元组集合子集的保密计算

李顺东,杜佶欣,余佳桐,吴川宇
(陕西师范大学计算机科学学院,陕西西安 710062)

摘 要: 安全多方计算是现代密码学的一个重要研究分支,它能够有效保护数据,防止隐私数据被不当获取或利用,同时确保参与者在共享数据时仍能保持数据隐私和完整性. 其中,集合子集的保密计算是支撑保密数据查询、保密数据外包、相似文档检索以及其他隐私数据安全共享的关键技术. 现有方案主要聚焦于单一元素集合的子集判定,对元组集合缺乏有效支持,且在实用性、安全性与效率层面存在以下挑战:需对元组集合进行两次独立子集判定,导致计算效率低下,且中间结果可能暴露非子集关系的敏感数据;难以有效保护单一元素集合的隐私(尤其在需要保护集合交集与势的场景下),而元组集合所需保护的信息量更大,隐私泄露风险显著加剧;现有单一元素子集协议可能存在误判;同时,现有方案缺乏对某一参与方持有多个元组集合的高效批量判定. 针对上述挑战,本文首次提出某个参与方有多个集合,且集合元素是元组情况下的集合子集的保密计算协议,针对参与方有无全集设计了不同的协议. 所提协议通过单次执行即可同步判定一个元组集合是否为其他多个元组集合的子集,避免分步计算导致的中间结果泄露风险. 本文协议可显著提升效率,并具备广泛的适用性,同时本文所提协议不仅保护了参与双方元组集合的势,也保护了元组集合子集的势与具体元素. 在解决有全集的两方计算时,Alice 只需从 Bob 发送的加密数据中进行选择,避免了复杂的模指数运算,从而降低了计算成本;在解决无全集的两方计算时,结合集合的多项式表示,Bob 只需将自己的数据代入 Alice 发送的加密多项式中,即可计算集合的子集. 最后,本文使用公认的模拟范式证明了两个协议是安全的,且实验表明了方案是可行的.

关键词: 密码学;安全多方计算;同态加密;加密选择;多项式
基金项目: 国家重点研发计划(No.2022YFB2703001)
中图分类号: TP309 **文献标识码:** A **文章编号:** 0372-2112(2025)08-2750-14
电子学报 URL:http://www.ejournal.org.cn **DOI:**10.12263/DZXB.20250389

Secure Computation of Subsets from Tuple Sets

LI Shun-dong, DU Ji-xin, YU Jia-tong, WU Chuan-yu
(*School of Computer Science*, *Shaanxi Normal University*, *Xi'an*, *Shaanxi* 710062, *China*)

Abstract: Secure multi-party computation is an important research branch of modern cryptography. It can effectively protect data, prevent privacy data from being improperly acquired or exploited, and simultaneously ensure that participants maintain data privacy and integrity while sharing data. Among its applications, secure computation for set subset relationships is a key technology underpinning private data queries, confidential data outsourcing, similar document retrieval, and other secure sharing of private data. Existing schemes primarily focus on subset determination for sets composed of single elements and lack effective support for complex sets composed of tuples. Furthermore, they face the following challenges in terms of practicality, security, and efficiency: existing schemes require performing two independent subset determinations on the tuple set, resulting in low computational efficiency, and intermediate results may expose sensitive data unrelated to the subset relationship; Existing schemes struggle to effectively protect the privacy of single-element sets (especially in scenarios requiring protection of set intersections and cardinalities), while the amount of information requiring protection in tuple sets is larger, significantly exacerbating privacy leakage risks; Existing subset protocols for single-element sets may yield erroneous judgments; Simultaneously, existing schemes lack support for efficient batch determination when one participant holds multiple tuple sets. To address the above challenges, this paper proposes, for the first time, secure computation protocols for set subsets where one participant holds multiple sets and the set elements are tuples, designing distinct schemes for scenarios where participants possess or lack a universal set. The proposed protocols enable the synchronous de-

收稿日期:2025-05-17;录用日期:2025-07-29;责任编辑:覃怀银

termination, through a single execution, of whether one tuple set is a subset of multiple other tuple sets, avoiding the privacy leakage risk of intermediate results inherent in stepwise computation. The protocols in this paper significantly enhance efficiency and possess broad applicability. Furthermore, the protocols proposed in this paper not only protect the cardinality of the tuple sets held by both participating parties but also protect the cardinality and specific elements of the tuple subset itself. Specifically, for two-party computations with a universal set, Alice selects from encrypted data sent by Bob, thus avoiding complex modular exponentiation and reducing computational costs. For scenarios without a universal set, using polynomial representations of sets, Bob simply substitutes his data into the encrypted polynomial sent by Alice to compute subset confidentiality. Finally, using established simulation paradigms, this paper proves the security of the protocols, with experimental validation demonstrating the feasibility of the approaches.

Key words: cryptography; secure multi-party computation; homomorphic encryption; encryption selection; polynomial

Foundation Item(s): National Key Research and Development Program of China (No.2022YFB2703001)

1 引言

随着网络技术的发展和数字化进程的加速,用户的个人信息和企业数据越来越容易被泄露或被非法访问. 如何在数字化时代解决多个数据拥有者之间的隐私合作计算问题,成为密码学界研究的焦点. 安全多方计算能够有效保护数据,防止数据被不当获取或利用,同时确保参与者在共享数据时仍能保持数据隐私和完整性.

安全多方计算[1~5]是指在无可信第三方情况下,多个参与方协同完成计算任务,并保证每个参与者除计算结果外不能得到其他参与者输入的任何信息. 安全多方计算的概念于1982年由姚期智教授提出,以解决百万富翁问题. 百万富翁问题涉及两个富人如何在没有可信第三方的情况下比较他们的财富,而不泄露具体财产状况. 这种场景引发了安全多方计算这一理论的初步探索. 安全多方计算的基本目标是允许多个参与者在进行合作计算时保持私密数据的隐私性,同时确保计算的正确性和安全性. Goldwasser[6]预言安全多方计算将成为计算科学中一个必不可少的组成部分. 随后,Ben-Or等人[7]进一步发展了安全多方计算理论,使其逐渐成为现代密码学的一个重要组成部分.

集合论是现代数学的基础,在其他学科中承担着基础性的作用. 由于大量的实际问题能够抽象为集合问题,集合论中的数学思想被应用到越来越多的领域. 集合的保密计算是安全多方计算领域研究的热点问题之一. 其中,集合子集的保密计算问题是保密科学计算研究的重要内容之一,它可以被描述为:给定多个参与者各自持有的集合,如何计算这些集合的子集,而不泄露集合的私密信息.

集合子集问题在社交网络分析、保密数据挖掘[8]、外包计算[9]、接触者追踪[10]等研究中有现实意义和应用价值. 在这些应用中,多个数据拥有者通常希望合作解决问题,但又不愿意直接共享其数据,因为数据可能包含敏感信息或私密内容. 安全多方计算为这个问题提供了解决方案,允许参与者在保护隐私的前提下进行合作计算.

现有的集合子集保密计算问题的解决方案主要集中在处理单一元素组成的集合. 保密计算集合子集问题是保密计算集合交集问题的特例,因此现有研究集合子集的保密计算问题都基于对集合交集问题的研究. 最著名的方案是Freedman等人[11]构造的关于交集保密计算问题的协议,Kissner等人[12]在探索Freedman交集方案的基础上给出了多方并集保密计算协议. 该协议中利用了集合的多项式表示方法,一个集合被表示成一个多项式的根,这样多个多项式乘积的根相当于多个集合的并集,所以并集计算问题就转化为先计算多项式的乘积,再计算这个乘积的根的问题.

文献[13]在给定全集的情况下,应用矩阵表示多重集,结合同态加密方案,设计了两方整数多重集的交集或并集保密计算协议. 文献[14]高效解决了在有全集的情况下n个参与者和Alice想要保密判断Alice的集合是否包含n个集合的交集问题. 文献[15]在有全集的情况下,结合加密算法的同态性,将集合的安全多方计算问题转化为向量的安全计算问题. 文献[16]将有理数域上元素与集合关系问题转化为整数范围内向量内积问题,进一步结合Paillier加密方案设计了集合运算的保密计算协议. 文献[17]提出并解决了根据空间众包中的工人兴趣为其分配任务的问题,设计了基于分组的模糊距离收集方法. 但上述文献均只关注单一元素组成的集合,且现有的针对单一元素组成的集合的方案,难以保护参与方集合的势[18,19]或计算结果存在误差[20].

文献[21~24]虽未直接研究集合子集保密计算问题,但在安全多方计算领域的技术方案为本研究提供了重要参考. 文献[21]基于Paillier加密方案设计了“点与区间”及“区间与区间”关系的两方保密计算协议,其核心思想是通过同态加密实现数值型数据的隐

私关系判定. 文献[22]针对现有的轨迹隐私保护模型大多难以抵御复杂背景知识攻击的问题,提出了一种基于差分隐私的轨迹隐私保护方法,既能有效保护轨迹数据中用户的隐私,也能保证数据的可用性. 文献[23]为了解决基于位置的服务(Location Based Service,LBS)在收集用户位置数据时造成的隐私泄露,提出一种本地化差分隐私位置发布模型. 文献[24]提出一种基于本地差分隐私(Local Differential Privacy,LDP)的面向离群点的真值发现算法,首先对工人提交的数据添加拉普拉斯噪声,以确保工人隐私不被泄露,然后细化工人和任务重要性权重分配,并根据提交值之间的相似性对工人进行分组,从而在有效保护工人隐私的同时提高估计"真值"的精度.

上述方案[21~24]虽在特定场景(数值比较/相等判定)中有效,但难以迁移至元组集合子集保密计算场景,主要原因在于元组集合子集的保密计算问题涉及元组间复杂的多维度关系,而现有方案仅处理标量或简单区间,且文献[21~24]使用基于噪声的隐私保护机制,计算结果可能存在误差.

