河南理工大学

# 全日制硕士学位论文

## 安全多方计算集合交集问题研究

申请人姓名：秦榕霞

指导教师：张静

学位类别：工学硕士

专业名称：计算机科学与技术

研究方向：网络与信息安全

**河南理工大学计算机科学与技术学院**

**二〇二二年五月**

中图分类号：TP309　　密　　级：公开
UDC：004　　单位代码：10460

# 安全多方计算集合交集问题研究
# Research on Set Intersection of Secure Multi-party Computation



| | | | |
|---|---|---|---|
| 申请人姓名 | 秦榕霞 | 学位类别 | 工学硕士 |
| 专业名称 | 计算机科学与技术 | 研究方向 | 网络与信息安全 |
| 导　　师 | 张静 | 职　　称 | 副教授 |
| 提交日期 | 2022.05 | 答辩日期 | 2022.05 |



河南理工大学

# 摘 要

5G、互联网和通信技术蓬勃发展的今天，信息共享已成为实现信息资源价值最大化的有效手段。在大数据、区块链和物联网等领域，越来越多的用户通过多方之间数据的协同计算有效实现信息资源共享，来获得更高水平的服务以及更加便捷舒适的生活方式。然而，在信息共享的过程之中，用户隐私的数据也面临着被泄露的风险。如何在多方协同计算过程中实现数据的隐私保护成为人们密切关注的问题。安全多方计算在充分利用数据进行协同计算时，可以保护数据的隐私不被泄露，成为数据隐私保护计算的重要研究对象。集合交集问题作为安全多方计算领域的特定应用研究问题，具有重要的理论和实际研究意义。本文主要对安全多方计算领域的集合交集问题进行研究，研究工作如下：

（1）针对两方之间隐私集合交集计算过程中的公平性问题和计算复杂度较高问题，提出一种外包的公平两方隐私集合交集协议。协议利用哈希映射算法对集合元素以哈希桶为单位进行分类，减少元素的比较次数；同时结合门限加密算法和云服务器，将双方相应哈希桶内的元素以密文的形式在服务器上比较进行交集计算，降低参与用户的计算负担。协议执行结束后参与双方可同时得知交集计算结果，实现了协议的公平性。

（2）针对特定应用场景中安全计算参与用户共同信息的数量而不揭露具体信息的要求，提出一种公平的两方隐私集合交集势协议。协议将哈希映射算法和可交换加密算法进行结合，以哈希桶为单位在密文形式下计算双方相应哈希桶内的相等元素的个数来获得双方集合的交集势。参与双方并行完成协议的每个阶段，确保参与双方能够同时得到交集势结果，实现了协议的公平性。

（3）针对多个参与方之间的隐私集合交集计算问题，提出一种多方隐私集合交集协议。协议不预先设定所有参与者共有的集合，以哈希映射算法和等值比较协议为基本构建模块分别计算第一个参与者与其他参与者之间的交集信息，利用 ElGamal 门限加密算法的加法同态性以哈希桶为单位将所有交集信息以密文形式汇总得到多方之间的交集。经过理论和实验分析表明，协议的综合性能较优。

**关键词：**安全多方计算；集合交集；集合交集势；隐私保护

# Abstract

With the flourishing development of 5G, Internet and communication technology, information sharing has become an effective means to maximize the value of information resources. In fields such as big data, blockchain and Internet of Things, more and more users effectively realize information resource sharing through data collaborative computing among multiple parties, so as to obtain a higher level of services and a more convenient and comfortable lifestyle. However, when information is shared, the private data of users also face the risk of disclosure. How to achieve data privacy protection in the process of multi-party collaborative computing has become an issue of close attention. Secure multi-party computing can protect the privacy of data from being leaked when making full use of data for collaborative computing. It has become an important research object of data privacy protection computing. Private set intersection is a special applied research problem in secure multi-party computing and it has important theoretical and practical research significance. This paper mainly studies the set intersection problem in the field of secure multi-party computing. The research work is as follows:

(1) Aiming at the fairness problem and high computational complexity in the calculation process of privacy set intersection between two parties, an outsourced fair two-party privacy set intersection protocol is proposed. The protocol uses the hash mapping algorithm to classify the set elements by hash bin, reducing the number of element comparisons. At the same time, combined with the threshold encryption algorithm and the cloud server, the elements in the corresponding hash bin of both sides are compared on the server in the form of ciphertext for intersection calculation, so as to reduce the computational burden of participating users. After the protocol is executed, the two parties can know the intersection calculation results at the same time, which realizes the fairness of the protocol.

(2) Aiming at the requirement of safely calculating the amount of common information of participating users without disclosing specific information in a specific application scenario, a fair two-party privacy set intersection cardinality protocol is proposed. The protocol combines the hash mapping algorithm with the commutative encryption algorithm. Taking the hash bin as the unit, the number of equal elements in the corresponding hash bin of both sides is calculated in the form of ciphertext to obtain the intersection potential of the sets of both

sides. Participating parties complete each stage of the agreement in parallel to ensure that both parties can get the results of the intersection at the same time, and the fairness of the agreement is realized.

(3) Aiming at the calculation of the intersection of privacy sets between multiple parties, a multi-party privacy set intersection protocol is proposed. The protocol does not preset the set shared by all participants. It uses hash mapping algorithm and equivalent comparison protocol as the basic building blocks to calculate the intersection information between the first participant and other participants respectively. In addition, the protocol uses the additive homomorphism of the ElGamal threshold encryption algorithm to gather all the intersection information in ciphertext in the unit of hash bin to obtain the intersection between multiple parties. Theoretical and experimental analysis shows that the comprehensive performance of the protocol is superior.

**Keywords：** Secure multi-party computing; Set intersection; Set intersection cardinality; Privacy protection

# 目　录

# 图清单

# 表清单

# 变量注释表

| | |
|---|---|
| $pk$ | 系统公钥 |
| $sk$ | 系统私钥 |
| $n$ | 协议参与者的个数 |
| $pk_1, pk_2, \ldots, pk_n$ | 参与者生成公钥 |
| $sk_1, sk_2, \ldots, sk_n$ | 参与者生成私钥 |
| $G$ | 循环群 |
| $p$ | 大素数 |
| $g$ | 生成元 |
| $H$ | 加密时的哈希函数 |
| $P_1, P_2, \ldots, P_n$ | 协议参与者 |
| $X, Y$ | 两方安全协议中参与者的隐私集合 |
| $X_1, X_2, \ldots, X_n$ | 多方安全协议中参与者的隐私集合 |
| $n_1, n_2, \ldots, n_n$ | 参与者隐私集合大小 |
| $T_1, T_2, \ldots, T_n$ | 哈希表 |
| $m$ | 哈希表中哈希桶的个数 |
| $b$ | 哈希桶的大小 |
| $h_1, h_2$ | 预处理元素的哈希函数 |
| $D$ | $D=(g, g^a, g^b, g^{ab}) \in G^4$ 判定性 Diffie-Hellman 四元组分布 |
| $R$ | $R=(g, g^a, g^b, g^c) \in G^4$ 随机四元组分布 |

# 1 绪论

# 1 Introduction

安全多方计算作为密码学的一个子领域，它允许多个数据所有者执行协作计算，而无需从本地数据库中取出原始数据，保证了“数据可用不可见”，成为平衡隐私保护和数据共享的一个重要工具。集合交集问题作为安全多方计算领域的特定应用问题，具有重要的理论研究意义和实际应用研究价值。本章主要对安全多方计算领域中集合交集问题的研究背景与意义进行了阐述，并分析了集合交集问题的国内外研究现状，介绍了本文的主要工作，最后给出了论文的组织结构。

## 1.1 研究背景与意义（Research Background and Significance）

5G、物联网、边缘计算等技术的广泛应用给人们的日常生活带来了便利，也给人们隐私数据的安全性带来了隐患。随着多起用户隐私泄露事件的发生，公民的隐私数据保护遇到了严峻的挑战，如何在不侵犯隐私的情况下利用用户的大量数据信息成为一个重要问题。安全多方计算（Secure Multi-party Computation，SMC）在充分利用数据进行协同计算时，可以保护数据的隐私不被泄露，在解决这一难题中发挥了巨大作用，成为数据隐私保护计算方面的重要研究对象。

安全多方计算[1]是一种通用的密码学原语，以保护隐私的方式实现了多方之间的联合计算。作为密码学领域的一个重要的基础研究课题，安全多方计算解决了分布式环境中互不信任的参与方以一种安全的方式对来自多个参与者的私有数据执行协同计算的问题。形式上，在安全多方计算场景中，持有私有输入的两个或多个参与方使用这些输入来执行一些联合功能计算。在计算结束后，需要确保所有参与者输入数据的隐私性以及计算结果的正确性。此外，联合功能计算是一个通用的概念，可以用于大多数的加密任务，例如加密、身份验证、零知识证明[2]、承诺方案、不经意传输[3]和其他非加密协议（电子投票、机器学习、基因组数据处理等）。因此，安全多方计算是密码学领域最普遍、最基本的研究课题，任何涉及多方的加密任务都可视为安全多方计算。对安全多方计算的深入研究促进了零知识证明、不经意传输、密钥共享[4]等底层密码原语的发展，为交互式协议的可证明安全性奠定了理论基础，促进了现代密码学的发展。

集合交集问题[5-7]的安全多方计算是数据隐私保护问题的一个重要研究内容。其旨在构建安全协议，以允许多个互不信任的参与者共同计算隐私输入集合的交集，即使面对不诚实的行为也能确保输出的正确性，同时维护输入信息的隐私性。其具体可分为集合交集计算、集合交集势计算以及其他集合交集的变体计算等问题。目前对于集合交集

问题的研究主要分为理论研究和具体应用研究两个研究方向。理论研究主要包括安全模型、可行性和复杂度等方面的研究。具体应用方面的研究目的是将理论研究中产生的结果转化为解决现实世界中的隐私安全问题的具体集合交集协议。随着云计算[8]、移动计算[9]、物联网[10]等新兴技术的日益普及，针对具体应用方面的集合交集问题研究工作受到了更加广泛的关注。作为一种数据隐私计算工具，集合交集计算成为解决这些领域信息共享过程中数据隐私问题的有效技术手段。如智能医疗系统[11]中，允许合法用户访问患者的体征数据，通过将患者体征数据和医疗系统数据库之间执行集合交集协议，根据交集结果为患者提供更准确有效的治疗，从而减轻医护人员的负担。集合交集协议在为患者提供了更精准的医疗服务的同时也保护了患者数据的隐私。

综上所述，集合交集问题作为安全多方计算领域的特定应用研究，不仅促进了现代密码学底层密码原语的发展，对密码学的研究工作进行了有效补充，同时也是解决现实世界中隐私安全问题的有效技术手段。具有较高的理论和实际应用研究价值。

## 1.2 国内外研究现状（Research Status at Home and Abroad）

安全多方计算的概念起源于 Yao 在 1982 年[12]提出的百万富翁问题：“两个百万富翁想知道谁更富有，但他们都不想对方得知自己财富的具体信息。”该情况的解决方案本质上是安全地执行两方函数比较。随后在 1986 年，Yao[13]以不经意传输协议为基本构建工具，对双方的输入信息进行双重加密，给出了两方之间通用的安全计算协议。Goldreich 等人[14]在 1987 年对姚氏两方安全计算的百万富翁问题进行了扩展，提出了一个通用的安全多方计算框架来解决多方的安全计算问题，并将多方协议分为半诚实模型和恶意敌手模型。该框架为研究更多的多方协同计算问题提供了理论依据，吸引了越来越多的学者对安全多方计算的理论分析以及实际应用进行研究，对安全多方计算的发展具有重大意义。目前，针对安全多方计算领域研究的问题主要有：科学计算[15]、统计分析、数据挖掘[16]、计算几何[17]和集合问题等方面。集合交集问题的安全多方计算是隐私保护问题的一个重要研究内容，指持有各自私密集合的多个参与者共同进行集合的交集运算，除正确的计算结果之外无法得知其他参与者的任何信息。集合交集问题的研究始于 2002 年 Clifton 等人[18]利用单向映射函数计算集合的交集势，从而实现数据挖掘领域中的数据隐私保护。随着对集合交集问题研究的更加深入，根据计算结果的不同，集合交集问题可进一步的分为集合交集计算[19-23]、集合交集势计算[24-27]和集合交集和计算[28]等具体研究问题。此外，根据参与方个数不同，集合交集问题又可分为两方的集合交集问题研究和多方的集合交集问题研究，如图 1-1 所示。

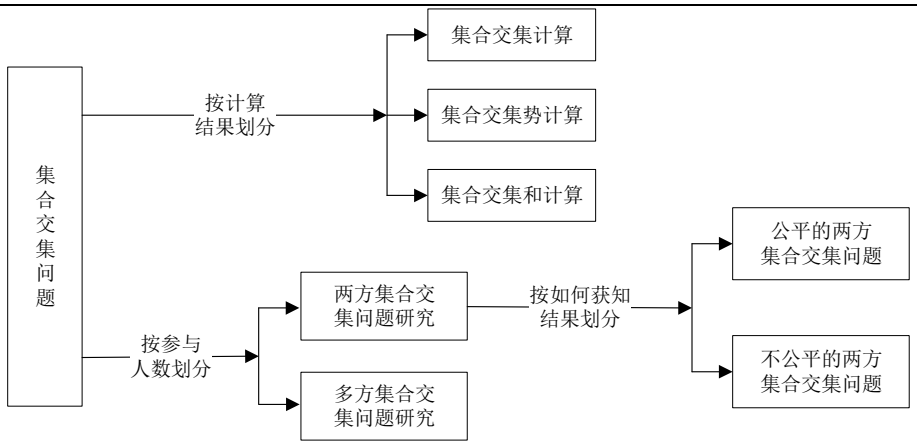



**图 1-1 集合交集问题研究内容**
**Fig.1-1 Research content of set intersection problem**

本文主要针对集合交集问题中两方之间的集合交集计算、多方之间的集合交集计算以及两方之间的集合交集势计算进行研究。

### 1.2.1 两方集合交集计算

两方之间的隐私集合交集(Private Set Intersection，PSI)计算是目前安全多方计算领域最重要和最实用的技术之一，其允许参与的双方共同计算他们私有集合的交集，计算结束后一个或两个参与用户可以得到交集结果。自 Agrawal 等人[29]提出 PSI 的概念以来，众多学者做了一系列工作来构建 PSI 协议。目前对于 PSI 协议的设计分为两个方向，协议执行结束后只有一方得到交集计算结果的非公平性 PSI 协议和协议执行结束后参与双方同时得知交集计算结果的公平性 PSI 协议。

Freedman 等人[30]于 2004 年通过不经意多项式求值和同态加密，首次构造出一种只有一方能得知交集结果的 PSI 协议。但该协议要求参与者的集合元素取自于一个大指数域，不利于现实生活中的应用。在此之后，为了提高效率和安全级别，基于混淆电路[31]、不经意伪随机函数[32]、Bloom 过滤器[33]、不经意传输以及云计算等工具的非公平性 PSI 协议陆续被提出。Huang 等人利用混淆电路设计了一个 PSI 协议[34]，可适用于拥有不同集合元素个数的集合求交集计算。此外，在该 PSI 协议中添加一个简单的信息审计机制，即可直接用于执行任意复杂的安全计算。Pinkas 等人基于混淆电路和不经意可编程伪随机函数（Oblivious Programmable Pseudo-Random Functions，OPPRF）实现了具有线性通信复杂性的 PSI 协议[35]，在效率方面相较之前基于混淆电路的 PSI 协议较优。之后，Pinkas 等人[36]又基于可扩展的不经意传输协议构造了不经意伪随机函数（Oblivious Pseudo-Random Functions，OPRF），之后将 OPRF 与 hash 算法结合提出一个 PSI 协议，并在半诚实敌手模型下证明了该协议的安全性。Kavousi 等人[37]提出一个

PSI 协议，该协议以 OPRF 和乱码 Bloom 过滤器为主要组件，避免了昂贵的计算操作，具有很高的可扩展性。[38]中的协议允许用户将其私有数据集存储在云服务器上，并将交集的计算委托给服务器。此外协议在决策 Diffie Hellman（DDH）假设下设计了一种新的基于恶意敌手模型的外包型 PSI 构造，是第一个在没有交互式设置的情况下解决恶意敌手模型下 PSI 问题的协议。Abadi 等人[39]设计的 PSI 协议也使用云服务器存储私有数据集。协议主要由哈希表和多项式构造完成，不存在任何公钥加密操作。然而，各方执行协议时都需要事先建立安全通道，否则攻击者很容易窃听各诚实参与方之间传输的信息。文献[40]在[39]的基础上提出一种改进的基于云服务器的 PSI 方案。该方案不需要任何安全通道，在保密性和复杂性方面均优于[39]中的方案。以上 PSI 协议均属于只有一方得知交集计算结果的不公平的 PSI 协议。

针对参与双方能够同时得知交集计算结果的公平性 PSI 协议的研究始于 2005 年 Kissner 和 Song 等人提出的第一个公平性 PSI 协议[41]。该协议基于多项式的数学特性，并使用同态加密以密文形式计算两个参与者私密集合的交集。协议的公平性取决于构造协议时使用的门限加密方案的公平性。Camenisch 等人提出了另一种基于 Camenisch-Lysyanskaya 签名和不经意多项式的公平的认证集 PSI 协议[42]。但该协议的计算开销与集合元素个数之间呈二次线性关系，计算复杂度相对较高。该协议中参与者之间的公平性通过公平交换方案实现。但是，如果输入信息没有经过验证，无法参与协议的执行。Kim 等人将素数表示技术与门限加法同态加密相结合，在半诚实安全模型中首次实现了线性复杂度的公平性 PSI 协议[43]。协议中参与双方之间的公平性由门限解密的性质来实现。Dong 等人在同态加密和不经意多项式的基础上提出了一个具有半诚实仲裁器的公平性 PSI 协议[44]。协议中的仲裁器可以处理冲突，但不能知道用户的私有输入和输出。Debnath 等人[45]构造出一个基于合数阶群的恶意模型下安全的交互性不经意伪随机函数（mutual OPRF， mOPRF），以维护 PSI 协议的公平性，并采用同态加密算法来保护数据的隐私性。然而，由于该协议是基于合数阶群进行构造的，其复杂度高达 $48N$（$N$ 为参与者隐私集合中的元素个数），相对基于素数阶群构造的协议较高。在文献[45]的基础上，Debnath 等人[46]提出了另一种使用素数阶群的 PSI 协议。协议采用乘法同态加密 ElGamal 和分布式 ElGamal 密码体制[47]保证数据的安全性，使用离线半诚实仲裁器实现公平性。但为了保证恶意模型下的安全性，采用了可验证的 Cramer-Shoup 密码[48]系统，使得协议更加复杂。[49]中的 PSI 协议同样是基于素数阶群进行构造的, 协议使用同态加密算法来保证数据的安全性，并结合半诚实离线仲裁器来实现两个参与者之间的公平性。然而，协议整体执行的计算开销为 $23N+7$，复杂度仍然较高。

### 1.2.2 多方集合交集计算

对于多方之间的 PSI 协议，最早是由 Lai 等人[50]提出的。但在该协议，各参与方集合中 Bloom 过滤器编码的部分是以明文形式发送给其他参与方的，因此各参与方的隐私集合信息在协议执行时将会被泄漏给相关的参与者。之后对多方 PSI 协议的研究可分为两种，一种为协议的各参与者在执行协议前预先设定一个共有集合，所有参与者的集合元素均来自其中，从而方便进行交集计算；另一种为不预先设定共有集合，各参与者需首先对集合元素进行处理来进行多方交集计算。

在预先设定一个共有集合的多方 PSI 协议的研究工作中，李顺东等人[51]预先设定所有参与者共同拥有的集合，并设计一种 0-r 编码方式将每个参与者集合中的元素编码为 0 或者随机数，同时结合哥德尔编码和 ElGamal 公钥加密算法，构造出一个能够抵抗合谋攻击的多方 PSI 方案。窦家维等人[52]同样设定所有参与者的集合元素均属于一个共有的集合，以编码的方式将集合元素转化为 1 或者随机元素，同时利用同态加密算法，将集合安全计算问题转化为相应的数组安全计算问题，从而实现多个参与方之间的集合交集计算。Badrinarayanan 等人[53]以多项式和门限全同态加密算法为基础构建了一个具有次线性通信复杂性的门限多方 PSI 协议，其中所有参与者的集合元素均来自一个共同的有限域。在较弱的阈值加性同态加密假设下，协议所有参与方拥有的私密集合的并集和交集的并集最大相差值为门限值数。以上协议的参与方均在协议执行前沟通确定了一个共有集合，且保证每个参与方的隐私集合是属于该共有集合的。这在多方集合交集计算中方便了对多个集合的元素进行处理，减少了协议的计算开销。但在实际应用中，多个数据拥有者在不泄露隐私的情况下确定一个包含所有交集信息的共有集合是相当复杂且难以实现的。

对于不预先设定共有集合的多方 PSI 协议，Hazay 等人[54]以星形网络拓扑结构为基础，利用已知的双方 PSI 协议，安全地实现多个参与者之间的集合交集计算功能。协议指定一个参与方与其他所有参与方进行通信，最大限度地减少了通信通道的使用，但该协议的通信复杂性随着参与方的数量而变化。Miyaji 等人[55]基于 Bloom 过滤器和 ElGamal 加密算法提出了一个实用型多方 PSI 协议。协议中各参与方拥有的隐私数据集的大小是独立于其他参与方的，且由每个参与方执行的计算复杂度与参与者的人数无关。但该协议不能防止参与者之间的共谋攻击，无法完全保证数据的隐私性。在文献[58]的基础上，Bay 等人[56]又基于 Bloom 过滤器和门限同态公钥加密，提出了一个在半诚实模型中实现安全性的多方 PSI 协议。协议通过让每个参与方单独对集合元素进行随机化来解决参与者之间的共谋问题。此外，该多方 PSI 协议可进一步改进为门限多方 PSI 协

