单云环境下强隐私保护的多维多重集相似度阈值精确查询方案

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

摘 要 集合相似度查询在现实生活中具有广泛应用，但由于它只允许每个元素出现一次，这限制了其在某些场景下的表达能力，无法描述复杂现象。多重集的特性使其能够更加全面地描述复杂现象，增强数据灵活性和表达力。因此，多重集的相似度阈值查询更具实用性。随着云计算的发展，将数据存储和查询外包给云服务器成为数据拥有者的一个有吸引力的选择。然而，这种数据外包极易泄露数据隐私。为了保护数据隐私，数据拥有者在外包数据之前都要将数据加密，而在外包的密文数据上进行相似度查询就成为一个挑战。本文提出了一种新的保护隐私的相似度阈值查询方案，不仅能够解决多重集相似度的保密查询问题，还能够同时基于数据向量和关键词（两种数据类型）为查询用户提供查询结果。具体而言，我们首先设计了一个基于Jaccard相似度的多重集相似度阈值查询协议，然后通过0-1编码构造向量，结合Paillier密码系统设计了一个可以对不同类型的数据进行高效、准确的并行查询协议，并提出了单云服务器下的多维多重集相似度阈值查询方案。最后，本文使用公认的模拟范例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 隐私保护；多重集；相似度；同态运算；模型
中图法分类号 TP309 **DOI号** 10.11897/SP.J.1016.2025.02430

Strong Privacy-Preserving Scheme for Exact Multi-Dimensional Multiset Similarity Threshold Queries in Single-Cloud Environments

LI Shun-Dong DU Ji-Xin WU Chuan-Yu YU Jia-Tong
（School of Computer Science, Shaanxi Normal University, Xi'an 710062）

Abstract Set similarity queries have broad applications in the Internet of Things (IoT), but since they only allow each element to appear once, their expressive power is limited in certain scenarios, failing to describe complex phenomena. In contrast, the characteristics of multisets enable them to more comprehensively describe complex applications, thus enhancing the flexibility and expressiveness of data. Therefore, multiset similarity threshold queries are more practical in real-life applications. With the development of cloud computing, outsourcing data storage and querying to cloud servers has become an attractive option for data owners. However, this data outsourcing poses a significant risk to data privacy. To protect privacy, data owners must encrypt their data before outsourcing it. Conducting similarity queries on encrypted data is a challenging task. In cloud environments, privacy-preserving query models for multidimensional multisets face several significant challenges. One of the foremost issues is the inefficiency in multidimensional data processing. Existing solutions, which are typically designed for single-dimensional datasets, require separate query on independent datasets. This approach not only reduces operational

收稿日期：2025-01-15；在线发布日期：2025-07-10。本课题得到国家重点研发计划（No. 2022YFB2703001）资助。**李顺东**，博士，教授，博士生导师，中国计算机学会（CCF）会员，主要研究领域为密码学与信息安全。E-mail：shundong@snnu.edu.cn。**杜佶欣**，硕士研究生，主要研究领域为信息安全。**吴川宇**，硕士研究生，主要研究领域为信息安全。**余佳桐**，硕士研究生，主要研究领域为信息安全。

efficiency but also exposes privacy, as it may inadvertently disclose failed query types. Another critical challenge is the vulnerability to frequency leakage. Unlike set queries, which focus on verifying the existence of elements, multiset queries must manage occurrence frequencies. Even when the content is encrypted, frequency distributions can reveal underlying dataset characteristics, enabling semi-honest participants to infer raw data through frequency analysis. Furthermore, high computational complexity remains a significant hurdle. While current homomorphic encryption schemes effectively protect set operations, such as element presence, they fail to jointly protect both element content and frequency without imposing prohibitive overhead. Finally, traditional similarity metrics like Jaccard or cosine similarity fall short in handling multisets, as they ignore frequency differences. This oversight necessitates frequency-aware adaptations to ensure accurate threshold queries. To address these challenges, we propose a novel solution that incorporates several key innovations. Our approach begins with a Jaccard-based multiset similarity protocol that includes lightweight query tokenization, enabling client-side efficiency, and a redesigned frequency-embedded Jaccard metric that reduces computational complexity. Additionally, we introduce a 0-1 encoding and Paillier based multidimensional protocol that enables basic cross-dimensional queries using compact 0-1 encoded vectors and facilitates parallel processing of heterogeneous data, such as vectors and keywords, within a single execution. The protocol also leverages Paillier encryption to prevent distribution analysis and incorporates a unified framework that eliminates partial-failure leaks. Our framework represents the first practical solution tailored for single-cloud IoT environments. It ensures end-to-end privacy for datasets, queries, thresholds, and access patterns. The security of the system is formally proven using the simulation paradigm in semi-honest cloud environments, guaranteeing that it can support dynamic data updates without compromising security. Additionally, the framework natively extends to cosine similarity metrics, addressing the inadequacy of traditional similarity measures. This work establishes a new paradigm for privacy-preserving complex data analytics in resource-constrained cloud environments.

Keywords　privacy-preserving; multiset; similarity query; homomorphic operation; model

1　引　言

网络技术的发展和数字化进程的加速，为人们的生活带来了极大的便利。在临床诊断[1]、智能推荐[2]、社交网络分析[3]等各种物联网应用中，都需要进行集合相似性查询。集合相似性查询，旨在为用户快速找到与特定集合相似的其他集合。例如，在医疗保健中，通过集合相似性查询查找与当前患者相似的先前患者，这些相似患者的信息可以帮助医生做出更准确的临床诊断，其治疗方案和治疗效果可以辅助医生为当前患者制定更好的治疗方案。在社交媒体平台上，通过分析用户的兴趣和互动行为，系统能够推荐与用户兴趣相似的内容或好友，从而提升用户的参与度和满意度。在智能交通中，实时监测交通数据并进行相似性查询，可以识别交通拥堵模式，从而优化信号灯控制和路线规划，减少拥堵。

集合相似度有多种度量方式，其中最常见的包括欧几里得距离、曼哈顿距离、Jaccard系数和余弦相似度。Jaccard相似度作为一种重要的度量方式，其定义为两个集合交集的大小与并集的大小之比。Jaccard相似度的意义在于，它能够直观地反映出两个集合的重叠程度，尤其适用于稀疏数据场景。与其他度量方式相比，Jaccard相似度在处理集合数据时具有明显的优势，因为它不受集合大小的影响，更加专注于集合间的共同元素，从而提供了更可靠的相似度评估。

数据的激增和云技术的进步促使越来越多的用户和企业将数据外包到具有强大计算能力的云服务器上。然而，由于云服务提供商可以直接访问数据库，这带来了严重的隐私问题。例如，对于医疗数据

库，患者的个人信息可能会被泄露。因此，在将数据外包到云之前，对数据进行加密是有必要的。

目前，有许多研究致力于解决加密数据上的集合相似性查询，并提出了几种保护隐私的相似性查询方案。然而，它们大多不能很好地平衡查询效率、安全性和准确性。具体来说，文献[4,5]中的方案通过安全的两方计算实现了集合相似性查询，但不适用于外包计算的场景；基于对称可搜索加密(SSE)和局部敏感哈希(LSH)技术的方案[6-8]，因为查询结果可能存在误判，所以它们只能提供近似的相似性查询，而不能提供精确的相似性查询；虽然文献[9,10]的相似性查询方案能够返回精确的查询结果，但是它们都是在两个云服务器的模型下构建的。然而，涉及的云服务器越多，支出越多，隐私保护也越困难。在一些实际场景中，为了减少云开销，数据拥有者可能更倾向于使用单个云服务器；文献[11]在查询过程中泄露了原有数据库及中间计算结果；其他方案均未保护访问模式隐私[12]，即数据记录中满足查询条件的记录，正如文献[13,14]所述，泄露的访问模式可能会引起推理攻击，从而泄露敏感信息。

集合相似性查询基本上都研究标准集合的相似性问题，但由于标准集合只允许每个元素出现一次，这使得其无法描述复杂现象，从而限制了在某些场景下的表达能力，尤其是在需要统计元素重复次数时，会导致重要信息的丢失。例如，在社交网络中，集合只能记录用户之间的互动对象，无法反映互动的次数，可能导致对用户关系的误判，不适合需要深入分析用户行为的场景。

多重集是集合概念的推广。为了避免混淆，本文中将不允许出现重复元素的集合称为标准集。多重集的特性使其能够更加全面地反映真实情况，从而增强数据处理的灵活性与表达力。因此，在需要考虑元素频率和多样性的实际问题中，多重集显得尤为重要。例如，多重集可以记录互动次数，提供更准确的互动频繁度，并在社交网络的相似性查询中提供更全面和精确的分析能力，从而更好地反映用户之间的紧密联系和相似性。

当前关于集合相似性查询的研究不断增多，但基于多重集的相似性查询仍然是一个尚未被充分探索的领域。现有的研究大多集中于标准集合的相似性查询，忽视了多重集在相似性查询中的独特优势。因此，我们提出多重集相似度阈值查询方案，该方案重点解决了以下四个挑战：

(1)多维多重集查询：现有方案大多适用于单维数据集查询，如果要实现多维数据集的相似度阈值查询，就必须在两个独立的数据集上执行两个独立的查询。这种方式不可避免地会导致查询效率低下，此外，组合使用方案还会在查询失败时泄露哪些查询类型未被满足，从而引发隐私泄露问题。因此，如何通过一次查询有效解决多维多重集相似度阈值查询问题，是一个亟待解决的关键问题。

(2)频率泄露风险：在现有的集合查询中，查询目标是判断一个元素是否存在，而在多重集查询中，除了检查元素是否存在外，还需要考虑元素的重复次数，即使元素内容加密，频率分布特征仍可能暴露数据集特性，如半诚实参与者可通过频率分析推断原始数据。因此，如何处理频率信息并确保其在加密环境中的安全性，是一个全新的挑战。

(3)加密方案的计算复杂性：现有的隐私保护方案，通常专注于集合相关计算的隐私保护(如判断元素是否存在)，很少考虑既保护元素内容又保护它们出现的频率，因此需要设计更复杂的同态加密方案以支持在多重集上的相关计算，并且确保计算复杂度不会大幅度提升。

(4)相似度度量的准确性：现有集合查询中的相似度通常基于集合之间的交集或并集，而在多重集查询中，相似度度量可能需要考虑元素的频率差异。例如，Jaccard相似度或余弦相似度在处理多重集时需要进行相应的调整，确保频率信息不会影响计算结果。因此，相似度度量的设计和实现需要根据频率进行扩展。如何设计一个既能保护隐私又能保证查询准确性的方案，是多重集相似度查询中的一个关键问题。

针对上述挑战，本文提出了一种在单云服务器环境中，隐私保护的多维多重集相似度阈值查询方案。我们的贡献如下：

(1)设计了多重集Jaccard相似度阈值查询协议，在该协议中，通过对原始集的预处理，使查询过程中的计算代价大大降低，且减少了查询令牌阶段的计算成本，使查询用户只需要轻量级的计算操作即可得到查询结果。

(2)通过0-1编码构造向量，结合Paillier密码系统设计了一个多维多重集相似度阈值查询协议，该协议可以对不同类型的数据进行高效、准确的并行查询，并用模拟范例证明协议是安全的。

(3)基于这两个协议，提出了一个高效且隐私保护的单云服务器下的精确多维多重集相似度阈值查

询方案。该方案可以在保护数据集、查询集、相似度阈值以及访问模式的隐私性的同时实现高效的多维多重集相似度阈值查询。据我们所知，该方案是第一个适用于物联网环境的实用且具有强隐私性的多维多重集相似度阈值查询方案。

(4)本文提出的保护隐私的多维多重集相似度阈值查询方案易于扩展，在保护数据安全的同时，保持了数据的灵活性和原始特征，方便进行各种计算，使其能够适应数据动态更新的应用场景，同时可以处理基于余弦相似度度量标准的多维多重集相似度阈值查询问题。

本文内容组织如下：第2节介绍相关研究；第3节阐述问题与系统模型；第4节详细描述了两个核心协议与具体查询过程；第5节对本方案的安全性和正确性进行分析；第6节对本文的效率进行理论分析并给出实验结果；第7节进行总结与展望。

2　相关工作

2.1　隐私保护的单维集合相似度查询

该问题可以表述为：给定一个查询集合Y和一个相似度阈值τ，以及多个加密的数据集合$\{[M_i]\}$，在不泄露任何隐私信息的情况下，从$\{[M_i]\}$中找出与Y的相似度大于等于τ的所有集合，即查询结果为$\{M_i | J_{sim}(M_i, Y)\geqslant\tau\}$。其中，$J_{sim}(M_i, Y)$表示$M_i$与$Y$的Jaccard系数。

为了解决这个问题，Blundo等人[5]设计了一种利用最小哈希技术求样本集合的Jaccard相似度的方案；文献[6-8]提出了基于SSE和LSH技术的方案；文献[11]设计了一种基于支点的kd-Tree，并在过滤和精化框架中有效地实现了集合相似度阈值查询。

然而，文献[6-8]的查询结果可能存在误判，文献[11]在计算过程中泄露了原有数据集和中间计算结果。现有方案难以在准确性、数据隐私性和查询效率之间找到良好的平衡。

