# 基于秘密共享的可验证分层洗牌协议设计及其应用方案

张艳硕[1], 满子琪[1], 周幸妤[1], 杨亚涛[2], 谢绒娜[1]

[1]北京电子科技学院密码科学与技术系 北京 中国 100070
[2]北京电子科技学院电子与通信工程系 北京 中国 100070

**摘要** 现有的基于秘密共享的洗牌协议存在着一些问题, 例如: 多集中于理论框架的设计, 缺少每一流程实现的具体算法; 多采用公钥的解决方案, 在处理大规模数据集时效率不是很高; 缺乏一定的适用性, 在一些应用领域不是很实用。鉴于这些局限, 本文设计了一种基于秘密共享的可验证分层洗牌协议。与此同时, 为了结合具体应用场景, 本文还设计了一种基于洗牌协议的隐私保护方案。本协议通过不经意传输协议构建改进的份额转换算法, 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层, 将复杂的洗牌任务分为多个简易的子任务, 提高了大规模数据下的效率; 引入可验证的思想, 从而使协议的安全性得到了有效保证。本文对所提出协议的正确性进行了严格分析; 运用理想-现实模拟范式对安全性进行了评估; 并与相应的协议在时间开销、安全性、算法时间复杂度等方面进行了对比分析。结果表明, 本文提出的基于秘密共享的可验证分层洗牌协议能够满足恶意模型下的安全性标准; 在处理大规模数据集时有一定的效率优势; 提升了协议的适用性, 进一步推广了在当下环境中的应用。

**关键词** 秘密共享; 洗牌; 安全多方计算; 隐私保护
**中图法分类号** TN92 DOI 号 10.19363/J.cnki.cn10-1380/tn.2026.03.11

# Design of Verifiable Layered Shuffling Protocol based on Secret Sharing and Its Application Scheme

ZHANG Yanshuo[1], MAN Ziqi[1], ZHOU Xingyu[1], YANG Yatao[2], XIE Rongna[1]

[1] Department of Cryptology Science and Technology, Beijing Electronic Science & Technology Institute, Beijing 100070, China
[2] Department of Electronic and Communication Engineering, Beijing Electronic Science & Technology Institute, Beijing 100070, China

**Abstract** The existing shuffling protocols based on secret sharing have some problems, for instance: the existing shuffling protocols mainly focus on the design of theoretical framework, and lack the specific algorithm for each step of the process; Most of the solutions of the existing shuffling protocols using public key are not efficient when dealing with large data sets. The existing shuffling protocol lacks some applicability and is not very practical in some application fields. In view of these limitations, this paper designs a verifiable layered shuffling protocol based on secret sharing. At the same time, in order to combine the specific application scenario, this paper also designed a privacy protection scheme based on the shuffling protocol. In this protocol, an improved share conversion algorithm is constructed on the basis of inadvertent transfer protocol, and the original data set is shuffled without exposing the original data set. The Benes arrangement network is used to realize the shuffling layer, and then the complex shuffling task is divided into multiple sub-tasks that are easy to implement, which improves the processing efficiency of large-scale data. Furthermore, the idea of verifiability is introduced, which ensures the security of the shuffling protocol by allowing participants to confirm that the shuffling process was correctly performed. The correctness of the proposed protocol is analyzed strictly in this paper. The ideal-reality simulation paradigm was used to evaluate the security of the shuffling protocol. The time cost, security and time complexity of algorithm of the shuffling protocol are compared with other protocols. The results of the protocol show that the verifiable layered shuffling protocol based on secret sharing can meet the security standard under the malicious model. It has certain advantages in efficiency when dealing with large-scale data sets. It improves the applicability of the protocol and further promotes its application in the current environment.

**Key words** secret sharing; shuffling protocol; privacy protection; secure multiparty computing

**通讯作者:** 张艳硕, CCF 高级会员, 博士, 副教授, Email: zhang_yanshuo@163.com。
本课题得到中央高校基本科研业务费(No. 3282024003)、国家自然科学基金项目(No. 62002003)资助。
收稿日期: 2024-07-29; 修改日期: 2025-01-02; 定稿日期: 2026-01-26

## 1 引言

数据交集上的函数计算可促进参与方获得对各自有价值的信息, 但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, 隐私交集协议[1]常被采用。通过隐私交集协议, 参与方能够在不泄露各自隐私数据的前提下, 计算出基于他们数据交集的函数结果[2]。然而, 这种情况下得到的数据交集也存在着一些问题和不足。例如: 数据交集可能在参与方之间不保密; 在一些场景下, 参与方不愿交集信息在他们之间呈露, 因为这很可能会泄露一些隐私信息, 现有的隐私交集协议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]提出了一种思路: 协议可以生成一个加密的标识向量, 用以标识哪些元素属于数据集的交集。通过将加密的数据元素与相应的加密标识向量进行洗牌, 参与者就可以利用标志向量丢弃不属于交集的元素。为了提高洗牌后的隐私性, 洗牌的顺序不能被任何一个参与方知晓, 因此洗牌的结果需要使用秘密共享进行保护。基于这种思路, Chase 等[4]在 2020 年的亚密会上, 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协议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, 在参与方之间确保交集数据的隐私性来发挥其独特作用, 可有效地实现隐私保护进而更有效解决上述存在的问题。

### 1.1 相关工作

基于秘密共享的洗牌协议是这样的一种协议: 它允许参与方共同对数据洗牌, 并获得结果的秘密共享[5]。其主要步骤如下: 参与方对原数据集进行洗牌, 生成洗牌数据集; 然后参与方对洗牌数据集进行秘密共享, 从而得到自己的秘密份额。在这过程中, 秘密共享保证了参与方以外的人无法获得数据集的内容, 从而确保了协议的秘密性; 而洗牌保证了参与方只能获得打乱后的数据集, 从而确保了秘密信息的安全性。

基于秘密共享的洗牌协议近年来得到了不断的发展。Li 等人[6]于 2012 年提出了一种具有较高计算效率的洗牌协议, 并在本文中指出洗牌协议与秘密共享相结合的重要意义, 从而为后续的研究提供了思路。Zhao 等人[7]于 2016 年提出了一种基于冲突检测判断的洗牌协议, 该协议避免了第三方, 然而反复的基于冲突检测判断造成了效率的低下。Chen 等人[8]于 2020 年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, 然而具有较为复杂的过程。Attrapadung 等人[9]于 2021 年提出一种用于对数据集应用线性分组动作的两方洗牌协议。Han 等人[10]于 2022 年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 等人[11]于 2023 年基于洗牌协议提出一种采用全阈值和半诚信安全的模型。张艳硕等人[12]于 2023 年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。Shriram 等人[13] 在同一年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。Liang 等人[14]于 2023 年提出了一种基于秘密共享的循环洗牌协议, 其通过增加洗牌的次数来提高洗牌的随机性与具体效果, 尽管这可能会带来一些额外的计算开销, 但也为洗牌协议的改进提供了新的思路。满子琪等人[15]于 2024 年提出了一种基于弹性秘密共享的洗牌协议, 这种协议利用了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。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, 同时为隐私保护提供了创新的解决方案。

