(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 115982785 B
(45)授权公告日 2023.06.30

(21)申请号 202310258366.8
(22)申请日 2023.03.17
(65)同一申请的已公布的文献号
申请公布号 CN 115982785 A
(43)申请公布日 2023.04.18
(73)专利权人 北京富算科技有限公司
地址 100020 北京市朝阳区东三环中路9号19层2201
(72)发明人 尤志强 卞阳
(74)专利代理机构 北京超凡宏宇专利代理事务所(特殊普通合伙) 11463
专利代理师 唐正瑜
(51)Int.Cl.
G06F 21/71(2013.01)

(56)对比文件
CN 110557245 A,2019.12.10
CN 114327371 A,2022.04.12
CN 114584294 A,2022.06.03
CN 114844635 A,2022.08.02
CN 115080615 A,2022.09.20
US 9450938 B1,2016.09.20

审查员 徐晓

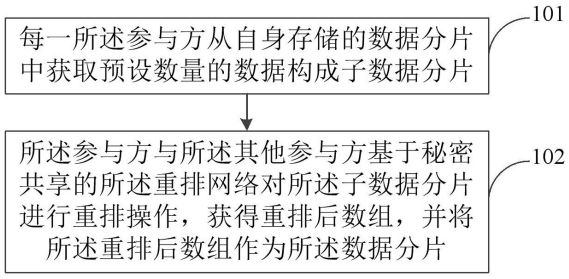
权利要求书2页 说明书14页 附图9页

(54)发明名称

多方安全的数据重排方法、装置、电子设备及存储介质

(57)摘要

本申请提供一种多方安全的数据重排方法、装置、电子设备及存储介质。涉及多方安全计算领域，方法包括：参与方按照如下步骤进行多轮迭代，直至完成对数据分片中所有的数据被重排，获得对原始数据进行重排后的重排数据；步骤包括：每一参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片；参与方与其他参与方基于秘密共享的重排网络对子数据分片进行重排操作，获得重排后数组，并将重排后数组作为数据分片。本申请针对每次迭代重排，各参与方均从数据分片中选取预设数量的数据，通过秘密共享的重排网络进行数据的重排，由于预设数量可以为任意长度，从而解决了现有技术中只能对固定长度的数据进行重排的问题。



CN 115982785 B

1.一种多方安全的数据重排方法,其特征在于,应用于多方安全计算系统中的参与方,所述多方安全计算系统中包括多个参与方,每一所述参与方中存储有原始数据对应的一数据分片;每一所述参与方随机生成一个重排网络,并与其他参与方秘密共享所述重排网络;所述方法包括:

所述参与方按照如下步骤进行多轮迭代,直至完成对所述数据分片中所有的数据被重排,获得对所述原始数据进行重排后的重排数据;所述步骤包括:

每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片;

所述参与方与其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作,获得重排后数组,并将所述重排后数组作为所述数据分片。

2.根据权利要求1所述的方法,其特征在于,所述每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片,包括:

各参与方同步生成随机种子,并根据所述随机种子从所述数据分片中获取预设数量的数据,构成所述子数据分片。

3.根据权利要求2所述的方法,其特征在于,所述根据所述随机种子从所述数据分片中获取预设数量的数据,包括:

各参与方基于所述随机种子,以概率 $p_0=\frac{n-2^m}{n}$ 从上一次已经采样过的数据中进行不放回采样,以概率 $p_1=\frac{2^m}{n}$ 从上一次未采样过的数据中进行不放回采样,获取所述预设数量的数据;或,

各参与方基于所述随机种子,根据概率 $\frac{2^m}{n}$ 以有放回的方式从所述数据分片中获取 j 次所述预设数量的数据;

其中,$(1-\Pi_i^j(1-\frac{2^m}{n}))>p_2$,$n$ 为所述原始数据的长度;2^m 为所述预设数量;p_2 为所述原始数据中参与重排的各个数据的概率。

4.根据权利要求1所述的方法,其特征在于,所述重排网络包括多个交换层,每一所述交换层包括多个交换门,每一所述交换门对应一个交换系数,所述交换系数用于表征是否对输入的数据进行交换;所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作,包括:

根据公式 $\begin{cases}A_{out}=A_{in}*(1-k)+B_{in}*k\\B_{out}=B_{in}*(1-k)+A_{in}*k\end{cases}$ 对输入到所述交换门的数据进行重排操作;其中,A_{out} 为所述交换门输出所述参与方的输出数据;A_{in} 为所述参与方输入到所述交换门的输入数据;k 为所述交换系数;B_{out} 为所述交换门输出所述其他参与方的输出数据;B_{in} 为其他参与方输入到所述交换门的输入数据。

5.根据权利要求1所述的方法,其特征在于,所述重排网络包括完备网络、双调合并网络或随机网络。

6.根据权利要求1所述的方法，其特征在于，所述原始数据的长度为n，根据公式$\lceil m \rceil = \log_2 n$确定所述预设数量。

7.根据权利要求1-6任一项所述的方法，其特征在于，各参与方之间通过比特进行秘密共享所述重排网络。

8.一种多方安全的数据重排装置，其特征在于，应用于多方安全计算系统中的参与方，所述多方安全计算系统中包括多个参与方，每一所述参与方中存储有原始数据对应的一数据分片；每一所述参与方随机生成一个重排网络，并与其他参与方秘密共享所述重排网络；所述装置包括：

所述参与方按照如下步骤进行多轮迭代，直至完成对所述数据分片中所有的数据被重排，获得对所述原始数据进行重排后的重排数据；所述步骤包括：

每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片；

所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，获得重排后数组，并将所述重排后数组作为所述数据分片。

9.一种电子设备，其特征在于，包括：处理器、存储器和总线，其中，

所述处理器和所述存储器通过所述总线完成相互间的通信；

所述存储器存储有可被所述处理器执行的程序指令，所述处理器调用所述程序指令能够执行如权利要求1-7任一项所述的方法。

10.一种非暂态计算机可读存储介质，其特征在于，所述非暂态计算机可读存储介质存储计算机指令，所述计算机指令被计算机运行时，使所述计算机执行如权利要求1-7任一项所述的方法。

多方安全的数据重排方法、装置、电子设备及存储介质

技术领域

[0001] 本申请涉及多方安全计算领域，具体而言，涉及一种多方安全的数据重排方法、装置、电子设备及存储介质。

背景技术

[0002] 安全多方计算(Multi-Party Computation, 简称MPC)主要解决多个参与方在不互相透露各自输入的前提下，如何联合完成计算的问题。

[0003] MPC目前的应用前景越来越广泛，企业、政府、学术机构和个人间的协同业务需求日趋强烈。一个典型的场景是，人工智能迅猛发展的历程中，数据隐私的需求愈加强烈。AI训练所需的数据，在很多商业场景里由于隐私合规性的原因无法获得，导致无法完成训练或者训练效果很差。隐私AI (Privacy AI)正试图利用MPC来解决AI计算中的隐私保护问题，即如何在AI训练涉及的数据方不直接暴露明文数据的前提下，完成协同训练和协同预测。

[0004] 安全两方计算是国内应用比较广泛的多方安全计算模式。目前的安全两方计算缺少高效的对数据进行shuffle的方法。基于shuffle network对数据进行shuffle，这种重排网络只能对固定长度的数据进行重排。

发明内容

[0005] 本申请实施例的目的在于提供一种多方安全的数据重排方法、装置、电子设备及存储介质，用以实现对任意长度的数据进行重排。

[0006] 第一方面，本申请实施例提供一种多方安全的数据重排方法，应用于多方安全计算系统中的参与方，所述多方安全计算系统中包括多个参与方，每一所述参与方中存储有原始数据对应的一数据分片；每一所述参与方随机生成一个重排网络，并与其他参与方秘密共享所述重排网络；所述方法包括：

[0007] 所述参与方按照如下步骤进行多轮迭代，直至完成对所述数据分片中所有的数据被重排，获得对所述原始数据进行重排后的重排数据；所述步骤包括：

[0008] 每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片；

[0009] 所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，获得重排后数组，并将所述重排后数组作为所述数据分片。

[0010] 本申请实施例针对每次迭代重排，各参与方均从数据分片中选取预设数量的数据，通过秘密共享的重排网络进行数据的重排，由于预设数量可以为任意长度，从而解决了现有技术中只能对固定长度的数据进行重排的问题。

[0011] 在任一实施例中，所述每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片，包括：

[0012] 各参与方同步生成随机种子，并根据所述随机种子从所述数据分片中获取预设数量的数据，构成所述子数据分片。

[0013] 本申请实施例基于随机种子从数据分片中选取待重排的数据，使得各个参与方能够获取到各自数据分片中对应位置的数据，以确保各参与方采样的一致性。

[0014] 在任一实施例中，所述根据所述随机种子从所述数据分片中获取预设数量的数据，包括：

[0015] 各参与方基于所述随机种子，以概率 $p_0=\frac{n-2^m}{n}$ 从上一次已经采样过的数据中进行不放回采样，以概率 $p_1=\frac{2^m}{n}$ 从上一次未采样过的数据中进行不放回采样，获取所述预设数量的数据；或，

