Received January 20, 2021, accepted February 3, 2021, date of publication February 10, 2021, date of current version February 18, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3058574

Local Privacy-Preserving Dynamic Worker Locations in Spatial Crowdsourcing

FENG LIN[1,2], JIANHAO WEI[3], (Student Member, IEEE), JUNYI LI[3], (Member, IEEE), JIANMING ZHANG[4], (Member, IEEE), AND BO YIN[4], (Member, IEEE)

[1]Provincial Key Laboratory of Informational Service for Rural Area of Southwestern Hunan, Shaoyang University, Shaoyang 422000, China
[2]College of Information Engineering, Shaoyang University, Shaoyang 422000, China
[3]College of Computer Science and Electronic Engineering, Hunan University, Changsha 410082, China
[4]School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha 410114, China

Corresponding author: Jianhao Wei (jianhao@hnu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61872131, Grant 61972056, and Grant 61902040; and in part by the Scientific Research of Fund of Hunan Provincial Education Department under Grant 20B519, Grant 19B512, and Grant 20B015.

ABSTRACT An outsourcing service named spatial crowdsourcing (SC) becomes popular, whereby the SC-server allocates nearby tasks to the workers based on the outsourced task and worker locations. Exposing real locations can cause serious privacy leakage. However, traditional differential privacy (DP) and encryption methods do not consider the dynamic worker location and correlation privacy. Here, a Local DP-based dynamic worker location protection (LDPDW) scheme is proposed to achieve high-quality task allocation and locally protect the correlation and location privacy of dynamic workers. Specifically, LDPDW generates noisy high correlated graph classes and obfuscates the worker locations in a static case by adopting a LDP-based correlation graph (LDPCG) algorithm and distance score-based LDP (DSLDP) algorithm, thereby achieving controlled noise addition and ensuring the correlation and location privacy. To support the privacy-preserving dynamic locations, a dynamic correlation graph-based location obfuscation (DCGLO) algorithm is proposed to allocate reasonable privacy budget ϵ, which ensures the data utility. Finally, a linear acceptance model-based task allocation (LAMTA) algorithm is used to allocate tasks to the workers with high acceptance rates. Privacy analysis and the extensive experimental results show that our LDPDW scheme follows ϵ-LDP while allocating tasks with high data utility.

INDEX TERMS Task allocation, location privacy, local differential privacy, data utility.

I. INTRODUCTION

A new *Spatial Crowdsourcing* (SC) platform has become popular promoted by the rapid development of mobile devices, whereby the *SC-server* allocates the tasks outsourced by the *requesters* to appropriate *workers* based on the task and worker locations [1]–[3]. Since task allocation involves large-scale data storage and matching calculations, it is performed by SC-servers. SC services have been applied to many fields, such as urban planning [4], environmental testing [5] and traffic monitoring [6]. Some popular SC platforms contain DiDi [7], Waze [8], OpenStreetMap [9] and Uber [10], etc. In particular, the ridership worldwide of Uber reached 1.7 billion trips in the second quarter of 2019, and its net revenue reached 14.1 billion dollars in 2019 [11].

To obtain the tasks, the workers require outsourcing their locations to the SC-server. However, publishing real locations may cause the serious privacy leakage due to sensitive workers information including health status [12], [13], daily activities [14] and personal track [15], [16], etc. The untrusted SC-server may leak the worker locations to the attackers once it is hacked. The workers can be monitored or stolen indentities when the attackers know their locations [15], [17]. For example, as a popular SC service platform, the workers in Waze can be monitored by attackers based on exposed locations [18]. Once the workers' locations are leaked, they will not participate in the SC services. Therefore, protecting the workers' location privacy becomes very important for improving the quality of SC services.

To achieve the privacy-preserving worker locations in SC services, traditional privacy-preserving methods such as homomorphic encryption methods in [14], [20]–[22]

The associate editor coordinating the review of this manuscript and approving it for publication was Yassine Maleh.

and anonymity methods in [23]–[25] can ensure that the SC-server allocates tasks to workers without knowing the worker locations. However, the homomorphic encryption method does not provide high efficiency [26], and the works in [27], [28] have showed that the anonymity method may be vulnerable caused by the strong background knowledge attacks and the reference attacks. Recently, a popular *Differential Privacy* (DP) method [29], [30] has been used to protect the location privacy in SC services [27], [31], [32]. For example, To *et al* [33], [34] protected the worker location privacy in SC by using a DP-based private spatial decomposition (PSD) method. Gong *et al.* [35] protected the location and reputation privacy of workers by using a reputation-based DP method and relying on a cellular service provider (CSP). In [36], Wang *et al.* proposed an agent-based differential privacy (DADP) framework to publish the crowdsourced data with strong privacy guarantees. However, above DP methods protect the worker location privacy by adopting a trusted thord party (TTP). The TTP is vulnerable by hacking since it stores all workers' original location information. The location privacy of users may be leaked once the TTP is attacked. To overcome this shortcoming, a novel **Local Differential Privacy** (LDP) method allows the users to locally obfuscate their own data [37]–[39], [41]. Wang *et al.* [40] introduced a distortion geo-obfuscation with LDP method to locally protect the task allocation privacy in SC. The work in [42] adopted the geo-indistinguishability method to ensure the location privacy in vehicle-based spatial crowdsourcing. To *et al.* [43] proposed a LDP framework to protect the location privacy of workers in SC based on geo-indistinguishability. However, these LDP methods only protect the worker location privacy at one time point without considering protecting the location privacy of dynamic workers. In practice, the SC services require the online workers to perform the tasks so that the workers' locations change dynamically [43], [44]. Exposing dynamic worker locations can increase the privacy disclosure risk [34], [43]. Besides, the future work in [43] also considers protecting the location correlation privacy of workers since public worker trajectories following specific movement patterns may leak the location privacy [45]. Therefore, achieving the local location and correlation privacy protection of dynamic workers in SC becomes very important.

In this paper, we explore using LDP to solve the problems of *both local location privacy and correlation privacy of dynamic workers in SC*, in which the SC-server performs high-quality task matching from the noisy dynamic locations outsourced by the workers themselves. To achieve this goal, we need to face the following three challenges: (i) since the workers' locations are correlated, which can leak the location privacy [43], [45]. Thus how to protect the correlation privacy between dynamic locations is challenging. (ii) To protect the dynamic location privacy, the workers' dynamic locations need to be locally obfuscated. However, traditional LDP methods have low accuracy for the dynamic location publishing since the amount of noise added at each location increases as the released dataset increases [37], [43]. Therefore, how to use the LDP to obfuscate the dynamic worker locations locally while ensuring the data utility is a challenge. (iii) Most importantly, since the workers' dynamic locations and correlations are obfuscated by LDP, this will reduce the quality of task allocation. It is a challenge to achieve the balance between dynamic worker privacy and task allocation having high data utility.

Here, a LDP-based dynamic worker location protection (LDPDW) scheme is proposed to solve the above challenges. For the first challenge, a LDP-based correlation graph (LDPCG) algorithm is proposed to cluster the high correlated locations into different graph classes and add laplace noise to the high correlated classes for protecting the correlation privacy and ensuring the correlation utility. Towards the second challenge, we first propose a distance score-based LDP (DSLDP) algorithm to obfuscate the worker locations in a static case by designing a smaller distance score sensitivity, thereby adding controlled noise. Further, to support the privacy-preserving dynamic locations, we allocate a fraction of privacy budget ϵ to generate noisy correlated graph structure and use remaining privacy budget to obfuscate dynamic worker locations by adopting a dynamic correlation graph-based location obfuscation (DCGLO) algorithm. For the third challenge, our LDPDW scheme requires achieving ϵ-LDP to ensure the dynamic worker location and correlation privacy. A linear acceptance model-based task allocation (LAMTA) algorithm is proposed to allocate the workers with high acceptance rates. Besides, our LDPDW scheme also has high data utility by extensive experiment evaluation.

