(19)国家知识产权局

(12)发明专利

(10)授权公告号 CN 114615282 B
(45)授权公告日 2022.08.23

(21)申请号 202210499569.1

(22)申请日 2022.05.10

(65)同一申请的已公布的文献号
申请公布号 CN 114615282 A

(43)申请公布日 2022.06.10

(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 赵东 陈立峰 卞阳

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

(51)Int.Cl.
H04L 67/1074(2022.01)
H04L 67/10(2022.01)

(56)对比文件
CN 111563261 A,2020.08.21
US 2021218576 A1,2021.07.15
CN 110708363 A,2020.01.17
陈莉等.基于分布式线性方程组求解的安全多方计算协议.《信息网络安全》.2013,(第09期),全文.

审查员 王勇

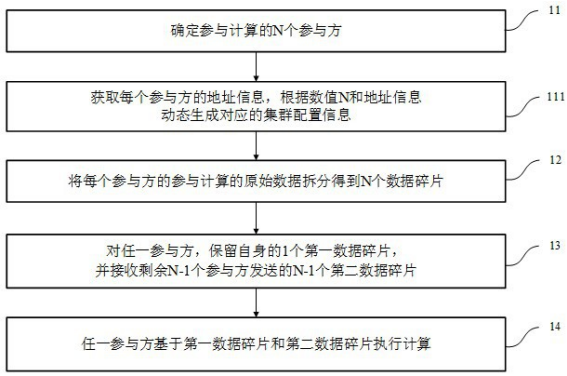
权利要求书3页 说明书9页 附图4页

(54)发明名称

多方安全计算方法、电子设备及可读存储介质

(57)摘要

本发明公开了一种多方安全计算方法、电子设备及可读存储介质，多方安全计算方法包括：确定参与计算的N个参与方；将每个参与方的参与计算的原始数据拆分得到N个数据碎片；对任一参与方，保留自身的1个第一数据碎片，并接收剩余N-1个参与方发送的N-1个第二数据碎片；任一参与方基于第一数据碎片和第二数据碎片执行计算。本申请数据节点与计算节点绑定且等量，不会发生参与的数据方超过计算方可能导致的数据完全出门情况，且数据方始终持有一个碎片不出门，其他参与方即使发生n-1方的合谋，都很难破解真实的原始数据。N作为变量参数使得本方案能够平滑支持N方的安全计算，在保证数据安全前提下，平滑扩展多方安全计算，更具通用性。



CN 114615282 B

1.一种多方安全计算方法,其特征在于,所述多方安全计算方法包括:

确定参与计算的N个参与方;

将每个参与方的参与计算的原始数据拆分得到N个数据碎片;

对任一参与方,保留自身的1个第一数据碎片,并接收剩余N-1个参与方发送的N-1个第二数据碎片;

所述任一参与方基于所述第一数据碎片和所述第二数据碎片执行计算;

所述多方安全计算方法还包括:

所述任一参与方配置生成至少一组与三个三元组种子对应的三元组碎片;其中,所述三个三元组种子包括第一种子、第二种子和第三种子,第一种子和第二种子的乘积等于第三种子;

当多方计算包括乘法计算和/或与操作时,所述任一参与方基于所述第一数据碎片和所述第二数据碎片执行计算的步骤具体包括:

所述任一参与方基于所述第一数据碎片、所述第二数据碎片、对应的三元组碎片执行计算;

若N不小于3,所述任一参与方配置生成至少一组与三个三元组种子对应的三元组碎片的步骤具体包括:

所述任一参与方配置生成与所述三个三元组种子对应的多组三元组碎片;

所述任一参与方基于所述第一数据碎片、所述第二数据碎片、对应的三元组碎片执行计算的步骤具体包括:

从所述任一参与方持有的数据碎片中选取与任意两个原始数据对应的数据碎片作为目标数据碎片;

选取任意一组三元组碎片作为目标三元组碎片;

基于两个目标数据碎片和一组目标三元组碎片计算得到一中间碎片;

将从剩余的数据碎片中任意选取的一个数据碎片和所述中间碎片作为新的目标数据碎片,并返回所述选取任意一组三元组碎片作为目标三元组碎片的步骤,直至所有数据碎片全部执行完计算。

2.如权利要求1所述的多方安全计算方法,其特征在于,所述多方安全计算方法还包括:

获取每个参与方的地址信息,根据数值N和所述地址信息动态生成对应的集群配置信息;

其中,每个参与方基于所述集群配置信息实现与任意参与方的通信,以实现任意两个参与方之间的数据传输。

3.如权利要求1所述的多方安全计算方法,其特征在于,所述基于两个目标数据碎片和一组目标三元组碎片计算得到一中间碎片的步骤具体包括:

确定第i个参与方持有的两个目标数据碎片为$(X)_i$,$(Y)_i$;

其中,$(X)_i$为第i个参与方持有的与原始数据X对应的数据碎片,$(Y)_i$为第i个参与方持有的与原始数据Y对应的数据碎片,i∈0~N-1;

确定第i个参与方持有的目标三元组碎片为$(a)_i$、$(b)_i$、$(c)_i$；

其中，$a*b=\sum_i(a)_i*\sum_i(b)_i=\sum_i(c)_i=c$；$(a)_i$、$(b)_i$、$(c)_i$分别为第$i$个参与方持有的与种子a、b、c对应的三元组碎片；