[0016] 各参与方基于所述随机种子，根据概率 $\frac{2^m}{n}$ 以有放回的方式从所述数据分片中获取 j 次所述预设数量的数据；

[0017] 其中，$j=(1-\Pi_i^j(1-\frac{2^m}{n}))>p_2$，$n$ 为所述原始数据的长度；2^m 为所述预设数量；p_2 为所述原始数据中参与重排的各个数据的概率。

[0018] 本申请实施例通过无放回或有放回的方式从各个数据分片中获取待重排的数据，保证数据分片中的每个数据均能够被作为待重排的数据。

[0019] 在任一实施例中，所述重排网络包括多个交换层，每一所述交换层包括多个交换门，每一所述交换门对应一个交换系数，所述交换系数用于表征是否对输入的数据进行交换；所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，包括：

[0020] 根据公式 $\begin{cases}A_{out}=A_{in}*(1-k)+B_{in}*k\\B_{out}=B_{in}*(1-k)+A_{in}*k\end{cases}$ 对输入到所述交换门的数据进行重排操作；其中，A_{out} 为所述交换门输出所述参与方的输出数据；A_{in} 为所述参与方输入到所述交换门的输入数据；k 为所述交换系数；B_{out} 为所述交换门输出所述其他参与方的输出数据；B_{in} 为其他参与方输入到所述交换门的输入数据。

[0021] 本申请实施例通过每次从数据分片中选取重排网络所需数据量的数据进行重排，满足了重排网络每次重排的数据的数量要求。

[0022] 在任一实施例中，所述重排网络包括完备网络、双调合并网络或随机网络。

[0023] 本申请实施例中的重排网络可以为完备网络、双调合并网络或随机网络，将上述网络用于数据的重排，使得数据重排可以有多种网络选择。

[0024] 在任一实施例中，所述原始数据的长度为n，根据公式 $\lceil m\rceil=\log_2 n$ 确定所述预设数量。

[0025] 本申请实施例通过上述公式确定的预设数量能够满足重排网络对数据数量的要求。

[0026] 在任一实施例中，各参与方之间通过比特进行秘密共享所述重排网络。

[0027] 本申请实施例中各参与方之间通过比特的方式进行秘密共享重排网络，进一步减

少隐私计算时的通信量级，提高计算性能。

[0028] 第二方面，本申请实施例提供一种多方安全的数据重排装置，应用于多方安全计算系统中的参与方，所述多方安全计算系统中包括多个参与方，每一所述参与方中存储有原始数据对应的一数据分片；每一所述参与方随机生成一个重排网络，并与其他参与方秘密共享所述重排网络；所述装置包括：

[0029] 所述参与方按照如下步骤进行多轮迭代，直至完成对所述数据分片中所有的数据被重排，获得对所述原始数据进行重排后的重排数据；所述步骤包括：

[0030] 每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片；

[0031] 所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，获得重排后数组，并将所述重排后数组作为所述数据分片。

[0032] 第三方面，本申请实施例提供一种电子设备，包括：处理器、存储器和总线，其中，

[0033] 所述处理器和所述存储器通过所述总线完成相互间的通信；

[0034] 所述存储器存储有可被所述处理器执行的程序指令，所述处理器调用所述程序指令能够执行第一方面的方法。

[0035] 第四方面，本申请实施例提供一种非暂态计算机可读存储介质，包括：

[0036] 所述非暂态计算机可读存储介质存储计算机指令，所述计算机指令使所述计算机执行第一方面的方法。

[0037] 本申请的其他特征和优点将在随后的说明书阐述，并且，部分地从说明书中变得显而易见，或者通过实施本申请实施例了解。本申请的目的和其他优点可通过在所写的说明书、权利要求书、以及附图中所特别指出的结构来实现和获得。

附图说明

[0038] 为了更清楚地说明本申请实施例的技术方案，下面将对本申请实施例中所需要使用的附图作简单地介绍，应当理解，以下附图仅示出了本申请的某些实施例，因此不应被看作是对范围的限定，对于本领域普通技术人员来讲，在不付出创造性劳动的前提下，还可以根据这些附图获得其他相关的附图。

[0039] 图1为本申请实施例提供的一种多方安全的数据重排方法流程示意图；

[0040] 图2为本申请实施例提供的一种选取子数据分片的示意图；

[0041] 图3为本申请实施例提供的perfect shuffle network的网络结构示意图；

[0042] 图4为本申请实施例提供的一种交换门的原理图；

[0043] 图5为本申请实施例提供的参与方1产生交换系数的原理图；

[0044] 图6为本申请实施例提供的参与方2产生交换系数的原理图；

[0045] 图7为本申请实施例提供的减法算子原理图；

[0046] 图8为本申请实施例提供的乘法算子原理图；

[0047] 图9为本申请实施例提供的一种bitonic merge network的网络结构示意图；

[0048] 图10为本申请实施例提供的一种数据比较示意图；

[0049] 图11为本申请实施例提供的一种随机网络结构示意图；

[0050] 图12为本申请实施例提供的一种对数据进行重排的方法流程示意图；

[0051] 图13为本申请实施例提供的一种多方安全的数据重排装置结构示意图；

[0052] 图14为本申请实施例提供的电子设备实体结构示意图。

具体实施方式

[0053] 下面将结合附图对本申请技术方案的实施例进行详细的描述。以下实施例仅用于更加清楚地说明本申请的技术方案,因此只作为示例,而不能以此来限制本申请的保护范围。

[0054] 除非另有定义,本文所使用的所有的技术和科学术语与属于本申请的技术领域的技术人员通常理解的含义相同;本文中所使用的术语只是为了描述具体的实施例的目的,不是旨在于限制本申请;本申请的说明书和权利要求书及上述附图说明中的术语“包括”和“具有”以及它们的任何变形,意图在于覆盖不排他的包含。

[0055] 在本申请实施例的描述中,技术术语“第一”“第二”等仅用于区别不同对象,而不能理解为指示或暗示相对重要性或者隐含指明所指示的技术特征的数量、特定顺序或主次关系。在本申请实施例的描述中,“多个”的含义是两个以上,除非另有明确具体的限定。

[0056] 在本文中提及“实施例”意味着,结合实施例描述的特定特征、结构或特性可以包含在本申请的至少一个实施例中。在说明书中的各个位置出现该短语并不一定均是指相同的实施例,也不是与其它实施例互斥的独立的或备选的实施例。本领域技术人员显式地和隐式地理解的是,本文所描述的实施例可以与其它实施例相结合。

[0057] 在本申请实施例的描述中,术语“和/或”仅仅是一种描述关联对象的关联关系,表示可以存在三种关系,例如A和/或B,可以表示:单独存在A,同时存在A和B,单独存在B这三种情况。另外,本文中字符“/”,一般表示前后关联对象是一种“或”的关系。

[0058] 在本申请实施例的描述中,术语“多个”指的是两个以上(包括两个),同理,“多组”指的是两组以上(包括两组),“多片”指的是两片以上(包括两片)。

[0059] 在本申请实施例的描述中,除非另有明确的规定和限定,技术术语“安装”“相连”“连接”“固定”等术语应做广义理解,例如,可以是固定连接,也可以是可拆卸连接,或成一体;也可以是机械连接,也可以是电连接;可以是直接相连,也可以通过中间媒介间接相连,可以是两个元件内部的连通或两个元件的相互作用关系。对于本领域的普通技术人员而言,可以根据具体情况理解上述术语在本申请实施例中的具体含义。

[0060] 为便于对本申请实施例的理解,对本申请实施例中所涉及到的相关概念进行解释。

[0061] Shuffle,就是基于某一给定的顺序,对数据进行重排列,以达到随机乱序的目的。举个例子:数组A=[0,1,2,3,4,5,6,7,8,9], 对数组进行重排列后:$A^{'}$=[5,7,9,2,3,8,0,1,4,6]。简单的说,重排列就是对数据的元素按照一种指定的方式调换位置。对数据进行重排列,在机器学习算法、联合统计、全匿联邦学习中经常用到,比如机器学习XGBOOST的分箱逻辑、全匿踪联邦学习中匿踪对齐及匿踪推理评估等。

[0062] 多方安全计算是指在无可信第三方的情况下,多个参与方共同计算一个目标函数,并且保证每一方仅获取自己的计算结果,无法通过计算过程中的交互数据推测出其他任意一方的输入数据。

[0063] 以两个参与方为例:

[0064] 对于数据x,秘密分享方式如下: 若x = x1 + x2, 则x1和x2是x的秘密分享。

[0065] 当有两个计算参与方时，一方拥有数据x1，另一方拥有数据x2，则这两个参与方就算各自拥有数据x的其中一份分片数据了，并且任意一方无法单独推测原始数据x。

[0066] 全匿踪联邦学习，是一种保护用户交集及非交集等全流程敏感数据的联邦学习范式。

[0067] 随机种子：计算机中产生的随机数是伪随机数，所谓的‘伪’，意思是这些数其实是有规律的，只不过因为算法规律太复杂，很难看出来而已。但是，再厉害的算法，如果没有一个初始值，它也不可能凭空造出一系列随机数来，上述种子就是该初始值。

[0068] random随机数的生成过程为：可以将这套用于生成随机数的复杂的算法看成一个黑盒，把准备好的种子输入黑盒中，黑盒输出两个结果，一个是随机数，另一个是保证能生成下一个随机数的新的种子，把新的种子放进黑盒，又得到一个新的随机数和一个新的种子，依此类推。