由单一元素组成的集合只包含一个维度,无法提供足够的数据维度来描述复杂的现象或支持多维度的数据分析,无法展示数据的多样性和复杂性,难以描述数据之间复杂的关系,从而导致其在更复杂的应用场景中受到限制. 在车联网(IoV)和物联网(IoT)中,设备间的数据交互需同时满足高效性、隐私保护和严格的身份-状态绑定验证. 例如,车辆在遭遇事故或故障时,需向周围车辆及路侧单元(RSU)广播紧急信号,广播格式为"车辆ID,实时状态码". 若只广播车辆ID,则恶意车辆可能伪造高优先级信号(如虚假事故警报)引发交通混乱甚至事故. 若只广播实时状态码,则无法定位是哪一辆车需要救援. 智能家居设备(如智能门锁)需定期接收固件更新,更新请求格式为"设备序列号,固件版本哈希",厂商需保密合法设备的序列号与版本映射关系,避免合法设备被降级到漏洞版本. 单一元素组成的集合无法同时满足两个属性的判定. 因此,这种单一元素集合无法支持多维度的安全分析,在安全多方计算的复杂场景中,需要考虑多个数据维度之间的复杂关系,而单一元素集合的局限性使其难以胜任这些任务.

元组集合所包含的信息往往不是单一标识,而是由多个属性组合而成. 元组集合的子集判断,能够在不泄露完整数据内容情况下,同步进行多维度信息匹配. 与单一元素集合子集判断不同,该判定要求元组集合中的每个元素必须同时满足所有的严格等价关系,即只有当元组的每个属性值均完全匹配时,才判定为子集关系. 这样的集合关系判断增加了元素判定的严格度、精确度更高,在电商、供应链、金融、医疗等领域具有广泛应用. 例如,在供应链管理和物流追踪中,货物信息由产品ID、批次号组成,管理者需要判断某个批次清单是否为订单中的一部分,用于追踪与溯源. 在医疗数据共享与隐私保护中,患者记录由患者ID、时间戳组成,判断外部数据集是否为内部已知数据的子集,保证只有匹配且经过授权的记录被访问或共享. 在身份认证与访问控制中,用户权限由用户ID、角色组成,判断查询权限集合是否为用户已有权限集合的子集,实现精准访问控制.

此外,已有的协议针对参与方各有一个集合的计算问题,对某一参与方拥有多个集合的保密计算研究较少. 例如,网络安全中心持有一个大型恶意代码集合数据库,记录了各种已知病毒、木马、僵尸网络等样本的特征码,而个人或企业在日常网络运营过程中,也会从入侵检测等途径获得一些可疑的程序样本集合. 但同时,个人或企业又不希望将获得的完整代码明文透露给第三方. 另一方面,安全中心出于保护知识产权的考虑,也不愿公开其全部恶意代码特征库的内容. 在这种场景下,双方都有保护自身数据隐私的需求,通过对集合子集问题的保密计算,可以得知程序中是否存在已知恶意代码集的某些片段,从而指导后续处理.

目前关于单一元素组成集合的子集关系的研究逐渐增多,但元组集合子集关系的判定仍然是一个未被充分探索的领域,我们以列表的方式与已有工作[14,20]和文献[25]进行了对比分析,如表1所示.

表1 相关方案特性对比分析

协议	元组集合	交集的势	交集元素	准确性	中间结果
文献[14]协议5	×	√	√	√	×
文献[20]协议3.1.2	×	√	√	√	×
文献[20]协议3.2.2	×	√	√	×	×
文献[20]协议4.2.1	×	×	×	×	×
文献[20]协议4.2.2	×	×	×	√	×
文献[25]	×	×	×	×	×
本文协议1	√	√	√	√	√
本文协议2	√	√	√	√	√

注:元组集合:"√"表示参与计算的是元组集合,"×"表示参与计算的是单一元素集合;交集的势、交集元素与中间结果:"√"表示无隐私泄露,"×"表示存在隐私泄露;准确性:"√"表示计算结果准确,"×"表示不准确.

本文分别在有全集和无全集的两种情形下提出了元组集合子集的保密计算协议. 本文协议主要解决了以下4个挑战.

(1)元组集合子集关系判定:目前的方案主要适用于单一元素集合的子集关系判定,而要实现元组集合

子集的判定,通常需要在两个独立的数据集上分别执行两次子集关系判定. 这种方式不可避免地导致查询效率低下,并且在组合使用协议时,可能泄露不满足子集关系的部分数据,即中间计算结果可能带来隐私泄露问题.

(2)增强的元组集合隐私保护:现有方案往往难以有效保护参与方单一元素集合的隐私,尤其是在处理集合交集及其势时,而元组集合所需保护的信息量更大.

(3)准确子集关系判定:当前的单一元素集合子集关系判定协议可能存在误差.

(4)支持一方多集合的高效批量判定:针对某一参与方拥有多个元组集合这一实际应用场景,现有方案缺乏有效支持.

针对以上问题,本文提出了两个能够解决元组集合子集的保密计算协议,这两个协议能够保护每个参与方集合的势,计算结果准确且应用范围更广,适用于某一参与方拥有多个集合的情形,同时通过模拟范例对协议的安全性进行了严格证明.

本文提出的两个不同协议,分别针对集合是否具有全集的情形,解决了元组集合的保密计算问题. 尽管本文的协议主要适用于两方计算,但易于扩展至多方计算,具体贡献如下.

(1)提出并解决了元组集合子集的保密计算问题. 本文协议能够通过一次执行同步判定一个元组集合是否为其他多个元组集合的子集,无需在两个独立的数据集上进行分步计算,消除中间计算结果带来的隐私泄露问题,具备较广泛的适用性.

(2)本文提出的方案不仅保护参与双方元组集合的势,还保护了元组集合子集的势与具体元素,具有更高的安全性.

(3)本文提出并解决了在某一参与方拥有多个元组集合的情况下,如何高效且准确地进行元组集合子集关系的保密计算问题,在有全集和无全集的两种情形下,设计了不同的解决方案. 在有全集的两方计算中,Alice仅需从Bob发送的加密数据中选择,无需进行复杂的模指数运算;而在无全集的两方计算中,结合集合的多项式表示,Bob只需将自己的数据代入Alice发送的加密多项式中,即可完成子集关系的计算. 此外,本文的方案可扩展至多方保密计算.

(4)本文通过模拟范例对协议的安全性进行了严格证明,并进行了效率实验,实验结果表明这两个协议是可行的.

2　相关工作

隐私信息检索(Private Information Retrieval, PIR)和隐私集合求交(Private Set Intersection, PSI)问题均用于保护参与方的数据隐私. PIR侧重于隐私检索,保护客户端的查询隐私(即服务器不知道客户端检索了哪个数据项),而PSI侧重于隐私比较,判断集合间的包含关系,保护集合元素的隐私(即除了交集结果外,各方不会泄露自己集合中的其他元素).

PIR的核心是隐私检索单条数据(例如从数据库获取第i项),而子集判定是验证整个集合的包含关系(需检查所有元素是否被包含),要用PIR解决子集判定问题,需要对查询集A中的每个元素执行一次PIR(检查其是否在目标集B中). 尽管这种方法会泄露$|A|$,且效率极低(需要$O(|A|$次PIR调用),但它仍是一种可行的解决集合子集的保密判定问题的方式.

同时,本文研究的问题可以看作是PSI的一个特例. 具体来说,当判断集合A是不是集合B的子集时,可以通过求集合A和集合B的交集来实现. 如果交集结果与集合A相等,那么A就是B的子集. 因此,PSI在某些情境下可以被用来解决集合子集的保密判定问题.

2.1　隐私信息检索

隐私信息检索是信息安全和隐私保护中的一个基本问题,其目标是设计一种检索机制,使用户能够从存储在单个或多个远程服务器上的消息数据集中检索一条消息,并且不泄露用户的检索信息. PIR方案可以被简单概括为:

(1)设数据库D由N个索引i和它对应数据项d_i组成,即$D=\{(1,d_1),(2,d_2),\cdots,(N,d_N)\}$. 用户希望根据自己已知的索引$i$检索数据库中对应的数据项$d_i$,且不泄露索引$i$. 用户对检索索引进行了隐藏处理,向数据库服务器发送处理过的索引.

(2)数据库服务器接收到经过隐藏处理的索引信息,执行相应的检索操作,并返回与查询匹配的数据. 由于索引信息经过处理,服务器无法得知用户检索的是哪个数据项d_i,也无法识别数据项d_i的具体内容.

(3)用户对返回的数据进行进一步处理,获取相应的数据项d_i作为检索结果.

本文研究的问题是参与方Alice拥有n个集合$M_i(i=1,2,\cdots,n)$,每个集合分别拥有元素$M_i=\left\{\left(x_{i1},w_{i1}\right),\left(x_{i2},w_{i2}\right),\cdots,\left(x_{ik_i},w_{ik_i}\right)\right\}$,另一参与方Bob拥有集合$Y=\left\{\left(y_1,z_1\right),\left(y_2,z_2\right),\cdots,\left(y_s,z_s\right)\right\}$. 双方保密计算集合$M_i$和$Y$的子集关系,并在$Y$是$M_i$的子集时返回1,其他情况返回0. 其中,$\{x_{ij}\}(i=1,2,\cdots,n;j=1,2,\cdots,k_i)$,$\{y_j\}(j=1,2,\cdots,S)$为关键字集合,$\{w_{ij}\}(i=1,2,\cdots,n;j=1,2,\cdots,k_i)$,$\{z_j\}(j=1,2,\cdots,S)$为关联值集合. 具体要求是:计算双方无法得知除子集关系外的任何信息,且任

意一方仅知道关键字或关联值时，无法得知双方集合的子集关系，也无法获取对应的关联值或关键字.

本文研究的问题与PIR问题是不同的，PIR方案无法解决本文研究的问题，本文提出的方案也无法解决PIR问题.

本文对近年来提出的许多PIR改进和优化方案也进行了研究，包括在多服务器环境中检索单条消息[26~29]、多条消息[30]、在单服务器环境中检索单条消息[31]和多条消息[32,33]. 现有PIR方案[26~33]均无法解决本文提出的问题.

