(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 115587382 B
(45)授权公告日 2023.04.11

(21)申请号 202211598564.0

(22)申请日 2022.12.14

(65)同一申请的已公布的文献号
申请公布号 CN 115587382 A

(43)申请公布日 2023.01.10

(73)专利权人 富算科技(上海)有限公司
地址 200135 上海市浦东新区自由贸易试验区浦东大道1200号2层A区

(72)发明人 尤志强 卞阳 赵东

(74)专利代理机构 上海弼兴律师事务所 31283
专利代理师 罗朗 林嵩

(51)Int.Cl.
G06F 21/60(2013.01)
G06F 21/62(2013.01)

(56)对比文件
US 2015089243 A1,2015.03.26
US 2022179946 A1,2022.06.09

审查员 叶珊

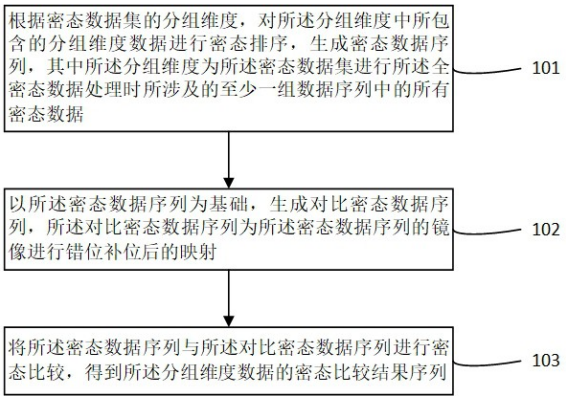
权利要求书2页 说明书16页 附图4页

(54)发明名称

全密态数据处理方法、装置、设备、介质

(57)摘要

本公开为一种全密态数据处理方法、装置、设备、介质。适用于多方安全计算中多个参与节点中的任一参与节点，全密态数据处理方法包括：根据密态数据集的分组维度，对分组维度中所包含的分组维度数据进行密态排序，生成密态数据序列，其中分组维度为密态数据集进行全密态数据处理时所涉及的至少一组数据序列中的所有密态数据；以密态数据序列为基础，生成对比密态数据序列，对比密态数据序列为密态数据序列的镜像进行错位补位后的映射；将密态数据序列与对比密态数据序列进行密态比较，得到分组维度数据的密态比较结果序列。实现了全密态数据处理，统计计算无需遍历全部数据以实现节约计算成本，而且隐私计算过程中数据全程密态以保证数据安全。



CN 115587382 B

1.一种全密态数据处理方法,适用于多方安全计算中多个参与节点中的任一参与节点,其特征在于,所述全密态数据处理方法包括:

根据密态数据集的分组维度,对所述分组维度中所包含的分组维度数据进行密态排序,生成密态数据序列,其中所述分组维度为所述密态数据集进行所述全密态数据处理时所涉及的至少一组数据序列中的所有密态数据;

以所述密态数据序列为基础,生成对比密态数据序列,所述对比密态数据序列为所述密态数据序列的镜像进行错位补位后的映射;

将所述密态数据序列与所述对比密态数据序列进行密态比较,得到所述分组维度数据的密态比较结果序列;

将所述密态比较结果序列提取非零数据,以形成类别序列;

将所述分组维度数据与所述类别序列进行一致性比较,以形成分类矩阵;

将聚合统计维度数据进行密态分类;其中,将聚合统计维度中的所有密态数据所形成的序列与分类矩阵进行点乘计,得到聚合统计矩阵;

根据聚合统计矩阵进行聚合统计,所述聚合统计矩阵用于表征所述聚合统计维度的密态数据所属的分组。

2.根据权利要求1所述的全密态数据处理方法,其特征在于,所述分组维度所包含的分组维度数据进行密态排序之前,还包括:

对所述参与节点的数据分别进行加密,生成所述密态数据集。

3.根据权利要求1所述的全密态数据处理方法,其特征在于,以所述密态数据序列为基础,生成对比密态数据序列包括:

新建一个与所述密态数据序列大小一致的空白序列;

所述空白序列依次将所述密态数据序列的第2位至末尾的数据复制到所述空白序列的首位至倒数第二位中,以形成对比密态数据序列。

4.根据权利要求1所述的全密态数据处理方法,其特征在于,以所述密态数据序列为基础,生成对比密态数据序列还包括:

新建密态数据副本序列,复制所述密态数据序列内容至所述密态数据副本序列;

将所述密态数据副本序列的第2位至末尾的数据依次向所述密态数据副本序列首位方向前进一位替换原数据,以形成对比密态数据序列。

5.根据权利要求1所述的全密态数据处理方法,其特征在于,将所述密态数据序列与所述对比密态数据序列进行密态比较,以取得所述分组维度数据的密态比较结果序列包括:

将所述密态数据序列与所述对比密态数据序列通过密态比较算子进行密态比较,以获取密态布尔结果;

将所述密态布尔结果通过取反算子,以获得取反密态布尔结果;

将所述反密态布尔结果通过转换规则,以获取密态算数类型序列,所述转换规则为布尔值True转换为1,布尔值False转换为0;

将所述密态算数类型序列与所述密态数据序列进行点乘计算,获得所述密态比较结果序列。

6.根据权利要求1所述的全密态数据处理方法,其特征在于,将所述分组维度数据与所述类别序列进行一致性比较,以形成分类矩阵包括:

将所述分组维度数据与所述类别序列进行所述密态比较，以获得取密态布尔结果矩阵；

所述密态布尔结果矩阵通过转换规则，以获取分类矩阵，所述转换规则为布尔值True转换为1，布尔值False转换为0。

7.一种全密态数据处理装置，适用于多方安全计算中多个参与节点中的任一参与节点，其特征在于，所述全密态数据处理装置包括：

密态排序模块，用于根据密态数据集的分组维度，对所述分组维度中所包含的分组维度数据进行密态排序，生成密态数据序列，其中所述分组维度为所述密态数据集进行所述全密态数据处理时所涉及的至少一组数据序列中的所有密态数据；

镜像生成模块，用于以所述密态数据序列为基础，生成对比密态数据序列，所述对比密态数据序列为所述密态数据序列的镜像进行错位补位后的映射；

密态比较模块，用于将所述密态数据序列与所述对比密态数据序列进行密态比较，得到所述分组维度数据的密态比较结果序列；

密态过滤模块，用于将所述密态比较结果序列提取非零数据，以形成类别序列；

密态分组模块，用于将所述分组维度数据与所述类别序列进行一致性比较，以形成分类矩阵；

将聚合统计维度数据进行密态分类；其中，将聚合统计维度中的所有密态数据所形成的序列与分类矩阵进行点乘计，得到聚合统计矩阵；

根据聚合统计矩阵进行聚合统计，所述聚合统计矩阵用于表征所述聚合统计维度的密态数据所属的分组。

8.一种电子设备，包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序，其特征在于，所述处理器执行所述计算机程序时实现权利要求1至6任一项所述的全密态数据处理方法。