[0069] 本申请发明人经过长期研究发现，目前重排网络，例如：Shuffle Network，只能对长度为2^n的数据进行重排，这将限制待重排数据的长度，从而对待重排数据要求较为苛刻，并且这种重排网络不是针对隐私计算场景的，而是普通明文计算场景。为了解决该技术问题，使得重排网络能够对任意长度的数据进行重排，提出了一种多方安全的数据重排方法，该方法通过多次迭代进行重排，每次迭代均从原始数据中选取满足重排网络要求的预设数量的数据进行本轮迭代，并且本轮迭代获得的结果作为下一轮迭代的输入，直到将原始数据中所有的数据均参与过重排为止。本申请实施例提出的重排网络通过秘密共享的方式适用与隐私计算场景，实现了安全的、密态的shuffle任务，并且能够针对任意长度的数据进行重排。

[0070] 可以理解的是，本申请实施例提供的多方安全的数据重排方法可以应用于电子设备，该电子设备包括终端以及服务器；其中终端具体可以为智能手机、平板电脑、计算机、个人数字助理(Personal Digital Assitant，PDA)等；服务器具体可以为应用服务器，也可以为Web服务器。

[0071] 图1为本申请实施例提供的一种多方安全的数据重排方法流程示意图，如图1所示，该方法应用于多方安全计算系统中的参与方，所述多方安全计算系统中包括多个参与方，每一所述参与方中存储有原始数据对应的一数据分片；可以理解的是，参与方的数据分片组合后可以构成完整的原始数据，并且，每个参与方的数据分片中数据的位置与原始数据中对应数据位置是相同的，例如：原始数据为[x,y,z]，共有两个参与方，参与方1中的数据分片为[x1,y1,z1]，参与方2中的数据分片为[x2,y2,z2]，其中，x1和x2构成x，y1和y2构成y，z1和z2构成z。每一所述参与方随机生成一个重排网络，并与其他参与方秘密共享所述重排网络。为了便于描述，本申请实施例中均以多方安全计算系统包括两个参与方(即参与方1和参与方2)为例进行说明，该方法包括：

[0072] 步骤101：每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片。

[0073] 其中，数据分片可以以数组的形式存储在参与方中，可以理解的是，参与方可以是终端，也可以是服务器。预设数量为根据原始数据的长度确定的，例如：假设原始数据的长度为n，根据公式$\lceil m \rceil = \log_2 n$确定m的取值，然后预设数量为$2^m$。可以理解的是，本申请实

施例中的参与方可以是指需要进行多方联合建模任务的各数据提供方，例如：银行和运营商之间进行联邦学习建模；汽车厂商与保险公司进行联合建模等。原始数据是指模型训练的样本数据，可以包括样本id、特征数据、标签数据等。可以理解的是，在对模型进行训练时，为了不暴露交集样本信息，可以先对原始数据进行重排，打乱原有的id与特征数据的对应关系，从而使得各参与方无法推知id与特征的对应关系。在模型训练或者评估的时候，由于无需id参与，仅需特征参与，因此可以支持模型在样本乱序且密态的形式下，进行训练或者评估，不会对最终结果产生影响。

[0074] 步骤102：所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，获得重排后数组，并将所述重排后数组作为所述数据分片，并继续执行步骤101，直至完成对所述数据分片中所有的数据被重排，获得对所述原始数据进行重排后的重排数据。

[0075] 其中，参与方1本地随机产生一个重排网络1，并与参与方2秘密共享该重排网络1，利用该重排网络1对参与方1选出来的子数据分片1和参与方2选出来的子数据分片2进行重排，获得重排后数组。参与方2本地随机产生一个重排网络2，并与参与方1秘密共享该重排网络2，利用该重排网络2对参与方1选出来的子数据分片1和参与方2选出来的子数据分片2进行重排，获得重排后数组。从而，参与方1和参与方2完成了对原始数据中的 2^m 个数据的重排。原始数据中还有 $(n-2^m)$ 个数据没有经过重排，因此还要进行下一轮的重排操作，将本轮重排后数组作为下一轮重排的输入，并继续执行步骤101，直至所有的数据都被重排为止，获得重排后数组。

[0076] 本申请实施例针对每次迭代重排，各参与方均从数据分片中选取预设数量的数据，通过秘密共享的重排网络进行数据的重排，由于预设数量可以为任意长度，从而解决了现有技术中只能对固定长度的数据进行重排的问题。

[0077] 在上述实施例的基础上，所述每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片，包括：

[0078] 各参与方同步生成随机种子，并根据所述随机种子从所述数据分片中获取预设数量的数据，构成所述子数据分片。

[0079] 其中，随机种子的作用是保证获取到的数据分片中的数据保持对应顺序的关系。例如：F1和F2是原始数据F的两个分片数据，当从F1中选取了某些数据时，也需要从F2中选出对应位置的数据，这样可以保证数据的正确性和完整性。如图2所示，F=[10,-1,5,-8]，F1=[2,5,-10,-1]，F2=[8,-6,15,-7]。当从F1中选择了2和-10构成子数据分片1后，也应当从F2中选择对应位置的数据，即从F2中选择8和15构成子数据分片2。另外，每迭代一次，随机种子需要进行更新，例如：可以每次加1，用于下一次迭代时生成随机索引，并利用随机索引从数据分片中选择对应位置的数据。参与方1和参与方2中的随机种子需保持不同，即随机种子对应的值应相同。

[0080] 本申请实施例基于随机种子从数据分片中选取待重排的数据，使得各个参与方能够获取到各自数据分片中对应位置的数据，以确保各参与方采样的一致性。

[0081] 在上述实施例的基础上，各参与方在从各自存储的数据分片中确定参与重排的数据时，可通过如下方式确定：

[0082] 第一种:各参与方基于所述随机种子,以概率 $p_0=\frac{n-2^m}{n}$ 从上一次已经采样过的数据中进行不放回采样,以概率 $p_1=\frac{2^m}{n}$ 从上一次未采样过的数据中进行不放回采样,获取所述预设数量的数据。并不断重复上述过程,直至将数据分片中的数据都经过重排为止。

[0083] 第二种:各参与方基于所述随机种子,根据概率 $\frac{2^m}{n}$ 以有放回的方式从所述数据分片中获取 j 次所述预设数量的数据。

[0084] 其中,采样的次数 j 次满足 $(1-\Pi_i^j(1-\frac{2^m}{n}))>p_2$,n 为所述原始数据的长度;2^m 为所述预设数量;p_2 为所述原始数据中参与重排的各个数据的概率,并且,p_2 可根据实际情况进行设定,例如可以为0.99,0.95等。

[0085] 本申请实施例通过无放回或有放回的方式从各个数据分片中获取待重排的数据,保证数据分片中的每个数据均能够被作为待重排的数据。

[0086] 在上述实施例的基础上,重排网络可以根据实际情况进行选择,例如:可以是perfect shuffle network、bitonic merge network和random network等,下面针对每种重排网络进行介绍。

[0087] 一、perfect shuffle network

[0088] 图3为本申请实施例提供的perfect shuffle network的网络结构示意图,如图3所示,重排网络包括多个交换层,每一所述交换层包括多个交换门,图3中每个小方框是一个交换门,每一所述交换门对应一个交换系数,所述交换系数用于表征是否对输入的数据进行交换;所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作。交换系数可以用 k 表示,k 的取值可以为0或1,当 k 的取值为0时,表示不交换,当 k 的取值为1时,表示交换。图4为本申请实施例提供的一种交换门的原理图,A_{in} 为参与方1选取的用来进行重排的数据,B_{in} 为参与方2选取的用来进行重排的数据。秘密共享重排网络就是对每个交换门的交换系数进行秘密共享,因此,交换系数k是碎片化后的分片数据的形式,以秘密共享碎片化的方式存在。每个交换门执行的计算都在密文下进行,即,输入数据 A_{in} ,B_{in} 是秘密共享的密文,交换系数由其中一个参与方产生,之后在各个参与方之间秘密共享,因此也是秘密共享的密文。

[0089] 图5为本申请实施例提供的参与方1产生交换系数的原理图,如图5所示,参与方1产生交换系数k,k可以由[k]1+[k]2构成,参与方1将[k]2发送给参与方2,参与方1中的交换系数为[z]1=1-[k]1,参与方2的交换系数为[z]2=1-[k]2。

[0090] 图6为本申请实施例提供的参与方2产生交换系数的原理图,如图5所示,参与方2产生交换系数k,k可以由[k]1+[k]2构成,参与方2将[k]1发送给参与方1,参与方1中的交换系数为[z]1=1-[k]1,参与方2的交换系数为[z]2=1-[k]2。

[0091] 每个交换门内执行的算法也是秘密共享的算法。该算法可以是乘法和/或减法等。

[0092] 在进行重排时，可根据公式$\begin{cases} A_{out} = A_{in} * (1-k) + B_{in} * k \\ B_{out} = B_{in} * (1-k) + A_{in} * k \end{cases}$对输入到所述交换门的数据进行重排操作；其中，$A_{out}$为所述交换门输出所述参与方的输出数据；$k$为所述交换系数；$B_{out}$为所述交换门输出所述其他参与方的输出数据。