除此之外，上述方案仅适用于单维数据集查询。如果要实现多维数据集的相似度阈值查询，就必须在两个独立的数据集上执行两个独立的查询。具体来说，首先应用相似度查询方案获取相似度大于等于阈值的候选记录，然后再应用关键词查询方案，验证每个候选记录的关键字集是否是查询集的子集。

如文献[15-17]的方案是专门为保护隐私的相似性查询而提出的，文献[15]在单个云上使用非对称标量积保持加密(ASPE)技术，文献[16,17]使用修改的ASPE来保护数据隐私。而文献[18-21]中的方案研究了与关键字查询相关的问题，文献[18-20]实现了基于密文策略属性基加密(CP-ABE)的布尔关键字搜索，文献[21]将多关键字查询与访问控制相结合。这两类方案的组合使用是实现上述丰富查询最直接的方法。

然而，这种方式不可避免地会导致查询效率低下。此外，组合使用方案还会在查询失败时泄露哪些查询类型未被满足，从而引发隐私泄露问题。因此，需要设计一种高效的多维集合相似度阈值查询方案，而不仅仅是将两种查询方案组合使用。

2.2　隐私保护的多维集合相似度阈值查询

该问题可以简单描述为：给定一个查询集合$\overline{Y}$，该集合由一个w维数据向量和一个关键字集合组成，即

$$\overline{Y}=\{(Y, O_y)|Y=(y_1, \cdots, y_w), O_y=\{o_1, \cdots, o_b\}\}。$$

同时，给定一个相似度阈值τ。

已知n个加密的数据集合$\{[M_i]\}$，每个集合$\{[M_i]\}$由u维数据向量和一个关键字集合组成，即

$$\{[M_i]\}=\{(X_i, O_i)|X_i=(m_{i1}, \cdots, m_{iu}), o_i=\{o_{i1}, \cdots, o_{ia_i}\}\}。$$

在不泄露任何隐私信息的情况下，从M_i中找出满足以下条件的多重集

$$\{M_i|J_{sim}(M_i, Y)\geqslant\tau\wedge O_y\subseteq O_i\}。$$

文献[22]基于电子医疗场景，使查询用户能够同时根据患者的多维生理特征和症状关键词(两种数据类型)查询相似患者的历史纪录。方案设计了一个二进制决策(BD-PB)树，同时为两种数据类型构建索引，结合Hilbert排除条件和多项式函数性质，设计了一种可以实现过滤验证的基于BD-PB树的查询算法，并利用矩阵加密技术与部分同态加密方案(SHE)，实现了在云服务器上的安全查询。

然而该方案在构建BD-PB树及查询过程中需要选择两个n维向量作为枢轴来协助计算，这两个n维向量的选择会影响查询效率及查询结果的准确性，导致查询结果出现误判或漏判。

以上分析表明，隐私保护的多维集合相似度阈

值查询问题尚未完全解决。一方面，现有的大多数方案将针对不同数据类型的查询方案组合使用，导致中间计算结果泄露；另一方面，现有方案的查询结果存在误差。因此，如何在多维相似度阈值查询方案中平衡安全性、准确性与查询效率，仍是当前研究面临的挑战。

3 预备知识和系统模型

本节首先介绍构造协议用到的一些预备知识；其次，描述了问题定义、系统模型和安全性需求。

3.1 预备知识

(1)Jaccard系数：给定两个集合A与B，Jaccard系数定义为A与B交集的大小与A与B并集的大小的比值，定义如下：

$$J_{sim}(A,B)=\frac{|A\cap B|}{|A\cup B|},$$

其中，$J_{sim}(A,B)\in[0,1]$，Jaccard系数越大，样本相似度越高。

(2)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：给定安全参数k，生成两个k比特的大素数p,q，若k为1024，则相当于对称加密算法中112比特密钥的安全性。令$N=p\times q$，$\eta=\mathrm{lcm}(p-1,q-1)$。定义函数$L(x)=\frac{x-1}{N}$，随机选择一个生成元$g\in Z_N^*$，使得$\gcd(\mathrm{L}(g^\eta \bmod N^2),N)=1$，则系统的公钥为$pk=(g,N)$，私钥为$sk=\eta$。

•加密：要加密明文$m\in Z_N$，选取随机数$r\in Z_N^*$，密文$c=g^m r^N \bmod N^2$。

•解密：对密文$c\in Z_{N^2}^*$，计算

$$m=\frac{L\left(c^\eta \bmod N^2\right)}{L\left(g^\eta \bmod N^2\right)} \bmod N。$$

•加法同态性：

$$[[m_1]]\times[[m_2]]=g^{m_1}r_1^N g^{m_2}r_2^N \bmod N^2=$$
$$g^{m_1+m_2}(r_1r_2)^N \bmod N^2=[[m_1+m_2]] \bmod N^2。$$

若明文m_2已知时，还可以得到以下性质：

$$[[m_1]]^{m_2} \bmod N^2=[[m_1m_2]] \bmod N^2。$$

(3)密文重随机化：指参与者在不解密的情况下将消息m的密文转化为另一个密文，相当于对明文m的重新加密。本文通过重随机化使每个参与者都无法了解其余参与者选择了哪些数据，替换了哪些数据，进而实现保密替换。

对密文消息M重随机化的具体步骤为：设明文消息M加密后的密文为

$$[[M]]=g^m r^N \bmod N^2。$$

参与者选择随机数r_i，并计算

$$g^m r^N r_i^N \bmod N^2=g^m(r\cdot r_i)^N \bmod N^2=$$
$$g^m r'^N \bmod N^2=[[M]]。$$

其中，r_i独立于原始随机数r，r_i^N是重随机化因子，确保新密文的随机性更新，r'^N即$(r\cdot r_i)^N$。

(4)安全性定义：在本文中，我们考虑所有参与者都是半诚实的。具体来说，协议的参与者在查询过程中遵循协议的要求，但他们可能保留从查询过程中获得的信息。在协议执行后，他们可能会试图利用这些信息来获得其他方的私人信息。半诚实模型，也称为诚实但好奇模型，是安全多方计算的重要模型。

假设有两个参与方P_1和P_2利用协议π保密计算多项式时间函数$f(x,y)$。设P_1和P_2分别输入私有数据x和y。在协议执行过程中，P_1获得的信息序列记为

$$view_1^\pi(x,y)=(x,r_1,M_1^1,\cdots,M_1^t),$$

其中，r_1是P_1选择的随机数，$M_1^j(j=1,2,\cdots,t)$表示P_1收到的第j个消息。P_2得到的信息序列也可类似定义。

定义1. （半诚实模型下的安全性）参与者均为半诚实时，若存在模拟器S_1和S_2，使得

$$\{S_1(x,f_1(x,y))\}_{x,y}\overset{c}{\equiv}\{view_1^\pi(x,y)\}_{x,y} \quad (1)$$

$$\{S_2(x,f_2(x,y))\}_{x,y}\overset{c}{\equiv}\{view_2^\pi(x,y)\}_{x,y} \quad (2)$$

则称协议π保密地计算f，其中$\overset{c}{\equiv}$表示计算不可区分。

3.2 问题定义

本文以集合相似度作为相似性标准，以Jaccard系数作为相似性度量。对于不同类型的数据，Jaccard相似度可以以不同的方式定义。

(1)相似性标准定义：

集合相似度阈值查询：给定一个相似度阈值τ，返回与查询集合相似度大于等于τ的集合。

(2)多重集及多重集相似度阈值查询：

多重集：多重集是数学中的一个概念，是集合概念的推广。在一个集合中，相同的元素只能出现

一次，而在多重集之中，同一个元素可以出现多次。一个元素出现的次数称为该元素的重数。每个元素都有一个重数，而所有元素的重数的最大值定义为多重集的重数。

多重集的全集：多重集的全集Q定义为

$$Q=\{(q_1,r_1),(q_2,r_2),\cdots,(q_k,r_k)\}$$
$$(q_1<q_2<\cdots<q_k),$$

q_i的出现次数为r_i，$|Q|=\sum_{i=1}^{k}r_i=\lambda$。

多重集的交集与并集：

设多重集

$$A=\{(q_1,s_1),\cdots,(q_k,s_k)\},$$
$$B=\{(q_1,t_1),\cdots,(q_k,t_k)\},$$

其中，$0\leqslant s_i,t_i\leqslant r_i$。

定义2.

$$A\cap B=\{(q_1,u_1),\cdots,(q_k,u_k)\},$$

其中，$u_i=\min(s_i,t_i)$。

$$A\cup B=\{(q_1,u_1),\cdots,(q_k,u_k)\},$$

其中，$u_i=\max(s_i,t_i)$。

一维多重集相似度阈值查询：

假设$\mathcal{M}=\{M_1,M_2,\cdots,M_n\}$是一组多重集，并且每个多重集都是某个多重集的子集，即给定一个查询多重集Y和一个相似度阈值τ，多重集相似度阈值查询是从$\mathcal{M}$中找出与查询多重集Y的相似度大于等于相似度阈值τ的多重集，即查询结果为

$$\{M_i|J_{sim}(M_i,Y)\geqslant\tau\}。$$

(3)多维多重集相似度阈值查询：

假设多重集$M_i(i=1,2,\cdots,n)$由一个u维数据向量和一个关键字集合组成，即

$$M_i=\{(X_i,O_i)|X_i=((x_{i1},s_{i1})\cdots,$$
$$(x_{iu},s_{iu})),o_i=\{o_{i1},\cdots,o_{ia_i}\}\}。$$

给定一个查询多重集，该多重集由一个w维数据向量和一个关键字集合组成，即

$$\overline{Y}=\{(Y,O_y)|Y=((y_1,t_1)\cdots,$$
$$(y_w,t_w)),O_y=\{o_1,\cdots,o_b\}\},$$

并给定一个相似度阈值τ。其中，s_{ij}表示在M_i中，x_{ij}的出现次数；t_j表示在Y中，y_j的出现次数，$X_i,Y\subseteq Q$且$O_i,O_y\subseteq O=\{O_1,O_2,\cdots,O_h\}$，$|O|=h$。

多维多重集相似度查询是从$\mathcal{M}$中找出满足

$$\{M_i|J_{sim}(M_i,Y)\geqslant\tau\wedge O_y\subseteq O_i\}$$

条件的多重集。

3.3　系统模型

在我们的系统模型中，考虑了一个基于云的多重集相似度阈值查询模型，系统模型如图1所示。

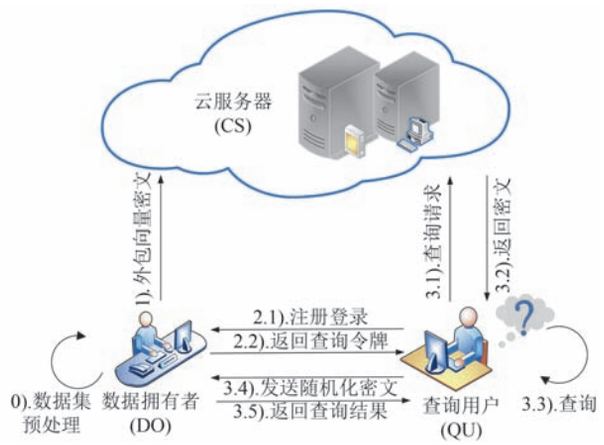

图1　系统模型

该模型由三部分组成，即数据所有者、云服务器和查询用户，每一方的数据和操作如下：

(1)数据拥有者(DO)：假设DO有多个多维多重集M_i，由于DO的存储空间有限，他更倾向于求助强大的云服务器来存储和管理他的数据。然而，由于云服务器是半诚实的，并且DO的数据中可能包含一些敏感信息，为了保护数据的隐私性，他倾向于在将数据外包给云服务器之前对其进行加密。

(2)云服务器(CS)：云服务器具有丰富的存储空间和强大的计算能力。一方面，它向DO提供存储服务，即为DO存储外包数据，节省了DO的存储空间。另一方面，它使得DO无需实时在线，也不需要维护数据，降低了DO数据维护的开销。同时，它接收查询用户QU的注册信息用于登录认证，向QU显示DO上传的密文。

(3)查询用户(QU)：查询用户QU已在CS注册。QU有待查询的一维(多维)多重集$Y(\overline{Y})$和一个相似度阈值τ，希望得到相似度大于等于规定阈值的多维多重集M_i。

3.4　安全性需求

我们假设此模型中的所有各方都是半诚实的。也就是说，DO，QU和CS都会忠实地遵循协议，且不会相互攻击，但可能对数据集和查询的信息感到好奇，试图从计算过程中获得的数据推断加密数据集的明文、查询范围和查询结果，以及其他间接隐私信息。

此外，我们假设DO，QU和CS不合谋。查询还必须满足以下要求：

(1)数据隐私性:CS除了数据集的大小,对确切的数据一无所知;QU除了查询结果,无法得到数据多重集M_i的具体元素及其势。当数据全集Q为公共信息或不敏感数据时,通常无需进行保护,因为这些信息不涉及数据隐私或敏感内容。例如,学生成绩是0-100分,人的年龄全集可以设定为0-120岁,人的正常体温通常在36.5℃到37.5℃之间,以及人体的基本生理学指标(如血压范围、心率等),这些都是广为人知的常识性数据,通常不会泄露任何数据隐私,因此可以认为它们不需要特别保护。

