BPPF: Bilateral Privacy-Preserving Framework for Mobile Crowdsensing

LIU Junyu, YANG Yongjian, WANG En

(Jilin University, Changchun 130012, China)

Abstract: With the emergence of mobile crowdsensing (MCS), merchants can use their mobile devices to collect data that customers are interested in. Now there are many mobile crowdsensing platforms in the market, such as Gigwalk, Uber and Checkpoint, which publish and select the right workers to complete the task of some specific locations (for example, taking photos to collect the price of goods in a shopping mall). In mobile crowdsensing, in order to select the right workers, the platform needs the actual location information of workers and tasks, which poses a risk to the location privacy of workers and tasks. In this paper, we study privacy protection in MCS. The main challenge is to assign the most suitable worker to a task without knowing the task and the actual location of the worker. We propose a bilateral privacy protection framework based on matrix multiplication, which can protect the location privacy between the task and the worker, and keep their relative distance unchanged.

Keywords: mobile crowdsensing; task allocation; privacy preserving

DOI: 10.12142/ZTECOM.202102004

https://kns.cnki.net/kcms/detail/34.1294.TN.20210517.1615.002.html, published online May 17, 2021

Manuscript received: 2021-03-11

Citation (IEEE Format): J. Y. Liu, Y. J. Yang, and E. Wang. "BPPF: bilateral privacy-preserving framework for mobile crowdsensing," *ZTE Communications*, vol. 19, no. 2, pp. 20 – 28, Jun. 2021. doi: 10.12142/ZTECOM.202102004.

1 Introduction

Mobile crowdsensing (MCS), as a critical component of the Internet of Things (IoT)[1], relies on various sensors in mobile devices to collect and transmit data through a wireless network. Nowadays, mobile devices are essential for our daily activities, including businesses, communications, and entertainments. According to Gartner statistics, the number of worldwide smartphones sales in 2018 was 1. 55 billion. Anyone with a smartphone can become a participant in the MCS system with wide coverage, which has gained popularity in recent years and become an appealing paradigm for sensing and collecting data[2].

In the traditional mobile crowdsensing, there are three entities: the crowdsensing platform, the worker, and the task requester. The task requester has some tasks to be completed in some places and is willing to pay for the task, the worker is the person who registers on the mobile crowdsensing platform to get the reward for completing the task, and the crowdsensing platform provides services for the task requester. The purpose of the platform is to recruit suitable workers for the task requester to complete the task under the condition of the minimum compensation. The distance between workers and tasks is highly related to the cost of workers who complete tasks. Most of the existing recruitment mechanisms take the distance between tasks and workers as the cost of workers completing tasks. Therefore, the total distance between workers and tasks is taken as the optimization objective of the algorithm in hope of minimizing the total cost of workers.

In order to recruit suitable workers for the task, the platform needs the location information of the task and workers. If users upload their actual location, it will bring danger to the user's privacy and eliminate the user's enthusiasm to participate in mobile crowdsensing activities. The invasion of the privacy of workers may occur when a worker uploads his or her actual location information on the platform, which may infer the worker's identity and preference based on the worker's location information, and disclose the privacy information to a third party for profit. On the other hand, to select the right worker, the task requester publishes the task information on the platform, and the platform and all workers can see the information, which poses a risk to the privacy of the task requester, because in certain cases the location and content of the task needs to be protected for the requester. For example, people with health problems at home can seek help through mobile crowdsensing, but publishing her health problems and her home address clearly violates her privacy.

Therefore, many privacy protection methods have been proposed to deal with the problem of privacy leakage in mobile crowdsensing, such as anonymity, obfuscation and encryption[3]. By adding noise to the user's actual location and confusing it with other locations[4], the secret sharing technology is used to encrypt the privacy information[5], and k-anonymity is used to cluster a group of users to protect the user's privacy information[6]. However, most of the existing privacy protection work mainly focuses on the privacy protection of workers with little consideration of task privacy protection. Some works[7] consider the privacy of both workers and tasks, but they need an online trusted third party (TTP) to assist in the task allocation phase, which undoubtedly leads to high communication overhead and unnecessary delay in the task allocation phase.

Inspired by the privacy protection problem, we focus on designing a bilateral privacy protection framework, which chooses the right worker to minimize the total moving distance while protecting the location information of workers and tasks without an online TTP in the task allocation phase. Specifically, the whole geographic space is divided into several sub-regions. The task requester maps the location of the task to the sub-region, uses a matrix to transfer the location information, then hides the actual location matrix, and uploads the confusing location matrix. Similarly, workers also map their positions to their sub-regions, use a matrix to transfer the position information, hide the actual location matrix, and upload the confusing location matrix. After receiving the confusing location matrix from the task requester and the worker, the platform uses the confusing location information instead of the actual location information to select the appropriate worker.

Obviously, if the location of the task and the location of the worker are both confusing, it is difficult for the platform to choose the right worker. Therefore, the first challenge of the bilateral privacy protection framework is to solve this problem: how to protect the location information of tasks and workers while keeping the relative distance constant. In this regard, we design a novel privacy protection method based on matrix multiplication, with which we can protect not only the location privacy of workers, but also the location privacy of tasks. At the same time, we can ensure that the platform can accurately measure the relative distance between tasks and workers through the confusing location matrix, so that the platform can select the right workers to minimize the total moving distance.

The contributions of our work are summarized as follows:

1) We propose a privacy protection framework to protect the location privacy of tasks and workers without an online TTP in the task allocation phase.

2) We design a novel location privacy protection method based on matrix multiplication, which can protect the location privacy of tasks and workers at the same time and retain the relative distance information after confusing the location, so that the platform can quantify the relative distance between workers and tasks through the confused location.

3) We have done a lot of experiments on real datasets to verify our proposed method. The experimental results show that our method outperforms the state-of-the-art method.