We summarize the contributions as follows:

- We propose a LDPDW scheme to achieve high-quality task allocation and perform the locally privacy-preserving locations and correlations of dynamic workers in SC.
- To protect the correlation and location privacy, we propose a LDPCG algorithm and a DSLDP algorithm to generate noisy high correlated graph classes and obfuscate the worker location at each time point, respectively, which ensures controlled noise addition. Further, a LAMTA algorithm uses a linear acceptance model to perform high-quality task allocation.
- To support the locally dynamic worker location protection, a DCGLO algorithm dynamically obfuscates the worker locations and correlations by allocating privacy budget reasonably, which considers the data utility.
- Strict privacy analysis and extensive experiments confirm that our LDPDW scheme follows ϵ-LDP and can allocate the tasks with high data utility.

We organize this paper's rest contents as follows. We describe the problem formulation in Section II. The preliminaries is introduced in Section III. Section IV details the proposed algorithms in our LDPDW scheme and Section V analyzes the proposed algorithms' security and time complexity. We evaluate our scheme's performance in Section VI.

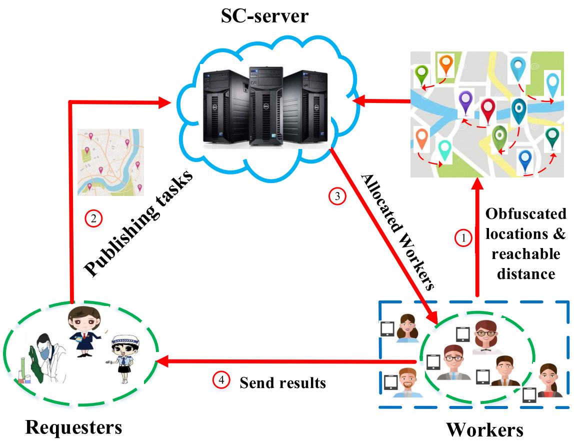


FIGURE 1. System model.

Finally, the related works are introduced in Section VII and the conclusions are summarized in Section VIII.

II. PROBLEM FORMULATION

Here, we introduce the system model, the threat model and the design goals.

A. SYSTEM MODEL

In Fig. 1, our system model consists of three entities, such as *Requesters*, *Workers* and *SC-server*.

- *Workers*: Here, the workers perform tasks voluntarily without any rewards [32], [34], [35]. To obtain the tasks, the workers with mobile devices need to outsource their locations and reachable distance to the SC-server. Due to sensitive privacy, the locations are first obfuscated locally by the workers before outsourcing them. When some workers are allocated by the SC-server, these allocated workers start to perform the tasks, and return the results to the corresponding requesters.
- *Requesters*: When a requester wants to obtain the analysis results for a task, he/she publishes the task locations to the SC-server. After receiving the performed results from the allocated workers, the requester will perform mining analysis.
- *SC-server*: The SC-server can be considered as a cloud server who has huge storage and computing capabilities [32], [43]. Upon receiving the task locations and obfuscated worker locations, the SC-server searches all obfuscated worker locations, allocates the workers to perform the task based on the reachable distance submitted by the workers, and feedbacks the allocated workers to the corresponding workers.

In the following, our system model's workflow is introduced. Given n workers, for a worker W_i ($i \in [1, n]$), he/she has a location set $L_{W_i} = \{L_{W_{it_1}}, L_{W_{it_2}}, \cdots, L_{W_{it_m}}\}$ at m timestamps and a reachable distance R_{W_i}, where $L_{W_{it_j}}$ is the worker W_i's location at j-th ($j \in [1, m]$) timestamp, and the R_{W_i} represents a circular region centered at $L_{W_{it_j}}$. Each worker W_i first adopts a **LDP-based correlation graph** (LDPCG) algorithm to obfuscate the high correlations between $L_{W_{it_j}}$ and $L_{W_{it_k}}$ ($j, k \in [1, m]$). After that, we obtain the location set $L^c_{W_i} = \{L^c_{W_{it_1}}, L^c_{W_{it_2}}, \cdots, L^c_{W_{it_m}}\}$ with noisy correlations. Then W_i uses a novel **distance score-based LDP** (DSLDP) algorithm to obfuscate the location set $L^c_{W_i}$ and outsources the obfuscated worker information $(\widetilde{L}_{W_i}, R_{W_i}) = (\{\widetilde{L}_{W_{it_1}}, \cdots, \widetilde{L}_{W_{it_m}}\}, R_{W_i})$ to the SC-server (Step 1). To support the privacy-preserving dynamic worker location publication, a **dynamic correlation graph-based location obfuscation** (DCGLO) algorithm is proposed to publish dynamic obfuscated worker locations. When a requester wants to publish the task T, he/she publishes the task location L_T to the SC-server (Step 2). Upon receiving the task location L_T and obfuscated worker information $(\widetilde{L}_{W_i}, R_{W_i})$, the SC-server uses a **linear acceptable model** (LCM) to allocate suitable workers $\{W_{z_1}, W_{z_2}, \cdots, W_{z_k}\}$ close to the task location L_T by searching all noisy worker information $(\widetilde{L}_{W_i}, R_{W_i})$ ($i \in [1, n]$), and returns the allocated workers and the task location $(\{W_{z_1}, W_{z_2}, \cdots, W_{z_k}\}, L_T)$ to the corresponding workers (Step 3). Finally, the allocated worker W_{z_r} ($r \in [1, k]$) decides whether to perform the task T according to the actual distance between the actual worker location L_{W_i} and the task location L_T, and returns the performed results to the requester (Step 4).

B. ATTACK MODEL

Here, we aim at the dynamic worker location and correlation privacy in SC. The workers' identifies are protected by the anonymity method. Besides, the TLS-like scheme in [46] can ensure the communication security.

In our attack model, following the works in [14], [20]–[22], [34], [43], we take the SC-server as an "*honest-but-curious*" entity who performs task allocation agreement but also leaks the workers' real locations to the attackers when it compromises caused by hacking. Since the workers and the requesters have their own data, they are trusted. To protect the worker location and correlation privacy locally, the workers use LDP method to obfuscate the high correlations and dynamic locations, respectively.

C. DESIGN GOALS

In this paper, the worker locations will be protected. However, protecting the worker privacy can complicate task allocation and increase system overhead since the SC-server requires allocating more workers to performs tasks. Besides, the obfuscated locations will lead to incorrect allocation results. Therefore, we design the following goals.

- ***Location and correlation Privacy***. The worker location privacy and correlation privacy cannot be leaked when the SC-server performs task allocation from the obfuscated locations of the workers.
- ***Data Utility***. The system model should have high data utility, that is the tasks accepted by workers account for a high ratio of all tasks.

TABLE 1. Main symbols.

