面向众包的隐私保护任务匹配方案

宋甫元[1]，丁思洋[1]，王威[2]，姜琴[1]，付章杰[1]
（1.南京信息工程大学计算机学院、网络空间安全学院，江苏 南京 210044；2.西安交通大学信息与通信工程学院，陕西 西安 710049）

摘　要：众包作为一种新兴的任务执行和数据感知方式，在众多领域中得到了广泛的应用。然而，不可信的众包平台可能泄露用户隐私，因此用户需对上传的数据进行加密。众包平台通过加密空间关键词查询，在密文环境下匹配工人兴趣和位置以满足任务需求。为了实现密文环境下安全高效的众包任务匹配，提出一种基于空间关键词相似性查询的隐私保护任务匹配方案，通过Geohash算法和位图方法对位置和关键词进行编码，将空间关键词相似性查询转化为向量内积计算。安全性分析和实验结果表明，所提方案在任务匹配效率上优于现有SOTA方案，并能够有效保护任务请求者和工人的隐私。

关键词：众包；任务匹配；相似性查询；隐私保护
中图分类号：TP391
文献标志码：A
DOI：10.11959/j.issn.1000−436x.2025090

Privacy-preserving task matching scheme for crowdsourcing

SONG Fuyuan[1], DING Siyang[1], WANG Wei[2], JIANG Qin[1], FU Zhangjie[1]
1. School of Computer Science, Nanjing University of Information Science & Technology, Nanjing 210044, China
2. School of Information and Communications Engineering, Xi'an Jiaotong University, Xi'an 710049, China

Abstract: Crowdsourcing has become a crucial paradigm for task execution and data collection, with task matching serving as a fundamental application. Due to the potential untrustworthiness of crowdsourcing platforms, which may lead to the leakage of users' private information, users are required to encrypt their data prior to uploading. To fulfill task matching while preserving privacy, the crowdsourcing platform employs encrypted spatial keyword queries to perform task matching of workers' interests and locations. To achieve secure and efficient crowdsourcing task matching, a privacy-preserving spatial keyword similarity-based task matching (SKSTM) scheme for crowdsourcing was proposed. SKSTM encoded locations and keywords by using the Geohash algorithm and bitmap representation, transforming spatial keyword similarity search into inner product calculations. Security analysis and experimental results demonstrate that SKSTM outperforms state-of-the-art schemes in task matching while effectively preserving the privacy of both task requesters and workers.

Keywords: crowdsourcing, task matching, similarity search, privacy protection

0　引言

众包作为一种重要的数据采集与任务匹配模式，已被广泛应用于多个领域，包括医疗健康监测、环境感知、路径规划以及社会调查等[1-2]。众包这一概念由Howe[3]首次提出，其核心思想是在互联网平台上，任务请求者可以将传统上由特定组织或个体完成的任务分发给一个开放的、大规模的网络群体，而工人则基于自身的兴趣、技能和位置

收稿日期：2025−02−13；修回日期：2025−05−06
通信作者：付章杰，fzj@nuist.edu.cn
基金项目：国家自然科学基金资助项目（No.62302230, No.62302229, No.U22B2062）；中国博士后科学基金资助项目（No.2024M751480）
Foundation Items：The National Natural Science Foundation of China (No.62302230, No.62302229, No.U22B2062), China Postdoctoral Science Foundation (No.2024M751480)

选择合适的任务，并完成匹配的任务。这种模式不仅能够帮助任务请求者降低成本、提高任务分配的灵活性，同时也为工人提供了多样化的任务选择和经济收益，使任务匹配过程更加高效。随着配备全球定位系统（GPS, global positioning system）的移动智能设备的日益普及，地理文本数据在过去几年中快速增长，其可以从一系列基于位置服务（LBS, location-based service）和社交网络自发地生成。例如，随着移动设备的广泛使用，Twitter每天能产生1 000万条带有地理标签的推文。类似地，在其他应用程序（如LinkedIn、Flickr、Netflix等）中，每天都有数以百万计的地理文本内容生成[4]。众包服务提供商可以通过这种地理标记并结合关键词为用户提供相应的服务。

任务匹配在众包中起着重要的作用，云服务器根据任务请求者的任务需求与工人的兴趣和位置进行任务匹配。具体来说，工人首先将自己的兴趣和位置上传至云服务器，任务请求者根据任务需求（位置范围查询和关键词模糊搜索）设置2个阈值r和τ，由云服务器执行阈值相似度搜索以实现基于空间兴趣的任务匹配。也就是说，工人兴趣与任务关键词之间的Jaccard相似度应大于或等于τ，工人位置与任务位置之间的欧氏距离应小于或等于r。通过空间关键词相似性查询，云服务器可以查询数据库中工人的兴趣和位置是否满足任务请求者的要求。因此需要设计一种能够支持空间关键词相似性查询的任务匹配方案。

由于数据的爆炸式增长，众包服务提供商倾向于将空间关键词数据外包给云服务器，以减轻本地存储和计算负担，但这种操作容易导致隐私泄露。例如，2019年，Facebook数亿用户数据（包括账户ID、点赞、评论等）被存储在未受保护的Amazon S3中，任何人都能公开访问。这次事件凸显了云环境下可能会带来的严重安全风险，并引发了人们对数据管理和隐私保护的广泛担忧。对于众包提供商来说，在将敏感数据外包给云服务器之前对其进行加密是一种可行的方法。然而，加密本质上破坏了原始数据的语义，从而限制了加密数据查询等服务。目前，基于空间关键词查询的隐私保护任务匹配得到了广泛的研究[5-9]。但是，现有的任务匹配方法存在诸多不足，例如，文献[5]仅支持单关键字查询，文献[10]仅支持多关键字精确搜索，文献[11]依赖于同态加密保护数据机密性，任务匹配效率低。因此，如何面向众包环境设计一种基于空间关键字相似性搜索的任务匹配方法，实现安全高效的任务匹配迫在眉睫。

针对密文环境下任务匹配效率低，难以支持空间关键词相似性查询问题，本文采用Geohash算法和BM位图对位置和关键词集进行编码，将空间关键词相似性查询转化成对应的向量内积操作，并使用矩阵加密方法对向量进行加密，提出基于空间关键词相似性查询的隐私保护任务匹配（SKSTM, spatial keyword similarity-based task matching）方案，实现高效且安全的任务匹配。本文的主要贡献可以概括为以下几点。

1) 本文采用Geohash算法和BM位图对用户位置和关键词编码，根据编码后的用户位置和关键词集合实现高效的位置关键词查询。本文使用Jaccard相似度作为关键词集合相似度匹配机制，将关键词集合相似度计算转换成向量的内积计算，实现多关键词模糊查询。基于此，本文提出了SKSTM方案，能够在密文环境下执行安全高效的任务匹配。

2) SKSTM将空间关键词相似性查询转换成向量的内积计算，并使用基于矩阵加密的安全向量内积计算进行任务匹配，仅进行一次内积计算就能同时判断工人的位置和关键词是否满足任务需求，从而极大地提升密文环境下任务匹配效率。

3) 本文通过安全性分析，证明SKSTM在不可区分选择明文攻击（IND-CPA, indistinguishable chosen-plaintext attack）模型下的安全性，并采用3种真实数据集进行实验，以证明本文方案的高效性。实验结果表明，SKSTM方案在任务匹配效率上远高于现有的空间关键词查询方案。

1 相关工作

1.1 面向众包的隐私保护任务匹配

随着众包用户的快速增长，众包中的任务匹配受到了广泛的关注。任务请求者通常根据兴趣、位置或声誉寻找合适的工人。文献[12]将任务匹配描述为多对一匹配问题，其中许多工人竞争同一任务，并且最符合任务要求的工人将被分配到该任务，该方案考虑了任务需求和工人利益。近年来，隐私泄露问题在空间众包的任务匹配中愈发

严峻。然而，现有的大多数研究集中在任务和工人信息的隐私保护上。文献[13]基于轻量级隐私保护方案。文献[14]基于曼哈顿距离评估方法来衡量扰动位置下工人与任务的相关性。文献[15]提出了基于工人长短期时空偏好的众包任务分配方案。文献[16]提出了隐私保护的在线多任务分配方案，结合同态加密技术和路径规划算法，在保护用户隐私的同时，显著减少了工人的总移动距离。文献[17]设计了一个区块链辅助、可公开验证、隐私保护的众包任务方案，以支持众包平台灵活和个性化的任务-工人匹配。文献[18]提出了基于聚类的众包空间任务分配优先队列算法。文献[19]提出了一种众包任务推荐方案，利用多项式函数保护任务需求和工人兴趣的隐私，设计了一种关键词偏离方法，实现多任务需求和多工人兴趣之间的多关键词匹配。然而，该方案只支持布尔关键词查询，即确定查询关键词集合是否完全包含在给定关键词集合中，这在众包应用场景下缺乏实用性。

1.2　空间关键词查询

空间关键词查询的目的是找到既满足空间范围又满足查询关键词条件的对象。由于在LBS中的广泛应用，一些隐私保护的空间关键词查询方案已被设计来保护外包数据和查询请求的隐私[20-22]。文献[23]提出了一种在加密数据上进行关键词搜索的公钥加密方案，该方案使服务器能够测试加密的关键词集中是否包含特定的关键词，且不会泄露任何隐私信息，但该方案只支持单关键词查询。文献[24]提出了一种隐私保护的加权邻近匹配空间关键字查询方案，利用目标导向空间关键字树并结合全同态加密协议，实现对加密地理文本数据的高效查询，同时保护用户查询隐私。文献[25]提出了一种基于布隆过滤器的空间文本编码方法，将空间和文本信息映射到布隆过滤器中，并使用对称矩阵加密算法对其进行加密，采用基于内积的匹配操作进行密文查询。文献[6]采用隐藏向量加密和布隆过滤器提出了隐私保护的布尔空间关键词查询方案。然而上述方案只考虑关键词精确查询，无法实现关键词模糊搜索，基于此，文献[10]提出了一种使用矩阵加密的隐私保护空间关键词相似性查询方案。然而，该方案采用欧氏距离相似度测量关键词相似性，仅适用于数值向量计算。对于关键词集合来说，它无法有效衡量集合之间的相似性，且受关键词数量和分布的影响较大，Jaccard相似度匹配机制更为适用。文献[11]提出了一个隐私保护的空间关键词相似性查询方案，能够安全地确定范围约束，计算关键词Jaccard相似度，并隐藏访问模式。然而，由于该方案采用同态加密保护数据机密性，需要进行复杂的指数运算和双线性映射，密文查询效率较低。

与以往的工作不同，本文的目标是设计一个面向众包的隐私保护空间关键词相似度任务匹配方案，可以实现位置和关键词相似性匹配，同时具有较高的匹配效率和安全性。为突出SKSTM的不同之处，表1将本文方案与现有相关方案在功能性（加密方式、多关键词、查询形式、查询效率）和安全性（已知明文攻击（KPA, known plaintext attack）、选择明文攻击（CPA, chosen plaintext attack））方面进行对比。

表1　　方案对比

方案	加密方式	多关键词	查询形式	查询效率	安全性
文献[7]	SHVE	√	Boolean	高	CPA
文献[10]	MSSAC	√	Euclidean	高	CPA
文献[11]	FHE	√	Jaccard	低	CPA
文献[25]	ASPE	√	Boolean	高	KBA
文献[26]	EASPE	√	Boolean	高	CPA
文献[27]	PKE	×	Boolean	较高	KBA
SKSTM	EASPE	√	Jaccard	高	CPA

2　问题提出

2.1　系统模型

如图1所示，本文的系统模型共由4个实体组成，分别是云服务器、密钥管理中心（KMC, key management center）、工人以及任务请求者。

云服务器：云服务器具有强大的存储和计算能力，能存储任务请求者和工人上传的加密信息并进行任务匹配。

密钥管理中心：密钥管理中心是一个受信任的第三方，用于验证注册者的身份和生成密钥，并将生成的密钥发送给相应实体。

工人：工人首先在密钥管理中心注册，注册成功后获得加密密钥。工人将自己的位置和兴趣关键

词加密上传至云服务器进行任务匹配。云服务器将匹配成功的加密任务信息发送给工人。工人执行任务获得任务结果，然后使用对称密钥加密任务结果，并将加密任务结果返回给云服务器。