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。在这些应用场景中, 数据需要在一定程度上被处理, 然而数据所有者不愿意个人信息外泄; 此时, 基于秘密共享的洗牌协议便能有效保护敏感数据的隐私性, 确保数据处理过程的公正性, 并防止恶意篡改。在隐私保护方面, 基于秘密共享的洗牌协议通过数据混淆技术, 有助于维护个人隐私安全; 在商业机密保护方面, 同样的协议使得多个合作实体在不公开原始数据的情况下, 能够安全地进行数据处理。由于合作方仅需使用经过洗牌处理的数据, 而无需直接交换原始信息, 这对涉及敏感或机密信息的合作项目尤为重要。

### 1.2 贡献

针对目前研究缺乏每一流程的具体实现算法、适用性较低、难以适用于大规模数据集等问题, 提出了一种基于秘密共享的单边洗牌协议, 并在此基础上设计了一种基于秘密共享的可验证分层洗牌协议。具有分层思想处理大规模数据集、功能灵活提升协议适用性、改进的份额转换算法确保协议安全性等创新点。本文的主要贡献如下。

**安全性方面:** 引入份额转换算法, 并基于不经意传输协议对其进行改进, 确保参与者无法获知原

始数据集的内容, 有效地完成了数据的洗牌过程。引入了可验证的思想, 使得洗牌更为透明, 有效防止恶意参与者的欺骗行为。采用理想和真实范例, 于半诚实以及恶意模型下证明了协议的安全性。

**效率方面:** 提出了一种新型洗牌算法, 可广泛应用于数据集的洗牌, 并通过对比分析表现该算法在确保安全性基础上的高效性能。利用 Benes 排列网络构建分层对协议进行优化, 减少了协议运行的资源消耗, 在数据集较大时提高了协议的性能。

**实用性方面:** 给出了在隐私保护方案中的具体应用; 所提出的协议可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方法; 协议的可验证性设计提高了场景的适用性; 支持分布式的特性使得其具有较为广泛的应用性。

### 1.3 主题结构

第 1 节为引言, 第 2 节给出了协议涉及的预备知识; 第 3 节介绍了符号、系统模型与安全模型; 第 4 节对协议涉及的算法进行了介绍; 并利用其于第 5 节对协议进行了设计; 第 6 节对所设计的协议进行了分析, 指明了协议的正确性、安全性、效率; 第 7 节结合所设计的基于秘密共享的可验证分层洗牌协议构造了隐私保护的方案设计; 并于第 8 节进行了总结。

## 2 预备知识

本文提出的洗牌协议基于秘密共享机制; 利用 Benes 排列网络将复杂的洗牌操作分解为多个易于处理的洗牌操作; 应用了可验证的思想; 采用不经意传输协议构建份额转换算法, 以此完成了协议的整体实施。

本节简要回顾下洗牌协议、秘密共享、份额转换等预备知识。

### 2.1 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], 主要用于处理以下这种情况: 在样本均衡的情况下, 初始数据可能是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。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。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者无法知道其他参与者的拥有的数据信息, 也无法判断某个元素是否在其中[20]。

Chaum[21]于 1981 年首次提出了洗牌协议的概念, 旨在隐藏信息流的内容及来源。后来, Katz 等人[22]利用公钥方式设计了一种基于单同态加密的两方洗牌协议, 为洗牌协议的发展做出了贡献。Mohassel 等人[23]提出了一种适用于置换网络的基于对称密码的洗牌协议, 为不同应用场景提供了更多选择。刘涵阅等人[24]于 2021 年提出了一种基于折叠技术的洗牌协议, 并对其效率进行了较为详细的分析, 为洗牌协议的实际应用提供了更多的理论支持。Jho 等人[25]于 2023 年提出了一种具有统一属性的键控分区的洗牌协议, 为洗牌协议的进一步发展开辟了新的方向。这些研究推动了洗牌协议在信息安全领域的不断发展和完善。

洗牌协议在隐私保护和机器学习模型训练中发挥重要作用[26]。它减少了模型对数据顺序的依赖, 从而避免过拟合或引入偏差。这使得模型在训练时能够更好地泛化, 处理来自不同样本的数据, 而不会因样本顺序的影响而产生偏差。最终, 达到保护数据隐私和安全的目的, 确保任何参与者无法知晓其他参与者所持数据信息, 或判断某个元素的存在。

### 2.2 可验证洗牌

可验证洗牌的主要内容为: 参与方可以在不泄露洗牌相关信息以及不恶意更改洗牌数据集的前提下, 证明洗牌的完成。

本文参考文献[27]。该文献采用零知识证明提出了一种可验证洗牌, 其具体内容为: 首先参与方 $A$ 生成一个向量 $V=(v_1,...,v_m)$ 并利用加密生成一个向量 $E(V)=(E(v_1),...,E(v_m))$, 然后参与方 $B$ 可以对加密后的向量进行洗牌得到 $E(V')$。参与方 $B$ 可以用零知识证明告诉参与方 $A$, $E(V')$ 只是 $E(V)$ 的洗牌, 没有进行其他诸如更改数据的操作。

### 2.3 秘密共享

秘密共享协议是一种保障参与方之间安全共享秘密信息的协议[28]。其核心思想是将秘密信息分配给不同的参与者, 并要求他们合作才能恢复原始的秘密。这种特性确保了即便部分参与者的数据遭受攻击并导致信息泄露, 攻击者仍然无法获得完整的原数据集[29]。

秘密共享最早由 Shamir 在 1979 年提出[30], 基于数学原理, 确保了未获得足够信息的攻击者无法恢复出秘密信息。Shamir 后又提出了 $(t,n)$ 门限方案。当参与者所持有的坐标数量大于等于门限 $t$ 时, 利用拉格朗日差值多项式法即可求出这个秘密。张剑等人[31]于 2022 年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, 高等级的参与者的权限大于低等级的参与者。宋云等人[32]于 2022 年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。肖健等人[33]于 2023 年提出了一种基于多答案保护的弹性秘密共享方案。该方案只需要用户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。

秘密共享的实施不仅可以防止数据泄露, 还可以有效应对恶意攻击和数据盗窃行为[30]。在当今数字化时代, 秘密共享协议作为一种重要的安全机制, 被广泛应用于保护用户隐私和数据安全的各个领域。

### 2.4 份额转换

份额转换算法是一种确保参与者在不了解某个其他参与者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。

假设有两个参与者 $p_0$ 和 $p_1$, 其中 $p_0$ 执行洗牌 $S$, $p_1$ 拥有数据集 $X$, 他们想获得数据集 $X$ 的洗牌共享。抛开隐私性要求, 可以由 $p_0$ 直接对数据集 $X$ 进行洗牌, 然后将洗牌后的数据集进行秘密共享。但是这种方法使得 $p_0$ 获得了数据集 $X$, 造成了 $p_1$ 隐私性的泄露。因此, 这种方法是行不通的。

份额转换算法最早由 Chase 等[4]于 2020 年的亚密会上提出, 该方法实现了隐私保护下数据的洗牌。份额转换算法可以实现一个简易的基于秘密共享的洗牌协议, 其具体流程如下:

(1) $p_0$ 拥有洗牌 $S$ 和 $C=S(a)\oplus b$; $p_1$ 拥有数据集 $X$ 和随机掩码 $a$ 和 $b$;

(2) $p_1$ 将 $x\oplus a$ 发送给 $p_0$ 并设置自己的秘密份额为 $b$;

(3) $p_0$ 设置自己的秘密份额为 $S(x+a)\oplus C$ ,这个式子化简后为 $S(X)\oplus b$;