2.2 隐私集合求交

隐私集合求交是安全多方计算中的一种密码学技术，它允许参与计算的双方，在不获取对方额外信息（除交集外的其他信息）的基础上，计算出双方数据的交集. 隐私集合求交在数据共享、广告转化率、联系人发现等领域有着广泛的应用空间. PSI问题可以被简单概括如下.

（1）输入：参与方P_1持有集合$S_1=\{x_1,x_2,\cdots,x_m\}$，参与方$P_2$持有集合$S_2=\{y_1,y_2,\cdots,y_n\}$.

（2）输出：双方（或指定一方）获得交集$I=S_1\cap S_2$.

文献[34]设计了一个新的两方计算协议，即安全秘密共享检索（S^3R），在客户端不泄露谓词和额外数据的情况下，获得满足给定谓词的交集，但是该方案对服务器泄露了集合交集的势. 文献[35]提出了一种结合了特征检索的私有集合交集协议，防止输入集合的势与集合交集的势的泄露，从而增强了隐私保护. 文献[36]证明了PSI问题可以转化为一个多消息对称的私有信息检索（MM-SPIR）问题，并给出了相应的限制条件，但是该方案需在多个不合谋的数据库上实现，并且泄露了交集的势. 文献[37]通过将数据集中的数据区分为匹配属性和分析属性，并对加密后的匹配属性进行比较，在保持隐私的情况下计算匹配属性的交集，然后对分析属性进行整合，实现数据集成，但该方案对云服务器和每一个参与方都泄露了交集的势，并且没有保护参与方$P_i(i=1,2,\cdots,n)$分析属性的隐私性. 文献[38]提出了一种基于伪随机函数和不经意键值存储的高效的两方外包隐私集合交集基数计算协议，旨在计算集合交集的势. 文献[39]针对多方集合运算下特有的隐私泄露问题进行改进，提出了一种改进的基于全集编码的多方隐私集合求交集协议，但无法保护交集元素和交集的势的隐私. 文献[40]提出了一个基于云辅助的两方集合交集协议ED-PSI，无法保护交集元素和交集的势的隐私. 文献[41]提出了一个高效的超阈值多方隐私集合运算协议，在双云服务器环境下实现，但无法保护集合交集的势. 文献[42]提出一种基于隐私交集基数的拼车方案，该方案充分利用了布隆过滤器的特性，通过将集合元素映射到固定大小的数组上，有效减小了元素比较的次数，但计算结果可能存在误差. 文献[43]设计了支持任意类型计算的安全两方隐私保护集合交集方案，但计算结果也可能存在误差. 同时，文献[40~43]都在云服务器环境下进行计算，引入了额外的服务器开销.

上述方案仅解决单一元素集合的隐私求交问题，且都泄露了交集的势或其他隐私信息，不满足本文应用场景的安全性需求，亦无法解决本文提出的问题.

进一步，本文研究了PSI问题的扩展，即私有交集求和问题（Private Set Intersection with Sum）和带标签的PSI问题（Labeled Private Set Intersection）.

PSI-SUM是隐私集合求交（PSI）的扩展协议[44,45]，允许两方在计算集合交集的同时，对交集元素关联的数值属性进行求和，且不泄露交集外的元素及其关联值. PSI-SUM问题可以被简单概括如下.

（1）输入：参与方P_1持有集合

$$S_1=\{(x_1,v_1),(x_2,v_2),\cdots,(x_m,v_m)\}$$

其中，x_i为属性，v_i为对应数值. 参与方P_2持有集合$S_2=\{y_1,y_2,\cdots,y_n\}$.

（2）输出：P_1和P_2共同获得交集$I=\{x_i|x_i\in S_1\cap S_2\}$，并计算$\mathrm{SUM}=\sum_{(x_i,v_i)\in S_1,\ x_i\in I}v_i$或单向输出：仅$P_2$获得SUM.

Labeled PSI是PSI的扩展协议[46,47]，允许一方在计算集合交集后，获取对方集合中交集元素关联的标签或附加数据，同时保护非交集元素的标签隐私. Labeled PSI问题可以被简单概括为：

（1）输入：参与方P_1持有集合

$$S_1=\{(x_1,l_1),(x_2,l_2),\cdots,(x_m,l_m)\}$$

其中，l_i为标签. 参与方P_2持有集合$S_2=\{y_1,y_2,\cdots,y_n\}$.

（2）输出：P_2获得交集$I=\{x_i|x_i\in S_1\cap S_2\}$，$P_1$无输出（或仅确认交集存在）.

上述方案仅将集合视为单一元素集合，是在单一元素集合求交问题上进行扩展，仍无法解决本文提出的问题.

我们对与本文研究最接近的文献[14,20]进行了具体分析.

文献[14]提出并解决了3个新问题：集合交（并）集的势与阈值关系的保密判定、元素与集合交（并）集关系的保密判定、集合与集合交（并）集关系的保密判定. 对与本文研究最接近的文献[14]协议5进行深入分析. 文献[14]协议5仅研究在有全集的情形下的单一元素集合交集关系的保密判定，若应用于元组集合子集的保密计算，则必须在两个独立的数据集上分别执行两

次子集关系判定,这样的计算会泄露集合的大量信息.

文献[20]在第3节给出了集合取自已知全集(协议3.1.2)以及集合取自较大全集(协议3.2.2)的子集保密计算协议,这两个协议都针对单一元素的集合设计,未考虑如何对元组类型的集合进行编码加密,也没有考虑其中一个参与方拥有多个集合的情形. 尽管文献[20]的协议3.2.2能解决集合取自较大全集的子集保密计算问题,但该协议可能导致误判. 文献[20]的协议4.2.1和协议4.2.2研究的是多方交集保密计算问题,无法应用于两方计算场景. 本文分析了与研究问题相关的文献[20]的两方交集保密计算协议4.2.1和4.2.2,这两个协议分别适用于集合取自无限集合和有限集合的情况. 它们同样仅适用于单一元素的集合,并且假设双方各自拥有一个集合. 在这种情况下,计算双方都会得到交集的内容. 除此之外,文献[20]的协议4.2.1使用哈希函数进行保密计算,可能产生误判.

尽管文献[20]的协议5.1.2和5.2.2也研究了集合的保密计算问题,但本文研究的是集合子集的保密计算,重点在于判断集合元素是否满足包含关系. 文献[20]的协议5.1.2和5.2.2研究的是集合并集的保密计算,侧重于合并和去重,且它们同样仅适用于单一元素的集合. 因此,这两个协议不适用于本文的研究问题.

以上分析表明,元组集合子集的保密计算是一个尚未解决的新问题. 一方面,现有的大多数方案仅针对单一元素组成的集合,协议的组合使用会导致中间计算结果泄露,这可能导致严重的隐私泄露和安全风险;另一方面,现有协议的计算结果存在误差. 因此,如何在元组集合的保密计算中平衡安全性、准确性与协议效率,仍是当前研究面临的挑战.

3　基础知识

3.1　半诚实模型

安全多方计算的协议运行环境分为半诚实参与者模型和恶意攻击者模型[48,49]. 在半诚实模型中,参与者按照协议的步骤执行,并向其他参与者发送正确的信息,不会篡改或偏离协议. 然而,他们可能会记录协议执行过程中收集到的所有信息,并试图根据收集到的信息分析出其他参与者的私密数据.

3.2　模拟范例

Goldreich等人[48,49]利用比特承诺和零知识证明理论设计了一个编译器,这个编译器可以将在半诚实参与者条件下保密计算函数f的协议π自动生成在恶意参与者条件下也能保密计算f的协议π'. 新的协议π'可以迫使恶意参与者以半诚实方式参与协议的执行,否则就会被发现. 因此,大多数情况下,我们只设计半诚实模型下的协议.

本文研究半诚实模型下的两方计算. 两方计算是一个将任意给定的输入对映射为输出对的随机过程,此过程用函数表示为$f(x,y)\to\left(f_1(x,y),f_2(x,y)\right)$,即对于每个输入对$(x,y)$,输出对是随机变量$\left(f_1(x,y),f_2(x,y)\right)$,记这样的函数为$f=f(f_1,f_2)$.

假设Alice和Bob利用协议π保密计算多项式时间函数$f(x,y)$. 设Alice和Bob分别输入x和y. 在协议执行过程中,Alice获得的信息序列记为

$$\text{view}_1^{\pi}(x,y)=\left(x,\ c_1,\ M_1^1,\ \cdots,M_1^t\right)\tag{1}$$

其中,c_1是Alice选择的随机数,$M_1^j\left(j=1,2,\cdots,t\right)$表示Alice收到的第$j$个消息. Bob得到的信息序列也可类似定义.

定义1　参与者均为半诚实时,若存在模拟器S_1和S_2,得

$$\begin{aligned}&\left\{S_1\left(x,f_1(x,y)\right)\right\}_{x,y}\overset{c}{\equiv}\left\{\text{view}_1^{\pi}(x,y)\right\}_{x,y}\\&\left\{S_2\left(x,f_2(x,y)\right)\right\}_{x,y}\overset{c}{\equiv}\left\{\text{view}_2^{\pi}(x,y)\right\}_{x,y}\end{aligned}\tag{2}$$

则称协议π保密地计算f,其中$\overset{c}{\equiv}$表示计算不可区分. 模拟范例是安全多方计算证明安全性的一种常用方法. 协议在半诚实模型下的安全性,需构造满足式(1)和式(2)的模拟器来证明.

3.3　Paillier密码系统

Paillier密码系统,是一种概率公钥加密系统. 该系统具有加法同态性,方案是语义安全的,具体描述如下.

密钥生成$\text{Gen}(k)$. 给定安全参数k,生成两个k比特的大素数p、q,并令$N=p\times q$,$\lambda=\text{lcm}(p-1,q-1)$. 定义函数$L(x)=\dfrac{x-1}{N}$,随机选择一个生成元$g\in Z_N^*$,使得$\gcd(L(g^{\lambda}\bmod N^2),N)=1$,则系统的公钥为$pk=(g,N)$,私钥为$sk=\lambda$.