任务请求者：任务请求者首先在密钥管理中心注册，注册成功后获得加密密钥。任务请求者将任务请求（包括位置范围和查询关键词）通过密钥加密上传至云服务器，进行任务匹配。云服务器匹配成功后，将工人完成的任务结果返回给任务请求者。

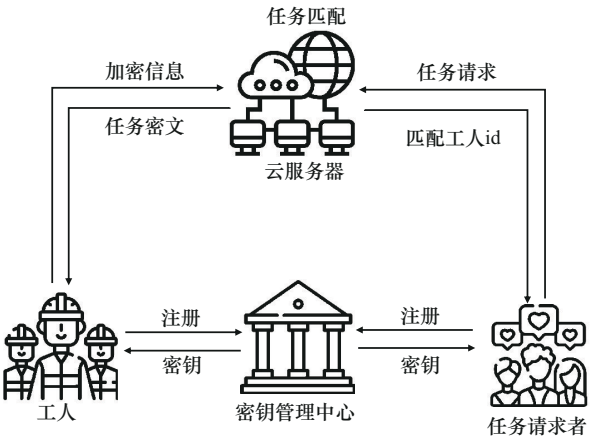


图1　系统模型

2.2　威胁模型

在本文的威胁模型中，密钥管理中心是完全可信的，密钥管理中心与其他参与实体之间的通信是安全的；任务请求者是完全受信任的；云服务器是半诚实的，云服务器遵循所设计的协议，但可能试图获取工人和任务请求者的一些敏感信息（如任务请求、任务关键词以及工人的兴趣和位置）；工人是半诚实的，他们可能会尝试学习任务请求者和其他工人的敏感信息。此外，本文假定云服务器与工人之间不会合谋。云服务器作为大型服务提供商（如iCloud、美团平台），需维护商业信誉与合规性，主动发起合谋攻击将面临法律风险和声誉损失，因此合谋动机较低。此外，合谋攻击的检测技术（如审计日志、行为分析）已逐渐成熟，此类攻击一旦被发现，将对云服务提供商的运营造成重大影响[27]。基于云服务器可能获得的可用信息，本文考虑如下2种攻击模型。

1) KPA：云服务器可以获得若干明文-密文对（如工人的兴趣和相应任务密文、任务请求者的任务需求和相应的查询陷门）。云服务器尝试使用这些明文-密文对获得用于解密其他密文的密钥。

2) CPA：云服务器可以选择任意明文，获得对应的密文，比已知明文攻击的攻击能力更强，例如，云服务器可以观察多个密文以及对应的明文。在该模型中，云服务器除了知道加密空间数据、索引和陷门之外，还可以访问与其选择的明文相对应的对象密文。

2.3　问题陈述

假设$D=\{o_1,o_2,\cdots,o_n\}$是一个空间关键词数据库。数据库D中的每个对象$o_i\left(i\in[1,n]\right)$可以表示成$\{W_i,L_i\}$。其中，$W_i=\{w_{i,1},w_{i,2},\cdots,w_{i,n_i}\}$是一个关键词集合，代表工人的兴趣描述；$L_i=(x_i,y_i)$表示工人位置的纬度和经度。每个工人$u_i$根据其位置和兴趣从云服务器获得任务分配。在任务请求者u_q向云服务器发布新的任务需求Q之前，任务请求者设置用于任务匹配的2个约束阈值。首先，任务请求者设置一个关键词相似度阈值τ，工人兴趣关键词和任务请求的关键词Jaccard相似度应当大于或等于τ。其次，任务请求者生成位置范围阈值r，工人位置和任务位置之间的欧氏距离应当在阈值r内。

对于任务匹配，云服务器需要计算工人兴趣与任务关键词之间的Jaccard相似度，以及工人与任务之间的欧氏距离，并选择Jaccard相似度大于或等于τ，且欧氏距离小于或等于r的工人。如图2所示，共有5个工人，其中，u_1,u_3,u_5满足关键词约束，u_4,u_5满足位置范围约束。因此，工人u_5满足请求者u_q发布的任务需求。最后，云服务器将加密的任务分配给工人u_5来完成。

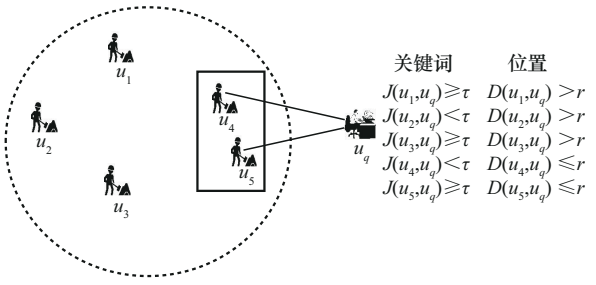


图2　众包任务匹配示例

2.4　设计目标

本文的目标是实现安全高效的众包任务匹配，同时确保在上述威胁模型下任务请求者的需求以及工人的位置和兴趣不会泄露。本文SKSTM方案应

达到以下目标。

位置关键词相似性匹配：本文方案应支持多工人多请求者的位置和多关键词相似性任务匹配。

数据隐私：工人的位置和兴趣信息，以及任务请求者的任务需求的机密性均应受到保护，不应泄露给云服务器或未经授权的实体，确保任何第三方无法获取这些敏感数据的具体内容。

陷门不可链接性：陷门应随机化，云服务器无法从加密陷门中推测查询结果与查询陷门之间的内在关系，同时任意 2 个查询陷门在任务匹配过程中无法关联。

查询结果隐私：云服务器返回的查询结果不应泄露给他人，而工人和任务请求者可以获得符合查询条件正确的查询结果。

效率：实现隐私保护将不可避免地产生额外的计算成本。因此，本文方案的目标是在执行隐私保护的空间关键词相似性任务匹配时最大限度地降低计算成本。

3　预备知识

本节主要回顾相关的背景知识，包括 Geohash 算法[18,21,28]、Jaccard 相似度匹配、增强型安全内积计算方案 EASPE [8]。

3.1　Geohash 算法

Geohash 算法是一种将经纬度坐标编码为字符串的空间索引方法，通过递归二分的方式将平面空间划分为网格，并交替对经度和纬度进行二进制编码，最终将二进制编码转换为 Base32 编码。例如，在图 3 中，点 1“wtmk70k”表示中心点经纬度为 (31.3835, 120.0354)±76 m 的矩形范围。Geohash 编码的精度如表 2 所示。

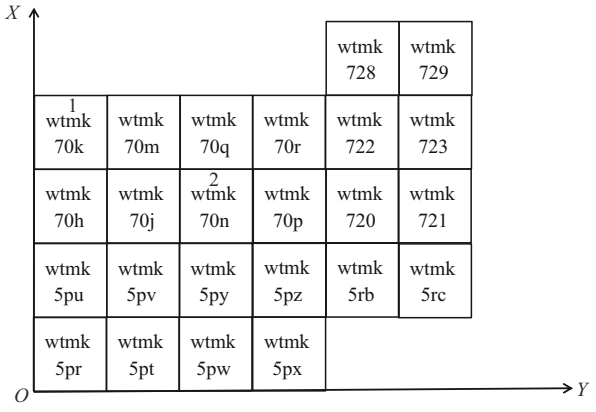


图 3　Geohash 编码示例

表 2　　Geohash 编码的精度

Geohash 编码长度	网格宽度/km	网格高度/km	误差/km
1	5 000	5 000	±2 500
2	1 250	625	±630
3	156	156	±78
4	39.1	19.5	±20
5	4.89	4.89	±2.4
6	1.22	0.61	±0.61
7	0.153	0.153	±0.076
8	0.038 2	0.019 1	±0.019
9	0.004 77	0.004 77	±0.004 8
10	0.001 19	0.001 29	±0.001 2

本文通过匹配 Geohash 编码来确定两点之间的距离。例如，在图 3 中，点 1 和点 2 的 Geohash 编码具有相同的前缀“wtmk70”，则可以知道点 1 在点 2 的±610 m 范围内。因此，本文通过判断 2 个 Geohash 编码的相同前缀的个数来判断空间点是否在查询点的查询范围内。当 Geohash 编码的长度为 8 时，精度约为 19 m，那么一个 8 位的 Geohash 编码就可以精确对应真实世界中的一个空间点。若 Geohash 编码长度增加至 9 位，其精度可达约 4.8 m，10 位时精度可进一步提高至约 1.2 m。然而，Geohash 长度的增加会带来额外的计算和存储开销，同时对匹配范围的控制也更加严格，可能导致可匹配的候选工人数量减少。此外，较高精度的 Geohash 编码可能泄露更精确的位置信息，如编码长度为 9 位时攻击者可以通过编码后的范围推测出工人的大致位置，导致工人的位置隐私泄露。因此，本文选择 8 位 Geohash 编码，在保证空间精度的同时兼顾匹配效率与隐私保护。

3.2　Jaccard 相似度匹配

假设在众包系统中共有 n 个工人 $\{o_1,o_2,\cdots,o_n\}$，其中每个工人 o_i 有位置和关键词 2 个属性，$o_i=\{o_i.l,o_i.k\}$。

根据任务的要求，云服务器基于空间关键词相似性搜索执行任务匹配。本文通过以下方式定义基于空间关键词相似性的众包任务匹配。

定义 1　给定空间关键词字典 $\mathcal{W}\{w_1,w_2,\cdots,w_a\}$、任务请求 $Q=\{R,Q.k,\tau\}$、查询范围

$R=(Q.l,r)$和关键词相似性阈值τ，空间关键词相似性查询返回匹配的数据记录，满足：1)位置范围约束，$o_i.l\in R$，即工人o_i的位置应在查询范围R之内；2)关键词约束，与文献[11]相同。本文使用Jaccard相似度来计算关键词相似度，即

$$J(o_i.k,Q.k)=\frac{|o_i.k\cap Q.k|}{|o_i.k\cup Q.k|} \tag{1}$$

为了满足任务的要求，工人o_i和查询Q之间的关键词相似度$J(o_i.k,Q.k)\geqslant\tau$。

首先，对工人数据集中的关键词集合$\mathcal{W}$按照频率进行降序排序得到新的关键词集合$\mathcal{W}'$。然后，按照$\mathcal{W}'$为$Q.k$和$o_i.k$生成位图BM_q和BM_i。图4是一个关键词字典长度为9的位图构建示例。根据文献[29]中的结论，将BM_q中的前$m-(\lfloor(1-\tau)m+1\rfloor)$位设置为0，从而得到$\varepsilon(\mathrm{BM}_q)$，其中$m=|Q.k|$，$\tau$是关键词相似度阈值。只要满足$\mathrm{BM}_i\circ\varepsilon(\mathrm{BM}_q)>0$，那么$Q.k$和$o_i.k$两者之间的Jaccard相似度就大于$\tau$，$\circ$表示内积运算。详细证明过程见文献[29]。

3.3 增强型安全内积计算方案EASPE

ASPE[30]是一种重要的加密数据相似性搜索技术，它可以在密文环境下计算2个向量的内积。然而，文献[31]中证明了ASPE在选择明文攻击模型下是不安全的。因此，本文使用文献[8]中提出的增强型安全内积计算方案EASPE进行加密。

EASPE的具体步骤如下，假设p和q是2个d维向量。

1) 密钥生成：EASPE.KeyGen$(\lambda)\rightarrow$sk。输入安全参数λ，输出密钥sk。sk$=\{\boldsymbol{s},\boldsymbol{M}_1,\boldsymbol{M}_2,\pi,r_1,r_2,r_3,r_4,r_5,r_6\}$，其中，$s$是一个$(d+3)$维的随机二进制向量，$\boldsymbol{M}_1$和$\boldsymbol{M}_2$是2个$(d+3)\times(d+3)$的随机可逆矩阵，$\pi$是一个$\mathbb{R}^{d+3}\rightarrow\mathbb{R}^{d+3}$的随机置换，$r_1,r_2,r_3,r_4,r_5,r_6$是6个随机数，满足$r_1r_4+r_2r_5+r_3r_6=0$。