议，即返回门限数量的集合中的所有交集元素。

### 1.2.3 集合交集势计算

作为隐私保护集合交集问题的一种限制性更强的变体，隐私集合交集势（PSI-CA）问题中持有私有集合的双方联合计算交集的大小，无法获得其他任何信息。Cristofaro 等人[57]基于 Diffee-Hellman 密钥交换提出了一个高效的 PSI-CA 协议，其计算和通信复杂度与集合大小成线性关系，此外协议引入了授权集合交集势的概念，即客户端的输入信息须预先被授权才能执行 PSI-CA 协议，从而防止客户端窃取服务器集合中的元素。Debnath 等人[58]利用 Bloom 过滤器，提出了一个授权 PSI-CA 协议，协议中客户端的输入集合信息同样需要可信第三方授权，来防止服务器的元素被窃取，并证明该授权 PSI-CA 协议在二次剩余假设下是安全的。Davidson 等人[59]使用 Bloom 过滤器数据结构和加法同态加密设计了一个具有线性计算和通信复杂性的 PSI-CA 协议，但在计算过程中会泄露参与双方集合的大小。Tajima 等人[60]基于完全同态加密和 Bloom 过滤器，提出了外包的 PSI-CA 协议。但协议要求半诚实的云服务器不与任何参与方勾结，忽略了现实生活中云服务器与参与方之间存在的合谋攻击。2020 年，Lv 和 Ye 等人[61]设计了非平衡私有数据集情况下的 PSI-CA 协议，即参与双方中接收方拥有私密集合中元素的数量远远小于发送方拥有私密集合中元素的数量。此外协议在使用低功耗移动物联网设备时，通过 Bloom 过滤器，接收方可以比使用对发送方私有数据集进行加密的方法更容易计算出输出结果。上述协议在执行结束后均只有一方能够得知交集势的计算结果，均为非公平性 PSI-CA 协议。

对于公平的 PSI-CA 协议，Debnath 等人在文献[46]中提出一个公平且高效的 PSI-CA 协议，其在判定性 Diffifie-Hellman 假设[62]下针对恶意敌手是安全的。协议使用可信的第三方来实现公平性，但在现实生活中，不存在完全可信的第三方，其可能不忠或被收买，因此构造高效的 PSI-CA 协议且保持公平性仍然是一个具有挑战性的问题。Debnath[63]等人通过使用离线的第三方仲裁器解决了这个问题，协议中该仲裁器被假定是半可信的，且无法访问参与方的隐私数据信息。Debnath 等人又在文献[47]中将提出的公平性 PSI 协议扩展为 PSI-CA 协议，同样由离线的半诚实仲裁器来维护协议的公平性。基于文献[63]，Debnath 等人[64]使用 Bloom 过滤器，提出来一种能够隐藏参与集合大小的公平的 PSI-CA 协议，将计算开销降为线性并对协议在随机谕言机模型下的安全性进行了证明。

## 1.3 本文主要工作（Main Contributions of This Paper）

本文以安全多方计算方面的集合交集问题为研究主题，在分析国内外研究现状之后，针对两方之间的隐私集合交集计算问题进行深入研究，并对其变体问题隐私集合交集势计算和扩展问题多方之间的隐私集合交集计算进行深入分析后，从公平性、高效性和隐私性三个方面出发，致力于设计出能满足两方之间公平性的高效两方隐私集合交集协议和两方隐私集合交集势协议，以及更加安全高效的多方隐私集合交集协议。本文主要的研究工作如图 1-2 所示。

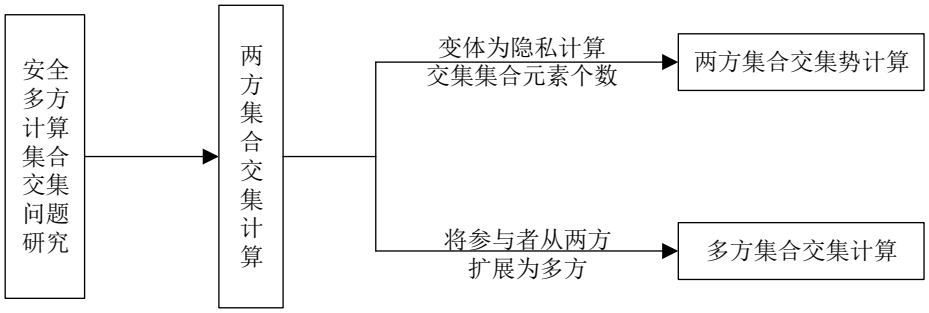



**图 1-2 本文主要研究工作**
**Fig.1-2 The main research work of this paper**

具体的研究内容为以下三个方面：

（1）针对两方之间的隐私集合交集计算问题，构造出一个外包的公平两方隐私集合交集协议。协议改进了文献[36]中预处理集合元素的方法，增加了存储在每个哈希桶中元素的数量，并取消额外存储设备的使用，这样只需对两个哈希表中的元素以桶对桶的方式比较就可以计算交集，进一步减少了集合元素的比较次数，降低了协议的计算成本。此外，采用 ElGamal 门限加密算法加密哈希表中的元素，以确保参与双方哈希桶内的元素可在密文的形式下进行比较，使得用户可以正确安全地计算交集，实现了协议的数据隐私性。最后，引入一个半诚实的服务器作为第三方，将集合元素的比较工作转移到服务器上，进一步降低了用户的计算负担，同时允许两个参与用户并行处理集合元素，确保在协议执行结束后，双方可以同时得知交集的结果，实现了协议的公平性。

（2）针对两方之间的隐私集合交集势计算问题，构造出一个公平的两方隐私集合交集势协议。协议中参与双方分别使用布谷鸟哈希算法和简单哈希算法将集合元素映射入相应的哈希桶中，从而将每个哈希桶中的集合元素分为一类来预处理元素，这样即可将一方哈希桶内的元素与另一方相应哈希桶内的元素进行对比，减少双方所有集合元素一对一进行比较时的次数，降低协议的计算成本。同时，采用基于 ElGamal 的可交换加密算法对双方存储在哈希表中的集合元素加密，根据可交换加密中使用不同用户的公钥

对消息明文进行多次不同顺序的加密后对应的密文仍相等的特性，确保参与双方可在密文形式下对哈希桶内的元素进行比较，在正确计算集合交集势的同时能够保证数据的隐私性。此外，协议的执行由双方并行完成，保证了双方之间的公平性，使得两个参与者能够同时得知交集势计算结果。最后，从计算开销、通信开销和是否实现公平性三个方面与现有交集势计算协议进行比较，结果表明本文所构造协议的计算和通信复杂度较低，协议整体性能较优。

（3）针对多方之间的隐私集合交集计算问题，构造出一个安全高效的多方隐私集合交集协议。经过对现有多方隐私集合交集计算协议的分析总结，发现目前对于多个参与方之间的隐私计算问题，多数协议会预先设定一个所有参与方共有的集合，各方私有集合均为共有集合的子集来方便对集合元素预先处理从而能更加快速便捷的计算多方之间的交集。然而，在实际应用中，无法在不泄露任何信息的情况下准确设定多个用户之间的共有集合。针对这种情况，所构造协议中令 $P_1$ 使用布谷鸟哈希，其他参与方 $P_2,\ldots,P_n$ 使用简单哈希算法来预先对集合元素进行处理，使得 $P_1$ 分别与 $P_2,\ldots,P_n$ 以哈希桶为单位执行等值比较协议。同时，采用 $(n,n)$ 门限加密算法，保证所有参与者共同掌握交集元素信息且不泄露参与者集合中的任何信息。最后，经过理论和实验分析，表明所构造协议在没有预先设定共有集合的情况下达到与设定共有集合的多方隐私集合交集协议相近的复杂度，远远低于其他同样没有预先设定共有集合的多方隐私集合交集协议。

## 1.4 论文组织结构（Paper Organization）

本文主要分为以下六个章节，具体安排如下：

第一章是绪论。主要说明了研究安全多方计算集合交集问题的目的和意义，介绍了国内外对于安全两方隐私集合交集计算、安全两方隐私集合交集势计算和安全多方隐私集合交集计算问题的研究现状。

第二章是预备知识。对本文所涉及的基础知识进行介绍，主要为安全多方计算和集合交集问题的定义、密码学方面的相关知识、简单哈希和布谷鸟哈希两种哈希映射算法以及协议使用的安全分析模型。

第三章构造了一个外包的公平两方隐私集合交集计算协议。给出基于服务器使用ElGamal门限加密算法和哈希映射算法构造协议的具体过程，对协议的正确性和安全性进行了证明，并从理论和具体实验两方面对协议的性能进行了分析。

第四章设计了一个公平的两方隐私集合交集势计算协议。描述了使用哈希映射算法

和可交换加密算法构造协议计算交集势的过程，给出了协议的正确性和安全性证明过程并对协议的性能从理论和实验两方面进行了分析。

第五章提出了一个多方的隐私集合交集计算协议。对采用哈希映射算法、等值比较协议和门限加密算法为基本模块构建协议计算多方之间集合交集的过程进行了详细描述，给出了协议的正确性和安全性证明以及理论和实验两方面的性能分析。

第六章是总结与展望。对本文的研究工作进行全面的总结，并指出下一步的研究方向。

# 2 预备知识

# 2. Preliminaries

构造具体方案之前，我们在本章预先对方案中所涉及的基础理论知识进行了介绍。其中 2.1 节介绍安全多方计算方面的相关知识，2.2 节介绍密码学方面的相关知识，2.3 节介绍哈希映射算法，2.4 节介绍安全分析模型，最后 2.5 节对本章内容进行总结。

## 2.1 安全多方计算相关知识（Knowledge of Secure Multi-party Computing）

### 2.1.1 定义

安全多方计算是分布式环境中一种保护数据安全隐私的多方计算算法，旨在解决一组互不信任的参与方之间的安全协同计算问题，在不泄露原始数据的情况下为数据需求者提供多方协同计算能力。根据参与方个数的不同，可具体分为安全的两方计算和多方计算。

（1）两方计算

安全两方计算协议[65]允许参与双方对他们的私有数据进行隐私计算。协议执行结束后，一方或双方除了计算结果外，无法得知对方的任何输入信息。其基本模型如图 2-1 所示。

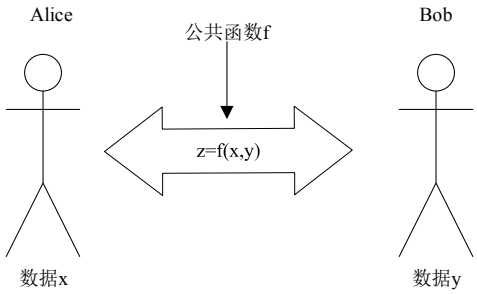



**图 2-1 安全两方计算基本模型图**
**Fig.2-1 Basic model diagram of security two-party computation**

具体过程描述为：假设存在两个参与者 Alice 和 Bob，且 Alice 拥有数据 $x$，Bob 拥有数据 $y$；参与双方确定一个公共函数 $f(\cdot)$，计算 $z = f(x,y)$ 并公布计算结果 $z$。此外在执行安全两方计算协议时，一般由加密算法来保证数据的隐私性。

（2）多方计算

安全多方计算协议指多个参与者分别输入自己的私有数据进行协同计算，且要求每个参与者除计算结果外无法得知其他参与者的任何信息。其基本模型如图 2-2 所示。具

体描述为：

假设存在 $n$ 个参与者 $P_1, P_2, \ldots, P_n$，每个参与者 $P_i(i=1,\ldots,n)$ 拥有私密数据 $x_i$，确定函数 $f(\cdot)$ 并共同计算 $f(x_1, x_2, \ldots, x_n) = (y_1, y_2, \ldots, y_n)$。协议执行过程中同样由加密算法来保证数据的隐私性。

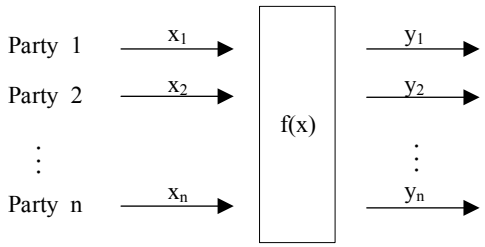



**图 2-2 安全多方计算基本模型图**

**Fig.2-2 Basic model diagram of security multi-party computation**

## 2.1.2 计算模型

（1）参与者类型

根据参与者的具体行为，可将安全多方计算协议中的参与者分为以下三种类型：

诚实参与者：诚实的参与者在整个安全多方计算协议执行过程中严格遵循协议的规则来执行协议，对自己的输入信息以及获得的输出信息严格保密。

半诚实参与者：半诚实的参与者同意严格遵循协议的规则来执行协议，但会将自己的输入信息以及获得的输出信息发送给攻击者。

恶意参与者：在整个安全多方计算协议的执行过程中，恶意的参与者不仅会将自己的输入信息以及获得的输出信息发送给攻击者，还可能篡改协议的输入信息、中间的输出信息和输出结果，甚至可能终止协议的执行。

（2）具体计算模型

根据参与者的类型，可将安全多方计算协议的计算模型分为两类：

半诚实模型：若协议的参与者均为诚实的或者半诚实的，则该计算模型为半诚实模型。

恶意模型：若协议中存在恶意的参与者，则该计算模型为恶意模型。恶意模型的构建相对半诚实模型而言更为复杂，故对于半诚实模型下的安全多方计算的研究较多。

## 2.1.3 安全需求

考虑到安全多方计算协议执行过程中存在不诚实参与者不诚实行为的可能性，安全多方计算需满足以下安全需求：

隐私性：除了自己的输入信息以及协议的输出结果之外，参与方无法获得其他参与者的任何信息。

正确性：每个参与者收到的输出结果都是正确的。

输入的独立性：每个参与者的输入信息都是独立的。

输出的传递性：半诚实以及恶意的参与者不能阻止诚实的参与者得到协议的输出结果。

公平性：所有参与者在协议执行结束时能同时得到计算结果。

## 2.2 密码学相关知识（Knowledge of Cryptography）

### 2.2.1 ElGamal 加密算法

（1）ElGamal 加密具体算法

ElGamal 加密算法[49]主要由密钥生成（KeyGen）、加密（Encrypt）和解密（Decrypt）三个算法构成。

KeyGen：给定安全参数 $\lambda$，以及循环群 $G$，其阶为 $p$，生成元为 $g$。随机选取 $sk \leftarrow \mathbb{Z}_p$，计算 $pk = g^{sk} \bmod p$，则 $sk$ 为私钥，$pk$ 为公钥。

Encrypt：对于给定的消息 $m \in G$，选取一个随机数 $r$，计算密文

$$E(m) = (c_1, c_2) = (g^r, m \cdot pk^r)$$

Decrypt：对于给定的密文 $E(m)$，计算 $g^m = \dfrac{c_2}{(c_1)^{sk}} = \dfrac{m \cdot pk^r}{(g^r)^{sk}}$。

满足乘法同态性 $E(m_1) \times E(m_2) = E(m_1 \cdot m_2)$，即：

$$\begin{aligned} E(m_1) \times E(m_2) &= (g^{r_1}, m_1 \cdot pk^{r_1}) \times (g^{r_2}, m_2 \cdot pk^{r_2}) \\ &= (gE(m_1)^{r_1+r_2}, (m_1 \cdot m_2) \cdot pk^{r_1+r_2}) \\ &= E(m_1 \cdot m_2) \end{aligned}$$

（2）ElGamal 加密算法变体

相对于 ElGamal 加密算法，ElGamal 加密算法的变体[66]在加密时将 $m \cdot pk^r$ 改为 $g^m \cdot pk^r$。其同样由密钥生成（KeyGen）、加密（Encrypt）和解密（Decrypt）三部分构成。

KeyGen：给定安全参数 $\lambda$，以及循环群 $G$，其阶为 $p$，生成元为 $g$。随机选取 $sk \leftarrow \mathbb{Z}_p$，计算 $pk = g^{sk} \bmod p$，则 $sk$ 为私钥，$pk$ 为公钥。

Encrypt：对于给定的消息 $m \in G$，选取一个随机数 $r$，计算密文

$$E(m) = (c_1, c_2) = (g^r, g^m \cdot pk^r)$$

Decrypt：对于给定的密文 $E(m)$，计算 $g^m = \dfrac{c_2}{(c_1)^{sk}} = \dfrac{g^m \cdot pk^r}{(g^r)^{sk}}$。

满足加法同态性 $E(m_1)\times E(m_2)=E(m_1+m_2)$，即：

$$\begin{aligned}E(m_1)\times E(m_2)&=(g^{r_1},g^{m_1}\cdot pk^{r_1})\times(g^{r_2},g^{m_2}\cdot pk^{r_2})\\&=(g^{r_1+r_2},g^{m_1+m_2}\cdot pk^{r_1+r_2})\\&=E(m_1+m_2)\end{aligned}$$

### 2.2.2 加密体制

（1）门限加密体制

门限加密体制[67]是一种以多方安全计算的方式进行的密码操作，可以有效保护需要多方授权才能使用的隐私数据。其一般由初始化（Initialize）、加密（Encrypt）、各自分别解密（RespDec）和联合解密（ComDec）四部分构成。

Initialize：$n$ 个参与者共同确定门限值 $t$，获取各自的私钥 $x_i(i\in[1,n])$ 并联合计算公钥 $pk$。

Encrypt：使用公钥 $pk$ 对给定消息 $m$ 加密，得到密文 $E(m)$。

RespDec：每个参与者分别使用自己的私钥 $x_i$ 对 $E(m)$ 进行解密，生成解密碎片 $D_i(m)$。

ComDec：聚合解密碎片，只有聚合不少于 $t$ 个不同的解密碎片 $D_i(m)$ 时才能完全解密 $E(m)$。

这种 $n$ 个参与者中至少需 $t$ 个人合作才可以完全解密密文的密码体制称为 $(n,t)$ 门限加密体制。

（2）可交换加密体制

在可交换加密体制[68]中，允许使用不同用户的公钥对消息明文进行多次加密，且消息在经过不同顺序的公钥加密后，对应的密文仍相等。

根据文献[69]中的定义，令 $M$ 为明文空间，$K$ 为密钥空间，则可交换加密算法中存在双射关系：$f:M\times K\to M$。即对于给定的消息 $m\in M$，随机选取两个密钥 $a,b\in K$，存在可交换加密算法：$f_a(f_b(m))=f_b(f_a(m))$。其一般流程如下：

Initialize：运行密钥生成算法，参与者 $P_1$ 获取密钥对 $(pk_1,sk_1)$，$P_2$ 获取密钥对 $(pk_2,sk_2)$。

Encrypt：对于给定消息 $m$，$P_1$ 使用公钥 $pk_1$ 对 $m$ 加密得到密文 $E_{pk_1}(m)$，$P_2$ 使用公钥 $pk_2$ 对 $m$ 加密得到密文 $E_{pk_2}(m)$。

Re-Encrypt：$P_2$ 对密文 $E_{pk_1}(m)$ 重加密得到 $E_{pk_2}(E_{pk_1}(m))$，$P_1$ 对密文 $E_{pk_2}(m)$ 重加密得到 $E_{pk_1}(E_{pk_2}(m))$，则有 $E_{pk_2}(E_{pk_1}(m))=E_{pk_1}(E_{pk_2}(m))$。

Decrypt：对于密文 $E_{pk_2}(E_{pk_1}(m))$（或 $E_{pk_1}(E_{pk_2}(m))$），$P_1$ 使用私钥 $sk_1$ 半解密为

$E_{pk_2}(m)$，$P_2$再使用私钥$sk_2$将$E_{pk_2}(m)$完全解密为明文$m$。

### 2.2.3 不经意传输

不经意传输（Oblivious transfer，OT）是由发送方和接收方组成的两方密码协议[70]，广泛应用于安全的两方和多方计算协议中，主要有 1-out-of-2 和 1-out-of-n 两种基本 OT 形式：

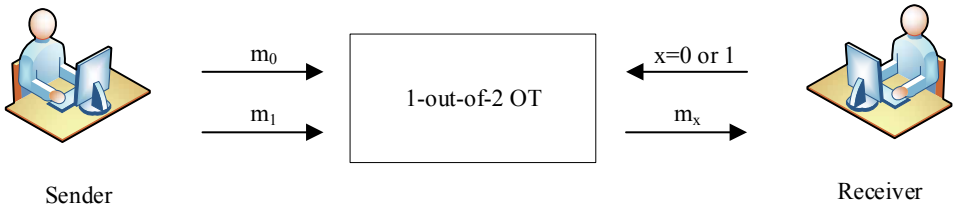



**图 2-3 1-out-of-2 OT 协议**
**Fig.2-3 1-out-of-2 OT protocol**

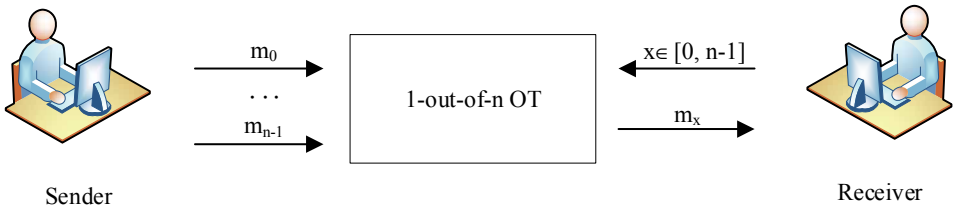



**图 2-4 1-out-of-n OT 协议**
**Fig.2-4 1-out-of-n OT protocol**

在 1-out-of-2 OT 协议中，发送方输入两个消息$(m_0, m_1)$，接收方输入一个选择比特$b$得到消息$m_b$；在 1-out-of-n OT 协议中，发送方输入$n$个消息$(m_0, \ldots, m_{n-1})$，接收方输入一个选择比特$b$得到消息$m_b$。无论是 1-out-of-2 OT 协议还是 1-out-of-n OT 协议，在执行结束后，接收方除$m_b$外无法得知发送方的其他任何消息，发送方也无法得知接收方接收的具体信息。

