但不参与核实事务或构造区块。其中一些用户或代理103可以充当事务中的发送者和接收者。其它用户可以与区块链150交互,而不必充当发送者或接收者。例如,一些当事方可以充当存储区块链150的副本(例如,已经从区块链节点104获得区块链的副本)的存储实体。

[0375] 各方103中的一些或所有当事方可以作为不同网络的一部分连接,例如覆盖在区块链网络106之上的网络。区块链网络的用户(经常称为“客户端”)可以被称为是包含区块链网络106的系统的一部分;然而,这些用户不是区块链节点104,因为它们不执行区块链节点所需的角色。相反,每一方103可以与区块链网络106交互,从而通过连接到区块链节点106(即,与区块链节点106通信)来利用区块链150。出于说明目的,示出了双方103及其相应的设备102:第一方103a及其相应的计算机设备102a,以及第二方103b及其相应的计算机设备102b。应当理解的是,更多此类当事方103及其相应的计算机设备102可能存在并参与系统100,但为了方便起见,未进行说明。每一方103均可以是个人或组织。仅出于说明目的,在本文中,第一方103a称为爱丽丝,第二方103b称为鲍勃,但应当理解的是,这并不仅限于爱丽丝或鲍勃,且本文对爱丽丝或鲍勃的任何引用均可分别用“第一方”和“第二方”替换。

[0376] 每一方103的计算机设备102包括相应的处理装置,其包括一个或更多个处理器,例如一个或更多个CPU、图形处理单元(GPU)、其他加速器处理器、特定应用程序处理器和/或FPGA。每一方103的计算机设备102还包括存储器,即采用非暂时性计算机可读介质形式的计算机可读存储器。该存储器可包括一个或更多个存储器单元,其采用一个或更多个存储器介质,例如诸如硬盘等磁介质、诸如SSD、闪存或EEPROM等电子媒介和/或诸如光盘驱动器等的光学介质。每一方103的计算机设备102上的存储器存储软件,其包括被设置为在处理装置上运行的至少一个客户端应用程序105的相应实例。应当理解的是,在本文中归因于给定方103的任何行动均可通过在相应计算机设备102的处理装置上运行的软件执行。每一方103的计算机设备102包括至少一个用户终端,例如台式或笔记本电脑、平板电脑、智能手机或诸如智能手表等的可穿戴设备。给定方103的计算机设备102还可包括一个或更多个其他网络资源,诸如通过用户终端访问的云计算资源。

[0377] 客户端应用程序105最初可通过例如从服务器下载的适当计算机可读存储介质,或通过诸如可移动SSD、闪存密钥、可移动EEPROM、可移动磁盘驱动器、软盘或磁带等的可移动存储设备、诸如CD或DVD ROM等的光盘或可移动光驱等提供至任何给定方103的计算机设备102。

[0378] 客户端应用程序105至少包括“钱包”功能。这有两个主要功能。其中一个功能是使相应方103能够创建、授权(例如签名)事务152并将其发送到一个或更多个位比特币节点104,然后在区块链节点104的网络中传播,从而包括在区块链150中。另一个功能是向相应方汇报其目前拥有的数字资产数额。在基于输出的系统中,该第二功能包括整理分散在区

块链150中属于相关方的各种事务152的输出中定义的数额。

[0379] 注意：虽然各种客户端功能可以描述为集成到给定客户端应用程序105中，但这不一定是限制性的，相反，在本文中所描述的任何客户端功能可以在由两个或更多个不同应用程序组成的套件中实现，例如经由API进行接口连接或一个应用程序作为另一个应用程序的插件。更通俗地说，客户端功能可以在应用层或诸如操作系统的较低层或这些层的任意组合实现。下面将根据客户端应用程序105进行描述，但应当理解的是，这不是限制性的。

[0380] 每个计算机设备102上的客户端应用程序或软件105的实例可操作地耦合到网络106的区块链节点104中的至少一个。这可以启用客户端105的钱包功能，以将事务152发送至网络106。客户端105还可联络区块链节点104，以在区块链150中查询相应方103作为接收者的任何事务（或实际上在区块链150中检查其它方的事务，因为在实施例，区块链150是在某种程度上通过其公开可见性提供事务信任的公共设施）。每个计算机设备102上的钱包功能被配置为根据事务协议制定和发送事务152。如上所述，每个区块链节点104运行软件，该软件被配置为根据区块链节点协议核实事务152并转发事务152以便在区块链网络106中传播。事务协议和节点协议相互对应，给定事务协议和给定节点协议一起实现给定的事务模型。相同的事务协议用于区块链150中的所有事务152。网络106中的所有节点104使用相同的节点协议。

[0381] 当给定方103（比方说爱丽丝）希望发送拟包含在区块链150中的新事务152_j时，她将根据相关事务协议（使用其客户端应用程序105中的钱包功能）制定新事务。然后，她将事务152从客户端应用程序105发送到她所连接的一个或更多个区块链节点104。例如，这可能是与爱丽丝的计算机102最佳连接的区块链节点104。当任何给定区块链节点104接收新事务152_j时，其将根据区块链节点协议及其相应的角色进行处理。这包括首先检查新接收的事务152_j是否满足变为“有效”的特定条件，具体示例稍后将详细讨论。在一些事务协议中，有效条件可通过事务152中包含的脚本在每个事务的基础上进行配置。或者，条件可仅仅是节点协议的内置功能，或通过组合脚本和节点协议进行定义。

[0382] 如果新接收的事务152_j通过有效性测试（即：“有效”的条件下），接收事务152_j的任何区块链节点104将向在区块链节点104处维护的有序事务集154中添加新的核实有效事务152。进一步地，接收事务152_j的任何区块链节点104随后将核实有效事务152传播至网络106中的一个或更多个其它区块链节点104。由于每个区块链节点104应用相同的协议，因此假定事务152_j有效，这意味着事务很快将在整个网络106中传播。

[0383] 一旦进入在给定区块链节点104处维护的未决事务有序池154，该区块链节点104将开始争相解决其各自的包含新事务152的池154的最新版本上的工作量证明难题（请记住，其它区块链节点104可以尝试基于不同的事务池154来解决难题。但是，首先解决难题的人将定义包括在最新区块151中的事务集合。最终，区块链节点104将解决有序池154的一部分的难题，该有序集154包括爱丽丝的事务152_j）。一旦包括新事务152_j的池154完成工作量证明，其将不可变地成为区块链150中区块151中的一个区块的一部分。每个事务152包括指向早前事务的指针，因此事务的顺序也被不可变地记录下来。

[0384] 不同的区块链节点104可以首先接收给定事务的不同实例，并且因此在一个实例被发布到新区块151中之前具有关于哪个实例“有效”的冲突视图，此时所有区块链节点104同意所发布的实例是唯一的有效实例。如果区块链节点104将一个实例接受为有效实例，然

后发现第二实例已记录在区块链150中,则区块链节点104必须接受这一点,并将丢弃(即,视为无效)其最初接受的实例(即,在区块151中尚未公布的实例)。

[0385] 作为基于账户的事务模型的一部分,由一些区块链网络操作的另一种类型的事务协议可称为“基于账户的”协议。在基于账户的情况下,每个事务均不通过参考过去事务序列中先前事务的UTX0来定义转移的数额,而是通过参考绝对账户余额进行定义。所有账户的当前状态由网络的节点单独存储到区块链中,并不断更新。在此类系统中,事务使用账户的运行事务记录(也称为“头寸”)进行排序。该值由发送者签名作为其加密签名的一部分,并作为事务引用计算的一部分进行哈希处理。此外,可选的数据字段也可以在事务中签名。例如,如果数据字段中包含上一事务的ID,该数据字段可指向上一事务。

[0386] 4.2. 基于UTX0的模型

[0387] 图2示出了示例性事务协议。这是基于UTX0的协议的示例。事务152(简称“Tx”)是区块链150的基本数据结构(每个区块151包括一个或更多个事务152)。下面将通过参考基于输出或基于“UTX0”的协议进行描述。但这并不限于所有可能的实施例。应当注意的是,虽然参考比特币描述了示例性基于UTX0的协议,但是它同样可以在其它示例区块链网络上实现。

[0388] 在基于UTX0的模型中,每个事务(“Tx”)152包括数据结构,其包括一个或更多个输入202和一个或更多个输出203。每个输出203可包括未花费事务输出(UTX0),其可用作另一新事务的输入202的来源(如果UTX0尚未赎回)。UTX0包括指定数字资产数额的值。这表示分布式分类账上的一组通证。UTX0还可包含其来源事务的事务ID以及其它信息。事务数据结构还可包括标头201,其可包括输入字段202和输出字段203的大小指示符。标头201还可包括事务的ID。在实施例中,事务ID是事务数据(不含事务ID本身)的哈希值,且存储在提交至节点104的原始事务152的标头201中。

[0389] 比方说爱丽丝103a希望创建转移相关数字资产数额至鲍勃103b的事务152j。在图2中,爱丽丝的新事务152j标记为“Tx₁”。该新事务获取在序列中先前事务152i的输出203中锁定至爱丽丝的数字资产数额,并至少将此类数额中的一部分转移至鲍勃。在图2中,先前事务152i标记为“Tx₀”。Tx₀和Tx₁只是任意的标记,其不一定意味着Tx₀指区块链151中的第一事务且Tx₁指池154中的下一事务。Tx₁可指向仍具有锁定至爱丽丝的未花费输出203的任何先前(即先行)事务。

[0390] 当爱丽丝创建其新事务Tx₁时,或至少在她将该新事务发送至网络106时,先前事务Tx₀可能已经有效并包括在区块链150的区块151中。该事务此时可能已包括在区块151中的一个区块中,或者可能仍在有序集154中等待,在这种情况下,该事务将很快包括在新区块151中。或者,Tx₀和Tx₁可以创建并一起发送至网络106;或者,如果节点协议允许缓冲“孤立”事务,Tx₀甚至可以在Tx₁之后发送。本文事务序列上下文中使用的“先前”和“后续”一词是指由事务中指定的事务指针定义的序列中的事务顺序(哪个事务指向哪个其他事务等等)。它们同样可以替换为“前任”和“继任”、“先行”和“后代”或“父项”和“子项”等。这不一定指其创建、发送至网络106或到达任何给定区块链节点104的顺序。然而,指向先前事务(先行事务或“父事务”)的后续事务(后代事务或“子事务”)不会有效除非父事务有效。在父事务之前到达区块链节点104的子事务被视为孤立事务。根据节点协议和/或节点行为,其可被丢弃或缓冲一段时间,以等待父事务。

[0391] 先前事务 Tx_0 的一个或多个输出203中的一个包括特定的UTX0,标记为 $UTX0_0$ 。每个UTX0包括指定UTX0表示的数字资产数额的值以及锁定脚本,该锁定脚本定义后续事务的输入202中的解锁脚本必须满足的条件,以使后续事务有效,从而成功赎回UTX0。通常情况下,锁定脚本将数额锁定至特定方(该数额的事务的受益人)。即,锁定脚本定义解锁条件,该解锁条件通常包括以下条件:后续事务的输入中的解锁脚本包括先前事务被锁定到的一方的加密签名。

[0392] 锁定脚本(亦称scriptPubKey)是节点协议识别的域特定语言中写入的一段代码。此类语言的特定示例称为“脚本(Script)”(S大写),其可由区块链网络所使用。锁定脚本指定花费事务输出203所需的信息,例如爱丽丝签名的要求。解锁脚本出现在事务的输出中。解锁脚本(亦称scriptSig)是提供满足锁定脚本标准所需信息的域特定语言中写入的一段代码。例如,其可包含鲍勃的签名。解锁脚本出现在事务的输入202中。

