Location Privacy-Preserving Task Recommendation With Geometric Range Query in Mobile Crowdsensing

Chuan Zhang, *Student Member, IEEE*, Liehuang Zhu, *Member, IEEE*, Chang Xu, Jianbing Ni, *Member, IEEE*, Cheng Huang, *Member, IEEE*, and Xuemin Shen, *Fellow, IEEE*

Abstract—In mobile crowdsensing, location-based task recommendation requires each data requester to submit a task-related geometric range to crowdsensing service providers such that they can match suitable workers within this range. Generally, a trusted server (i.e., database owner) should be deployed to protect location privacy during the process, which is not desirable in practice. In this paper, we propose the location privacy-preserving task recommendation (PPTR) schemes with geometric range query in mobile crowdsensing without the trusted database owner. Specifically, we first propose a PPTR scheme with linear search complexity, named PPTR-L, based on a two-server model. By leveraging techniques of polynomial fitting and randomizable matrix multiplication, PPTR-L enables the service provider to find the workers located in the data requester's arbitrary geometric query range without disclosing the sensitive location privacy. To further improve query efficiency, we design a novel data structure for task recommendation and propose PPTR-F to achieve faster-than-linear search complexity. Through security analysis, it is shown that our schemes can protect the confidentiality of workers' locations and data requesters' queries. Extensive experiments are performed to demonstrate that our schemes can achieve high computational efficiency in terms of geometric range query.

Index Terms—Task recommendation, location, privacy, geometric range query, mobile crowdsensing

1 Introduction

The fast development of wireless communications and mobile devices has given rise to mobile crowdsensing, an emerging paradigm that enables data requesters to publish tasks and a crowd of end-users sense, collect, and process data with their capable sensing and computing devices, such as smartphones, tablet computers, and wearable devices [1], [2], [3]. This "sensing as a service'' [4] has greatly broadened the availability of sensory data while reducing the time and money needed on each task. Currently, thanks to the flourish of various on-board sensors (e.g., GPS, camera, compass, etc.), a large variety of mobile crowdsensing systems have been developed, supporting a broad range of applications, including traffic monitoring [5], health-care services [6], [7], and social recommendation [8].

- *Chuan Zhang is with the School of Computer Science and Technology, Beijing Institute of Technology, Beijing 100811, China, and also with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: chuanz@bit.edu.cn.*
- *Liehuang Zhu and Chang Xu are with the School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing 100811, China. E-mail: {liehuangz, xuchang}@bit.edu.cn.*
- *Jianbing Ni is with the Department of Electrical and Computer Engineering, Queen's University, Kingston, ON K7L 3N6, Canada. E-mail: jianbing.ni@queensu.ca.*
- *Cheng Huang and Xuemin Shen are with the Department of Electrical and Computer Engineering, University of Waterloo, Waterloo, ON N2L 3G1, Canada. E-mail: {c225huan, sshen}@uwaterloo.ca.*

Manuscript received 28 Sept. 2020; revised 10 Mar. 2021; accepted 11 May 2021. Date of publication 17 May 2021; date of current version 3 Nov. 2022. (Corresponding author: Liehuang Zhu.) Digital Object Identifier no. 10.1109/TMC.2021.3080714

In a typical mobile crowdsensing system, end-users are registered as workers in the crowdsensing platform (referred to as *service provider*). When new crowdsensing tasks come, the service provider recommends the tasks to a proper subset of workers according to the correlation between their locations and the published tasks [9]. In this *task recommendation* process, geometric range query (GRQ) is a fundamental search functionality that can help data requesters to find the workers inside a certain geometric region, such as rectangle, triangle, and circle, to perform the data sensing tasks [10]. For example, a healthcare center attempts to identify if there is a virus outbreak in a specific area and thus it is supposed to find the users inside the area to collect their health information; a driver wants to know the traffic conditions in his/her destination area and thus he/she prefers to find the drivers inside the destination area to collect their real-time traffic information. However, due to the potential threats from both external and inside attackers, workers' locations and data requesters' queries may be disclosed during the geometric range query process, leading to data abuse, economic loss, or other disastrous results. For example, driven by profits, the service provider may try to obtain workers' location information and sell them to advertisers, retailers, or other companies and individuals. Adversaries who get workers' locations may launch attacks such as surveillance, tracking, theft, or even robbery, putting workers in grave danger. Thus, to encourage workers to be engaged in mobile crowdsensing, the location privacy should be carefully considered in geometric range query of task recommendation.

Recently, significant efforts have been devoted to protecting the location privacy when considering geometric range

query [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], but most of them have their inherent weaknesses, making them hard to be directly applied in the scenario of task recommendation. In [13], [14], [15], [18], the public-key based mechanisms are investigated to realize range query over encrypted spatial data. Unfortunately, due to the time-consuming homomorphic operations, these schemes are computationally expensive. To enable more efficient geometric range query, a series of symmetric searchable encryption (SSE) based approaches [10], [11], [12], [16], [17], [20] have been proposed. Compared with the public-key based mechanisms, these schemes essentially adopted a *database owner*-centric model, i.e., a database owner is employed to encrypt users' locations and share a same search key to data requesters. The success of these SSE-based schemes relies on an important assumption that the database owner and all data requesters are fully trusted, which however is a strong assumption in mobile crowdsensing. In reality, it is vulnerable that the service provider may collude with a data requester to harvest workers' sensitive locations and data requesters' queries. Moreover, the *database owner*-centric model also sacrifices system scalability since the search key and workers' encrypted data need to be updated after each revocation or update of data requesters, which introduces a great deal of computation and communication overhead towards the system. In addition to the above weaknesses, how to achieve arbitrary geometric range query in mobile crowdsensing is another challenge, and most of the existing works [11], [13], [14], [15], [18], [19] only support circle-based or rectangle-based query. Since the search area in mobile crowdsensing may be arbitrary, e.g., a town, a street, or a community, many useless results will be returned by the circular or rectangular range query schemes.

To resolve the above issues, in this paper, we propose a location Privacy-Preserving Task Recommendation (PPTR) scheme which supports efficient arbitrary geometric range query and does not rely on any trusted database owner. By employing a two-server model and leveraging the techniques of randomizable matrix multiplication and polynomial fitting, PPTR enables the service provider to perform worker selection in an efficient and privacy-preserving manner. Specifically, the contributions of this paper are summarized as follows.

- We formulate the problem of privacy-preserving task recommendation with geometric range query in mobile crowdsensing, identify a system model without the trusted database owner, and establish a well-defined threat model based on adversaries' different attack capabilities.
- We propose a location privacy-preserving task recommendation scheme (PPTR-L) over encrypted spatial data to allow the service provider to find the workers inside a data requester's arbitrary geometric query range. Specifically, we design a key derivation approach based on matrix decomposition, represent workers' locations and data requesters' queries by curvilinear equations, and protect data privacy using randomizable matrices. By exploiting the characteristic of matrices, we enable the service provider to obtain the location matching results, without knowing workers' geographic locations and data requesters' queries. Since all computations are based on matrix multiplication, PPTR-L introduces little computational overhead to the workers and data requesters.
- To further improve query efficiency, we propose a more efficient scheme named PPTR-F to achieve *faster-than-linear* search complexity. Specifically, we consider data requesters' historical search behaviors and label the historical search results with indexes. When new queries come, the service provider can filter a large number of workers by performing few matching operations. Experiments conducted on Python show that PPTR-F can save at most 98 percent time cost compared with PPTR-L in the query process.

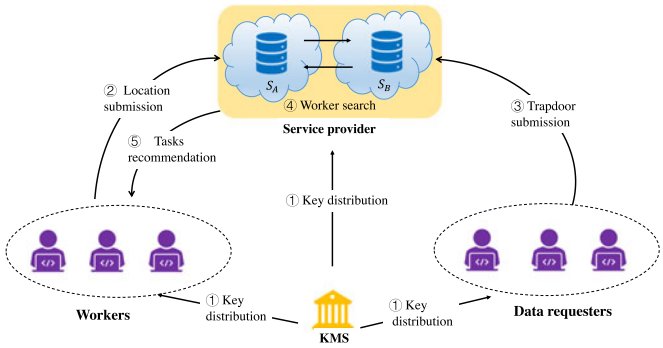


Fig. 1. System model.

The remainder of this paper is organized as follows. In Section 2, we introduce system model, threat model, and design goals. In Section 3, we introduce the background including the technique of polynomial fitting and the definition of geometric range query in task recommendation. After that, we elaborate details of our proposed schemes in Section 4 and analyze their security in Section 5. In Section 6, we conduct experiments to evaluate our schemes' performance. Finally, we give a brief introduction of related work in Section 7 and conclude our work in Section 8.

2 Models and Design Goals

This section identifies the system model, threat model, and introduces the design goals.

2.1 System Model

As shown in Fig. 1, the architecture of our schemes mainly consists of four entities, namely a key management server (KMS), a service provider, workers, and data requesters.

- KMS is responsible to generate encryption and re-encryption keys for participating entities (Step ①). After initializing the system, KMS will stay offline.
- The service provider has unlimited storage space and computational abilities that can store workers' location information and provide crowdsensing task recommendation services for data requesters (Step ④,⑤). In this model, two non-colluding cloud service providers S_A, S_B, such as Amazon and Microsoft, serve as the service provider.
- Workers are data providers, and they receive tasks according to their locations. To protect the location privacy, they encrypt their locations before sending them to the service provider (Step ②).

- Data requesters expect to find workers located in a specific geographic region, while not disclosing their query privacy. To achieve this goal, they generate trapdoors for the queries and upload them to the service provider (Step ③).

2.2 Threat Model

In our model, we assume that the KMS is fully trusted and the communication between KMS and any other entity is secure. Workers, data requesters, and the service provider are semi-honest, which is, they will honestly perform the designed protocol, but try to derive private information from encrypted locations and trapdoors. Based on the information adversaries may derive, we consider the following two level attacks.

- *Level-I attack*: Adversaries can observe a number of ciphertexts, but cannot obtain their corresponding plaintexts. This attack is known as the Ciphertext-Only-Attack (COA) and it happens when an external attacker eavesdrops a user's communication.
- *Level-II attack*: The adversary can obtain more information than what it gains in *Level-I* attack. It is possible for an external attacker to know a worker's location via physical observation. If there is a collusion between a cloud and some registered workers/data requesters, the adversary can even select a number of plaintexts and get their corresponding ciphertexts. This attack corresponds to the Chosen-Plaintext-Attack (CPA) in cryptography.

In general, two clouds are non-colluding: a cloud will not share its private key to the other cloud. This is a common assumption in most existing two-server based mechanisms [2], [21], [22], [23]. In addition, we assume that a cloud may collude with some valid users, but the colluding users can only submit limited number of locations or queries to the system. This assumption is reasonable since (1) a worker corresponds to only one location in the system; (2) a data requester usually needs to pay for his/her search queries. In fact, we do allow a cloud to find some workers in a certain area if the cloud submits search queries to the system, since this is the aim of task recommendation, and the other cloud and the returned workers can get paid by the cloud's search behaviors. Moreover, the focus of this paper is privacy-preserving geometric range query in task recommendation, and thus the privacy of crowdsensing tasks and user identity is not considered. In practice, users can encrypt the tasks [3] or adopt randomizable techniques such as pseudonyms [24], anonymous authentication [25], and one-way hash chain [23] to hide their real identity information. The Sybil attack is not taken into account either, as there have been many mechanisms [26], [27], [28] proposed to defend against such an attack. It should be mentioned that we do allow the clouds to be aware of some leaked information, including (1) the total number of encrypted locations and trapdoors, i.e., *Size Pattern*; (2) the identities returned for a specific geometric range query, i.e., *Access Pattern*; and (3) whether an encrypted location is searched for two different trapdoors, i.e., *Search Pattern*. These information are default to be known in most searchable encryption schemes [10], [11], [12], [16], [17].

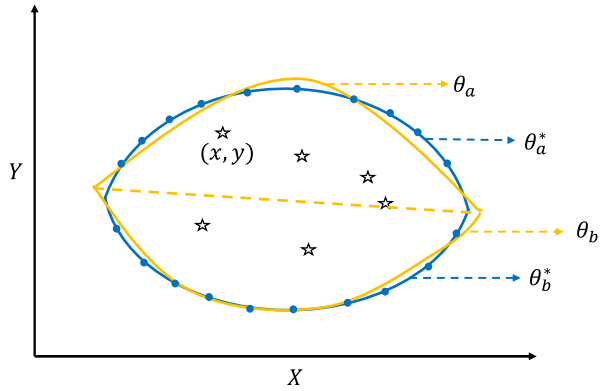


Fig. 2. Polynomial curve fitting.

2.3 Design Goals

According to system and threat models, our schemes should fulfill the following goals.

- *Data privacy*: The privacy of locations and queries should be well protected. That is, given the ciphertexts, adversaries, including curious entities and external attackers, cannot derive the sensitive location or query information from the ciphertexts.
- *Query accuracy*: The accuracy of query results should be guaranteed. That is, given a query, our schemes should find the workers that are located in the data requester's geometric query range.
- *Query efficiency*: The query efficiency should be guaranteed. That is, given a query, our schemes should efficiently handle a large number of matching operations.

3 Background

In this section, we first review the polynomial fitting technique and based on this, we give the definition of task recommendation with geometric range query.

The polynomial fitting is a data processing technique that can construct a curve to best fit a series of data points. Due to its effectiveness and scalability, in this paper, we utilize the polynomial fitting technique to generate the trapdoor for the geometric range query. As shown in Fig. 2, a random geometric range (i.e., the orange region) can be fitted by two curves (i.e., the blue curves) with curvilinear equations being $\theta_a^* = a_0 + a_1 x + \cdots + a_n x^n$ and $\theta_b^* = b_0 + b_1 x + \cdots + b_n x^n$, respectively, where $\{a_i, b_i\}_{i=0}^n$ are coefficients of the curvilinear equations and n is the highest degree of the equation. Given a geometric range (represented as θ_a^*, θ_b^*) and a point with geographic coordinate (x_i, y_i), the following steps are performed to identify whether the point is in the query range.