[0093] 图7为本申请实施例提供的减法算子原理图，如图7所示，参与方1与参与方2需要安全计算x和y之间的差值，参与方1持有原始数据x，参与方2持有原始数据y。参与方1利用加法碎片化方法，将原始数据x拆分为[x]1和[x]2两个分片数据，其中，x=[x]1+[x]2，同理，参与方2利用加法碎片化方法，将原始数据y拆分为[y]1和[y]2两个分片数据，其中，y=[y]1+[y]2。参与方1将[x]2发送给参与方2，参与方2将[y]1发送给参与方1。此时，参与方1持有数据分片[x]1和[y]1，参与方2持有数据分片[x]2和[y]2，然后各参与方各自在本地执行[z]1=[x]1-[y]1和[z]2=[x]2-[y]2，从而得到碎片化状态下的x-y的拆分后的分片数据。可以理解的是，当[z]1+[z]2就是x-y对应的值。

[0094] 图8为本申请实施例提供的乘法算子原理图，如图8所示，为了进行乘法计算，附加的信息就不再是简单的常数c，而是一个三元组a，b，c，满足：$\langle c \rangle = \langle a \rangle \cdot \langle b \rangle$，根据共享的a，b，c可以计算出e和f：

[0095] $$\langle e \rangle^{\alpha} = \langle x \rangle^{\alpha} - \langle a \rangle^{\bar{\alpha}}$$
$$\langle f \rangle^{\bar{\alpha}} = \langle y \rangle^{\alpha} - \langle b \rangle^{\alpha}$$

[0096] 可以理解的是，上述公式为参与方1的计算方式，参与方2的计算方式与参与方1的计算方式类似，此处不再赘述。

[0097] 双方各自计算自己的e和f并共享，最终双方都可以得到真正的e和f值：

[0098] $$e = \mathrm{Re}c(\langle e \rangle^{\mathrm{A}}, \langle e \rangle^{\mathrm{B}})$$
$$f = \mathrm{Re}c(\langle f \rangle^{\mathrm{A}}, \langle f \rangle^{\mathrm{B}})$$

[0099] 最后的乘法结果是：

[0100] $$\langle x \cdot y \rangle^{\mathrm{A}} = f \cdot \langle a \rangle^{\mathrm{A}} + e \cdot \langle b \rangle^{\mathrm{A}} + \langle c \rangle^{\mathrm{A}}$$
$$\langle x \cdot y \rangle^{\mathrm{B}} = e \cdot f + f \cdot \langle a \rangle^{\mathrm{B}} + e \cdot \langle b \rangle^{\mathrm{B}} + \langle c \rangle^{\mathrm{B}}$$

[0101] 经过上述计算，即完成了对任意长度数据在两方场景下，无需可信第三方，高效快速地基于秘密共享机制进行安全shuffle。

[0102] 二、bitonic merge network

[0103] 图9为本申请实施例提供的一种bitonic merge network的网络结构示意图，如图9所示，图9示出的是对16个数据进行升序排序，图中的箭头代表comparator(比较器)。如果网络上的两条线接在同一个comparator的两端，那么这两个线上此时的数据要进行比较，其中数值较大的放在箭头所指的方向，如图10所示。

[0104] 图9中可以分为三种区域，第一区域中，在上半区域中的值要和下半部分进行比较，在同一红色区域中所有箭头指向方向相同(向下或向上)。当这样一个红色区域的箭头所指方向为下时，当接收长度为n的Bitonic序列，经过该红色区域计算后，最小n/2个元素

会被调至的上半区域，最大的n/2个元素会被调至下半区域，且上下两个区域的序列仍为Bitonic序列。

[0105] 第二区域中，接收长度为n的Bitonic序列，然后把它传递给一个同样需要输入大小为n的Bitonic序列的红色区域，将计算结果传递给两个方向相同，需要输入大小为n/2的Bitonic序列的红色区域。然后每个区域又分别再传递给两个方向相同、且需要输入大小为n/(2×2)的Bitonic序列的红色区域...依此类推。经过第二区域计算后，输入的Bitonic序列变成一个完全递增的序列。

[0106] 第三区域与第二区域的计算方法相同，最终输出完全递减的序列。

[0107] 由于Bitonic sort网络最后一部分是第二区域，因而最后整体的输出是一个递增序列。根据实际业务的需求，可以输出最终的密态序列作为shuffle结果，也可以输出倒数K轮的交换结果作为最终的shuffle结果，这些序列都可以是较好的乱序序列，无法推知与原始序列的对应关系。

[0108] 三、random network

[0109] 图11为本申请实施例提供的一种随机网络结构示意图，如图11所示，每一个圆圈表示一个样本，样本之间的交换对象选择是完全随机的，交换门的形成是完全基于随机选择机制形成，呈现不可测规律。比如1和2可以进行可选交换，通过交换门决定是否进行密态交换。同样的，1也可以与6进行交换门的计算。在每一轮中，1与2、1与6可以独立存在，也可以同时存在。每一轮交换的总体数量依然是2^m，因为需要成双成对抽取。

[0110] 在上述实施例的基础上，各参与方之间进行重排网络的秘密共享时，可以通过比特进行秘密共享，例如：针对交换系数0或1的共享，如果表示成int64类型，其大小为64bit，本申请可将0或1使用1bit来表示，从而进一步减少隐私计算的通信量级，进而提升整体的计算性能。

[0111] 在另一实施例中，图12为本申请实施例提供的一种对数据进行重排的方法流程示意图，如图12所示，包括：

[0112] 步骤1201：原始id数值转换；参与方1和参与方2分别对其存储的原始id进行映射转换，如图13所示。

[0113] 步骤1202：样本矩阵增广；表1为参与方1的原始特征数据D_0，表2为参与方2的原始特征数据D_1。将原始特征数据D_0和原始特征数据D_1进行特征数量同步，即，将特征矩阵即标签进行补齐，得到增广后的特征数据D_0'，D_1'，如表3和表4所示；

[0114] 表1

[0115]

id	Y	X_{a1}	X_{a2}	X_{a3}
124360	0	1.3	5.2	3
328492	1	2.5	3.3	-2
572683	0	-1	0.5	0.2
930913	1	0.9	0.12	1

[0116] 表2

[0117]

id	X_{b1}	X_{b2}
748329	0.89	1.41
328492	2.3	1.9
930913	-1.2	-0.1

[0118] 表3

[0119]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
124360	0	1.3	5.2	3	0	0
328492	1	2.5	3.3	-2	0	0
572683	0	-1	0.5	0.2	0	0
930913	1	0.9	0.12	1	0	0

[0120] 表4

[0121]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
748329	0	0	0	0	0.89	1.41
328492	0	0	0	0	2.3	1.9
930913	0	0	0	0	-1.2	-0.1

[0122] 步骤1203：增广矩阵碎片化；对x进行碎片化 $Shr^{i}_{A(x)}$ ：参与方1选择 $r \in_R Z_{2^l}$ ，使 $\langle x \rangle^{i}_{A} = x - r$ ，并且发送 r 给 P_{1-i} ，使 $\langle x \rangle^{1-i}_{A} = r$ 。参与方2对 $D_0^{'}$ 进行秘密共享（id也需要碎片化）。如表5所示：

[0123] 表5

[0124]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<124360>	<0>	<1.3>	<5.2>	<3>	<0>	<0>
<328492>	<1>	<2.5>	<3.3>	<-2>	<0>	<0>
<572683>	<0>	<-1>	<0.5>	<0.2>	<0>	<0>
<930913>	<1>	<0.9>	<0.12>	<1>	<0>	<0>

[0125] 同理，参与方2对 $D_1^{'}$ 进行本地碎片化并秘密共享，如表6所示：

[0126] 表6

[0127]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<748329>	<0>	<0>	<0>	<0>	<0.89>	<1.41>
<328492>	<0>	<0>	<0>	<0>	<2.3>	<1.9>
<930913>	<0>	<0>	<0>	<0>	<-1.2>	<-0.1>

[0128] 需要注意的是,<x>表示x的碎片态。

[0129] MPC Concat执行得到拼接后的碎片矩阵 D_f ,如表7所示:

[0130] 表7

[0131]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<124360>	<0>	<1.3>	<5.2>	<3>	<0>	<0>
<328492>	<1>	<2.5>	<3.3>	<-2>	<0>	<0>
<572683>	<0>	<-1>	<0.5>	<0.2>	<0>	<0>
<930913>	<1>	<0.9>	<0.12>	<1>	<0>	<0>
<748329>	<0>	<0>	<0>	<0>	<0.89>	<1.41>
<328492>	<0>	<0>	<0>	<0>	<2.3>	<1.9>
<930913>	<0>	<0>	<0>	<0>	<-1.2>	<-0.1>

[0132] 步骤1204:密态样本重排;采用上述实施例所提供的重排方法对上述表7中的数据进行重排,可获得如表8所示的重排后数据。

[0133] 表8

[0134]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<930913>	<1>	<0.9>	<0.12>	<1>	<0>	<0>
<572683>	<0>	<-1>	<0.5>	<0.2>	<0>	<0>
<328492>	<1>	<2.5>	<3.3>	<-2>	<0>	<0>
<328492>	<0>	<0>	<0>	<0>	<2.3>	<1.9>
<748329>	<0>	<0>	<0>	<0>	<0.89>	<1.41>
<930913>	<0>	<0>	<0>	<0>	<-1.2>	<-0.1>
<124360>	<0>	<1.3>	<5.2>	<3>	<0>	<0>

