(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 116319084 B
(45)授权公告日 2023.09.29

(21)申请号 202310552246.9

(22)申请日 2023.05.17

(65)同一申请的已公布的文献号
申请公布号 CN 116319084 A

(43)申请公布日 2023.06.23

(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 涂志鹏 张伟奇

(74)专利代理机构 北京超凡宏宇知识产权代理有限公司 11463
专利代理师 唐正瑜

(51)Int.Cl.
H04L 9/40(2022.01)
G06F 21/62(2013.01)

(56)对比文件
CN 106507133 A,2017.03.15
CN 110033348 A,2019.07.19
CN 113377850 A,2021.09.10
CN 115941168 A,2023.04.07
US 2014257919 A1,2014.09.11
彭加飞,金三九,沈毅.利用计算机进行区组随机分组的一种新方法及其应用.中国卫生统计.2003,(第03期),
彭加飞,金三九,沈毅.利用计算机进行区组随机分组的一种新方法及其应用.中国卫生统计.2003,(第03期),全文.

审查员 冉建国

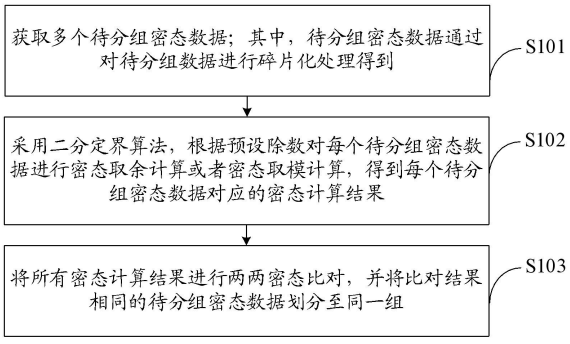
权利要求书2页 说明书12页 附图3页

(54)发明名称

一种随机分组的方法及装置、计算机程序产品、电子设备

(57)摘要

本申请提供一种随机分组的方法及装置、计算机程序产品、电子设备，其中，方法包括：获取多个待分组密态数据；其中，待分组密态数据通过对待分组数据进行碎片化处理得到；根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算，得到每个待分组密态数据对应的密态计算结果；将所有密态计算结果进行两两密态比对，并将比对结果相同的待分组密态数据划分至同一组。通过在保护原始的待分组数据不暴露的前提下，以碎片态的形式进行多方安全计算，从而有效的保证了数据的隐私性以及安全性。同时，通过二分定界算法对待分组密态数据进行密态取余计算或者密态取模计算，能够得到准确度较高的计算结果，进一步能够提高随机分组的效果。



CN 116319084 B

1.一种随机分组的方法,其特征在于,包括:

获取多个待分组密态数据;其中,所述待分组密态数据通过对待分组数据进行碎片化处理得到;

采用二分定界算法,根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算,得到每个待分组密态数据对应的密态计算结果;

将所有密态计算结果进行两两密态比对,并将比对结果相同的待分组密态数据划分至同一组;

所述采用二分定界算法,根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算,得到每个待分组密态数据对应的密态计算结果,包括:

针对一个待分组密态数据,获取初始的密态左边界、初始的密态右边界以及该待分组密态数据与所述预设除数之间的密态除法结果;

从所述初始的密态左边界以及所述初始的密态右边界开始,循环执行二分定界流程,直至满足循环结束条件,并将循环结束后得到的密态左边界确定为所述密态计算结果;

其中,所述二分定界流程包括:

根据当前的密态左边界以及当前的密态右边界计算密态中间值;

根据所述密态中间值与所述密态除法结果之间的大小关系,对所述当前的密态左边界以及所述当前的密态右边界进行更新,得到新的密态左边界以及新的密态右边界。

2.根据权利要求1所述的随机分组的方法,其特征在于,所述根据当前的密态左边界以及当前的密态右边界计算密态中间值,包括:

根据如下公式计算所述密态中间值:

$$mid = left + (right - left)/2;$$

其中,mid为所述密态中间值,$left$为所述当前的密态左边界,$right$为所述当前的密态右边界。

3.根据权利要求1所述的随机分组的方法,其特征在于,所述根据所述密态中间值与所述密态除法结果之间的大小关系,对所述当前的密态左边界以及所述当前的密态右边界进行更新,得到新的密态左边界以及新的密态右边界,包括:

当所述密态中间值大于所述密态除法结果时,将所述当前的密态右边界更新为所述密态中间值;或者,

当所述密态中间值小于所述密态除法结果时,将所述当前的密态左边界更新为所述密态中间值。

4.根据权利要求1所述的随机分组的方法,其特征在于,所述根据所述密态中间值与所述密态除法结果之间的大小关系,对所述当前的密态左边界以及所述当前的密态右边界进行更新,得到新的密态左边界以及新的密态右边界,包括:

计算所述密态除法结果与所述密态中间值的差值,得到对应的密态符号位;其中,所述差值大于0时所述密态符号位为0,否则所述密态符号位为1;

根据如下公式对所述当前的密态左边界以及所述当前的密态右边界进行更新:

$$right' = msb\_mid \times mid + (1 - msb\_mid) \times right;$$

$$left' = (1 - msb\_mid) \times mid + msb\_mid \times left;$$

其中，$right'$为所述新的密态右边界，msb_mid为所述密态符号位，mid为所述密态中间值，$right$为所述当前的密态右边界，$left'$为所述新的密态左边界，$left$为所述当前的密态左边界。

5.根据权利要求1-4任一项所述的随机分组的方法，其特征在于，所述预设除数的大小与分组组数的大小相等。

6.一种随机分组的装置，其特征在于，包括：

获取模块，用于获取多个待分组密态数据；其中，所述待分组密态数据通过对待分组数据进行碎片化处理得到；

计算模块，用于根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算，得到每个待分组密态数据对应的密态计算结果；

分组模块，用于将所有密态计算结果进行两两密态比对，并将比对结果相同的待分组密态数据划分至同一组；

所述计算模块具体用于：

针对一个待分组密态数据，获取初始的密态左边界、初始的密态右边界以及该待分组密态数据与所述预设除数之间的密态除法结果；

从所述初始的密态左边界以及所述初始的密态右边界开始，循环执行二分定界流程，直至满足循环结束条件，并将循环结束后得到的密态左边界确定为所述密态计算结果；

其中，所述二分定界流程包括：

根据当前的密态左边界以及当前的密态右边界计算密态中间值；

根据所述密态中间值与所述密态除法结果之间的大小关系，对所述当前的密态左边界以及所述当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界。

7.一种电子设备，其特征在于，包括：处理器、存储器和总线；

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计算机程序指令，所述处理器调用所述计算机程序指令能够执行如权利要求1-5任一项所述的随机分组的方法。