9.一种计算机可读存储介质，其上存储有计算机程序，其特征在于，所述计算机程序被处理器执行时实现权利要求1至6任一项所述的全密态数据处理方法。

全密态数据处理方法、装置、设备、介质

技术领域

[0001] 本公开涉及隐私计算技术领域，尤其涉及一种全密态数据处理方法、装置、设备、介质。

背景技术

[0002] 信息安全越来越得到各类机构重视，对实际业务中所传输的数据进行加密保护成为一种普遍认知。隐私计算正是应对在信息保护前提下进行数据价值流通的创新手段。在隐私计算过程中，各参与节点将数据加密成密文后对密文进行传输，并且在密文状态下执行具体的算子逻辑计算，以实现数据安全的目的。

[0003] 然而，在隐私计算过程中，尤其是涉及复杂的分组统计计算，比如groupby算子的时候，如何高效且安全地进行计算是业内面临的难点。目前，业内常见方案中，分组统计算子需要遍历全部数据，当处理海量数据时，会花费巨大计算成本。而且，分组统计算子进行分组时会将数据的比较结果进行明文恢复，然而明文在隐私计算中出现被视为一种安全隐患。综上所述，现有技术存在计算成本高且计算过程中数据存在安全隐患的缺陷。

发明内容

[0004] 本公开要解决的问题是为了克服现有技术中隐私计算过程中分组统计计算的计算成本高且计算过程中数据存在安全隐患的缺陷，提供一种全密态数据处理方法、装置、设备、介质。

[0005] 本公开是通过下述技术方案来解决上述技术问题：

[0006] 根据本公开的一方面，提供了一种全密态数据处理方法，适用于多方安全计算中多个参与节点中的任一参与节点，其特征在于，所述全密态数据处理方法包括：

[0007] 根据密态数据集的分组维度，对所述分组维度中所包含的分组维度数据进行密态排序，生成密态数据序列，其中所述分组维度为所述密态数据集进行所述全密态数据处理时所涉及的至少一组数据序列中的所有密态数据；

[0008] 以所述密态数据序列为基础，生成对比密态数据序列，所述对比密态数据序列为所述密态数据序列的镜像进行错位补位后的映射；

[0009] 将所述密态数据序列与所述对比密态数据序列进行密态比较，得到所述分组维度数据的密态比较结果序列。

[0010] 较佳地，所述得到所述分组维度数据的密态比较结果序列之后，还包括：

[0011] 将所述密态比较结果序列提取非零数据，以形成类别序列；

[0012] 将所述分组维度数据与所述类别序列进行一致性比较，以形成分类矩阵。

[0013] 较佳地，所述分组维度所包含的分组维度数据进行密态排序之前，还包括：

[0014] 对所述参与节点的数据分别进行加密，生成所述密态数据集。

[0015] 较佳地，以所述密态数据序列为基础，生成对比密态数据序列包括：

[0016] 新建一个与所述密态数据序列大小一致的空白序列；

[0017] 所述空白序列依次将所述密态数据序列的第2位至末尾的数据复制到所述空白序列的首位至倒数第二位中，以形成对比密态数据序列。

[0018] 较佳地，以所述密态数据序列为基础，生成对比密态数据序列还包括：

[0019] 新建密态数据副本序列，复制所述密态数据序列内容至所述密态数据副本序列；

[0020] 将所述密态数据副本序列的第2位至末尾的数据依次向所述密态数据副本序列首位方向前进一位替换原数据，以形成对比密态数据序列。

[0021] 较佳地，将所述密态数据序列与所述对比密态数据序列进行密态比较，以取得所述分组维度数据的密态比较结果序列包括：

[0022] 将所述密态数据序列与所述对比密态数据序列通过密态比较算子进行密态比较，以获取密态布尔结果；

[0023] 将所述密态布尔结果通过取反算子，以获得取反密态布尔结果；

[0024] 将所述反密态布尔结果通过转换规则，以获取密态算数类型序列，所述转换规则为布尔值True转换为1，布尔值False转换为0；

[0025] 将所述密态算数类型序列与所述密态数据序列进行点乘计算，获得所述密态比较结果序列。

[0026] 较佳地，将所述分组维度数据与所述类别序列进行一致性比较，以形成分类矩阵包括：

[0027] 将所述分组维度数据与所述类别序列进行所述密态比较，以获得取密态布尔结果矩阵；

[0028] 所述密态布尔结果矩阵通过转换规则，以获取分类矩阵，所述转换规则为布尔值True转换为1，布尔值False转换为0。

[0029] 根据本公开的另一方面，提供了一种全密态数据处理装置，包括：

[0030] 数据准备模块，用于对所述参与节点的数据分别进行加密，生成所述密态数据集；

[0031] 密态排序模块，用于根据密态数据集的分组维度，对所述分组维度中所包含的分组维度数据进行密态排序，生成密态数据序列，其中所述分组维度为所述密态数据集进行所述全密态数据处理时所涉及的至少一组数据序列中的所有密态数据；

[0032] 镜像生成模块，用于以所述密态数据序列为基础，生成对比密态数据序列，所述对比密态数据序列为所述密态数据序列的镜像进行错位补位后的映射；所述以所述密态数据序列为基础，生成对比密态数据序列包括：

[0033] 新建一个与所述密态数据序列大小一致的空白序列；

[0034] 所述空白序列依次将所述密态数据序列的第2位至末尾的数据复制到所述空白序列的首位至倒数第二位中，以形成对比密态数据序列；

[0035] 所述以所述密态数据序列为基础，生成对比密态数据序列还包括：

[0036] 新建密态数据副本序列，复制所述密态数据序列内容至所述密态数据副本序列；

[0037] 将所述密态数据副本序列的第2位至末尾的数据依次向所述密态数据副本序列首位方向前进一位替换原数据，以形成对比密态数据序列；

[0038] 密态比较模块，用于将所述密态数据序列与所述对比密态数据序列进行密态比较，得到所述分组维度数据的密态比较结果序列包括：将所述密态数据序列与所述对比密态数据序列通过密态比较算子进行密态比较，以获取密态布尔结果；

[0039] 将所述密态布尔结果通过取反算子,以获得取反密态布尔结果;

[0040] 将所述反密态布尔结果通过转换规则,以获取密态算数类型序列,所述转换规则为布尔值True转换为1,布尔值False转换为0;

[0041] 将所述密态算数类型序列与所述密态数据序列进行点乘计算,获得所述密态比较结果序列;

