Hindawi
Security and Communication Networks
Volume 2023, Article ID 9477320, 13 pages
https://doi.org/10.1155/2023/9477320

Research Article

pMATE: A Privacy-Preserving Map Retrieval Task Assignment Scheme in Spatial Crowdsourcing

Peicong He, Yang Xin, Zhengwen Li, and Yixian Yang

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China

Correspondence should be addressed to Peicong He; hepeicong@bupt.edu.cn and Yang Xin; yangxin@bupt.edu.cn

Received 4 August 2022; Revised 23 November 2022; Accepted 7 December 2022; Published 30 January 2023

Academic Editor: Youwen Zhu

Spatial crowdsourcing (SC) task assignment is to find the optimal worker for the task from abundant alternative workers based on the information of the task and workers, such as location, time, and ability. This information will undoubtedly reveal the privacy of both the task and workers. The disclosure of private information is a crucial issue constraining the development of SC. To this end, various privacy-preserving task assignments have been proposed to protect privacy by obfuscating or encrypting information. Fuzzy processing will limit matching accuracy, while encrypted information will increase the time cost of data computation. Therefore, this paper proposes a privacy-preserving map retrieval task assignment scheme (pMATE), which can divide the map and accurately retrieve the optimal workers according to this division. In pMATE, relevant information about tasks and workers is encrypted, and neighboring workers are searched based on the task presence partition. The task location can also be hidden in that partition. Partitioned retrieval reduces the amount of encrypted data needed to be matched. Furthermore, to reduce the problem of multiple communications during encrypted data comparison, we propose the Find MinNumber (FMN) algorithm, which can determine the optimal worker or top-*k* optimal workers need only two communications. Experimental evaluations of real-world data show that pMATE is efficient and accurate.

1. Introduction

With the increasingly powerful functions of intelligent mobile terminals as well as the convenience and high speed of network access, spatial crowdsourcing (SC) [1, 2] as a new business cooperation model, becomes more and more popular and is widely used in urban services and data collection [3–5], such as Uber, Didi, OpenStreetMap, etc. There are three characteristics in SC tasks, respectively, the location, the required capabilities, and the deadline. In other words, SC tasks are based on a specific space scope. They need to be completed within a specified time, for example, picking up passengers from Xizhimen subway station at 3 pm on September 5 and taking them to Tiananmen Square, or taking a picture of Tiananmen Square at sunset on September 6 and recording the sunset time. The total revenue of Meituan takeaway and Didi was 96.3 billion yuan and 173.8 billion yuan, respectively, in the Chinese market in 2021. The vast market and broad application prospects have turned SC into a research hotspot.

As depicted in Figure 1, there are three components in SC, task requesters (TRs), task workers (TWs), and spatial crowdsourcing server (SC-server). TRs are the initiators of SC tasks, searching for a suitable worker or workers to help her/him accomplish the task. TWs are SC task completers looking for a competent task to complete to earn a paycheck. TRs and TWs are SC-server users, an online service platform for matching TRs and TWs with some rules. The typical task assignment strategies are considered ability, distance, and time. That is, whether the ability to complete the task is available, whether it can be completed within the specified time, and the worker's distance from the task point. Therefore, TRs and TWs must upload their information to the SC-server to accomplish the matches. Although the uploaded pieces of information make task assignments more accessible and accurate, it also increases the risk of leakage of

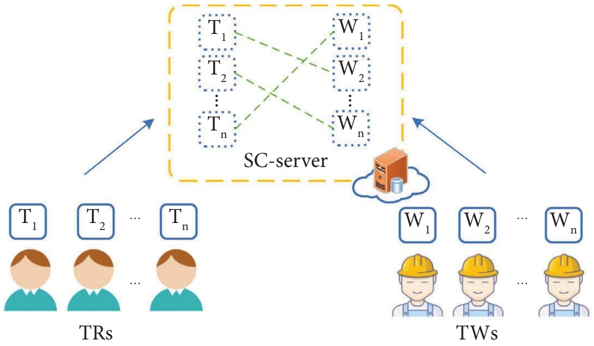


Figure 1: Tasks assignment model in SC.

user privacy information because more users' relevant information is uploaded.

Unfortunately, in the SC model, there is no wholly trusted party among the three parties. As shown in Figure 1, adversaries can disguise themselves as TRs or TWs to participate in SC tasks, listen to service requests sent by TRs or TWs to the SC-server or even hack the SC-server to obtain task requests from TRs and TWs. Adversaries may infer the users' physical state, living habits, and religion based on the information uploaded by users [6]. For example, Alice accepts the task of delivering medicine to Bob. People who know this information can know that Bob is in poor health and can even infer what disease Alice has according to the medicine. If the address is a residential area and the delivery time is after hours, the address is likely Alice's residence. It follows that disclosing such task information would pose a potential threat to users.

However, data privacy and security are becoming more and more vital in calculating user data [7]. Users would be less likely to participate in SC out of fear of the risk of their privacy being exposed. The loss of users is highly detrimental to SC development, and meager user participation will make assigning SC tasks challenging. Therefore, many privacy-preserving task assignment schemes have been proposed to ensure that task information is not available to anyone other than the TR and TW of that task. They can be broadly classified into two categories based on the protection mechanism. One is to fuzz the information, while the other is to encrypt the information.

Blurring processes protect user privacy by constructing a cloak zone where users cannot be precisely distinguished. This indistinguishability leads to finding only TWs without determining whether they are optimal. The distance traveled is the primary condition for determining the optimal TW, that is, the distance from the TW to the task location, provided that the TW can complete the task. TWs are more likely to choose a less distant job than the jobs they can do. In this way, they have a low cost of arriving at the location and can complete more tasks and earn more money. On the other hand, for TRs, the closer TW may arrive at the destination faster. Their wait times are short, so they have a better service experience. In contrast to the fuzzing process, methods of encryption protection can achieve exact matching. However, unfortunately, encryption protection methodsare quite time-consuming due to the need to perform calculations and comparisons of encrypted data.

Hence, we would like to construct a privacy-preserving scheme that combines the advantages of fuzzy processing and cryptographic protection. It can protect the privacy of TRs and TWs and assign tasks accurately and efficiently at the same time. Dividing the map and performing encrypted task matching based on the division is a promising solution. However, a task assignment strategy for encrypted information partition retrieval needs to address the following challenges.

Challenge 1: How to achieve matching without knowing the specific information of TRs and TWs. It is easy to match TRs with TWs without privacy protection simply by comparing and filtering their information to select TWs who meet the conditions. As mentioned above, information about TRs and TW needs to be processed to protect their privacy. Matching these ambiguous or encrypted messages is the first difficulty that needs to be solved.

Challenge 2: How to determine the search area without knowing the exact location of TRs and TWs. In order to reduce the number of encrypted samples to be compared, a subregion search is undoubtedly a better approach. However, two issues need to be addressed to accomplish this. The first one is to determine that the optimal TW exists in the retrieval region if the TWs exist in the retrieval region. The second one is how to expand the retrieval region if no TW satisfies the conditions in the retrieval region. After encrypting the user's location, information is no longer a visible difference. It is challenging to classify a group of undifferentiated data. Proposing a partitioned retrieval method that can be feasible is another difficulty that needs to be solved.

Challenge 3: How to determine the optimal TW among the TWs who satisfy the conditions. The key to this issue is that no additional information is leaked during the determination process. The optimal TW typically is the one that is the closest to the task location and satisfies the task conditions. Identifying the closest TW to the task location is easy without protecting privacy. However, achieving this in the context of privacy protection is not easy. Because to protect users' privacy, their location information needs to be processed. It is challenging to calculate and compare the distance of the processed locations and not disclose the information in the process.

To address the above challenges, we propose a privacy-preserving map retrieval task assignment scheme that can divide the map and accurately retrieve the optimal TWs according to this division, called pMATE. In pMATE, we divide the map with Voronoi and encrypt the task information of TRs and TWs with the Paillier cryptosystem. Our method reduces the candidate TWs' number to be compared by retrieving from the partition where the task is located. The proposed Find R and Get TWs algorithms can solve the problem that no suitable TWs are available in the search zone, and the search range needs to be expanded. The comparison of encrypted data is achieved using the homomorphic nature of Paillier. Furthermore, we propose the FMN algorithm to find the optimal TW or TWs from candidates.