Notation	Description	Notation	Description
$L_{W_i}, L^*_{W_i}$	Neighboring datasets	l_{loi}, l_{lai}	longitude, latitude
n	Workers' number	m	Timestamps' number
ϵ_1, ϵ_2	Privacy budgets	Ψ	Query function
$\widetilde{\Psi}$	Privacy mechanism	R_{W_i}	Reachable distance
$\Delta\Psi$	Query sensitivity	$L_{W_{it_k}}$	Location at time t_k
$S(\bullet)$	Location correlation	$D(\bullet)$	Euclidean distance
σ_1, σ_2	Thresholds	AC_T	Task acceptance

- ***System Overhead***. Since the allocated workers' number can affect the task allocation complexity and communication overhead, the efficiency of system model should not be significantly decreased.

III. PRELIMINARIES

In this section, the definitions of both worker locations and local differential privacy (LDP) in [37]–[39] are introduced, respectively. Table 1 gives the main notations.

Worker Locations: To represent the locations of dynamic workers, we first define the worker W_i's location $L_{W_{it_j}}$ at timestamp t_j is $L_{W_{it_j}} = (l^{tj}_{loi}, l^{tj}_{lai})$, where l^{tj}_{loi} and l^{tj}_{lai} are the longitude and latitude of the location $L_{W_{it_j}}$, respectively.

*Definition 1: (**Dynamic Worker Locations**). Given worker W_i's location $L_{W_{it_j}} = (l^{tj}_{loi}, l^{tj}_{lai})$ at timestamp t_j, for m timestamps, the original dynamic worker location dataset L_{W_i} can be denoted by*:

$$\begin{aligned} L_{W_i} &= (L_{W_{it_1}}, L_{W_{it_2}}, \cdots, L_{W_{it_m}}) \\ &= \{(l^{tj}_{loi}, l^{tj}_{lai}) | j = 1, 2, \cdots, m\} \end{aligned}$$

Local Differential Privacy (LDP): Here, we first give the concept of LDP. Two datasets L_{W_i} and $L^*_{W_i}$ are neighboring datasets iff they satisfy $L_{W_i} = L^*_{W_i} \cup L_{W_{it_j}}$ or $L^*_{W_i} = L_{W_i} \cup L_{W_{it_j}}$. A privacy mechanism $\widetilde{\Psi}$ gets ϵ-LDP when a Laplace noise $Lap(\frac{\Delta\Psi}{\epsilon})$ ia added to the query function Ψ, where ϵ is the privacy budget and $\Delta\Psi$ is the query sensitivity.

*Definition 2: (**Local Differential Privacy**) [37]. For two neighboring datasets W_i and W^*_i, given a privacy budget ϵ, a privacy mechanism $\widetilde{\Psi}(W_i) \in Range(\Upsilon)$. $\widetilde{\Psi}$ will achieve ϵ-LDP when it has*:

$$Pr[\widetilde{\Psi}(W_i) \in \Upsilon] \leq e^{\epsilon} \times Pr[\widetilde{\Psi}(W^*_i) \in \Upsilon]$$

*Definition 3: (**Query Sensitivity**) [29]. For the neighboring datasets W_i and W^*_i, a query function $\Psi(W_i) \rightarrow \Re^h_i$, thus we calculate the query sensitivity $\Delta\Psi$ as*:

$$\Delta\Psi = max_{W_i, W^*_i} \|\Psi(W_i) - \Psi(W^*_i)\|_1$$

*Definition 4: (**Laplace Mechanism**) [29]. If $\Psi(W_i) = (y_1, \cdots, y_h)$, $\widetilde{\Psi}$ will achieve ϵ-LDP when $\widetilde{\Psi}(W_i)$ satisfies*:

$$\widetilde{\Psi}(W_i) = \Psi(W_i) + Lap(\frac{\Delta\Psi}{\epsilon})$$

IV. LDP-BASED DYNAMIC WORKER LOCATION PROTECTION SCHEME

In this section, we propose a LDP-based dynamic worker location protection (LDPDW) scheme to achieve the locally privacy-preserving dynamic worker locations and correlations while allocating high-quality tasks. The proposed LDPDW scheme contains four modules, such as ***LocCorPro***, ***LocObfuca***, ***DynaLocPro*** and ***TaskAlloc***.

LocCorPro Module: Given the original dataset $L(W_i) = \{L_{W_{it_1}}, L_{W_{it_2}}, \cdots, L_{W_{it_m}}\}$ at m timestamps, the worker W_i first uses a LDP-based correlation graph (LDPCG) algorithm to calculate the correlation score S_{jk} between the locations $L_{W_{it_j}}$ and $L_{W_{it_k}}$ by using a complete undirect graph, and cluster the high-correlated locations into graph classes. Finally, each graph class is added Laplace noise based on LDP method so that the correlations are obfuscated.

LocObfuca Module: In this module, each worker adopts a distance score-based LDP (DSLDP) to obfuscate the location set $L^c_{W_i}$ and obtains the noisy locations $(\widetilde{L}(W_i), R_{W_i}) = \{(\widetilde{L}_{W_{it_1}}, R_{W_i}), \cdots, (\widetilde{L}_{W_{it_m}}, R_{W_i})\}$, where R_{W_i} is the reachable distance.

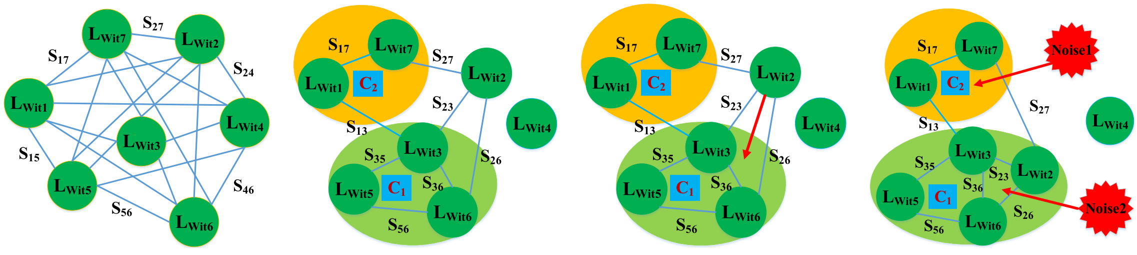
DynaLocPro Module: To support the privacy-preserving dynamic worker location publication, a dynamic correlation graph-based location obfuscation (DCGLO) algorithm is proposed to obtain the obfuscated dynamic worker location set $\{(\widetilde{L}^{t_{m_1}}(W_i), R_{W_i}), (\widetilde{L}^{t_{m_2}}(W_i), R_{W_i}), \cdots, (\widetilde{L}^{t_{m_k}}(W_i), R_{W_i})\}$ for different timestamps $\{t_{m_1}, t_{m_2}, \cdots, t_{m_k}\}$.

TaskAlloc Module: At any time, upon receiving the task location L_T and all the workers' obfuscated location information $\{(\widetilde{L}(W_1), R_{W_1}), (\widetilde{L}(W_2), R_{W_2}), \cdots, (\widetilde{L}(W_n), R_{W_n})\}$, the SC-server first searches the workers close to the task location by calculating the distances between the obfuscated locations and the real locations, then adopts a Linear acceptance model to obtain the allocated workers $\{W_{z_1}, W_{z_2}, \cdots, W_{z_k}\}$, and finally notifies these corresponding workers by returning the allocated workers and the task location information $(\{W_{z_1}, W_{z_2}, \cdots, W_{z_k}\}, L_T)$.