计算得到第i个参与方的中间数据$(e)_i$和$(f)_i$；其中，$(e)_i=(X)_i-(a)_i$，$(f)_i=(Y)_i-(b)_i$；

计算得到所有参与方的总中间数据e和f；

其中，$e=\sum_i(e)_i$，$f=\sum_i(f)_i$；

得到第i个参与方的中间碎片的计算结果为

$(z)_i=\max(0,\ 1-i)*e*f+f*(a)_i+e*(b)_i+(c)_i$。

4.如权利要求1所述的多方安全计算方法，其特征在于，所述任一参与方配置生成至少一组与三个三元组种子对应的三元组碎片的步骤中：

若原始数据为数值型数据且多方计算为乘法计算，则生成数值型的三元组碎片；

若原始数据为布尔型数据且多方计算为与操作，则生成布尔型的三元组碎片。

5.如权利要求1所述的多方安全计算方法，其特征在于，所述任一参与方配置生成至少一组与三个三元组种子对应的三元组碎片具体包括：

所述任一参与方随机生成与第一种子和第二种子对应的第一三元组碎片和第二三元组碎片；

所述任一参与方生成一对公钥和私钥；

所述任一参与方将所述第一三元组碎片或所述第二三元组碎片经公钥加密后发送至剩余参与方；

所述任一参与方根据自身的第一三元组碎片、第二三元组碎片和剩余参与方发送的加密后的三元组碎片执行计算得到多个计算结果；

所述任一参与方基于私钥对剩余参与方返回的基于加密后的三元组碎片得到的计算结果进行解密；

所述任一参与方基于自身的计算结果和剩余参与方返回的解密后的计算结果，得到与第三种子对应的第三三元组碎片。

6.如权利要求1所述的多方安全计算方法，其特征在于，所述多方安全计算方法还包括：

原始数据为数值型数据时，设定多方计算的计算精度；

基于所述计算精度对原始数据进行扩展处理；

其中，所述扩展处理包括将预设参数与所述原始数据相乘得到扩展后的数据，不同计算精度对应不同的预设参数，所述计算精度与所述预设参数成正比。

7.一种电子设备，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任一项所述的多方安全计算方法。

8.一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任一项所述的多方安全计算方法。

多方安全计算方法、电子设备及可读存储介质

技术领域

[0001] 本发明属于计算机安全领域，特别涉及一种多方安全计算方法、电子设备及可读存储介质。

背景技术

[0002] 现有技术中，在安全多方计算的实现中，先将任务划分为N个子任务，然后将子任务分发给N个节点(包括计算发起方和计算参与方)分别进行计算并返回计算结果，考虑到应用场景等，传统多方计算的MPC协议(安全多方计算协议)主要针对仅2方或仅3方等的限制计算节点的情况，当需要扩展到更多方计算时，则需要大量修改调整，以实现算子的主要逻辑的改动，甚至有的mpc协议可能都无法扩展应对，且调整后数据存在完全出门的风险，需要重新进行算法设置。

发明内容

[0003] 本发明要解决的技术问题是为了克服现有技术中的上述缺陷，提供一种多方安全计算方法、电子设备及可读存储介质。

[0004] 本发明是通过下述技术方案来解决上述技术问题：

[0005] 一种多方安全计算方法，所述多方安全计算方法包括：

[0006] 确定参与计算的N个参与方；

[0007] 将每个参与方的参与计算的原始数据拆分得到N个数据碎片；

[0008] 对任一参与方，保留自身的1个第一数据碎片，并接收剩余N-1个参与方发送的N-1个第二数据碎片；

[0009] 所述任一参与方基于所述第一数据碎片和所述第二数据碎片执行计算。

[0010] 较佳地，所述多方安全计算方法还包括：

[0011] 获取每个参与方的地址信息，根据数值N和所述地址信息动态生成对应的集群配置信息；

[0012] 其中，每个参与方基于所述集群配置信息实现与任意参与方的通信，以实现任意两个参与方之间的数据传输。

[0013] 较佳地，所述多方安全计算方法还包括：

[0014] 所述任一参与方配置生成至少一组与三个三元组种子对应的三元组碎片；其中，所述三个三元组种子包括第一种子、第二种子和第三种子，第一种子和第二种子的乘积等于第三种子；

[0015] 当多方计算包括乘法计算和/或与操作时，所述任一参与方基于所述第一数据碎片和所述第二数据碎片执行计算的步骤具体包括：

[0016] 所述任一参与方基于所述第一数据碎片、所述第二数据碎片、对应的三元组碎片执行计算。

[0017] 较佳地，若N不小于3，所述任一参与方配置生成至少一组与三个三元组种子对应

的三元组碎片的步骤具体包括：

[0018] 所述任一参与方配置生成与所述三个三元组种子对应的多组三元组碎片；

[0019] 所述任一参与方基于所述第一数据碎片、所述第二数据碎片、对应的三元组碎片执行计算的步骤具体包括：

[0020] 从所述任一参与方持有的数据碎片中选取与任意两个原始数据对应的数据碎片作为目标数据碎片；

[0021] 选取任意一组三元组碎片作为目标三元组碎片；

[0022] 基于两个目标数据碎片和一组目标三元组碎片计算得到一中间碎片；

[0023] 将从剩余的数据碎片中任意选取的一个数据碎片和所述中间碎片作为新的目标数据碎片，并返回所述选取任意一组三元组碎片作为目标三元组碎片的步骤，直至所有数据碎片全部执行完计算。

[0024] 较佳地，所述基于两个目标数据碎片和一组目标三元组碎片计算得到一中间碎片的步骤具体包括：

[0025] 确定第i个参与方持有的两个目标数据碎片为 $\langle X\rangle_i$，$\langle Y\rangle_i$；

[0026] 其中，$\langle X\rangle_i$为第i个参与方持有的与原始数据X对应的数据碎片，$\langle Y\rangle_i$为第i个参与方持有的与原始数据 Y对应的数据碎片，$i\in 0\sim N-1$；

[0027] 确定第 i个参与方持有的目标三元组碎片为 $\langle a\rangle_i$、$\langle b\rangle_i$、$\langle c\rangle_i$；

[0028] 其中，$a*b=\sum_i\langle a\rangle_i*\sum_i\langle b\rangle_i=\sum_i\langle c\rangle_i=c$；$\langle a\rangle_i$、$\langle b\rangle_i$、$\langle c\rangle_i$分别为第 i个参与方持有的与种子a、b、c对应的三元组碎片；

[0029] 计算得到第 i个参与方的中间数据 $\langle e\rangle_i$和 $\langle f\rangle_i$；其中，$\langle e\rangle_i=\langle X\rangle_i-\langle a\rangle_i$，$\langle f\rangle_i=\langle Y\rangle_i-\langle b\rangle_i$；

[0030] 计算得到所有参与方的总中间数据 e和 f；

[0031] 其中，$e=\sum_i\langle e\rangle_i$，$f=\sum_i\langle f\rangle_i$ ；

[0032] 得到第 i个参与方的中间碎片的计算结果为

$\langle z\rangle_i=\max(0,\ 1-i)*e*f+f*\langle a\rangle_i+e*\langle b\rangle_i+\langle c\rangle_i$。

[0033] 较佳地，所述任一参与方配置生成至少一组与三个三元组种子对应的三元组碎片的步骤中：

[0034] 若原始数据为数值型数据且多方计算为乘法计算，则生成数值型的三元组碎片；

[0035] 若原始数据为布尔型数据且多方计算为与操作，则生成布尔型的三元组碎片。

[0036] 较佳地，所述任一参与方配置生成至少一组与三个三元组种子对应的三元组碎片具体包括：

[0037] 所述任一参与方随机生成与第一种子和第二种子对应的第一三元组碎片和第二三元组碎片；

[0038] 所述任一参与方生成一对公钥和私钥；

[0039] 所述任一参与方将所述第一三元组碎片或所述第二三元组碎片经公钥加密后发送至剩余参与方；

[0040] 所述任一参与方根据自身的第一三元组碎片、第二三元组碎片和剩余参与方发送的加密后的三元组碎片执行计算得到多个计算结果；

[0041] 所述任一参与方基于私钥对剩余参与方返回的基于加密后的三元组碎片得到的

计算结果进行解密；

[0042] 所述任一参与方基于自身的计算结果和剩余参与方返回的解密后的计算结果，得到与第三种子对应的第三三元组碎片。

[0043] 较佳地，所述多方安全计算方法还包括：

[0044] 原始数据为数值型数据时，设定多方计算的计算精度；

[0045] 基于所述计算精度对原始数据进行扩展处理；

[0046] 其中，所述扩展处理包括将预设参数与所述原始数据相乘得到扩展后的数据，不同计算精度对应不同的预设参数，所述计算精度与所述预设参数成正比。

[0047] 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，所述处理器执行所述计算机程序时实现上述的多方安全计算方法。