The contributions of this paper are concluded as follows.

(1) We propose a privacy-preserving map retrieval task assignment solution approach that combines fuzzy processing with cryptographic retrieval. The specific location of the user is hidden in the partition. The encrypted search in the partition guarantees exact matches and reduces the number of samples to be compared.

(2) pMATE supports dynamic partition search. The proposed algorithms Find R and Get TWs can dynamically expand the search range according to the number of suitable TWs in the partition and ensure that the optimal TWs are in the search area.

(3) Our method can pick out the optimal TW or top-k TWs from the candidates that require only two communications with Find MinNumber (FMN) and Find Equal (FE) algorithms. In pMATE, we perturb the difference of encrypted distance and pass it to TR. The optimal TW is found with the help of TR and without compromising privacy.

The rest of this paper is organized as follows. We introduce the related work and preliminaries in Sections 2 and 3. Section 4 describes the system model, threat model, design goals, and problem formulation. Then, we introduce pMATE's preparation stage, the task request stage, the task assignment stage, and the algorithm involved in these stages in Section 5. We analyze the security of pMATE in Section 6. Section 7 reports and evaluates the experimental results. Finally, we conclude Section 8.

2. Related Work

2.1. The Privacy-Preserving Task Assignment with Fuzzification Techniques. The primary purpose of this protection method is to construct an indistinguishable region by processing the data. Data belonging to an indistinguishable partition has a standard attribute class but cannot differentiate between them. Anonymous technology is a typical representative of this. It is like seeing a group of people but cannot distinguish between them. Well-known anonymous technologies include k-anonymity [8] and l-diversity [9]. With these techniques, many task assignments have been proposed [10–13]. In [10], Kazemi and Shahabi proposed a PiRi framework to construct a cloaking region before connecting the SC-server. In this mode, instead of each participant raising a separate query, only a representative group of participants raises queries to the SC-server and shares their results with those who did not. Vu et al. [11] partition user location into groups by locality-sensitive hashing (LSH). They assume that users are all credible and hide the user's location information by establishing the group-satisfied k-anonymity. Pournajaf et al. [12] expanded the TWs' specific location into a cloaked area and set a limited travel distance. SC-server match TRs and TWs according to them. However, it can only protect the TWs' location and cannot find the nearest one. Hu et al. [13] extended Pournajaf's work. They extended the travel budget constraint in [12] to an area of space denoted by a rectangle R that employees are willing to travel. The anonymous techniques aim to protect user privacy by expanding the location range or integrating multiple user locations. These typically only protect one of TW and TR's private information and do not complete an exact match. Like the anonymous technique, perturbation techniques can protect users' privacy by differential privacy or perturbation of geographical position [14–17]. All of them use geocasting [18] for search workers. To et al. [14] use differential privacy protection and custom privacy budgets to protect worker location privacy. They need a trusted third party to preprocess the data. Then to expand their research [17], they address the moving TWs challenge by investigating continuously released privacy budget allocation techniques and using a Kalman filter-based post-processing technique to reduce inaccuracies from noise addition. Gong et al. [15] introduced reputation information for quality control based on [14]. Zhang et al. [16] use a similar framework. Instead of [14] using an adaptive grid to publish a cleaned location view to SC-server, they constructed a contour plot to represent the spatial distribution of TWs to introduce less noise than previous techniques. Both the anonymization technique and perturbation technique need to obfuscate the source data. This way protects users' privacy within that range but makes the accurate matching of task challenging. Compared with the existing solutions, our proposed pMATE protects both TR and TW's privacy and performs precise task matching.

2.2. The Privacy-Preserving Task Assignment with Encryption Techniques. Another way to protect user privacy is to use cryptography to encrypt data. This approach has a stronger theoretical basis for security, but efficiency is its disadvantage. Although Shu et al. [19] achieve bidirectional privacy-preserving task assignment through proxy re-encryption, their proposed solution requires additional servers such as proxy servers or fog nodes. They proposed another scheme [20] called pMatch, a proxy-free task matching via Shamir secret sharing. It can implement a privacy-preserving task distribution scheme without a proxy server. Wang et al. [6] proposed the PWSM to assign tasks by minimizing the travel distance. It only protects the TWs' privacy by obfuscating information about them. In [21], Ni et al. utilize a random matrix and the grid-encoded location to protect both TRs and TWs. Liu et al. [22] combine the Paillier cryptosystem with Yao's garbled circuits to find the nearest TWs and use Geohash to find the approximate nearest workers. But it can only perform fuzzy search due to its partitioning mechanism. Zhao et al. [23] propose the iTAM scheme that uses the Paillier cryptosystem to protect the privacy of both TRs and TWs and the corresponding matching protocol. However, it is inefficient because it needs to retrieve all ciphertext information. These solutions either do not match precisely or are less efficient. Therefore, we propose a privacy-preserving map retrieval task assignment scheme (pMATE), which can divide the map and accurately retrieve the optimal workers according to this division. In pMATE, we use the Paillier cryptosystem to match encrypted information exactly and reduce the number of comparisons by the Voronoi diagram.

3. Preliminaries

This section introduces essential concepts for clearly elaborating our model and methods. In our framework, we adopt the Voronoi diagram and Paillier cryptosystem to ensure the efficient and accurate matching of TWs and tasks while protecting their privacy.

3.1. Voronoi Diagram. Voronoi diagram (VD) [24, 25] is a way of partitioning a plane into Voronoi cells by electing particular points in this plane. Let $VD()$ be the function of the partitioning plane. These elected points are known as generator points denoted by the set $\mathbb{G}$, $\mathbb{G} = \{g_1, g_2, \ldots, g_n\}$, and the plane can be viewed as a set $\mathbb{U}$, $\mathbb{U} = R^2$. The result of this partition is that all points in its cell have the shortest Euclidean distance from the cell's generator point than others. We adopt $d(v, g)$ as the Euclidean distance between v and g. The v and g are the points in the plane. With the $\mathbb{G}$, we can obtain a partition of $\mathbb{U}$ denoted as follow.

$$\begin{cases} VD(\mathbb{U}) = \{\mathbb{U}_1, \mathbb{U}_2, \ldots, \mathbb{U}_n\}, \\ \mathbb{U}_i = \{v \mid d(v, g_i) \le d(v, g_{\text{others}})\}. \end{cases} \tag{1}$$

3.2. Paillier Cryptosystem. Paillier cryptosystem [26] is a homomorphic public key encryption scheme. Its properties can be used in ciphertext calculations to preserve users' privacy.

Key Generation: We set $N = pq$ where p and q are independent large prime numbers, and they are selected randomly such that $\gcd(N, (p-1)(q-1)) = 1$. Let $\lambda = \operatorname{lcm}(p-1, q-1)$, and select random integer g where $g \in \mathbb{Z}^*_{N^2}$, and then compute $\mu = (L(g^\lambda \bmod N^2))^{-1} \bmod N$, where the function L is defined as $L(x) = (x - 1/N)$. The public key and private key are $pk = (g, N)$ and $sk = (\lambda, \mu)$.

Encryption: Given a message $m \in \mathbb{Z}_N$ to be encrypted, it is encrypted by the public key pk to $\mathscr{C} = [\![m]\!]_{pk} = g^m \cdot r^N \bmod N^2$, where r is a selected random integer $r \in \mathbb{Z}^*_N$.

Decryption: Given a ciphertext $\mathscr{C}$ to be decrypted, it is decrypted by the private key sk to $m = D_{sk}(\mathscr{C}) = L(\mathscr{C}^\lambda \bmod N^2) \cdot \mu \bmod N$.

Its properties are listed as follows where $x, y \in \mathbb{Z}_N$.

(1) Additive homomorphism:

$$D_{sk}([\![x]\!] \cdot [\![y]\!]) = D_{sk}([\![x + y]\!]). \tag{2}$$

(2) Scalar-multiplicative homomorphism:

$$D_{sk}([\![x]\!]^y) = D_{sk}([\![x \cdot y]\!]). \tag{3}$$

4. System Overview and Problem Statement

In this section, we illustrate the system model, threat model, and design goals of pMATE. Then we introduce the problem formulation in SC task allocation.