8.一种计算机可读存储介质，其特征在于，所述计算机可读存储介质存储计算机程序指令，所述计算机程序指令被计算机运行时，使所述计算机执行如权利要求1-5任一项所述的随机分组的方法。

一种随机分组的方法及装置、计算机程序产品、电子设备

技术领域

[0001] 本申请涉及多方安全计算技术领域，具体而言，涉及一种随机分组的方法及装置、计算机程序产品、电子设备。

背景技术

[0002] 随着企业对数据保护意识的增强，多方安全计算作为一种有效的隐私保护技术逐步在企业业务中得到应用。在多方安全计算中，算子是最底层、最基础、最重要的计算单元，复杂的统计以及机器学习都需要建立在算子的基础之上。其中，在随机分组的场景中，取余计算以及取模计算是其中一种非常重要的算子之一，例如：在广告推荐、活动策略上线、负载均衡、AB实验等场景中均会涉及到取余或者取模计算。

[0003] 但是，在现有技术中，在保护数据安全的前提下进行随机分组，由于取余或者取模计算的准确度较低，会导致随机分组的效果较差。

发明内容

[0004] 本申请实施例的目的在于提供一种随机分组的方法及装置、计算机程序产品、电子设备，用以解决现有技术中在保护数据安全的前提下进行随机分组，由于取余或者取模计算的准确度较低，会导致随机分组的效果较差的技术问题。

[0005] 第一方面，本申请实施例提供一种随机分组的方法，包括：获取多个待分组密态数据；其中，所述待分组密态数据通过对待分组数据进行碎片化处理得到；采用二分定界算法，根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算，得到每个待分组密态数据对应的密态计算结果；将所有密态计算结果进行两两密态比对，并将比对结果相同的待分组密态数据划分至同一组。

[0006] 在上述方案中，通过在保护原始的待分组数据不暴露的前提下，以碎片态的形式进行多方安全计算，从而有效的保证了数据的隐私性以及安全性。同时，通过二分定界算法对待分组密态数据进行密态取余计算或者密态取模计算，能够得到准确度较高的计算结果，进一步能够提高随机分组的效果。

[0007] 在可选的实施方式中，所述采用二分定界算法，根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算，得到每个待分组密态数据对应的密态计算结果，包括：针对一个待分组密态数据，获取初始的密态左边界、初始的密态右边界以及该待分组密态数据与所述预设除数之间的密态除法结果；从所述初始的密态左边界以及所述初始的密态右边界开始，循环执行二分定界流程，直至满足循环结束条件，并将循环结束后得到的密态左边界确定为所述密态计算结果；其中，所述二分定界流程包括：根据当前的密态左边界以及当前的密态右边界计算密态中间值；根据所述密态中间值与所述密态除法结果之间的大小关系，对所述当前的密态左边界以及所述当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界。

[0008] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算

对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0009] 在可选的实施方式中，所述根据当前的密态左边界以及当前的密态右边界计算密态中间值，包括：根据如下公式计算所述密态中间值：

[0010] $$mid = left + (right - left)/2;$$

[0011] 其中，mid为所述密态中间值，$left$为所述当前的密态左边界，$right$为所述当前的密态右边界。

[0012] 在上述方案中，在二分定界算法中，可以通过确定密态中间值以对密态左边界以及密态右边界进行更新，从而可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0013] 在可选的实施方式中，所述根据所述密态中间值与所述密态除法结果之间的大小关系，对所述当前的密态左边界以及所述当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界，包括：当所述密态中间值大于所述密态除法结果时，将所述当前的密态右边界更新为所述密态中间值；或者，当所述密态中间值小于所述密态除法结果时，将所述当前的密态左边界更新为所述密态中间值。

[0014] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0015] 在可选的实施方式中，所述根据所述密态中间值与所述密态除法结果之间的大小关系，对所述当前的密态左边界以及所述当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界，包括：计算所述密态除法结果与所述密态中间值的差值，得到对应的密态符号位；其中，所述差值大于0时所述密态符号位为0，否则所述密态符号位为1；根据如下公式对所述当前的密态左边界以及所述当前的密态右边界进行更新：

[0016] $$right' = msb\_mid \times mid + (1 - msb\_mid) \times right;$$

[0017] $$left' = (1 - msb\_mid) \times mid + msb\_mid \times left;$$

[0018] 其中，$right'$为所述新的密态右边界，msb_mid为所述密态符号位，mid为所述密态中间值，$right$为所述当前的密态右边界，$left'$为所述新的密态左边界，$left$为所述当前的密态左边界。

[0019] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0020] 在可选的实施方式中，所述预设除数的大小与分组组数的大小相等。

[0021] 在上述方案中，在密态取余计算或者密态取模计算的过程中，其预设除数的大小可以与分组组数的大小，这样，经过对密态取余计算或者密态取模计算的计算结果进行比对后，可以直接将待分组密态数据划分为分组组数对应的组，从而可以提高分组的效率。

[0022] 第二方面，本申请实施例提供一种随机分组的装置，包括：获取模块，用于获取多

个待分组密态数据;其中,所述待分组密态数据通过对待分组数据进行碎片化处理得到;计算模块,用于采用二分定界算法,根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算,得到每个待分组密态数据对应的密态计算结果;分组模块,用于将所有密态计算结果进行两两密态比对,并将比对结果相同的待分组密态数据划分至同一组。

[0023] 在上述方案中,通过在保护原始的待分组数据不暴露的前提下,以碎片态的形式进行多方安全计算,从而有效的保证了数据的隐私性以及安全性。同时,通过二分定界算法对待分组密态数据进行密态取余计算或者密态取模计算,能够得到准确度较高的计算结果,进一步能够提高随机分组的效果。

[0024] 在可选的实施方式中,所述计算模块具体用于:针对一个待分组密态数据,获取初始的密态左边界、初始的密态右边界以及该待分组密态数据与所述预设除数之间的密态除法结果;从所述初始的密态左边界以及所述初始的密态右边界开始,循环执行二分定界流程,直至满足循环结束条件,并将循环结束后得到的密态左边界确定为所述密态计算结果;其中,所述二分定界流程包括:根据当前的密态左边界以及当前的密态右边界计算密态中间值;根据所述密态中间值与所述密态除法结果之间的大小关系,对所述当前的密态左边界以及所述当前的密态右边界进行更新,得到新的密态左边界以及新的密态右边界。

[0025] 在上述方案中,通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界,从而可以得到准确度较高的计算结果;同时,还能够支持大批量数据的计算,降低了计算和通信的开销。

[0026] 在可选的实施方式中,所述计算模块还用于:根据如下公式计算所述密态中间值:

[0027] $$mid = left + (right - left)/2;$$

[0028] 其中,mid为所述密态中间值,$left$为所述当前的密态左边界,$right$为所述当前的密态右边界。

[0029] 在上述方案中,在二分定界算法中,可以通过确定密态中间值以对密态左边界以及密态右边界进行更新,从而可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界,从而可以得到准确度较高的计算结果;同时,还能够支持大批量数据的计算,降低了计算和通信的开销。

[0030] 在可选的实施方式中,所述计算模块还用于:当所述密态中间值大于所述密态除法结果时,将所述当前的密态右边界更新为所述密态中间值;或者,当所述密态中间值小于所述密态除法结果时,将所述当前的密态左边界更新为所述密态中间值。

[0031] 在上述方案中,通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界,从而可以得到准确度较高的计算结果;同时,还能够支持大批量数据的计算,降低了计算和通信的开销。

[0032] 在可选的实施方式中,所述计算模块还用于:计算所述密态除法结果与所述密态中间值的差值,得到对应的密态符号位;其中,所述差值大于0时所述密态符号位为0,否则所述密态符号位为1;根据如下公式对所述当前的密态左边界以及所述当前的密态右边界进行更新:

[0033] $$right' = msb\_mid \times mid + (1 - msb\_mid) \times right;$$

[0034] $$left' = (1 - msb\_mid) \times mid + msb\_mid \times left;$$

[0035] 其中，$right'$为所述新的密态右边界，msb_mid为所述密态符号位，mid为所述密态中间值，$right$为所述当前的密态右边界，$left'$为所述新的密态左边界，$left$为所述当前的密态左边界。

[0036] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0037] 在可选的实施方式中，所述预设除数的大小与分组组数的大小相等。

[0038] 在上述方案中，在密态取余计算或者密态取模计算的过程中，其预设除数的大小可以与分组组数的大小，这样，经过对密态取余计算或者密态取模计算的计算结果进行比对后，可以直接将待分组密态数据划分为分组组数对应的组，从而可以提高分组的效率。

[0039] 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；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计算机程序指令，所述处理器调用所述计算机程序指令能够执行如第一方面所述的随机分组的方法。

[0040] 第四方面，本申请实施例提供一种计算机可读存储介质，所述计算机可读存储介质存储计算机程序指令，所述计算机程序指令被计算机运行时，使所述计算机执行如第一方面所述的随机分组的方法。

[0041] 为使本申请的上述目的、特征和优点能更明显易懂，下文特举本申请实施例，并配合所附附图，作详细说明如下。

附图说明

[0042] 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。

[0043] 图1为本申请实施例提供的一种随机分组的方法的流程图；

[0044] 图2为本申请实施例提供的一种对二分定界流程进行循环执行的示意图；

[0045] 图3为本申请实施例提供的一种对用户ID进行随机分组的示意图；

[0046] 图4为本申请实施例提供的另一种对用户ID进行随机分组的示意图；

[0047] 图5为本申请实施例提供的一种随机分组的装置的结构框图；

[0048] 图6为本申请实施例提供的一种电子设备的结构框图。

具体实施方式

[0049] 下面将结合本申请实施例中的附图，对本申请实施例中的技术方案进行描述。

[0050] 在随机分组的场景下，如果需要保护用户流量或者用户人群被分配给某个组别的信息，那么就需要使用密态取模计算或者密态取余算子计算，以保护分组信息不被暴露。

[0051] 其中，取余计算或者取模计算，是指计算整数除法中被除数未被除尽部分，且余数的取值范围为0到除数（不包含除数）之间的整数。取余计算和取模接口计算的区别在于整

数商的计算方式不同，取余运算的整数商参考靠近0原则，而取模运算的整数商参考商值小原则。

[0052] 可以理解的是，对于整型数，取余和取模的步骤是一样的。因此，对于整数a和b，若想求其余数和模，则有：整数商：$c = a / b$；取余/取模：$r = a - b \times c$。

[0053] 因此，在本申请实施例提供的随机分组的方法中，可以采用多方安全计算(Secure Muti-Party Computation，MPC)。其中，MPC是允许一组相互独立的数据所有方在互不信任且不信任任何公开的第三方的条件下，以各自的秘密为输入联合完成某个函数的计算。可以理解的是，在本申请实施例中，涉及的加法、减法、除法都是采用的MPC的碎片态计算算子。

[0054] 需要说明的是，当本申请实施例提供的随机分组的方法涉及第一参与方以及第二参与方时，本申请实施例对第一参与方以及第二参与方的具体实施方式不作具体的限定，本领域技术人员可以根据实际应用场景的不同，进行合适的调整；此外，第一参与方以及第二参与方均可以作为执行主体执行本申请实施例提供的随机分组的方法，本申请实施例对此同样不作具体的限定。

[0055] 下面对本申请实施例提供的随机分组的方法进行详细的介绍。请参照图1，图1为本申请实施例提供的一种随机分组的方法的流程图，该随机分组的方法可以包括如下步骤：

[0056] 步骤S101：获取多个待分组密态数据；其中，待分组密态数据通过对待分组数据进行碎片化处理得到。

[0057] 步骤S102：采用二分定界算法，根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算，得到每个待分组密态数据对应的密态计算结果。

[0058] 步骤S103：将所有密态计算结果进行两两密态比对，并将比对结果相同的待分组密态数据划分至同一组。

[0059] 具体的，在上述步骤S101中，通过对待分组数据进行碎片化处理，可以得到待分组密态数据。可以理解的是，经过碎片化处理得到的待分组密态数据，对于参与方来说，其真实的数据是未知的。

[0060] 需要说明的是，本申请实施例对待分组数据的具体实施方式不作具体的限定，本领域技术人员可以根据实际情况进行合适的调整；举例来说，待分组数据可以为用户身份标识(Identity Document，ID)、用户流量等。

[0061] 此外，本申请实施例对碎片化处理的具体实施方式同样不作具体的限定，本领域技术人员可以结合现有技术进行合适的调整。

[0062] 再者，本申请实施例对获取多个待分组密态数据的具体实施方式也不作具体的限定，本领域技术人员同样可以结合实际情况进行合适的调整。举例来说，可以接收外部设备发送的待分组密态数据；或者，可以通过对待分组数据进行碎片化处理得到对应的待分组密态数据；或者，可以通过MPC碎片交互得到待分组密态数据等。

[0063] 可以理解的是，作为一种实施方式，待分组数据均为大于0的数。

[0064] 在上述步骤S102中，可以采用二分定界算法，根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算，从而得到每个待分组密态数据对应的密态计算结果。

[0065] 针对上述二分界定法：假定针对被除数x以及除数y，$x,y \in Z_{2^l}$，可以根据二分定界算法确定$\alpha \le \frac{x}{y} < \alpha + 1, \alpha \in Z_{2^l}$中的$\alpha$；可以用$\beta = \frac{x}{y}$表示除法结果，其为一个小数；因此，可以通过确定$left \le \beta = \frac{x}{y} < right$中的左边界$left$，来得到对应的$\alpha$。