(2)查询隐私:CS不会得到QU的查询多重集$Y(\overline{Y})$和相似度阈值τ。

(3)结果隐私性:CS不会得到查询结果的明文。

(4)间接隐私:CS不能从中间计算结果中推断出任何隐私信息及访问模式。

我们没有对其他攻击方法,如信道攻击、恶意云服务器与查询用户合谋攻击等进行广泛的讨论,这些问题将作为我们未来工作的一部分来解决。

4 多维多重集相似度阈值查询方案

使用Paillier密码系统对加密向量进行相似度阈值查询的过程包括三个步骤:(1)查询用户QU注册;(2)数据拥有者DO对多重集数据进行预处理,并将加密的向量外包给CS;(3)QU登录CS,CS作出响应,QU获取相似度大于等于规定阈值的多重集。

4.1 概 述

在该方案中,DO执行安全的向量加密算法,并帮助QU解密密文。当QU希望查询相似度大于等于规定阈值的多重集时,QU首先向DO申请注册。当QU使用注册密码登录时,CS会向QU显示所有加密的向量。这时,QU执行查询响应算法,以获取相似度大于等于规定阈值的多重集的密文。最后,DO帮助QU解密密文,以获得查询结果。

4.2 查询用户注册

在这个阶段,QU向DO注册,提供必要的信息以获得权限,并随机选择一个字符串s。然后,使用哈希函数H,计算$H(s)$,将$H(s)$存储,并将s用作登录密码。

4.3 向量的预处理和加密存储

本节首先介绍一维多重集相似度阈值查询的构建过程和查询原理,然后将其扩展到多维多重集相似度阈值查询的原理。

4.3.1 一维多重集相似度阈值查询方案

设

$$M_i=\{(x_{i1}, s_{i1}), \cdots, (x_{iu}, s_{iu})\}\subseteq Q$$

为多重集数据集。查询请求为

$$Y=\{(y_1, t_1), \cdots, (y_w, t_w)\}\subseteq Q$$

以及阈值$\tau=\frac{\tau_1}{\tau_2}$。

我们的计算原理是

$$\frac{|A\cap B|}{|A\cup B|}\geqslant\frac{\tau_1}{\tau_2}\Leftrightarrow|A\cap B|\times\tau_2-|A\cup B|\times\tau_1\geqslant 0 \tag{3}$$

即在A,B的密文上判断

$$|A\cap B|\times\tau_2-|A\cup B|\times\tau_1\geqslant 0 \tag{4}$$

是否成立,具体做法如下:

(1)我们采用另一种记法来记多重集Q,即

$$Q=\underbrace{\{\overbrace{q_1, \cdots, q_1}^{r_1}, \cdots, \overbrace{q_k, \cdots, q_k}^{r_k}\}}_{\lambda} \tag{5}$$

DO先将Q中的元素重新编号,使得每个元素都有一个唯一的标识符。重新编号后的多重集表示为

$$Q'=\{q_1, q_2, \cdots, q_\lambda\} \tag{6}$$

其中,每个原始元素被映射到新的标识符。

DO根据Q'构造向量$V_i=(v_{i1}, \cdots, v_{i\lambda})$,若$q_j\in M_i$,则$v_{ij}=1$;若$q_j\notin M_i(1\leqslant j\leqslant\lambda)$,则$v_{ij}=0$,即

$$v_{ij}=\begin{cases}1, & q_j\in M_i\\ 0, & q_j\notin M_i\end{cases}(1\leqslant j\leqslant\lambda) \tag{7}$$

(2)当QU想要获取与Y的相似度大于等于某个阈值的多重集时,只需根据查询请求对V_i进行操作:当$q_j\in Y(1\leqslant j\leqslant\lambda)$时,选择$v_{ij}$将其全部相加,记为$C_{i1}$,即$|A\cap B|$:

$$C_{i1}=v_{i1}+\cdots+v_{iw}=\dot{v}_{ij} \tag{8}$$

若$q_j\in Y$,将V_i中的v_{ij}使用1替换;若$q_j\notin Y(1\leqslant j\leqslant\lambda)$,则保持$V_i$中的$v_{ij}$数据不变,记为$\dot{V}_i$,即

$$\dot{v}_{ij}=\begin{cases}1, & q_j\in Y\\ v_{ij}, & q_j\notin Y\end{cases}(1\leqslant j\leqslant\lambda) \tag{9}$$

将$\dot{v}_{ij}$全部相加,记为C_{i2},即$|A\cup B|$:

$$C_{i2}=\dot{v}_{i1}+\cdots+\dot{v}_{i\lambda}=\sum_{j=1}^{\lambda}\dot{v}_{ij} \tag{10}$$

(3)QU根据τ_1,τ_2,C_{i1}和C_{i2}计算得出C_i,即

$$C_i=C_{i1}\times\tau_2-C_{i2}\times\tau_1 \tag{11}$$

若$C_i\geqslant 0$,M_i即为满足条件的多重集,详细计算过程见示例1。

示例1.　DO有多重集

$$M_1=\{q_1,q_2,q_3,q_3,q_5\},$$
$$M_2=\{q_1,q_1,q_4,q_4,q_5\},$$
$$M_3=\{q_1,q_2,q_4,q_5,q_5\}。$$

QU想要查找与$Y=\{q_1,q_3,q_3,q_5\}$的相似度大于等于阈值$\tau=\frac{\tau_1}{\tau_2}=\frac{2}{3}$的多重集。已知全集

$$Q=\{q_1,q_1,q_2,q_3,q_3,q_4,q_4,q_5,q_5\},$$

即$|Q|=\sum_{i=1}^{k}r_i=9$。

(1)DO使用另一种记法来记多重集Q，

$$Q=\underbrace{\left\{\overbrace{q_1,\cdots,q_1}^{r_1},\cdots,\overbrace{q_k,\cdots,q_k}^{r_k}\right\}}_{\lambda}。$$

然后将Q中的元素重新编号，使得每个元素都有一个唯一的标识符。重新编号后的多重集表示为

$$Q'=\{q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8,q_9\},$$

其中，每个原始元素被映射到新的标识符。

根据4.3.1节将$\boldsymbol{M}_i$构造为向量，若$q_j\in\boldsymbol{M}_i$，则$v_{ij}=1$；若$q_j\notin\boldsymbol{M}_i(1\leqslant j\leqslant\lambda)$，则$v_{ij}=0$。

以构造向量$\boldsymbol{V}_1$为例：

$\{q_1,q_3,q_4,q_5,q_8\}\subseteq\boldsymbol{M}_1$，因此，$v_{11}=v_{13}=v_{14}=v_{15}=v_{18}=1$；$\{q_2,q_6,q_7,q_5,q_9\}\notin\boldsymbol{M}_1$，因此，$v_{12}=v_{16}=v_{17}=v_{19}=0$。

综上所述，得到向量$\boldsymbol{V}_1$、$\boldsymbol{V}_2$、$\boldsymbol{V}_3$：

$$\boldsymbol{V}_1=(1,0,1,1,1,0,0,1,0),$$
$$\boldsymbol{V}_2=(1,1,0,0,0,1,1,1,0),$$
$$\boldsymbol{V}_3=(1,0,1,0,0,0,1,1,1)$$

随后发送给QU。

(2)QU选择计算

$$C_{11}=v_{11}+v_{14}+v_{15}+v_{18}=4,$$
$$C_{12}=v_{21}+\cdots+v_{28}=2,$$
$$C_{13}=v_{31}+\cdots+v_{38}=2。$$

对$\boldsymbol{V}_1,\boldsymbol{V}_2,\boldsymbol{V}_3$进行替换，得到：

$$\dot{\boldsymbol{V}}_1=(1,0,1,1,1,0,0,1,0),$$
$$\dot{\boldsymbol{V}}_2=(1,1,0,1,1,1,1,1,0),$$
$$\dot{\boldsymbol{V}}_3=(1,0,1,1,1,1,0,1,1)。$$

计算$C_{12}=\sum_{j=1}^{9}\dot{\boldsymbol{v}}_{1j}=5$，$C_{22}=\sum_{j=1}^{9}\dot{\boldsymbol{v}}_{2j}=7$，$C_{32}=\sum_{j=1}^{9}\dot{\boldsymbol{v}}_{3j}=7$。

结合$\tau=\frac{2}{3}$，$\tau_1=2$，$\tau_2=3$得出

$$C_1=4\times3-5\times2=2\geqslant0,$$
$$C_2=2\times3-7\times2=-8<0,$$
$$C_3=2\times3-7\times2=-8<0,$$

则满足查询条件的多重集为M_1。

DO利用明文数据预处理多重集信息，并将预处理向量加密为密文，外包给服务器CS。算法1描述了一维多重集相似度阈值查询方案中向量生成和加密向量元素的过程。

4.3.2　多维多重集相似度阈值查询方案

设

$$\boldsymbol{M}_i=\left\{\left((x_{i1},s_{i1}),\cdots,(x_{iu},s_{iu})\right),o_{i1},\cdots,o_{ia_i}\right\}$$

为多维多重集($i=1,2,\cdots,n$)。给定一个查询多重集

算法1　安全向量加密算法

输入：Paillier密码系统的公钥pk，DO输入多重集$\boldsymbol{M}_i(i=1,2,\cdots,n)$.

输出：密文向量$[[\boldsymbol{V}_1]],[[\boldsymbol{V}_2]],\cdots,[[\boldsymbol{V}_n]]$.

1. for $i\in\{1,\cdots,n\}$ do
2.　　for $j\in\{1,\cdots,\lambda\}$ do
3.　　DO encrypts v_{ij} by using pk to obtain ciphertext element $[[v_{ij}]]$.
4.　　end for
5. end for
6. Output ciphertext vector $[[\boldsymbol{V}_1]],[[\boldsymbol{V}_2]],\cdots,[[\boldsymbol{V}_n]]$.

$$\overline{Y}=\left\{\left((y_1,t_1),\cdots,(y_w,t_w)\right),o_1,\cdots,o_b\right\},$$

以及阈值$\tau=\frac{\tau_1}{\tau_2}$。计算原理具体如下：

(1)采用另一种记法来记数据多重集Q，即

$$Q=\underbrace{\left\{\overbrace{q_1,\cdots,q_1}^{r_1},\cdots,\overbrace{q_k,\cdots,q_k}^{r_k}\right\}}_{\lambda}\tag{12}$$

DO根据Q和关键字集合O构造向量$\boldsymbol{V}_i$，即

$$\boldsymbol{V}_i=\left(\overbrace{v_{i1},v_{i2},\cdots,v_{i\lambda}}^{\text{数据多重集}},\overbrace{v_{i(\lambda+1)},\cdots,v_{i(\lambda+h)}}^{\text{关键字集合}}\right)\tag{13}$$

当$1\leqslant j\leqslant\lambda$时，若$q_j\in\boldsymbol{M}_i$，则$v_{ij}=1$；若$q_j\notin\boldsymbol{M}_i$，则$v_{ij}=0$，即

$$v_{ij}=\begin{cases}1, & q_j\in\boldsymbol{M}_i\\ 0, & q_j\notin\boldsymbol{M}_i\end{cases}(1\leqslant j\leqslant\lambda)\tag{14}$$

当$\lambda<j\leqslant\lambda+h$时，若$o_j\in\boldsymbol{M}_i$，则$v_{ij}=0$；若$o_j\notin\boldsymbol{M}_i$，则$v_{ij}=1$，即

$$v_{ij}=\begin{cases}0, & o_j\in\boldsymbol{M}_i\\ 1, & o_j\notin\boldsymbol{M}_i\end{cases}\lambda<j\leqslant(\lambda+h)\tag{15}$$

(2)当QU想要查询满足条件

$$\left\{\boldsymbol{M}_i|J_{sim}(\boldsymbol{M}_i,Y)\geqslant\tau\wedge O_y\subseteq O_i\right\}$$

的多重集，只需根据$\overline{Y}$对V_i进行操作：

当$q_j\in Y(1\leqslant j\leqslant\lambda)$时，选择$v_{ij}$将其全部相加，记为$C_{i1}$，即$|A\cap B|$：

$$C_{i1}=v_{i1}+\cdots+v_{iw}=\sum_{j=1}^{w}v_{ij} \tag{16}$$

当$o_j\in O_y(\lambda<j\leqslant\lambda+h)$时，选择$v_{ij}$将其全部相加，记为$C_{i3}$，即

$$C_{i3}=v_{i1}+\cdots+v_{ib}=\sum_{j=1}^{b}v_{ij} \tag{17}$$

若$q_j\in Y$，将V_i中的v_{ij}使用1替换；若$q_j\notin Y(1\leqslant j\leqslant\lambda)$，则保持$V_i$中的$v_{ij}$数据不变，记为$\dot{V}_i$，即

$$\dot{v}_{ij}=\begin{cases}1, & q_j\in Y\\ v_{ij}, & q_j\notin Y\end{cases}(1\leqslant j\leqslant\lambda) \tag{18}$$