[0048] 一种计算机可读存储介质，其上存储有计算机程序，所述计算机程序被处理器执行时实现上述的多方安全计算方法。

[0049] 本发明的积极进步效果在于：数据节点与计算节点绑定，并且数据节点与计算节点是等量，因此不会发生参与的数据方超过计算方可能导致的数据完全出门情况，另外，因为数据方始终持有1/N碎片在自己本节点不出门，其他参与方即使发生n-1方的合谋，都很难破解真实的原始数据，因此能够应对合谋的风险场景。再者，N作为变量参数使得本方案能够平滑支持N方的安全计算，既能应对2方、3方的计算，而且可以在保证数据安全前提下，平滑扩展至4方及以上的多方安全计算，更具通用性。

附图说明

[0050] 图1为本发明实施例1的多方安全计算方法的流程图。

[0051] 图2为本发明实施例1的多方安全计算方法的一种具体实现方式的流程图。

[0052] 图3为本发明实施例1的多方安全计算方法中三元组碎片生成示意图。

[0053] 图4为本发明实施例1的多方安全计算方法中N不小于3时步骤141的具体实现流程图。

[0054] 图5为本发明实施例2的电子设备的结构示意图。

具体实施方式

[0055] 下面通过实施例的方式进一步说明本发明，但并不因此将本发明限制在所述的实施例范围之中。

[0056] 实施例1

[0057] 一种多方安全计算方法，如图1所示，多方安全计算方法包括：

[0058] 步骤11、确定参与计算的N个参与方；

[0059] 步骤12、将每个参与方的参与计算的原始数据拆分得到N个数据碎片；

[0060] 步骤13、对任一参与方，保留自身的1个第一数据碎片，并接收剩余N-1个参与方发送的N-1个第二数据碎片；

[0061] 需要说明的是，剩余N-1个参与方为参与计算的除了任一参与方自身外的其他参与方。

[0062] 步骤14、任一参与方基于第一数据碎片和第二数据碎片执行计算。

[0063] 本实施例中，步骤11之后，参见图1，多方安全计算方法还包括：

[0064] 步骤111、获取每个参与方的地址信息，根据数值N和地址信息动态生成对应的集群配置信息；

[0065] 其中，每个参与方基于集群配置信息实现与任意参与方的通信，以实现任意两个参与方之间的数据传输。

[0066] 其中，1个节点既是参与方角色同时也是计算方角色，参与方与计算方进行角色绑定，参与方节点提供数据，计算方节点参与协同计算，参与方数量决定了计算方数量。根据参与方的地址信息，生成计算方的集群配置信息，该N方计算以集群配置信息作为参数，触发MPC协议中N方算子的配置初始化。

[0067] 上述方案中，参与方1将其持有的原始数据X，通过算法运算的方式拆分为(X-X2-X3-……-XN)、X2、X3…XN等N份，各碎片求和为原始X值。然后参与方1将这N份碎片分别传输给N个计算方，使得每一个计算方有且仅有一份碎片。由于参与方角色与计算方角色绑定，因此，参与方1与计算方1是同一个节点，不涉及通信，即碎片1相当于是保留在参与方1的本地，只有N-1份碎片涉及通信传输。由于数据节点与计算节点绑定，并且数据节点与计算节点是等量，因此不会发生参与的数据方超过计算方可能导致的数据完全出门情况，另外，因为数据方始终持有1/N碎片在自己本节点不出门，其他参与方即使发生n-1方的合谋，都很难破解真实的原始数据，因此能够应对合谋的风险场景。再者，N作为变量参数使得本方案能够平滑支持N方的安全计算，既能应对2方、3方的计算，而且可以在保证数据安全前提下，平滑扩展至4方及以上的多方安全计算，更具通用性。

[0068] 需要说明的是，对不同的数据类型，做了两种不同的数据拆分方式，对于数值型的数据拆分，通过上述的减法实现，而对于布尔型的数据拆分，通过异或的方式进行分碎片。比如原始数据为布尔型的XB，分碎片方式如下：XB=(XB^XB2^XB3^…^XBN)^XB2^XB3^…^XBN，即数据碎片1为XB^XB2^XB3^…^XBN，数据碎片2为XB2，数据碎片3为XB3，数据碎片N为XBN。

[0069] 以三方加法算子为例，三个数据方Alice、Bob、Clare，假如alice持有X，Bob持有Y，Clare持有Z，那么要秘密求和，经过碎片化分发：Alice持有X1、Y1、Z1三份碎片；Bob持有X2、Y2、Z2三份碎片；Clare持有X3、Y3、Z3三份碎片。其中X=X1+X2+X3，Y=Y1+Y2+Y3，Z=Z1+Z2+Z3。

[0070] 各节点在本地执行各自的碎片加法，即：

[0071] Alice：A=X1+Y1+Z1，Bob：B=X2+Y2+Z2，Clare：C=X3+Y3+Z3。

[0072] 然后，各自接收其他方的计算结果，完成最终计算：SUM_result=A+B+C，其中，原始数据不出门实现求和，其他参与方无法破解原始的X、Y、Z数据。

[0073] 本实施例中，提供多方安全计算方法的一种具体实现方式，如图2所示，步骤14之前，多方安全计算方法还包括：

[0074] 步骤131、任一参与方配置生成至少一组与三个三元组种子对应的三元组碎片；其中，所述三个三元组种子包括第一种子、第二种子和第三种子，第一种子和第二种子的乘积等于第三种子；

[0075] 当多方计算包括乘法计算和/或与操作时，步骤14具体包括：

[0076] 步骤141、任一参与方基于第一数据碎片、第二数据碎片、对应的三元组碎片执行计算。

[0077] 其中，若原始数据为数值型数据且多方计算为乘法计算，则步骤131中生成数值型的三元组碎片；若原始数据为布尔型数据且多方计算为与操作，则步骤131中生成布尔型的三元组碎片。另外，若多方计算涉及多步计算，如果在数值型数据计算后进行布尔型数据计算，则将数值型数据转换为布尔型数据进行后续的计算，反之也同样执行对应的数据转换。