2 Related Work

In this section, we briefly introduce the related work of task allocation optimization and privacy protection.

2.1 Task Allocation Optimization

Task allocation aims to allocate appropriate workers to tasks based on some optimization objectives, such as the cost of recruiting tasks on the platform, the task coverage, and the quality of task completion. In Ref. [8], the authors take the number of recruited workers as the optimization objective in the task allocation stage. The fewer the number of workers, the less the recruitment cost. The platform focuses on selecting a minimum number of workers and ensures that tasks can be covered. In Ref. [9], the authors propose a task coverage oriented assignment method based on the worker analysis and worker attribute model. They propose to migrate certain qualified workers to the less popular tasks for increasing the task coverage, and meanwhile optimize other performance factors. In Ref. [10], the authors take the quality of task completion as the optimization objective of task allocation.

In the location-based mobile crowdsensing task allocation, the recruited workers often need to move from their current location to the task location to complete the data collection task. Therefore, in this kind of mobile sensing, the cost of workers completing the task is closely related to the workers' moving distance. Therefore, in order to minimize the recruitment cost, the platform often takes minimizing the total moving distance as the priority of task allocation. In Ref. [11], the authors propose a tabu search algorithm to select workers, considering various task requirements (sensors, location accessibility and reliability) and work specifications (available sensor set and speed) as well as task completion sequence, and the total moving distance of all workers is minimized. In Ref. [12], the authors study two kinds of multi-task assignment schemes which make the number of completed tasks the most and the total moving distance the shortest. In Ref. [13], the authors propose ActiveCrowd, a worker selection framework for multitask MCS environments under two situations, for time-sensitive tasks. Workers are required to move to the task venue intentionally, and the goal is to minimize the total distance moved. For delay-tolerant tasks, the goal is to minimize the total number of workers. None of the above work has considered the issue of privacy protection.

2.2 Privacy-Preserving Task allocation

The problem of privacy protection has attracted more and more attention. Until now, many privacy protection technologies have been used in MCS, such as anonymization, obfuscation, encryption, and authentication[3], for the location-based mobile crowdsensing. In Ref. [14], the authors formulate a mixed-integer nonlinear programming problem to minimize the expected travel distance of the selected workers under the constraints of differential and distortion privacy. WANG et al.[15] propose a probabilistic winner selection mechanism (PWSM) to minimize the total travel distance with the obfuscated information from workers, by allocating each task to the worker most likely to be closest to it. The location privacy of workers is protected by adding Laplace noise to the distance between tasks and workers. WANG et al.[16] propose a location aggregation method, which groups users into a group to realize k-anonymity. However, all the above work only considers the location privacy of workers, while ignores the location privacy of tasks. Recently, there have been some works that consider both tasks' and workers' location privacy. For example, NI et al.[17] propose SPOON, a strong privacy-preserving mobile crowdsensing scheme supporting accurate task allocation based on geographic information and credit points of mobile users. Although this scheme protects the location privacy of workers and tasks, it can only determine whether the tasks and workers are in the same grid. In general, if there is no worker in the grid where a task is located, the platform needs to recruit workers from the nearby grid. This scheme is not suitable for this situation. YUAN et al.[18] devise a grid-based location protection method, which can protect the locations of workers and tasks. However, this method requires the task requester to calculate the neighbor grid code in a certain range in advance. If the workers in the range cannot meet the requirements of the task, the platform may need to recruit from outside the range. This method is not suitable for this situation. ZENG et al.[7] utilize the multi-secret sharing scheme to preserve location privacy in the MCS task assignment. However, this scheme is only suitable for edge computing environments with fog nodes, but not universal for traditional mobile crowdsensing. Different from the above work, the bilateral privacy-preserving framework (BPPF) in this paper can protect the location privacy of tasks and workers, and it is convenient for the platform to measure the relative distance between tasks and workers in any range at the same time. Our proposed BPPF has universality, which is suitable for most mobile sensing scenarios.

3 Preliminaries

In this section, we briefly introduce the models of the system including the threat model and the privacy model, and the design purpose. Table 1 lists the notations frequently used in this paper.

3.1 System Model

Our system mainly includes four entities: task requesters, workers, crowdsensing platform and offline TTP.

1) Task requesters: Task requesters can be individuals, groups and organizations. They have some location-based data collection tasks, such as collecting traffic flow data of a certain road section, monitoring air pollution of a certain area, and taking photos to investigate commodity prices of a supermarket. They do not have the resources to complete the task, so task requesters pay a certain amount of reward to upload the task to the platform for crowdsensing services.

2) Workers: Workers are users with mobile smart devices (such as mobile phones, tablets and smart-watches). They register as workers on the mobile sensing platform in advance, and use their own devices to complete the tasks assigned by the platform and get paid.

3) Crowdsensing platforms: They have enough computing resources and storage resources to provide mobile sensing services for task requesters. They receive some location-based task requests from task requesters and assign appropriate workers to the task. When a worker completes a task, the platform receives the data uploaded by the worker and sends it to the task requester.

4) Offline-TTP: The offline-TTP is responsible for generating the user's secret key, which is used to encrypt and decrypt the content of the task, as well as the disturbance matrix.

Definition 1 (location matrix): In this paper, we use a $k \times k$

▼Table 1. Notation List

Notations	Description
w	A worker
t	A task
$\boldsymbol{L}_w$	A matrix to represent the current location of a user
$\boldsymbol{L}_t$	A matrix to represent the sensing area of a task
$\boldsymbol{R}_w$	Confusion matrix of workers
$\boldsymbol{R}_T$	Confusion matrix of task requester
k	The size of the location matrix
α	The matrix assignment variable
$K_\#$	Task encryption key
SOD ($\boldsymbol{R}$)	Sum of diagonal elements of matrix $\boldsymbol{R}$

matrix to represent the grid of locations of tasks and workers.