将$\dot{v}_{ij}$全部相加，记为C_{i2}，即$|A\cup B|$：

$$C_{i2}=\dot{v}_{i1}+\cdots+\dot{v}_{i\lambda}=\sum_{j=1}^{\lambda}\dot{v}_{ij} \tag{19}$$

(3) QU根据τ_1，τ_2，C_{i1}，C_{i2}和C_{i3}计算得出C_i，即

$$C_i=C_{i1}\times\tau_2-C_{i2}\times\tau_1-C_{i3}\times\beta_i \tag{20}$$

其中，β_i是QU选择的随机数，$\beta_i\geqslant0$。若$C_i\geqslant0$，$\boldsymbol{M}_i$即为满足条件的多重集，详细计算过程见示例2。

示例2 DO有多重集

$$\boldsymbol{M}_1=\{(q_1,q_2,q_3,q_3,q_5),o_1,o_3,o_5\},$$
$$\boldsymbol{M}_2=\{(q_1,q_1,q_4,q_4,q_5),o_1,o_2,o_4\},$$
$$\boldsymbol{M}_3=\{(q_1,q_2,q_4,q_5,q_5),o_3,o_4\}$$

QU想要根据$\overline{Y}=\{(q_1,q_3,q_3,q_5),o_3,o_5\}$查询满足条件

$$\{\boldsymbol{M}_i|J_{sim}(\boldsymbol{M}_i,Y)\geqslant\tau\wedge O_y\subseteq O_i\}$$

的多重集。其中，数据多重集的全集

$$Q=\{q_1,q_1,q_2,q_3,q_3,q_4,q_4,q_5,q_5\},$$

即$\lambda=9$；关键字集合全集

$$O=\{o_1,o_2,o_3,o_4,o_5\},$$

即$h=5$，并且$\tau=\dfrac{\tau_1}{\tau_2}=\dfrac{2}{3}$。

(1) DO先将Q中的元素重新编号，使得每个元素都有一个唯一的标识符。重新编号后的多重集表示为

$$Q'=\{q_1,q_2,q_3,q_4,q_5,q_6,q_7,q_8,q_9\},$$

其中，每个原始元素被映射到新的标识符。随后将中O的元素重新编号，得到

$$O'=\{o_{10},o_{11},o_{12},o_{13},o_{14}\}$$

根据4.3.2节将$\boldsymbol{M}_i$构造为向量，若$q_j\in\boldsymbol{M}_i$，则$v_{ij}=1$；若$q_j\notin\boldsymbol{M}_i(1\leqslant j\leqslant\lambda)$，则$v_{ij}=0$。以构造向量$V_1$为例，$q_1,q_3,q_4,q_5,q_8\subseteq\boldsymbol{M}_1$，因此，$v_{11}=v_{13}=v_{14}=v_{15}=v_{18}=1$；$q_2,q_6,q_7,q_5,q_9\notin\boldsymbol{M}_1$，因此，$v_{12}=v_{16}=v_{17}=v_{19}=0$。

若$o_j\in\boldsymbol{M}_i$，则$v_{ij}=0$；若$o_j\notin\boldsymbol{M}_i(\lambda<j\leqslant\lambda+h)$则$v_{ij}=1$。以构造向量$\boldsymbol{V}_1$为例，$\{o_1,o_3,o_5\}\in\boldsymbol{M}_1$，因此，$v_{10}=v_{12}=v_{14}=0$；$\{o_2,o_4\}\notin\boldsymbol{M}_1$，因此，$v_{11}=v_{13}=1$。

综上所述，得到向量$\boldsymbol{V}_1$，$\boldsymbol{V}_2$，$\boldsymbol{V}_3$：

$$\boldsymbol{V}_1=\left(\overbrace{1,0,1,1,1,0,0,1,0}^{\text{数据多重集}},\overbrace{0,1,0,1,0}^{\text{关键字集}}\right),$$
$$\boldsymbol{V}_2=(1,1,0,0,0,1,1,1,0,0,0,1,0,1),$$
$$\boldsymbol{V}_3=(1,0,1,0,0,0,1,1,1,1,1,0,0,1)$$

(2) QU首先计算$|A\cap B|$：

$$C_{11}=v_{11}+v_{14}+v_{15}+v_{18}=4,$$
$$C_{21}=v_{21}+\cdots+v_{28}=2,$$
$$C_{31}=v_{31}+\cdots+v_{38}=2$$

然后计算关键字子集：

$$C_{13}=v_{112}+v_{114}=0+0=0,$$
$$C_{23}=v_{212}+v_{214}=1+1=2,$$
$$C_{33}=v_{312}+v_{314}=0+1=1$$

计算$|A\cup B|$：

$$\dot{\boldsymbol{V}}_1=(1,0,1,1,1,0,0,1,0),$$
$$\dot{\boldsymbol{V}}_2=(1,1,0,1,1,1,1,1,0),$$
$$\dot{\boldsymbol{V}}_3=(1,0,1,1,1,1,0,1,1)$$

得出$C_{12}=\sum_{j=1}^{9}\dot{v}_{1j}=5$，$C_{22}=\sum_{j=1}^{9}\dot{v}_{2j}=7$，$C_{32}\sum_{j=1}^{9}\dot{v}_{3j}=7$。结合$\dfrac{\tau_1}{\tau_2}=\dfrac{2}{3}$及选择的随机数$\beta_1$，$\beta_2$，$\beta_3$得

$$C_1=4\times3-5\times2-0\times\beta_1=2-0\beta_1\geqslant0,$$
$$C_2=2\times3-7\times2-2\times\beta_2=-8-2\beta_2<0,$$
$$C_3=2\times3-7\times2-1\times\beta_3=-8-\beta_3<0$$

则满足查询条件的多重集为$\boldsymbol{M}_1$。

多维多重集相似度阈值查询方案中加密向量元素的过程与算法1相同，故省略。

4.4 登录与查询响应

在一维多重集相似度阈值查询方案中：

(1) CS存储密文向量$[[V_1]]$，$[[V_2]]$，…，$[[V_n]]$。QU使用口令登录CS，CS向QU显示所有加密向量$[[V_1]]$，$[[V_2]]$，…，$[[V_n]]$。

(2) DO与QU都知道全集Q，因此，QU想要得

到与查询请求 Y 大于等于阈值 τ 的多重集，只需要在密文向量上进行选择计算 C_i。

(3)为了保护 DO 与 QU 的数据隐私性，QU 选择随机数 $\alpha_i(\alpha_i>0)$，计算 $[[C_i+\alpha_i]]$，并发送给 DO。

(4) DO 收到 QU 发来的密文 $[[C_i']]=[[C_i+\alpha_i]]$，使用私钥 sk 对其进行解密，将全部解密结果返回给 QU。

(5) QU 可以计算 $C_i'+\alpha_i-\alpha_i=C_i$，但 DO 不知道 C_i 与 α_i 的具体数值，具体过程见算法2。

在多维多重集相似度阈值查询方案中：

(1) CS 存储密文向量 $[[\boldsymbol{V}_1]]$，$[[\boldsymbol{V}_2]]$，…，$[[\boldsymbol{V}_n]]$。QU 使用口令登录 CS，CS 向 QU 显示所有加密向量 $[[\boldsymbol{V}_1]]$，$[[\boldsymbol{V}_2]]$，…，$[[\boldsymbol{V}_n]]$。

算法2.　一维多重集查询响应算法

输入：QU 输入查询多重集 Y，CS 输入密文向量 $[[V_1]]$，$[[V_2]]$，…，$[[V_n]]$.

输出：QU 获得与查询多重集 Y 的相似度大于等于阈值 τ 的多重集 $\boldsymbol{M}_i$.

1. QU logs in to the server using a password
2. CS displays all ciphertext vectors $[[V_1]]$, $[[V_2]]$, …, $[[V_n]]$ to QU
3. for $i\in\{1,\cdots,n\}$ do
4. QU selects from the ciphertext based on the query multiset Y and computes $[[C_i]]$
5. for $j\in\{1,\cdots,\lambda\}$ do
6. calculate $|A\cap B|$:
7. $[[C_{i1}]]=\left[\left[\sum_{j=1}^{w}v_{ij}\right]\right]$
8. if $q_j\in Y$
9. Replace $[[v_{ij}]]$ with $[[1]]$ to get $[[\dot{v}_{ij}]]$
10. end if
11. calculate $|A\cup B|$:
12. $[[C_{i2}]]=\left[\left[\sum_{j=1}^{\lambda}\dot{v}_{ij}\right]\right]$
13. end for
14. combine similarity thresholds τ_1, τ_2:
15. $[[C_i]]=[[C_{i1}]]^{\tau_2}\times[[-C_{i2}]]^{\tau_1}$
16. using Paillier homomorphic property:
17. $[[C_i]]=[[\tau_2C_{i1}-\tau_1C_{i2}]]$
18. QU selects a random number $\alpha_i(\alpha_i>0)$ and computes:
19. $[[C_i']]=[[C_i]]\times[[\alpha_i]]=[[C_i+\alpha_i]]$
20. QU sends $[[C_i']]=[[C_i+\alpha_i]]$ to DO
21. end for
22. for $i\in\{1,2,\cdots,n\}$ do
23. DO receives the ciphertext $[[C_i']]=[[C_i+\alpha_i]]$ sent by QU and decrypts it using the private sk, returning the decrypted result C_i'
24. QU receives C_i' computes:
25. if $C_i'>0$
26. computes $C_i'+\alpha_i-\alpha_i=C_i$
27. end if
28. end for

(2) DO 与 QU 已知数据多重集全集 Q 与关键字全集 O，QU 想要得到满足查询条件

$$\{\boldsymbol{M}_i|J_{sim}(\boldsymbol{M}_i,Y)\geqslant\tau\wedge O_y\subseteq O_i\}$$

的多重集，只需要在密文向量上进行选择计算 C_i，具体为

$$[[C_i]]=[[C_{i1}]]^{\tau_2}\times[[-C_{i2}]]^{\tau_1}\times[[-C_{i2}]]^{\beta_i}$$

为了保护 DO 与 QU 的数据隐私性，QU 选择随机数 $\alpha_i,\beta_i(\alpha_i>0,\beta_i\geqslant0)$，计算

$$[[C_i']]=[[C_i]]\times[[\alpha_i]]=[[C_i+\alpha_i]],$$

并发送给 DO。

(3) DO 收到 QU 发来的密文 $[[C_i']]$，使用私钥 sk 对其进行解密，将全部解密结果返回给 QU。

(4) QU 通过计算 $C_i=C_i'-\alpha_i$ 得到最终查询结果。

多维多重集的查询响应算法和一维多重集的查询响应算法相同，故省略。

4.5　动态更新

参考文献[23]，我们提出了对一维多重集相似度阈值查询和多维多重集相似度阈值查询的动态更新方案：

(1)简单情形下的动态更新：

①删除操作：若删除多个多重集 $\boldsymbol{M}_i$，仅需直接删除对应行的加密向量。

②插入操作：对于新增多重集 $\boldsymbol{M}_i$，仅需根据计算原理对 $\boldsymbol{M}_i$ 进行编码和加密，然后将加密后的密文上传至 CS，无需对现有密文进行任何修改。

(2)复杂情形下的动态更新：

①多维数据集全集 Q 与关键字全集 O 不发生变化，仅 M_i 中的元素需要插入或删除：

在这种情况下，DO 使用公钥加密1或0，对需要插入的数据向量元素在其相应位置使用 $[[1]]$ 替换，对需要删除的数据向量元素在其相应位置使用 $[[0]]$ 替换，对需要插入或删除的关键字元素，使用

相反的替换方式。为了防止CS根据密文推测更新内容，所有密文必须在更新后进行重随机化。这样可以确保即使密文发生了变化，CS也无法从加密数据中推测出具体的更新内容。之后将更新后的密文上传到CS，用于后续查询。

②多维数据集全集Q与关键字全集O发生变化：

若删除某个数据或关键字，仅需直接删除该列对应的全部密文。

若新增数据或关键字，则需生成新的密文列，即按照计算原理对M_i相应位置进行编码，再将对应列的每个元素加密，上传给CS。

4.6 基于余弦相似度的多重集相似度阈值查询

通过对一维多重集相似度阈值查询方案扩展，即DO计算$\boldsymbol{M}_i$编码为$\boldsymbol{V}_i$后的$[[\boldsymbol{V}_i]]$，将原先λ维的向量$\boldsymbol{V}_i$扩展为$\lambda+1$维向量$\boldsymbol{V}_i'$，即可实现多样化相似度查询。具体计算过程如下：

我们的计算原理是，

$$\frac{A\cdot B}{|A||B|}\geqslant\frac{\tau_1}{\tau_2}\Leftrightarrow(A\cdot B)\times\tau_2-(|A||B|\times\tau_2)\geqslant 0$$