[0066] 需要说明的是，本申请实施例对预设除数的具体大小不作具体的限定，本领域技术人员可以根据实际情况进行合适的调整。举例来说，预设除数的大小可以与后续进行分组的分组组数的大小相等；或者，预设除数可以为随机确定的一个整数等。

[0067] 其中，上述步骤S102的具体实施方式将在后续实施例中进行详细的介绍，此处暂不说明。

[0068] 在上述步骤S103中，通过将所有密态计算结果进行两两密态比对，可以将比对结果相同的待分组密态数据划分至同一组，从而实现了对待分组数据的分组。

[0069] 可以理解的是，对于参与方来说，每一组内待分组密态数据的真实数据以及每一组具体的划分情况均是未知的，这样，保证了全流程处于秘密状态。

[0070] 在上述方案中，通过在保护原始的待分组数据不暴露的前提下，以碎片态的形式进行多方安全计算，从而有效的保证了数据的隐私性以及安全性。同时，通过二分定界算法对待分组密态数据进行密态取余计算或者密态取模计算，能够得到准确度较高的计算结果，进一步能够提高随机分组的效果。

[0071] 进一步的，在上述实施例的基础上，下面对通过二分定界算法进行密密态取余计算或者密态取模计算的具体实施方式进行详细的介绍。在该种实施方式中，上述步骤S102具体可以包括如下步骤：

[0072] 步骤1)，针对一个待分组密态数据，获取初始的密态左边界、初始的密态右边界以及该待分组密态数据与预设除数之间的密态除法结果。

[0073] 步骤2)，从初始的密态左边界以及初始的密态右边界开始，循环执行二分定界流程，直至满足循环结束条件，并将循环结束后得到的密态左边界确定为密态计算结果。

[0074] 具体的，在上述步骤1)中，可以获取初始的密态左边界以及初始的密态右边界。其中，为了便于描述，以明文为例，作为一种实施方式，假设初始的密态左边界可以为0，初始的密态右边界可以为$2^{l-1}-1$；因此，需要在$[0,\ 2^{l-1}-1]$的范围内找到上述α。

[0075] 同样的，为了便于描述，以明文为例，作为另一种实施方式，在64bit场景中，l的大小可以为64。

[0076] 在上述步骤2)中，通过循环对密态左边界以及密态右边界进行更新，直到满足循环结束条件时，可以得到最后一次更新得到的密态左边界以及密态右边界。

[0077] 需要说明的是，本申请实施例对上述循环结束条件的具体实施方式不作具体的限定，本领域技术人员可以根据实际情况进行合适的调整。举例来说，循环结束条件可以为循环次数达到预设次数；或者，循环结束条件可以为密态左边界与密态右边界之间的差值小于预设阈值等。

[0078] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0079] 进一步的，在上述实施例的基础上，上述二分定界流程可以包括如下步骤：

[0080] 步骤1)，根据当前的密态左边界以及当前的密态右边界计算密态中间值。

[0081] 步骤2)，根据密态中间值与密态除法结果之间的大小关系，对当前的密态左边界以及当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界。

[0082] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0083] 进一步的，在上述实施例的基础上，上述根据当前的密态左边界以及当前的密态右边界计算密态中间值的步骤，具体可以包括如下步骤：

[0084] 根据如下公式计算密态中间值：

[0085] $$mid = left + (right - left)/2;$$

[0086] 其中，mid为密态中间值，$left$为当前的密态左边界，$right$为当前的密态右边界。

[0087] 在上述方案中，在二分定界算法中，可以通过确定密态中间值以对密态左边界以及密态右边界进行更新，从而可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0088] 进一步的，在上述实施例的基础上，上述根据密态中间值与密态除法结果之间的大小关系，对当前的密态左边界以及当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界的步骤，具体可以包括如下步骤：

[0089] 当密态中间值大于密态除法结果时，将当前的密态右边界更新为密态中间值；或者，当密态中间值小于密态除法结果时，将当前的密态左边界更新为密态中间值。

[0090] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0091] 进一步的，在上述实施例的基础上，上述根据密态中间值与密态除法结果之间的大小关系，对当前的密态左边界以及当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界的步骤，具体可以包括如下步骤：

[0092] 步骤1)，计算密态除法结果与密态中间值的差值，得到对应的密态符号位；其中，差值大于0时密态符号位为0，否则密态符号位为1。

[0093] 步骤2)，根据如下公式对当前的密态左边界以及当前的密态右边界进行更新：

[0094] $$right' = msb\_mid \times mid + (1 - msb\_mid) \times right;$$

[0095] $$left' = (1 - msb\_mid) \times mid + msb\_mid \times left;$$

[0096] 其中，$right'$为新的密态右边界，msb_mid为密态符号位，mid为密态中间值，$right$为当前的密态右边界，$left'$为新的密态左边界，$left$为当前的密态左边界。

[0097] 进一步的，在上述实施例的基础上，请参照图2，图2为本申请实施例提供的一种对二分定界流程进行循环执行的示意图。

[0098] 可以看出，针对被除数x以及除数y，首先判断循环是否结束，若未结束，则执行上述二分定界流程：

[0099] 首先，根据当前的密态左边界以及当前的密态右边界计算密态中间值：

[0100] $$mid = left + (right - left)/2;$$

[0101] 然后，计算密态除法结果与密态中间值的差值，得到对应的密态符号位；再然后，根据上述密态符号位，对当前的密态左边界以及当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界：

[0102] $$right' = msb\_mid \times mid + (1 - msb\_mid) \times right;$$

[0103] $$left' = (1 - msb\_mid) \times mid + msb\_mid \times left;$$

[0104] 循环执行上述二分定界流程，直至循环结束，将循环结束后得到的密态左边界确定为密态计算结果。

[0105] 为了便于描述，以明文为例，假设初始的密态左边界为0、初始的密态右边界为7、被除数为3、预设除数为4，那么，第一次执行二分定界流程可以得到：密态中间值为$[0 + (7 - 0)/2] = 3$，密态除法结果与密态中间值的差值为$(0.75 - 3) = -2.25$，则符号位为1，更新后的密态左边界为0、更新后的密态右边界为3；第二次执行二分定界流程可以得到：密态中间值为$[0 + (3 - 0)/2] = 1$，密态除法结果与密态中间值的差值为$(0.75 - 1) = -0.25$，则符号位为1，更新后的密态左边界为0、更新后的密态右边界为1；若此时满足循环结束条件，则可以将密态计算结果确定为0。可以看出，上述密态计算结果符合$3\%4 = 3 - 0 * 4 = 3$。

[0106] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0107] 进一步的，在上述实施例的基础上，预设除数的大小与分组组数的大小相等。