- Calculate and check if $\theta_a^*(x_i) - y_i > 0$. If so, moving to next step; otherwise, stopping immediately.
- Calculate and check if $\theta_b^*(x_i) - y_i < 0$. If so, this point is in the query range; otherwise, stopping immediately.

Note that, since the polynomial fitting technique is an approximation algorithm, the fitting curves may not perfectly match the actual search range and accordingly polynomial

fitting error will be inevitably introduced. However, some techniques such as orthogonal family can be employed to control the fitting error into a very small range. Interested readers can find more details in [10] and [29].

Based on the polynomial fitting technique, we now define task recommendation with geometric range query (TGRQ). In a TGRQ scheme, a worker sends his/her location to the service provider, while the data requester outsources crowdsensing tasks together with the geometric range requirement. Based on the above information, the service provider should recommend the tasks to the workers within the query range.

Definition 1. *A TGRQ scheme consists of three algorithms as follows.*

- $\mathsf{LocBuild}((x_i, y_i)) \rightarrow l_i$: Given a location (x_i, y_i), this algorithm outputs $l_i = (1, x_i, x_i^2, \ldots, x_i^n, y_i)$.
- $\mathsf{QueryGen}(\theta_a^*, \theta_b^*) \rightarrow (l_a, l_b)$: Given the geometric query range, i.e., θ_a^*, θ_b^*, this algorithm outputs the query $l_a = (a_0, a_1, \ldots, a_n, -1), l_b = (b_0, b_1, \ldots, b_n, -1)$.
- $\mathsf{Query}(l_i, (l_a, l_b)) \rightarrow \mathsf{R}$: Given l_i and the query (l_a, l_b), this algorithm outputs $\mathsf{R} = 1$ if $l_i \circ l_a > 0$ & $l_i \circ l_b < 0$ are satisfied, where $l_i \circ l_a$, $l_i \circ l_b$ denote the inner products of l_i, l_a and l_i, l_b, respectively. Otherwise, this algorithm outputs 0.

4 Proposed PPTR Schemes

In this section, we propose our PPTR schemes, which mainly consist of seven phases: (1) System Setup, (2) Key Generation, (3) Location Encryption, (4) Location Transformation, (5) Trapdoor Generation, (6) Trapdoor Transformation, and (7) Worker Query. Especially, we first propose a basic scheme with linear search complexity, namely PPTR-L, based on the polynomial fitting technique and randomizable matrix multiplication. Then, we improve PPTR-L by designing a novel data structure and propose PPTR-F to reduce the search complexity from linear to faster-than-linear. For ease of presentation, we use S_A, S_B to denote the two clouds, respectively.

4.1 PPTR-L: A Linear Scheme

(1) System Setup : Given the highest degree (i.e., n) of the curvilinear equation, KMS first generates four random invertible matrices $\{\mathcal{M}_1, \mathcal{M}_A, \mathcal{M}_B\} \in \mathbb{R}^{(n+8)\times(n+8)}$ as the master key.

(2) Key Generation : Given a worker's identity u_i, KMS first selects a random $(n+8) \times (n+8)$ invertible matrix $\mathcal{A}_i$, a random $(n+8) \times (n+8)$ lower triangular matrix $\mathcal{I}_i$ with the main diagonal being $(0, 0, 1, \ldots, 1, 1, 1, 0, 0)$ as

$$\begin{bmatrix} 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ * & * & 1 & 0 & \cdots & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ * & * & \cdots & * & 1 & 0 & 0 \\ * & * & \cdots & * & * & 0 & 0 \\ * & * & \cdots & * & * & 0 & 0 \end{bmatrix},$$

and computes $\mathcal{A}_i' = \mathcal{M}_1 \mathcal{I}_i \mathcal{A}_i^{-1}$. For simplicity of presentation, we use $*$ to denote random values in $\mathcal{I}_i$. For a data requester u_q, KMS selects a random $(n+8) \times (n+8)$ invertible matrix $\mathcal{B}_q$, a random lower triangular matrix $\mathcal{S}_q$ similar to $\mathcal{I}_i$, and then computes $\mathcal{B}_q' = \mathcal{B}_q^{-1} \mathcal{S}_q \mathcal{M}_1^{-1}$.

Then, KMS calculates perturbed matrix $\mathcal{R}_i = \mathcal{A}_i \mathcal{I}_i'$ for each worker, $\mathcal{R}_q = \mathcal{S}_q' \mathcal{B}_q$ for each data requester, where $\mathcal{I}_i', \mathcal{S}_q'$ are lower triangular matrices with main diagonal being $(0, 0, \ldots, 0, \gamma_{i,1}, \gamma_{i,2}, 0, 0)$ and $(0, 0, \ldots, 0, \gamma_{q,1}, \gamma_{q,2}, 0, 0)$, $\{\gamma_{i,1}, \gamma_{i,2}, \gamma_{q,1}, \gamma_{q,2}\} > 0$, respectively. At last, KMS sends the encryption keys $\mathcal{A}_i$ to the worker u_i, $\mathcal{B}_q$ to the data requester u_q, and sends the re-encryption keys $(\mathcal{M}_A, \mathcal{M}_B, \mathcal{A}_i', \mathcal{R}_i, \mathcal{R}_q)$ to the cloud S_A, $\mathcal{M}_B^{-1} \mathcal{B}_q'$ to the cloud S_B.

(3) Location Encryption : u_i retrieves the location coordinate $\mathsf{p}_i = (x_i, y_i)$ from the GPS device, and then performs the following operations to encrypt his/her location.

- First, u_i extends p_i to an $(n+8)$-dimension vector $l_i = (0, 0, 1, x_i, x_i^2, \ldots, x_i^n, y_i, 0, 0, 0, 0)$, and generates a random $(n+8) \times (n+8)$ lower triangular matrix $\mathcal{P}_i$ similar to $\mathcal{I}_i$ with main diagonal being l_i.
- Next, u_i uses his/her encryption key to encrypt the extended matrix $\mathcal{P}_i$ as

$$\mathcal{C}_i = \mathcal{A}_i \mathcal{P}_i, \tag{1}$$

and sends the encrypted location along with his/her identity to cloud S_A.

(4) Location Transformation : On receiving the ciphertext from u_i, S_A first finds the corresponding re-encryption keys $(\mathcal{M}_A, \mathcal{A}_i', \mathcal{R}_i)$, selects a random lower triangular matrix $\mathcal{D}_i$ with main diagonal being $(0, 0, v_i, \ldots, v_i, 1, 1, 0, 0)$ where v_i is a positive random value, and then re-encrypts $\mathcal{C}_i$ as shown in Eq. (2) before sending it to the cloud S_B

$$\begin{aligned} \widetilde{\mathcal{C}}_i &= \mathcal{A}_i'(\mathcal{C}_i + \mathcal{R}_i)\mathcal{D}_i \mathcal{M}_A \\ &= \mathcal{A}_i' \mathcal{A}_i (\mathcal{P}_i + \mathcal{I}_i') \mathcal{D}_i \mathcal{M}_A \\ &= \mathcal{M}_1 \mathcal{I}_i (\mathcal{P}_i + \mathcal{I}_i') \mathcal{D}_i \mathcal{M}_A. \end{aligned} \tag{2}$$

(5) Trapdoor Generation : Given a geometric query range, a data requester u_q generates the search trapdoor as follows.

- First, u_q employs the polynomial fitting technique to build two curvilinear equations $\theta_a^*(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ and $\theta_b^*(x) = b_0 + b_1 x + b_2 x^2 + \cdots + b_n x^n$ to fit the search area. Then, u_q extracts the coefficients of the two curvilinear equations to generate the geometric range query as $(\mathsf{q}_a, \mathsf{q}_b)$, where $\mathsf{q}_a = (a_0, a_1, \ldots, a_n), \mathsf{q}_b = (b_0, b_1, \ldots, b_n)$.
- Next, u_q extends q_a to an $(n+8)$-dimension vector $l_a = (0, 0, a_0, a_1, \ldots, a_n, -1, 0, 0, 0, 0)$ and extends q_b to an $(n+8)$-dimension vector $l_b = (0, 0, b_0, b_1, \ldots, b_n, -1, 0, 0, 0, 0)$. Then, u_q generates two random $(n+8) \times (n+8)$ lower triangular matrices $\mathcal{Q}_a, \mathcal{Q}_b$ with the main diagonal being l_a, l_b, respectively, and encrypts the matrices as

$$\mathcal{T}_a = \mathcal{Q}_a \mathcal{B}_q, \quad \mathcal{T}_b = \mathcal{Q}_b \mathcal{B}_q. \tag{3}$$

Finally, u_q sends the trapdoor $\mathcal{T}_a, \mathcal{T}_b$ along with his/her identity u_q to S_A.

(6) Trapdoor Transformation : On receiving the query information from u_q, S_A first finds u_q's re-encryption keys $(\mathcal{M}_A, \mathcal{M}_B, \mathcal{R}_q)$, selects two random lower triangular matrices

$\mathcal{D}_a, \mathcal{D}_b$ with the main diagonal being $(0, 0, v_a, \ldots, v_a, r_{a,1}, r_{a,2}, 0, 0), (0, 0, v_b, \ldots, v_b, r_{b,1}, r_{b,2}, 0, 0)$, respectively, where v_a, $v_b, r_{a,1}, r_{a,2}, r_{b,1}, r_{b,2}$ are positive random values, and re-encrypts the trapdoor as

$$\begin{aligned}\widetilde{\mathcal{T}}_a &= \mathcal{M}_A^{-1}\mathcal{D}_a(\mathcal{T}_a + \mathcal{R}_q)\mathcal{M}_B \\ &= \mathcal{M}_A^{-1}\mathcal{D}_a(\mathcal{Q}_a\mathcal{B}_q + \mathcal{S}'_q\mathcal{B}_q)\mathcal{M}_B \\ &= \mathcal{M}_A^{-1}\mathcal{D}_a(\mathcal{Q}_a + \mathcal{S}'_q)\mathcal{B}_q\mathcal{M}_B, \\ \widetilde{\mathcal{T}}_b &= \mathcal{M}_A^{-1}\mathcal{D}_b(\mathcal{T}_b + \mathcal{R}_q)\mathcal{M}_B \\ &= \mathcal{M}_A^{-1}\mathcal{D}_b(\mathcal{Q}_b + \mathcal{S}'_q)\mathcal{B}_q\mathcal{M}_B.\end{aligned} \tag{4}$$

Then, S_B finds u_q's re-encryption key $\mathcal{M}_B^{-1}\mathcal{B}'_q$ to re-encrypt the trapdoor as

$$\begin{aligned}\overline{\mathcal{T}}_a &= \widetilde{\mathcal{T}}_a\mathcal{M}_B^{-1}\mathcal{B}'_q \\ &= \mathcal{M}_A^{-1}\mathcal{D}_a(\mathcal{Q}_a + \mathcal{S}'_q)\mathcal{B}_q\mathcal{M}_B\mathcal{M}_B^{-1}\mathcal{B}'_q \\ &= \mathcal{M}_A^{-1}\mathcal{D}_a(\mathcal{Q}_a + \mathcal{S}'_q)\mathcal{S}_q\mathcal{M}_1^{-1}, \\ \overline{\mathcal{T}}_b &= \widetilde{\mathcal{T}}_b\mathcal{M}_B^{-1}\mathcal{B}'_q \\ &= \mathcal{M}_A^{-1}\mathcal{D}_b(\mathcal{Q}_b + \mathcal{S}'_q)\mathcal{S}_q\mathcal{M}_1^{-1}.\end{aligned} \tag{5}$$

(7) Worker Query : With the transformed trapdoor, S_B retrieves workers by calculating the inner products of l_i and l_a, l_b in ciphertexts. Specifically, given an encrypted location $\widetilde{\mathcal{C}}_i$ and the trapdoor $(\overline{\mathcal{T}}_a, \overline{\mathcal{T}}_b)$, S_B computes $\widetilde{\mathcal{C}}_i\overline{\mathcal{T}}_a, \widetilde{\mathcal{C}}_i\overline{\mathcal{T}}_b$ and further calculates their trace as

$$\begin{aligned}\Delta_a &= tr(\widetilde{\mathcal{C}}_i\overline{\mathcal{T}}_a) \\ &= tr(\mathcal{M}_1\mathcal{I}_i(\mathcal{P}_i + \mathcal{I}'_i)\mathcal{D}_i\mathcal{D}_a(\mathcal{Q}_a + \mathcal{S}'_q)\mathcal{S}_q\mathcal{M}_1^{-1}) \\ &= v_iv_a(\theta_a^*(x_i) - y_i) + r_{a,1}\gamma_{q,1}\gamma_{i,1} + r_{a,2}\gamma_{q,2}\gamma_{i,2}, \\ \Delta_b &= tr(\widetilde{\mathcal{C}}_i\overline{\mathcal{T}}_b) \\ &= tr(\mathcal{M}_1\mathcal{I}_i(\mathcal{P}_i + \mathcal{I}'_i)\mathcal{D}_i\mathcal{D}_b(\mathcal{Q}_b + \mathcal{S}'_q)\mathcal{S}_q\mathcal{M}_1^{-1}) \\ &= v_iv_b(\theta_b^*(x_i) - y_i) + r_{b,1}\gamma_{q,1}\gamma_{i,1} + r_{b,2}\gamma_{q,2}\gamma_{i,2}.\end{aligned} \tag{6}$$

To eliminate the influence of the perturbations $r_{a,1}\gamma_{q,1}\gamma_{i,1} + r_{a,2}\gamma_{q,2}\gamma_{i,2}$ and $r_{b,1}\gamma_{q,1}\gamma_{i,1} + r_{b,2}\gamma_{q,2}\gamma_{i,2}$, we should control the range of random values. Specifically, if we assume $|\theta_a^*(x_i) - y_i|$ and $|\theta_b^*(x_i) - y_i|$ is larger than a very small positive value A and the random values $\{r_{a,1}, r_{a,2}, r_{b,1}, r_{b,2}, \gamma_{i,1}, \gamma_{i,2}, \gamma_{q,1}, \gamma_{q,2}\}$ are smaller than a constant positive value B, $\{v_i, v_a, v_b\} \gg \sqrt{\frac{2B^3}{A}}$ should be satisfied. Then, if $\Delta_a > 0$ & $\Delta_b < 0$, we have $\mathsf{R}_{iq} = 1$. This means worker u_i is in the data requester u_q's search area and S_B should add the worker's identity in a result set RS_q. Otherwise, S_B selects another encrypted location and repeats the above query process. After retrieving all encrypted locations, S_B is able to get a set of workers that are in the geometric query range, and it is expected to recommend the task to the workers in $\mathsf{RS}_q = \{u_i\}_{\mathsf{R}_{iq}=1}$.