[0078] 本实施例中，前述触发的MPC 协议预配置模块还用于生成N 方三元组种子(基于同态加密方式生成三元组碎片)，并将其加入到集群配置信息中，该配置信息包含N 方的三元组碎片，因此可以提供所需N 个计算方数量信息，共同完成上述的触发MPC 协议中N 方算子的配置初始化。

[0079] 具体的，参见图2，步骤131具体包括：

[0080] 步骤1311、任一参与方随机生成与第一种子和第二种子对应的第一三元组碎片和第二三元组碎片；

[0081] 步骤1312、任一参与方生成一对公钥和私钥；

[0082] 步骤1313、任一参与方将第一三元组碎片或第二三元组碎片经公钥加密后发送至剩余参与方；

[0083] 需要说明的是，剩余参与方为参与计算的除了任一参与方自身外的其他参与方。

[0084] 步骤1314、任一参与方根据自身的第一三元组碎片、第二三元组碎片和剩余参与方发送的加密后的三元组碎片执行计算得到多个计算结果；

[0085] 步骤1315、任一参与方基于私钥对剩余参与方返回的基于加密后的三元组碎片得到的计算结果进行解密；

[0086] 步骤1316、任一参与方基于自身的计算结果和剩余参与方返回的解密后的计算结果，得到与第三种子对应的第三三元组碎片。

[0087] 需要说明的是，三元组碎片生成过程中，任一参与方还生成与剩余参与方对应的随机数以用于三元组碎片的生成过程。

[0088] 需要说明的是，本实施例中，三个三元组种子是一个虚拟中间变量，实际运行过程中，基于同态加密的方式在每个参与方分别配置生成两个随机三元组碎片，也即与第一种子对应的第一碎片以及与第二种子对应的第二碎片，所有参与方的第一碎片之和与所有参与方的第二碎片之和的乘积等于第三种子碎片之和。需要说明的是，碎片由各参与方自行配置，在运算过程中，会将其中一个碎片进行加密发送至其他参与方进而得到第三种子碎片。

[0089] 举个具体示例进行说明，参见图3，参与方包括：P1…Pn；

[0090] 以P1参与方为例，随机生成与第一种子和第二种子对应的第一三元组碎片A1和第二三元组碎片B1，并生成一对公钥Puk1和私钥Prk1；

[0091] P1将A1经公钥加密后得到PA1并发送至其他参与方；图中r12…r1N为P1对其他参与方生成的随机数。

[0092] P1根据A1、B1和其他参与方发送PA2…PAN执行计算得到Z11、Z12…Z1N，然后接收其他参与方返回的将Z21…ZN1，基于自身的私钥Prk21对Z21…ZN1进行解密，然后根据Z11和解密后的Z21…ZN1计算得到C1。

[0093] 关于三元组碎片的有效性检验如下：

[0094] A*B=(A1+A2+A3+...+AN)*(B1+B2+B3+...+BN)

[0095] =A1*B1+A1*B2+A1*B3+...+A1*BN+A2*B1+A2*B2+A2*B3+...+A2*BN+A3*B1+A3*B2+A3*B3+...+A3*BN+AN*B1+AN*B2+AN*B3+...+AN*BN

[0096] =A1*B1+A1*B2+A1*B3+...+A1*BN+(r21+r31+...+rN1)-(r12+r13+...+r1N)+A2*B1+A2*B2+A2*B3+...+A2*BN+(r12+r32+...+rN2)-(r21+r23+...+r2N)+A3*B1+A3*B2+A3*B3+...+A3*BN+(r13+r23+...+rN3)-(r31+r32+...+r3N)+AN*B1+AN*B2+AN*B3+...+AN*BN+(r1N+r2N+r3N+...)-(rN1+rN2+rN3+...)

[0097] =C1+C2+C3+...+CN

[0098] =C

[0099] 本实施例中,若N不小于3,步骤131具体包括:

[0100] 任一参与方配置生成与三个三元组种子对应的多组三元组碎片;

[0101] 需要说明的是,上述实现方式中,优选的是生成N-1组三元组碎片,进而对任意两个原始数据的碎片计算时,分别使用不同的三元组碎片加入计算,亦或者生成小于N-1组的三元组碎片,部分三元组碎片重复执行不同的数据计算也是可行的,需要说明的是,在重复使用同一组三元组碎片的情况下,要随机的比如基于洗牌式的方式进行挑选,避免连续选取同一组碎片进行计算,以避免数据被解密攻破。

[0102] 另外,若某一参与方存有多个原始数据并参与到多方计算的情况下,针对每个原始数据分别生成与一随机的三元组种子对应的三元组碎片以参与到多方计算中。

[0103] 进一步的,如图4所示,步骤141具体包括:

[0104] 步骤1411、从任一参与方持有的数据碎片中选取与任意两个原始数据对应的数据碎片作为目标数据碎片;

[0105] 步骤1412、选取任意一组三元组碎片作为目标三元组碎片;

[0106] 步骤1413、基于两个目标数据碎片和一组目标三元组碎片计算得到一中间碎片;

[0107] 步骤1414、将从剩余的数据碎片中任意选取的一个数据碎片和中间碎片作为新的目标数据碎片,并返回步骤1412选取新的目标三元组碎片,直至所有数据碎片全部执行完计算。

[0108] 其中,步骤143具体包括:

[0109] 确定第i个参与方持有的两个目标数据碎片为$\langle X\rangle_i$,$\langle Y\rangle_i$;

[0110] 其中,$\langle X\rangle_i$为第i个参与方持有的与原始数据X对应的数据碎片,$\langle Y\rangle_i$为第i个参与方持有的与原始数据Y对应的数据碎片,$i\in 0\sim N-1$;

[0111] 确定第i个参与方持有的目标三元组碎片为$\langle a\rangle_i$、$\langle b\rangle_i$、$\langle c\rangle_i$;

[0112] 其中,$a*b=\sum_i\langle a\rangle_i*\sum_i\langle b\rangle_i=\sum_i\langle c\rangle_i=c$;$\langle a\rangle_i$、$\langle b\rangle_i$、$\langle c\rangle_i$分别为第$i$个参与方持有的与种子a、b、c对应的三元组碎片;

[0113] 计算得到第i个参与方的中间数据$\langle e\rangle_i$和$\langle f\rangle_i$;其中,$\langle e\rangle_i=\langle X\rangle_i-\langle a\rangle_i$,$\langle f\rangle_i=\langle Y\rangle_i-\langle b\rangle_i$;

[0114] 计算得到所有参与方的总中间数据e和f;

[0115] 其中,$e=\sum_i\langle e\rangle_i$,$f=\sum_i\langle f\rangle_i$;