(4) 通过份额转换算法, 参与方 $p_0$ 和 $p_1$ 获得了各自的秘密份额, 参与方 $p_0$ 和 $p_1$ 协同即可恢复洗牌数据集 $S(X)$。

通过份额转换算法, 参与方 $\boldsymbol{p}_0$ 没有获得关于数据集以及 $\boldsymbol{X}$ 随机掩码 $\boldsymbol{a}$、$\boldsymbol{b}$ 的任何信息, 参与方 $\boldsymbol{p}_1$ 没有获得关于洗牌 $\boldsymbol{S}$ 的任何信息, 但是他们共享了一个洗牌数据集 $\boldsymbol{S}(\boldsymbol{X})$。

## 3 相关模型

根据参考文献[30], 基于秘密共享的洗牌协议的模型构建可分为两部分, 分别为系统模型和安全模型。

本节先介绍了协议的相关符号, 紧接着对系统模型进行了构建, 包括协议执行的环境、协议结构等要素的定义。接着, 本文形式化地定义了安全模型, 针对半诚实对手, 明确了安全性需求。

### 3.1 符号定义

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, 本节先介绍了协议所涉及的符号和概念定义, 如表 1 所示。

**表 1 符号定义**

**Table 1 Symbol Definition**

| 符号 | 定义 |
|---|---|
| $\{p_0,p_1\}$ | 协议的参与者 |
| $S_i(X)$ | $p_i$ 对 $X$ 的洗牌 |
| $D$ | 基于秘密共享的单边洗牌协议 |
| $DD$ | 运行两次 $D$ 的基于秘密共享的可验证分层洗牌协议 |
| $N$ | 数据集的长度 |
| $w$ | 数据集每个元素的位数 |
| $a$, $b$, $c$, $v$ | 有 $N$ 个元素的向量 |
| $v[i]$ | $v$ 的第 $i$ 个元素 |

### 3.2 系统模型

基于秘密共享的洗牌协议的系统模型为其形式化的定义。本文参考了文献[30]给出其形式化定义, 作为基于秘密共享的可验证分层洗牌协议的研究基础。

**定义 1** 在基于秘密共享的单边洗牌协议中, 一方进行洗牌 $S$, 另外一方则提供数据集 $\boldsymbol{X}$, 输出是洗牌数据集 $\boldsymbol{X}$ 的秘密份额。可表示如下:

$$F_{D[N,q]}(S,x)=(r,S(x)-r);$$

**定义 2** 基于秘密共享的可验证分层洗牌协议可由基于秘密共享的单边洗牌协议执行两次组成。

在基于秘密共享的可验证分层洗牌协议中, 洗牌由参与方依次共同执行:

$$S=S^0\cdot S^1;$$

而数据集 $x_0+x$ 也由双方共同提供, 可以考虑将洗牌或者秘密共享的数据作为输入的等效功能, 这种情况可看作单边洗牌协议执行两次的结果。

$$F_{DD[N,q]}(x_0,x_1)=(r,S(x_0+x_1)-r);$$

**定义 3** 基于秘密共享的洗牌协议的正确性。如果一个基于秘密共享的洗牌协议是正确的, 那么对于参与方 $p_0,p_1,\ldots,p_n$ 的份额 $r_0,r_1,\ldots,r_n$, 其满足如下条件:

$$r_0\oplus r_1\oplus\ldots\oplus r_n=S(x_0\oplus x_1\oplus\ldots\oplus x_n);$$

注: 为了计算得通俗易懂, 可将异或符号表示为加减符号, 不影响结果的正确性。

依据文献[30], 结合本文所提出的分层等算法, 基于秘密共享的单边洗牌协议的系统模型可定义如下:

**定义 4** 基于秘密共享的单边洗牌协议的系统模型主要包括四个部分, 分别是洗牌协议的初始化、洗牌分层、洗牌份额转换以及洗牌份额的生成, 相关功能见图 1。

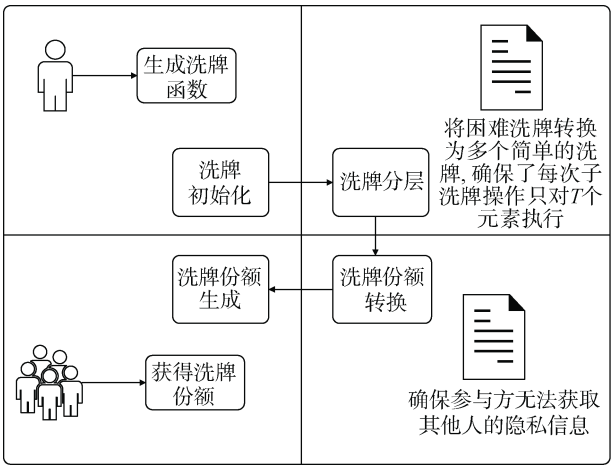


**图 1 基于秘密共享的单边洗牌协议的系统模型**

**Figure 1 System model of unilateral shuffling protocol based on secret sharing**

其具体描述可如下。

(1) 洗牌协议的初始化: 输入洗牌参数 $k$, 通过洗牌算法输出洗牌对应的函数 $S$。

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

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

(4) 洗牌份额生成: 输入洗牌函数 $S$, 输出 $p_0$ 和 $p_1$ 的份额, $p_0$ 和 $p_1$ 可联合恢复洗牌数据集。

基于秘密共享的可验证分层洗牌协议的系统模型可定义如下。

**定义 5** 基于秘密共享的可验证分层洗牌协议主要包括如下流程: 洗牌协议的初始化、两个基于秘密共享的单边洗牌协议的执行以及洗牌份额的生成。其具体描述如下。

(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函数 $S_1$ 和 $S_2$;

(2) 两个基于秘密共享的单边洗牌协议的执行: 输入一方的洗牌 $S$, 另一方的数据集 $X$, 输出 $p_0$ 和 $p_1$ 的份额。

(3) 洗牌份额的生成: 输入洗牌 $S_0$ 和 $S_1$, 输出 $p_0$ 和 $p_1$ 的份额, $p_0$ 和 $p_1$ 可联合恢复洗牌数据集。

### 3.3 安全模型

基于秘密共享的洗牌协议的安全模型为: 为保证协议安全需要满足的定义, 本节参考了文献[34]并给出了相关模型。由于协议的安全性与协议的应用环境有着很大的关系[34], 故先定义应用环境如下:

(1) 参与者没有完全安全的通信信道;

(2) 参与者只有有限的计算能力;

(3) 参与者的性质是不变的, 即参与者要么一直是恶意参与者, 要么一直是半诚实参与者。

根据文献[34], 在基于秘密共享的洗牌协议中, 参与者大多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。因此半诚实模型下协议的安全性是一个重要的考虑方向。

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 $F$。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-现实这个实验[35]。

在理想实验中, 存在一个可信第三方, 可以接收参与者的输入, 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者。理想实验可表示如下:

$\mathrm{IDEAL}^{F}_{\mathrm{sim},b}(q,x_0,x_1)$: sim 将 $x_0$, $x_1$ 发送给 $\boldsymbol{F}$, 并计算 $F(x_0,x_1)$ 得到输出 $(y_0,y_1)$, 模拟器 $\mathrm{sim}(1^q,b,x_b,y_b)$ 为参与者 $p_b$ 产生了一个模拟视图 $\mathrm{view}_b$。理想实验的输出为 $(\mathrm{view}_b,y_{1-b})$。