4.2 Correctness Analysis of PPTR-L

Here, we analyze the correctness of PPTR-L. In our design, the lower triangular matrix, e.g., $\mathcal{P}_i$, is a special kind of square matrix where the elements above the main diagonal are zero and its trace, i.e., $tr(\mathcal{P}_i)$, is the sum of the main diagonal entries. From the linear algebra, we have the following lemmas.

Lemma 1. Given a square matrix $\mathcal{P}$ and an invertible matrix $\mathcal{M}$, $tr(\mathcal{P}) = tr(\mathcal{M}\mathcal{P}\mathcal{M}^{-1})$ is satisfied.

Lemma 2. Given two lower triangular matrices $\mathcal{A}, \mathcal{B}$ with the main diagonal being $\mathbf{A}, \mathbf{B}$, respectively, $tr(\mathcal{A}\mathcal{B}) = \mathbf{A} \circ \mathbf{B}$ is satisfied.

Based on the above two lemmas, we have *Theorem 1*.

Theorem 1. In PPTR-L, PPTR-L.$\mathsf{Query}(\widetilde{\mathcal{C}}_i, (\overline{\mathcal{T}}_a, \overline{\mathcal{T}}_b)) \rightarrow \mathsf{R}_{iq}$ is correct if TGRQ.$\mathsf{Query}(l_i, (l_a, l_b)) \rightarrow \mathsf{R} = \mathsf{R}_{iq}$.

Proof. Recall the details of TGRQ, as described in Section 3, we know $l_i \circ l_a = \theta_a^*(x_i) - y_i, l_i \circ l_b = \theta_b^*(x_i) - y_i$ and we have $\theta_a^*(x_i) - y_i > 0$ & $\theta_b^*(x_i) - y_i < 0 \rightarrow \mathsf{R} = 1$. Recall the details of PPTR-L, as described in Section 4.1, we have $\Delta_a > 0$ & $\Delta_b < 0 \rightarrow \mathsf{R}_{iq} = 1$. To demonstrate the correctness of PPTR-L, we should prove the sign (i.e., positive or negative) of $\theta_a^*(x_i) - y_i$ (or $\theta_b^*(x_i) - y_i$) is equal to the sign of Δ_a (or Δ_b).

According to the system construction of PPTR-L, the vector l_i is first extended to a lower triangular matrix $\mathcal{P}_i$ with the main diagonal being l_i, encrypted as $\mathcal{C}_i$ in the worker side, and re-encrypted as $\widetilde{\mathcal{C}}_i$ in the clouds. Similarly, the query l_a (or l_b) is extended and encrypted as $\overline{\mathcal{T}}_a$ or $\overline{\mathcal{T}}_b$ in the clouds. With $\widetilde{\mathcal{C}}_i, \overline{\mathcal{T}}_a, \overline{\mathcal{T}}_b$, S_B calculates $\Delta_a = tr(\widetilde{\mathcal{C}}_i\overline{\mathcal{T}}_a)$, $\Delta_b = tr(\widetilde{\mathcal{C}}_i\overline{\mathcal{T}}_b)$. We first take Δ_a as an example and give the analysis below.

Step 1. Since $\mathcal{M}_1$ is an invertible matrix, based on Lemma 1, we know that $\Delta_a = tr(\mathcal{M}_1\mathcal{I}_i(\mathcal{P}_i + \mathcal{I}'_i)\mathcal{D}_i\mathcal{D}_a(\mathcal{Q}_a + \mathcal{S}'_q)\mathcal{S}_q\mathcal{M}_1^{-1}) = tr(\mathcal{I}_i(\mathcal{P}_i + \mathcal{I}'_i)\mathcal{D}_i\mathcal{D}_a(\mathcal{Q}_a + \mathcal{S}'_q)\mathcal{S}_q)$.

Step 2. Since $\mathcal{I}_i, \mathcal{P}_i, \mathcal{I}'_i, \mathcal{D}_i, \mathcal{D}_a, \mathcal{Q}_a, \mathcal{S}'_q, \mathcal{S}_q$ are lower triangular matrices with the main diagonal being $\mathbf{I}_i = (0, 0, 1, \ldots, 1, 1, 1, 0, 0), l_i, \mathbf{I}'_i = (0, \ldots, 0, \gamma_{i,1}, \gamma_{i,2}, 0, 0), \mathbf{D}_i = (0, 0, v_i, \ldots, v_i, 1, 1, 0, 0), \mathbf{D}_a = (0, 0, v_a, \ldots, v_a, r_{a,1}, r_{a,2}, 0, 0), l_a, \mathbf{S}'_q = (0, \ldots, 0, \gamma_{q,1}, \gamma_{q,2}, 0, 0), \mathbf{S}_q = (0, 0, 1, \ldots, 1, 1, 1, 0, 0)$, respectively, based on Lemma 2, we have $\Delta_a = \mathbf{I}_i \circ (l_i + \mathbf{I}'_i) \circ \mathbf{D}_i \circ \mathbf{D}_a \circ (l_a + \mathbf{S}'_q) \circ \mathbf{S}_q$, which is equal to $v_iv_a(\theta_a^*(x_i) - y_i) + r_{a,1}\gamma_{q,1}\gamma_{i,1} + r_{a,2}\gamma_{q,2}\gamma_{i,2}$.

Step 3. Due to the assumption $\{v_i, v_a\} \gg \sqrt{\frac{2B^3}{A}}, |\theta_a^*(x_i) - y_i| > A, \{r_{a,1}, r_{a,2}, \gamma_{i,1}, \gamma_{i,2}, \gamma_{q,1}, \gamma_{q,2}\} < B$, we can derive that $\theta_a^*(x_i) - y_i > 0 \rightarrow \Delta_a > 0$ and $\theta_a^*(x_i) - y_i < 0 \rightarrow \Delta_a < 0$. Hence, the sign of $\theta_a^*(x_i) - y_i$ is equal to the sign of Δ_a.

We then analyze the correctness of Δ_b. Following similar analysis, we can derive $\theta_b^*(x_i) - y_i > 0 \rightarrow \Delta_b > 0$ and $\theta_b^*(x_i) - y_i < 0 \rightarrow \Delta_b < 0$, which means the sign of $\theta_b^*(x_i) - y_i$ is also equal to the sign of Δ_b. Therefore, the correctness of PPTR-L is demonstrated. □

4.3 PPTR-F: A Faster-Than-Linear Scheme

Although PPTR-L can effectively identify workers in an arbitrary geographic region, it still suffers from the issue of efficiency. As described above, S_B needs to calculate the trace between every worker's encrypted location and the data requester's trapdoor, which introduces linear search complexity. To improve search efficiency, we design a novel data structure to reduce the search complexity. Specifically, our new data structure is inspired by the following

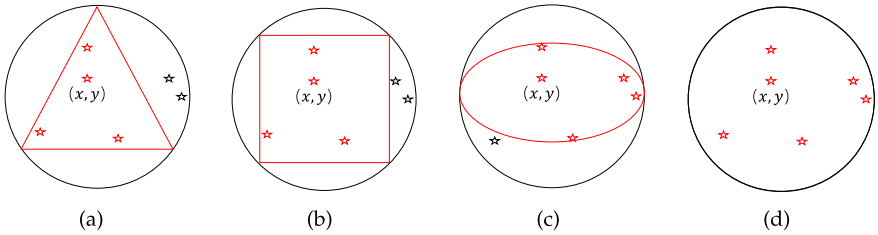


Fig. 3. Examples of circle covering (the red region denotes the query range and the black circle denotes the newly generated circle).

observations in PPTR-L: (1) a data requester's search result is a collection labeled with his/her query; (2) if there is an intersection between two queries, a worker in one collection is possible to be in the other collection, while conversely if the two queries have no intersections, a worker in one collection is impossible to be in the other collection. Thereby, we denote a data requester's query as the father node and the query results, i.e., a set of workers, as the children nodes of this father node. Given a geometric range query, the search process of PPTR-F is conducted as follows.

- *Step I:* Staring from the father node, if there is an intersection between the father node and the given query, moving to *Step II*; otherwise, repeating *Step I*.
- *Step II:* For each children node, checking if this node belongs to the geometric range. If so, adding this children node into the result set RS; otherwise, repeating *Step II*.
- *Step III:* For a node that has not been added in a father node, checking if it is in the geometric range. If so, adding this node in RS; otherwise, repeating *Step III*.

However, due to the arbitrariness of the geometric query range, it is difficult to determine whether there is an intersection between two queries. To deal with this challenge, we can generate a circle to cover the original search area, as shown in Fig. 3, and use the circles to judge whether two queries intersect. More specifically, given two circles, we calculate the difference between the distance of their centers and the sum of their radii to determine the relationship between the circles, and based on this further identify the relationship between two queries, i.e., if two circles intersect with each other (as shown in Figs. 4a, 4b, and 4c, there may be an intersection between two query results, and if two circles are separated without intersection (as shown in Fig. 4d), it is impossible to have intersections between the two queries. Consequently, we denote the newly generated circle, i.e., the black circle in Fig. 3, as the father node and the workers in the query range, i.e., the nodes in the red region in Fig. 3, as children nodes.

To protect the query privacy, the father node, i.e., the circle, should be encrypted before uploading to the clouds. Obviously, we can also use the same encryption method presented in PPTR-L to protect data privacy. Thus, the key part of PPTR-F is to check whether two circles intersect in the ciphertext domain. To the end, PPTR-F is implemented with the following procedure. In the phase of Trapdoor Generation, u_q additionally generates a circle $C_q((x_q, y_q), R_q)$ to cover the fitted area, where (x_q, y_q) denotes the coordinate of the center of the circle and R_q denotes the radius. Based on $C_q((x_q, y_q), R_q)$, u_q generates two random $(n+8) \times (n+8)$ lower triangular matrices $\mathcal{C}_{q,1}, \mathcal{C}_{q,2}$ with the main diagonal being $(0, 0, x_q, y_q, R_q, (x_q^2 + y_q^2 - R_q^2), 1, 0, \ldots, 0)$ and $(0, 0, -2x_q, -2y_q, -2R_q, 1, (x_q^2 + y_q^2 - R_q^2), 0, \ldots, 0)$, respectively. Then, by performing similar operations of Location Encryption, Location Trans*formation*, Trapdoor Generation, and Trapdoor Transformation, S_B derives $\widetilde{\mathcal{E}}_{q,1} = \mathcal{M}_1\mathcal{I}_q(\mathcal{C}_{q,1} + \mathcal{I}'_q)\mathcal{D}_{q,1}\mathcal{M}_A$ and $\overline{\mathcal{E}}_{q,2} = \mathcal{M}_A^{-1}\mathcal{D}_{q,2}(\mathcal{C}_{q,2} + \mathcal{S}'_q)\mathcal{S}_q\mathcal{M}_1^{-1}$, which are regarded as the father node and the search circle token, respectively. After performing Worker Query, the workers in $\mathcal{RS}_q$ are set as the children nodes of $\widetilde{\mathcal{E}}_{q,1}$. Given a new search token $(\overline{\mathcal{E}}_{k,2}, (\overline{\mathcal{T}}_a, \overline{\mathcal{T}}_b))$ from u_k, S_B first selects a father node, e.g., $\widetilde{\mathcal{E}}_{q,1}$, and calculates the value as shown in Eq. (7) to identity if there may be an intersection between two queries

$$\begin{aligned} ComS_{qk} &= tr(\widetilde{\mathcal{E}}_{q,1}\overline{\mathcal{E}}_{k,2}) \\ &= tr(\mathcal{M}_1\mathcal{I}_q(\mathcal{C}_{q,1} + \mathcal{I}'_q)\mathcal{D}_{q,1}\mathcal{M}_A \\ &\quad \mathcal{M}_A^{-1}\mathcal{D}_{k,2}(\mathcal{C}_{k,2} + \mathcal{S}'_k)\mathcal{S}_k\mathcal{M}_1^{-1}) \\ &= v_q v_k((x_k - x_q)^2 + \\ &\quad (y_k - y_q)^2 - (R_k + R_q)^2) \\ &\quad + r_{k,1}\gamma_{q,1}\gamma_{k,1} + r_{k,2}\gamma_{q,2}\gamma_{k,2}. \end{aligned} \tag{7}$$

To eliminate the influence of the perturbation $r_{k,1}\gamma_{q,1}\gamma_{k,1} + r_{k,2}\gamma_{q,2}\gamma_{k,2}$, if we assume $|(x_k - x_q)^2 + (y_k - y_q)^2 - (R_k + R_q)^2)|$ is larger than a very small value C, and $\{r_{k,1}, r_{k,2}, \gamma_{q,1}, \gamma_{q,2}, \gamma_{k,1}, \gamma_{k,2}\}$ are smaller than a constant positive value D, $\{v_q, v_k\} \gg \sqrt{\frac{2D^3}{C}}$ should be satisfied. Then, if $ComS_{qk} < 0$, S_B selects the children nodes in $\widetilde{\mathcal{E}}_{q,1}$ to compute $\{\Delta_a, \Delta_b\}$ to further check whether $\widetilde{\mathcal{E}}_{q,1}$'s children nodes are in the geometric range. Otherwise, S_B selects another father node $\widetilde{\mathcal{E}}_{q' \neq q,1}$ and continues to compute $ComS_{q'k}$. The details of PPTR-F are given in Fig. 5.