4.1. System Model. As shown in Figure 2, our system mainly consists of SC-server, TWs, and TRs. We consider privacy-preserving task assignments in SC, and the main focus is on detecting the closest TWs while satisfying the task capability and time constraints.

The complete task assignment process has four steps, which are described below.

Step 1: Task requirement. TRs submit their task requests to the SC-server. The requested information includes the TR's task description without private information, encrypted task information, and public key.

Step 2: Search for candidate TWs. SC-server broadcasts the task description, encrypted task information, and public key to search for interested TWs based on the partition where the task is located. Interested TWs send task requests encrypted by the public key to SC-server. If SC-server does not receive enough task requests, it will expand the broadcast range and continue searching for alternative TWs until it finds a sufficient number of TWs.

Step 3: Encrypted information calculation. SC-server calculates the encrypted information uploaded by TWs and TRs according to the task assignment constraints.

Step 4: Search for optimal TW. SC-server disrupts the calculation results and collaborates with the TR, who issued the task to find the optimal TW without compromising privacy.

4.2. Threat Model. As mentioned above, the potential threat to the system model comes from three sources: TRs, TWs, and SC-server. We assume SC-server is honest-but-curious, which follows the protocols but wants to pry into users' private information, and most of TRs and TWs ask for help in the platform. This assumption is reasonable because SC-server needs to be approved and registered before it goes online. It can ensure the trust of SC-Server to a certain extent. Then we assume a strong adversary α in our model with the following capabilities. The goal of α is to obtain the privacy information of TRs or TWs.

(1) α can eavesdrop on all communication channels to obtain the encrypted message.

(2) α can disguise a TR to issue task request to SC-server and obtain the information returned from SC-server.

(3) α can disguise a TW to compete for a task and obtain the information about the task.

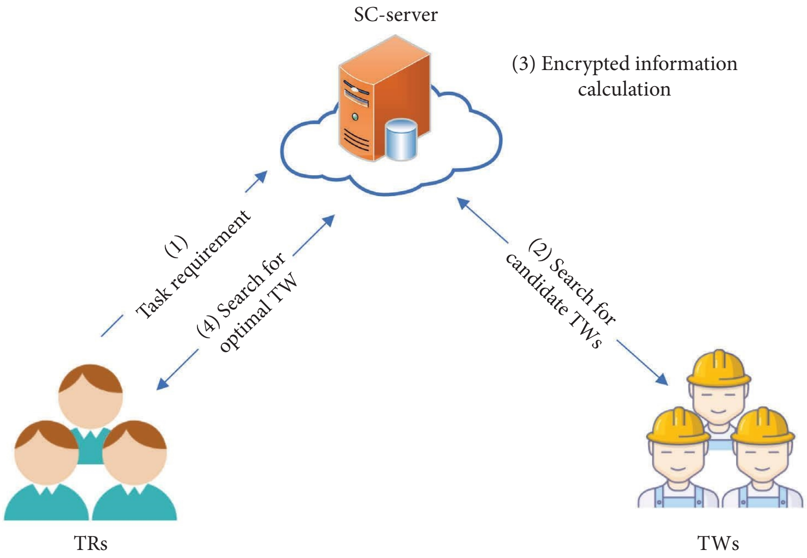

SC-server
(3) Encrypted information calculation
(1) Task requirement
(4) Search for optimal TW
(2) Search for candidate TWs
TRs
TWs

Figure 2: System model of pMATE.

(4) α can compromise SC-server to obtain exchange messages among TRs, TWs, and SC-server.

Note that α is restricted from compromising both the TRs and SC-server simultaneously.

Remark 1. In our framework, if α compromise both SC-server and TRs simultaneously, it could get the TWs' location information. However, if α compromise both SC-server and TWs at the same time, it could only get the range of TR's location information who submits task request. Because only the TR has the private key, only he/she can decrypt the location information. Although location will not be exposed in this situation, there will be a certain amount of information leakage. Furthermore, it is a general assumption that there is no CS/TR or CS/TW collusion attack among the three.

4.3. Design Goals. Given these three potential threats, we want to design a scheme that can protect users' task information from leakage for both TRs and TWs until the task assignment is done.

Privacy. In the process of an SC task, for TRs, its task information should be confidential in the task application process. Both SC-server and TWs cannot get the plaintext of task information. Every TR should encrypt his/her task information before sending a task request to SC-server. For TWs, they need to encrypt their task information too. In addition, their ciphertext of location information cannot be obtained by TR, who issues the task. Because TR has the private key, he/she can decrypt the ciphertext.

Calculability. The SC-server should have ciphertext computing capability since the task information it receives is encrypted.

Efficiency. Comparing and calculating encrypted data is time-consuming. In SC, there is a large number of such operations, and it increases with the number of users. Computational efficiency is an issue to consider. So we devised a method to reduce this time consumption and improve efficiency.

4.4. Problem Formulation. To transform the real problem into a mathematical problem, we formulate this problem as follows.

Definition 1 (Task). Let $\mathbb{T} = \{T_1, T_2, ..., T_{\mathrm{n}}\}$ be a task collection, and each task is represented as $T_{\mathrm{i}} = \{\mathrm{id}, \mathrm{lat}, \mathrm{lng}, t_R, \mathrm{cap}_R\}$. id is the task number, $(\mathrm{lat}, \mathrm{lng})$ is the latitude and longitude of the task's location, t_R is the latest start time of the task, cap_R is the capability required by the task. It indicates the task type or required sensors, such as transportation, photo-taking, temperature acquisition, etc.

Definition 2 (TW). Let $\mathbb{W}_{\mathrm{i}} = \{W_1, W_2, \ldots, W_{\mathrm{n}}\}$ be a TW collection for T_{i}, and each TW is represented as $W_{\mathrm{i}} = \{\mathrm{id}, \mathrm{lat}, \mathrm{lng}, t_W, \mathrm{cap}_W\}$. id is the TW's number, $(\mathrm{lat}, \mathrm{lng})$ is the latitude and longitude of TW's location, t_W is the arrival time of the TW, cap_W is the capability of the TW, like cap_R.

Definition 3 (Travel Distance). Travel Distance (TD) is the Euclidean distance from the location of a TW to the task position. It can be calculated according to the latitude and longitude of the TR and TW, the TR's location is $(\mathrm{lat}_R, \mathrm{lng}_{\mathrm{R}})$, and the TW's location is $(\mathrm{lat}_W, \mathrm{lng}_{\mathrm{W}})$. The location information is preprocessed into integers to satisfy the Paillier cryptosystem. Let $\mathbb{D} = \{d_1, d_2, \ldots, d_n\}$ be a collection of TW's TD who want to get this task. d_i is the No. i TW's TD. The TD's formula is as follows.

$$d_W = \sqrt{\left(\mathrm{lat}_R - \mathrm{lat}_W\right)^2 + \left(\mathrm{lng}_\mathrm{R} - \mathrm{lng}_\mathrm{W}\right)^2}. \qquad (4)$$

Definition 4 (Task Assignment). In the SC, it is the fundamental purpose to match TWs for specific tasks. In this paper, we use equation (5) as the matching rule. Supposing a TR and TW are denoted by $T_i = \{i, \mathrm{lat}_R, \mathrm{lng}_\mathrm{R}, t_R, \mathrm{cap}_R\}$ and $W_j = \{j, \mathrm{lat}_W, \mathrm{lng}_\mathrm{W}, t_W, \mathrm{cap}_W\}$. If the following constraints are met

$$\{t_W \leq t_R, \qquad (5a)$$

$$\{d_W = \min(\mathbb{D}), \qquad (5b)$$

$$\{\mathrm{cap}_\mathrm{R} = \mathrm{cap}_W. \qquad (5c)$$

The TW is considered to be the best match for this task. (5a) means that the TW can arrive at the task location on time. In (5a), $t_W = t'_W + \Delta t$. t'_W is the earliest departure time for TW to go to the destination, and Δt is the estimated time spent on the journey. Due to the unclear destinations, we set Δt as twice the time TW goes to the generator point of the task in the cell. (5b) means that the TW's TD is minimum. (5c) means that the TW is adequately capable of this task.

5. Design of pMATE

5.1. Framework Workflow. We present the workflow of pMATE. The whole workflow is divided into three stages: the preparation stage, the task request stage, and the task assignment stage.

Specifically, as a request task example shown in Figure 3, there are seven steps in the workflow without the preparation stage.