[0116] 得到第i个参与方的中间碎片的计算结果为

$\langle z\rangle_i=\max(0,1-i)*e*f+f*\langle a\rangle_i+e*\langle b\rangle_i+\langle c\rangle_i$。

[0117] 上述方案中,乘法以及与操作的计算算子借助Beaver三元组来执行计算。以两方

乘法为例,对beaver三元组进行介绍,然后结合多方计算举个具体示例阐述上述方案。

[0118] 以Alice、Bob两方为例,假如各方持有原始数据X,Y。

[0119] 1)经过碎片化并分发:

[0120] Alice:<x>0,<y>0,Bob:<x>1,<y>1;

[0121] 2)各参与方通过同态加密方式生成三元组碎片,其中,三元组种子为a、b、c:

[0122] Alice:<a>0,<b>0,<c>0,Bob:<a>1,<b>1,<c>1;

[0123] 3)每一方持有的碎片信息为:

[0124] Alice:<x>0,<y>0,<a>0,<b>0,,<c>0,Bob:<x>1,<y>1,<a>1,<b>1,<c>1;

[0125] 4)三方各自计算自己的<e>=<x>-<a>,<f>=<y>-<b>,用来盲化原始数据<x>,<y>,得到各方持有:

[0126] Alice:<x>0,<y>0,<a>0,<b>0,<c>0,<e>0,<f>0,Bob:<x>1,<y>1,<a>1,<b>1,<c>1,<e>1,<f>1;

[0127] 5)三方共享自己的<e>,<f>,计算出e,f,得到各方持有:

[0128] Alice:<x>0,<y>0,<a>0,<b>0,<c>0,e,f,Bob:<x>1,<y>1,<a>1,<b>1,<c>1,e,f;

[0129] 6)各方自行执行计算,各方持有:

[0130] Alice:<z>1=f*<a>0+e*<b>0+<c>0,Bob:<z>2=e*f+f*<a>1+e*<b>1+<c>1;

[0131] 7)计算最终结果

[0132] <z>1+<z>2=f*<a>0+e*<b>0+<c>0+e*f+f*<a>1+e*<b>1+<c>1

[0133] =e*f+f*a+e*b+c

[0134] =(x-a)*(y-b)+(y-b)*a+(x-a)*b+c

[0135] =xy-bx-ay+ab+ay-ab+bx-ab+c

[0136] =xy-ab+c

[0137] =xy-ab+ab

[0138] =xy

[0139] 由此,可以得证:两方乘法成立。

[0140] 需要说明的是,当拓展到N方计算时,执行两两原始数据对应的数据碎片依次执行计算即可完成对所有数据的计算。

[0141] 本实施例中,多方安全计算方法还包括:

[0142] 原始数据为数值型数据时,设定多方计算的计算精度;基于计算精度对原始数据进行扩展处理;

[0143] 其中,扩展处理包括将预设参数与原始数据相乘得到扩展后的数据,不同计算精度对应不同的预设参数,计算精度与预设参数成正比。

[0144] 需要说明的是,mpc的运算一般是基于整型碎片,浮点型数值会通过fixed-point被转成整型。比如若涉及12.345677*13.892832,计算得到的数值带有多位小数,那么,在mpc计算过程中,若直接进行整型处理然后计算,在浮点型数值初始转换的过程就会有部分精度丢失,但由于整型类型的限制,比如64bit,保留过多在计算过程中会发生数值溢出。为了尽可能提高精度,会尽可能多得保留小数位数。通过扩展,能够支持更多小数位数保留,且计算过程中不发生溢出,在恢复结果进行rescale,就可以尽可能逼近真实的小数位数,比如可达到15位小数。目前采取fixed-point方式进行控制,比如64bit下,原值12.3456会

通过12.3456*2^{13}进行扩大后再保留整数部分，丢弃小数部分，在最终恢复的时候进行rescale。

[0145] 另外，前述部分列举了加法和乘法的执行逻辑，阐述了如何将密态计算任务拆分为N个子任务执行。另外，基于加法、乘法、与操作等基础算子，可以进一步衍生实现更复杂的N方运算算子，如减法、求和、除法、平方、指数、多项式函数等运算；或者是联合统计算子，如中位数、排序等。

[0146] 本实施中，数据节点与计算节点绑定，并且数据节点与计算节点是等量，因此不会发生参与的数据方超过计算方可能导致的数据完全出门情况，另外，因为数据方始终持有1/N碎片在自己本节点不出门，其他参与方即使发生n-1方的合谋，都很难破解真实的原始数据，因此能够应对合谋的风险场景。再者，N作为变量参数使得本方案能够平滑支持N方的安全计算，既能应对2方、3方的计算，而且可以在保证数据安全前提下，平滑扩展至4方及以上的多方安全计算，更具通用性。

[0147] 实施例2

[0148] 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，所述处理器执行所述计算机程序时实现实施例1所述的多方安全计算方法。

[0149] 图5为本实施例提供的一种电子设备的结构示意图。图5示出了适于用来实现本发明实施方式的示例性电子设备90的框图。图5显示的电子设备90仅仅是一个示例，不应对本发明实施例的功能和使用范围带来任何限制。

[0150] 如图5所示，电子设备90可以以通用计算设备的形式表现，例如其可以为服务器设备。电子设备90的组件可以包括但不限于：至少一个处理器91、至少一个存储器92、连接不同系统组件(包括存储器92和处理器91)的总线93。

[0151] 总线93包括数据总线、地址总线和控制总线。

[0152] 存储器92可以包括易失性存储器，例如随机存取存储器(RAM)921和/或高速缓存存储器922，还可以进一步包括只读存储器(ROM)923。

[0153] 存储器92还可以包括具有一组(至少一个)程序模块924的程序工具925，这样的程序模块924包括但不限于：操作系统、一个或者多个应用程序、其它程序模块以及程序数据，这些示例中的每一个或某种组合中可能包括网络环境的实现。

[0154] 处理器91通过运行存储在存储器92中的计算机程序，从而执行各种功能应用以及数据处理。

[0155] 电子设备90也可以与一个或多个外部设备94(例如键盘、指向设备等)通信。这种通信可以通过输入/输出(I/O)接口95进行。并且，电子设备90还可以通过网络适配器96与一个或者多个网络(例如局域网(LAN)，广域网(WAN)和/或公共网络，例如因特网)通信。网络适配器96通过总线93与电子设备90的其它模块通信。应当明白，尽管图中未示出，可以结合电子设备90使用其它硬件和/或软件模块，包括但不限于：微代码、设备驱动器、冗余处理器、外部磁盘驱动阵列、RAID(磁盘阵列)系统、磁带驱动器以及数据备份存储系统等。