A. PRIVACY-PRESERVING LOCATION CORRELATIONS

The workers in SC services have specific movement patterns and these locations are correlated [43]. The works in [37], [45] also showed that the high correlated data may leak the user privacy. Especially, the locations with high correlations can reveal the gender of the user [45]. Therefore, protecting the location correlation privacy of workers in SC becomes important. The works in [37], [45] add noise to all location correlations. However, these methods can reduce the data utility since the low correlated locations are add noise and they cannot leak the privacy. The Pearson correlation coefficient in work [41] cannot be used to calculate the correlations between locations since the difference between the longitude and latitude attributes in the locations can affect the correlation [45]. Here, a **LDP-based correlation graph** (LDPCG) is proposed to protect the location correlation privacy. It contains: correlation graph construction, correlated location clustering and correlation noise addition.

First, since any two locations may have correlations, a complete undirect graph is used to represent the correlations of all locations $L(W_i) = \{L_{W_{it_1}}, L_{W_{it_2}}, \cdots, L_{W_{it_m}}\}$.



(a) Complete graph generation (b) Base graph clustering (c) Correlated location clustering (d) Correlation noise addition

FIGURE 2. **Privacy-preserving location correlations.**

In Fig. 2(a), each node is a location $L_{W_it_j}$ and each edge is a correlation score S_{jk}. Since the longitude and latitude in a location are independent of each other, we use a location correlation score to calculate the correlation S_{jk} between any locations $L_{W_it_j}$ and $L_{W_it_k}$. Given the locations $L_{W_it_j} = (l_{loi}^{tj}, l_{lai}^{tj})$ and $L_{W_it_k} = (l_{loi}^{tk}, l_{lai}^{tk})$ ($j, k \in [1, m]$), the correlation score S_{jk} is calculated as:

$$\begin{aligned} S_{ij} &= \exp(-|l_{loi}^{tj} - l_{loi}^{tk}|) \times \exp(-|l_{lai}^{tj} - l_{lai}^{tk}|) \\ &= exp(-|l_{loi}^{tj} - l_{loi}^{tk}| - |l_{lai}^{tj} - l_{lai}^{tk}|) \end{aligned} \tag{1}$$

When $|l_{loi}^{tj} - l_{loi}^{tk}|$ and $|l_{lai}^{tj} - l_{lai}^{tk}|$ are close to 0, the S_{ij} is close to 1. It means that the $L_{W_it_j}$ and $L_{W_it_k}$ are highly correlated. Otherwise, the $L_{W_it_j}$ and $L_{W_it_k}$ have low correlation. This change trend is in line with actual phenomena.

Second, inspired by the work in [41], since the low correlation weights S_{jk} (i.e., $S_{jk} < \sigma_1$) in the graph cannot leak the privacy, they are deleted. For example, in Fig. 2(b), the low correlated weights $\{S_{12}, S_{14}, S_{15}, S_{24}, \cdots,\}$ are deleted. Here, the locations having high correlations are clustered into different graph classes. In the following, we describe how to cluster the high correlated locations: (a) Assuming that high correlation scores with descending ranking are $\{S_{j_1k_1}, S_{j_2k_2}, \cdots, S_{j_pk_p}\}$, the biggest complete subgraph C_1 is calculated by searching the biggest correlated nodes both $L_{W_{ij_1}}$ and $L_{W_{ik_1}}$'s neighboring nodes. As shown in Fig. 2(b), if the locations $L_{W_it_3}$ and $L_{W_it_6}$ have the maximum correlation score S_{36}, they and their neighbor nodes $\{L_{W_it_2}, L_{W_it_5}\}$ can form two largest complete subgraphs $\{L_{W_it_2}, L_{W_it_3}, L_{W_it_6}\}$ and $\{L_{W_it_3}, L_{W_it_5}, L_{W_it_6}\}$. The $\{L_{W_it_3}, L_{W_it_5}, L_{W_it_6}\}$ is considered as a class C_1 when $S_{23} + S_{26} < S_{35} + S_{56}$; (b) we reobtain a new correlation ranking by moving the nodes in C_1 and the nodes linked to C_1, and the steps (a) and (b) are continued to generate the class C_l until there is no new correlation ranking; (c) According to the correlation scores, the nodes linked to the classes $\{C_1, C_2, \cdots C_q\}$ are clustered into different classes. In Fig. 2(c), the node $L_{W_it_2}$ is linked to the classes $\{C_1, C_2\}$. When $S_{23} + S_{26} > S_{27}$, we cluster the node $L_{W_it_2}$ into the class C_1. Therefore, after performing above steps, the high correlated locations are clustered into the classes $\{C_1, C_2, \cdots C_q\}$.

Finally, the locations in each class C_l ($l \in [1, q]$) have high correlations and different classes have different correlations. Some classes need to be added more noise since they have higher correlations, the locations in different classes are personally added different noise based on the LDP method instead of all locations. For the class C_l ($l \in [1, q]$), given any location $L_{W_it_{l_j}}$ in the C_l, based on the LDP's Definition 2 in Section III, the LDPCG algorithm adds Laplace noise to the $L_{W_it_{l_j}}$. Since each location is a two-dimensional vector containing longitude and latitude, a random unit vector $U_l = (y_{l_1}, y_{l_2})$ following Bernoulli distribution is used to the noise addition. That is, the noisy correlated location $L^c_{W_it_{l_j}} = L_{W_it_{l_j}} + U_l \times Lap_{l_j}(\frac{\Delta S(C_l)}{\epsilon_1})$, where $\Delta S(C_l)$ is the query sensitivity (e.g., following Theorem 1) for the class C_l ($l \in [1, q]$). After adding noise to the correlations, we combine the location set $L^c_{W_it}$ that are adding noise and the low correlated locations $L_{low}(W_i) = \{L_{W_it_{m_1}}, \cdots, L_{W_it_{m_r}}\}$ without adding noise (e.g., the node $L_{W_it_4}$ in Fig. 2) to obtain whole noisy correlated location set $L^c_{W_i} = \{L^c_{W_it_1}, L^c_{W_it_2}, \cdots, L^c_{W_it_m}\}$.

Theorem 1: For each original class $C_l = \{L_{W_it_{l_1}}, L_{W_it_{l_2}}, \cdots, L_{W_it_{l_v}}\}$*, when the correlation score between the locations* $L_{W_it_{l_j}}$ *and* $L_{W_it_{l_k}}$ *is* S_{jk}*, thus the query sensitivity* $\Delta S(C_l) = \max\limits_{L_{W_it_{l_j}}, L_{W_it_{l_k}} \in C_l} (S_{jk}) \leq 1$.

Proof: The Appendix A gives the detail proofs. ■

B. WORKER LOCATION PROTECTION

Although the location correlations are protected in Section IV-A, the workers' real locations are not fully protected, which can leak the worker privacy. In this section, we propose a distance score-based LDP (DSLDP) algorithm to obfuscate the real worker locations. Our DSLDP algorithm contains two phases: (a) location distance score calculation, and (b) location obfuscation.

First, the worker locations $L^c_{W_it}$ consist of a trajectory sequence that contains m locations $\{L^c_{W_it_1}, L^c_{W_it_2}, \cdots, L^c_{W_it_m}\}$. Therefore, the location sequence $L^c_{W_it}$ can be split into $m - 1$ segments and any two neighbor locations form a segment. The distance between two locations can leak the worker privacy since these locations follow specific movement patterns. Here, we allocate a distance score DS_{t_j} to the distance segment between the neighbor locations $L^c_{W_it_j}$ and $L^c_{W_it_{j+1}}$ ($j \in [1, m - 1]$). The distance score DS_{t_j} is defined as:

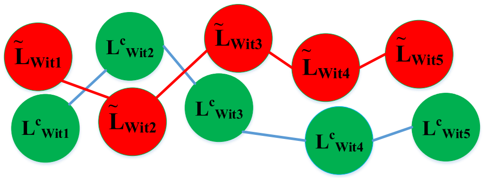


FIGURE 3. Privacy-preserving worker Locations.