In ①, a TR sends its encrypted location information, the capability required to complete the task, the cell number obtained during the preparation stage, the public key, and the latest start time of the task to SC-server to initiate a task request. The SC-server numbers the task request and puts it into $\mathbb{T}$.

In ②, the SC-server selects a task request from $\mathbb{T}$ according to the recent start time and broadcasts it to TWs using the Find *R* and Get TWs algorithm based on the cell number received.

In ③, TWs who want to get the task and meet the time and ability constraints reply to a message to SC-server based on the information they received. SC-server catches the response by the TWs Respond function, numbers these TWs, and puts their information into $\mathbb{W}_i$. The *i* is the task number.

In ④, SC-server determines whether the number in the set $\mathbb{W}_i$ meets the requirements. If it does not, SC-server will expand R and repeat ②.After the number of TWs meets the requirements, SC-server selects the TWs in $\mathbb{W}_i$ and calculates the encrypted distance between them and the task point respectively, calculates the difference of ciphertext distance, and forms the difference into a sequence.

In ⑤, the SC-server processes and disarranges the sequence and sends it to the TR.

In ⑥, the TR decrypts the sequence and returns the relationship of size to SC-server.

In ⑦, SC-server finds the nearest TW or TWs based on the relationship and associates the TR with TW or TWs.

5.2. Preparation Stage. In the previous part, we introduced the workflow of pMATE. Before that, there is a preparation stage in which we partition the map with VD. Ciphertext calculation and comparison are time-consuming, especially when many ciphertext calculations and comparisons are required. Because the data is encrypted, the value cannot be confirmed. All ciphertext value differences need to be calculated and compared each time. Moreover, these calculations are very time-consuming. Therefore, we plan to divide the map into regions to reduce the number of comparison samples and the time cost and improve efficiency. SC-server can locate an approximate range and expand its search until it matches the suitable TW or TWs when a TR submits a task request based on her/his partition number. Since the map information is publicly available in advance, it is easy to identify the authenticity.

Compared with many other partition methods, the VD algorithm can guarantee a unique result and the shortest distance from the point in the cell to the generator point. With that, we can ensure the uniqueness of the partition and that the points in the cell have the same properties when we choose the intersections as the generator points of VD. For a cell, each point in it is the shortest distance to this intersection which belongs to the traffic road network. This division of the map according to road conditions is more realistic.

VD as a fundamental structure has been researched and applied in many fields. In our work, to facilitate the Get TWs algorithm, some information about VD needs to be stored. The left part of Figure 4 shows an example of the VD partition, and the right part presents its storage structure. Specifically, C_1 is the cell number, V_1 is its generator point, and $\{p_1, p_2, p_3, p_4, p_5\}$ is the set of its vertexes. The Cell Number list records each cell. Each cell links a logic space that contains its generator called Generator Point, the set of its vertexes called Vertex, and the set of neighbor cells' numbers called Adjacency. Since the map information can be obtained in advance, these calculations can also be completed before the task request stage, so the waiting time for the request will not increase. This stage is called the preparation stage.

5.3. Task Request Stage. ① to ③ in the workflow is the task request stage. In this stage, a TR submits a task request to SC-server. Then SC-server looks for TWs who want to get the task by the following algorithms. When a user wants to enjoy the services of the SC-server, she has to register with SC-

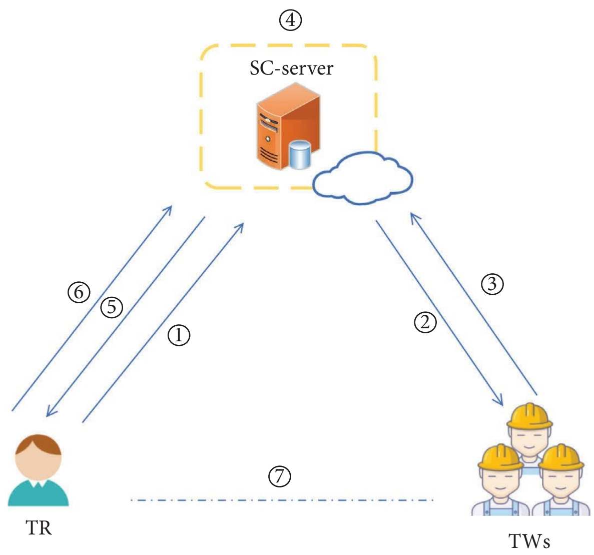
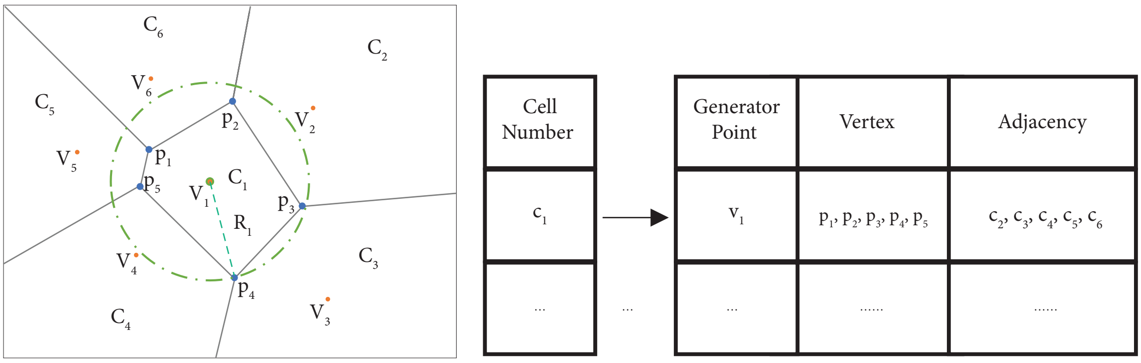


FIGURE 3: Workflow of pMATE.



Cell Number		Generator Point	Vertex	Adjacency
c_1	→	v_1	p_1, p_2, p_3, p_4, p_5	c_2, c_3, c_4, c_5, c_6
...	...	...	......	......

FIGURE 4: Voronoi diagram and voronoi diagram storage structure.

Server and get VD information for the preparation phase. In other words, she needs to know her location's cell number. When submitting a task request to SC-server, she needs to query her cell number according to the VD information obtained during registration. Furthermore, the map's information is public, so she can quickly determine its reliability by comparing it with VD.

In our framework, Euclidean distance is the key to matching tasks. Expanding equation (4), we can get the following:

$$
\begin{aligned}
d_{RW} &= \sqrt{(lat_R - lat_W)^2 + (\ln g_R - \ln g_W)^2} \\
&= \sqrt{lat_R^2 + lat_W^2 - 2lat_R \cdot lat_W + \ln g_R^2 + \ln g_W^2 - 2\ln g_R \cdot \ln g_W}, \\
d_{RW}^2 &= lat_R^2 + lat_W^2 - 2lat_R \cdot lat_W + \ln g_R^2 + \ln g_W^2 - 2\ln g_R \cdot \ln g_W.
\end{aligned} \tag{6}
$$

With the properties of Paillier, we can get the following:

$$
\llbracket d_{RW}^2 \rrbracket = \llbracket lat_R^2 \rrbracket \cdot \llbracket lat_W^2 \rrbracket \cdot \llbracket 2lat_R \rrbracket^{-lat_W} \cdot \llbracket \ln g_R^2 \rrbracket \cdot \llbracket \ln g_W^2 \rrbracket \cdot \llbracket 2\ln g_R \rrbracket^{-\ln g_W}. \tag{7}
$$

Due to the $d_{RW} \geq 0$, $d_{RW_1} \geq d_{RW_2} \Leftrightarrow d_{RW_1}^2 \geq d_{RW_2}^2$. So, we have to compare d_{RW}^2.