[0108] 在上述方案中，在密态取余计算或者密态取模计算的过程中，其预设除数的大小可以与分组组数的大小，这样，经过对密态取余计算或者密态取模计算的计算结果进行比对后，可以直接将待分组密态数据划分为分组组数对应的组，从而可以提高分组的效率。

[0109] 下面基于三种具体的应用场景，对本申请实施例提供的随机分组的方法进行介绍。

[0110] 首先介绍第一种应用场景：某银行与某保险机构需要对两家机构的共同用户随机分组，进行隐私保护前提下的商业推广活动，不能暴露用户的敏感信息，同时也不能暴露用户的分组信息，避免对推广活动产生影响。

[0111] 其中，为了便于描述，采用明文数据来描述，可以理解的是，真实计算场景是采用MPC的算子进行碎片态计算实现。请参照图3，图3为本申请实施例提供的一种对用户ID进行随机分组的示意图。在该示意图中，假设：银行持有用户群A，用户群A包括以下ID的用户[121,534,781]；保险机构持有用户群B，用户群B包括与喜爱ID的用户[534,781, 994, 236]。

[0112] 第一步，分别对上述两组用户ID进行哈希运算。

[0113] 第二步,对哈希运算的结果进行碎片化处理。

[0114] 第三步,银行与保险机构进行MPC碎片交互。

[0115] 第四步,对碎片态的ID进行求取交集,得到交集ID碎片。以上述明文为例,交集用户为[533,781]。

[0116] 第五步,对交集ID碎片进行碎片态取模,得到取模的碎片结果。以上述明文为例,假如分为2组(组别为0和1两组),对交集用户ID进行取模计算,分别得到取模结果为0和1。

[0117] 第六步,将取模碎片态结果与分组组别进行MPC碎片态比较计算,得到是否相等的碎片结果。以上述明文为例,将0与组别0、1分别做比较,显然0与0相等,那么比较结果为1,而0与1不同,结果为0,所以用户ID 533分组后,每个组别中的结果为:组别0持 [533],组别1持有[0];同理,对用户ID 781也做同样的处理,得到更新后的分组结果:组别0持有[533, 0],组别1持有[0, 781]。

[0118] 第七步,对每一组别通过MPC排序过滤0碎片无效数据。以上述明文为例,对每一组别进行排序,得到组别0持有 [533, 0],组别1持有[781,0];再过滤出非0的元素,得到最终的随机分组结果为组别0持有[533],组别1持有[781],完成最终的密态分组。

[0119] 接下来介绍第二种应用场景:在互联网公司中,当用户规模达到一定的量级之后,数据驱动能够帮助公司更好的决策和发展。在公司各个团队中,经常会面临不同的产品设计方案的选择或者多个算法方案的决策,比如顶部导航栏的排序方案一二三、派单算法一二三等;此时,可以采用AB实验进行决策分析。

[0120] 一次完整的AB实验可以分为以下几步:第一步:设计实验方案,包括确定实验对象,划分实验组,确定实验提升目标等;第二步:进行人群分组,一般是一个空白组加一个或多个实验组;第三步:将需要实验的策略,方案或者功能施加到各个组,收集数据;第四步:对实验关心的指标进行分析观察。

[0121] 采用随机分组的方式进行上述AB实验,可以实现为对实验对象的某个ID字段进行哈希后对100取模/取余计算,根据结果值进入不同的桶,多个不同的组分别占有一定比例的桶。实验对象在哈希取模之后,会得到0 ~ 99的一个数,即为该实验对象落入的桶。这个桶所属的组就是该实验对象的组。

[0122] 作为一种实施方式,可以对随机分组进行简单的优化,即进行Re-randomization。在每次跑完随机分组之后,验证随机分组结果组间的差异是否小于实验设定的阈值。当各组的观察指标小于阈值或者重新分组次数大于最大允许分组次数后,停止分组。请参照图4,图4为本申请实施例提供的另一种对用户ID进行随机分组的示意图。

[0123] 接下来介绍第三种应用场景:对用户流量进行随机分组,以减轻单组机器的负载。假定:将机器共计分为10组,一直处于在线状态,每组包含两台真实DB服务器,每台DB均使用内存数据库;由于分成了10个DB节点,因此,在对每条数据存取时必须要均匀分配到这10个节点上。

[0124] 以客户数据存取为例:假设每条客户数据的主键是手机号,首先,自定义一个哈希函数,将输入的字符串转成一个Int类型的值,例如:Hash(手机号)=>返回值(int 类型值);然后,对客户手机号进行哈希计算得到一个int值;然后,再将哈希之后的值对总DB节点数(10)进行取余/取模计算,得到的余数在0-9之间;最后,由于0-9分别对应一个DB节点,因此,可以将客户数据均匀分布到多个DB节点上。

[0125] 请参照图5，图5为本申请实施例提供的一种随机分组的装置的结构框图，该随机分组的装置500包括：获取模块501，用于获取多个待分组密态数据；其中，所述待分组密态数据通过对待分组数据进行碎片化处理得到；计算模块502，用于采用二分定界算法，根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算，得到每个待分组密态数据对应的密态计算结果；分组模块503，用于将所有密态计算结果进行两两密态比对，并将比对结果相同的待分组密态数据划分至同一组。

[0126] 在上述方案中，通过在保护原始的待分组数据不暴露的前提下，以碎片态的形式进行多方安全计算，从而有效的保证了数据的隐私性以及安全性。同时，通过二分定界算法对待分组密态数据进行密态取余计算或者密态取模计算，能够得到准确度较高的计算结果，进一步能够提高随机分组的效果。

[0127] 进一步的，在上述实施例的基础上，所述计算模块502具体用于：针对一个待分组密态数据，获取初始的密态左边界、初始的密态右边界以及该待分组密态数据与所述预设除数之间的密态除法结果；从所述初始的密态左边界以及所述初始的密态右边界开始，循环执行二分定界流程，直至满足循环结束条件，并将循环结束后得到的密态左边界确定为所述密态计算结果；其中，所述二分定界流程包括：根据当前的密态左边界以及当前的密态右边界计算密态中间值；根据所述密态中间值与所述密态除法结果之间的大小关系，对所述当前的密态左边界以及所述当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界。

[0128] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0129] 进一步的，在上述实施例的基础上，所述计算模块502还用于：根据如下公式计算所述密态中间值：

[0130] $$mid = left + (right - left)/2;$$

[0131] 其中，mid为所述密态中间值，$left$为所述当前的密态左边界，$right$为所述当前的密态右边界。

[0132] 在上述方案中，在二分定界算法中，可以通过确定密态中间值以对密态左边界以及密态右边界进行更新，从而可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0133] 进一步的，在上述实施例的基础上，所述计算模块502还用于：当所述密态中间值大于所述密态除法结果时，将所述当前的密态右边界更新为所述密态中间值；或者，当所述密态中间值小于所述密态除法结果时，将所述当前的密态左边界更新为所述密态中间值。