[0042] 密态过滤模块,用于将所述密态比较结果序列提取非零数据,以形成类别序列;

[0043] 密态分组模块,用于将所述分组维度数据与所述类别序列进行一致性比较,以形成分类矩阵包括:将所述分组维度数据与所述类别序列进行所述密态比较,以获得取密态布尔结果矩阵;

[0044] 所述密态布尔结果矩阵通过转换规则,以获取分类矩阵,所述转换规则为布尔值True转换为1,布尔值False转换为0。

[0045] 根据本公开的另一方面,提供了一种电子设备,包括存储器、处理器及存储在存储器上并可在处理器上运行的计算机程序,其特征在于,所述处理器执行所述计算机程序时实现上述任一项所述的全密态数据处理方法。

[0046] 根据本公开的另一方面,提供了一种计算机可读存储介质,其上存储有计算机程序,其特征在于,所述计算机程序被处理器执行时实现上述一项所述的全密态数据处理方法。

[0047] 本公开的积极进步效果在于:本公开的全密态数据处理方法实现了全密态数据处理,统计计算无需遍历全部数据以实现节约计算成本,而且隐私计算过程中数据全程密态以保证数据安全。

[0048] 应当理解,本部分所描述的内容并非旨在标识本公开的实施例的关键或重要特征,也不用于限制本公开的范围。本公开的其它特征将通过以下的说明书而变得容易理解。

附图说明

[0049] 附图用于更好地理解本方案,不构成对本公开的限定。其中:

[0050] 图1为本公开一示例性实施例提供的一种全密态数据处理方法的流程图;

[0051] 图2为本公开一示例性实施例提供的一种全密态数据处理装置的模块示意图;

[0052] 图3为本公开一示例性实施例提供的一种电子设备的结构示意图;

[0053] 图4为本公开一示例性实施例提供的一种全密态数据处理方法的比较算子原理示意图。

具体实施方式

[0054] 以下结合附图对本公开的示范性实施例做出说明,其中包括本公开实施例的各种细节以助于理解,应当将它们认为仅仅是示范性的。因此,本领域普通技术人员应当认识到,可以对这里描述的实施例做出各种改变和修改,而不会背离本公开的范围和精神。同样,为了清楚和简明,以下的描述中省略了对公知功能和结构的描述。

[0055] 本公开的技术方案适用于多方安全计算中多个参与节点中的任一参与节点。本公开一示例性实施例提供一种全密态数据处理方法,如图1所示,该方法包括以下步骤:

[0056] 步骤101、根据密态数据集的分组维度,对分组维度中所包含的分组维度数据进行

密态排序，生成密态数据序列，其中分组维度为密态数据集进行全密态数据处理时所涉及的至少一组数据序列中的所有密态数据。

[0057] 其中，分组维度所包含的分组维度数据进行密态排序之前，对参与节点的数据分别进行加密，生成密态数据集。

[0058] 在一个实施例中，多方安全计算中有两个参与节点A和B，分别持有数据a和数据b，如表1、表2所示。

[0059] 表1 参与节点A的数据a

[0060]

aid	site_id	count
1	1	45
2	2	100
3	1	230
4	5	70

[0061] 表2 参与节点B的数据b

[0062]

aid	site_id	count
5	1	35
6	2	10
7	5	205
8	4	13
9	2	21

[0063] 如表1、表2所示，数据a和数据b均含有字段aid、site_id、count。其中分组维度为site_id，聚合统计维度为count。

[0064] 在本实施例中，使用的加密手段是碎片化加密，比如数据x的碎片化加密表示为<x>。其中，碎片化的执行逻辑为：例如参与节点A与参与节点B，对于原始数据x，执行完碎片化后，将会被拆分为x-r与r两个分片数据，并且其中分片r发送给对端参与方持有，这个过程为秘密共享。执行完之后，原始数据x就以碎片态的形式存在，各方持有原始数据的其中一个分片，可以看成为x=[(节点A,x-r),(节点B,r)]。以<x>表示碎片态数据，该碎片态数据也就是密态数据。

[0065] 数据a和b的碎片化加密处理结果如表3、表4所示：

[0066] 表3 参与节点A的碎片化数据<a>

[0067]

aid	site_id	count
<1>	<1>	<45>
<2>	<2>	<100>
<3>	<1>	<230>
<4>	<5>	<70>

[0068] 表4 参与节点B的碎片化数据<b>

[0069]

aid	site_id	count
<5>	<1>	<35>
<6>	<2>	<10>
<7>	<5>	<205>
<8>	<4>	<13>
<9>	<2>	<21>

[0070] 各参与节点的数据执行完碎片化加密以后，执行密态拼接，两个参与节点的碎片化密态数据进行纵向拼接，得到密态数据集D，如表5所示：

[0071] 表5

[0072]

Aid	site_id	count
<1>	<1>	<45>
<2>	<2>	<100>
<3>	<1>	<230>
<4>	<5>	<70>
<5>	<1>	<35>
<6>	<2>	<10>
<7>	<5>	<205>
<8>	<4>	<13>
<9>	<2>	<21>

[0073] 由于密态数据集D的分组维度是site_id，将对密态数据集D中的分组维度site_id中的所有分组维度数据进行密态排序，生成密态数据序列。其中，密态排序通过执行多方安全计算的密态排序算子实现，密态排序算子包括：密态快速排序、密态双调排序、密态归并排序、密态选择排序、密态冒泡排序、密态堆排序。

[0074] 当执行密态排序算子时，分组维度site_id中的数据在加密状态下进行升序或降

序排序。同时，其他aid、count列所在的密态数据也跟随site_id中的数据进行排序。其中，密态数据的结果在没有恢复成明文之前，对于各参与节点来说都是未知的，不能推理出任何信息。例如，用密态排序算子对密态数据集D进行升序排列后，如表6所示：

[0075] 表6

[0076]

aid	site_id	count
〈1〉	〈1〉	〈45〉
〈3〉	〈1〉	〈230〉
〈5〉	〈1〉	〈35〉
〈2〉	〈2〉	〈100〉
〈6〉	〈2〉	〈10〉
〈9〉	〈2〉	〈21〉
〈8〉	〈4〉	〈13〉
〈4〉	〈5〉	〈70〉
〈7〉	〈5〉	〈205〉

[0077] 其中，分组维度site_id所包含的所有密态数据的序列为密态数据序列。

[0078] 步骤102、以密态数据序列为基础，生成对比密态数据序列，对比密态数据序列为密态数据序列的镜像进行错位补位后的映射。如表7所示：

[0079] 表7

[0080]

site_id（密态数据序列）	site_id（对比密态数据序列）
〈1〉	〈1〉
〈1〉	〈1〉
〈1〉	〈2〉
〈2〉	〈2〉
〈2〉	〈2〉
〈2〉	〈4〉
〈4〉	〈5〉
〈5〉	〈5〉
〈5〉	〈null〉

