IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 11, 2022, accepted March 17, 2022, date of publication March 23, 2022, date of current version March 30, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3161544

Differentially Private Task Allocation Algorithm Under Preference Protection

JUNTAO HAN[1] **AND SHUYUE CAI**[2]

[1]School of E-Commerce and Logistics Management, Henan University of Economics and Law, Zhengzhou, Henan 450046, China
[2]School of Management Engineering, Zhengzhou University, Zhengzhou, Henan 450001, China

Corresponding author: Juntao Han (hanjuntao126@126.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61802110, and in part by the Characteristic Backbone Discipline in Henan Province's "Modern Service Discipline Group."

ABSTRACT Mobile crowdsensing has been widely applied as a kind of perception paradigm, and task allocation is a fundamental research issue in mobile crowdsensing. Existing task allocation algorithms under differential privacy are not suitable for preference protection scenarios as they may inject too much noise. To this end, in this paper, we propose a differentially private task allocation algorithm with preference protection, referred to as SLEPT. In SLEPT, we divide privacy budget into three parts. Specifically, we first use one part of privacy budget to perturb the location of each worker. Then we use another part of privacy budget to perturb the preference information of him. In particular, to relieve the problem that perturbation may lead to that tasks will not be allocated, we propose a two-phase preference collection mechanism called TPC. Finally, we propose a task allocation sequential updating mechanism TASU using the remaining privacy budget. It aims to reduce the travel distance of workers and improve the success rate of task allocation. Theoretical analysis shows that SLEPT satisfies differential privacy. Time complexity analysis shows that it is linearly related to the number of tasks. The results on two public datasets verify the effectiveness of SLEPT. It is worth noting that although SLEPT is proposed for task allocation, its idea is also applicable to other crowdsensing scenarios, such as high-dimensional data collection.

INDEX TERMS Crowdsensing, differential privacy, geo-indistinguishability, preference protection, task allocation.

I. INTRODUCTION

Mobile crowdsensing is a kind of perception paradigm with excellent application prospects. It makes full use of the processing power of smartphones with multiple sensors. It enables ordinary people to complete tasks that had to be done by professionals in the past [1]. Due to the advantages of crowdsensing, it has attracted extensive attention from the academia and the industry. It has a wide range of applications in the real world, such as environmental monitoring [2] and intelligent transportation [3]. As a typical example, WAZE, an application for traffic monitoring and route navigation, has obtained more than 100 million downloads on Google Play, with a user score of 4.6 (5 grades in total) [4].

The running process of a typical crowdsensing system is as follows. First, participants are registered as candidate workers. When a new task arrives, each worker first selects the task he wants to do and submits his location to the server. Then the platform selects an appropriate subset of candidate workers to complete the task. Finally, these workers go to the target location to complete the task and submit the perceived result values to the initiator of crowdsensing. The task allocation [5] from the process is a core step and the basis of crowdsensing. In particular, the travel distance of workers to the task location and the number of tasks successfully allocated are essential issues to be considered in task allocation. If the travel distance is too long, a worker may not be willing to perform the task. It needs the consumption of lots of material resources. For the initiators of crowdsensing, increasing travel distance will reduce their stickiness to the crowdsensing platform, such as high salary expenditure and large delay of sensing results. Moreover, a low number of successful task allocations will lead to the initiators of crowdsensing being unwilling to allocate tasks on the platform

The associate editor coordinating the review of this manuscript and approving it for publication was Diana Gratiela Berbecaru.

again. Therefore, according to the previous work, we use travel distance and the number of successfully assigned tasks as the utility measures of task allocation.

Suppose the platform knows the location of the candidate workers. In that case, directly allocating tasks to nearby workers can minimize the travel distance. However, the leakage of workers' locations often leads to the leakage of home addresses, work units and other information, which leads to workers' unwillingness to participate in the crowdsensing system. Moreover, the preference information of workers (that is, the set of tasks that workers want to do) will also reveal the location information of workers from the side. The reason is that workers may first choose the tasks as the candidate task whose locations are close to their homes or work unit [6]–[8].

The existing privacy protection technology based on Cloaking [9] is often vulnerable to background knowledge attacks. For example, suppose the adversary foresees that the user is a student. Then he can safely infer that the user is in the school area when the Cloaking area includes schools and government offices. Furthermore, some existing solutions need the participation of users and crowdsensing platforms, and the support of additional trusted platforms, which make them challenging to deploy. For example, the cellular service provider which needs workers plays an important coordinating role between the user and the crowdsensing platform to protect privacy. However, in practice, the cellular service provider may lack the motivation to participate. In addition, there is no scheme to protect workers' preferences [10].

To this end, differential privacy (DP) [11] has emerged as the gold standard for privacy protection recently. Compared with the traditional Cloaking-based technology, it provides a quantifiable privacy protection effect that has nothing to do with background knowledge. Primarily, local differential privacy (LDP) [12] and Geo-Indistinguishability (Geo-I) [13] are used to protect workers' preference information and location privacy. Both of them do not need trusted server settings. They can provide users with quantifiable privacy protection strength as the same as DP. Typical LDP and Geo-I implementation mechanisms are random response (RR) [14] and Planar Laplacian (PL) [13], respectively.

Data are perturbed locally before the system uploads them to the server, fundamentally protecting user privacy. At present, the LDP model has been used in many software to provide privacy protection, such as Google's Chrome Browser [14], Apple's iOS [15], and Microsoft's Windows Insiders [16]. Geo-I model has been applied to many software to provide location privacy protection, such as LP-Guardian [17] and LP-Doctor [18]. Therefore, this paper will use Geo-I and LDP to protect workers' location privacy and preference privacy.

However, the direct adoption of LDP and Geo-I to task allocation under preference protection will face two technical obstacles:

1) Perturbing whether a task is in a worker's preference set requires a lot of privacy budget segmentation. For example, suppose that the system allocates 100 tasks, and a worker $\boldsymbol{u}$ has 60 tasks left after eliminating the tasks he can't do. Then, to protect $\boldsymbol{u}$'s preference information, the privacy budget ε needs to be divided into 60 parts. Then using the existing LDP implementation mechanism, such as RR, may make the collected tasks in preference not required by u.
2) Suppose that a task $\boldsymbol{t}$ is only in the preference sets of workers $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$. After protecting workers' preferences, it may not be in the preference set of any workers. Then the system will never allocate the task $\boldsymbol{t}$.