In this stage, for the TR who wants to submit the task to SC-server, she needs to upload her cell number, the latest start time t_R, and $\llbracket \mathrm{lat}_R^2 \rrbracket$, $\llbracket 2\mathrm{lat}_R \rrbracket$, $\llbracket \mathrm{lng}_R^2 \rrbracket$, $\llbracket 2\mathrm{lng}_{\mathrm{R}} \rrbracket$ to SC-server. SC-server broadcasts the task description, the task time, cell number, public key, and encrypted location information by the Geo-Broadcast [19] function to find TWs who want to get this job. Due to the cells of VD being irregular convex quadrilateral, it is difficult to determine the broadcast radius. As shown in Figure 4, the R_1 is the broadcast radius we want to find for C_1 in the first round. R_1 is chosen as the broadcast radius to avoid the situation where the nearest TW is in the adjacent cell of C_1 and is missed. We can see that the algorithm for finding TWs is divided into two parts. Part of it is to determine the broadcast radius based on the search area and broadcast to find TWs. Another part is expanding the search area when no TWs are found. So, we propose the Find R algorithm to find the suitable broadcast radius for the Get TWs algorithm. Find R's initialization is finished as part of the Get TWs algorithm.

Assuming a TR submits a task request, she uploads her cell number to SC-server, and SC-server can get her cell generator point, cell vertexes, and adjacency cell number according to the information obtained during the preparation stage. Let the cell generator point be the central point called v, $\mathbb{C}$ be the set of hunting cells, $\mathbb{N}$ be the vertex sets of $\mathbb{C}$, $\mathbb{L}$ be the vertex sets of adjacency cells of $\mathbb{C}$, $\mathbb{K}$ be the set of the farthest possible point. $\mathbb{K}$ is the subtraction of $\mathbb{L}$ and $\mathbb{N}$, $\mathbb{K} = \mathbb{L} - \mathbb{N}$. In the Algorithm 1, C_1.Vertex $\cap C_2$.Vertex $\cap \cdots \cap C_n$.Vertex represents all vertexes of the search area. Calculate the distance between all points in $\mathbb{K}$ and v. The maximum value is the search radius, R. With R, we can get the Get TWs algorithm described as follows. It is a circular algorithm that collects TWs who want to get task number i into $\mathbb{W}_i$. For simplicity, we adopt $\mathbb{W}$ instead of $\mathbb{W}_i$.

In Algorithm 2, we need to initialize those sets: $\mathbb{W}$, $\mathbb{L}$, $\mathbb{N}$, $\mathbb{K}$, $\mathbb{C}_{\mathrm{now}}$, $\mathbb{C}_{\mathrm{before}}$, $\mathbb{C}_{\mathrm{cache}}$. Those sets' initial values are null. When we run Algorithm 1, we can get the broadcast radius according to the search area. We need to update the search area to get TWs. The TWs who want to get this task need to calculate how long it takes to get from their position to the generator of the task in the cell, and they can get the estimated arrival time t_W, and upload t_W, $\llbracket \mathrm{lat}_W^2 \rrbracket$, $\llbracket 2\mathrm{lat}_R \rrbracket^{-\mathrm{lat}_W}$, $\llbracket \mathrm{lng}_W^2 \rrbracket$, $\llbracket 2\mathrm{lng}_{\mathrm{R}} \rrbracket^{-\mathrm{lng}_{\mathrm{W}}}$ to SC-server for task matching. The Worker Respond function is a listener that puts received replies into the $\mathbb{W}$. In Algorithm 2, t is the configurable waiting time. After the Geo-Broadcast function is executed, we wait for the specified time. Then we put the information of responded TWs into the set $\mathbb{W}$ by the Worker Respond function during this time. If the $\mathbb{W}$ does not meet the quantity requirement, we would wait some time and do it again, or expand the search area $\mathbb{C}_{\mathrm{now}}$ and run again until the $\mathbb{W}$ is satisfiedsati. Whether to expand the radius search depends on the urgency that the task needs to be completed. In Algorithm 2, we only give the way to expand the search area $\mathbb{C}_{\mathrm{now}}$. The waiting way is just a matter of setting a suspension and wake time.

5.4. Task Assignment Stage. ④ to ⑦ in the workflow is the task assignment stage. By the task request stage mentioned above, the SC-server got $\mathbb{W}$. In this stage, its goal is to find the TW who meets the above constraints and is closest to the task location in the $\mathbb{W}$.

The TWs can arrive on time at the task's location based on previous calculations. This record of TWs' arrival time is only used as proof of later failure to arrive on time. The capability required to complete the task is sensitive information that exposes the information of the task and TW. It can be protected by encrypting the task capabilities. An additional ciphertext comparison is required in the encrypted task capability mode to confirm that the task requirements are met. The ciphertext comparison operation is the same as the encrypted location distance comparison. For the convenience of description, the plaintext task capability description is chosen in this paper. SC-server broadcasts this task information to find the suitable TW or TWs. Next, we need to solve another critical problem: to figure out the nearest TW or TWs.

We can determine how far each TW is from the task according to the information they uploaded with Paillier's properties. In mathematics, it is recorded as $\llbracket \mathbb{D} \rrbracket = \{\llbracket d_1 \rrbracket, \llbracket d_2 \rrbracket, \ldots, \llbracket d_n \rrbracket\}$, $\llbracket d_i \rrbracket$ representing the encrypted distance from the TW labelled i to the task location. Let M be the matrix for recording the difference of encrypted distance, $m_{ij} = \llbracket d_i - d_j \rrbracket^{x_{ij}}$. In this equation, x_{ij} is randomly either 1 or −1 to perturb. Notes that $m_{ij} = \llbracket d_i - d_j \rrbracket^1 = \llbracket d_j - d_i \rrbracket^{-1} = m_{ji}^{-1}$. The value of m_{ij} represents the size relationship between d_i and d_j. m_{ij} also records the relation between them. So we transmit one of them to the TR. Let the transmitted collection be the $\mathbb{M}$. However, if we transmit $\mathbb{M}$ to the TR directly, the TR can deduce all distance magnitude relationships. To avoid such information leakage, we need to disrupt the order of $\mathbb{M}$. As shown in Figure 5, the disrupting order of $\mathbb{M}$ is $\mathbb{A}$, $\mathbb{A} = \{a_1, a_2, \ldots, a_{n(n-1)/2}\}$. SC-server selects the value of $\mathrm{m_{ij}}$ in $\mathbb{M}$ orderly, puts it into $\mathbb{A}$ randomly, and transmits $\mathbb{A}$ to the TR.

Since the value of elements in $\mathbb{A}$ is encrypted with the TR's public key, she can decrypt the processed collection with her private key, get the relationship, and record them in $\mathbb{R}$, $\mathbb{R} = \{R_1, R_2, ..., R_{n(n-1)/2}\}$. For $a_n = \llbracket A - B \rrbracket$, if $A - B > 0$, $R_n = 1$, if $A - B < 0$, $R_n = -1$, if $A - B = 0$, $R_n = 0$. The TR can easily judge the relationship between A and B but can not judge which is A and which is B. Because, when $x_{ij} = 1$, $a_n = \llbracket d_i - d_j \rrbracket$; when $x_{ij} = $-1, $a_n = \llbracket d_j - d_i \rrbracket$. In other words, getting the R-value is easy for the TR, but the relationship between A and B is difficult. She only has the value of a_n, sets the value of R_n by a_n, and returns the $\mathbb{R}$ to the SC-server.

Nowadays, for SC-server, getting the TW closest to the task location $\mathbb{R}$ is a problem. To efficiently retrieve the information of the same element in $\mathbb{A}$, $\mathbb{R}$, and $\mathbb{M}$, we introduce $\mathbb{M}$'. As shown in Figure 6, it stores the id of the matrix element in M and links the subscript table that stores the subscript of the element in $\mathbb{A}$ and the x table that stores the value of x_{ij}. Obviously, in a task, $\mathbb{A}$, $\mathbb{R}$, $\mathbb{M}'$, and $\mathbb{M}$ have the same sequence length. The TR got the $\mathbb{R}$ according to the $\mathbb{A}$. So, there is a one-to-one correspondence between $\mathbb{R}$ and $\mathbb{A}$. With the Subscript table, we can get the logic relationship represented by dotted arrows between $\mathbb{M}'$ and $\mathbb{R}$. To facilitate calculations, we open a space R to record the value of $\mathbb{R}$ in

```
**Input:** the set of hunting cells $\mathbb{C}$, central point $v$.
**Output:** $R$
$\mathbb{L} \leftarrow C_1$.Vertex $\cap C_2$.Vertex $\cap \cdots \cap C_n$. Vertex; //$\mathbb{C} = \{C_1, C_2, \ldots, C_n\}$
$\mathbb{K} \leftarrow \mathbb{L} - \mathbb{N}$; //initialization in Get TWs algorithm
$R \leftarrow \max(d(v, K_i))$; //$i = 1$ to $n$;
$\mathbb{N} \leftarrow \mathbb{K}$;
**Return** $R$;
```