在现实实验中, 参与方可执行协议 $\prod$ 并与其他参与者进行交互。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者的隐私信息。现实实验可表示如下:

$\mathrm{REAL}^{\prod}_{b}(q,x_0,x_1)$: 在安全参数 $q$ 下, 协议以 $x_0$, $x_1$ 为输入, 分别独立地输出 $y_0$, $y_1$。现实实验的输出为 $(\mathrm{view}_b,y_{1-b})$。

**定义 6** 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 $x_0$ 和 $x_1$ 以及腐败方 $b\in\{0,1\}$ ( $\boldsymbol{p_b}$ 为腐败方), 在理想实验 $\mathrm{IDEAL}^{F}_{\mathrm{sim},b}(q,x_0,x_1)$ 与现实实验 $\mathrm{REAL}^{\prod}_{b}(q,x_0,x_1)$ 中的输出是不可区分的, 那么协议 $\prod$ 在半诚实安全模

型下是安全的。

根据文献, 在某些情况下, 半诚实模型协议的安全性是不够的[36]。因为虽然在半诚实模型下,协议可防止参与者好奇或者失误造成的隐私信息泄露, 但如果隐私信息的价值足够高, 参与者可能会对协议进行攻击。恶意模型下的安全性取决于如下的理想-现实实验[34]。

$\mathrm{IDEAL}^{F}_{\mathrm{sim},b}(q;\{x_0,x_1\})$: sim 将 $x_0$, $x_1$ 发送给 $\boldsymbol{F}$, 并计算 $F(x_0,x_1)$ 得到输出 $(y_0,y_1)$, 模拟器 $\mathrm{sim}(1^q,b,x_b,y_b)$ 为参与者 $p_b$ 产生了一个模拟视图 $\mathrm{view}_b$。理想实验的输出为 $(\mathrm{view}_b,y_{1-b})$。

$\mathrm{REAL}^{\prod}_{b}(q;\{x_0,x_1\})$: 在安全参数 $q$ 下, 协议以 $x_0$, $x_1$ 为输入, 分别独立的输出 $y_0$, $y_1$。现实实验的输出为 $(\mathrm{view}_b,y_{1-b})$。

**定义 7** 如果存在一个可提取现实世界中任意概率多项式时间敌手行为的模拟器 sim, 使得对于所有参与方的输入 $x_0$ 和 $x_1$ 包括腐败方 $b\in\{0,1\}$ ( $\boldsymbol{p}_b$ 为腐败方), 在理想实验 $\mathrm{IDEAL}^{F}_{\mathrm{sim},b}(q;\{x_0,x_1\})$ 与现实实验 $\mathrm{REAL}^{\prod}_{b}(q;\{x_0,x_1\})$ 中的输出是不可区分的, 那么协议 $\prod$ 在恶意模型下是安全的。

## 4 洗牌协议的相关算法设计

本部分主要对第 5 节洗牌协议的设计运用到的算法进行了构造, 主要可分为洗牌算法、份额转换算法以及分层算法。

### 4.1 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 $X$ 、洗牌参数 $k$。

(2) 将 $X$ 分成 $y+1$ 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

$$X=\{x_1^1,x_2^1,...,x_k^1,...,x_{\mathrm{pend}}^{y+1}\};$$

(3) 将 $xx_i^j$ 和 $xx_i^{j+1}$ 按 $j$ 升序连接, 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 $S$。

### 4.2 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人的隐私信息。

$\{v_i[j]\}_{i,j\in N^2}$ 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 $(c,(a,b))$。在这个算法中, 参与方 $p_0$ 和 $p_1$ 遵循如下的规则:

$p_0$ 学习所有的元素除了 $v_1[S(1)],...,v_N[S(N)]$; $p_1$ 学习整个向量 $v$, 但是其不知道洗牌 $S$。

份额转换算法的具体流程如下所示:

(1) 参与方 $\boldsymbol{p}_0$ 和 $\boldsymbol{p}_1$ 并行地执行 $N$ 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是 $v_1'$, $v_1$。

(2) 对于每个 $i\in N$, $p_0$ 设置:

$$c[i]\leftarrow\sum_{j\neq S(i)}v_i'[j]-\sum_{j\neq i}v_j'[S(i)];$$

$\boldsymbol{p}_0$ 的输出为 $c=(c[1],...,c[N])$。

(3) 对于每个 $i\in N$, $p_1$ 设置 $a_i$ 和 $b_i$ 分别为矩阵的列项和和行项和, 即

$$b_i=\sum_j v_i[j],\ a_i=\sum_j v_j[i];$$

$\boldsymbol{p}_1$ 设置自己的输出为 $(a,b)$, 这里的 $a=(a[1],...,a[N])$, $b=(b[1],...,b[N])$。因此, 由上述可知, 经过份额转换算法后, $p_0$ 输出 $c$, $p_1$ 输出 $(a,b)$。

这个算法的正确性可如下得证。

### 4.3 分层算法

分层算法将困难洗牌转换为多个简单的洗牌, 确保了每次子洗牌操作只对 $T$ 个元素执行。

4.2 节提出的份额转换算法的运行时间与 $N^2$ 呈正相关。通过分层算法, 协议使得份额转换算法的作用对象由原先大数据的洗牌转变成多个互不相交的小数据集的洗牌。在提高大规模数据的效率中具有重要的作用。

对洗牌 $S$ 执行分层算法后, 可将 $S$ 分为几个不相交的洗牌组合:

$$S=S^1\cdot S^2...S^d;$$

洗牌分层基于 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, 故每一层都是个排列。

如果输入以索引 $1,...,N$ 编号, 那么每个索引可以用二进制表示为 $\sigma_1,...,\sigma_n$, 算法的具体步骤如下所示:

(1) 令 $T=2^t, t\in N$， $N=2^n$， $d=2\left\lfloor\frac{n}{t}\right\rfloor-1$，同理 $d=2\left\lfloor\frac{\log N}{\log T}\right\rfloor-1$。$S^1$ 由前 $t$ 层组成，$S^2$ 由 $St+1,...,2t$ 组成，以此类推，除了中间的洗牌 $S^{\left\lfloor\frac{d}{2}\right\rfloor+1}$ 容纳了 $2t-1$ 层。

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

(3) 因此，对于每个子洗牌 $S^i$， $i\neq\left\lfloor\frac{d}{2}\right\rfloor+1$，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。而对于中间的洗牌 $S^{\left\lfloor\frac{d}{2}\right\rfloor+1}$ 容纳了 $2t-1$ 层，只洗牌了每一组的 $\sigma_1,...,\sigma_{n-t}x$，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。

## 5 洗牌协议的设计

5.1 节设计一种基于秘密共享的单边洗牌协议。一个完整的基于秘密共享的可验证分层洗牌协议要求所有参与者完成两轮基于秘密共享的单边洗牌协议操作。5.2 节设计出了一种基于秘密共享的可验证分层洗牌协议。下面是洗牌协议的相关设计。

### 5.1 基于秘密共享的单边洗牌协议的设计

基于秘密共享的单边洗牌协议本质上构成了基于秘密共享的可验证分层洗牌协议流程的一半。在该协议框架中，参与者之一承担洗牌操作 $S$，而另一参与者负责分发数据份额 $X$。基于秘密共享的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 4.1 节提出的洗牌算法输出洗牌对应的函数 $S$。