加密$\text{Enc}(m)$. 若明文$m\in Z_N$,选取随机数$r\in Z_N^*$,密文$c=g^m r^N\bmod N^2$.

解密$\text{Dec}(c)$. 对密文$c\in Z_{N^2}^*$,计算$m=\dfrac{L\left(c^{\lambda}\bmod N^2\right)}{L\left(g^{\lambda}\bmod N^2\right)}\bmod N$.

加法同态性:

$$\begin{aligned}E(m_1)\times E(m_2)&=g^{m_1}r_1^N g^{m_2}r_2^N\bmod N^2\\&=g^{m_1+m_2}\left(r_1r_2\right)^N\bmod N^2\\\text{Dec}\left(E(m_1)\times E(m_2)\right)&=m_1+m_2.\end{aligned}$$

若明文m_2已知时,还可以得到以下性质:

$$\text{Dec}\left(E(m_1)^{m_2}\bmod N^2\right)=m_1m_2$$

3.4 集合的多项式表示

若$X=(x_1, x_2, \cdots, x_n)$为一个集合，则此集合可表示为一个$n$次的多项式：

$$f(x)=(x-x_1)(x-x_2)\cdots(x-x_n)$$
$$=a_0+a_1x+\cdots+a_nx_n=\sum_{i=0}^{n}a^i x^i$$

定理1 若$a\in X\Leftrightarrow f(a)=0$.

从这个定理可以看出，要判断一个元素是否属于一个集合，只需要判断这个元素是不是这个集合转化的多项式的根即可.

4 有全集限制的两方保密计算

4.1 问题描述

Alice拥有n个集合$M_i(i=1, 2, \cdots, n)$，每个集合分别拥有元素$M_i=\{(x_{i1}, w_{i1}), (x_{i2}, w_{i2}), \cdots, (x_{ik_i}, w_{ik_i})\}$. Bob拥有集合$Y=\{(y_1, z_1), (y_2, z_2), \cdots, (y_s, z_s)\}$. 其中，$\{x_{ij}\}(i=1,2,\cdots,n; j=1,2,\cdots,k_i)$，$\{y_j\}(j=1,2,\cdots,S)$为关键字集合，$\{w_{ij}\}(i=1,2,\cdots,n; j=1,2,\cdots,k_i)$，$\{z_j\}(j=1,2,\cdots,S)$为关联值集合.

$$\{x_{ij}\}_{i=1,j=1}^{n,k_i}, \{y_j\}_{j=1}^{s}\subseteq Q=\{q_1, q_2, \cdots, q_k\};$$
$$\{w_{ij}\}(i=1,2,\cdots,n; j=1,2,\cdots,k_i), \{z_j\}(j=1,2,\cdots,S)\subseteq$$
$$U=\{u_1, u_2, \cdots, u_t\}, (u_1<u_2<\cdots<u_t).$$

在不泄露双方隐私的情况下，确定Y是不是M_i的子集.

4.2 计算原理

Alice和Bob根据M_i、Y和Q、U执行下面操作.

(1)Bob根据关键字全集Q、关联值全集U构造$k\times t$的矩阵$\boldsymbol{V}$. 当且仅当$q_j\in Y(1\leqslant j\leqslant k)$且$u_l\in Y$时，矩阵$\boldsymbol{V}$中第$j$行第$l$列元素为1，即$v_{jl}=1$；其余情况$v_{jl}=0$($l=1,2,\cdots,t$).

(2)Bob将编码后的$k\times t$矩阵$\boldsymbol{V}$发给Alice，同时向Alice发送Y的势s.

(3)Alice根据M_1对矩阵进行操作：当$q_j\in M_1$且$u_l\in M_1$时，选择v_{jl}，将其全部相加，并减去Y的势s，记为R_1，即

$$R_1=v_{11}+v_{12}+\cdots+v_{jl}+\cdots+v_{k_1k_1}-s=\sum_{j=1}^{k_1}\sum_{l=1}^{k_1}v_{jl}+(-s) \quad (3)$$

以此类推，Alice对每一个集合M_i进行相同的操作，得出R_i.

$$R_i=v_{i1}+v_{i2}+\cdots+v_{jl}+\cdots+v_{k_ik_i}-s=\sum_{j=1}^{k_i}\sum_{l=1}^{k_i}v_{jl}+(-s) \quad (4)$$

若$R_i=0$，则Y是M_i的子集，记为$F(M_i, Y)=1$；若$R_i\neq 0$，则Y不是M_i的子集，记为$F(M_i, Y)=0$.

为叙述方便，给出如下定义：

$$F(M_i, Y)=\begin{cases}1, & \text{如果}R_i=0\\ 0, & \text{如果}R_i\neq 0\end{cases}$$

以上所述是有全集限制下的元组集合与元组集合子集关系的判定原理，计算过程无安全性可言. 因此，我们基于Paillier加密算法设计出安全的保密计算协议.

4.3 举例说明

Alice有集合$M_1=\{(A, 1), (B, 3), (C, 7), (D, 7)\}$，$M_2=\{(A, 2), (B, 4), (C, 6)\}$，$M_3=\{(C, 6), (C, 7), (D, 7)\}$. Bob有一个集合$Y=\{(A, 1), (D, 7)\}$. 全集$Q=\{A, B, C, D\}$，$U=\{1, 2, 3, 4, 5, 6, 7\}$. Alice和Bob想在不泄露双方隐私的情况下，确定Y是不是M_i的子集.

(1)Bob根据全集构造矩阵

$$\boldsymbol{V}_1=\begin{array}{c}\\A\\B\\C\\D\end{array}\begin{array}{c}\begin{array}{ccccccc}1&2&3&4&5&6&7\end{array}\\\begin{bmatrix}1&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&0\\0&0&0&0&0&0&1\end{bmatrix}\end{array}$$

随后将其与Y的势$s=2$发送给Alice.

(2)Alice根据M_1在矩阵$\boldsymbol{V}$中选择$v_{11}=1$，$v_{23}=0$，$v_{37}=0$，$v_{47}=1$，计算得出$R_1=1+0+0+1-2=0$. 以此类推，Alice根据M_2选择$v_{23}=0$，$v_{24}=0$，$v_{36}=0$，计算得出$R_2=0+0+0-2=-2$. 根据M_3选择$v_{36}=0$，$v_{37}=0$，$v_{47}=1$，计算得出$R_3=0+0+1-2=-1$.

(3)Bob根据$R_1=0$，$R_2=-2$，$R_3=-1$得出Y是M_1的子集.

4.4 元组集合子集的保密计算协议(有全集)

4.4.1 协议1的正确性

定理2 协议1能正确地保密计算有全集限制的元组集合与元组集合的子集关系.

证明 Alice先加密选择，若q_j为Alice和Bob共有元素，则选择出的$q_j=x_{ij}=y_j$，$u_l=z_j=w_{ij}$，即$v_{jl}=1$；若q_j仅为Bob的元素，Alice不会选择；若q_j仅为Alice的元素，则选择出的$v_{jl}=0$. Alice选择出的数据的和若为集合Y的势，即$\sum_{j=1}^{k_i}\sum_{l=1}^{k_i}v_{jl}+(-s)=0$，则$Y$必是$M_i$的子集，反之则不是. 添加随机数$\alpha_i$对Bob隐藏了集合信息且对正确性没有影响. 因此，协议1能正确地计算有全集限制的元组集合与元组集合的子集关系. 证毕.

4.4.2 协议1的安全性

定理3 协议1在半诚实模型下是安全的.

证明 下面在半诚实模型下，构造模拟器S_1和S_2来证明定理3. 在协议1中，

$$\text{view}_1^{\pi}(M_i, Y)=\{M_1, M_2, \cdots, M_n, c_1, E(\boldsymbol{V}), E(s)\}$$
$$\text{view}_2^{\pi}(M_i, Y)=\{Y, c_2, E(R_1), E(R_2), \cdots, E(R_n)\}$$

协议1　元组集合子集的保密计算协议(有全集)

输入: Alice输入集合$M_i=\{(x_{i1},w_{i1}),(x_{i2},w_{i2}),\cdots,(x_{ik_i},w_{ik_i})\}(i=1,2,\cdots,n)$, Bob输入集合$Y=\{(y_1,z_1),(y_2,z_2),\cdots,(y_s,z_s)\}$.

其中,$\{x_{ij}\}(i=1,2,\cdots,n;j=1,2,\cdots,k_i),\{y_j\}(j=1,2,\cdots,S)\subseteq Q=\{q_1,q_2,\cdots,q_k\}$;$\{w_{ij}\}(i=1,2,\cdots,n;j=1,2,\cdots,k_i),\{z_j\}(j=1,2,\cdots,S)\subseteq U=\{u_1,u_2,\cdots,u_t\},(u_1<u_2<\cdots<u_t)$. Q、U分别为关键字、关联值集合

输出: $F(M_i,Y)$

准备:Bob运行Paillier算法,生成公钥和私钥.

(1)Bob将Y的势s和矩阵$\boldsymbol{V}$中的每个元素加密后得到$E(s)$和$E(\boldsymbol{V})$,并发送给Alice. 当$q_j\in Y$且$u_l\in Y$,$E(v_{jl})=E(1)$,若$u_l\notin Y$,则$E(v_{jl})=E(0)$. 当$q_j\notin Y$时,$E(v_{jl})=E(0)$.

(2)Alice根据M_i对$E(\boldsymbol{V})$进行选择. 当$q_j\in M_i$且$u_l\in M_i$,选择$E(v_{jl})$,随后计算

$$E(R_i)=\sum_{j=1}^{k_i}\sum_{l=1}^{k_i}E\left(\alpha_i(v_{jl}-s)\right)=\left(E(v_{i1})\times E(v_{i2})\times\cdots\times E(v_{jl})\times\cdots\times E(v_{k_ik_i})\times E(-s)\right)^{\alpha_i}$$

其中,α_i为Alice选择的随机数,$\alpha_i>0(i=1,2,\cdots,n)$. 随后将结果发送给Bob.

