高效的多方非平衡隐私集合交集协议

张 恩[1),2)] 李金磊[1)] 郑 东[1),3)] 禹 勇[4)] 刘登辉[1)]

[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)](西安邮电大学无线网络安全技术国家工程研究中心 西安 710121)

[4)](陕西师范大学人工智能与计算机学院 西安 710062)

摘 要 多方隐私集合交集(Multiparty Private Set Intersection,MPSI)协议允许多个参与方各自持有私有集合,在不泄露除交集以外任何信息的前提下,安全计算所有集合的交集,在泄露凭证检查、金融反欺诈和联邦学习等领域具有广泛应用。然而,现有非平衡隐私集合交集协议主要针对两方参与场景,缺乏针对多方参与场景的高效解决方法。为了解决MPSI及其变体协议在非平衡数据集上效率低下的问题,本文提出一种非平衡双中心零共享的方法。该方法结合零共享和不经意键值存储技术,将多方非平衡计算归约为两方非平衡计算,有效降低了通信和计算开销。然后,通过将该方法和两方非平衡隐私集合交集及其变体协议相结合,构建了一种新的高效且可扩展的多方非平衡隐私集合交集(Multiparty Unbalanced Private Set Intersection, MUPSI)及其变体协议。实验结果表明,在相同条件下,客户端的集合规模为2^{10},服务器端的集合规模为2^{27}时,本文提出的MUPSI协议的服务器端在线阶段耗时比目前最优的协议缩短约20%。此外,在32个参与方的场景下,客户端的集合规模为2^{10},服务器端的集合规模为2^{27}时,客户端在线阶段耗时约10秒,验证了该协议在大规模参与方以及集合规模差异显著场景下的有效性。

关键词 多方非平衡隐私集合交集;不经意键值存储;零共享;全同态加密;布谷鸟哈希

中图法分类号 TP309 **DOI号** 10.11897/SP.J.1016.2026.00193

Efficient Multiparty Unbalanced Private Set Intersection Protocol

ZHANG En[1),2)] LI Jin-Lei[1)] ZHENG Dong[1),3)] YU Yong[4)] LIU Deng-Hui[1)]

[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)](National Engineering Laboratory of Wireless Security Technology, Xi'an University of Posts and Telecommunications, Xi'an 710121)

[4)](School of Artificial Intelligence and Computer Science, Shaanxi Normal University, Xi'an 710062)

Abstract Multiparty Private Set Intersection (MPSI) protocols enable multiple participants to securely compute the intersection of their private datasets without revealing any extra information beyond the intersection itself, serving as a critical primitive for privacy-preserving data collaboration. Such protocols hold extensive practical application value. In compromised credential checking scenarios, they assist platforms in cross-verifying leaked account data while preventing the exposure of complete user lists. In financial anti-fraud work, they identify cross-institutional

收稿日期:2025-03-21;在线发布日期:2025-10-22。本课题得到国家密码科学基金(2025NCSF02025)、国家自然科学基金联合基金重点项目(U24B20149,U23A20302)、国家自然科学基金(No. 62372157)、陕西省重点研发计划重点产业创新链项目(2024GX-ZDCYL-01-09)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,主要研究领域为信息安全、密码协议和区块链。E-mail:zhangenzdrj@163.com。**李金磊**,硕士研究生,主要研究领域为信息安全和密码协议。**郑 东(通信作者)**,博士,教授,博士生导师,主要研究领域为密码学理论和云计算安全。E-mail:zhengdong@xupt.edu.cn。**禹 勇**,博士,教授,博士生导师,主要研究领域为公钥密码理论及应用、人工智能安全和区块链安全。**刘登辉**,硕士研究生,主要研究领域为信息安全和密码学。

fraud rings by matching and analyzing suspicious transaction records across different banks. In the field of federated learning, they realize effective alignment of sample spaces for distributed training data, all while safeguarding data privacy throughout the process. However, existing unbalanced Private Set Intersection (PSI) protocols mainly focus on two-party scenarios, where one party holds a large-scale dataset and the other holds a small-scale dataset. In multi-party unbalanced scenarios (e.g., one server holds massive data while multiple clients each hold only small datasets), directly applying traditional MPSI schemes usually encounters significant efficiency bottlenecks. Although some existing works have optimized MPSI protocols for unbalanced scenarios, their performance still degrades rapidly when there is a large difference in the size of datasets among participants. To address the inefficiency of MPSI and its variant protocols on unbalanced datasets, we propose an Unbalanced Bicentric Zero-Sharing (UBZS) method. This method combines the zero-sharing mechanism with Oblivious Key-Value Store (OKVS) technology. It designates two core participants as Pivot and Leader respectively, encodes the shared values using OKVS, and then distributes the encoded values to all participants. By converting multi-party unbalanced computation tasks into two-party interactions between the Pivot and the Leader, this method effectively eliminates redundant operations for ordinary clients and significantly reduces the overall communication and computational overhead. Furthermore, by integrating this bicentric framework with the optimized two-party Unbalanced Private Set Intersection (UPSI) and its variants, we construct an efficient and scalable Multi-Party Unbalanced Private Set Intersection (MUPSI) protocol and its variants (e.g., supporting the computation of the cardinality of multi-party unbalanced private set intersection). The experimental results show that under the same conditions, when the client's set size is 2^{10} and the server's set size is 2^{27}, the client online phase time consumption of the proposed MUPSI protocol is about 20% shorter than that of the current optimal protocol. Moreover, when the client's set size is 2^{10} and the server's set size is 2^{27}, in a scenario with 32 participants, the client's online execution time is approximately 10 seconds. These results validate the protocol's effectiveness in scenarios with a large number of participants and significant differences in set sizes. To further verify the efficiency of the UBZS method, which serves as the core technology of the MUPSI protocol, we conducted comparative experiments with the previously proposed Balanced Bicentric Zero-Sharing (BZS) protocol under Local Area Network (LAN) conditions. The experiments focused on the variation of time consumption as the size of the server-side dataset increased (in a scenario with one server and multiple clients). In the experiments, the server-side set size was sequentially set to 2^{20}, 2^{22}, 2^{24} and 2^{27}. The results show that the server-side time consumption of the BZS protocol was 0.053 seconds, 0.176 seconds, 0.622 seconds, and 4.675 seconds respectively, exhibiting a significant upward trend with the increase in data scale. In contrast, the UBZS method consistently maintained low and stable computational overhead under the same conditions, with corresponding time consumption values of 0.001 seconds, 0.001 seconds, 0.002 seconds, and 0.004 seconds. This result indicates that UBZS effectively avoids the significant time growth exhibited by BZS when processing large-scale datasets. It further verifies the efficiency of the unbalanced optimization design in reducing computational overhead and provides important technical support for the effectiveness of the MUPSI protocol in unbalanced multi-party scenarios.

Keywords multiparty unbalanced private set intersection; oblivious key-value store; zero-sharing; fully homomorphic encryption; cuckoo hashing

1 引　言

随着网络技术的发展,数据已成为驱动各行业发展的重要资源。为构建更全面的竞争优势,企业和组织日益重视跨机构数据协作。然而,如何在保障敏感数据安全的同时实现高效的数据协作,已成为学术界和工业界共同面临的核心挑战。尤其是在多参与方协作时,涉及参与方之间数据泄露和合谋攻击等多重问题,该问题将更加复杂。

多方隐私集合交集协议是应对上述隐私保护难题的核心技术手段之一。MPSI 协议是一种密码学协议,允许多个参与方各自拥有一个私有集合。该协议的目标是在不泄露除交集以外任何信息的前提下,由参与方共同计算出所有私有集合的交集。MPSI 协议在跨机构数据协作等领域具有广泛应用[1-2]。传统的 MPSI 协议通常建立在各参与方数据集规模相近且计算资源对等的假设基础之上。然而,在分布式数据库查询、跨机构数据协同分析等实际应用场景中,参与方的数据集规模往往呈现显著的差异(通常可达 3～5 个数量级),这种数据分布的非平衡性导致传统 MPSI 协议面临严重的性能瓶颈。

针对数据规模失衡导致的性能瓶颈,研究者提出了 MUPSI 协议[3]。该协议架构包含两类角色,其中参与方 P_n 作为服务器节点持有大规模私有集合 X_n(典型规模为 2^{20}、2^{24}、2^{27}),其余 $n-1$ 个参与方 $P_1,\cdots,P_{n-1}$ 作为客户端节点各自持有小规模私有集合(典型规模为 2^{10}、2^{11}、2^{12}),协议执行后所有参与方共同获得交集 $\bigcap_{i=1}^{n}X_i$,同时保证除该交集外,任何参与方均无法推断其他方的集合元素或统计特征。MUPSI 协议可应用于在线推荐系统[4](如交友网站)、机密数据共享[5](如安全事件信息)、边境管控[6](针对犯罪企图)、禁飞名单比对[7]以及网络安全[8](如僵尸网络检测和基于可疑 IP 集合的入侵检测)等领域。下面是 3 个具体的应用场景:

(1)凭证泄露检查(Compromised Credential Checking,C3)[9-10]是一个典型应用场景。客户端需要检查其私有凭证集合是否与服务器维护的大规模泄露凭证数据库存在交集。客户端的凭证集合规模通常为几百或几千条记录,而服务器维护的泄露凭证数据库则可达数十亿条记录,并持续增长。

(2)机构可能需要检查客户是否同时存在于多家银行的黑名单或高风险名单中(银行的数据规模根据银行的规模不同,差异可能非常大);贷款人也需要验证借款人是否在多家银行负债。MUPSI 协议可以在不泄露任何其他私人信息的前提下,帮助他们找出这些可疑人员[11]。

(3)大型医院(持有大规模数据集)和小型诊所(持有小规模数据集)可以使用 MUPSI 协议匹配出共同的患者记录。在识别出交集后,可针对这些重叠数据应用联邦学习,进行联合分析,并保护非重叠数据的隐私[12]。这对于分布式机器学习模型尤为重要。

在非平衡隐私集合交集协议(Unbalanced Private Set Intersection,UPSI)的研究领域中,现有工作主要集中于两方参与场景的协议设计。然而,当协议需要扩展到多方参与场景时,尤其是在参与方的集合规模呈现显著差异的情况下(例如大集合规模达到 2^{27} 而小集合规模仅为 2^{10}),现有方法在计算效率和通信开销方面仍面临挑战[3]。为了解决上述问题,本文的主要贡献总结如下:

(1)通过结合无条件零共享和不经意键值存储技术提出了一种称为非平衡双中心零共享的方法(Unbalanced Bicentric Zero-Sharing,UBZS),将多方非平衡计算问题转化为两方非平衡计算问题,在保证安全性的同时,有效降低了通信和计算复杂度。

(2)通过将 UBZS 与 Cong 等人[13]提出的当前通信开销最优的两方 UPSI 协议相结合,构建了一种高效且可扩展的 MUPSI 协议。此外,本文对 Mahdavi 等人[14]提出的标准 UPSI 协议进行修改,构建了一种新的两方非平衡隐私集合交集基数协议。在此基础上,结合 UBZS,构建了一种高效且可扩展的多方非平衡隐私集合交集基数协议(Multi-party Unbalanced Private Set Intersection Cardinality,MUPSI-CA)。

(3)对 UBZS、MUPSI 和 MUPSI-CA 协议分别进行了实验评估。针对 UBZS 协议,在相同条件下,本文提出的 UBZS 协议在持有大集合的参与方在线执行时,相较于 Gao 等人[15]提出的双中心零共享(Bicentric Zero Sharing,BZS)协议更具优势,且大集合规模越大,优势越突出。针对 MUPSI 协议,在相同的实验环境下,客户端的集合规模为 2^{10},服务器端的集合规模为 2^{27} 时,本文提出的 MUPSI 协议的客户端在线阶段耗时比文献[15]中协议缩短约 20%。此外,在 32 个参与方的场景下,客户端的集合规模为 2^{10},服务器端的集合规模为 2^{27} 时,协议中客户端在线执行时间约 10 秒,验证了该协议在大规模参与方以及集合规模差异显著场景下的有效性。