Algorithm 1 DSLDP Algorithm

```
Input: The dataset $L^c_{W_it} = (L^c_{W_it_1}, \cdots, L^c_{W_it_m})$, privacy
       budget $\epsilon_2$
Output: Noisy Locations $\widetilde{L}_{W_it} = (\widetilde{L}_{W_it_1}, \cdots, \widetilde{L}_{W_it_m})$
Initialize $\widetilde{L}_{W_it} = \emptyset$;
for each location $L^c_{W_it_j}, L^c_{W_it_j} \in L^c_{W_it}$ do
    $d(L^c_{W_it_j}, L^c_{W_it_{j+1}}) = \sqrt{(l^{ct_j}_{loi} - l^{ct_{j+1}}_{loi})^2 + (l^{ct_j}_{lai} - l^{ct_{j+1}}_{lai})^2}$;
for each location $L^c_{W_it_j} \in L^c_{W_it}$ do
    $DS_{t_j} = \frac{d(L^c_{W_it_j}, L^c_{W_it_{j+1}})}{\sum_{j=1}^{m-1} d(L^c_{W_it_j}, L^c_{W_it_{j+1}})}$;
    Generate a random unit vector $\Upsilon_j$ following the
    Bernoulli distribution;
    $\widetilde{L}_{W_it_j} = L^c_{W_it_j} + Lap(\frac{\Delta DS(L^c_{W_it})}{\epsilon_2}) \times \Upsilon_j$;
    $\widetilde{L}_{W_it} = \widetilde{L}_{W_it} \bigcup \widetilde{L}_{W_it_j}$
return $\widetilde{L}_{W_it}$;
```

*Definition 5: (**Distance Score**). Given any two neighbor locations $L^c_{W_it_j} = (l^{ct_j}_{loi}, l^{ct_j}_{lai})$ and $L^c_{W_it_{j+1}} = (l^{ct_{j+1}}_{loi}, l^{ct_{j+1}}_{lai})$ ($j \in [1, m-1]$) in noisy correlated location set $L^c_{W_it}$, their location distance $d(L^c_{W_it_j}, L^c_{W_it_{j+1}}) = \sqrt{(l^{ct_j}_{loi} - l^{ct_{j+1}}_{loi})^2 + (l^{ct_j}_{lai} - l^{ct_{j+1}}_{lai})^2}$, thus the distance score DS_{t_j} is calculated as:*

$$DS_{t_j} = \frac{d(L^c_{W_it_j}, L^c_{W_it_{j+1}})}{\sum_{j=1}^{m-1} d(L^c_{W_it_j}, L^c_{W_it_{j+1}})}$$

Second, to protect the worker location privacy, we obfuscate the real worker locations. That is, each real location $L_{W_it_j}$ should be added Laplace noise. Therefore, given a privacy budget ϵ_2, the noisy location $\widetilde{L}_{W_it_j} = L^c_{W_it_j} + Lap(\frac{\Delta DS(L^c_{W_it})}{\epsilon_2}) \times \Upsilon_j$ ($j \in [1, m]$), where $\Upsilon_j = (x_{j1}, x_{j2})$ is a random vector that satisfies the Bernoulli distribution, and $\Delta DS(L^c_{W_it})$ is the query sensitivity of distance score (e.g., following Theorem 2). For example, in Fig. 3, based on our DSLDP algorithm, the real worker locations $\{L^c_{W_it_1}, L^c_{W_it_2}, L^c_{W_it_3}, L^c_{W_it_4}, L^c_{W_it_5}\}$ are obfuscated into the noisy locations $\{\widetilde{L}_{W_it_1}, \widetilde{L}_{W_it_2}, \widetilde{L}_{W_it_3}, \widetilde{L}_{W_it_4}, \widetilde{L}_{W_it_5}\}$.

Our DSLDP algorithm is shown in Algorithm 1. First, the distance $d(L^c_{W_it_j}, L^c_{W_it_{j+1}})$ between two locations $L^c_{W_it_j}$ and $L^c_{W_it_{j+1}}$ is calculated (Lines 2 to 3). Then DSLDP algorithm calculates the distance score DS_{t_j} and adds Laplace noise to the real location $L^c_{W_it_j}$ (Lines 4 to 7). Finally, the noisy worker location set $\widetilde{L}_{W_it}$ is obtained (Lines 8 to 9).

Algorithm 2 DCGLO Algorithm

```
Input: Dynamic locations $\{L_{W_it_1}, \cdots, L_{W_it_m}\}$, privacy
       budgets $\epsilon_1, \epsilon_2$, parameters $h_1$
Output: Noisy locations $\widetilde{L}_{W_it} = (\widetilde{L}_{W_it_1}, \cdots, \widetilde{L}_{W_it_m})$
Initialize $\widetilde{L}_{W_it} = \emptyset$;
for any two locations $L_{W_it_{j_1}}$ and $L_{W_it_{j_2}}$ do
    $\widetilde{S}_{j_1j_2} = S_{j_1j_2} + Lap(\frac{1}{\epsilon_1 h_1})$;
Calculate noisy correlated classes $\widetilde{C} = \{\widetilde{C}_1, \cdots, \widetilde{C}_l\}$ at
time $t_j$ based on LDPCG algorithm;
for each $\widetilde{C}_p \in \widetilde{C}$ do
    for each time $t_k \in [t_j, t_m]$ do
        Cluster high correlated locations into different
        classes;
        $\widetilde{L}_{t_j}(\widetilde{C}_p) = L_{t_j}(\widetilde{C}_p) + U_p \times Lap(\frac{\Delta S(\widetilde{C}_p)}{(1-g_1)\epsilon_1})$;
    $\widetilde{C} \longleftarrow$ update $\{\widetilde{L}_{t_j}(\widetilde{C}_p), \cdots, \widetilde{L}_{t_m}(\widetilde{C}_p)\}$;
$L^c_{W_it} = \widetilde{C} \bigcup \{L_{W_it_{m_1}}, \cdots, L_{W_it_{m_r}}\}$ without edges;
for each location $L^c_{W_it_j} \in L^c_{W_it}$ do
    Calculate distance score $DS_{t_j}$ based on Algorithm 2;
    $\widetilde{L}_{W_it_j} = L^c_{W_it_j} + Lap(\frac{m \times \Delta DS(L^c_{W_it})}{\epsilon_2}) \times \Upsilon_j$ for an unit
    vector $\Upsilon_j$;
    $\widetilde{L}_{W_it} = \widetilde{L}_{W_it} \bigcup \widetilde{L}_{W_it_j}$;
return $\widetilde{L}_{W_it}$;
```

Theorem 2: Given an original location dataset $L^c_{W_it} = (L^c_{W_it_1}, \cdots, L^c_{W_it_m})$, for two neighbor locations $L^c_{W_it_j}$ and $L^c_{W_it_{j+1}}$, their distance score is DS_{t_j}. Thus the query sensitivity $\Delta DS(L^c_{W_it}) = \max_{L^c_{W_it_j}, L^c_{W_it_{j+1}} \in L^c_{W_it}} (DS_{t_j} + DS_{t_{j+1}}) \leq 2$.