(3)Bob使用私钥依次解密$E(R_i)$:

①若$R_i=0$,输出$F(M_i,Y)=1$,表明Y是M_i的子集;

②若$R_i\neq 0$,输出$F(M_i,Y)=0$,表明Y不是M_i的子集.

其中,$M_i=\{(x_{i1},w_{i1}),(x_{i2},w_{i2}),\cdots,(x_{ik_i},w_{ik_i})\}$, $Y=\{(y_1,z_1),(y_2,z_2),\cdots,(y_s,z_s)\}$分别是Alice和Bob的输入,$c_1$是Alice计算时选择的随机数集合,$E(R_i)$是Alice发给Bob的密文集合,$c_2$是Bob加密时选择的随机数集合,$E(\boldsymbol{V})$、$E(s)$是Bob发给Alice的加密矩阵和加密后的$Y$的势.

首先,构造模拟器S_1模拟$\text{view}_1^{\pi}(M_i,Y)$来证明Bob数据的安全性,$S_1$模拟过程如下.

(1)接收输入$(M_i,F(M_i,Y))$,根据$F(M_i,Y)$的值,S_1选择集合$\tilde{Y}=\{(\tilde{y}_1,\tilde{z}_1),(\tilde{y}_2,\tilde{z}_2),\cdots,(\tilde{y}_s,\tilde{z}_s)\}$使得$F(M_i,Y)=F(M_i,\tilde{Y})$.

(2)S_1根据协议1将$\tilde{Y}$编码为矩阵$\tilde{\boldsymbol{V}}$,并加密为$E(\tilde{\boldsymbol{V}})$,加密$\tilde{Y}$的势为$E(\tilde{s})$.

Paillier加密算法是语义安全的,因此真实环境得到的$E(\tilde{\boldsymbol{V}})$、$E(\tilde{s})$和模拟得到的$E(\boldsymbol{V})$、$E(s)$是计算不可区分的. 令

$$S_1(M_i,F(M_i,Y))=\{M_1,M_2,\cdots,M_n,r_1,E(\tilde{\boldsymbol{V}}),E(\tilde{s})\},$$

则有下式成立:

$$\{S_1(M_i,F(M_i,Y))\}_{M_i,Y}\overset{c}{\equiv}\{\text{view}_1^{\pi}(M_i,Y)\}_{M_i,Y}.$$

类似地,构造模拟器S_2模拟$\text{view}_2^{\pi}(M_i,Y)$来证明Alice数据的安全性,$S_2$模拟过程如下.

(1)接收输入$(Y,F(M_i,Y))$,根据$F(M_i,Y)$的值,S_2选择$M_i'=\{(x_{i1}',w_{i1}'),(x_{i2}',w_{i2}'),\cdots,(x_{ik_i}',w_{ik_i}')\}$使得$F(M_i,Y)=F(M_i',Y)$.

(2)S_2根据协议1及M_i'对加密后的V和s进行选择,当$q_j\in M_i'$且$u_l\in M_i'$时,选择v_{jl},随后计算

$$E(R_i')=\sum_{j=1}^{k_i}\sum_{l=1}^{k_i}E\left(\alpha_i(v_{jl}-s)\right)=\left(E(v_{i1})\times E(v_{i2})\times\cdots\times E(v_{jl})\times\cdots\times E(v_{k_ik_i})\times E(-s)\right)^{\alpha_i}.$$

Paillier加密算法是语义安全的,因此真实环境得到的$E(R_i)$和模拟得到的$E(R_i')$是计算不可区分的. 令

$$S_2(Y,F(M_i,Y))=\{Y,r_2,E(R_1'),E(R_2'),\cdots,E(R_n')\}$$

则有下式成立:

$$\{S_2(M_i,F(M_i,Y))\}_{M_i,Y}\overset{c}{\equiv}\{\text{view}_2^{\pi}(M_i,Y)\}_{M_i,Y}$$

因此,协议1在半诚实模型下是安全的.

5　无集限制的两方保密计算

5.1　问题描述

Alice拥有n个集合$M_i(i=1,2,\cdots,n)$每个集合分别拥有元素$M_i=\{(x_{i1},w_{i1}),(x_{i2},w_{i2}),\cdots,(x_{ik_i},w_{ik_i})\}$. Bob拥有集合$Y=\{(y_1,z_1),(y_2,z_2),\cdots,(y_s,z_s)\}$. 其中,$\{x_{ij}\}(i=1,2,\cdots,n;j=1,2,\cdots,k_i)$,$\{y_j\}(j=1,2,\cdots,S)$为关键字集合,$\{w_{ij}\}(i=1,2,\cdots,n;j=1,2,\cdots,k_i)$和$\{z_j\}(j=1,2,\cdots,S)$为关联值集合. $\{x_{ij}\}(i=1,2,\cdots,n;j=1,2,\cdots,k_i),\{y_j\}(j=1,2,\cdots,S)\subseteq Q=\{q_1,q_2,\cdots,q_k\}$,$(1\leqslant k\leqslant 100)$. 在不泄露双方隐私的情况下,确定$Y$是不是$M_i$的子集.

5.2　基础计算原理

(1)由于n个集合的势不一定相同,而多项式的最高次对应了集合的势,为不泄露集合的势,以n个集合的势的最大值α为参照值,在n个集合$M_i(i=1,2,\cdots,n)$中添加$(0,0)$直至每个集合的势都等于α.

(2) Alice拥有集合$M_1=\{(x_{11},w_{11}),(x_{12},w_{12}),\cdots,(x_{1k_1},w_{1k_1})\}$,添加$(0,0)$,即

$$M_1'=\{(x_{11}',w_{11}'),(x_{12}',w_{12}'),\cdots,(x_{1k_1}',w_{1k_1}'),\underbrace{(0,0),\cdots,(0,0)}_{\alpha-k_1}\}\tag{5}$$

则此集合可表示为一个α次的二元多项式:

$$\begin{aligned}f_1(x,w)&=(x+w-x_{11}-w_{11})\times(x+w-x_{12}-w_{12})\times\cdots\\&\quad\times(x+w-x_{1j}-w_{1j})\times\cdots\times(x+w-0)\\&=\prod_{j=1}^{\alpha}(x+w-x_{1j}-w_{1j})\end{aligned}\tag{6}$$

以此类推，将Alice拥有的其他集合表示为多项式得

$$f_i(x,w)=\left(x+w-x_{i1}-w_{i1}\right)\times\left(x+w-x_{i2}-w_{i2}\right)\times\cdots\times\left(x+w-x_{ij}-w_{ij}\right)\times\cdots\times(x+w-0)=\prod_{j=1}^{\alpha}\left(x+w-x_{ij}-w_{ij}\right)\tag{7}$$

(3)Alice将构造好的n个多项式按顺序发送给Bob.

(4) Bob将集合Y中元素依次代入多项式$f_i(x,\ w)$，即

$$G_i(y,z)=\sum_{j=1}^{s}f_i\left(y_j,z_j\right)=f_i\left(y_1,z_1\right)+f_i\left(y_2,z_2\right)+\cdots+f_i\left(y_s,z_s\right)\tag{8}$$

(5)若$G_i\left(y,z\right)=0$，则Y是M_i的子集，记为$F(M_i,Y)=1$；若$G_i\left(y,z\right)\neq0$，则Y不是M_i的子集，记为$F(M_i,Y)=0$.

为叙述方便，给出如下定义：

$$F(M_i,Y)=\begin{cases}1, & \text{如果}G_i\left(y,z\right)=0\\0, & \text{如果}G_i\left(y,z\right)\neq0\end{cases}$$

以上所述是无全集限制下的元组集合与元组集合子集关系的判定原理，因为这样构造的多项式次数较高，项数较多，效率较低. 我们基于上述原理设计出更高效的判定原理.

5.3 高效计算原理

由于将二元组转化为多项式的方法计算复杂性较高，我们设法将二元组转化为一个数字，然后用数字集合交集的保密计算方法来解决二元组集合的问题. 二元组$\left(x_{ik_i},w_{ik_i}\right)$编码为数字的方法如下：设$1\leqslant k\leqslant100$，$x_{ij}=q_j$，$\left(x_{ik_i},w_{ik_i}\right)$的编码为$\left[\left(x_{ik_i},w_{ik_i}\right)\right]=\left[(100+j)\left\|w_{ik_i}\right.\right]$.

有了上述编码，计算过程如下.

(1)因n个集合的势不一定相同，而多项式的最高次对应了集合的势，为不泄露集合的势，以n个集合的势的最大值α为参照值，Alice在n个集合$M_i(i=1,2,\cdots,n)$中添加$(0,0)$，直至每个集合的势都等于α.

(2)Alice通过上述编码方法把二元组集合转化为数字集合M_i'，Bob也用同样的方法将集合Y转化为数字集合Y'.

(3)Alice根据M_i'构建一元α次多项式$f_i(x)$：

$$f_i(x)=\left(x-x_{i1}\right)\times\left(x-x_{i2}\right)\times\cdots\times\left(x-x_{i\alpha}\right)=\prod_{r=1}^{\alpha}\left(x-x_{ir}\right)\tag{9}$$

其中，$f_i(x)$的系数分别为$a_{i0},a_{i1},\cdots,a_{i\alpha}$. 随后，将其按顺序发送给Bob.

(4)Bob将$Y'=\{y_1',y_2',\cdots,y_s'\}$的元素代入每个多项式$f_i(x)$，计算

$$G_i(y')=f_i(y_1')+f_i(y_2')+\cdots+f_i(y_s')=\sum_{j=1}^{s}f_i(y_j')\tag{10}$$

若$G_i\left(y'\right)=0$，则Y是M_i的子集，记为$F(M_i,Y)=1$；若$G_i\left(y'\right)\neq0$，则Y不是M_i的子集，记为$F(M_i,Y)=0$. 为叙述方便，给出如下定义：

$$F(M_i,Y)=\begin{cases}1, & \text{如果}G_i\left(y'\right)=0\\0, & \text{如果}G_i\left(y'\right)\neq0\end{cases}$$

以上所述是更高效的无全集限制下的元组集合与元组集合子集关系的判定原理，计算过程无安全性可言. 因此，我们基于Paillier加密算法设计出安全的保密计算协议.