To overcome these two obstacles, we propose SLEPT (taSk aLlocation prEference ProTection) algorithm. In SLEPT, we divide the privacy budget into three parts: ε_1, ε_2, and ε_3. First, the PL and the part ε_1 are used to perturb the position of each worker. Then, the part ε_2 is used to collect the distribution of preference information of each worker. Next, each worker uses the part ε_3 to perturb his preference set. Finally, the system allocates the tasks according to the perturbed preference sets and location information.

In summary, the main contributions of this paper are as follows:

1) A novel differentially private task allocation algorithm SLEPT is proposed. We formally give its privacy and complexity. The main idea is that the server adaptively allocates the privacy budget, collects the preference set information of each worker. And then, it needs to assign tasks serially and more to ensure that the travel distance is as small as possible. SLEPT is not only suitable for task allocation but also high-dimensional data collection.
2) In SLEPT, we design a two-phase preference collection (TPC). In TPC, the server adaptively allocates the privacy budget according to the number of times each task appears in the preference set.
3) In SLEPT, we develop a mechanism of task allocation updating serially (TASU). In TASU, each worker chooses the nearest task until the system assigns all tasks or traverses all workers.
4) Privacy analysis shows that the proposed SLEPT algorithm satisfies differential privacy. Experimental results on two real datasets demonstrate the effectiveness of the proposed scheme.

The other parts of this paper are arranged as follows. Section 2 describes the related work involved in this paper. Section 3 analyzes the details of SLEPT. Section 4 verifies the effectiveness. We summarize this paper in Section 5.

II. RELATED WORK

A. GEO-INDISTINGUISHABILITY

Husain *et al.* [12] extended the traditional differential privacy for processing numerical data to location protection scenarios and proposed Geo-Indistinguishability (Geo-I). In particular,

they implemented Geo-I using planar Laplacian (PL). Bordenabe *et al.* [19] explored constructing a mechanism to minimize the loss of service quality. They used linear programming technology to obtain the optimal noise function. Yu *et al.* [20] considered Geo-I and expected reasoning error to be two complementary concepts of location privacy and conducted formal research on them. Oya *et al.* [21] studied other aspects of privacy to avoid ''error'' choices. They further proposed a new mechanism and proved its effectiveness, which is optimal in terms of comparing adversaries' average errors.

Pyrgelis *et al.* [22] evaluated the impact of releasing aggregate location time-series on the privacy of individuals contributing to the aggregation. Chatzikokolakis *et al.* [23] studied these methods to improve the utility of location obfuscation. They provided such solutions for both infinite (continuous or discrete) and large but finite domains of locations, using a Bayesian remapping procedure as a key ingredient. ElSalamouny and Gambs [24] proposed the noise functions to satisfy a generic location privacy notion, obfuscating a user's location. Wang *et al.* [25] proposed a method to protect the location in mobile crowd sensing using local differential privacy preference. Takagi *et al.* [26] found the additional privacy loss of Geo-I for LBSs over road networks. They further proposed a new privacy concept to protect location privacy and designed a graph index mechanism. Oya *et al.* [27] provided an alternative formulation of Geo-I as an adversary error. They used it to show the tradeoff between privacy and utility.

B. LOCAL DIFFERENTIAL PRIVACY

In recent years, the research on local differential privacy has received great attention. Duchi *et al.* [28] proposed a data collection framework that satisfied the local differential privacy (LDP) mean calculation and statistical risk minimization based on information theory. Erlingsson *et al.* [14] proposed the RAPPOR mechanism based on randomized response, collecting binary attribute values by LDP. Based on RAPPOR, Fanti *et al.* [29] extended it to more complex statistical tasks based on expectation-maximization algorithm, such as joint distribution statistics and association testing. They expanded the scope of RAPPOR to classification attributes containing a large number of unknown values (such as the homepage data of user's browser). However, when the data dimension is high, the mechanism has high time complexity and slow convergence speed. Kairouz *et al.* [30] proposed an LDP mechanism for frequency estimation of binary single attribute data and proved that it is optimal in the case of low privacy. After that, they further studied how to deal with categorical data with any number of values [31]. Bassily and Smith [32] proposed an asymptotically optimal privacy scheme, which can construct a concise histogram of classification attributes under the condition of LDP.

Nguyên *et al.* [33] proposed a data collection method called Harmony. In particular, for each piece of high-dimensional data, the way randomly selects a dimension of the data. If the dimension corresponds to continuous data, the collection method is based on continuous value. If the dimension corresponds to discrete data, it is collected based on the discrete collection method. To obtain the frequent items of multidimensional data, Qin *et al.* [34] proposed a two-phase data collection method called LDP Miner. In the first phase, the candidate space of frequent items is initially determined from noise data based on a concise histogram mechanism. In the second phase, the method obtains accurate frequent items based on RAPPOR mechanism. Wang *et al.* [35] proposed an optimized LDP implementation mechanism for collecting numerical single attribute data, and gave their multi-attribute extension schemes.

C. TASK ALLOCATION BASED ON DIFFERENTIAL PRIVACY

To *et al.* [36] introduced a private framework of differential privacy to enable workers to participate without compromising their location privacy. In particular, they proposed an analysis model to measure the probability of task completion. They found the appropriate partition to ensure a high success rate task allocation in the case of uncertain worker location. Wang *et al.* [37] used the Geo-I method to protect the location and privacy of workers and mixed integer nonlinear programming to minimize the expected travel distance of selected workers. Wang *et al.* [38] provided a personalized probabilistic winner selection mechanism considering the number of workers with different protection needs. It assigned each task to a maximum probability with the closest task location. Wang *et al.* [39] proposed a method to maximize the work efficiency of mobile workers and a future location coverage protection scheme under location privacy guarantee. To *et al.* [40] proposed a three-stage framework to compromise the location privacy of staff and tasks. They designed three techniques to quantify the probability of realizability between tasks and workers. Gong *et al.* [41] proposed a new framework to achieve high task coverage through evaluation. In addition, there was an incentive pricing mechanism to guide workers to collect sensing data in low worker density areas. For the first time, Tao *et al.* [42] tried to carry out differential private online task allocation competition ratio under the premise of ensuring security. Song *et al.* [43] used the SAT model to solve the task assignment problem requiring multiple skilled workers. The task assignment result had the shortest worker travel distance and the least cost of employing workers, and proposed two greedy algorithms for task allocation. Béziaud *et al.* [44] solved the problem of privacy protection in a task assignment scenario requiring workers with different skills. Perturbing the worker's skill vector was in the way of satisfying differential privacy, so that the crowdsourcing platform could assign tasks without knowing the exact skill points of workers.