[0393] 因此在示出的示例中, Tx_0 的输出203中的 $UTX0_0$ 包括锁定脚本[Checksig P_A],该锁定脚本需要爱丽丝的签名Sig P_A ,以赎回 $UTX0_0$ (严格来说,是为了使试图赎回 $UTX0_0$ 的后续事务有效)。 $[Checksig P_A]$ 包含爱丽丝的公私密钥对中的公钥 P_A 的表示(即哈希)。 Tx_1 的输入202包括指向 Tx_1 的指针(例如,通过其事务ID($TxID_0$),其在实施例中的是整个事务 Tx_0 的哈希值)。 Tx_1 的输入202包括在 Tx_0 中标识 $UTX0_0$ 的索引,以在 Tx_0 的任何其他可能输出中对其进行标识。 Tx_1 的输入202进一步包括解锁脚本<Sig P_A >,该解锁脚本包括爱丽丝的加密签名,该签名由爱丽丝通过将其密钥对中的私钥应用于预定的部分数据(有时在密码学中称为“消息”)创建。爱丽丝需要签名以提供有效签名的数据(或“消息”)可通过锁定脚本、节点协议或其组合进行定义。

[0394] 当新事务 Tx_1 到达区块链节点104时,该节点应用节点协议。这包括一起运行锁定脚本和解锁脚本,以检查解锁脚本是否满足锁定脚本中定义的条件(其中该条件可包括一个或多个标准)。在实施例中,这涉及并置两个脚本:

[0395] <Sig PA >< PA >||[Checksig PA]

[0396] 其中“||”表示并置,“<...>”表示将数据放在堆栈上,“[...]”表示由锁定脚本组成的函数(在该示例中指基于堆栈的语言)。同样,脚本可以使用公共堆栈一个接一个地运行,而不是并置脚本。无论采用哪种方式,当一起运行时,脚本使用爱丽丝的公钥 P_A (包括在 Tx_0 的输出的锁定脚本中),以认证 Tx_1 的输入中的解锁脚本是否包含爱丽丝签名预期部分的数据时的签名。也需要包括预期的部分数据本身(“消息”),以便执行此认证。在实施例中,签名的数据包括整个 Tx_1 (因此不需要包括一个单独的元素来明文指定签名的部分数据,因为其本身便已存在)。

[0397] 本领域技术人员将熟悉通过公私密码进行验证的细节。基本上而言,如果爱丽丝已使用其私钥加密签署消息,则给定爱丽丝的公钥和明文中的消息,诸如节点104等其它实体可验证消息必须已经由爱丽丝签名。签署通常包括对消息进行哈希,签署哈希值和将此标记到消息作为签名,从而使公钥的任何持有者能够验证签名。因此,应当注意的是,在实施例中,在本文中对签名特定数据片段或事务部分等的任何引用可以意味着对该数据片段或事务部分的哈希值进行签名。

[0398] 如果 Tx_1 中的解锁脚本满足 Tx_0 的锁定脚本中指定的一个或多个条件(因此,在所示示例中,如果在 Tx_1 中提供了爱丽丝的签名并进行验证),则区块链节点104认为 Tx_1 有效。

这意味着区块链节点104会将 Tx_1 添加到待处理事务有序池154。区块链节点104还会将事务 Tx_1 转发到网络106中的一个或多个其它区块链节点104,以便其会在整个网络106中传播。一旦 Tx_1 有效并包括在区块链150中,这会将 $UTXO_0$ 从 Tx_0 定义为已花费。应当注意的是, Tx_1 仅在花费未花费事务输出203时才有效。如果其试图花费另一事务152已经花费的输出,则即使满足所有其它条件, Tx_1 也将无效。因此,区块链节点104还需要检查先前事务 Tx_0 中引用的 $UTXO$ 是否已经花费(即,其是否已经形成另一有效事务的有效输入)。这是为何区块链150对事务152施加定义的顺序很重要的原因之一。在实践中,给定区块链节点104可维护单独的数据库,标记已花费事务152的 $UTXO$ 203,但最终定义 $UTXO$ 是否已花费取决于是否在区块链150中形成了另一有效事务的有效输入。

[0399] 如果给定事务152的所有输出203中指定的总数额大于其所有输入202所指向的总数额,则这是大多数事务模型中的另一失效依据。因此,此类事务不会传播或包括在区块链151中。

[0400] 请注意,在基于 $UTXO$ 的事务模型中,给定 $UTXO$ 需要作为一个整体使用。不能“留下” $UTXO$ 中定义为已花费的一部分数额,而同时又花费另一部分。但 $UTXO$ 的数额可以在下一事务的多个输出之间分割。例如, Tx_0 的 $UTXO_0$ 中定义的数额可以在 Tx_1 中的多个 $UTXO$ 之间分割。因此,如果爱丽丝不想将 $UTXO_0$ 中定义的所有数额都给鲍勃,她可以使用剩余部分在 Tx_1 的第二输出中自己找零,或者支付给另一方。

[0401] 在实践中,爱丽丝通常还需要包括用于比特币节点104的费用,该比特币节点104在区块151中成功包含爱丽丝的事务104。如果爱丽丝未包括此类费用,则 Tx_0 可能会被区块链节点104拒绝,并且因此尽管在技术上有效,但可能不会传播并且包括在区块链150中(如果区块链节点104不希望接受事务152,节点协议不强迫区块链节点104接受)。在一些协议中,事务费用不需要其自身的单独输出203(即不需要单独的 $UTXO$)。相反,输入202指向的总数额与给定事务152的输出203指定的总数额之间的任何差额都将自动提供给发布事务的区块链节点104。例如,假设指向 $UTXO_0$ 的指针是 Tx_1 的唯一输入,并且 Tx_1 仅具有一个输出 $UTXO_1$ 。如果在 $UTXO_0$ 中指定的数字资产数额大于在 $UTXO_1$ 中指定的数额,则可以由赢得工作量证明竞赛以创建包含 $UTXO_1$ 的区块的节点104分配该差值。替代地或附加地,这不一定排除可以在其自身事务152的其中一个 $UTXO$ 203中明确指定事务费用。

[0402] 爱丽丝和鲍勃的数字资产由区块链150中任何位置的任何事务152中的锁定至他们的 $UTXO$ 组成。因此,通常情况下,给定方103的资产分散在整个区块链150的各种事务152的 $UTXO$ 中。区块链150中的任何位置均未存储定义给定方103的总余额的一个数字。客户端应用程序105的钱包功能的作用是将锁定至相应方且在其它随后事务中尚未花费的各种 $UTXO$ 值整理在一起。为实现这一点,其可以查询存储在任何一个比特币节点104处的区块链150的副本。

[0403] 应当注意的是,脚本代码通常用示意图表示(即使用非精确语言)。例如,可以使用操作码(opcode)来表示特定功能。“OP...”是指脚本语言的特定操作码。举例来说,OP_RETURN是脚本语言操作码,当在锁定脚本的开始处在操作码前加上OP_FALSE时,操作码创建事务的不可花费输出,该输出可以在事务内存储数据,从而将数据不可改变地记录在区块链150中。例如,数据可包括需存储在区块链中的文件。

[0404] 通常,事务的输入包含对应于公钥PA的数字签名。在实施例,这基于使用椭圆曲

线secp256k1的ECDSA。数字签名对特定的数据段进行签名。在实施例中,对于给定事务,签名将对部分事务输入以及部分或全部事务输出进行签名。对输出的特定部分进行签名取决于SIGHASH标志。SIGHASH标志通常是包含在签名末尾的4字节代码,用于选择签名的输出(并因此在签名时固定)。

[0405] 锁定脚本有时称为“scriptPubKey”,指其通常包括相应事务被锁定到的当事方的公钥。解锁脚本有时称为“scriptSig”,指其通常提供相应的签名。但是更通俗地说,在区块链150的所有应用中,UTXO赎回的条件并不一定包括对签名进行验证。更通俗地说,脚本语言可用于定义任何一个或更多个条件。因此,可以优选更为通用的术语“锁定脚本”和“解锁脚本”。

[0406] 4.3. 侧信道

[0407] 如图1所示,爱丽丝和鲍勃的计算机设备102a、120b中的每个计算机设备上的客户端应用程序都可以包括附加通信功能。此附加功能可使爱丽丝103a建立与鲍勃103b的单独侧信道107(在任何一方或第三方的鼓动下)。侧信道107使得能够脱离区块链网络交换数据。此类通信有时称为“链下”通信。例如,这可用于在爱丽丝与鲍勃之间交换事务152,而不将该事务(尚未)注册到区块链网络106上或将其发布到链150上,直到其中一方选择将其广播到网络106上。以这种方式共享事务有时称为共享“事务模板”。事务模板可能缺少形成完整事务所需的一个或更多个输入和/或输出。替代地或附加地,侧信道107可用于交换任何其它事务相关数据,例如密钥、议付数额或条款、数据内容等。

[0408] 通过与区块链网络106相同的分组交换网络101可建立侧信道107。替代地或附加地,侧信道301可以经由诸如移动蜂窝网络的不同网络或者诸如无线局域网的局域网建立,甚至经由爱丽丝和鲍勃的设备102a、102b之间的直接有线或无线链路建立。通常,在本文中任何地方所指的侧信道107可以包括经由一项或更多项联网技术或通信介质的任何一条或更多条链路,这些链路用于“链下”交换数据,即脱离区块链网络106交换数据。在使用多条链路的情况下,链下链路束或集合整体上可以称为侧信道107。因此,应当注意的是,如果说爱丽丝和鲍勃通过侧信道107交换某些信息或数据等,则这不一定意味着所有这些数据都必须通过完全相同的链路或甚至相同类型的网络发送。

[0409] 侧信道107可以包括采用已知安全通信技术的安全信道,以在爱丽丝和鲍勃等各方之间实现安全、私有的链下通信。例如,安全信道可以基于通过安全信道进行通信的各方之间共享的共享秘密。例如,可以使用此类信道在验证方103V和目标方103T之间进行通信,从而使验证方103V能够向目标方持有的PUF 302/500提交质询,并接收回对应的响应。

[0410] 5. 基于区块链的PUF身份证明

[0411] 如前几节所述,用作响应记录的响应数据可以存储在公共区块链上,而不是采用可信第三方系统602。响应数据是在建立时确定的数据,稍后可以由验证方103V(“鲍勃”)用来测试目标方103T(“爱丽丝”)对目标身份的断言。再次应当注意的是,爱丽丝和鲍勃只是任意标签,爱丽丝和鲍勃在这里的角色不一定与第4节(其中鲍勃花费爱丽丝的事物的输出)中给出的区块链系统总体概述中的角色相同。

[0412] 如前所述,给定CR对 $\{C_i, R_i\}$ 的响应数据(无论是存储在链上还是其他地方)可以包括在建立阶段702确定并存储以供验证方103V将来参考的以下各项中的任一项:

[0413] i) 质询 C_i 和/或响应 R_i 的显式值(明文或加密);或

[0414] ii) 链接到主质询 C_m 的响应 R_i 的显式值,从该主质询可以导出相应响应 R_i 的特定质询 C_i ;或

[0415] iii) 响应 R_i 的证明(例如,哈希或双重哈希)以及质询 C_i 的显式值;或

[0416] iv) 链接到主质询 C_m 的响应 R_i 的证明(例如,哈希或双重哈希),从该主质询可以导出相应响应 R_i 的特定质询;或

[0417] v) 从响应 R_i 导出的公钥-私钥对的公钥。

[0418] 如图9所示,无论采取何种形式,在建立阶段702,此类响应数据901可以存储在区块链150上记录的事务152S的输出203中。这在下文中可以称为存储事务。例如,可以使用上面第4节中讨论的技术将其记录在链上,再次应当注意的是,该节中的爱丽丝不一定是目标方103T,该节中的鲍勃不一定是验证方103V——事实上,现在被称为爱丽丝的目标方103T可以是制定并发送要记录在链上的存储事务152S的人。在另一示例中,可信第三方可以为目标方103T制定存储事务模板,通过包括在建立时生成的响应数据901完成该模板,然后转发到链上记录。目标方103T可以将存储事务152S直接发送给区块链节点104中的一个节点,以通过区块链网络106传播,或者可以经由可信第三方等另一方间接发送。在又一示例中,目标方103T可以将其响应数据901发送给可信第三方,以便可信第三方将该响应数据制定为存储事务152并发送到链上记录。

