高效的概率门限隐私集合交集协议

张 恩[1),2)] 刘登辉[1)] 杜瑞颖[3)]

[1)](河南师范大学计算机与信息工程学院 河南 新乡 453007)

[2)](河南省教育人工智能与个性化学习重点实验室 河南 新乡 453007)

[3)](武汉大学国家网络安全学院 武汉 430072)

摘 要 概率门限隐私集合交集(Probabilistic Threshold Private Set Intersection,PTPSI)是门限隐私集合交集的一种概率变体,当交集数量处于给定区间内时,会以一定概率计算交集,交集数量越多,计算交集的概率越大。相比确定型门限隐私集合交集协议,PTPSI在拼车、联邦学习等场景中展现了更高的效率。然而,现有针对半诚实敌手的PTPSI协议在门限测试阶段依赖昂贵的通用电路计算机制,其计算开销与参与方数量呈指数关系,不能有效扩展至多方场景。针对此问题,首先在半诚实模型下基于双中心零共享技术(Bicentric Zero-Sharing,BZS)设计一种高效的PTPSI协议。在5方场景下,每个参与方集合大小为$n=2^{20}$,门限值设为$0.5n$,现有协议的运行时间为45.40秒,改进后的协议总运行时间为9.62秒,通信量为187.39 MB,速度提升4.72倍。当参与方数量从5方扩展到32方时,协议总运行时间为9.92秒。为进一步抵抗合谋攻击,提出第二个隐私增强的PTPSI协议,使用不经意伪随机函数来限制聚合参与方的恶意查询,同样场景下,该协议时间成本为30.25秒。两个协议都能抵抗特定$N-1$个参与方的合谋攻击,且随着参与方数量增多,与现有协议相比,优势更加明显。

关键词 概率门限隐私集合交集;门限隐私集合交集;双中心零共享技术;不经意键值存储;不经意伪随机函数

中图法分类号 TP309 **DOI号** 10.11897/SP.J.1016.2026.00309

Efficient Probabilistic Threshold Private Set Intersection Protocol

ZHANG En[1),2)] LIU Deng-Hui[1)] DU Rui-Ying[3)]

[1)](College of Computer and Information Engineering, Henan Normal University, Xinxiang, Henan 453007)

[2)](Key Laboratory of Artificial Intelligence and Personalized Learning in Education of Henan Province, Xinxiang, Henan 453007)

[3)](School of Cyber Science and Engineering, Wuhan University, Wuhan 430072)

Abstract Probabilistic threshold private set intersection (PTPSI) is a probabilistic variant of threshold private set intersection (TPSI), where the intersection is computed with a certain probability when the intersection size falls within a given range. The larger the intersection size, the higher the probability of computing the intersection. The protocol is divided into two phases. The first phase is the threshold testing phase: this phase is designed such that the probability of passing it increases with the size of the intersection among all parties. The protocol then proceeds to the intersection calculation phase, where all parties will execute a standard private set intersection (PSI) protocol to obtain the final result. Compared to deterministic TPSI protocols, PTPSI demonstrates higher efficiency in scenarios such as ride-sharing and federated learning. However, existing PTPSI protocols against semi-honest adversaries rely on expensive generic circuit-based computation mechanisms during the threshold testing phase, with computational costs exponentially related to the number of participants, thus failing to scale efficiently to multi-

收稿日期:2025-02-24;在线发布日期:2025-07-07。本课题得到国家自然科学基金(62372157)资助。**张 恩**(通信作者),博士,教授,硕士生导师,中国计算机学会(CCF)高级会员,主要研究领域为密码协议设计、安全多方计算和区块链。E-mail:zhangenzdrj@163.com。**刘登辉**,硕士研究生,主要研究领域为密码协议设计和安全多方计算。**杜瑞颖**(通信作者),博士,教授,博士生导师,主要研究领域为网络安全和隐私保护。E-mail:duraying@126.com。

party scenarios. To address this issue, this paper first designs an efficient PTPSI protocol based on bicentric zero-sharing (BZS) in the semi-honest model. The main idea is to utilize BZS techniques to reduce the number of participants in the threshold testing phase from multiple parties to two central parties. These two central parties perform the threshold test, and if the test is passed, they proceed to compute the intersection. This approach reduces the communication and computational overhead among participants. In a five-party setting, with each participant's set size being $n=2^{20}$ and the threshold value set to $0.5n$, the runtime of the existing protocol is 45.40 seconds, while the improved protocol has a total runtime of 9.62 seconds with a communication volume of 187.39 MB, achieving a 4.72× speedup. When the number of parties increased from 5 to 32, the total runtime of the protocol was 9.92 seconds. Experimental results demonstrate that the protocol exhibits high scalability. To further resist collusion attacks from multiple parties, a second privacy-enhanced PTPSI protocol is proposed in the semi-honest model. This protocol utilizes an oblivious pseudorandom function to effectively restrict malicious queries from aggregated participants. The core idea is that before the threshold testing phase, each party must act as an OPRF receiver and commit to its own input set. When parties collude, they cannot obtain any information beyond the intersection because they lack the OPRF keys of the honest parties. In the same scenario, assume there are five participants, each with participant's set size being $n=2^{20}$ and the threshold value set to $0.5n$, the time cost of the protocol is 30.25 seconds. Through extensive experimental simulations, this paper reproduces the results of existing protocols under the same conditions and compares them in terms of both computational overhead and communication overhead. The two protocols proposed in this paper can resist collusion attacks from a specific number of $N-1$ participants, and their advantages over existing protocols become more pronounced as the number of participants increases. Finally, algorithmic analysis and experimental results demonstrate these protocols outperform existing ones in terms of performance.

Keywords probabilistic threshold private set intersection; threshold private set intersection; bicentric zero-sharing; oblivious key-value stores; oblivious pseudorandom function

1 引 言

隐私集合交集(Private Set Intersection,PSI)[1]允许多个参与方,在保护其各自集合隐私的条件下计算这些集合的交集,并且不泄露除交集以外的任何信息。近年来,该技术发展逐渐成熟,按照参与方数量分类,可以分为两方PSI[2-7]与多方PSI[8-13]。PSI在许多应用中具有重要价值,比如僵尸网络检测[14],计算广告转化率[15]和接触者追踪[16]等。然而,标准的PSI无法适用于以下情形:在拼车服务中,多个用户通常只有在大多数轨迹相交时才有拼车的需求[17];在数据挖掘和机器学习中,当数据在各个参与方之间垂直分布时,只有在公共数据集比较大时,各方才会有合作的动力[18]。

针对上述场景,2004年Freedman等人[1]提出并设计了一种门限隐私集合交集(Threshold Private Set Intersection,TPSI)协议,只有当参与方之间的交集数量达到预设门限t时,才能进行PSI协议。随后对于这一主题的研究逐渐深入[19-25]。尽管理论上TPSI是可行的,但如何构建一种高效且可扩展的多方TPSI协议,同时确保交集基数不被泄露,目前仍然是个困难问题。

针对该问题,Liu等人[26]于2023年提出概率门限隐私集合交集(Probabilistic Threshold Private Set Intersection,PTPSI)计算协议的定义,它是标准TPSI的一个概率变体,如图1所示,PTPSI允许参与方在交集数量为$[\alpha,\beta]$范围时以$F(|I|)$的概率计算交集,F为一个递增函数,该函数的输出范围为$[0,1]$,$|I|$为参与方的交集数量。当$|I|\geqslant\beta$时,计算交集的概率$F(|I|)\geqslant 1-\delta$,当$|I|\leqslant\alpha$时,计算交集的概率$F(|I|)\leqslant\delta$,δ为一个可忽略函数。为实现PTPSI

协议，Liu等人[26]将经典的Goldwasser-Sipser[27]两方证明系统扩展到多方场景，设计出一种概率集合大小测试(Probabilistic Set Size Test，PSST)方法，但该方法需要依赖昂贵的多方通用电路，计算开销与参与方数量呈指数关系，使得其无法适应于多参与方场景，成为整个协议的瓶颈。

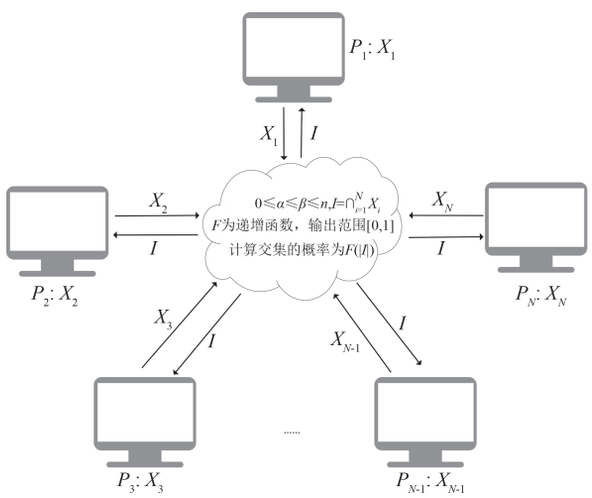

图1　(α,β)－PTPSI示意图

为解决上述问题，本文的主要贡献如下。

(1)提出一种新的多方概率集合大小测试方法。基于双中心零共享技术，将多方概率集合大小测试归结到两个中心参与方，有效降低参与方之间交互的通信量与计算开销。通过验证两方概率集合大小测试是否通过，来决定是否计算交集。基于新的多方概率集合大小测试方法，设计出一种可扩展的多方概率门限隐私集合交集协议，并且没有使用开销较大的公钥算法。

(2)为抵抗中心参与方合谋，提出一种隐私增强的多方概率集合大小测试方法。利用不经意伪随机函数来限制合谋方的恶意查询，进而设计一种隐私增强的多方概率门限隐私集合交集协议。

(3)通过实验仿真，本文在相同的环境下复现了文献[26]协议的实验，并与所提出方案进行对比，以证明协议的有效性。对于(1)中的概率门限隐私集合交集协议，在5方场景下，每个参与方集合大小为$n=2^{20}$，门限值设为$0.5n$，文献[26]的协议运行时间为45.40秒，通信为3403.12 MB，改进后的协议在相同条件下总时间为9.62秒，通信量为187.39 MB，相比之下，速度提升4.72倍，当参与方数量从5方扩展到32方时，协议总时间为9.92秒，通信量为773.90 MB，具有较高的可扩展性。

2　相关工作

现有的TPSI协议主要集中在两方场景[17-21]，对于多方场景下的研究[23-26]仍存在计算开销大或隐私泄露等问题。下面做具体分析。

首先介绍两方场景下的TPSI工作，Freedman等人[1]首次提出了TPSI的概念，但是他们没有给出具体代码实现。Zhao等人[19]提出了一种TPSI的具体实现方法，他们设计了一种门限秘密传输协议，该协议可以通过多项式评估或者混淆布隆过滤器实现，进而可以构造出两方门限隐私集合交集协议。但是该协议会泄露交集基数。

Hallgren等人[17]解决了交集泄露问题并将门限隐私集合交集应用于共享拼车领域。他们基于半同态加密方案设计了一种两方TPSI协议，该协议通信复杂度为$O(n^2)$，计算复杂度为$O(n^3)$，这使得在长距离拼车场景下，协议的性能较差。

Ghosh等人[20]进一步优化TPSI的通信开销，他们指出TPSI协议的通信下界仅依赖门限值t。他们提出一种新型的不经意线性函数评估方案，并通过代数方法设计出一种两方TPSI协议，该协议的通信复杂度为$O(t^2)$。但他们对多方场景下的TPSI协议仅进行了理论分析，并没有给出具体实现，且由于需要大量公钥操作，协议的计算开销较大。

Hu等人[21]将TPSI应用拓展到云计算领域。利用第三代全同态加密技术，在半诚实场景下构造了一种安全的两方门限隐私集合交集协议，客户端计算与通信复杂度均为$O(n)$，能够应用在客户端集合较小、服务器集合较大的典型云计算场景下。然而由于协议使用大量同态操作，使得其无法适用于客户端拥有大输入集合的场景。

接下来对多方场景下TPSI的工作进行介绍，Badrinarayanan等人[23]对文献[20]的协议进行扩展，设计出了一个多方TPSI协议。该协议采用门限全同态加密实现，其通信复杂度为$O(Nt)$，N为参与方的数量。由于门限全同态加密涉及复杂的加解密和同态计算操作，对设备计算资源要求较高，导致协议计算效率较低。

张恩等人[24]基于弹性秘密共享方法，结合布隆过滤器设计了一种轻量级的TPSI协议，然后通过不经意传输，实现抵抗参与方合谋的目的。他们协议的通信复杂度为$O(N^2m)$，m为混淆布隆过滤器的长度，由于多项式的重构需要大量计算开销，他们

的协议在大数据集场景下效率较低。

Ghosh等人[25]对多方TPSI协议进行了理论分析，将文献[20]中的通信复杂度从$O(t^2)$降低至$O(t\text{polylog}(t))$。他们从加法同态加密和不经意传输等简单假设出发构建了一个多方TPSI协议，避免了对昂贵的全同态加密的依赖，降低了计算成本。但他们的协议仍处于理论研究阶段，并没有给出具体的实现。