[0134] 在上述方案中，通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界，从而可以得到准确度较高的计算结果；同时，还能够支持大批量数据的计算，降低了计算和通信的开销。

[0135] 进一步的，在上述实施例的基础上，所述计算模块502还用于：计算所述密态除法结果与所述密态中间值的差值，得到对应的密态符号位；其中，所述差值大于0时所述密态

符号位为0,否则所述密态符号位为1;根据如下公式对所述当前的密态左边界以及所述当前的密态右边界进行更新:

[0136] $right' = msb\_mid \times mid + (1 - msb\_mid) \times right$;

[0137] $left' = (1 - msb\_mid) \times mid + msb\_mid \times left$;

[0138] 其中,$right'$为所述新的密态右边界,msb_mid为所述密态符号位,mid为所述密态中间值,$right$为所述当前的密态右边界,$left'$为所述新的密态左边界,$left$为所述当前的密态左边界。

[0139] 在上述方案中,通过二分定界算法可以快速确定密态取余计算或者密态取模计算对应的计算结果的上下边界,从而可以得到准确度较高的计算结果;同时,还能够支持大批量数据的计算,降低了计算和通信的开销。

[0140] 进一步的,在上述实施例的基础上,所述预设除数的大小与分组组数的大小相等。

[0141] 在上述方案中,在密态取余计算或者密态取模计算的过程中,其预设除数的大小可以与分组组数的大小,这样,经过对密态取余计算或者密态取模计算的计算结果进行比对后,可以直接将待分组密态数据划分为分组组数对应的组,从而可以提高分组的效率。

[0142] 请参照图6,图6为本申请实施例提供的一种电子设备的结构框图,该电子设备600包括:至少一个处理器601,至少一个通信接口602,至少一个存储器603和至少一个通信总线604。其中,通信总线604用于实现这些组件直接的连接通信,通信接口602用于与其他节点设备进行信令或数据的通信,存储器603存储有处理器601可执行的机器可读指令。当电子设备600运行时,处理器601与存储器603之间通过通信总线604通信,机器可读指令被处理器601调用时执行上述随机分组的方法。

[0143] 例如,本申请实施例的处理器601通过通信总线604从存储器603读取计算机程序并执行该计算机程序可以实现如下方法:步骤S101:获取多个待分组密态数据;其中,待分组密态数据通过对待分组数据进行碎片化处理得到。步骤S102:采用二分定界算法,根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算,得到每个待分组密态数据对应的密态计算结果。步骤S103:将所有密态计算结果进行两两密态比对,并将比对结果相同的待分组密态数据划分至同一组。

[0144] 其中,处理器601包括一个或多个,其可以是一种集成电路芯片,具有信号的处理能力。上述的处理器601可以是通用处理器,包括中央处理器(Central Processing Unit,简称CPU)、微控制单元(Micro Controller Unit,简称MCU)、网络处理器(Network Processor,简称NP)或者其他常规处理器;还可以是专用处理器,包括神经网络处理器(Neural-network Processing Unit,简称NPU)、图形处理器(Graphics Processing Unit,简称GPU)、数字信号处理器(Digital Signal Processor,简称DSP)、专用集成电路(Application Specific Integrated Circuits,简称ASIC)、现场可编程门阵列(Field Programmable Gate Array,简称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。并且,在处理器601为多个时,其中的一部分可以是通用处理器,另一部分可以是专用处理器。

[0145] 存储器603包括一个或多个,其可以是,但不限于,随机存取存储器(Random Access Memory,简称RAM),只读存储器(Read Only Memory,简称ROM),可编程只读存储器

(Programmable Read-Only Memory,简称PROM),可擦除可编程只读存储器(Erasable Programmable Read-Only Memory,简称EPROM),电可擦除可编程只读存储器(Electric Erasable Programmable Read-Only Memory,简称EEPROM)等。

[0146] 可以理解,图6所示的结构仅为示意,电子设备600还可包括比图6中所示更多或者更少的组件,或者具有与图6所示不同的配置。图6中所示的各组件可以采用硬件、软件或其组合实现。于本申请实施例中,电子设备600可以是,但不限于台式机、笔记本电脑、智能手机、智能穿戴设备、车载设备等实体设备,还可以是虚拟机等虚拟设备。另外,电子设备600也不一定是单台设备,还可以是多台设备的组合,例如服务器集群,等等。

[0147] 本申请实施例还提供了一种计算机可读存储介质,所述计算机可读存储介质存储计算机程序指令,所述计算机程序指令被计算机运行时,使所述计算机执行前述方法实施例所述的随机分组的方法。

[0148] 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。

[0149] 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。

[0150] 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。

[0151] 需要说明的是,功能如果以软件功能模块的形式实现并作为独立的产品销售或使用时,可以存储在一个计算机可读取存储介质中。基于这样的理解,本申请的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的部分可以以软件产品的形式体现出来,该计算机软件产品存储在一个存储介质中,包括若干指令用以使得一台计算机设备(可以是个人计算机,服务器,或者网络设备等)执行本申请各个实施例所述方法的全部或部分步骤。而前述的存储介质包括:U盘、移动硬盘、只读存储器(Read-Only Memory,ROM)随机存取存储器(Random Access Memory,RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