[0419] 响应数据901可以存储在存储事务152S的不可花费输出中。例如,如果使用脚本协议(在BTC或BCH等区块链协议中,OP_RETURN的任何包含都会使输出不可花费,而在BSV等其他协议中,OP_FALSE和OP_RETURN是使输出不可花费所必需的),则可以通过OP_RETURN或OP_FALSE后跟OP_RETURN使输出不可花费。比特币(BTC)、比特币现金(BTC)和比特币SV(BSV)是前面描述的区块链系统的不同示例性实现方式。

[0420] 或者,响应数据901可以嵌入存储事务152S的可花费输出中。例如,可以通过包含OP_RETURN而不包含OP_FALSE来保持可花费。在另一示例中,可以将数据直接包含在OP_DROP代码之前,从而将数据嵌入到可花费的锁定脚本中。这同样适用于BTC、BCH和BSV。

[0421] 在实施例中,可以存储给定目标方103T的多个不同CR对 $\{C_i, R_i\}$ 的集的响应数据901。这些响应数据可以存储在存储事务152的相同输出203或不同输出203中,或采用一些存储在相同输出中而一些存储在不同输出中的组合。这些响应数据可以存储在相同的存储事务152S中,或不同CR对的响应数据901可以存储在不同的存储事务152S中,或采用一些存储在相同事务中而一些存储在不同事务中的组合。

[0422] 应当注意的是,链上存储不一定局限于基于帐户的模型。在替代部署中,响应数据901可以存储在基于帐户的模型的一个或多个事务的一个或多个智能合约中。

[0423] 在验证阶段704,当验证方103V希望验证目标的身份时,他/她访问区块链150,以便从存储事务152S获得与特定CR对相对应的响应数据901。在实施例中,这向验证方103V提供了与特定质询 C_i 相对应的响应 R_i ,或该响应 R_i 的证明(例如,哈希或双重哈希)。验证方103V还向目标方103V提交质询 C_i ,并且作为响应,接收回目标方103T(或其设备)通过将接收的质询 C_i 输入到PUF模块603而生成的(声称的)响应 R'_i 。然后,验证方103V将返回的响应 R'_i 与从链上的存储事务152S中检索的版本进行比较,或对接收的用于证明的响应应用相同的变换(例如, $H(R'_i)$ 或 $H^2(R'_i)$),并将其与从链上的存储事务152S中检索的证明进行比较。无论采用哪种方式,如果比较结果匹配,则验证目标。

[0424] 验证方103V可以通过区块链网络106的节点104中的一个节点访问区块链150,或替代地,通过从任何外部方获得响应数据来访问区块链150,该外部方也可以提供该数据(即,事务)包含在区块链中的默克尔证明。

[0425] 在响应数据901存储在区块链150等公共介质上的实施例,可能希望实际响应值 R_i 本身不被公开或不被不受限制地公开。否则,任何恶意方都可以查看链上的 R_i ,然后在受到 C_i 质询时假装是目标方103T。因此,相反,可以优选地仅存储 R_i 的证明(例如, $H(R_i)$ 或 $H^2(R_i)$)作为保存在链上的响应数据901,或以加密形式存储 R_i 的显式值。或者在某些情况下,证明可以以加密形式存储在链上。

[0426] 在可能存在多个验证方的情况下,以加密形式存储 R_i 或其证明使得目标方103T或可信第三方能够控制哪些验证方103V能够获得与CR对中的哪些CR对相对应的存储数据901。这可以通过以下方式实现:仅将响应数据901的某个片段的解密密钥提供给给定的验证方,并且仅将响应数据901的另一个片段的解密密钥提供给另一验证方。解密密钥的分发可以由目标方103T或可信第三方管理。向每个验证方或验证方的子集提供其自己的一个或多个解密密钥的子集,用于访问响应数据901的各片段的相应子集(例如,CR对)。优选地,这些子集相互排斥。然而,在其他实现方式中,这些子集可以重叠(例如,同一组织内的不同组可以访问CR对的重叠子集)。

[0427] 作为一种变体,如果响应数据901(例如,CR对)存储在第三方系统602中而不是存储在链上,则可以采用其他手段来确保每个验证方只能访问其自己的CR对子集(或更一般地,响应数据),而不是(或同时)分发解密密钥。例如,可信第三方系统602可以为每个验证方维护受密码保护的帐户,要求他们登录该帐户才能访问他们的一个或多个质询,并且该帐户只允许他们访问自己的一个或多个CR对。

[0428] 此类方案可能有利于安全性。如果要向一个验证方103V公开给定CR对的响应 R_i ,则可能希望不要将同一CR对用于另一验证方103V。否则,第一验证方103V可以使用现在已知的响应 R_i 向另一验证方假装他/她是目标方103T。然而,如果可以访问响应数据901的所有潜在验证方103V都是可信的,那么采取步骤防止这种情况就不是必要的。

[0429] 在进一步的变体中,存储在链上的响应数据901可以采取目标方103T的公钥的形式,该公钥是在建立时基于对应的响应 R_i 生成的公钥-私钥对的公钥(例如,将其用作种子)。在这种情况下,验证方103V从存储事务152S访问公钥,并使用该公钥验证由目标方103T使用对应私钥进行签名的消息。在一些情况下,公钥可以以加密形式存储在链上,以便可以为不同的验证方103V分配不同的公钥。

[0430] 同样,如图9所示,在采用输出(例如,基于UTXO的)模型的实施例,可以利用这一点来提供用于管理CR对(或从中导出的密钥)的有效机制。这里的管理可以包括更新或撤销CR对或密钥,例如,一旦该CR对或密钥已经消耗(在验证中使用)。

[0431] 为此,在区块链150上记录新的修改事务152M。该修改器事务具有指向存储事务152S的输出203中的一个输出的输入202,其中存储了要撤销或更新的响应数据901的片段。这可以称为“花费”、“兑换”或“分配”该输出(但应当注意的是,这并不一定意味着货币价值的转移)。在验证方103V识别的第二层协议级别上,这意味着不再使用指向的存储事务152S或输出203中的响应数据901。如果修改器事务152M本身在自己的输出中的一个输出中包含响应数据901',则这表示新的响应数据901'表示替换之前的响应数据901(例如,新的CR

对)。如果验证方访问区块链150以查找在验证操作中使用的响应数据,则将使用更新的响应数据901',而不是替换的响应数据。另一方面,如果修改器事务152U不包含替换响应数据901,则只需撤销其所指向的存储事务152S或输出203中的响应数据901。

[0432] 在一些实施例中,响应数据901嵌入到存储事务152S的可花费输出中,并且可以通过花费(即,分配或兑换)存储响应数据901(例如,CR对)的特定输出203进行撤销或更新。在一些此类实施例中,与不同CR对相对应的响应数据901的不同片段可以存储在单一存储事务203的单独输出203中,并且可以单独调用或更新。

[0433] 在其他实施例中,响应数据901存储在存储事务152S的不可花费输出中,可以通过花费(即,分配或兑换)存储事务152S的不同可花费输出进行撤销或更新。在一些此类实施例中,可以通过花费同一事务152S的相同可花费输出撤销或更新与多个不同CR对(存储在相同或不同的不可花费输出中)相对应的存储数据901的多个片段。

[0434] 作为示例性用例,与CR对相对应的响应数据901的片段记录一旦消耗(即,在验证中使用),就可以被撤销或更新。无论响应数据901是Ri的显式记录、证明、还是从Ri导出的公钥,这都适用。无论哪种方式,出于安全原因,这都可能是有利的,因此现在已经发布到世界上的响应数据不再可用。

[0435] 修改器事务152M可以由目标方103T制定并发送到链上记录。该事务可以直接发送到区块链节点104进行传播,也可以经由中间方间接发送到节点104。或者,可信第三方可以发送模板事务,让目标方完成(例如,通过签名和/或添加替换响应数据901'),然后直接或间接地转发到节点104,以在链上记录。作为另一种可能性,可信第三方可以制定修改器事务152M(可能基于模板或从目标方103T发送的一些数据,例如,包括替换响应数据901'),然后可信第三方可以将修改器事务152M发送到节点104,以在链上记录。应当注意的是,所有这些选项也可以应用于在区块链150上记录存储事务152S的方式。

[0436] 根据上面讨论的各种概念,提供了一种系统,用于i)将身份(或公钥等其他相关信息)链接到UTX0,并使用该UTX0的花费状态作为身份凭证有效性的代理;以及ii)建立一组事务来执行高效的身份管理操作,例如建立、撤销、更新和验证。该过程的效率更高,因为通信数量减少了,因为各方都可以咨询区块链,查看何时消耗CRP或何时撤销身份,而不是各方都需要一直相互通信。

[0437] 例如,如前所述,通过最大限度地减少对第三方KYC(know your customer,了解客户)提供者处理验证中使用的CRP数据的依赖,此类技术可以用于扩展将身份链接到PUF设备的框架。这一目标可以通过用公共区块链部分取代KYC提供者的角色或更确切地说是一些函数来实现,由此用户可以独立于任何第三方实例化自己的与ePUF设备相关的身份凭证。

[0438] 不一定完全避开可信第三方在身份系统中的角色,但无论采用哪种方式,身份管理过程都可以得到改善,从而至少减少他们在该过程中的参与以及给他们带来的相关负担。

[0439] 5.1.将PUF身份链接到UTX0集

[0440] 如前几节中所讨论的,使用区块链可以改善身份系统的第一方面是通过使用公共区块链的未花费事务输出集(UTX0集)来管理与PUF身份相关的CRP。

[0441] 在本节中,公开了两种不同的示例性机制,用于将CRP映射到UTX0集的成员,并使

用它们的‘已花费’或‘未花费’状态来指示每个特定CRP是否在身份验证过程中消耗。第一种机制涉及将CRP数据嵌入到可花费UTX0中,第二种机制涉及将CRP数据与可花费UTX0配对。在任何一种情况下,与CRP或有关身份相关的附加数据也可以可选地包含在系统中。

[0442] 5.1.1. 嵌入可花费UTX0:第一种机制是将CRP绑定到可花费UTX0,这些UTX0是包含脚本的事务输出,其条件可以由未来的输入满足,因此可以由未来的花费事务消耗。

[0443] 实现此类嵌入有许多不同的选项,但就本文的目的而言,通常至少包括以下锁定脚本:

[0444] $[Checksig\ P]OP_RETURN\langle Rep(C,R)\rangle$

[0445] 其中 $[Checksig\ P]$ 是标准支付到公钥哈希(P2PKH)锁定脚本, $Rep(C,R)$ 是特定质询-响应对 (C,R) 的表示。

[0446] 可以简单地通过对花费事务提供有效签名 $Sig\ P$ 来解锁该锁定脚本,其中签名被视为针对公钥P的有效签名。应当注意的是,在核实花费事务时,不会考虑操作码 OP_RETURN 后面的任何数据,因此对于区块链核实者来说,该数据可以被视为任意的无格式数据。