为解决以上问题，Liu等人[26]于2023年首次提出概率门限隐私集合交集的概念，这是TPSI的一种概率变体，该协议会以一定的概率去计算交集，当交集数量越多时，计算交集的概率就会越大。他们的协议巧妙的绕过了Ghosh等人[20]提出的TPSI的通信下界。他们将经典的Goldwasser-Sipser协议[27]推广到多方场景，设计了一种多方概率集合大小测试协议，尽管该协议在参与方数量较少时表现出较高的效率，但由于其核心构建块交集计数功能函数($\mathcal{F}_{\cap-\text{count}}$)依赖于昂贵的多方通用电路，协议在多方场景下，其计算开销与参与方数量呈指数关系，这一特性在实际应用中极具挑战性。例如，在涉及10个以上参与方的联合分析场景中，现有协议所需的计算资源和等待时间将远超可接受范围，严重制约其在大规模医疗协作、拼车、数据挖掘等高隐私任务中的实用性。

针对此问题，本文首先在两方概率集合大小测试的基础上提出了高效的多方PTPSI协议，针对多个参与方，首先进行多方概率门限测试，若测试通过，则各参与方进行多方PSI协议得到交集；否则，协议终止。实验表明，随着参与方增多，协议计算效率不受影响。为抵抗两个中心参与方的合谋攻击，提出一种隐私增强的多方PTPSI协议，通过使用不经意伪随机函数来限制合谋参与方的恶意查询。算法分析与实验结果显示，本文所设计的协议与现有的方案相比具有更好的性能。

3 基础知识

本文设计的协议在半诚实模型下主要基于两两独立的哈希函数、Goldwasser-Sipser协议、不经意伪随机函数、不经意键值存储、双中心零共享技术，下面介绍以上技术的相关概念及基础知识，所使用的符号及说明如表1所示。

表1 符号说明表

符号	说明	符号	说明
κ	计算安全参数	$\mathbb{N}$	非负整数集
λ	统计安全参数	$\|D\|$	集合D中元素个数
N	参与方数量	$\oplus$	逐比特异或
t	门限值	$\bigoplus_{i=1}^{N}x_i$	元素$x_1\oplus x_2\oplus\cdots\oplus x_N$
q	合谋方数量	$\bigcap_{i=1}^{N}X_i$	集合$X_1,\cdots,X_N$的交集
n	参与方集合大小	$[n]$	集合$\{1,2,\cdots,n\}$
P_i	参与方i	$[a,b]$	整数集合$\{a,a+1,\cdots,b\}$
X_i	参与方i的集合	$s^{i,j}$	P_i作为发送方与P_j执行OPRF协议产生的密钥
X_i^j	参与方i的第j个集合	m	哈希函数的输入位长
x_i^j	第j个参与方的第i个元素	l	哈希函数的输出位长
T_i	参与方i的OKVS数据结构	$\mathcal{S}$	键值对集合
∅或⊥	一个空集	Sim	模拟器

3.1 概率门限隐私集合交集

概率门限隐私集合交集是指N个参与方，每个参与方拥有一个集合，PTPSI允许参与方在交集数量$|I|$为$[\alpha,\beta]$范围时以$F(|I|)$的概率计算交集，F为一个递增函数，该函数的输出范围为$[0,1]$，$|I|$为参与方的交集数量。(α,β)－PTPSI的理想功能函数$\mathcal{F}_{\text{PTPSI}}$如下。

参数：参与方的数量为N，每个参与方拥有的集合元素个数为n。令函数$F(\cdot):\mathbb{N}\to[0,1]$是一个递增的函数，该函数输出区间为$[0,1]$。$\delta$为可忽略函数，令$0<\alpha<\beta\leqslant n$，使得$F(\alpha)<\delta$，$F(\beta)>1-\delta$。

输入：等待每个参与方$P_i(i\in[1,N])$输入自己的集合$X_i=\{x_1^i,\cdots,x_n^i\}\subseteq\{0,1\}^*$。

采样一个均匀随机的字符串r。

使用r从伯努利分布中采样一个随机比特b，该伯努利分布输出为1的概率为函数值$F(|I|)$，其中

$I=\bigcap_{i=1}^{N}X_i$，代表参与方的交集。

输出：若$b=1$，输出交集(I,r)；否则输出$(\perp,r)$。

3.2　两两独立的哈希函数

定义 1.　首先假设存在一个哈希函数簇$\mathcal{H}_{m,l}=\{h:\{0,1\}^m\to\{0,1\}^l\}$，这里的参数$l<m$，如果对于所有的$x\neq x'\in\{0,1\}^m$和$y,y'\in\{0,1\}^l$，满足以下条件：

$$\Pr_{h\leftarrow\mathcal{H}_{m,l}}\left[h(x)=y\wedge h(x')=y'\right]=2^{-2l} \tag{1}$$

则称其为两两独立的哈希函数。

引理 1.　存在一个两两独立的哈希函数簇$\mathcal{H}_{m,l}$，大小为2^{2m}，能够使用$2m$比特的随机串构造函数$h\in\mathcal{H}_{m,l}$。

证明.　详见附录A。

本文使用两两独立的哈希函数，该函数满足独立性要求，又便于快速实现，且在数学上拥有良好的概率特性，便于在设计协议时进行概率分析。

3.3　Goldwasser-Sipser(GS)协议

Goldwasser 和 Sipser 提出了一个经典的 Arthur-Merlin 协议[27]，Merlin作为证明者，他被给定一个集合$D\subseteq\{0,1\}^m$，他需要向作为验证者的Arthur证明集合D中的元素个数大于一个确定的边界。GS协议允许验证者通过以下步骤确定$|D|\geqslant t$或者$|D|\leqslant\frac{t}{2}$，其中t为门限值。先设置一个共同的参数l，l需要满足以下条件：$2^{l-2}\leqslant t\leqslant 2^{l-1}$，令$p=t\cdot 2^{-l}\in\left[\frac{1}{4},\frac{1}{2}\right]$。具体交互步骤如下：

(1) 验证者随机选择一个哈希函数$h\leftarrow\mathcal{H}_{m,l}$和一个点$y\leftarrow\{0,1\}^l$，把$(h,y)$发给证明者。

(2) 证明者寻找$x\in D$且$h(x)=y$，然后把x发送给验证者，如果找不到符合条件的x，证明者终止协议。

(3) 验证者检查x的成员属性以及是否$h(x)=y$，若满足上述条件，则接受。

引理 2.　如果$|D|\geqslant t$，那么验证者接受的概率至少为$\frac{3p}{4}$，如果$|D|\leqslant\frac{t}{2}$，那么验证者接受的概率至多为$\frac{p}{2}$。

证明.　详见附录B。

可以通过重复实验来减少两种情况下的随机误差和不确定性。使用(k,τ,l)－GS表示在重复执行k次GS协议(参数为l)后，如果验证者发现有τ次通过，那么验证者就接受。此时的(k,τ,l)－GS协议验证者接受的概率取决于集合D的大小。

定义 2.　设$D\subseteq\{0,1\}^m$，然后定义一个函数$F_{(k,\tau,l)-\mathrm{GS}}(\cdot):\mathbb{N}\to[0,1]$，函数输入为$D$中元素个数，输出$(k,\tau,l)$－GS协议中验证者接受的概率。

定理 1.　设$2^{l-2}\leqslant t\leqslant 2^{l-1}$，令$\alpha=\frac{t}{2}$，$\beta=t$和$\tau=\frac{5p}{8}k$，这里$p=t\cdot 2^{-l}\in\left[\frac{1}{4},\frac{1}{2}\right]$，当函数输入为$\beta$时，$F_{(k,\tau,\ell)-\mathrm{GS}}(\beta)\geqslant 1-e^{-\frac{pk}{96}}$，当函数输入为$\alpha$时，$F_{(k,\tau,\ell)-\mathrm{GS}}(\alpha)\leqslant e^{-\frac{pk}{72}}$。

可使用切尔诺夫界[28]进行证明。详见附录C。

需要说明一下参数之间的关系，当确定了集合D中元素的长度以及门限值t之后，可以设置一个参数l，l需要满足以下条件：$2^{l-2}\leqslant t\leqslant 2^{l-1}$。进而可以计算$p=t\cdot 2^{-l}\in\left[\frac{1}{4},\frac{1}{2}\right]$。当确定安全参数$\lambda$之后，通过定理1，可以计算出$(k,\tau,l)$－GS协议至少需要重复的轮数$k$。

3.4　不经意伪随机函数

不经意伪随机函数(Oblivious Pseudorandom Function，OPRF)是一个安全两方计算协议，涉及接收方和发送方。当前基于不经意传输技术设计的OPRF效率最高，其可分为单点OPRF[29,30]和多点OPRF[2,31-32]。单点OPRF允许接收方输入一个元素x，而多点OPRF允许接收方输入一个集合$X=\{x_1,\cdots,x_n\}$。本文使用多点OPRF作为基本构建块。协议允许接收方输入一个集合$\{x_1,\cdots,x_n\}$，接收方可以得到伪随机函数值集合$\{F_s(x_1),\cdots,F_s(x_n)\}$，$F_{(\cdot)}(\cdot)$为一个伪随机函数；发送方得到伪随机函数密钥$s$。由于接收方无法得到伪随机密钥$s$，当使用OPRF值对经过编码之后的数据结构进行查询时，接收方只能使用$\{F_s(x_1),\cdots,F_s(x_n)\}$来查询自己的输入集合$\{x_1,\cdots,x_n\}$所对应的值，从而达到限制接收方对数据结构进行无限查询的目的。理想功能函数$\mathcal{F}_{\mathrm{OPRF}}$如下：

参数：发送方S，接收方R，一个伪随机函数$F_{(\cdot)}(\cdot):\{0,1\}^\kappa\times\{0,1\}^*\to\{0,1\}^m$，计算安全参数$\kappa$。

输入：R输入集合$\{x_1,\cdots,x_n\}\subseteq\{0,1\}^*$。

输出：R得到伪随机函数输出$\{F_s(x_i)\}_{i\in[n]}$，S得到伪随机密钥s。

3.5　不经意键值存储

键值对存储(Key-Value Stores，KVS)是一种

安全的数据结构，主要由编码算法Encode(•)和解码算法Decode(•, •)组成[33-35]：

Encode(•)：编码算法要求输入键值对集合$\mathcal{S}=\{(x_i,v_i)\}_{i\in[n]}$，得到一个数据结构$T$。

Decode(•, •)：解码算法要求输入数据结构T，一个查询键x'，得到一个值v'。

KVS的正确性需要保证对任意键值对集合$\mathcal{S}$：

Pr[Encode($\mathcal{S}$)=$\perp$]可以忽略不计。

当Encode($\mathcal{S}$)=$T\neq\perp$，而且键值对$(x,v)\in\mathcal{S}$，Decode($\mathcal{S},x$)=v。

定义3. 如果值的集合$\{v_1,\cdots,v_n\}$随机生成，编码键值对集合$\mathcal{S}_1=\{(x_1^1,v_1),\cdots,(x_n^1,v_n)\}$与$\mathcal{S}_2=\{(x_1^2,v_1),\cdots,(x_n^2,v_n)\}$所得到的结果在计算意义上是不可区分的，此时KVS是一个不经意键值存储(Oblivious Key-Value Stores，OKVS)[34]。

定义4. 如果OKVS的解码算法满足：Decode(T,x)=$\bigoplus_{i=1}^{N}$Decode(T_i,x)，数据结构T满足以下条件：$T=\bigoplus_{i=1}^{N}T_i$，解码算法可以把任何与T格式相同的数据结构作为输入，此时的OKVS满足线性性质[34]。

3.6 双中心零共享

Gao等人[36]于2024年提出一个双中心零共享(Bicentric Zero-Sharing，BZS)原语，主要思想是让每个参与方的每个元素对应一个秘密份额值，然后将这些秘密份额值由两个中心参与方进行匹配与聚合，其理想功能函数$\mathcal{F}_{\text{BZS}}$如下：

参数：现有N个参与方$P_1,\cdots,P_N$，每个参与方$P_i(i\in[1,N])$拥有一个集合X_i，秘密份额的长度为m。

输入：等待每个参与方$P_i(i\in[1,N])$输入$X_i=\{x_1^i,\cdots,x_n^i\}\subseteq\{0,1\}^*$。

输出：集合$A=\{a_1,\cdots,a_n\}\leftarrow\{\{0,1\}^m\}^n$和集合$B=\{b_1,\cdots,b_n\}\leftarrow\{\{0,1\}^m\}^n$，满足当$x_j^{N-1}=x_j^N\in\bigcap_{i=1}^{N}X_i$时$a_j=b_j$。然后把$A$给$P_{N-1}$，把$B$给$P_N$。

$\mathcal{F}_{\text{BZS}}$的正确性需要保证当$x\in\bigcap_{i=1}^{N}X_i$时，设此时P_{N-1}中的某个键值对(x_i^{N-1},a_i)满足$x_i^{N-1}=x$，P_N中的键值对(x_j^N,b_j)满足$x_j^N=x$，那么此时$a_i=b_j$。

3.7 安全模型

本文协议在半诚实模型下[37,38]是安全的。在半诚实模型下，参与方诚实的执行协议，但会在协议执行过程中记录得到的中间信息，并通过分析其获得的所有信息来推测其他参与方的隐私。当协议中参与方可以计算的任何内容均来源于其输入和输出时，那么该协议被视为安全的。参与方的视图在给定输入输出的情况下可以被模拟。