(1)DO将原有向量$\boldsymbol{V}_i=(v_{i1},\cdots,v_{i\lambda})$扩展为$\lambda+1$维向量$\boldsymbol{V}_i'=(v_{i1},v_{i2},\cdots,v_{i\lambda},v_{i(\lambda+1)})$，其中，

$$v_{i(\lambda+1)}=|\boldsymbol{V}_i|=\sqrt{v_{i1}^2+v_{i2}^2+\cdots+v_{i\lambda}^2}$$

DO使用公钥pk加密向量$\boldsymbol{V}_i'$的每一维，之后上传给CS。

(2)QU使用口令登录，CS向QU显示所有密文向量。QU根据全集Q将自己的查询多重集Y进行编码，得到向量Y'。若$q_j\in Y$，则$y_j'=1$；若$q_j\notin Y$($1\leqslant j\leqslant\lambda$)，则$y_j'=0$。之后，计算

$$|\boldsymbol{Y}'|=\sqrt{y'^2_1+y'^2_2+\cdots+y'^2_\lambda}$$

1)QU首先计算$\boldsymbol{V}_i\cdot\boldsymbol{Y}$：

$$[[C_{i1}]]=\left[\left[\sum\nolimits_{j=1}^{\lambda}(v_{ij}\times y'_j)\right]\right]=$$

$$[[v_{i1}]]^{y_1'}\times[[v_{i2}]]^{y_2'}\times\cdots[[v_{i\lambda}]]^{y_\lambda'}$$

2)之后，QU选择$[[v_{i\lambda+1}]]$，计算$|\boldsymbol{V}_i|\cdot|\boldsymbol{Y}'|$：

$$[[C_{i2}]]=[[v_{i(\lambda+1)}]]^{|Y'|}=$$

$$\left[\left[\sqrt{v_{i1}^2+\cdots+v_{i\lambda}^2}\times\sqrt{y'^2_1+\cdots+y'^2_\lambda}\right]\right]$$

3)结合相似度阈值τ_1,τ_2计算：

$$[[C_i]]=[[C_{i1}]]^{\tau_2}\times[[-C_{i2}]]^{\tau_1}$$

(3)QU选择随机数α_i，计算：

$$[[C_i']]=[[C_i]]\times[[\alpha_i]]=[[C_i+\alpha_i]],$$

并发送给DO。

(4)DO收到QU发来的密文$[[C_i']]$，使用私钥sk对其进行解密，将全部解密结果C_i'返回给QU。

(5)QU收到C_i'后计算$C_i=C_i'-\alpha_i$，得到最终查询结果。

多维多重集相似度阈值查询方案实现多样化相似度查询的方法类似，只需在计算$[[C_{i2}]]$后，按原有方案计算$[[C_{i3}]]$和$[[C_i']]$即可。

5 方案分析

5.1 正确性分析

在一维相似度阈值查询方案中，CS只存储密文，由DO加解密一维多重集，QU在密文上进行选择与同态计算。QU根据查询多重集Y在密文向量$[[\boldsymbol{V}_1]],[[\boldsymbol{V}_2]],\cdots,[[\boldsymbol{V}_n]]$上进行选择，计算$\boldsymbol{V}_i$与$Y$交集的势，计算原理为：

$$|\boldsymbol{V}_i\cap Y|=|\{q_j\in Q|q_j\in V_i\wedge q_j\in Y\}|=(v_{i1}\wedge y_1)+(v_{12}\wedge y_2)+\cdots+(v_{1\lambda}\wedge y_\lambda)=\sum\nolimits_{j=1}^{\lambda}(v_{1j}\wedge y_j)$$

具体计算过程如下：

若q_j为DO和QU共有元素，则选择出的$[[v_{ij}]]$解密后得到的明文为1，若q_j仅为DO的元素，QU不会选择；若q_j仅为QU的元素，则选择出的$[[v_{ij}]]$解密后得到的明文为0，计算$\left[\left[\sum_{j=1}^{\lambda}v_{ij}\right]\right]$即得出$DO$和$QU$一维多重集交集的势的密文。

之后，QU计算$\boldsymbol{V}_i$与Y并集的势，计算原理为：

$$|V_i\cup Y|=|\{q_j\in Q|q_j\in\boldsymbol{V}_i\vee q_j\in Y\}|=(v_{i1}\vee y_1)+(v_{i2}\vee y_2)+\cdots+(v_{2\lambda}\vee y_\lambda)=\sum\nolimits_{j=1}^{w}(v_{1j}\vee y_j)$$

具体计算过程如下：

根据查询请求Y对密文向量$[[\boldsymbol{V}_1]],[[\boldsymbol{V}_2]],\cdots,[[\boldsymbol{V}_n]]$进行保密替换，当$q_j\in Y$时，使用$[[1]]$替换$[[\boldsymbol{V}_i]]$中的$[[v_{ij}]]$，若$q_j\notin Y$($1\leqslant j\leqslant\lambda$)，则保持$[[\boldsymbol{V}_i]]$中$[[v_{ij}]]$数据不变，记为$[[\dot{v}_{ij}]]$，计算$\left[\left[\sum_{j=1}^{\lambda}\dot{v}_{ij}\right]\right]$，即得出$DO$和$QU$一维多重集并集的势的密文。根据Jaccard系数的定义与计算原理

$$\frac{|A\cap B|}{|A\cup B|}\geqslant\frac{\tau_1}{\tau_2}\Leftrightarrow|A\cap B|\times\tau_2-|A\cup B|\times\tau_1\geqslant 0$$

即计算

$$[[C_i]]=[[C_{i1}]]^{\tau_2}\times[[-C_{i2}]]^{\tau_1},$$
$$[[C_i']]=[[C_i]]\times[[\alpha_i]]=[[C_i+\alpha_i]]。$$

DO将解密结果发送给QU，QU计算$C_i=C_i'-\alpha_i$，得出最终满足条件的多重集。QU添加随机数α_i对DO隐藏了查询多重集的信息但对正确性没有影响。

在多维相似度阈值查询方案中，其他步骤都与一维相似度阈值查询相同，仅增添在密文上判断$O_y\subseteq O_i$的计算过程，计算原理为

$$O_y\subseteq O_i\Leftrightarrow\forall o_j\left(o_j\in O_y\Rightarrow o_j\in O_i\right),$$
$$O_y\subseteq O_i\Leftrightarrow\sum_{j=1}^{b}v_{ij}=0。$$

具体计算过程如下：

当$o_j\in O_y\left(\lambda<j\leqslant\lambda+h\right)$时，选择$v_{ij}$将其全部相加，若$o_j$为$O_y$和$O_i$共有元素，则选择出的$\left[\left[v_{ij}\right]\right]$解密后得到的明文为0，若$o_j$仅为$O_i$的元素，$QU$不会选择；若$o_j$仅为$O_y$的元素，则选择出的$\left[\left[v_{ij}\right]\right]$解密后得到的明文为1，计算$\left[\left[\sum_{j=1}^{b}v_{ij}\right]\right]$即可判断$O_y\subseteq O_i$。$QU$添加随机数$\alpha_i$，$\beta_i$隐藏了查询多重集的信息但对正确性没有影响。

5.2　安全性分析

在一维多重集相似度阈值查询方案中，CS唯一可用的信息是DO的外包加密向量$[[V_1]]$，$[[V_2]]$，…，$[[V_n]]$，QU的登录信息，以及多重集全集的势。由于Paillier密码系统在语义上是安全的，CS无法获得关于多重集的任何信息，也没有接收来自QU的查询信息。因此，在证明中可以忽略CS，接下来主要构建模拟器S_1和S_2，利用满足定义1的模拟范例证明QU和DO的安全性。

首先，构造S_1证明QU的安全性。在实际执行协议中，DO获得的信息为

$$view_{DO}^{\pi}\left(\left([[V_1]],\cdots,[[V_n]]\right),Y\right)=$$
$$\left\{\left([[V_1]],\cdots,[[V_n]]\right),[[C_1']],\cdots,[[C_n']]\right\}。$$

对于输入$([[V_1]],\cdots,[[V_n]],\perp)$，模拟器$S_1$随机选择$\widetilde{C}_i'$，用$DO$的公钥$pk$对其加密，得到$\left[\left[\widetilde{C}_1'\right]\right],\cdots,\left[\left[\widetilde{C}_n'\right]\right]$，即

$$S_1\left(\left([[V_1]],\cdots,[[V_n]]\right),\perp\right)=$$
$$\left\{\left([[V_1]],\cdots,[[V_n]]\right),\left[\left[\widetilde{C}_1'\right]\right],\cdots,\left[\left[\widetilde{C}_n'\right]\right]\right\}。$$

由于Paillier加密算法是语义安全的，所以真实环境得到的$[[C_i']]$和模拟得到的$\left[\left[\widetilde{C}_i'\right]\right]$是计算不可区分的。虽然$DO$可以解密$[[C_i']]$，但$[[C_i']]$是经使用$QU$随机数$\alpha_i$混淆的，$DO$解密$[[C_i']]$只得到$C_i'=C_i+\alpha_i$。

由于α_i的随机性，DO无法推断出C_i的具体值。因此：

$$\left\{S_1\left([[V_1]],\cdots,[[V_n]],\perp\right)\right\}\overset{c}{\equiv}$$
$$\left\{view_{DO}^{\pi}\left(\left([[V_1]],\cdots,[[V_n]]\right),Y\right)\right\}。$$

其次，构造模拟器S_2证明DO的安全性。QU从CS获得的数据与从DO获得相同的数据是等价的。

在实际执行协议中，QU获得的信息为

$$view_{QU}^{\pi}\left(\left([[V_1]],\cdots,[[V_n]]\right),Y\right)=$$
$$\left\{\left(y_1,t_1\right),\cdots,\left(y_w,t_w\right),\tau_1,\tau_2,\right.$$
$$\left.\alpha_i,r,\left([[V_1]],\cdots,[[V_n]]\right),C_1',\cdots,C_n'\right\}。$$

(1)对于输入$(Y,C_1',\cdots,C_n')$，模拟器S_2随机选择多重集$\widetilde{M}_1,\cdots,\widetilde{M}_n$，使得在$\widetilde{M}_1,\cdots,\widetilde{M}_n$选择计算出的结果为$C_1',\cdots,C_n'$。

(2)S_2模拟算法2，加密$\widetilde{M}_1,\cdots,\widetilde{M}_n$中所有元素，得到密文向量$\left[\left[\widetilde{V}_1\right]\right],\cdots,\left[\left[\widetilde{V}_n\right]\right]$。

(3)S_2根据Y，在加密向量中选择对应元素，并计算$[[C_i']]=[[C_i+\alpha_i]]$。

(4)S_2使用私钥解密$[[C_i']]$，得到C_i'。

在模拟协议过程中，S_2选择的随机数记为r，令：

$$S_2(Y,C_1',\cdots,C_n')=\left\{\left(y_1,t_1\right),\cdots,\left(y_w,t_w\right),\right.$$
$$\left.\tau_1,\tau_2,\alpha_i,r,\left(\left[\left[\widetilde{V}_1\right]\right],\cdots,\left[\left[\widetilde{V}_n\right]\right]\right),C_1',\cdots,C_n'\right\}。$$

Paillier加密算法是语义安全的，所以在模拟过程中得到的$\left[\left[\widetilde{V}_1\right]\right],\cdots,\left[\left[\widetilde{V}_n\right]\right]$和实际方案执行中得到的信息$[[V_1]]$，…，$[[V_n]]$是不可区分的。

因此有：

$$\left\{S_2(Y,C_1',\cdots,C_n')\right\}\overset{c}{\equiv}$$
$$\left\{view_{QU}^{\pi}\left([[V_1]],\cdots,[[V_n]]\right),Y\right\}。$$

综上所述，所提出的方案在半诚实模型下是安全的。

多维多重集相似度阈值查询方案与一维多重集相似度阈值查询方案的安全性相同，故省略。

5.3 误差分析

多重集的全集Q定义为

$$Q=\{(q_1, r_1), (q_2, r_2), \cdots, (q_k, r_k)\},$$

$$q_1<q_2<\cdots<q_k。$$

q_i的出现次数为r_i，$|Q|=\sum_{i=1}^{k}r_i=\lambda$。

采用另一种记法来记数据多重集Q，即

$$Q'=\underbrace{\{\overbrace{q_1, \cdots, q_1}^{r_1}, \cdots, \overbrace{q_k, \cdots, q_k}^{r_k}\}}_{\lambda}。$$

DO根据Q'构造向量$V_i=(v_{i1}, \cdots, v_{i\lambda})$，若$q_j\in M_i$，则$v_{ij}=1$；若$q_j\notin M_i$（$1\leqslant j\leqslant\lambda$），则$v_{ij}=0$，即

$$v_{ij}=\begin{cases}1, & q_j\in M_i\\ 0, & q_j\notin M_i\end{cases}(1\leqslant j\leqslant\lambda)。$$

通过这样的编码，将重复元素考虑入内，确保编码后的向量能够准确反映每个元素在原始多重集中的出现次数，因此，交集和并集的计算不会存在误差，具体证明过程如下：

定义3： 对于多重集M_1和M_2，其Jaccard系数应基于元素的最小出现次数（交集）和最大出现次数（并集）：

$$J_D(M_1, M_2)=\frac{\sum_{q_j\in Q}\min\left(M_1(q_j), M_2(q_j)\right)}{\sum_{q_j\in Q}\max\left(M_1(q_j), M_2(q_j)\right)}。$$