（1） 基于 OT 的不经意伪随机函数

基于 OT 协议构造的不经意伪随机函数(Oblivious Pseudo-Random Function，OPRF)是一个在两方之间执行的协议[71]，其功能如下：

1) 接收方输入元素$x$；

2) 随机选择一个密钥$k$；

3) 发送$k$给发送方，发送$F(k,x)$（密钥$k$加密后的元素）给接收方。

在理想状态下，OPRF 执行结束后，发送方除密钥$k$外无法得知任何额外信息，接收方除$F(k,x)$外得不到任何额外信息。

（2）基于 OT 的等值比较协议

基于OT协议构造的等值比较协议[72]是进行集合交集计算的一个基础协议，主要用来比较两个输入元素是否相等。协议执行结束后双方共同享有元素比较的结果，单独一方无法得知。其具体功能如下：

1） 发送方输入元素 $x$，接收方输入元素 $y$；

2） 若 $x=y$，则令 $e\leftarrow 0$，否则令 $e\leftarrow 1$；

3） 选取随机数 $r$ 发送给发送方，$e+r$ 发送给接收方。

在理想状态下，等值比较协议执行结束后，参与双方只能得知共享的元素比较结果，无法得知对方的输入信息。

## 2.3 哈希映射算法（Hash Algorithm）

### 2.3.1 简单映射

采用简单哈希映射算法[73]将集合元素映射到哈希表的 $m$ 个哈希桶中，集合中每个元素需存储两次，元素存储步骤如下：

1）确定两个哈希函数 $h_1,h_2:\{0,1\}^*\rightarrow\{1,\ldots,m\}$；

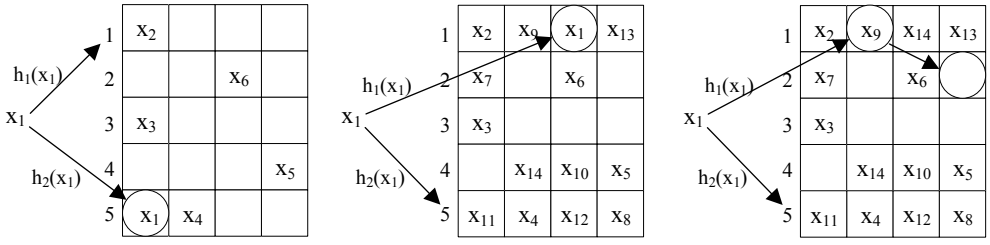
2）对于待存储的集合元素 $x$，计算其对应的哈希桶号 $h_1(x)$ 和 $h_2(x)$；

3）在哈希表的 $m$ 个哈希桶中，寻找桶号为 $h_1(x)$ 和 $h_2(x)$ 的两个哈希桶；

4）分别在哈希桶 $h_1(x)$ 和 $h_2(x)$ 中存入元素 $x$。

### 2.3.2 布谷鸟映射

采用布谷鸟哈希映射算法[74]将集合元素映射到哈希表的 $m$ 个哈希桶中，集合中每个元素只存储一次，元素在插入哈希桶时的情况举例如图 2-5 所示。




（a）两个桶均未满　（b）一个桶满一个桶未满　（c）两个桶均已满

**图 2-5 元素在插入哈希桶中时的三种情况举例**

**Fig.2-5 Examples of three situations when an element is inserted into hash bin**

具体映射方式如下：

1）确定两个哈希函数 $h_1,h_2:\{0,1\}^*\rightarrow\{1,\ldots,m\}$（与简单哈希映射中的相同）；

2）对于待存储的集合元素 $x$，计算其对应的哈希桶号 $h_1(x)$ 和 $h_2(x)$；

3）在哈希表的 $m$ 个哈希桶中，寻找桶号为 $h_1(x)$ 和 $h_2(x)$ 的两个哈希桶，执行以下步骤：

①若哈希桶 $h_1(x)$ 和 $h_2(x)$ 中均未存满，随机选择其中一个存入元素 $x$；

②若两个哈希桶中一个已满一个未满，则将元素插入未满的桶中；

③若两个哈希桶均已存满元素，则随机选择一个哈希桶，将其中任意一个原有数据 $y$ 移出，将 $x$ 插入 $y$ 空出来的位置，被移出的数据 $y$ 重新从步骤 2）开始执行来插入哈希表中。

## 2.4 安全分析模型（Security Analysis Model）

### 2.4.1 判定性 Diffie-Hellman(DDH)假设

定义 1. 给定阶为 $p$ 的循环群 $G$，其中 $p$ 是一个大素数，$g$ 为 $G$ 的生成元，且 $a,b,c \leftarrow \mathbb{Z}_p$。则对于随机四元组 $R=(g,g^a,g^b,g^c)\in G^4$ 和 DDH 四元组 $D=(g,g^a,g^b,g^{ab})\in G^4$ 两个分布，任意敌手 $\mathcal{A}$ 区分分布 $R$ 和 $D$ 的优势 $\mathrm{Adv}_{\mathcal{A}}^{DDH}(\lambda)=\left|\Pr[\mathcal{A}(R)=1]-\Pr[\mathcal{A}(D)=1]\right|$ 是可忽略的，其中 $\lambda$ 为安全参数。

### 2.4.2 敌手模型

在隐私保护集合交集计算协议中，存在概率多项式时间（Probability-Polynomial Time，PPT）的敌手 $\mathcal{A}$，参考文献[75]对其模型和能力进行了定义。在该模型中，敌手 $\mathcal{A}$ 必须忠实地执行协议，但可以根据协议执行过程中获得的信息来试图了解更多信息。敌手的具体能力见表 2-1。

**表 2-1 敌手能力分析表**

**Tab.2-1 Analysis of the adversary's capabilities**

| Types | Descriptions |
|---|---|
| C1 | 敌手能够得知信道中传送的消息 |
| C2 | 敌手可腐败参与者中的其中一个来获取其私钥和隐私数据 |
| C3 | 敌手可与服务器合谋从而获取服务器上存储的所有信息 |
| C4 | 敌手可与参与者中的其中一部分合谋从而获取他们掌握的所有信息 |

### 2.4.3 基于随机谕言机的安全模型

隐私保护集合交集计算协议的安全性定义需满足协议的参与者除最后的计算结果外无法得知其他参与者的任何信息。参考文献[76]中的安全模型，在随机谕言机（Random oracle）模型下使用选择明文攻击下的不可区分性（IND-CPA）游戏来证明隐私集合交集计算协议安全性。如果敌手腐败一个参与者后，在概率多项式时间内仍无法根据 oracle 询问中得到的信息来准确判断密文的具体明文信息，则说明隐私集合交集计算协议是安

全的。游戏过程如下：

Initialize：确定挑战用户实体集合$U$以及两个等长的消息$M_0$和$M_1$。

Setup：接收安全参数$\lambda$，挑战者运行密钥生成算法生成公私钥对，公开公钥，保留私钥。

Queries：敌手$\mathcal{A}$通过 oracle 询问与实体$U$进行交互，oracle 询问对敌手$\mathcal{A}$在实际攻击中的能力进行了模拟。协议中所涉及到的 oracle 询问如下：

1) Execute ($U$)：该询问对敌手的窃听能力进行了模拟（即表 2-1 中的 C1），将协议实际执行过程中实体间传送的所有消息发送给$\mathcal{A}$。
2) Send ($U,a,M$)：敌手$\mathcal{A}$发送消息$M$给实体$U_a^i$（$i$表示在协议并发执行中的第$i$次，$a$为执行第$i$次协议的实体集合$U^i$中第$a$个参与者），$U_a^i$根据协议对$M$进行处理，将处理后的信息返回给$\mathcal{A}$。
3) Corrupt ($U,a$)：返回$U_a^i$的私钥和其隐私输入数据。该询问模拟了敌手的腐败能力（即表 2-1 中的 C2），且$\mathcal{A}$只能腐败参与者中的一个。
4) Collude ($S$)：该询问模拟了敌手的合谋能力（即表 2-1 中的 C3），返回服务器上存储的所有信息。
5) Collude ($V$)：该询问同样模拟了敌手的合谋能力（即表 2-1 中的 C4），$V \subset U$，返回集合$V$中所有参与者拥有的信息。

Challenge：敌手$\mathcal{A}$将消息$M_0$和$M_1$发送给挑战者，挑战者随机选取$b \leftarrow_R \{0,1\}$，加密$M_b$并返回密文$c$。

Test：$\mathcal{A}$根据上述 oracle 询问中得到的信息输出对于$b$的猜测$b'$，若$b'=b$则$\mathcal{A}$赢得游戏（记为事件$Succ$）。

敌手$\mathcal{A}$赢得 IND-CPA 游戏的优势定义为：

$$\mathrm{Adv}_{\mathcal{A}}(\lambda)=\Pr\left[Succ(\mathcal{A})\right]-1/2=\Pr\left[b'=b\right]-1/2 \quad (1)$$

定义 **2**. 若对于任意的 PPT 敌手$\mathcal{A}$，存在一个可忽略的函数$\varepsilon(\cdot)$使得$\mathrm{Adv}_{\mathcal{A}}(\lambda)=\Pr\left[Succ(\mathcal{A})\right]-1/2\le\varepsilon(\lambda)$，则称协议是 IND-CPA 安全的。

## 2.5 本章小结（Summary of This Chapter）

本章对方案中所涉及到的知识点进行了介绍。首先给出了安全多方计算的定义、计算模型以及安全需求；其次对密码学方面的相关知识进行了介绍，主要包括 ElGamal 加密算法、加密体制和不经意传输；接着对简单映射和布谷鸟映射两种哈希映射算法进行了介绍；最后简要介绍了两种安全模型，基于随机谕言机的安全模型和基于模拟器的半诚实安全模型。

# 3 外包的公平两方隐私集合交集协议

# 3. Outsourced Fair two-party Privacy Set Intersection protocol

PSI 计算作为一种允许数据信息秘密共享的高效加密技术，在充分利用数据进行交集计算时，可以确保存储在服务器上的数据的安全性。因此，成为大数据环境下解决共同数据隐私共享问题的重要研究对象。本章对外包环境下两方之间的隐私集合交集问题进行研究，提出一个基于云服务器的公平性两方 PSI 协议，来解决两方隐私集合的交集计算问题以及两方之间的公平性问题。

## 3.1 引言（Introduction）

两方的 PSI 协议在两个参与者共同完成集合交集计算后，参与的一方或双方可获得交集结果，且能保证各自数据集合的隐私性，能够在不泄露隐私数据的情况下实现双方信息共享。但对 PSI 问题的研究仍存在以下问题：

（1）用户双方之间的不公平性。大多数 PSI 协议在执行结束后，只有一个参与用户能够获得交集的计算结果，是不公平的 PSI 协议，无法满足某些具体应用场景中，两个用户希望同时得到交集结果的需求。如电子医疗信息匹配过程中，两个用户将各自的病历信息作为输入集合来执行 PSI 协议，双方希望协议执行结束后能够同时得到交集结果，来实现患者之间的医疗信息共享。

（2）公平性 PSI 协议的低效性。在少数的 PSI 协议当中，通过协议构造过程中使用的具有公平性的构造模块或者离线半诚实的第三方仲裁器，来保证两个参与双方之间获得结果的公平性，实现了双方能够同时得到交集结果的公平性 PSI 协议。然而，通过这些方法在实现公平性的同时却导致协议更加复杂，使得协议的计算复杂度较高，无法满足协议的高效性。

本章以哈希映射算法为基础，使用 ElGamal 门限加密算法保证数据的隐私性，同时结合云服务器，提出一个外包的公平两方隐私集合交集协议（OF-PSI），在实现公平性的同时降低了协议的计算复杂度，并在随机谕言机模型下证明了协议的 IND-CPA 安全性。

## 3.2 系统模型（System Model）

OF-PSI 协议中主要涉及到三个实体：两个参与用户 $P_1, P_2$ 和一个半诚实的服务器 $S$。其中 $P_1$ 拥有隐私数据集合 $X=\{x_1,\ldots x_{n_1}\}$，$P_2$ 拥有隐私数据集合 $Y=\{y_1,\ldots,y_{n_2}\}$。他们希

望通过服务器来计算 $X$ 和 $Y$ 的交集 $X \cap Y$，而不透露自己集合的任何信息。协议的系统模型如图 3-1 所示。

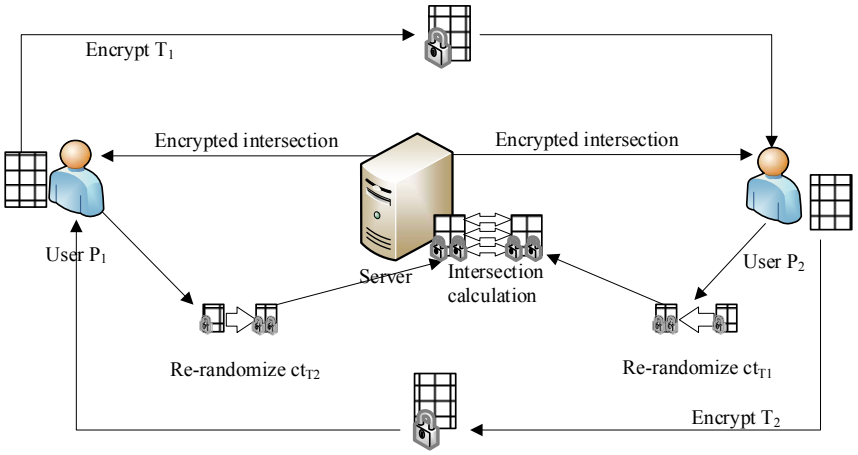



**图 3-1 OF-PSI 协议系统模型图**
**Fig.3-1 System model of OF-PSI protocol**

（1）哈希映射分类预处理集合元素

参考文献[36]，在协议 OF-PSI 中使用哈希映射的方式对集合元素进行分类预处理。参与者 $P_1$ 和 $P_2$ 商定使用两个哈希函数 $\{h_1, h_2\}:\{0,1\}^* \to \{1,\ldots,m\}$ 来生成每个集合元素在哈希表中的对应的两个哈希桶位置。其中 $P_1$ 使用布谷鸟哈希，将集合 $X$ 中每个元素 $x_i (i=1,\ldots,n_1)$，在哈希表 $T_1$ 中其对应的两个哈希桶 $h_1(x_i)$ 和 $h_2(x_i)$ 中选择一个来存入 $x_i$，每个哈希桶存储 4 个元素，根据文献[36]可知，此时哈希表利用率超过 90%；$P_2$ 使用简单哈希，将集合 $Y$ 中的每个元素 $y_j (j=1,\ldots,n_2)$，在哈希表 $T_2$ 中其对应的两个哈希桶位置 $h_1(y_j)$ 和 $h_2(y_j)$ 内均存储 $y_j$，每个哈希桶存储 $b$（$O(\log n_2 / \log\log n_2)$）个元素[77]。这样即对集合元素进行了分类，桶号相等的哈希桶内的元素为一类。

（2）ElGamal 门限加密集合元素

使用基于 ElGamal 的 $(2,2)$ 门限加密算法[78]在正确计算交集的同时保护用户集合信息的隐私。参与者 $P_a (a=1,2)$ 使用公钥 $pk$ 加密哈希表中的元素后，将其发送给另一方进行重随机化。结合哈希算法可知，对于 $T_1$ 哈希桶内存在的任意元素 $\forall x_i \in X$，如果有相同的元素 $y_j \in Y$ 存储在 $T_2$ 相应的哈希桶中，那么在这两个编号相同的哈希桶中必然有 $E^*(x_i) = E^*(y_j)$。因此，ElGamal 门限加密算法的使用确保了参与双方相等元素的加密结果相同，保证了能够以密文执行元素的比较操作。

（3）桶对桶求集合交集

引入服务器来进行元素的比较工作。参与者 $P_a (a=1,2)$ 将通过 ElGamal 门限加密后的哈希表发送到服务器，服务器在密文的情况下以桶对桶的方式比较两个哈希表中的元

素，即将$T_1$哈希桶内的元素与$T_2$相应的哈希桶内的元素进行比较。$m$个哈希桶中所有相等的密文元素即为密文形式下的交集$X \cap Y$。$P_1$和$P_2$联合对密文交集进行并行解密，即可同时得到明文交集$X \cap Y$。其中，$m$个哈希桶中的元素比较工作可并行进行，减少了OF-PSI协议的时间开销。

## 3.3 具体方案（Specific Scheme）

输入：$P_1$输入隐私数据集合$X=\{x_1,\ldots,x_{n_1}\}$，

$P_2$输入隐私数据集合$Y=\{y_1,\ldots,y_{n_2}\}$。

输出：交集$X \cap Y$

（1）初始化阶段

步骤 1-1：$P_1$和$P_2$分别运行 ElGamal 的 KeyGen 算法，生成各自的公私钥对$(pk_a,sk_a),a=1,2$，公布公钥$pk_a$，保留私钥$sk_a$，则系统公钥为$pk=pk_1\cdot pk_2$；

步骤 1-2：$P_1$和$P_2$确定哈希函数$H:\{0,1\}^* \to G$，商定哈希表中桶的数量$m$以及元素预处理哈希函数$h_1,h_2:\{0,1\}^* \to \{1,\ldots,m\}$。

（2）元素预处理阶段

步骤 2-1：对于$x_i \in X, i\in[n_1]$，$P_1$计算其在哈希表$T_1$中对应的哈希桶位置$h_1(x_i)$和$h_2(x_i)$，随机选择一个插入$x_i$；在集合$X$所有元素存入哈希表$T_1$后，使用随机元素对$T_1$中的空闲位置进行填充；

步骤 2-2：对于$y_j \in Y, j\in[n_2]$，$P_2$计算其在哈希表$T_2$中对应的哈希桶位置$h_1(y_j)$和$h_2(y_j)$，在两个位置均插入$y_j$；在集合$Y$所有元素存入哈希表$T_2$后，使用随机元素对$T_2$中的空闲位置进行填充。

（3）数据加密阶段

步骤 3-1：$P_1$以哈希桶为单位对$T_1$中的元素$x_i(i=1,\ldots,n_1)$进行加密，对于$k\in[1,m]$，选取随机数$r_1^k$计算$ct_{T_1^k}=E(x_i^k)=(g^{r_1^k},g^{H(x_i^k)}\cdot pk^{r_1^k})$，并以置换序列$s_1$将$ct_{T_1}=(ct_{T_1^1},\ldots,ct_{T_1^m})$置换后发送给$P_2$；

步骤 3-2：$P_2$以哈希桶为单位对$T_2$中的元素$y_j(j=1,\ldots,n_2)$进行加密，对于$k\in[1,m]$，选取随机数$r_2^k$计算$ct_{T_2^k}=E(y_j^k)=(g^{r_2^k},g^{H(y_j^k)}\cdot pk^{r_2^k})$，并以置换序列$s_2$将$ct_{T_2}=(ct_{T_2^1},\ldots,ct_{T_2^m})$置换后发送给$P_1$；

步骤 3-3：对于$k\in[1,m]$，$P_1$计算$ct_{T_2^k}^*=E^*(y_j^k)=(g^{r_1^k+r_2^k},g^{H(y_j^k)}\cdot pk^{r_1^k+r_2^k})$，对接收到的$ct_{T_2}$进行重随机化，并将$ct_{T_2}^*=(ct_{T_2^1}^*,\ldots,ct_{T_2^m}^*)$以置换序列$s_1$置换后发送给服务器$S$；

步骤 3-4：对于$k\in[1,m]$，$P_2$计算$ct_{T_1^k}^*=E^*(x_i^k)=(g^{r_1^k+r_2^k},g^{H(x_i^k)}\cdot pk^{r_1^k+r_2^k})$，对接收到的$ct_{T_1}$

进行重随机化，并将$ct_{T_1}^* = (ct_{T_1^1}^*, \ldots, ct_{T_1^m}^*)$以置换序列$s_2$置换后发送给$S$。

（4）交集计算阶段

步骤 4-1：$S$接收到的$ct_{T_1}^*$和$ct_{T_2}^*$后，对于$k \in [1,m]$，计算$ct_{I^k} = ct_{T_1^k}^* \cap ct_{T_2^k}^*$，密文形式下的交集即为$ct_I = ct_{I^1} \cup ct_{I^2} \cup \cdots \cup ct_{I^m}$，公布$ct_I$；

步骤 4-2：$P_1$和$P_2$对接收到的$ct_I$联合解密，分别得到明文交集$I = \{z_1, \ldots, z_w\}$。

## 3.4 方案分析（Scheme Analysis）

### 3.4.1 正确性分析

**定理 1.** OF-PSI 协议可以正确的计算两方集合的交集。

**证明：**若双方集合元素$X = \{x_1, \ldots x_{n_1}\}$和$Y = \{y_1, \ldots, y_{n_2}\}$在分类预处理的情况下，经过 ElGamal 门限加密后能够以桶对桶的形式正确计算出交集$X \cap Y$，则说明 OF-PSI 协议是正确的。

在元素预处理阶段，$P_1$使用布谷鸟哈希将集合$X$中每个元素$x_i (i = 1, \ldots, n_1)$在$h_1(x_i)$或$h_2(x_i)$中存储，$P_2$使用简单哈希将集合$Y$中每个元素$y_j (j = 1, \ldots, n_2)$在$h_1(y_j)$和$h_2(y_j)$中存储。若存在$x_i = y_j$，则有$h_1(x_i) = h_1(y_j)$，$h_2(x_i) = h_2(y_j)$，无论$x_i$选择$h_1(x_i)$或$h_2(x_i)$进行存储，$h_1(y_j)$和$h_2(y_j)$中均有$y_j$与之相对应。因此该预处理方法能够正确且完整的的对元素进行分类，即该阶段是正确的。

在数据加密阶段，经过$(2,2)$门限加密及重随机化后，哈希表$T_1$中元素信息转变为密文信息$ct_{T_1}^* = E^*(x_i^k) = (g^{r_1^k + r_2^k}, g^{H(x_i^k)} \cdot pk^{r_1^k + r_2^k})$，哈希表$T_2$中元素信息转变为密文信息$ct_{T_2}^* = E^*(y_j^k) = (g^{r_1^k + r_2^k}, g^{H(y_j^k)} \cdot pk^{r_1^k + r_2^k})$。若在第$k$个哈希桶中存在$x_i^k = y_j^k$，则有$E^*(x_i^k) = E^*(y_j^k)$，故该阶段是正确的。