2 相关工作

2.1 非平衡隐私集合交集协议技术现状

非平衡隐私集合交集[3,13-14,16-19]协议是隐私集合交集(Private Set Intersection,PSI)协议的一种变体,其典型特征是接收方的集合规模远小于发送方。接收方通常为计算能力和通信带宽受限的低功耗设备,扮演客户端角色;发送方则扮演服务器角色。尽管已有诸多高效的 PSI 协议[20-28],但这些协议的通信复杂度通常与大集合的规模呈线性或亚线性关系。这种特性使得它们在集合大小差异显著的 UPSI 场景下效率低下,甚至由于通信开销过大而无法在实际中应用。

针对 PSI 协议在非平衡场景下效率低下的问题,目前通信开销表现最佳的解决方法[13,18-19]主要依赖于全同态加密(Fully Homomorphic Encryption, FHE)。FHE 最显著的特点是允许在密文上执行任意计算,解密后的结果与在明文上执行相同计算的结果一致。全同态加密为 UPSI 协议的实现提供了强大的技术支撑。

Chen 等人[19]提出了基于 FHE 的 UPSI 协议,通过优化同态计算中函数的乘法电路深度,降低了计算复杂度,从而提升协议的效率和实用性,为后续研究奠定了重要的理论和实践基础。该协议的核心思想是服务器基于其输入集合构建一个插值多项式 $P(x)$,$P(x)$ 的根集合即为服务器的输入集合。客户端将其输入集合中的每个元素加密后发送给服务器,服务器则利用 FHE 对元素密文进行同态计算,计算 $P(x)$ 在这些密文上的值。根据多项式的性质,若客户端某个元素 x_0 属于交集,则 $P(x_0)=0$,反之 $P(x_0)\neq 0$。

在后续研究中,Chen 等人[18]基于文献[19]进行了改进,通过结合不经意伪随机函数(Oblivious Pseudorandom Function,OPRF),构建了安全性更高的协议,并将其安全性从半诚实模型提升到恶意模型。此外,文献[18]还使协议能够处理任意比特长度的元素,增强了协议的通用性和实用性。Cong 等人[13]则在文献[18]的基础上,利用 Extremal Postage Stamp Bases 和 Paterson Stockmeyer 算法进一步优化了计算和通信效率。

近年来,研究者们也探索了基于 FHE 与恒定权重编码算法相结合的 UPSI 协议[14]。尽管这一协议在离线计算方面有所提升,但在通信开销和在线执行速度等关键指标上,仍然难以超越 Chen 等人[18-19]和 Cong 等人[13]提出的基于多项式插值和 FHE 优化的方法。

2.2 多方隐私集合交集协议技术现状

MPSI 协议近年来受到了研究者的广泛关注和深入研究,并在平衡集合场景下取得了显著进展[15,29-35],例如在协议效率、安全性和功能性等方面都涌现出许多优秀的解决方法。然而,针对非平衡集合场景下的 MPSI 协议(即参与方集合规模差异显著的情况),如何设计高效且通用的方法仍然是一个重要的研究挑战。

Lai 等人[36]提出的协议是早期 MPSI 协议的代表性工作之一。然而,该协议存在安全漏洞:它将每个参与方集合的元素编码为布隆过滤器(Bloom Filter,BF),并以明文形式广播给其他参与方。由于布隆过滤器本身存在固有的概率性误判,该协议会不可避免地泄露参与方集合的部分信息。为了解决这一问题,Miyaji 等人[37-38]提出了基于 BF 和加法同态加密的 MPSI 协议。通过使用同态加密技术保护 BF 中的内容,避免信息泄露。文献[38]中还引入了阈值隐私集合交集(Threshold Private Set Intersection,TPSI)协议的概念,即允许参与方当且仅当交集大小超过预设的阈值时才能获得交集。但文献[37]和文献[38]并未提供严格的安全性证明或分析,无法保证协议是否满足标准的安全性定义。早期的 MPSI 协议大多缺乏具体的实现和性能评估,因此在实际应用中价值有限。

Kolesnikov 等人[32]提出了首个基于不经意传输(Oblivious Transfer,OT)的 MPSI 协议,为后续相关研究奠定了重要的基础。该协议的构造主要基于零共享技术和不经意可编程伪随机函数(Oblivious Programmable Pseudorandom Function, OPPRF)。文献[32]以 Ishai 等人[27]提出的基于 OT 的 OPRF 为基础,结合混淆布隆过滤器(Garbled Bloom Filter, GBF)、多项式和哈希表,构造了三个 OPPRF 实例。通过将 OPPRF 和零共享协议相结合,文献[32]分别在半诚实模型和增强半诚实模型下构造了高效的 MPSI 协议。

Chandran 等人[39]对文献[32]中的协议进行了改进,实现了 5 倍左右的性能提升,同时设计并实现了电路 MPSI 和阈值 MPSI 协议。然而,文献[39]的协议考虑的是一个较弱的敌手,即假设最多可以腐败 $t<n/2$ 的参与方(诚实大多数)。这种降低安全性的假设可以移除文献[32]中的性能瓶颈(条件

零共享),他们使用 (n,t) 秘密共享代替条件零共享。秘密共享方法确保了在协议执行过程中,即使任意 t 个或更少的参与方合谋,也无法获取任何中间计算结果,从而保护了参与方的隐私。尽管如此,文献[39]的实验结果表明,在一个包含 15 个参与方(7 个合谋参与方)且每个参与方集合规模为 2^{20} 的广域网(WAN)环境下,MPSI 协议执行时间约为 244 秒,这在许多对实时性要求较高的应用场景中,例如在线广告匹配或金融交易欺诈检测等,是难以接受的。

Ben-Efraim 等人[29]结合半诚实 MPSI 协议[8]和恶意两方 PSI 协议[40],提出了一种能够抵抗恶意敌手攻击的 MPSI 协议。实验结果表明,在一个包含 8 个参与方且每个参与方集合规模为 2^{20} 的局域网(LAN)环境下,协议的在线执行时间约为 78 秒,该协议的扩展性与先前的大多数 MPSI 协议类似,仍有待提高。

Nevo 等人[31]基于文献[32]的工作,进一步研究了 MPSI 协议的恶意安全性,并区分了无合谋和合谋两种场景,提出了相应的抵抗恶意敌手攻击的 MPSI 协议。针对无合谋场景,他们设计了一种基于对称密码学原语的高效协议,但该协议无法抵抗合谋攻击。为此,他们提出了另一种利用 OPPRF 和 ZeroXOR 技术的通用协议,有效地增强了抗合谋攻击的能力,从而提供了更强的安全性保证。

Gao 等人[15]在文献[31]的工作基础上,提出了一种双中心零共享协议,为高效实现 MPSI 及其变体提供了一种新的途径。该协议的核心思想是将多方计算问题归约为两方计算问题,从而显著降低了计算和通信开销。BZS 协议不仅适用于标准 MPSI 协议,还可以应用于 MPSI 的各种变体,例如阈值多方隐私集合交集(Threshold Multiparty Private Set Intersection, TMPSI)协议和多方隐私集合交集基数(Multiparty Private Set Intersection Cardinality, MPSI-CA)协议等。在广域网(WAN)环境下,针对 15 个参与方(其中 14 个合谋),每个参与方集合规模为 2^{20} 的实验表明,基于 BZS 实现的 MPSI 协议的执行时间比文献[31]的方法缩短了约 95%,通信开销降低了约 96%。这些结果表明,BZS 协议在大规模参与方时具有显著的性能优势。

在 PSI 协议的研究领域,当前技术在两方非平衡场景协议设计和多方平衡场景协议构建方面已经取得了显著的成果。然而,当两方非平衡扩展到多方非平衡场景时,尤其是在参与方的集合规模呈现显著差异的情况下,现有协议在计算效率和通信开销方面还没有高效的解决方法。

3　预备知识

本文提出的协议涉及不经意键值存储、无条件零共享、全同态加密、恒定权重编码和布谷鸟哈希等技术。本节将对上述技术的相关概念及基础知识进行描述。本文所使用的符号及其说明如表 1 所示。

表 1　符号定义

符号	说明	符号	说明
λ	统计安全参数	c	全同态加密的密文值
κ	计算安全参数	w	恒定权重编码中的权重大小
σ	参与方输入元素的位长	ℓ	伪随机函数的输出长度
m	哈希函数映射长度	N_i	参与方 P_i 的集合大小
n	参与方个数	$X_i=\{x_1,\cdots,x_{N_i}\}$	参与方 P_i 的集合
h	布谷鸟哈希的个数	I_{CA}	参与方交集基数
ε	布谷鸟哈希中的常量	I	参与方交集
k	伪随机函数的密钥	S_i	参与方 P_i 的 OKVS 结构
t	腐败方的个数	$\prod^{n,N_i}$	协议有 n 个参与方,集合大小分别为 $N_{i,i\in[1,n]}$

3.1　无条件零共享

零共享(Zero-Sharing, ZS)[15,31-32]是一种将秘密值分解成若干份额的技术。ZS 的优势在于它的抗合谋性,使其成为构造抗合谋攻击 MPSI 协议的常用技术手段之一。无条件零共享的核心思想是参与方 $P_{i,i\in[1,n-1]}$ 对 $j\in[i,n]$ 取样随机种子 r_i^j,并将随机种子 r_i^j 发送给 P_j。对于某个集合元素 x,P_i 可以计算关于元素 x 的零共享的份额,即

$$s_i(x)=(\bigoplus_{j=1}^{i-1}F(r_j^i,x))\oplus(\bigoplus_{j=i+1}^{n}F(r_i^j,x)) \quad (1)$$

若存在某个元素 x 是交集元素,则有 $\bigoplus_{i=1}^{n}s_i(x)=0$。

3.2　不经意键值存储

定义 1. 键值存储(Key-Value Store, KVS)[21,25-26]是一种数据结构,由键的集合 X、值的集合 Y 和一组函数 H 组成:

$\text{Encode}_H(\cdot)$：该算法的输入为一个包含 n 个键值对的集合 $\{(x_i,y_i)\}_{i\in[1,n]}$，其中 $x_i\in X$ 和 $y_i\in Y$。算法编码该输入集合，并输出一个数据结构 S 或以可忽略的概率输出符号 $\perp$。

$\text{Decode}_H(\cdot,\cdot)$：给定数据结构 S 和输入值 x，该算法执行解码操作，并输出键值 x 映射的值 y。

正确性：对于 $(x,y)\in A$ 和 $S\neq\perp$ 所有不同的键 $A\subseteq X\times Y$：

$$S\leftarrow\text{Encode}_H(A)\Rightarrow\text{Decode}_H(S,x)=y \quad (2)$$

定义 2. 如果对于任意两个不同键集合 $\{x_1^0,x_2^0,\cdots,x_n^0\}$ 和 $\{x_1^1,x_2^1,\cdots,x_n^1\}$，编码算法 $\text{Encode}_H(\cdot)$ 对这两个集合的编码均成功（即不输出 $\perp$）的情况下，输出结果 $\mathcal{R}(x_1^0,x_2^0,\cdots,x_n^0)$ 和 $\mathcal{R}(x_1^1,x_2^1,\cdots,x_n^1)$ 在计算意义上是不可区分的，则称一个 KVS 是 OKVS。本文中，后续将使用 Encode 表述 Encode_H，将使用 Decode 表述 Decode_H。$\mathcal{R}$ 的详细描述如算法 1 表示。

算法 1. $\mathcal{R}$ 构造算法

输入：$x_1,x_2,\cdots,x_n$

输出：数据结构 S

1. FOR $i=1$ to n
2. $\quad y_i\leftarrow Y$
3. END FOR
4. S $\leftarrow$ Encode($\{(x_1,y_1),(x_2,y_2),\cdots,(x_n,y_n)\}$)
5. RETURN S