The platform divides the whole geographic space covered by its services into $k \times k$ grids according to a certain granularity[19]. It is represented by a matrix $\boldsymbol{L} = \{\psi_{ij} | 0 \leqslant i \leqslant k-1, 0 \leqslant j \leqslant k-1\}$. Each element in the matrix corresponds to the divided grid in the actual geographic space. Therefore, the task requester uses a location matrix $\boldsymbol{L}_t$ to represent the location of the task. If the task is located in the i-th row and the j-th column of the grid space, the element corresponding to the task location matrix is represented by a large number, and the farther the elements of other positions in the position matrix are from the task, the smaller the number is. Similarly, workers use the same method to generate the location matrix $\boldsymbol{L}_w$.

Definition 2 (travel distance): In the location-based MCS task, we need the worker's current location and the task's location for task allocation. Because the worker needs to cross the city block to reach the task's location, we use Manhattan distance as the travel distance between the worker and the task, i.e, if the location index of a task in the grid space is (i, j) and the worker's location index is (m, n), the travel distance between tasks and workers is shown as Eq. (1):

$$d(t,w) = |i - m| + |j - n|. \tag{1}$$

3.2 Threat Model

The danger in MCS mainly comes from two aspects: external threats and internal threats[20]. External threats come from the outside of the MCS system and are initiated by external entities that have no contact with the MCS system. The main way is to steal the communication information of internal entities in the MCS system by monitoring the communication channel of entities in the MCS system, thus threatening the privacy of users. In this paper, the encryption technology is used to prevent such attacks. Internal threats come from the inside of the MCS system and are initiated by entities participating in MCS system activities. In this paper, we only consider privacy threats from the platform. We assume that offline-TTP is completely trusted, and similar to the traditional MCS privacy protection mechanism[7, 18, 21], we assume that the platform is honest-but-curious, which means that they are not only honest with the pre-defined protocol, but also curious about the privacy of the requester and worker.

3.3 Privacy Model

We divide the privacy model into two categories as follows:

1) Bilateral location privacy: The location of workers and tasks needs to be protected. In this paper, the location of tasks and workers is represented by a matrix $\boldsymbol{L}$, so we need to confuse the matrix $\boldsymbol{L}$ to protect the location privacy of workers and tasks.

2) Task content privacy: The task content may be sensitive to task requesters in some cases, so we need to protect task content from the platform.

3.4 Problem Formulation

Then, our problem has emerged as follows:

Bilateral privacy-preserving task allocation problem: In MCS, there is a set of tasks distributed in different locations $T = \{t_1, t_2, \ldots, t_m\}$, and a set of workers distributed in different locations $W = \{w_1, w_2, \ldots, w_n\}$. The honest-but-curious platform selects a set of workers based on the confusing information of tasks and workers. As shown in Eq. (2), the objective is to minimize the total travel distance of selected workers, where $\left(t_j, w_j\right)$ represents an assignment.

$$\sum_{j=1}^{|T|} \mathrm{d}(t_j, w_j) . \tag{2}$$

4 Overview of BPPF

In this section, we show the details of BPPF, a novel bilateral privacy protection framework, which can protect the location privacy of tasks and workers, as well as the content of tasks, while the task allocation phase does not need the participation of online TTP.

4.1 Service Setup

In order to build the MCS service, the platform first divides the whole geographic space into $k \times k$ grid space (for example, longitude and latitude), and uses the matrix $\boldsymbol{L}_{k \times k}$ to indicate that each element in the matrix corresponds to a grid area in the grid space, and randomly selects value $\alpha \in (k^2, +\infty)$.

4.2 User Registration

In order to participate in MCS activities and prove that they are legal users, task requesters and workers must register in offline TTP in advance and generate authenticated ID: $Sign_{SK}(ID)$, where SK is the authenticated private key of offline-TTP, $Sign_{SK}(ID)$ can be verified by the corresponding authentication public key of the offline-TTP, i.e., if $Verify_{PK}(Sign_{SK}(ID)) = ID$, then the user is legal, and the signature of the TTP can prevent malicious attacks from simply forging ID to participate in MCS activities. At the same time, offline-TTP generates two matrices randomly, namely $\boldsymbol{R}_T$, $\boldsymbol{R}_W \left(\boldsymbol{R}_T \times \boldsymbol{R}_W = E\right)$, and symmetric key $K_\#$, where $K_\#$ is used to encrypt and decrypt the task content, and R_T, R_W is the confusion matrix of task requester and worker to protect their location privacy.

4.3 Generation and Confusion of Location Matrix

Firstly, the whole geographic space is represented by a $k \times k$ matrix, and each element in the matrix represents a small grid in the grid space. In order to retain the relative distance information between the task and the worker after confusion, a novel assignment rule of location matrix is designed in this paper. The specific details are shown in Algorithm 1. If the task requester has a task in one of the small grids, the elements in the corresponding location matrix are represented by a maximum number, and the farther the other elements are from the task location, the smaller the assignment is. That is, if the task is located in the position with the matrix index of (i, j), then the position element $\psi_{x,y}$ is assigned the value of α^{2k-2}. The farther the other positions are from (i, j), the smaller the element ψ assignment is. An example is shown in Fig. 1. After the task requester and the worker generate the corresponding location matrix $\boldsymbol{L}$, in order to protect the location privacy of the task and the worker, they need to transform the actual location matrix $\boldsymbol{L}$ into the privacy-protected location matrix $\boldsymbol{L}^*$. Finally, the worker and the task requester upload the confusing location matrix to the platform.