Algorithm 1: Find R.

```
**Input:** Cell number of TR in the cell $x$, $t$;
**Output:** $\mathbb{W}$//$\mathbb{W} = \{W_1, W_2, \ldots, W_n\}$ the set of TWs
Initialize $\mathbb{W}$, $\mathbb{L}$, $\mathbb{N}$, $\mathbb{K}$, $\mathbb{C}_{\text{now}}$, $\mathbb{C}_{\text{before}}$, $\mathbb{C}_{\text{cache}} \leftarrow \varnothing$;
Initialize float $R \leftarrow 0$;
Initialize $v \leftarrow x$. Generator Point;
Initialize $\mathbb{C}_{\text{now}} \leftarrow x$;
**While** $\mathbb{W} = \varnothing$ **do**
  $R \leftarrow$ Find $R$ ($\mathbb{C}_{\text{now}}$, $v$);
    Geo-Broadcast ($R$);
    Wait ($t$);
    $W \leftarrow$ Worker Respond();
    $\mathbb{C}_{\text{cache}} \leftarrow \mathbb{C}_{\text{now}}$;
    $\mathbb{C}_{\text{now}} \leftarrow \mathbb{C}_{\text{now}}$.Adjacency $- \mathbb{C}_{\text{cache}} - \mathbb{C}_{\text{before}}$;
    $\mathbb{C}_{\text{before}} \leftarrow \mathbb{C}_{\text{cache}}$;
**Return** $\mathbb{W}$;
```

Algorithm 2: Get TWs.

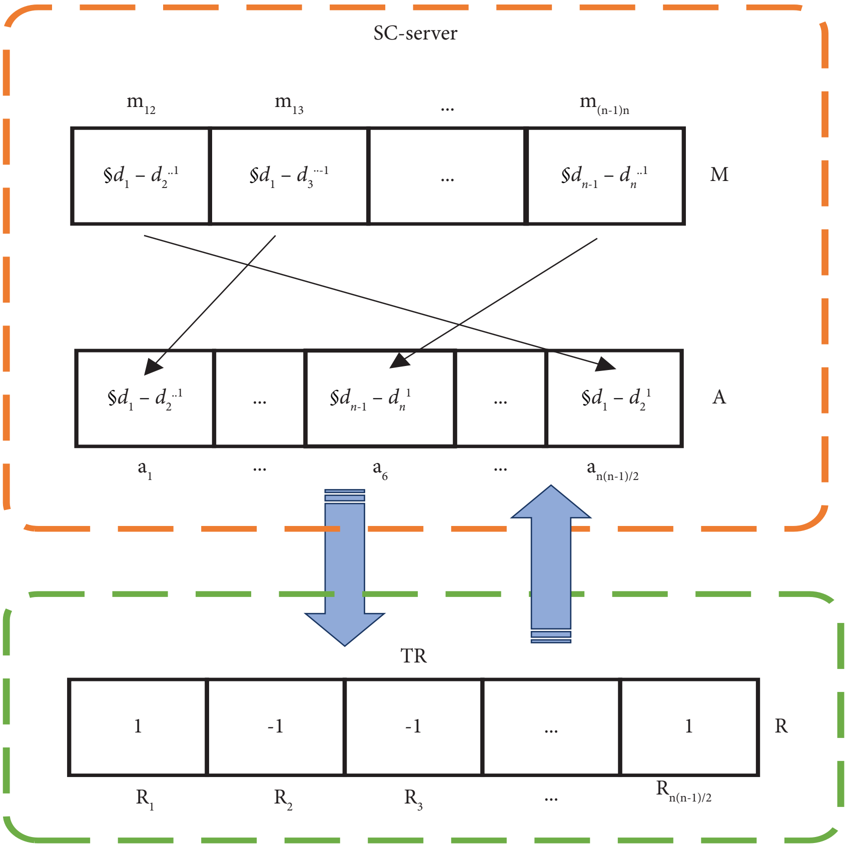
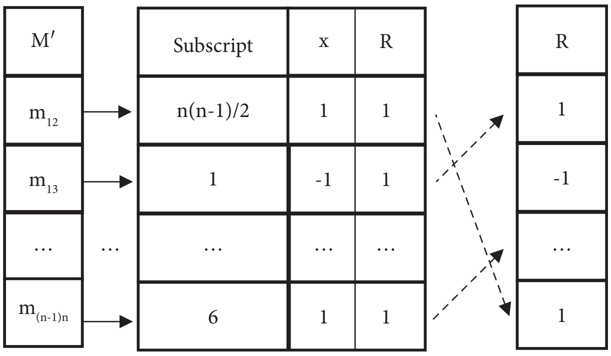


Figure 5: Process of transmit between TR and SC-server.



M′
m$_{12}$
m$_{13}$
…
m$_{(n-1)n}$

Subscript	x	R
n(n-1)/2	1	1
1	-1	1
…	…	…
6	1	1

R
1
-1
…
1

Figure 6: Storage structure and relationship of tables.

the linked table. It can help us to find the corresponding relation quickly.

With the relation, the following analysis can be obtained. Assuming $a_n = [\![d_i - d_j]\!]^{x_{ij}}$, when $x_{ij} = 1$, $a_n = [\![d_i - d_j]\!]$. If the $R_n = 1$, we can get $d_i - d_j > 0$. If $R_n = -1$, $d_i - d_j < 0$. When $x_{ij} = -1$, $a_n = [\![d_j - d_i]\!]$. If the $R_n = 1$, we can get $d_j - d_i > 0$. If $R_n = -1$, $d_j - d_i < 0$. So we can obtain the relationship shown in the following equation.

$$\{x_{ij} * R_n = 1 \Rightarrow d_i > d_j, \tag{8a}$$

$$\{x_{ij} * R_n = -1 \Rightarrow d_i < d_j, \tag{8b}$$

$$\{x_{ij} * R_n = 0 \Rightarrow d_i = d_j, \tag{8c}$$

with this relationship and storage structure of tables, we can judge the size relationship between d_{n-1} and d_n by the value of $x_{(n-1)n}$ times R_b. The value of $x_{(n-1)n}$ and R_b can obtain by $\mathbb{M}'$, $x_{(n-1)n} = m_{(n-1)n}.x$, $R_b = m_{(n-1)n}.R$. if we traverse the whole $\mathbb{M}'$, we will get the number of TW closest to the task location. However, sometimes, when a task requires multiple TWs to complete, SC-server should find the top-k nearest TWs for it. If we find the nearest TW and do not make any changes, we will get the same TW every time we run the program. For the algorithm to be used multiple times, we inverted the value of its x after getting the nearest TW's number. With this change, if SC-server wants to find the top-k nearest TWs, it just needs to perform k times. However, in the particular case of equality, the transitivity of the relationship is broken, causing the algorithm to fail. So we propose Algorithm 3 to find an equality relation. When Algorithm 4 finds the nearest TW number, it passes that number to Algorithm 3, which finds all equal distant numbers and returns them. After finding enough TWs to meet the requirements, SC-server establishes the link between the TR and TW or TWs by the number.

In Algorithm 3, we look for an equal value based on the number of the minimum value already found. According to the number of the minimum value, locate its row in the matrix M. Equal numbers can be found by traversing the value of R of this row. We put these equal minimums into a queue and modify their x values item by item.

6. Security Analyses

In this section, we analyze the security of the pMATE scheme from the perspective of potential risks that existed in the task request and task assignment stage.

Theorem 1. *pMATE can protect TR's task location privacy during the task request stage if the Paillier cryptosystem is secure.*

Proof. In pMATE, a TR encrypts his/her task location before he/she uploads them to SC-server, as mentioned above, like $[\![\mathrm{lat}_R^2]\!]$, $[\![2\mathrm{lat}_R]\!]$, $[\![\mathrm{lng}_R^2]\!]$, $[\![2\mathrm{lng}_R]\!]$. SC-server will send them to TWs. Since only the TR has the private key, SC-server, TWs, and even no matter who gets this information, it will not be able to decrypt if the Pailier cryptosystem is secure. It makes sure the privacy of TR location information.

