基于加性秘密共享的洗牌协议的设计

张艳硕[1]，满子琪[1]，周幸妤[1]，杨亚涛[2]，胡荣磊[2]
（1. 北京电子科技学院密码科学与技术系，北京 100070；2. 北京电子科技学院电子与通信工程系，北京 100070）

摘　要：针对现有基于秘密共享的洗牌协议缺少流程实现的具体算法、解决方案多采用公钥、处理大规模数据集时效率低、适用性不足等问题，提出了一种单边洗牌协议，并在此基础上设计了一种基于加性秘密共享的洗牌协议。通过不经意传输协议构建份额转换算法，在不暴露原数据集的前提下完成了洗牌；利用Benes排列网络实现洗牌分解，将复杂的洗牌任务分解为多个子任务，提高了大规模数据集的处理效率；最终通过加性秘密共享，确保将洗牌份额安全地分配给参与方。对所提协议的正确性进行了严格分析，并运用理想-现实模拟范式对其安全性进行了评估。与现有文献相比，所提协议在安全性上能够达到当前安全标准，并在处理大规模数据集时有较高的效率。此外，所提协议的适用性得到了提升，进一步促进了其在当下环境中的应用。

关键词：加性秘密共享；洗牌协议；隐私保护；安全多方计算
中图分类号：TN92
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2024146

Design of shuffling protocol based on additive secret sharing

ZHANG Yanshuo[1], MAN Ziqi[1], ZHOU Xingyu[1], YANG Yatao[2], HU Ronglei[2]
1. Department of Cryptographic Science and Technology, Beijing Electronic Science and Technology Institute, Beijing 100070, China
2. Department of Electronic and Communication Engineering, Beijing Electronic Science and Technology Institute, Beijing 100070, China

Abstract: Aiming at the problems such as lack of specific algorithms for process implementation, using public keys in most of the solutions, low efficiency in dealing with large-scale data sets, and lack of applicability, a unilateral shuffling protocol was proposed, and on this basis, a shuffling protocol based on additive secret sharing was designed. The share conversion algorithm was constructed through the casual transfer protocol, and the shuffling was completed without exposing the original data set. The shuffling task was decomposed into multiple sub-tasks by the Benes arrangement network, which improved the efficiency of large-scale data sets. Finally, through the additive secret sharing, the shuffling shares were safely distributed to the participants. The correctness of the proposed shuffling protocol was analyzed strictly, and its security property was evaluated by using an ideal-reality simulation paradigm. Compared with the existing literature, the proposed protocol can meet the current security standards in security, and has high efficiency in processing large-scale data sets. It improves the applicability of the protocol and further promotes its application in the current environment.

Keywords: additive secret sharing, shuffling protocol, privacy protection, secure multiparty computing

收稿日期：2024–03–18；**修回日期：**2024–07–17
通信作者：张艳硕，zhang_yanshuo@163.com
基金项目：中央高校基本科研业务费资金资助项目（No.3282024003）；“信息安全”国家级一流本科专业建设点基金资助项目（No.2017YFB0801803）；北京市自然科学基金资助项目（No.4232034）
Foundation Items: The Fundamental Research Funds for the Central Universities (No.3282024003),“Information Security”National First-Class Undergraduate Program Construction Point (No.2017YFB0801803), Beijing Natural Science Foundation (No.4232034)

0　引言

数据交集上的函数计算可促使参与方获得对各自有价值的信息。然而，这一过程也面临着数据隐私保护难题的挑战。在此类场景中，隐私交集计算常被采用[1]。通过隐私交集计算，参与方能够在不泄露各自私密信息的前提下，共同计算出基于他们数据交集的函数结果[2]。

然而，数据交集本身也需要保密，现有的隐私交集计算并不能高效地生成加密后的交集结果。针对这一难题，Ciampi 等[3]提出协议可以生成一个加密标识向量，用以标识哪些元素属于数据交集。因此，可将加密的数据元素与相应的加密标识向量进行洗牌，随后参与方就可以利用该加密标识向量丢弃不属于数据交集的元素。为了确保原始元素与洗牌后的元素之间不被关联，洗牌输出的结果需要使用秘密共享的方式进行保护。Chase 等[4]在 2020 年的亚洲密码学会议上，正式提出了基于秘密共享的洗牌协议的概念。基于秘密共享的洗牌协议是安全多方计算领域的一个重要组成部分，通过对数据集进行结合秘密共享的洗牌处理，可有效地保护数据隐私。

基于秘密共享的洗牌协议允许参与方联合洗牌数据，并获得结果的秘密共享[5]。其主要步骤如下：首先参与方对原数据集进行洗牌，生成洗牌数据集；然后参与方对生成的洗牌数据集进行秘密共享，得到自己的秘密份额。在这一过程中，秘密共享保证了除参与方以外的人无法获得数据集的内容，确保了协议的隐私性；洗牌协议保证了参与方只能获得打乱后的数据集，确保了协议的安全性。

Pinkas 等[6]提出了基于洗牌协议的安全多方计算方案，提高了计算效率和可扩展性。Zhao 等[7]提出了一种洗牌协议。在该协议中，每个参与方都会选择一个数据位置，如果该位置与其他参与方没有冲突，则确定这个位置，否则重新选择。该协议避免了第三方，然而反复的冲突检测判断导致计算效率较低。Chen 等[8]提出了一种基于秘密共享和洗牌的数据发布方案。该方案保证了发布数据的原始性和不可链接性，但是具有较为复杂的计算过程，造成了不必要的开销。Attrapadung 等[9]提出了一种用于对数据集应用线性分组动作的高效两方洗牌协议。Han 等[10]实现了一种基于秘密共享的私有数据库可扩展洗牌协议。Belorgey 等[11]基于洗牌协议提出了一种采用全阈值和半诚信安全的模型。张艳硕等[12]对基于秘密共享的洗牌协议进行了综述，为该领域的研究提供了全面的概述和总结。Liang 等[13]提出了一种循环洗牌协议，通过增加洗牌的次数来提高洗牌的随机性与具体效果，同时也带来了额外的开销。Pranav 等[14]设计了一种基于秘密共享且提供快速在线阶段的洗牌协议，为实际应用提供了更高效的解决方案。满子琪等[15]提出了一种基于弹性秘密共享的洗牌协议，该协议利用弹性秘密共享的特性，为洗牌过程增加了一层安全性和灵活性。上述这些研究为基于秘密共享的洗牌协议的发展提供了新的思路和技术支持，同时也为信息安全和隐私保护提供了更多创新的解决方案。

目前，基于秘密共享的洗牌协议已在电子投票[16]、协同过滤[17]、大数据随机抽样[18]等多个领域得到了应用，并展示了其广泛的实用性。在这些应用中，数据所有者不希望他们的个人信息被泄露，但是数据需要在某种程度上被处理，而基于秘密共享的洗牌协议可以用于保护敏感数据的隐私，确保数据处理的公平性和防止恶意操作。由于参与方不需要直接共享原始数据，仅通过洗牌后的数据进行合作，这对涉及机密或敏感信息的合作项目具有重要价值。

针对目前研究缺乏每一流程实现的具体算法、适用性不足、不适用于大规模数据集处理等问题，本文设计了一种基于加性秘密共享的洗牌协议，具有分解算法处理大规模数据集、改进的份额转换算法确保协议安全性、功能灵活提升协议适用性等创新点。本文的主要贡献如下。

1) 安全性方面。本文引入份额转换算法，基于不经意传输协议对其进行构建，确保了参与方无法获得原始数据集的内容，有效地完成了数据的洗牌过程，并增强了数据的隐私性。同时引入秘密共享的方法，有效地防止了恶意参与方的欺骗行为。最后，采用理想–现实模型范式，在半诚实模型下证明了协议的安全性。

2) 效率方面。本文提出了一种新型洗牌算法，可广泛应用于数据集的洗牌，对比分析表明该算法在确保安全性的基础上具有较好的性能。利用 Benes 排列网络实现洗牌分解算法对协议进行优化，减少了协议运行过程中的资源消耗，在数据集较大时提高了协议的性能。