Algorithm 1. Generating privacy-preserving location matrix

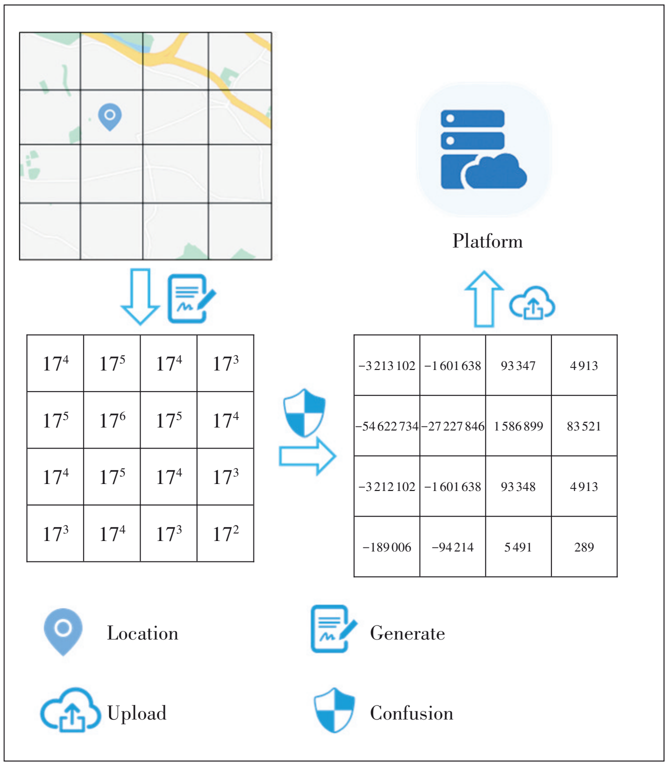
```
Input: task Location index $(i,j)$ , worker Location index $(m,n)$
Output: task confusion location matrix $\boldsymbol{L}_t^*$, user confusion location matrix $\boldsymbol{L}_w^*$
Initialization: $\boldsymbol{L}_t \leftarrow 0$, $\boldsymbol{L}_w \leftarrow 0$
for each $\psi_{(x,y)} \in \boldsymbol{L}_t$ do
    $\psi_{x,y} = \alpha^{2k - 2 - |i - x| - |j - y|}$
end for
for each $\psi_{(x,y)} \in \boldsymbol{L}_w$ do
    $\psi_{x,y} = \alpha^{2k - 2 - |m - x| - |n - y|}$
end for
$\boldsymbol{L}_w^* = \boldsymbol{L}_w \times \boldsymbol{R}_W$
$\boldsymbol{L}_t^* = \boldsymbol{R}_T \times \boldsymbol{L}_t^T$
Return $L_w^*$, $L_t^*$
```

4.4 Privacy-Preserving Task Submission and Worker Participation

The task requester submits an encrypted task to the platform. For a task, the task requester submits its verifiable ID, the encrypted task content $Enc(K_{\#}, t)$, and the location matrix after confusion $\boldsymbol{L}_T^*$, i.e., $task \leqslant t$, S_t, $Enc(K_{\#}, t)$, $\boldsymbol{L}_t^* >$, where S_t denotes verifiable ID, $Enc(K_{\#}, t)$ represents the task content encrypted with symmetric secret key, and L^*_T represents the location matrix of the task after confusion. After the task requester submits the task to the platform, the platform first passes the verification S_t to check whether the task requester is a legal user, and then the platform assigns the right worker to the task. In order to participate in MCS activities, workers registered on the platform need to upload their verifiable ID and the confusing location matrix $\boldsymbol{L}_w^*$ to the platform, i.e, $user = < w, S_w, \boldsymbol{L}_w^* >$, where S_w denotes verifiable ID, and $\boldsymbol{L}_w^*$ represents the location matrix of the task after the confusion. After receiving the worker's information, the platform first checks whether the worker is legal, then stores the user's information locally and adds it to the candidate worker set.

4.5 Privacy-Preserving Task Matching and Distribution

After receiving the messages from the task and the worker, the platform allocates workers based on the confusion location matrix of task and worker $\boldsymbol{L}_t^*$, $\boldsymbol{L}_w^*$. First, the platform computes $\boldsymbol{R} = \boldsymbol{L}_t^* \times \boldsymbol{L}_w^*$, and then takes the sum of diagonal elements of result matrix $\boldsymbol{R}$, SOD ($\boldsymbol{R}$) as the standard to measure the distance between a pair of tasks and workers. The larger the value of SOD ($\boldsymbol{R}$), the shorter the actual distance. The details will be explained in Theorem 1.



▲Figure 1. An example of generating privacy-preserving location matrix

Each task needs one worker to complete, and the purpose of the platform is to minimize the total moving distance of the assigned workers. Because the relative distance SOD ($\boldsymbol{R}$) obtained by the platform is inversely proportional to the actual distance, we can transform the problem into a maximum weight bipartite graph matching problem. The task and the worker are two disjoint point sets in the bipartite graph. The relative distance between the task and the worker SOD ($\boldsymbol{R}$) is used as the weight of the edge between the task and the worker. Our goal is to find a set of edge sets to maximize the total weight, that is, the total moving distance of assigned workers is the minimum.

Theorem 1. The actual distance d between workers and tasks is inversely proportional to the relative distance SOD ($\boldsymbol{R}$). That is, if $d_1 < d_2$, then $SOD_1(\boldsymbol{R}) > SOD_2(\boldsymbol{R})$.