在交集计算阶段，服务器$S$以桶为单位将$T_1$哈希桶中的密文元素与$T_2$相应哈希桶中的密文元素进行对比，得到：

$$\begin{aligned} ct_I &= ct_{I^1} \cup ct_{I^2} \cup \cdots \cup ct_{I^m} \\ &= (ct_{T_1^1}^* \cap ct_{T_2^1}^*) \cup (ct_{T_1^2}^* \cap ct_{T_2^2}^*) \cup \cdots \cup (ct_{T_1^m}^* \cap ct_{T_2^m}^*) \end{aligned}$$

$P_1$和$P_2$使用各自的私钥联合对$ct_I$进行解密即可同时得到明文交集$X \cap Y$，因此交集计算阶段也是正确的。

综上，OF-PSI 协议能够正确的计算出两方集合的交集。

### 3.4.2 安全性分析

**定理 2.** 令$G$是一个阶为大素数$p$的循环群，$\mathcal{A}$是在多项式时间$t$内攻击 IND-CPA 安全性的 PPT 敌手。设$\mathcal{A}$可发送少于$q_{exe}$次的 Execute 询问和$q_{send}$次的 Send 询问，最多

$q_h$ 次的 random oracle 询问，因此有：

$$\mathrm{Adv}_{OF-PSI}^{IND-CPA}(\mathcal{A}) \le q_h \mathrm{Adv}_{\mathcal{A}}^{DDH}(t + \frac{(1/3)\cdot q_{send} + q_{exe} + 1}{p^2}) + \frac{q_h^2 + ((1/3)\cdot q_{send} + q_{exe})^2}{2p} \quad (3)$$

**证明**：假设存在 PPT 敌手 $\mathcal{A}$ 攻击 OF-PSI 协议的 IND-CPA 安全性，构建一个 PPT 敌手 $\mathcal{B}$ 通过 $\mathcal{A}$ 攻击 DDH 假设。如果 $\mathcal{A}$ 能够以不可忽略的优势 $\varepsilon(\lambda)$ 攻破 IND-CPA 安全性，$\mathcal{B}$ 可以不可忽略的优势攻破 DDH 假设。

Initialize：$\mathcal{A}$ 确定挑战用户实体集合 $U$ 以及两个等长的消息 $M_0$ 和 $M_1$ 发送给 $\mathcal{B}$。在 OF-PSI 协议中存在三个实体，两个参与者 $P_1, P_2 \in$ **User** 以及一个服务器 $S \in$ **Server**，参与用户实体 $P_1^i, P_2^i (i \in \mathbb{Z})$ 和服务器实体 $S^i$ 为执行第 $i$ 次 OF-PSI 协议的一组实体，用户实体集合表示为 $U \in$ **User** $\cup$ **Server**。

Setup：接收安全参数 $\lambda$，$\mathcal{B}$ 运行 KeyGen 算法生成公私钥对 $(pk_1, sk_1)$ 和 $(pk_2, sk_2)$，公开系统公钥 $pk = pk_1 \cdot pk_2$，保留私钥 $sk_1$ 和 $sk_2$，则 $\mathcal{B}$ 的挑战四元组为 $T = (g, pk, g^{r_1+r_2}, Z)$。

Queries：OF-PSI 协议中所涉及到的 oracle 询问如下：

1) Hash $H(M)$ (or $H'(M)$)：若在列表 $L$（或 $L'$）中已存在记录 $(M, r)$，则返回 $r$；否则，随机选取 $r \in G$，将记录 $(M, r)$ 存储在列表 $L$（或 $L'$）中，返回 $r$。
2) Send $(P^i, a, Start)$：对于哈希表中的 $m$ 个哈希桶，若 $a = 1$，随机选择 $r_1^k (k \in [1, m])$，计算 $ct_{T_1} = E(x^k) = (g^{H(x^k)} \cdot pk^{r_1^k})$，返回 $ct_{T_1}$；若 $a = 2$，随机选择 $r_2^k (k \in [1, m])$，计算 $ct_{T_2} = E(y^k) = (g^{H(y^k)} \cdot pk^{r_2^k})$，返回 $ct_{T_2}$。
3) Send $(P^i, a, ct_{T_a})$：若 $a = 1$，计算 $ct_{T_1}^* = E^*(x^k) = (g^{H(x^k)} \cdot pk^{r_1^k + r_2^k})$，返回 $ct_{T_1}^*$；若 $a = 2$，计算 $ct_{T_2}^* = E^*(y^k) = (g^{H(y^k)} \cdot pk^{r_1^k + r_2^k})$，返回 $ct_{T_2}^*$。
4) Send $(S^i, (ct_{T_1}^*, ct_{T_2}^*))$：对于 $k \in [1, m]$，计算 $ct_{I^k} = ct_{T_1^k}^* \cap ct_{T_2^k}^*$ 及 $ct_I = ct_{I^1} \cup ct_{I^2} \cup \cdots \cup ct_{I^m}$，返回 $ct_I$。
5) Execute $(P_1^i, P_2^i, S^i)$：基于 Send 询问的成功模拟情况，返回 $ct_{T_1} \leftarrow \mathrm{Send}(P_1^i, Start)$，$ct_{T_2} \leftarrow \mathrm{Send}(P_2^i, Start)$，$ct_{T_1}^* \leftarrow \mathrm{Send}(P_2^i, ct_{T_1})$，$ct_{T_2}^* \leftarrow \mathrm{Send}(P_1^i, ct_{T_2})$ 和 $ct_I \leftarrow \mathrm{Send}(S^i, (ct_{T_1}^*, ct_{T_2}^*))$。
6) Corrupt $(P^i, a)$：若 $a = 1$，返回 $sk_1$ 和数据集合 $X$；若 $a = 2$，返回 $sk_2$ 和数据集合 $Y$。
7) Collude $(S^i)$：返回 $S^i$ 上存储的 $ct_{T_1}^*$、$ct_{T_2}^*$ 和 $ct_I$。

Challenge：$\mathcal{B}$ 随机选取一个 $b \leftarrow_R \{0,1\}$，加密 $M_b$。选取两个随机数 $r_1$ 和 $r_2$，计算 $E(M_b) = (g^{r_1+r_2}, g^{H(M_b)} \cdot Z)$ 并返回密文 $E(M_b)$。

Test：$\mathcal{A}$ 根据密文 $E(M_b)$ 以及上述 oracle 询问中得到的信息输出对于 $b$ 的猜测 $b'$，若 $b'=b$ 则返回 1，表示 $(g, pk, g^{r_1+r_2}, Z)$ 为 DDH 四元组分布；若 $b'\neq b$ 则返回 0，表示 $(g, pk, g^{r_1+r_2}, Z)$ 为随机四元组分布。

$$\mathrm{Adv}_{\mathcal{B}}^{DDH}(\lambda)=\left|\Pr[\mathcal{B}(T)=1]-\Pr[\mathcal{B}(T)=0]\right|=\left|\Pr[Succ]-1/2\right|=\mathrm{Adv}_{OF-PSI}^{IND-CPA}(\mathcal{A})$$

进一步通过一系列 hybrid 游戏 $\mathrm{Exp}_n(0\le n\le 4)$ 来证明定理 2。从游戏 $\mathrm{Exp}_0$ 敌手真实的攻击开始，到游戏 $\mathrm{Exp}_4$ 敌手没有任何优势时结束：

游戏 $\mathrm{Exp}_0$：在 random oracle 模型中，$\mathrm{Exp}_0$ 与真实的攻击相对应，由定义 2 可得：

$$\mathrm{Adv}_{OF-PSI}^{IND-CPA}(\mathcal{A})=\Pr[Succ_0]-1/2$$

游戏 $\mathrm{Exp}_1$：通过维持列表 $L$（及 $L'$）中的记录来模拟 oracle 哈希 $H:\{0,1\}^*\to G$（及 $\mathrm{Exp}_3$ 中的 $H'$），其余部分同 $\mathrm{Exp}_0$ 一样，仍与真实的攻击相对应。因此可知 $\mathrm{Exp}_1$ 与现实中敌手攻击的情况是不可区分的，故有：

$$\left|\Pr[Succ_1]-\Pr[Succ_0]\right|\approx 0$$

游戏 $\mathrm{Exp}_2$：$\mathrm{Exp}_2$ 与 $\mathrm{Exp}_1$ 一样对 oracle 哈希 $H$ 和 $H'$ 进行了模拟，但是中止了在 $H(x)$ 和 $H(y)$ 中存在哈希碰撞的执行。根据生日悖论可知，在发送 $q_h$ 次的 random oracle 询问、$q_{exe}$ 次的 Execute 询问和 $q_{send}$ 次的 Send 询问时，模拟 $H(x)$ 和 $H(y)$ 发生碰撞的概率最多可能为 $(q_h^2+(q_{send}/3+q_{exe})^2)/2p$，因此有：

$$\left|\Pr[Succ_2]-\Pr[Succ_1]\right|\le\frac{q_h^2+((1/3)\cdot q_{send}+q_{exe})^2}{2p}$$

游戏 $\mathrm{Exp}_3$：$\mathrm{Exp}_3$ 中，使用 oracle 哈希 $H'$ 代替 $H$ 来计算 $E'(x)=(g^{H'(x)}\cdot pk^{r_1})$ 和 $E'(y)=(g^{H'(y)}\cdot pk^{r_2})$，其余部分与 $\mathrm{Exp}_2$ 仍相同，$E'(x)$ 和 $E'(y)$ 与 $E(x)$ 和 $E(y)$ 是完全独立的。游戏 $\mathrm{Exp}_3$ 与 $\mathrm{Exp}_2$ 是完全不可区分的除非有事件 $\mathrm{AskH}_3$ 发生：$\mathcal{A}$ 对 oracle 哈希 $H$ 关于 $E(x)$ 和 $E(y)$ 进行了询问。此外，无论 Challenge 阶段中 $b$ 如何选取，$\mathcal{A}$ 对于其的猜测都是随机的，故可知：

$$\left|\Pr[Succ_3]-\Pr[Succ_2]\right|\le\Pr[\mathrm{AskH}_3],$$
$$\Pr[Succ_3]=1/2$$

游戏 $\mathrm{Exp}_4$：$\mathrm{Exp}_4$ 中，通过 DDH 随机自归约的属性来模拟协议的执行。给定一个 DDH 实例 $(A=g^x, B=g^y)$，随机选取 $\alpha,\beta\in\mathbb{Z}_p$，令 $E(x)=A^\alpha$，$E(y)=B^\beta$，则存在 DDH 四元组 $DDH(g, A, B, DDH(E(x), E(y)))$。由于事件 $\mathrm{AskH}_4$ 同样指 $\mathcal{A}$ 对 oracle 哈希 $H$ 关于 $E(x)$ 和 $E(y)$ 进行了询问，且存在：

$$DDH(E(x), E(y))=DDH(A^\alpha, B^\beta)=DDH(A, B)^{\alpha\beta}$$

故可知

$$\Pr[\mathrm{AskH}_3]=\Pr[\mathrm{AskH}_4]$$

$$\Pr[\mathrm{AskH}_4]\le q_h\mathrm{Adv}_{\mathcal{A}}^{DDH}(t+\frac{(1/3)\cdot q_{send}+q_{exe}+1}{p^2})$$

综上可得：

$$\mathrm{Adv}_{OF-PSI}^{IND-CPA}(\mathcal{A})\le q_h\mathrm{Adv}_{\mathcal{A}}^{DDH}(t+\frac{(1/3)\cdot q_{send}+q_{exe}+1}{p^2})+\frac{q_h^2+((1/3)\cdot q_{send}+q_{exe})^2}{2p}$$

结合定义 2 可知，OF-PSI 协议是 IND-CPA 安全的。

## 3.5 性能分析（Performance Analysis）

### 3.5.1 理论分析

将 OF-PSI 协议与文献[38, 45, 49]从计算复杂度、通信复杂度和是否具有公平性三个方面进行对比分析，具体如表 3-1 所示。

**表 3-1 OF-PSI 协议与相关协议对比分析**

**Tab.3-1 Comparison analysis of OF-PSI protocol and related protocols**

| 协议 | 计算复杂度 | | 通信复杂度 | 是否公平 |
|---|---|---|---|---|
| | 用户 | 服务器 | | |
| [38] | $9N+19$ | $6N+4$ | $6N+28$ | 否 |
| [45] | $48N$ | — | $103N$ | 是 |
| [49] | $23N+7$ | — | $12N+4$ | 是 |
| OF-PSI | $6.2N$ | 0 | $6N$ | 是 |

（1）计算复杂度

OF-PSI 协议中，使用 ElGamal 门限加密算法加密数据时产生的计算开销为协议的主要计算开销，我们使用加密时所执行模幂运算的次数来表示 OF-PSI 协议的计算复杂度。参与者 $P_1$ 在加密数据时执行了 $2m+n_1$ 次模幂运算，在解密时执行了 $w$ 次模幂运算；$P_2$ 在加密数据时执行了 $2m+2n_2$ 次模幂运算，解密时执行了 $w$ 次模幂运算；服务器只参与了元素比较工作，没有产生模幂运算。因此 OF-PSI 协议共执行了 $4m+n_1+2n_2+2w$ 次模幂运算，其中 $m$ 为哈希表中哈希桶的数量，$w$ 为交集集合中元素的个数。令 $N=\max\{n_1,n_2\}$，交集元素个数 $w\in[1,N]$，我们取其最大值 $N$。由 3.2 节可知 $N=4m\cdot 0.9$，故 $4m\approx 1.2N$，因此本协议执行模幂运算的次数为 $6.2N$。

在文献[38]的协议中使用 ElGamal 加密算法加密数据，参与用户双方共执行 $9N+19$ 次模幂运算，服务器执行了 $6N+4$ 次模幂运算，故文献[38]共执行了 $15N+13$ 次模幂运算。文献[45]使用基于 Paillier 决策复合剩余假设的 Camenisch–Shoup 加密算法加密数据，协议中没有服务器的参与，参与双方共执行了 $48N$ 次的模幂运算。文献[49]中使用分布式 ElGamal 加密算法加密数据，同样不涉及服务器，共执行 $23N+7$ 次模幂运算。

（2）通信复杂度

使用协议执行过程中传送的密文数量来表示协议的通信开销。OF-PSI 协议中，参与者 $P_1$ 和 $P_2$ 均在数据加密阶段发送了 $n_1+2n_2$ 个密文，因此 OF-PSI 协议共传送了 $2n_1+4n_2$ 个密文。令 $N=\max\{n_1,n_2\}$，故本协议整体传送密文的个数为 $6N$。在文献[38]的协议中参与用户和服务器共传送了 $6N+28$ 个密文。文献[45]中协议的参与双方共传送了 $103N$ 个密文。文献[49]的协议中参与双方整体传送的密文数量为 $12N+4$。

综上可以看出，OF-PSI 协议在实现公平性的同时，计算复杂度和通信复杂度均低于文献[45]和[49]中公平的 PSI 协议以及文献[38]中非公平的 PSI 协议。

### 3.5.2 实验分析

为了进一步验证 3.5.1 节中理论分析的结果，我们通过实验对 OF-PSI 协议与文献[41, 48, 52]中协议在运算方面所耗的时间进行了对比分析。实验平台：Windows10 64 位操作系统，处理器： AMD Ryzen 5 4600H Radeon Graphics 3.00 GHz 16gb RAM，编译环境：My Eclipse 2017。

由于 OF-PSI 协议与[38, 45, 49]中的协议均使用同态加密算法对数据进行加密，因此首先比较四个协议在不同模数位长的情况下执行同态加密所耗的时间。在本次实验中，设置集合元素个数 $N=1000$，模数长度分别为 128 bit、256 bit、512 bit 和 1024 bit 时，四个协议分别执行同态加密所需的时间如图 3-2 所示。

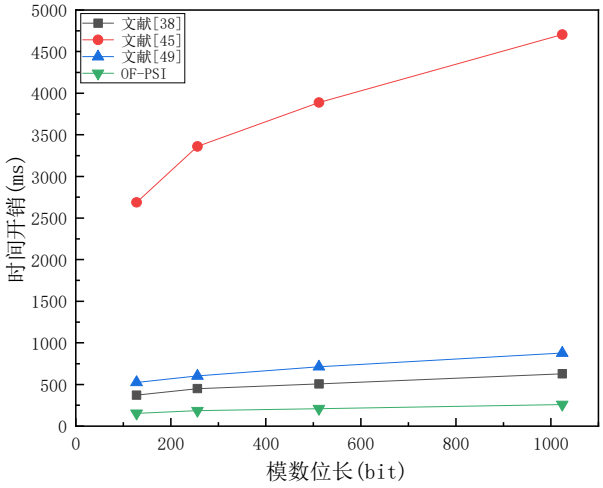



**图 3-2 不同模数位长时执行同态加密时间开销**

**Fig.3-2 Time cost to perform homomorphic encryption of different modulus length**

随后进一步比较了四个协议在不同集合元素个数下执行所需的时间。在该实验中，将模数位长固定为 1024 bit，在集合元素个数分别为 $2^{10}$、$2^{12}$、$2^{14}$、$2^{16}$、$2^{18}$ 和 $2^{20}$ 时，四个协议的执行所耗时间如图 3-3 所示。

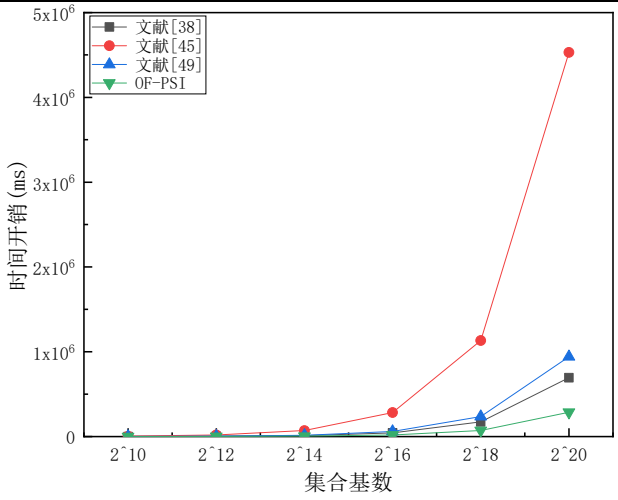



**图 3-3 不同集合元素个数时 OF-PSI 协议时间开销**
**Fig.3-3 Time cost of OF-PSI protocol with different set cardinality**

从图 3-2 可以看出，随着模数位长的不断增加，四个协议执行同态加密的时间也随之增加。其中，OF-PSI 协议的时间开销相对文献[38, 45, 49]中协议的时间开销而言较低，且增长趋势最为缓慢。该实验的时间开销取决于方案中使用模幂运算的数量。由表 3-1 可知，OF-PSI 协议计算中涉及的模幂运算次数最少，因此模数位长的增加对 OF-PSI 协议的影响最小。

对比分析图 3-3 的运行结果可知，四个协议的执行时间随着数据集合元素的增加而不断增长。其中，OF-PSI 的执行时间开销的增长速度远远慢于文献[38, 45, 49]中协议执行所耗时间的增长速度。这是由于 OF-PSI 协议中使用哈希算法预先对集合元素进行了分类处理。因此，随着数据集合元素的不断增加，OF-PSI 协议的时间开销曲线增长缓慢，而其他三个协议的时间开销曲线增长显著。

## 3.6 本章小结（Summary of This Chapter）

本章针对集合交集问题中两方之间的隐私集合交集计算问题进行了讨论，提出一种基于服务器的公平性 PSI 协议。方案使用哈希算法对集合元素进行预处理，将所有元素一一比较的方式改为以哈希桶为单位只对相应桶内元素进行比较的方式来计算交集，减少了元素比较的次数。方案引入服务器作为第三方，并将交集的计算交给服务器，减少双方的计算负担的同时使得双方可同时得到交集的计算结果，实现了协议的公平性。此外，该方案采用 ElGamal 门限加密算法，保证了数据的安全性。只有双方合作时所有密文才能完全解密，因此能够有效地抵抗服务器与任意一方的合谋攻击，并证明了该方案在 DDH 假设下是 IND-CPA 安全的。通过理论分析和实验评估，表明本文提出的 OF-PSI 协议的计算开销低于其他的公平性 PSI 协议。

# 4 公平的两方隐私集合交集势协议

# 4. Fair two-party Privacy Set Intersection cardinality protocol

作为隐私集合交集问题的变体，隐私集合交集势问题在多方协同计算过程中能够实现数据的隐私保护，在社交网络、边缘物联网、电子医疗等要求得到共同信息的数量且不泄露共同信息的特定应用场景得到了广泛应用。本章在第 3 章的基础上对两方之间的集合交集势计算问题进行研究，提出一个公平的隐私集合交集势协议，在实现两个参与方之间公平性的同时解决两方隐私集合的交集势计算问题。

## 4.1 引言（Introduction）

两方的 PSI-CA 协议在两个拥有隐私数据集合的参与者共同完成集合的交集势计算后，参与的一方或双方可获得交集势结果，不泄露各自的数据集合的信息且不揭露交集集合元素具体信息，能够在保护隐私的情况下安全解决计算共有信息数量的问题。但在构造 PSI-CA 协议时需面临以下问题：

（1）隐私性。在实际生活中，两个参与用户将各自的隐私数据信息作为输入信息，通过计算两者拥有共同隐私数据信息的数量，来作为用户进行下一步行动的依据。但在执行计算的过程中，用户的个人隐私信息也被暴露在网络中，面临隐私泄露的风险，如何在保证用户隐私数据安全的情况下计算双方共有信息的数量是 PSI-CA 协议需要解决的问题。