To sum up, no existing researches can conduct task allocation with high utility while protecting workers' locations and preference information as they could inject too much noise and result in many tasks cannot be assigned.

TABLE 1. Frequent notations.

Variable	Description
M	The number of workers
N	The number of tasks
U	The set of workers
T	The set of tasks
t_i	The i-th task
u_i	The i-th worker
S_i	The preference set of the i-th worker
Q	Task statistics after perturbation
F	The array of normalization of Q array
R	The result of task allocation
LDP	Local Differential Privacy
RR	Random Response mechanism
PL	Planar Laplacian mechanism

III. PRELIMINARIES

Table 1 shows the frequent notations which may be used in this paper.

A. TASK ALLOCATION

As shown in (1), given the worker set U and task set T, this paper needs to minimize the sum of the distances from workers to tasks, where $\boldsymbol{I}$ indicates whether a task is allocated to a worker. The first restrictive condition demonstrates that a task can only be done by one worker. The second restrictive condition shows that each worker can only do one task at most. The third restrictive condition indicates that each task is expected to be allocated as much as possible. Suppose the server knows the actual location of each worker. In that case, it knows the distance from each worker to each task, so we can directly use the existing linear programming tools to solve the problem.

$$\begin{aligned} &\min_{I} \sum_{u_i \in U} \sum_{t_j \in T} I\left(u_i, t_j\right) d\left(u_i, t_j\right) \\ &s.t.\ \ I\left(u_i, t_j\right) \in \{0, 1\} \\ &\quad \sum_{t_j \in T} I\left(u_i, t_j\right) \leq 1 \\ &\quad \sum_{u_i \in U} I\left(u_i, t_j\right) = 1. \end{aligned} \tag{1}$$

B. SYSTEM AND THREAT MODEL

Fig. 1 presents an overview of the system in this paper. The sequence number represents the steps to be executed. This system includes a worker set, a task set, and a server. The server receives the information from the workers and then allocates the tasks. In particular, the server first accepts the task set and starts to allocate the tasks. Then the server receives the candidate worker set and the related information of the workers. Then the server allocates the tasks. Finally, the server notifies the related workers to do their corresponding tasks.

In this paper, we assume a semi-trusted environment and entities. All entities will honestly execute the algorithm process, but will steal the privacy of workers during execution. The attackers of the system are workers and servers. In addition, third-party attackers can observe the geographic location information uploaded by workers to the server (by packet capture methods in computer network, etc.). It mains that it can obtain almost the same information as that obtained by the server.

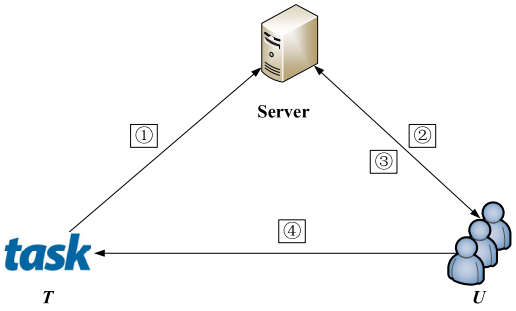

FIGURE 1. System overview.

C. GEO-INDISTINGUISHABILITY

The formal definition of Geo-Indistinguishability (Geo-I) is as follows:

Definition 3.1 (Geo-I): Suppose there is a random algorithm M, it's the domain of definition is $\boldsymbol{X}$ and range of value is $\boldsymbol{Z}$. If for any two position points $\boldsymbol{l_1}$ and $\boldsymbol{l_2}$, and any output $l^* \in Z$, the following inequality holds:

$$P\left[l^* \,|l_1\right] \leq e^{\varepsilon r} \cdot P\left[l^* \,|l_2\right], \tag{2}$$

then the algorithm M satisfies ε -Geo-I, where ε is the privacy protection parameter and $d\left(l_1, l_2\right) \leq r$.

The typical Geo-I is implemented by Planar Laplacian (PL), and the specific process is as follows:

The probability density function of the Planar Laplacian is as follows:

$$D\left(l_0\right)(l) = \frac{\varepsilon^2}{2\pi} \exp\left(-\varepsilon d\left(l_0, l\right)\right), \tag{3}$$

$d(l_0, l)$ represents the perturbation from the actual position $\boldsymbol{l_0}$ to the fuzzy position $\boldsymbol{l}$. For the convenience of representation, it is converted to the following polar coordinate form:

$$D\left(r, \theta\right) = \frac{\varepsilon^2}{2\pi} r \exp\left(-\varepsilon r\right), \tag{4}$$

where r and θ denote radius and angle, respectively. We obtain the boundary integral for r and θ respectively by calculation:

$$\begin{aligned} D_{\varepsilon,r}\left(r\right) &= \int_0^{2\pi} D\left(r, \theta\right) d\theta = \varepsilon^2 r e^{-\varepsilon r}, \\ D_{\varepsilon,\theta}\left(\theta\right) &= \int_0^{\infty} D\left(r, \theta\right) dr = \frac{1}{2\pi}. \end{aligned} \tag{5}$$

Then the generation method of radius r is as follows:

$$r = C_{\varepsilon}^{-1}(p) = -\frac{1}{\varepsilon}\left(W_{-1}\left(\frac{p-1}{e}\right)+1\right), \tag{6}$$

where W_{-1} represents Lambert W function. Then the angle is generated randomly within $[0,\ 2\pi)$. Finally, the noise adding method is as follows:

$$l_{0'} = l_0 + (r * \cos(\theta), r * \sin(\theta)). \tag{7}$$

D. LOCAL DIFFERENTIAL PRIVACY

The formal definition of Local Differential Privacy (LDP) is as follows:

Definition 3.2 (LDP): Suppose there is a random algorithm M whose domain is $Dom(M)$ and whose range is $Ran(M)$. For any two data records $t, t' \in Dom(M)$, and any output $t* \in Ran(M)$, if the following inequality holds:

$$P[M(t) = t^*] \le e^{\varepsilon} \cdot P[M(t') = t^*], \tag{8}$$

then algorithm $\boldsymbol{M}$ satisfies ε -LDP, where ε is the privacy protection parameter.