其中，Q是全集

$$Q=\underbrace{\{\overbrace{q_1, \cdots, q_1}^{r_1}, \cdots, \overbrace{q_k, \cdots, q_k}^{r_k}\}}_{\lambda}。$$

$M_1(q_j)$表示元素q_j在多重集M_1中的出现次数，$M_2(q_j)$表示元素q_j在多重集M_2中的出现次数。

设元素q_j在多重集M_1中出现m次，在多重集M_2中出现n次，若按本文的0-1编码方式进行编码，则M_1被编码为$V_1=(v_{11}, v_{12}, \cdots, v_{1\lambda})$，$M_2$被编码为$V_2=(v_{21}, v_{22}, \cdots, v_{2\lambda})$。

$$|M_1\cap M_2|=|\{q_j\in Q|q_j\in M_1\wedge q_j\in M_2\}|=$$

$$(v_{11}\wedge v_{21})+\cdots+(v_{1\lambda}\wedge v_{2\lambda})=\sum_{j=1}^{\lambda}v_{1j}\wedge v_{2j}$$

$$|M_1\cup M_2|=|\{q_j\in Q|q_j\in M_1\vee q_j\in M_2\}|=$$

$$(v_{11}\vee v_{21})+\cdots+(v_{1\lambda}\vee v_{2\lambda})=\sum_{j=1}^{\lambda}v_{1j}\vee v_{2j}。$$

根据Jaccard系数的定义

$$J_{sim}=\frac{|A\cap B|}{|A\cup B|},$$

得出：

$$J_{sim}=\frac{|M_1\cap M_2|}{|M_1\cup M_2|}=\frac{\sum_{j=1}^{\lambda}v_{1j}\wedge v_{2j}}{\sum_{j=1}^{\lambda}v_{1j}\vee v_{2j}}=$$

$$\frac{\sum_{q_j\in Q}\min\left(M_1(q_j), M_2(q_j)\right)}{\sum_{q_j\in Q}\max\left(M_1(q_j), M_2(q_j)\right)}。$$

可知，经过0-1编码后，Jaccard相似度计算仍旧不会存在误差。

6 方案效率分析

本节通过理论分析和实验对所提方案进行分析并给出实验结果。

我们并未将所提方案与现有的隐私保护相似度阈值查询方案进行性能对比，原因在于本文提出的方案是第一个在隐私保护相似度阈值查询中同时考虑多维多重集相似性和保护访问模式隐私的工作。由于该方案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方案的差异，我们在表1中提供了详细的特性对比。

在本文第二节中已对现有方案[11，22，24-28]进行具体分析，它们皆是解决在标准集合上的相似度查询问题，本文研究在一维或多维多重集上的相似度阈值查询问题，现有方案无法解决本文提出的问题。本文首先对与方案最接近的文献[11]进行具体分析：

文献[11]给出了隐私保护的一维标准集合的相似性查询方案，没有考虑集合是多重集的情况，也没有考虑在有多种数据类型的情况下如何解决隐私保护的相似性查询问题。

如果用文献[11]中的方案解决本文提出的隐私保护的一维多重集相似度阈值查询问题，则会导致计算结果有误差，因为文献[11]的方案没有考虑对从多重集中的元素进行编码，则在执行过程中只会选择重复元素中的一个，在计算Jaccard相似度时，得出的交集与并集的势并不是真实值。

同时，如果用文献[11]中的方案解决本文提出的隐私保护的多维多重集相似度阈值查询问题，则

表1　性能对比分析

性能	文献[22]	文献[24]	文献[25]	文献[26]	文献[27]	文献[28]	本文协议1	本文协议2
集合维度	多维	一维	一维	多维	多维	多维	一维	多维
多重集	×	×	×	×	×	×	√	√
关键字查询	*E*	*J*	-	*J*	*J*	*J*	*J*	*J*
单云服务器	×	√	-	-	×	-	√	√
合谋攻击	×	×	×	×	×	×	√	√
访问模式隐私	×	×	×	√	√	×	√	√
中间结果隐私	×	×	√	√	×	×	√	√
查询准确性	×	×	√	×	√	×	√	√

注：集合维度：“多维”表示参与计算的是多维集合，“一维”表示参与计算的是一维集合；多重集：“√”表示参与计算的是多重集，“×”表示参与计算的是标准集；关键字查询：“*E*”表示使用欧氏距离衡量相似度，“*J*”表示使用Jaccard系数，“-”表示其他；单云服务器：“×”表示没有使用单云服务器，“√”表示使用单云服务器；合谋攻击：“×”表示无法抵抗合谋攻击，“√”表示能够抵抗合谋攻击；访问模式隐私、中间结果隐私：“×”表示没有保护，“√”表示保护；查询准确性：“×”表示计算结果不准确，“√”表示准确。

需要分步执行该方案，即先对关键字集合进行相似性查询，再对数据向量进行查询。若两次查询返回不同的查询结果，则会泄露哪些集合只满足关键字集合的相似度阈值判定，哪些集合只满足数据向量的判定。

除此之外，文献[11]在构建kd-Tree时需数据拥有者从数据集中随机选择k个集合发送给查询用户，泄露了数据集信息；云服务器在计算过程中需进行过滤与精化的分步计算，泄露了中间计算结果。

因此文献[11]提出的方案不满足本文应用场景所需的安全性与准确性，不能解决本文提出的问题。

文献[22]提出了一种面向关键字的隐私保护多维相似度查询方案，但该方案没有考虑集合是多重集的情况，也未能保护访问模式隐私。此外，文献[22]采用欧氏距离来衡量两个n维数据向量的相似度，而本文则采用Jaccard系数来衡量两个多重集中数据向量的相似度，即将两个λ维数据向量通过编码转换为两个多重集。选择Jaccard系数的原因在于欧氏距离反映了在数值向量中数值之间的几何距离，本文关注的是多重集之间的相似度，Jaccard相似度通过计算交集与并集的比值，能够更准确地反映集合之间的相似度，且被广泛应用。

文献[22]不适用于解决本文提出的问题，未考虑将向量转化为集合计算Jaccard相似度时会存在重复出现的元素，不适用于解决一维或多维多重集相似度阈值查询问题，而我们的方案经过扩展可以计算使用欧氏距离衡量相似度的多维相似度查询。

文献[22]假设*DO*和*QU*是诚实的，但这一假设在现实应用场景中并不合理。实际上，*DO*可能会对多个*QU*获得的查询结果的分布产生兴趣，而*QU*也可能对*DO*的数据集产生好奇。在我们的安全性需求中，*DO*和*QU*被视为半诚实的，且我们不仅防止*QU*和*CS*的合谋，还防止*CS*和*DO*的合谋。文献[22]中的云服务器在计算过程中需要进行过滤和精化的分步计算，这会泄露中间计算结果，从而进一步增加隐私泄露的风险，因此文献[22]不满足本文提出的安全性需求。

同时，文献[22]使用了两个云服务器，其中一个持有*DO*的私钥。然而，在现实应用中，很难确保这两个云服务器完全不合谋，这种做法可能会增加*DO*私钥和数据集泄露的风险。尽管文献[22]在构建BD-PB树时，将数据向量和关键字编码为相同长度的向量后进行加密，但云服务器在对BD-PB树进行搜索和剪枝时，执行的判断是不同的：当查询到数据向量时，云服务器需要对两个密文进行大小比较；而查询到关键字时，仅需判断密文是否为零。这种做法可能会向云服务器泄露更多关于BD-PB树的隐私信息。

此外，为提高查询效率，文献[22]引入了Hilbert排除条件，并在构造BD-PB树时，基于每个节点的两个枢轴递归构建树结构。这两个枢轴是从每个节点对应数据集中的最大值和最小值选取的。枢轴选择不当可能导致BD-PB树无法构建成功，或者查询结果出现误差。因此，基于上述差异，本文没有与文献[22]进行性能对比。

我们也对文献[24-28]进行了深入分析，除表1所示外，文献[25]仅以集合交集的势的最大值定义最大相似度，无法反映重复元素对相似度的影响，而以Jaccard系数衡量相似度能够避免这种局限性。如果用文献[25]解决我们提出的问题，如何定义相似度最大值都会使查询结果存在误差，且解决多维多重集相似度阈值查询需要两次编码两次查询，会泄露更多中间计算结果，而用我们的方案稍作修改就可以解决文献[25]提出的问题。尽管文献[26-28]研究的问题是在任意空间范围中的关键字相似性查

询，但它们都以Jaccard系数衡量相似度，因此我们也对其进行了深入分析，具体如表1所示。

综上所述，我们的方案可以通过一次查询，解决一维或多维多重集相似度阈值查询问题，且能得到准确的计算结果，比文献[22,24-28]提出的方案安全性更高、通信复杂性更低且应用范围更广。

6.1 理论分析

6.1.1 计算复杂性分析

本文方案以模指数运算次数作为复杂性度量指标，因为模指数运算是本文协议中最耗时的操作，记模指数运算的时间为T_e，模乘运算的时间为T_m，选择$g=1+hN$后，在加密和解密过程中只需进行1次模指数运算。

在一维多重集相似度阈值查询方案中，DO利用向量表示一维多重集，需要加密的数据和多重集全集的势λ有直接关联，若QU查询满足条件$\{\boldsymbol{M}_i|(X_i, Y)\geqslant\tau\}$的多重集，$DO$加密一个向量需要$\lambda$次模指数运算，$\lambda$次模乘运算，计算开销为$\lambda(T_e+T_m)$。

若想一次预处理，将n个向量全部预处理外包存储于服务器中，DO总共需要$n\lambda$次模指数运算，$n\lambda$次模乘运算，计算开销为$n\lambda(T_e+T_m)$。

本方案只需要一次预处理供多用户多次查询，假设查询次数为l，如果加密一个向量，每次查询的平均成本是$\frac{\lambda(T_e+T_m)}{l}$。如果加密$n$个向量，每次查询的平均成本是$n\lambda\frac{(T_e+T_m)}{l}$，将这种计算开销称为平摊计算开销。当$l\to\infty$，$\frac{\lambda(T_e+T_m)}{l}\to0$，$n\lambda\frac{(T_e+T_m)}{l}\to0$。因此，多次查询时，可以忽略向量加密算法的计算开销。

QU查询时，使用口令登录服务器选择对应向量中的密文，计算

$$C_i=C_{i1}\times\tau_2-C_{i2}\times\tau_1,$$

即计算$C_{i1}\times\tau_2$需要τ_2次模乘运算，其余计算同理。加上最后对所有密文相乘的1次模乘运算，计算$[[C_i']]=[[C_i+\alpha_i]]$需要1次模指数运算，1次模乘运算，则$QU$查询一个多重集的过程共需要1次模指数运算，$(\tau_2+\tau_1+1)$次模乘运算，查询$n$个多重集需要$n$次模指数运算，$n(\tau_2+\tau_1+1)$次模乘运算。

最后DO解密需要n次模指数运算，n次模乘运算，所以查询者对n个多重集查询1次的计算开销为$2nT_e+n(\tau_2+\tau_1+2)T_m$。

在多维多重集相似度阈值查询方案中，DO利用向量表示多维多重集，需要加密的数据和多重集全集的势与关键字全集的势有直接关联，假设多重集全集的势与关键字全集的势之和为$\lambda+h$，若QU查询满足条件

$$\{\boldsymbol{M}_i|J_{sim}(\boldsymbol{M}_i, Y)\geqslant\tau\wedge O_y\subseteq O_i\}$$

的多重集时，DO加密一个向量需要$\lambda+h$次模指数运算，$\lambda+h$次模乘运算，计算开销为$(\lambda+h)(T_e+T_m)$。若想一次预处理，将n个向量全部预处理外包存储于服务器中，DO总共需要$n(\lambda+h)$次模指数运算，$n(\lambda+h)$次模乘运算，计算开销为$n(\lambda+h)(T_e+T_m)$。

本方案只需要一次预处理供多用户多次查询，假设查询次数为l，如果加密一个向量，每次查询的平均成本是$\frac{(\lambda+h)(T_e+T_m)}{l}$。如果加密$n$个向量，每次查询的平均成本是$n(\lambda+h)\frac{(T_e+T_m)}{l}$，将这种计算开销称为平摊计算开销。当$l\to\infty$，$\frac{(T_e+T_m)n(\lambda+h)}{l}\to0$，$\frac{(T_e+T_m)n(\lambda+h)}{l}\to0$。因此，多次查询时，可以忽略向量加密算法的计算开销。

QU查询时，使用口令登录服务器选择对应向量中的密文，计算

$$C_i'=C_{i1}\times\tau_2-C_{i2}\times\tau_1-C_{i3}\times\beta_i,$$

计算$C_{i1}\times\tau_2$需要τ_2次模乘运算，其余计算同理。加上最后对所有密文相乘的2次模乘运算，则QU查询一个多重集的过程共需要$\tau_2+\tau_1+\beta_i+2$次模乘运算，查询n个多重集需要$n(\tau_2+\tau_1+\beta_i+2)$次模乘运算。