4.4 Correctness Analysis of PPTR-F

To demonstrate the correctness of PPTR-F, we have the following Theorems.

Theorem 2. In PPTR-F, PPTR-F.Query$((\widetilde{\mathcal{E}}_{q,1}, \widetilde{C}_i), (\overline{\mathcal{E}}_{k,2}, (\overline{\mathcal{T}}_a, \overline{\mathcal{T}}_a))) \rightarrow \mathsf{R}_{ik}$ is correct if TGRQ.Query$(l_i, (l_a, l_b)) \rightarrow \mathsf{R} = \mathsf{R}_{ik}$.

Proof. As shown in Fig. 4, we know that a point inside a circle is possible to be in the other circle if there is an intersection between two circles (as shown in Figs. 4a and 4b), and a point inside a circle is impossible to be in the other circle if two circles are tangential or separate (as shown in Figs. 4c and 4d). Based on this principle, in PPTR-F, we first check if there is an intersection between the two circles by calculating $ComS_{qk}$, and then retrieve the points in the circle if $ComS_{qk} < 0$. Thus, to demonstrate the

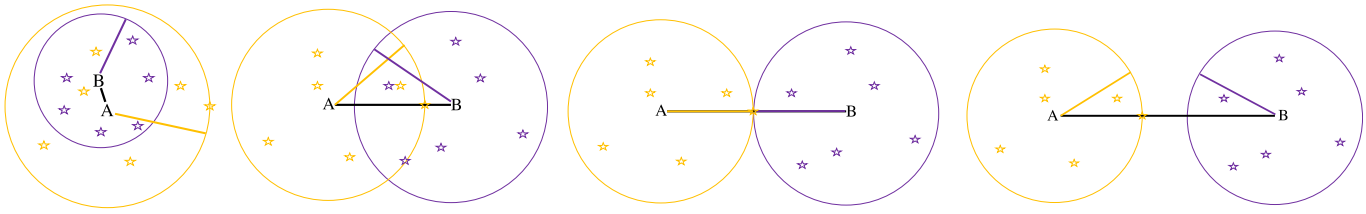


Fig. 4. All possible relationships between two circles.

correctness of PPTR-F, we should prove the correctness of $ComS_{qk}$, i.e., $ComS_{qk} < 0$ if and only if two circles $C_q((x_q, y_q), R_q), C_k((x_k, y_k), R_k)$ intersect with each other, which is given and demonstrated in *Theorem 3*. □

Theorem 3. In PPTR-F, $ComS_{qk} < 0$ if and only if the circle $C_q((x_q, y_q), R_q)$ intersects with the circle $C_k((x_k, y_k), R_k)$.

Proof. According to the geometric properties, if there is an intersection between two circles $C_q((x_q, y_q), R_q)$ and $C_k((x_k, y_k), R_k)$, the distance of the centers of the two circles should be smaller than the sum of their radii, i.e., $(x_q - x_k)^2 + (y_q - y_k)^2 < (R_k + R_q)^2$ should be satisfied. Recall the details of PPTR-F and the proof of *Theorem 1*, we have $ComS_{qk} = v_q v_k((x_q - x_k)^2 + (y_q - y_k)^2 - (R_q + R_k)^2) + r_{k,1}\gamma_{q,1}\gamma_{k,1} + r_{k,2}\gamma_{q,2}\gamma_{k,2}$. Due to the assumption $|(x_k - x_q)^2 + (y_k - y_q)^2 - (R_k + R_q)^2)| > C$, $\{r_{k,1}, r_{k,2}, \gamma_{q,1}, \gamma_{q,2}, \gamma_{k,1}, \gamma_{k,2}\} < D$, $\{v_q, v_k\} \gg \sqrt{\frac{2D^3}{C}}$, we know that when $(x_q - x_k)^2 + (y_q - y_k)^2 < (R_q + R_k)^2$,

★ (1) System Setup: Given the highest degree of the curvilinear equation, KMS generates the master key as

$$\{\mathcal{M}_1, \mathcal{M}_A, \mathcal{M}_B\}.$$

★ (2) Key Generation: Given a worker identity u_i, KMS generates the encryption keys and re-encryption keys as

$$\begin{aligned}\text{Encryption key} &: \mathcal{A}_i,\\ \text{Re-encryption key} &: \mathcal{A}'_i, \mathcal{R}_i, \mathcal{M}_A, \text{ where},\\ &\mathcal{A}'_i \leftarrow \mathcal{M}_1\mathcal{I}_i\mathcal{A}_i^{-1}.\end{aligned}$$

For a data requester u_q, KMS generates

$$\begin{aligned}\text{Encryption key} &: (\mathcal{A}_q, \mathcal{B}_q),\\ \text{Re-encryption key} &: (\mathcal{A}'_q, \mathcal{R}_q, \mathcal{M}_A, \mathcal{B}'_q, \mathcal{R}_q, \mathcal{M}_B, \mathcal{M}_B^{-1}\mathcal{B}_q),\\ &\text{where, } \mathcal{A}'_q \leftarrow \mathcal{M}_1\mathcal{I}_q\mathcal{A}_i^{-1},\\ &\mathcal{B}'_q = \mathcal{B}_q^{-1}\mathcal{S}_q\mathcal{M}_1^{-1}.\end{aligned}$$

★ (3) Location Encryption: Given a user u_i's location vector l_i, u_i generates a matrix $\mathcal{P}_i$ according to l_i, and encrypts $\mathcal{P}_i$ as

$$\mathcal{C}_i \leftarrow \mathcal{A}_i\mathcal{P}_i.$$

★ (4) Location Transformation: Given u_i's encrypted location $\mathcal{C}_i$, S_A re-encrypts it as

$$S_A : \widetilde{\mathcal{C}}_i \leftarrow \mathcal{A}'_i(\mathcal{C}_i + \mathcal{R}_i)\mathcal{D}_i\mathcal{M}_A.$$

★ (5) Trapdoor Generation: Given a data requester u_q's search query l_a, l_b, u_q first generates two matrices $\mathcal{Q}_a, \mathcal{Q}_b$ according to l_a, l_b, and then generates the trapdoor as

$$\mathcal{T}_a \leftarrow \mathcal{Q}_a\mathcal{B}_q, \quad \mathcal{T}_b \leftarrow \mathcal{Q}_b\mathcal{B}_q.$$

Then, let $C((x_q, y_q), R_q)$ be a circle that covers the search area, u_q generates two matrices $\mathcal{C}_{q,1}, \mathcal{C}_{q,2}$, and calculates

$$\mathcal{E}_{q,1} \leftarrow \mathcal{A}_q\mathcal{C}_{q,1}, \quad \mathcal{E}_{q,2} \leftarrow \mathcal{C}_{q,2}\mathcal{B}_q.$$

After that, u_q sends $(u_q, (\mathcal{T}_a, \mathcal{T}_b), \mathcal{E}_{q,1}, \mathcal{E}_{q,2})$ to S_A.

★ (6) Trapdoor Transformation: On receiving the trapdoor, clouds find u_q's re-encryption keys and calculate

$$\begin{aligned}&S_A : \overline{\mathcal{T}}_a \leftarrow \mathcal{M}_A^{-1}\mathcal{D}_a(\mathcal{T}_a + \mathcal{R}_q)\mathcal{M}_B, S_B : \overline{\mathcal{T}}_a \leftarrow \widetilde{\mathcal{T}}_a\mathcal{M}_B^{-1}\mathcal{B}'_q.\\ &S_A : \overline{\mathcal{T}}_b \leftarrow \mathcal{M}_A^{-1}\mathcal{D}_b(\mathcal{T}_b + \mathcal{R}_q)\mathcal{M}_B, S_B : \overline{\mathcal{T}}_b \leftarrow \widetilde{\mathcal{T}}_b\mathcal{M}_B^{-1}\mathcal{B}'_q.\\ &S_A : \widetilde{\mathcal{E}}_{q,1} \leftarrow \mathcal{A}'_q(\mathcal{E}_{q,1} + \mathcal{R}_q)\mathcal{D}_{q,1}\mathcal{M}_A.\\ &S_A : \widetilde{\mathcal{E}}_{q,2} \leftarrow \mathcal{M}_A^{-1}\mathcal{D}_{q,2}(\mathcal{T}_j + \mathcal{R}_q)\mathcal{M}_B,\\ &S_B : \overline{\mathcal{E}}_{q,2} \leftarrow \widetilde{\mathcal{E}}_{q,2}\mathcal{M}_B^{-1}\mathcal{B}'_q.\end{aligned}$$

★ (7) Worker Query: Upon receiving u_q's search trapdoor, S_B performs the search process as:

- Step I: For each father node $\widetilde{\mathcal{E}}_{k,1}, k \neq q$, S_B calculates
$$ComS_{qk} = tr(\widetilde{\mathcal{E}}_{k,1}\overline{\mathcal{E}}_{q,2}).$$
If $ComS_{qk} < 0$, moving to next step to search its children node; otherwise, repeating this step.
- Step II: When retrieving a children node u_i in $\widetilde{\mathcal{E}}_{k,1}$, S_B performs
$$\Delta_a = tr(\widetilde{\mathcal{C}}_i\overline{\mathcal{T}}_a), \quad \Delta_b = tr(\widetilde{\mathcal{C}}_i\overline{\mathcal{T}}_b).$$
If $\Delta_a > 0$ & $\Delta_b < 0$, S_B adds this node into the result set RS_q; otherwise, S_B selects another children node and repeats this step.
- Step III: For the non-children node u_l, S_B computes
$$\Delta_a = tr(\widetilde{\mathcal{C}}_l\overline{\mathcal{T}}_a), \quad \Delta_b = tr(\widetilde{\mathcal{C}}_l\overline{\mathcal{T}}_b).$$
If $\Delta_a > 0$ & $\Delta_b < 0$, S_B adds this node into RS_q and sets $\widetilde{\mathcal{E}}_{q,1}$ as the father node of u_l; otherwise, S_B selects another non-children node and repeats this step.

Finally, S_B recommends the crowdsensing task to workers in RS_q.

Fig. 5. Details of PPTR-F.

1: *Init:* Given a security parameter λ, $\mathcal{A}$ generates two location databases $\mathsf{DB}_0 = (p_{0,1}, \cdots, p_{0,t})$ and $\mathsf{DB}_1 = (p_{1,1}, \cdots, p_{1,t})$, and then submits them to $\mathcal{C}$, where $p_{i,j}, i \in \{0,1\}, j \in [1,t]$ is a location point and each element is selected from $[1, 2^\lambda]$.
2: *Setup:* $\mathcal{C}$ runs System Setup, Key Generation to generate the master key, a user's encryption key, and it keeps them secret.
3: *Challenge:* With $\mathsf{DB}_0, \mathsf{DB}_1$ obtained in *Init*, $\mathcal{C}$ flips a coin $b \in \{0,1\}$ and calculates $\mathcal{C}_{b,j}$ through Location Encryption. After that, $\mathcal{C}$ returns $\mathsf{EDB}_b \leftarrow \{\mathcal{C}_{b,j}\}_{i=1}^{t}$ to $\mathcal{A}$.
4: *Guess:* $\mathcal{A}$ takes a guess b' of b.
5: *Output:* This experiment outputs 1 if $b = b'$. Otherwise, this experiment outputs 0.

Fig. 6. *Level-I* attack experiment $\mathsf{Level-I}_{\mathcal{A},\mathsf{PPTR}}$.

$ComS_{qk} < 0$, and when $(x_q - x_k)^2 + (y_q - y_k)^2 > (R_q + R_k)^2$, $ComS_{qk} > 0$. Therefore, the correctness of *Theorem 3* is demonstrated and accordingly the correctness of *Theorem 2* is demonstrated as well. □

5 Security Analysis

In this section, we analyze the security of our proposed schemes. Since there are mainly three operations, i.e., data encryption (Location Encryption, Trapdoor Generation), data re-encryption (Location Transformation, Trapdoor Transformation), and data query (Worker Query), involved in both PPTR-L and PPTR-F, we will discuss them separately in the following sections. In addition, since the operations of location data and search query are almost the same, we take the location data as an example in the discussion of data encryption and data re-encryption. For ease of presentation, we use PPTR to denote the schemes of PPTR-L and PPTR-F.

5.1 Data Encryption

5.1.1 Security Against Level-I Attack

Under the *Level-I* attack, adversaries can observe a number of ciphertexts, but have no idea of the corresponding plaintexts. In the following, we first give an experiment $\mathsf{Level-I}_{\mathcal{A},\mathsf{PPTR}}$, as shown in Fig. 6, to simulate the security game played between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

We define the security of PPTR under the *Level-I* attack based on $\mathsf{Level-I}_{\mathcal{A},\mathsf{PPTR}}$.