[0081] 在一个实施例中，可通过新建一个与密态数据序列大小一致的空白序列；再将空白序列依次将密态数据序列的第2位至末尾的数据复制到空白序列的首位至倒数第二位中，以形成对比密态数据序列。

[0082] 在一个实施例中，也可新建密态数据副本序列，复制密态数据序列内容至密态数据副本序列；再将密态数据副本序列的第2位至末尾的数据依次向密态数据副本序列首位方向前进一位替换原数据，以形成对比密态数据序列。

[0083] 步骤103、将密态数据序列与对比密态数据序列进行密态比较，得到分组维度数据的密态比较结果序列。如表8所示：

[0084] 表8

[0085]

site_id（密态数据序列）	site_id（对比密态数据序列）	密态比较结果	密态比较结果取反	密态比较结果取反转换成密态算数类型（B2A）	B2A与密态数据序列乘积
<1>	<1>	<True>	<False>	<0>	<0>
<1>	<1>	<True>	<False>	<0>	<0>
<1>	<2>	<False>	<True>	<1>	<1>
<2>	<2>	<True>	<False>	<0>	<0>
<2>	<2>	<True>	<False>	<0>	<0>
<2>	<4>	<False>	<True>	<1>	<2>
<4>	<5>	<False>	<True>	<1>	<4>
<5>	<5>	<True>	<False>	<0>	<0>
<5>	<null>	<False>	<True>	<1>	<5>

[0086] 将密态数据序列与对比密态数据序列通过密态比较算子进行密态比较，以获取密态布尔结果。其中，密态比较结果序列使用布尔值，布尔值包括True和False。由于是密态比较，将布尔值表述为<True>和<False>以表示为加密状态。

[0087] 如图4所示为密态比较算子的原理示意图。首先，P1，P2分别拥有明文x和y，并将x碎片化为$[x]_1^A$、$[x]_2^A$，y碎片化为$[y]_1^A$、$[y]_2^A$，秘密共享其中部分碎片，此时P1拥有$([x]_1^A,\ [y]_1^A)$，P2拥有$([x]_2^A,\ [y]_2^A)$。

[0088] 其次，P1计算碎片$[z]_1^A = [x]_1^A - [y]_1^A$，P2计算碎片$[z]_2^A = [x]_2^A - [y]_2^A$。

[0089] 然后创建布尔零碎片$([a]_1^B,\ [a]_2^B)$和算数零碎片$([b]_1^A,\ [b]_2^A)$。此时P1拥有$[a]_1^B$，$[b]_1^A$，P2拥有$[a]_2^B$，$[b]_2^A$。创建零碎片的主要目的是为了进入加法电路，并保持其值不变。

[0090] 接下来P1计算$[op_1]_1^B = ([z]_1^A + [b]_1^A) \oplus [a]_1^B$，$[op_2]_1^B = 0$，P2计算$[op_1]_2^B = [a]_2^B$，$[op_2]_2^B = [z]_2^A + [b]_2^A$。并通过加法电路计算$OP = PPA(op_1,\ op_2)$。

[0091] 最后通过计算OP&Mask得到最后的符号位bit，其中$Mask = 0x1 \ll nbit$（nbit是固定位数，一般有32位，64位）。

[0092] 将密态布尔结果通过取反算子，以获得取反密态布尔结果。其中<True>和<False>互为取反值。

[0093] 将反密态布尔结果通过转换规则，以获取密态算数类型序列，转换规则为布尔值<True>转换为<1>，布尔值<False>转换为<0>。其中，转换规则通过多方安全计算的B2A算子实现，B2A算子将密态的布尔值<False>转为密态算术碎片结果<0>，密态的布尔值<True>转为密态算术碎片态结果<1>。

[0094] 将密态算数类型序列与密态数据序列进行点乘计算，获得所述密态比较结果序列。如表9所示：

[0095] 表9

[0096]

site_id（密态数据序列）	密态算数类型序列	密态比较结果序列
<1>	<0>	<0>
<1>	<0>	<0>
<1>	<1>	<1>
<2>	<0>	<0>
<2>	<0>	<0>
<2>	<1>	<2>
<4>	<1>	<4>
<5>	<0>	<0>
<5>	<1>	<5>

[0097] 然后，进行密态排序，将密态比较结果序列提取非零数据，以形成类别序列。例如执行升序密态排序，如表10所示：

[0098] 表10

[0099]

site_id（密态数据序列）	密态算数类型序列	密态比较结果序列
〈1〉	〈1〉	〈1〉
〈2〉	〈1〉	〈2〉
〈4〉	〈1〉	〈4〉
〈5〉	〈1〉	〈5〉
〈1〉	〈0〉	〈0〉
〈1〉	〈0〉	〈0〉
〈2〉	〈0〉	〈0〉
〈2〉	〈0〉	〈0〉
〈5〉	〈0〉	〈0〉

[0100] 提取非零数据后，如表11所示：

[0101] 表11

[0102]

site_id（密态数据序列）	转换成密态算数类型	与乘积
〈1〉	〈1〉	〈1〉
〈2〉	〈1〉	〈2〉
〈4〉	〈1〉	〈4〉
〈5〉	〈1〉	〈5〉

[0103] 类别序列如表12所示：

[0104] 表12

[0105]

类别序列
<1>
<2>
<4>
<5>

[0106] 其中类别序列，表示为数据可根据类别序列的数据进行分类，本实施例中类别序列包含<1>、<2>，<4>，<5>，则说明数据总共有四类即枚举数为4，类别分别为<1>、<2>，<4>，<5>，即枚举值分别为<1>、<2>，<4>，<5>。

[0107] 将分组维度数据与类别序列进行一致性比较，以形成分类矩阵。首先，将分组维度数据与类别序列进行密态比较，以获得取密态布尔结果矩阵，如表13所示：

[0108] 表13

[0109]

密态分组	第一分组	第二分组	第三分组	第四分组
密态比较	<1>	<2>	<4>	<5>
<1>	<True>	<False>	<False>	<False>
<2>	<False>	<True>	<False>	<False>
<1>	<True>	<False>	<False>	<False>
<5>	<False>	<False>	<False>	<True>
<1>	<True>	<False>	<False>	<False>
<2>	<False>	<True>	<False>	<False>
<5>	<False>	<False>	<False>	<True>
<4>	<False>	<False>	<True>	<False>
<2>	<False>	<True>	<False>	<False>