5.4 举例说明

Alice有集合$M_1=\{(A,\ 1),\ (B,\ 3),\ (C,\ 7),\ (D,\ 7)\}$，$M_2=\{(A,\ 2),\ (B,\ 4),\ (C,\ 6)\}$，$M_3=\{(C,\ 6),\ (C,\ 7),\ (D,\ 7)\}$. Bob有一个集合$Y=\{(A,\ 1),\ (D,\ 7)\}$. $Q=\{A,\ B,\ C,\ D\}$. Alice和Bob想在不泄露双方隐私的情况下，确定Y是不是M_i的子集.

(1)Bob根据Q和Y构造$Y'=\{1\,011,\ 1\,047\}$，随后计算得到

$b_{10}=1$，$b_{11}=(1\,011)^1,\cdots,b_{14}=(1\,011)^4$；$b_{20}=1$，$b_{21}=(1\,047)^1,\cdots,b_{24}=(1\,047)^4$.

(2)令$\alpha=4$，Alice对M_i进行相同编码后构建多项式$f_i(x)$：

$$f_1(x)=(x-1\,011)\times(x-1\,023)\times\cdots\times(x-1\,047),$$
$$\cdots$$
$$f_3(x)=(x-1\,036)\times(x-1\,037)\times\cdots\times(x-0),$$

则多项式$f_1(x)$的系数为$a_{10},\ a_{11},\ \cdots,\ a_{14}$；$f_2(x)$的系数为$a_{20},\ a_{21},\cdots,\ a_{24}$；$f_3(x)$的系数为$a_{30},\ a_{31},\ \cdots,\ a_{34}$. 随后Alice将其依次发送给Bob.

(3)Bob将$b_{10},\ b_{11},\ \cdots,\ b_{14}$依次代入多项式$f_1(x)$，得$f_1\left(y_1'\right)=a_{10}\times b_{10}+a_{11}\times b_{11}+\cdots+a_{14}\times b_{14}=0$. 随后将$b_{20},\ b_{21},\cdots,\ b_{24}$依次代入多项式$f_1(x)$，得$f_1\left(y_2'\right)=a_{10}\times b_{20}+a_{11}\times b_{21}+\cdots+a_{14}\times b_{24}=0$. 以此类推，$f_2\left(y_1'\right)\neq0$，$f_2\left(y_2'\right)\neq0$，$f_3\left(y_1'\right)\neq0$，$f_3\left(y_2'\right)=0$.

(4)Bob计算得出$G_1\left(y'\right)=0+0=0$，$G_2\left(y'\right)\neq0$，$G_3\left(y'\right)\neq0$，因此Y是M_1的子集.

5.5 元组集合子集的保密计算协议(无全集)

5.5.1 协议2的正确性

定理4 协议2能正确地保密计算无全集限制的元组集合与元组集合的子集关系.

证明 Alice将M_i元素隐藏在多项式$f_i(x)$的系数中，且对多项式的系数进行了加密，Bob无法从中获得M_i中的元素，又因为Alice在M_i中添加元素$(0,0)$，确保所有多项式$f_i(x)$的最高次项次数相同，从而保护了M_i

协议2　元组集合子集的保密计算协议(无全集)

输入: Alice输入集合 $M_i=\{(x_{i1},w_{i1}),(x_{i2},w_{i2}),\cdots,(x_{ik_i},w_{ik_i})\}$. ($i=1,2,\cdots,n$), Bob输入集合 $Y=\{(y_1,z_1),(y_2,z_2),\cdots,(y_s,z_s)\}$.

其中,$\{x_{ij}\}(i=1,2,\cdots,n;j=1,2,\cdots,k_i)$,$\{y_j\}(j=1,2,\cdots,S)\subseteq Q=\{q_1,q_2,\cdots,q_k\}$,$Q$为关键字集合

输出: $F(M_i,Y)$

准备:Alice运行Paillier算法,生成公钥和私钥.

(1)Alice按照计算原理2构建多项式$f_i(x)$,得到的$f_i(x)$的系数为$a_{i0},a_{i1},\cdots,a_{i\alpha}$,将其加密为$E(a_{i0}),E(a_{i1}),\cdots,E(a_{i\alpha})$,随后按顺序发送给Bob.

(2)Bob根据计算原理2得出b_{jr},计算$E(f_i(y'_j))$,即

$$E(f_i(y'_j))=E(a_{i0})^{b_{j0}}\times E(a_{i1})^{b_{j1}}\times\cdots\times E(a_{i\alpha})^{b_{j\alpha}}$$
$$=E(a_{i0}b_{j0}+a_{i1}b_{j1}+\cdots+a_{i\alpha}b_{j\alpha})=E\left(\sum_{r=0}^{\alpha}a_{ir}\times b_{jr}\right)$$

(3)随后,Bob根据$E(f_i(y'_j))$及选择的随机数γ_i计算$E(G_i(y'))$,即

$$E(G_i(y'))=E\left(\sum_{j=1}^{s}\gamma_i f_i(y_j')\right)$$
$$=E(f_i(y'_1))^{\gamma_i}\times E(f_i(y'_2))^{\gamma_i}\times\cdots\times E(f_i(y'_s))^{\gamma_i}$$

并将其按顺序发送给Alice.

(4)Alice使用私钥依次解密$E(G_i(y'))$:

①若$G_i(y')=0$,输出$F(M_i,Y)=1$,表明Y是M_i的子集;

②若$G_i(y')\neq 0$,输出$F(M_i,Y)=0$,表明Y不是M_i的子集.

的势. Bob将Y中元素依次代入加密后的多项式并计算$E(G_i(y'))$. 随后Alice解密$E(G_i(y'))$,若解密结果为0,则表明Y中的元素是该多项式的根,从而得出Y是M_i的子集;若解密结果不为0,则表明Y中的元素不是该多项式的根,从而得出Y不是M_i的子集. Bob在计算$E(G_i(y'))$时添加了随机数γ_i,Alice无法从中推断出Y的相关信息,从而保证了Y的隐私性. 因此,协议2在无全集限制下能正确地保密计算Y是不是M_i的子集. 证毕.

5.5.2　协议2的安全性

定理5　协议2在半诚实模型下是安全的.

证明　下面在半诚实模型下,构造模拟器S_1和S_2来证明定理5. 在协议2中,

$$\text{view}_1^{\pi}(M_i,Y)=\{M_1,M_2,\cdots,M_n,c_1,E(G_1(y')),\cdots,E(G_n(y'))\}$$
$$\text{view}_2^{\pi}(M_i,Y)=\{Y,c_2,E(a_{i0}),E(a_{i1}),\cdots,E(a_{i\alpha})\}.$$

其中,$M_i=\{(x_{i1},w_{i1}),(x_{i2},w_{i2}),\cdots,(x_{ik_i},w_{ik_i})\}$,$Y=\{(y_1,z_1),(y_2,z_2),\cdots,(y_s,z_s)\}$分别是Alice和Bob的输入,$c_1$是Alice加密时选择的随机数集合,$E(G_i(y'))$是Bob发给Alice的密文集合,$c_2$是Bob计算时选择的随机数集合,$E(a_{ij})$是Alice发给Bob的多项式系数.

首先,构造模拟器S_1模拟$\text{view}_1^{\pi}(M_i,Y)$来证明Bob数据的安全性,$S_1$模拟过程如下.

(1)接收输入$(M_i,F(M_i,Y))$,根据$F(M_i,Y)$的值,S_1选择集合$\tilde{Y}=\{(\tilde{y}_1,\tilde{z}_1),(\tilde{y}_2,\tilde{z}_2),\cdots,(\tilde{y}_s,\tilde{z}_s)\}$使得$F(M_i,Y)=F(M_i,\tilde{Y})$.

(2)S_1根据协议2得出$\tilde{b}_{jr}$,随后计算

$$E(f_i(y'_j))=E(a_{i0})^{\tilde{b}_{j0}}\times E(a_{i1})^{\tilde{b}_{j1}}\times\cdots\times E(a_{i\alpha})^{\tilde{b}_{j\alpha}}$$
$$=E(a_{i0}\tilde{b}_{j0}+a_{i1}\tilde{b}_{j1}+\cdots+a_{i\alpha}\tilde{b}_{j\alpha})=E\left(\sum_{r=0}^{\alpha}a_{ir}\times\tilde{b}_{jr}\right)$$

并根据$E(f_i(y'_j))$计算

$$E(G_i(\tilde{y}'))=E\left(\sum_{j=1}^{s}\gamma_i f_i(y'_j)\right)$$
$$=E(f_i(\tilde{y}'_1))^{\gamma_i}\times E(f_i(\tilde{y}'_2))^{\gamma_i}\times\cdots\times E(f_i(\tilde{y}'_s))^{\gamma_i}$$

Paillier加密算法是语义安全的,因此真实环境得到的$E(G_i(y'))$和模拟得到的$E(G_i(\tilde{y}'))$是计算不可区分的. 令

$$S_1(M_i,F(M_i,Y))=$$
$$\{M_1,M_2,\cdots,M_n,c_1,E(G_1(\tilde{y}')),E(G_2(\tilde{y}')),\cdots,E(G_n(\tilde{y}'))\},$$

则有下式成立:

$$\{S_1(M_i,F(M_i,Y))\}_{M_i,Y}\overset{c}{\equiv}\{\text{view}_1^{\pi}(M_i,Y)\}_{M_i,Y}$$

类似地,构造模拟器S_2模拟$\text{view}_2^{\pi}(M_i,Y)$来证明Alice数据的安全性,$S_2$模拟过程如下.