Definition 2. *PPTR is secure against the Level-I attack if for any polynomial-time adversary $\mathcal{A}$, it has at most a negligible advantage* $\mathsf{negl}(\lambda)$, *such that*

$$|\mathsf{Pr}(\mathsf{Level-I}_{\mathcal{A},\mathsf{PPTR}}(1^\lambda) = 1) - \frac{1}{2}| \leq \mathsf{negl}(\lambda).$$

Theorem 4. *PPTR is secure against Level-I attack.*

Proof. As described in the security model, a *Level-II* attack is more powerful than a *Level-I* attack. That is, if PPTR is secure against the *Level-II* attack, it can also defend against the *Level-I* attack. Therefore, we skip the proof of *Theorem 4* and focus on the proof of *Theorem 5*. □

1: *Init:* Given a security parameter λ, $\mathcal{A}$ generates two location databases $\mathsf{DB}_0 = (p_{0,1}, \cdots, p_{0,t})$ and $\mathsf{DB}_1 = (p_{1,1}, \cdots, p_{1,t})$, where $p_{i,j}, i \in \{0,1\}, j \in [1,t]$ is a location point and each element is selected from $[1, 2^\lambda]$.
2: *Setup:* $\mathcal{C}$ runs System Setup, Key Generation to generate the master key, a user's encryption key, and it keeps them secret.
3: *Phase 1:* $\mathcal{A}$ uploads $p_{i,j}, i \in \{0,1\}, j \in [1,t]$ to $\mathcal{C}$. Then, $\mathcal{C}$ responses with a ciphertext $\mathcal{C}_{i,j}$ through Location Encryption.
4: *Challenge:* With $\mathsf{DB}_0, \mathsf{DB}_1$ selected in *Init*, $\mathcal{C}$ flips a coin $b \in \{0,1\}$ and calculates $\mathcal{C}_{b,j}$ via Location Encryption. After that, $\mathcal{C}$ returns $\mathsf{EDB}_b \leftarrow \{\mathcal{C}_{b,j}\}_{i=1}^{t}$ to $\mathcal{A}$.
5: *Phase 2:* $\mathcal{A}$ adaptively selects a number of messages and submits them to $\mathcal{C}$.
6: *Guess:* $\mathcal{A}$ takes a guess b' of b.
7: *Output:* This experiment outputs 1 if $b = b'$. Otherwise, this experiment outputs 0.

Fig. 7. *Level-II* attack experiment $\mathsf{Level-II}_{\mathcal{A},\mathsf{PPTR}}$.

5.1.2 Security Against Level-II Attack

In addition to the ciphertexts, a *Level-II* attacker has the ability to obtain their corresponding plaintexts. This attack happens when an external attacker has oracle access to a worker's encryption process. To prove that PPTR has CPA-security, we first give an experiment $\mathsf{Level-II}_{\mathcal{A},\mathsf{PPTR}}$, as shown in Fig. 7, to simulate the game between an adversary $\mathcal{A}$ and a challenger $\mathcal{C}$.

Definition 3. *PPTR is secure against the Level-II attack if for any polynomial-time adversary $\mathcal{A}$, it has at most a negligible advantage* $\mathsf{negl}(\lambda)$, *such that*

$$|\mathsf{Pr}(\mathsf{Level-II}_{\mathcal{A},\mathsf{PPTR}}(1^\lambda) = 1) - \frac{1}{2}| \leq \mathsf{negl}(\lambda).$$

Theorem 5. *PPTR is secure against the Level-II attack.*

Proof. According to the above analysis, we should prove that $\mathcal{A}$ cannot distinguish $\mathcal{C}_{0,j}$ and $\mathcal{C}_{1,j}$, even though $\mathcal{A}$ has the ability of oracle access to Location Encryption.

Suppose a message in DB_0 is a location point $p_k = (x, y)$. Following the procedure in Location Encryption, p_k is extended to an $(n+8) \times (n+8)$ random lower triangular matrix $\mathcal{P}_k$ with the main diagonal being $(0, 0, 1, x, \ldots, x^n, y, 0, 0, 0, 0)$ and is then encrypted as $\mathcal{C}_k = \mathcal{A}_k \mathcal{P}_k$. According to the law of matrix multiplication, $\mathcal{C}_k$ can be written as

$$\mathcal{C}_k = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,n+6} & 0 & 0 \\ c_{2,1} & c_{2,2} & \cdots & c_{2,n+6} & 0 & 0 \\ c_{3,1} & c_{3,2} & \cdots & c_{3,n+6} & 0 & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ c_{n+7,1} & c_{n+7,2} & \cdots & c_{n+7,n+6} & 0 & 0 \\ c_{n+8,1} & c_{n+8,2} & \cdots & c_{n+8,n+6} & 0 & 0 \end{bmatrix}.$$

Let $\mathbf{A}_i$ be the ith row vector of $\mathcal{A}_k$, $\mathbf{P}_j$ be the jth column vector of $\mathcal{P}_k$, $a_{i,j}$ be the entry in the ith row and jth column of $\mathcal{A}_k$, and $p_{i,j}$ be the entry in the ith row and jth column of $\mathcal{P}_k$, we will have

$$c_{i,j} = \mathbf{A}_i \circ \mathbf{P}_j = a_{i,1}p_{1,j} + a_{i,2}p_{2,j} \cdots + a_{i,n+8}p_{n+8,j}, \tag{8}$$

where

$$\begin{cases} p_{i,j} = *, \ 1 \le j < i \le n+8, i \ne 2, j \ne n+7, \\ p_{i,j} = x^{j-3}, \ \ 3 \le i = j \le n+3, \\ p_{n+4,n+4} = y, \\ p_{i,j} = 0, \ \ \text{otherwise}. \end{cases}$$

Without loss of generality, we assume that (x, y) can be any value selected by $\mathcal{A}$ and then we observe the last one non-zero column vector of $\mathcal{C}_k$, i.e., $\{c_{i,n+6}\}_{i=1}^{n+6}$. From Eq. (8), we can see $c_{i,n+6} = a_{i,n+7}p_{n+7,n+6} + a_{i,n+8}p_{n+8,n+6}$. Since $a_{i,n+7}$, $a_{i,n+8}$ are fixed and $p_{n+7,n+6}, p_{n+8,n+6}$ are one-time random values selected by $\mathcal{C}$, it is obvious that $c_{i,n+6}$ is also a random value and it changes with $p_{n+7,n+6}, p_{n+8,n+6}$. Following the same justification, we can obtain that $\mathcal{C}_k$ is a random matrix and each non-zero element in $\mathcal{C}_k$ contains at least two random values determined by $\mathcal{C}$.

In *Phase 1* and *Phase 2* of the experiment $\mathsf{Level-II}_{\mathcal{A},\mathsf{PPTR}}$, $\mathcal{A}$ is able to select different $p_{i,j}$ each time and observe its corresponding ciphertext $\mathcal{C}_{i,j}$. However, since $\mathcal{P}_{i,j}$ is an one-time random matrix that is determined by $\mathcal{C}$, the ciphertexts look random to $\mathcal{A}$. That is, given a ciphertext encrypted by using the message selected by $\mathcal{A}$, $\mathcal{A}$ cannot distinguish which message is encrypted. Therefore, even $\mathcal{A}$ has the ability of oracle access to Location Encryption, it can only take a random guess b' of b with the probability

$$|\mathsf{Pr}(\mathsf{Level-II}_{\mathcal{A},\mathsf{PPTR}}(1^\lambda) = 1) - \frac{1}{2}| \le \mathsf{negl}(\lambda).$$

□

5.2 Data Re-Encryption

In this subsection, we demonstrate that PPTR can resist against the *Level-II* attack in the phase of data re-encryption. In reality, such an attack may occur when a cloud colludes with several workers or data requesters. Without loss of generality, we assume that a cloud, S_A for example, has the ability of oracle access to Location Encryption, and it can run the *Level-II* attack as shown in Fig. 7. Then, we have the following definition and theorem.

Definition 4. *PPTR is secure against the Level-II attack if for a cloud, it has at most a negligible advantage* $\mathsf{negl}(\lambda)$*, such that*

$$|\mathsf{Pr}(\mathsf{Level-II}_{\mathsf{Cloud},\mathsf{PPTR}}(1^\lambda) = 1) - \frac{1}{2}| \le \mathsf{negl}(\lambda).$$

Theorem 6. *PPTR is secure against the Level-II attack.*

Proof. Suppose there is a ciphertext $\mathcal{C}_i$ that is generated by worker u_i from the location (x, y) via the algorithm Location *Encryption*. As proven in *Theorem 5*, $\mathcal{C}_i$ looks random and thus S_A cannot identify which message is encrypted. With the re-encryption key, S_A obtains $\widetilde{\mathcal{C}}_i = \mathcal{M}_1\mathcal{I}_i(\mathcal{P}_i + \mathcal{I}'_i)\mathcal{D}_i\mathcal{M}_A$ through the algorithm Location Transformation. It is observed that $\mathcal{M}_1, \mathcal{M}_A$ are fixed matrices similar to A_i, and $\mathcal{I}_i, \mathcal{P}_i, \mathcal{I}'_i, \mathcal{D}_i$ are random lower triangular matrices similar to $\mathcal{P}_k$. With similar analysis in *Theorem 5*, we can prove that PPTR is secure against the *Level-II* attack.

Note that, since the master key $\mathcal{M}_1$ is involved in $\widetilde{C}_i$, workers' privacy may be violated if a cloud can obtain the master key. In the following, we demonstrate that a cloud cannot recover the master key from the information it has even if it colludes with a number of workers.

Given a set of workers' encryption keys $\{\mathcal{A}_l\}_{l=1}^t$, S_A can obtain a series of ciphertexts $\mathcal{F}_l = \mathcal{M}_1\mathcal{I}_l$ by calculating $\mathcal{A}'_l\mathcal{A}_l$. Let $f_{i,j}^{(l)}, m_{i,j}, o_{i,j}^{(l)}$ be the entry in ith row and jth column in $\mathcal{F}_l, \mathcal{M}_1, \mathcal{I}_l$ respectively, without loss of generality, we select an element in the last non-zero column, e.g., $f_{1,n+6}^{(l)}$, and we assume $(x, y) = (0, 0)$, then, we can derive

$$\begin{cases} f_{1,n+6}^{(1)} = m_{1,n+7}o_{n+7,n+6}^{(1)} + m_{1,n+8}o_{n+8,n+6}^{(1)}, \\ f_{1,n+6}^{(2)} = m_{1,n+7}o_{n+7,n+6}^{(2)} + m_{1,n+8}o_{n+8,n+6}^{(2)}, \\ \cdots, \\ f_{1,n+6}^{(t)} = m_{1,n+7}o_{n+7,n+6}^{(t)} + m_{1,n+8}o_{n+8,n+6}^{(t)}. \end{cases} \tag{9}$$

In Eq. (9), there are t equations with $2t + 2$ unknowns (i.e., $m_{1,n+7}, m_{1,n+8}, \{o_{n+7,n+6}^{(l)}, o_{n+8,n+6}^{(l)}\}_{l=1}^t$), such that $m_{1,n+7}, m_{1,n+8}$ cannot be determined or even approximately resolved. Thus, S_A can only take a random guess $m'_{1,n+7}, m'_{1,n+8}$ of $m_{1,n+7}, m_{1,n+8}$. Mathematically, if we assume the element in the master key $\mathcal{M}_1$ is randomly selected from $[1, 2^\lambda]$, where λ denotes the bit size (in practice, the element can be integer or decimal), then the probability of identifying the elements $m_{1,n+7}, m_{1,n+8}$ is $\mathsf{Pr}[m'_{1,n+7} = m_{1,n+7}, m'_{1,n+8} = m_{1,n+8}] < \frac{1}{2^\lambda \cdot 2^\lambda} = \frac{1}{2^{2\lambda}}$. Since $\frac{1}{2^{2\lambda}} \to 0$, $\mathsf{Pr}[m'_{1,n+7} = m_{1,n+7}, m'_{1,n+8} = m_{1,n+8}] \to 0$. Hence, we can conclude that the master key $\mathcal{M}_1$ is kept secret. □

5.3 Data Query

With $\widetilde{\mathcal{C}}_i, \overline{\mathcal{T}}_a, \overline{\mathcal{T}}_b$, S_B can obtain the matrices $\Gamma_a = \mathcal{M}_1\mathcal{I}_i(\mathcal{P}_i + \mathcal{I}'_i)\mathcal{D}_i\mathcal{D}_a(\mathcal{Q}_a + \mathcal{S}'_q)\mathcal{S}_q\mathcal{M}_1^{-1}$ and $\Gamma_b = \mathcal{M}_1\mathcal{I}_i(\mathcal{P}_i + \mathcal{I}'_i)\mathcal{D}_i\mathcal{D}_b(\mathcal{Q}_b + \mathcal{S}'_q)\mathcal{S}_q\mathcal{M}_1^{-1}$. Since $\mathcal{M}_1, \mathcal{M}_1^{-1}$ are unknown matrices (as proven in *Theorem 6*) and $\mathcal{I}_i(\mathcal{P}_i + \mathcal{I}'_i)\mathcal{D}_i\mathcal{D}_a(\mathcal{Q}_a + \mathcal{S}'_q)\mathcal{S}_q$, $\mathcal{I}_i(\mathcal{P}_i + \mathcal{I}'_i)\mathcal{D}_i\mathcal{D}_b(\mathcal{Q}_b + \mathcal{S}'_q)\mathcal{S}_q$ are random lower triangular matrices, it is obvious that Γ_a, Γ_b also have CPA-security and S_B cannot derive any sensitive information of l_i, l_a, l_b from Γ_a and Γ_b.