[0110] 将密态布尔结果通过转换成密态算数类型序列，转换规则为布尔值<True>转换为<1>，布尔值<False>转换为<0>。其中，转换规则通过多方安全计算的B2A算子实现。分类矩阵，如表14所示：

[0111] 表14

[0112]

密态分组	第一分组	第二分组	第三分组	第四分组
密态比较	〈1〉	〈2〉	〈4〉	〈5〉
〈1〉	〈1〉	〈0〉	〈0〉	〈0〉
〈2〉	〈0〉	〈1〉	〈0〉	〈0〉
〈1〉	〈1〉	〈0〉	〈0〉	〈0〉
〈5〉	〈0〉	〈0〉	〈0〉	〈1〉
〈1〉	〈1〉	〈0〉	〈0〉	〈0〉
〈2〉	〈0〉	〈1〉	〈0〉	〈0〉
〈5〉	〈0〉	〈0〉	〈0〉	〈1〉
〈4〉	〈0〉	〈0〉	〈1〉	〈0〉
〈2〉	〈0〉	〈1〉	〈0〉	〈0〉

[0113] 得到分类矩阵，将聚合统计维度数据进行密态分类。其中，聚合统计维度为count，其中的所有密态数据所形成的序列与分类矩阵进行点乘计，得到聚合统计矩阵。如表15所示：

[0114] 表15

[0115]

分类矩阵				count	聚合统计矩阵			
第一分组	第二分组	第三分组	第四分组		第一分组	第二分组	第三分组	第四分组
〈1〉	〈0〉	〈0〉	〈0〉	〈45〉	〈45〉	〈0〉	〈0〉	〈0〉
〈0〉	〈1〉	〈0〉	〈0〉	〈100〉	〈0〉	〈100〉	〈0〉	〈0〉
〈1〉	〈0〉	〈0〉	〈0〉	〈230〉	〈230〉	〈0〉	〈0〉	〈0〉
〈0〉	〈0〉	〈0〉	〈1〉	〈70〉	〈0〉	〈0〉	〈0〉	〈70〉
〈1〉	〈0〉	〈0〉	〈0〉	〈35〉	〈35〉	〈0〉	〈0〉	〈0〉
〈0〉	〈1〉	〈0〉	〈0〉	〈10〉	〈0〉	〈10〉	〈0〉	〈0〉
〈0〉	〈0〉	〈0〉	〈1〉	〈205〉	〈0〉	〈0〉	〈0〉	〈205〉
〈0〉	〈0〉	〈1〉	〈0〉	〈13〉	〈0〉	〈0〉	〈13〉	〈0〉
〈0〉	〈1〉	〈0〉	〈0〉	〈21〉	〈0〉	〈21〉	〈0〉	〈0〉

[0116] 根据聚合统计矩阵，可以看出聚合统计维度的密态数据所属的分组，进而可进行聚合统计。比如，进行求和统计如表16所示：

[0117] 表16

[0118]

	聚合统计矩阵			
分组	第一分组	第二分组	第三分组	第四分组
	〈45〉	〈0〉	〈0〉	〈0〉
	〈0〉	〈100〉	〈0〉	〈0〉
	〈230〉	〈0〉	〈0〉	〈0〉
	〈0〉	〈0〉	〈0〉	〈70〉
	〈35〉	〈0〉	〈0〉	〈0〉
	〈0〉	〈10〉	〈0〉	〈0〉
	〈0〉	〈0〉	〈0〉	〈205〉
	〈0〉	〈0〉	〈13〉	〈0〉
	〈0〉	〈21〉	〈0〉	〈0〉
sum	310	131	13	275

[0119] 其中，聚合统计不局限sum，可以是任意算子，包括：求各分组平均avg、方差var、最小值min、最大值max。通过上述步骤实现了，对数据的密态分组，并且可以根据密态分组的结果进行相应的聚合计算。

[0120] 图2为本公开一示例性实施例提供的一种全密态数据处理装置的模块示意图，该全密态数据处理装置包括：

[0121] 数据准备模块，用于对所述参与节点的数据分别进行加密，生成所述密态数据集；

[0122] 密态排序模块，用于根据密态数据集的分组维度，对所述分组维度中所包含的分组维度数据进行密态排序，生成密态数据序列，其中所述分组维度为所述密态数据集进行所述全密态数据处理时所涉及的至少一组数据序列中的所有密态数据；

[0123] 镜像生成模块，用于以所述密态数据序列为基础，生成对比密态数据序列，所述对比密态数据序列为所述密态数据序列的镜像进行错位补位后的映射；

[0124] 可选地，镜像生成模块用于新建一个与所述密态数据序列大小一致的空白序列；

[0125] 所述空白序列依次将所述密态数据序列的第2位至末尾的数据复制到所述空白序列的首位至倒数第二位中，以形成对比密态数据序列；

[0126] 可选地，镜像生成模块还用于新建密态数据副本序列，复制所述密态数据序列内容至所述密态数据副本序列；

[0127] 将所述密态数据副本序列的第2位至末尾的数据依次向所述密态数据副本序列首位方向前进一位替换原数据，以形成对比密态数据序列；

[0128] 密态比较模块，用于将所述密态数据序列与所述对比密态数据序列进行密态比较，得到所述分组维度数据的密态比较结果序列；

[0129] 可选地，密态比较模块，用于将所述密态数据序列与所述对比密态数据序列通过密态比较算子进行密态比较，以获取密态布尔结果；

[0130] 将所述密态布尔结果通过取反算子，以获得取反密态布尔结果；

[0131] 将所述反密态布尔结果通过转换规则，以获取密态算数类型序列，所述转换规则为布尔值True转换为1，布尔值False转换为0；

[0132] 将所述密态算数类型序列与所述密态数据序列进行点乘计算，获得所述密态比较结果序列；

[0133] 密态过滤模块，用于将所述密态比较结果序列提取非零数据，以形成类别序列；

[0134] 密态分组模块，用于将所述分组维度数据与所述类别序列进行一致性比较，以形成分类矩阵；

[0135] 可选地，密态分组模块，用于将所述分组维度数据与所述类别序列进行所述密态比较，以获得取密态布尔结果矩阵；

[0136] 所述密态布尔结果矩阵通过转换规则，以获取分类矩阵，所述转换规则为布尔值True转换为1，布尔值False转换为0。

[0137] 根据本公开的实施例，本公开还提供了一种电子设备、一种可读存储介质和一种计算机程序产品。

[0138] 图3示出了可以用来实施本公开的实施例的示例电子设备800的示意性框图。电子设备旨在表示各种形式的数字计算机，诸如，膝上型计算机、台式计算机、工作台、个人数字助理、服务器、刀片式服务器、大型计算机、和其它适合的计算机。电子设备还可以表示各种形式的移动装置，诸如，个人数字处理、蜂窝电话、智能电话、可穿戴设备和其它类似的计算装置。本文所示的部件、它们的连接和关系、以及它们的功能仅仅作为示例，并且不意在限制本文中描述的和/或者要求的本公开的实现。