最后，DO解密需要n次模指数运算，n次模乘运算，所以查询者对n个多重集查询1次的计算开销为$nT_e+n(\tau_2+\tau_1+\beta_i+3)T_m$。

6.1.2 通信复杂性分析

本方案以交互次数和传输的密文数量来衡量通信成本。

表1以通信次数评估通信开销，DO需要与其他实体通信两次，即DO将向量密文外包，返回给QU解密结果，各需一次通信，CS收到查询请求后将向量密文发送给经过身份验证的QU需要一次通信，QU将同态计算后的结果发送给DO需要一次通信，共计四次。

在Paillier密码系统中，选取两个k比特的大素数p,q，并令$N=p\times q$，密文$c=g^m r^N \bmod N^2$，因此，使用Paillier加密算法加密一个密文所需的平均比特数是$2k$比特。在一维多重集查询方案中，DO和CS各需传输n个势为λ的多重集的密文，即$2kn\lambda$比特，QU需传输n个同态计算后的密文，即$2kn$比特，DO将解密后的最终查询结果以明文形式发送给查询用户，为$2kn$比特；同理可得，在多维多重集查询方案中DO和CS各需传输n个势为$\lambda+h$的多重集的密文，为$2kn(\lambda+h)$比特，QU需传输n个同态计算后的密文，为$2kn$比特，DO将解密后的最终查询结果以明文形式发送给查询用户，为$2kn$比特。

具体分析见表2和表3。

表2　本文方案的计算复杂性和通信复杂性

	加密外包成本	查询成本	通信次数
一维多重集查询	$n\lambda(T_e+T_m)$	$2nT_e+n$ $(\tau_2+\tau_1+2)T_m$	4
多维多重集查询	$n(\lambda+h)(T_e+T_m)$	nT_e+n $(\tau_2+\tau_1+\beta_i+3)T_m$	4

表3　通信复杂性（传输密文数量，单位：比特）

	向量密文外包	返回密文向量	发送同态计算密文	返回明文查询结果
一维多重集查询	$2kn\lambda$	$2kn\lambda$	$2kn$	$2kn$
多维多重集查询	$2kn(\lambda+h)$	$2kn(\lambda+h)$	$2kn$	$2kn$

6.2　实验分析

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：本文方案均采用Paillier算法，设定素数的比特数为1024，该设定提供112比特的等效对称加密密钥安全强度。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：图2表示在一维多重集相似度阈值查询方案中，执行时间随多重集全集的势λ与多重集的个数n变化的二维图。由图2可知，执行时间随多重集全集的势λ与多重集的个数n增加而增长。

图3表示在多维多重集相似度阈值查询方案中，执行时间随多重集全集的势与关键字全集的势之和$\lambda+h$与多重集的个数n变化的二维图。由图3可知，执行时间随多重集全集的势与关键字全集的势之和$\lambda+h$与多重集的个数n增加而增长。

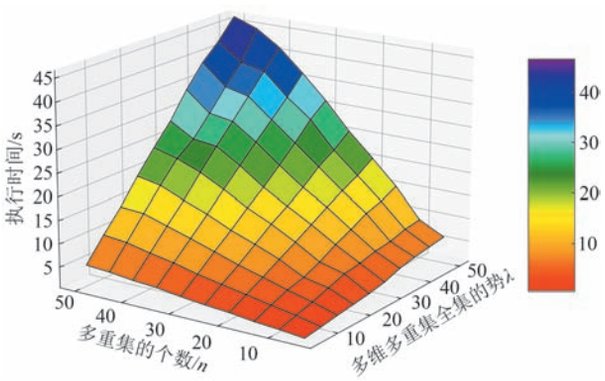

图2　一维多重集相似度阈值查询方案中执行时间随多重集的势和多重集的个数的变化规律

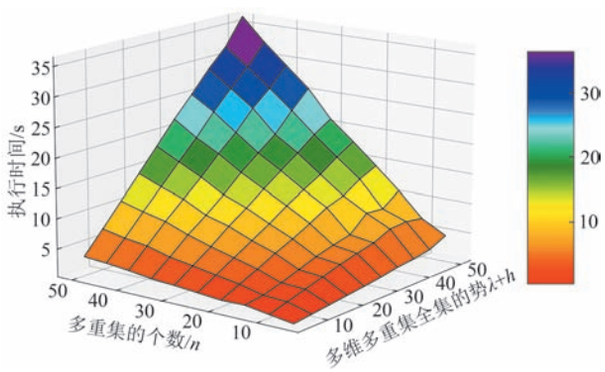

图3　多维多重集相似度阈值查询方案中执行时间随多重集的势和多重集的个数的变化规律

(1)一维多重集加密的实验结果：

图4(a)和(b)表示在一维多重集相似度阈值查询方案中，加密所有多重集所需的时间。

由图4(a)可知，当多重集的个数$n=50$时，一维多重集相似度阈值查询方案的加密时间随多重集全集的势λ增加而增长。

由图4(b)可知，当多重集全集的势$\lambda=50$时，一维多重集相似度阈值查询方案的加密时间随多重集的个数n增加而增长。

(2)一维多重集相似度阈值查询方案中的查询实验结果：

图4(c)表示在一维多重集相似度阈值查询方案中，当多重集的个数$n=50$时，所需的查询时间。

(3)多维多重集加密的实验结果：

图5(a)和(b)表示在多维多重集相似度阈值查询方案中，加密所有多重集所需的时间。

由图5(a)可知，当多重集全集的势$\lambda+h=100$时，多维多重集相似度阈值查询方案的加密时间随多重集的个数n增加而增长。

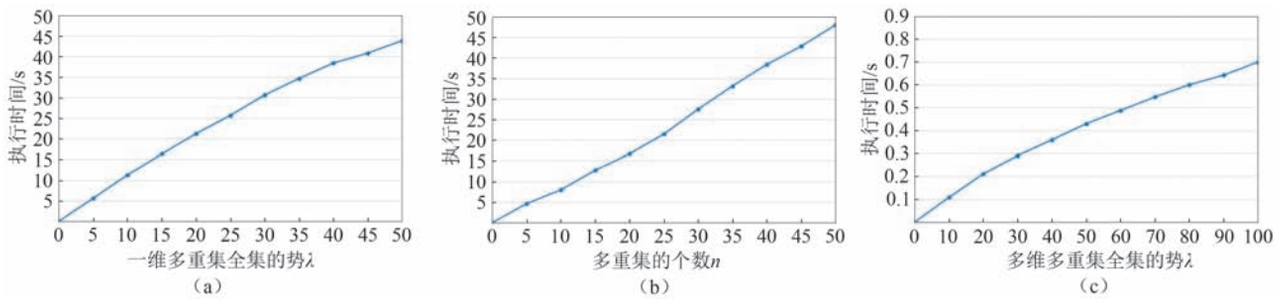

图4 一维多重集相似度阈值查询方案数据集加密和查询所需执行时间

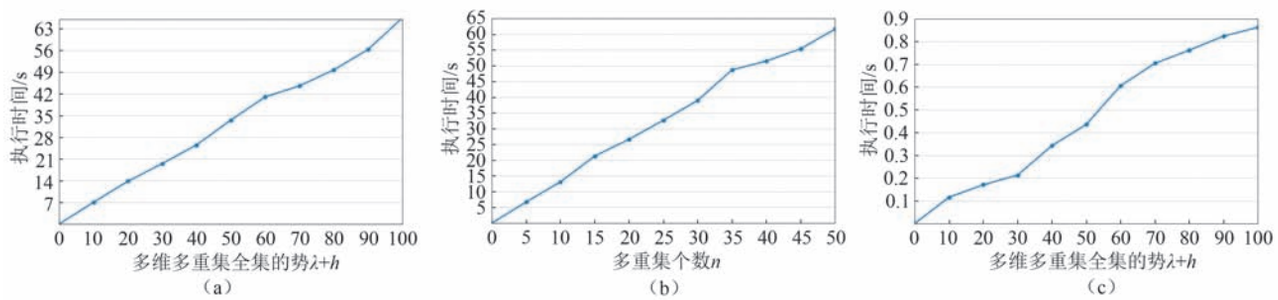

图5 多维多重集相似度阈值查询方案数据集加密和查询所需执行时间

由图5(b)可知,当多重集的个数n=50时,多维多重集相似度阈值查询方案的加密时间随多重集全集的势$\lambda+h$增加而增长。

(4)多维多重集相似度阈值查询方案中的查询实验结果:

图5(c)表示在多维多重集相似度阈值查询方案中,当多重集的个数n=50时,所需的查询时间。

图6表示在一维多重集相似度阈值查询方案中,数据集加密所需的执行时间随多重集全集的势λ与多重集的个数n变化的柱状图。由图6可知,数据集加密时间随λ增大而增长,随n增大而增长。

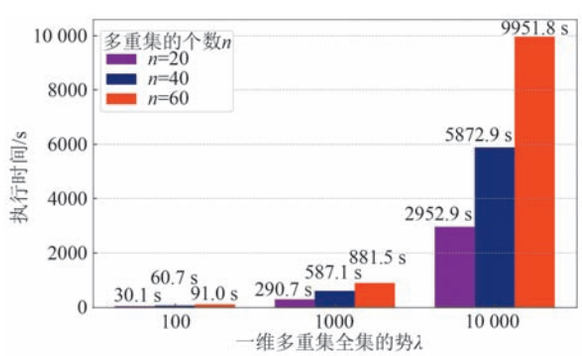

图6 一维多重集相似度阈值查询方案加密时间

图7表示在一维多重集相似度阈值查询方案中,查询所需的执行时间随多重集全集的势λ与多重集的个数n变化的折线图。由图7可知,查询时间随λ增大而增长,随n增大而增长。

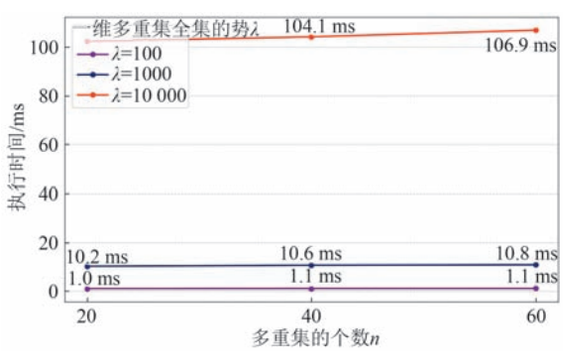

图7 一维多重集相似度阈值查询方案查询时间

图8表示在多维多重集相似度阈值查询方案中,数据集加密所需的执行时间随多重集全集的势$\lambda+h$与多重集的个数n变化的柱状图。由图8可知,数据集加密时间随$\lambda+h$增大而增长,随n增大而增长。

图9表示在多维多重集相似度阈值查询方案中,查询所需的执行时间随多重集全集的势$\lambda+h$

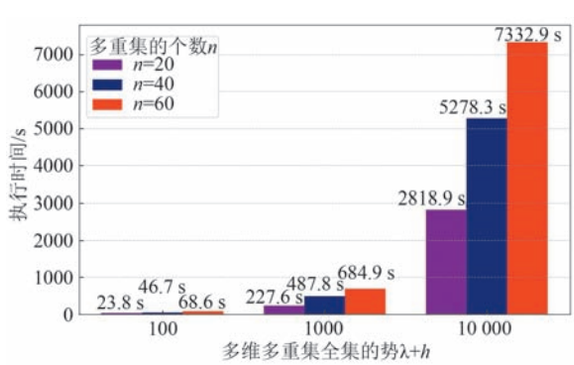

图8 多维多重集相似度阈值查询方案加密时间

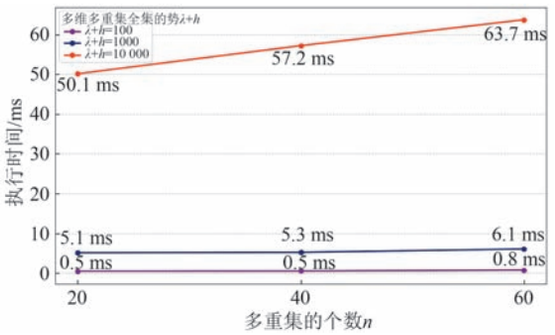

图9　多维多重集相似度阈值查询方案查询时间

与多重集的个数n变化的折线图。由图9可知，查询时间随λ增大而增长，随n增大而增长。

实验没有在公开数据集上验证的主要原因是现有的公共数据集通常以标准集的形式存储数据，缺乏本文方案所需的一维或多维多重集，尽管在实验部分采用的数据集存在一定限制，我们仍模拟现实数据，并扩大多重集势的范围以确保结论的可靠性，即将一维（多维）多重集的势从λ=100（$\lambda+h$=100）、扩展至λ=10 000（$\lambda+h$=10 000），覆盖小规模（λ=100）（$\lambda+h$=100）、中规模（λ=1000）（$\lambda+h$=1000）、大规模（λ=10 000）（$\lambda+h$=10 000）三类场景。新增图9至图10(d)展示执行时间与一维（多维）多重集全集的势λ（$\lambda+h$），多重集个数n的关系曲线。