Then, we observe the final results $\Delta_a = tr(\Gamma_a) = v_i v_a(\theta_a^*(x_i) - y_i) + r_{a,1}\gamma_{q,1}\gamma_{i,1} + r_{a,2}\gamma_{q,2}\gamma_{i,2}$, $\Delta_b = tr(\Gamma_b) = v_i v_b(\theta_b^*(x_i) - y_i) + r_{b,1}\gamma_{q,1}\gamma_{i,1} + r_{b,2}\gamma_{q,2}\gamma_{i,2}$. Given a fixed location l_i, it is possible that S_B selects two queries l_1, l_2 such that $\Delta_a > 0$ while $\Delta_b < 0$. In other words, the final results may reveal some information about l_i, l_1, l_2. However, we emphasize that the disclosed information is what we expect since we need to know whether $\theta_a^*(x_i) - y_i$ and $\theta_b^*(x_i) - y_i$ are positive or not from Δ_a and Δ_b. It is possible that S_B may collude with some workers or data requesters. In this case, the values S_B knows are $\theta_a^*(x_i) - y_i, \Delta_a$ and $\theta_b^*(x_i) - y_i, \Delta_b$. Since $\gamma_{i,1}, \gamma_{i,2}, \gamma_{q,1}, \gamma_{q,2}$ are unknown and $v_a, r_{a,1}, r_{a,2}, v_b, r_{b,1}, r_{b,2}$ are one-time random values that are associated with l_a, l_b generated by S_A, the final results Δ_a, Δ_b only reveal the sign of Δ_a, Δ_b and no more sensitive information can be derived by S_B.

TABLE 1
Query Accuracy Comparison of Different Degrees

	degree = 3	degree = 5	degree = 7	degree = 10
Triangle	93.72%	95.82%	98.00%	99.15%
Circle	98.11%	99.29%	99.64%	99.80%
Rectangle	93.87%	97.03%	98.21%	99.12%
Closed curve	83.09%	95.17%	99.36%	99.99%

6 Performance Analysis

In this section, we numerically analyze the complexity of our schemes and implement our schemes as a geometric range query system. Specifically, we conduct experiments on a laptop and an Android phone, and implement our schemes in Python with Numpy library and matplotlib library. The laptop with 2.3 GHz, Intel Core i5, 16 GB RAM is used as the clouds, and the Android phone with 4G RAM, Kirin 659 processor is used as the worker and data requester, respectively. We randomly generate longitude and latitude coordinates as workers' location data, and the number of encrypted locations ranges from 1×10^5 to 1×10^6. For the geometric range query, we randomly generate two curvilinear equations to cover several points and select the coefficients of the curvilinear equations as the query information. In comparison, a most recent SSE-based spatial keywords scheme PBRQ-L [17] supporting arbitrary geometric range query, a privacy-preserving task assignment scheme ETA [15] supporting circle-based query, and a randomizable matrix based location-aware task recommendation scheme PPTA [3] supporting arbitrary sensing area query, are also implemented.

6.1 Query Accuracy

We first analyze the accuracy of our schemes. As described in Sections 3 and 4, the polynomial fitting technique is adopted to generate the location vector and trapdoor. However, since the polynomial fitting technique is an approximation algorithm, fitting errors will be inevitably introduced. In this part, we conduct a series of experiments to analyze the query accuracy, i.e., fitting accuracy, by selecting different query ranges, including a triangle, a circle, a rectangle, and a random closed curve. For the datasets, we assume the range of spatial data is within [0,1000]. In Table 1, we show the precision of different degrees under different query ranges. It can be observed that with the increase of the degree, the query accuracy becomes higher regardless of the query ranges. When the degree is 10, the query accuracy can reach 99 percent or even more. To show the query accuracy clearly, we also plot the original ranges and the polynomial fitting curves, which can be seen in Figs. 8, 9, 10, and 11.

6.2 Theoretical Analysis

We then analyze the complexities of PPTR-L and PPTR-F in terms of computational, communication, and storage overhead before providing experimental results, which are summarized in Table 2. For clear description, we assume the time cost of each matrix multiplication is $\mathsf{T_M}$, which is equal to $O((n+8)^3)$, and the size of a matrix is $|\mathbf{M}|$. To encrypt the location information, a worker needs to perform 1 matrix multiplication in Location Encryption and the clouds need to perform 3 matrix multiplications in Location Transformation for both PPTR-L and PPTR-F, which costs $\mathsf{T_M}$ and $3\mathsf{T_M}$, respectively. In the phase of Trapdoor Generation, to generate a search trapdoor, PPTR-L needs to perform 2 matrix multiplications on the data requester side, while 2 more matrix multiplications are needed in PPTR-F which is caused by the additional generation of the search circle trapdoor and the father node. Accordingly, PPTR-L and PPTR-F need 8 and 15 matrix multiplications in Trapdoor Transformation, respectively. In the phase of Worker Query, each trace calculation causes 1 matrix multiplication. Thus, to process a geometric

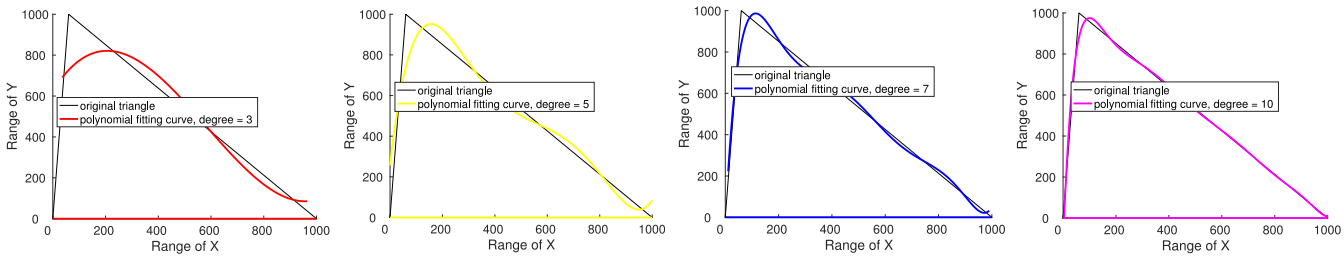


Fig. 8. Query accuracy for triangular range. (a) degree = 3; (b) degree = 5; (c) degree = 7; (d) degree = 10.

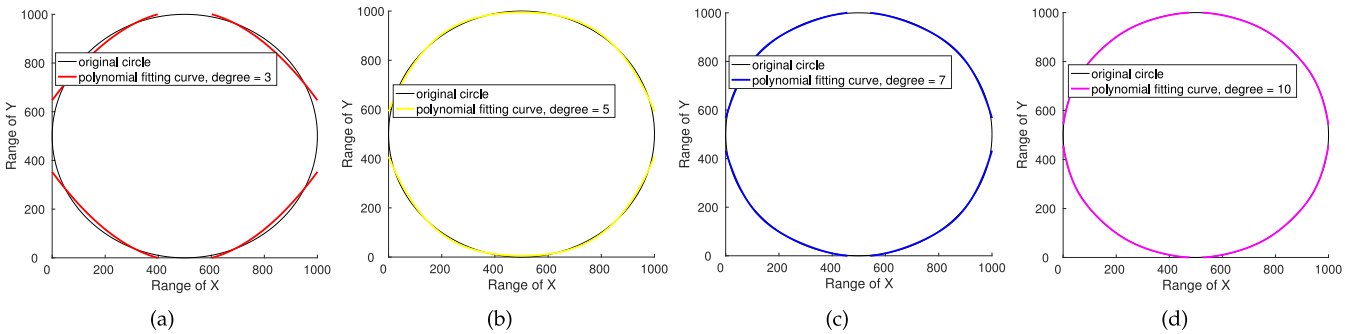


Fig. 9. Query accuracy for circular range. (a) degree = 3; (b) degree = 5; (c) degree = 7; (d) degree = 10.

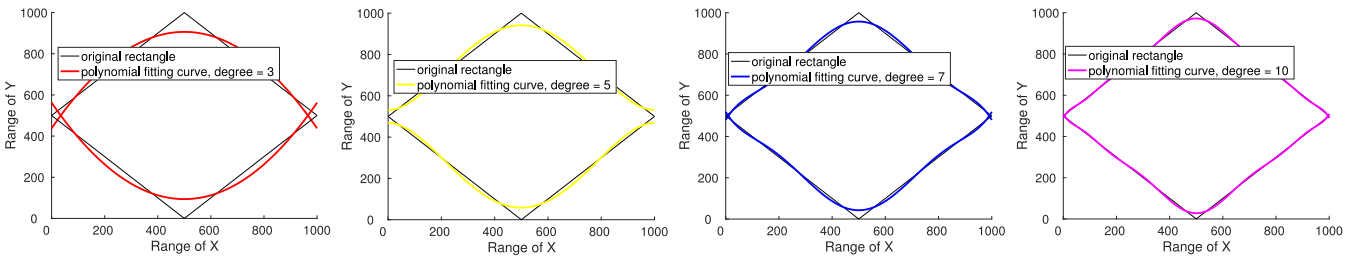


Fig. 10. Query accuracy for rectangular range. (a) degree = 3; (b) degree = 5; (c) degree = 7; (d) degree = 10.

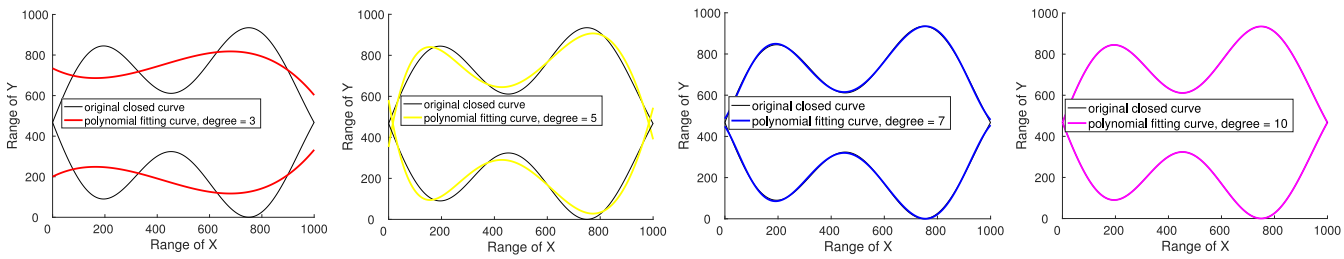


Fig. 11. Query accuracy for a random closed curve range. (a) degree = 3; (b) degree = 5; (c) degree = 7; (d) degree = 10.

range query, the maximum time cost of Worker Query in PPTR-L is $2m\mathsf{T_M}$, where m denotes the number of encrypted location data. For PPTR-F, since a new data structure is designed to speed up the search process, with more father nodes, fewer matrix multiplications will be performed. Suppose there are K father nodes and among them, there are e father nodes that have interaction with the search circle trapdoor, the cloud needs to perform $2(m-(K-e)z)+K$ matrix multiplications, where z denotes the number of children nodes in a father node and $(m-Kz)$ is the number of nodes that have not been added in father nodes.

In Table 3, we compare the computational costs and communication overhead of our schemes with the schemes [3], [15], [17]. The scheme [17] is based on the symmetric key encryption SHVE [30] and it adopts a vector containing w bits to denote a user's location. We use $\mathsf{T_{PRF}}$ to denote the time cost to compute a pseudorandom function (PRF), $\mathsf{T_{XOR}}$ to denote the time cost to perform an exclusive-or operation, $\mathsf{T_{Enc}}$ to denote the time cost to perform symmetric encryption, $\mathsf{T_{Dec}}$ to denote the time cost to perform symmetric decryption, and $|\mathsf{PRF}|$ to denote the size of the PRF output. The scheme [15] is based on Paillier cryptosystem and it uses the longitude and latitude to denote a user's location. We use $\mathsf{P_{Enc}}$ to denote the time cost to perform Paillier encryption, $\mathsf{P_{Dec}}$ to denote the time cost to perform Paillier decryption, and use $|\mathsf{Paillier}|$ to denote the size of the Paillier ciphertext. The scheme [3] is based on randomizable matrix multiplication and it uses an element in the matrix to denote a geographic region. The dimension of the matrix is equal to $\sqrt{W}$, where W denotes the total number of regions. We use $\widetilde{\mathsf{T}}_{\mathsf{M}}$ to denote the time cost to perform a matrix multiplication and use $|\widetilde{\mathsf{M}}|$ to denote the size of the matrix. The number of regions a data requester requests is assumed to be k. Note that, the schemes [15], [17] are based on symmetric key encryption or homomorphic Paillier encryption, while our schemes are based on randomizable matrix multiplication, which can avoid expensive cryptographic operations. Although the scheme [3] is also based on randomizable matrix multiplication, it needs more time to perform data encryption and data query, since the dimension of the

TABLE 2
A Summary of Theoretical Analysis

Scheme	Phase	Entity	Computation overhead	Communication overhead	Storage overhead
PPTR-L	Location Encryption	Worker	$\mathsf{T_M}$	$\lvert\mathsf{M}\rvert$	–
	Location Transformation	Clouds	$3\mathsf{T_M}$	$\lvert\mathsf{M}\rvert$	$\lvert\mathsf{M}\rvert$
	Trapdoor Generation	Data requester	$2\mathsf{T_M}$	$2\lvert\mathsf{M}\rvert$	–
	Trapdoor Transformation	Clouds	$8\mathsf{T_M}$	$2\lvert\mathsf{M}\rvert$	–
	Worker Query	Clouds	$2m\mathsf{T_M}$	–	–
PPTR-F	Location Encryption	Worker	$\mathsf{T_M}$	$\lvert\mathsf{M}\rvert$	–
	Location Transformation	Clouds	$3\mathsf{T_M}$	$\lvert\mathsf{M}\rvert$	$\lvert\mathsf{M}\rvert$
	Trapdoor Generation	Data requester	$4\mathsf{T_M}$	$4\lvert\mathsf{M}\rvert$	–
	Trapdoor Transformation	Clouds	$15\mathsf{T_M}$	$4\lvert\mathsf{M}\rvert$	–
	Worker Query	Clouds	$(2(m-(K-e)z)+K)\mathsf{T_M}$	–	$\lvert\mathsf{M}\rvert$