Proof. Assume the task's location is (i,j), the corresponding location matrix is $\boldsymbol{L}_t$, the worker's location is (m,n), the corresponding location matrix is $\boldsymbol{L}_w$, $\boldsymbol{R} = \boldsymbol{L}_t^* \times \boldsymbol{L}_w^*$, and SOD ($\boldsymbol{R}$) is the sum of diagonal elements of matrix $\boldsymbol{R}$. The formulas are:

$$d_1 < d_2 \ , \tag{3}$$

$$SOD_1(\boldsymbol{R}) > SOD_2(\boldsymbol{R}), \tag{4}$$

where d_1 is the Manhattan distance between task 1 and worker 1 and d_2 is the Manhattan distance between task 2 and worker 2. Formulas (3) and (4) can be transformed into the following form:

$$\left|i_1 - m_1\right| + \left|j_1 - n_1\right| < \left|i_2 - m_2\right| + \left|j_2 - n_2\right|. \tag{5}$$

$$\sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \alpha^{4k-4-\left(\left|i_1 - x\right| + \left|j_1 - y\right| + \left|m_1 - x\right| + \left|n_1 - y\right|\right)} > \sum_{x=0}^{k-1} \sum_{y=0}^{k-1} \alpha^{4k-4-\left(\left|i_2 - x\right| + \left|j_2 - y\right| + \left|m_2 - x\right| + \left|n_2 - y\right|\right)}. \tag{6}$$

The expansion of $SOD(\boldsymbol{R})$ is the accumulation of k^2 items, which is recorded as ζ. For the next proof, first of all, we want to get the value range of ζ. According to the properties of the exponential function, the minimum value of ζ is not less than 0. When the index number is the largest, the value of ζ is the largest. Therefore, the problem of finding the maximum value of ζ is to find a point (x, y) in the two-dimensional space of $\{(x, y) \mid 0 \leqslant x \leqslant k-1, 0 \leqslant y \leqslant k-1\}$ so that the sum of the Manhattan distance from this point to point (i,j) plus the Manhattan distance to point (m,n) is the minimum. According to the image, the minimum value of $|i - x| + |j - y| + |m - x| + |n - y|$ is $|i - m| + |j - n| = d$, so the maximum value of ζ is α^{4k-4-d}, and finally the value range of ζ is $(0, \alpha^{4k-4-d}]$. So formula (6) can be transformed into formula (7).

$$\alpha^{4k-4-d_1} + \zeta_1 + \zeta_2 + \cdots + \zeta_{k^2-1} > \alpha^{4k-4-d_2} + \zeta_1^{,} + \zeta_2^{,} + \cdots + \zeta_{k^2-1}. \tag{7}$$

By narrowing the left side of inequality (7) and enlarging the right side, we get inequality (8):

$$\alpha^{4k-4-d_1} > k^2 * \alpha^{4k-4-d_2}. \tag{8}$$

We divide both sides of inequality (8) by α^{4k-4-d_2}, and then the inequality (9) is obtained.

$$\alpha^{d_2 - d_1} > k^2, \tag{9}$$

where d_1 and d_2 is the Manhattan distance in the grid space and the minimum value of d_2 is d_1+1. From inequality (9), we can draw the following conclusion: if $\alpha > k^2$ and $d_1 < d_2$, $\mathrm{SOD}_1(\boldsymbol{R}) > \mathrm{SOD}_2(\boldsymbol{R})$ must be true.

4.6 Task Extraction and Execution

After receiving the task assigned by the platform, the worker first performs the decryption operation to obtain the task content $Dec(K_\#, Enc(K_\#, t))$. Only authenticated legitimate users have the secret $K_\#$. After the worker completes the task, the collected data is encrypted with the secret key and uploaded to the platform $< w, S_w, Enc(K_\#, date) >$. After receiving the information forwarded by the platform, the task requester first verifies whether w is a legitimate user and then decrypts the sensor data with the secret key.

5 Experimental Evaluation

In this section, we evaluate the performance of BPPF by theoretical analysis and extensive experiments based on real-world datasets.

5.1 Experimental Setup

1) Datasets. In the simulations, we adopt two widely used real-world datasets: Feeder[22] and GeoLife[23]. Feeder contains four kinds of data, i.e., the cellphone call detail records data, smartcard data, taxicab GPS data, and bus GPS data collected from Shenzhen. GeoLife is from the user's mobile phone data which records the user's mobile trajectory data. For the Feeder dataset, we preserve the tax ID, the latitude, and the the longitude to construct the worker's attributes. The selected users locate in the area with a latitude ranging from 22.488899 to 22.7491, and a longitude ranging from 113.801048 to 114.241135. For the GeoLife dataset, we construct workers' attributes by extracting the user ID, the latitude and longitude. The selected users locate in the area with a latitude ranging from 40.013225 to 40.424714, and a longitude ranging from 116.327406 to 116.655039. We split the urban area of Feeder and GeoLife into 40 × 40 grids, each with the size of 1 km × 1 km.

2) Baseline approaches. In this paper, we compare our strategy with the following benchmarks: No-privacy, the strategy which is the no-privacy-version of our strategy without the confusion strategies; PriRadar[18], a grid-based bilateral privacy protection framework. For the convenience of comparison, we set the hash table $H(t) = 1$, $H(w) = 1$. This means that a task needs to be completed by one worker, and a worker can only complete one task, meanwhile we do not set the noise hash code.

3) Evaluation metrics. We use the following metrics to evaluate the compared algorithms: the number of completed tasks and the travel distance, which are the main metrics to evaluate our strategy.

4) Setting. All the algorithms were implemented in Java. All the experiments were evaluated on a notebook with Intel Core i7-4720HQ central processing unit (CPU), of which the clock rate is 2.60 GHz and the memory is 8.00 GB. The operation system is 64-bit Windows 10.