[0139] 如图3所示，设备800包括计算单元801，其可以根据存储在只读存储器(ROM)802中的计算机程序或者从存储单元808加载到随机访问存储器(RAM)803中的计算机程序，来执行各种适当的动作和处理。在RAM 803中，还可存储设备800操作所需的各种程序和数据。计算单元801、ROM 802以及RAM 803通过总线804彼此相连。输入/输出(I/O)接口805也连接至总线804。

[0140] 设备800中的多个部件连接至I/O接口805，包括：输入单元806，例如键盘、鼠标等；输出单元807，例如各种类型的显示器、扬声器等；存储单元808，例如磁盘、光盘等；以及通信单元809，例如网卡、调制解调器、无线通信收发机等。通信单元809允许设备800通过诸如因特网的计算机网络和/或各种电信网络与其他设备交换信息/数据。

[0141] 计算单元801可以是各种具有处理和计算能力的通用和/或专用处理组件。计算单元801的一些示例包括但不限于中央处理单元(CPU)、图形处理单元(GPU)、各种专用的人工智能(AI)计算芯片、各种运行机器学习模型算法的计算单元、数字信号处理器(DSP)、以及任何适当的处理器、控制器、微控制器等。计算单元801执行上文所描述的各个方法和处理，例如全密态数据处理方法。例如，在一些实施例中，全密态数据处理方法可被实现为计算机软件程序，其被有形地包含于机器可读介质，例如存储单元808。在一些实施例中，计算机程

序的部分或者全部可以经由ROM 802和/或通信单元809而被载入和/或安装到设备800上。当计算机程序加载到RAM 803并由计算单元801执行时，可以执行上文描述的全密态数据处理方法的一个或多个步骤。备选地，在其他实施例中，计算单元801可以通过其他任何适当的方式(例如，借助于固件)而被配置为执行全密态数据处理方法。

[0142] 本文中以上描述的系统和技术的各种实施方式可以在数字电子电路系统、集成电路系统、场可编程门阵列(FPGA)、专用集成电路(ASIC)、专用标准产品(ASSP)、芯片上系统的系统(SOC)、复杂可编程逻辑设备(CPLD)、计算机硬件、固件、软件、和/或它们的组合中实现。这些各种实施方式可以包括：实施在一个或者多个计算机程序中，该一个或者多个计算机程序可在包括至少一个可编程处理器的可编程系统上执行和/或解释，该可编程处理器可以是专用或者通用可编程处理器，可以从存储系统、至少一个输入装置、和至少一个输出装置接收数据和指令，并且将数据和指令传输至该存储系统、该至少一个输入装置、和该至少一个输出装置。

[0143] 用于实施本公开的方法的程序代码可以采用一个或多个编程语言的任何组合来编写。这些程序代码可以提供给通用计算机、专用计算机或其他可编程数据处理装置的处理器或控制器，使得程序代码当由处理器或控制器执行时使流程图和/或框图中所规定的功能/操作被实施。程序代码可以完全在机器上执行、部分地在机器上执行，作为独立软件包部分地在机器上执行且部分地在远程机器上执行或完全在远程机器或服务器上执行。

[0144] 在本公开的上下文中，机器可读介质可以是有形的介质，其可以包含或存储以供指令执行系统、装置或设备使用或与指令执行系统、装置或设备结合地使用的程序。机器可读介质可以是机器可读信号介质或机器可读储存介质。机器可读介质可以包括但不限于电子的、磁性的、光学的、电磁的、红外的、或半导体系统、装置或设备，或者上述内容的任何合适组合。机器可读存储介质的更具体示例会包括基于一个或多个线的电气连接、便携式计算机盘、硬盘、随机存取存储器(RAM)、只读存储器(ROM)、可擦除可编程只读存储器(EPROM或快闪存储器)、光纤、便捷式紧凑盘只读存储器(CD-ROM)、光学储存设备、磁储存设备、或上述内容的任何合适组合。

[0145] 为了提供与用户的交互，可以在计算机上实施此处描述的系统和技术，该计算机具有：用于向用户显示信息的显示装置(例如，CRT(阴极射线管)或者LCD(液晶显示器)监视器)；以及键盘和指向装置(例如，鼠标或者轨迹球)，用户可以通过该键盘和该指向装置来将输入提供给计算机。其它种类的装置还可以用于提供与用户的交互；例如，提供给用户的反馈可以是任何形式的传感反馈(例如，视觉反馈、听觉反馈、或者触觉反馈)；并且可以用任何形式(包括声输入、语音输入或者、触觉输入)来接收来自用户的输入。

[0146] 可以将此处描述的系统和技术实施在包括后台部件的计算系统(例如，作为数据服务器)、或者包括中间件部件的计算系统(例如，应用服务器)、或者包括前端部件的计算系统(例如，具有图形用户界面或者网络浏览器的用户计算机，用户可以通过该图形用户界面或者该网络浏览器来与此处描述的系统和技术的实施方式交互)、或者包括这种后台部件、中间件部件、或者前端部件的任何组合的计算系统中。可以通过任何形式或者介质的数字数据通信(例如，通信网络)来将系统的部件相互连接。通信网络的示例包括：局域网(LAN)、广域网(WAN)和互联网。

[0147] 计算机系统可以包括客户端和服务器。客户端和服务器一般远离彼此并且通常通

过通信网络进行交互。通过在相应的计算机上运行并且彼此具有客户端-服务器关系的计算机程序来产生客户端和服务器的关系。服务器可以是云服务器，也可以为分布式系统的服务器，或者是结合了区块链的服务器。

[0148] 应该理解，可以使用上面所示的各种形式的流程，重新排序、增加或删除步骤。例如，本发公开中记载的各步骤可以并行地执行也可以顺序地执行也可以不同的次序执行，只要能够实现本公开公开的技术方案所期望的结果，本文在此不进行限制。

[0149] 上述具体实施方式，并不构成对本公开保护范围的限制。本领域技术人员应该明白的是，根据设计要求和其他因素，可以进行各种修改、组合、子组合和替代。任何在本公开的精神和原则之内所作的修改、等同替换和改进等，均应包含在本公开保护范围之内。