[0447] 上面脚本中 OP_RETURN 码后面的数据是质询-响应对 (C,R) 的表示 $Rep(C,R)$ 。这种表示可以有多种方式,具体取决于有关用例。然而,合理的示例是使用只有拥有PUF的证明者爱丽丝知道的密钥 k 来加密CRP。在这种情况下,可以使用以下任何一种表示:

[0448] $Rep(C,R) = Encrypt(C,k)$,

[0449] $Rep(C,R) = Encrypt(R,k)$,

[0450] $Rep(C,R) = Encrypt(C || R,k)$ 。

[0451] 这些表示将允许爱丽丝在稍后分别检索或证明其UTX0中包含的质询、响应或CRP。

[0452] 额外的脚本负担:可以扩展前面示出的基本锁定脚本,以便在将来花费输出的输入脚本中包含附加条件。此类额外条件的合理示例是以下脚本:

[0453] $[Checksig\ P][Hash\ PuzzleH^2(R)]OP_RETURN\langle Rep(C,R)\rangle$

[0454] 其中 $[Hash\ PuzzleH^2(R)] = OP_HASH160\langle H(R)\rangle OP_EQUAL$ 。应当注意的是,可以使用其他哈希函数操作码。该修改脚本现在要求花费者除了为公钥P提供有效签名外,还要显示质询R的哈希值。这里的思路是,在一些场景中,这可以用作知识证明,证明花费者知道与有关质询R相关的信息 $H(R)$ 。

[0455] 事务模型:假设已经确定要使用的事务锁定脚本的确切结构,那么就可以选择如何构建包含这些脚本的事务,作为一种证明存储、证明和管理CRP的方式。

[0456] 本文公开了将CRP和相关联的锁定脚本一对一地映射到UTX0。换句话说,包含此类脚本的每个UTX0将恰好对应于与特定PUF设备相关的一个CRP。

[0457] 关于如何将这些UTX0组织为事务,有几种选项。最有可能的选项如下:

[0458] 1. 每个事务一个CRP。

[0459] 2. 每个事务一个CRP集。

[0460] 3. 每个事务一个PUF。

[0461] 第一种选项可能适用于某些情况(例如,用于很少使用的PUF、用于更新遗嘱),并且具有多个CRP之间没有明显链接的优点。这在需要极度隐私的情况下也可能有用,因为一个CRP的消耗和披露可以独立于任何其他CRP。

[0462] 下面表1中的事务是第一种选项的示例性实现方式。可以看出,事务仅包括单个输

入和输出,因此每个CRP将包含在不同的事务中。当其输出花费时,除了用于审计目的之外,该事务与身份系统的相关性实际上就结束了。

$TxID_{CRP-Set}$			
输入		输出	
输出点	脚本	值	脚本
$TxID_{prev}$	$< SIG_A > < P_A >$	x BSV	$[Checksig P_A^1] OP_RETURN <Rep(C_1, R_1)>$

[0464] 表1:单个事务中映射到UTXO的单个CRP。

[0465] 第二种选项是在单个事务中将许多CRP分别映射到相应的UTXO,该选项可能更适合于CRP消耗的预期频率相当高的银行卡等用例。下面表2中的事务显示了如何实现这一点。

[0466] 应当注意的是,可能由爱丽丝生成的输入签名可以用于对整个输出集进行签名。这提供了从一个公钥 P_A 到多个UTXO(因此到多个CRP)的一对多链接,同时保持UTXO到不同CRP本身的一对一映射。还假设每个输出/CRP都有自己的关联公钥(全部归爱丽丝所有),以避免重复使用。

$TxID_{CRP-Set}$			
输入		输出	
输出点	脚本	值	脚本
$TxID_{prev}$	$< SIG_A > < P_A >$	x BSV	$[Checksig P_A^1] OP_RETURN <Rep(C_1, R_1)>$
[0467]		x BSV	$[Checksig P_A^2] OP_RETURN <Rep(C_2, R_2)>$
		x BSV	$[Checksig P_A^3] OP_RETURN <Rep(C_3, R_3)>$
		...	
		x BSV	$[Checksig P_A^n] OP_RETURN <Rep(C_n, R_n)>$

[0468] 表2:映射到单个事务的相应UTXO的CRP集

[0469] 上面示出的选项也可以很好地与随时间更新CRP集的实施例集成,其中每次生成更新集时,可以为该集发行新的事务。此外,还可以通过并行独立的(即,在链上不相关的)事务,同时为同一PUF生成并发布多个不同的CRP集。这可能有助于与多个不同的可信第三方(例如,不同的银行)建立身份,这样身份都是独立建立的,但仍由同一PUF固定。

[0470] 第三种选项是使用单个事务来表示单个PUF,这只是第二种选项更严格的版本,其中不可能进行更新。这可能适用于包含PUF的设备被赋予特定‘寿命’的情况,在这种情况下,在向用户发放新设备之前,包含PUF的设备只能用于预定次数的身份验证。

[0471] 5.1.2. 与可花费UTXO配对

[0472] 将CRP嵌入可花费UTXO的一种替代方案是简单地将它们与这些输出配对。在这种情况下,与现有的数字证书工作的不同之处在于,事务可以由爱丽丝构建并签名,因为她可能希望独立于任何第三方来证明身份。

$TxID_{CRP-Set}$			
输入		输出	
输出点	脚本	值	脚本
$TxID_{prev}$	$< SIG_A > < P_A >$	x BSV	OP_DUP OP_HASH160 $< P_A^1 >$ OP_EQUALVERIFY OP_CHECKSIG
[0473]		0 BSV	OP_0 OP_RETURN $< Rep(C_1, R_1) >$
		x BSV	OP_DUP OP_HASH160 $< P_A^2 >$ OP_EQUALVERIFY OP_CHECKSIG
		0 BSV	OP_0 OP_RETURN $< Rep(C_2, R_2) >$
		...	
		x BSV	OP_DUP OP_HASH160 $< P_A^n >$ OP_EQUALVERIFY OP_CHECKSIG
		0 BSV	OP_0 OP_RETURN $< Rep(C_n, R_n) >$

[0474] 表3: 包含映射到CRP的可花费UTXO的事务。

[0475] 在上图中可以看到包含与n个CRP相关的2n个输出的示例性事务, 其中每个可花费输出可以映射到所述CRP中的一个CRP, CRP表示本身已包含在对应的不可花费输出 (例如, OP_FALSE OP_RETURN) 中。还应当注意的是, 将CRP组织为事务和UTXO的三种可能的变体在这里同样适用。

[0476] 5.1.3. 讨论

[0477] CRP管理的好处: 将CRP映射到UTXO的概念可以显著改善CRP管理以及对前几节中身份协议用户的处理。优点在于, 可以在一定程度上将CRP的存储和查找卸载到区块链网络106, 以及可以方便从中进行可靠检索的服务提供者。

[0478] 通过将特定PUF的所有‘实时’CRP映射到UTXO, 可以通过查询UTXO集的状态来获得关于身份系统中给定PUF当前可用的CRP的准确信息, 从而改进CRP更新过程。

[0479] 下面是使用所描述的区块链和UTXO-CRP映射约定的简单流程示例:

[0480] 1. 爱丽丝获得PUF设备, 并将CRP集列举为 $(C_1, R_1), (C_2, R_2), \dots, (C_n, R_n)$ 。

[0481] 2. 爱丽丝生成如表2所示的事务 $TxID_{CRP-Set}$, 并向区块链网络进行广播。

[0482] 3. 随着时间的推移, 爱丽丝会消耗多个CRP来向第三方验证她的身份。

[0483] 4. 爱丽丝现在希望检查她是否有足够的CRP来满足她在接下来一周的预期活动:

[0484] 1. 爱丽丝查询区块链节点104或类似SPV的服务提供者, 询问 $TxID_{CRP-Set}$ 的哪些UTXO当前未花费;

[0485] 2. 区块链节点或服务提供者以仍未花费的事务 $TxID_{CRP-Set}$ 的输出数量进行响应。

[0486] 5. 如果返回的数量不够, 爱丽丝可以与她的可信第三方一起执行身份更新流程, 或者只是列举更多CRP以获得独立建立的身份。否则, 爱丽丝不会采取任何动作。

[0487] 嵌入与配对: 选择是将CRP嵌入到可花费输出中, 还是简单地将其与输出配对, 使爱丽丝可以在区分这些情况的两种不同好处之间进行选择。

[0488] 如果CRP嵌入到可花费输出中, 这将激励维护区块链网络106的区块链节点104保持这些输出的数据随时可用。这意味着对爱丽丝的查询的响应可能更快, 更重要的是, 区块链节点更有可能将这些事务输出的原始数据传回爱丽丝。

[0489] 如前所述, 如果包括CRP的表示 $Rep(C, R)$, 使其包含质询、响应或两者的原始 (或混

淆)数据,则这意味着爱丽丝将能够从区块链网络106检索相关信息。这使得爱丽丝能够替换本地存储,并使用区块链150操作更轻量级的系统,因为将数据嵌入到可花费输出中会增加她的数据无论如何都具有高可用性的可能性。

[0490] 相比之下,如果CRP只与可花费输出配对,则爱丽丝可能只能确定有多少CRP可供其使用,但不一定能从比特币节点检索表示数据本身。这可能意味着,如果爱丽丝不在本地维护自己的CRP集,则她必须咨询区块链节点网络106外部的代理。

[0491] 使用双重哈希:在上面的示例性实现方式中,可以看到双重哈希 $H^2(\text{Data})$ 可以用作一些Data的链上表示。以这种方式使用双重哈希的原因是,允许单个哈希也在链上显示,原则上就像一方知道 $H(\text{Data})$ 的知识证明一样,而 $H(\text{Data})$ 又连接到Data。

[0492] 这在(例如)PUF身份的情况下可能有用,在这种情况下,爱丽丝在链上将 $H^2(R)$ 记录为可以由第三方提供 $H(R)$ 来满足的花费负担,前提是爱丽丝已经与他们共享了R的实际值。

[0493] 多方签名:本节中详细说明的事务可能包括多个不同方的更多签名,以帮助爱丽丝证明PUF身份,这也是合理的。例如,爱丽丝和第三方身份提供者可能都希望对CRP事务的一个或多个输入进行签名,以提高验证者对爱丽丝身份的信任。如果对应签名者是可以证明爱丽丝用于对区块链事务进行签名的一个或多个公钥的认证中心,那么这一点尤其重要。作为仅多个签名(即“multi-sig”)的替代方案,可以通过门限签名或密钥拆分技术(例如,Shamir秘密共享)等手段将多方纳入签名流程中。

[0494] 5.2.使用事务进行高效的身份管理

[0495] 区块链可以与基于PUF的身份系统(例如,前面介绍的那些系统)结合使用的另一种方式是作为撤销由PUF设备保护的身份密钥或通行证的有效手段。

[0496] 在之前的数字证书管理工作中,已知在链上发布和撤销证书,同时还有对应的证书验证过程。考虑这样一种场景,即爱丽丝在链上证明其基于PUF的身份时,很乐意与认证中心合作。爱丽丝在链上注册身份证书的流程如下:

[0497] 1. 认证中心(CA)验证爱丽丝的身份。

[0498] 2. CA创建证书事务。该事务具有以下输入和输出:

[0499] a. 输入:CA的UTX0,带有包含CA签名和公钥的解锁脚本。

[0500] b. 输出1:P2PKH锁定脚本。

[0501] c. 输出2:包含爱丽丝的公钥的OP_RETURN输出。

[0502] 3. 事务被广播,一旦被挖掘,CA就会向爱丽丝提供事务ID $TXID_{CTX-PK_A}$ 。

[0503] 该过程最终导致爱丽丝和认证中心合作生成由CA进行签名的事务,其中包含一个不可花费输出,包括爱丽丝公钥证书,以及与证书配对的可花费输出,CA可以使用该输出来撤销证书。