3) 实用性方面。本文协议可以应用到现有的安全多方计算框架中，为数据处理提供了一种新的隐私保护方法。同时，本文协议的可验证性设计使其在需要对数据处理过程进行审计的场景中特别有用，支持分布式的特性使其具有较为广泛的应用性。

1 预备知识

本文提出的洗牌协议基于加性秘密共享机制，利用Benes排列网络将复杂的洗牌任务分解为多个易于处理的子任务，采用不经意传输协议构建份额转换算法，运用秘密共享机制，将数据集划分为多个子份额，以此完成了协议的整体实施。

1.1 洗牌协议

洗牌协议是一种故意打乱一组数据并产生随机序列的协议[19]，主要用于处理在样本均衡的情况下，初始数据可能以某种规律或按照某种顺序进行排列。

洗牌协议具有“乱序”的性质[20]。针对数据元素进行重新排列，洗牌协议可以打乱数据样本的顺序，从而消除或减少数据样本的有序性。经过洗牌后，原有的有序数列为一个随机数列，最终确保任意一个参与方都无法知道其他参与方所拥有的数据信息，也无法判断某个元素是否在其中[21]。

洗牌协议最初由Chaum[22]提出，用以混淆和隐匿信息流的内容和来源。刘涵阅等[23]提出了一种基于折叠技术的洗牌协议。Jho等[24]提出了一种具有统一属性的键控分区的洗牌协议。然而，上述协议需要一个可信的第三方来执行洗牌。

洗牌协议在隐私保护和机器学习模型训练中发挥着重要作用[24]，减少了模型对数据顺序的依赖，从而避免了过拟合或引入偏差。这使得该模型在训练时能够更好地泛化，处理来自不同样本的数据，不会因样本顺序的变化影响而产生偏差。最终达到保护数据隐私和安全的目的，确保任何参与方都无法知道其他参与方所拥有的数据信息，或判断某个元素是否存在。

1.2 秘密共享

秘密共享协议[25]将秘密信息分给不同的参与方，并要求他们只有合作才能恢复出这个秘密信息。该协议保证了即使部分参与方的数据遭受攻击并导致秘密信息泄露，攻击者仍然无法得到原数据集[26]。

秘密共享最早由Shamir[27]提出，确保了未获得足够信息的攻击者无法恢复出秘密信息。Shamir后来又提出了(t,n)门限方案。当参与方所拥有的坐标数量大于或等于门限t时，利用拉格朗日差值多项式法即可求出这个秘密信息。张剑等[28]提出了一种基于多项式插值的多等级秘密共享方案，在该方案中，高等级参与方的权限大于低等级参与方。宋云等[29]基于极小线性码构造了一个适用于一般存取结构的抗内存泄露的可验证多级秘密共享方案。肖健等[30]提出了一种基于多答案保护的弹性秘密共享方案。该方案将一个秘密分发给不同的有多密保问题的服务器。在秘密的重构阶段，用户只需向达到阈值的部分服务器提供达到阈值的部分密保问题的答案就能重构该秘密信息。

1.3 份额转换

份额转换算法是一种确保参与方在不了解其他参与方数据集内容的情况下完成数据集洗牌并实现秘密共享的算法。为体现份额转换算法在基于秘密共享的洗牌协议中的作用，现构造如下的假设环境。假设有2个参与方p_0和p_1，p_0执行洗牌S，p_1拥有数据集$\boldsymbol{x}$，他们想获得数据集$\boldsymbol{x}$的洗牌共享。抛开隐私性要求，可以由p_0直接对数据集$\boldsymbol{x}$进行洗牌，然后将洗牌后的数据集进行秘密共享，但是这种方法造成了p_1隐私信息的泄露。

份额转换算法最早由Chase等[4]于2020年的亚洲密码学会议上提出，该算法较好地解决了参与方隐私泄露的问题，其具体实现流程如下。

1) p_0拥有洗牌S和$\boldsymbol{c}=S(\boldsymbol{a})\oplus\boldsymbol{b}$，$p_1$拥有数据集$\boldsymbol{x}$及随机掩码$\boldsymbol{a}$和$\boldsymbol{b}$。

2) p_1将$\boldsymbol{x}\oplus\boldsymbol{a}$发送给$p_0$，并设置自己的秘密份额为$\boldsymbol{b}$。

3) p_0设置自己的秘密份额为$S(\boldsymbol{x}+a)\oplus\boldsymbol{C}$，简化后为$S(\boldsymbol{x})\oplus\boldsymbol{b}$。

4) p_0和p_1联合即可恢复出洗牌数据集$S(\boldsymbol{x})$。

2 相关模型

根据文献[31]，基于加性秘密共享的洗牌协议的构建可分为两部分，分别为系统模型和安全模型。

2.1 符号说明

为确保相关模型以及后续协议内容表述的准确

性和一致性，本节首先介绍了本文协议所涉及的符号定义，如表1所示。

表1　符号定义

符号	定义
$\{p_0,p_1\}$	协议的参与方
$S_i(\boldsymbol{x})$	p_i对数据集$\boldsymbol{x}$的洗牌
D	单边洗牌协议
DD	运行2次D的基于秘密共享的可验证分解洗牌协议
N	数据集的长度
w	数据集每个元素的位数
$\boldsymbol{a},\boldsymbol{b},\boldsymbol{c},\boldsymbol{v}$	有N个元素的向量
$\boldsymbol{v}[i]$	$\boldsymbol{v}$的第i个元素

2.2　系统模型

基于加性秘密共享的洗牌协议的系统模型为其形式化的定义。本文参考文献[31]给出其形式化定义，作为基于加性秘密共享的洗牌协议的研究基础。

定义1　基于加性秘密共享的洗牌协议由单边洗牌协议执行2次组成。

在单边洗牌协议中，一方进行洗牌S，另一方则提供数据集$\boldsymbol{x}$，输出是洗牌数据的秘密共享。

$$F_{D[N,q]}(S,\boldsymbol{x})=(\boldsymbol{r},S(\boldsymbol{x})-\boldsymbol{r}) \tag{1}$$

在基于加性秘密共享的洗牌协议中，洗牌由参与方共同执行，即$S=S^0S^1$，那么函数可以定义为

$$F_{\mathrm{DD}[N,q]}(\boldsymbol{x}_0,\boldsymbol{x}_1)=(\boldsymbol{r},S(\boldsymbol{x}_0+\boldsymbol{x}_1)-\boldsymbol{r}) \tag{2}$$

定义2　单边洗牌协议的系统模型主要包括4个部分，分别为洗牌协议的初始化、洗牌分解、洗牌份额转换和洗牌份额生成，相关功能如图1所示，具体描述如下。

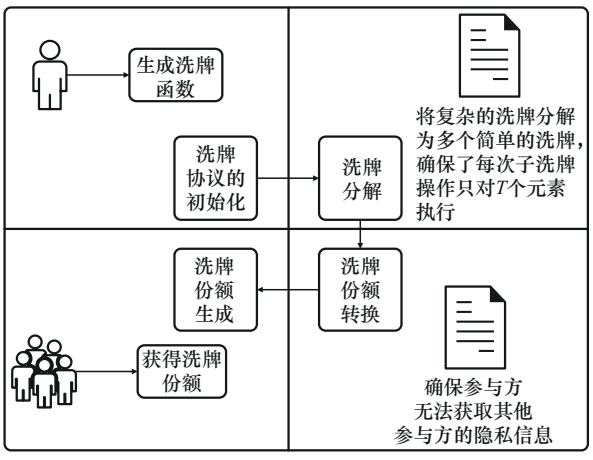

图1　单边洗牌协议的系统模型

1) 洗牌协议的初始化：输入洗牌参数k，通过洗牌算法输出对应的洗牌函数S。

2) 洗牌分解：输入洗牌函数S，数据集长度N，输出洗牌组合$S=S^1\cdots S^d$。

3) 洗牌份额转换：输入洗牌函数S，输出$(\boldsymbol{c},(\boldsymbol{a},\boldsymbol{b}))$。

4) 洗牌份额生成：输入洗牌函数S，输出p_0和p_1的洗牌份额，p_0和p_1可联合恢复出洗牌数据集。

基于加性秘密共享的洗牌协议的系统模型定义如下。

定义3　基于秘密共享的可验证分解洗牌协议主要包括洗牌协议的初始化、2个单边洗牌协议的执行和洗牌份额生成，具体描述如下。

1) 洗牌协议的初始化：输入洗牌参数k_1和k_2，通过洗牌算法输出洗牌对应的洗牌函数S_1和S_2。

2) 2个单边洗牌协议的执行：输入一方的洗牌S和另一方的数据集$\boldsymbol{x}$，输出p_0和p_1的洗牌份额。

3) 洗牌份额生成：输入洗牌函数S_0和S_1，输出p_0和p_1的洗牌份额，p_0和p_1可联合恢复出洗牌数据集。

2.3　安全模型

基于秘密共享的洗牌协议的安全模型为保证协议安全需要满足的定义，本节参考了文献[32]并给出了相关安全模型。由于协议的安全性与应用环境有着很大的关系，故先定义应用环境如下。