2) 索引加密：EASPE.Enc$(\boldsymbol{p},\mathrm{sk})\rightarrow\boldsymbol{C}$。首先根据数据向量$\boldsymbol{p}$生成扩充向量$\boldsymbol{p}_v=(\boldsymbol{p},r_1,r_2,r_3)$，然后对$\boldsymbol{p}_v$进行置换得到$\widehat{\boldsymbol{p}_v}=\pi(\boldsymbol{p}_v)$。根据二进制向量$\boldsymbol{s}$，生成$\widehat{\boldsymbol{p}_v}'[k]$和$\widehat{\boldsymbol{p}_v}''[k]$，如式(2)所示。

$$\begin{cases}\widehat{\boldsymbol{p}_v}'[k]=\widehat{\boldsymbol{p}_v}''[k]=\widehat{\boldsymbol{p}_v}[k],\boldsymbol{s}[k]=0\\ \widehat{\boldsymbol{p}_v}'[k]+\widehat{\boldsymbol{p}_v}''[k]=\widehat{\boldsymbol{p}_v}[k],\boldsymbol{s}[k]=1\end{cases} \tag{2}$$

最后得到加密索引，如式(3)所示。

$$\boldsymbol{C}=(\boldsymbol{M}_1^{\mathrm{T}}\widehat{\boldsymbol{p}_v}',\boldsymbol{M}_2^{\mathrm{T}}\widehat{\boldsymbol{p}_v}'') \tag{3}$$

3) 陷门生成：EASPE.TrapGen$(\boldsymbol{q},\mathrm{sk})\rightarrow\boldsymbol{T}_Q$。首先根据查询向量$\boldsymbol{q}$生成扩充向量$\boldsymbol{q}_v=(\boldsymbol{q},r_4,r_5,r_6)$，然后对$\boldsymbol{q}_v$进行置换得到$\widehat{\boldsymbol{q}_v}=\pi(\boldsymbol{q}_v)$。根据二进制向量$\boldsymbol{s}$，生成$\widehat{\boldsymbol{q}_v}'[k]$和$\widehat{\boldsymbol{q}_v}''[k]$，如式(4)所示。

$$\begin{cases}\widehat{\boldsymbol{q}_v}'[k]=\widehat{\boldsymbol{q}_v}''[k]=\widehat{\boldsymbol{q}_v}[k],\boldsymbol{s}[k]=1\\ \widehat{\boldsymbol{q}_v}'[k]+\widehat{\boldsymbol{q}_v}''[k]=\widehat{\boldsymbol{q}_v}[k],\boldsymbol{s}[k]=0\end{cases} \tag{4}$$

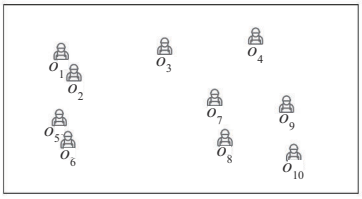
最后，得到加密陷门，如式(5)所示。

$$\boldsymbol{T}_Q=(\boldsymbol{M}_1^{-1}\widehat{\boldsymbol{q}_v}',\boldsymbol{M}_2^{-1}\widehat{\boldsymbol{q}_v}'') \tag{5}$$

4) 内积计算：EASPE.Query$(\boldsymbol{C},\boldsymbol{T}_Q)\rightarrow\boldsymbol{p}^{\mathrm{T}}\cdot\boldsymbol{q}$。通过式$\boldsymbol{C}^{\mathrm{T}}\cdot\boldsymbol{T}_Q=\widehat{\boldsymbol{p}_v}'^{\mathrm{T}}\boldsymbol{M}_1\cdot\boldsymbol{M}_1^{-1}\widehat{\boldsymbol{q}_v}'+\widehat{\boldsymbol{p}_v}''^{\mathrm{T}}\boldsymbol{M}_2\cdot\boldsymbol{M}_2^{-1}\widehat{\boldsymbol{q}_v}''=\boldsymbol{p}_v^{\mathrm{T}}\cdot\boldsymbol{q}_v=\boldsymbol{p}^{\mathrm{T}}\cdot\boldsymbol{q}$，云服务器可以获得$\boldsymbol{C}$和$\boldsymbol{T}_Q$的内积。EASPE方案引入了3个随机数和一个随机置换π，这使EASPE方案随机性更强，能够抵御选择明文攻击。

4 方案SKSTM构造

本节介绍了SKSTM方案的框架，详细阐述了



$o_1.k$	w_1, w_3
$o_2.k$	w_2, w_5
$o_3.k$	w_4, w_7, w_8, w_9
$o_4.k$	w_1, w_5, w_9
$o_5.k$	w_6

$o_6.k$	w_1, w_2, w_7, w_8
$o_7.k$	w_8, w_9
$o_8.k$	w_4, w_7, w_8, w_9
$o_9.k$	w_2, w_5, w_6
$o_{10}.k$	w_1, w_3, w_8

关键词	w_8	w_1	w_9	w_2	w_5	w_7	w_3	w_4	w_6
频次	5	4	4	3	3	3	2	2	2

BM_1 0 1 0 0 0 0 1 0 0 0　BM_2 0 0 0 1 1 0 0 0 0 0　BM_3 1 0 1 0 0 0 1 0 1 0 …

图4 一个关键词字典长度为9的位图构建示例

该方案中位置和关键词的匹配方式。

4.1　方案框架

本文方案包括密钥生成、数据加密、陷门生成和查询4个阶段。任务请求者和工人分别向密钥管理中心提交各自的信息，认证通过后完成注册。在密钥生成阶段，密钥管理中心生成加密密钥并发送给工人和任务请求者。在数据加密阶段，工人将自己的空间数据对象 o_i 编码为索引向量 $\boldsymbol{o_i.v}$，并将 $\boldsymbol{o_i.v}$ 加密为 $\boldsymbol{C}_i$，然后将 $\boldsymbol{C}_i$ 发送给云服务器。在陷门生成阶段，任务请求者将查询请求 Q 编码为向量 $\boldsymbol{Q.v}$，并将其加密为查询陷门 $\boldsymbol{T}_Q$。另外，任务请求者使用ASE算法加密任务明文，然后将查询陷门 $\boldsymbol{T}_Q$ 和任务密文提交给云服务器。在任务匹配阶段，云服务器计算加密索引 $\boldsymbol{C}_i$ 和陷门 $\boldsymbol{T}_Q$ 的内积，若内积值大于或等于0，则将任务密文 $\bar{\mathcal{M}}$ 发送给工人。工人完成任务后，将任务结果提交给云服务器。最后，云服务器将任务结果返回给任务请求者，任务请求者为查询行为支付对应的费用。

4.2　SKSTM方案

设 $\mathcal{W}=\{w_1,w_2,\cdots,w_a\}$ 是总关键词集合，$D=\{o_1,o_2,\cdots,o_n\}$ 是一个工人的空间关键词数据集。每个工人 o_i 具有唯一标识 id_i，并且可以表示为 $\{o_i.l,o_i.k\}$，其中，$o_i.l=\{o_i.x,o_i.y\}$ 表示工人 o_i 的空间地理位置，$o_i.k=\{w_{i,1},w_{i,2},\cdots,w_{i,n_i}\}\in\mathcal{W}$ 表示工人 o_i 的兴趣关键词。对数据集中的关键词集合 $\mathcal{W}$ 按照频率进行降序排序，得到包含关键词词频的关键词集合 $\mathcal{W}'$。任务查询 $Q=\{R=(Q.l,r),Q.k,\tau,\mathcal{M}\}$，其中，$R$ 表示位置范围查询，$Q.k$ 表示任务的关键词，τ 表示关键词相似性查询的阈值，$\mathcal{M}$ 是任务的明文信息。

密钥生成。密钥管理中心通过调用EASPE.KeyGen生成加密密钥sk，输入安全参数 λ，输出密钥sk。$d=32\times8+a+1$，a 是关键词字典的长度。$\mathrm{sk}=\{K,\boldsymbol{s},\boldsymbol{M}_1,\boldsymbol{M}_2,\pi,r_1,r_2,r_3,r_4,r_5,r_6\}$，其中，$K$ 是一个对称加密密钥，用于任务明文的加密和解密（本文采用AES对称加密算法），$\boldsymbol{s}$ 是一个 $(d+3)$ 维的随机二进制向量，$\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 是2个 $(d+3)\times(d+3)$ 的随机可逆矩阵，π 是一个 $\mathbb{R}^{d+3}\rightarrow\mathbb{R}^{d+3}$ 的随机置换，r_1,r_2,r_3,r_4,r_5,r_6 是6个随机数，满足 $r_1r_4+r_2r_5+r_3r_6=0$。

数据加密。具体过程如算法1所示。对于 D 中的每个工人对象 o_i，工人首先使用Geohash算法将 $o_i.l$ 映射为一个8字符的Geohash编码，并根据Base 32将Geohash编码转换为一个8维顺序向量 $\widetilde{\boldsymbol{o_i.l}}$。然后，将 $\widetilde{\boldsymbol{o_i.l}}$ 的每一维转换为32维空间向量。最后，生成(32×8)维空间向量 $\boldsymbol{o_i.lv}$，如式(6)所示。

$$\boldsymbol{o_i.lv}=\{\boldsymbol{o_i.lv}_1,\boldsymbol{o_i.lv}_2,\cdots,\boldsymbol{o_i.lv}_8\} \tag{6}$$

对于工人对象 o_i 所拥有的关键词集合 $o_i.k$，根据词频排序后的关键词集合 $\mathcal{W}'$ 生成位图 BM_i。最后生成一个 $(32\times8+a+1)$ 维的向量 $\boldsymbol{o_i.v}$，如式(7)所示。

$$\boldsymbol{o_i.v}=(\boldsymbol{o_i.lv},\mathrm{BM}_i,-1) \tag{7}$$

最后，工人使用EASPE.Enc将空间数据集 D 加密为 $\mathrm{Enc}(D)=\{C_i\ 1\leqslant i\leqslant n\}$，并将加密数据集发送给云服务器。

算法1　数据加密

输入　$D=\{o_1,o_2,\cdots,o_n\}$

输出　$\mathrm{Enc}(D)$

初始化集合 $\mathrm{Enc}(D)=\varnothing$；

1)for $1\leqslant i\leqslant n$ do

2) 将 $o_i.l$ 映射为一个8字符的Geohash编码，并根据Base 32将Geohash编码转换为一个8维顺序向量 $\widetilde{\boldsymbol{o_i.l}}$；

3)　for $1\leqslant j\leqslant 8$ do

4)$\widetilde{\boldsymbol{o_i.l}}$ 的第 j 维转换为32维空间向量 $\boldsymbol{o_i.lv}_j$；

5)　end for

6) end for

7)$\boldsymbol{o_i.v}=(\boldsymbol{o_i.lv},\mathrm{BM}_i,-1)$；

8)$\boldsymbol{C}_i=\mathrm{EASPE.Enc}(\boldsymbol{o_i.v},\mathrm{sk})$；

9)$\mathrm{Enc}(D).\mathrm{add}(\boldsymbol{C}_i)$；

return $\mathrm{Enc}(D)$

陷门生成。具体过程如算法2所示。给定查询请求 $Q=\{R=(Q.l,r),Q.k,\tau,\mathcal{M}\}$。任务请求者首先根据 r 将 $Q.l$ 映射为 t 位的Geohash编码，并将其转换为 t 维顺序向量 $\widetilde{\boldsymbol{Q.l}}$。然后，任务请求者将每个维度编码为32维查询空间向量 $\boldsymbol{Q.lv}$。如果 $t<8$，则剩余的 $(8-t)$ 个32维查询空间向量的每个维度被设置为0。最后，将 Q 转换为(32×8)维空间向量

$\boldsymbol{Q.lv}$，如式(8)所示。

$$\boldsymbol{Q.lv}=\left\{\boldsymbol{Q.lv}_1,\boldsymbol{Q.lv}_2,\cdots,\boldsymbol{Q.lv}_8\right\} \tag{8}$$