[0504] 本文公开的实施例混合使用上面针对数字证书概述的方法以及建立基于PUF的身份的方法,例如前面描述的方法中的一种方法。此处添加到PUF身份系统中的元素是为了让通用可信第三方(类似于CA)能够通过花费UTX0来‘撤销’CRP或相关公钥。

[0505] 可信第三方撤销爱丽丝公钥证书的情况与前面讨论的加密身份建立有关。

[0506] 在使用链上存储或证明的CR对(CRP)的情况下,本文公开的实施例提供了一种方案,该方案允许可信第三方在身份验证过程中使用CRP后撤销这些CRP。示例性方法如下:

- [0507] 1. 爱丽丝和可信第三方执行身份建立协议(例如,如前所述)。
- [0508] 2. 爱丽丝和可信第三方现在希望使用区块链来管理在1中生成的CRP,或在步骤1之后可以获得的CRP:
- [0509] a. 爱丽丝创建CRP映射事务 $TxID_{CRP-Set}$,该事务将CRP映射到事务输出。如下面的表4所示。
- [0510] b. 爱丽丝和可信第三方都对 $TxID_{CRP-Set}$ 进行签名。
- [0511] 3. CRP映射事务 $TxID_{CRP-Set}$ 在区块链区块中进行广播和发布。

$TxID_{CRP-Set}$			
输入		输出	
输出点	脚本	值	脚本
[0512] $TxID_{prev}$	$< SIG_A > < P_A >$ $< SIG_{TTP} > < P_{TTP} >$	x BSV	$[Checksig P_{TTP}] OP_RETURN <Rep(C_1, R_1)>$
		x BSV	$[Checksig P_{TTP}] OP_RETURN <Rep(C_2, R_2)>$
		x BSV	$[Checksig P_{TTP}] OP_RETURN <Rep(C_3, R_3)>$
		...	
		x BSV	$[Checksig P_{TTP}] OP_RETURN <Rep(C_n, R_n)>$

- [0513] 表4: 允许可信第三方撤销/花费CRP的CRP映射事务。
- [0514] 在该过程中创建的映射事务如上面的表4所示。这与前面表2中所示的CRP映射事务非常相似,不同之处在于可信第三方和爱丽丝都对输入进行签名,并且映射到CRP的UTXO中的每个UTXO都可以由可信第三方通过在未来的事务中花费来撤销。
- [0515] 这是有利的,因为它允许在没有直接通信的情况下处理CRP的撤销,并且TTP可以代表用户执行撤销,这进一步减轻了系统中爱丽丝的负担,并使她的身份管理能够变得更加轻量级。
- [0516] 6. 结论
- [0517] 一旦给出本文的公开内容,所公开技术的其它变体或用例对于本领域技术人员可能变得显而易见。本公开的范围不受所描述的实施例限制,而仅受随附权利要求限制。
- [0518] 例如,上面的一些实施例已经根据比特币网络106、比特币区块链150和比特币节点104进行了描述。然而,应当理解的是,比特币区块链是区块链150的一个特定示例,并且上述描述通常可以应用于任何区块链。也就是说,本发明决不限于比特币区块链。更一般地,以上对比特币网络106、比特币区块链150和比特币节点104的任何引用可以分别参考区块链网络106、区块链150和区块链节点104来替换。区块链、区块链网络和/或区块链节点可以共享如上所述的比特币区块链150、比特币网络106和比特币节点104的部分或全部所述特性。
- [0519] 在本发明的优选实施例中,区块链网络106是比特币网络,并且比特币节点104至少执行对区块链150的区块151进行创建、发布、传播和存储中的所有所述功能。不排除可能存在仅执行这些功能中的一个或部分功能但不是全部功能的其它网络实体(或网络元件)。也就是说,网络实体可以执行传播和/或存储区块的功能,而不创建和发布区块(请记住,这些实体不被认为是优选的比特币网络106的节点)。
- [0520] 在本发明的其他实施例中,区块链网络106可以不是比特币网络。在这些实施例

中,不排除节点可以执行对区块链150的区块151进行创建、发布、传播和存储中的至少一个或部分功能但不是所有功能。例如,在这些其它区块链网络上,“节点”可用于指被配置为创建和发布区块151但不存储和/或传播这些区块151到其它节点的网络实体。

[0521] 甚至更通俗地说,上面对术语“比特币节点”104的任何引用可以用术语“网络实体”或“网络元件”代替,其中这样的实体/元件被配置为执行对区块进行创建、发布、传播和存储中的一些或全部角色。这种网络实体/元件的功能可以在硬件中实现,方法与上面参照区块链节点104所述的方式相同。

[0522] 应当理解的是,上述实施例仅通过示例的方式进行描述。更通俗地说,可根据下述任何一个或更多个语句提供一种方法、装置或程序。

[0523] 语句1.一种计算机实现的方法,用于使一个或多个验证方能够验证包括目标方或所述目标方的设备的目标的身份;所述方法包括,在建立阶段,由除所述一个或多个验证方中的任一验证方之外的建立方:将一个或多个质询的集合输入到PUF模块中,所述PUF模块包括物理不可克隆函数PUF,用以基于所述PUF生成一个或多个响应的相应的集;以及,将所述响应的所述集中的每个响应的相应响应数据记录存储在所述目标方或所述一个或多个验证方的任何设备外部的数据存储器中,所述数据存储器是第三方计算机设备的一部分、或是公共对等发布介质(public peer-to-peer publication medium),从而向所述一个或多个验证方提供所述响应数据记录,以在后续验证阶段验证所述目标的所述身份,其中每个响应的所述响应数据记录包括相应的响应或从中导出的数据。

[0524] 语句2.根据语句1所述的方法,其中所述建立方是所述目标方,所述方法由所述目标方使用所述目标方的计算机设备执行。

[0525] 语句3.根据语句2所述的方法,其中所述PUF模块集成在所述目标方的所述计算机设备中。

[0526] 语句4.根据语句2所述的方法,其中所述PUF模块是所述目标方的所述计算机设备的外部的外围设备(peripheral)。

[0527] 语句5.根据语句2至4中任一项所述的方法,其中所述数据存储器是所述第三方计算机设备的一部分,所述目标方从所述第三方接收所述质询集,输入到所述PUF模块的所述质询集是从所述第三方接收的所述质询集。

[0528] 语句6.根据语句1所述的方法,其中所述建立方是可信第三方。

[0529] 语句7.根据语句6所述的方法,其中所述数据存储器是所述第三方计算机设备的一部分,所述方法由所述可信第三方使用所述第三方计算机设备执行。

[0530] 语句8.根据语句7所述的方法,其中所述第三方计算机设备包括服务器设备,所述服务器设备包括位于一个或多个地理位置的一个或多个服务器单元。

[0531] 语句9.根据语句7或8所述的方法,其中所述PUF模块是所述第三方计算机设备的外部外围设备或可移动组件,所述方法包括:在所述建立阶段之后以物理方式将所述PUF模块发送给所述目标方,以使所述目标方能够在所述验证阶段响应来自所述验证方的质询。

[0532] 语句10.根据语句1至6中任一项所述的方法,其中所述数据存储器是对等发布介质。

[0533] 语句11.根据语句10所述的方法,其中所述对等发布介质是区块链。

[0534] 语句12.根据语句11所述的方法,其中所述区块链采用基于输出的事务模型,所述

区块链包括多个事务,每个事务具有至少一个输入和至少一个输出,其中每个事务的所述输入指向另一个事务的所述输出,其中所述响应数据记录存储在所述区块链上记录的一个或多个事务的一个或多个输出中。

[0535] 语句13.根据语句12所述的方法,其中所述方法还包括:通过在区块链上记录新事务来更新或撤销所述响应数据记录中的过期响应数据记录,所述新事务具有指向存储所述过期响应数据记录的所述事务的所述输出的输入。

[0536] 语句14.根据前述任一项语句所述的方法,其中所述存储步骤包括:通过网络访问所述数据存储器,并通过所述网络发送数据,以在所述数据存储器中创建所述响应数据记录。

[0537] 语句15.根据前述任一项语句所述的方法,其中所述一个或多个验证方是多个验证方。

[0538] 语句16.根据语句15所述的方法,其中所述存储仅向所述验证方中的每个验证方提供所述响应数据记录中的一个或多个响应数据记录的相应子集,不同子集向所述验证方中的不同验证方提供记录。

[0539] 语句17.根据语句16所述的方法,其中所述子集相互排斥(exclusive)。

[0540] 语句18.根据前述任一项语句所述的方法,其中所述响应数据记录中的每个响应数据记录包括相应响应的显式值(explicit value)。

[0541] 语句19.根据语句18所述的方法,其中每个响应数据记录的值仅以加密形式存储在所述数据存储器中。

[0542] 语句20.根据语句19所述的方法,其中每个响应数据记录需要不同的相应解密密钥来解密。

[0543] 语句21.根据从属于语句16的语句20所述的方法,其中,通过将所述解密密钥中的不同解密密钥分发给不同的验证方,向所述验证方中的不同验证方提供所述不同子集。

[0544] 语句22.根据语句1至17中任一项所述的方法,其中,所述响应数据记录中的每个响应数据记录仅包括相应响应的证明(attestation),而不包括所述响应的显式值,所述证明包括相应响应的变换,所述变换不公开所述响应,但所述变换使验证方能够通过由所述目标返回的相应响应的另一实例执行相同的变换并与所述证明进行比较来验证所述目标的所述身份。

[0545] 语句23.根据语句22所述的方法,其中所述变换包括哈希或双重哈希。

[0546] 语句24.根据前述任一项语句所述的方法,所述方法包括:从所述验证方中的一个验证方接收请求,所述请求包括所述质询集中的一个质询;以及,在响应中,将所述质询的所述另一实例输入到所述PUF模块中,以生成对应响应的另一实例,并将所述响应的所述另一实例返回给所述验证方,以使所述验证方能够基于与所述数据存储器中存储的所述响应的比较来验证所述目标的所述身份。

[0547] 语句25.根据语句1至17中任一项所述的方法,其中所述响应数据记录中的每个响应数据记录包括从所述相应响应推导出的相应公钥-私钥对的相应公钥。

[0548] 语句26.根据语句25所述的方法,其中所述推导包括使用所述相应响应作为生成所述相应公钥-私钥对的算法中的种子。

[0549] 语句27.根据语句25或26所述的方法,所述方法包括:向所述验证方中的一个验证

方,发送、或提供由所述目标使用所述公钥中的一个公钥进行签名的消息。

[0550] 语句28.根据前述任一项语句所述的方法,其中所述PUF模块包括PUF、接口逻辑、和确定性变换函数;其中所述接口逻辑通过以下方式引起所述响应集的生成:接收所述质询集;将基本质询输入到所述PUF中,以生成对应的主要响应;以及,将所述接收的质询集中的每个质询与所生成的基本响应一起输入到所述变换函数中,以生成所述响应集中的相应响应,所述变换函数是来自所接收的质询集的相应质询和所生成的基本响应的函数。

[0551] 语句29.根据前述任一项语句所述的方法,其中所述响应数据记录中的每个响应数据记录与所述相应质询的指示相关联地存储在所述数据存储器中,从而使所述验证方能够确定所述相应质询、或能够基于所述相应质询查找所述相应响应,所述相应质询将用于在验证中对所述目标提出质询。

[0552] 语句30.根据语句29所述的方法,其中每个相应质询的所述指示包括与所述相应响应数据相关联地存储的所述相应质询的显式值。

[0553] 语句31.根据语句29所述的方法,其中所述质询的所述指示包括主质询,从所述主质询可以导出所述质询集。

[0554] 语句32.根据前述任一项语句所述的方法,其中所述目标方是多个目标方中的一个目标方,对于所述多个目标方,所述数据存储器记录了对所述多个目标方的对应PUF的质询所产生的响应数据集。