Random Response (RR) is a classic implementation mechanism of LDP. It answers the true value with probability $p = \frac{e^{\varepsilon}}{k-1+e^{\varepsilon}}$, and answers other values with probability $q = \frac{1}{k-1+e^{\varepsilon}}$, where k is the value range of the task. When everyone sends the result of privacy protection to the server, the server counts the number of the tag v as $\boldsymbol{I}$. Then the unbiased estimation of $\boldsymbol{I}$ is as follows:

$$I_v = \frac{I - kq}{p - q}. \tag{9}$$

E. COMBINATORIAL PROPERTIES

For some complex privacy protection problems, the RR or PL algorithm usually needs to be applied many times, and the privacy budget needs to be allocated reasonably. Specifically, we have the following theorems to guarantee these complex algorithms also satisfy differential privacy [11]:

Theorem 1 (Sequential Compositionality): Suppose that there are the random algorithms $M_1, M_2, \ldots, M_n$, whose corresponding privacy parameters are $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$. When these random algorithms are used on the same data set, their algorithms provide $\left(\sum_{i=1}^{n} \varepsilon_i\right)$ -differential privacy.

Theorem 2 (Parallel Compositionality): Suppose that the random algorithms $M_1, M_2, \ldots, M_n$ satisfy differential privacy, and their corresponding privacy parameters are $\varepsilon_1, \varepsilon_2, \ldots, \varepsilon_n$. When these random algorithms act on disjoint data sets $D_1, D_2, \ldots, D_n$, the composed algorithm $M(M_1, M_2, \ldots, M_n)$ provides $\max(\varepsilon_i)$-differential privacy for these sets.

Theorem 3 (Post-processing Property): Suppose that the random algorithm M_A satisfies differential privacy and random algorithm M_B working on the output of M_A, then M_B also satisfies differential privacy with the same level of privacy protection as M_A.

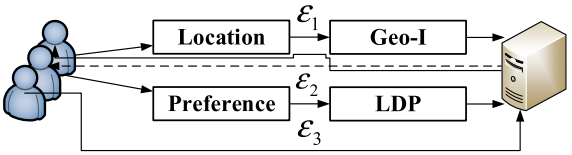

FIGURE 2. Overview of SLEPT.

F. PROBLEM DEFINITION

Suppose there are N tasks $T = \{t_1, t_2, \ldots\ldots, t_N\}$, and M candidate workers participate in the task allocation of privacy protection $U = \{u_1, u_2, \ldots\ldots, u_M\}$. Each task has its two-dimensional position coordinates (longitude and latitude). Each worker has its position coordinates and preference task set S_i. S_i is a collection of tasks that worker i want to do. In particular, the location of tasks is open to the public, and the workers' locations and the set of preference tasks need to be protected.

Moreover, we need to satisfy Geo-I for the location of workers and LDP for the preference set of workers. The goal of the server is to allocate all tasks on a minimal total travel distance. We assume that each worker can only do one task, and each task can only be selected by one worker.

IV. THE PROPOSED ALGORITHM

A. OVERVIEW OF SLEPT

Fig. 2 shows the overview of the SLEPT. As can be seen, SLEPT includes the following five stages:

Stage 1: Each worker uses the privacy budget ε_1 to call Planar Laplacian (PL) to perturb his location and submit the obfuscated location to the server;

Stage 2: Each worker uses private budget ε_2 to call Random Response (RR) to perturb the tasks in his own preference set;

Stage 3: The server conveys statistics over the distribution of tasks after perturbation and sends statistical information to each worker (i.e., the dotted line in the figure indicates the feedback information from the server);

Stage 4: Based on the statistical information, each worker perturbs his own preference set unevenly by using privacy budget ε_3, and then selects a task nearest to him until all tasks are allocated, or all workers are traversed;

Stage 5: The server informs the selected worker to go to the task location to do the task.

B. TWO-PHASE PREFERENCE COLLECTION

To collect preference information of workers and facilitate the task allocation, we design a Two-phase Preference Collection (TPC).

In the first phase of TPC, each worker uses privacy budget ε_2 to perturb the tasks in his preference set S through calling the RR mechanism. For each task in S, the privacy budget is $\frac{\varepsilon_2}{|S|}$. After he perturbed preference set task is sent to the server, the server invokes (9) to get the unbiased estimation after perturbation. In this paper, Q is used to represent the unbiased estimation array.

TABLE 2. An example of preference set.

u_1	t_1	t_3
u_2	t_2	t_4
u_3	t_2	t_3
u_4	t_1	t_4

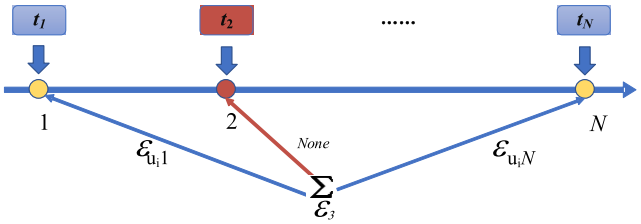

FIGURE 3. Overview of TPC for phase 2.

For example, suppose there are four workers $\{u_1, u_2, u_3, u_4\}$ and four tasks $\{t_1, t_2, t_3, t_4\}$, and the preference information of each worker is shown in Table 2. The privacy budget of each worker for each task in the preference set is $\frac{\varepsilon_2}{2}$. After calling the RR mechanism, the preference task of u_1 may become t_2 and t_3. In this way, each worker sends the perturbed preference set to the server to get the count information of each task. Such as $Q = \{2, 1, 3, 2\}$, it means that task t_1 appears in the preference set of two workers. Then the server calls (9) for unbiased estimation and gets $Q = \{2, 2, 2, 2\}$. And then, we normalize the array to get $F = \{0.25, 0.25, 0.25, 0.25\}$ (specific calculation method: $\frac{2}{2+2+2+2}$, where the numerator represents an element in the data Q, and the denominator represents the sum of all elements).

Fig. 3 is the schematic diagram of the second phase of TPC. In addition to uneven privacy budget allocation, this paper combines the task allocation sequential updating (TASU) mechanism (introduced in the next part) to further avoid the segmentation of privacy budget. In particular, if a task has been allocated to the previous worker u_{i-1}, then for the following worker u_i, if the allocated task is still in the preference set of u_i. We don't split the privacy budget for this task allocated. As shown in Fig.3, task t_2 (indicated in red) is not necessary to split the privacy budget.

In the second phase of TPC, we find that the fundamental reason why some tasks are not allocated is that they become preference tasks less frequently. In order to protect the preference, the perturbed preference tasks may not be in any worker's preference set, or a task may be in a worker's preference set who has been allocated to other tasks. As a result, the task can never be allocated. Based on this observation, this paper proposes that when the preference set of workers is perturbed, we can allocate a higher privacy budget should be allocated to the tasks with a lower frequency to avoid the above situation as far as possible. Therefore, this paper allocates the privacy budget according to the F array.