1) 参与方没有完全安全的通信信道。

2) 参与方只有有限的计算能力。

3) 参与方的性质是不变的，即参与方要么一直是恶意参与方，要么一直是半诚实参与方。

根据文献[33]可知，在基于秘密共享的洗牌协议中，参与方会遵循协议的规定步骤，但会尝试从接收到的信息中学习尽可能多的额外信息。因此，本文主要考虑半诚实模型下协议的安全性。

令$\prod$是一个两方协议，设置协议的安全参数为q，理想世界中存在一个函数F，模拟器sim可以接收参与方的输入并发送给函数F。半诚实模型下的两方协议评估函数F的安全性取决于如下理想–现实实验[34]。

在理想实验中，存在一个可信的第三方，可以接收所有参与方的输入，诚实地执行理想函数F的计算，然后将计算结果公布给所有参与方。理想实验描述如下。

$\mathrm{IDEAL}^{F}_{\mathrm{sim},b}(q,\boldsymbol{x}_0,\boldsymbol{x}_1)$。sim将$\boldsymbol{x}_0$和$\boldsymbol{x}_1$发送给函数$F$，并计算$F(\boldsymbol{x}_0,\boldsymbol{x}_1)$得到输出$(\boldsymbol{y}_0,\boldsymbol{y}_1)$，模拟器$\mathrm{sim}(1^q,b,\boldsymbol{x}_b,\boldsymbol{y}_b)$为参与方$p_b$产生了一个模拟视图$\mathrm{view}_b$。这个实验的输出为$(\mathrm{view}_b,\boldsymbol{y}_{1-b})$。

在现实实验中，参与方可执行两方协议$\prod$并与其他参与方进行交互。存在腐败方p_b可通过观察协议公开的信息来窃取其他参与方的隐私信息。现实实验描述如下。

$\mathrm{REAL}^{\prod}_{b}(q,\boldsymbol{x}_0,\boldsymbol{x}_1)$。在安全参数$q$下，两方协议$\prod$以$\boldsymbol{x}_0$和$\boldsymbol{x}_1$为输入，分别独立地输出$\boldsymbol{y}_0$和$\boldsymbol{y}_1$。这个实验的输出为$(\mathrm{view}_b,\boldsymbol{y}_{1-b})$。

定义4 如果存在一个概率多项式时间模拟器sim，使得对于所有的输入$\boldsymbol{x}_0$和$\boldsymbol{x}_1$以及腐败方$\boldsymbol{b}\in\{0,1\}$（p_b为腐败方），在理想实验$\mathrm{IDEAL}^{F}_{\mathrm{sim},b}(q,\boldsymbol{x}_0,\boldsymbol{x}_1)$与现实实验$\mathrm{REAL}^{\prod}_{b}(q,\boldsymbol{x}_0,\boldsymbol{x}_1)$中的输出是不可区分的，说明两方协议$\prod$在半诚实安全模型下是安全的。

3 基于加性秘密共享的洗牌协议的算法设计

本节主要对第5节洗牌协议的设计运用到的算法进行了构造，主要分为洗牌算法、份额转换算法和分解算法。

3.1 洗牌算法

洗牌算法是协议执行流程中参与方对数据集进行打乱的具体算法，这里用函数S表示。

本节提出了一种具有较高效率的洗牌算法。在该算法中，参与方可根据实际情况改变洗牌参数k的值，其具体的流程如下。

1) 输入长度为N的数据集$\boldsymbol{x}$和洗牌参数k。

2) 将$\boldsymbol{x}$分成$\boldsymbol{y}+1$段，每段元素数量为k，最后一段元素数量为$\mathrm{pend}\in[1,k]$，这些数据集可以表示为

$$\boldsymbol{x}=\{\boldsymbol{x}_1^1,\boldsymbol{x}_2^1,\cdots,\boldsymbol{x}_k^1,\cdots,\boldsymbol{x}_{\mathrm{pend}}^{y+1}\} \tag{3}$$

3) 将$\boldsymbol{x}_i^j$和$\boldsymbol{x}_i^{j+1}$按j升序排列，从而组成pend个数组。

4) 按i的值升序排列组成一个新的数组，该数组即为洗牌后的数据集。

通过上述的洗牌算法，参与方获得了一个可作用于数据集的洗牌函数S。

3.2 份额转换算法

本文借助不经意传输协议构建了份额转换算法，用于确保参与方无法获取其他参与方的隐私信息。

$\{\boldsymbol{v}_i[j]\}_{i,j\in N^2}$是一个$N\times N$的矩阵，份额转换算法的输入为$p_0$的洗牌，输出为$(\boldsymbol{c},(\boldsymbol{a},\boldsymbol{b}))$。在这个算法中，参与方$p_0$和$p_1$遵循如下规则。

1) p_0学习所有的元素除了$v_1[S(1)],\cdots,v_N[S(N)]$。

2) 学习整个矩阵，但是其不知道洗牌函数S。

份额转换算法的具体流程如下。

1) 参与方p_0和p_1并行的执行N次不经意传输协议，p_0用$S(i)$作为输入，p_0和p_1执行不经意传输协议后的输出分别为$\boldsymbol{v}_1$和$\boldsymbol{v}_1$。

2) 对于每个$i\in N$，p_0设置$\boldsymbol{c}[i]\leftarrow\sum_{j\neq S(i)}\boldsymbol{v}_i'[j]-\sum_{j\neq i}\boldsymbol{v}_j'[S(i)]$，并设置自己的输出为$\boldsymbol{c}=(\boldsymbol{c}[1],\cdots,\boldsymbol{c}[N])$。

3) 对于每个$i\in N$，p_1设置$\boldsymbol{a}_i$与$\boldsymbol{b}_i$分别为矩阵的列项和与行向和，即

$$\boldsymbol{b}_i=\sum_j\boldsymbol{v}_i[j],\boldsymbol{a}_i=\sum_j\boldsymbol{v}_j[i]。$$

p_1设置自己的输出为$(\boldsymbol{a},\boldsymbol{b})$，这里的$\boldsymbol{a}=(\boldsymbol{a}[1],\cdots,\boldsymbol{a}[N])$，$\boldsymbol{b}=(\boldsymbol{b}[1],\cdots,\boldsymbol{b}[N])$。综上所述，经过份额转换算法后，$p_0$输出$\boldsymbol{c}$，$p_1$输出$(\boldsymbol{a},\boldsymbol{b})$。

3.3 分解算法

分解算法将复杂的洗牌分解为多个简单的洗牌，确保了每次子洗牌操作只对T个元素执行。

3.2节提出的份额转换算法的运行时间与N^2成正相关。通过分解算法，本文协议使得份额转换算法的作用对象由原来大数据集的洗牌转换成多个互不相交的小数据集的洗牌。在提高大规模数据处理效率中具有重要的作用。

对洗牌函数S执行分解算法后，可将S分为几个互不相交的洗牌组合。

$$S=S^1S^2\cdots S^d \tag{4}$$

洗牌分解基于Benes排列网络，这个网络有$2\log N-1$层，每层有$\frac{N}{2}$个元素交换，故每一层都是一个排列。如果输入以索引$1,\cdots,N$编号，那么每个索引可以用二进制表示为$\sigma_1,\cdots,\sigma_n$，其具体步骤如下。