5.2 Number of Completed Tasks Evaluation

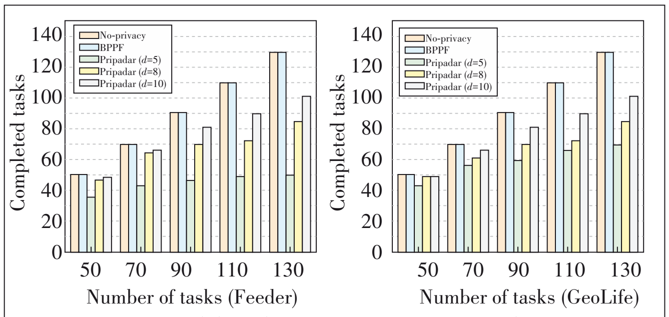
We first compare the performance of No-privacy, BPPF, and Pripadar in the number of tasks completed. In the specific experiment, we randomly select the user information from the dataset to construct the worker attributes. As shown in Fig. 2, we can see the No-privacy method and our BPPF method can allocate all tasks by changing the number of tasks from 50 to 130, because both methods can find a maximum matching scheme in task allocation, while the number of tasks completed by Pripadar method increases with the increase of search distance under the same number of tasks. This is because the effect of Pripadar is related to the search distance (*d*). With the increase of search distance, the platform has more workers to choose and a task is more likely to be assigned to a worker. Then we compare the numbers of completed tasks with different algorithms with respect to different numbers of workers, and we change the number of workers from 50 to 130. As shown in Fig. 3, we can see that the No-privacy and BPPF methods can assign all tasks even when the number of workers is equal to the number of tasks. While in the case of the same search distance, the number of tasks completed by Pripadar method increases with the increase of the number of workers, because with the increase of the number of workers, there are more workers to choose within the search distance of a task.

5.3 Travel Distance Evaluation

We then evaluate the travel distance. The travel distance is defined as the sum of the Manhattan distances between all assigned workers and task pairs. For consistency, we set the search distance (*d*) of Pripadar so that all tasks can be assigned. As shown in Fig. 4, the moving distances of the three methods increase with the number of tasks. We can observe that the No-privacy method and our BPPF method are always better than Pripadar. As shown in Fig. 5, the travel distance of the three methods decreases with the increase of the number of workers. This is because with the increase of the number of workers, a task is more likely to be assigned to a closer worker. The result of GeoLife dataset does not change obviously. Because the user location distribution of GeoLife dataset is relatively concentrated, while the user location distribution of Feeder is relatively scattered, the workers assigned by GeoLife dataset are already the optimal workers when the number of workers is small. Even if the number of workers increases, the change of moving distance will not be greatly reduced. From Figs. 4 and 5, we can observe that the No-privacy method and our BPPF method are always better than Pripadar, and our BPPF method has almost no difference with No-privacy, which indicates that our method can achieve accurate task allocation under the condition of protecting tasks' and workers' positions.

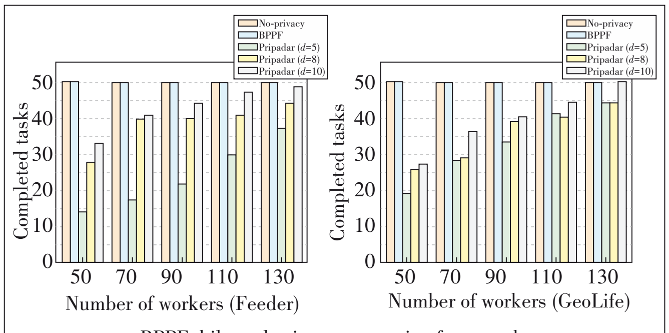
5.4 Location Privacy

We also do experiments to verify our proposed bilateral privacy protection mechanism based on the matrix. We set the task location as (0, 0), then 100 workers are set in different grids, and the relative distance SOD (***R***) calculated by the platform is shown in Fig. 6. We can observe that the farther the worker is from the task, the smaller the relative distance is,



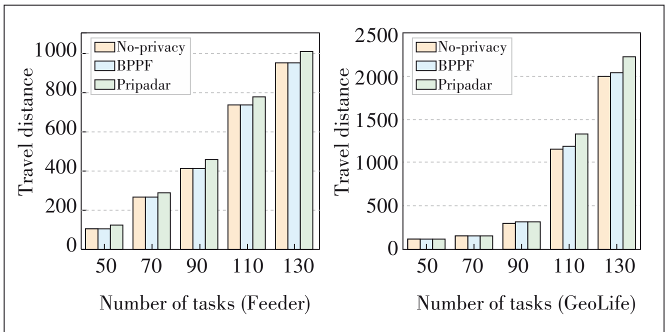
BPPF: bilateral privacy-preserving framework

▲Figure 2. Completed tasks vs. number of tasks



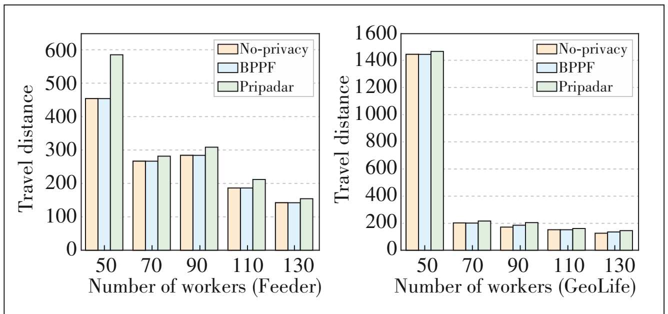
BPPF: bilateral privacy-preserving framework