TABLE 3
A Comparison of Theoretical Analysis

Scheme	Phase	Entity	Computation overhead	Communication overhead	Storage overhead
PPTR-L	Location Encryption	Worker	$\mathsf{T_M}$	$\lvert\mathsf{M}\rvert$	–
	Trapdoor Generation	Data requester	$2\mathsf{T_M}$	$2\lvert\mathsf{M}\rvert$	–
	Worker Query	Clouds	$2m\mathsf{T_M}$	–	–
PPTR-F	Location Encryption	Worker	$\mathsf{T_M}$	$\lvert\mathsf{M}\rvert$	–
	Trapdoor Generation	Data requester	$4\mathsf{T_M}$	$4\lvert\mathsf{M}\rvert$	–
	Worker Query	Clouds	$(2(m-(K-e)z)+K)\mathsf{T_M}$	–	$\lvert\mathsf{M}\rvert$
PBRQ-L [17]	Location Encryption	Worker	$w\mathsf{T_{PRF}}$	$w\lvert\mathsf{PRF}\rvert$	$w\lvert\mathsf{PRF}\rvert$
	Trapdoor Generation	Data requester	$w\mathsf{T_{PRF}}+\mathsf{T_{Enc}}$	$\lvert\mathsf{PRF}\rvert$	–
	Worker Query	Cloud	$m(w\mathsf{T_{XOR}}+\mathsf{T_{Dec}})$	–	–
ETA [15]	Location Encryption	Worker	$2\mathsf{P_{Enc}}$	$2\lvert\mathsf{Paillier}\rvert$	$2\lvert\mathsf{Paillier}\rvert$
	Trapdoor Generation	Data requester	$2\mathsf{P_{Enc}}$	$2\lvert\mathsf{Paillier}\rvert$	–
	Worker Query	Clouds	$m(5\mathsf{P_{Enc}}+\mathsf{P_{Dec}})$	$(10m+4)\lvert\mathsf{Paillier}\rvert$	–
PPTA [3]	Location Encryption	Worker	$\widetilde{\mathsf{T}}_{\mathsf{M}}$	$\lvert\widetilde{\mathsf{M}}\rvert$	$\lvert\widetilde{\mathsf{M}}\rvert$
	Trapdoor Generation	Data requester	$k\widetilde{\mathsf{T}}_{\mathsf{M}}$	$k\lvert\widetilde{\mathsf{M}}\rvert$	–
	Worker Query	Cloud	$km\mathsf{T_M}$	–	–

matrix in their scheme grows linearly with the number of geometric regions. The performance of these schemes will be evaluated in Section 6.3.

6.3 Performance Evaluation

Setup and Implementation. In the scheme [17], Quadtree and Gray code are used to represent a worker's location and the technique of Symmetric-key Hidden Vector Encryption [30] is leveraged to protect location privacy. To support arbitrary geometric range query, the size of Gray code needs to be set large enough. Similar to [17], we set the length of Gray code as 100 and the size of symmetric key is 128 bits. In the scheme [15], we implement the Paillier cryptosystem-based scheme, i.e., LATE mechanism, and set the secret key as 512 bits. In the scheme [3], the number of total geographic regions, i.e., W, is assumed to be 10000, and the number of regions a data requester requires, i.e., k, is assumed to be 1. According to the query accuracy analyzed in Section 6.1, the degree of the polynomial fitting function in our schemes is set as 10. Each element in the matrix is set as $[1, 2^{32}]$. The number of encrypted locations ranges from 1×10^5 to 1×10^6. The phases of Location Encryption, Trapdoor Generation are implemented on the smartphone, while other phases are implemented on the laptops.

Experimental Results. Figs. 12a and 12b show the time costs of Location Encryption, Trapdoor Generation for all schemes. It is observed that our schemes are more efficient than the schemes [3], [15], [17]. This is because (1) our schemes are implemented based on matrix multiplication, while the schemes [15], [17] rely on the cryptographic operations; (2) our schemes enable the dimension of location data to be stable thanks to the adoption of the polynomial fitting technique, while the size of data needed to be encrypted in the schemes [3], [17] depends on the search scope. Figs. 12c and 12d show the running time of Location Transformation, Trapdoor Transformation. We can see from Fig. 12d that the running time of PPTR-F is more than that of PPTR-L, which is caused by additional re-encryption of the search circle trapdoor and the father node. In Fig. 12e, we plot the running time of Worker Query and the number of encrypted locations m ranges from 2×10^5 to 1×10^6. For a single geometric query, not surprisingly, the query time of PPTR-L increases linearly with the number of encrypted locations, which confirms the complexity analysis in Table 2. As described in Section 4.3, a new data structure is designed in PPTR-F to speed up the search time. To better describe the influence of this data structure, we use $\eta=\frac{z*K}{m}$ to denote the percentage of workers that are in father nodes. Especially, we assume $z=100, e=1$, and $\eta\in[25\%, 100\%]$. As demonstrated in Fig. 12e, when $\eta=100\%$, i.e., all workers are in father nodes, PPTR-F can save about 98 percent time cost compared with PPTR-L. When selecting different η, as shown in Fig. 12f, the search time in PPTR-F is inversely proportional to the value of η, which however is still more efficient than PPTR-L. As a result, we demonstrate that PPTR-F is a *faster-than-linear* scheme and the more search operations are conducted, the higher efficiency will be achieved. We also compare the computational costs between our schemes and the schemes [3], [15], [17] in the phase of Worker Query. It is observed from Table 4 that our schemes are extremely efficient. Especially, the query time of our linear scheme, i.e., PPTR-L, is faster than PBRQ-L and is significantly more efficient than the other two public key-based schemes. The communication overhead between the participating parties is summarized in Table 5.

7 Related Work

In this section, we briefly review some existing works related to privacy-preserving task recommendation and privacy-preserving geometric range query.

7.1 Privacy-Preserving Task Recommendation

Task recommendation is a key issue in mobile crowdsensing, and its security and privacy issues have been widely investigated recently. By employing techniques of Paillier cryptosystem and Yao's garbled circuits, Liu *et al.* [15] proposed a task assignment scheme to find the nearest worker to the published

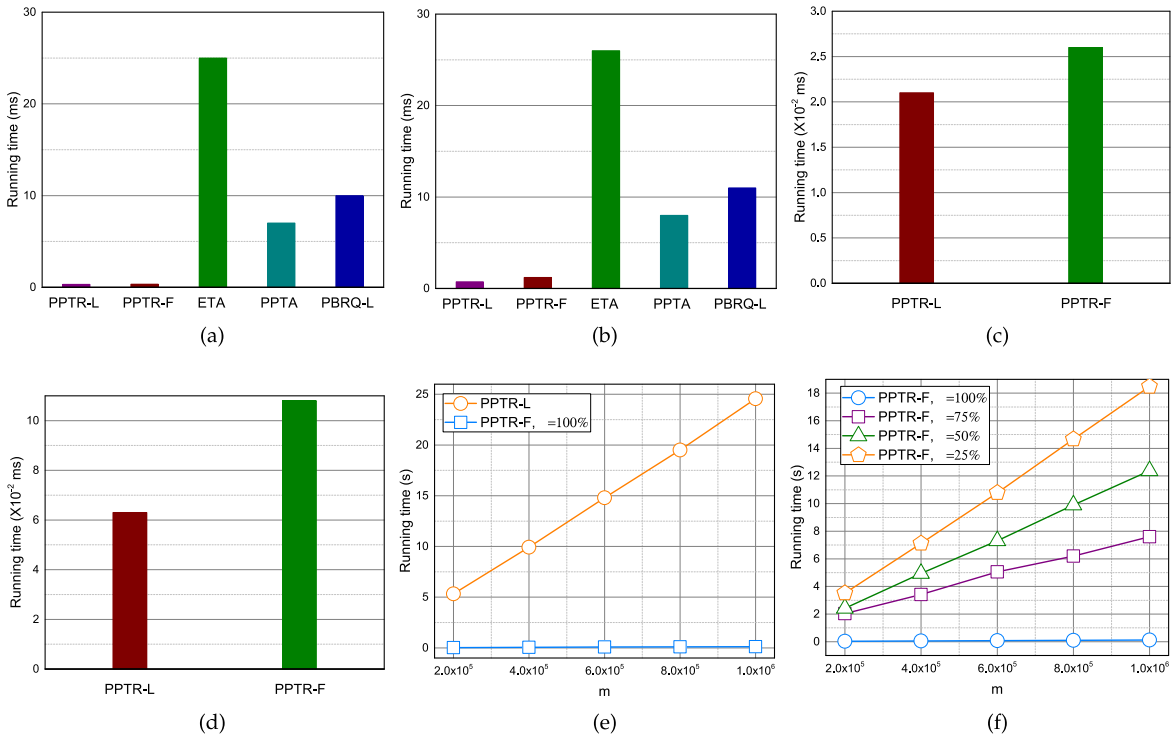


Fig. 12. Experimental results. (a) The computational costs of Location Encryption; (b) The computational costs of Trapdoor Generation; (c) The computational costs of Location Transformation; (d) The computational costs of Trapdoor Transformation; (e) The computational costs of Worker Query; (f) The computational costs of Worker Query in PPTR-F with different η.

task. Based on this scheme, Zhai *et al.* [31] proposed a novel secure and truthful task assignment scheme. To protect the location privacy, they used travel costs to represent the location and designed an anonymity-based data aggregation protocol to prevent the service provider from inferring a worker's actual location. Wang *et al.* [32] proposed a privacy-preserving task allocation scheme to select users having the shortest travel distances to different tasks. By leveraging the technique of differential privacy, each participant can obfuscate his location without the need of a trusted database owner. Using the same technique, Wang *et al.* [33] proposed a personalized location privacy-preserving task allocation scheme. Different from [32], their scheme required each worker to upload the obfuscated distance as well as the personal privacy budget. With a designed Probabilistic Winner Selection Mechanism (PWSM) protocol, their scheme can allocate tasks to workers that are more likely to be closest to the tasks. In addition to the above location privacy-preserving task recommendation schemes, there are some other privacy-preserving task recommendation schemes [3], [20], [34], [35], [36], [37], [38] that focus on the privacy of tasks or workers' trustworthiness.

7.2 Privacy-Preserving Geometric Range Query

As a fundamental search functionality of the spatial database, geometric range query has acquired increasing attention, and there have been many works on privacy-preserving geometric range query. Specifically, Zhu *et al.* [13] utilized the BGN cryptosystem to protect the location privacy and applied hash tables to evaluate the results range. Nevertheless, due to the time-consuming homomorphic operations, lots of computational costs are introduced. To improve query efficiency, Wang *et al.* presented several schemes [11], [12], [16] by leveraging techniques, such as Shen-Shi-Waters (SSW) encryption [39].

TABLE 4
Query Time Comparison Among the Schemes, Where h Denotes Hour and s Denotes Second

m	1×10^5	2×10^5	3×10^5	4×10^5	5×10^5
ETA [15]	0.79 h	1.69 h	2.63 h	3.42 h	4.11 h
PPTA [3]	49.78 s	99.39 s	148.45 s	202.04 s	286.35 s
PBRQ-L [17]	3.18 s	6.36 s	9.89 s	12.85 s	15.73 s
PPTR-L	2.72 s	5.32 s	7.44 s	9.93 s	12.46 s
PPTR-F (25%)	1.83 s	3.52 s	5.31 s	7.11 s	8.92 s
PPTR-F (50%)	1.23 s	2.42 s	3.58 s	4.94 s	6.06 s
PPTR-F (75%)	1.03 s	2.03 s	2.75 s	3.41 s	4.30 s
PPTR-F (100%)	0.02 s	0.04 s	0.06 s	0.08 s	0.10 s

TABLE 5
Communication Overhead Comparison Among the Schemes, Where L.E Denotes Location Encryption, L.T Denotes Location Transformation, T.G Denotes Trapdoor Generation, and T.T Denotes Trapdoor Transformation

	L.E	L.T	T.G	T.T
	u_i-S_A	$S_A - S_B$	u_q-S_A	S_A-S_B
ETA [15]	0.5 KB	-	0.5 KB	-
PPTA [3]	39.8 KB	-	39.8 KB	-
PBRQ-L [17]	1.56 KB	-	0.02 KB	-
PPTR-L	1.29 KB	1.29 KB	2.58 KB	2.58 KB
PPTR-F	1.29 KB	1.29 KB	5.16 KB	5.16 KB

TABLE 6
Comparison With Prior Works

Scheme	No database owner	Search method	Faster-than-linear	Security	Performance
PPTA [3]	✓	Arbitrary	×	IND-CPA	High
EGRQ [10]	×	Arbitrary	✓	KBA	Very High
CRSE [11]	×	Circle	×	IND-CPA	Low
GRSE [12]	×	Arbitrary	✓	IND-CPA	Low
SRQC [13]	×	Circle	×	IND-CPA	High
EPPD [14]	✓	Circle	×	IND-CPA	High
ETA [15]	✓	Circle	×	IND-CPA	Low
FastGeo [16]	×	Arbitrary	✓	IND-CPA	High
PBRQ-L [17]	×	Arbitrary	×	IND-CPA	High
PPTR-L	✓	Arbitrary	×	IND-CPA	High
PPTR-F	✓	Arbitrary	✓	IND-CPA	Very high

Wang *et al.* [11] utilized circles to represent queries and performed geometric range query by evaluating whether a point insides the constructed circles, which however cannot support arbitrary geometric range query. To address this shortcoming, Wang *et al.* [12] adopted the Bloom filter to represent workers' spatial data and the geometric range query, and decide whether a worker is located in the query range by calculating the inner product of two Bloom filters. However, with the increase of search scope, a large Bloom filter is required and lots of computational costs will be introduced. Then, Wang *et al.* [16] improved the query efficiency and accuracy by converting workers' spatial data and queries to an equality-vector form. Xu *et al.* [10] leveraged the techniques of polynomial fitting and secure kNN [40] to realize efficient and privacy-preserving arbitrary geometric range query with access control [41]. The above *database owner*-centric schemes will incur the following two significant shortcomings: (1) user privacy may be violated since all search users hold the same secret key; (2) user scalability is sacrificed, as their key-sharing methods need to update the secret key and encrypted spatial database after every user revocation, leading to high computational and communication overhead.