1) 令$T=2^t$，$t\in N$，$d=2\left[\frac{n}{t}\right]-1$，$S^1$由前$t$层

组成，S^2 由 $St+1,\cdots,2t$ 组成，依次类推，中间的洗牌函数 $S^{\left\lfloor\frac{d}{2}\right\rfloor+1}$ 容纳了 $2t-1$ 层。

2) Benes 排列网络在每一组中只是排列元素 $\sigma_1,\cdots,\sigma_{i(t-1)},\sigma_{i(t+1)},\cdots,\sigma_n$，这里的 x 包含所有的 t 长度字符串。剩余的 $n-t$ 位 $\sigma_1,\cdots,\sigma_{i(t-1)},\sigma_{i(t+1)},\cdots,\sigma_n$ 是固定的。

3) 对于每个子洗牌函数 S^i，$i\neq\left\lfloor\frac{d}{2}\right\rfloor+1$，都容纳了 $2^{n-t}=\frac{N}{T}$ 个不相交的洗牌，每个洗牌都作用于 $T=2^t$ 个元素。对于中间的洗牌函数 $S^{\left\lfloor\frac{d}{2}\right\rfloor+1}$，容纳了 $2t-1$ 层，只洗牌了每一组的 $\sigma_1,\cdots,\sigma_{n-t}x$，因此也可以被表示为 $\frac{N}{T}$ 不相交的洗牌组合，每个洗牌作用于 T 个元素。

4　基于加性秘密共享的洗牌协议的协议设计

4.1　单边洗牌协议的设计

单边洗牌协议本质上构成了基于秘密共享机制的洗牌协议流程的一半。在该协议框架中，其中一个参与方承担洗牌操作 S，而另一个参与方负责分发数据份额 x。单边洗牌协议的具体实施步骤如下。

1) 洗牌协议的初始化。输入洗牌参数 k，通过 3.1 节提出的洗牌算法输出对应的洗牌函数 S。

2) 洗牌分解。输入洗牌函数 S，数据集长度 N。参与方 p_0 通过 3.3 节提出的分解算法输出洗牌的分解 $S_0=S_0^1\cdots S_0^d$。

3) 洗牌份额转换。对于每个洗牌函数 S_i，参与双方依据 3.2 节提出的份额转换算法，执行 $\frac{N}{T}$ 次份额转换算法。

对于每个 i，p_1 通过 3.2 节提出的份额转换算法得到了 $\boldsymbol{a}^{(i,1)},\cdots,\boldsymbol{a}^{(i,\frac{N}{T})}$ 和 $\boldsymbol{b}^{(i,1)},\cdots,\boldsymbol{b}^{(i,\frac{N}{T})}$，分别简记为向量 $\boldsymbol{a}^{(i)}$ 和向量 $\boldsymbol{b}^{(i)}$。

利用向量 $\boldsymbol{a}^{(i)}$ 和向量 $\boldsymbol{b}^{(i)}$，参与方 p_0 可通过式(5)得到 $\boldsymbol{c}^{(i,1)},\cdots,\boldsymbol{c}^{(i,\frac{N}{T})}$，简记为向量 $\boldsymbol{c}^{(i)}$。

$$\boldsymbol{c}^{(i)}=\boldsymbol{b}^{(i)}-S_0^i(\boldsymbol{a}^{(i)}) \tag{5}$$

4) 洗牌份额生成。利用上面得到的向量 $\boldsymbol{a}^{(i)}$、$\boldsymbol{b}^{(i)}$ 和 $\boldsymbol{c}^{(i)}$，对于每个 $i\in 1,\cdots,d-1$，p_1 计算 $\boldsymbol{\delta}^{(i)}$ 并将其发送给 p_0。

$$\boldsymbol{\delta}^{(i)}=\boldsymbol{a}^{(i+1)}-\boldsymbol{b}^{(i)} \tag{6}$$

p_1 还发送 $\boldsymbol{m}=\boldsymbol{x}+\boldsymbol{a}^{(1)}$，采样并发送一个随机的 $\boldsymbol{w}$。p_1 的输出（即洗牌份额）为 $\boldsymbol{b}=\boldsymbol{w}-\boldsymbol{b}^{(d)}$。$p_0$ 计算如式(7)所示。

$$\begin{aligned}\boldsymbol{c}=\boldsymbol{c}^{(d)}+S_0^d(&\boldsymbol{\delta}^{(d-1)}+\boldsymbol{c}^{(d-1)}+S_0^{d-1}\\&(\boldsymbol{\delta}^{(d-2)}+\boldsymbol{c}^{(d-2)}+\cdots+S_0^2(\boldsymbol{\delta}^{(1)}+\boldsymbol{c}^{(1)})))\end{aligned} \tag{7}$$

并输出 $S(\boldsymbol{m})+\boldsymbol{c}-\boldsymbol{w}$，这也是 p_0 的洗牌份额。

通过上述提出的基于秘密共享的单边洗牌协议，p_0 得到了洗牌份额 $S(\boldsymbol{m})+\boldsymbol{c}-\boldsymbol{w}$，$p_1$ 得到了洗牌份额 $\boldsymbol{b}=\boldsymbol{w}-\boldsymbol{b}^{(d)}$。实际上，此时 p_0 和 p_1 可联合恢复出洗牌数据集 $S(\boldsymbol{x})$。在 5.1 节，将结合 3.2 节中的正确性定义，对本文协议的正确性进行分析。

4.2　双边洗牌协议的设计

在本文设计的基于加性秘密共享的洗牌协议中，参与方 p_0 和 p_1 各拥有一个数据集 $\boldsymbol{x}_0$ 和 $\boldsymbol{x}_1$，且各拥有一次洗牌的机会 S_0 和 S_1。p_0 和 p_1 想在双方数据集的交集上计算一些函数，且不想给彼此泄露数据集交集的内容。

在上述安全多方计算的要求下，参与方各自的数据集 $\boldsymbol{x}_0$ 和 $\boldsymbol{x}_1$ 需要保密，数据集交集的内容也需要保密，那么参与方需要得到的是洗牌数据集的秘密份额。根据 2.2 节系统模型的定义，基于加性秘密共享的洗牌协议的具体实施步骤如下。

1) 洗牌协议的初始化。p_0 和 p_1 输入洗牌参数 k_1 和 k_2，通过 3.1 节提出的洗牌算法分别确定各自的洗牌函数 S_1 和 S_2。

2) 第一个基于秘密共享的单边洗牌协议的执行。p_0 和 p_1 运行 4.1 节提出的单边洗牌协议对 $\boldsymbol{x}_1$ 应用洗牌函数 S_0，使得 p_0 获得 $\boldsymbol{x}_0^{(1)}$，p_1 获得 $\boldsymbol{x}_1^{(1)}$；然后 p_0 计算 $\boldsymbol{x}_0^{(2)}=S_0(\boldsymbol{x}_0)+\boldsymbol{x}_0^{(1)}$。

3) 第二个基于秘密共享的单边洗牌协议的执行。p_0 和 p_1 运行 4.1 节提出的单边洗牌协议对 $\boldsymbol{x}_0^{(2)}$ 应用洗牌函数 S_1，使得 p_0 获得 $\boldsymbol{x}_0^{(3)}$，p_1 获得 $\boldsymbol{x}_1^{(3)}$；然后 p_1 计算 $\boldsymbol{x}_1^{(4)}=S_1(\boldsymbol{x}_1^{(1)})+\boldsymbol{x}_1^{(3)}$。

4) 洗牌份额生成。p_0 输出 $\boldsymbol{x}_0^{(3)}$，p_1 输出 $\boldsymbol{x}_1^{(4)}$。

通过上述提出的基于秘密共享的可验证分解洗牌协议，p_0 获得洗牌份额 $\boldsymbol{x}_0^{(3)}$，p_1 获得洗牌份额 $\boldsymbol{x}_1^{(4)}$。实际上，此时 p_0 和 p_1 可联合恢复出洗牌数据集 $S_1(S_0(\boldsymbol{x}_0+\boldsymbol{x}_1))$。在 5.1 节，将结合 2.2 节中的正确性定义，对该协议的正确性进行分析。