定义5. 令$f:(\{0,1\}^*)^N\rightarrow(\{0,1\}^*)^N$，为一个$N$元函数，$f_i(X_1,\cdots,X_N)$表示$f(X_1,\cdots,X_N)$的第$i$个分量，对于索引集合$C=\{i_1,\cdots,i_q\}\subseteq[1,N]$，定义$f_C(X_1,\cdots,X_N)$为$f$在索引$C$处的子序列，即$f_{i_1}(X_1,\cdots,X_N),\cdots,f_{i_q}(X_1,\cdots,X_N)$。设$\pi$为计算$f$的多方协议，令$\vec{X}=(X_1,\cdots,X_N)$为参与方的输入。协议执行过程中第$i$个参与方的视图记为$\text{VIEW}_i^{\pi}(\vec{X})$。对于索引集合$C=\{i_1,\cdots,i_q\}$，此时协议执行过程中参与方$C=\{i_1,\cdots,i_q\}\subseteq[1,N]$的视图可以表示为：

$$\text{VIEW}_C^{\pi}(\bar{X})=(C,\text{VIEW}_{i_1}^{\pi}(\bar{X}),\cdots,\text{VIEW}_{i_q}^{\pi}(\bar{X})) \quad (2)$$

若f为一个确定型函数，若存在概率多项式时间算法S，其对于任意腐败参与方的索引集合$C=\{i_1,\cdots,i_q\}\subseteq[N]$满足以下条件：

$$\{\text{S}(C,(X_{i_1},\cdots,X_{i_q}),f_C(\bar{X}))\}\overset{c}{\equiv}\{\text{VIEW}_C^{\pi}(\bar{X})\} \quad (3)$$

则协议π在半诚实敌手存在的情况下是安全的。$\overset{c}{\equiv}$表示计算不可区分。

4 高效可扩展的多方PTPSI协议

本章提出一种可拓展的高效多方PTPSI协议，4.1节描述Liu等人[26]提出的两方概率集大小测试协议，4.2节提出本文的高效多方PTPSI协议，4.3节给出具体的安全证明。

4.1 两方概率集合大小测试协议

本节给出两方概率集合大小测试(Two-party PSST，TPSST)的理想功能函数$\mathcal{F}_{\text{TPSST}}$和具体协议$\pi_{\text{TPSST}}$。

GS协议可以用来实现TPSST。设Y_1和Y_2是参与方P_1,P_2的输入集合，交集$I=Y_1\cap Y_2$。GS协议的第一步是验证者选择哈希函数发送给证明者计算。TPSST可以让两方通过投币算法共同抽样一个两两独立的哈希函数$h\leftarrow\mathcal{H}_{m,l}$，各自计算满足哈希函数条件的输入集。GS协议的第二步是证明者把符合条件的元素发送验证者进行验证，TPSST可以通过使用一个交集计数功能函数$\mathcal{F}_{\cap-\text{count}}$来验证两方的交集数量是否超过一个确定的门限值。

TPSST的一个核心组件为交集计数功能函数$\mathcal{F}_{\cap-\text{count}}$，$\mathcal{F}_{\cap-\text{count}}$如图2所示，其主要思想如算法1所

示。可以使用SPDZ电路框架[39-41]来实现这一理想功能函数。基于交集计数功能函数$\mathcal{F}_{\cap-\text{count}}$可以得到一种安全高效的TPSST协议。

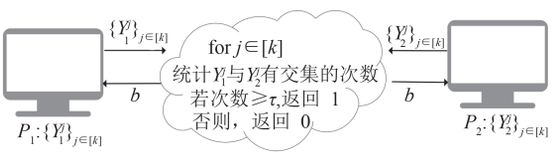

图2　$\mathcal{F}_{\cap-\text{count}}$示意图

算法1.　$\mathcal{F}_{\cap-\text{count}}$交集计数算法

输入：P_1, P_2输入集合$\{Y_1^j\}_{j\in[k]}$, $\{Y_2^j\}_{j\in[k]}$, k和τ。

输出：一个比特b。

1.　$sum=0$
2.　FOR $j=1$ TO k DO
3.　　IF$\{Y_1^j\}$与$\{Y_2^j\}$有交集
4.　　$sum++$
　　CONTINUE//此轮循环结束,继续下一轮
5.　　END IF
6.　END FOR
7.　IF $sum\geqslant\tau$
8.　　RETURN 1
9.　ELSE
10.　　RETURN 0

两方概率集合大小测试协议的理想功能函数$\mathcal{F}_{\text{TPSST}}$如下。

参数：现有两个参与方$P_i(i\in[1,2])$分别持有一个集合$X_i=\{x_1^i,\cdots,x_n^i\}\subseteq\{0,1\}^*$，函数$F(\bullet)$：$\mathbb{N}\to[0,1]$是递增的函数，该函数输出区间为$[0,1]$。

输入：P_1输入集合X_1，P_2输入集合X_2。

采样一个随机字符串r。

使用r从伯努利分布中采样一个随机比特b，该伯努利分布输出1的概率为$F(|I|)$，这里的$I=\bigcap_{i=1}^{N}X_i$，代表参与方的交集。

输出：输出(b,r)。

两方概率集合大小测试协议π_{TPSST}实现如下。

参数：现有两个参与方$P_i(i\in[1,2])$分别持有一个集合$X_i=\{x_1^i,\cdots,x_n^i\}\subseteq\{0,1\}^m$，$k,\tau,l$为整数。协议具体步骤如下：

(1) 对于$j\in[k]$(并行执行)

所有参与方共同抽样一个两两独立的哈希函数$h\leftarrow\mathcal{H}_{m,l}$和一个点$y\leftarrow\{0,1\}^l$。每个参与方$P_i(i\in[1,2])$计算$X_i^j=\{x\in X_i:h(x)=y\}$。

(2) P_1, P_2共同调用理想功能函数$\mathcal{F}_{\cap-\text{count}}$，输入为步骤(1)计算的集合$\{X_1^j\}_{j\in[k]}$和$\{X_2^j\}_{j\in[k]}$，以及参数$k,\tau$。

(3) 所有参与方从$\mathcal{F}_{\cap-\text{count}}$中得到一个输出比特$b$。

在两方场景下该协议可以用来判断交集数量是否达到预设门限，虽然在调用$\mathcal{F}_{\cap-\text{count}}$时所使用的SPDZ电路框架对于大输入集合效率较低，但由于GS协议所使用的两两独立的哈希函数减少了$\mathcal{F}_{\cap-\text{count}}$中每个参与方$P_i(i\in[1,2])$输入集$\{X_i^j\}_{j\in[k]}$的元素个数，因此每次比较集合中包含的元素个数为$O(1)$(可通过切比雪夫不等式进行证明，详见附录D)，两方场景下该协议是高效的。

若π_{TPSST}协议输出比特为1，代表参与方交集数量达到预设门限值，那么就可以执行一个标准的PSI协议来获取交集。协议的安全性已在文献[26]中得到证明。

4.2　多方概率门限隐私集合交集协议

本节提出一种高效的多方PTPSI协议，基本思想是首先让所有参与方调用$\mathcal{F}_{\text{BZS}}$，$\mathcal{F}_{\text{BZS}}$主要使用算法2实现，P_{N-1}获得集合A，A中元素a_j与P_{N-1}的输入集合X_{N-1}中每个元素$x_j^{N-1}(j\in[n])$为一一对应关系。P_N获得集合B，集合B中元素与P_N输入集中每个元素也为一一对应关系。然后让P_{N-1}与P_N调用$\mathcal{F}_{\text{TPSST}}$，如果未通过测试，则结束协议；如果通过测试，为防止P_N的恶意查询，不能直接将集合A发送给P_N，需要让P_N作为接收方与P_{N-1}运行一次OPRF协议，把P_N的查询范围固定为集合B。之后P_{N-1}将集合$\{F_s(a_j)\}_{j\in[n]}$发送给P_N，P_N通过计算对比得到交集I。交集查找算法如算法3所示，协议π_{PTPSI}的流程图如图3所示。

协议π_{PTPSI}如下。

参数：现有N个参与方$P_1,\cdots,P_N$，每个参与方P_i拥有一个集合$X_i=\{x_1^i,\cdots,x_n^i\}\subseteq\{0,1\}^*$，计算安全参数$\kappa$，统计安全参数$\lambda$，$k,\tau,l$是三个整数，OKVS编码算法Encode$(\bullet)$和解码算法Decode$(\bullet,\bullet)$，理想功能函数$(\mathcal{F}_{\text{BZS}},\mathcal{F}_{\text{TPSST}},\mathcal{F}_{\text{OPRF}})$。协议过程如下。

份额共享阶段：

(1) 每个参与方$P_i(i\in[1,N])$使用他们的输入集$X_i(i\in N)$共同调用$\mathcal{F}_{\text{BZS}}$，P_{N-1}获得集合A，P_N获得集合B。

门限测试阶段：

(2) P_{N-1}与P_N调用$\mathcal{F}_{\text{TPSST}}$，$P_{N-1}$输入集合$A$，$P_N$输入集合$B$。

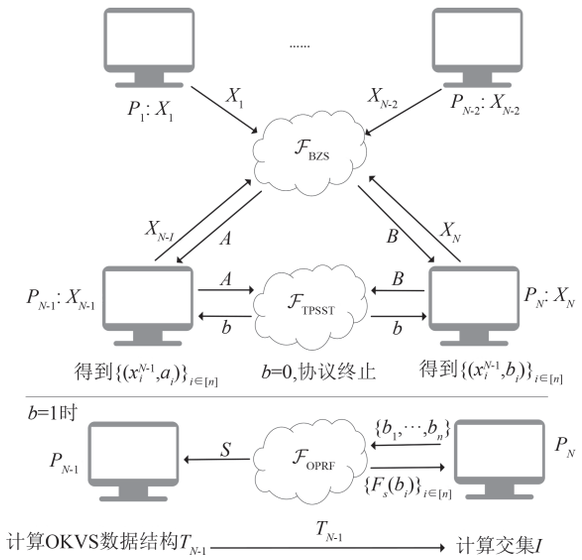

图3 π_{PTPSI}协议流程图

(3) 从步骤(2)得到输出比特b,如果$b=0$,终止协议。如果$b=1$,执行下一步。

求交集阶段:

(4) P_N作为接收方与P_{N-1}调用$\mathcal{F}_{OPRF}$,P_N的输入为集合$B=\{b_1,\cdots,b_n\}$,得到输出集合$\{F_s(b_i)\}_{i\in[n]}$;P_{N-1}得到OPRF密钥s,计算集合$\{F_s(a_i)\}_{i\in[n]}$。P_{N-1}得到一个键值对集合$\mathcal{S}_{N-1}=\{(a_j,F_s(a_j))\}_{j\in[n]}$。

(5) P_{N-1}使用OKVS编码算法Encode($\mathcal{S}_{N-1}$)计算得到一个数据结构T_{N-1},将T_{N-1}发送给P_N,P_N使用OKVS解码算法计算Decode(T_{N-1},b_j),求得交集$I=\{x_j^N|F_s(b_j)\oplus\text{Decode}(T_{N-1},b_j)=0\}_{j\in[n]}$。

算法2. 双中心零共享算法

输入:每个参与方$P_i(i\in[1,N])$输入集合X_i。

输出:输出集合A给P_{N-1},输出集合B给P_N。

1. //P_N初始化
2. FOR $i=1$ TO n DO
3. $b_i=PRF(x_i^N)$//PRF代表伪随机函数
4. END FOR //P_N初始化集合B
5. $T_N=$Encode($\{(x_i^N,b_i)\}_{i\in[n]}$)//Encode表示编码算法
6. FOR $i=1$ TO $N-2$ DO
7. $seed_i\leftarrow\{0,1\}^\kappa$//$seed$表示种子
8. $T_i=PRG(seed_i)$//PRG表示伪随机生成器
9. END FOR //P_N初始化发送给$P_i(i\in[N-2])$的种子
10. $T_{N-1}=T_N\oplus\oplus_{j=1}^{N-2}T_j$//计算发送给$P_{N-1}$的数据结构
11. //P_N把$seed_i$发送给$P_i(i\in[N-2])$,T_{N-1}发给P_{N-1}
12. //$P_i(i\in[1,N-1])$的执行步骤
13. FOR $i=1$ TO $N-1$ DO
14. IF $i>=1\&\&i<N-1$//$P_i(i\in[N-2])$
15. $T_i=PRG(seed_i)$//P_i计算T_i
16. FOR $j=1$ TO n DO
17. $y_j^i=$Decode(T_i,x_j^i)//解码算法
18. END FOR//$P_i(i\in[N-2])$执行解码算法
19. $T_i=$Encode($\{(x_j^i,y_j^i)\}_{j\in[n]}$)//$T_i$发给$P_{N-1}$
20. ELSE IF $i==N-1$//P_{N-1}的执行算法
21. $T=\oplus_{e=1}^{N-1}T_e$//OKVS的线性
22. FOR $j=1$ TO n DO
23. $a_j=$Decode(T,x_j^{N-1})
24. END FOR
25. $A=\{a_1,\cdots,a_n\}$//P_{N-1}计算得到集合A
26. END IF
27. END FOR