总的来说，对于任意两组编码结构 S_0 和 S_1，S_0 和 S_1 在计算意义上是不可区分的。

3.3 全同态加密

全同态加密[41]是一种加密技术，允许在不解密数据的情况下对密文进行任意计算。解密后的结果与对明文进行相同计算的结果一致。然而，FHE 通常计算开销较大。为了提高效率，在实际应用中常采用分层全同态加密（Leveled Fully Homomorphic Encryption，LFHE），该方案通过限制所支持的算术电路深度（即电路的层数）来降低计算复杂度。本文协议采用了 LFHE 以平衡安全性和效率。一个 LFHE 通常由一系列随机化算法构成，算法描述如下：

（1）FHE.Gen($1^\lambda,\partial$) 是一个密钥生成算法。以安全参数 λ 和辅助参数 ∂ 作为输入，输出公钥 pk、私钥 sk 和评估密钥 evk 的密钥三元组 (pk,sk,evk)。

（2）FHE.Enc(m,pk) 是一个加密算法。以明文消息 m 和公钥 pk 作为输入，输出密文 c。

（3）FHE.Dec(c,sk) 是一个解密算法。以密文 c 和私钥 sk 作为输入，输出明文消息 m。

（4）FHE.Eval($f,(c_1,c_2,\cdots,c_n),evk$) 是一个评估算法。以评估密钥 evk、函数 f 和一组输入 $c_1,c_2,\cdots,c_n$ 作为输入，其中 $c_1,c_2,\cdots,c_n$ 可以是明文加密后的密文或先前同态计算的结果。该算法输出一个密文 c'，该密文是对输入密文进行同态计算 f 的结果。

对于任何同态计算，其输出密文的分布与对相应明文计算结果进行新加密所得密文的分布在计算上是不可区分的，则称 FHE 实现了电路隐私。通过这种方式，可以有效地隐藏在加密数据上被评估的电路。详细请参考文献[41]。

3.4 非平衡隐私集合交集协议

本节给出理想安全模型下两方非平衡隐私集合交集协议的形式化定义。UPSI 协议的理想功能 $\mathcal{F}_{\text{UPSI}}^{2,N_i}$ 定义如下：

参数：接收方 P_1 持有一个大小为 N_1 的集合 X_1；发送方 P_2 持有一个大小为 N_2 的集合 X_2，其中 $N_2\gg N_1$。

功能：等待接收方 P_1 输入 X_1；等待发送方 P_2 输入 X_2；接收方 P_1 得到交集集合 $I=|X_1\cap X_2|$。

本文采用 Cong 等人[13]提出的协议来实现理想功能函数 $\mathcal{F}_{\text{UPSI}}^{2,N_i}$。

3.5 非平衡隐私集合交集基数协议

非平衡隐私集合交集基数（Unbalanced Private Set Intersection Cardinality，UPSI-CA）是 UPSI 协议的一种变体。本节给出理想安全模型下两方非平衡隐私集合交集基数协议的形式化定义，UPSI-CA 协议的理想功能 $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$ 定义如下：

参数：接收方 P_1 持有一个大小为 N_1 的集合 X_1；发送方 P_2 持有一个大小为 N_2 的集合 X_2，其中 $N_2\gg N_1$。

功能：等待接收方 P_1 输入 X_1；等待发送方 P_2 输入 X_2；接收方 P_1 得到交集基数 $I_{\text{CA}}=|X_1\cap X_2|$。

本文对 Mahdavi 等人[14]提出的两方 UPSI 协议进行修改来实现理想功能函数 $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$。

3.6 哈希函数

布谷鸟哈希（Cuckoo Hashing）[42-43]是一种将 n 个元素以大于等于 $1-2^{-\lambda}$ 的概率成功插入到 m 个 bin 中的哈希方法，其使用 h 个哈希函数 $H_1,H_2,\cdots,H_h$，其中 $m=(1+\varepsilon)\cdot n$，$\varepsilon>0$ 为常数。映射过程如下：对于待插入的元素 x，随机选择 $i\in[1,h]$，如果 $H_i(x)$ 对应的 bin 为空，则将 x 直接插入

该 bin。如果 $H_i(x)$ 对应的 bin 不为空,则将该 bin 中原有的元素 y 踢出,将 x 插入到 bin,然后递归地尝试插入 y。此过程重复进行,直到所有元素都插入成功或达到预设的递归深度阈值。若达到递归深度阈值,则插入操作失败,需要重新进行哈希。布谷鸟哈希的部分变体采用了一种称为“stash”的辅助存储结构,用于存储无法成功插入的元素。本文采用无 stash 的布谷鸟哈希方法。

3.7　恒定权重编码

恒定权重(Constant-Weight,CW)码是一种特殊的二进制码,其所有码字均具有相同的汉明重量(码字中“1”的数量)。本文将码字共享的公共汉明重量记为 w。函数 CW-Encode(x,l,w) 用于将元素 x 编码为长度为 l 比特,汉明重量为 w 的 CW 码字集合。本文使用 Mahdavi 等人[44]中的算术恒定权重相等算法来比较恒定权重码字和汉明重量 w。详细细节如算法 2 所示。

算法 2. CW-EQ($\cdot$) 构造算法

输入:两个字符串 x 和 y

输出:一个比特 e

1. $w'=0$
2. FOR $i=1$ to l
3. $\quad w'=w'+x_i \cdot y_i$
4. END FOR
5. $e=(1/w!)\prod_{i=1}^{w}(w'-i)$ //其中 $w!$ 表示阶乘
6. RETURN e

3.8　安全模型

本文在半诚实安全模型下对协议的安全性进行了形式化证明。在半诚实模型中,规定所有参与方必须严格遵循协议的执行规范。然而,好奇的参与方可能会试图根据协议执行过程中获得的信息来推断结果之外的额外信息。本节将给出半诚实模型下协议安全性的正式定义[45]。

定义 3. 设 $f:(\{0,1\}^*)^n \to (\{0,1\}^*)^n$ 是一个 n 元函数,其中 $f_i(\bar{\boldsymbol{X}})$ 表示 $f(\bar{\boldsymbol{X}})$ 的第 i 个输出分量,$\bar{\boldsymbol{X}}=(X_1,X_2,\cdots,X_n)$ 是输入向量。设 $\prod$ 是计算函数 f 的 n 方协议。假设敌手腐化了参与方集合 $C=\{i_1,i_2,\cdots,i_t\}\subseteq[1,n]$。记 $f_C(\bar{\boldsymbol{X}})=(f_{i_1}(\bar{\boldsymbol{X}}),\cdots,f_{i_t}(\bar{\boldsymbol{X}}))$ 和 $\boldsymbol{X}_C=(X_{i_1},X_{i_2},\cdots,X_{i_t})$。在真实世界中,敌手可以获取协议的输出 $\text{Output}^{\prod}(\bar{\boldsymbol{X}})$,以及被腐败的参与方视图 $\text{View}_C^{\prod}(\bar{\boldsymbol{X}})=(C,\text{View}_{i_1}^{\prod}(\bar{\boldsymbol{X}}),\cdots,\text{View}_{i_t}^{\prod}(\bar{\boldsymbol{X}}))$。在理想世界中,存在一个概率多项式时间(Probabilistic Polynomial-Time,PPT)算法的模拟器 Sim,Sim 可以仅根据被腐化参与方的输入 $\boldsymbol{X}_C$ 和函数在腐败方上的输出 $f_C(\bar{\boldsymbol{X}})$ 来模拟敌手的视图。对于所有可能 $C\subseteq[1,n]$,真实世界和理想世界是计算不可区分的,即

$$\begin{aligned}&\{\text{Sim}(C,\boldsymbol{X}_C,f_C(\bar{\boldsymbol{X}})),f(\bar{X})\}_{\bar{\boldsymbol{X}}}\\&\overset{c}{\equiv}\{\text{View}_C^{\prod}(\bar{\boldsymbol{X}}),\text{Output}^{\prod}(\bar{\boldsymbol{X}})\}_{\bar{\boldsymbol{X}}}\end{aligned}\tag{3}$$

特别地,当 f 是确定性函数,即函数输出与真实输出不同的概率 $Pr[f(\bar{\boldsymbol{X}})\neq\text{Output}^{\prod}(\bar{\boldsymbol{X}})]$ 可忽略不计时,式 (3)可简化为

$$\{\text{Sim}(C,\boldsymbol{X}_C,f_C(\bar{\boldsymbol{X}}))\}_{\bar{\boldsymbol{X}}}\overset{c}{\equiv}\{\text{View}_C^{\prod}(\bar{\boldsymbol{X}})\}_{\bar{\boldsymbol{X}}}\tag{4}$$

4　非平衡双中心零共享

4.1　技术概述

将多方计算问题转化为两方计算问题的方法已在现有研究中实现[31,35]。在这些工作中,每个参与方为自身集合中的每个元素生成对应的伪随机值,并由两个中心参与方进行聚合。当且仅当元素 x 属于所有参与方的交集时,两个中心参与方所持有的聚合值才相等。

Gao 等人[15]利用上述思想提出双中心零共享方法,该方法将所有参与方的私有集合作为输入,并将两个聚合集合分别返回给两个中心参与方,从而支持后续两方 PSI,MPSI 的结果由两方 PSI 的结果输出。对于 $i\in[1,n]$,每个参与方 P_i 拥有一个集合 X_i,假设 P_1 和 P_n 分别为两个中心方 Pivot 和 Leader。下面是对双中心零共享应用到非平衡场景的讨论。

情形 1:P_n 持有大规模集合,当 $P_2,P_3,\cdots,P_{n-1}$ 充当客户端持有小规模集合时,P_n 聚合 $P_2,P_3,\cdots,P_{n-1}$ 发送的编码数据结构,该场景下协议时间复杂度与大规模集合相关,当 P_n 持有集合规模越大,协议效率越低。

情形 2:P_1 持有大规模集合,在协议执行过程中 P_1 需要发送编码数据结构给 P_n。该过程的通信开销与大规模集合相关,在非平衡场景下协议通信开销较大。

非平衡双中心零共享,其核心思想与双中心零共享相同,但 UBZS 比 BZS 更适合非平衡场景。对于 $i\in[1,n]$ 和 $j\in[1,N_i]$,每个参与方 P_i 拥有一个集合 $X_i=\{x_i^j\}$,N_i 表示集合 X_i 的大小,其中

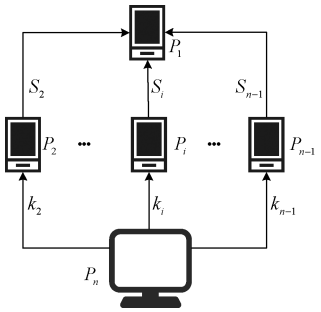

图 1 UBZS 协议流程图

N_n 远远大于 $N_1,N_2,\cdots,N_{n-1}$。协议的目标是使 P_1 和 P_n 分别获得集合 $A=\{a^1,a^2,\cdots,a^{N_1}\}$ 和 $B=\{b^1,b^2,\cdots,b^{N_n}\}$。当且仅当存在 $j\in[1,N_1]$ 和 $j'\in[1,N_n]$，满足 $x_n^{j'}=x_1^j$ 且 x_1^j 属于前 $n-1$ 个参与方集合的交集，即 $x_1^j\in\bigcap_{i=1}^{n-1}X_i$ 时，才有 $a^j=b^{j'}$ 成立。图 1 给出了 UBZS 协议的简化流程图。UBZS 协议的理想功能 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 定义如下：

参数：对于 $i\in[1,n]$ 和 $j\in[1,N_i]$，每个参与方 P_i 拥有一个集合 $X_i=\{x_i^j\}$，N_i 表示集合 X_i 的大小，且 N_n 远远大于 $N_1,N_2,\cdots,N_{n-1}$。

功能：等待参与方 P_i 的输入 X_i；P_1 获得集合 $A=\{a^1,a^2,\cdots,a^{N_1}\}$；$P_n$ 获得集合 $B=\{b^1,b^2,\cdots,b^{N_n}\}$。

4.2 协议构造

OKVS 和伪随机函数是 UBZS 构建的主要组成部分，详细协议如算法 3 所示。具体流程如下：

参数：对于 $i\in[1,n]$ 和 $j\in[1,N_i]$，每个参与方 P_i 拥有一个集合 $X_i=\{x_i^j\}$，N_i 表示集合 X_i 的大小，其中 N_n 远远大于 $N_1,N_2,\cdots,N_{n-1}$。伪随机函数 $F:\{0,1\}^\kappa\times\{0,1\}^\sigma=\{0,1\}^\ell$。