[0156] 应当注意，尽管在上文详细描述中提及了电子设备的若干单元/模块或子单元/模块，但是这种划分仅仅是示例性的并非强制性的。实际上，根据本申请的实施方式，上文描述的两个或更多单元/模块的特征和功能可以在一个单元/模块中具体化。反之，上文描述的一个单元/模块的特征和功能可以进一步划分为由多个单元/模块来具体化。

[0157] 实施例3

[0158] 一种计算机可读存储介质，其上存储有计算机程序，所述计算机程序被处理器执行时实现实施例1所述的多方安全计算方法。

[0159] 其中，可读存储介质可以采用的更具体可以包括但不限于：便携式盘、硬盘、随机存取存储器、只读存储器、可擦拭可编程只读存储器、光存储器件、磁存储器件或上述的任意合适的组合。

[0160] 在可能的实施方式中，本发明还可以实现为一种程序产品的形式，其包括程序代码，当所述程序产品在终端设备上运行时，所述程序代码用于使所述终端设备执行实现实施例1所述的多方安全计算方法。

[0161] 其中，可以以一种或多种程序设计语言的任意组合来编写用于执行本发明的程序代码，所述程序代码可以完全地在用户设备上执行、部分地在用户设备上执行、作为一个独立的软件包执行、部分在用户设备上部分在远程设备上执行或完全在远程设备上执行。

[0162] 虽然以上描述了本发明的具体实施方式，但是本领域的技术人员应当理解，这仅是举例说明，本发明的保护范围是由所附权利要求书限定的。本领域的技术人员在不背离本发明的原理和实质的前提下，可以对这些实施方式做出多种变更或修改，但这些变更和修改均落入本发明的保护范围。

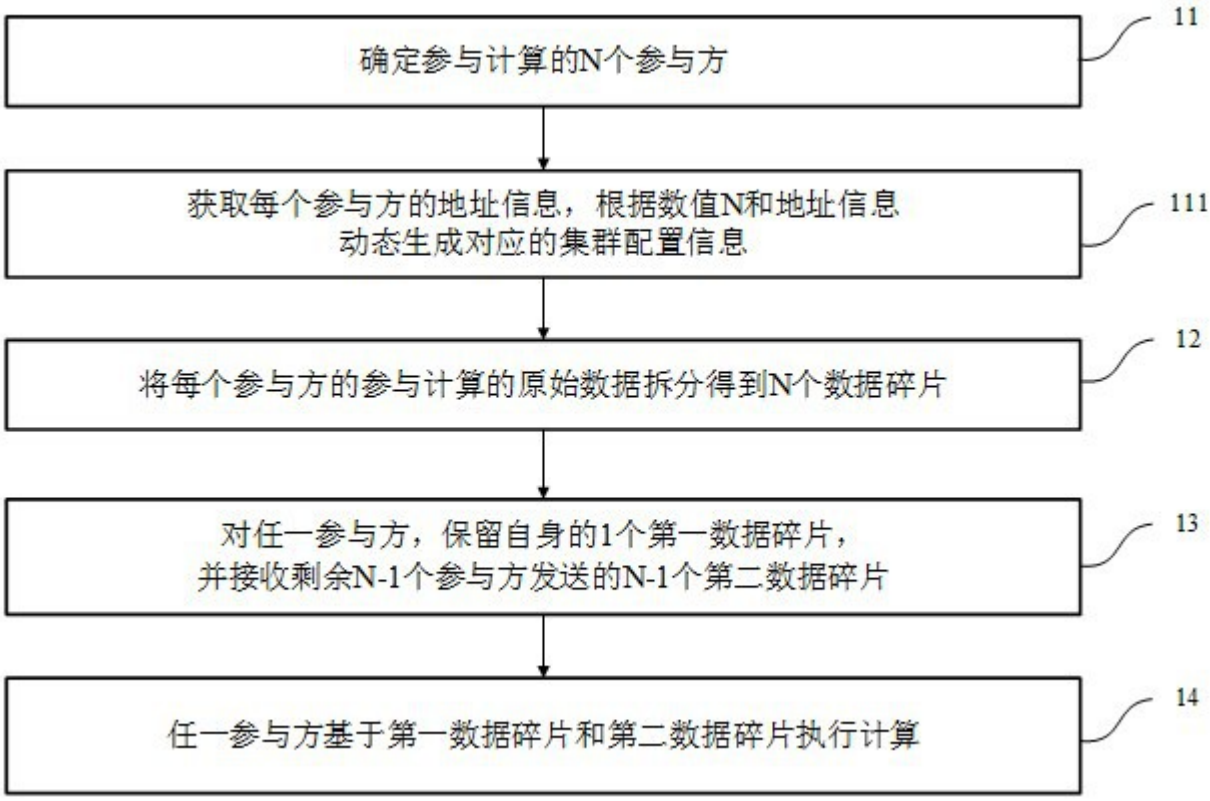

确定参与计算的N个参与方
11
获取每个参与方的地址信息，根据数值N和地址信息
动态生成对应的集群配置信息
111
将每个参与方的参与计算的原始数据拆分得到N个数据碎片
12
对任一参与方，保留自身的1个第一数据碎片，
并接收剩余N-1个参与方发送的N-1个第二数据碎片
13
任一参与方基于第一数据碎片和第二数据碎片执行计算
14