对于任务请求者Q所拥有的关键词集合$Q.k$，根据词频排序后的关键词集合$\mathcal{W}'$生成位图BM_q。然后将BM_q中的前$m-\left(\lfloor(1-\tau)m+1\rfloor\right)$位设置为0，从而得到$\varepsilon\left(\mathrm{BM}_q\right)$，其中$m=|Q.k|$。任务请求者任意选择一个大于$m$的随机整数$\alpha$，生成查询向量$\boldsymbol{Q.v}$，如式(9)所示。

$$\boldsymbol{Q.v}=\left(\alpha\boldsymbol{Q.lv},\varepsilon\left(\mathrm{BM}_q\right),\alpha\tau\right) \tag{9}$$

最后，任务请求者通过调用陷门生成算法EASPE.TrapGen将任务请求Q加密为陷门$\boldsymbol{T}_Q$，并使用AES对称加密算法加密任务明文$\mathcal{M}$。将陷门$\boldsymbol{T}_Q$和加密后的任务$\bar{\mathcal{M}}$发送给云服务器。

算法2 陷门生成

输入 $Q=\left\{R=\left(Q.l,r\right),Q.k,\tau,\mathcal{M}\right\}$

输出 $\boldsymbol{T}_Q$，$\bar{\mathcal{M}}$

1)根据r将$Q.l$映射为t位Geohash编码，并将其转换为t维顺序向量$\widetilde{\boldsymbol{Q.l}}$；

2) for 1⩽ j ⩽ 8 do

3) if $j<t$ then

4) 将第j维$\widetilde{\boldsymbol{Q.l}}$编码为32维查询空间向量$\boldsymbol{Q.lv}_j$；

5) else

6) 将$\boldsymbol{Q.lv}_j$剩余的维度元素全部设置为0；

7) end if

8) end for

9)$\boldsymbol{Q.lv}=\left\{\boldsymbol{Q.lv}_1,\boldsymbol{Q.lv}_2,\cdots,\boldsymbol{Q.lv}_8\right\}$；

10)将$Q.k$转换为位图BM_q；

11)将BM_q前$m-\left(\lfloor(1-\tau)m+1\rfloor\right)$位设置为0，得到$\varepsilon\left(\mathrm{BM}_q\right)$；

12)$\boldsymbol{Q.v}=\left(\alpha\boldsymbol{Q.lv},\varepsilon\left(\mathrm{BM}_q\right),\alpha\tau\right)$；

13)$\boldsymbol{T}_Q=\mathrm{EASPE.TrapGen}\left(\boldsymbol{Q.v},\mathrm{sk}\right)$；

14)$\bar{\mathcal{M}}=\mathrm{AES}\left(\mathcal{M},K\right)$；

return $\boldsymbol{T}_Q$，$\bar{\mathcal{M}}$

任务匹配。云服务器收到加密索引$\mathrm{Enc}(D)$和查询陷门$\boldsymbol{T}_Q$后，计算$\boldsymbol{C}_i$和$\boldsymbol{T}_Q$的内积，匹配合适的工人。然后，将任务密文$\bar{\mathcal{M}}$分发给满足查询约束条件的工人，工人执行任务。具体过程如算法3所示。工人在收到$\bar{\mathcal{M}}$后，使用对称密钥K进行解密，获得任务明文$\mathcal{M}$。

算法3 任务匹配

输入 $\mathrm{Enc}(D)=\{\boldsymbol{C}_1,\boldsymbol{C}_2,\cdots,\boldsymbol{C}_n\}_{1\leqslant i\leqslant n},\boldsymbol{T}_Q$

输出 查询结果R

1)初始化查询结果$R=\varnothing$

2)for 1⩽ i ⩽ n do

3) 计算$\boldsymbol{C}_i\circ\boldsymbol{T}_Q=\boldsymbol{o}_i.\boldsymbol{v}\circ\boldsymbol{Q.v}$；

4) if $\boldsymbol{C}_i\circ\boldsymbol{T}_Q\geqslant 0$ then

5) end if

6)end for

return R

正确性分析如下。

定理1 若索引$\boldsymbol{C}_i$与陷门$\boldsymbol{T}_Q$的内积大于或等于0，则工人o_i与任务Q匹配成功。

证明 对于每个$o_i\in D$，云服务器通过调用EASPE.Query计算$\boldsymbol{C}_i\circ\boldsymbol{T}_Q$。其中，$\boldsymbol{C}_i=\mathrm{EASPE.Enc}\left(\boldsymbol{o}_i.\boldsymbol{v},\mathrm{sk}\right)$，具体生成过程如下。首先对$\boldsymbol{o}_i.\boldsymbol{v}$引入随机数$r_1,r_2,r_3$，得到向量$\boldsymbol{o}_{iv}=\left(\boldsymbol{o}_i.\boldsymbol{v},r_1,r_2,r_3\right)$。然后通过随机置换$\pi$对$\boldsymbol{o}_{iv}$进行置换得到$\widehat{\boldsymbol{o}_{iv}}=\pi\left(\boldsymbol{o}_{iv}\right)$。最后根据随机生成的二进制向量$\boldsymbol{s}$将$\widehat{\boldsymbol{o}_{iv}}$拆分成如下2个向量

$$\begin{cases}\widehat{\boldsymbol{p}_v}'[k]=\widehat{\boldsymbol{p}_v}''[k]=\widehat{\boldsymbol{p}_v}[k],\boldsymbol{s}[k]=0\\ \widehat{\boldsymbol{p}_v}'[k]+\widehat{\boldsymbol{p}_v}''[k]=\widehat{\boldsymbol{p}_v}[k],\boldsymbol{s}[k]=1\end{cases}$$

获得加密数据$\boldsymbol{C}_i=(\boldsymbol{M}_1^{\mathrm{T}}\widehat{\boldsymbol{o}_{iv}}',\boldsymbol{M}_2^{\mathrm{T}}\widehat{\boldsymbol{o}_{iv}}'')$。

此外，任务请求者加密的查询陷门$\boldsymbol{T}_Q=\mathrm{EASPE.TrapGen}\left(\boldsymbol{Q.lv},\mathrm{sk}\right)$，具体生成过程如下：首先对$\boldsymbol{Q.lv}$引入随机数$r_4,r_5,r_6$，得到向量$\boldsymbol{Q}_v=\left(\boldsymbol{Q.lv},r_4,r_5,r_6\right)$。然后通过随机置换$\pi$对$\boldsymbol{Q.lv}$进行置换得到$\widehat{\boldsymbol{Q}_v}=\pi\left(\boldsymbol{Q}_v\right)$。根据二进制向量$\boldsymbol{s}$将$\widehat{\boldsymbol{Q}_v}$拆分成2个向量

$$\begin{cases}\widehat{\boldsymbol{Q}_v}'[k]=\widehat{\boldsymbol{Q}_v}''[k]=\widehat{\boldsymbol{Q}_v}[k],\boldsymbol{s}[k]=1\\ \widehat{\boldsymbol{Q}_v}'[k]+\widehat{\boldsymbol{Q}_v}''[k]=\widehat{\boldsymbol{Q}_v}[k],\boldsymbol{s}[k]=0\end{cases}$$

查询陷门$\boldsymbol{T}_Q=(\boldsymbol{M}_1^{-1}\widehat{\boldsymbol{Q}_v}',\boldsymbol{M}_2^{-1}\widehat{\boldsymbol{Q}_v}'')$。

索引与陷门内积为

$$\begin{aligned}&\boldsymbol{C}_i^{\mathrm{T}}\circ\boldsymbol{T}_Q=\\&(\boldsymbol{M}_1\widehat{\boldsymbol{o}_{iv}}',\boldsymbol{M}_2\widehat{\boldsymbol{o}_{iv}}'')^{\mathrm{T}}\circ(\boldsymbol{M}_1^{-1}\widehat{\boldsymbol{Q}_v}',\boldsymbol{M}_2^{-1}\widehat{\boldsymbol{Q}_v}'')=\\&\left(\widehat{\boldsymbol{o}_{iv}}'\boldsymbol{M}_1\right)\circ\left(\boldsymbol{M}_1^{-1}\widehat{\boldsymbol{Q}_v}'\right)+\left(\widehat{\boldsymbol{o}_{iv}}''\boldsymbol{M}_2\right)\circ\left(\boldsymbol{M}_2^{-1}\widehat{\boldsymbol{Q}_v}''\right)=\\&\boldsymbol{o}_i.\boldsymbol{v}\circ\boldsymbol{Q.v}+r_1r_4+r_2r_5+r_3r_6\end{aligned}$$

由于$r_1r_4 + r_2r_5 + r_3r_6 = 0$，因此有

$$\boldsymbol{C}_i^{\mathrm{T}} \circ \boldsymbol{T}_Q = \boldsymbol{o}_i.\boldsymbol{v} \circ \boldsymbol{Q}.\boldsymbol{v} =$$

$$\left(\boldsymbol{o}_i.\boldsymbol{lv}, \mathrm{BM}_i, -1\right) \circ \left(a\boldsymbol{Q}.\boldsymbol{lv}, \boldsymbol{\varepsilon}\left(\mathrm{BM}_q\right), a\tau\right) =$$

$$\alpha\left(\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t\right) + \mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right)$$

对于查询结果$\boldsymbol{C}_i^{\mathrm{T}} \circ \boldsymbol{T}_Q$，本文考虑以下4种情况。

1) 工人o_i与任务请求Q位置和关键词都匹配

o_i和Q的位置匹配，即o_i在位置查询范围内，$\boldsymbol{o}_i.\boldsymbol{l}$和$\boldsymbol{Q}.\boldsymbol{l}$对应的Geohash编码有$t$位相同，$\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t = 0$。

o_i和Q的关键词匹配，$o_i.k$和$Q.k$之间的Jaccard相似度大于或等于查询相似度阈值τ，$\mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right) \geqslant 0$。

因此，索引与陷门的内积$\boldsymbol{C}_i^{\mathrm{T}} \circ \boldsymbol{T}_Q = \alpha(\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t) + \mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right) \geqslant 0$。

2) 工人o_i与任务请求Q位置匹配，关键词不匹配

o_i和Q的位置匹配，即o_i在位置查询范围内，$o_i.l$和$Q.l$对应的Geohash编码有t位相同，$\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t = 0$。

o_i和Q的关键词不匹配，$o_i.k$和$Q.k$之间的Jaccard相似度小于查询相似度阈值τ，$\mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right) < 0$。

因此，索引与陷门的内积$\boldsymbol{C}_i^{\mathrm{T}} \circ \boldsymbol{T}_Q = \alpha\left(\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t\right) + \mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right) < 0$。

3) 工人o_i与任务请求Q位置不匹配，关键词匹配

o_i和Q的位置不匹配，即o_i不在位置查询范围内。由于$Q.l$对应Geohash编码的长度为t，o_i和Q的位置不匹配，因此$Q.lv \circ o_i.lv$一定是一个小于t的整数，即$\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t \leqslant -1$。$\alpha$是一个大于$m$的随机整数，因此$\alpha\left(\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t\right) < -m$。

o_i和Q的关键词匹配，$o_i.k$和$Q.k$之间的Jaccard相似度大于或等于查询相似度阈值τ，$\mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right) \geqslant 0$。此外，$\left|\varepsilon\left(\mathrm{BM}_q\right)\right| \leqslant m$，$\mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right) \leqslant m$。

因此，索引与陷门的内积$\boldsymbol{C}_i^{\mathrm{T}} \circ \boldsymbol{T}_Q = \alpha\left(\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t\right) + \mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right) < 0$。

4) 工人o_i与任务请求Q位置和关键词都不匹配

o_i和Q的位置不匹配，即o_i不在位置查询范围内。由于$Q.l$对应Geohash编码的长度为t，o_i和Q的位置不匹配，因此$\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv}$一定是一个小于$t$的整数，即$\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t \leqslant -1$。$\alpha$是一个大于$m$的随机整数，因此$\alpha\left(\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t\right) < -m$。

o_i和Q的关键词不匹配，$o_i.k$和$Q.k$之间的Jaccard相似度小于查询相似度阈值τ，$\mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right) < 0$。

因此，索引与陷门的内积$\boldsymbol{C}_i^{\mathrm{T}} \circ \boldsymbol{T}_Q = \alpha(\boldsymbol{Q}.\boldsymbol{lv} \circ \boldsymbol{o}_i.\boldsymbol{lv} - t) + \mathrm{BM}_i \circ \varepsilon\left(\mathrm{BM}_q\right) < 0$。