客户端离线阶段：

(1)参与方 P_n 随机取样 $n-2$ 个伪随机函数 F 的密钥，分别记为 $k_2,k_3,\cdots,k_{n-1}$。P_n 使用 $k_2,k_3,\cdots,k_{n-1}$ 计算 $x'^j_n=F(k_2,x_n^j)\oplus\cdots\oplus F(k_{n-1},x_n^j)_{j\in[1,N_n]}$，并将计算结果组成的集合记为 $X'_n=\{x'^1_n,x'^2_n,\cdots,x'^{N_n}_n\}$。

客户端在线阶段：

(1)对于 $i\in[2,n-1]$，参与方 P_n 将密钥 k_i 分发给参与方 P_i。

(2)对于 $i\in[2,n-1]$，参与方 P_i 使用密钥 k_i 计算 $x'^j_i=F(k_i,x_i^j)_{j\in[1,N_i]}$，并将计算结果组成的集合记为 $X'_i=\{x'^1_i,x'^2_i,\cdots,x'^{N_i}_i\}$。

(3)参与方 $P_2,P_3,\cdots,P_{n-1}$ 分别使用各自集合 $X_2,X_3,\cdots,X_{n-1}$ 作为键的集合，$X'_2,X'_3,\cdots,X'_{n-1}$ 作为值的集合进行 OKVS 编码。具体来说，对于 $i\in[2,n-1]$，参与方 P_i 计算 $S_i\leftarrow\text{Encode}(\{(x_i^j,x'^j_i)\})_{j\in[1,N_i]}$。参与方 $P_2,P_3,\cdots,P_{n-1}$ 分别持有 $S_2,S_3,\cdots,S_{n-1}$。

(4)参与方 $P_2,P_3,\cdots,P_{n-1}$ 将编码结果 $S_2,S_3,\cdots,S_{n-1}$ 发送给 P_1。参与方 P_1 对收到的 $S_2,S_3,\cdots,S_{n-1}$ 进行 OKVS 进行异或运算并将其结果进行解码操作。即对于 $j\in[1,N_1]$，P_1 计算 $x'^j_1=\text{Decode}(\bigoplus_{i=2}^{n-1}S_i,x_1^j)$，并将计算结果组成的集合记为 $X'_1=\{x'^1_1,x'^2_1,\cdots,x'^{N_1}_1\}$。

(5)对于 $i\in[1,N_1]$，参与方 P_1 将 x'^i_1 设为 a^i，构造集合 $A=\{a^1,a^2,\cdots,a^{N_1}\}$。对于 $j\in[1,N_n]$，参与方 P_n 将 x'^j_n 设为 b^j，构造集合 $B=\{b^1,b^2,\cdots,b^{N_n}\}$。

算法 3. $\prod_{\text{UBZS}}^{n,N_i}$ 协议构造算法

输入：对于 $i\in[1,n]$，每个参与方 P_i 输入集合 X_i。

输出：输出集合 A 给 P_1，输出集合 B 给 P_n。

1. FOR $i=2$ to $n-1$//客户端离线阶段
2. $k_i\leftarrow\{0,1\}^\kappa$
3. END FOR
4. // P_n 使用密钥 k_i 计算伪随机函数
5. FOR $j=1$ to N_n
6. $x'^j_n=0$
7. FOR $i=2$ to $n-1$
8. //计算同一元素所有伪随机函数的异或
9. $x'^j_n=x'^j_n\oplus F(k_i,x_n^j)$
10. END FOR
11. ADD x'^j_n to X'_n
12. END FOR
13. FOR $i=2$ to $n-1$//客户端在线阶段
14. //参与方 P_i 使用得到的 k_i 计算伪随机函数
15. FOR $j=1$ to N_i
16. $x'^j_i=F(k_i,x_i^j)$
17. ADD x'^j_i to X'_i //设伪随机函数为新的集合
18. END FOR
19. // P_i 使用集合 X_i 和 X'_i 计算 OKVS 结构
20. $S_i\leftarrow\text{Encode}(X_i,X'_i)$
21. END FOR
22. // P_1 对其他参与方发送的 S_i 进行异或并解码
23. $S=S_2\oplus\cdots\oplus S_{n-1}$//OKVS 的线性性质
24. FOR $j=1$ to N_1
25. $x'^j_1=\text{Decode}(S,x_1^j)$
26. ADD x'^j_1 to X'_1
27. END FOR
28. SET $A\equiv X'_1$//赋值 X'_1 为输出集合 A
29. SET $B\equiv X'_n$ //赋值 X'_n 为输出集合 B

4.3 安全证明

定理 1. 当参与方 P_1 未被敌手腐化,算法 3 中的 $\prod_{\mathrm{UBZS}}^{n,N_i}$ 协议在半诚实模型下安全地实现了理想功能函数 $\mathcal{F}_{\mathrm{UBZS}}^{n,N_i}$。

证明. 正确性:算法 3 中 $\prod_{\mathrm{UBZS}}^{n,N_i}$ 协议的输出包含 $A=\{a^1,a^2,\cdots,a^{N_1}\}$ 和 $B=\{b^1,b^2,\cdots,b^{N_n}\}$,其中

$$b^j=\bigoplus_{i=2}^{n-1}F(k_i,x_n^j)_{j\in[1,N_n]} \tag{5}$$

$$a^j=\mathrm{Decode}(\bigoplus_{i=2}^{n-1}S_i,x_1^j)_{j\in[1,N_1]} \tag{6}$$

通过伪随机函数 F 的伪随机性和 OKVS 的不经意性,可以有效地确保每个 a^j 和 b^j 的均匀随机分布。当且仅当存在 $j\in[1,N_1]$ 和 $j'\in[1,N_n]$,使得 $x_n^{j'}=x_1^j$ 且 x_1^j 属于前 $n-1$ 个参与方集合的交集,即 $x_1^j\in X_1\cap x_2^j\in X_2\cap\cdots\cap X_{n-1}$ 时,才有 $a^j=b^{j'}$ 成立。具体证明如下:

对于 $j\in[1,N_1]$ 和 $j'\in[1,N_n]$,若存在 $x_1^j\in X_1$ 是交集,则存在 j' 使得 $x_n^{j'}=x_1^j$,根据 $\prod_{\mathrm{UBZS}}^{n,N_i}$ 协议计算 a^j 和 $b^{j'}$:

$$\begin{aligned}a^j&=\mathrm{Decode}(S_2,x_1^j)\oplus\cdots\oplus\mathrm{Decode}(S_{n-1},x_1^j)\\&=\mathrm{Decode}(S_2\oplus\cdots\oplus S_{n-1},x_1^j)\\&=F(k_2,x_1^j)\oplus\cdots\oplus F(k_{n-1},x_1^j)\end{aligned} \tag{7}$$

$$b^{j'}=F(k_2,x_n^{j'})\oplus\cdots\oplus F(k_{n-1},x_n^{j'}) \tag{8}$$

由于 $x_n^{j'}=x_1^j$,故 $a^j=b^{j'}$。

对于 $j\in[1,N_1]$,若 $x_1^j\in X_1$ 不在交集中,则存在 $i\in[2,n-1]$,使得 $x_1^j\notin X_i$。此时,由于 OKVS 的不经意性,在计算 $\mathrm{Decode}(S_i,x_1^j)$ 时,P_1 得到的值为随机值,而非对应的 $F(k_i,x_1^j)$,从而导致 $a^j\notin B$。

安全性:对于 $|\mathcal{C}|=t$ 和 $|\mathcal{H}|=n-t$,设 $\mathcal{C}$ 和 $\mathcal{H}$ 分别表示合谋参与方和诚实参与方的集合。由于本文对无条件零共享进行了简化,使其只能实现最多抵抗有条件的 $n-1$ 参与方的合谋,在 UBZS 协议中表现为参与方 P_1 不能参与合谋。为了说明如何在理想世界中模拟 $\mathcal{C}$ 的视图,本文根据 P_n 是否参与合谋分为二种情况进行讨论。

情形 1:P_n 不参与合谋。在这种情况下,$P_1\in\mathcal{H}$ 和 $P_n\in\mathcal{H}$。模拟器 $\mathrm{Sim}_{\mathrm{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)$ 需要模拟诚实参与方发送给合谋集合 $\mathcal{C}$ 的信息。模拟如下,对于所有的 $P_i\in\mathcal{C}$,此时只有诚实参与方 P_n 发送对应伪随机函数的密钥给合谋参与方 P_i,因此,Sim 随机取样伪随机函数的密钥 k'_i 并将 k'_i 添加到视图。

由于伪随机函数 F 的伪随机性和 OKVS 的不经意性,对于 $\bar{\boldsymbol{X}}=(X_1,X_2,\cdots,X_n)$,可知模拟器 Sim 构造的视图与真实世界中敌手的视图在计算意义上不可区分。

$$\{\mathrm{Sim}_{\mathrm{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)\}_{\bar{\boldsymbol{X}}}\overset{c}{\equiv}\{\mathrm{View}_c^{\prod_{\mathrm{UBZS}}^{n,N_i}}(\bar{\boldsymbol{X}})\}_{\bar{\boldsymbol{X}}} \tag{9}$$

情形 2:P_n 参与合谋,在这种情况下,$P_n\in\mathcal{C}$ 和 $P_1\in\mathcal{H}$。模拟器 $\mathrm{Sim}_{\mathrm{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),B)$ 需要模拟诚实参与方发送给合谋集合 $\mathcal{C}$ 的信息。模拟如下,通过观察 UBZS 协议的执行步骤发现,在这种情况下,诚实参与方并没有发送任何信息给合谋集合。此时,合谋集合无法获取任何诚实参与方的多余信息。

对于 $\bar{X}=(X_1,X_2,\cdots,X_n)$,可知模拟器 Sim 构造的视图与真实世界中敌手的视图在计算意义上不可区分。

$$\{\mathrm{Sim}_{\mathrm{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),B)\}_{\bar{\boldsymbol{X}}}\overset{c}{\equiv}\{\mathrm{View}_c^{\prod_{\mathrm{UBZS}}^{n,N_i}}(\bar{\boldsymbol{X}})\}_{\bar{\boldsymbol{X}}} \tag{10}$$

证毕.