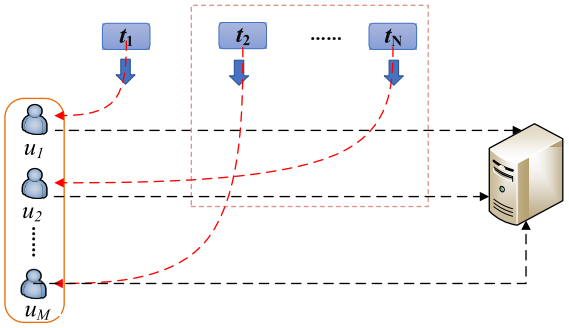

FIGURE 4. Overview of TASU.

Pseudocode 1 describes the two-phase preference collection (TPC) approach.

Pseudocode 1. The Procedure of Two-phase Preference Collection (TPC)

Input: Privacy budget ε_2 and ε_3; Worker set U; Preference set S;
Output: Normalized array F;
1: **for** u_i in U do
2: Worker i uses ε_2 to call RR to distribute the task sequence number in S;
3: Worker upload the perturbed task set to the server;
4: **end**
5: The unbiased estimation of each tag for each task is calculated according to (9);
6: Calculate the normalized array F;
7: **for** u_i in U do
8: Determine whether a task has been allocated to decide whether to split ε_3
9: **end**
10: **return** F;

C. TASK ALLOCATION UPDATING

To further reduce the segmentation of privacy budget when collecting preference sets, we develop the Task Allocation Sequential Updating method (TASU).

In particular, as shown in Fig. 4, this paper traverses workers one by one. Each worker first calls PL to perturb their location using the privacy budget ε_1. He needs to partition ε_3 using the F array, perturb their preference set, and then selects the task closest to their perturbed position. Particularly, suppose task t_1 has been allocated to u_1. In that case, it does not need to split the privacy budget for task t_1 even if it is in the preference set of u_2. The traversal process continues until all tasks are allocated, or all workers are traversed. Finally, suppose some tasks are not allocated, and some workers are not selected, then, they will be allocated to workers closest to them in this paper. Pseudocode 2 describes the task allocation sequential updating method (TASU).

Pseudocode 2. The Method of Task Allocation Sequential Updating (TASU)

Input: Privacy budgets ε_1 and ε_3; Worker set U; Task set T; Preference set S; Normalized array F;
Output: The result of task allocation R;
for u_i in U do
Worker i uses ε_1 to call PL to perturb his location;
The worker uploads the perturbed location to the server;
end
for u_i in U do
  for t_j in S_i do
    If t_j has been allocated, then no privacy budget is split for it;
    Workers choose the nearest task to do;
  end
end
The server marks the non-allocated task set as TT;
The server marks the unselected worker set as UU;
for t_j in TT do
  for u_i in UU do
    If t_j is in S_i, then t_j is allocated to u_i;
    Remove the worker who has been allocated from UU;
  end
end
return R;

D. PRIVACY ANALYSIS

Theorem 4: SLEPT satisfies ε-differential privacy.

Proof: In SLEPT, only **Stage 1**, **Stage 2** and **Stage 4** need to contact the original locations or preference information.

In **Stage 1**, workers' operations satisfy ε_1-Geo-I. In **Stage 2** and according to **Theorem 1**, worker's operations satisfy ε_2-LDP. In **Stage 4** and according to **Theorem 1**, worker's operations satisfy ε_3-LDP.

According to **Theorem 1**, each worker's operations satisfy ε-differential privacy, where $\varepsilon = \varepsilon_1 + \varepsilon_2 + \varepsilon_3$. According to**Theorem 3**, the server operations satisfy ε-differential privacy.

According to **Theorem 2**, the overall system satisfies differential privacy.

E. COMPLEXITY ANALYSIS

From the part of IV.A, we can see that the algorithm consists of five stages in total. In **Stage 1**, each worker uses the PL mechanism to add noise and consumes $O(M)$ totally, where M is the number of workers. In **Stage 2**, each worker perturbs his preference set and consumes $O(M|S|)$ totally, where $|S|$ represents the length of the preference set of workers. In **Stage 3**, the server transmission of the statistical information of each task consumes $O(N)$, where N is the number of tasks. In **Stage 4**, each worker perturbs his preference set partially. Specifically, the first worker perturbs $|S|$ tasks, and the second worker perturbs $|S|-1$ tasks, and so on until the N-th worker perturbs the last one task. The perturbation process consumes $O\left(\frac{|S|(1+|S|)}{2}\right)$ in total. Because in **Stage 5** is that the corresponding workers carry out their task, the server doesn't consume time. To sum up, the algorithm consumes$O\left((1+|S|)\left(\frac{|S|}{2}+M\right)+N\right)$. We can see that SLEPT algorithm is linearly related to the number of tasks through the time complexity.

V. EXPERIMENT

A. DATASET

We use two publicly available datasets collected from Foursquare to assign tasks: New York (NYC), and Tokyo (TKY). In particular, NYC contains 227428 check-in points and TKY has 573708 check-in points. In this paper, 300 check-in points are randomly selected as task locations and 500 check-in points as workers' positions for task allocation.

B. EXPERIMENTAL SETUP

We generally use Average Travel Distance (ATD), and Unassigned Number of Tasks (UNT) to evaluate the utility of the final noise task allocation results.

Equation (10) shows that it is ATD, and the experimental results are expressed in km.

$$ATD = \frac{\sum d(R)}{|R|}. \tag{10}$$

where $|R|$ represents the number of tasks successfully allocated, and $d(R)$ represents the travel distance of the corresponding task of a worker path in R.

As shown in (11), it is UNT, where N represents the number of tasks.

$$UNT = N - |R|. \tag{11}$$

C. BASELINES

According to the analysis of related work, we find that the existing schemes are inapplicable to solve the problem of this paper. To verify the effectiveness of the proposed scheme, we compare the SLEPT algorithm with the following design scheme.