通过对以上4种情况的分析，可以得到如下结论：计算$\boldsymbol{C}_i \circ \boldsymbol{T}_Q$并判断是否大于或等于0，等价于确定工人$o_i$和任务$Q$是否匹配，若$\boldsymbol{C}_i \circ \boldsymbol{T}_Q \geqslant 0$，则工人$o_i$满足任务$Q$的查询要求，反之则不满足要求。证毕。

部署可行性分析如下。

在实际应用中，密钥管理是影响系统安全性和可行性的重要因素。本文方案依赖于密钥管理中心进行密钥的生成、分发和更新，以确保任务请求者和工人在任务匹配过程中能够在密文环境下进行安全计算。KMC作为一个受信任的第三方，主要负责认证用户身份并为其分配加密密钥。在系统初始化阶段，任务请求者和工人分别向KMC注册身份信息，KMC通过密钥生成算法生成相应的加密密钥，并通过安全信道（如SSL/TLS连接）将密钥安全地分发给工人和任务请求者。工人和任务请求者获得密钥后，可使用该密钥对任务需求和兴趣关键词进行加密，并将加密后的数据提交至云服务器进行任务匹配。

为了进一步增强系统的安全性，KMC需要支持密钥更新机制。由于长期使用的密钥可能面临密钥隐私泄露风险，因此需要设计有效的密钥更新机制。密钥更新机制较易实现，例如，任务请求者和工人可以在一定时间间隔后向KMC申请新密钥，或者当用户身份发生变更（如任务请求者撤销某个工人的访问权限）时，KMC也可以强制触发密钥更新。更新后的密钥仍然通过安全信道分发给相关用户，以确保数据的机密性不受影响。

值得注意的是，KMC 在完成密钥生成、分发和更新后，可以处于离线状态，仅在有新用户注册或密钥更新时才重新上线。云服务器在任务匹配过程中无须与 KMC 进行交互，这一设计减少了系统对中心化密钥管理的依赖，同时提升了整体的可扩展性和抗攻击能力。此外，任务请求者和工人可以本地存储自身的密钥，或者采用去中心化存储机制（如区块链或分布式身份管理）来提升密钥管理的灵活性。

5 安全性分析

本节分析了本文方案 SKSTM 在第 2 节中定义的威胁模型下的安全性。由于进行 CPA 的敌手比发起 KPA 的敌手拥有更多的信息，因此发起 CPA 的敌手拥有更强大的攻击能力。若方案在 CPA 环境下是安全的，则方案也可以抵御 KPA。因此，本文证明 SKSTM 可以抵御 IND-CPA。

5.1 数据加密

SKSTM 是安全参数 λ 上的隐私保护任务匹配方案，敌手 A 和挑战者 C 在选择明文攻击下的安全实验定义如下。

①初始化：给定安全参数 λ，敌手 A 生成 2 个相同维度的空间关键词数据集 $D_0=\left(d_{01},d_{02},\cdots,d_{0n}\right)$ 和 $D_1=\left(d_{11},d_{12},\cdots,d_{1n}\right)$，并发送给 C，其中 d_{ij} 是一个空间文本数据，$i\in\{0,1\}$，$j\in\{1,n\}$。

②密钥生成：挑战者 C 运行 EASPE.KeyGen 来生成密钥，密钥对 A 保密。

③阶段 1：A 将 d_{ij} 提交给 C，$i\in\{0,1\}$，$j\in\{1,n\}$。然后，C 通过运行 EASPE.Enc 返回密文 $\boldsymbol{C}_{ij}$。

④挑战：C 选择 $b\in\{0,1\}$，通过 EASPE.Enc 计算出 d_{bj} 的密文 $\boldsymbol{C}_{bj}$。然后，C 将 $\boldsymbol{C}_{bj}$ 返回给 A。

⑤阶段 2：A 选择一些消息，并将它们提交给 C。

⑥猜测：敌手 A 猜测 b 的值为 b'。

定义 2 如果对于任何概率多项式时间敌手 A，至多有一个几乎可忽略的优势 $\mathrm{negl}(\lambda)$，使

$$\mathrm{Adv}_{\mathrm{SKSTM},A}^{\mathrm{IND\text{-}CPA}}\left(1^{\lambda}\right)=\left|\Pr\left(b'=b\right)-\frac{1}{2}\right|\leqslant\mathrm{negl}(\lambda)$$

其中，$\mathrm{negl}(\lambda)$ 表示一个可忽略的函数。则 SKSTM 方案在数据加密阶段可以达到 IND-CPA 安全。

定理 2 在 IND-CPA 模型下，SKSTM 可以达到数据安全。

证明 令 A 为一个概率多项式时间敌手，定义 $\mathcal{E}(n)=\Pr\left[\mathrm{Adv}_{\mathrm{SKSTM},A}^{\mathrm{IND\text{-}CPA}}\left(1^{\lambda}\right)=1\right]$。下面介绍模拟实验的步骤。①敌手 A 随机选择 2 个空间数据消息 D_0，D_1。②A 选择一个随机 $b\leftarrow\{0,1\}$，并且挑战密文 $\boldsymbol{C}_{0i}$ 和 $\boldsymbol{C}_{1i}$ 给 A。③敌手 A 可继续问询解密预言机，尽管它不能对解密预言机问询它的挑战密文。④A 输出一个比特 b'。如果 $b'=b$，该实验的输出被定义为 1，否则为 0。在数据模拟实验执行中，基于定义 2，假设 D_i 中的一个消息 $d_{ij}=\left(d_{ij}.l,d_{ij}.k\right)$，根据数据加密 EASPE.Enc 的过程，$d_{ij}$ 被扩展为 $\boldsymbol{d}_{ij}.\boldsymbol{v}=\left(\boldsymbol{d}_{ij}.\boldsymbol{lv},\mathrm{BM}_{ij},-1,r_1,r_2,r_3\right)$，其中 r_1,r_2,r_3 是 3 个随机数。然后通过随机置换生成 $\widehat{\boldsymbol{d}_{ijv}}=\pi\left(\boldsymbol{d}_{ijv}\right)$，最后生成加密数据 $\boldsymbol{C}_{ij}=(\boldsymbol{M}_1^{\mathrm{T}}\widehat{\boldsymbol{d}_{ijv}}',\boldsymbol{M}_2^{\mathrm{T}}\widehat{\boldsymbol{d}_{ijv}}'')$。虽然 A 可以对加密预言机问询，并得到对应的密文 $\boldsymbol{C}_{ij}$。然而 $\widehat{\boldsymbol{d}_{ijv}}$ 是由挑战者 C 确定的随机向量，π 是一个随机置换，r_1,r_2,r_3 是 3 个随机数，$\boldsymbol{M}_1$ 和 $\boldsymbol{M}_2$ 是 2 个随机可逆矩阵，因此密文 $\boldsymbol{C}_{ij}$ 对于敌手 A 来说是随机的。换言之，给定一个由敌手选定的加密数据，A 无法区分对应的明文数据。存在一个可忽略函数 $\mathrm{negl}(\lambda)$，使 $\Pr[b\neq b']=\Pr'[b\neq b']$ 满足

$$\Pr'[b=b'\wedge b\neq b']=\Pr'[b=b'|b\neq b']\cdot$$

$$\Pr'[b\neq b']=\Pr[b=b'\wedge b\neq b']=$$

$$\left|\Pr(b'=b)-\frac{1}{2}\right|\leqslant\mathrm{negl}(\lambda)$$

$$\mathrm{Adv}_{\mathrm{SKSTM},A}^{\mathrm{IND\text{-}CPA}}\left(1^{\lambda}\right)=\left|\Pr(b'=b)-\frac{1}{2}\right|\leqslant\mathrm{negl}(\lambda)$$

因此，SKSTM 在 IND-CPA 安全模型下可以达到数据安全。证毕。

5.2 陷门生成

模拟选择明文攻击模型下敌手 A 和挑战者 C 之间的安全实验定义如下。

① 初始化：给定安全参数 λ，敌手 A 生成 2 个相同维度的空间关键词查询请求 $Q_0=\left(R_0=\left(Q_0.l,r_0\right),Q_0.k,\tau_0,\boldsymbol{M}_0\right)$ 和 $Q_1=\left(R_1=\left(Q_1.l,r_1\right),Q_1.k,\tau_1,\boldsymbol{M}_1\right)$，并发送给 C，其中 Q_i 是一个空间文本查询请求，$i\in\{0,1\}$。

②密钥生成：挑战者 C 运行 EASPE.KeyGen 来生成密钥，密钥对 A 保密。

③阶段 1：A 将 Q_i 提交给 C，$i\in\{0,1\}$。然后，

C通过运行EASPE.TrapGen返回陷门$\boldsymbol{T}_{Qi}$。

④挑战：C选择$b\in\{0,1\}$，通过EASPE.TrapGen计算出Q_i的陷门$\boldsymbol{T}_{Qi}$。然后，C将$\boldsymbol{T}_{Qi}$返回给A。

⑤阶段2：A选择一些消息并将它们提交给C。

⑥猜测：敌手A猜测b的值为b'。

定义3　如果对于任何概率多项式时间敌手A，至多有一个几乎可忽略的优势$\mathrm{negl}(\lambda)$，使

$$\mathrm{Adv}_{\mathrm{SKSTM},A}^{\mathrm{IND\text{-}CPA}}\left(1^{\lambda}\right)=\left|\Pr\left(b'=b\right)-\frac{1}{2}\right|\leqslant\mathrm{negl}(\lambda)$$

其中，$\mathrm{negl}(\lambda)$表示一个可忽略的函数。则SKSTM在陷门生成阶段实现了IND-CPA安全性。

定理3　在CPA安全模型下，SKSTM在陷门生成阶段可以达到IND-CPA安全。

证明　在陷门生成阶段，基于定义3，注意到任务请求者经过了随机扩充、置换，以及向量分裂和矩阵随机扰动，得到加密后的查询陷门$\boldsymbol{T}_{Qi}$。具体而言，根据陷门生成算法EASPE.TrapGen，Q_i首先被随机扩展$\boldsymbol{Q}_i.\boldsymbol{v}=\left(\alpha\boldsymbol{Q}_i.\boldsymbol{lv},\varepsilon\left(\mathrm{BM}_{qi}\right),\alpha\tau_i,r_4,r_5,r_6\right)$，其中$r_4,r_5,r_6$是3个随机数。然后，通过随机置换生成$\widehat{\boldsymbol{Q}_{vi}}=\pi\left(\boldsymbol{Q}_i.\boldsymbol{v}\right)$。最后，生成查询陷门$\boldsymbol{T}_{Qi}=(\boldsymbol{M}_1^{-1}\widehat{\boldsymbol{Q}_{vi}}',\boldsymbol{M}_2^{-1}\widehat{\boldsymbol{Q}_{vi}}'')$。然而$\widehat{\boldsymbol{Q}_{vi}}$是由挑战者$C$确定的随机向量，$\pi$是一个随机置换，$r_4$、$r_5$、$r_6$、$\alpha$是随机数，$\boldsymbol{M}_1$和$\boldsymbol{M}_2$是2个随机可逆矩阵，因此，陷门$\boldsymbol{T}_{Qi}$对于敌手$A$来说是随机的。存在一个可忽略函数$\mathrm{negl}(\lambda)$，使$\Pr[b\neq b']=\Pr'[b\neq b']$满足

$$\mathrm{Adv}_{\mathrm{SKSTM},A}^{\mathrm{IND\text{-}CPA}}\left(1^{\lambda}\right)=\left|\Pr\left(b'=b\right)-\frac{1}{2}\right|\leqslant\mathrm{negl}(\lambda)$$