[0135] 步骤1205:密态id排序;

[0136] 步骤1206:密态对齐;

[0137] 步骤1207:输出最终对其样本碎片结果,如表9所示:

[0138] 表9

[0139]

id	Y	X_{a1}	X_{a2}	X_{a3}	X_{b1}	X_{b2}
<328492>	<1>	<2.5>	<3.3>	<-2>	<2.3>	<1.9>
</>	<0>	<0>	<0>	<0>	<0>	<0>
<930913>	<1>	<0.9>	<0.12>	<1>	<-1.2>	<-0.1>

[0140] 图13为本申请实施例提供的一种多方安全的数据重排装置结构示意图,该装置可以是电子设备上的模块、程序段或代码。应理解,该装置与上述图1方法实施例对应,能够执

行图1方法实施例涉及的各个步骤，该装置具体的功能可以参见上文中的描述，为避免重复，此处适当省略详细描述。所述装置包括多个参与方，例如参与方1，参与方2，…，参与方N，其中N为大于1的整数。该装置运行在多方安全计算系统中的参与方中，多方安全计算系统中包括多个参与方，每一所述参与方中存储有原始数据对应的一数据分片；每一所述参与方随机生成一个重排网络，并与其他参与方秘密共享所述重排网络；所述装置包括：

[0141] 所述参与方按照如下步骤进行多轮迭代，直至完成对所述数据分片中所有的数据被重排，获得对所述原始数据进行重排后的重排数据；所述步骤包括：

[0142] 每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片；

[0143] 所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，获得重排后数组，并将所述重排后数组作为所述数据分片。

[0144] 在上述实施例的基础上，所述每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片，包括：

[0145] 各参与方同步生成随机种子，并根据所述随机种子从所述数据分片中获取预设数量的数据，构成所述子数据分片。

[0146] 在上述实施例的基础上，所述根据所述随机种子从所述数据分片中获取预设数量的数据，包括：

[0147] 各参与方基于所述随机种子，以概率 $p_0=\frac{n-2^m}{n}$ 从上一次已经采样过的数据中进行不放回采样，以概率 $p_1=\frac{2^m}{n}$ 从上一次未采样过的数据中进行不放回采样，获取所述预设数量的数据；或，

[0148] 各参与方基于所述随机种子，根据概率 $\frac{2^m}{n}$ 以有放回的方式从所述数据分片中获取 j 次所述预设数量的数据；

[0149] 其中，$(1-\Pi_i^j(1-\frac{2^m}{n}))>p_2$，$n$ 为所述原始数据的长度；2^m 为所述预设数量；p_2 为所述原始数据中参与重排的各个数据的概率。

[0150] 在上述实施例的基础上，所述重排网络包括多个交换层，每一所述交换层包括多个交换门，每一所述交换门对应一个交换系数，所述交换系数用于表征是否对输入的数据进行交换；所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，包括：

[0151] 根据公式 $\begin{cases}A_{out}=A_{in}*(1-k)+B_{in}*k\\B_{out}=B_{in}*(1-k)+A_{in}*k\end{cases}$ 对输入到所述交换门的数据进行重排操作；其中，A_{out} 为所述交换门输出所述参与方的输出数据；A_{in} 为所述参与方输入到所述交换门的输入数据；k 为所述交换系数；B_{out} 为所述交换门输出所述其他参与方的输出数据；B_{in} 为其他参与方输入到所述交换门的输入数据。

[0152] 在上述实施例的基础上，所述重排网络包括完备网络、双调合并网络或随机网络。

[0153] 在上述实施例的基础上，所述原始数据的长度为n，根据公式$\lceil m \rceil = \log_2 n$确定所述预设数量。

[0154] 在上述实施例的基础上，各参与方之间通过比特进行秘密共享所述重排网络。

[0155] 图14为本申请实施例提供的电子设备实体结构示意图，如图14所示，所述电子设备，包括：处理器(processor)1401、存储器(memory)1402和总线1403；其中，

[0156] 所述处理器1401和存储器1402通过所述总线1403完成相互间的通信；

[0157] 所述处理器1401用于调用所述存储器1402中的程序指令，以执行上述各方法实施例所提供的方法，例如包括：所述参与方按照如下步骤进行多轮迭代，直至完成对所述数据分片中所有的数据被重排，获得对所述原始数据进行重排后的重排数据；所述步骤包括：每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片；所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，获得重排后数组，并将所述重排后数组作为所述数据分片。

[0158] 处理器1401可以是一种集成电路芯片，具有信号处理能力。上述处理器1401可以是通用处理器，包括中央处理器(Central Processing Unit，CPU)、网络处理器(Network Processor，NP)等；还可以是数字信号处理器(DSP)、专用集成电路(ASIC)、现场可编程门阵列(FPGA)或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。其可以实现或者执行本申请实施例中公开的各种方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。

[0159] 存储器1402可以包括但不限于随机存取存储器(Random Access Memory，RAM)，只读存储器(Read Only Memory，ROM)，可编程只读存储器(Programmable Read-Only Memory，PROM)，可擦除只读存储器(Erasable Programmable Read-Only Memory，EPROM)，电可擦除只读存储器(Electrically Erasable Programmable Read-Only Memory，EEPROM)等。

[0160] 本实施例公开一种计算机程序产品，所述计算机程序产品包括存储在非暂态计算机可读存储介质上的计算机程序，所述计算机程序包括程序指令，当所述程序指令被计算机执行时，计算机能够执行上述各方法实施例所提供的方法，例如包括：所述参与方按照如下步骤进行多轮迭代，直至完成对所述数据分片中所有的数据被重排，获得对所述原始数据进行重排后的重排数据；所述步骤包括：每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片；所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，获得重排后数组，并将所述重排后数组作为所述数据分片。

[0161] 本实施例提供一种非暂态计算机可读存储介质，所述非暂态计算机可读存储介质存储计算机指令，所述计算机指令使所述计算机执行上述各方法实施例所提供的方法，例如包括：所述参与方按照如下步骤进行多轮迭代，直至完成对所述数据分片中所有的数据被重排，获得对所述原始数据进行重排后的重排数据；所述步骤包括：每一所述参与方从自身存储的数据分片中获取预设数量的数据构成子数据分片；所述参与方与所述其他参与方基于秘密共享的所述重排网络对所述子数据分片进行重排操作，获得重排后数组，并将所述重排后数组作为所述数据分片。

[0162] 在本申请所提供的实施例中，应该理解到，所揭露装置和方法，可以通过其它的方

式实现。以上所描述的装置实施例仅仅是示意性的，例如，所述单元的划分，仅仅为一种逻辑功能划分，实际实现时可以有另外的划分方式，又例如，多个单元或组件可以结合或者可以集成到另一个系统，或一些特征可以忽略，或不执行。另一点，所显示或讨论的相互之间的耦合或直接耦合或通信连接可以是通过一些通信接口，装置或单元的间接耦合或通信连接，可以是电性，机械或其它的形式。

[0163] 另外，作为分离部件说明的单元可以是或者也可以不是物理上分开的，作为单元显示的部件可以是或者也可以不是物理单元，即可以位于一个地方，或者也可以分布到多个网络单元上。可以根据实际的需要选择其中的部分或者全部单元来实现本实施例方案的目的。

[0164] 再者，在本申请各个实施例中的各功能模块可以集成在一起形成一个独立的部分，也可以是各个模块单独存在，也可以两个或两个以上模块集成形成一个独立的部分。

[0165] 在本文中，诸如第一和第二等之类的关系术语仅仅用来将一个实体或者操作与另一个实体或操作区分开来，而不一定要求或者暗示这些实体或操作之间存在任何这种实际的关系或者顺序。

[0166] 以上所述仅为本申请的实施例而已，并不用于限制本申请的保护范围，对于本领域的技术人员来说，本申请可以有各种更改和变化。凡在本申请的精神和原则之内，所作的任何修改、等同替换、改进等，均应包含在本申请的保护范围之内。