（2）公平性。PSI-CA 协议中的公平性指参与的一方获得交集势结果的同时，另一个参与方也应获得交集势计算结果。但经过对目前的 PSI-CA 协议研究发现，大部分的 PSI-CA 协议在执行结束后均先由一个参与方得到交集势结果，再告知给另一个参与方，没有实现参与双方之间的公平性，无法满足实际应用当中参与双方希望同时得到交集势计算结果的需求。

（3）高效性。PSI-CA 问题是针对大数据环境下人们协同安全计算的特定应用问题，具有较高的效率是 PSI-CA 协议被广泛应用的关键。但保护数据安全的加密算法的引入增加了协议的计算负担，保证参与双方公平性的构造模块的使用进一步增加了协议的计算复杂度，使得协议更加复杂，无法在保证隐私性和公平性的同时实现协议的高效性。

基于 OF-PSI 协议，本章同样采用哈希映射算法来预先对集合元素进行处理，但使用可交换加密算法代替门限加密算法且不需要服务器的参与，提出了一种能够实现参与双方之间公平性的隐私集合交集势协议（FPSI-CA），来满足特定应用场景中不揭露交

集集合元素具体信息，只需使用双方集合交集的势来进行下一步计算的要求。经过对协议的理论和实验分析，表明在集合元素个数一定的情况下，FPSI-CA 协议的计算开销远远低于其他公平的交集势协议，达到与不公平的隐私集合交集势协议相近的复杂度，协议整体性能得到提升。

## 4.2 系统模型（System Model）

FPSI-CA 协议中共涉及到两个实体：参与者 $P_1$ 和参与者 $P_2$。其中 $P_1$ 拥有隐私数据集 $X=\{x_1,\ldots x_{n_1}\}$，$P_2$ 拥有隐私数据集 $Y=\{y_1,\ldots,y_{n_2}\}$。他们希望计算计算 $X$ 和 $Y$ 的交集势 $|X\cap Y|$，而不透露交集集合中元素的具体信息和各自集合中的任何信息。协议的系统模型如图 4-1 所示。

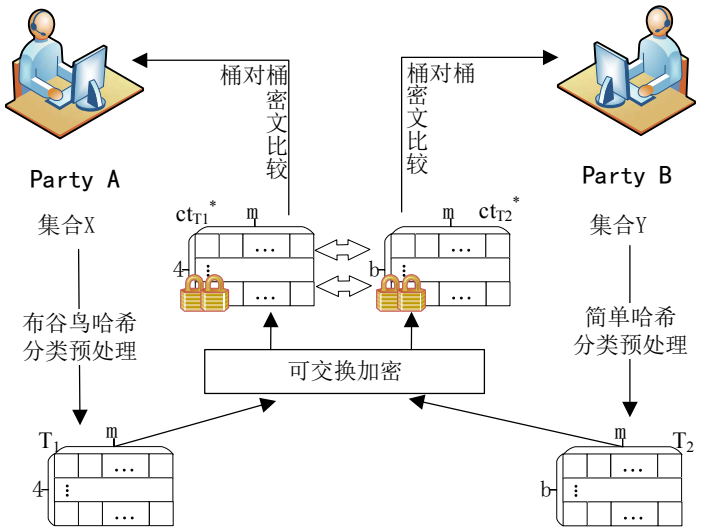



**图 4-1 FPSI-CA 协议系统模型图**
**Fig.4-1 System model of FPSI-CA protocol**

（1）哈希映射分类预处理集合元素

参与者 $P_1$ 和 $P_2$ 商定使用两个哈希函数 $\{h_1,h_2\}:\{0,1\}^*\rightarrow\{1,\ldots,m\}$ 来生成每个集合元素在哈希表中对应的两个哈希桶位置。其中 $P_1$ 使用布谷鸟哈希[79]，对于集合 $X$ 中每个元素 $x_i(i=1,\ldots,n_1)$，计算其在哈希表 $T_1$ 中对应的两个哈希桶 $h_1(x_i)$ 和 $h_2(x_i)$，并选择一个将 $x_i$ 存入，每个哈希桶存储 4 个元素；$P_2$ 使用简单哈希，对于集合 $Y$ 中的每个元素 $y_j(j=1,\ldots,n_2)$，计算其在哈希表 $T_2$ 中对应的两个哈希桶位置 $h_1(y_j)$ 和 $h_2(y_j)$，在其中均存入 $y_j$，每个哈希桶存储 $b$ 个元素。这样参与者 $P_1$ 和 $P_2$ 就将各自集合的元素进行了分类，桶号相等的哈希桶内的元素为一类。

（2）交换加密集合元素

方案使用基于 ElGamal 的交换加密算法，在保护用户集合信息隐私性的同时保证了

哈希表$T_1$和$T_2$中相等元素在加密后密文也相同。参与者$P_a(a=1,2)$使用自己的公钥$pk_a$加密哈希表中的元素后，将其发送给另一方进行重加密。这样，$T_1$中存放的集合$X$中的每个元素$x_i(i=1,\ldots,n_1)$转化为密文$E_{pk_2}(E_{pk_1}(x_i))$；$T_2$中存放的集合$Y$中的每个元素$y_j(j=1,\ldots,n_2)$转化为密文$E_{pk_1}(E_{pk_2}(y_j))$。结合哈希算法可知，对于$T_1$哈希桶内存在的任意元素$x_i \in X$，如果有相同的元素$y_j \in Y$存储在$T_2$相应的哈希桶中，那么在这两个编号相同的哈希桶中必然有$E_{pk_2}(E_{pk_1}(x_i)) = E_{pk_1}(E_{pk_2}(y_j))$。因此，基于 ElGamal 的交换加密算法的使用确保了参与双方能够在密文的形式下以哈希桶为单位执行元素的比较操作。

（3）桶对桶交集势计算

参与者$P_1$和$P_2$分别以桶对桶的方式比较两个哈希表中的密文元素，即将$T_1$哈希桶内的元素与$T_2$相应的哈希桶内的元素进行比较。$m$个哈希桶中所有相等的密文元素的个数即为交集势$|X \cap Y|$。因不存在解密操作且所有密文元素均经过了双方的两重加密，故在不知道对方私钥的情况下无法得知交集元素的具体情况以及对方的集合元素信息。

## 4.3 具体方案（Specific Scheme）

输入：$P_1$输入隐私数据集合$X=\{x_1,\ldots x_{n_1}\}$，

$P_2$输入隐私数据集合$Y=\{y_1,\ldots,y_{n_2}\}$。

输出：交集$|X \cap Y|$

（1）初始化阶段

步骤 1-1：$P_1$和$P_2$分别运行 ElGamal 的 KeyGen 算法，生成各自的公私钥对$(pk_a, sk_a), a=1,2$；

步骤 1-2：$P_1$和$P_2$确定哈希函数$H:\{0,1\}^* \to G$，商定哈希表中桶的数量$m$以及元素预处理哈希函数$h_1, h_2:\{0,1\}^* \to \{1,\ldots,m\}$。

（2）元素预处理阶段

步骤 2-1：对于$x_i \in X, i \in [n_1]$，$P_1$计算其在哈希表$T_1$中对应的哈希桶位置$h_1(x_i)$和$h_2(x_i)$，随机选择一个插入$x_i$；在集合$X$中所有元素存入哈希表$T_1$后，使用随机元素对$T_1$中的空闲位置进行填充；

步骤 2-2：对于$y_j \in Y, j \in [n_2]$，$P_2$计算其在哈希表$T_2$中对应的哈希桶位置$h_1(y_j)$和$h_2(y_j)$，在两个位置均插入$y_j$；在集合$Y$中所有元素存入哈希表$T_2$后，使用随机元素对$T_2$中的空闲位置进行填充。

（3）数据加密阶段

步骤 3-1：$P_1$以哈希桶为单位对$T_1$中的元素$x_i(i=1,\ldots,n_1)$进行加密，对于$k \in [1,m]$，

选取随机数 $r_1^k$ 计算 $ct_{T_1} = E_{pk_1}(x_i^k) = H(x_i^k) \cdot pk_1^{r_1^k}$，并以置换序列 $s_1$ 将 $ct_{T_1}$ 发送给 $P_2$；$P_2$ 以哈希桶为单位对 $T_2$ 中的元素 $y_j(j=1,\ldots,n_2)$ 进行加密，对于 $k \in [1,m]$，选取随机数 $r_2^k$ 计算 $ct_{T_2} = E_{pk_2}(y_j^k) = H(y_j^k) \cdot pk_2^{r_2^k}$，并以置换序列 $s_2$ 将 $ct_{T_2}$ 发送给 $P_1$；

步骤 3-2：$P_1$ 对接收到的 $ct_{T_2}$ 进行重加密，对于 $k \in [1,m]$，计算 $ct_{T_2}^* = E_{pk_1}(E_{pk_2}(y_j^k)) = H(y_j^k) \cdot pk_2^{r_2^k} \cdot pk_1^{r_1^k}$，将 $ct_{T_2}^*$ 以置换序列 $s_1$ 发送给 $P_2$；对于 $k \in [1,m]$，$P_2$ 计算 $ct_{T_1}^* = E_{pk_2}(E_{pk_1}(x_i^k)) = H(x_i^k) \cdot pk_1^{r_1^k} \cdot pk_2^{r_2^k}$ 来对接收到的 $ct_{T_1}$ 进行重加密，并将 $ct_{T_1}^*$ 以置换序列 $s_2$ 发送给 $P_1$。

（4）交集势计算阶段

对于 $ct_{T_1}^*$ 和 $ct_{T_2}^*$，$P_1$ 和 $P_2$ 分别进行如下计算：

步骤 4-1：对于 $k \in [1,m]$，设置 $w_k$ 的初始值为 0，比较 $ct_{T_1^k}^*$ 与 $ct_{T_2^k}^*$ 中的密文元素，遇到相等密文元素则 $w_k = w_k + 1$；

步骤 4-2：交集势 $|X \cap Y|$ 即为 $w = \sum_{k=1}^{m} w_k$。

## 4.4 方案分析（Scheme Analysis）

### 4.4.1 正确性分析

**定理 3.** FPSI-CA 协议可以正确的计算两方集合的交集势。

**证明：**对于隐私集合 $X = \{x_1, \ldots x_{n_1}\}$ 和 $Y = \{y_1, \ldots, y_{n_2}\}$，若参与双方能够正确计算出交集势 $|X \cap Y|$，则说明 FPSI-CA 协议是正确的。

在元素预处理阶段，$P_1$ 和 $P_2$ 均使用哈希函数 $h_1, h_2 : \{0,1\}^* \to \{1,\ldots,m\}$ 将元素映射入哈希表中，不同的是，$P_1$ 在每个元素 $x_i \in X(i=1,\ldots,n_1)$ 对应的两个哈希桶 $h_1(x_i)$ 或 $h_2(x_i)$ 中只选择一个来存储 $x_i$，在每个元素 $y_j \in Y(j=1,\ldots,n_2)$ 对应的两个哈希桶 $h_1(y_j)$ 和 $h_2(y_j)$ 中都存入 $y_j$。这样若存在 $x_i = y_j$，则有 $h_1(x_i) = h_1(y_j)$，$h_2(x_i) = h_2(y_j)$，无论 $x_i$ 选择 $h_1(x_i)$ 或 $h_2(x_i)$ 进行存储，$h_1(y_j)$ 和 $h_2(y_j)$ 中均有 $y_j$ 与之相对应。因此该预处理方法能够正确且完整的对元素进行分类，该阶段是正确的。

在数据加密阶段，哈希表 $T_1$ 中存储的元素信息经交换加密后转变为密文信息 $ct_{T_1}^* = E_{pk_2}(E_{pk_1}(x_i)) = H(x_i) \cdot pk_1^{r_1^k} \cdot pk_2^{r_2^k}$，哈希表 $T_2$ 中存储的元素信息经交换加密后转变为密文信息 $ct_{T_2}^* = E_{pk_1}(E_{pk_2}(y_j^k)) = H(y_j^k) \cdot pk_2^{r_2^k} \cdot pk_1^{r_1^k}$。根据可交换加密的特点，在任意哈希桶中若存在 $x_i^k = y_j^k (k \in [1,m])$，则必有 $E_{pk_2}(E_{pk_1}(x_i)) = E_{pk_1}(E_{pk_2}(y_j))$，故该阶段是正确的。

在交集势计算阶段，$P_1$ 和 $P_2$ 分别以桶为单位对 $T_1$ 哈希桶中的密文元素与 $T_2$ 相应哈希桶中的密文元素进行比较，遇到相等元素则计数器加 1，得到：

$$\begin{aligned}|X\cap Y| &= \left|ct_{I^1}\cup ct_{I^2}\cup\cdots\cup ct_{I^m}\right| \\ &= \left|(ct^*_{T_1^1}\cap ct^*_{T_2^1})\right|+\left|(ct^*_{T_1^2}\cap ct^*_{T_2^2})\right|+\cdots+\left|(ct^*_{T_1^m}\cap ct^*_{T_2^m})\right|\end{aligned}$$

因此交集势计算阶段也是正确的。

综上，FPSI-CA 协议能够正确的计算出两方集合的交集势。

### 4.4.2 安全性分析

**定理 4**. 令 $G$ 是一个阶为大素数 $p$ 的循环群，$\mathcal{A}$ 是在多项式时间 $t$ 内攻击 IND-CPA 安全性的 PPT 敌手。设 $\mathcal{A}$ 可发送少于 $q_{exe}$ 次的 Execute 询问和 $q_{send}$ 次的 Send 询问，最多 $q_h$ 次的 random oracle 询问，因此有：

$$\mathrm{Adv}^{IND-CPA}_{FPSI-CA}(\mathcal{A})\le q_h\mathrm{Adv}^{DDH}_{\mathcal{A}}(t+\frac{(1/2)\cdot q_{send}+q_{exe}+1}{p^2})+\frac{q_h^2+((1/2)\cdot q_{send}+q_{exe})^2}{2p} \quad (4)$$

**证明**：假设存在 PPT 敌手 $\mathcal{A}$ 攻击 FPSI-CA 协议的 IND-CPA 安全性，构建一个 PPT 敌手 $\mathcal{B}$ 通过 $\mathcal{A}$ 攻击 DDH 假设。如果 $\mathcal{A}$ 能够以不可忽略的优势 $\varepsilon(\lambda)$ 攻破 IND-CPA 安全性，$\mathcal{B}$ 可以不可忽略的优势攻破 DDH 假设。

Initialize：$\mathcal{A}$ 确定挑战用户实体集合 $U$ 以及两个等长的消息 $M_0$ 和 $M_1$ 发送给 $\mathcal{B}$。在 FPSI-CA 协议中存在两个实体，参与者 $P_1\in U_1$，$P_2\in U_2$。实体 $U_1^i,U_2^i(i\in\mathbb{Z})$ 为执行第 $i$ 次 FPSI-CA 协议的一组实体，实体集合表示为 $U\in U_1\cup U_2$。

Setup：接收安全参数 $\lambda$，$\mathcal{B}$ 运行 KeyGen 算法生成公私钥对 $(pk_1,sk_1)$ 和 $(pk_2,sk_2)$，公开公钥 $pk_1$ 和 $pk_2$，保留私钥 $sk_1$ 和 $sk_2$，则 $\mathcal{B}$ 的挑战四元组为 $T=(g,pk_2,g^{r_2},Z)$。

Queries：FPSI-CA 协议中所涉及到的 oracle 询问如下：

1) Hash $H(M)$(or $H'(M)$)：若在列表 $L$（或 $L'$）中已存在记录 $(M,r)$，则返回 $r$；否则，随机选取 $r\in G$，将记录 $(M,r)$ 存储在列表 $L$（或 $L'$）中，并返回 $r$。
2) Send ( $U^i,a,Start$ )：对于 $k\in[1,m]$，若 $a=1$，随机选择 $r_1^k$，计算 $ct_{T_1}=E_{pk_1}(x_i^k)=H(x_i^k)\cdot pk_1^{r_1^k}$，返回 $ct_{T_1}$；若 $a=2$，随机选择 $r_2^k$，计算 $ct_{T_2}=E_{pk_2}(y_j{}^k)=H(y_j{}^k)\cdot pk_2^{r_2^k}$，返回 $ct_{T_2}$。
3) Send $(U^i,a,ct_{T_a})$：若 $a=1$，计算 $ct^*_{T_1}=E_{pk_2}(E_{pk_1}(x_i))=H(x_i)\cdot pk_1^{r_1^k}\cdot pk_2^{r_2^k}$，返回 $ct^*_{T_1}$；若 $a=2$，计算 $ct^*_{T_2}=E_{pk_1}(E_{pk_2}(y_j{}^k))=H(y_j{}^k)\cdot pk_2^{r_2^k}\cdot pk_1^{r_1^k}$，返回 $ct^*_{T_2}$。
4) Execute ( $U_1^i,U_2^i$ )：基于 Send 询问的成功模拟情况，返回 $ct_{T_1}\leftarrow$ Send( $U_1^i,Start$ )，$ct_{T_2}\leftarrow$ Send( $U_2^i,Start$ )，$ct^*_{T_1}\leftarrow$ Send( $U_2^i,ct_{T_1}$ ) 和 $ct^*_{T_2}\leftarrow$Send$(U_1^i,ct_{T_2})$。
5) Corrupt $(P^i,a)$：若 $a=1$，返回 $sk_1$ 和隐私数据集合 $X$；若 $a=2$，返回 $sk_2$ 和

隐私数据集合$Y$。

Challenge：$\mathcal{B}$随机选取一个$b \leftarrow_R \{0,1\}$，加密$M_b$。选取两个随机数$r_1$和$r_2$，计算$E(M_b)=(g^{r_2},(H(M_b)\cdot pk_1^{r_1})\cdot Z)$，发送$E(M_b)$给$\mathcal{A}$。

Test：$\mathcal{A}$根据上述 oracle 询问中得到的信息输出对于$b$的猜测$b'$，若$b'=b$则返回 1，表示$Z=pk_2^{r_2}$，$(g,pk_2,g^{r_2},Z)$为 DDH 四元组分布；若$b'\neq b$则返回 0，表示$Z$为随机元素，$(g,pk_2,g^{r_2},Z)$为随机四元组分布。

$$\mathrm{Adv}_{\mathcal{B}}^{DDH}(\lambda)=\left|\Pr[\mathcal{B}(T)=1]-\Pr[\mathcal{B}(T)=0]\right|=\left|\Pr[Succ]-1/2\right|=\mathrm{Adv}_{FPSI-CA}^{IND-CPA}(\mathcal{A})$$

进一步通过一系列 hybrid 游戏$\mathrm{Exp}_n(0\le n\le 4)$来证明定理 4。从游戏$\mathrm{Exp}_0$敌手真实的攻击开始，到游戏$\mathrm{Exp}_4$敌手没有任何优势时结束：

游戏$\mathrm{Exp}_0$：在 random oracle 模型中，$\mathrm{Exp}_0$与真实的攻击相对应，由定义 2 知：

$$\mathrm{Adv}_{FPSI-CA}^{IND-CPA}(\mathcal{A})=\Pr[Succ_0]-1/2$$

游戏$\mathrm{Exp}_1$：通过维持列表$L$（及$L'$）中的记录来模拟 oracle 哈希$H:\{0,1\}^*\rightarrow G$（及$\mathrm{Exp}_3$中的$H'$），其余部分同$\mathrm{Exp}_0$一样，仍与真实的攻击相对应。因此可知$\mathrm{Exp}_1$与现实中敌手攻击的情况是不可区分的，故有：

$$\left|\Pr[Succ_1]-\Pr[Succ_0]\right|\approx 0$$

游戏$\mathrm{Exp}_2$：$\mathrm{Exp}_2$与$\mathrm{Exp}_1$一样对 oracle 哈希$H$和$H'$进行了模拟，但是中止了在$H(x)$和$H(y)$中存在哈希碰撞的执行。根据生日悖论可知，在发送$q_h$次的 random oracle 询问、$q_{exe}$次的 Execute 询问和$q_{send}$次的 Send 询问时，模拟$H(x)$和$H(y)$发生碰撞的概率最多可能为$(q_h^2+(q_{send}/3+q_{exe})^2)/2p$，因此有：

$$\left|\Pr[Succ_2]-\Pr[Succ_1]\right|\le\frac{q_h^2+((1/3)\cdot q_{send}+q_{exe})^2}{2p}$$

游戏$\mathrm{Exp}_3$：$\mathrm{Exp}_3$中，使用 oracle 哈希$H'$代替$H$来计算$E'(x)=(g^{H'(x)}\cdot pk^{r_1})$和$E'(y)=(g^{H'(y)}\cdot pk^{r_2})$，其余部分仍与$\mathrm{Exp}_2$相同，$E'(x)$和$E'(y)$与$E(x)$和$E(y)$是完全独立的。游戏$\mathrm{Exp}_3$与$\mathrm{Exp}_2$是完全不可区分的除非有事件$\mathrm{AskH}_3$发生：$\mathcal{A}$对 oracle 哈希$H$关于$E(x)$和$E(y)$进行了询问。此外，无论 Challenge 阶段中$b$如何选取，$\mathcal{A}$对于其的猜测都是随机的，故可知：

$$\left|\Pr[Succ_3]-\Pr[Succ_2]\right|\le\Pr[\mathrm{AskH}_3],$$
$$\Pr[Succ_3]=1/2$$

游戏$\mathrm{Exp}_4$：$\mathrm{Exp}_4$中，通过 DDH 随机自归约的属性来模拟协议的执行。给定一个 DDH 实例$(A=g^x,B=g^y)$，随机选取$\alpha,\beta\in\mathbb{Z}_p$，令$E(x)=A^{\alpha}$，$E(y)=B^{\beta}$，则存在四元组$DDH(g,A,B,DDH(E(x),E(y)))$，故$DDH(E(x),E(y))=DDH(A^{\alpha},B^{\beta})=DDH(A,B)^{\alpha\beta}$，

因此存在事件 $\mathrm{AskH}_4$：

$$\Pr[\mathrm{AskH}_4] \le q_h \mathrm{Adv}_{\mathcal{A}}^{DDH}(t + \frac{(1/3)\cdot q_{send} + q_{exe} + 1}{p^2})$$