[0152] 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。

[0153] 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。

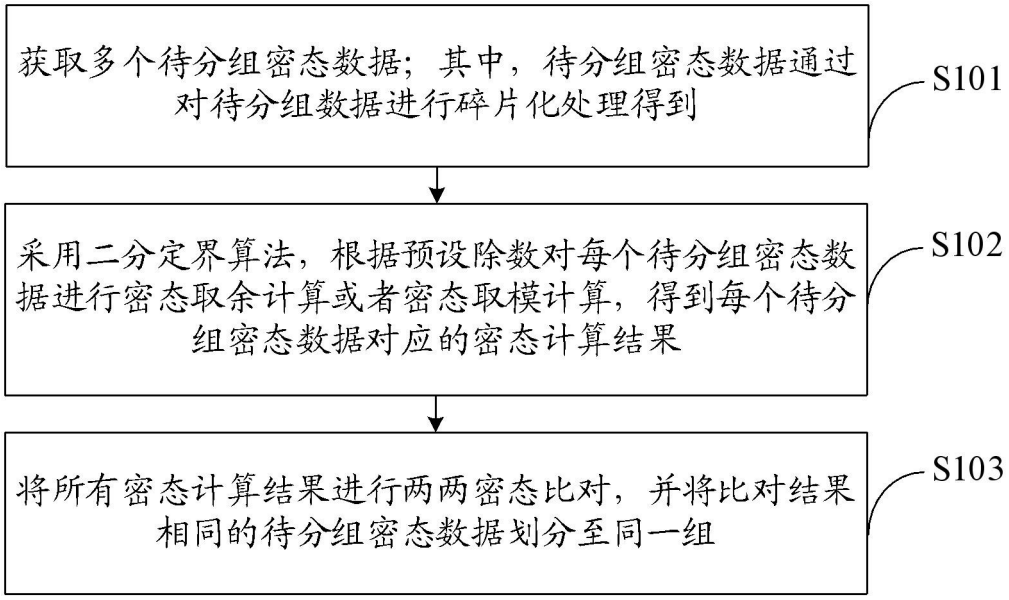

获取多个待分组密态数据；其中，待分组密态数据通过对待分组数据进行碎片化处理得到
S101
采用二分定界算法，根据预设除数对每个待分组密态数据进行密态取余计算或者密态取模计算，得到每个待分组密态数据对应的密态计算结果
S102
将所有密态计算结果进行两两密态比对，并将比对结果相同的待分组密态数据划分至同一组
S103

图1

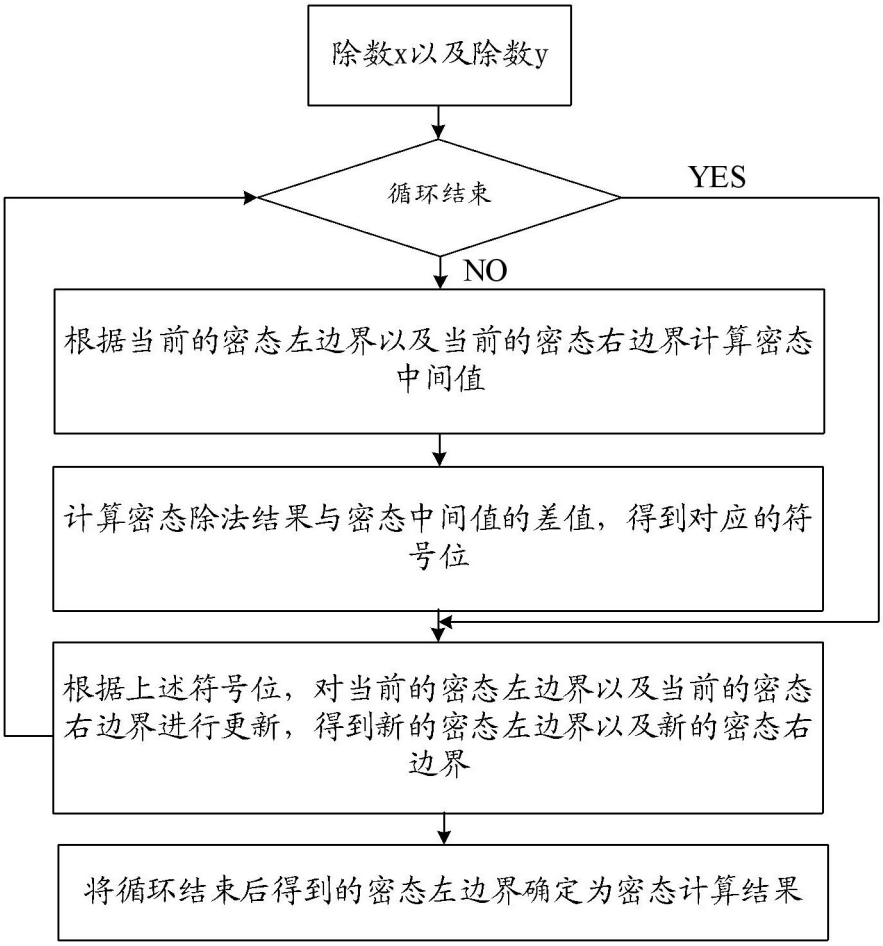

除数x以及除数y
循环结束
YES
NO
根据当前的密态左边界以及当前的密态右边界计算密态中间值
计算密态除法结果与密态中间值的差值，得到对应的符号位
根据上述符号位，对当前的密态左边界以及当前的密态右边界进行更新，得到新的密态左边界以及新的密态右边界
将循环结束后得到的密态左边界确定为密态计算结果