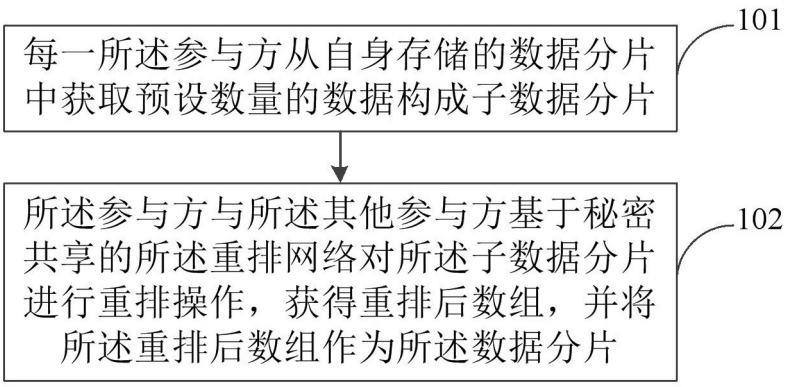


图1

F		F1		F2
10	=	2	+	8
-1		5		-6
5		-10		15
-8		-1		-7

图2

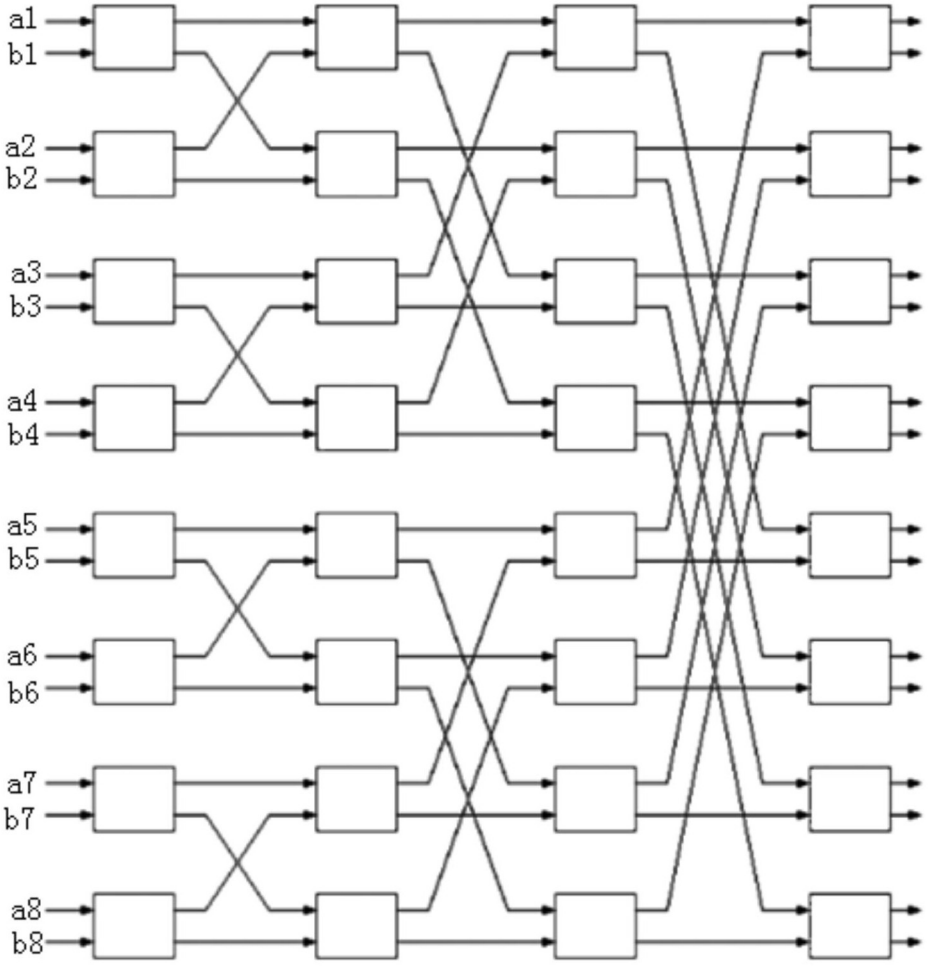

a1
b1
a2
b2
a3
b3
a4
b4
a5
b5
a6
b6
a7
b7
a8
b8

图3

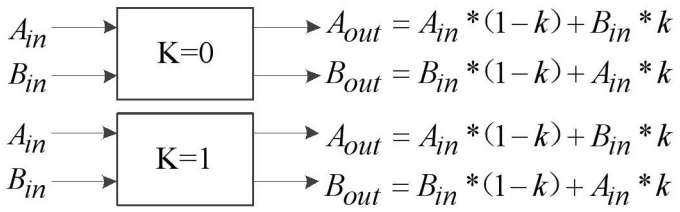

A_{in}
B_{in}
K=0
$A_{out} = A_{in} * (1-k) + B_{in} * k$
$B_{out} = B_{in} * (1-k) + A_{in} * k$
A_{in}
B_{in}
K=1
$A_{out} = A_{in} * (1-k) + B_{in} * k$
$B_{out} = B_{in} * (1-k) + A_{in} * k$

图4

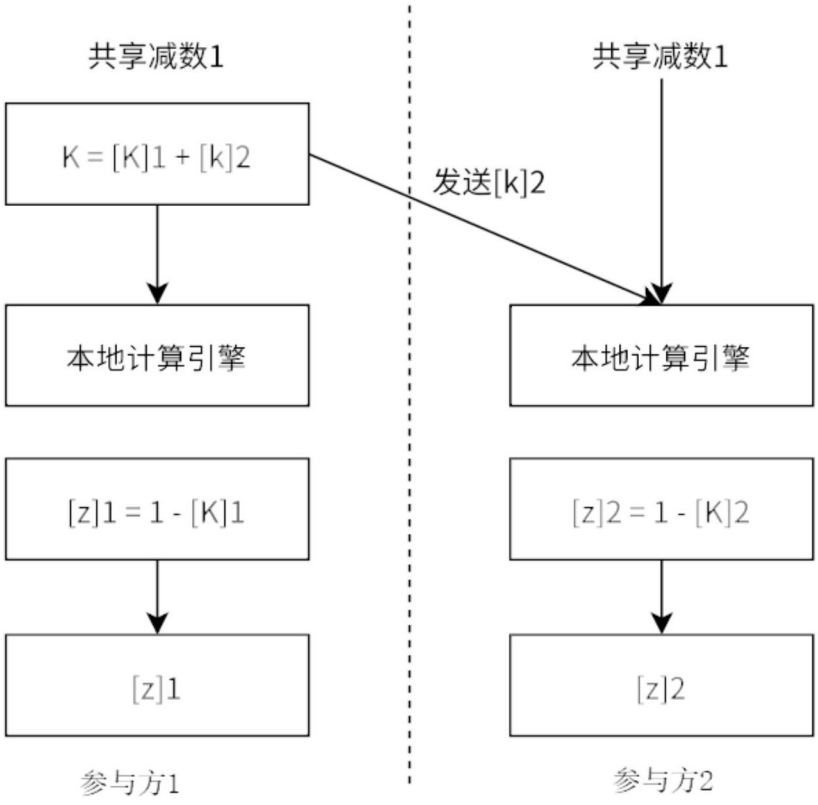

共享减数1
共享减数1
K = [K]1 + [k]2
发送[k]2
本地计算引擎
本地计算引擎
[z]1 = 1 - [K]1
[z]2 = 1 - [K]2
[z]1
[z]2
参与方1
参与方2

图5

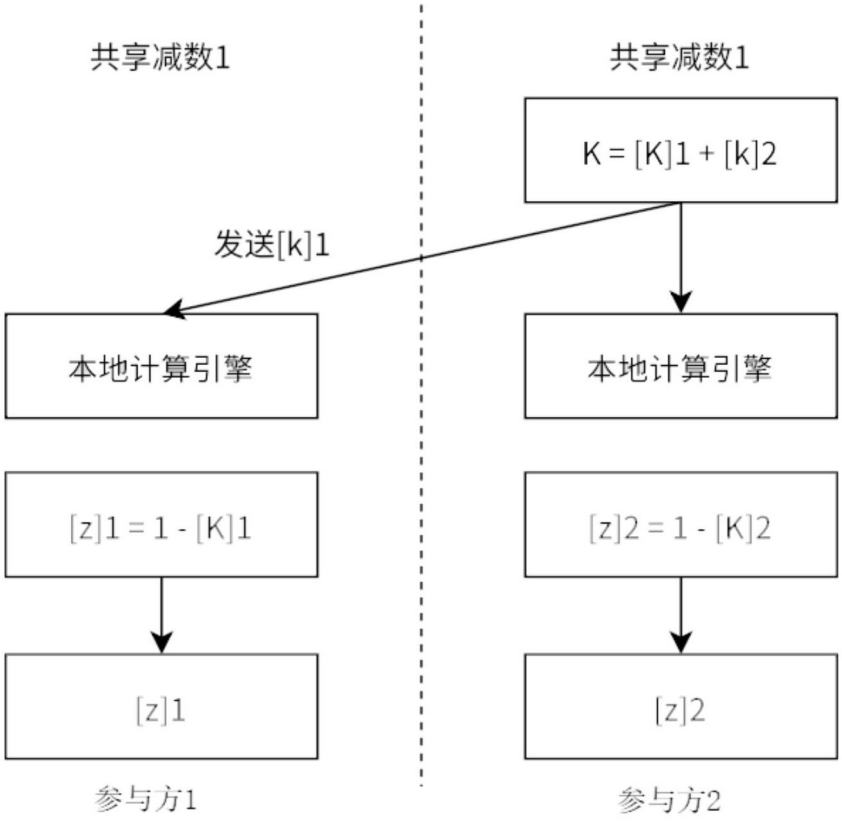

共享减数1
共享减数1
K = [K]1 + [k]2
发送[k]1
本地计算引擎
本地计算引擎
[z]1 = 1 - [K]1
[z]2 = 1 - [K]2
[z]1
[z]2
参与方1
参与方2