算法3. 交集查找算法

输入:数据结构T_{N-1},集合X_N,B和集合$\{F_s(b_i)\}_{i\in[n]}$。

输出:交集集合I。

1. $I=\varnothing$
2. FOR $i=1$ TO n DO
3. IF Decode(T_{N-1},b_i)$==F_s(b_i)$
4. ADD x_i TO I
5. END IF
6. END FOR
7. RETURN I

4.3 安全证明

定理2. 在($\mathcal{F}_{BZS}$,$\mathcal{F}_{TPSST}$,$\mathcal{F}_{OPRF}$)混合模型下,当P_N与P_{N-1}不合谋时本文4.2节的协议实现了3.1节中针对半诚实敌手的理想功能函数$\mathcal{F}_{PTPSI}$。

证明. 本文首先证明该协议的正确性,然后证明安全性。

正确性:参与方首先调用$\mathcal{F}_{BZS}$,根据$\mathcal{F}_{BZS}$的正确性知,当x_j^N为各参与方的交集元素时,b_j为集合A与B的交集元素。然后P_{N-1}与P_N调用$\mathcal{F}_{TPSST}$,当交集数量足够多时,$\mathcal{F}_{TPSST}$返回结果为1,代表通过门限测试。接下来P_N作为OPRF接收方与P_{N-1}调用$\mathcal{F}_{OPRF}$,当$a_i=b_j$时,$F_s(a_i)=F_s(b_j)$。当$a_i\neq b_j$,$F_s(a_i)$对于P_N来说是随机值r。根据OKVS的正确性,P_{N-1}编码集合$\{(a_i,F_s(a_i))\}_{i\in[n]}$:

$$T_{N-1}=\text{Encode}(\{(a_j,F_s(a_j))\}_{j\in[n]}) \quad (4)$$

编码失败的概率可忽略。由于P_N无法得到OPRF密钥s,因此P_N只能在其输入集合B的范围内,对P_{N-1}发送过来的数据结构T_{N-1}进行正确解码,

此时：

$$I=\{x_j^N|F_s(b_j)\oplus\text{Decode}(T_{N-1},b_j)=0\}_{j\in[n]} \quad (5)$$

结果可以分为以下两种情况。

交集的情况，假设$a_j=b_{j'}$：

$$\begin{aligned}&F_s(b_{j'})\oplus\text{Decode}(T_{N-1},b_{j'})=\\&F_s(b_{j'})\oplus F_s(a_j)=0\end{aligned} \quad (6)$$

非交集的情况：

$$\begin{aligned}&F_s(b_j)\oplus\text{Decode}(T_{N-1},b_j)=\\&F_s(b_j)\oplus r\end{aligned} \quad (7)$$

综上，协议π_{PTPSI}的输出与$\bigcap_{i=1}^{N}X_i$的输出保持一致，正确性得证。

安全性：本文采用理想-现实模型进行形式化证明。用C表示合谋参与方集合，$|C|=q$，H表示诚实参与方的集合，构造合谋方集合的模拟器Sim。需要说明，当合谋方集合C中只有一个参与方$P_i(i\in[1,N])$时，表示单个参与方被腐败的情况，下面对合谋情况的分析已包含单个参与方腐败的情形。接下来对合谋参与方集合C的视图进行模拟，本文分为以下三种情况进行讨论。

(1)P_N参与合谋，P_{N-1}诚实，则P_N在合谋方集合C，P_{N-1}在诚实方集合H中，此时C的模拟视图为：$\{\text{Sim}(C,(X_{i_1},\cdots,X_{i_q}),I)\}$。

(2)P_{N-1}参与合谋，P_N诚实，则P_{N-1}在合谋方集合C，P_N在诚实方集合H中，此时C的模拟视图为：$\{\text{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)\}$。

(3)P_{N-1}与P_N均诚实，则P_{N-1}与P_N均在诚实方集合H中，此时合谋方集合C的模拟视图为：$\{\text{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)\}$。

第一种情况：P_N被腐败，P_{N-1}诚实。则$P_N\in C$，$|C|=q$，$P_{N-1}\in H$，模拟器Sim接收到理想世界中$\mathcal{F}_{\text{PTPSI}}$的输出$I$以及$C$中每个合谋参与方$P_i$的集合$X_i$。模拟器$\text{Sim}(C,(X_{i_1},\cdots,X_{i_q}),I)$运行如下：

(1)Sim采样n个随机值$b_j'(j\in[n])$，与P_N的集合X_N中元素$x_j^N(j\in[n])$一一对应，令$B'=\{b_j'\}_{j\in[n]}$，集合$I'=\{b_j'|x_j^N\in I\}$。

(2)Sim接下来调用双中心零共享模拟器$\text{Sim}_{\text{BZS}}(C,(X_{i_1},\cdots,X_{i_q}),B')$，将$B'$添加到合谋参与方的视图中。

(3)Sim调用两方概率集大小测试模拟器$\text{Sim}_{\text{TPSST}}^{P_N}(B',b)$，将$b$添加到合谋参与方的视图中。

(4)Sim使用B'作为输入，调用两方不经意伪随机函数的模拟器$\text{Sim}_{\text{OPRF}}^{P_N}(B',\{F_s(b_j')\}_{j\in[n]})$，然后把$\{F_s(b_j')\}_{j\in[n]}$添加到合谋参与方的视图中。

(5)Sim模拟P_{N-1}发送给P_N的数据结构T_{N-1}，首先计算键值集合$\mathcal{S}_{N-1}=\{(b_j',F_s(b_j'))\}_{b_j'\in I'}$，并随机采样$n-|I|$个键值对到$\mathcal{S}_{N-1}$中，使得$|\mathcal{S}_{N-1}|=n$，使用OKVS编码算法$\text{Encode}(\mathcal{S}_{N-1})$得到数据结构$T_{N-1}'$，然后把$T_{N-1}'$添加到合谋方视图中。

因为在混合模型中对于合谋参与方而言$\text{Sim}_{\text{BZS}}(C,(X_{i_1},\cdots,X_{i_q}),B')$与$\text{VIEW}_C^{\pi_{\text{BZS}}}(X_1,\cdots,X_N)$在计算意义上无法区分，$\text{Sim}_{\text{TPSST}}^{P_N}(B',b)$与$\text{VIEW}_{P_N}^{\pi_{\text{TPSST}}}(A,B)$在计算意义上无法区分，以及$\text{Sim}_{\text{OPRF}}^{P_N}(B',\{F_s(b_j')\}_{j\in[n]})$与$\text{VIEW}_{P_N}^{\pi_{\text{OPRF}}}(A,B)$在计算意义上无法区分，所以合谋方$C$的视图在理想世界与现实世界是无法区分的：

$$\begin{aligned}&\{\text{Sim}(C,(X_{i_1},\cdots,X_{i_q}),I)\}\overset{c}{\equiv}\\&\{\text{VIEW}_C^{\pi_{\text{PTPSI}}}(X_1,\cdots,X_N)\}\end{aligned} \quad (8)$$

第二种情况：P_{N-1}被腐败，P_N诚实。则$P_{N-1}\in C$，$|C|=q$，$P_N\in H$，Sim接收到C中每个合谋参与方P_i的集合X_i。模拟器$\text{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)$运行如下。

(1)Sim采样n个随机值a_j'，得到一个集合$A'=\{a_j'\}_{j\in[n]}$。

(2)Sim接下来调用双中心零共享模拟器$\text{Sim}_{\text{BZS}}(C,(X_{i_1},\cdots,X_{i_q}),A')$，将$A'$添加到合谋参与方的视图中。

(3)Sim调用两方概率集大小测试模拟器$\text{Sim}_{\text{TPSST}}^{P_{N-1}}(A',b)$，将$b$添加到合谋参与方的视图中。

(4)Sim调用两方不经意伪随机函数模拟器$\text{Sim}_{\text{OPRF}}^{P_{N-1}}(\perp,s)$，把$s$添加到合谋参与方的视图中。

因为在混合模型中对于合谋参与方集合而言$\text{Sim}_{\text{BZS}}(C,(X_{i_1},\cdots,X_{i_q}),A')$与$\text{VIEW}_C^{\pi_{\text{BZS}}}(X_1,\cdots,X_N)$，$\text{Sim}_{\text{TPSST}}^{P_{N-1}}(A',b)$与$\text{VIEW}_{P_{N-1}}^{\pi_{\text{TPSST}}}(A,B)$，$\text{Sim}_{\text{OPRF}}^{P_{N-1}}(\perp,s)$与$\text{VIEW}_{P_{N-1}}^{\pi_{\text{OPRF}}}(A,B)$在计算意义上无法区分，所以合谋参与方$C$的视图在理想世界与现实世界是无法区分的：

$$\begin{aligned}&\{\text{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)\}\overset{c}{\equiv}\\&\{\text{VIEW}_C^{\pi_{\text{PTPSI}}}(X_1,\cdots,X_N)\}\end{aligned} \quad (9)$$

第三种情况：P_{N-1}与P_N均为诚实参与方，那么$P_{N-1},P_N\in H$，Sim接收到C中每个合谋参与方P_i的集合X_i。模拟器$\text{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)$运行如下。

(1) Sim直接调用双中心零共享模拟器$\text{Sim}_{\text{BZS}}(C,(X_{i_1},\cdots,X_{i_q}),\perp)$，然后将它的输出添加到视图中。

因为在混合模型中对于合谋参与方集合而言 $\mathrm{Sim}_{\mathrm{BZS}}(C,(X_{i_1},\cdots,X_{i_q}),\perp)$ 与 $\mathrm{VIEW}_C^{\pi_{\mathrm{BZS}}}(X_1,\cdots,X_N)$ 在计算意义上无法区分，所以合谋方 C 的视图在理想世界与现实世界是无法区分的：

$$\{\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)\}\overset{c}{\equiv}\{\mathrm{VIEW}_C^{\pi_{\mathrm{PTPSI}}}(X_1,\cdots,X_N)\} \tag{10}$$

证毕.

5 隐私增强的多方PTPSI协议

本章5.1节提出一种隐私增强的PTPSI协议，然后在5.2节给出协议具体的安全证明。

5.1 隐私增强的多方PTPSI协议

本节提出一种隐私增强的概率门限隐私集合交集(Enhanced PTPSI，EPTPSI)协议。虽然4.2节的协议是高效的，但它不允许参与方 P_{N-1} 与 P_N 合谋，因为当二者合谋时，在双中心零共享阶段，P_{N-1} 可以拿到 P_N 分发给 $P_i(i\in[1,N-2])$ 的种子，这样 P_{N-1} 就可以通过无限次查询 $P_i(i\in[1,N-2])$ 发送给他的数据结构 T_i，拿到任意 $P_i(i\in[1,N-2])$ 的隐私集合信息。同样，P_{N-1} 可以与 P_N 共享 $T_i(i\in[1,N-2])$，P_N 通过无限次查询 T_i 拿到任意 $P_i(i\in[1,N-2])$ 的隐私集合信息，这是不允许的。

通过对Wu等人[11]和Kolesnikov等人[29]的协议进行研究，本节构造出一种隐私增强的PTPSI协议。即使参与方 P_{N-1} 与 P_N 合谋，也只能得到所有参与方的交集信息，其他信息无法得到。主要思想是利用OPRF可以限制参与方无限查询的优势，分别让每个参与方 $P_i(i\in[1,N-1])$ 作为OPRF发送方，其余 $N-1$ 个参与方作为OPRF接收方实现秘密份额的分发，若 P_N 不参与合谋，由于 P_N 未向其他参与方发送任何关于他私有集合的信息，且各方在半诚实场景下诚实执行协议，协议是安全的。若 P_N 参与合谋，则必有一个参与方 $P_j(j\in[1,N-1])$ 是诚实方，合谋方由于无法拿到诚实参与方 P_j 使用的OPRF密钥 $s^{j,k}(k\in[1,N-1],k\neq j)$，所以该协议也是安全的，$\pi_{\mathrm{EPTPSI}}$ 具体协议如下。

参数：每个参与方 $P_i(i\in[1,N])$ 拥有一个输入集合 $X_i=\{x_1^i,\cdots,x_n^i\}\subseteq\{0,1\}^*$，计算安全参数 κ，统计安全参数 λ，k,τ,l 是三个整数，OKVS编码算法 $\mathrm{Encode}(\cdot)$ 和解码算法 $\mathrm{Decode}(\cdot,\cdot)$，理想功能 $\mathcal{F}_{\mathrm{OPRF}}$ 和 $\mathcal{F}_{\mathrm{TPSST}}$，伪随机函数 $F_{(\cdot)}(\cdot):\{0,1\}^\kappa\times\{0,1\}^*\to\{0,1\}^m$。协议过程如下。

份额共享阶段。

(1) 首先让每个参与方 $P_i(i\in[1,N-1])$ 作为发送方与其他参与方 $P_j(j\in[1,N],j\neq i)$ 调用 $\mathcal{F}_{\mathrm{OPRF}}$，$P_j$ 输入查询集合 $\{x_1^j,\cdots,x_n^j\}$，得到伪随机值集合 $\{F_{s^{i,j}}(x_e^j)\}_{e\in[n]}$；$P_i$ 得到与每个 $P_j(j\in[1,N],j\neq i)$ 执行 $\mathcal{F}_{\mathrm{OPRF}}$ 时的密钥 $s^{i,j}$。$P_i(i\in[1,N-1])$ 使用密钥计算集合 $\{c_j^i|\bigoplus_{e=1}^{i-1}F_{s^{i,e}}(x_j^i)\bigoplus\bigoplus_{e=i+1}^{N}F_{s^{i,e}}(x_j^i)\}_{j\in[n]}$。