[0555] 语句33.一种计算机设备,所述计算机设备包括:存储器,所述存储器包括一个或多个存储器单元;以及,处理装置,所述处理装置包括一个或多个处理单元,其中所述存储器存储被设置在所述处理装置上运行的代码,所述代码被配置为当在所述处理装置上运行时,执行根据前述任一项语句所述的方法。

[0556] 语句34.一种计算机程序,所述计算机程序包含在非暂时性计算机可读介质上并且被配置为当在一个或多个处理器上运行时,执行根据语句1至32中任一项所述的方法。

[0557] 语句35.一种计算机实现的方法,用于使一个或多个验证方能够验证目标的身份,所述目标包括目标方或所述目标方的设备;所述方法包括,在建立阶段,由可信第三方:从所述目标方接收响应数据,所述响应数据是由所述目标方将相应的质询输入到包括物理不可克隆函数PUF的PUF模块中以生成相应的响应而产生的;将所述一个或多个响应中的每个响应的相应响应数据记录存储在所述可信第三方的数据存储器中,每个响应数据记录包括所述相应响应或从中导出的数据;以及,向所述一个或多个验证方提供所述响应数据记录,以在后续验证阶段验证所述目标的所述身份,其中每个响应的所述响应数据记录包括所述相应响应或从中导出的数据。

[0558] 语句36.一种计算机设备,所述计算机设备包括:存储器,所述存储器包括一个或多个存储器单元;以及,处理装置,所述处理装置包括一个或多个处理单元,其中所述存储器存储被设置在所述处理装置上运行的代码,所述代码被配置为当在所述处理装置上运行时,执行根据语句35所述的方法。

[0559] 语句37.一种计算机程序,所述计算机程序包含在非暂时性计算机可读介质上并且被配置为当在一个或多个处理器上运行时,执行根据语句35所述的方法。

[0560] 语句38.一种计算机实现的方法,其中验证方验证目标的身份,所述目标包括目标方或所述目标方的设备;所述方法包括,由所述验证方:访问所述目标方或所述验证方的任

何设备外部的数据存储器中的响应数据记录,所述数据存储器是可信第三方的第三方计算机设备的一部分、或是公共对等发布介质,其中所述响应数据记录包括:a)响应值,所述响应值是通过将质询输入到包括与所述目标相关联的物理不可克隆函数PUF的PUF模块中而得到的,或b)证明,所述证明包括所述响应的变换;向所述目标发送包括所述质询的请求,并在响应中接收回所述响应的另一实例;执行比较,并在所述比较结果匹配的条件下验证所述目标的所述身份,其中所述比较包括:在a)所存储的所述响应的值与所述另一实例匹配的情况下,或在b)所述证明与应用于所述另一实例的相同变换匹配的情况下。

[0561] 语句39.一种计算机设备,所述计算机设备包括:存储器,所述存储器包括一个或多个存储器单元;以及,处理装置,所述处理装置包括一个或多个处理单元,其中所述存储器存储被设置在所述处理装置上运行的代码,所述代码被配置为当在所述处理装置上运行时,执行根据语句38所述的方法。

[0562] 语句40.一种计算机程序,所述计算机程序包含在非暂时性计算机可读介质上并且被配置为当在一个或多个处理器上运行时,执行根据语句38所述的方法。

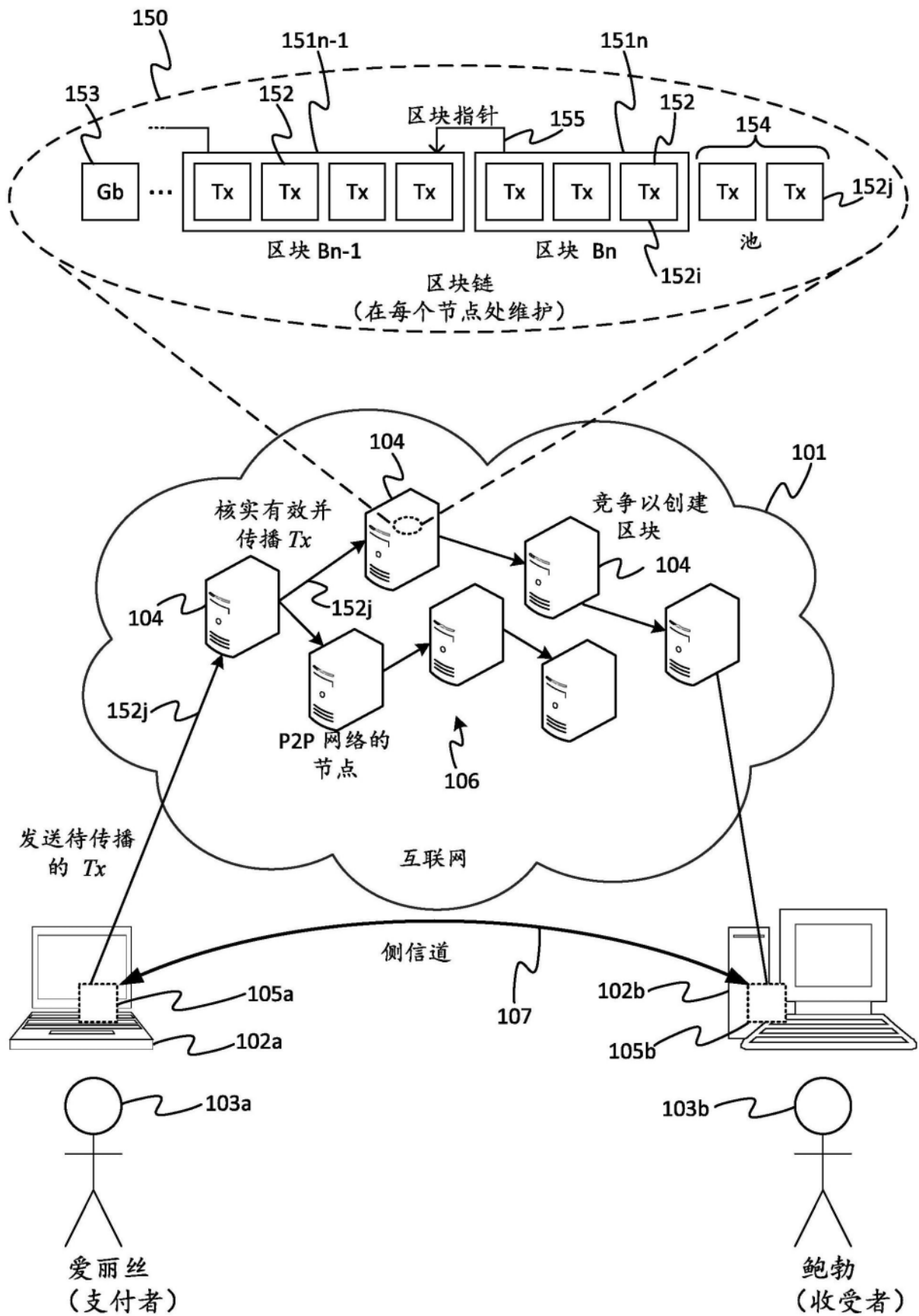
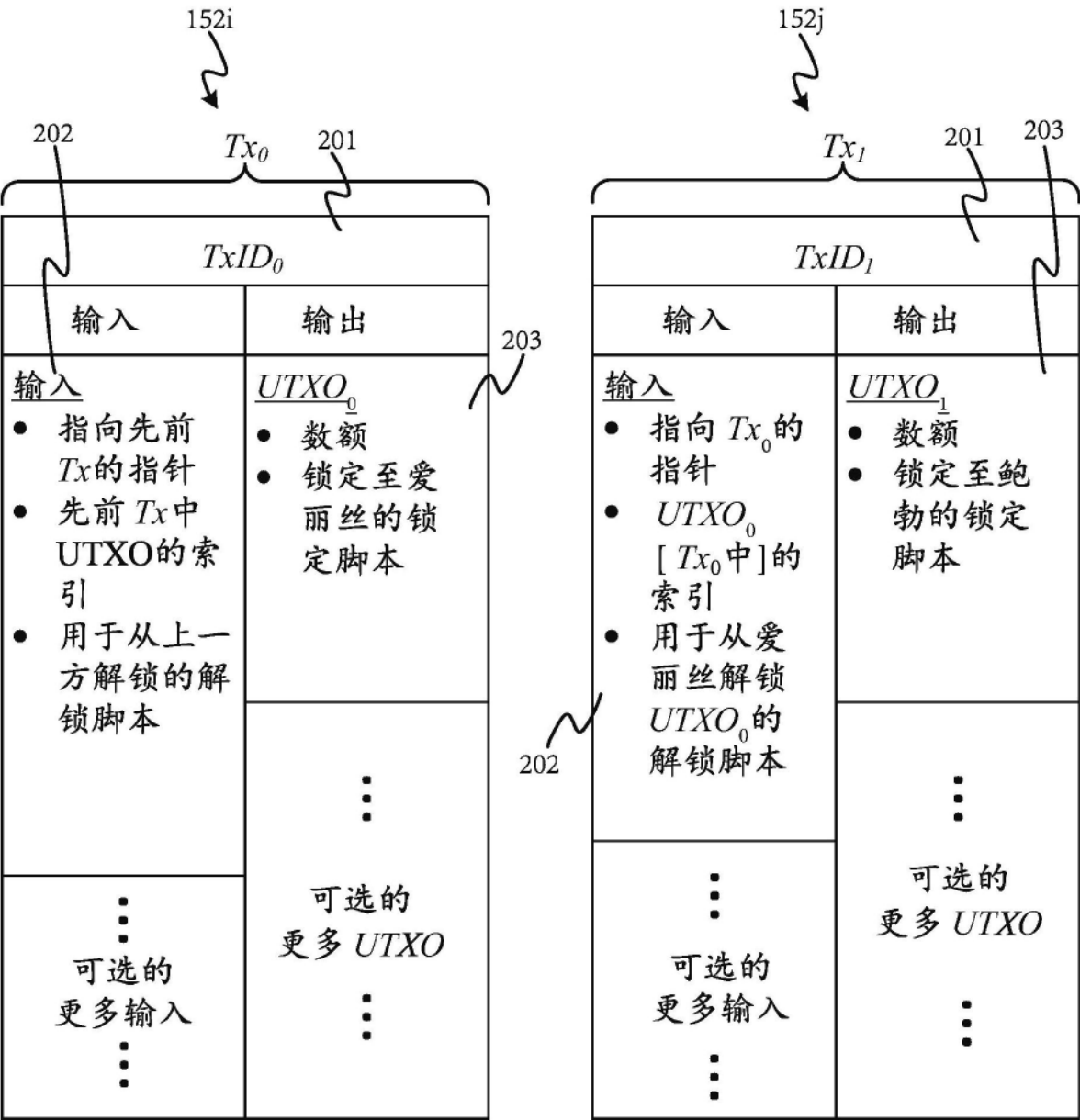


图1



从爱丽丝到鲍勃
的事务



通过运行：来自 Tx_0 的输出 203 的锁定脚本、和来自 Tx_1 的输入 202 的爱丽丝的解锁脚本来验证。此操作检查 Tx_1 中的爱丽丝的解锁脚本是否满足先前事务 Tx_0 中的锁定脚本中定义的条件。