5 洗牌协议的相关分析

5.1 正确性分析

根据文献[32]可知，基于秘密共享的洗牌协议满足正确性的标准为参与方可以在不泄露隐私信息的前提下，联合恢复出洗牌数据集。

定理1 单边洗牌协议使得参与方能够联合恢复出洗牌数据集，满足正确性分析。

证明 因为对于任意的i，都有$\boldsymbol{c}^{(i)}=\boldsymbol{b}^{(i)}-S_0^i(\boldsymbol{a}^{(i)})$。这意味着对于任意的$i$，都有

$$\begin{aligned}&\boldsymbol{\delta}^{(i)}+\boldsymbol{c}^{(i)}=\\&\boldsymbol{a}^{(i+1)}-\boldsymbol{b}^{(i)}+\boldsymbol{b}^{(i)}-S_0^i(\boldsymbol{a}^{(i)})=\\&\boldsymbol{a}^{(i+1)}-S_0^i(\boldsymbol{a}^{(i)})\end{aligned}\tag{8}$$

因此，最终参与方p_0产生的$\boldsymbol{c}$表示为

$$\begin{aligned}&\boldsymbol{c}^{(d)}+S_0^d(\boldsymbol{\delta}^{(d-1)}+\boldsymbol{c}^{(d-1)}+S_0^{d-1}(\boldsymbol{\delta}^{(d-2)}+\boldsymbol{c}^{(d-2)}+\\&\cdots+S_0^2(\boldsymbol{\delta}^{(1)}+\boldsymbol{c}^{(1)})))=\\&\boldsymbol{c}^{(d)}+S_0^d(\boldsymbol{a}^{(d)}-S_0^{d-1}(\boldsymbol{a}^{(d-1)})+\\&S_0^{d-1}(\boldsymbol{a}^{(d-1)}-S_0^{d-2}(\boldsymbol{a}^{(d-2)})+\cdots+S_0^2(\boldsymbol{a}^{(1)})))=\\&\boldsymbol{c}^{(d)}+S_0^d(\boldsymbol{a}^{(d)}-S_0^{d-1}(\cdots S_0^2(S_0^1(\boldsymbol{a}^{(1)}))))=\\&\boldsymbol{b}^{(d)}-S_0^d(\boldsymbol{a}^{(d)})+S_0^d(\boldsymbol{a}^{(d)}-S_0^{d-1}(\cdots S_0^2(S_0^1(\boldsymbol{a}^{(1)}))))=\\&\boldsymbol{b}^{(d)}-S_0^d(S_0^{d-1}(\cdots S_0^2(S_0^1(\boldsymbol{a}^{(1)}))))=\\&\boldsymbol{b}^{(d)}-S_0(\boldsymbol{a}^{(1)})\end{aligned}\tag{9}$$

而参与方p_0与p_1的输出（秘密份额）分别为

$$\begin{aligned}&S_0(\boldsymbol{m})+\boldsymbol{c}-\boldsymbol{w}=\\&S_0(\boldsymbol{x}+\boldsymbol{a}^{(1)})+\boldsymbol{c}-\boldsymbol{w}=\\&S_0(\boldsymbol{x})+S_0(\boldsymbol{a}^{(1)})+\boldsymbol{c}-\boldsymbol{w}\end{aligned}\tag{10}$$

$$\begin{aligned}&\boldsymbol{w}-\boldsymbol{b}^{(d)}=\\&\boldsymbol{w}-(\boldsymbol{c}+S_0(\boldsymbol{a}^{(1)}))=\\&\boldsymbol{w}-\boldsymbol{c}-S_0(\boldsymbol{a}^{(1)})\end{aligned}\tag{11}$$

可见参与方p_0与p_1的输出（秘密份额）联合可恢复出洗牌数据集$S_0(\boldsymbol{x})$。

证毕。

定理2 本文设计的基于秘密共享的洗牌协议使得参与方联合能够恢复出洗牌数据集$S_1(S_0(\boldsymbol{x}_0+\boldsymbol{x}_1))$，满足正确性分析。

证明 p_0输出$\boldsymbol{x}_0^{(3)}$，也是它的秘密份额，可进一步化简为

$$\begin{aligned}&\boldsymbol{x}_0^{(3)}=\\&S_1(\boldsymbol{x}_0^{(2)})-\boldsymbol{r}^{(3)}=\\&S_1(S_0(\boldsymbol{x}_0)+\boldsymbol{x}_0^{(1)})-\boldsymbol{r}^{(3)}=\\&S_1(S_0(\boldsymbol{x}_0)+\boldsymbol{r}^{(1)})-\boldsymbol{r}^{(3)}=\\&S_1(S_0(\boldsymbol{x}_0))+S_1(\boldsymbol{r}^{(1)})-\boldsymbol{r}^{(3)}\end{aligned}\tag{12}$$

p_1输出$\boldsymbol{x}_1^{(4)}$，也是它的秘密份额，可进一步化简为

$$\begin{aligned}&\boldsymbol{x}_1^{(4)}=\\&S_1(\boldsymbol{x}_1^{(1)})+\boldsymbol{x}_1^{(3)}=\\&S_1(\boldsymbol{x}_1^{(1)})+\boldsymbol{r}^{(3)}=\\&S_1(S_0(\boldsymbol{x}_1)-\boldsymbol{r}^{(1)})+\boldsymbol{r}^{(3)}=\\&S_1(S_0(\boldsymbol{x}_1))-S_1(\boldsymbol{r}^{(1)})+\boldsymbol{r}^{(3)}\end{aligned}\tag{13}$$

将上述p_0和p_1的输出联合后，可恢复出洗牌数据集$S_1(S_0(\boldsymbol{x}_0+\boldsymbol{x}_1))$。

证毕。

定理3 本文协议提出的洗牌算法满足正确性需求。

证明 设N为洗牌前的元素总数。本文提出的洗牌算法每次从两个半部分选择一个元素，然后交替排列，元素的总数保持不变。通过引入洗牌参数k，确保洗牌后的结果是足够随机的。在洗牌剩余的后半部分，由于其数目不足以构成一个长度为k的数组，故使用哈希函数生成哈希值。对于不同的元素，哈希函数返回不同的哈希值，确保了在洗牌后每个元素都有一个唯一的位置。

证毕。

5.2 安全性分析

根据2.3节安全模型所提出，如果一个基于秘密共享的洗牌协议在2.3节所提出的应用环境中是安全的，其需要满足定义4的要求，即模拟器能产生与现实实验无异的理想实验结果。接下来，本节将从p_0为腐败方和p_1为腐败方2个方面，验证现实实验和理想实验的输出是否一致。

当$b=0$时，即p_0为腐败方，模拟器$\mathrm{sim}(1^q,0,\boldsymbol{x}_0,\boldsymbol{y}_0)$将选择一个$S_0$和$\boldsymbol{x}_0^{(1)}$，设置$\boldsymbol{x}_0^{(2)}=S_0(\boldsymbol{x}_0)+\boldsymbol{x}_0^{(1)}$，模拟第一个单边洗牌协议为$\mathrm{sim}^D(1^q,0,S_0,\boldsymbol{x}_0^{(1)})$，模拟第二个单边洗牌协议为$\mathrm{sim}^D(1^q,0,\boldsymbol{x}_0^{(2)},\boldsymbol{y}_0)$。

当$b=1$时，即p_1为腐败方，模拟器$\mathrm{sim}(1^q,1,\boldsymbol{x}_1,\boldsymbol{y}_1)$将选择一个$S_1$和$\boldsymbol{x}_1^{(1)}$，设置$\boldsymbol{x}_1^{(3)}=\boldsymbol{y}_1-S_1(\boldsymbol{x}_1^{(1)})$，模拟第一个单边洗牌协议为$\mathrm{sim}^D(1^q,1,\boldsymbol{x}_1,\boldsymbol{x}_1^{(1)})$，模拟第二个单边洗牌协议为$\mathrm{sim}^D(1^q,1,s_1,\boldsymbol{x}_1^{(3)})$。