1) NoPriv: to verify the utility loss of privacy protection, we give this comparative method. The server uses the existing solving tools, combines with the actual information of workers to solve the (1), and gets the final task allocation result directly;
2) LPA (Linear Programming Approach) [38]: In this method, each worker first uses half of his privacy budget to perturb his preference information and send it to the server. The server can get the task allocation result by calling the algorithm in [38]. In particular, according to the original author's suggestion, the size of each grid is set as 1km * 1km;
3) PBA (Probability-based Approach) [41]: In this method, each worker calls the method in [40] when calculating the distance from the perturbation position to

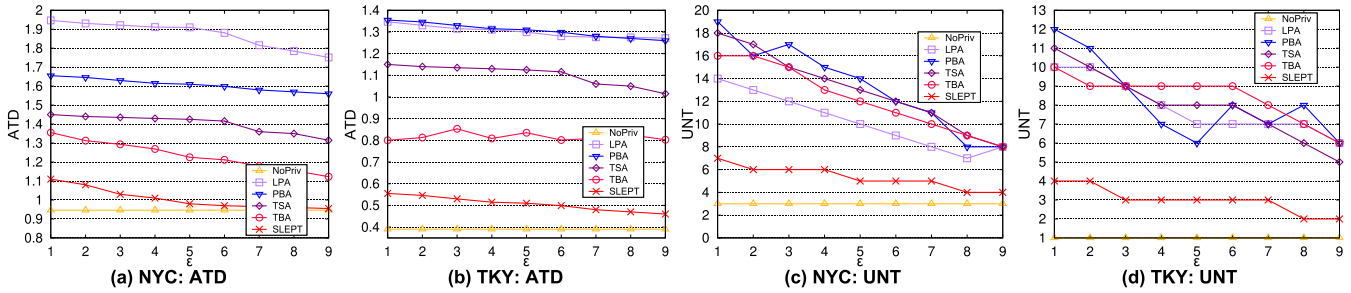

FIGURE 5. Performance comparison under different privacy parameter ε: (a) for the changes of ATD with the NYC, (b) for the changes of ATD with the TK, (c) for the changes of UNT with the NYC, (d) for the changes of UNT with the TKY.

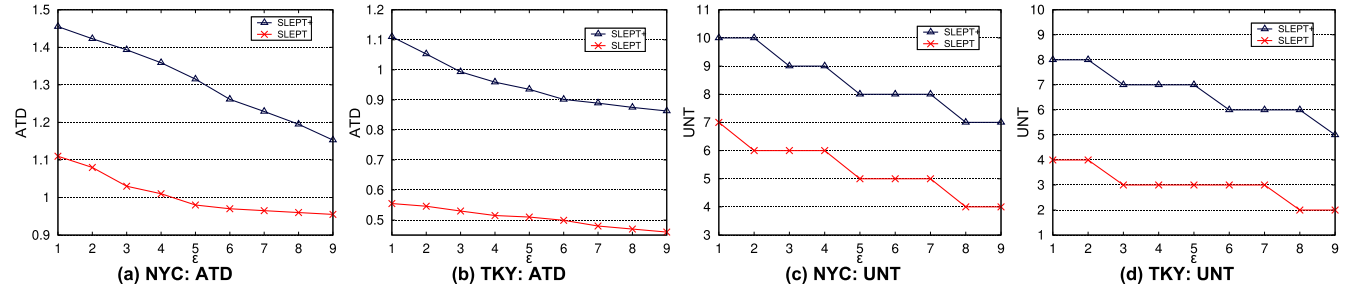

FIGURE 6. Effect of TPC: (a) for the changes of ATD with the NYC, (b) for the changes of ATD with the TKY, (c) for the changes of UNT with the NYC, (d) for the changes of UNT with the TKY.

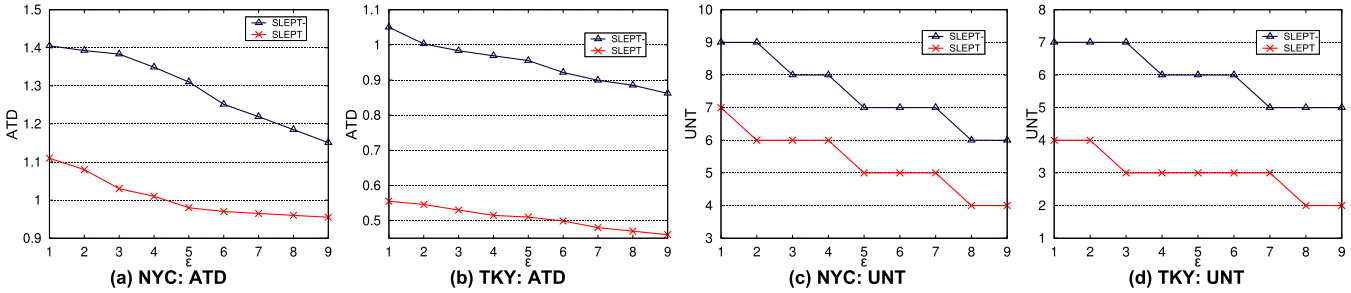

FIGURE 7. Effect of TASU: (a) for the changes of ATD with the NYC, (b) for the changes of ATD with the TKY, (c) for the changes of UNT with the NYC, (d) for the changes of UNT with the TKY.

the task position. When the probability of the distance value obtained more than the distance threshold (such as the distance from other tasks) is > 0.5, other tasks are allocated to the worker;

4) TSA (Two-Server Approach) [37]: In this method, each worker first uses half of his privacy budget to perturb his preference information and send it to the server, and then the server calls the algorithm in [36] to get the task allocation result;
5) TBA (Tree-based Approach) [43]: In this method, each worker first uses half of his privacy budget to perturb the preference information and sends it to the server. And then, the server calls the algorithm in [42] to get the task allocation result.

D. PERFORMANCE COMPARISON

1) THE IMPACT OF ε

To test the effect of privacy protection ε on algorithm utility, we set $\varepsilon \in \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$ to evaluate the performance of SLEPT under different privacy parameters. The experimental results are shown in Fig. 5. Fig. 5 (a) and Fig. 5 (b) separately show the changes of ATD (Average Travel Distance) corresponding to the NYC data set and the TKY data set. Fig. 5 (c) and Fig. 5 (d) separately show the changes of UNT (Unassigned Number of Tasks) corresponding to the NYC data set and the TKY data set.