5 多方非平衡隐私集合交集协议

本文通过将 UBZS 与当前通信开销最优的两方 UPSI 协议相结合,构建了一种高效且可扩展的 MUPSI 协议。具体而言,执行 UBZS 协议后,参与方 P_1 和 P_n 分别获得聚合集合 A 和 B。随后,P_n 作为发送方,输入集合 B、P_1 作为接收方,输入集合 A、P_1 和 P_n 调用 $\mathcal{F}_{\mathrm{UPSI}}^{2,N_i}$。最后,参与方 P_1 接收交集集合 $I=A\cap B$。

5.1 协议构造

基于 UBZS 协议构造的 MUPSI 协议 $\prod_{\mathrm{UMPSI}}^{n,N_i}$ 的详细协议如算法 4 所示。具体流程如下:

参数:对于 $i\in[1,n]$ 和 $j\in[N_1,N_i]$,每个参与方 P_i 拥有一个集合 $X_i=\{x_i^j\}$,N_i 表示集合 X_i 的大小,其中 N_n 远远大于 $N_1,N_2,\cdots,N_{n-1}$。

协议流程:

(1) $P_1,P_2,\cdots,P_n$ 调用 $\mathcal{F}_{\mathrm{UBZS}}^{n,N_i}$ 分别输入 $X_1,X_2,\cdots,X_n$。参与方 P_1 和 P_n 分别获得输出 $A=\{a^1,a^2,\cdots,a^{N_1}\}$ 和 $B=\{b^1,b^2,\cdots,b^{N_n}\}$。

(2) P_1 和 P_n 调用 $\mathcal{F}_{\mathrm{UPSI}}^{2,N_i}$,$P_n$ 作为发送方输入集合 B,P_1 作为接收方输入集合 A。参与方 P_1 获得交集 $I=A\cap B$。

算法 4. $\prod_{\text{UMPSI}}^{n,N_i}$ 协议构造算法

输入:对于 $i \in [1,n]$,每个参与方 P_i 输入集合 X_i。

输出:输出交集集合 I 给 P_1。

1. $P_1,P_2,\cdots,P_n$ INVOKE $\mathcal{F}_{\text{UBZS}}^{n,N_i}(X_1,X_2,\cdots,X_n)$

2. P_1 GET $A=\{a^1,a^2,\cdots,a^{N_1}\}$

3. P_n GET $B=\{b^1,b^2,\cdots,b^{N_n}\}$

4. P_1,P_n INVOKE $\mathcal{F}_{\text{UPSI}}^{2,N_i}(A,B)$

5. RETURN $I=A\cap B$

5.2 安全证明

本节分析 MUPSI 协议的正确性和安全性。$\prod_{\text{UMPSI}}^{n,N_i}$ 协议的安全性基于理想功能 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 和 $\mathcal{F}_{\text{UPSI}}^{2,N_i}$ 的安全性。具体证明如下:

定理 2. 在 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 和 $\mathcal{F}_{\text{UPSI}}^{2,N_i}$ 混合模型中,针对 P_1 不参与合谋的半诚实敌手,算法 4 中的 $\prod_{\text{UMPSI}}^{n,N_i}$ 协议是安全地。

证明. 正确性:本文提出的 $\prod_{\text{UMPSI}}^{n,N_i}$ 协议的正确性依赖于 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 和 $\mathcal{F}_{\text{UPSI}}^{2,N_i}$ 的正确性,$\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 的正确性确保了当且仅当存在 $j\in[1,N_1]$ 和 $j'\in[1,N_n]$,使得 $x_n^{j'}=x_1^j$ 且 x_1^j 属于前 $n-1$ 个参与方集合的交集,即 $x_1^j\in X_1\cap X_2\cap\cdots\cap X_{n-1}$ 时,才有 $a^j=b^{j'}$ 成立。$\mathcal{F}_{\text{UPSI}}^{2,N_i}$ 的正确性确保了正确得到交集。

安全性:对于 $|\mathcal{C}|=t$ 和 $|\mathcal{H}|=n-t$,设 $\mathcal{C}$ 和 $\mathcal{H}$ 分别表示合谋参与方和诚实参与方的集合。本文实现最多抵抗有条件的 $n-1$ 参与方的合谋,在 MUPSI 协议中表现为参与方 P_1 不能参与合谋。为了说明如何在理想世界中模拟 $\mathcal{C}$ 的视图,本文根据 P_n 是否参与合谋分为两种情况进行证明。

情形 1:P_n 参与合谋。在这种情况下,$P_1\in\mathcal{H}$ 和 $P_n\in\mathcal{C}$。模拟器 $\text{Sim}_{\text{UMPSI}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)$ 模拟如下:

(1)对于 $j\in[1,N_n]$,模拟器 Sim 随机取样 $b'^j\leftarrow\{0,1\}^\ell$,并构造集合 $B'=\{b'^j\}_{j\in[1,N_n]}$。

(2)模拟器 Sim 调用非平衡双中心零共享协议的模拟器 $\text{Sim}_{\text{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),B')$ 和两方 UPSI 协议的模拟器 $\text{Sim}_{\text{UPSI}}^{P_n}(B',\perp)$,并将两个模拟器的输出添加到 Sim 的视图。

在情形 1 场景下,P_1 不参与合谋,P_n 参与合谋。因此,本文的 $\prod_{\text{UMPSI}}^{n,N_i}$ 协议的安全性依赖于 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 和 $\mathcal{F}_{\text{UPSI}}^{2,N_i}$ 在混合模型中的安全性。在 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 混合模型中,$\text{Sim}_{\text{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),B')$ 的输出与真实世界中敌手的视图在计算意义上不可区分。在 $\mathcal{F}_{\text{UPSI}}^{2,N_i}$ 混合模型中,$\text{Sim}_{\text{UPSI}}^{P_n}(B',\perp)$ 的输出与真实世界中敌手的视图在计算意义上不可区分。因此,对于 $\bar{\boldsymbol{X}}=(X_1,X_2,\cdots,X_n)$,模拟器 Sim 构造的视图与真实世界中敌手的视图在计算意义上不可区分。

$$\{\text{Sim}_{\text{UMPSI}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)\}_{\bar{\boldsymbol{X}}}\overset{c}{\equiv}\{\text{View}_{\mathcal{C}}^{\prod_{\text{UMPSI}}^{n,N_i}}(\bar{\boldsymbol{X}})\}_{\bar{\boldsymbol{X}}}\tag{11}$$

情形 2:P_n 不参与合谋。在这种情况下,$P_1\in\mathcal{H}$ 和 $P_n\in\mathcal{H}$。模拟器 $\text{Sim}_{\text{UMPSI}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)$ 模拟如下:

(1)模拟器 Sim 调用非平衡双中心零共享协议的模拟器 $\text{Sim}_{\text{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)$,并将输出添加到模拟器 Sim 的视图。

在情形 2 场景下,P_1 和 P_n 都不参与合谋。因此,本文的 $\prod_{\text{UMPSI}}^{n,N_i}$ 协议的安全性依赖于 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 在混合模型中的安全性。在 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 混合模型中,$\text{Sim}_{\text{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)$ 的输出与真实世界中敌手的视图在计算意义上不可区分。因此,对于 $\bar{\boldsymbol{X}}=(X_1,X_2,\cdots,X_n)$,模拟器 Sim 构造的视图与真实世界中敌手的视图在计算意义上不可区分。

$$\{\text{Sim}_{\text{UMPSI}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)\}_{\bar{\boldsymbol{X}}}\overset{c}{\equiv}\{\text{View}_{\mathcal{C}}^{\prod_{\text{UMPSI}}^{n,N_i}}(\bar{\boldsymbol{X}})\}_{\bar{\boldsymbol{X}}}\tag{12}$$

证毕.

6 多方非平衡隐私集合交集基数协议

本文对 Mahdavi 等人[14]提出的两方 UPSI 协议进行了修改,构建了一种新的两方 UPSI-CA 协议。在此基础上,结合 UBZS,构建了一种高效且可扩展的 MUPSI-CA 协议。

6.1 两方非平衡隐私集合交集基数协议构造

本节描述了如何通过修改 Mahdavi 等人[14]提出的两方 UPSI 协议,构造一种两方 UPSI-CA 协议,记为 $\prod_{\text{UPSI-CA}}^{2,N_i}$ 详细协议如算法 5 所示。具体流程如下:

参数:接收方 P_1 输入大小为 N_1 的集合 X_1;发送方 P_2 输入大小为 N_2 的集合 X_2,其中 $N_2\gg N_1$。利用抛硬币的方式随机选取 3 个哈希函数 H_1、H_2、$H_3:\{0,1\}^\sigma\to[m]$,其中 $m=(1+\varepsilon)\cdot N_1$ 和 B 分别为哈希函数的存储大小和最大负载。

客户端离线阶段:

(1)发送方 P_2 分别使用 H_1、H_2、H_3 将集合 X_2 插入到哈希表 T_S,其中每个 bin 存储 B 个元素。对于 $i \in [1,m]$、$j \in [1,B]$ 和 $k \in [1,l]$,将哈希表 T_S 中存储的元素进行恒定权重编码,即 $T'_S[i][j] = \text{CW-Encode}(T_S[i][j],l,w)$。对 $T'_S[i][j]$ 表进行批处理 $pt_S[j][k] = [T'_S[1][j][k], T'_S[2][j][k],\cdots, T'_S[m][j][k]]$。

客户端在线阶段:

(1)接收方 P_1 使用布谷鸟哈希将集合 X_1 插入到大小为 m 个 bin 的哈希表 T_R,其中每个 bin 存储 1 个元素。对于 $i \in [1,m]$ 和 $k \in [1,l]$,将哈希表 T_R 中存储的元素进行恒定权重编码,即 $T'_R[i]=\text{CW-Encode}(T_R[i],l,w)$。对 $T'_R[i]$ 进行批处理 $pt_R[k] = [T'_R[1][k], T'_R[2][k],\cdots, T'_R[m][k]]$,对批处理之后的集合进行同态加密 $ct_R[k]=\text{Enc}(pt_R[k],\text{sk}_R)$。

(2) P_1 将密文批处理集合 ct_R 发送给 P_2。对于 $j \in [1,B]$,P_2 调用恒定权重编码的批处理比较协议,即 $ct_{eq}[j] \leftarrow \text{CW-EQ}(ct_R, pt_S[j])$。$P_2$ 将比较结果进行聚合 $c \leftarrow \sum_{i=1}^{m}(\sum_{j=1}^{B} ct_{eq}[j])$,$P_2$ 将密文结果发送给 P_1。

(3) P_1 解密密文 c,获得 $I_{\text{CA}} = | X_1 \cap X_2 |$。

算法 5. $\prod_{\text{UPSI-CA}}^{2,N_i}$ 协议构造算法

输入:参与方 P_1 和 P_2 分别输入集合 X_1 和 X_2。

输出:输出交集基数 $I_{\text{CA}} = | X_1 \cap X_2 |$ 给 P_1。

1. // P_2 将输入集合插入朴素哈希表
2. FOR $i = 1$ to N_2 //客户端离线阶段
3. FOR $j = 1$ to h
4. x_2^i INTO $T_S[H_j(x_2^i)]$
5. END FOR
6. END FOR
7. //对表 T_S 中的元素进行恒定权重编码
8. FOR $i = 1$ to m
9. FOR $j = 1$ to B
10. $T'_S[i][j] = \text{CW-Encode}(T_S[i][j],l,w)$
11. END FOR
12. END FOR
13. //对恒定编码之后的表进行批处理
14. FOR $j = 1$ to B
15. FOR $k = 1$ to l
16. $pt_S[j][k] = [T'_S[1][j][k],\cdots,T'_S[m][j][k]]$
17. END FOR
18. END FOR
19. //客户端在线
20. FOR $i = 1$ to N_1
21. // P_1 将输入集合插入到布谷鸟哈希表
22. x_1^i INTO T_R
23. END FOR
24. // P_1 将布谷鸟哈希表进行恒定权重编码
25. FOR $i = 1$ to m
26. $T'_R[i] = \text{CW-Encode}(T_R[i],l,w)$
27. END FOR
28. // P_1 将编码之后的集合批处理之后进行加密
29. FOR $k = 1$ to l
30. $pt_R[k] = [T'_R[1][k],\cdots,T'_R[m][k]]$
31. $ct_R[k] = \text{FHE.Enc}(pt_R[k],\text{sk}_R)$
32. END FOR
33. // P_2 对发送过来的密文进行同态比较
34. FOR $j = 1$ to B
35. $ct_{eq}[j] \leftarrow \text{CW-EQ}(ct_R, pt_S[j])$
36. END FOR
37. $c = \text{FHE.Enc}(0)$
38. // P_2 将同态比较之后的密文,同态相加
39. FOR $i = 1$ to m
40. FOR $j = 1$ to B
41. $c = c + ct_{eq}[i][j]$ //交集基数的密文
42. END FOR
43. END FOR
44. // P_1 对交集基数密文进行解密,从而获得交集基数
45. RETURN $I_{CA} = \text{FHE.Dec}(c,\text{sk}_R)$

6.2　多方非平衡隐私集合交集基数协议构造

基于 UBZS 协议构造的 MUPSI-CA 协议详细描述见算法 6。其具体流程如下:

参数:对于 $i \in [1,n]$ 和 $j \in [N_1,N_i]$,每个参与方 P_i 拥有一个集合 $X_i = \{x_i^j\}$,N_i 表示集合 X_i 的大小,其中 N_n 远远大于 $N_1, N_2,\cdots, N_{n-1}$。

协议流程:

(1) $P_1,\cdots,P_n$ 调用 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 分别输入 $X_1, X_2,\cdots, X_n$。参与方 P_1 和 P_n 分别获得输出 $A = \{a^1, a^2,\cdots, a^{N_1}\}$ 和 $B = \{b^1, b^2,\cdots, b^{N_n}\}$。