定理4 模拟器能产生与实际实验无异的理想实验结果。

证明

1) 当$b=0$时，运行现实实验，输出的是p_0的视图（它的输入是$\boldsymbol{x}_0$），来自2个单边洗牌协议的$\mathrm{view}_0^{(1)}$和$\mathrm{view}_0^{(2)}$包含了输出$\boldsymbol{x}_0^{(1)}$和$\boldsymbol{x}_0^{(3)}$，诚实方p_1的输入是$\boldsymbol{x}_1$，输出是$\boldsymbol{x}_1^{(4)}=S_1(\boldsymbol{x}_1^{(1)})+\boldsymbol{x}_1^{(3)}$。

① 游戏1。在4.2节的2)中，首先计算$F_D(S_0,\boldsymbol{x}_1)$，

例如，随机选择一个$\boldsymbol{r}^{(1)}$，然后令$\boldsymbol{x}_0^{(1)}=\boldsymbol{r}^{(1)}$，$\boldsymbol{x}_1^{(1)}=S_0(\boldsymbol{x}_1)-\boldsymbol{r}^{(1)}$，然后运行单边洗牌模拟器产生第一个视图$\text{view}_0^{(1)\prime}$。输出是$p_0$的视图（它的输入是$\boldsymbol{x}_0$，来自2个单边洗牌协议的$\text{view}_0^{(1)\prime}$和$\text{view}_0^{(2)\prime}$包含了输出$\boldsymbol{x}_0^{(1)}=\boldsymbol{r}^{(1)}$和$\boldsymbol{x}_0^{(3)}$，诚实方$p_1$的输入是$\boldsymbol{x}_1$，输出是$\boldsymbol{x}_1^{(4)}=S_1(\boldsymbol{x}_1^{(1)})+\boldsymbol{x}_1^{(3)}=S_1(S_0(\boldsymbol{x}_1)-\boldsymbol{r}^{(1)})+\boldsymbol{r}^{(3)}$与现实实验不可区分。

②游戏2。在4.2节的3)中，首先计算$F_D(S_1,\boldsymbol{x}_0^{(2)})$，例如，随机选择一个$\boldsymbol{r}^{(3)}$，然后令$\boldsymbol{x}_1^{(3)}=\boldsymbol{r}^{(3)}$，$\boldsymbol{x}_0^{(3)}=S_1(\boldsymbol{x}_0^{(2)})-\boldsymbol{r}^{(3)}$，然后运行单边洗牌模拟器产生第一个视图$\text{view}_0^{(2)\prime}$。输出是$p_0$的视图（它的输入是$\boldsymbol{x}_0$），来自2个单边洗牌协议的$\text{view}_0^{(1)\prime}$和$\text{view}_0^{(2)\prime}$包含了输出$\boldsymbol{x}_0^{(1)}=\boldsymbol{r}^{(1)}$和$\boldsymbol{x}_0^{(3)}=S_1(\boldsymbol{x}_0^2)-\boldsymbol{r}^{(3)}$，诚实方$p_1$的输入是$\boldsymbol{x}_1$，输出$\boldsymbol{x}_1^{(4)}=S_1(S_0(\boldsymbol{x}_1)-\boldsymbol{r}^{(1)})+\boldsymbol{x}_1^{(3)}=S_1(S_0(\boldsymbol{x}_1)-\boldsymbol{r}^{(1)})+\boldsymbol{r}^{(3)}$

与现实实验不可区分。

③游戏3。选择S、r和$\boldsymbol{x}_0^{(1)}$，令$S_1=SS_0^{-1}$、$\boldsymbol{x}_0^{(1)}=\boldsymbol{r}^{(1)}$和$\boldsymbol{r}^{(3)}=S_I(S_0(\boldsymbol{x}_0))+S_1(\boldsymbol{r}^{(1)})-\boldsymbol{r}$，除此之外，继续进行游戏2。输出是$p_0$的视图（它的输入是$\boldsymbol{x}_0$），来自2个单边洗牌协议的$\text{view}_0^{(1)\prime}$和$\text{view}_0^{(2)\prime}$包含了输出$\boldsymbol{x}_0^{(1)}=\boldsymbol{r}^{(1)}$和$\boldsymbol{x}_0^{(3)}$，诚实方$p_1$的输入是$\boldsymbol{x}_1$，输出是$\boldsymbol{x}_1^{(4)}$。由定理4可知

$$\boldsymbol{x}_1^{(4)}=S_1(S_0(\boldsymbol{x}_1)-\boldsymbol{r}^{(1)})+\boldsymbol{r}^{(3)} \tag{14}$$

可化为

$$\begin{aligned}&\boldsymbol{x}_1^{(4)}=S_1(S_0(\boldsymbol{x}_1)-\boldsymbol{x}_0^{(1)})+\\&S_1(S_0(\boldsymbol{x}_1))+S_1(\boldsymbol{r}^{(1)})-\boldsymbol{x}_0^{(3)}=\\&S(\boldsymbol{x}_1+\boldsymbol{x}_0)-\boldsymbol{x}_0^{(3)}\end{aligned} \tag{15}$$

因此，当$b=0$时，现实实验与理想实验是不可区分的。

2) 当$b=1$时，运行现实实验。

①游戏1。在4.2节的2)中，首先计算$F_D(S_0,\boldsymbol{x}_1)$，例如，随机选择一个$\boldsymbol{x}_0^{(1)}$，然后计算$\boldsymbol{x}_1^{(1)}=S_0(\boldsymbol{x}_1)-\boldsymbol{x}_0^{(1)}$，然后运行单边洗牌模拟器产生第一个视图。与现实实验不可区分。

②游戏2。在4.2节的3)中，首先计算$F_D(S_1,\boldsymbol{x}_0^{(2)})$，例如，随机选择一个$\boldsymbol{x}_1^{(3)}$，然后计算$\boldsymbol{x}_0^{(3)}=S_1(\boldsymbol{x}_0^2)-\boldsymbol{x}_1^{(3)}$，然后运行单边洗牌模拟器产生视图$\text{view}_0^{(2)\prime}$，这与现实实验不可区分。

③游戏3。选择一个随机的洗牌函数S，设$S_0=SS_1^{-1}$，$\boldsymbol{x}_1^{(3)}=S(\boldsymbol{x}_0+\boldsymbol{x}_1)-S_1(\boldsymbol{x}_1^{(1)})-\boldsymbol{x}_0^{(3)}$，这意味着$\boldsymbol{x}_1^{(4)}=S(\boldsymbol{x}_0+\boldsymbol{x}_1)-\boldsymbol{x}_0^{(3)}$。由于游戏2中的$\boldsymbol{x}_0^{(3)}=S_1(\boldsymbol{x}_0^2)-\boldsymbol{x}_1^{(3)}$可以化简为

$$S_1(S_0(\boldsymbol{x}_0)+\boldsymbol{x}_0^{(1)})-\boldsymbol{x}_0^{(3)} \tag{16}$$

也就等于

$$\begin{aligned}&S_1(S_0(\boldsymbol{x}_0+\boldsymbol{x}_1))-S_1(\boldsymbol{x}_1^{(1)})-\boldsymbol{x}_0^{(3)}=\\&S(\boldsymbol{x}_0+\boldsymbol{x}_1)-S_1(\boldsymbol{x}_1^{(1)})-\boldsymbol{x}_0^{(3)}\end{aligned} \tag{17}$$

这也与现实实验不可区分。

结果表明，该模拟器能产生与现实实验无异的理想实验结果。

证毕。

5.3　效率分析

本节主要对本文协议在通信开销和通信复杂度方面的效率进行分析。本文提出的基于加性秘密共享的洗牌协议采用了洗牌分解算法，旨在确保每次子洗牌操作仅对T个元素进行执行，从而增强了协议在处理大规模数据集时的效率。