Proof: Please refer to the Appendix B. ■

Finally, our DSLDP algorithm can generate noisy locations $\widetilde{L}_{W_it} = (\widetilde{L}_{W_it_1}, \cdots, \widetilde{L}_{W_it_m})$ locally of the worker W_i. The W_i can outsource them to the SC-server safely.

C. PRIVACY-PRESERVING DYNAMIC WORKER LOCATIONS

In practice, the workers can dynamically move locations to perform tasks. However, releasing dynamic worker locations can leak the privacy. Therefore, the dynamic worker locations should be protected. Given m timestamps, if we directly adopt the LDPCG algorithm in Section IV-A and the DSLDP algorithm in Section IV-B to protect the worker correlation and location privacy at each timestamp, based on the nature of LDP [37] and the sequential composition theorem in [29], the noisy worker location $\widetilde{L}_{W_{it}}$ at each timestamp t_j ($j \in [1, m]$) achieves ($\epsilon_1 + \epsilon_2$)-LDP. Further, the whole noisy worker locations $\widetilde{L}_{W_it} = (\widetilde{L}_{W_it_1}, \cdots, \widetilde{L}_{W_it_m})$ at m timestamps satisfy $m(\epsilon_1 + \epsilon_2)$-LDP. When m is larger, the total privacy budget $m(\epsilon_1 + \epsilon_2)$ is larger, which reduces the privacy protection level. In this section, we propose a **dynamic correlation graph-based location obfuscation** (DCGLO) algorithm to protect the dynamic worker location privacy while providing high data utility.

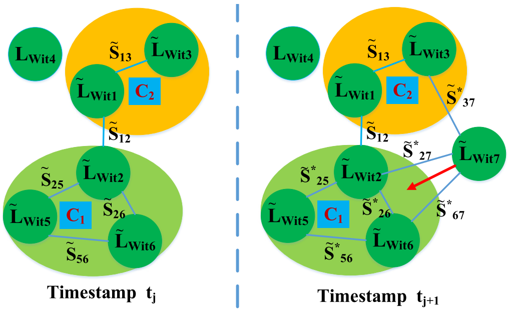


FIGURE 4. Locations and correlations in correlation classes.

Our **DCGLO** algorithm is shown in **Algorithm 2**. Given original dynamic worker locations $L(W_i) = \{L_{W_it_1}, L_{W_it_2}, \cdots, L_{W_it_m}\}$, our DCGLO algorithm contains two phases: **dynamic correlation privacy and dynamic location privacy**. First, **for the dynamic correlation privacy protection phase**, given a privacy budget ϵ_1, the correlations between any two locations at first j timestamps are added $Lap(\frac{1}{\epsilon_1 h_1})$ Laplace noise by allocating the privacy budget $\epsilon_1 h_1$ (e.g., $h_1 \in (0,1)$) (Lines 2 to 3). Based on the LDPCG algorithm in Section IV-A, DCGLO algorithm clusters the noisy correlation classes $\{\widetilde{C}_1, \cdots, \widetilde{C}_l\}$ (Line 4). In each class $\widetilde{C}_p$ at timestamps t_j to t_m, we cluster high correlated locations into the $\widetilde{C}_p$ and add $U_p \times Lap(\frac{\Delta S(\widetilde{C}_p)}{(1-g_1)\epsilon_1})$ noise to the locations in $\widetilde{C}_p$ by using privacy budget $(1-h_1)\epsilon_1$ (Lines 5 to 8). Then the noisy high correlated locations $L^c_{W_it}$ are generated by updating the classes $\{\widetilde{L}_{t_1}(\widetilde{C}_p), \cdots, \widetilde{L}_{t_m}(\widetilde{C}_p)\}$ at m timestamps (Lines 9 to 10). **For the dynamic location protection phase**, to achieve ϵ_2-LDP, based on the DSLDP algorithm in Section IV-B, the location $\widetilde{L}_{W_it_j}$ at each timestamp is added $Lap(\frac{m \times \Delta DS(L^c_{W_it})}{\epsilon_2}) \times \Upsilon_j$ noise with a privacy budget $\frac{\epsilon_2}{m}$ (Lines 11 to 14). Finally, the noisy dynamic worker locations $\widetilde{L}_{W_it} = (\widetilde{L}_{W_it_1}, \cdots, \widetilde{L}_{W_it_m})$ are obtained (Line 15). According to the sequential composition theorem in [29], the dynamic correlation privacy achieves $(\epsilon_1 h_1 + (1-h_1)\epsilon_1)$-LDP and the dynamic location privacy achieves $(m \times \frac{\epsilon_2}{m})$-LDP. In summary, our DCGLO algorithm for the whole dynamic worker location protection achieves $(\epsilon_1 + \epsilon_2)$-LDP, which ensures the privacy.

The noisy locations in dynamic location privacy can be directly calculated by adding noise with $\frac{\epsilon_2}{m}$. In the dynamic location correlation protection, since the locations and the correlations in the class at different timestamp are different. Here, we use an example to illustrate how we update the locations and correlations at different timestamps. The main updates contain: (i) location addition and (ii) correlation update. In Fig. 4, from the timestamps t_j to t_{j+1}, the location $L_{W_it_7}$ is added and linked to classes $\{\widetilde{C}_1, \widetilde{C}_2\}$. Based on the LDPCG algorithm, since noisy correlations $\widetilde{S}_{27} + \widetilde{S}_{67} > \widetilde{S}_{37}$, thus the location $L_{W_it_7}$ is clustered into the class $\widetilde{C}_1$. According to our DCGLO algorithm, we add noise to all the locations $\{L_{W_it_2}, L_{W_it_5}, L_{W_it_6}, L_{W_it_7}\}$ in $\widetilde{C}_1$. Therefore, the noisy correlations $\{\widetilde{S}_{25}, \widetilde{S}_{26}, \widetilde{S}_{27}, \widetilde{S}_{56}, \widetilde{S}_{67}\}$ are updated to $\{\widetilde{S}^*_{25}, \widetilde{S}^*_{26}, \widetilde{S}^*_{27}, \widetilde{S}^*_{56}, \widetilde{S}^*_{67}\}$. Correspondingly, the noisy correlations $\{\widetilde{S}_{12}, \widetilde{S}_{37}\}$ are updated to $\{\widetilde{S}^*_{12}, \widetilde{S}^*_{37}\}$ since the correlations $\{\widetilde{L}_{W_it_2}, \widetilde{L}_{W_it_2}\}$ are added noise.