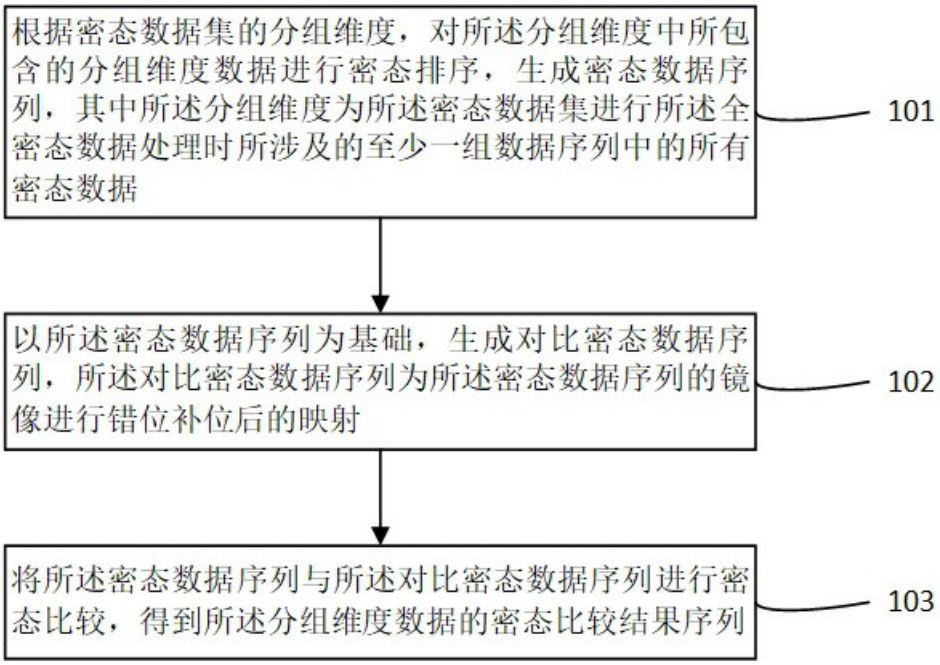

根据密态数据集的分组维度，对所述分组维度中所包含的分组维度数据进行密态排序，生成密态数据序列，其中所述分组维度为所述密态数据集进行所述全密态数据处理时所涉及的至少一组数据序列中的所有密态数据
101
以所述密态数据序列为基础，生成对比密态数据序列，所述对比密态数据序列为所述密态数据序列的镜像进行错位补位后的映射
102
将所述密态数据序列与所述对比密态数据序列进行密态比较，得到所述分组维度数据的密态比较结果序列
103