因此，SKSTM方案的陷门生成阶段可以安全抵御IND-CPA。证毕。

5.3　任务匹配

定理4　在CPA安全模型下，SKSTM在任务匹配阶段可实现陷门不可链接性。

证明　在任务匹配中，陷门的随机性可由随机数、随机置换、随机分裂，以及矩阵随机扰动实现。在陷门生成算法中，由于r_4、r_5、r_6是3个随机数，π是随机置换，$\boldsymbol{M}_1$，$\boldsymbol{M}_2$是2个随机可逆矩阵，$\boldsymbol{s}$是随机二进制向量，云服务器收到查询陷门后，进行任务匹配，返回与查询陷门匹配的索引。换言之，云服务器计算$\boldsymbol{C}_i^{\mathrm{T}}\circ\boldsymbol{T}_Q=\boldsymbol{o}_i.\boldsymbol{v}\circ\boldsymbol{Q}.\boldsymbol{v}+r_1r_4+r_2r_5+r_3r_6=\alpha\left(\boldsymbol{Q}.\boldsymbol{lv}\circ\boldsymbol{o}_i.\boldsymbol{lv}-t\right)+\mathrm{BM}_i\circ\varepsilon\left(\mathrm{BM}_q\right)$，由于$r_1,r_2,r_3,r_4,r_5,r_6$均为未知随机数，因此即使云服务器能够判断$\boldsymbol{C}_i^{\mathrm{T}}\circ\boldsymbol{T}_Q$是否大于或等于0，也无法获取$(\boldsymbol{o}_i.\boldsymbol{lv}\circ\boldsymbol{Q}.\boldsymbol{lv})-t=0$或是$\mathrm{BM}_i\circ\varepsilon\left(\mathrm{BM}_q\right)\geqslant 0$的关系。给定2个查询陷门$\boldsymbol{T}_{Q1}$和$\boldsymbol{T}_{Q2}$，本文有$\Pr[\boldsymbol{T}_{Q1}=\boldsymbol{T}_{Q2}]=\frac{1}{2^{260+a}}$，其中，$260+a$表示陷门向量的维度。此外，当$\boldsymbol{s}[j]=0$时，$\boldsymbol{T}_{Q1}[j]$和$\boldsymbol{T}_{Q2}[j]$随机分裂成2个值。假设对于任意$j$，且二进制向量中$\boldsymbol{s}[j]=0$，$\Pr\left[\left\{\boldsymbol{T}'_{Q1}[j],\boldsymbol{T}''_{Q1}[j]\right\}=\left\{\boldsymbol{T}'_{Q2}[j],\boldsymbol{T}''_{Q2}[j]\right\}\right]=\gamma$，本文有$\Pr\left[\left\{\boldsymbol{T}'_{Q1},\boldsymbol{T}''_{Q1}\right\}=\left\{\boldsymbol{T}'_{Q2},\boldsymbol{T}''_{Q2}\right\}\right]=\gamma^{h_0}$，其中，$h_0$表示二进制分裂向量$s$中0的个数。由于$\gamma\to 0$且$h_0\to 260+a$，本文有$\Pr\left[\boldsymbol{T}_{Q1}=\boldsymbol{T}_{Q2}\right]\to 0$。因此，云服务器无法推断2个查询陷门之间的关系，陷门不可链接性得证。证毕。

6　讨论

在隐私保护任务匹配场景中，恶意云服务器与恶意工人可能会合谋，窃取用户的隐私数据，造成安全威胁。为了防御合谋攻击，本文从以下几个方面进行讨论。①基于安全硬件的可信执行环境（TEE）：使用Intel SGX等TEE技术，将加密和任务匹配计算封装在受保护的可信执行环境中，即使云服务器与工人合谋，也无法泄露隐私数据。②基于代理重加密的抗合谋方法：通过结合代理重加密和时间陷门的访问控制策略，重加密密钥受时间限制，实现安全的任务匹配与访问权限控制，抵御恶意云服务器与工人的合谋攻击。③基于密钥管理的抗合谋机制：采用密钥分割技术，将密钥分解成多个部分，并分别由不同的可信方管理，防止云服务器与恶意工人的合谋。④白盒可追踪的属性基加密：通过在密钥生成过程中引入用户的可追踪特性，一旦发生合谋攻击，就可以追溯到泄露密钥的恶意工人，并撤销其权限。⑤基于零知识证明的可验证计算：通过零知识证明验证工人是否具备访问权限，而不直接暴露工人的属性信息，从而降低恶意工人合谋的可能性，同时结合零知识证明，验证云服务器返回的任务结果是否正确，防止云服务器合谋，恶意篡改任务结果。

7 实验评估

本节对本文提出的SKSTM方案的数据加密、陷门生成和查询进行理论分析和性能评估。SKSTM是用Python 3.12实现，并在Intel(R)Core (TM) i5-12400处理器和64位Windows 11操作系统的服务器上进行实验。本文实验基于3种真实场景下的多源地理社交数据集Yelp[32]、Gowalla [33]和Foursquare [34]。Gowalla、Foursquare和Yelp是3个广泛用于研究用户移动模式、社交网络关系和兴趣点推荐的地理位置数据集。本文提取这些数据集中的位置信息作为空间数据，提取类别属性作为关键词集，对SKSTM方案进行了理论分析和性能评估，并与现有的空间关键词方案PPSK[11]、PPSK+[11]、PSDQ[26]、PSDQ+[26]进行对比。需特别说明的是，文献[11]采用Jaccard相似度作为关键词集合的相似性度量机制，适用于空间关键词模糊查询场景；文献[26]采用空间关键词等值查询范式，与本文方案采用的加密原语相似，但不支持相似性计算。为了验证SKSTM在众包场景下匹配的效率，本文与现有的隐私保护众包任务匹配方案POTA[16]、VP^2-Match[17]在匹配时间上进行对比。实验结果表明，SKSTM任务匹配时间远低于POTA，约是VP^2-Match的40%，验证了基于空间关键词相似性匹配方法在众包环境下任务匹配的效率优势。

7.1 理论分析

本节从数据加密、陷门生成和任务匹配3个算法的计算和通信复杂度角度，对SKSTM、PSDQ[26]、PSDQ+[26]、PPSK[11]、PPSK+[11]的计算和通信复杂度进行了分析。表3和表4分别给出了5种方案的计算复杂度和通信复杂度对比。

SKSTM通过EASPE算法加密，加密密钥sk = $\{\boldsymbol{s},\boldsymbol{M}_1,\boldsymbol{M}_2,\pi,r_1,r_2,r_3,r_4,r_5,r_6\}$，其中$K$是一个对称加密密钥用于任务明文的加密和解密，$\boldsymbol{s}$是一个$(d+3)$维的随机二进制向量，$d=32\times 8+a+1$，$\boldsymbol{M}_1,\boldsymbol{M}_2$是2个$(d+3)\times(d+3)$的随机可逆矩阵，$\pi$是一个$\mathbb{R}^{d+3}\to\mathbb{R}^{d+3}$的随机置换，$a$是关键词字典的长度。假设每个空间关键词密文的大小为$|X|$，任务明文的大小为$|Y|$，AES算法加密轮数为r，任务密文大小为$|Z|$。在数据加密阶段，工人需要加密位置和关键词信息，假设众包环境共有n个工人，则计算复杂度为$O(2n(260+a)^2)$。工人仅上传自身的加密位置和兴趣关键词，云平台通信复杂度只有单个密文数据，因此在数据加密阶段的通信复杂度为$|X|$。陷门生成的计算复杂度为$O\left(2(260+a)^2+r\cdot\left\lceil\frac{Y}{128}\right\rceil\right)$，通信复杂度为$|X|+|Z|$。任务匹配阶段的计算复杂度为$O\big(2(260+a)n\big)$。

PSDQ同样采用EASPE算法进行数据加密，其计算和通信复杂度与SKSTM大致相同。

PSDQ+通过构造基于GR-tree的树形索引结构，在原PSDQ方案基础上进一步提高了查询效率。以一个9层二叉树为例，该索引结构额外产生了510个内部节点，因而在数据加密阶段，数据所有者需对整个数据集和新增内部节点进行加密，其计算复杂度为$O(2(n+510)(260+a)^2)$，其中，n表示数据对象的数量，a为关键词集合的大小。数据加密的通信复杂度为$(n+510)|X|$。在陷门生成阶段，其计算复杂度为$O(2(260+a)^2)$且通信复杂度为$|X|$。在任务匹配阶段，由于利用树索引实现了次线性搜索，任务匹配的计算复杂度为$O(2(260+a)\log n)$。PSDQ+在保持与原方案相同安全性的前提下，通过引入GR-tree结构和剪枝策略，实现了查询效率从

表3 5种方案的计算复杂度对比

方案	数据加密	陷门生成	任务匹配
SKSTM	$O(2n(260+a)^2)$	$O\left(2(260+a)^2+r\cdot\left\lceil\frac{Y}{128}\right\rceil\right)$	$O\big(2(260+a)n\big)$
PSDQ[26]	$O(2n(260+a)^2)$	$O(2(260+a)^2)$	$O(2(260+a)n)$
PSDQ+[26]	$O(2(n+510)(260+a)^2)$	$O(2(260+a)^2)$	$O(2(260+a)\log n)$
PPSK[11]	$O(n\cdot h)$	$O(t\cdot h)$	$O(C_{\text{secure}}(l_{\max})+O(n\cdot h))$
PPSK+[11]	$O(n\cdot h+\xi\cdot k^2)$	$O(t+\xi\cdot k)$	$O(B\cdot\xi\cdot k\cdot\log n+a\cdot(h+l))$

线性搜索到次线性搜索的显著提升，但在数据加密阶段增加了部分额外的计算和存储开销。

表 4　　5 种方案的通信复杂度对比

方案	数据加密	陷门生成
SKSTM	$\|X\|$	$\|X\|+\|Z\|$
PSDQ[26]	$n\|X\|$	$\|X\|$
PSDQ+[26]	$(n+510)\|X\|$	$\|X\|$
PPSK[11]	$(n\cdot h)\|X\|$	$\|X\|$
PPSK+[11]	$(n\cdot h+\xi\cdot k^2)\|X\|$	$\|X\|$

在 PPSK 方案中，数据所有者首先将每个数据对象的空间位置信息和关键词集合分别映射到布隆过滤器中，其中布隆过滤器的长度 $h=\frac{gi}{k}$，其中，g 表示使用的哈希函数数量，i 表示映射到布隆过滤器中的元素个数，k 表示向量分桶后子向量的长度，随后使用 FHE 对布隆过滤器中每一位进行加密，使每个数据对象需要执行 $O(h)$ 次加密操作，从而数据加密阶段的计算复杂度为 $O(n\cdot h)$，通信复杂度则为 $(n\cdot h)|X|$。在陷门生成阶段，数据用户根据查询请求中涉及的空间范围和关键词集合构造加密查询令牌，其计算复杂度为 $O(t\cdot h)$（t 表示查询时生成的布隆过滤器数量），通信复杂度为常数级。在任务匹配阶段，云服务器利用 SSMT 技术对每个加密数据对象与查询陷门进行内积计算，其计算复杂度为 $O(n\cdot h)$，并结合安全加法和乱码电路（计算复杂度记为 $O(C_{\text{secure}}(l_{\max}))$，$l_{\max}$ 为安全电路的比特位宽）实现对关键词相似度的安全判断，因此整个任务匹配过程的总计算复杂度为 $O(C_{\text{secure}}(l_{\max})+O(n\cdot h))$。总体而言，PPSK 方案通过将空间和文本信息分别转换为固定维度的加密向量，在保证数据、查询请求和查询结果隐私的同时，其计算和通信复杂度主要受数据量 n、布隆过滤器长度 h 以及安全电路参数 $l_{\max}$ 影响，从而在理论上实现了高效且安全的隐私保护查询。