(2) P_n 作为发送方,输入集合 B,P_1 作为接收方,输入集合 A,P_1 和 P_n 调用 $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$。参与方 P_1 获得交集基数 $I_{\text{CA}} = | A \cap B |$。

算法 6. $\prod_{\text{UMPSI-CA}}^{n,N_i}$ 协议构造算法

输入:对于 $i \in [1,n]$,每个参与方 P_i 输入集合 X_i。

输出:输出交集基数 I_{CA} 给 P_1。

1. $P_1,\cdots,P_n$ INVOKE $\mathcal{F}_{\text{UBZS}}^{n,N_i}(X_1, X_2,\cdots, X_n)$
2. P_1 GET $A = \{a^1, a^2,\cdots, a^{N_1}\}$

3. P_n GET $B=\{b^1,b^2,\cdots,b^{N_n}\}$

4. P_1,P_n INVOKE $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}(A,B)$

5. RETURN $I_{\text{CA}}=|A\cap B|$

6.3 安全证明

本节分析了 UPSI-CA 和 MUPSI-CA 协议的正确性和安全性。

6.3.1 两方非平衡隐私集合交集基数安全证明

定理 3. 在 $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$ 混合模型下，$\prod_{\text{UPSI-CA}}^{2,N_i}$ 协议实现了半诚实敌手安全的 $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$ 功能。

证明. 正确性：$\prod_{\text{UPSI-CA}}^{2,N_i}$ 协议的正确性依赖于全同态加密的正确性和哈希映射的成功概率。全同态加密保证了在密文上的计算能够正确反映在明文上的计算结果，而哈希映射的成功概率则决定了是否能够正确计算集合交集的概率。该协议以压倒性的概率 $1-2^{-\lambda}$ 正确计算了哈希映射。因此，$\prod_{\text{UPSI-CA}}^{2,N_i}$ 协议是正确的。

安全性：为了说明如何在理想世界中模拟敌手的视图，本文根据 P_1 和 P_2 是否被腐败分为两种情况进行证明。

情形 1：P_1 被腐败，$\text{Sim}_{\text{UPSI-CA}}^{P_1}(X_1,I_{\text{CA}})$ 旨在模拟诚实参与方 P_2 发送给腐败方 P_1 的视图。模拟过程如下：

(1)模拟器 Sim 从集合 X_1 中随机选取 I_{CA} 个值，并随机取样 N_2-I_{CA} 个随机值，以模拟诚实参与方 P_2 的输入集合 X'_2。

(2)模拟器 Sim 模拟协议客户端离线阶段第 1 步，使用朴素哈希将集合 X'_2 插入到哈希表 $T_{S'}$，其中每个 bin 存储 B 个元素。对于 $i\in[1,m]$、$j\in[1,B]$ 和 $k\in[1,l]$，将哈希表 $T_{S'}$ 中存储的元素进行恒定权重编码，即 $T'_{S'}[i][j]=\text{CW-Encode}(T_{S'}[i][j],l,w)$。对模拟得到的权重表 $T'_{S'}[i][j]$ 进行批处理，得到 $pt_{S'}[j][k]=[T'_{S'}[1][j][k],T'_{S'}[2][j][k],\cdots,T'_{S'}[m][j][k]]$。

(3)模拟器 Sim 接收到腐败方 P_1 发送的密文表 ct_R 后，对于 $j\in[1,B]$，调用恒定权重编码的批处理比较协议，用以模拟真实协议中通过 CW-EQ(ct_R，$pt_{S'}[j]$) 计算得到的比较结果 $ct'_{eq}[j]$。随后，模拟器 Sim 将所有比较结果进行聚合 $c'\leftarrow\sum_{i=1}^{m}(\sum_{j=1}^{B}ct'_{eq}[j])$，并将表 c' 添加到 Sim 的视图。

由于全同态加密的电路隐私性，腐败方 P_1 无法通过所得交集基数，区分出哪个是模拟器模拟的值，哪个是真实协议执行得到的结果值。因此，模拟器 Sim 模拟的视图与真实世界中敌手的视图在计算意义上是不可区分的。

$$\text{Sim}_{\text{UPSI-CA}}^{P_1}(X_1,I_{\text{CA}})\overset{c}{\equiv}\text{View}_{P_1}^{\prod_{\text{UPSI-CA}}^{2,N_i}}(X_1,X_2) \tag{13}$$

情形 2：P_2 被腐败，$\text{Sim}_{\text{UPSI-CA}}^{P_2}(X_2,\perp)$ 旨在模拟腐败发送方 P_2 的视图。模拟过程如下：

(1)模拟器 Sim 使用明文为零的加密密文来模拟协议中客户端在线阶段的第 3 步中发送方 P_2 发送的密文，并将零的密文添加到模拟器 Sim 的视图。

由于全同态加密的 IND-CPA 安全性，对于腐败方 P_2 而言，零的密文与真实元素的密文在计算意义上是不可区分的。因此，模拟器 Sim 模拟的视图与真实世界中敌手的视图在计算意义上不可区分。

$$\text{Sim}_{\text{UPSI-CA}}^{P_2}(X_2,\perp)\overset{c}{\equiv}\text{View}_{P_2}^{\prod_{\text{UPSI-CA}}^{2,N_i}}(X_1,X_2) \tag{14}$$

证毕.

6.3.2 多方非平衡隐私集合交集基数安全证明

定理 4. 在 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 和 $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$ 混合模型中，针对 P_1 不参与合谋的半诚实敌手，算法 6 中的 $\prod_{\text{UMPSI-CA}}^{n,N_i}$ 协议是安全地。

证明. 正确性：本文提出的 $\prod_{\text{UMPSI-CA}}^{n,N_i}$ 协议的正确性依赖于 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 和 $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$ 的正确性。$\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 的正确性确保了当且仅当存在 $j\in[1,N_1]$ 和 $j'\in[1,N_n]$，使得 $x_n^{j'}=x_1^j$ 且 x_1^j 属于前 $n-1$ 个参与方集合的交集，即 $x_1^j\in X_1\cap X_2\cap\cdots\cap X_{n-1}$ 时，才有 $a^j=b^{j'}$ 成立。$\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$ 的正确性确保了协议正确得到交集基数。

安全性：对于 $|\mathcal{C}|=t$ 和 $|\mathcal{H}|=n-t$，设 $\mathcal{C}$ 和 $\mathcal{H}$ 分别表示合谋参与方和诚实参与方的集合。本文只能实现最多抵抗有条件的 $n-1$ 参与方的合谋，在 MUPSI-CA 协议中表现为参与方 P_1 不参与合谋。为了说明如何在理想世界中模拟 $\mathcal{C}$ 的视图，本文根据 P_n 是否参与合谋分为两种情况进行证明。

情形 1：P_n 参与合谋。在这种情况下，$P_1\in\mathcal{H}$ 和 $P_n\in\mathcal{C}$。模拟器 $\text{Sim}_{\text{UMPSI-CA}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)$ 模拟如下：

(1)对于 $j\in[1,N_n]$，模拟器 Sim 随机取样 $b'^j\leftarrow\{0,1\}^{\ell}$，并构造集合 $B'=\{b'^j\}_{j\in[1,N_n]}$。

(2)模拟器 Sim 调用非平衡双中心零共享协议的模拟器 $\text{Sim}_{\text{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),B')$ 和两方 UPSI-CA 协议的模拟器 $\text{Sim}_{\text{UPSI-CA}}^{P_n}(B',\perp)$，并

将两个模拟器的输出添加到 Sim 的视图。

在情形 1 场景下,本文的 $\prod_{\text{UMPSI}}^{n,N_i}$ 协议的安全性依赖于 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 和 $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$ 在混合模型中的安全性。在 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 混合模型中,$\text{Sim}_{\text{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),B')$ 输出的视图与真实世界中敌手的视图在计算意义上不可区分。在 $\mathcal{F}_{\text{UPSI-CA}}^{2,N_i}$ 混合模型中,$\text{Sim}_{\text{UPSI-CA}}^{P_n}(B',\perp)$ 输出的视图与真实世界中敌手的视图在计算意义上不可区分。因此,对于 $\bar{X}=(X_1,\cdots,X_n)$,模拟器 Sim 模拟的视图与真实世界中敌手的视图在计算意义上不可区分。

$$\{\text{Sim}_{\text{UMPSI-CA}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)\}_{\bar{X}} \overset{c}{\equiv} \{\text{View}_{\mathcal{C}}^{\prod_{\text{UMPSI-CA}}^{n,N_i}}(\bar{\boldsymbol{X}})\}_{\bar{X}} \tag{15}$$

情形 2:P_n 不参与合谋。在这种情况下,$P_1\in\mathcal{H}$ 和 $P_n\in\mathcal{H}$。模拟器 $\text{Sim}_{\text{UMPSI-CA}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)$ 模拟如下:

(1) Sim 调用非平衡双中心零共享协议的模拟器 $\text{Sim}_{\text{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)$,并将输出添加到模拟器 Sim 的视图。

在情形 2 场景下,本文的 $\prod_{\text{UMPSI-CA}}^{n,N_i}$ 协议的安全性依赖于 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 在混合模型中的安全性。在 $\mathcal{F}_{\text{UBZS}}^{n,N_i}$ 混合模型中,$\text{Sim}_{\text{UBZS}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)$ 输出的视图与真实世界中敌手的视图在计算意义上不可区分。因此,对于 $\bar{\boldsymbol{X}}=(X_1,X_2,\cdots,X_n)$,模拟器 Sim 模拟的视图与真实世界中敌手的视图在计算意义上不可区分。

$$\{\text{Sim}_{\text{UMPSI-CA}}(\mathcal{C},(X_{i_1},X_{i_2},\cdots,X_{i_t}),\perp)\}_{\bar{X}} \overset{c}{\equiv} \{\text{View}_{\mathcal{C}}^{\prod_{\text{UMPSI-CA}}^{n,N_i}}(\bar{\boldsymbol{X}})\}_{\bar{X}} \tag{16}$$

证毕.

7　性能分析

7.1　UBZS 协议通信与计算复杂度分析

本文使用文献[21]中提出的 OKVS 作为构建模块,实例化本文的 UBZS 协议。在半诚实模型下,对于包含 N_i 个键值对的 OKVS 实例,OKVS 的渐近大小为 $O(\lambda N_i)$。OKVS 编码和每个键解码的渐近计算复杂度分别为 $O(\lambda N_i)$ 和 $O(\lambda)$。伪随机函数 F 的输出长度为 $\ell=\lambda+2\log N_i$。

表 2 和表 3 列出了文献[32]的零共享协议、文献[15]的双中心零共享协议与本文非平衡双中心零共享协议间的通信及计算复杂度对比情况。其中,文献[32]的零共享协议中有两个协议分别为有条件零共享(Conditional Zero-Sharing, CZS)和无条件零共享(Unconditional Zero-Sharing, UCZS)。

7.1.1　通信复杂度

在 UBZS 协议中,对于 $i\in[2,n-1]$,P_n 向 P_i 分发长度为 κ 的伪随机函数的密钥,P_n 渐近通信复杂度为 $O(\kappa n)$。P_1 在 UBZS 阶段不发送信息不计通信复杂度。本文中 P_i 和 P_1 持有小集合且规模近似,涉及集合规模时统一使用 N_1 表示,P_n 持有大集合使用 N_n 表示。P_i 对集合进行 OKVS 编码并发送编码结构给 P_1,P_i 渐近通信复杂度为 $O(\lambda N_1)$。

表 2　UBZS 通信复杂度对比表

协议	P_1	P_i	P_n
CZS[32]	$O(\lambda n N_n)$	$O(\lambda n N_n)$	$O(\lambda n N_n)$
UCZS[32]	$O(\kappa n)$	$O(\kappa(n-i))$	None
BZS[15]	$O(\lambda N_1+n\kappa)$	$O(\lambda N_1)$	None
UBZS	None	$O(\lambda N_1)$	$O(\kappa n)$