(2) 每个参与方 $P_i(i\in[1,N-1])$ 计算伪随机值集合 $\{v_j^i|c_j^i\bigoplus\bigoplus_{e=1}^{i-1}F_{s^{e,i}}(x_j^i)\bigoplus\bigoplus_{e=i+1}^{N-1}F_{s^{e,i}}(x_j^i)\}_{j\in[n]}$。$P_N$ 使用作为OPRF接收方得到的伪随机值计算集合 $W_N=\{w_j^N|\bigoplus_{e=1}^{N-1}F_{s^{e,N}}(x_j^N)\}_{j\in[n]}$。

(3) 由步骤(2)，每个参与方 $P_i(i\in[1,N-2])$ 能得到一个键值对集合 $\mathcal{S}_i=\{(x_j^i,v_j^i)\}_{j\in[n]}$，由OKVS编码算法 $\mathrm{Encode}(\mathcal{S}_i)$ 得到数据结构 T_i，把 T_i 发给 P_{N-1}，P_{N-1} 使用OKVS解码算法 $\mathrm{Decode}(T_i,x)$ 计算集合 $W_{N-1}=\{w_j^{N-1}|v_j^{N-1}\bigoplus\bigoplus_{e=1}^{N-2}\mathrm{Decode}(T_e,x_j^{N-1})\}_{j\in[n]}$，令 $\mathcal{S}_{N-1}=\{(x_j^{N-1},w_j^{N-1})\}_{j\in[n]}$。

门限测试阶段。

(4) P_{N-1} 与 P_N 调用 $\mathcal{F}_{\mathrm{TPSST}}$，$P_{N-1}$ 的输入为 W_{N-1}，P_N 的输入为 W_N。

(5) 从步骤(4)中得到输出比特 b，如果 $b=0$ 终止协议。如果 $b=1$，执行下一步。

求交集阶段。

(6) P_{N-1} 通过OKVS编码算法 $\mathrm{Encode}(\mathcal{S}_{N-1})$ 得到一个数据结构 T_{N-1}，然后把 T_{N-1} 发送给 P_N。

(7) P_N 使用OKVS解码算法 $\mathrm{Decode}(T_{N-1},x)$ 求交集 $I=\{x_j^N|w_j^N\bigoplus\mathrm{Decode}(T_{N-1},x_j^N)=0\}_{j\in[n]}$。

5.2 安全证明

定理3. 在 $(\mathcal{F}_{\mathrm{OPRF}},\mathcal{F}_{\mathrm{TPSST}})$ 混合模型下，当 P_N 与 P_{N-1} 不合谋时，本文5.1节的协议 π_{EPTPSI} 实现了3.1节中针对半诚实敌手的理想功能函数 $\mathcal{F}_{\mathrm{TPSST}}$。

证明. 本文首先证明该协议的正确性，然后证明协议的安全性。

正确性：为证明正确性，本文把元素 x 分为在交集中和不在交集中两种情况考虑。

在协议步骤(1)与步骤(2)中，每个发送方 $P_i(i\in[1,N-1])$ 可以调用 $\mathcal{F}_{\mathrm{OPRF}}$ 产生的所有密钥计算元素 $x_j^i(j\in[n])$ 对应的OPRF的异或值：

$$c_j^i=\bigoplus_{e=1}^{i-1}F_{s^{i,e}}(x_j^i)\bigoplus\bigoplus_{e=i+1}^{N}F_{s^{i,e}}(x_j^i) \tag{11}$$

每个接收方 $P_j(j\in[1,N],j\neq i)$ 可以得到正确的OPRF值。当 $i\in[1,N-1]$ 时，每个参与方 P_i 可以计算集合元素 $x_j^i(j\in[n])$ 对应的值 v_j^i：

$$v_j^i = c_j^i \oplus \bigoplus_{e=1}^{i-1} F_{s^{e,i}}(x_j^i) \oplus \bigoplus_{e=i+1}^{N-1} F_{s^{e,i}}(x_j^i) \quad (12)$$

参与方P_N计算集合元素$x_j^N(j\in[n])$对应的值w_j^N：

$$w_j^N = \bigoplus_{e=1}^{N-1} F_{s^{e,N}}(x_j^N) \quad (13)$$

在步骤(3)中，每个参与方$P_i(i\in[1,N-2])$通过OKVS编码算法将x与其对应的值v正确编码到数据结构T_i中：把T_i发送给P_{N-1}。P_{N-1}通过OKVS解码算法计算元素$x_j^{N-1}(j\in[n])$对应的w_j^{N-1}：

$$w_j^{N-1} = v_j^{N-1} \oplus \bigoplus_{e=1}^{N-2} \mathrm{Decode}(T_e, x_j^{N-1}) \quad (14)$$

如果P_{N-1}与P_N调用$\mathcal{F}_{\mathrm{TPSST}}$返回值为1，代表通过门限测试。$P_{N-1}$编码$S_{N-1}=\{(x_j^{N-1}, w_j^{N-1})\}_{j\in[n]}$得到数据结构$T_{N-1}$，将$T_{N-1}$发送给$P_N$，$P_N$计算$x_j^N(j\in[n])$解码值与$w_j^N$的异或值：

$$w_j^N \oplus \mathrm{Decode}(T_{N-1}, x_j^N) \quad (15)$$

第一种情况：x为交集元素时，由(14)(15)知

$$w_j^N \oplus \mathrm{Decode}(T_{N-1}, x) = 0 \quad (16)$$

第二种情况：x为非交集元素时，此时分为以下三种子情况。

如果$x\notin X_i(i\in[1,N-2])$，由OPRF的安全性，在公式(12)中，P_i得到的v_j^i为一个随机值，由OKVS的正确性，公式(14)中P_{N-1}的解码结果也为一个随机值，则公式(15)中的解码结果对于P_N来说是一个不等于0的随机值。

如果$x\notin X_{N-1}$，由于参与方P_{N-1}没有OPRF密钥$s^{i,N-1}(i\in[1,N-2])$，则公式(12)中，P_{N-1}得到的v_j^i对于P_{N-1}而言为一个随机值，公式(14)中的w_j^{N-1}也为一个随机值，由OKVS的不经意性，公式(15)中P_N的解码结果也为一个不等于0的随机值。

如果$x\notin X_N$，由于参与方P_N没有OPRF密钥$s^{i,N}(i\in[1,N-1])$，则公式(13)中P_N得到的OPRF异或值w_j^N为一个随机值，公式(15)中的结果对于P_N来说是一个不等于0的随机值。以上三种子情况，都不会被当作交集元素输出。

正确性得证。

安全性：本文采用理想-现实模型进行形式化证明。用C表示合谋参与方集合，$|C|=q$。H表示诚实参与方的集合。接下来构造模拟器Sim模拟合谋参与方C的视图。需要说明，当集合C中只有一个参与方$P_i(i\in[1,N])$时，表示单个参与方被腐败的情况，下面对合谋情况的分析已包含单个参与方腐败的情形。接下来对合谋参与方集合C的视图进行模拟，本文分为以下三种情况进行讨论。

(1) P_N参与合谋，P_{N-1}诚实，则P_N在合谋方集合C，P_{N-1}在诚实方集合H中，此时C的模拟视图为：$\{\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),I)\}$。

(2) P_{N-1}参与合谋，P_N诚实，则P_{N-1}在合谋方集合C，P_N在诚实方集合H中，此时C的模拟视图为：$\{\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)\}$。

(3) P_{N-1}与P_N均诚实，则P_{N-1}与P_N在诚实方集合H，此时合谋方集合C的模拟视图为：$\{\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)\}$。

第一种情况：P_N被腐败，则$P_N\in C$，$|C|=q$，$P_{N-1}\in H$。Sim接收到理想世界$\mathcal{F}_{\mathrm{PTPSI}}$的输出$I$以及$C$中每个合谋参与方$P_i$的集合$X_i$。模拟器$\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),I)$运行如下：

(1) Sim调用模拟器$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(X_i,\{F_{s^{d,i}}(x_j^i)\}_{j\in[n]})$，来模拟每个$P_i(P_i\in C)$作为OPRF接收方与每个$P_d(P_d\in H)$调用$\mathcal{F}_{\mathrm{OPRF}}$时的视图，将每次调用后的结果集合$\{F_{s^{d,i}}(x_j^i)\}_{j\in[n]}$添加到合谋方的视图中。

(2) Sim调用OPRF模拟器$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(\perp, s^{i,d})$，来模拟每个$P_i(P_i\in C\backslash P_N)$作为OPRF发送方与每个$P_d(P_d\in H)$调用$\mathcal{F}_{\mathrm{OPRF}}$时的视图，将每次调用的结果$s^{i,d}$添加到合谋方集合的视图中。

(3) Sim计算P_N的OPRF伪随机值集合$W_N=\{w_j^N|\bigoplus_{e=1}^{N-1}F_{s^{e,N}}(x_j)\}_{j\in[n]}$并调用TPSST模拟器$\mathrm{Sim}_{\mathrm{TPSST}}^{P_N}(W_N,b)$，将$b$添加到视图中。

(4) Sim模拟诚实参与方P_{N-1}发送给P_N的数据结构T_{N-1}(协议步骤6)，计算键值对集合$\mathcal{S}_{N-1}=\{(x_i,\bigoplus_{e=1}^{N-1}F_{s^{e,N}}(x_i))\}_{x_i\in I}$，并随机采样$n-|I|$个键值对到$\mathcal{S}_{N-1}$中，使得$|\mathcal{S}_{N-1}|=n$，使用OKVS编码算法$\mathrm{Encode}(\mathcal{S}_{N-1})$得到数据结构$T'_{N-1}$，然后把$T'_{N-1}$添加到合谋方集合的视图中。

因为在混合模型下对于合谋参与方C来说$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(X_i,\{F_{s^{d,i}}(x_j^i)\}_{j\in[n]})$与$\mathrm{VIEW}_{P_i}^{\pi_{\mathrm{OPRF}}}(X_d,X_i)$在计算意义上无法区分，$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(\perp,s^{i,d})$与$\mathrm{VIEW}_{P_i}^{\pi_{\mathrm{OPRF}}}(X_i,X_d)$在计算意义上无法区分，$\mathrm{Sim}_{\mathrm{TPSST}}^{P_N}(W_N,b)$与$\mathrm{VIEW}_{P_N}^{\pi_{\mathrm{TPSST}}}(W_{N-1},W_N)$在计算意义上无法区分，合谋方无法得到诚实参与方$P_d\in H$的密钥$s^{d,j}(j\in[1,N-1],j\neq d)$，所以合谋方集合$C$的视图在理想世界与现实世界中是无法区分的：

$$\{\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),I)\} \overset{c}{\equiv} \{\mathrm{VIEW}_C^{\pi_{\mathrm{EPTPSI}}}(X_1,\cdots,X_N)\} \quad (17)$$

第二种情况：P_{N-1}被腐败，则$P_{N-1}\in C$，$|C|=q$，$P_N\in H$，Sim接收到每个合谋参与方P_i（$P_i\in C$）的集合X_i。模拟器$\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)$运行如下：

（1）Sim调用模拟器$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(X_i,\{F_{s^{d,i}}(x_j^i)\}_{j\in[n]})$，来模拟每个$P_i$（$P_i\in C$）作为OPRF接收方与每个$P_d$（$P_d\in H\backslash P_N$）调用$\mathcal{F}_{\mathrm{OPRF}}$时的视图，将每次调用时的结果集合$\{F_{s^{d,i}}(x_j^i)\}_{j\in[n]}$添加到合谋方的视图中。

（2）Sim调用OPRF模拟器$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(\perp,s^{i,d})$，来模拟每个$P_i$（$P_i\in C$）作为OPRF发送方与每个$P_d$（$P_d\in H$）调用$\mathcal{F}_{\mathrm{OPRF}}$时的视图，将每次调用的结果$s^{i,d}$添加到合谋方集合的视图中。

（3）Sim模拟每个诚实参与方P_d（$P_d\in H\backslash P_N$）发送给腐败参与方P_{N-1}的数据结构T_d（协议步骤3），随机采样n个键值对得到一个键值集合$\mathcal{S}_d'=\{(x'^d_j,v'^d_j)\}_{j\in[n]}$，Sim编码该键值集合得到一个数据结构$T_d'$，由OPRF的安全性知，$P_{N-1}$对于模拟器采样随机值生成的$T_d'$与真实协议执行过程中生成的$T_d$（因为在真实协议执行过程中生成的$T_d$对于$P_{N-1}$来说也是随机值）是无法区分的，二者是计算不可区分的。将T_d'添加到合谋方集合的视图中。

（4）Sim计算P_{N-1}的OPRF伪随机值集合W_{N-1}并调用TPSST模拟器$\mathrm{Sim}_{\mathrm{TPSST}}^{P_{N-1}}(W_{N-1},b)$，将$b$添加到视图中。