图2

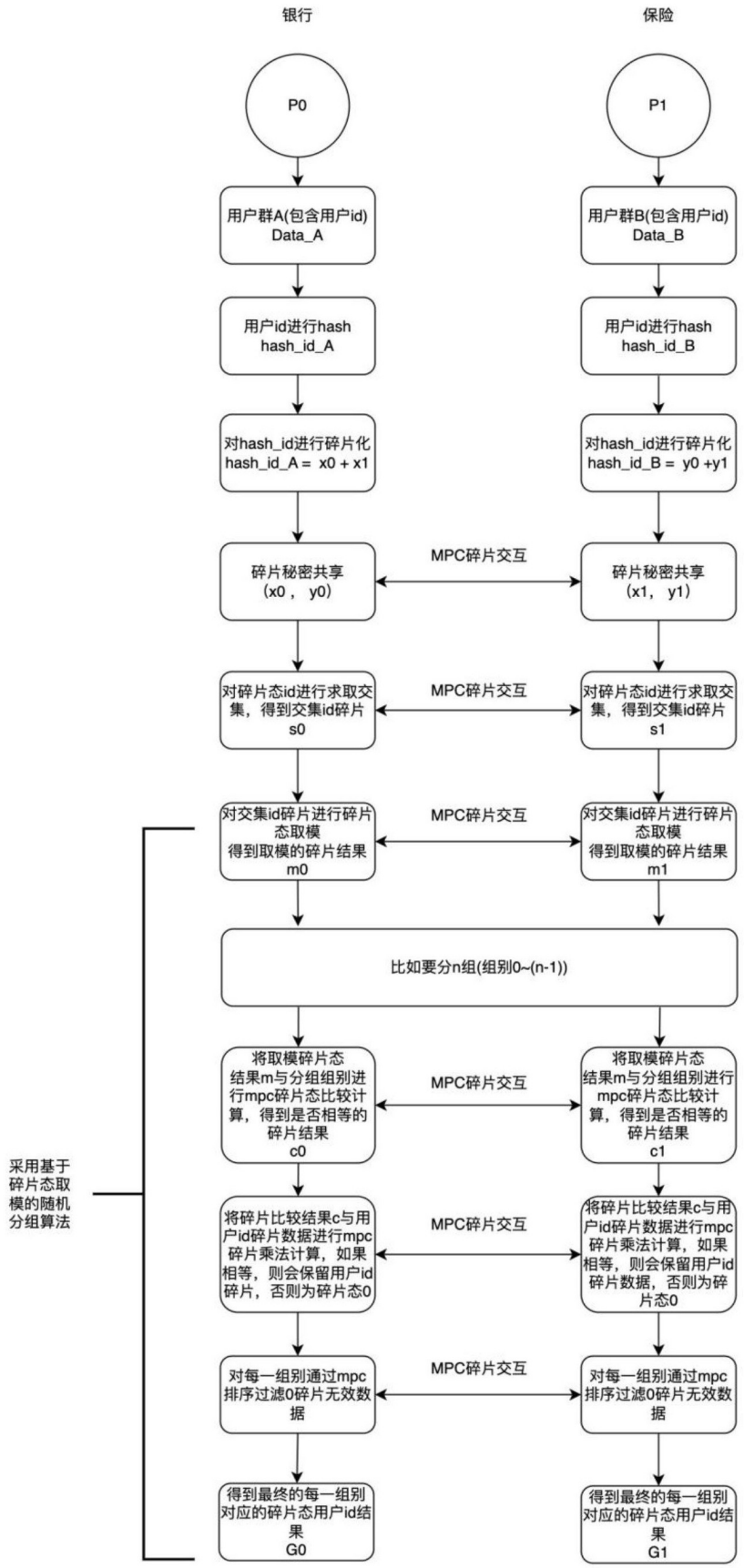

银行
保险
P0
P1
用户群A(包含用户id)
Data_A
用户群B(包含用户id)
Data_B
用户id进行hash
hash_id_A
用户id进行hash
hash_id_B
对hash_id进行碎片化
hash_id_A = x0 + x1
对hash_id进行碎片化
hash_id_B = y0 +y1
碎片秘密共享
(x0， y0)
MPC碎片交互
碎片秘密共享
(x1， y1)
对碎片态id进行求取交集，得到交集id碎片
s0
MPC碎片交互
对碎片态id进行求取交集，得到交集id碎片
s1
对交集id碎片进行碎片态取模
得到取模的碎片结果
m0
MPC碎片交互
对交集id碎片进行碎片态取模
得到取模的碎片结果
m1
比如要分n组(组别0~(n-1))
将取模碎片态结果m与分组组别进行mpc碎片态比较计算，得到是否相等的碎片结果
c0
MPC碎片交互
将取模碎片态结果m与分组组别进行mpc碎片态比较计算，得到是否相等的碎片结果
c1
采用基于碎片态取模的随机分组算法
将碎片比较结果c与用户id碎片数据进行mpc碎片乘法计算，如果相等，则会保留用户id碎片，否则为碎片态0
MPC碎片交互
将碎片比较结果c与用户id碎片数据进行mpc碎片乘法计算，如果相等，则会保留用户id碎片数据，否则为碎片态0
对每一组别通过mpc排序过滤0碎片无效数据
MPC碎片交互
对每一组别通过mpc排序过滤0碎片无效数据
得到最终的每一组别对应的碎片态用户id结果
G0
得到最终的每一组别对应的碎片态用户id结果
G1

图3

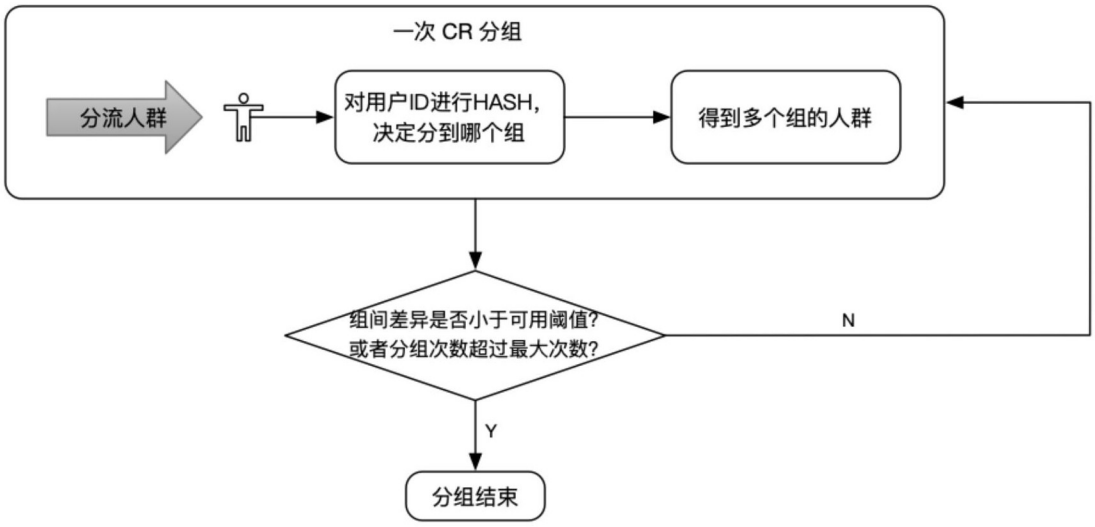

一次 CR 分组
分流人群
对用户ID进行HASH，
决定分到哪个组
得到多个组的人群
组间差异是否小于可用阈值?
或者分组次数超过最大次数?
N
Y
分组结束

图4

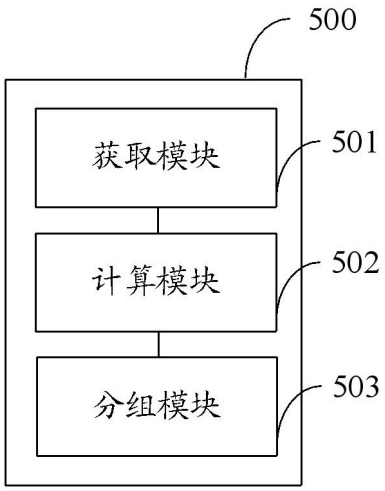

500
获取模块
501
计算模块
502
分组模块
503

图5

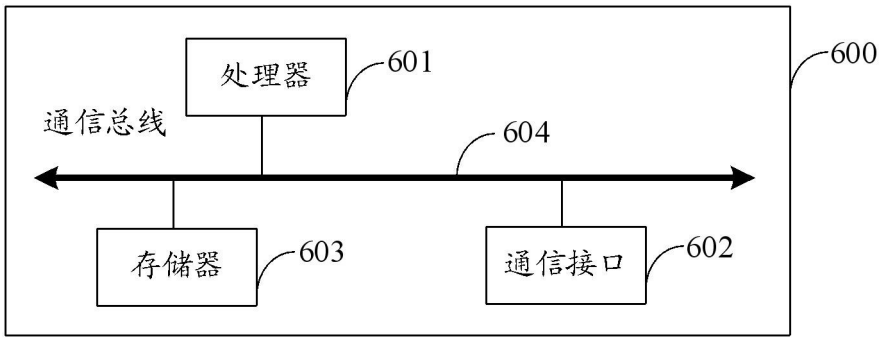

处理器
601
600
通信总线
604
存储器
603
通信接口
602

图6