**(2) 洗牌分层。**

输入洗牌函数 $S$，数据集长度 $N$。参与方 $p_0$ 通过 4.2 节提出的分层算法输出洗牌的分层 $S_0=S_0^1\circ...\circ S_0^d$;

**(3) 洗牌份额转换。**

对于每个 $S_i$，参与双方依据 4.2 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 4.2 节提出的份额转换算法得到了 $a^{(i,1)},...,a^{\left(i,\frac{N}{T}\right)}$ 和 $b^{(i,1)},...,b^{\left(i,\frac{N}{T}\right)}$，分别简记为向量 $a^{(i)}$ 和向量 $b^{(i)}$。

利用向量 $\boldsymbol{a}^{(i)}$ 和向量 $\boldsymbol{b}^{(i)}$， 参与者 $p_0$ 可通过下述式子得到 $c^{(i,1)},...,c^{\left(i,\frac{N}{T}\right)}$，简记为向量 $c^{(i)}$:

$c^{(i)}=b^{(i)}-S_0^i(a^{(i)})$;

参与者 $p_0$ 可利用 2.2 节的可验证洗牌来证明自己洗牌操作的可验证性。

**(4) 洗牌份额生成。**

利用上面得到的向量 $\boldsymbol{a}^{(i)}$、$\boldsymbol{b}^{(i)}$、$c^{(i)}$，对于每个 $i\in1,...,d-1$， $p_1$ 计算 $\delta^{(i)}$ 并将其发送给 $p_0$:

$\delta^{(i)}=a^{(i+1)}-b^{(i)}$;

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

$p_0$ 计算如下式:

$c=c^{(d)}+S_0^d(\delta^{(d-1)}+c^{(d-1)}+S_0^{d-1}$
$(\delta^{(d-2)}+c^{(d-2)}+...+S_0^2(\delta^{(1)}+c^{(1)})))$

并输出 $S(m)+c-w$，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 $S(m)+c-w$， $p_1$ 得到了洗牌份额 $b=w-b^{(d)}$。实际上，此时 $p_0$ 和 $p_1$ 可协同恢复洗牌数据集 $S(X)$。在后续 6.1 节，将结合 3.2 节的正确性定义，对该协议的正确性进行分析。

### 5.2 基于秘密共享的可验证分层洗牌协议的设计

在基于秘密共享的可验证分层洗牌协议中，参与方 $p_0$ 和 $p_1$ 各拥有一个数据集 $x_0$ 和 $x_1$，且分别拥有洗牌 $S_0$ 和 $S_1$。 $\boldsymbol{p}_0$ 和 $\boldsymbol{p}_1$ 想在双方数据集的交集上计算一些函数，且不想给彼此暴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 $x_0$ 和 $x_1$ 需要保密，数据集的交集也需要保密，那么参与方需要得到的是洗牌数据集的秘密份额。根据 3.2 节系统模型的定义，基于秘密共享的可验证分层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 4.1 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 5.1 节提出的基于秘密共享的单边

洗牌协议对 $x_1$ 应用洗牌 $S_0$，使得 $p_0$ 获得 $x_0^{(1)}$，$p_1$ 获得 $x_1^{(1)}$；然后 $p_0$ 计算：$x_0^{(2)} = S_0(x_0) + x_0^{(1)}$；

**(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 5.1 节提出的基于秘密共享的单边洗牌协议对 $x_0^{(2)}$ 应用洗牌 $S_1$，使得 $p_0$ 获得 $x_0^{(3)}$，$p_1$ 获得 $x_1^{(3)}$；然后 $p_1$ 计算：$x_1^{(4)} = S_1(x_1^{(1)}) + x_1^{(3)}$；

**(4) 洗牌份额的生成。**

$p_0$ 输出 $x_0^{(3)}$，$p_1$ 输出 $x_1^{(4)}$。

通过上述提出的基于秘密共享的可验证分层洗牌协议，$p_0$ 得到了洗牌份额 $x_0^{(3)}$，$p_1$ 获得洗牌份额 $x_1^{(4)}$。实际上，此时 $p_0$ 和 $p_1$ 可协同恢复洗牌数据集 $S_1(S_0(x_0 + x_1))$。在后续 6.1 节，将结合 3.2 节中的正确性定义，对该协议的正确性进行分析。

## 6 洗牌协议的分析

在第 5 节中，本文提出了一种基于秘密共享的单边洗牌协议，在此协议的基础上，构造出了一种基于秘密共享的可验证分层洗牌协议。本节将对这两种协议从正确性、安全性、效率、性能四个方面进行分析。

### 6.1 正确性分析

根据文献[37]，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的前提下，联合恢复出洗牌数据集。

**定理 1.** 基于秘密共享的单边洗牌协议使得参与方能够协同恢复出洗牌数据集，满足正确性分析。

证明. 因为对于任意 $i$，都有

$$c^{(i)} = b^{(i)} - S_0^i(a^{(i)})$$

这意味着对于任意的 $i$，都有

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

因此，最终参与方 $p_0$ 产生的 $c$ 为

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

而参与方 $p_0$ 与 $p_1$ 的输出(秘密份额)分别为

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

可见参与方 $p_0$ 与 $p_1$ 的输出(秘密份额)组合后，可恢复出洗牌数据集 $S_0(x)$。

成立。证毕。

**定理 2**. 本文设计的基于秘密共享的可验证分层洗牌协议使得参与方联合能够恢复出洗牌数据集 $S_1(S_0(x_0 + x_1))$，满足正确性分析。

证明. $p_0$ 输出 $x_0^{(3)}$，也是它的秘密份额，这可进一步化简为

$$\begin{aligned}&x_0^{(3)}\\ &= S_1(x_0^{(2)}) - r^{(3)}\\ &= S_1(S_0(x_0) + x_0^{(1)}) - r^{(3)}\\ &= S_1(S_0(x_0) + r^{(1)}) - r^{(3)}\\ &= S_1(S_0(x_0)) + S_1(r^{(1)}) - r^{(3)}\end{aligned}$$

$p_1$ 输出 $x_1^{(4)}$，也是它的秘密份额，这可进一步化简为

$$\begin{aligned}&x_1^{(4)}\\ &= S_1(x_1^{(1)}) + x_1^{(3)}\\ &= S_1(x_1^{(1)}) + r^{(3)}\\ &= S_1(S_0(x_1) - r^{(1)}) + r^{(3)}\\ &= S_1(S_0(x_1)) - S_1(r^{(1)}) + r^{(3)}\end{aligned}$$

将上述 $\boldsymbol{p}_0$ 的输出和 $\boldsymbol{p}_1$ 的输出联合后，可得到洗牌数据集 $S_1(S_0(x_0 + x_1))$。

成立。证毕。

**定理 3.** 本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。在进行洗牌时，引入洗牌参数 $k$，其确保了洗牌后的结果具有足够的随机性。

接下来，考虑洗牌的后半部分。由于后半部分的元素个数可能不足以构成一个长度为 $k$ 的数组，为了保证结果的随机性，我们采用哈希函数生成哈希值来表示这些元素的位置。假设哈希函数为 $H(x)$，其中 $x$ 表示元素的值。对于不同的元素，哈希函数将返回不同的哈希值，从而保证了在洗牌后每个元