图10(a)表示在多维多重集相似度阈值查询方案中，当h=100，n=60时，数据集加密所需的执行时间随多重集全集的势$\lambda+h$变化的折线图。由图10(a)可知，数据集加密时间随$\lambda+h$增大而增长。

图10(b)表示在多维多重集相似度阈值查询方案中，当h=100，n=60时，查询所需的执行时间随多重集全集的势$\lambda+h$变化的折线图。由图10(b)可知，查询时间随$\lambda+h$增大而增长。

图10(c)表示在多维多重集相似度阈值查询方案中，当λ=100，n=60时，数据集加密所需的执行时间随多重集全集的势$\lambda+h$变化的折线图。由图10(c)可知，数据集加密时间随$\lambda+h$增大而增长。

图10(d)表示在多维多重集相似度阈值查询方案中，当λ=100，n=60时，查询所需的执行时间随多重集全集的势$\lambda+h$变化的折线图。由图10(d)可知，查询时间随$\lambda+h$增大而增长。

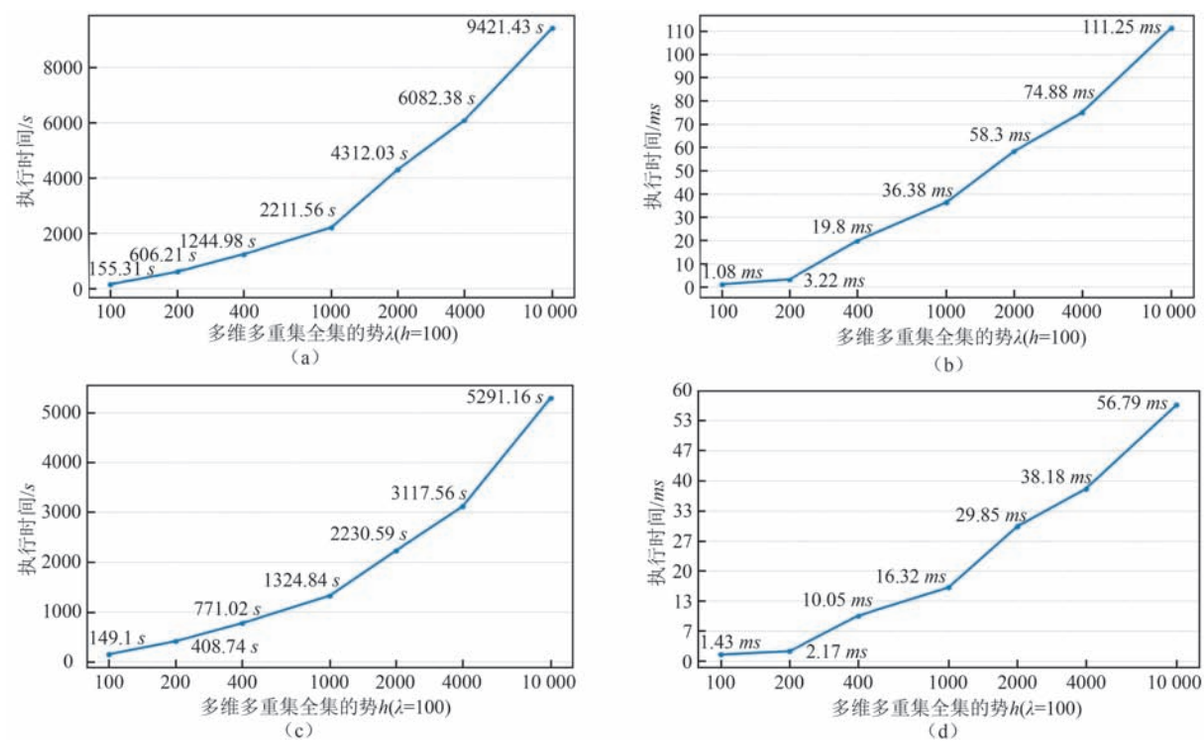

图10　多维多重集相似度阈值查询方案数据集加密和查询所需执行时间

7　总结与展望

多维多重集相似度阈值查询在物联网中具有重要的应用价值。本文提出了一个具有强隐私性的多维多重集阈值查询方案。首先，我们利用0-1编码和Paillier密码系统设计了两个多重集Jaccard相似度阈值查询协议，这两个协议能够高效、准确地进行

并行查询，适用于不同的数据类型。随后，基于这两个协议，我们提出了一个在单云服务器下高效且隐私保护的精确多维多重集相似度阈值查询方案。最后，通过模拟范例证明了该方案在半诚实模型下的安全性。尽管本文提出的方案有效解决了多维多重集相似度阈值查询问题，但仍有一些待进一步改进或探索的方面，例如，如何降低数据拥有者的计算开销，使该方案能够更广泛地应用，将成为我们未来研究的一个重要方向之一。

参 考 文 献

[1] Le, T. T. N., Phuong T. V. X. Privacy preserving jaccard similarity by cloud-assisted for classification. Wireless Pers Commun, 2020, 112: 1875-1892

[2] C. Dai. An intelligent recommendation system for improvisation playing and singing built with database technology// Proceedings of the 2023 International Conference on Computer Science and Automation Technology (CSAT). Shanghai, China, 2023: 64-70

[3] Q xiao, S yang, et al. Multi-resolution odd sketch for mining extended jaccard similarity of dynamic streaming sets. IEEE Transactions on Network Science and Engineering, 2024(11): 2399-2414

[4] Ma Xiu-Lian, Duan Yu-Wei, Li Shun-Dong. Asecure computing protocol for set similarity problems. Journal of Cryptologic Research, 2024, 11(5): 1029-1043
(马秀莲，段雨薇，李顺东. 集合相似问题的保密计算. 密码学报，2024，11(5): 1029-1043)

[5] Carlo Blundo, Emiliano De Cristofaro, Paolo Gasti. EsPRESSO: efficient privacy-preserving evaluation of sample set similarity. Computer Security, 2014, 3(22): 355-381

[6] C Guo, W Liu, et al. Secure similarity search over encrypted non-uniform datasets. IEEE Transactions on Cloud Computing, 2022, 10(3): 2102-2117

[7] J Fan, F Yuan. Recognition of junk short messages based on local sensitive hash knn algorithm//Proceedings of the 2022 International Conference on Artificial Intelligence, Information Processing and Cloud Computing (AIIPCC). Kunming, China, 2022: 356-359

[8] W Xu, J Zhang, et al. Privacy-preserving multi-cloud based dynamic symmetric searchable encryption//Proceedings of the 2021 2nd International Conference on Computer Communication and Network Security (CCNS). Xining, China, 2021: 176-181

[9] Jun Z, Shiqing H, et al. Privacy-preserving similarity computation in cloud-based mobile social networks. IEEE Access, 2020(8): 111889-111898

[10] B Zhao, Y Li, X Liu, Hwee Hwa Pang, H. DengRobert. FREED: an efficient privacy-preserving solution for person re-identification//Proceedings of the 2022 IEEE Conference on Dependable and Secure Computing (DSC). Edinburgh, UK, 2022: 1-8

[11] Y Zheng, et al. EPSet: efficient and privacy-preserving set similarity range query over encrypted data. IEEE Transactions on Services Computing, 2024, 17(2): 524-536

[12] Y, Zheng, et al. Efficient and privacy-preserving similarity range query over encrypted time series data. IEEE Transactions on Dependable and Secure Computing, 2021, 19(4): 2501-2516

[13] Islam, Mohammad Saiful, et al. Access pattern disclosure on searchable encryption: ramification, attack and mitigation// Proceedings of the 19th Annual Network and Distributed System Security Symposium (NDSS 2012. San Diego, USA, 2012: 1-15

[14] Kellaris Georgios, et al. Generic attacks on secure outsourced databases//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. New York, USA, 2016: 1329-1340

[15] Y Zheng, R Lu. Efficient privacy-preserving similarity range query based on pre-computed distances in eHealthcare// Proceedings of the GLOBECOM 2020-2020 IEEE Global Communications Conference. Taipei, China, 2020: 1-6

[16] Y Zheng, R Lu, Y Guan, J Shao. Efficient privacy-preserving similarity range query with quadsector tree in eHealthcare. IEEE Transactions on Services Computing, 2021, 15(5): 2742-2754

[17] Y Zheng, R Lu, S Zhang. Achieving privacy-preserving weighted similarity range query over outsourced ehealthcare data//Proceedings of the ICC 2022-IEEE International Conference on Communications. Seoul, Republic of Korea, 2022: 1251-1256

[18] Q Huang, G Yan, Y Yang. Privacy-preserving traceable attributebased keyword search in multi-authority medical cloud. IEEE Transactions on Cloud Computing, 2023, 11(1): 678-691

[19] B Shan, Y Yao, W Li, X Zuo. Fuzzy keyword search over encrypted cloud data with dynamic fine-grained access control// Proceedings of the 2022 IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Wuhan, China, 2022: 1340-1347

[20] C Li, et al. Efficient medical big data management with keywordsearchable encryption in healthchain. IEEE Systems Journal, 2022, 16(4): 5521-5532

[21] S Niu, M Song, L Fang, F Yu, S Han. Keyword search over encrypted cloud data based on blockchain in smart medical applications. Computer Communications, 2022(192): 33-47

[22] Z Zhang, H Bao, R Lu, C Huang. KMSQ: efficient and privacy-preserving keyword-oriented multidimensional similarity query in ehealthcare. IEEE Internet of Things Journal, 2024, 11(5): 7918-7934

[23] D Li, X Zhao, H Li, K Fan. Volume-hiding multidimensional verifiable dynamic searchable symmetric encryption scheme for cloud computing. IEEE Internet of Things Journal, 2024, 11(23): 37437-37451

[24] Y Zheng, R Lu, Y Guan, J Shao, H. Zhu. Achieving efficient and privacy-preserving exact set similarity search over encrypted data. IEEE Transactions on Dependable and Secure Computing, 2022, 19(2): 1090-1103

[25] Z Chen, et al. A new functional encryption scheme supporting privacy-preserving maximum similarity for web service platforms. IEEE Transactions on Information Forensics and Security, 2025, 20: 2621-2631

[26] S Zhang, et al. Performance enhanced secure spatial keyword similarity query with arbitrary spatial ranges. IEEE Transactions on Information Forensics and Security, 2024, 19: 5272-5285

[27] C Wang, et al. PPSKSQ: towards efficient and privacy-preserving spatial keyword similarity query in Cloud. IEEE Transactions on Cloud Computing, 2025: 1-16

[28] S Zhang, S Ray, R Lu, Y Guan, Y Zheng, J Shao. Efficient and privacy-preserving spatial keyword similarity query over encrypted data. IEEE Transactions on Dependable and Secure Computing, 2023, 20(5): 3770-3786

LI Shun-Dong, Ph. D., professor. His main research interests include cryptography and information security.

DU Ji-Xin, M. S. candidate. Her main research interest is information security.

WU Chuan-Yu, M. S. candidate. Her main research interest is information security.

YU Jia-Tong, M. S. candidate. Her main research interest is information security.

Background

Set similarity queries are vital for IoT applications, enabling data-driven decisions by evaluating dataset relationships. Cloud computing advancements demand scalable techniques for real-time big data processing.

Cloud outsourcing offers cost-effective storage and computation but exposes sensitive data to provider access risks. Robust protection mechanisms are essential to prevent breaches.

To protect sensitive data in cloud environments, encryption serves as a fundamental safeguard. However, performing efficient similarity queries on encrypted data presents significant challenges, as traditional set-similarity methods cannot be directly applied—encryption inherently obscures the underlying data structure. To overcome this limitation, methods must be employed to enable accurate similarity assessments without compromising data confidentiality.

Existing research has proposed multiple cryptographic approaches, including homomorphic encryption and secure multi-party computation (MPC). Nevertheless, achieving an optimal balance among query efficiency, security guarantees, and result accuracy remains an open challenge. Some methods provide strong security but suffer from impractical computational overhead, while others prioritize performance at the expense of rigorous privacy protection.

A demanding scenario involves multiset similarity threshold queries, where duplicate elements complicate comparisons. Such queries—for instance, those based on the Jaccard similarity metric—require to determine whether the similarity between two multisets exceeds a predefined threshold. Current solutions often fail to address this effectively, highlighting the demand for novel methods that simultaneously ensure provable security, computational efficiency, and practical accuracy in multiset operations.

In this article, we present a privacy-preserving multidimensional multiset threshold query scheme designed to address these challenges. Our approach builds upon two key protocols: multiset Jaccard similarity threshold query protocols using 0-1 encoding and the Paillier cryptosystem. The 0-1 encoding method ensures that multisets are represented in a way compatible with cryptographic operations, while the Paillier cryptosystem allows secure computation on encrypted data, preserving privacy throughout the query process.

Building on these protocols, we propose an privacy-preserving exact multidimensional multiset similarity threshold query scheme, designed for use with a single cloud server. This scheme enables queries to be executed on encrypted multisets without compromising data security. By combining cryptographic techniques with querying protocols, our scheme offers a practical solution to the challenge of performing similarity threshold queries on encrypted multisets in the cloud.

This work was supported by the National Key Research and Development Program of China (No. 2022YFB2703001).