由于事件 $\mathrm{AskH}_4$ 同样指 $\mathcal{A}$ 对 oracle 哈希 $H$ 关于 $E(x)$ 和 $E(y)$ 进行了询问，故可知：

$$\Pr[\mathrm{AskH}_3] = \Pr[\mathrm{AskH}_4]$$

综上可知：

$$\mathrm{Adv}_{FPSI-CA}^{IND-CPA}(\mathcal{A}) \le q_h \mathrm{Adv}_{\mathcal{A}}^{DDH}(t + \frac{(1/2)\cdot q_{send} + q_{exe} + 1}{p^2}) + \frac{q_h^2 + ((1/2)\cdot q_{send} + q_{exe})^2}{2p}$$

结合定义 2 可知，FPSI-CA 协议是 IND-CPA 安全的。

## 4.5 性能分析（Performance Analysis）

### 4.5.1 理论分析

本节从计算复杂度、通信复杂度和是否具有公平性三个方面对 FPSI-CA 协议与文献[49]和文献[61]中的 PSI-CA 协议进行对比分析，具体如表 4-1 所示。

**表 4-1 FPSI-CA 协议与相关协议对比分析**

**Tab.4-1 Comparison analysis of FPSI-CA protocol and related protocols**

| 协议 | 计算复杂度 | | 通信复杂度 | 是否公平 |
|---|---|---|---|---|
| | 模乘运算 | 哈希运算 | | |
| [49] | $(13\sqrt{2}+9)n_1 + (14\sqrt{2}+9)n_2$ | $n_1$ | $6n_1 + 7n_2$ | 是 |
| [61] | $n_1 + 3n_2$ | $14n_1 + 14n_2$ | $n_1 + 2n_2$ | 否 |
| FPSI-CA | $2n_1 + 4n_2$ | $3n_1 + 6n_2$ | $2n_1 + 4n_2$ | 是 |

（1）计算复杂度

FPSI-CA 协议中，使用哈希算法处理数据和可交换加密算法加密数据时的计算开销为协议的主要计算开销，故使用协议所执行的哈希运算和模乘运算的次数来表示 FPSI-CA 协议的计算复杂度。在元素预处理阶段，参与者 $P_1$ 执行了 $2n_1$ 次哈希运算，$P_2$ 执行了 $4n_2$ 次哈希运算；在数据加密阶段，$P_1$ 执行了 $n_1$ 次哈希运算和 $n_1 + 2n_2$ 次模乘运算，$P_2$ 执行了 $2n_2$ 次哈希运算和 $n_1 + 2n_2$ 次模乘运算；在交集势计算阶段本协议以密文形式对元素比较得到交集势，不涉及哈希运算和模乘运算；因此 FPSI-CA 协议共执行了 $3n_1 + 6n_2$ 次哈希运算，$2n_1 + 4n_2$ 次模乘运算。文献[49]中的协议整体执行了 $n_1$ 次哈希运算和 $(13\sqrt{2}+9)n_1 + (14\sqrt{2}+9)n_2$ 次模乘运算。文献[61]的协议中，$P_1$ 执行了 $14n_1$ 次哈希运算和 $n_1 + n_2$ 次模乘运算，$P_2$ 执行了 $14n_2$ 次的哈希运算和 $2n_2$ 次的模乘运算，因次文献[61]中协议共执行了 $14n_1 + 14n_2$ 次哈希运算和 $n_1 + 3n_2$ 次模乘运算。

（2）通信复杂度

使用协议执行过程中传送的密文数量来表示协议的通信开销。FPSI-CA 协议中，参

与者 $P_1$ 发送了 $n_1+2n_2$ 个密文给 $P_2$，$P_2$ 同样发送 $n_1+2n_2$ 个密文给 $P_1$，协议中共产生密文传送的数量为 $2n_1+4n_2$。因此 FPSI-CA 协议的通信复杂度为 $2n_1+4n_2$。文献[49]中，参与者 $P_1$ 和 $P_2$ 之间共传输了 $6n_1+7n_2$ 个密文，故协议的通信复杂度为 $6n_1+7n_2$。文献[61]中，$P_1$ 发送了 $n_1+n_2$ 个密文给 $P_2$，$P_2$ 发送了 $n_2$ 个密文给 $P_1$，其通信复杂度为 $n_1+2n_2$。

综上可以看出，FPSI-CA 协议在实现公平性的同时，其计算复杂度和通信复杂度相较文献[49]中公平的 PSI-CA 协议而言均较低，已趋近于文献[61]中的非公平性 PSI-CA 协议。

### 4.5.2 实验分析

通过实验对 FPSI-CA 协议与文献[49, 61]中协议在运算方面所耗的时间进行了对比分析，来进一步验证 4.5.1 节中理论分析的结果。实验平台：Windows10 64 位操作系统，处理器： AMD Ryzen 5 4600H Radeon Graphics 3.00 GHz 16gb RAM，编译环境：My Eclipse 2017。

FPSI-CA 协议与文献[49]和[61]中的协议均使用基于 ElGamal 的公钥加密算法来对数据进行加密，因此首先比较三个协议在不同模数位长的情况下加密数据时执行模乘运算所耗的时间。在本次实验中，设置集合元素个数 $N=1000$，模数长度分别为 128 bit、256 bit、512 bit 和 1024 bit 时，三个协议分别执行模乘运算所需的时间如图 4-2 所示。

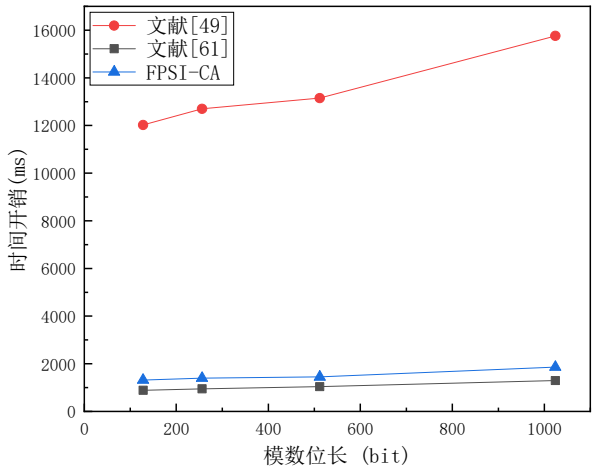



**图 4-2 不同模数位长时执行模乘运算时间开销**

**Fig.4-2 Time cost to perform modular multiplication of different modulus length**

从图 4-2 可以看出，模数位长在不断增加时，FPSI-CA 协议与文献[49]和[61]中协议的计算开销均随之不断增长。其中 FPSI-CA 协议与文献[61]中不公平的 PSI-CA 协议开销差距较小，且相较文献[49]中公平的 PSI-CA 协议而言增长缓慢。该实验的时间开销取决于协议中使用的模乘运算的数量，由表 4-1 可知，

FPSI-CA 协议与文献[61]的协议的在加密计算中使用模乘运算的次数相对文献[49]中协议较少，因此模数位长的增长对 FPSI-CA 协议与文献[61]中协议的影响相对文献[49]的协议较小。

之后加入哈希运算，进一步比较了三个协议在不同集合元素个数下执行所需的时间，在该实验中，将模数位长固定为 1024 bit，在集合元素个数分别为 $2^{10}$、$2^{11}$、$2^{12}$、$2^{13}$、$2^{14}$ 和 $2^{15}$ 时，四个协议执行时所耗的时间如图 4-3 所示。

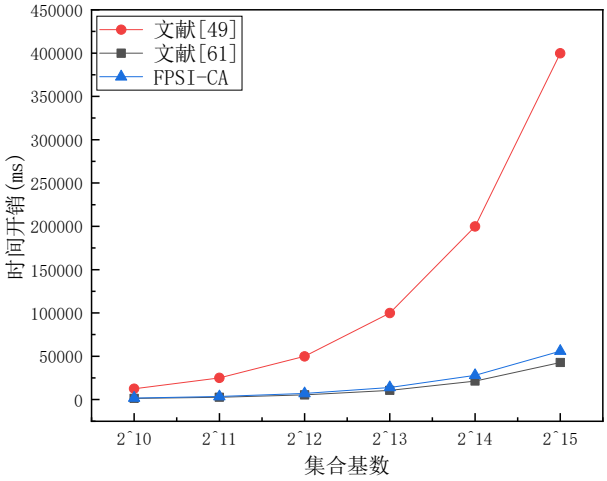



**图 4-3 不同集合元素个数时 FPSI-CA 协议时间开销**
**Fig.4-3 Time cost of FPSI-CA protocol with different set cardinality**

对比图 4-3 中运行结果可知，三个协议的执行时间随着数据集合元素的增加而不断增长。其中 FPSI-CA 协议的时间开销虽略高于文献[61]中不公平的 PSI-CA 协议，但远低于文献[49]中公平性 PSI-CA 协议急速增长的时间开销。这是由于 FPSI-CA 协议与文献[61]中的协议均使用哈希算法对集合元素进行了预处理，所以随着集合元素个数的不断增加时间开销曲线的增长较为缓慢且两者之间差距较小。但是文献[49]中的协议没有预先处理集合元素，所以其时间开销曲线增长较快。

## 4.6 本章小结（Summary of This Chapter）

本章主要讨论了隐私集合交集问题中两方之间的集合交集势计算问题，提出一种能够实现参与双方之间公平性的 PSI-CA 协议。方案使用哈希算法预先对集合元素进行分类存储，以哈希桶为单位只对相应哈希桶内元素进行比较来计算交集势，减少了元素比较的次数。方案采用交换加密算法，确保集合元素可以密文形式进行交集势计算，保证了数据集合的安全性。方案执行过程中参与双方的并行处理使得双方能够同时获得交集势结果，实现了双方之间的公平性。最后通过理论分析和实验评估，表明本文提出的

FPSI-CA 协议在实现公平性的同时具备较低的计算复杂度。

# 5 多方的隐私集合交集协议

# 5. Multi-party Privacy Set Intersection protocol

作为两方集合交集问题的扩展问题，多方隐私集合交集计算可在不泄露原始数据的前提下为数据需求方提供多方协同计算集合交集的能力，成为解决互不信任的参与方协同隐私计算问题的重要研究对象，在多个企业之间进行商业合作，以及不同医疗机构数据分析等场景中得到了广泛应用。本章对隐私保护环境下多方之间的集合交集问题进行研究，提出一个多方隐私集合交集协议，来解决互不信任的多个参与者之间隐私数据集合的交集计算问题。

## 5.1 引言（Introduction）

多方的 PSI 协议允许多个参与者分别输入自己的私有数据进行协同计算，计算完成后，每个参与者除计算结果外无法得知其他参与者的任何信息，使得互不信任的多个参与方在分布式环境下能够实现协同隐私计算。但在构造多方 PSI 协议时需解决以下问题：

（1）数据隐私性问题。多方的 PSI 协议是以两方 PSI 协议为基础进行实现的，在协议执行的过程中，不可避免的会产生中间计算结果，即两个参与者的集合交集结果，这导致参与者隐私数据集合的部分信息被泄露。多方的 PSI 协议要求每个参与者除计算结果外无法得知其他参与者的任何信息，如何在安全计算多个参与者隐私集合交集的同时保证每个参与者数据集合的隐私性是多方 PSI 协议需要解决的问题。

（2）参与者合谋的问题。多方 PSI 协议是由多个参与者共同执行完成的，但在实际生活中，无法保证每个参与者都是诚实的，因此在多方 PSI 协议的多个参与者中，存在一些参与者合谋，通过对协议执行过程中得到的信息进行分析，来获取其他参与者的隐私数据信息的问题。

（3）集合元素的处理问题。为了提高协议的计算效率，多方 PSI 协议需首先对集合元素进行处理。对于集合元素的处理方法，多数研究选择在协议执行前预先设定一个所有参与方共有的集合，每个参与方的集合属于共有集合，之后通过编码的方式对集合元素进行处理，这使得协议的复杂度大为减少。但在具体应用中，在多个实际参与方之间不泄露隐私的设定一个共同的信息集合是难以实现的。因此设计高效且可实行的集合元素处理方法同样是多方 PSI 协议需要解决的问题。

本章提出了一种能够在不预先设定共有集合的情况下实现多个参与方之间进行隐私计算的集合交集协议（M-PSI）。协议使用哈希算法预先对各方集合元素进行处理，

同时采用$(n,n)$门限加密算法保证所有参与者共同掌握交集元素信息且不泄露参与者集合中的任何信息，并在半诚实模型下证明了协议能够抵抗合谋攻击。最后对协议进行理论和实验分析，表明 M-PSI 协议在没有预先设定共同集合的情况下达到与设定有共同集合的多方隐私集合交集协议相近的复杂度，远远低于其他同样没有预先设定共同集合的多方隐私集合交集协议。

## 5.2 系统模型（System Model）

M-PSI 协议中，$n$ 个参与者 $P_1,\ldots,P_n$ 分别拥有私密集合 $X_a=\{x_1^a,\ldots,x_{n_a}^a\}$，$a\in[1,n]$，各方希望在不泄露自己隐私数据的情况下计算 $n$ 个集合的交集。协议的系统模型如图 5-1 所示。

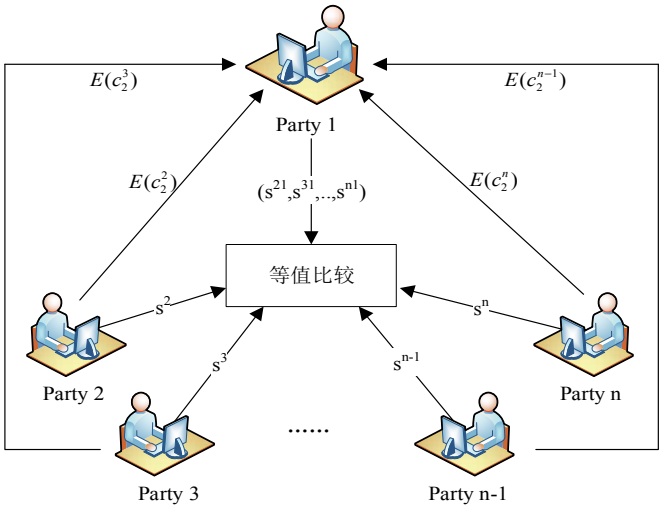



**图 5-1 M-PSI 协议系统模型图**
**Fig.5-1 System model of M-PSIprotocol**

（1）哈希映射分类预处理集合元素

参与者 $P_1,\ldots,P_n$ 商定使用两个哈希函数$\{h_1,h_2\}:\{0,1\}^*\to\{1,\ldots,m\}$来生成每个集合元素在哈希表中对应的两个哈希桶位置。其中 $P_1$ 使用布谷鸟哈希，对于集合 $X_1$ 中每个元素 $x_i^1(i=1,\ldots,n_1)$，计算其在哈希表 $T_1$ 中对应的两个哈希桶 $h_1(x_i^1)$ 和 $h_2(x_i^1)$，并选择一个将 $x_i^1$ 存入，每个哈希桶存储 1 个元素；$P_2,\ldots,P_n$ 使用简单哈希，对于各自集合 $X_a(a=2,\ldots,n)$ 中的每个元素 $x_i^a(i=1,\ldots,n_a)$，计算其在哈希表 $T_a$ 中对应的两个哈希桶位置 $h_1(x_i^a)$ 和 $h_2(x_i^a)$，在其中均存入 $x_i^a$，每个哈希桶存储 $b$ 个元素。以这样的方式，参与者 $P_1,\ldots,P_n$ 就将各自集合的元素进行了分类，桶号相等的哈希桶内的元素为一类。

（2）桶对桶执行等值比较

$P_1$ 分别与 $P_2,\ldots,P_n$ 执行等值比较协议，计算 $P_1$ 与 $P_a(a=2,\ldots,n)$ 之间交集的同时保证

其他参与者的集合信息不被泄露。$P_2,\ldots,P_n$ 首先使用盲化信息 $s_l^a$ 将哈希表中每个哈希桶中的 $b$ 个元素以盲化多项式 $P_a(x)$ 的方式发送给 $P_1$，$P_1$ 将相应哈希桶中的元素代入 $P_a(x)$ 得到盲化信息 $s_l^{a1}$。之后 $P_1$ 与 $P_a$ 输入各自的盲化信息执行等值比较协议，共享每个哈希桶中的元素是否为 $P_1$ 和 $P_a$ 之间交集元素的信息，使得单独的 $P_1$ 或 $P_a$ 一方无法得知两方之间具体的交集信息，为多方之间交集计算的顺利执行创造条件的同时保证了数据的隐私性。

（3）门限加密等值信息计算交集

方案基于 $(n,n)$ 门限加密算法的加法同态性，在汇总 $P_1$ 与 $P_2,\ldots,P_n$ 之间交集信息的同时保护用户集合信息的隐私。将 $P_1$ 分别与 $P_2,\ldots,P_n$ 之间两方的集合交集信息利用加同态算法汇总，并联合 $n$ 个参与者的私钥进行解密。若解密后某个汇总的信息为 0，则其对应的哈希桶内元素为交集 $X_1\cap\cdots\cap X_n$ 中的元素。

## 5.3 具体方案（Specific Scheme）

输入：$n$ 个参与者 $P_a(a\in[1,n])$ 输入各自的私密集合 $X_a=\{x_1^a,\ldots,x_{n_a}^a\}$

输出：$X_1\cap\cdots\cap X_n$

（1）初始化阶段

步骤 1-1：$P_1,\ldots,P_n$ 和 $P_2$ 确定哈希函数 $H:\{0,1\}^*\to G$

步骤 1-2：参与者 $P_1,\ldots,P_n$ 分别运行 ElGamal 的 KeyGen 算法，生成各自的公私钥对 $(pk_a,sk_a),a=1,\ldots,n$，公布公钥 $pk_a$，保留私钥 $sk_a$，则系统公钥为 $pk=pk_1pk_2\cdots pk_n$。

步骤 1-3：对于 $x_i^1\in X_1,i\in[1,n_1]$，$P_1$ 计算其在哈希表 $T_1$ 中对应的哈希桶位置 $h_1(x_i^1)$ 和 $h_2(x_i^1)$，随机选择一个插入 $x_i^1$；在集合 $X_1$ 中所有元素存入哈希表 $T_1$ 的 $m$ 个桶后，使用随机元素对 $T_1$ 中的空闲位置进行填充；

步骤 1-4：参与者 $P_2,\ldots,P_n$ 对于各自集合 $X_a(a\in[2,n])$ 中的元素 $x_i^a,i\in[1,n_a]$，分别计算其在哈希表 $T_a$ 中对应的哈希桶位置 $h_1(x_i^a)$ 和 $h_2(x_i^a)$，在两个位置均插入 $y_j$；集合 $X_a$ 中所有元素存入哈希表 $T_a$ 的 $m$ 个桶后，使用随机元素对 $T_a$ 中的空闲位置进行填充。

（2）等值比较阶段

对于 $l\in[1,m]$，$P_1$ 分别与 $P_2,\ldots,P_n$ 以哈希桶为单位执行 OPRF 协议：

步骤 2-1：$P_a$ 作为发送方无输入，$P_1$ 作为接收方输入 $T_1$ 第 $l$ 个哈希桶中元素 $x_l^1$，执行 OPRF 协议后 $P_1$ 接收到加密元素 $F_{k_a}(x_l^1)$，$P_a$ 接收到密钥 $k_a$；

步骤 2-2：$P_a$ 使用密钥 $k_a$ 对 $T_a$ 第 $l$ 个哈希桶中元素 $x_{l1}^a,\ldots,x_{lb}^a$ 进行加密，任选一随机

数 $s_l^a$，计算 $P_a(x)=s_l^a+\prod_{j=1}^{b}(x-F_{k_a}(x_{lj}^a))$，并将 $P_a(x)$ 发送给 $P_1$；

步骤 2-3：$P_1$ 接收到多项式 $P_a(x)$ 后，将 $F_{k_a}(x_l^1)$ 代入其中计算 $s_l^{a1}=P(F_{k_a}(x_l^1))$。

步骤 2-4：对于 $l\in[1,m]$，$P_1$ 分别与 $P_2,\ldots,P_n$ 以哈希桶为单位执行等值比较协议，$P_1$ 作为发送方输入 $s_l^{a1}$，$P_a$ 作为接收方输入 $s_l^a$，执行等值比较协议后 $P_1$ 得到 $c_{l1}^a=t_l^a$，$P_a$ 得到 $c_{l2}^a=e_l^a+t_l^a$，$t_l^a$ 为大于 0 的随机数；

（3）交集计算阶段

步骤 3-1：$P_a$ 选择随机数 $r_{l2}^a$，使用公钥 $pk$ 对 $c_{l2}^a$ 进行加密得到 $E(c_{l2}^a)$，并将 $E(c_{l2}^a)$ 发送给 $P_1$；

步骤 3-2：$P_1$ 选择随机数 $r_{l1}^a$，使用公钥 $pk$ 对 $-c_{l1}^a$ 进行加密得到 $E(-c_{l1}^a)$，计算

$$E(c_l)=\prod_{a=2}^{n}(E(c_{l2}^a)\cdot E(-c_{l1}^a))$$

步骤 3-3：$P_1$ 公布 $(E(c_1),E(c_2),\ldots,E(c_m))$；

步骤 3-4：$n$ 个参与者使用各自的私钥联合解密 $(E(c_1),\ldots,E(c_m))$ 得到 $(c_1,\ldots,c_m)$，若 $c_l=0$，$(1\le l\le m)$，则元素 $x_l^1$ 即为交集中的元素；

步骤 3-5：$P_1$ 输出交集 $X_1\cap\cdots\cap X_n=\{x_l^1\,|\,c_l=0,\ l=1,\ldots,m\}$。

## 5.4 方案分析（Scheme Analysis）

### 5.4.1 正确性分析

**定理 5.** M-PSI 协议可以正确的计算 $n$ 个参与者私密集合的交集。

**证明：**若 $n$ 个参与者的私密集合 $X_1,\ldots,X_n$ 在分类预处理的情况下，执行 OPRF 协议和等值比较协议后，经过 $(n,n)$ 门限加密能够以桶对桶的形式正确计算出交集 $X_1\cap\cdots\cap X_n$，则说明 M-PSI 协议是正确的。