素都具有一个唯一的位置。

因此, 通过前半部分的交替排列和后半部分的哈希值生成, 我们可以得到一个满足正确性需求的洗牌结果。

成立。证毕。

## 6.2 安全性分析

根据 3.3 节安全模型所提出, 如果一个基于秘密共享的洗牌协议在 3.3 节所提出的应用环境中是安全的, 其需要满足定义 6 的要求, 即模拟器能产生与实际实验无异的理想实验结果。下面, 本节将从 $p_0$ 为腐败方以及 $p_1$ 为腐败方两个方面, 验证实际实验和理想实验的输出是否一致。

当 $b$=0 时, 即 $\boldsymbol{p}_0$ 为腐败方, 模拟器 $\mathrm{sim}(1^q,0,x_0,y_0)$ 将选择一个 $S_0$ 和 $x_0^{(1)}$, 设置 $x_0^{(2)}=S_0(x_0)+x_0^{(1)}$, 模拟第一个基于秘密共享的单边洗牌协议为 $\mathrm{sim}^D(1^q,0,S_0,x_0^{(1)})$, 模拟第二个基于秘密共享的单边洗牌协议为 $\mathrm{sim}^D(1^q,0,x_0^{(2)},y_0)$。

当 $b$=1 时, 即 $\boldsymbol{p}_1$ 为腐败方, 模拟器 $\mathrm{sim}(1^q,1,x_1,y_1)$ 将选择一个 $S_1$ 和 $x_1^{(1)}$, 设置 $x_1^{(3)}=y_1-S_1(x_1^{(1)})$, 模拟第一个基于秘密共享的单边洗牌协议为 $\mathrm{sim}^D(1^q,1,x_1,x_1^{(1)})$, 模拟第二个基于秘密共享的单边洗牌协议为 $\mathrm{sim}^D(1^q,1,s_1,x_1^{(3)})$。

**定理 4.** 模拟器能产生与实际实验无异的理想实验结果。

(1) 当 $b=0$ 时, 运行现实这个实验, 输出的是 $p_0$ 的视图(它的输入是 $x_0$, 来自两个单边洗牌协议的 $\mathrm{view}_0^{(1)}$ 和 $\mathrm{view}_0^{(2)}$ 包含了输出 $x_0^{(1)}$ 和 $x_0^{(3)}$, 诚实方 $p_1$ 的输入是 $x_1$, 输出是 $x_1^{(4)}=S_1(x_1^{(1)})+x_1^{(3)}$。

游戏 1. 在步骤 1 中, 首先计算 $F_D(S_0,x_1)$, 例如, 随机地选择一个 $r^{(1)}$, 然后令 $x_0^{(1)}=r^{(1)}$, $x_1^{(1)}=S_0(x_1)-r^{(1)}$, 然后运行基于秘密共享的单边洗牌模拟器产生第一个视图 $\mathrm{view}_0^{(1)\prime}$。输出是 $p_0$ 的视图(它的输入是 $x_0$, 来自两个基于秘密共享的单边洗牌协议的 $\mathrm{view}_0^{(1)\prime}$ 和 $\mathrm{view}_0^{(2)\prime}$ 包含了输出 $x_0^{(1)}=r^{(1)}$ 和 $x_0^{(3)}$, 诚实方 $p_1$ 的输入是 $x_1$, 输出:

$$x_1^{(4)}=S_1(x_1^{(1)})+x_1^{(3)}=S_1(S_0(x_1)-r^{(1)})+r^{(3)}$$

与现实实验不可区分。

游戏 2. 在步骤 3 中, 首先计算 $F_D(S_1,x_0^{(2)})$, 例如, 随机选择一个 $r^{(3)}$, 然后令 $x_1^{(3)}=r^{(3)}$, $x_0^{(3)}=S_1(x_0^2)-r^{(3)}$, 然后运行基于秘密共享的单边洗牌模拟器产生第一个视图 $\mathrm{view}_0^{(2)\prime}$。输出是 $p_0$ 的视图(它的输入是 $x_0$, 来自两个基于秘密共享的单边洗牌协议的 $\mathrm{view}_0^{(1)\prime}$ 和 $\mathrm{view}_0^{(2)\prime}$ 包含了输出 $x_0^{(1)}=r^{(1)}$ 和 $x_0^{(3)}=S_1(x_0^2)-r^{(3)}$, 诚实方 $p_1$ 的输入是 $x_1$, 输出:

$$\begin{aligned}&x_1^{(4)}\\&=S_1(S_0(x_1)-r^{(1)})+x_1^{(3)}\\&=S_1(S_0(x_1)-r^{(1)})+r^{(3)}\end{aligned}$$

与现实实验不可区分。

游戏 3. 选择 $S$, $r$, $x_0^{(1)}$, 令 $S_1=S\circ S_0^{-1}$, $x_0^{(1)}=r^{(1)}$, $r^{(3)}=S_1(S_0(x_0))+S_1(r^{(1)})-r$, 除此之外, 继续进行游戏 2。输出是 $p_0$ 的视图(它的输入是 $x_0$, 来自两个基于秘密共享的单边洗牌协议的 $\mathrm{view}_0^{(1)\prime}$ 和 $\mathrm{view}_0^{(2)\prime}$ 包含了输出 $x_0^{(1)}=r^{(1)}$ 和 $x_0^{(3)}$), 诚实方 $p_1$ 的输入是 $x_1$, 输出是 $x_1^{(4)}$。由上述正确性定理可知, $x_1^{(4)}=S_1(S_0(x_1)-r^{(1)})+r^{(3)}$, 可化为

$$\begin{aligned}&x_1^{(4)}\\&=S_1(S_0(x_1)-x_0^{(1)})+S_1(S_0(x_1))+S_1(r^{(1)})-x_0^{(3)}\\&=S(x_1+x_0)-x_0^{(3)}\end{aligned}$$

因此, $b=0$ 时, 现实实验与理想实验是不可区分的。

(2)当 $b=1$ 时, 运行现实这个实验,

游戏 1: 在步骤 1 中, 首先计算 $F_D(S_0,x_1)$, 例如, 随机选择一个 $x_0^{(1)}$, 然后计算 $x_1^{(1)}=S_0(x_1)-x_0^{(1)}$, 然后运行基于秘密共享的单边洗牌模拟器产生第一个视图。与现实实验不可区分。

游戏 2: 在步骤 3 中, 首先计算 $F_D(S_1,x_0^{(2)})$, 例如, 随机选择一个 $x_1^{(3)}$, 然后计算 $x_0^{(3)}=S_1(x_0^2)-x_1^{(3)}$, 然后运行基于秘密共享的单边洗牌模拟器产生视图 $\mathrm{view}_0^{(2)\prime}$。这与现实实验不可区分。

游戏 3: 选择一个随机的 $S$, 设 $S_0=S\circ S_1^{-1}$, $x_1^{(3)}=S(x_0+x_1)-S_1(x_1^{(1)})-x_0^{(3)}$, 这意味 $x_1^{(4)}=S(x_0+x_1)-x_0^{(3)}$。由于游戏 2 中的 $x_0^{(3)}=S_1(x_0^2)-x_1^{(3)}$ 可以化简为

$$S_1(S_0(x_0)+x_0^{(1)})-x_0^{(3)}$$

也就等于:

$$\begin{aligned}&S_1(S_0(x_0+x_1))-S_1(x_1^{(1)})-x_0^{(3)}\\&=S(x_0+x_1)-S_1(x_1^{(1)})-x_0^{(3)}\end{aligned}$$