Different from the above works, we aim to design an efficient and privacy-preserving task recommendation scheme that can support efficient and arbitrary geometric range query and without any trusted database owner. A comparison of our PPTR schemes with prior related schemes is given in Table 6.

8 Conclusion

In this paper, we have proposed two novel privacy-preserving task recommendation (PPTR) schemes for mobile crowdsensing. For the first scheme, named PPTR-L, it enables the service provider to find the workers that are inside a data requester's arbitrary query range, while not disclosing the sensitive location information of both workers and data requesters. To further improve efficiency, by considering data requesters' historical search behaviors, we have designed an enhanced scheme, named PPTR-F, to achieve faster-than-linear search complexity. Our schemes are highly practical for real-world mobile crowdsensing applications, since they do not rely on the trusted database owner to handle the location data and do not require heavy cryptographic operations. For the future work, in addition to the location information, we will consider workers' interests and design a spatial keyword-based privacy-preserving task recommendation scheme.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grants 61972037, 61402037, 61872041, and U1836212, the Natural Sciences and Engineering Research Council (NSERC) of Canada, and the China Scholarship Council.

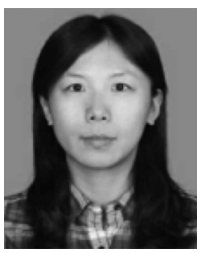
References

[1] Y. Li *et al.*, "Ptasim: Incentivizing crowdsensing with POI-tagging cooperation over edge clouds," *IEEE Trans. Ind. Inform.*, vol. 16, no. 7, pp. 4823–4831, Jul. 2019.
[2] C. Miao, L. Su, W. Jiang, Y. Li, and M. Tian, "A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems," in *Proc. IEEE INFOCOM, Conf. Comput. Commun.*, pp. 1–9, 2017.
[3] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 19, no. 6, pp. 1317–1331, Jun. 2019.
[4] Y. Hui, Z. Su, and S. Guo, "Utility based data computing scheme to provide sensing service in Internet of Things," *IEEE Trans. Emerging Top. Comput.*, vol. 7, no. 2, pp. 337–348, Apr.–Jun. 2019.
[5] L. Zhu, C. Zhang, C. Xu, X. Du, N. Guizani, and K. Sharif, "Traffic monitoring in self-organizing vanets: A privacy-preserving mechanism for speed collection and analysis," *IEEE Wirel. Commun.*, vol. 26, no. 6, pp. 18–23, Dec. 2019.
[6] J. Liang, Z. Qin, S. Xiao, J. Zhang, H. Yin, and K. Li, "Privacy-preserving range query over multi-source electronic health records in public clouds," *J. Parallel Distrib. Comput.*, vol. 135, pp. 127—–139, Jan. 2020.
[7] J. Liang, Z. Qin, S. Xiao, L. Ou, and X. Lin, "Efficient and secure decision tree classification for cloud-assisted online diagnosis services," *IEEE Trans. Dependable Secure Comput.*, early access, Jun. 14, 2019, doi: 10.1109/TDSC.2019.2922958.
[8] Z. Zhao, H. Lu, D. Cai, X. He, and Y. Zhuang, "User preference learning for online social recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2522–2534, Sep. 2016.
[9] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: A survey," *VLDB J.*, vol. 29, no. 1, pp. 217–250, 2020.

[10] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 4, pp. 870–885, Apr. 2018.
[11] B. Wang, M. Li, H. Wang, and H. Li, "Circular range search on encrypted spatial data," in *Proc. Int. Conf. Distrib. Comput. Syst.*, 2015, pp. 182–190.
[12] B. Wang, M. Li, and H. Wang, "Geometric range search on encrypted spatial data," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 4, pp. 704–719, Apr. 2016.
[13] H. Zhu, R. Lu, C. Huang, L. Chen, and H. Li, "An efficient privacy-preserving location-based services query scheme in outsourced cloud," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7729–7739, Sep. 2016.
[14] C. Huang, R. Lu, H. Zhu, J. Shao, A. Alamer, and X. Lin, "EPPD: Efficient and privacy-preserving proximity testing with differential privacy techniques," in *Proc. IEEE Int. Conf. Commun.*, 2016, pp. 1–6.
[15] A. Liu *et al.*, "Privacy-preserving task assignment in spatial crowdsourcing," *J. Comput. Sci. Technol.*, vol. 32, no. 5, pp. 905–918, 2017.
[16] B. Wang, M. Li, and L. Xiong, "FastGeo: Efficient geometric range queries on encrypted spatial data," *IEEE Trans. Dependable Secure Comput.*, vol. 16, no. 2, pp. 245–258, Mar./Apr. 2019.
[17] X. Wang *et al.*, "Search me in the dark: Privacy-preserving boolean range query over encrypted spatial data," in *Proc. INFOCOM Conf. Comput. Commun.*, 2020, pp. 2253–2262.
[18] X. Zhang, R. Lu, J. Shao, H. Zhu, and A. A. Ghorbani, "Secure and efficient probabilistic skyline computation for worker selection in MCS," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11524–11535, Dec. 2020.
[19] N. Cui, J. Li, X. Yang, B. Wang, M. Reynolds, and Y. Xiang, "When geo-text meets security: Privacy-preserving boolean spatial keyword queries," in *Proc. IEEE 35th Int. Conf. Data Eng.*, 2019, pp. 1046–1057.
[20] Y. Zheng, R. Lu, Y. Guan, J. Shao, and H. Zhu, "Achieving efficient and privacy-preserving exact set similarity search over encrypted data," *IEEE Trans. Dependable Secure Comput.*, early access, Jun. 23, 2020, doi:10.1109/TDSC.2020.3004442.
[21] K. Xue, S. Li, J. Hong, Y. Xue, N. Yu, and P. Hong, "Two-cloud secure database for numeric-related SQL range queries with privacy preserving," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 7, pp. 1596–1608, Jul. 2017.
[22] X. Liu, B. Qin, R. H. Deng, R. Lu, and J. Ma, "A privacy-preserving outsourced functional computation framework across large-scale multiple encrypted domains," *IEEE Trans. Comput.*, vol. 65, no. 12, pp. 3567–3579, Dec. 2016.
[23] C. Zhang, L. Zhu, C. Xu, X. Liu, and K. Sharif, "Reliable and privacy-preserving truth discovery for mobile crowdsensing systems," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1245–1260, May/Jun. 2021.
[24] J. Xu, K. Xue, Q. Yang, and P. Hong, "PSAP: Pseudonym-based secure authentication protocol for NFC applications," *IEEE Trans. Consumer Electron.*, vol. 64, no. 1, pp. 83–91, Feb. 2018.
[25] Q. Yang, K. Xue, J. Xu, J. Wang, F. Li, and N. Yu, "AnFRA: Anonymous and fast roaming authentication for space information network," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 2, pp. 486–497, Feb. 2018.
[26] C. Piro, C. Shields, and B. N. Levine, "Detecting the sybil attack in mobile Ad hoc networks," in *Proc. Securecomm Workshops*, 2016, pp. 1–11.
[27] Y. Yao *et al.*, "Multi-channel based sybil attack detection in vehicular ad hoc networks using RSSI," *IEEE Trans. Mobile Comput.*, vol. 18, no. 2, pp. 362–375, Feb. 2019.
[28] A. Vasudeva and M. Sood, "Survey on sybil attack defense mechanisms in wireless ad hoc networks," *J. Netw. Comput. Appl.*, vol. 120, pp. 78–118, 2018.
[29] P. Strobach, "Solving cubics by polynomial fitting," *J. Comput. Appl. Math.*, vol. 235, no. 9, pp. 3033–3052, 2011.
[30] S. Lai *et al.*, "Result pattern hiding searchable encryption for conjunctive queries," in *Proc. ACM SIGSAC Conf. Comput. Commun.*, 2018, pp. 745–762.
[31] D. Zhai *et al.*, "Towards secure and truthful task assignment in spatial crowdsourcing," *World Wide Web*, vol. 22, no. 5, pp. 2017–2040, 2019.
[32] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proc. Int. Conf. World Wide Web*, 2017, pp. 627–636.
[33] Z. Wang *et al.*, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1330–1341, Jun. 2018.
[34] J. Zhang, Q. Zhang, and S. Ji, "A fog-assisted privacy-preserving task allocation in crowdsourcing," *IEEE Internet Things J.*, vol. 7, no. 9, pp. 8331–8342, Sep. 2020.
[35] J. Shu, X. Liu, X. Jia, K. Yang, and R. H. Deng, "Anonymous privacy-preserving task matching in crowdsourcing," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3068–3078, Aug. 2018.
[36] J. Shu, X. Jia, K. Yang, and H. Wang, "Privacy-preserving task recommendation services for crowdsourcing," *IEEE Trans. Serv. Comput.*, vol. 14, no. 1, pp. 235–247, Jan./Feb. 2021.
[37] J. Hao, C. Huang, G. Chen, M. Xian, and X. Shen, "Privacy-preserving interest-ability based task allocation in crowdsourcing," in *Proc. IEEE Int. Conf. Commun.*, 2019, pp. 1–6.
[38] W. Tang, K. Zhang, J. Ren, Y. Zhang, and X. Shen, "Privacy-preserving task recommendation with win-win incentives for mobile crowdsourcing," *Inform. Sci.*, vol. 527, pp. 477–492, 2020.
[39] E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in *Proc. Theory Cryptogr. Conf.*, pp. 457–473, 2009.
[40] D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proc. IEEE Symp. Secur. Privacy*, 2000, pp. 44–55.
[41] Y. Miao, R. Deng, K.-K. R. Choo, X. Liu, and H. Li, "Threshold multi-keyword search for cloud-based group data sharing," *IEEE Trans. Cloud Comput.*, early access, Jun. 3, 2020, doi: 10.1109/TCC.2020.2999775.

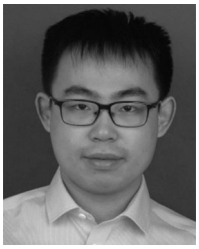
Chuan Zhang (Student Member, IEEE) received the bachelor's degree in network engineering from the Dalian University of Technology, Dalian, China, in 2015. He is currently working toward the PhD degree at the School of Computer Science and Technology, Beijing Institute of Technology. His current research interests include secure data services in cloud computing, security and privacy in IoT, and Big Data security.

Liehuang Zhu (Member, IEEE) received the PhD degree in computer science from the Beijing Institute of Technology, Beijing, China, in 2004. He is currently a professor at the School of Cyberspace Science and Technology, Beijing Institute of Technology. His research interests include security protocol analysis and design, group key exchange protocols, wireless sensor networks, and cloud computing.



Chang Xu received the PhD degree in computer science from Beihang University, Beijing, China, in 2013. She is currently an associate professor at the School of Cyberspace Science and Technology, Beijing Institute of Technology. Her research interests include security and privacy in VANET, and Big Data security.

Jianbing Ni (Member, IEEE) received the BE and ME degrees from the School of Computer Science and Technology, University of Electronic Science and Technology of China, China, in 2011 and 2014, respectively, and the PhD degree in 2018. He is currently an assistant professor at the Department of Electrical and Computer Engineering, Queen's University. From September 2018 to June 2019, he was a postdoctoral fellow with the Department of Electrical and Computer Engineering, University of Waterloo. His research interest includes blockchain, privacy-preserving machine learning, and applied cryptography.



Cheng Huang (Member, IEEE) received the PhD degree in electrical and computer engineering from the University of Waterloo, in 2020. He is currently a postdoctoral research fellow at the Department of Electrical and Computer Engineering, University of Waterloo. His research interests include applied cryptography, cyber security, and privacy in the mobile network.

Xuemin Shen (Fellow, IEEE) received the PhD degree in electrical engineering from Rutgers University, New Brunswick, NJ, USA, in 1990. He is currently a university professor at the Department of Electrical and Computer Engineering, University of Waterloo, Canada. His research interests include network resource management, wireless network security, social networks, 5G and beyond, and vehicular ad hoc and sensor networks. He is currently a registered professional Engineer of Ontario, Canada, an Engineering institute of Canada fellow, a Canadian Academy of Engineering fellow, a Royal Society of Canada fellow, a Chinese Academy of Engineering foreign fellow, and a distinguished lecturer of the IEEE Vehicular Technology Society and Communications Society. He was the recipient of the R.A. Fessenden Award in 2019 from the IEEE, Canada, James Evans Avant Garde Award in 2018 from the IEEE Vehicular Technology Society, Joseph LoCicero Award in 2015, Education Award in 2017 from the IEEE Communications Society, the Excellent Graduate Supervision Award in 2006, five times Outstanding Performance Award from the University of Waterloo, and the Premier's Research Excellence Award (PREA) in 2003 from Ontario, Canada. He was the Technical Program Committee chair or co-chair fof the IEEE Globecom'16, the IEEE Infocom'14, the IEEE VTC'10 Fall, the IEEE Globecom'07, the Symposia Chair for the IEEE ICC'10, the Tutorial Chair for the IEEE VTC'11 Spring, and the Chair of the IEEE Communications Society Technical Committee on Wireless Communications. He is currently the editor-in-chief of the *IEEE Internet of Things Journal* and the vice president of Publications of the IEEE Communications Society.