(1)接收输入$(Y,F(M_i,Y))$,根据$F(M_i,Y)$的值,S_2选择$\tilde{M}_i=\{\tilde{x}_{i1},\tilde{x}_{i2},\cdots,\tilde{x}_{ik_i}\}$,使得$F(\tilde{M}_i,Y)=F(M_i,Y)$.

(2)S_2根据计算原理2将$\tilde{M}_i$构造为$\tilde{M}_i'$,并根据$\tilde{M}_i'$构造多项式$f_i(\tilde{x})$,多项式的系数分别为$\tilde{a}_{i0},\tilde{a}_{i1},\cdots,\tilde{a}_{i\alpha}$,加密为$E(\tilde{a}_{i0}),E(\tilde{a}_{i1}),\cdots,E(\tilde{a}_{i\alpha})$.

Paillier加密算法是语义安全的,因此真实环境得到的$E(a_{ij})$和模拟得到的$E(\tilde{a}_{ij})$是计算不可区分的. 令

$$S_2(Y,F(M_i,Y))=\{Y,c_2,E(\tilde{a}_{i0}),E(\tilde{a}_{i1}),\cdots,E(\tilde{a}_{i\alpha})\},$$

则有下式成立:

$$\{S_2(M_i,F(M_i,Y))\}_{M_i,Y}\overset{c}{\equiv}\{\text{view}_2^{\pi}(M_i,Y)\}_{M_i,Y}$$

因此,协议2在半诚实模型下是安全的. 证毕.

6　方案性能分析

在本文第2节中已对现有方案[14,20]和[34~43]进行具体分析,它们皆在研究单一元素集合交集的保密计算问题,而本文研究的是元组集合子集的保密计算.

我们并未将所提方案与现有的集合交集保密计算问题进行性能对比，原因在于本文提出的协议是第一个在某一参与方拥有多个集合，既保护参与双方集合的势，也保护集合子集的势的情形下进行元组集合子集的保密计算的工作. 由于本文提出的协议具备更严格的安全性需求和更具挑战性的功能特性，这些因素不可避免地会引入额外开销，若将具有不同安全性需求或功能特性的方案进行性能对比，既不合理，亦有失公平. 尽管如此，为了探究所提方案与现有工作的差异，我们在表1中提供了详细的特性对比(见引言部分). 下面对本文方案进行分析. 方案以模指数运算次数作为复杂性度量指标，其中Paillier算法每一次加密需要2次模指数运算，解密一次需要1次模指数运算.

6.1 计算复杂性分析

协议1的复杂性 协议1中，Bob将$Y=\{(y_1,z_1),(y_2,z_2),\cdots,(y_s,z_s)\}$编码为矩阵$\boldsymbol{V}$，加密$\boldsymbol{V}$和$Y$的势$s$共需模指数运算$2(kt+1)$次，其中$k$、$t$分别为关键字集合、关联值集合的势. Alice根据M_i对加密后的$\boldsymbol{V}$进行选择，不参与解密. 随后参与方Bob需解密Alice发送的n个计算结果，进行模指数运算n次. 协议1共需$2(kt+1)+n$次模指数运算.

协议2的复杂性 协议2中，Alice对这n个集合M_i进行编码，并构造多项式，对多项式的系数进行加密，需加密$(\alpha+1)n$次，共需模指数运算$2(\alpha+1)n$次，Bob仅将Y中元素编码计算后代入加密多项式. 随后Alice需解密Bob发送的n个计算结果，进行模指数运算n次. 协议2共需$2(\alpha+1)n+n$次模指数运算.

6.2 通信复杂性分析

我们采用通信轮数来衡量通信复杂性，协议1和协议2均需要3轮通信，具体分析见表2.

表2 协议计算复杂性与通信复杂性

协议	计算复杂性	通信复杂性
协议1	$2(kt+1)+n$	3
协议2	$2(\alpha+1)n+n$	3

注：n表示某一参与方拥有的集合个数；k、t分别表示在有全集限制的两方保密计算中，关键字全集、关联值全集的势；α表示在无全集限制的两方保密计算中，某一参与方拥有的n个集合的势的最大值.

6.3 实验数据分析

测试环境：Windows 11 64位操作系统，处理器是12th Gen Intel(R) Core(TM) i5-12400 2.50 GHz，内存是16 GB，在PyCharm用Python 3.11语言运行实现.

实验方法：本文协议1和2均采用Paillier算法，设定素数的比特数为1 024，所有实验进行100次，统计平均值(忽略协议中预处理数据时间)，具体如下.

由图1可知，在协议1中，固定Alice的集合个数n=50时，执行时间和关键字集合的势k与关联值集合的势t有关. 图1表示执行时间随关键字集合的势和关联值集合的势变化的二维图. 通过分析可知，执行时间随关键字集合的势k和关联值集合的势t增加而增长.

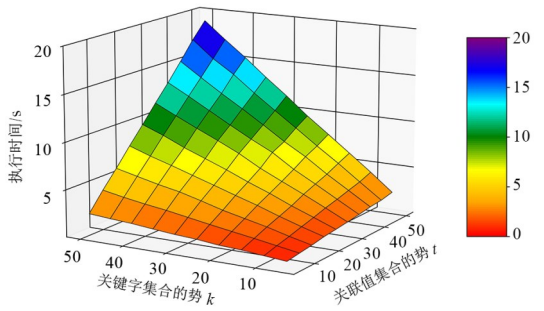


图1 协议1中执行时间随关键字、集合的势和关联值的势的变化规律

由图2可知，在协议2中，执行时间与Alice拥有的集合的势的最大值α和集合的个数n有关. 图2表示执行时间随Alice拥有的集合的势的最大值α和集合的个数n变化的二维图. 通过分析可知，执行时间随Alice拥有的集合的势的最大值α和集合的个数n增加而增长.

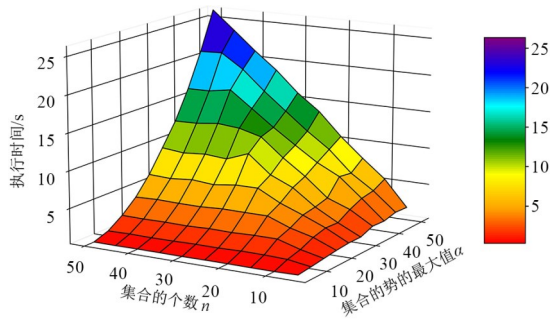


图2 协议2中执行时间随集合的势的最大值和集合的个数的变化规律

7 总结与展望

保密计算子集问题在安全多方计算中具有重要的研究意义和应用价值. 本文提出了元组类型的集合子集的保密计算问题的安全多方计算协议. 首先，设计了适用于有全集情况下的协议，利用加密选择的方法保护参与双方集合的势，并确保了计算结果的准确性. 随后，利用集合的多项式表示解决了无全集情况下的元组集合子集的保密计算问题. 此外，本文的协议可以有效应对在该问题中某一参与方拥有多个集合的情形. 最后，通过模拟范例证明了这两个协议在半诚实模型下的安全性. 尽管协议有效解决了元组集合子集的保密计算问题，但仍有一些待进一步改进或探索的方面. 例如，如何将这些计算协议应用于云环境中，以减少用

户的计算负担和降低计算复杂度,从而使其能够得到更广泛的应用,将成为我们未来研究的一个重要方向.

参考文献

[1] YAO A C. Protocols for secure computations[C]//23rd Annual Symposium on Foundations of Computer Science (sfcs 1982). Piscataway: IEEE, 2008: 160-164.

[2] FREEDMAN M J, HAZAY C, NISSIM K, et al. Efficient set intersection with simulation-based security[J]. Journal of Cryptology, 2016, 29(1): 115-155.

[3] GOLDREICH O. Cryptography and cryptographic protocols[J]. Distributed Computing, 2003, 16(2): 177-199.

[4] KUMAR S N. Review on network security and cryptography[J]. International Transaction of Electrical and Computer Engineers System, 2015, 3(1): 1-11.

[5] ZHAO C, ZHAO S N, ZHAO M H, et al. Secure multi-party computation: Theory, practice and applications[J]. Information Sciences, 2019, 476: 357-372.

[6] GOLDWASSER S. Multi party computations: Past and present[C]//Proceedings of the Sixteenth Annual ACM Symposium on Principles of Distributed Computing - PODC'97. New York: ACM, 1997: 1-6.

[7] BEN-OR M, WIGDERSON A. Completeness theorems for non-cryptographic fault-tolerant distributed computation[C]//Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing - STOC'88. New York: ACM, 1988: 1-10.

[8] TEO S G, CAO J N, LEE V C S. DAG: A general model for privacy-preserving data mining[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 32(1): 40-53.

[9] WANG F W, ZHU H, LIU X M, et al. Privacy-preserving collaborative model learning scheme for E-healthcare[J]. IEEE Access, 2019, 7: 166054-166065.

[10] DUONG T, PHAN D H, TRIEU N. Catalic: Delegated PSI cardinality with applications to contact tracing[M]//Advances in Cryptology - ASIACRYPT 2020. Cham: Springer International Publishing, 2020: 870-899.

[11] FREEDMAN M J, NISSIM K, PINKAS B. Efficient private matching and set intersection[M]//Advances in Cryptology - EUROCRYPT 2004. Berlin, Heidelberg: Springer, 2004: 1-19.

[12] KISSNER L, SONG D. Privacy-preserving set operations[M]//Advances in Cryptology - CRYPTO 2005. Berlin, Heidelberg: Springer, 2005: 241-257.

[13] 窦家维, 陈明艳 . 多重集的保密计算及应用[J]. 电子学报, 2020, 48(1): 204-208.
DOU J W, CHEN M Y. Secure multiset operations and their applications[J]. Acta Electronica Sinica, 2020, 48(1): 204-208. (in Chinese)

[14] 赵雪玲, 家珠亮, 李顺东 . 集合交集问题的安全计算[J]. 密码学报, 2022, 9(2): 294-307.
ZHAO X L, JIA Z L, LI S D. A secure multiparty intersection computation[J]. Journal of Cryptologic Research, 2022, 9(2): 294-307. (in Chinese)

[15] WANG W L, LI S D, DOU J W, et al. Privacy-preserving mixed set operation[J]. Information Sciences, 2020, 525(7): 67-81.