这也与现实实验不可区分。

结果表明, 该仿真器能产生与实际实验无异的理想实验结果。证毕。

### 6.3 效率分析

本节主要对所提出的协议在通信开销、通信复杂度方面的效率进行分析。

本文提出的两种基于秘密共享的洗牌协议采用了洗牌分层算法, 旨在确保每次子洗牌操作仅对 $T$ 个元素进行处理, 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, 本协议的 $T$ 的最佳的参数取值范围在 16~256, 这为实际应用提供了重要的参考依据。

基于秘密共享的单边洗牌协议一共运行 $dN/T$ 次份额转换算法, 其中 $d=2\left\lceil\frac{\log N}{\log T}\right\rceil-1$。

在这些运算中, 通信开销是一个关键指标, 它反映了在协议执行过程中信息传输所需的资源消耗。基于秘密共享的单边洗牌协议的通信开销为 $(d+1)Nw$;

基于秘密共享的单边洗牌协议的计算开销等同于并行运行 $dN/T$ 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。基于秘密共享的可验证分层洗牌协议执行三轮, 通信复杂度与 $qN\log N+\frac{Nw\log N}{\log T}$ 成比例。

通过对通信和计算开销的合理管理与优化, 基于秘密共享的洗牌协议能够在保障安全性的同时, 实现高效的数据洗牌操作, 为隐私保护和数据安全提供了有效的解决方案。

### 6.4 对比分析

本节将本文协议与文献[8, 25]提出的协议进行了比较。

文献[25]采用了一种基于公钥的解决办法, 然而并未对洗牌过程进行分层处理。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。特别是在洗牌操作方面, 未采用分层的洗牌会导致通信复杂度较高。为 $(N\log N-\frac{N}{2})(q+4w)+2Nw$。

文献[8]的通信复杂度与文献[25]的通信复杂度处于同一个数量级。

相比之下, 本文协议采用了洗牌分层算法, 以确保每次子洗牌操作只对 $\boldsymbol{T}$ 个元素执行。这种分层策略在处理大规模数据集时具有显著的优势, 可以有效降低通信开销, 并提高整体效率。通过在洗牌过程中对数据进行分层处理, 本协议能够更加灵活地适应不同规模的数据集, 从而提升了协议的性能和实用性。

本协议通信复杂度与 $qN\log N+\frac{Nw\log N}{\log T}$ 成比例。因此当数据集元素较小时, 本协议在效率方面并无明显优势, 甚至时间开销因为洗牌分层算法的存在, 要高于上述文献。但是当数据集元素较大时, 本文在时间开销上具有一定的优势。

文献[8]集中于洗牌算法的设计, 单次执行洗牌算法的时间复杂度为 $O(N)$, 与本文相当。与本文相比, 该文献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。通过多次循环的洗牌过程, 该文献能够更加均匀地打乱数据的顺序, 从而增加了数据的难以预测性, 提高了安全性水平。

然而, 这种多次循环的洗牌方法也带来了整体开销的增加。由于需要对数据进行多次洗牌, 因此整体的计算和通信开销也相应增加。尽管这种洗牌方法在提高安全性方面具有优势, 但其在性能方面可能会受到一定的影响。特别是在处理大规模数据集时, 多次循环的洗牌过程可能会导致较高的计算和通信开销, 从而影响了整体效率。

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, 实现较高效率的数据处理, 为安全计算提供了一种可行的解决方案。

协议的对比分析如表 2 所示。

经过表 2 的对比分析, 可以看出: 在安全性方面, 本协议能够满足安全多方计算环境下的基本要求。在效率方面, 在数据集较小的情况下, 本协议与其他相比并没有显著的优势; 然而, 当处理的数据集较大时, 本协议的优点就凸显出来了, 它具有更低的时间消耗。

**表 2 协议的对比分析**

**Table 2 Comparative analysis of protocols**

| 协议 | 时间开销 | 安全性 | 洗牌算法 |
|---|---|---|---|
| 文献[25] | 元素较多时高; 元素较少时低 | 半诚实下的安全性 | 未给出具体的洗牌算法 |
| 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 |
| 本协议 | 元素较少时高; 元素较多时低 | 半诚实下的安全性 | 时间复杂度为 $O(N)$ |

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。本协议在这方面表现出色, 因为它具有以下几个优点:

首先, 本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, 同时也能够灵活适用于各种不同的具体场景和应用需求。举例来说, 当应用于协同过滤等场景时, 本协议可以有效地保护用户的个人偏好和隐私信息, 从而实现用户数据的安全共享和合理利用。

其次, 本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、易于理解, 使得协议的实施更加高效。

表 3 展示了在实用性方面, 本协议与其他文献的对比情况。

**表 3 实用性的对比分析**

**Table 3 Comparative analysis of practicality**

| 协议 | 实用性 |
|---|---|
| 文献[25] | 采用公钥解决方案, 利用数学方法, 较复杂 |
| 文献[21] | 公钥解决方案, 需要多次循环 |
| 本协议 | 可与隐私交集计算紧密结合; 多采用异或运算, 运算结构较简单 |

## 7 基于洗牌协议的隐私保护方案的设计

本文提出了一种基于秘密共享的单边洗牌协议, 并在此基础上构造了一种基于秘密共享的可验证分层洗牌协议, 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、大数据抽样、随机存取存储器程序。下面将以商家——广告商的模型为例, 基于洗牌协议设计一种隐私保护方案。

### 7.1 模型构建

在商家——广告商的模型中, 广告供应商拥有一个数据集 $x_0$, 其中包含了看到广告的用户信息; 商家拥有另一个数据集 $x_1$, 其中包含了购买的用户信息。商人企图通过广告商提供的广告给自己带来额外的收入, 因此商人想知道所购买的广告有没有具体的效果。

然而用户点击广告和购买行为都是敏感信息。泄露这些信息可能导致用户隐私被侵犯。且根据相关数据保护法律, 个人数据的处理需要保护用户的隐私, 这包括防止未经授权的数据共享和暴露。因此需要确保数据交集在双方之间保密的情况下, 得到对双方各自有益的信息。

### 7.2 方案设计

本节利用基于秘密共享的可验证分层洗牌协议, 对商人——广告商的模型下的隐私保护方案进行了设计, 主要步骤如下所示。

**(1) 初始化阶段。**

广告供应商和商家输入洗牌参数 $k_1$ 和 $k_2$, 通过 4.1 节提出的洗牌算法确定洗牌函数 $S_1$ 和 $S_2$。

**(2) 洗牌协议的执行。**

首先广告供应商和商家运行 5.1 节提出的基于秘密共享的单边洗牌协议对 $x_1$ 应用洗牌 $S_0$, 使得广告供应商获得 $x_0^{(1)}$, 商家获得 $x_1^{(1)}$;

接着广告供应商计算: $x_0^{(2)} = S_0(x_0) + x_0^{(1)}$;

然后广告供应商和商家运行 5.1 节提出的基于秘密共享的单边洗牌协议对 $x_0^{(2)}$ 应用洗牌 $S_1$, 使得广告供应商获得 $x_0^{(3)}$, 商家获得 $x_1^{(3)}$;

最后商家计算: $x_1^{(4)} = S_1(x_1^{(1)}) + x_1^{(3)}$;