▲Figure 3. Completed tasks vs. number of workers



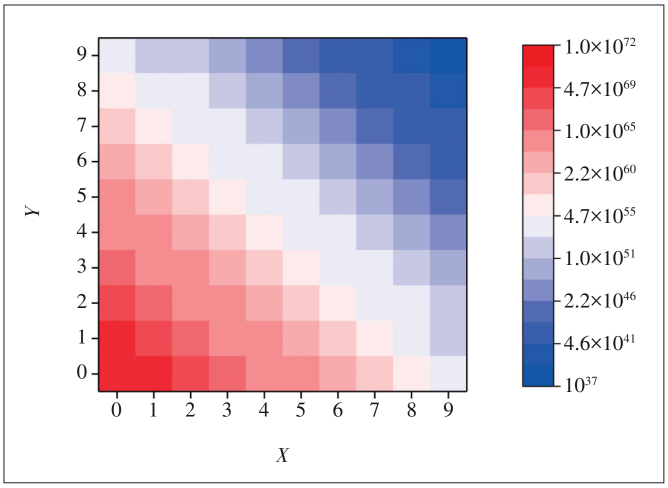
BPPF: bilateral privacy-preserving framework

Figure 4. Travel distance vs. number of tasks



BPPF: bilateral privacy-preserving framework

▲Figure 5. Travel distance vs. number of workers



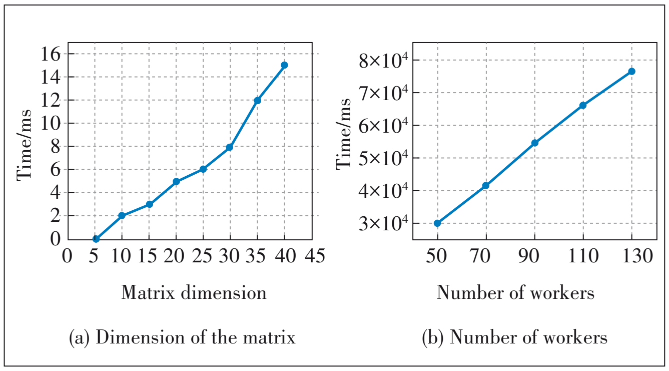
▲**Figure 6. Sum of diagonal elements of result matrix *R***

which also proves Theorem 1 mentioned in Section 4.

The sensing region of a task is represented by a matrix $\boldsymbol{L}_t$, which is randomized by a confusing matrix $\boldsymbol{R}_T$ to generate the privacy preserving location matrix $\boldsymbol{L}_t^*$, and the worker's location is also replaced by a privacy preserving location matrix $\boldsymbol{L}_w^*$. The platform can not know any location information of tasks or workers through these two matrices $\boldsymbol{L}_t^*$ and $\boldsymbol{L}_w^*$. The elements and location mapping rules of the actual location matrix are changed by matrix multiplication $\boldsymbol{L}_t \times \boldsymbol{R}_T$, and this change varies with the confusion matrix $\boldsymbol{R}_T$, so the platform cannot find the information of the actual location matrix from the privacy protected location matrix. After the platform receives the privacy protection location mentioned above, the user's actual location is hidden in the grid. Even if the user's actual grid is obtained, it can protect the user's location privacy to a certain extent.

5.5 Computational Overhead

Finally, we discuss the computation cost of our method. Because our privacy-preserving method is based on matrix multiplication, the computation cost in BPPF depends on the dimension of the matrix, such as the granularity of the grid. The smaller the granularity, the larger the precision and the size of the matrix. For task requesters and workers, the computation cost is $M_{k \times k} \times M_{k \times k}$, which means one matrix multiplication operation. For the platform, in order to allocate tasks, the platform needs to calculate the sum of diagonal elements of two matrix multiplication result matrices, so the platform can only calculate diagonal elements, and the computation cost is $1/k \times \left(M_{k \times k} \times M_{k \times k}\right) + O\left(n^3\right)$, where $O(n^3)$ means the time complexity of the weighted bipartite graph matching algorithm. The time taken by a user to generate the privacy protection location matrix in the case of different size matrices is shown in Fig. 7. It is acceptable for users to spend such computing resources to ensure privacy. Finally, Fig. 7 shows the running time of the whole framework with a fixed number of tasks of 50.



(a) Dimension of the matrix

(b) Number of workers

Figure 7. Time vs. dimension of the matrix and number of workers

6 Conclusions

In this paper, we study the problem of bilateral privacy protection in mobile sensing. We map the location of workers and tasks to the grid, use a novel location matrix generation method to represent the user's location information, and propose a location matrix obfuscation method based on matrix multiplication, which can preserve the relative distance information between tasks and workers while protecting their location privacy. Finally, extensive simulations based on real world datasets verify the performance of our method.

References

[1] LUO T, HUANG J W, KANHERE S S, et al. Improving IoT data quality in mobile crowd sensing: a cross validation approach [J]. IEEE Internet of Things journal, 2019, 6(3): 5651 – 5664. DOI: 10.1109/JIOT.2019.2904704

[2] GANTI R K, YE F, LEI H. Mobile crowdsensing: current state and future challenges [J]. IEEE communications magazine, 2011, 49(11): 32 – 39. DOI: 10.1109/MCOM.2011.6069707

[3] LIU Y T, KONG L H, CHEN G H. Data-oriented mobile crowdsensing: A comprehensive survey [J]. IEEE communications surveys & tutorials, 2019, 21(3): 2849 – 2885. DOI: 10.1109/COMST.2019.2910855

[4] WANG L Y, ZHANG D Q, YANG D Q, et al. Sparse mobile crowdsensing with differential and distortion location privacy [J]. IEEE transactions on information forensics and security, 2020, 15: 2735 – 2749. DOI: 10.1109/TIFS.2020.2975925

[5] XIAO M J, GAO G J, WU J, et al. Privacy-preserving user recruitment protocol for mobile crowdsensing [J]. IEEE/ACM transactions on networking, 2020, 28 (2): 519 – 532. DOI: 10.1109/TNET.2019.2962362

[6] ZHANG Y H, LI M, YANG D J, et al. Tradeoff between location quality and privacy in crowdsensing: An optimization perspective [J]. IEEE Internet of Things journal, 2020, 7(4): 3535 – 3544. DOI: 10.1109/JIOT.2020.2972555