图1

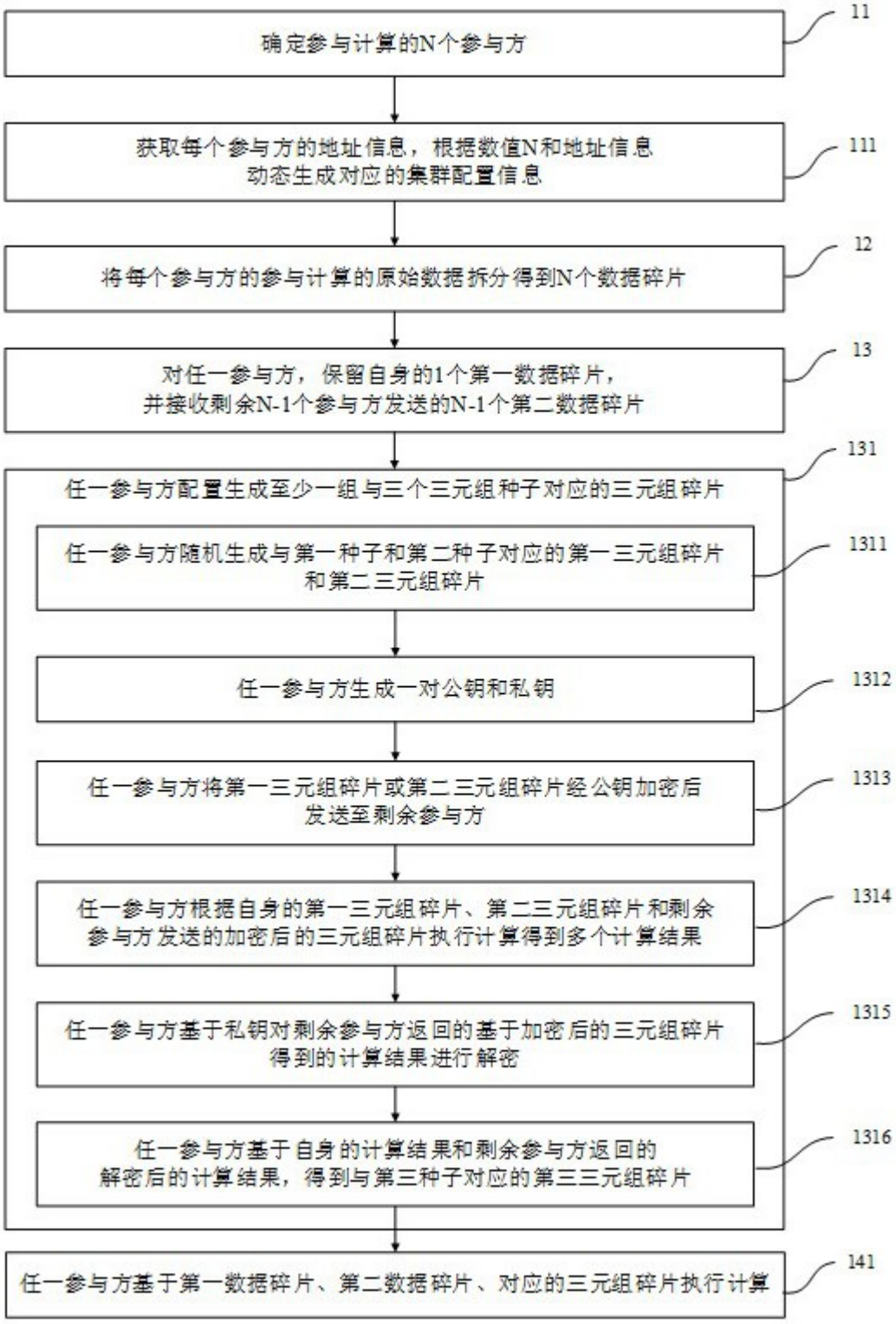

确定参与计算的N个参与方
11
获取每个参与方的地址信息，根据数值N和地址信息
动态生成对应的集群配置信息
111
将每个参与方的参与计算的原始数据拆分得到N个数据碎片
12
对任一参与方，保留自身的1个第一数据碎片，
并接收剩余N-1个参与方发送的N-1个第二数据碎片
13
任一参与方配置生成至少一组与三个三元组种子对应的三元组碎片
131
任一参与方随机生成与第一种子和第二种子对应的第一三元组碎片
和第二三元组碎片
1311
任一参与方生成一对公钥和私钥
1312
任一参与方将第一三元组碎片或第二三元组碎片经公钥加密后
发送至剩余参与方
1313
任一参与方根据自身的第一三元组碎片、第二三元组碎片和剩余
参与方发送的加密后的三元组碎片执行计算得到多个计算结果
1314
任一参与方基于私钥对剩余参与方返回的基于加密后的三元组碎片
得到的计算结果进行解密
1315
任一参与方基于自身的计算结果和剩余参与方返回的
解密后的计算结果，得到与第三种子对应的第三三元组碎片
1316
任一参与方基于第一数据碎片、第二数据碎片、对应的三元组碎片执行计算
141