[16] 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[J]. 计算机学报, 2020, 43(8): 1397-1413.
DOU J W, LIU X H, WANG W L. Privacy preserving two-party rational set computation[J]. Chinese Journal of Computers, 2020, 43(8): 1397-1413. (in Chinese)

[17] 张朋飞, 翟睿辰, 程祥, 等 . 满足地理不可区分性的偏好感知多对多任务分配算法[J]. 电子学报, 2025, 53(3): 878-894.
ZHANG P F, ZHAI R C, CHENG X, et al. A preference-aware many-to-many task allocation algorithm under geo-indistinguishability[J]. Acta Electronica Sinica, 2025, 53(3): 878-894. (in Chinese)

[18] HU J W, ZHAO Y J, HONG MENG TAN B, et al. Enabling threshold functionality for private set intersection protocols in cloud computing[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 6184-6196.

[19] WU A X, XIN X J, ZHU J H, et al. Cloud-assisted laconic private set intersection cardinality[J]. IEEE Transactions on Cloud Computing, 2024, 12(1): 295-305.

[20] 周素芳 . 集合相关问题的保密计算研究[D]. 西安: 陕西师范大学, 2019.
ZHOU S F. Research on Secure Multiparty Computation of Set-Related Problems[D]. Xi'an: Shaanxi Normal University, 2019. (in Chinese)

[21] 窦家维, 王颖囡, 葛雪 . 区间关系保密计算若干问题研究[J]. 电子学报, 2021, 49(1): 50-57.
DOU J W, WANG Y N, GE X. Some research on secure interval relation computation[J]. Acta Electronica Sinica, 2021, 49(1): 50-57. (in Chinese)

[22] 袁水莲, 皮德常, 胥萌 . 基于差分隐私的轨迹隐私保护方法[J]. 电子学报, 2021, 49(7): 1266-1273.
YUAN S L, PI D C, XU M. Trajectory privacy protection method based on differential privacy[J]. Acta Electronica Sinica, 2021, 49(7): 1266-1273. (in Chinese)

[23] 康海燕, 冀源蕊 . 基于本地化差分隐私的时序位置发布方案研究[J]. 电子学报, 2022, 50(9): 2222-2232.
KANG H Y, JI Y R. Research on time-serial location data publication based on local differential privacy[J]. Acta Electronica Sinica, 2022, 50(9): 2222-2232. (in Chinese)

[24] 朱伊波, 方贤进, 张朋飞, 等. 本地差分隐私下面向离群点的真值发现算法研究[J]. 电子学报, 2025, 53(5): 1541-1558.
ZHU Y B, FANG X J, ZHANG P F, et al. A study of truth discovery algorithms for forward outliers under local differential privacy[J]. Acta Electronica Sinica, 2025, 53(5): 1541-1558. (in Chinese)

[25] XIONG L, JIANG Z L, HUANG Y, et al. Efficient private set intersection based on functional encryption[C]// The 2022 4th International Conference on Data Intelligence and Security. Piscataway: IEEE, 2022: 9-15.

[26] BANAWAN K, ARASLI B, WEI Y P, et al. The capacity of private information retrieval from heterogeneous uncoded caching databases[J]. IEEE Transactions on Information Theory, 2020, 66(6): 3407-3416.

[27] WANG N, HEIDARZADEH A, SPRINTSON A, et al. A new approach to harnessing side information in multi-server private information retrieval[C]//The IEEE International Symposium on Information Theory (ISIT). Piscataway: IEEE, 2024: 2646-2651.

[28] CHEN Z, WANG Z Y, JAFAR S ALI. The capacity of T-private information retrieval with private side information[J]. IEEE Transactions on Information Theory, 2020, 66(8): 4761-4773.

[29] ERHILI L, HEIDARZADEH A. Achieving capacity of PIR with private side information with low sub-packetization and without MDS codes[C]//2024 IEEE International Symposium on Information Theory. Piscataway: IEEE, 2024: 2652-2657.

[30] SIAVOSHANI M J, SHARIATPANAHI S P, MADDAH-ALI M A. Private information retrieval for a multi-message scenario with private side informationin[J]. IEEE Transactions on Communications, 2021, 69(5): 3235-3244.

[31] LU Y X, JAFAR S A. On single server private information retrieval with private coded side information[J]. IEEE Transactions on Information Theory, 2023, 69(5): 3263-3284.

[32] HEIDARZADEH A, SPRINTSON A. The linear capacity of single-server individually-private information retrieval with side information[C]//The IEEE International Symposium on Information Theory (ISIT). Piscataway: IEEE, 2022: 2833-2838.

[33] HSU H C, LIU Z Y, TSO R, et al. Multi-value private information retrieval using homomorphic encryption[C]// 2020 15th Asia Joint Conference on Information Security. Piscataway: IEEE, 2020: 82-88.

[34] LING G W, TANG F, CAI C C, et al. P²FRPSI: Privacy-preserving feature retrieved private set intersection[J]. IEEE Transactions on Information Forensics and Security, 2024, 19: 2201-2216.

[35] LING G W, TANG P, SHAN J Y, et al. More efficient, privacy-enhanced, and powerful privacy-preserving feature retrieval private set intersection[J]. IEEE Transactions on Information Forensics and Security, 2025, 20: 4815-4827.

[36] WANG Z S, BANAWAN K, ULUKUS S. Private set intersection: A multi-message symmetric private information retrieval perspective[J]. IEEE Transactions on Information Theory, 2022, 68(3): 2001-2019.

[37] CHEN Y C, HUANG K C. JEDI: Joint and effective privacy preserving outsourced set intersection and data integration protocols[J]. IEEE Transactions on Information Forensics and Security, 2023, 18: 4504-4514.

[38] 赵昊. 两种隐私集合交集计算协议研究[D]. 成都: 电子科技大学, 2024.
ZHAO H. Research on Two Types of Private Set Intersection Computation Protocols[D]. Chengdu: University of Electronic Science and Technology of China, 2024. (in Chinese)

[39] 徐银. 隐私保护下集合运算研究[D]. 济南: 山东大学, 2023.
XU Y. Research on Privacy-Preserving Set Operations[D]. Jinan: Shandong University, 2023. (in Chinese)

[40] 罗磊. 基于云辅助的隐私集合交集计算协议的研究[D]. 上海: 华东师范大学, 2023.
LUO L. Researches on Private Set Intersection Computation Protocols Under the Cloud Environments[D]. Shanghai: East China Normal University, 2023. (in Chinese)

[41] 马立驹. 隐私保护的集合运算研究[D]. 济南: 山东师范大学, 2024.
MA L J. Research on Privacy-Preserving Set Operations[D]. Jinan: Shandong Normal University, 2024. (in Chinese)

[42] 苏代钊. 面向隐私保护的集合求交计算及其应用研究[D]. 成都: 西华大学, 2024.
SU D Z. Research on Privacy-Preserving Set Intersection Computation and Its Applications[D]. Chengdu: Xihua University, 2024. (in Chinese)

[43] 李婷. 基于大数据平台支持任意计算和模糊匹配的隐私保护集合交集方案[D]. 西安: 西安电子科技大学, 2024.
LI T. Privacy Protection Set Intersection Scheme Based on Big Data Platform Supporting Arbitrary Computing and Fuzzy Matching[D]. Xi'an: Xidian University, 2024. (in Chinese)

[44] 李顺东, 赵雪玲, 家珠亮. 集合交集元素和的保密计算[J].

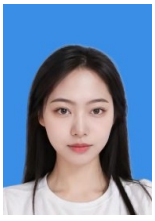
电子学报, 2023, 51(1): 86-92.
LI S D, ZHAO X L, JIA Z L. Private intersection-sum computation[J]. Acta Electronica Sinica, 2023, 51(1): 86-92. (in Chinese)
[45] 马秀莲, 张倦倦, 李顺东. 保密计算交集对应元素和的最大值[J]. 电子学报, 2023, 51(7): 1835-1841.
MA X L, ZHANG J J, LI S D. Maximum value of sum of intersection elements of secret calculation[J]. Acta Electronica Sinica, 2023, 51(7): 1835-1841. (in Chinese)
[46] LIU Q, GUO X J, YANG K, et al. Labeled private set intersection from distributed point function[J]. IEEE Transactions on Information Forensics and Security, 2025, 20: 2970-2983.
[47] YANG Y B, HU Y W, LI R F, et al. LSE: Efficient symmetric searchable encryption based on labeled PSI[J]. IEEE Transactions on Services Computing, 2024, 17(2): 563-574.
[48] GOLDREICH O, MICALI S, WIGDERSON A. How to play ANY mental game[C]//Proceedings of the Nineteenth Annual ACM Conference on Theory of Computing - STOC'87. New York: ACM, 1987: 218-229.
[49] GOLDREICH O. Foundations of Cryptography Volume II Basic Applications[M]. Cambridge, UK: Cambridge University Press, 2004.

作者简介

李顺东　男,1963年12月出生于河南省平顶山市. 现为陕西师范大学计算机科学学院教授、博士生导师. 主要研究方向为密码学与信息安全.
E-mail: shundong@snnu.edu.cn

杜佶欣　女,1999年12月出生于山西省太原市. 现为陕西师范大学计算机科学学院硕士研究生. 主要研究方向为密码学与信息安全.
E-mail: djixin@snnu.edu.cn

余佳桐　女,2000年6月出生于四川省资阳市. 现为陕西师范大学计算机科学学院硕士研究生. 主要研究方向为密码学与信息安全.
E-mail: yujiatong@snnu.edu.cn



吴川宇　女,2000年1月出生于河南省驻马店市. 现为陕西师范大学计算机科学学院硕士研究生. 主要研究方向为密码学与信息安全.
E-mail: wuchuanyu@snnu.edu.cn