图6

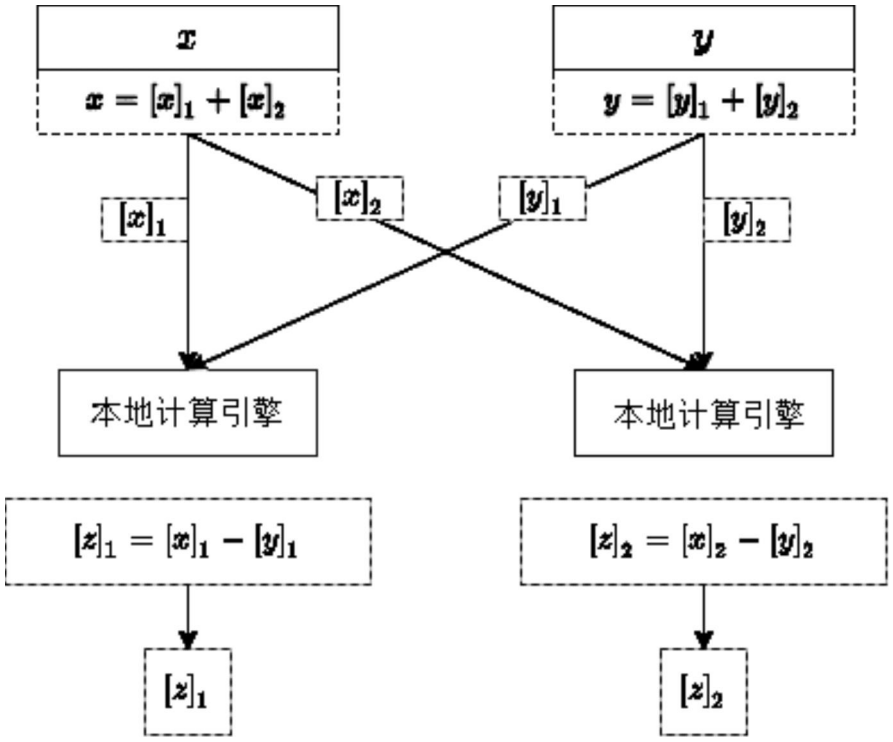

x
$x = [x]_1 + [x]_2$
y
$y = [y]_1 + [y]_2$
$[x]_1$
$[x]_2$
$[y]_1$
$[y]_2$
本地计算引擎
本地计算引擎
$[z]_1 = [x]_1 - [y]_1$
$[z]_2 = [x]_2 - [y]_2$
$[z]_1$
$[z]_2$

图7

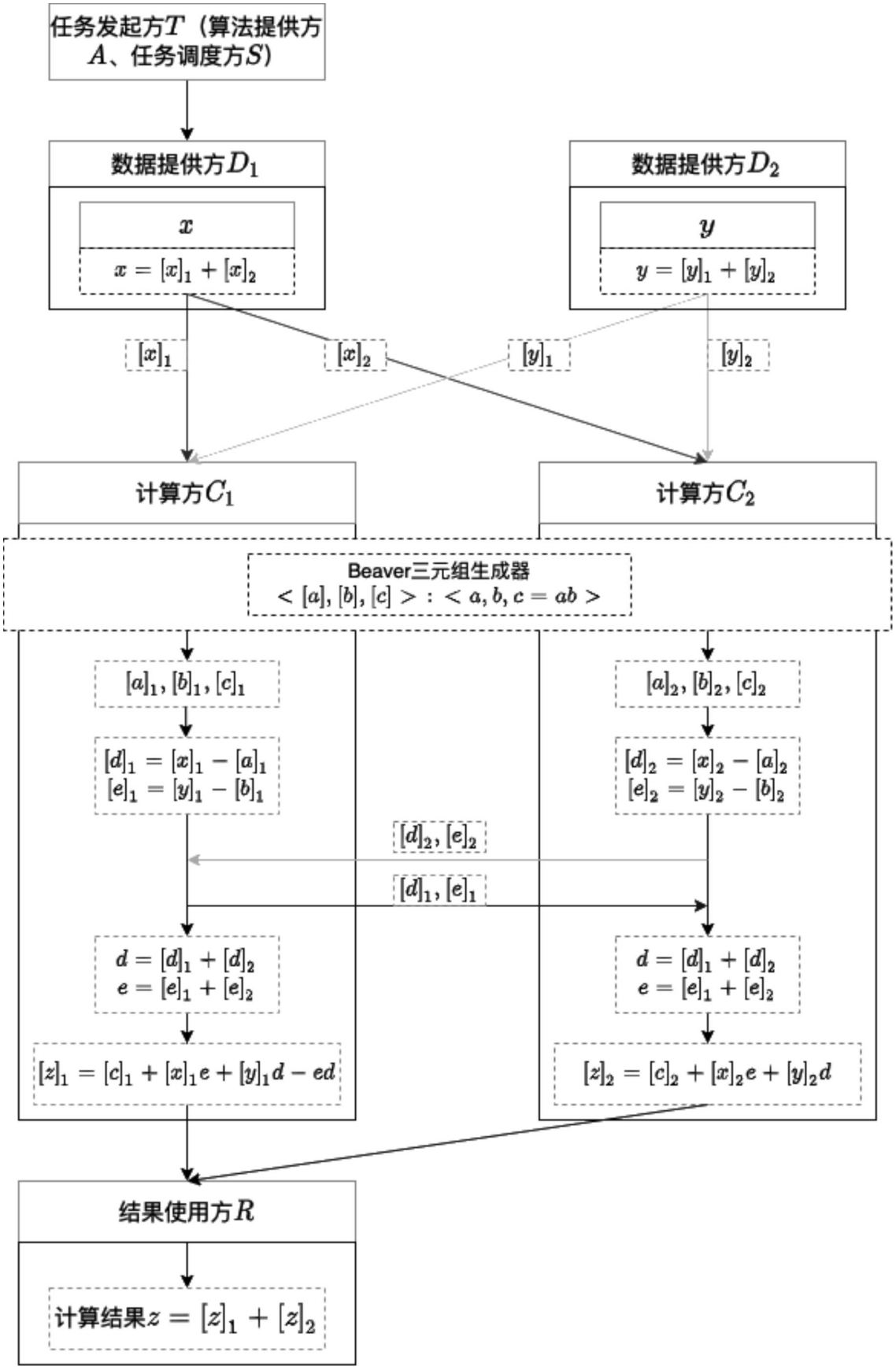

任务发起方T（算法提供方A、任务调度方S）
数据提供方D_1
x
$x = [x]_1 + [x]_2$
数据提供方D_2
y
$y = [y]_1 + [y]_2$
$[x]_1$
$[x]_2$
$[y]_1$
$[y]_2$
计算方C_1
计算方C_2
Beaver三元组生成器
$< [a], [b], [c] > : < a, b, c = ab >$
$[a]_1, [b]_1, [c]_1$
$[a]_2, [b]_2, [c]_2$
$[d]_1 = [x]_1 - [a]_1$
$[e]_1 = [y]_1 - [b]_1$
$[d]_2 = [x]_2 - [a]_2$
$[e]_2 = [y]_2 - [b]_2$
$[d]_2, [e]_2$
$[d]_1, [e]_1$
$d = [d]_1 + [d]_2$
$e = [e]_1 + [e]_2$
$d = [d]_1 + [d]_2$
$e = [e]_1 + [e]_2$
$[z]_1 = [c]_1 + [x]_1 e + [y]_1 d - ed$
$[z]_2 = [c]_2 + [x]_2 e + [y]_2 d$
结果使用方R
计算结果$z = [z]_1 + [z]_2$