图2

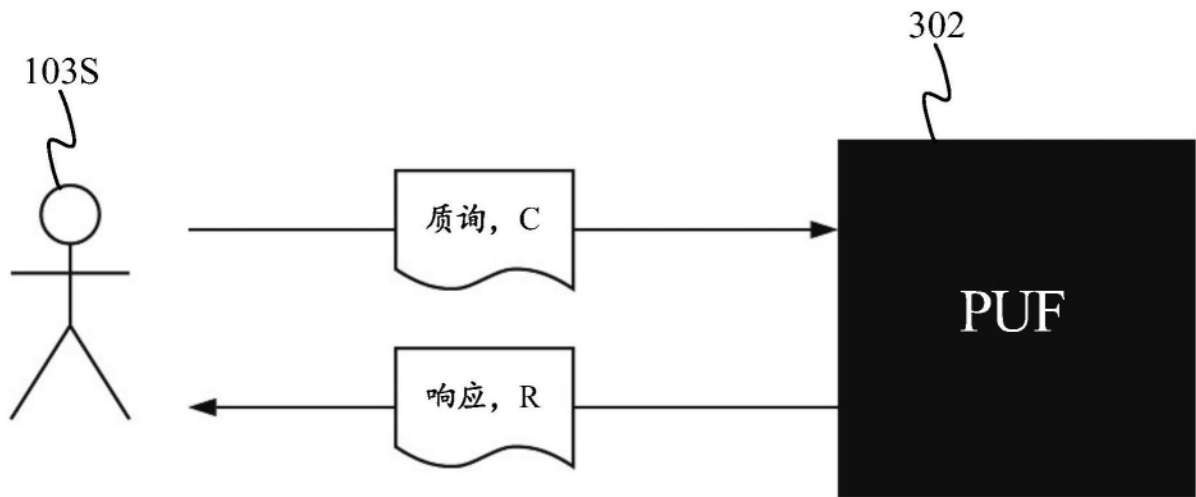


图3

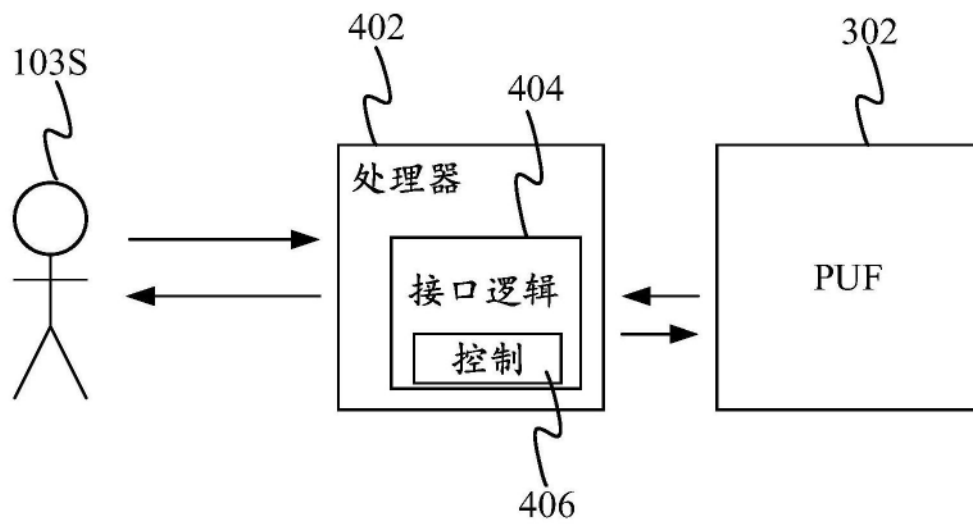


图4

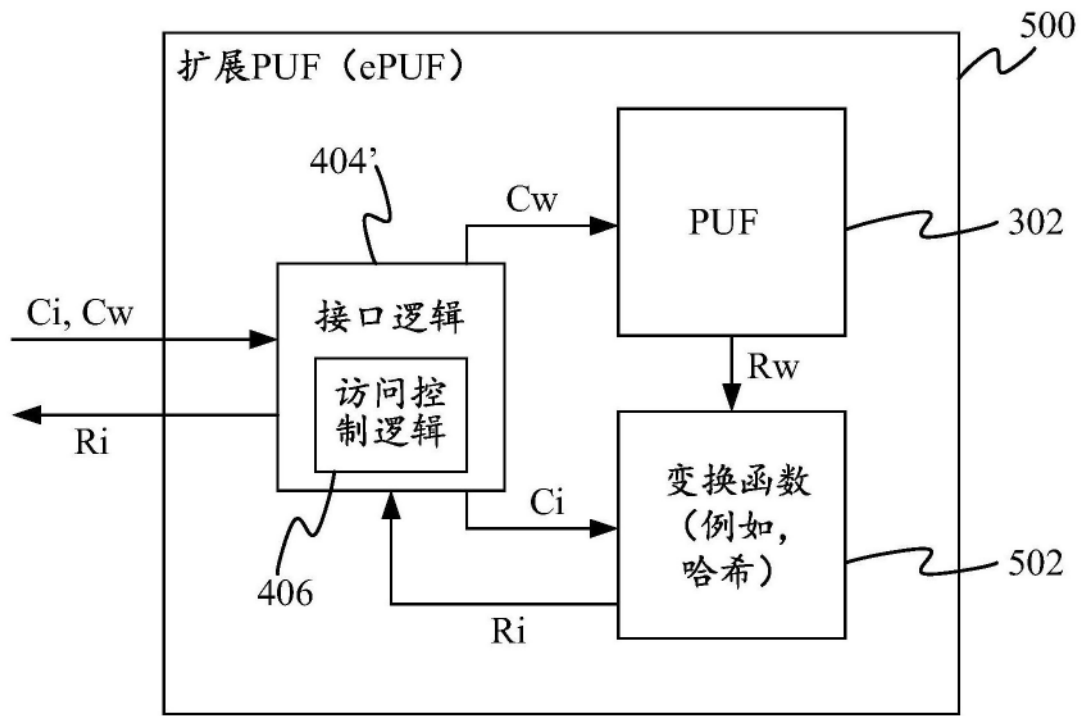


图5A

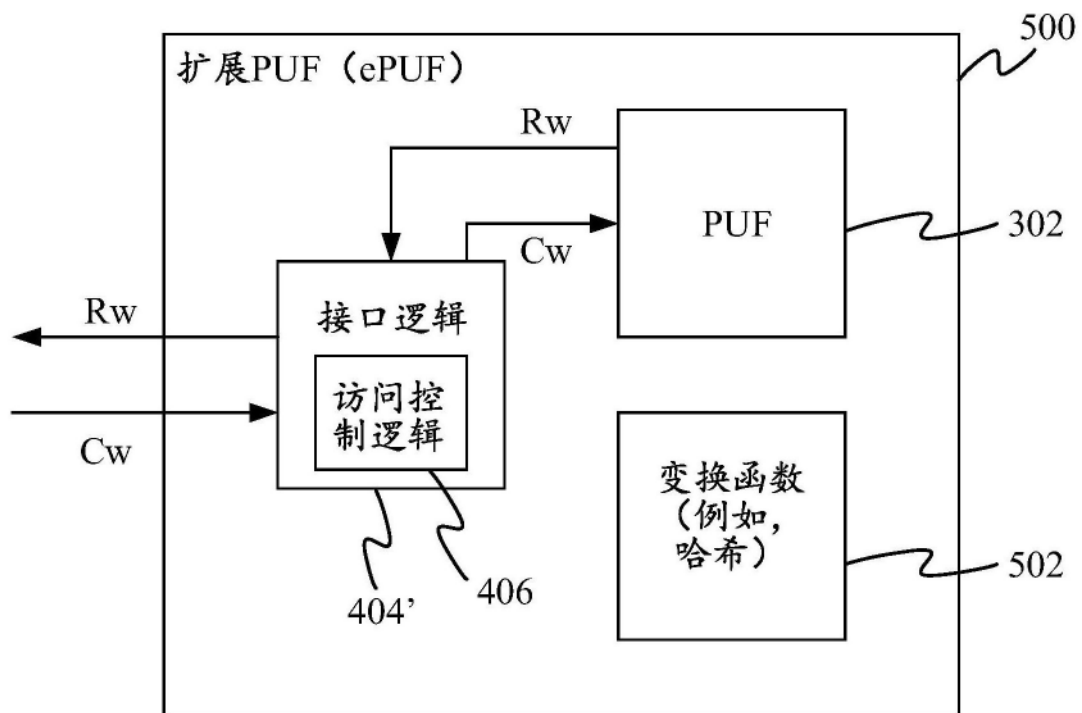


图5B

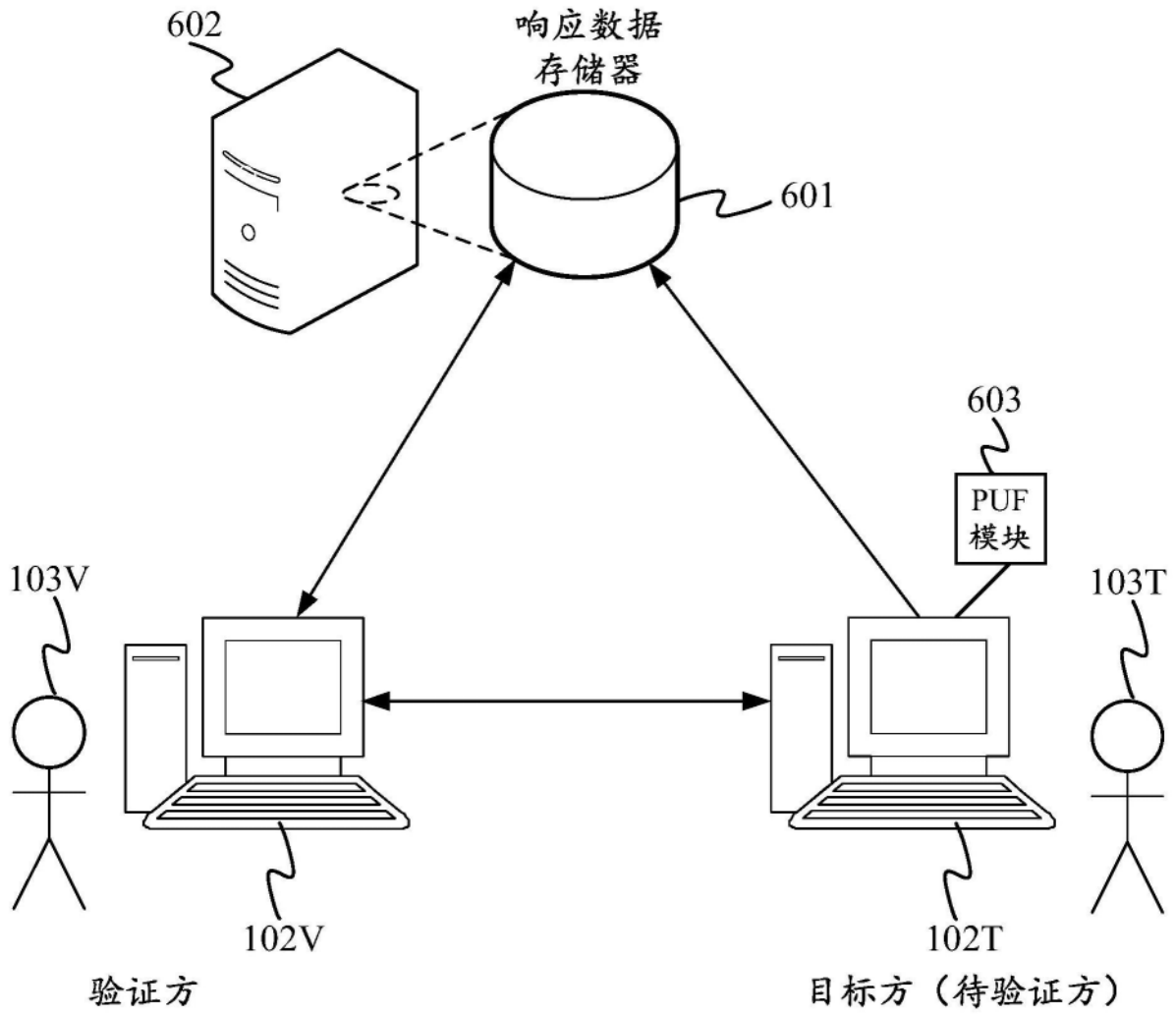