经过测试验证，本文协议的T的最佳的参数取值范围为16~256，这为实际应用提供了重要的参考依据。单边洗牌协议一共运行$\frac{dN}{T}$次份额转换算法，其中$d=2\left\lfloor\frac{\log N}{\log T}\right\rfloor-1$。

在这些运算中，通信开销是一个关键的衡量指标，它反映了在协议执行过程中信息传输所需的资源消耗$\left(N\log N-\frac{N}{2}\right)(q+4w)+2Nw$。单边洗牌协议的通信开销为$(d+1)Nw$。单边洗牌协议的计算开销等同于并行运行$\frac{dN}{T}$次份额转换算法的开销。

除此之外，通信复杂度也是一个重要的衡量指标，它反映了协议执行过程中所涉及的信息传递量和传输质量。基于加性秘密共享的洗牌协议执行3轮，通信复杂度与$qN\log N+\frac{Nw\log N}{\log T}$成比例。

5.4　性能分析

本节将本文协议与文献[8]和文献[13]提出的协议进行了比较。

文献[8]采用了一种基于公钥的解决办法，并未对洗牌进行分解。这导致在处理元素较多的数据集时，其效率可能会受到一定程度的影响。其通信复杂度为$\left(N\log N-\frac{N}{2}\right)(q+4w)+2Nw$。文献[13]的通信复杂度与文献[8]处于同一个数量级。

相比之下，本文协议采用洗牌分解算法，确保了每次子洗牌操作只对T个元素执行，其通信复杂度与$qN\log N+\frac{Nw\log N}{\log T}$成比例。显而易见，当数据集元素较小时，本文协议并无明显优势，甚至通信开销因为洗牌分解算法的存在，要高于上述文献。但是当数据集元素较大时，本文协议在通信开销上具有一定的优势。

文献[13]集中于洗牌算法的设计，其单次执行的洗牌算法的通信复杂度为$O(N)$，与本文协议相当。与本文协议相比，文献[13]采用了一种多次循环的洗牌方法，这导致了在洗牌后数据的均匀性更高，从而增强了数据的安全性。然而，这种多次循环的洗牌方法也带来了整体开销的增加。

与之相比，本文提出的洗牌算法虽然在安全性方面略逊一筹，但在综合考虑安全性和性能的情况下仍然具有一定的优势。本文协议能够在保证数据安全的前提下，实现较高的数据处理效率，为安全计算提供了一种可行的解决方案。协议的常见对比分析如表2所示。

经过表2的对比分析可以看出，在数据集较小的情况下，本文协议与其他协议相比并没有显著的优势。然而，当处理的数据集较大时，本文协议的优点就凸显出来了，它具有更低的通信开销。

除了效率与安全性外，实用性是在制定协议时必须高度重视的方面。本文协议在这方面表现出色，因为它具有以下优点。

首先，本文协议具有与多种隐私保护计算技术的兼容性。这一特点使得本文协议不仅能够提升数据的安全性和隐私保护水平，同时也能够灵活应用于各种不同的场景和满足不同应用需求。

其次，本文协议广泛采用简便的异或操作。相较于使用公钥加密等复杂方法，异或操作的运算过程更为简洁和易于理解，使得该协议的实施更加高效。

各协议的实用性对比分析如表3所示。

表3 各协议的实用性对比分析

协议	实用性
文献[8]	采用公钥解决方案，偏向于数学逻辑，较为复杂
文献[13]	公钥解决方案，需要多次循环
本文协议	可与隐私交集计算紧密结合；多采用异或运算，通俗易懂

6 前景应用

本文介绍的洗牌协议采用加性秘密共享机制，在数据隐私保护及安全计算领域具有一定的应用潜力，可应用于协同过滤、随机存取存储器程序和大数据抽样。本节对本文协议在实际应用中的运作进行简要概述。

1) 协同过滤。协同过滤要求参与方在保证各自数据隐私性的同时，筛选出满足要求的交集元素。本文提出的基于加性秘密共享的洗牌协议可以将加密的数据元素与加密标识向量一起洗牌，它将数据元素和对应的标识位（表示元素是否属于交集）混合在一起，然后重新排列。洗牌确保了没有参与方知道具体哪个标识位对应原始数据集中的哪个元素。当加密标识向量被揭示出来时，所有参与方会知道有哪些标识位是1（表示元素在交集中）和0（表示元素不在交集中），但是由于他们不知道这些标识位对应原始数据集中的哪些元素，交集之外的元素可以被安全地丢弃。

2) 随机存取存储器程序。随机存取存储器程序安全计算的一个基本构件是不经意式随机访问存储器（ORAM, oblivious random access memory），它可以隐藏计算过程中的内存访问。初始化ORAM的一种简单方法是对每个输入项执行一次ORAM写操作，但成本较高。本文提出的基于加性秘密共享的洗牌协议可以提高其效率，参与方使用基于秘密共享的洗牌协议来对其条目进行置换，然后将它们存储为ORAM内存。与此同时，ORAM具有较多大元素，而本文协议在元素较大的情况下也具有一定的效率优势。

3) 大数据抽样。抽样是大数据分析的一项基本操作，其中一个关键要求就是所选取的数据集必

表2 协议的常见对比分析

协议	通信开销	安全性	洗牌算法
文献[8]	元素较多时高；元素较少时低	半诚实、恶意下的安全性	单次为$O(N)$，然而需要循环
文献[13]	元素较多时高；元素较少时低	半诚实下的安全性	单次为$O(N)$，然而需要循环未给出具体的洗牌算法
本文协议	元素较少时高；元素较多时低	半诚实下的安全性	通信复杂度为$O(N)$

须具有代表性，能够准确地反映整体数据的特征和趋势。在这方面，本文提出的基于加性秘密共享的洗牌协议的洗牌功能发挥了重要作用。通过对数据集进行打乱顺序，本文协议能够消除数据原本可能存在的特定排列模式，使得具有相似特征的数据不再聚集。经过洗牌处理后的数据集更具有随机性和代表性，能够更好地反映整体数据集的分布情况，从而提高了抽样结果的准确性和可信度。

7　结束语

本文设计了一种基于加性秘密共享的洗牌协议，充分利用了秘密共享、洗牌协议、份额转换等密码学基础理论，以确保协议的正确执行和数据安全。在协议的执行过程中，本文采用洗牌分解思路对协议进行优化，通过将洗牌过程分解成多个子洗牌，提高了本文协议在大规模数据下的执行效率。同时，本文提出了一种高效的洗牌算法，通过引入份额转换算法并加以改进，保证了协议的安全性。

与现有的学术成果相比，本文协议在安全性方面虽然并未表现出显著优势，但在实用性和大规模数据处理效率方面却具备了一定的优势。因此，本文协议在用户数据隐私保护方面具有一定的优势，同时具有较好的适用性，可满足大规模数据集处理的需求，也可应用于数据共享、隐私保护、安全计算等方面。

参考文献：

[1] YAO A C. Protocols for secure computations[C]//Proceedings of the 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982). Piscataway: IEEE Press, 1982: 160-164.

[2] 韩伟力, 宋鲁杉, 阮雯强, 等. 安全多方学习: 从安全计算到安全学习[J]. 计算机学报, 2023, 46(7): 1494-1512.
HAN W L, SONG L S, RUAN W Q, et al. Secure multi-party learning: from secure computation to secure learning[J]. Chinese Journal of Computers, 2023, 46(7): 1494-1512.

[3] CIAMPI M, ORLANDI C. Combining private set-intersection with secure two-party computation[C]//International Conference on Security and Cryptography for Networks. Berlin: Springer, 2018: 464-482.

[4] CHASE M, GHOSH E, POBURINNAYA O. Secret shared shuffle[J]. Cryptology ePrint Archive, 2020, 11(1): 342-372.

[5] LAUD P. Linear-time oblivious permutations for spdz[C]//International Conference on Cryptology and Network Security. Berlin: Springer 2021: 245-252.

[6] PINKAS B, SCHNEIDER T, ZOHNER M. Secure multiparty computation goes live[C]//Proceedings of the 2012 ACM Conference on Computer and Communications Security. New York: ACM Press, 2012: 605-616.