图8

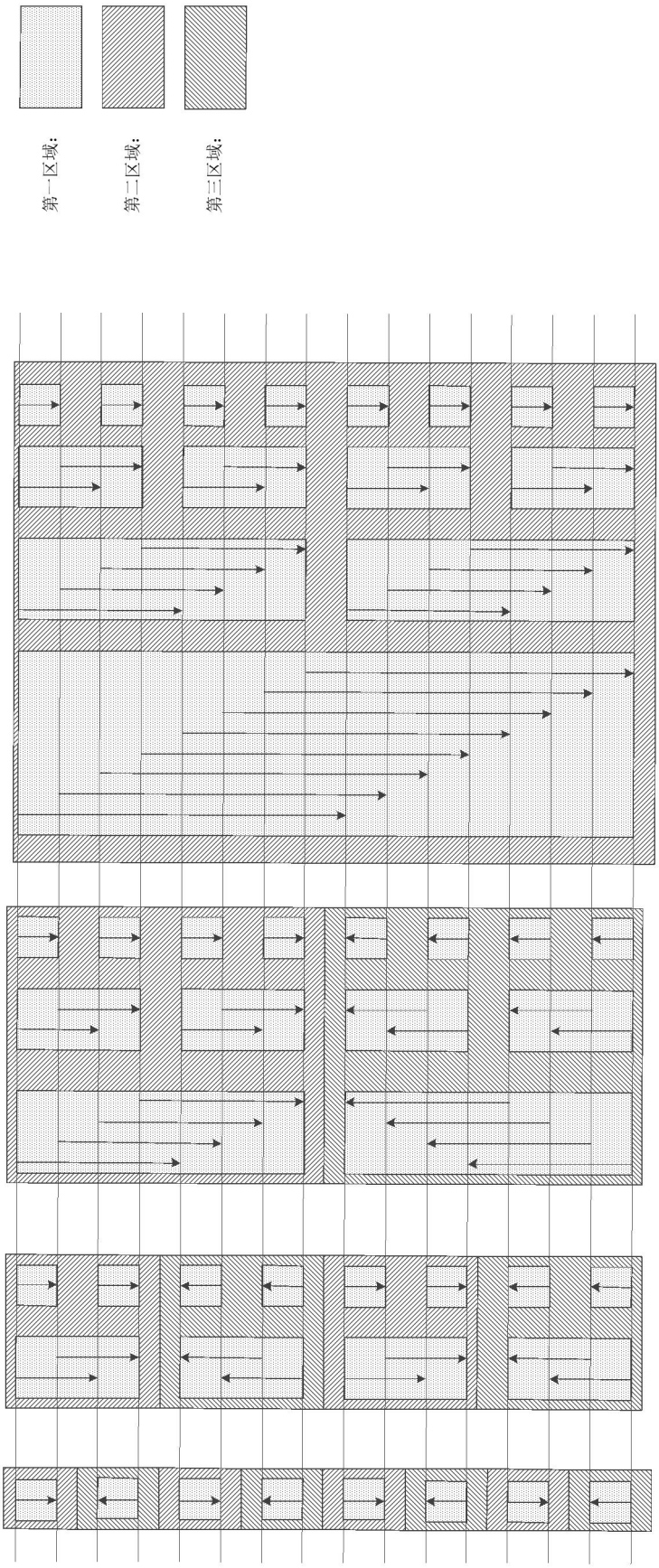


图9

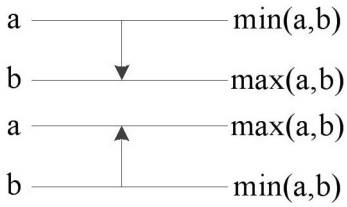


图10

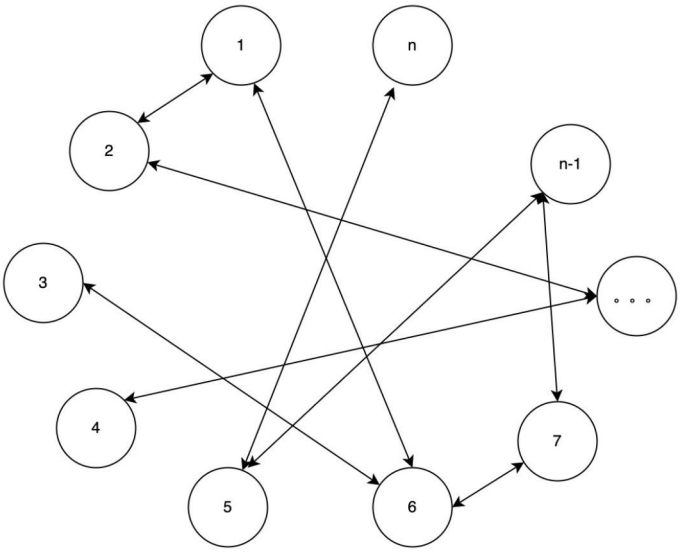


图11

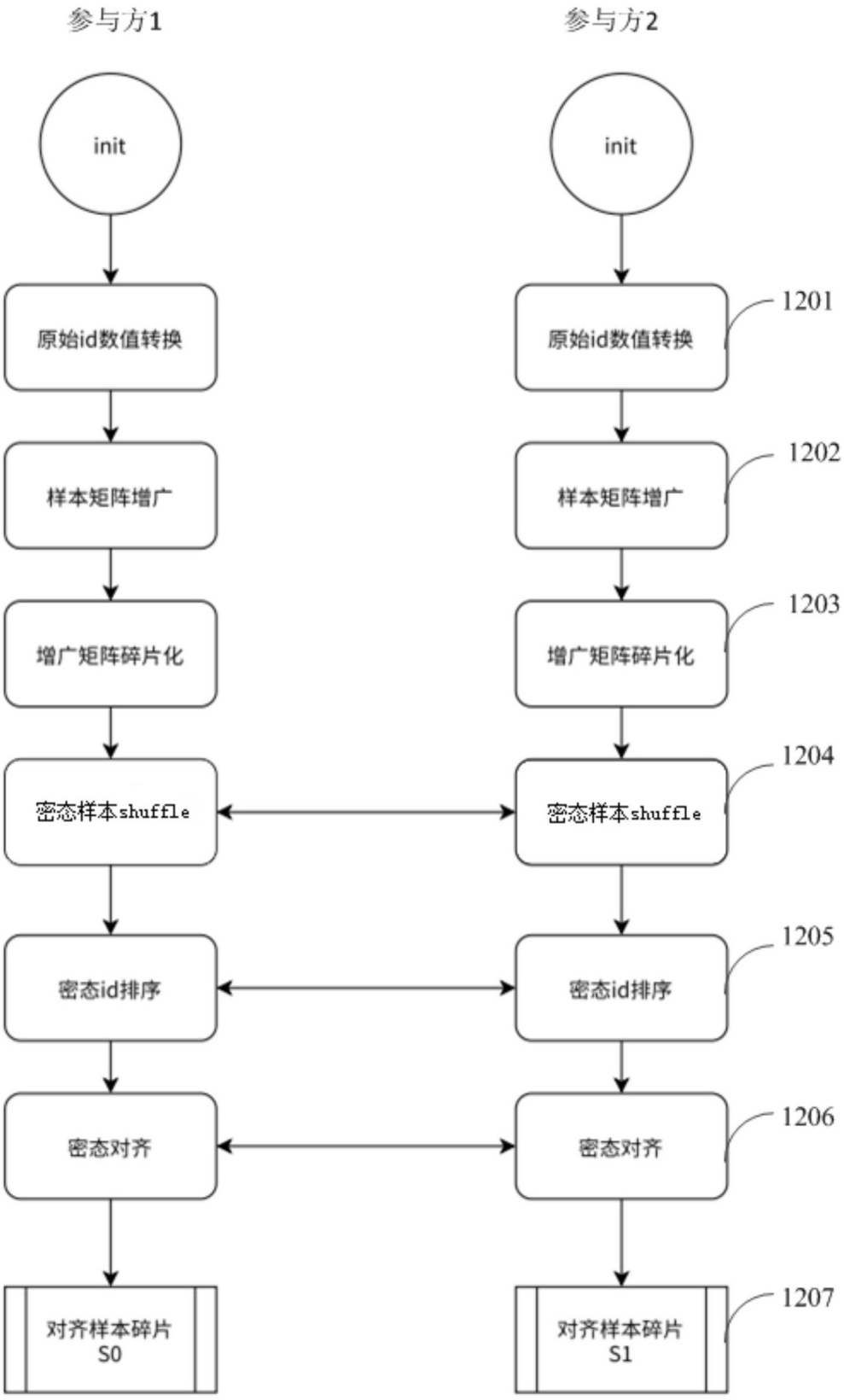

参与方1
参与方2
init
init
原始id数值转换
原始id数值转换
1201
样本矩阵增广
样本矩阵增广
1202
增广矩阵碎片化
增广矩阵碎片化
1203
密态样本shuffle
密态样本shuffle
1204
密态id排序
密态id排序
1205
密态对齐
密态对齐
1206
对齐样本碎片
S0
对齐样本碎片
S1
1207

图12

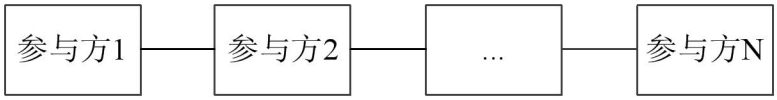


图13

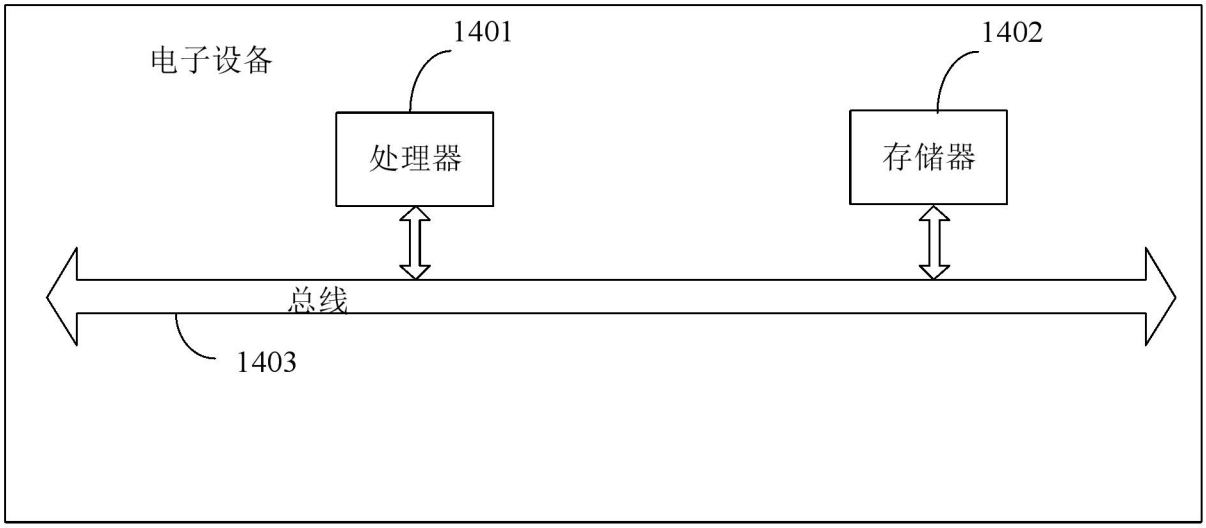


图14