因为在混合模型中对于合谋方C来说$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(X_i,\{F_{s^{d,i}}(x_j^i)\}_{j\in[n]})$与$\mathrm{VIEW}_{P_i}^{\pi_{\mathrm{OPRF}}}(X_d,X_i)$在计算意义上无法区分，$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(\perp,s^{i,d})$与$\mathrm{VIEW}_{P_i}^{\pi_{\mathrm{OPRF}}}(X_i,X_d)$在计算意义上无法区分，$\mathrm{Sim}_{\mathrm{TPSST}}^{P_{N-1}}(W_{N-1},b)$与$\mathrm{VIEW}_{P_{N-1}}^{\pi_{\mathrm{TPSST}}}(W_{N-1},W_N)$在计算意义上无法区分，所以合谋方$C$的视图在理想世界与现实世界是无法区分的：

$$\{\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)\}\overset{c}{\equiv}\{\mathrm{VIEW}_{C}^{\pi_{\mathrm{EPTPSI}}}(X_1,\cdots,X_N)\} \tag{18}$$

第三种情况：P_{N-1}与P_N诚实，则$P_{N-1},P_N\in H$，$|C|=q$，Sim接收到每个合谋参与方P_i（$P_i\in C$）的集合X_i。模拟器$\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)$运行如下：

（1）Sim调用模拟器$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(X_i,\{F_{s^{d,i}}(x_j^i)\}_{j\in[n]})$，来模拟每个$P_i$（$P_i\in C$）作为OPRF接收方与每个$P_d$（$P_d\in H\backslash P_N$）调用$\mathcal{F}_{\mathrm{OPRF}}$时的视图，将每次调用时的结果集合$\{F_{s^{d,i}}(x_j^i)\}_{j\in[n]}$添加到合谋方的视图中。

（2）Sim调用OPRF模拟器$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(\perp,s^{i,d})$，来模拟每个$P_i$（$P_i\in C$）作为OPRF发送方与$P_d$（$P_d\in H$）调用OPRF模拟器$\mathcal{F}_{\mathrm{OPRF}}$时的视图，将每次调用的结果$s^{i,d}$添加到合谋方集合的视图中。

因为在混合模型下对于合谋参与方C来说$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(X_i,\{F_{s^{d,i}}(x_j^i)\}_{j\in[n]})$与$\mathrm{VIEW}_{P_i}^{\pi_{\mathrm{OPRF}}}(X_d,X_i)$在计算意义上无法区分，$\mathrm{Sim}_{\mathrm{OPRF}}^{P_i}(\perp,s^{i,d})$与$\mathrm{VIEW}_{P_i}^{\pi_{\mathrm{OPRF}}}(X_i,X_d)$在计算意义上无法区分，所以合谋参与方$C$的视图在理想世界与现实世界中是无法区分的：

$$\{\mathrm{Sim}(C,(X_{i_1},\cdots,X_{i_q}),\perp)\}\overset{c}{\equiv}\{\mathrm{VIEW}_{C}^{\pi_{\mathrm{EPTPSI}}}(X_1,\cdots,X_N)\} \tag{19}$$

证毕.

5.3 小 结

协议π_{PTPSI}与π_{EPTPSI}也可以应用在非平衡场景，为隐藏参与方的集合数量，可以让小集合参与者补充随机元素与大集合数量一致，然后与其他参与方执行协议。PTPSI协议只有在通过门限测试之后，才允许各参与方得到交集，不会泄露用户拥有的数据集大小。两个协议中的门限值是公开的，每个参与方都知道门限值大小。协议允许参与方在交集数量为$[\alpha,\beta]$范围时以$F(|I|)$的概率计算交集，F为一个递增函数，该函数的输出范围为$[0,1]$，$|I|$为参与方交集数量。当$|I|\geqslant\beta$时，计算交集的概率为$F(|I|)\geqslant 1-\delta$，当$|I|\leqslant\alpha$时，计算交集的概率$F(|I|)\leqslant\delta$，δ为一个可忽略函数。在门限测试阶段，协议使用SPDZ安全电路框架进行门限值的比较，不会泄露具体的交集大小，门限值t不需要进行模糊化处理。

PTPSI的理想功能函数允许交集数量位于$[\alpha,\beta]$范围时有一定的概率输出交集，协议π_{PTPSI}与π_{EPTPSI}均实现了这一理想功能函数且可以大幅度降低门限测试阶段的开销。在许多应用场景中，并不需要非常精确的门限值比较，比如在大数据集合的垂直联邦学习中，当参与方的交集数量$|I|=t-1$时仍有助于各参与方训练模型，本文设计的PTPSI协议更适合这种场景。

6 实验与分析

本章6.1节分析4.2节与5.1节协议的计算复杂度；6.2节分析4.2节与5.1节协议的通信复杂度；6.3节给出协议具体的实验分析以及与文献[26]的比较结果。

本文使用C++来实现所有协议，并在ubuntu20.04系统中进行仿真实验，其中设备的

CPU 为 Intel（R）Xeon（R）CPU E5-2630 v4 @ 2.20 GHz，内存为 128 G。协议的计算安全参数 $\kappa=128$，统计安全参数 $\lambda=40$。两两独立的哈希函数输入位长为 $m=\lambda+2\log n$。

6.1　计算复杂度

本节首先对4.2节中协议 π_{PTPSI} 的计算复杂度进行分析。根据文献[36]知在双中心零共享阶段每个 $P_i(i\in[1,N-2])$ 的计算复杂度均为 $O(n\lambda)$，P_{N-1} 与 P_N 的计算复杂度均为 $O(Nn\lambda)$。P_{N-1} 与 P_N 调用理想功能函数 $\mathcal{F}_{\text{TPSST}}$，需要 k 轮测试，计算复杂度为 $O((k\log n+n+k)\lambda)$，P_{N-1} 与 P_N 调用 $\mathcal{F}_{\text{OPRF}}$，计算复杂度为 $O(n\lambda)$。

接着对5.1节协议 π_{EPTPSI} 的计算复杂度进行分析。每个参与方 $P_i(i\in[1,N-2])$ 需要作为接收方与 $P_j(j\in[1,N-1],j\neq i)$ 调用 $\mathcal{F}_{\text{OPRF}}$，计算复杂度为 $O(Nn\lambda)$，接着进行 OKVS 编码，计算复杂度为 $O(n\lambda)$。P_{N-1} 首先作为发送方与其他参与方调用 $\mathcal{F}_{\text{OPRF}}$，计算复杂度为 $O(Nn\lambda)$，接着查询 $P_i(i\in[1,N-2])$ 发送过来的 OKVS 数据结构 T_i，计算复杂度为 $O((N-2)n\lambda)$，然后与 P_N 调用理想功能函数 $\mathcal{F}_{\text{TPSST}}$，需要 k 轮测试，计算复杂度为 $O((k\log n+n+k)\lambda)$。P_N 首先作为接收方与 $P_j(j\in[1,N-1])$ 调用 $\mathcal{F}_{\text{OPRF}}$，计算复杂度为 $O((N-1)n\lambda)$，接着与 P_{N-1} 调用理想功能函数 $\mathcal{F}_{\text{TPSST}}$，计算复杂度为 $O((k\log n+n+k)\lambda)$。

6.2　通信复杂度

本节首先分析4.2节中协议 π_{PTPSI} 的通信复杂度。根据文献[36]可知在双中心零共享阶段 $P_i(i\in[1,N-2])$ 的通信复杂度均为 $O(n\lambda)$，P_{N-1} 未发送信息，P_N 的通信复杂度为 $O((N-2)\kappa+n\lambda)$，根据文献[26]知 P_{N-1} 与 P_N 调用 $\mathcal{F}_{\text{TPSST}}$，通信复杂度为 $O(k\lambda+\kappa)$。如果 $\mathcal{F}_{\text{TPSST}}$ 结果输出为1，P_{N-1} 需要作为发送方与 P_N 执行 OPRF 协议通信复杂度为 $O(n\lambda)$。P_N 首先与 P_{N-1} 调用理想功能函数 $\mathcal{F}_{\text{TPSST}}$，通信复杂度为 $O(k\lambda+\kappa)$，如果 $\mathcal{F}_{\text{TPSST}}$ 输出为1，P_N 作接收方与 P_{N-1} 调用 $\mathcal{F}_{\text{OPRF}}$，通信复杂度为 $O(n\lambda)$。

接着对5.1节协议 π_{EPTPSI} 的通信复杂度进行分析。每个参与方 $P_i(i\in[1,N-2])$ 需要作为接收方与其他参与方 $P_j(j\in[1,N-1],j\neq i)$ 调用 $\mathcal{F}_{\text{OPRF}}$，通信复杂度为 $O(Nn\lambda)$，接着发送 OKVS 编码得到的数据结构给 P_{N-1}，通信复杂度为 $O(n\lambda)$。P_{N-1} 首先作为接收方与其他参与方 $P_j(j\in[1,N-2])$ 调用 $\mathcal{F}_{\text{OPRF}}$，通信复杂度为 $O(Nn\lambda)$。再与 P_N 调用 $\mathcal{F}_{\text{TPSST}}$，通信复杂度为 $O(k\lambda+\kappa)$。如果 $\mathcal{F}_{\text{TPSST}}$ 结果输出为1，P_{N-1} 作为发送方与 P_N 调用 $\mathcal{F}_{\text{OPRF}}$，通信复杂度为 $O(n\lambda)$。P_N 首先作为 OPRF 接收方与 $P_j(j\in[N-1])$ 调用 $\mathcal{F}_{\text{OPRF}}$，通信复杂度为 $O((N-1)n\lambda)$，接着与 P_{N-1} 调用 $\mathcal{F}_{\text{TPSST}}$，通信复杂度为 $O(k\lambda+\kappa)$。如果 $\mathcal{F}_{\text{TPSST}}$ 输出为1，P_N 作为接收方与 P_{N-1} 调用 $\mathcal{F}_{\text{OPRF}}$，通信复杂度为 $O(n\lambda)$。

文献[26]提出的多方概率集合大小测试协议与本文两个协议的门限测试阶段的通信与计算开销对比如表2所示。

6.3　实验结果

在测试实验数据之前，本文设置 $t=0.5n$，$\beta=t$，$\alpha=\frac{1}{2}t$，其中 t 为门限值。两两独立哈希函数的输入位长 $m=\lambda+2\log n$。δ,p,k 这三个数据之间的关系如表3所示，表3列出在给定参数 δ 与 p 时，执行GS协议需要并行重复的轮数 k。

表2　概率集合大小测试开销对比

协议	$P_1,\cdots,P_{N-2}$		P_{N-1}		P_N	
	通信开销	计算开销	通信开销	计算开销	通信开销	计算开销
文献[26]	$O(Nk\lambda+\kappa)$	$O((k\log n+n+k)\lambda)$	$O(Nk\lambda+\kappa)$	$O((k\log n+n+k)\lambda)$	$O(Nk\lambda+\kappa)$	$O((k\log n+n+k)\lambda)$
π_{PTPSI}	$O(n\lambda)$	$O(n\lambda)$	$O(k\lambda+\kappa)$	$O(Nn\lambda+k\log n\lambda)$	$O(N\kappa+n\lambda)$	$O(Nn\lambda+k\log n\lambda)$
π_{EPTPSI}	$O(Nn\lambda)$	$O(Nn\lambda)$	$O(Nn\lambda+k\lambda+\kappa)$	$O(Nn\lambda+k\log n\lambda)$	$O(Nn\lambda+k\lambda+\kappa)$	$O(Nn\lambda+k\log n\lambda)$

表3结果显示，当 $p=\frac{1}{2}$ 时，如果 δ 大小为统计安全参数 2^{-40}，此时需要并行重复的轮数值至少为5324轮。在后续实验测试中，为确保误差可忽略，本文设置的默认轮数为 $k=8000$。

本文对文献[26]的半诚实多方交集计数功能函数 $\mathcal{F}_{\cap-\text{count}}$ 在LAN下进行测试，固定各方数据集大小 $n=2^{20}$，协议在不同参与方数量与不同重复轮数的运行时间如表4所示。

表4结果显示，该组件依赖昂贵的通用电路机

表3 k值大小

p	δ:$F(\beta)\geqslant 1-\delta$			δ:$F(\alpha)\leqslant\delta$		
	2^{-20}	2^{-30}	2^{-40}	2^{-20}	2^{-30}	2^{-40}
$\frac{1}{4}$	5324	7985	10647	3993	5989	7985
$\frac{1}{2}$	2662	3993	5324	1997	2995	3993

制，随着参与方数量增多，计算开销与参与方数量呈指数关系。但两方场景下协议效率较高。因实验条件限制，多方场景下，SPDZ阶段计算开销较大，本文实验设备只能测试五个参与方的实验结果。

表4 $\mathcal{F}_{\cap-\text{count}}$不同参与方数量的运行时间（单位：s）

参与方	2方	3方	4方	5方
5000	1.13	1.82	11.81	19.97
6000	1.34	2.24	13.39	21.69
7000	1.53	2.98	14.59	27.74
8000	1.95	5.25	16.13	37.73