As can be seen from Fig. 5, the utility of all algorithms become better with the increase of ε,. That is, ATD is smaller and UNT is also smaller. This is because with the increase of ε, the total noise in all algorithm examples becomes to be reduced. In addition, the SLEPT algorithm performs best. That's because the two-phase preference collection (TPC) algorithm and the task allocation sequential updating method (TASU) in this paper can significantly reduce the noise in the algorithm. As for LPA, PBA, TSA and TBA algorithms, the improper collection of preference tasks leads to many tasks being allocated to non-optimal workers. So their ATD values are large. At the same time, incorrect

collection of preference tasks will lead to a mismatch between the preference set and the worker's preference. In such a case, even if the server assigns them to such workers, they will refuse to perform tasks in the final task execution stage.

2) THE EFFECTIVENESS OF TPC

To verify the effectiveness of the proposed TPC module, we set this part of experiments. The experimental results are shown in Fig. 6. The horizontal ordinate represents the total privacy budget, and the vertical ordinate represents the ATD or UNT. In addition, SLEPT+ is used to describe the method when not executing the TPC module. That is to say, it uses one-half of the privacy budget to perturb the locations of workers. It uses the other half to collect the preference information of each worker and then traverses each worker for task allocation.

3) THE EFFECTIVENESS OF TASU

To verify the effectiveness of the TASU module, we set these comparative experiments. The experimental results are shown in Fig. 7. The horizontal ordinate represents the total privacy budget, and the vertical ordinate represents the ATD or UNT. SLEPT- is used to represent the method when not executing the TASU module. That is, it uses ε_1 to perturb the locations of each worker and adopts TPC to collect the preference information. Then, the server performs the greedy task allocation based on the locations and preference information.

As shown in Fig. 7, the SLEPT algorithm is significantly better than SLEPT-. By sequential updating, we can allocate the task to the corresponding worker who is really preferred it. At the same time, greedy allocation will cause tasks not to be allocated to the proper workers. In such cases, on the one hand, if tasks are not allocated to the optimal workers, ATD will increase. On the other hand, if there are no workers who prefers to do the leaved tasks, UNT will increase.

VI. CONCLUSION

Focusing on the task allocation problem under preference protection, we propose a task allocation algorithm SLEPT while satisfying differential privacy. We show it satisfies ε-differential privacy. In particular, to improve the success rate of task allocation as much as possible, we design a two-phase preference collection mechanism TPC. To reduce the travel distance of workers and improve the success rate of task allocation, we develop a task allocation sequential updating mechanism TASU. Experimental results on two public datasets verify the effectiveness of SLEPT. In addition, the idea of SLEPT can be used for other applications in the context of crowdsensing scenarios with privacy protection.

REFERENCES

[1] X. Li, J. Li, Y. Liu, Z. Ding, and A. Nallanathan, "Residual transceiver hardware impairments on cooperative NOMA networks," *IEEE Trans. Wireless Commun.*, vol. 19, no. 1, pp. 680–695, Jan. 2020.

[2] X. Li, Q. Wang, H. Peng, H. Zhang, D.-T. Do, K. M. Rabie, R. Kharel, and C. Charles, "A unified framework for HS-UAV NOMA networks: Performance analysis and location optimization," *IEEE Access*, vol. 8, pp. 13329–13340, 2020.

[3] X. Li, M. Zhao, M. Zeng, S. Mumtaz, V. G. Menon, Z. Ding, and O. A. Dobre, "Hardware impaired ambient backscatter NOMA systems: Reliability and security," *IEEE Trans. Commun.*, vol. 69, no. 4, pp. 2723–2736, Apr. 2021.

[4] Q. Hu, S. Wang, X. Cheng, J. Zhang, and W. Lv, "Cost-efficient mobile crowdsensing with spatial-temporal awareness," *IEEE Trans. Mobile Comput.*, vol. 20, no. 3, pp. 928–938, Mar. 2021.

[5] F. Yucel and E. Bulut, "User satisfaction aware maximum utility task assignment in mobile crowdsensing," *Comput. Netw.*, vol. 172, May 2020, Art. no. 107156.

[6] J. Duguépéroux and T. Allard, "From task tuning to task assignment in privacy-preserving crowdsourcing platforms," in *Transactions on Large-Scale Data-and Knowledge-Centered Systems XLIV* (Lecture Notes in Computer Science), vol. 12380. Berlin, Germany: Springer, 2020, pp. 67–107.

[7] M. Dai, J. Li, Z. Su, W. Chen, Q. Xu, and S. Fu, "A privacy preservation based scheme for task assignment in Internet of Things," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2323–2335, Oct. 2020.

[8] E. Aloufi, R. Alharthi, I. Alrashdi, A. Alqazzaz, D. Alsulami, and M. Zohdy, "TASC: Efficient task assignment in spatial crowdsourcing with workers privacy protection," in *Proc. EIT*, Chicago, IL, USA, Jul. 2020, pp. 546–550.

[9] Y.-H. Lin, "Nearly cloaking for the elasticity system with residual stress," *Asymptotic Anal.*, vol. 106, no. 1, pp. 1–23, 2018.

[10] Y. Gong, C. Zhang, Y. Fang, and J. Sun, "Protecting location privacy for task allocation in ad hoc mobile cloud computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 110–121, Jan./Mar. 2015.

[11] C. Dwork, "Differential privacy," in *Proc. ICALP*, vol. 2, 2006, pp. 1–12.

[12] H. Husain, B. Borja, C. Zac, and R. Nock, "Local differential privacy for sampling," in *Proc. AISTATS*, 2020, pp. 3404–3413.

[13] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM SIGSAC-CCS*, Berlin, Germany, 2013, pp. 901–914.

[14] Ú. Erlingsson, V. Pihur, and A. Korolova, "RAPPOR: Randomized aggregatable privacy-preserving ordinal response," in *Proc. ACM SIGSAC-CCS*, Scottsdale, AZ, USA, Nov. 2014, pp. 1054–1067.

[15] J. Tang, A. Korolova, X. Bai, X. Wang, and X. Wang, "Privacy loss in Apple's implementation of differential privacy on MacOS 10.12," 2017, *arXiv:1709.02753*.

[16] D. Bolin, J. Kulkarni, and S. Yekhanin, "Collecting telemetry data privately," in *Proc. Adv. Neural. Inf. Process Syst.*, 2017, pp. 3571–3580.

[17] K. Fawaz and K. G. Shin, "Location privacy protection for smartphone users," in *Proc. ACM SIGSAC-CCS*, Scottsdale, AZ, USA, Nov. 2014, pp. 239–250.

[18] K. Fawaz, H. Feng, and K. G. Shin, "Anatomization and protection of mobile apps' location privacy threats," in *Proc. USENIX Secur. Symp.*, 2015, pp. 753–768.