Theorem 2. *pMATE can protect TWs' location privacy during the task request stage if the Paillier cryptosystem is secure.*

Proof. In pMATE, TWs need to upload their encrypted location to SC-server based on the TR's encrypted location received, as mentioned above, like $[\![\mathrm{lat}_W^2]\!]$, $[\![2\mathrm{lat}_R]\!]^{-\mathrm{lat}_W}$, $[\![\mathrm{lng}_W^2]\!]$, $[\![2\mathrm{lng}_R]\!]^{-\mathrm{lng}_W}$. SC-server owns encrypted information but not the private key. Conversely, the TR has the private key but no encrypted information. So both of them cannot get the location information of TWs if the Pailier cryptosystem works and there is no collusion between them.

Theorem 3. *pMATE does not reveal location privacy during the task assignment stage. In other words, pMATE can protect the location privacy of both TR and TWs during the TD comparison process.*

Proof. In the TD comparison process, the SC-server can figure out encrypted TDs, as mentioned above, like $[\![\mathbb{D}]\!] = \{[\![d_1]\!], [\![d_2]\!], \ldots, [\![d_n]\!]\}$. And it figures out $\mathbb{M}$, which sequence consisted of the difference of TDs, and gets $\mathbb{A}$ by disrupting the order of $\mathbb{M}$ randomly. Since SC-server does not have the private key, it sends $\mathbb{A}$ to the TR to determine the size relationship of TD. Now the TR can know the value of each element in $\mathbb{A}$ by the private key. However, since $\mathbb{A}$ is perturbed twice, TR cannot obtain any information to infer the location information of TW. Specifically, the first perturbation is to shuffle the positions of elements in $\mathbb{M}$. It makes it impossible for the TR to determine which two TDs are compared based on the element's position. The second perturbation is the addition of a random number x_{ij}. As mentioned above, $a_n = [\![d_i - d_j]\!]^{x_{ij}}$, when $x_{ij} = 1$, $a_n = [\![d_i - d_j]\!]$, when $x_{ij} = -1$, $a_n = [\![d_j - d_i]\!]$. The TR only knows the value of a_n, but cannot get the size relationship of d_i and d_j. Because the TR has no additional information to infer, it is similar to the one-time pad. For example, if you get the number 5, you cannot speculate which two numbers this number is calculated from. Further, suppose $a_n = A - B$, if you get the value of a_n, you cannot infer the size relationship

```
Input: int $i$;
Output: int $x[]$;        //the queue of the nearest TWs' number
Initialize int $x[]$;
Initialize int $n$, $k$;
put $i$ into $x[]$;
for $j = 1$ to $n$ do
  if ($m_{ij}.R = 0$)
    put the $j$ into $x[]$;
while $x[n = 0]$ ! = null do;
    for $k = 1$ to int $x[n]-1$ do
    $m_{kx[n]}.x \leftarrow m_{kx[n]}.x * -1$;
    for $k =$ int $x[n] + 1$ to $n$ do
    $m_{x[n]k}.x \leftarrow m_{x[n]k}.x * -1$;
  $n$++;
return $x[]$;
```

Algorithm 3: FE.

```
Input: $\mathbb{M}$';
Output: int $x[]$;        //an array of the nearest TWs' numbers
Initialize int $k$, $p$, $i \leftarrow n - 1$, $j \leftarrow n$;        //$n = \mathbb{D}$.length
While $i \neq 1$ do
        $p \leftarrow m_{ij}.R * m_{ij}.x$
    if $p = -1$        //$d_i < d_j$
        $j \leftarrow i$, $i \leftarrow i - 1$;
            else        //$d_i \geq d_j$
        $i \leftarrow i - 1$;
$p \leftarrow m_{ij}.R * m_{ij}.x$
  if $p = 1$        //return $j$, set $j$ max
    $x[] =$ FE ($j$);
    return $x[]$;
    else        //return 1, set 1 max
    $x[] =$ FE (1);
            return $x[]$;
```

Algorithm 4: FMN.

between A and B. All in all, the TR cannot get the location information of TWs. The TR returns the relation $\mathbb{R}$ to SC-server, which cannot get the location information of TWs too.

7. Performance Analysis

7.1. Experiment Setting. In this section, we will evaluate the performance of our proposed pMATE through some experiments. Similar to the work [22], we experimental data through a real-world dataset, Gowalla. Gowalla is a historical record set which records the social network users' check-in location information on social networks. We choose 110 check-in data in San Francisco with a latitude from 37.75431 to 37.80062 and a longitude from −122.42691 to −122.39382. We choose 100 check-in data as the TWs. The remaining ten pieces of data are used as the TRs. And TRs all satisfy the task conditions.

The experimental machine is a personal computer (PC) with a Core i7 CPU and 16 GB RAM and a smartphone with Kirin 990 CPU and 8 GB RAM. We use the smartphone to simulate TW or TR encryption of a task and test its time consumption. We use the PC to simulate task allocation in SC-server. We use Baidu Map to find the longitude and latitude of the crossroads in the experimental area, San Francisco, and generate VD according to it. Instead of CSP's broadcast function, we look for TWs within the radius. We evaluate the scheme's performance using encryption time, task allocation time, and task TD.

7.2. Evaluation Results. All methods, including ours, need to encrypt the task information before TRs and TWs apply for services to SC-server. In pMatch [20], TRs and TWs mainly perform exponential operations in the bilinear map. Assume that their number of sensors is three. They do not need to perform decryption operations. SC-server can compare the encrypted information. As same as [22, 23], we choose the Paillier cryptosystem to encrypt the task information. So we

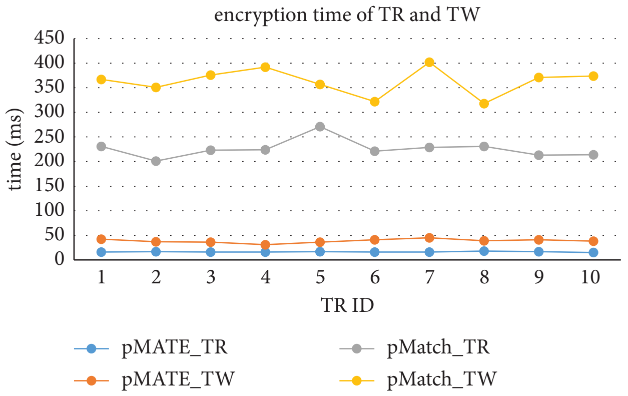


Figure 7: The comparison of encryption time between pMATE and pMatch.

experiment with the encryption time of TRs and TWs between pMATE and pMatch. As shown in Figure 7, we record the encryption time of all TRs and their corresponding TW, respectively.

After comparing the time of encryption, we mainly conduct comparative experiments on task allocation time. We randomly select 10 TWs from all TWs for task allocation for each TR and record the time to complete the task allocation in Figure 8. Similar to previous schemes [22, 23], pMATE is to find the nearest TW according to TD. But in ETA [22], they try to find an approximation to the nearest TW. As shown in Figure 8, the pMATE's consume-time is usually less than others because it uses the VD to shrink the set of matching TWs. The ETA [22] also wants to reduce the number of samples. However, its effect fluctuates wildly due to it being without an extension algorithm. Its time consumption depends entirely on the number of TWs randomly appearing in its partitioned area. In ETA [22], there is no specific rule for its division. However, if it is too large, it will contain too many TWs; if it is too small, there may be no TWs. In the experiment of this paper, if there are no TWs, the area will be doubled and re-divided until there are TWs. However, we can see that they have reduced the time consumption because they all use the method of reducing the number of samples. The record shows that the pMATE and ETA still have a similar time consumption on 2, 4, 5, and 8. Analyzing the experimental data, we found that the number of TWs in their search area is almost the same. But for the rest, pMATE can significantly improve the matching efficiency. A possible explanation is that the pMATE places the TR near the center of the search area. In iTAM [23], it is more time consuming than both ETA and pMATE due to the need to retrieve all tasks. In pMatch [20], its primary operations are bilinear map operations resulting in a high time overhead.