表5给出两方场景下4.1节协议π_{TPSST}在不同数据集合大小和不同并行重复轮数下的总运行时间。

表5 π_{TPSST}在不同集合大小下的运行时间（单位：s）

集合大小	2^{12}	2^{16}	2^{18}	2^{20}
5000	2.12	2.27	2.32	6.48
6000	2.47	2.69	3.02	6.69
7000	2.64	3.14	3.42	6.88
8000	3.12	3.19	4.03	7.30

针对4.2节协议π_{PTPSI}与5.1节协议π_{EPTPSI}，设置参与方集合大小$n=2^{20}$，在不同的轮数k与不同参与方数量下两种协议的运行时间对比如表6所示。固定轮数$k=8000$，不同参与方数量和不同集合大小下两种协议的运行时间对比如表7所示。实验环境均在LAN下。

表6的实验结果显示，对于同一种协议，在参与方数量固定的情况下，随着重复轮数的增加，协议运行时间变化较小。表7的实验结果显示，当集合大小固定时，随着参与方数量的增加，协议π_{PTPSI}展现了较好的可扩展性。

本文针对计算时间与通信开销两个维度，将所设计协议与文献[26]提出的多方概率门限隐私集合交集协议进行了系统性对比。为

表6 不同重复次数k下的运行时间对比（单位：s）

重复次数		5000	6000	7000	8000
5方	π_{PTPSI}	8.80	9.01	9.20	9.62
	π_{EPTPSI}	29.43	29.64	29.83	30.25
10方	π_{PTPSI}	8.86	9.07	9.26	9.68
	π_{EPTPSI}	37.26	37.47	37.66	38.08
16方	π_{PTPSI}	8.88	9.09	9.28	9.70
	π_{EPTPSI}	59.25	59.46	59.65	60.07
32方	π_{PTPSI}	9.10	9.31	9.50	9.92
	π_{EPTPSI}	166.27	166.48	166.67	167.09

表7 不同集合大小下的运行时间对比（单位：s）

集合大小		2^{12}	2^{16}	2^{18}	2^{20}
5方	π_{PTPSI}	3.20	3.36	4.55	9.62
	π_{EPTPSI}	3.26	4.24	8.81	30.25
10方	π_{PTPSI}	3.21	3.36	4.57	9.68
	π_{EPTPSI}	3.41	4.97	11.49	38.08
16方	π_{PTPSI}	3.21	3.37	4.59	9.70
	π_{EPTPSI}	3.66	6.43	16.16	60.07
32方	π_{PTPSI}	3.22	3.42	4.66	9.92
	π_{EPTPSI}	9.76	19.23	45.35	167.09

保证比较的公平性，在文献[26]协议的概率集合大小测试阶段通过后，使用目前最快的多方隐私集合交集协议[36]作为其PSI阶段的开销。

固定$\mathcal{F}_{\cap-\text{count}}$中设置的并行重复轮数$k=8000$，本文测试了两个协议分别在不同的网络环境（LAN，200 Mbps，100 Mbps），与不同参与方数量（3,4,5,10,16,32），及不同集合大小（$n=2^{16},2^{18},2^{20}$）下的运行时间与通信开销，并在相同的实验环境下复现了文献[26]的协议，对比实验如表8所示。

表8实验结果显示，本文协议可以很好地扩展到多方场景，其中协议π_{PTPSI}在32方场景下，其运行时间为9.92秒，通信开销为773.90 MB。在多方场景下随着参与方数量从5方扩展到32方，其运行时间变化为0.3秒，通信开销增加586.51 MB，其可扩展性显著增强，适合运用在多方场景下。协议π_{EPTPSI}相比于π_{PTPSI}实现了更强的隐私性，从实验数据可以看出，本文方案的通信开销及计算开销均优于文献[26]。

表8　与文献[26]的对比

开销		计算开销(秒)									通信开销(MB)		
协议		文献[26]			π_{PTPSI}			π_{EPTPSI}			文献[26]	π_{PTPSI}	π_{EPTPSI}
		LAN	200M	100M	LAN	200M	100M	LAN	200M	100M			
2^{16}	3方	4.86	12.20	24.02	**3.34**	**3.61**	**6.68**	4.02	4.30	7.46	246.56	**64.96**	67.86
	4方	15.73	31.74	62.56	**3.35**	**3.63**	**6.71**	4.08	4.47	7.57	703.74	**66.30**	77.14
	5方	29.59	84.14	170.28	**3.36**	**3.64**	**6.73**	4.24	5.03	8.66	2443.25	**67.65**	89.91
	10方	*	*	*	**3.36**	**3.65**	**6.74**	4.97	9.58	18.63	*	**74.36**	206.04
	16方	*	*	*	**3.37**	**3.66**	**7.44**	6.43	20.47	40.61	*	**82.42**	460.43
	32方	*	*	*	**3.42**	**4.45**	**9.26**	19.23	189.04	1060.19	*	**103.91**	1752.32
2^{18}	3方	6.96	18.63	35.96	**4.52**	**5.01**	**8.48**	8.14	8.79	12.16	377.09	**84.66**	93.82
	4方	18.23	35.09	68.66	**4.53**	**5.03**	**8.94**	8.63	9.31	14.74	781.23	**91.08**	125.48
	5方	36.24	100.32	214.45	**4.55**	**5.04**	**9.42**	8.81	11.15	18.38	2909.69	**95.49**	168.74
	10方	*	*	*	**4.57**	**6.01**	**11.71**	11.49	28.09	51.38	*	**122.58**	559.21
	16方	*	*	*	**4.59**	**7.39**	**14.47**	16.16	65.03	124.69	*	**155.07**	1410.03
	32方	*	*	*	**4.66**	**11.17**	**21.79**	45.35	526.29	1916.27	*	**241.73**	5720.49
2^{20}	3方	12.80	29.41	55.54	**9.5**	**10.43**	**16.98**	27.85	28.8	35.45	552.53	143.94	**125.19**
	4方	23.70	51.49	97.33	**9.52**	**11.28**	**18.87**	28.82	33.77	45.27	1137.96	**165.67**	303.55
	5方	45.40	145.60	286.28	**9.62**	**12.27**	**20.73**	30.25	41.15	59.26	3403.12	**187.39**	471.97
	10方	*	*	*	**9.68**	**16.95**	**29.80**	38.08	105.89	188.32	*	**296.00**	1988.79
	16方	*	*	*	**9.70**	**22.46**	**41.01**	60.07	249.31	466.31	*	**426.34**	5293.24
	32方	*	*	*	**9.92**	**37.43**	**70.61**	167.09	956.99	2717.01	*	**773.90**	22021.22

注：*表示开销太大，无法计算

7　结束语

为了将现有的PTPSI协议有效扩展至多方场景，本文在半诚实模型下提出并实现了两个PTPSI协议。与以往的协议相比，第一种协议π_{PTPSI}避免了大量的公钥算法，该协议随着参与方数量的增多，运行效率不受影响。为了增强隐私性，设计了一种隐私增强的概率门限隐私集合交集协议π_{EPTPSI}，两个协议都能抵抗特定$N-1$个参与方的合谋攻击。算法分析与实验结果显示，本文所设计的协议与现有的方案相比具有更好的性能。

协议的安全性建立在半诚实模型下，在半诚实模型下，参与方严格遵守协议的执行规范，但会在协议执行过程中试图通过记录和分析其获得的所有中间信息来推测其他参与方的隐私信息。在恶意模型下敌手可以通过构造特殊集合来进行试探攻击获取诚实方的隐私信息，未来工作中，将会对恶意模型下的多方概率门限隐私集合交集协议进行深入研究。

参　考　文　献

[1] Freedman M J, Nissim K, Pinkas B. Efficient private matching and set intersection//Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Interlaken, Switzerland, 2004: 1-19

[2] Chase M, Miao P. Private set intersection in the internet setting from lightweight oblivious PRF//Proceedings of the CRYPTO 2020 - 40th Annual International Cryptology Conference. California, USA, 2020: 34-63

[3] Rindal P, Rosulek M. Improved private set intersection against malicious adversaries//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Paris, France, 2017: 235-259

[4] Rindal P, Rosulek M. Malicious-secure private set intersection via dual execution//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Texas, USA, 2017: 1229-1242

[5] Wei Li-Fei, Wang Qin, Zhang Lei, Chen Cong-Cong, Chen Yu-Jiao, Ning Jian-Ting. Efficient private set intersection protocols with semi-trusted cloud server aided. Journal of Software, 2023, 34(02): 932 - 944 (in Chinese)
(魏立斐，王勤，张蕾，陈聪聪，陈玉娇，宁建廷. 半可信云服务器辅助的高效隐私交集计算协议. 软件学报，2023，34(02)932-

944)

[6] Gong Lin-Ming, Wang Dao-Shun, Liu Mo-Meng, Gao Quan-Li, Shao Lian-He, Wang Ming-Ming. PSI computation based on no matching errors. Chinese Journal of Computers, 2020, 43(09): 1769-1790 (in Chinese)
(巩林明，王道顺，刘沫萌，高全力，邵连合，王明明. 基于无匹配差错的PSI计算. 计算机学报，2020，43(09)：1769-1790)

[7] Yang Jia-Hui, Chen Lan-Xiang, Mu Yi, Zeng Ling-Fang, Xue Yu-Jie. PSI protocol with structured encryption. Chinese Journal of Computers, 2022, 45(12): 2652-2666 (in Chinese)
(杨佳辉，陈兰香，穆怡，曾令仿，薛玉洁. 结构化加密的PSI协议. 计算机学报，2022，45(12)：2652-2666)

[8] Zhang E, Liu F H, Lai Q, et al. Efficient multi-party private set intersection against malicious adversaries//Proceedings of the 2019 ACM SIGSAC conference on cloud computing security workshop. London, UK, 2019: 93-104

[9] Chandran N, Dasgupta N, Gupta D, et al. Efficient linear multiparty PSI and extensions to circuit/quorum PSI//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual, 2021: 1182-1204

[10] Nevo O, Trieu N, Yanai A. Simple, fast malicious multiparty private set intersection//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual, 2021: 1151-1165

[11] Wu M, Yuen T H, Chan K Y. O-Ring and K-Star: Efficient multi-party private set intersection//Proceedings of the 33th Conference on USENIX Security Symposium. Pennsylvania, USA, 2024: 6489-6506

[12] Zhang Lei, He Chong-De, Wei Li-Fei. Efficient and malicious secure three-party private set intersection computation protocols for small sets. Journal of Computer Research and Development, 2022, 59(10): 2286-2298 (in Chinese)
(张蕾，贺崇德，魏立斐. 高效且恶意安全的三方小集合隐私交集计算协议. 计算机研究与发展，2022，59(10)：2286-2298)

[13] Wei Li-Fei, Liu Ji-Hai, Zhang Lei, Wang Qin, He Chong-De. Survey of privacy preserving oriented set intersection computation. Journal of Computer Research and Development, 2022, 59(08): 1782-1799 (in Chinese)
(魏立斐，刘纪海，张蕾，王勤，贺崇德. 面向隐私保护的集合交集计算综述. 计算机研究与发展，2022，59(08)：1782-1799)

[14] Nagaraja S, Mittal P, Hong C Y, et al. BotGrep: Finding P2P bots with structured graph analysis//Proceedings of the 19th Conference on USENIX Security Symposium. Washington, USA, 2010: 95-110

[15] Pinkas B, Schneider T, Segev G, et al. Phasing: Private set intersection using permutation-based hashing//Proceedings of the 24th Conference on USENIX Security Symposium. Washington, USA, 2015: 515-530

[16] Duong T, Phan D H, Trieu N. Catalic: Delegated PSI cardinality with applications to contact tracing//Proceedings of the International Conference on the Theory and Application of Cryptology and Information Security. Daejeon, Republic of Korea, 2020: 870-899

[17] Hallgren P, Orlandi C, Sabelfeld A. PrivatePool: Privacy-preserving ridesharing//Proceedings of the IEEE 30th Computer Security Foundations Symposium. California, USA, 2017: 276-291

[18] Mohassel P, Zhang Y. SecureML: A system for scalable privacy-preserving machine learning//Proceedings of the 2017 IEEE Symposium on Security and Privacy. California, USA, 2017:19-38

[19] Zhao Y, Chow S S M. Are you the one to share? Secret transfer with access structure. Proceedings on Privacy Enhancing Technologies, 2017, 2017(1): 149-169

[20] Ghosh S, Nilges T. An algebraic approach to maliciously secure private set intersection//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Darmstadt, Germany, 2019: 154-185

[21] Hu J, Zhao Y, Tan B H M, et al. Enabling threshold functionality for private set intersection protocols in cloud computing. IEEE Transactions on Information Forensics and Security, 2024, 19: 6184-6196

[22] Branco P, Döttling N, Pu S. Multiparty cardinality testing for threshold private intersection//Proceedings of the IACR International Conference on Public-Key Cryptography. Virtual, 2021: 32-60

[23] Badrinarayanan S, Miao P, Raghuraman S, et al. Multi-party threshold private set intersection with sublinear communication//Proceedings of the IACR International Conference on Public-Key Cryptography. Virtual, 2021: 349-379

[24] Zhang En, Qin Lei-Yong, Yang Ren-Lin, Li Gong-Li. Multi-party threshold private set intersection protocol based on robust secret sharing. Journal of Software, 2023, 34(11): 5424-5441 (in Chinese)
张恩，秦磊勇，杨刃林，李功丽. 基于弹性秘密共享的多方门限隐私集合交集协议. 软件学报，2023，34(11)：5424-5441)