图6

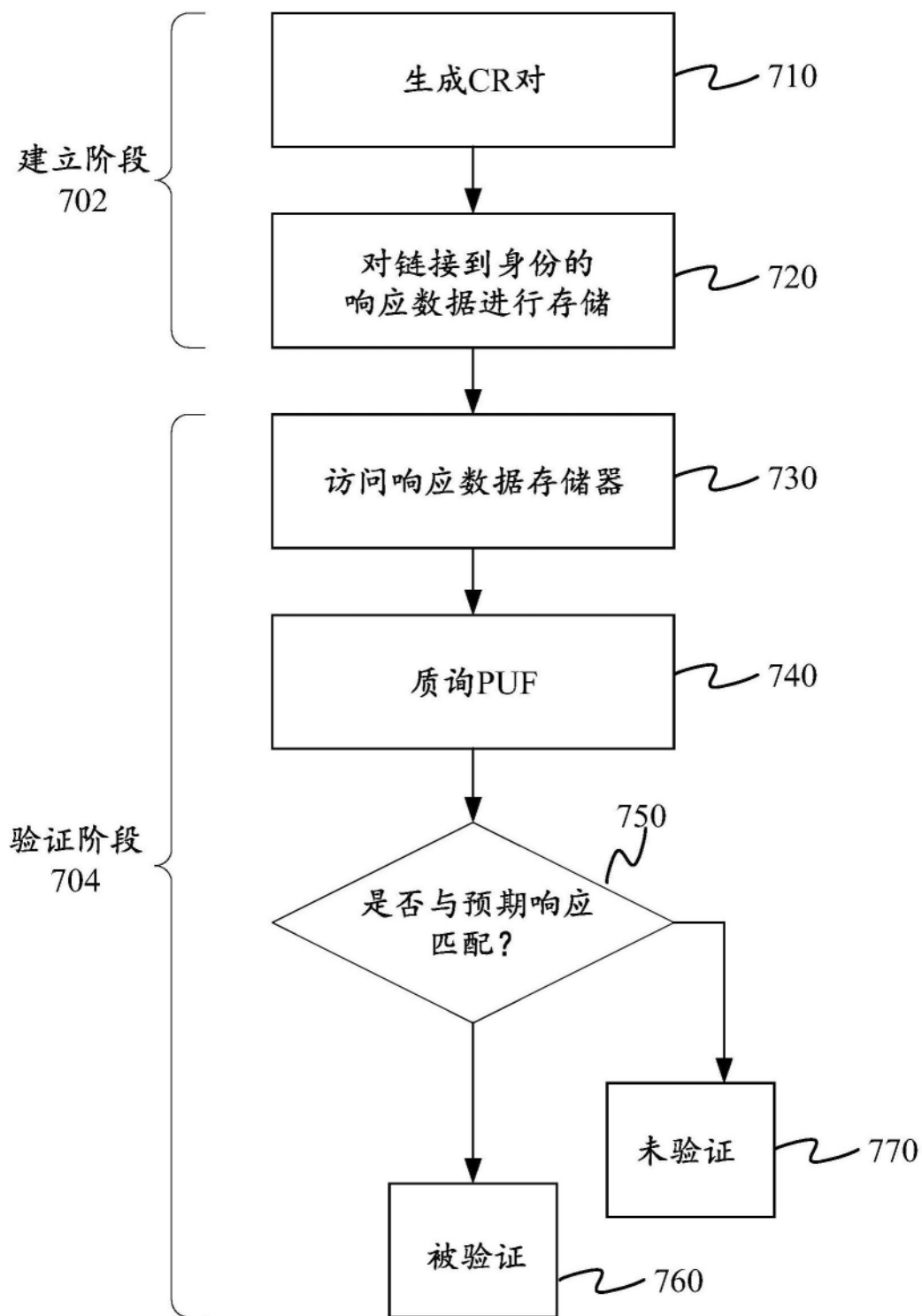


图7

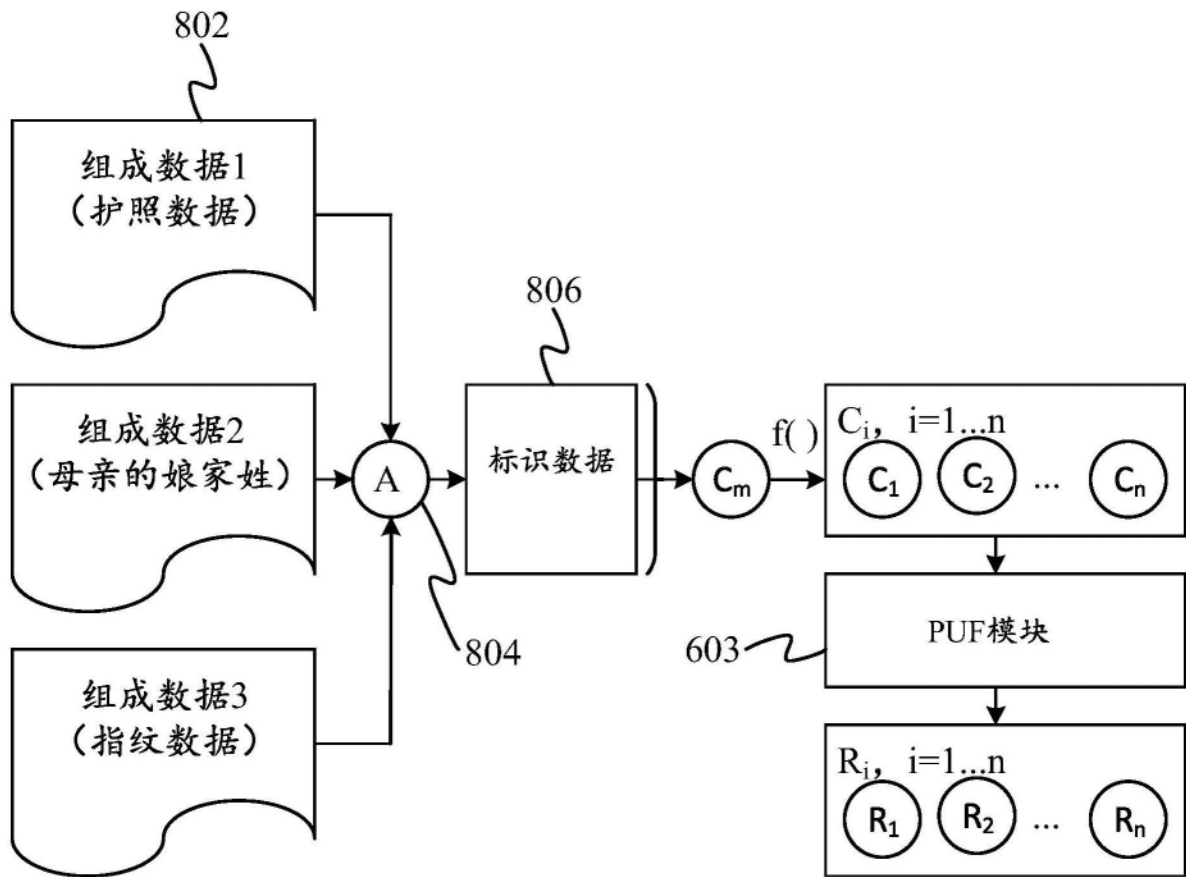


图8A

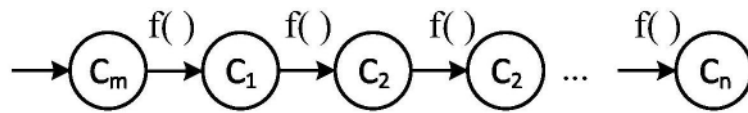


图8B

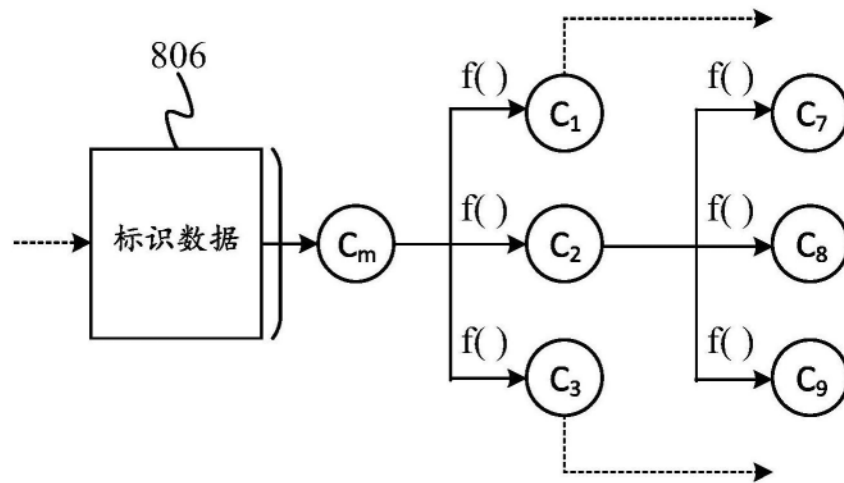


图8C

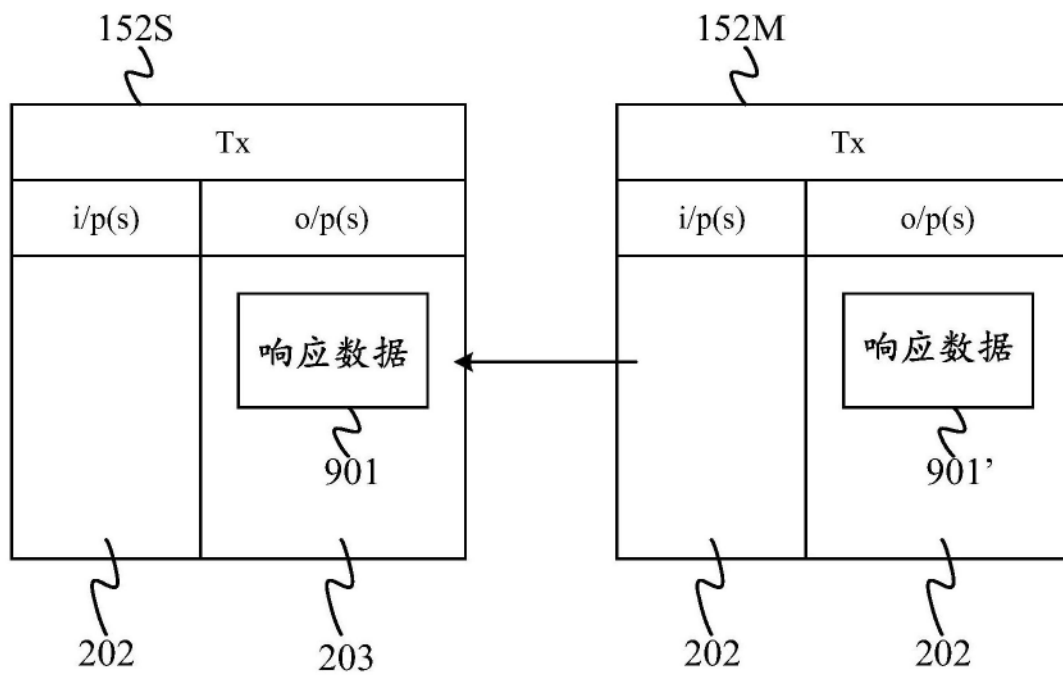


图9