注:None 表示没有向其他参与方发送信息。

表 3　UBZS 计算复杂度对比表

协议	P_1	P_i	P_n
CZS[32]	$O(\lambda n N_n)$	$O(\lambda n N_n)$	$O(\lambda n N_n)$
UCZS[32]	$O(\lambda n N_n)$	$O(\lambda n N_n)$	$O(\lambda n N_n)$
BZS[15]	$O(\lambda n N_1)$	$O(\lambda N_1)$	$O(\lambda N_n+\lambda n N_1)$
UBZS	$O(\lambda n N_1)$	$O(\lambda N_1)$	None

注:None 表示没有进行计算。

7.1.2　计算复杂度

在 UBZS 协议中,客户端离线阶段 P_n 使用 $n-2$ 个长度为 κ 的伪随机函数的密钥,分别计算 N_n 个元素的 $n-2$ 个伪随机函数值。因此,P_n 在客户端离线阶段的渐近计算复杂度为 $O(nN_n)$,此复杂度不计入协议在线执行的复杂度。客户端在线阶段,P_n 向其他参与方发送伪随机函数密钥,无需进行计算。对于 $i\in[2,n-1]$,P_i 使用密钥 k_i 计算 N_i 个元素的伪随机函数值,并进行 OKVS 编码,P_i 渐近计算复杂度为 $O(\lambda N_i)$。P_1 接收来自 $P_2,P_3,\cdots,P_{n-1}$ 的 $n-2$ 个 OKVS 结构,P_1 使用接收到的数据结构计算其异或值,得到一个新的数据结构,然后使用 N_1 个元素对新的数据结构进行解码,P_1 渐近计算复杂度为 $O(\lambda N_1)$。

7.2　MUPSI 协议通信与计算复杂度分析

对于 P_1 和 P_n 来说,MUPSI 协议的计算和通信复杂度是 UBZS 协议和两方 UPSI 协议的总和。对于 $P_2,P_3,\cdots,P_{n-1}$,由于不需要额外的操作,MUPSI 协议的计算和通信复杂度等于 UBZS 协议。文献[13]中的 UPSI 协议接收方和发送方的渐近通信复杂度

分别为 $O(N_1)$ 和 $O(N_1)$，渐近计算复杂度分别为 $O(N_1)$ 和 $O(N_n\log N_n)$。MUPSI 协议中的 P_1 和 P_n 在 UPSI 协议中充当接收方和发送方的角色。

综上所述，MUPSI 协议中对于参与方 P_1 的渐近通信和计算复杂度分别为 $O(N_1)$ 和 $O(\lambda n N_1)$，对于 $P_2,P_3,\cdots,P_{n-1}$ 的渐近通信和计算复杂度分别为 $O(\lambda N_1)$ 和 $O(\lambda N_1)$，对于 P_n 的渐近通信和计算复杂度分别为 $O(\lambda n N_1+N_n\log N_n)$ 和 $O(\kappa n+N_1)$。

7.3 MUPSI-CA 协议通信与计算复杂度分析

对于 P_1 和 P_n 来说，MUPSI-CA 协议的计算和通信复杂度是 UBZS 协议和两方 UPSI-CA 协议的总和。对于 $P_2,P_3,\cdots,P_{n-1}$，由于不需要额外的操作，MUPSI-CA 协议的计算和通信复杂度等于 UBZS 协议。本文通过修改 Mahdavi 等人[14]提出的两方 UPSI 协议，构造一种新的两方 UPSI-CA 协议，在 UPSI-CA 协议中接收方和发送方的渐近通信复杂度分别为 $O(N_1)$ 和 $O(1)$，渐近计算复杂度分别为 $O(N_1)$ 和 $O(N_n)$。MUPSI-CA 协议中的 P_1 和 P_n 在两方 UPSI-CA 协议中充当接收方和发送方的角色。

综上所述，MUPSI-CA 协议中对于参与方 P_1 的渐近通信和计算复杂度分别为 $O(N_1)$ 和 $O(\lambda n N_1)$，对于参与方 $P_2,P_3,\cdots,P_{n-1}$ 的渐近通信和计算复杂度均为 $O(\lambda N_1)$，对于参与方 P_n 的渐近通信和计算复杂度分别为 $O(\kappa n)$ 和 $O(\lambda n N_1+N_n)$。

7.4 实验结果与分析

本文的协议使用 C++实现，并在 Ubuntu-20.04 系统上进行性能评估，实验平台配置如下：CPU 为 Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20 GHz，128 GB 内存。统计安全参数 $\lambda=40$，字符串长度 $\sigma=128$ 比特。本次实验利用 Linux 的 tc 命令模拟两种网络连接环境。局域网(LAN)设置中，网络延迟为 0.02 ms，带宽为 10 Gbps；广域网(WAN)设置中，延迟为 40 ms，带宽为 100 Mbps。所有实验结果均为 10 次独立运行的平均值。

7.4.1 非平衡双中心零共享协议性能分析

为了验证 UBZS 协议的可扩展性，本文进行了实验评估，表 4 展示了 Gao 等人[15]提出的 BZS 方案和本文提出的 UBZS 方案的性能对比。测试环境设置为：小集合的规模为 2^{10}，参与方数量为 32。其中，P_1 和 P_n 分别作为 BZS 和 UBZS 中的两个中心参与方，P_n 持有大集合，其余 $P_1,P_2,\cdots,P_{n-2}$ 持有小集合。因 UBZS 和 BZS 协议的中间参与方 $P_{i,i\in[2,n-1]}$ 的计算开销相近，为评估最坏情况下的性能，表中数据取中间参与方的最大的执行时间。UBZS 协议包含客户端不在线的离线时间，这部分未计入协议的时间。P_n 在执行协议时使用 32 个线程，$P_1,P_2,\cdots,P_{n-1}$ 则为 4 线程。实验结果表明，本文提出的 UBZS 协议在持有大集合的参与方在线执行时，相比 Gao 等人[15]的 BZS 协议更具优势，且大集合与小集合的规模差异越大，优势越突出。

表 4 BZS 与 UBZS 对比表

	大集合规模	协议	总时间(s)		
			P_1	P_i	P_n
LAN	2^{20}	BZS[15]	0.002	0.001	0.053
		UBZS	0.002	0.009	**0.001**
	2^{22}	BZS[15]	0.003	0.001	0.176
		UBZS	0.003	0.009	**0.001**
	2^{24}	BZS[15]	0.006	0.001	0.622
		UBZS	0.006	0.009	**0.002**
	2^{27}	BZS[15]	0.007	0.001	4.675
		UBZS	0.007	0.009	**0.004**
WAN	2^{20}	BZS[15]	0.004	0.003	0.059
		UBZS	0.004	0.011	**0.003**
	2^{22}	BZS[15]	0.005	0.003	0.186
		UBZS	0.005	0.011	**0.003**
	2^{24}	BZS[15]	0.008	0.003	0.631
		UBZS	0.008	0.011	**0.004**
	2^{27}	BZS[15]	0.010	0.003	4.682
		UBZS	0.010	0.011	**0.006**

7.4.2 MUPSI 协议的性能分析

本文在实现 MUPSI 协议中的两方 UPSI 时，采用了通信最优的参数配置，以最大程度地减少通信开销，但同时也导致了运行时间略有增加。为了评估本文 MUPSI 协议的性能，将与文献[3]和文献[15]中的协议进行了对比。文献[3]未提供源代码，本次对比基于其公开的实验数据进行。由于实验环境(包括硬件及软件配置等)存在差异，直接对比可能存在偏差。此外，文献[15]缺乏完整的 MUPSI 实现，因此，本文依据其原理分别测试各功能模块的开销，通过累加方式评估整体性能。

表 5 MUPSI 协议时间开销对比

大集合规模	协议	总时间(s)		
		P_1	P_i	P_n
2^{20}	文献[15]	0.671	0.009	0.367
	MUPSI	0.671	0.009	**0.314**
2^{22}	文献[15]	4.237	0.009	8.717
	MUPSI	4.237	0.009	**8.541**
2^{24}	文献[15]	4.488	0.009	9.883
	MUPSI	4.488	0.009	**9.261**
2^{27}	文献[15]	9.789	0.009	25.638
	MUPSI	9.789	0.009	**20.956**

表 5 展示了在 LAN 环境下，当参与方数量为

32 且小集合规模为 2^{10} 时,本文的 MUPSI 协议与文献[15]在不同大集合规模下的对比情况。实验结果表明,本文的 MUPSI 协议性能优于文献[15]的,且随着大集合规模的增大,其优势更为明显。

图 2 展示了服务器端集合规模从 2^{10} 扩展到 2^{14} 时,客户端和服务器的在线运行时间变化。实验结果表明,随着服务器端集合规模的不断增大,本文提出的 MUPSI 协议性能优于文献[3]中的协议。

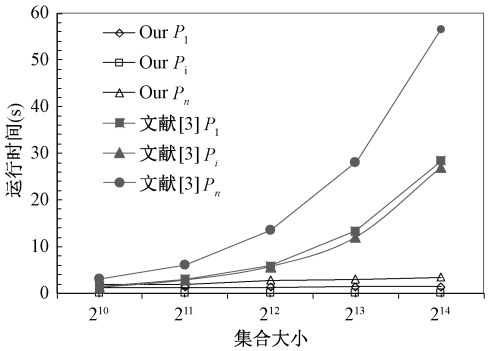

图 2　MUPSI 协议与文献[3]协议的开销对比

表 6 展示了当参与方数量为 32 且小集合规模为 2^{10} 时,本文实现的 MUPSI 协议在不同大集合规模下的运行时间和通信开销。其中,P_1,P_2,…,P_{n-1} 持有小集合为 2^{10},P_n 持有大集合,具体规模在表中列出,以展示其对 MUSPI 协议的影响。对于 $i \in [2,n-1]$,由于 P_i 执行相同的操作,表中 P_i 的时间代表了这些中间参与方的平均运行时间。

根据表 6 和图 3 展示的信息可知,除 P_1 和 P_n 之外的参与方的通信开销和在线运行时间主要取决于其自身集合的大小。在 32 个参与方的实验中,在 LAN 环境下,当大集合的规模为 2^{27} 时,协议完成交集计算仅需约 13 秒。这表明本文提出的 MUPSI 协议具有更强的适应性和扩展性。

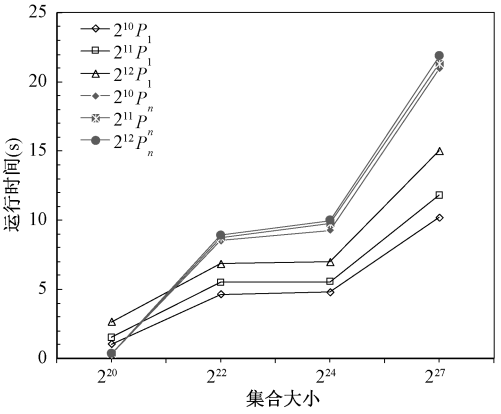

图 3　MUPSI 性能对比

7.4.3　MUPSI-CA 协议的性能分析

表 7 列出了 MUPSI-CA 协议的性能评估数据。实验表明,当参与方数量从 16 扩展到 128 时,总运行时间仅增加了约 1%。例如,在大集合规模为 2^{24} 设置下,16 个参与方运行时间为 120.301 秒,而 128 个参与方运行时间为 120.438 秒。这一结果表明

表 6　MUPSI 协议性能测试表

	大集合规模	在线运行时间(s)			通信开销(MB)		
		P_1	P_i	P_n	P_1	P_i	P_n
LAN	2^{20}	0.671	0.009	0.314	2.123	0.023	1.629
	2^{22}	4.237	0.009	8.541	3.420	0.023	0.216
	2^{24}	4.488	0.009	9.261	3.423	0.023	0.676
	2^{27}	9.789	0.009	20.956	3.423	0.023	4.634
WAN	2^{20}	1.287	0.011	0.414	2.123	0.023	1.629
	2^{22}	5.183	0.011	7.696	3.420	0.023	0.216
	2^{24}	5.571	0.011	8.469	3.423	0.023	0.676
	2^{27}	13.432	0.011	20.156	3.423	0.023	4.634