Suitable parameter h_1 calculation: Since the dynamic correlation graph class generation is allocated privacy budget $h_1\epsilon_1$, and the noisy correlation calculation is allocated privacy budget $(1-h_1)\epsilon_1$, when h_1 is close to 1, the generated graph classes have high accuracy since they have larger privacy budget, but the noisy correlations have low data utility even are unusable. Here, a privacy budget allocation method is proposed to ensure the data utility. For simplicity, we allocate the privacy budget $\frac{(1-h_1)\epsilon_1}{m}$ to each class correlation $S(\widetilde{C}_p)$ $(p \in [j, m])$. Based on the nature of LDP [37], the class correlation $S(\widetilde{C}_p)$ follows $Lap(\frac{\Delta S(\widetilde{C}_p)m}{(1-g_1)\epsilon_1})$ Laplace distribution. Each class $\widetilde{C}_p$ should have at least two locations since the correlation $S(\widetilde{C}_p)$ cannot be null. Based on the $Lap(\frac{\Delta S(\widetilde{C}_p)m}{(1-g_1)\epsilon_1})$ Laplace distribution [29], [34], the probability $P(h_1)$ that satisfies $\widetilde{S}(\widetilde{C}_p) > 0$ is: $P(h_1) = 1 - \frac{1}{2}\exp(-\frac{\widetilde{S}(\widetilde{C}_p)(1-g_1)\epsilon_1}{\Delta S(\widetilde{C}_p)m})$. According to the Laplace distribution [29], the noisy standard deviation of a two-dimensional location in $\widetilde{C}_p$ is $\frac{4\sqrt{2}\Delta S(\widetilde{C}_p)}{\epsilon_1 g_1}$. To ensure the high probability $P(h_1)$ that $\widetilde{C}_p$ has at least two locations, the noisy class correlation $\widetilde{S}(\widetilde{C}_p)$ should satisfy $\widetilde{S}(\widetilde{C}_p) \geq \frac{4\sqrt{2}\Delta S(\widetilde{C}_p)}{\epsilon_1 g_1}$. When $\widetilde{S}(\widetilde{C}_p) = \frac{4\sqrt{2}\Delta S(\widetilde{C}_p)}{\epsilon_1 g_1}$, thus $P(h_1)$ is:

$$P(h_1) = 1 - \frac{1}{2}\exp(-\frac{4\sqrt{2}(1-h_1)}{mh_1}) \tag{2}$$

D. TASK ALLOCATION

Since the outsourced worker locations $\widetilde{L}_{W_it} = (\widetilde{L}_{W_it_1}, \cdots, \widetilde{L}_{W_it_m})$ contain noise, the SC-server cannot accurately allocate suitable tasks to nearby workers from the unknown worker locations. To achieve task allocation with high data utility, a *Linear Acceptance Model* (LAM) is first to proposed.

1) LINEAR ACCEPTANCE MODEL

Following the works in [32], [34], [43], in our paper, the workers perform tasks voluntarily without considering rewards. In real SC scenes, since the workers like to perform nearby tasks and do not want to move large distance to perform tasks, thus the move distance is a key for task allocation. That is, the acceptance rate that a worker can perform tasks decreases as the move distance increases. Once the distance between worker location and task location is bigger than the reachable maximum distance of the worker, the acceptance rate is 0. For simplicity, we use a linear function to evaluate the acceptance rate since the acceptance rate decreases as the distance increases. Therefore, the acceptance rate $PA(\widetilde{L}_{W_it_j})$ of a worker at location $\widetilde{L}_{W_it_j}$ is defined.

*Definition 6: (**Linear Acceptance Model**). Given a worker's location $\widetilde{L}_{W_it_j}$ and the task location L_T, let $d(\widetilde{L}_{W_it_j}, L_T)$ be the distance between worker location $\widetilde{L}_{W_it_j}$ and task location L_T. If the reachable distance outsourced by the worker W_i is R_{W_i}, the acceptance rate $PA(\widetilde{L}_{W_it_j})$ that the* *worker W_i performs the task T is*:

$$PA(\widetilde{L}_{W_it_j}) = \begin{cases} 1 - \dfrac{d(\widetilde{L}_{W_it_j}, L_T)}{R_{W_i}}, & d(\widetilde{L}_{W_it_j}, L_T) \le R_{W_i} \\ 0, & d(\widetilde{L}_{W_it_j}, L_T) > R_{W_i}. \end{cases}$$

Some SC services such as environmental testing [5] and traffic monitoring [6] may require multiple workers to perform the same task. Therefore, we give the acceptance rate model that multiple workers perform the same tasks.

*Definition 7: (**Acceptance Rate of Multiple Workers**). For n workers, if a task T requires at leat k workers to perform it and the worker W_i's acceptance rate is $PA(\widetilde{L}_{W_it_j})$, thus the acceptance rate PA_T^k that at leat k workers perform the task T is calculated as*:

$$PA_T^k = 1 - \sum_{l=0}^{k-1} \binom{n}{l} (PA(\widetilde{L}_{W_it_j}))^l (1 - PA(\widetilde{L}_{W_it_j}))^{n-l}$$

2) ALLOCATED TASK CALCULATION

At the time point t_j, upon receiving the task location L_T and all the workers' obfuscated location information $\{(\widetilde{L}_{W_1t_j}, R_{W_1}), (\widetilde{L}_{W_2t_j}, R_{W_2}), \cdots, (\widetilde{L}_{W_nt_j}, R_{W_n})\}$, the SC-server searches all obfuscated worker locations to calculate the workers who close to the task location L_T. Therefore, we should achieve the following two goals: (i) the SC-server should allocate sufficient workers so that the task can have high acceptance rate; and (ii) the number of allocated workers should be as small as possible so that the we can reduce the system overhead. However, these two goals are in conflict. To achieve the balance between the acceptance rate and the system overhead, a **linear acceptance model-based task allocation** (LAMTA) algorithm is proposed as shown in **Algorithm 3**. Our LAMTA contains three phases: calculating the workers who may perform the task, multi-worker acceptance rate calculation, and task allocation.

First, based on the reachable maximum distance R_{W_i} outsourced by the worker W_i ($i \in [1, n]$), when the distance $d(\widetilde{L}_{W_it_j}, L_T)$ between the worker location $\widetilde{L}_{W_it_j}$ and task location L_T is smaller than R_{W_i} (**Lines** 2 **to** 4 **in Algorithm 3**), we consider that it is possible for the worker W_i to perform the task T. Therefore, we obtain the workers $W_p = \{W_{p_1}, W_{p_2}, \cdots, W_{p_l}\}$ who may perform the task (Line 5).

Second, since a task may require multiple workers to perform it, to ensure that the task T can be accepted, we set threshold σ_2 ($\sigma_2 \in (0, 1]$). When multi-workers' acceptance rate $PA_T^k \ge \sigma_2$, we consider that these k workers can complete the task T. To ensure high acceptance rate, according to the linear acceptance mode, we calculate the acceptance rate $PA(\widetilde{L}_{W_{p_r}t_j}) = 1 - \frac{d(\widetilde{L}_{W_{p_r}t_j}, L_T)}{R_{W_{p_r}}}$ of the worker W_{p_r} in the dataset W_p ($r \in [1, l]$) (Line 6). When at least k workers in dataset W_p perform the task, these workers' acceptance rate is: $PA_T^k = 1 - \sum_{r=0}^{k-1} \binom{l}{u} (PA(\widetilde{L}_{W_{p_r}t_j}))^u (1 - PA(\widetilde{L}_{W_{p_r}t_j}))^{l-u}$.

Finally, the SC-server performs task allocation. To allocate sufficient workers and achieve high acceptance rate, we first sort all the acceptance rates in descending order, denoted as $PA = \{PA(\widetilde{L}_{W_{z_1}t_j}), PA(\widetilde{L}_{W_{z_2}t_j}), \cdots, PA(\widetilde{L}_{W_{z_l}t_j})\}$ (Line 7).