[25] Ghosh S, Simkin M. Threshold private set intersection with better communication complexity//Proceedings of the IACR International Conference on Public-Key Cryptography. Georgia, USA, 2023: 251-272

[26] Liu F H, Zhang E, Qin L. Efficient multiparty probabilistic threshold private set intersection//Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. Copenhagen, Denmark, 2023: 2188-2201

[27] Goldwasser S, Sipser M. Private coins versus public coins in interactive proof systems//Proceedings of the eighteenth annual ACM symposium on Theory of computing. California, USA, 1986: 59-68

[28] Chernoff H. A note on an inequality involving the normal distribution. The Annals of Probability, 1981, 9(3): 533-535

[29] Kolesnikov V, Kumaresan R, Rosulek M, et al. Efficient batched oblivious PRF with applications to private set intersection//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria, 2016: 818-829

[30] Kolesnikov V, Matania N, Pinkas B, et al. Practical multi-party private set intersection from symmetric-key techniques//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Texas, USA, 2017: 1257-1272

[31] Rindal P, Schoppmann P. VOLE-PSI: Fast OPRF and circuit-PSI from vector-OLE//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Zagreb, Croatia, 2021: 901-930

[32] Raghuraman S, Rindal P. Blazing fast PSI from improved OKVS and subfield VOLE//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. California, USA, 2022: 2505-2517

[33] Pinkas B, Rosulek M, Trieu N, et al. PSI from PaXoS: Fast, malicious private set intersection//Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques. Zagreb, Croatia, 2020: 739-767

[34] Garimella G, Pinkas B, Rosulek M, et al. Oblivious key-value stores and amplification for private set intersection//Proceedings of the CRYPTO 2021-41th Annual International Cryptology Conference. Virtual, 2021: 395-425

[35] Bienstock A, Patel S, Seo J Y, et al. Near-optimal oblivious key-value stores for efficient PSI, PSU and volume-hiding multi-maps//Proceedings of the 32th Conference on USENIX Security Symposium. California, USA, 2023:301-318

[36] Gao Y, Luo Y, Wang L, et al. Efficient scalable multi-party private set intersection (-variants) from bicentric zero-sharing//Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security. Utah, USA, 2024: 4137-4151

[37] Goldreich O. The foundations of cryptography—Volume 2: Basic application. Cambridge: Cambridge university press, 2004

[38] Canetti R. Universally composable security: A new paradigm for cryptographic protocols//Proceedings of the 42nd IEEE Symposium on Foundations of Computer Science. Nevada, USA, 2001: 136-145

[39] Damgård I, Pastro V, Smart N, et al. Multiparty computation from somewhat homomorphic encryption//Proceedings of the CRYPTO 2012-32th Annual International Cryptology Conference, California, USA, 2012: 643-662

[40] Damgård I, Keller M, Larraia E, et al. Practical covertly secure MPC for dishonest majority-or: breaking the SPDZ limits//Proceedings of the European Symposium on Research in Computer Security, Egham, UK, 2013: 1-18

[41] Keller M. MP-SPDZ: A versatile framework for multi-party computation//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. Virtual, 2020: 1575-1590

附录A

引理1. 存在一个两两独立的哈希函数簇$\mathcal{H}_{m,l}$，大小为2^{2m}，能够使用$2m$比特的随机串来构造函数$h\in\mathcal{H}_{m,l}$。

证明. 令乘法计算在域$\mathbb{F}_{2^m}$内，设$x\neq x'\in\{0,1\}^m$和$y,y'\in\{0,1\}^l$，令哈希函数簇形式满足以下条件：

$$\mathcal{H}=\left\{h_{a,b}(x)=a\cdot x+b \bmod 2^l: a,b\in\mathbb{F}_{2^m}\right\} \tag{20}$$

那么

$$\begin{aligned}&\Pr_{h\leftarrow\mathcal{H}_{m,l}}\left[h(x)=y\wedge h(x')=y'\right]=\\&\Pr_{h\leftarrow\mathcal{H}_{m,l}}\left[\begin{pmatrix}ax+b=y \bmod 2^l\end{pmatrix}\right]=\\&\Pr_{h\leftarrow\mathcal{H}_{m,l}}\left[\begin{pmatrix}a=(y-y')(x-x')^{-1} \bmod 2^l\end{pmatrix}\right]=\\&2^{-2l}\end{aligned}\tag{21}$$

证毕.

附录B

引理2. 如果$|D|\geqslant t$，那么验证者接受的概率至少为$\frac{3p}{4}$，如果$|D|\leqslant\frac{t}{2}$，那么验证者接受的概率至多为$\frac{p}{2}$。

证明. 令集合$D\subset\{0,1\}^m$且$|D|\leqslant 2^{l-1}$。随机采样一个两两独立的哈希函数$h\leftarrow\mathcal{H}_{m,l}$和点$y\leftarrow\{0,1\}^l$，那么有

$$\frac{3}{4}\frac{|D|}{2^l}\leqslant\Pr_{h,y}\left[\exists x\in D: h(x)=y\right]\leqslant\frac{|D|}{2^l} \tag{22}$$

因为$p=t\cdot 2^{-l}$，当$|D|\leqslant\frac{t}{2}$时，明显

$$\Pr_{h,y}\left[\exists x\in D: h(x)=y\right]\leqslant\frac{|D|}{2^l}=\frac{p}{2} \tag{23}$$

当$|D|\geqslant t$时：

$$\begin{aligned}&\Pr_{h,y}\left[\exists x\in D: h(x)=y\right]\geqslant\sum_{x\in D}\Pr_h\left[h(x)=y\right]-\\&\sum_{x\neq x'\in D}\Pr_h\left[h(x)=y\wedge h(x')=y\right]=\\&|D|\frac{1}{2^l}-\frac{|D||D-1|}{2}\cdot\frac{1}{2^{2l}}\geqslant\\&\frac{|D|}{2^l}-\frac{1}{2}\left(\frac{|D|}{2^l}\right)^2\geqslant\\&\frac{|D|}{2^l}-\frac{1}{4}\frac{|D|}{2^l}=\frac{3}{4}p\end{aligned}\tag{24}$$

证毕.

附录C

定理1. 设置$2^{l-2}\leqslant t\leqslant 2^{l-1}$，令$\alpha=\frac{t}{2}$，$\beta=t$和$\tau=\frac{5p}{8}k$，这里$p=t\cdot 2^{-l}\in\left[\frac{1}{4},\frac{1}{2}\right]$，当函数的输入为

β时，输出满足$F_{(k,\tau,\ell)-\mathrm{GS}}(\beta)\geqslant 1-e^{-\frac{pk}{96}}$，当函数的输入为$\alpha$时，满足$F_{(k,\tau,\ell)-\mathrm{GS}}(\alpha)\leqslant e^{-\frac{pk}{72}}$。

证明. 使用切尔诺夫界来证明定理1，用M_i代表验证者在第i轮$(k,\tau,l)-$GS实验中确定证明者是否通过测试，若通过测试，令$M_i=1$；否则令$M_i=0$。设$M=\sum_{i\in[k]}M_i$，代表验证者在k次GS协议中，总共通过测试的次数，考虑以下两种情况。

假定$|D|=\alpha$，那么M的期望值为$\mu=\frac{p}{2}k$，对于任何$\sigma>0$，根据切尔诺夫界的上界计算可以得到：

$$\Pr\left[M\geqslant(1+\sigma)\mu\right]\leqslant e^{-\frac{\sigma^2}{2+\sigma}\mu} \tag{25}$$

当设置$\sigma=\frac{1}{4}$时：

$$\Pr\left[M\geqslant\frac{5p}{8}\cdot k\right]\leqslant e^{-\frac{pk}{72}} \tag{26}$$

假定$|D|=\beta$，那么M的期望值为$\mu=\frac{3p}{4}k$，对于任何$\sigma\in(0,1)$，根据切尔诺夫界的下界计算可以得到：

$$\Pr\left[M\leqslant(1-\sigma)\mu\right]\leqslant e^{-\frac{\sigma^2}{2}\mu} \tag{27}$$

当设置$\sigma=\frac{1}{6}$时：

$$\Pr\left[M\geqslant\frac{5p}{8}\cdot k\right]=1-\Pr\left[M\leqslant\frac{5p}{8}\cdot k\right]\geqslant 1-e^{-\frac{pk}{96}} \tag{28}$$

证毕.

附录D

引理3. 对于第$i(i\in[k])$轮GS测试，$|X_1^i|$与$|X_2^i|$的期望值为$n\cdot 2^{-l}$，方差为$n\cdot(2^{-l}-2^{-2l})$，概率空间由两两独立的哈希函数来确定。

证明. 对任意的集合$D=\{x_1,\cdots,x_n\}\subseteq\{0,1\}^m$，一个随机的哈希函数$h\leftarrow\mathcal{H}_{m,l}$和一个点$y\leftarrow\{0,1\}^l$，令集合$X=\{x_i\in D|h(x)=y\}$。那么$|X|$的期望值为$n\cdot 2^{-l}$，方差为$n\cdot(2^{-l}-2^{-2l})$。

对于每个x_i，用M_i表示一个二元随机变量，当且仅当$h(x_i)=y$时，$M_i=1$。那么M_i的期望值$E(M_i)=2^{-l}$，方差值为$V(M_i)=2^{-l}-2^{-2l}$。因为哈希函数h两两之间相互独立，随机变量M_i的关系也如此，所以这代表方差是线性的，那么$|X|$期望与方差大小分别为$n\cdot 2^{-l}$与$n\cdot(2^{-l}-2^{-2l})$。可以使用切比雪夫不等式对集合$|X_1^i|$与$|X_2^i|$的范围进行推导，以此证明$|X_1^i|$与$|X_2^i|$的大小不可能超出期望值太多。

首先给出切比雪夫不等式的数学形式，令μ表示随机变量X的均值，σ^2表示方差，那么对于任意的正数ε，有

$$\Pr\left[\left||X|-\mu\right|\geqslant\varepsilon\right]\leqslant\frac{\sigma^2}{\varepsilon^2} \tag{29}$$

等价的，因为$|X|$的期望值为$n\cdot 2^{-l}$，方差为$n\cdot(2^{-l}-2^{-2l})$，那么

$$\Pr\left[\left||X|-n\cdot 2^{-l}\right|\geqslant\varepsilon\right]\leqslant\frac{n\cdot(2^{-l}-2^{-2l})}{\varepsilon^2}<\frac{n\cdot 2^{-l}}{\varepsilon^2} \tag{30}$$

这意味着使用SPDZ实现$\mathcal{F}_{\cap-\mathrm{count}}$时，每一轮只需要计算较少的集合元素，即$n\cdot 2^{-l}$。又因为GS协议中门限值$t$与集合元素大小$n$满足关系$t=O(n)$，且$2^l=O(n)$，因此$n\cdot 2^{-l}=O(1)$。所以SPDZ可以高效的实现$\mathcal{F}_{\cap-\mathrm{count}}$。

证毕.

ZHANG En, Ph. D., professor. His main research interests include cryptographic protocol design, secure multi-party computation and blockchain.

LIU Deng-Hui, M. S. candidate. His main research interests include cryptographic protocol design and secure multi-party computation.

DU Rui-Ying, Ph. D., professor. Her main research interests include network security and privacy protection.

Background

In recent years, research on threshold private set intersection (TPSI) has gained significant attention. TPSI

requires that the intersection is computed only when the size of the intersection among all parties exceeds a specified threshold. However, how to design an efficient TPSI protocol without leaking the size of the intersection remains an important challenge. To address this, researchers proposed probabilistic threshold private set intersection (PTPSI), a probabilistic variant of TPSI where the intersection is computed with a certain probability when the intersection size falls within a given range. Compared to deterministic TPSI protocols, PTPSI demonstrates higher efficiency in scenarios such as ride-sharing and federated learning.

However, existing PTPSI protocols secure against semi-honest adversaries rely on expensive generic circuit-based computations during the threshold testing phase, resulting in computational costs exponentially related to the number of participants. This inefficiency prevents them from scaling effectively to larger multi-party scenarios.

This paper overcomes this limitation by proposing an efficient PTPSI protocol based on bicentric zero-sharing (BZS) in the semi-honest model, which makes the protocol's running time independent of the number of participants. In a five-party setting, with each participant's set size being $n=2^{20}$ and the threshold value set to $t=0.5n$, the runtime of the existing protocol is 45.40 seconds, while the improved protocol has a total runtime of 9.62 seconds with a communication volume of 187.39 MB, achieving a 4.72× speedup. To resist collusion attacks, we further develop a privacy-enhanced PTPSI protocol that leverages oblivious pseudorandom functions to constrain malicious queries from colluding parties. In the same scenario, the time cost of the protocol is 30.25 seconds. Experimental results and algorithmic analysis demonstrate that our protocol achieves superior performance.

This research was supported by the National Natural Science Foundation of China (No. 62372157). Over the past decade, the research team's core areas of focus have include network security, federated learning cryptographic protocol design, privacy protection, and blockchain technology. Research team has published more than 80 papers in prominent domestic and international journals and conferences such as ACM CCS, Information Sciences, Journal of Software, and IEEE Transactions on Vehicular Technology.