Another important indicator is the accuracy of the match. In other words, whether the matching TW is the nearest one from the task location. ETA, iTAM, and ours are all trying to find the nearest TW. So, we calculate the TD of the selected TW to the task location in each group. To compare obviously, we introduced a set of matching results in No-privacy. As shown in Figure 9, pMATE and iTAM can find the nearest TW, except for the ETA method. However, iTAM found the nearest TW by searching for all TWs. So although it can find the nearest TW, it takes a high time consumption. ETA reduces the search range but does not provide the corresponding search algorithm. So there is only a high chance of finding the nearest TW. Analysis of the incorrect situation of the ETA shows that the selected TW is only closest within the search area and not globally closest. It also verifies the deviation due to the lack of search algorithms.

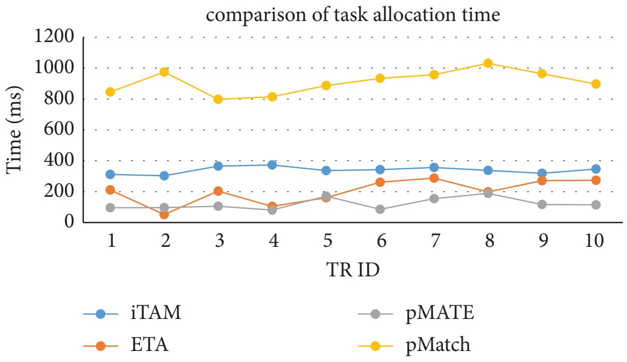


Figure 8: The comparison of task allocation time among iTAM, ETA, pMATE.

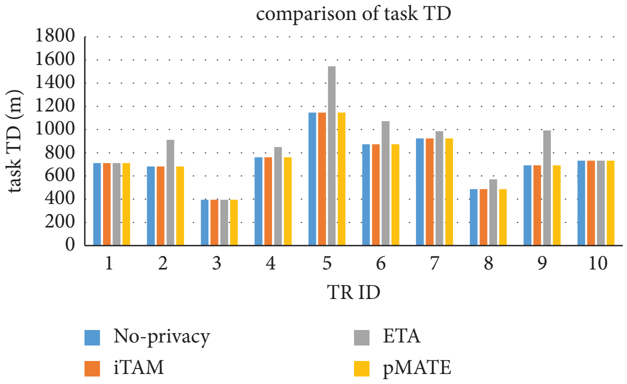


Figure 9: The comparison of task TD among No-privacy, iTAM, ETA, pMATE.

8. Conclusion

In this paper, we proposed pMATE, a privacy-preserving task assignment scheme to solve the optimization problems for minimum travel distance. We use the Paillier cryptosystem to encrypt the task information of both TRs and TWs to protect their privacy. To improve the retrieval efficiency, we use VD to divide the map and propose Find R and Get TWs algorithms to search for the candidate TWs. For these encrypted data of candidates and the structures of Voronoi storage, we propose FMN and FE algorithms that can efficiently complete the data comparison to find the TW who is the minimum travel distance to the task location. Finally, we verified the effectiveness and superiority of the scheme from three aspects: encryption time, task assignment time, and accuracy through experiments. In future work, we will focus on packaging multiple tasks and assigning them to a single worker to increase efficiency.

Data Availability

In this paper, we use the Gowalla dataset. The URL of the dataset is https://snap.stanford.edu/data/loc-gowalla.html.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

National Key R&D Program of China No. 2020YFB1805403; Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2017BDKFJJ015, No. 2018BDKFJJ008, No. 2018BDKFJJ020, No. 2018BDKFJJ021).

References

[1] W. Feng, Z. Yan, H. Zhang, K. Zeng, Y. Xiao, and Y. T. Hou, "A survey on security, privacy, and trust in mobile crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2971–2992, 2018.

[2] Y. Tong, Z. Zhou, Y. Zeng, L. Chen, and C. Shahabi, "Spatial crowdsourcing: a survey," *The VLDB Journal*, vol. 29, no. 1, pp. 217–250, 2020.

[3] Y. Liu, X. Zhang, Y. Li, J. Zhou, X. Li, and G. Zhao, "Graph-based facial affect analysis: a review," *IEEE Transactions on Affective Computing*, pp. 1–20, 2022.

[4] Y. Liu, F. Yang, C. Zhong, Y. Tao, B. Dai, and M. Yin, "Visual tracking via salient feature extraction and sparse collaborative model," *AEU-International Journal of Electronics and Communications*, vol. 87, pp. 134–143, 2018.

[5] Y. Li, J. Wei, Y. Liu, J. Kauttonen, and G. Zhao, "Deep learning for micro-expression recognition: a survey," *IEEE Transactions on Affective Computing*, vol. 13, no. 4, pp. 2028–2046, 2022.

[6] Z. Wang, J. Hu, R. Lv et al., "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1330–1341, 2019.

[7] B. Zhao, X. Liu, W. N. Chen, and R. Deng, "CrowdFL: privacy-preserving mobile crowdsensing system via federated learning," *IEEE Transactions on Mobile Computing*, p. 1, 2022.

[8] L. Sweeney, "k-anonymity: a model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, 2002.

[9] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam, "l-diversity: privacy beyond k-anonymity," *ACM Transactions on Knowledge Discovery from Data*, vol. 1, no. 1, pp. 106–115, 2007.

[10] L. Kazemi and C. Shahabi, "A privacy-aware framework for participatory sensing," *ACM Sigkdd Explorations Newsletter*, vol. 13, no. 1, pp. 43–51, 2011.

[11] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for k-anonymous location privacy in participatory sensing," in *Proceedings of the 2012 Proceedings - IEEE INFOCOM*, IEEE, Orlando, FL, Florida, March 2012.

[12] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations,"vol. 1, pp. 73–82, in *Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management*, vol. 1, pp. 73–82, IEEE, Brisbane, QLD, Australia, October 2014.

[13] J. Hu, L. Huang, L. Li, M. Qi, and W. Yang, "Protecting location privacy in spatial crowdsourcing," in *Proceedings of the Asia-Pacific Web Conference*, pp. 113–124, Springer, Cham, Guangzhou, China, September 2015.

[14] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proceedings of the VLDB Endowment*, vol. 7, no. 10, pp. 919–930, 2014.

[15] Y. Gong, C. Zhang, Y. Fang, and J. Sun, "Protecting location privacy for task allocation in ad hoc mobile cloud computing," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 110–121, 2018.

[16] L. Zhang, X. Lu, P. Xiong, and T. Zhu, "A differentially private method for reward-based spatial crowdsourcing," in *Proceedings of the International Conference on Applications and Techniques in Information Security*, pp. 153–164, Springer, Berlin, Heidelberg, Germany, November 2015.

[17] H. To, G. Ghinita, L. Fan, and C. Shahabi, "Differentially private location protection for worker datasets in spatial crowdsourcing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 4, pp. 1–949, 2016.

[18] J. C. Navas and T. Imielinski, "GeoCast—geographic addressing and routing," in *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pp. 66–76, Piscataway, NJ, USA, September 1997.

[19] J. Shu, X. Jia, K. Yang, and H. Wang, "Privacy-preserving task recommendation services for crowdsourcing," *IEEE Transactions on Services Computing*, vol. 14, no. 1, pp. 235–247, 2018.

[20] J. Shu, K. Yang, X. Jia, X. Liu, C. Wang, and R. H. Deng, "Proxy-free privacy-preserving task matching with efficient revocation in crowdsourcing," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 1, pp. 117–130, 2021.

[21] J. Ni, K. Zhang, Q. Xia, X. Lin, and X. Shen, "Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 6, pp. 1317–1331, 2020.

[22] A. Liu, Z. X. Li, G. F. Liu et al., "Privacy-preserving task assignment in spatial crowdsourcing," *Journal of Computer Science and Technology*, vol. 32, no. 5, pp. 905–918, 2017.

[23] B. Zhao, S. Tang, X. Liu, X. Zhang, and W. N. Chen, "iTAM: bilateral privacy-preserving task assignment for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 12, pp. 3351–3366, 2021.

[24] M. Erwig, "The graph Voronoi diagram with applications," *Networks*, vol. 36, no. 3, pp. 156–163, 2000.

[25] Y. Liu, X. Zhang, Y. Lin, and H. Wang, "Facial expression recognition via deep action units graph network based on psychological mechanism," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 12, no. 2, pp. 311–322, 2020.

[26] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the International conference on the theory and applications of cryptographic techniques*, pp. 223–238, Springer, Berlin, Heidelberg, Germany, May 1999.