[7] ZENG B, YAN X F, ZHANG X L, et al. BRAKE: bilateral privacy-preserving and accurate task assignment in fog-assisted mobile crowdsensing [J]. IEEE systems journal, 9278, (99): 1 – 12. DOI: 10.1109/JSYST.2020.3009278

[8] GAO G J, XIAO M J, WU J, et al. DPDT: A differentially private crowd-sensed data trading mechanism [J]. IEEE Internet of Things journal, 2020, 7(1): 751 – 762. DOI: 10.1109/JIOT.2019.2944107

[9] SONG S W, LIU Z D, LI Z J, et al. Coverage-oriented task assignment for mobile crowdsensing [J]. IEEE Internet of Things journal, 2020, 7(8): 7407 – 7418. DOI: 10.1109/JIOT.2020.2984826

[10] WANG T, XIE X K, CAO X, et al. On efficient and scalable time-continuous spatial crowdsourcing: full version [EB/OL]. (2020-10-29) [2021-01-25]. https://arxiv.org/abs/2010.15404

[11] AKTER S, YOON S. Location-aware task assignment and routing in mobile crowd sensing [C]//2020 International Conference on Information and Communication Technology Convergence (ICTC). Jeju, Korea (South): IEEE, 2020: 51 – 53. DOI: 10.1109/ICTC49870.2020.9289316

[12] LIU Y, GUO B, WANG Y, et al. TaskMe: multi-task allocation in mobile crowd sensing[C]//Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing. Heidelberg, Germany: ACM, 2016: 403 – 414. DOI: 10.1145/2971648.2971709

[13] GUO B, LIU Y, WU W L, et al. ActiveCrowd: A framework for optimized multitask allocation in mobile crowdsensing systems [J]. IEEE transactions on human - machine systems, 2017, 47(3): 392 – 403. DOI: 10.1109/THMS.2016.2599489

[14] WANG L Y, YANG D Q, HAN X, et al. Mobile crowdsourcing task allocation with differential-and-distortion geoobfuscation [J]. IEEE transactions on dependable and secure computing, 2021, 18(2): 967 – 981. DOI: 10.1109/TDSC.2019.2912886

[15] WANG Z B, HU J H, LV R, et al. Personalized privacy-preserving task allocation for mobile crowdsensing [J]. IEEE transactions on mobile computing, 2019, 18(6): 1330 – 1341. DOI: 10.1109/TMC.2018.2861393

[16] WANG X, LIU Z, TIAN X H, et al. Incentivizing crowdsensing with location-privacy preserving [J]. IEEE transactions on wireless communications, 2017, 16(10): 6940 – 6952. DOI: 10.1109/TWC.2017.2734758

[17] NI J B, ZHANG K, XIA Q, et al. Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing [J]. IEEE transactions on mobile computing, 2020, 19(6): 1317 – 1331. DOI: 10.1109/TMC.2019.2908638

[18] YUAN D, LI Q, LI G L, et al. PriRadar: A privacy-preserving framework for spatial crowdsourcing [J]. IEEE transactions on information forensics and security, 2020, 15: 299 – 314. DOI: 10.1109/TIFS.2019.2913232

[19] YAN K, LUO G C, ZHENG X, et al. A comprehensive location-privacy-awareness task selection mechanism in mobile crowd-sensing [J]. IEEE access, 2019, 7: 77541 – 77554. DOI: 10.1109/ACCESS.2019.2921274

[20] GISDAKIS S, GIANNETSOS T, PAPADIMITRATOS P. Security, privacy, and incentive provision for mobile crowd sensing systems [J]. IEEE Internet of Things journal, 2016, 3(5): 839 – 853. DOI: 10.1109/JIOT.2016.2560768

[21] ZHUO G Q, JIA Q, GUO L K, et al. Privacy-preserving verifiable data aggregation and analysis for cloud-assisted mobile crowdsourcing[C]//IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications. San Francisco, USA: IEEE, 2016: 1 – 9. DOI: 10.1109/INFOCOM.2016.7524547

[22] ZHANG D S, ZHAO J J, ZHANG F, et al. Feeder: supporting last-mile transit with extreme-scale urban infrastructure data [C]//Proceedings of the 14th International Conference on Information Processing in Sensor Networks. Seattle, USA: IPSN, 2015: 226 – 237. DOI: 10.1145/2737095.2737121

[23] ZHENG Y, ZHANG L Z, XIE X, et al. Mining interesting locations and travel sequences from GPS trajectories [C]//Proceedings of the 18th international conference on World Wide Web. Madrid, Spain: ACM, 2009: 791 – 800. DOI: 10.1145/1526709.1526816

Biographies

LIU Junyu received his bachelor's degree in computer science and technology from Jilin University, China in 2019. Currently, he is pursuing for the master's degree in computer science and technology at Jilin University. His current research interests include mobile crowdsensing and privacy preserving in mobile computing.

YANG Yongjian received his B.E. degree in automatization from Jilin University of Technology, China in 1983, M.E. degree in computer communication from Beijing University of Post and Telecommunications, China in 1991, and Ph.D. in software and theory of computer from Jilin University, China in 2005. He is currently a professor and a Ph.D. supervisor at Jilin University, Director of Key lab under the Ministry of Information Industry, and Standing Director of the Communication Academy. His research interests include network intelligence management, wireless mobile communication and services, and wireless mobile communication.

WANG En (wangen@jlu.edu.cn) received his B.E. degree in software engineering from Jilin University, China in 2011, and his M.E. degree and Ph.D. in computer science and technology from Jilin University in 2013 and 2016. He is currently an associate professor in the Department of Computer Science and Technology, Jilin University. His current research interests include the efficient utilization of network resources, scheduling and drop strategy in terms of buffer-management, energy-efficient communication between human-carried devices, and mobile crowdsensing.