在初始化阶段，$P_1$ 使用布谷鸟哈希将集合 $X_1$ 中每个元素 $x_i^1(i=1,\ldots,n_1)$ 在 $h_1(x_i^1)$ 或 $h_2(x_i^1)$ 中存储，$P_2,\ldots,P_n$ 均使用简单哈希将其集合 $X_a(a\in[2,n])$ 中每个元素 $x_i^a(i=1,\ldots,n_a)$ 在 $h_1(x_i^a)$ 和 $h_2(x_i^a)$ 中存储。因此若 $P_1$ 与 $P_a$ 间存在相等元素，其均可在哈希表 $T_1$ 与 $T_a$ 相应的哈希桶中找到，即该阶段是正确的。

在等值比较阶段，$P_2,\ldots,P_n$ 分别作为发送方与 $P_1$ 首先执行 OPRF 协议后，$P_1$ 得到加密的元素 $F_{k_a}(x_l^1),l\in[1,m]$，$P_a(a\in[2,n])$ 得到密钥 $k_a$ 对自己的元素进行加密；采用多项式来表示 $P_a$ 哈希桶中元素，以哈希桶中的 $b$ 个加密元素为多项式的根，并使用随机数 $s_l^a$ 对多项式进行盲化，$P_1$ 相应哈希桶中的元素代入多项式后的结果 $s_l^1$ 与 $s_l^a$ 执行等值比较协议来判断 $x_l^1$ 是否为 $P_1$ 与 $P_a$ 的交集元素，故该阶段是正确的。

在交集计算阶段，$n$ 个参与者共享的等值比较结果利用 $(n,n)$ ElGamal 门限加密的加法同态性进行汇总，计算

$$\begin{aligned} E(c_l) &= \prod_{a=2}^{n}(E(c_{l2}^a)\cdot E(-c_{l1}^a)) \\ &= E((c_{l2}^2-c_{l1}^2)+\cdots+(c_{l2}^n-c_{l1}^n)) \\ &= g^{(c_{l2}^2-c_{l1}^2)+\cdots+(c_{l2}^n-c_{l1}^n)}\cdot pk^{(r_{l2}^2+r_{l1}^2)+\cdots+(r_{l2}^n+r_{l1}^n)} \end{aligned}$$

其中 $1\le l\le m$，对 $E(c_l)$ 联合解密得到 $g^{(c_{l2}^2-c_{l1}^2)+\cdots+(c_{l2}^n-c_{l1}^n)}$，若 $g^{(c_{l2}^2-c_{l1}^2)+\cdots+(c_{l2}^n-c_{l1}^n)}=1$ 即 $(c_{l2}^2-c_{l1}^2)+\cdots+(c_{l2}^n-c_{l1}^n)=0$ 则表明元素 $x_l^1$ 为 $n$ 个参与者共有的交集元素。$m$ 个哈希桶内的所有 $n$ 方交集元素即为交集 $X_1\cap\cdots\cap X_n$，故该阶段是正确的。

综上，M-PSI 协议能够正确的计算出两方集合的交集。

## 5.4.2 安全性分析

**定理 6.** 令 $G$ 是一个阶为大素数 $p$ 的循环群，$\mathcal{A}$ 是在多项式时间 $t$ 内攻击 IND-CPA 安全性的 PPT 敌手。设 $\mathcal{A}$ 可发送少于 $q_{exe}$ 次的 Execute 询问和 $q_{send}$ 次的 Send 询问，最多 $q_h$ 次的 random oracle 询问，因此有：

$$\mathrm{Adv}_{OF-PSI}^{IND-CPA}(\mathcal{A})\le q_h\mathrm{Adv}_{\mathcal{A}}^{DDH}(t+\frac{(1/5)\cdot q_{send}+q_{exe}+1}{p^2})+\frac{q_h^2+((1/5)\cdot q_{send}+q_{exe})^2}{2p} \quad (5)$$

**证明：**假设存在 PPT 敌手 $\mathcal{A}$ 攻击 OF-PSI 协议的 IND-CPA 安全性，构建一个 PPT 敌手 $\mathcal{B}$ 通过 $\mathcal{A}$ 攻击 DDH 假设。如果 $\mathcal{A}$ 能够以不可忽略的优势 $\varepsilon(\lambda)$ 攻破 IND-CPA 安全性，$\mathcal{B}$ 可以不可忽略的优势攻破 DDH 假设。

Initialize：$\mathcal{A}$ 确定挑战用户实体集合 $U$，合谋用户实体集合 $V$（$|V|=n-1$）以及两个等长的消息 $M_0$ 和 $M_1$ 发送给 $\mathcal{B}$。在 M-PSI 协议中存在 $n$ 个实体，参与者 $P_1\in U_1,P_2\in U_2,\ldots,P_n\in U_n$，参与用户实体 $P_1^i,\ldots,P_n^i(i\in\mathbb{Z})$ 为执行第 $i$ 次 M-PSI 协议的一组实体，用户实体集合表示为 $U\in U_1\cup\cdots\cup U_n$。

Setup：接收安全参数 $\lambda$，$\mathcal{B}$ 运行 KeyGen 算法生成 $n$ 个公私钥对 $(pk_1,sk_1),\ldots,(pk_n,sk_n)$，公开系统公钥 $pk=pk_1\cdots pk_n$，保留私钥 $sk_1\cdots sk_n$，则 $\mathcal{B}$ 的挑战四元组为 $T=(g,pk,g^{r_1+\cdots+r_n},Z)$。

Queries：FPSI-CA 协议中所涉及到的 oracle 询问如下：

1) Hash $H(M)$(or $H'(M)$)：若在列表 $L$（或 $L'$）中已存在记录 $(M,r)$，则返回 $r$；否则，随机选取 $r\in G$，将记录 $(M,r)$ 存储在列表 $L$（或 $L'$）中，并返回 $r$。
2) Send $(U^i,a,Start)$：对于 $l\in[1,m]$，随机选择 $s_l^a$，计算 $F_{k_a}(x_l^1)$（$a\in[2,n]$），

$P_a(x)=s_l^a+\prod_{j=1}^{b}(x-F_{k_a}(x_{lj}^a))$，并返回 $F_{k_a}(x_l^1)$， $s_l^a$ 和 $P_a(x)$。

3) Send $(U^i,a,F_{k_a}(x_l^1),P_a(x))$：计算 $s_l^{a1}=P(F_{k_a}(x_l^1))$，返回 $s_l^{a1}$。

4) Send $(U^i,a,s_l^a,s_l^{a1})$：根据 $s_l^a$ 和 $s_l^{a1}$ 的关系，选取一个大于 0 的随机数 $t_l^a$，计算 $c_{l1}^a=t_l^a$ 和 $c_{l2}^a=e_l^a+t_l^a$，并返回 $c_{l1}^a$ 和 $c_{l2}^a$。

5) Send $(U^i,a,c_{l2}^a)$：选取随机数 $r_{l2}^a$，计算 $E(c_{l2}^a)=g^{H(c_{l2}^a)}\cdot pk^{r_{l2}^a}$，并返回 $E(c_{l2}^a)$。

6) Send $(U_1^i,c_{l1}^a,E(c_{l2}^a))$：对于 $a\in[2,n]$，选取随机数 $r_{l1}^a$，计算 $E(-c_{l1}^a)=g^{H(-c_{l1}^a)}\cdot pk^{r_{l1}^a}$，$E(c_l)=\prod_{a=2}^{n}(E(c_{l2}^a)\cdot E(-c_{l1}^a))$，并返回 $E(c_l)$。

7) Execute $(U_1^i,U_2^i)$：基于 Send 询问的成功模拟情况，返回 $F_{k_a}(x_l^1),s_l^a,P_a(x)\leftarrow$ Send$(U^i,a,Start)$，$s_l^{a1}\leftarrow$ Send $(U^i,a,F_{k_a}(x_l^1),P_a(x))$，$c_{l1}^a,c_{l2}^a\leftarrow$ Send $(U^i,a,s_l^a,s_l^{a1})$，$E(c_{l2}^a)\leftarrow$ Send $(U^i,a,c_{l2}^a)$ 和 $E(c_l)\leftarrow$ Send $(U_1^i,c_{l1}^a,E(c_{l2}^a))$，其中 $a=2,\ldots,n$。

8) Collude $(V^i)$：若 $P_1^i\in V^i,P_n^i\notin V^i$，返回 $X_1,\ldots,X_{n-1}$，$sk_1,\ldots,sk_{n-1}$，$F_{k_1}(x_l^1),\ldots,F_{k_n}(x_l^1)$，$s_l^2,\ldots,s_l^{n-1}$，$P_2(x),\ldots,P_n(x)$，$s_l^{21},\ldots,s_l^{(n-1)1}$，$c_{l1}^2,\ldots,c_{l1}^n$，$c_{l2}^2,\ldots,c_{l2}^{n-1}$，$E(c_{l2}^2),\ldots,E(c_{l2}^n)$ 和 $E(c_l)$；若 $P_1^i\notin V^i,P_n^i\in V^i$，返回 $X_2,\ldots,X_n$，$sk_2,\ldots,sk_n$，$s_l^2,\ldots,s_l^n$，$P_2(x),\ldots,P_n(x)$，$c_{l2}^2,\ldots,c_{l2}^n$，$E(c_{l2}^2),\ldots,E(c_{l2}^n)$ 和 $E(c_l)$。

Challenge：$\mathcal{B}$ 随机选取一个 $b\leftarrow_R\{0,1\}$，加密 $M_b$。选取两个随机数 $r_1$ 和 $r_2$，计算 $E(M_b)=(g^{r_1+\cdots+r_n},g^{H(M_b)}\cdot Z)$ 并返回密文 $E(M_b)$。

Test：$\mathcal{A}$ 根据密文 $E(M_b)$ 以及上述 oracle 询问中得到的信息输出对于 $b$ 的猜测 $b'$，若 $b'=b$ 则返回 1，表示 $(g,pk,g^{r_1+\cdots+r_n},Z)$ 为 DDH 四元组分布；若 $b'\neq b$ 则返回 0，表示 $(g,pk,g^{r_1+\cdots+r_n},Z)$ 为随机四元组分布。

$$\mathrm{Adv}_{\mathcal{B}}^{DDH}(\lambda)=\left|\Pr[\mathcal{B}(T)=1]-\Pr[\mathcal{B}(T)=0]\right|=\left|\Pr[Succ]-1/2\right|=\mathrm{Adv}_{M-PSI}^{IND-CPA}(\mathcal{A})$$

进一步通过一系列 hybrid 游戏 $\mathrm{Exp}_n(0\le n\le 4)$ 来证明定理 6。从游戏 $\mathrm{Exp}_0$ 敌手真实的攻击开始，到游戏 $\mathrm{Exp}_4$ 敌手没有任何优势时结束：

游戏 $\mathrm{Exp}_0$：在 random oracle 模型中，$\mathrm{Exp}_0$ 与真实的攻击相对应，由定义 2 可得：

$$\mathrm{Adv}_{M-PSI}^{IND-CPA}(\mathcal{A})=\Pr[Succ_0]-1/2$$

游戏 $\mathrm{Exp}_1$：通过维持列表 $L$（及 $L'$）中的记录来模拟 oracle 哈希 $H:\{0,1\}^*\rightarrow G$（及 $\mathrm{Exp}_3$ 中的 $H'$），其余部分同 $\mathrm{Exp}_0$ 一样，仍与真实的攻击相对应。因此可知 $\mathrm{Exp}_1$ 与现实中敌手攻击的情况是不可区分的，故有：

$$\left|\Pr[Succ_1]-\Pr[Succ_0]\right|\approx 0$$

游戏 $\mathrm{Exp}_2$：$\mathrm{Exp}_2$ 与 $\mathrm{Exp}_1$ 一样对 oracle 哈希 $H$ 和 $H'$ 进行了模拟，但是中止了在 $H(c_{l1}^a)$

和$H(c_{l1}^a)$中存在哈希碰撞的执行。根据生日悖论可知，在发送$q_h$次的 random oracle 询问、$q_{exe}$次的 Execute 询问和$q_{send}$次的 Send 询问时，模拟$H(c_{l1}^a)$和$H(c_{l1}^a)$发生碰撞的概率最多可能为$(q_h^2+(q_{send}/5+q_{exe})^2)/2p$，因此有：

$$\left|\Pr[Succ_2]-\Pr[Succ_1]\right|\le\frac{q_h^2+((1/5)\cdot q_{send}+q_{exe})^2}{2p}$$

游戏$\mathrm{Exp}_3$：$\mathrm{Exp}_3$中，使用 oracle 哈希$H'$代替$H$来计算$E'(c_{l1}^a)=(g^{H'(c_{l1}^a)}\cdot pk^{r_{l1}^a})$和$E'(c_{l2}^a)=(g^{H'(c_{l2}^a)}\cdot pk^{r_{l2}^a})$，其余部分与$\mathrm{Exp}_2$仍相同，$E'(c_{l1}^a)$和$E'(c_{l2}^a)$与$E(c_{l1}^a)$和$E(c_{l2}^a)$是完全独立的。游戏$\mathrm{Exp}_3$与$\mathrm{Exp}_2$是完全不可区分的除非有事件$\mathrm{AskH}_3$发生：$\mathcal{A}$对 oracle 哈希$H$关于$E(c_{l1}^a)$和$E(c_{l2}^a)$进行了询问。此外，无论 Challenge 阶段中$b$如何选取，$\mathcal{A}$对于其的猜测都是随机的，故可知：

$$\left|\Pr[Succ_3]-\Pr[Succ_2]\right|\le\Pr[\mathrm{AskH}_3],$$
$$\Pr[Succ_3]=1/2$$

游戏$\mathrm{Exp}_4$：$\mathrm{Exp}_4$中，通过 DDH 随机自归约的属性来模拟协议的执行。给定一个 DDH 实例$(A=g^{c_{l1}^a},B=g^{c_{l2}^a})$，随机选取$\alpha,\beta\in\mathbb{Z}_p$，令$E(c_{l1}^a)=A^\alpha$，$E(c_{l2}^a)=B^\beta$，则存在 DDH 四元组$DDH(g,A,B,DDH(E(c_{l1}^a),E(c_{l2}^a)))$。由于事件$\mathrm{AskH}_4$同样指$\mathcal{A}$对 oracle 哈希$H$关于$E(c_{l1}^a)$和$E(c_{l2}^a)$进行了询问，且存在：

$$DDH(E(x),E(y))=DDH(A^\alpha,B^\beta)=DDH(A,B)^{\alpha\beta}$$

故可知

$$\Pr[\mathrm{AskH}_3]=\Pr[\mathrm{AskH}_4]$$
$$\Pr[\mathrm{AskH}_4]\le q_h\mathrm{Adv}_{\mathcal{A}}^{DDH}(t+\frac{(1/5)\cdot q_{send}+q_{exe}+1}{p^2})$$

综上可得：

$$\mathrm{Adv}_{OF-PSI}^{IND-CPA}(\mathcal{A})\le q_h\mathrm{Adv}_{\mathcal{A}}^{DDH}(t+\frac{(1/5)\cdot q_{send}+q_{exe}+1}{p^2})+\frac{q_h^2+((1/5)\cdot q_{send}+q_{exe})^2}{2p}$$

故可知，M-PSI 协议是 IND-CPA 安全的，且能够抵抗任意$(n-1)$个参与者的合谋攻击。

## 5.5 性能分析（Performance Analysis）

### 5.5.1 理论分析

本节从计算复杂度、通信复杂度和是否设定参与方共有集合三个方面对 M-PSI 协议与文献[52]和文献[56]中的 PSI 协议进行对比分析，具体如表 5-1 所示。其中$n$为协议的参与用户个数，$D$表示文献[52]所有参与者共同集合元素的个数，$N=\max\{n_1,\ldots,n_n\}$为$n$个参与者集合中所含元素最多的集合的元素个数。

**表 5-1 M-PSI 协议与相关协议对比分析**

**Tab.5-1 Comparison analysis of M-PSI protocol and related protocols**

| 协议 | 计算复杂度 | 通信复杂度 | 是否设定共有集合 |
|---|---|---|---|
| [52] | $4nD$ | $(2n-1)D$ | 是 |
| [56] | $(n-1)(14N/\log 2+16N)$ | $(n-1)(7N/\log 2+4N)$ | 否 |
| M-PSI | $6(n-1)N+2n$ | $3(n-1)N$ | 否 |

（1）计算复杂度

M-PSI 协议中，基于 ElGamal 的$(n,n)$门限加密算法加密数据时产生的计算开销为协议的主要计算开销，故使用加密时所执行模幂运算的次数来表示 M-PSI 协议的计算复杂度。参与者$P_1$在加密数据时执行了$2N(n-1)$次模幂运算，在解密时执行了$2N$次模幂运算；$P_2,\ldots,P_n$在加密数据时分别执行了$2N$次模幂运算，解密时同样分别执行了$2N$次模幂运算。因此 M-PSI 协议共执行了$6N(n-1)+2n$次模幂运算。文献[52]同样使用 ElGamal 门限加密算法加密数据，协议中$P_1,\ldots,P_n$均分别在加密时执行了$2D$模幂运算，解密时执行了$2D$次模幂运算，因次文献[52]中协议共执行了$4nD$次的模幂运算。文献[56]使用 Paillier 门限加密算法加密数据，其中$n$个参与者$P_1,\ldots,P_n$在加密时共执行了$(14N/\log 2+14N)(n-1)$次模幂运算，在解密时共执行了$2N(n-1)$次模幂运算，因此文献[56]整体执行了$(14N/\log 2+16N)(n-1)$次模幂运算。

（2）通信复杂度

使用协议执行过程中传送的密文数量来表示协议的通信开销。M-PSI 协议中，参与者$P_2,\ldots,P_n$发送了$2(n-1)N$个密文给$P_1$，$P_1$分别给$P_2,\ldots,P_n$发送了$N$个密文，故 M-PSI 协议中共产生密文传送的数量为$3(n-1)N$。文献[52]中，每个参与者传送$D$个密文给下一个参与者，产生的密文传递数量为$nD$，$P_1$在解密时又分别对$P_2,\ldots,P_n$发送了$D$个密文，因此文献[52]中参与者共传送了$(2n-1)D$个密文。文献[56]中，$P_1,\ldots,P_{n-1}$发送了$(n-1)(7N/\log 2+2N)$个密文给$P_n$，$P_n$发送了$2(n-1)N$个密文给$P_1,\ldots,P_{n-1}$，文献[56]中共传输了$(n-1)(7N/\log 2+4N)$个密文。

综上可以看出，M-PSI 协议在没有设定所有参与者共有集合时对$n$个集合进行预处理虽然增加了一些复杂度，但相较同样没有设定所有参与者共有集合的文献[56]而言 M-PSI 协议的计算复杂度和通信复杂度均较低。

### 5.5.2 实验分析

本小节通过实验对 M-PSI 协议与文献[52, 56]中协议在运算方面所耗的时间进行对比分析。实验平台：Windows10 64 位操作系统，处理器： AMD Ryzen 5 4600H Radeon Graphics 3.00 GHz 16gb RAM，编译环境：My Eclipse 2017。

M-PSI 协议与文献[52]和[56]中的协议均使用门限公钥加密算法来对数据进行加密，首先比较三个协议在不同模数位长的情况下加密数据时执行模幂运算所耗的时间。在本次实验中，设置参与者个数 $n=10$，集合元素个数 $D=N=1000$，模数长度分别为 128 bit、256 bit、512 bit 和 1024 bit 时，三个协议分别执行模幂运算所需的时间如图 5-2 所示。之后进一步比较三个协议在不同集合元素个数下执行所需的时间，在该实验中，参与者个数 $n=10$，模数位长固定为 1024 bit，在集合元素个数分别为 $2^{10}$、$2^{11}$、$2^{12}$、$2^{13}$、$2^{14}$ 和 $2^{15}$ 时，三个协议执行时所耗的时间如图 5-3 所示。

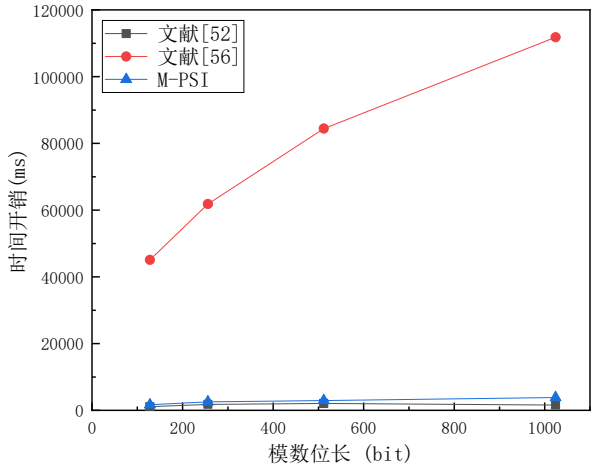



**图 5-2 不同模数位长时执行模幂运算时间开销**

**Fig.5-2 Time cost to perform modular exponentiation of different modulus length**

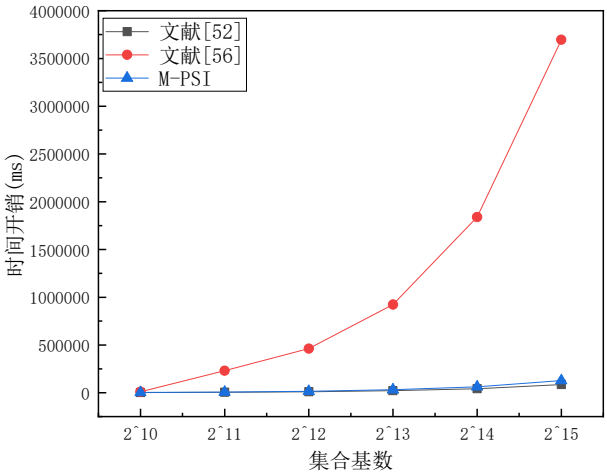



**图 5-3 不同集合元素个数时 M-PSI 协议时间开销**

**Fig.5-3 Time cost of M-PSI protocol with different set cardinality**

根据图 5-2 和图 5-3 可知，随着模数位长和集合元素个数的不断增加，M-PSI 协议与文献[52]和文献[56]中执行模幂运算的时间开销均随之不断增长。其中文献[56]增长速度最快，M-PSI 协议与文献[52]增长趋势较缓且两者之间差距较小。故可知，M-PSI 协议在没有设定所有参与者共有集合时达到了与设定有所有参与