表 7　MUPSI-CA 协议性能测试表

	大集合规模	各个参与方数量的在线运行时间(s)				通信开销(MB)		
		16	32	64	128	P_1	P_i	P_n
LAN	2^{20}	9.687	9.710	9.749	9.824	28.3902	0.0224	0.1070
	2^{22}	32.726	32.749	32.788	32.863	28.3992	0.0224	0.1070
	2^{24}	120.301	120.324	120.363	120.438	28.3959	0.0224	0.1074
WAN	2^{20}	10.354	10.369	10.374	10.454	28.3902	0.0224	0.1070
	2^{22}	33.795	33.806	33.815	33.983	28.3992	0.0224	0.1070
	2^{24}	122.634	122.651	122.714	122.842	28.3959	0.0224	0.1074

MUPSI-CA 协议在扩展到大规模参与方的场景时具有出色的可扩展性,参与方数量的增加对整体效率影响甚微。未来的工作可以针对两方 UPSI-CA 的计算和通信效率进行优化,从而进一步提升

MUPSI-CA 协议的整体性能。

8 结束语

本文提出了一种非平衡双中心零共享的方法，将多方非平衡计算问题转化为两方非平衡计算，有效降低了通信和计算复杂度。进一步地，将 UBZS 方法与当前通信开销最优的两方 UPSI 协议结合，构建了一种高效且可扩展的 MUPSI 协议。此外，对一种标准 UPSI 协议进行修改，构建了一种两方 UPSI-CA 协议。在此基础上，结合 UBZS 方法，从而构建了一种高效且可扩展的 MUPSI-CA 协议。

目前，本文设计的 MUPSI 协议安全性建立在半诚实模型下，且假设 P_1 不参与合谋。在未来的工作中，我们将研究和设计在恶意模型下能够抵抗任意合谋的 MUPSI 协议。

参 考 文 献

[1] Su Yi-Fei, Huang Cheng-Wei, Zhu Wen-Wei, et al. Multi-party diabetes mellitus risk prediction based on secure federated learning. Biomedical Signal Processing and Control, 2023, 85: 104881

[2] Chatterjee P, Das D, Rawat D B. Digital twin for credit card fraud detection: opportunities, challenges, and fraud detection advancements. Future Generation Computer Systems, 2024, 158: 410-426

[3] Ruan Ou, Yan Chang-Wang, Zhou Jing, et al. A practical multiparty private set intersection protocol based on bloom filters for unbalanced scenarios. Applied Sciences, 2023, 13(24): 13215

[4] Freedman M J, Nissim K, Pinkas B. Efficient private matching and set intersection//Proceedings of the Advances in Cryptology-EUROCRYPT 2004. Interlaken, Switzerland, 2004:1-19

[5] Pinkas B, Schneider T, Zohner M. Scalable private set intersection based on OT extension. ACM Transactions on Privacy and Security, 2018, 21(2): 1-35

[6] Wang Zhu-Sheng, Banawan K, Ulukus S. Multi-party private set intersection: an information-theoretic approach. IEEE Journal on Selected Areas in Information Theory, 2021, 2(1): 366-379

[7] Kerschbaum F. Outsourced private set intersection using homomorphic encryption//Proceedings of the 7th ACM Symposium on Information, Computer and Communications Security. Seoul, Republic of Korea, 2012:85-86

[8] Inbar R, Omri E, Pinkas B. Efficient scalable multiparty private set-intersection via garbled bloom filters//Proceedings of the Security and Cryptography for Networks. Amalfi, Italy, 2018:235-252

[9] Li L, Pal B, Ali J, et al. Protocols for checking compromised credentials//Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. London, UK, 2019:1387-1403

[10] Kales D, Rechberger C, Schneider T, et al. Mobile private contact discovery at scale//Proceedings of the 28th USENIX Security Symposium (USENIX Security 19). Santa Clara, USA, 2019:1447-1464

[11] Yang Xin-Peng, Cai Liang, Wang Ying-Hao, et al. Efficient unbalanced quorum PSI from homomorphic encryption//Proceedings of the 19th ACM Asia Conference on Computer and Communications Security. Singapore, 2024:1003-1016

[12] Angelou N, Benaissa A, Cebere B, et al. Asymmetric private set intersection with applications to contact tracing and private vertical federated machine learning. arXiv preprint arXiv:2011.09350, 2020

[13] Cong K, Moreno R C, Da Gama M B, et al. Labeled PSI from homomorphic encryption with reduced computation and communication//Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. Virtual, 2021:1135-1150

[14] Mahdavi R A, Lukas N, Ebrahimianghazani F, et al. PEPSI: Practically efficient private set intersection in the unbalanced setting//Proceedings of the 33rd USENIX Security Symposium (USENIX Security 24). Philadelphia, USA, 2024:6453-6470

[15] Gao Ying, Luo Yuan-Chao, Wang Long-Xin, et al. Efficient scalable multi-party private set intersection(−variants) from bicentric zero-sharing//Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security. Salt Lake City, USA, 2024:4137-4151

[16] Aranha D F, Lin C, Orlandi C, et al. Laconic private set-intersection from pairings//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. Los Angeles, USA, 2022:111-124

[17] Zhao Quan-Yu, Jiang Bing-Bing, Zhang Yuan, et al. Unbalanced private set intersection with linear communication complexity. Science China Information Sciences, 2024, 67(3): 132105

[18] Chen Hao, Huang Zhi-Cong, Laine K, et al. Labeled PSI from fully homomorphic encryption with malicious security//Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security. Toronto, Canada, 2018:1223-1237

[19] Chen Hao, Laine K, Rindal P. Fast private set intersection from homomorphic encryption//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA, 2017:1243-1255

[20] Yang Jia-Hui, Chen Lan-Xiang, Mu Yi, et al. PSI Protocol with structured encryption. Chinese Journal of Computers, 2022, 45(12):2652-2666 (in Chinese)
(杨佳辉，陈兰香，穆怡，等. 结构化加密的 PSI 协议. 计算

机学报,2022,45(12):2652-2666)

[21] 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. Los Angeles, USA, 2022:2505-2517

[22] Dou Jia-Wei, Liu Xu-Hong, Wang Wen-Li. Privacy preserving two-party rational set computation. Chinese Journal of Computers, 2020, 43(08):1397-1413 (in Chinese)
(窦家维,刘旭红,王文丽. 有理数域上两方集合的高效保密计算. 计算机学报,2020,43(08):1397-1413)

[23] Rindal P, Schoppmann P. VOLE-PSI: fast OPRF and circuit-PSI from vector-OLE//Proceedings of the Advances in Cryptology - EUROCRYPT 2021. Zagreb, Croatia, 2021: 901-930

[24] Gong Lin-Ming, Wang Dao-Shun, Liu Mo-Meng, et al. 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,等. 基于无匹配差错的 PSI 计算. 计算机学报,2020,43(09):1769-1790)

[25] Garimella G, Pinkas B, Rosulek M, et al. Oblivious key-value stores and amplification for private set intersection//Proceedings of the Advances in Cryptology-CRYPTO 2021. Virtual, 2021:395-425

[26] Pinkas B, Rosulek M, Trieu N, et al. PSI from PaXoS: Fast, malicious private set intersection//Proceedings of the Advances in Cryptology—EUROCRYPT 2020. Zagreb, Croatia, 2020:739-767

[27] 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

[28] Pinkas B, Schneider T, Zohner M. Faster private set intersection based on OT extension//Proceedings of the 23rd USENIX Conference on Security Symposium. New York, USA, 2014:797-812

[29] Ben-Efraim A, Nissenbaum O, Omri E, et al. PSImple: Practical multiparty maliciously-secure private set intersection//Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security. Nagasaki, Japan, 2022:1098-1112

[30] Bay A, Erkin Z, Hoepman J H, et al. Practical multi-party private set intersection protocols. IEEE Transactions on Information Forensics and Security, 2022, 17: 1-15

[31] 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. Virtua, 2021:1151-1165

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

[33] Zhang En, Qin Lei-Yong, Yang Ren-Lin, et al. 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,等. 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)

[34] Wei Li-Fei, Liu Ji-Hai, Zhang Lei, et al. Two cloud-assisted over-threshold multi-party private set intersection calculation protocol. Journal of Software, 2023, 34(11):5442-5456 (in Chinese)
(魏立斐,刘纪海,张蕾,等. 双云辅助的超阈值多方隐私集合交集计算协议. 软件学报,2023,34(11):5442-5456)

[35] Zhang En, Liu Feng-Hao, Lai Qiqi, et al. Efficient multi-party private set intersection against malicious adversaries//Proceedings of the Proceedings of the 2019 ACM SIGSAC Conference on Cloud Computing Security Workshop. 2019: 93-104

[36] Lai P K Y, Yiu Siu-Ming, Chow K P, et al. An efficient bloom filter based solution for multiparty private matching//Proceedings of the Security and Management, 2006:286-292

[37] Miyaji A, Nakasho K, Nishida S. Privacy-preserving integration of medical data: a practical multiparty private set intersection. Journal of Medical Systems, 2017, 41(3): 1-10

[38] Miyaji A, Nishida S. A scalable multiparty private set intersection//Proceedings of the Network and System Security. Cham, Switzerland, 2015:376-385

[39] 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

[40] Rindal P, Rosulek M. Improved private set intersection against malicious adversaries//Proceedings of the Advances in Cryptology—EUROCRYPT 2017. Paris, France, 2017:235-259

[41] Armknecht F, Boyd C, Carr C, et al. A guide to fully homomorphic encryption. Cryptology ePrint Archive, 2015

[42] Fotakis D, Pagh R, Sanders P, et al. Space efficient hash tables with worst case constant access time. Theory of Computing Systems, 2005, 38(2): 229-248

[43] Devroye L, Morin P. Cuckoo hashing: further analysis. Information Processing Letters, 2003, 86(4): 215-219

[44] Mahdavi R A, Kerschbaum F. Constant-weight PIR: Single-round keyword PIR via constant-weight equality operators//Proceedings of the 31st USENIX Security Symposium (USENIX Security 22). Boston, USA, 2022:1723-1740

[45] Goldreich O. Foundations of Cryptography. Volume 2. Cambridge,USA: Cambridge University Press, 2004

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

LI Jin-Lei, M. S. candidate. His main research interests include cryptographic protocol design and secure multi-party computation.

ZHENG Dong, Ph. D., professor. His main research interests include cryptography theory and cloud computing security.

YU Yong, Ph. D., professor. His main research interests include asymmetric cryptography theory and applications, artificial intelligence security and blockchain security.

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

Background

In the field of secure multi-party computation, the Unbalanced Private Set Intersection (UPSI) protocol, as a core technical paradigm supporting efficient intersection computation of distributed datasets, can effectively address the bottleneck problem of computational efficiency caused by uneven distribution of data resources among participants while ensuring the privacy and security of the participants. Although a series of theoretical breakthroughs have been achieved internationally in this area, the existing UPSI protocols are mostly focused on the scenario with two participants, and there is still no efficient solution for the scenario with multiple participants.

This paper proposes an Unbalanced Bicentric Zero-Sharing (UBZS) protocol under the semi-honest model, which transforms multi-party unbalanced computing problems into two-party unbalanced computing problems. Based on this architecture, this study designs and implements UMPSI and its variant protocols. Experimental evaluation results show that the proposed protocol outperforms existing methods in key indicators such as communication overhead, computational delay, and scalability.

This research has been supported by the National Cryptography Science Foundation (2025NCSF02025), the Key Project of Joint Fund of the National Natural Science Foundation of China (U24B20149, U23A20302), the National Natural Science Foundation of China (No. 62372157), and the Key Industry Innovation Chain Project of Shaanxi Provincial Key Research and Development Program (2024GX-ZDCYL-01-09). Before this work, our team had been continuously exploring in the field of cryptographic protocol design, network security, and blockchain. These findings have been published in various prestigious journals and conferences, such as ACM CCS, IEEE Transactions on Vehicular Technology, Information Sciences, IET Information Security, and Chinese Journal of Electronics.