[7] ZHAO X X, LI L J, XUE G L, et al. Efficient anonymous message submission[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(2): 217-230.

[8] CHEN J X, LIU G, LIU Y N. Lightweight privacy-preserving raw data publishing scheme[J]. IEEE Transactions on Emerging Topics in Computing, 2021, 9(4): 2170-2174.

[9] ATTRAPADUNG N, HANAOAKA G, MATSUDA T, et al. Oblivious linear group actions and applications[C]//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. New York: ACM Press, 2021: 630-650.

[10] HAN F, ZHANG L, FENG H W, et al. Scape: scalable collaborative analytics system on private database with malicious security[C]//Proceedings of the 2022 IEEE 38th International Conference on Data Engineering (ICDE). Piscataway: IEEE Press, 2022: 1740-1753.

[11] BELORGEY M G, CARPOV S, DEFORTH K, et al. Manticore: a framework for efficient multiparty computation supporting real number and Boolean arithmetic[J]. Journal of Cryptology, 2023, 36(3): 31.

[12] 张艳硕, 满子琪, 刘冰. 基于秘密共享的洗牌协议的对比分析[J]. 北京电子科技学院学报, 2023, 31(2): 10-19.
ZHANG Y S, MAN Z Q, LIU B. Comparative analysis of shuffling agreement based on secret sharing[J]. Journal of Beijing Electronic Science and Technology Institute, 2023, 31(2): 10-19.

[13] LIANG J T, SAGAN B E, ZHUANG Y. Cyclic shuffle-compatibility via cyclic shuffle algebras[J]. Annals of Combinatorics, 2024, 28(2): 615-654.

[14] PRANAV SHRIRAM A, KOTI N, KUKKALA V B, et al. Ruffle: rapid 3-party shuffle protocols[J]. Proceedings on Privacy Enhancing Technologies, 2023, 2023(3): 24-42.

[15] 满子琪, 张艳硕, 严梓洋, 等. 基于弹性秘密共享的多方洗牌协议[J]. 信息安全研究, 2024, 10(4): 347-352.
MAN Z Q, ZHANG Y S, YAN Z Y, et al. Multi-party shuffling protocol based on elastic secret sharing[J]. Journal of Information Security Research, 2024, 10(4): 347-352.

[16] SINGH H, SINHA A. A blockchain framework for E-voting[J]. Multimedia Tools and Applications, 2024, 83(20): 58875-58889.

[17] 陈宁江, 练林明, 欧平杰, 等. 基于图协同过滤模型的D2D协作缓存策略[J]. 通信学报, 2023, 44(7): 136-148.
CHEN N J, LIAN L M, OU P J, et al. D2D cooperative caching strategy based on graph collaborative filtering model[J]. Journal on Communications, 2023, 44(7): 136-148.

[18] NARASIMHULU K, ABARNA K T M, KUMAR B S, et al. A novel sampling-based visual topic models with computational intelligence for big social health data clustering[J]. The Journal of Supercomputing, 2022, 78(7): 9619-9641.

[19] 刘艺菲, 王宁, 王志刚, 等. 混洗差分隐私下的多维类别数据的收集与分析[J]. 软件学报, 2022, 33(3): 1093-1110.
LIU Y F, WANG N, WANG Z G, et al. Collecting and analyzing multidimensional categorical data under shuffled differential privacy[J]. Journal of Software, 2022, 33(3): 1093-1110.

[20] 陈景雪, 高克寒, 周尔强, 等. 物联网环境下鲁棒的源匿名联邦学习洗牌协议[J]. 计算机研究与发展, 2023, 60(10): 2218-2233.
CHEN J X, GAO K H, ZHOU E Q, et al. Robust source anonymous federated learning shuffle protocol in IoT[J]. Journal of Computer Research and Development, 2023, 60(10): 2218-2233.

[21] LI C L, CAI Q Q, LUO Y L. Data balancing-based intermediate data partitioning and check point-based cache recovery in spark environment[J]. The Journal of Supercomputing, 2022, 78(3): 3561-3604.
[22] CHAUM D L. Untraceable electronic mail, return addresses, and digital pseudonyms[J]. Communications of the ACM, 1981, 24(2): 84-90.
[23] 刘涵阅, 张春生. 基于洗牌算法的大数据抽样有效性分析[J]. 计算机应用研究, 2021, 38(10): 3049-3054.
LIU H Y, ZHANG C S. Analysis of sampling effectiveness of big data based on shuffling algorithm[J]. Application Research of Computers, 2021, 38(10): 3049-3054.
[24] JHO N S, LEE J. Partition and mix: generalizing the swap-or-not shuffle[J]. Designs, Codes and Cryptography, 2023, 91(6): 2237-2254.
[25] LI J, MAKKONEN O, GNILKE H O W. Efficient recovery of a shared secret via cooperation: applications to SDMM and PIR[J]. IEEE Journal on Selected Areas in Communications, 2022, 40(3): 871-884.
[26] ZHANG E, LI M, YIU S M, et al. Fair hierarchical secret sharing scheme based on smart contract[J]. Information Sciences, 2021, 546: 166-176.
[27] SHAMIR A. How to share a secret (1979)[J]. Communications of the ACM, 1979, 22(11): 612-613.
[28] 张剑, 林昌露, 黄可可, 等. 基于多项式插值的多等级秘密共享方案[J]. 密码学报, 2022, 9(4): 743-754.
ZHANG J, LIN C L, HUANG K K, et al. Polynomial interpolation based hierarchical secret sharing schemes[J]. Journal of Cryptologic Research, 2022, 9(4): 743-754.
[29] 宋云, 李志慧, 王文华. 一般存取结构上抗内存泄露的多级秘密共享[J]. 软件学报, 2022, 33(10): 3891-3902.
SONG Y, LI Z H, WANG W H. Memory leakage-resilient multi-stage secret sharing scheme with general access structures[J]. Journal of Software, 2022, 33(10): 3891-3902.
[30] 肖健, 杨敏, 孟庆树. 多答案保护秘密共享协议[J]. 武汉大学学报(理学版), 2023, 69(1): 51-59.
XIAO J, YANG M, MENG Q S. Multi-answer protected secret sharing protocol[J]. Journal of Wuhan University (Natural Science Edition), 2023, 69(1): 51-59.
[31] 李顺东, 王文丽, 陈明艳, 等. 抗主动攻击的保密比较协议[J]. 软件学报, 2022, 33(12): 4771-4783.
LI S D, WANG W L, CHEN M Y, et al. Comparing protocol against active attacks[J]. Journal of Software, 2022, 33(12): 4771-4783.
[32] 李超, 王健, 刘吉强. 基于区块链的轻量级匿名评审协议[J]. 信息安全学报, 2022, 7(5): 91-107.
LI C, WANG J, LIU J Q. Blockchain-based lightweight anonymous review system[J]. Journal of Cyber Security, 2022, 7(5): 91-107.
[33] JACK P K M, SHERMAN S M. CHOW. Secure-computation-friendly private set intersection from oblivious compact graph evaluation[C]// Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. New York: ACM press, 2022: 1086-1097.
[34] CANETTI R. Universally composable security: a new paradigm for cryptographic protocols[C]//Proceedings 42nd IEEE Symposium on Foundations of Computer Science. Piscataway: IEEE Press, 2001: 136-145.

[作者简介]

张艳硕（1979– ），男，陕西宝鸡人，博士，北京电子科技学院副教授、硕士生导师，主要研究方向为密码理论及其应用。

满子琪（2000– ），男，江苏宿迁人，北京电子科技学院硕士生，主要研究方向为密码理论及其应用。

周幸妤（2000– ），女，江苏镇江人，北京电子科技学院硕士生，主要研究方向为密码理论及其应用。

杨亚涛（1978– ），男，河南平顶山人，北京电子科技学院教授、博士生导师，主要研究方向为信息安全、同态加密、密码协议和算法。

胡荣磊（1977– ），男，河北衡水人，北京电子科技学院副研究员，主要研究方向为密码芯片安全、隐私保护与隐私计算、网络安全、物联网、区块链等。