者共有集合的文献[52]相近的计算开销，其复杂度远远低于同样没有设定所有参与者共有集合的文献[56]中协议的复杂度。

## 5.6 本章小结（Summary of This Chapter）

本章主要讨论了多方之间的隐私保护集合交集计算问题，提出了一种能够在不预先设定共同集合的情况下实现多方隐私集合交集计算的 M-PSI 协议。方案使用哈希算法预先对各方的集合元素进行分类存储，以哈希桶为单位使得 $P_1$ 分别与 $P_2,\ldots,P_n$ 执行等值比较协议。方案采用基于 ElGamal 的 $(n,n)$ 门限加密算法，在保证所有参与者共同掌握交集元素信息的同时不泄露参与者集合中的任何信息，确保多方之间的交集计算正确安全的进行。最后通过理论分析和实验评估，表明本文提出的 M-PSI 协议在没有预先设定共同集合的同时具备较低的计算复杂度。

# 6 总结与展望

# 6. Summary and Prospect

本章对全文的研究工作进行总结，指出安全多方计算领域的集合交集问题研究中仍存在的问题，并给出未来要进行研究的方向。

## 6.1 总结（Summary）

隐私保护的安全多方计算集合交集问题在充分利用数据进行交集或交集势计算时，可以保护隐私数据不被泄露，成为不侵犯用户隐私的情况下利用用户大量的数据进行信息共享的关键技术，具有重要的理论和实际研究价值。本文在现有安全多方计算集合交集问题协议的研究基础上，对两方之间的交集和交集势计算问题以及多方之间的交集计算问题，进行以下研究：

（1）针对两方之间的隐私集合交集计算问题，提出一个外包的公平两方隐私集合交集协议。协议采用布谷鸟哈希和简单哈希映射算法将双方集合元素映射入相应的哈希桶中，只需对两个哈希表中的元素以桶对桶的方式比较就可以计算交集，减少了集合元素的比较次数，降低了协议的计算成本；利用 ElGamal 门限加密算法加密哈希表中的元素，以确保参与双方哈希桶内的元素可在密文的形式下进行比较，使得用户可以正确安全地计算交集，实现了协议的数据隐私性；引入半诚实的服务器作为第三方，将集合元素的比较工作转移到服务器上，进一步降低了用户的计算负担，同时允许两个参与用户并行处理集合元素，确保在协议执行结束后，双方可以同时得知交集的结果，实现了协议的公平性。

（2）针对两方之间的隐私集合交集势计算问题，提出一个公平的两方隐私集合交集势协议。协议中参与双方分别使用布谷鸟哈希算法和简单哈希算法对集合元素进行预处理，将一方哈希桶内的元素与另一方相应哈希桶内的元素进行对比即可进行交集势计算，减少了双方所有集合元素一对一进行比较时的次数，降低了协议的计算成本；同时，根据可交换加密中使用不同用户的公钥对消息明文进行多次不同顺序的加密后对应的密文仍相等的特性，对双方存储在哈希表中的集合元素加密，确保参与双方可在密文形式下对哈希桶内的元素进行比较，在正确计算集合交集势的同时能够保证数据的隐私性；此外，双方并行完成协议的执行，保证了双方能够同时得知协议计算结果，确保了参与双方之间的公平性。本方案在实现公平性的同时达到较低的计算和通信复杂度，协议整体性能较优。

（3）针对多方之间的隐私集合交集计算问题，构造出一个安全高效的多方隐私集合交集协议。协议采用布谷鸟哈希和简单哈希映射算法将双方集合元素映射入相应的哈希桶中来预先对集合元素进行处理，利用两方之间的等值比较协议来共享第一个参与者与其他参与者之间的交集信息；此外，使用$(n,n)$门限加密算法，在不泄露参与者集合中的任何信息的情况下对所有参与者的交集元素信息进行汇总，得到多方之间的交集结果。最后，经过理论和实验分析，表明本方案在没有预先设定共同集合的情况下达到与设定有共同集合的多方隐私集合交集协议相近的复杂度，远远低于其他同样没有预先设定共同集合的多方隐私集合交集协议。

## 6.2 展望（Prospect）

本文主要对安全多方计算领域的集合交集问题进行研究，通过对集合交集方案的仔细分析，可以发现目前的理论和实际应用方案研究仍然存在一些问题，需要在未来继续深入展开，因此未来将以以下两方面作为研究重点：

（1）研究基于边缘辅助物联网的集合交集方案。边缘计算和物联网技术的发展与结合提升了物联网设备的智能化，将智能设备引入生活的方方面面，给我们的生活带来了便利。但所收集的敏感和私人数据也更容易受到攻击，物联网大数据的隐私保护和安全共享已成为当前研究的重点。因此，研究更加安全高效的集合交集计算方案应用于边缘辅助物联网是我们未来的研究方向。

（2）研究基于恶意敌手模型的集合交集方案。本文所构造的方案中，基于的安全模型中的敌手都是半诚实的，其可以窃听协议执行过程中传送的消息，但不能主动破坏协议的执行。然而在实际生活中，敌手大多为恶意的，为了得到参与者的隐私信息，在窃听通信通道的同时也会主动攻击或者篡改参与者传输的信息。半诚实的安全模型在该情况下不足以保证协议的安全性，因此设计基于恶意敌手模型的集合交集方案具有重要的实际应用价值，是我们未来的研究重点之一。

# 参考文献

[1] Huang Y. Secure multi-party computation[M]//Responsible Genomic Data Sharing. Academic Press, 2020: 123-134.

[2] Pathak A, Patil T, Pawar S, et al. Secure Authentication using Zero Knowledge Proof[C]//2021 Asian Conference on Innovation in Technology (ASIANCON). IEEE, 2021: 1-8.

[3] Costa B, Branco P, Goulão M, et al. Randomized Oblivious Transfer for Secure Multiparty Computation in the Quantum Setting[J]. Entropy, 2021, 23(8): 1001.

[4] 何苗, 柏粉花, 于卓, 沈韬.区块链中可公开验证密钥共享技术[J].浙江大学学报(工学版), 2022, 56(02): 306-312.

[5] Bay A, Erkin Z, Alishahi M, et al. Multi-Party Private Set Intersection Protocols for Practical Applications[C]//Vimercati, S. De Capitani di (ed.), SECRYPT 2021: Proceedings of the 18th International Conference on Security and Cryptography, July 6-8, 2021. Setubal: SciTePress, 2021: 515-522.

[6] 魏立斐, 刘纪海, 张蕾, 王勤, 贺崇德.面向隐私保护的集合交集计算综述[J].计算机研究与发展, 2021: 1-18.

[7] Qian Y , Shen J , Vijayakumar P , et al. Profile Matching for IoMT: A Verifiable Private Set Intersection Scheme[J]. IEEE Journal of Biomedical and Health Informatics, 2021, 25 (10), 3794-3803.

[8] Alam T. Cloud Computing and its role in the Information Technology[J]. IAIC Transactions on Sustainable Digital Innovation (ITSDI), 2021, 1: 108-115.

[9] Machidon O, Fajfar T, Pejovic V. Implementing Approximate Mobile Computing[C]//Proc. of the 2020 Workshop on Approximate Computing Across the Stack (WAX). 2020: 1-3.

[10] Nižetić S, Šolić P, González-de D L I, et al. Internet of Things (IoT): Opportunities, issues and challenges towards a smart and sustainable future[J]. Journal of Cleaner Production, 2020, 274: 122877.

[11] Zhong Y, Xu Z H, Cao L. Intelligent IoT-based telemedicine systems implement for smart medical treatment[J]. Personal and Ubiquitous Computing, 2021: 1-11.

[12] Yao A C. Protocols for secure computations[C]//23rd annual symposium on foundations of computer science (sfcs 1982). IEEE, 1982: 160-164.

[13] Yao A C C. How to generate and exchange secrets[C]//27th Annual Symposium on Foundations of

Computer Science (sfcs 1986). IEEE, 1986: 162-167.

[14] Goldreich O . How to play ANY mental game[J]. Stoc, 1987.

[15] 周素芳, 窦家维, 郭奕旻, 等.安全多方向量计算[J].计算机学报, 2017, 40(5): 1134-1150.

[16] Yang J, Li Y, Liu Q, et al. Brief introduction of medical database and data mining technology in big data era[J]. Journal of Evidence‐Based Medicine, 2020, 13(1): 57-69.

[17] 郭奕旻, 周素芳, 窦家维, 等.高效的区间保密计算及应用[J].计算机学报, 2017, 40(7): 1-16.

[18] Clifton C, Kantarcioglu M, Vaidya J, et al. Tools for privacy preserving distributed data mining[J]. ACM Sigkdd Explorations Newsletter, 2002, 4(2): 28-34.

[19] Alamati N, Branco P, Döttling N, et al. Laconic private set intersection and applications[C]//Theory of Cryptography Conference. Springer, Cham, 2021: 94-125.

[20] 巩林明, 王道顺, 刘沫萌, 高全力, 邵连合, 王明明.基于无匹配差错的 PSI 计算[J].计算机学报, 2020, 43(09): 1769-1790.

[21] 张静, 罗守山, 杨义先, 辛阳.安全两方集合交集云外包计算协议[J].北京邮电大学学报, 2019, 42(02): 13-18.

[22] Wang Y, Huang Q, Li H, Xiao M, Ma S, Susilo W. Private Set Intersection With Authorization Over Outsourced Encrypted Datasets[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 4050-4062.

[23] Qiu S, Zhang Z, Liu Y, Yan H, Cheng Y. SE-PSI: Fog/Cloud server-aided enhanced secure and effective private set intersection on scalable datasets with Bloom Filter[J]. Mathematical Biosciences and Engineering, 2022;19(2):1861-76.

[24] Liu B, Ruan O, Shi R, et al. Quantum private set intersection cardinality based on bloom filter[J]. Scientific Reports, 2021, 11(1): 1-9.

[25] Jolfaei AA, Mala H, Zarezadeh M. EO-PSI-CA: Efficient outsourced private set intersection cardinality[J]. Journal of Information Security and Applications, 2022 Mar 1;65:102996.

[26] 程楠, 赵运磊.一种高效的关于两方集合并/交集基数的隐私计算方法[J].密码学报, 2021, 8(02): 352-364.

[27] 马敏耀, 陈松良, 左羽.基于 Goldwasser-Micali 加密系统的隐私交集基数协议研究[J].计算机应用研究, 2018, 35(09): 2748-2751.

[28] Raghuvir Y A, Govindarajan S, Vijayakumar S, et al. Advancement on Security Applications of Private Intersection Sum Protocol[C]//Proceedings of the Future Technologies Conference. Springer, Cham, 2021: 104-116.

[29] Agrawal R, Evfimievski A, Srikant R. Information sharing across private databases[C]//Proceedings of the 2003 ACM SIGMOD international conference on Management of data. 2003: 86-97.

[30] Freedman M J, Nissim K, Pinkas B, et al. Efficient Private Matching and Set Intersection [M]// Springer. Advances in Cryptology-Eurocrypt 2004. Heidelberg: Springer, Berlin, 2004: 1-19.

[31] 张正, 张方国.混淆电路与不可区分混淆[J].密码学报, 2019, 6(05): 541-560.

[32] Garimella G, Pinkas B, Rosulek M, et al. Oblivious key-value stores and amplification for private set intersection[C]//Annual International Cryptology Conference. Springer, Cham, 2021: 395-425.

[33] 华文镝, 高原, 吕萌, 谢平.布隆过滤器研究综述[J].计算机应用, 2022: 1-22.

[34] Huang Y, Evans D, Katz J. Private set intersection: Are garbled circuits better than custom protocols?[C]//NDSS. 2012.

[35] Pinkas B, Schneider T, Tkachenko O, et al. Efficient circuit-based PSI with linear communication[C]//Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Cham, 2019: 122-153.

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

[37] Kavousi A, Mohajeri J, Salmasizadeh M. Efficient scalable multi-party private set intersection using oblivious prf[C]//International Workshop on Security and Trust Management. Springer, Cham, 2021: 81-99.

[38] Debnath S K, Sakurai K, Dey K and Kundu N. Secure Outsourced Private Set Intersection with Linear Complexity[C]//2021 IEEE Conference on Dependable and Secure Computing (DSC), 2021: 1-8.

[39] Abadi A , Terzis S , Metere R , et al. Efficient Delegated Private Set Intersection on Outsourced Private Datasets[J]. IEEE Transactions on Dependable and Secure Computing, 2019, 16(4): 608-624.

[40] Kavousi A, Mohajeri J, Salmasizadeh M. Improved secure efficient delegated private set intersection[C]//2020 28th Iranian Conference on Electrical Engineering (ICEE). IEEE, 2020: 1-6.

[41] Kissner L, Song D. Privacy-preserving set operations[C]//Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 2005: 241-257.

[42] Camenisch J, Zaverucha G M. Private intersection of certified sets[C]//International Conference on Financial Cryptography and Data Security. Springer, Berlin, Heidelberg, 2009: 108-127.

[43] Kim M, Lee H T, Cheon J H. Mutual private set intersection with linear complexity[C]//International Workshop on Information Security Applications. Springer, Berlin, Heidelberg, 2011: 219-231.

[44] Dong C, Chen L, Camenisch J, et al. Fair private set intersection with a semi-trusted arbiter[C]//IFIP

Annual Conference on Data and Applications Security and Privacy. Springer, Berlin, Heidelberg, 2013: 128-144.

[45] Debnath S K, Dutta R. Towards fair mutual private set intersection with linear complexity[J]. Security and Communication Networks, 2016, 9(11): 1589-1612.

[46] Debnath S K, Dutta R. Fair mPSI and mPSI-CA: Efficient constructions in prime order groups with security in the standard model against malicious adversary[J]. Cryptology ePrint Archive, 2016.

[47] Brandt F. Efficient Cryptographic Protocol Design Based on Distributed El Gamal Encryption[C]//Information Security & Cryptology-icisc, International Conference, Seoul, Korea, December, Revised Selected Papers. Springer-Verlag, 2005.

[48] Cramer R. A practical public key cryptosystem provably secure against adaptive chosen ciphertext attack[J]. Proc Crypto, 1998.

[49] Debnath S K, Dutta R. New realizations of efficient and secure private set intersection protocols preserving fairness[C]//International Conference on Information Security and Cryptology. Springer, Cham, 2016: 254-284.

[50] Lai P K Y, Yiu S M, Chow K P, et al. An Efficient Bloom Filter Based Solution for Multiparty Private Matching[C]//Security and Management. 2006: 286-292.

[51] 李顺东, 周素芳, 郭奕旻, 窦家维, 王道顺. 云环境下集合隐私计算. 软件学报, 2016, 27(6): 1549-1565.

[52] 窦家维, 刘旭红, 周素芳, 等. 高效的集合安全多方计算协议及应用[J]. 计算机学报, 2018, 41(8):17: 1844-1860.

[53] Badrinarayanan S, Miao P, Raghuraman S, et al. Multi-party threshold private set intersection with sublinear communication[C]//IACR International Conference on Public-Key Cryptography. Springer, Cham, 2021: 349-379.

[54] Hazay C, Venkitasubramaniam M. Scalable multi-party private set-intersection[C]//IACR International Workshop on Public Key Cryptography. Springer, Berlin, Heidelberg, 2017: 175-203.

[55] Miyaji A , Nakasho K , Nishida S . Privacy-Preserving Integration of Medical Data: A Practical Multiparty Private Set Intersection[J]. Journal of Medical Systems, 2017, 41(3):37.

[56] Bay A, Erkin Z, Hoepman J H, et al. Practical Multi-party Private Set Intersection Protocols[J]. IEEE Transactions on Information Forensics and Security, 2021, 17: 1-15.

[57] Cristofaro E D , Gasti P , Tsudik G . Fast and Private Computation of Cardinality of Set Intersection and Union[C]// Cans. Springer, Berlin, Heidelberg, 2012.

[58] Debnath S K, Dutta R. Secure and efficient private set intersection cardinality using bloom filter[C]//International Conference on Information Security. Springer, Cham, 2015: 209-226.

[59] Davidson A, Cid C. An efficient toolkit for computing private set operations[C]//Australa- sian Conference on Information Security and Privacy. Springer, Cham, 2017: 261-278.

[60] Tajima A, Sato H, Yamana H. Outsourced private set intersection cardinality with fully homomorphic encryption[C]//2018 6th International Conference on Multimedia Computing and Systems (ICMCS). IEEE, 2018: 1-8.

[61] Lv S, Ye J, Yin S, et al. Unbalanced private set intersection cardinality protocol with low communication cost[J]. Future Generation Computer Systems, 2020, 102: 1054-1061.

[62] Boneh D. The decision diffie-hellman problem[C]//International Algorithmic Number Theory Symposium. Springer, Berlin, Heidelberg, 1998: 48-63.

[63] Debnath S K , Dutta R . Provably Secure Fair Mutual Private Set Intersection Cardinality Utilizing Bloom Filter[C]// International Conference on Information Security and Cryptology. Springer, Cham, 2016.

[64] Debnath S K. Secure computation of private set intersection cardinality with linear complexity[M]//Cryptographic Security Solutions for the Internet of Things. IGI Global, 2019: 142-180.

[65] Ohata S, Nuida K. Communication-efficient (client-aided) secure two-party protocols and its application[C]//International Conference on Financial Cryptography and Data Security. Springer, Cham, 2020: 369-385.

[66] 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.

[67] Varghese P E, Nair L R. A STUDY ON THE EXISTING THRESHOLD CRYPTOGRAPHY TECHNIQUES[J]. International Journal of Advanced Research in Computer Science, 2020, 11(5).

[68] Huang K, Tso R. A commutative encryption scheme based on ElGamal encryption[C]//2012 International Conference on Information Security and Intelligent Control. IEEE, 2012: 156-159.

[69] Khayat S H. Using commutative encryption to share a secret[J]. Electrical Engineering Department, Ferdowsi University of Mashhad, Iran, 2008.

[70] Zhao S, Song X, Jiang H, et al. An Efficient Outsourced Oblivious Transfer Extension Protocol and Its Applications[J]. Security and Communication Networks, 2020, 2020.

[71] Kolesnikov V, Kumaresan R, Rosulek M, et al. Efficient batched oblivious PRF with applications to

private set intersection[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. 2016: 818-829.

[72] 宋祥福, 盖敏, 赵圣楠, 蒋瀚. 面向集合计算的隐私保护统计协议[J]. 计算机研究与发展, 2020, 57(10): 2221-2231.

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

[74] Fan B, Andersen D G, Kaminsky M, et al. Cuckoo filter: Practically better than bloom [C]//Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies. 2014: 75-88.

[75] Wang C, Wang D, Tu Y, et al. Understanding node capture attacks in user authentication schemes for wireless sensor networks[J]. IEEE Transactions on Dependable and Secure Computing, 2020.

[76] Wang D, Wang P. Two Birds with One Stone: Two-Factor Authentication with Security Beyond Conventional Bound[J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(4): 708-722.

[77] Hemenway Falk B, Noble D, Ostrovsky R. Private set intersection with linear communication from general assumptions[C]//Proceedings of the 18th ACM Workshop on Privacy in the Electronic Society. 2019: 14-25.

[78] Miao P, Patel S, Raykova M, Seth K and Yung M, Two-sided malicious security for private intersection-sum with cardinality[C]//Annual International Cryptology Conference, 2020: 3-33.

[79] 田美金, 马建峰, 刘志全，等. 一种改进 PSI 协议的基因数据隐私保护方案[J]. 西安电子科技大学学报, 2020, 47(4):8.

# 学位论文数据集

| 关键词* | 密级* | 中图分类号* | UDC | 论文资助 |
|---|---|---|---|---|
| 安全多方计算；集合交集；集合交集势；隐私保护 | 公开 | TP309 | 004 | 国家自然科学基金 |

| 学位授予单位名称* | 学位授予单位代码* | 学位类别* | 学位级别* |
|---|---|---|---|
| 河南理工大学 | 10460 | 工学 | 硕士 |

| 论文题名* | 并列题名* | 论文语种* |
|---|---|---|
| 安全多方计算集合交集问题研究 | Research on Set Intersection of Secure Multi-party Computation | 中文 |

| 作者姓名* | 秦榕霞 | 学号* | 211909010005 |
|---|---|---|---|
| 培养单位名称* | 培养单位代码* | 培养单位地址 | 邮编 |
| 河南理工大学 | 10460 | 河南省焦作市 | 454003 |
| 学科专业* | 研究方向* | 学制* | 学位授予年* |
| 计算机科学与技术 | 网络与信息安全 | 3 年 | 2022 年 |

| 论文提交日期* | 2022 年　5 月　21 日 |
|---|---|

| 导师姓名* | 张静 | 职称* | 副教授 |
|---|---|---|---|

| 评阅人 | 答辩委员会主席* | 答辩委员会成员 |
|---|---|---|
| | 甘勇 | |

电子版论文提交格式　文本（ √ ）　图像（　）　视频（　）　音频（　）

多媒体（　）其他（　）

推荐格式：Microsoft Word(DOC)；Adobe Reader（PDF）

| 电子版论文出版（发布）者 | 电子版论文出版（发布）地 | 权限声明 |
|---|---|---|
| | | |

| 论文总页数* | 59 |
|---|---|

注：共 33 项，其中带*为必填数据，为 22 项。