在 PPSK+方案中，数据所有者首先构建 FR-tree 索引以优化查询效率。FR-tree 基于 R*-树结构，将空间数据划分为多个最小边界矩形（MBR），并为每个非叶子节点维护关键词频率位图，通过分层过滤机制减少查询范围。数据加密阶段需将 FR-tree 的非叶子节点通过改进的谓词加密技术处理：每个 MBR 的向量维度为 $2T+a+\rho$，T 为空间维度上限，a 为关键词字典大小，ρ 为虚值数量。通过向量分桶技术将其分割为 $\xi=2T+a+\frac{\rho}{k}$ 个长度为 k 的子向量，并引入随机矩阵加密。数据加密总计算复杂度为 $O(n\cdot h+\xi\cdot k^2)$，通信复杂度为 $(n\cdot h+\xi\cdot k)|X|$。在陷门生成阶段，任务匹配用户需为 FR-tree 的非叶子节点构造加密查询向量，通过分桶技术将高维向量分割为 ξ 个子块，计算复杂度为 $O(t+\xi\cdot k)$，通信复杂度为常数级。任务匹配阶段云服务器通过计算加密 MBR 与查询向量的内积，快速过滤不满足条件的子树，仅需遍历 $O(\log n)$ 层非叶子节点，最终在 m 个候选叶子节点中执行 PPSK 的线性搜索。假设 FR-tree 的分支层数为 B，因此，总计算复杂度为 $O(B\cdot\xi\cdot k\cdot\log n+a\cdot(h+l))$。

7.2　性能评估

本节对 SKSTM 的数据加密、陷门生成和任务匹配过程进行了性能评估，并与 PPSK[11]、PPSK+[11]、PSDQ[26]、PSDQ+[26]进行时间开销对比，如表 5 所示。此外，为了验证 SKSTM 在众包任务匹配场景下的高效性，本文与现有的隐私保护众包任务匹配方案 POTA[16]、VP^2-Match[17]等在任务匹配时间上进行对比。

1) 数据加密

对于 SKSTM，影响计算成本的因素是数据集中数据规模大小 n（即数据集中工人的数量）和关键词字典大小（即 a）来确定。设 $a=200$，通过设置 n 为 2 000 到 20 000 测试 5 种方案的时间开销，实验结果如图 5(a)所示。设 n=12 000，通过设置 a 为 100 到 300 测试 5 种方案的时间开销，实验结果如图 5(b)所示。

实验结果表明，SKSTM 在数据加密时间开销上和 PSDQ 接近。由于 PSDQ+ 需要加密索引树的每个节点，因此加密过程计算开销相较于 PSDQ 和 SKSTM 略高。由于 PPSK 和 PPSK+使用同态加密，计算开销较高，因此加密时间开销高于 SKSTM。

2) 陷门生成

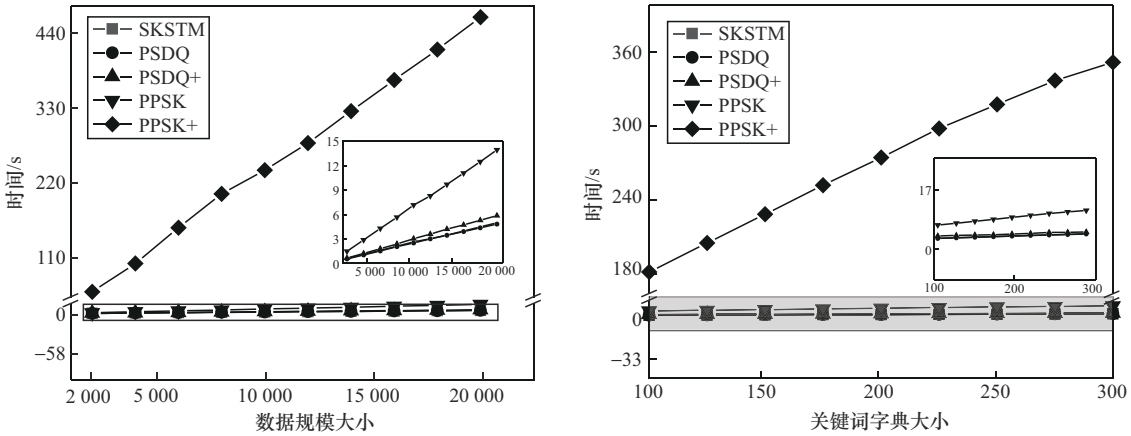
陷门生成是对任务请求者的任务需求进行一次性加密，同时使用 AES 算法对任务明文进行加密。由于 PSDQ 和 PSDQ+的陷门生成阶段相同，因此仅

表 5　　方案时间开销对比

阶段	变量		SKSTM/ms	PSDQ[24]/ms	PSDQ+[24]/ms	PPSK[9]/ms	PPSK+[9]/ms
数据加密	a=200	n=4 000	1 016	957	1 142	2 815	91 000
		n=8 000	2 038	1 983	2 315	5 620	196 000
		n=12 000	2 960	2 950	3 533	8 262	273 000
	n=12 000	a=100	2 306	2 282	2 987	6 084	178 000
		a=200	2 960	2 950	3 533	8 262	273 000
		a=300	3 604	3 560	4 110	10 240	352 000
陷门生成	a=200	m=4	11.1	8.5	8.5	182	7.9
		m=6	11.3	8.7	8.7	184	8.2
		m=8	10.9	8.4	8.4	188	8.3
	m=5	a=100	11.2	8.6	8.6	181	8.0
		a=200	11.0	8.7	8.7	182	8.1
		a=300	11.4	8.9	8.9	183	8.2
任务匹配	a=200	n=4 000	14.2	162	38.5	2 182 000	3 152
		n=8 000	27.3	348	82.4	4 390 000	6 294
		n=12 000	42.3	482	123.2	6 385 000	9 430
	n=12 000	a=100	40.3	480	120.6	6 194 000	9 312
		a=200	42.3	482	123.2	6 385 000	9 430
		a=300	**44.1**	483	124.8	6 476 000	9 552

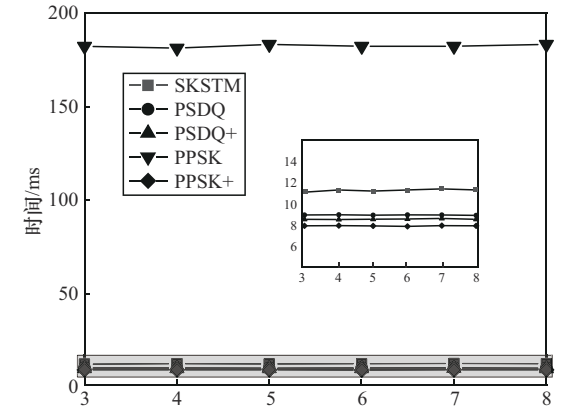
使用 PSDQ 作为对比。设 $a = 200$，通过改变查询关键词 Q_k 数目（即 m），测试 5 种方案的陷门生成时间开销，实验结果如图 6(a)所示。设 $m = 5$，通过改变 a 测试 5 种方案的陷门生成时间开销，实验结果如图 6(b)所示。

实验结果表明，SKSTM 在陷门生成计算开销上略高于 PSDQ 和 PPSK+，由于 SKSTM 在陷门生成过程中需要对关键词进行 Jaccard 相似度计算的处理，因此计算开销上相较于 PSDQ 和 PPSK 略高。PPSK 需要根据树的结构对查询陷门进行额外的处理，这一过程中的计算开销比 SKSTM 更高，因此时间开销更大。

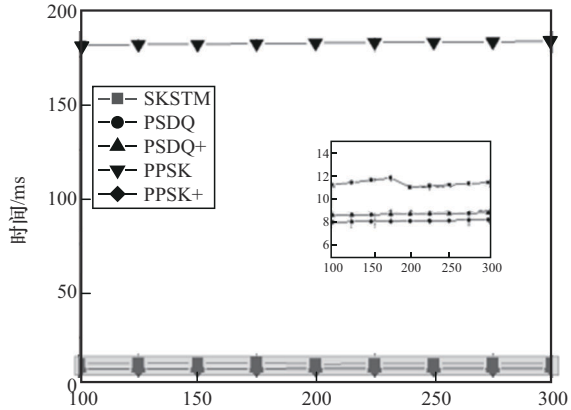


(a) 加密时间与数据规模大小关系　　(b) 加密时间与关键词字典大小关系

图 5　数据加密时间开销



(a) 陷门生成时间与关键词数目关系

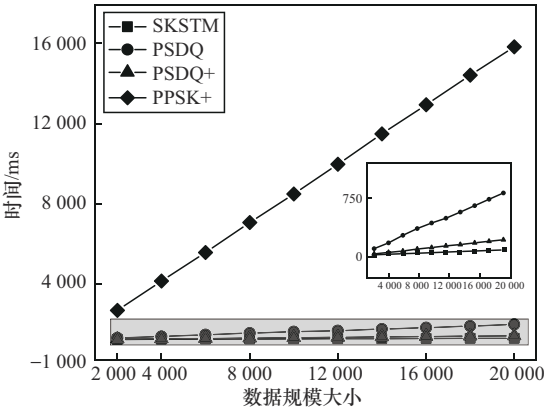


(b) 陷门生成时间与关键词字典大小关系

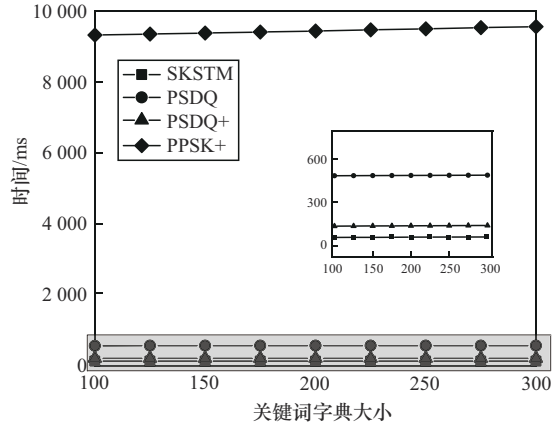
图6　陷门生成时间开销

3) 任务匹配

在任务匹配阶段，云服务器需要计算每个加密索引和陷门的乘积。因此，影响计算成本的因素是数据集中数据规模大小（即数据集中工人的数量）和关键词字典大小。由于文献[11]中的基础方案 PPSK 任务匹配时间过高（超过 1 000 s），因此在图 7 中不进行比较。设 a =200，通过将 n 设置为 2 000 到 20 000，测试这 5 种方案的任务匹配时间开销，实验结果如图 7(a)所示。设 n= 12 000，通过将 a 设置为 100 到 300，测试这 5 种方案的任务匹配时间开销，实验结果如图 7(b)所示。此外，由于文献[11]和文献[26]都是研究空间关键字查询的工作，没有应用在任务匹配场景下，因此在任务匹配时间开销实验中增加了与隐私保护任务匹配方案 POTA[16]、VP^2-Match[17]的对比。POTA 通过最小成本流模型优化任务分配，在保证数据隐私的同时，最小化工人的总移动距离，无须添加额外噪声。VP^2-Match[17]是一种基于区块链的众包任务匹配方案，支持隐私保护的多属性个性化任务匹配，并通过累加器实现任务匹配的可验证性和公平性。实验中设 a =200，POTA 方案的隐私预算为 0.1，通过将工人数量 n 设置为 200 到 1 000，测试这些方案在实际众包任务匹配场景下的时间开销，实验结果如表 6 所示。



(a) 任务匹配时间与数据规模大小关系



(b) 任务匹配时间与关键词字典大小关系

图7　任务匹配时间开销

表6　　众包任务匹配时间对比

方案	时间/ms				
	n=200	n=400	n=600	n=800	n=1 000
POTA[16]	2 716	5 310	7 904	10 498	13 092
VP^2-Match[17]	1.98	3.5	5.2	6.9	8.4
PSDQ[26]	8.9	17.2	26.4	35.2	48.7
PSDQ+[26]	2.3	4.2	5.0	9.3	11.6
PPSK[11]	112 000	218 000	340 000	442 000	524 000
PPSK+[11]	269	530	810	1 005	1 348
SKSTM	**0.72**	**1.43**	**2.15**	**2.87**	**3.58**

实验结果表明，SKSTM在任务匹配时间开销上接近PSDQ+，低于PSDQ，远低于PPSK+。由于SKSTM在任务匹配时只需要进行一次内积计算，可以同时判断位置和关键词是否满足任务匹配条件，因此在任务匹配时间开销上和文献[26]中的改进方案PSDQ+接近，低于基础方案PSDQ。PPSK+需要进行大量的同态加密计算，因此时间开销远高于SKSTM。PPSK方案由于匹配时间过长，不适用于众包任务匹配。同时SKSTM任务匹配时间远低于POTA[16]，约是VP^2-Match[17]的40%。原因在于POTA需要进行路径规划，VP^2-Match需要进行区块链上的额外操作，而SKSTM只需要进行一次内积计算。因此SKSTM在众包任务场景下任务匹配具有显著的效率优势。