[19] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proc. ACM SIGSAC-CCS*, Scottsdale, AZ, USA, Nov. 2014, pp. 251–262.

[20] L. Yu, L. Liu, and C. Pu, "Dynamic differential location privacy with personalized error bounds," in *Proc. NDSS*, San Diego, CA, USA, 2017, pp. 1–15.

[21] S. Oya, C. Troncoso, and F. Pérez-González, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *Proc. ACM SIGSAC-CCS*, Dallas, TX, USA, Oct. 2017, pp. 1959–1972.

[22] A. Pyrgelis, C. Troncoso, and E. De Cristofaro, "What does the crowd say about you? Evaluating aggregation-based location privacy," *Proc. Privacy Enhancing Technol.*, vol. 2017, no. 4, pp. 156–176, Oct. 2017.

[23] K. Chatzikokolakis, E. ElSalamouny, and C. Palamidessi, "Efficient utility improvement for location privacy," *Proc. Privacy Enhancing Technol.*, vol. 2017, no. 4, pp. 308–328, Oct. 2017.

[24] E. ElSalamouny and S. Gambs, "Optimal noise functions for location privacy on continuous regions," *Int. J. Inf. Secur.*, vol. 17, no. 6, pp. 613–630, Nov. 2018.

[25] J. Wang, Y. Wang, G. Zhao, and Z. Zhao, "Location protection method for mobile crowd sensing based on local differential privacy preference," *Peer-to-Peer Netw. Appl.*, vol. 12, no. 5, pp. 1097–1109, Sep. 2019.

[26] S. Takagi, Y. Cao, Y. Asano, and M. Yoshikawa, "Geo-graph-indistinguishability: Protecting location privacy for LBS over road networks," in *Proc. DBSec*, Charleston, SC, USA, 2019, pp. 143–163.
[27] S. Oya, C. Troncoso, and F. Pérez-González, "Is geo-indistinguishability what you are looking for?" in *Proc. WPES*, Dallas, TX, USA, Oct. 2017, pp. 137–140.
[28] J. C. Duchi, M. I. Jordan, and M. J. Wainwright, "Local privacy and statistical minimax rates," in *Proc. IEEE 54th ASFCS*. Berkeley, CA, USA: Microsoft New England Research, Oct. 2013, pp. 429–438.
[29] G. Fanti, V. Pihur, and Ú. Erlingsson, "Building a RAPPOR with the unknown: Privacy-preserving learning of associations and data dictionaries," *Proc. Privacy Enhancing Technol.*, vol. 2016, no. 3, pp. 41–61, 2016.
[30] P. Kairouz, S. Oh, and P. Viswanath, "Extremal mechanisms for local differential privacy," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 2879–2887, 2016.
[31] P. Kairouz, K. Bonawitz, and D. Ramage, "Discrete distribution estimation under local privacy," in *Proc. ICML*, New York, NY, USA, 2016, pp. 2436–2444.
[32] R. Bassily and A. Smith, "Local, private, efficient protocols for succinct histograms," in *Proc. ACM(STOC)*, Portland, OR, USA, Jun. 2015, pp. 127–135.
[33] T. T. Nguyên, X. Xiao, Y. Yang, S. Cheung Hui, H. Shin, and J. Shin, "Collecting and analyzing data from smart device users with local differential privacy," 2016, *arXiv:1606.05053*.
[34] Z. Qin, Y. Yang, T. Yu, I. Khalil, X. Xiao, and K. Ren, "Heavy hitter estimation over set-valued data with local differential privacy," in *Proc. ACM SIGSAC-CCS*, Oct. 2016, pp. 192–203.
[35] N. Wang, X. Xiao, Y. Yang, J. Zhao, S. C. Hui, H. Shin, J. Shin, and G. Yu, "Collecting and analyzing multidimensional data with local differential privacy," in *Proc. IEEE 35th ICDE*, Macau, China, Apr. 2019, pp. 638–649.
[36] H. To, G. Ghinita, L. Fan, and C. Shahabi, "Differentially private location protection for worker datasets in spatial crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 934–949, Apr. 2017.
[37] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location privacy-preserving task allocation for mobile crowdsensing with differential geo-obfuscation," in *Proc. Int. Conf. WWW*, Perth, WA, Australia, Apr. 2017, pp. 627–636.
[38] Z. Wang, J. Hu, R. Lv, J. Wei, Q. Wang, D. Yang, and H. Qi, "Personalized privacy-preserving task allocation for mobile crowdsensing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1330–1341, Jun. 2019.
[39] L. Wang, G. Qin, D. Yang, X. Han, and X. Ma, "Geographic differential privacy for mobile crowd coverage maximization," in *Proc. AAAI(CAI)*, New Orleans, LA, USA, 2018, pp. 200–207.
[40] H. To, C. Shahabi, and L. Xiong, "Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server," in *Proc. ICDE*, Paris, France, Apr. 2018, pp. 833–844.
[41] W. Gong, B. Zhang, and C. Li, "Privacy-aware online task assignment framework for mobile crowdsensing," in *Proc. IEEE ICC*, Shanghai, China, May 2019, pp. 1–6.
[42] Q. Tao, Y. Tong, Z. Zhou, Y. Shi, L. Chen, and K. Xu, "Differentially private online task assignment in spatial crowdsourcing: A tree-based approach," in *Proc. IEEE 36th ICDE*, Dallas, TX, USA, Apr. 2020, pp. 517–528.
[43] T. Song, F. Zhu, and K. Xu, "Specialty-aware task assignment in spatial crowdsourcing," in *Proc. Int. Conf. AISC*. Cham, Switzerland: Springer, 2018, pp. 243–254.
[44] L. Béziaud, T. Allard, and D. Gross-Amblard, "Lightweight privacy-preserving task assignment in skill-aware crowdsourcing," in *Proc. Int. Conf. DESA*. Cham, Switzerland: Springer, 2017, pp. 18–26.

JUNTAO HAN was born in 1984. He received the M.Sc. degree in communication and information system from Capital Normal University, Beijing, China, in 2011, and the Ph.D. degree from the Beijing University of Posts and Telecommunications, in 2014. Since 2017, he has been working with the School of E-Commerce and Logistics Management, Henan University of Economics and Law. His current research interests include services computing, complex networks, e-commence, and supply chain management.

SHUYUE CAI is currently pursuing the master's degree with the School of Management Engineering, Zhengzhou University. Her research interests include logistics supply chain management and shop scheduling optimization.