图1

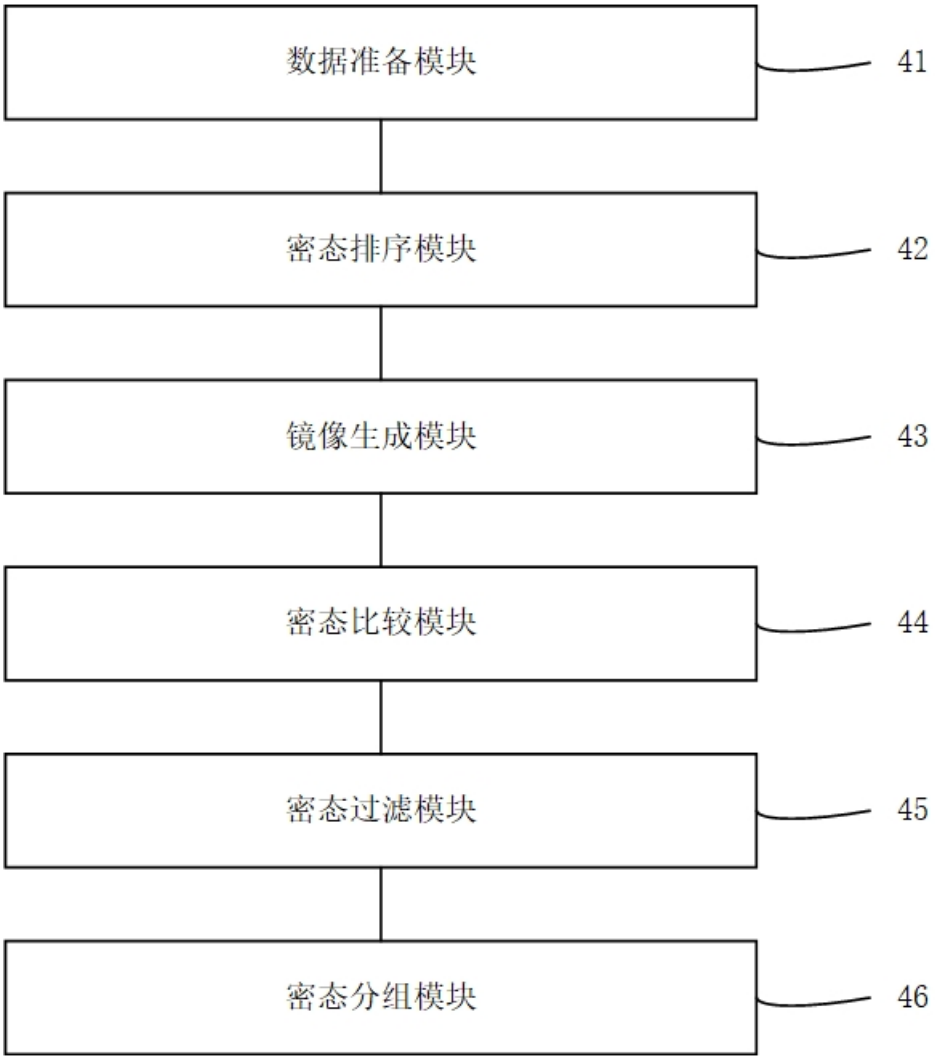


图2

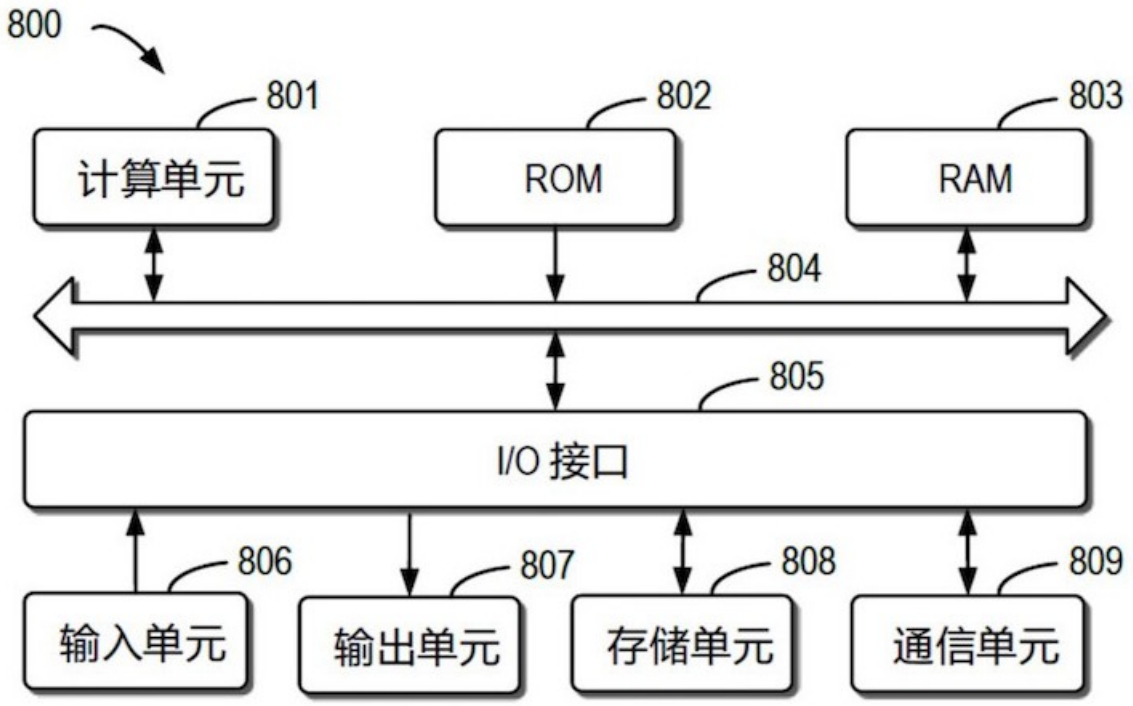

800
801
计算单元
802
ROM
803
RAM
804
805
I/O 接口
806
输入单元
807
输出单元
808
存储单元
809
通信单元

图3

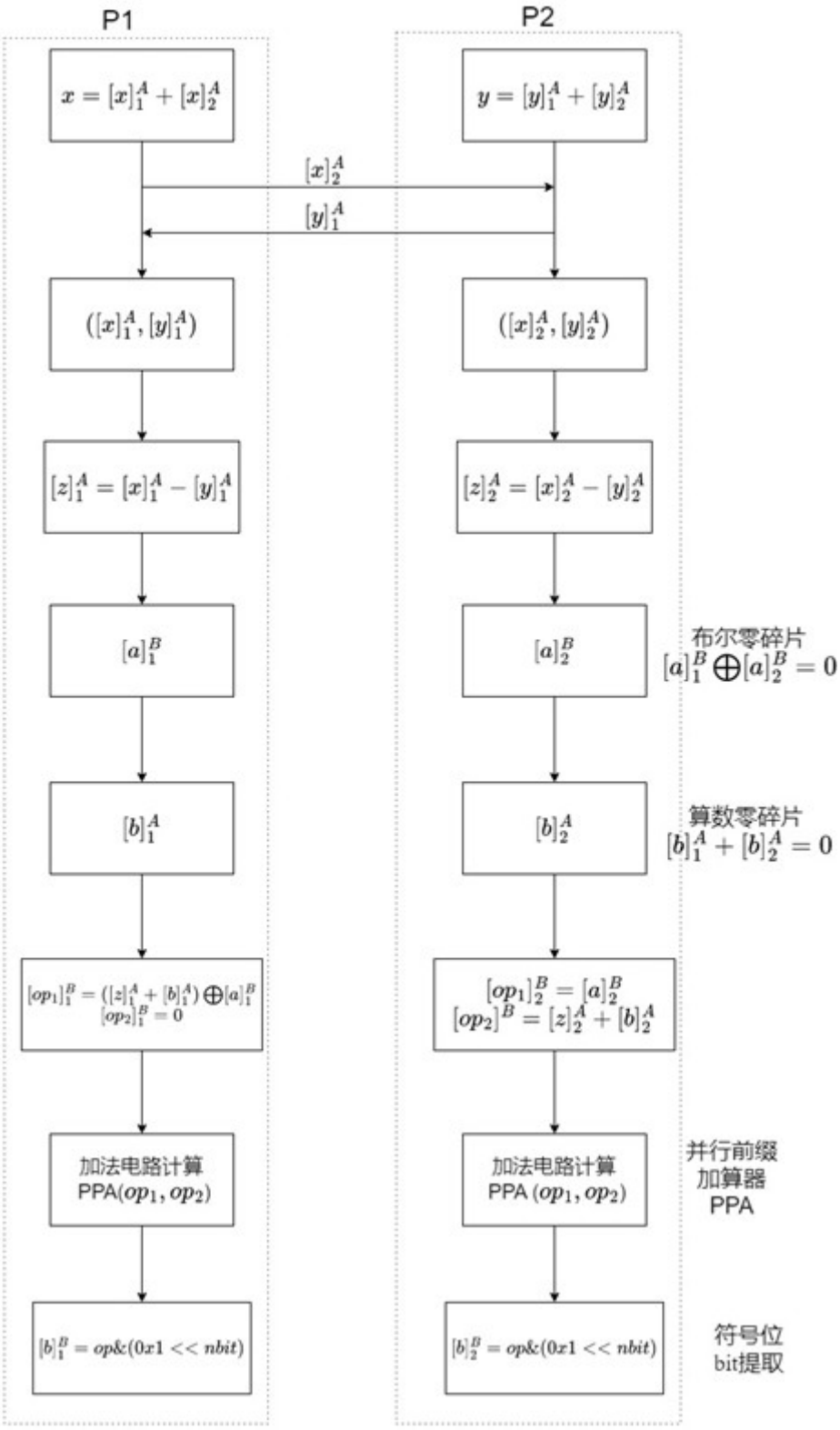

P1
P2
$x = [x]_1^A + [x]_2^A$
$y = [y]_1^A + [y]_2^A$
$[x]_2^A$
$[y]_1^A$
$([x]_1^A, [y]_1^A)$
$([x]_2^A, [y]_2^A)$
$[z]_1^A = [x]_1^A - [y]_1^A$
$[z]_2^A = [x]_2^A - [y]_2^A$
$[a]_1^B$
$[a]_2^B$
布尔零碎片
$[a]_1^B \bigoplus [a]_2^B = 0$
$[b]_1^A$
$[b]_2^A$
算数零碎片
$[b]_1^A + [b]_2^A = 0$
$[op_1]_1^B = ([z]_1^A + [b]_1^A) \bigoplus [a]_1^B$
$[op_2]_1^B = 0$
$[op_1]_2^B = [a]_2^B$
$[op_2]^B = [z]_2^A + [b]_2^A$
加法电路计算
PPA(op_1, op_2)
加法电路计算
PPA (op_1, op_2)
并行前缀
加算器
PPA
$[b]_1^B = op\&(0x1 << nbit)$
$[b]_2^B = op\&(0x1 << nbit)$
符号位
bit提取

图4