图2

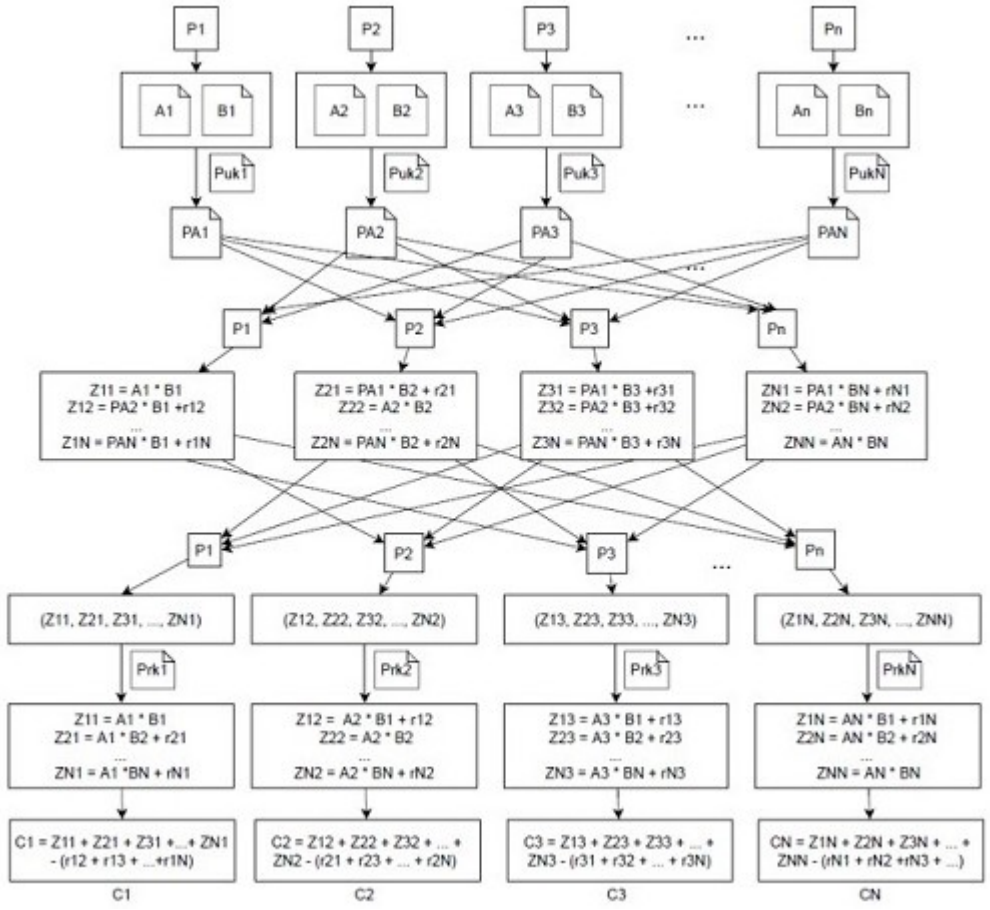

P1
P2
P3
Pn
A1
B1
A2
B2
A3
B3
An
Bn
Puk1
Puk2
Puk3
PukN
PA1
PA2
PA3
PAN
Z11 = A1 * B1
Z12 = PA2 * B1 +r12
Z1N = PAN * B1 + r1N
Z21 = PA1 * B2 + r21
Z22 = A2 * B2
Z2N = PAN * B2 + r2N
Z31 = PA1 * B3 +r31
Z32 = PA2 * B3 +r32
Z3N = PAN * B3 + r3N
ZN1 = PA1 * BN + rN1
ZN2 = PA2 * BN + rN2
ZNN = AN * BN
(Z11, Z21, Z31, ..., ZN1)
(Z12, Z22, Z32, ..., ZN2)
(Z13, Z23, Z33, ..., ZN3)
(Z1N, Z2N, Z3N, ..., ZNN)
Prk1
Prk2
Prk3
PrkN
Z11 = A1 * B1
Z21 = A1 * B2 + r21
ZN1 = A1 *BN + rN1
Z12 = A2 * B1 + r12
Z22 = A2 * B2
ZN2 = A2 * BN + rN2
Z13 = A3 * B1 + r13
Z23 = A3 * B2 + r23
ZN3 = A3 * BN + rN3
Z1N = AN * B1 + r1N
Z2N = AN * B2 + r2N
ZNN = AN * BN
C1 = Z11 + Z21 + Z31 +...+ ZN1 - (r12 + r13 + ...+r1N)
C2 = Z12 + Z22 + Z32 + ... + ZN2 - (r21 + r23 + ... + r2N)
C3 = Z13 + Z23 + Z33 + ... + ZN3 - (r31 + r32 + ... + r3N)
CN = Z1N + Z2N + Z3N + ... + ZNN - (rN1 + rN2 +rN3 + ...)
C1
C2
C3
CN

图3

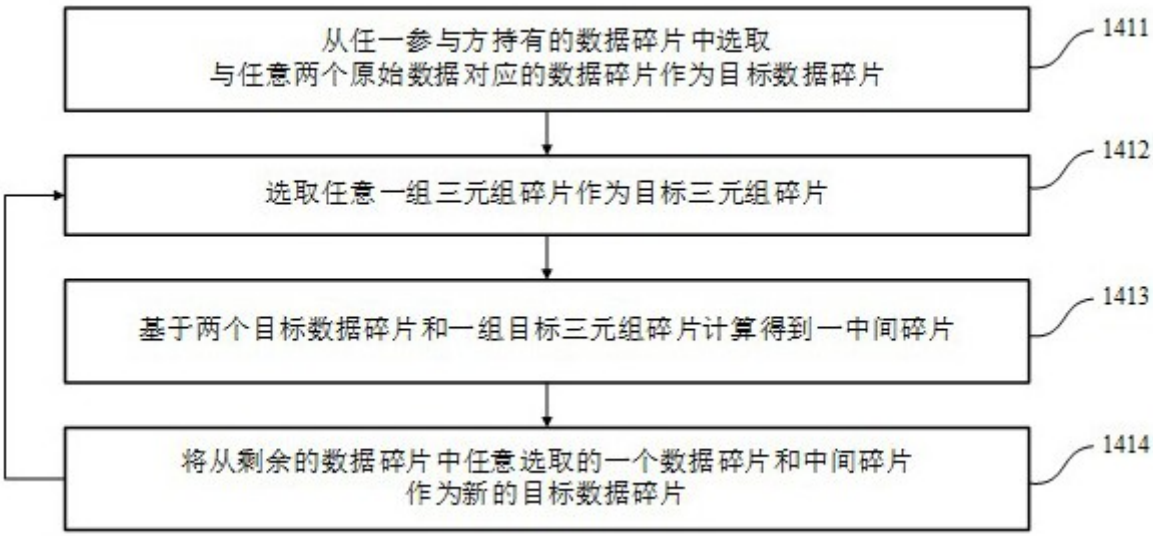

从任一参与方持有的数据碎片中选取
与任意两个原始数据对应的数据碎片作为目标数据碎片
1411
选取任意一组三元组碎片作为目标三元组碎片
1412
基于两个目标数据碎片和一组目标三元组碎片计算得到一中间碎片
1413
将从剩余的数据碎片中任意选取的一个数据碎片和中间碎片
作为新的目标数据碎片
1414

图4

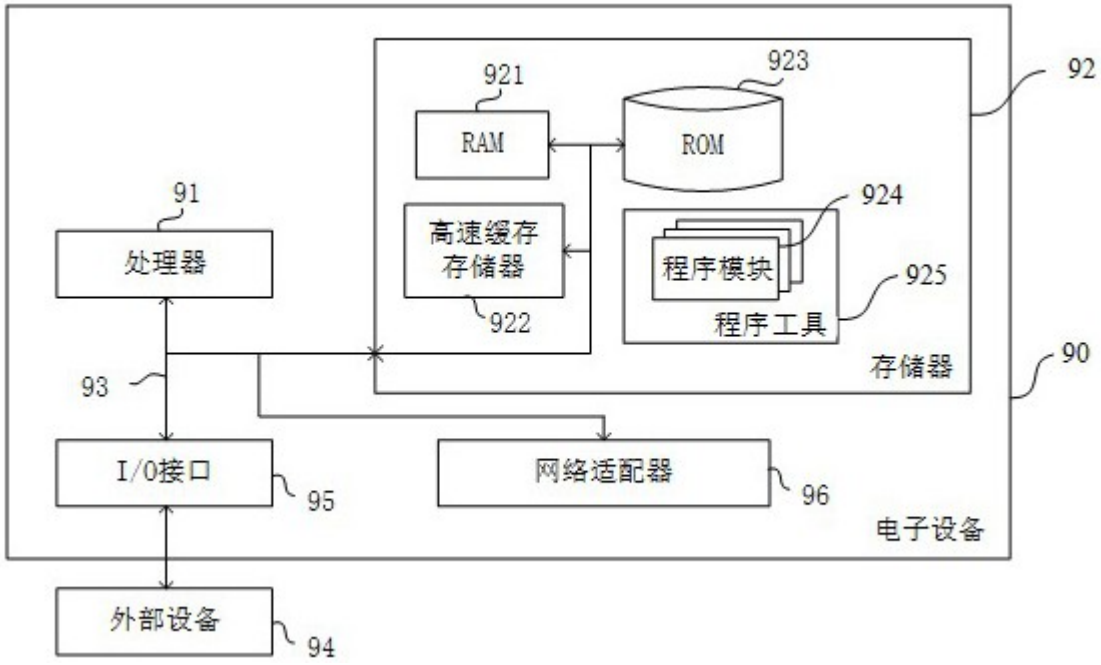

921
923
92
RAM
ROM
91
924
高速缓存
存储器
处理器
程序模块
925
922
程序工具
存储器
90
93
I/O接口
网络适配器
95
96
电子设备
外部设备
94

图5