4) 通信开销

图8给出了上述方案的通信开销实验对比。图8(a)设a=200，通过将n设置为2 000到10 000，测试这5种方案的数据加密通信开销。图8(b)设a=200，通过将a设置为100到300，测试这5种方案的陷门生成通信开销。众包环境下数据加密通信开销较低的原因是工人仅上传自身的加密位置和兴趣关键字，云平台通信开销只有单个密文数据，即密文大小$|X|$。然而其他对比方案在数据加密过程中，数据拥有者与云平台的通信开销取决于数据量的大小。因此本文SKSTM方案的数据加密通信开销是常数级，其他对比方案的数据加密通信开销的增长趋势是线性的。

8 结束语

针对密文环境下任务匹配效率低，难以支持空间关键词相似性查询问题，本文采用Geohash算法和BM位图对位置和关键词集进行编码，将空间关键词相似性查询转化成对应的向量内积计算，并使用矩阵加密方法对向量进行加密，提出SKSTM方案，实现了安全高效的任务匹配。本文给出了形式化的安全性分析，证明了SKSTM方案可以抵御选择明文攻击，并在真实数据集上进行了大量的实验，证明了SKSTM方案在加密算法和任务匹配算法的高效性。

参考文献：

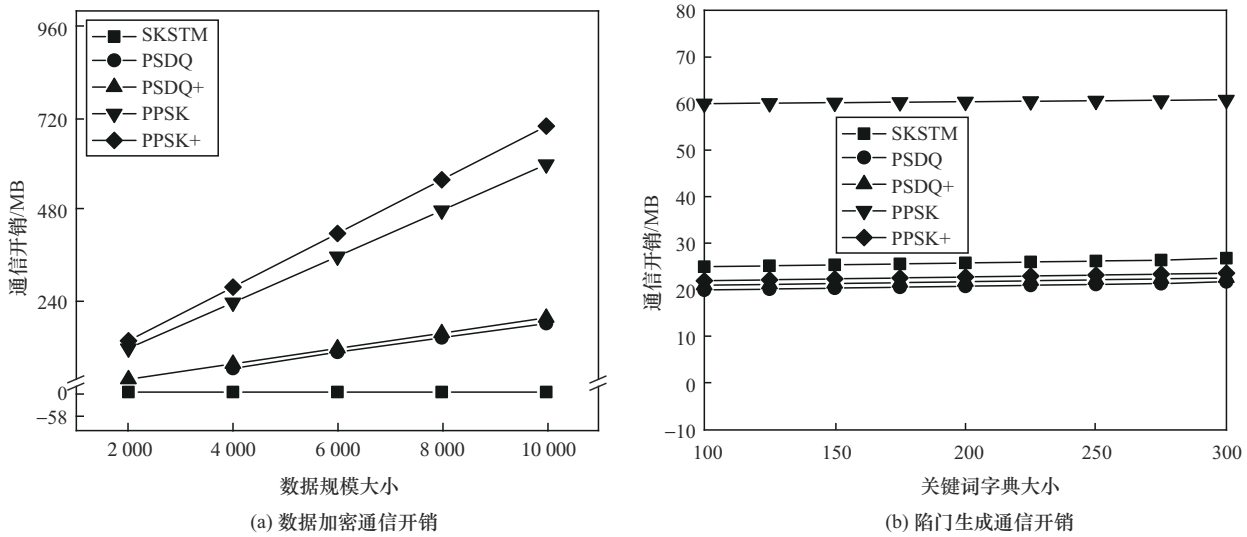
[1] YE Z K, WANG X Y, LIU Z S, et al. OBIR-tree: an efficient oblivious index for spatial keyword queries on secure enclaves[J]. Proceedings of the ACM on Management of Data, 2025, 3(1): 1-24.

[2] WANG X Y, MA J F, LIU X M, et al. Forward/backward and content private DSSE for spatial keyword queries[J]. IEEE Transactions on Dependable and Secure Computing, 2022, 20(4): 3358-3370.

[3] HOWE J. The rise of crowdsourcing[J]. Wired magazine, 2006, 14(6): 176-183.

[4] FU Z J, WANG Y, SUN X M, et al. Semantic and secure search over encrypted outsourcing cloud based on BERT[J]. Frontiers of Computer Science, 2021, 16(2): 162802.

[5] SONG F Y, LIANG J W, ZHANG C, et al. Achieving efficient and privacy-preserving location-based task recommendation in spatial crowdsourcing[J]. IEEE Transactions on Dependable and Secure Computing, 2024, 21(4): 4006-4023.



(a) 数据加密通信开销　　(b) 陷门生成通信开销

图8 通信开销

[6] WANG X Y, MA J F, LI F, et al. Enabling efficient spatial keyword queries on encrypted data with strong security guarantees[J]. IEEE Transactions on Information Forensics and Security, 2021, 16: 4909-4923.

[7] WANG X Y, MA J F, LIU X M, et al. Search me in the dark: privacy-preserving Boolean range query over encrypted spatial data[C]//Proceedings of the IEEE INFOCOM 2020 - IEEE Conference on Computer Communications. Piscataway: IEEE Press, 2020: 2253-2262.

[8] YANG Y T, MIAO Y B, CHOO K R, et al. Lightweight privacy-preserving spatial keyword query over encrypted cloud data[C]//Proceedings of the 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS). Piscataway: IEEE Press, 2022: 392-402.

[9] SONG F Y, QIN Z, LIU D X, et al. Privacy-preserving task matching with threshold similarity search via vehicular crowdsourcing[J]. IEEE Transactions on Vehicular Technology, 2021, 70(7): 7161-7175.

[10] SONG F Y, QIN Z, XUE L, et al. Privacy-preserving keyword similarity search over encrypted spatial data in cloud computing[J]. IEEE Internet of Things Journal, 2022, 9(8): 6184-6198.

[11] ZHANG S N, RAY S, LU R X, et al. Efficient and privacy-preserving spatial keyword similarity query over encrypted data[J]. IEEE Transactions on Dependable and Secure Computing, 2023, 20(5): 3770-3786.

[12] CHEN Y J, LI B C, ZHANG Q. Incentivizing crowdsourcing systems with network effects[C]//Proceedings of the IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications. Piscataway: IEEE Press, 2016: 1-9.

[13] HUANG K, LIU X M, FU S J, et al. A lightweight privacy-preserving CNN feature extraction framework for mobile sensing[J]. IEEE Transactions on Dependable and Secure Computing, 2021, 18(3): 1441-1455.

[14] LIN Y M, JIANG Y J, LI Y, et al. Privacy-preserving batch-based task assignment over spatial crowdsourcing platforms[J]. Computer Networks, 2024, 241: 110196.

[15] 王府鑫, 王宁, 曾奇雄. 基于工人长短期时空偏好的众包任务分配[J]. 软件学报, 2024, 35(10): 4710-4728.
WANG F X, WANG N, ZENG Q X. Long-and short-term spatio-temporal preference-aware task assignment in crowdsourcing[J]. Journal of Software, 2024, 35(10): 4710-4728.

[16] ZHANG C, LUO X Q, LIANG J W, et al. POTA: privacy-preserving online multi-task assignment with path planning[J]. IEEE Transactions on Mobile Computing, 2024, 23(5): 5999-6011.

[17] WU H Q, DÜDDER B, JIANG S R, et al. VP^2-match: verifiable privacy-aware and personalized crowdsourcing task matching via blockchain[J]. IEEE Transactions on Mobile Computing, 2024, 23(10): 9913-9930.

[18] MA Y, GAO X F, BHATTI S S, et al. Clustering based priority queue algorithm for spatial task assignment in crowdsourcing[J]. IEEE Transactions on Services Computing, 2024, 17(2): 452-465.

[19] WU Y M, TANG S H, ZHAO B W, et al. BPTM: blockchain-based privacy-preserving task matching in crowdsourcing[J]. IEEE Access, 2019, 7: 45605-45617.

[20] SHU J G, JIA X H, YANG K, et al. Privacy-preserving task recommendation services for crowdsourcing[J]. IEEE Transactions on Services Computing, 2021, 14(1): 235-247.

[21] WANG X Y, MA J F, LIU X M. Enabling efficient and expressive spatial keyword queries on encrypted data[C]//Proceedings of the ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Piscataway: IEEE Press, 2021: 2670-2674.

[22] ZHANG W T, ZHAO M Y, SUN Z Y, et al. VSpatial: enabling private and verifiable spatial keyword-based positioning in 6G-oriented IoT[J]. IEEE Journal on Selected Areas in Communications, 2024, 42(10): 2954-2969.

[23] ZHANG C, ZHU L H, XU C, et al. Location privacy-preserving task recommendation with geometric range query in mobile crowdsensing[J]. IEEE Transactions on Mobile Computing, 2022, 21(12): 4410-4425.

[24] SUN L L, LU R X, ZHENG Y D, et al. Efficient and privacy-preserving weighted nearby-fit spatial keyword query in cloud[J]. IEEE Internet of Things Journal, 2025(99): 1.

[25] BONEH D, DI CRESCENZO G, OSTROVSKY R, et al. Public key encryption with keyword search[C]// Advances in Cryptology - EUROCRYPT 2004. Berlin: Springer, 2004: 506-522.

[26] MIAO Y B, YANG Y T, LI X H, et al. Efficient privacy-preserving spatial data query in cloud computing[J]. IEEE Transactions on Knowledge and Data Engineering, 2024, 36(1): 122-136.

[27] CUI N N, LI J X, YANG X C, et al. When geo-text meets security: privacy-preserving Boolean spatial keyword queries[C]//Proceedings of the 2019 IEEE 35th International Conference on Data Engineering (ICDE). Piscataway: IEEE Press, 2019: 1046-1057.

[28] GUO R Y, QIN B, WU Y C, et al. Lu$_x$Geo: efficient and security-enhanced geometric range queries[J]. IEEE Transactions on Knowledge and Data Engineering, 2023, 35(2): 1775-1790.

[29] WONG W K, CHEUNG D W, KAO B, et al. Secure kNN computation on encrypted databases[C]//Proceedings of the 2009 ACM SIGMOD International Conference on Management of data. New York: ACM Press, 2009: 139-152.

[30] HUANG Q L, DU J B, YAN G Y, et al. Privacy-preserving spatio-temporal keyword search for outsourced location-based services[J]. IEEE Transactions on Services Computing, 2022, 15(6): 3443-3456.

[31] DAVIS N, RAINA G, JAGANNATHAN K. Taxi demand forecasting: a HEDGE-based tessellation strategy for improved accuracy[J]. IEEE Transactions on Intelligent Transportation Systems, 2018, 19(11): 3686-3697.

[32] ASGHAR N. Yelp dataset challenge: review rating prediction[J]. arXiv Preprint, arXiv: 1605.05362, 2016.

[33] CHO E, MYERS S A, LESKOVEC J. Friendship and mobility: user movement in location-based social networks[C]//Proceedings of the

17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York: ACM Press, 2011: 1082-1090.

[34] YANG D Q, ZHANG D Q, ZHENG V W, et al. Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2015, 45(1): 129-142.

[作者简介]

宋甫元（1991– ），男，江西上饶人，博士，南京信息工程大学讲师、硕士生导师，主要研究方向为隐私保护、物联网安全、应用密码学、数据安全等。

丁思洋（2000– ），男，江苏泰州人，南京信息工程大学硕士生，主要研究方向为隐私保护、应用密码学等。

王威（1990– ），男，山东菏泽人，博士，西安交通大学教授、博士生导师，主要研究方向为6G无线通信、无线系统安全、低空智联网与低空无人机监管等。

姜琴（1989– ），女，河南信阳人，博士，南京信息工程大学讲师、硕士生导师，主要研究方向为隐私保护、应用密码学、区块链安全等。

付章杰（1983– ），男，江苏南京人，博士，南京信息工程大学教授、博士生导师，主要研究方向为隐私保护、应用密码学、人工智能安全、区块链安全等。