**(3) 洗牌份额的生成。**

广告供应商获得 $x_0^{(3)}$, 商家获得 $x_1^{(4)}$。

### 7.3 方案简评

通过上述方案, 商家和广告供应商秘密共享了一个洗牌数据集 $S_1(S_0(x_0 + x_1))$, 他们可协同在数据交集上进行相关运算。但是无论单个商家还是单个广告供应商, 都不能够知晓数据交集上的相关信息, 进而保证了方案的隐私性。

## 8 结论

本文提出了一种基于秘密共享的单边洗牌协议, 并在此基础上构造了一种基于秘密共享的可验证分层洗牌协议, 并将其应用于具体的隐私保护方案之中。本文提出的两个协议充分利用了密码学理论中的秘密共享、洗牌算法以及份额转换等技术手段, 以确保数据洗牌过程的安全性和可验证性。

在具体内容方面, 本文采用了分层思想对协议进行优化, 通过将洗牌过程分解成多个层次, 提高了协议的执行效率和数据处理速度。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, 保证了协议的安全性。另外, 本文还引入了可验证的思想, 通过加入验证机制, 提高了协议的安全性, 并证明了协议在恶意模型下的安全性。

与现有的学术成果相比, 本文所提出的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。本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, 可适用于大规模数据处理的需求。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]. *23rd Annual Symposium on Foundations of Computer Science* (*sfcs 1982*), 1982: 160-164.

[2] 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.
(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.)

[3] Ciampi M, Orlandi C. Combining Private Set-Intersection with Secure Two-Party Computation[M]. Security and Cryptography for Networks. Cham Springer International Publishing2018: 464-482.

[4] Chase M, Ghosh E, Poburinnaya O. Secret-Shared Shuffle[C]. *Advances in Cryptology – ASIACRYPT 2020*, 2020: 342-372.

[5] Zhang Y Y, Li S Y, Shi Y X, et al. Secure Multi-Party Θ-Join Algorithm Toward Data Federation[J]. *Journal of Software*, 2023, 34(3): 1109-1125.
(张媛媛, 李书缘, 史烨轩, 等. 面向数据联邦的安全多方 θ -连接算法[J]. *软件学报*, 2023, 34(3): 1109-1125.)

[6] Li S S, Zhang C, Lin D D. Secure Multiparty Computation with Lazy Sharing[C]. *The 2024 on ACM SIGSAC Conference on Computer and Communications Security*, 2024: 795-809.

[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]. *The 2021 ACM SIGSAC Conference on Computer and Communications Security*, 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]. *2022 IEEE 38th International Conference on Data Engineering*, 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] 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.
(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.)

[13] 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.

[14] Liang J T, Sagan B E, Zhuang Y. Cyclic Shuffle-Compatibility *via* Cyclic Shuffle Algebras[J]. *Annals of Combinatorics*, 2024, 28(2): 615-654.

[15] 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.
(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.)

[16] Singh H, Sinha A. A Blockchain Framework for E-Voting[J]. *Multimedia Tools and Applications*, 2024, 83(20): 58875-58889.

[17] 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.
(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.)

[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] 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.
(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.)

[20] 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.

[21] Chaum D. Untraceable Electronic Mail, Return Addresses and Digital Pseudonyms[M]. Secure Electronic Voting. Boston, MA: Springer US, 2003: 211-219.

[22] Katz J, Malka L. Constant-round Private Function Evaluation with Linear Complexity[M]. Advances in Cryptology – ASIACRYPT 2011. Berlin, Heidelberg Springer2011: 556-571.

[23] Mohassel P, Sadeghian S. How to Hide Circuits in MPC an Efficient Framework for Private Function Evaluation[M]. Advances in Cryptology – EUROCRYPT 2013. Berlin, Heidelberg Springer2013: 557-574.

[24] 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.
(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.)

[25] Jho N S, Lee J. Partition and Mix: Generalizing the Swap-or-Not Shuffle[J]. *Designs, Codes and Cryptography*, 2023, 91(6): 2237-2254.

[26] Wu Z H, Wei Q, Ren K L, et al. Dynamic Defense for DDoS Attack Using Open Flow-Based Switch Shuffling Approach[J]. *Journal of Electronics & Information Technology*, 2017, 39(2): 397-404.
(武泽慧, 魏强, 任开磊, 等. 基于 OpenFlow 交换机洗牌的 DDoS 攻击动态防御方法[J]. *电子与信息学报*, 2017, 39(2): 397-404.)

[27] Bayer S, Groth J. Efficient Zero-Knowledge Argument for Correctness of a Shuffle[M]. Advances in Cryptology – EUROCRYPT 2012. Berlin, Heidelberg Springer2012: 263-280.

[28] Song X L, Li C. Dynamic Quantum Secret Sharing Scheme Based on Nonlocal Orthogonal Product States[J]. *Journal of Electronics & Information Technology*, 2024, 46(3): 1109-1118.
(宋秀丽, 李闯. 基于非局域性正交乘积态的动态量子秘密共享方案[J]. *电子与信息学报*, 2024, 46(3): 1109-1118.)

[29] 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.

[30] Shamir A. How to Share a Secret[J]. *Communications of the ACM*, 1979, 22(11): 612-613.

[31] 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.
(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.)

[32] 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.
(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.)

[33] 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.
(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.)

[34] Zhang E, Qin L Y, Yang R L, et al. Multi-Party Threshold Private Set Intersection Protocol Based on Robust Secret Sharing[J]. *Journal of Software*, 2023, 34(11): 5424-5441.
(张恩, 秦磊勇, 杨刃林, 等. 基于弹性秘密共享的多方门限隐私集合交集协议[J]. *软件学报*, 2023, 34(11): 5424-5441.)

[35] Canetti R. Universally Composable Security: A New Paradigm for Cryptographic Protocols[C]. *Proceedings 42nd IEEE Symposium on Foundations of Computer Science*, 2001: 136-145.

[36] Li C, Wang J, Liu J Q. Blockchain-Based Lightweight Anonymous Review System[J]. *Journal of Cyber Security*, 2022, 7(5): 91-107.
(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.)

[37] Jack P K Ma, Sherman S M. Chow. Secure-Computation-Friendly Private Set Intersection from Oblivious Compact Graph Evaluation[D]. AsiaCCS, Nagasaki, Japan, 2022: 1086-1097.

**张艳硕** 于 2009 年在中国科学院数学与系统研究院获得博士学位。现任北京电子科技学院副教授。研究领域为密码理论及其应用。研究兴趣包括: 区块链、安全协议等。Email: zhang_yanshuo@163.com

**满子琪** 于 2022 年在南京信息工程大学信息安全专业获得学士学位。现在北京电子科技学院网络空间安全专业攻读硕士学位。研究领域为密码协议及其应用。研究兴趣包括: 秘密共享、安全协议等。Email: 2673028450@qq.com

**周幸妤** 于 2023 年在东北大学秦皇岛分校获得学士学位。现在北京电子科技学院网络空间安全专业攻读硕士学位。研究领域为密码协议及其应用。研究兴趣包括: 安全协议等。Email: 2673028450@qq.com

**杨亚涛** 于 2009 年在北京邮电大学获得博士学位。现任北京电子科技学院教授。研究领域为密码学与通信安全等。研究兴趣包括: 同态加密、密码协议与算法等。Email: yy2008@163.com

**谢绒娜** 于 2020 年在西安电子科技大学获得博士学位。现任北京电子科技学院教授。研究领域为密码理论与协议。研究兴趣包括: 数据安全与隐私保护、安全体系结构与系统安全等。Email: 486503266@qq.com