Algorithm 3 DCGLO Algorithm

```
Input: Noisy worker locations
    $\{(\widetilde{L}_{W_1t_j}, R_{W_1}), (\widetilde{L}_{W_2t_j}, R_{W_2}), \cdots, (\widetilde{L}_{W_nt_j}, R_{W_n})\}$,
    task location $L_T$, threshold $\sigma_2$
Output: Allocated workers $\{W_{z_1}, W_{z_2}, \cdots, W_{z_k}\}$
Initialize $W_p = \emptyset$;
for each location $L_{W_it_{j_1}}$ do
    Calculate the distance $d(\widetilde{L}_{W_it_j}, L_T)$;
    if $d(\widetilde{L}_{W_it_j}, L_T) < R_{W_i}$ then
        $W_p = W_p \bigcup \widetilde{L}_{W_it_j}$;
        $PA(\widetilde{L}_{W_{p_r}t_j}) = 1 - \frac{d(\widetilde{L}_{W_{p_r}t_j}, L_T)}{R_{W_{p_r}}}$;
$PA = \{PA(\widetilde{L}_{W_{z_1}t_j}), \cdots, PA(\widetilde{L}_{W_{z_l}t_j})\}$ in descending order;
for each $k \in [1, l]$ do
    Select the workers with top-$r$ acceptance rates;
    $PA_T^k = 1 - \sum_{v=0}^{k-1} \binom{l}{v} (PA(\widetilde{L}_{W_{z_h}t_j}))^v (1 - PA(\widetilde{L}_{W_{z_h}t_j}))^{l-v}$;
    if $PA_T^k < \sigma_2$ then
        Goto Line 8;
    else
        Select the workers $\{W_{z_1}, W_{z_2}, \cdots, W_{z_k}\}$;
        Break;
return $\{W_{z_1}, W_{z_2}, \cdots, W_{z_k}\}$;
```

Then we select the worker W_{z_1} with the maximum acceptance rate $PA(\widetilde{L}_{W_{z_1}t_j})$. If $PA(\widetilde{L}_{W_{z_1}t_j}) < \sigma_2$, we continue to select the workers $\{W_{z_1}, W_{z_2}\}$ with top-2 acceptance rates, and calculate the acceptance rate PA_T^2, this step is continued to perform when $PA_T^2 < \sigma_2$ (Lines 8 to 12). To reduce the system overhead, when $PA_T^k \ge \sigma_2$, we no longer select the workers. Therefore, the workers $\{W_{z_1}, W_{z_2}, \cdots, W_{z_k}\}$ with top-k acceptance rates are allocated to perform the task T (Lines 13-15). The SC-server returns the allocated results $\{W_{z_1}, W_{z_2}, \cdots, W_{z_k}, L_T\}$ to the responding workers for performing the task T.

V. LDPDW SCHEME ANALYSIS

A. PRIVACY ANALYSIS

As described in Section IV, our LDPDW scheme contains four algorithms: *LDPCG, DSLDP, DCGLO* and *LAMTA*. Since the SC-server performs the LAMTA algorithm from the obfuscated worker locations and cannot infer the real worker locations, thus the LAMTA algorithm is secure. Since the LDPCG algorithm and the DSLDP algorithm all adopt LDP method by allocating the privacy budgets ϵ_1 and ϵ_2, respectively, thus we will first prove that the LDPCG algorithm achieves ϵ_1-LDP and the DSLDP algorithm achieves ϵ_2-LDP. Since the algorithm DCGLO protects the dynamic worker location privacy based on the the LDPCG algorithm and the DSLDP algorithm, the DCGLO algorithm will be proved to satisfy $(\epsilon_1 + \epsilon_2)$-LDP. Finally, the LDPDW scheme will be proved to achieve ϵ-LDP (i.e., $\epsilon = \epsilon_1 + \epsilon_2$).

Theorem 3: Our LDPCG algorithm follows ϵ_1-LDP.

Proof: Please refer to the Theorem 3 of Appendix C. ■

Theorem 4: Our DSLDP algorithm achieves ϵ_2-LDP.

Proof: Please refer to the Theorem 4 of Appendix C. ■

Theorem 5: The algorithm DCGLO achieves $(\epsilon_1 + \epsilon_2)$-LDP.

Proof: As described in Section IV-C, we protect the dynamic location correlation and location privacy based on a DCGLO algorithm. For the correlation privacy, DCGLO generates the noisy graph class structure by allocating the privacy budget $h_1\epsilon_1$, and then uses the privacy budget $(1 - h_1)\epsilon_1$ to calculate high correlated locations of all timestamps. According to the sequential composition of LDP in [29], this phase achieves $h_1\epsilon_1 + (1 - h_1)\epsilon_1$ (i.e., ϵ_1)-LDP. To ensure the dynamic location privacy, the worker location at each timestamp is allocated privacy budget $\frac{\epsilon_2}{m}$. For m timestamps, the whole noisy worker locations satisfy $(m \times \frac{\epsilon_2}{m})$-LDP. In summary, when $\epsilon = \epsilon_1 + \epsilon_2$, our DCGLO algorithm achieves ϵ-LDP. ■

Theorem 6: Our LDPDW scheme follows ϵ-LDP.

Proof: According to previous analysis, the *LAMTA* algorithm satisfies 0-LDP since it does not add any noise. The algorithms *LDPCG* and *DSLDP* are static privacy protection situations. *DCGLO* is a dynamic privacy protection situation. Therefore, for the static situation, based on Theorem 3 and Theorem 4, the *LDPCG* and *DSLDP* achieve ϵ_1-LDP and ϵ_2-LDP, respectively. According to the sequential composition of LDP in [29], our LDPDW scheme in static situation follows $\epsilon_1 + \epsilon_2$-LDP. In dynamic worker location privacy case, since *DCGLO* achieves $(\epsilon_1 + \epsilon_2)$-LDP based on above Theorem 5, thus LDPDW scheme also meets $(\epsilon_1 + \epsilon_2)$-LDP. Let $\epsilon = \epsilon_1 + \epsilon_2$, thus our LDPDW scheme follows ϵ-LDP. ■

B. TIME COMPLEXITY ANALYSIS

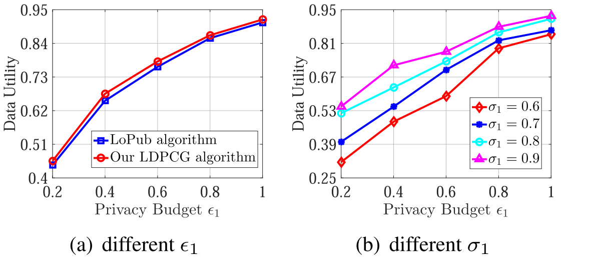
Here, we analyze the time complexities of four algorithms in our LDPDW scheme, such as algorithms LDPCG, DSLDP, DCGLO and TAMTA.

First, LDPCG algorithm calculates the correlations between m locations, and clusters the high-correlated locations into q graph classes. This requires $O(2m^2)$ time complexity. Further, each graph class is added noise. Thus the total time complexity of LDPCG algorithm is $O(2m^2 + q)$.

Second, for m locations, the algorithm DSLDP first calculates the distance scores between any two neighbor locations, which requires $O(m-1)$ time complexity. Then we obfuscate each location. Therefore, the total time complexity is $(2m-1)$.

Third, for the dynamic correlation privacy, similar to the LDPCG algorithm, the DCGLO algorithm first calculates the l noisy graph classes at first j timestamps. Its time complexity is $O(2j^2+l)$. Then, from j-th timestamp to the m-th timestamp, each graph class is added noise and updated, which requires $O(m - j + 1)$. For the dynamic location privacy, the location of each timestamp is obfuscated. Therefore, the DCGLO algorithm's time complexity is $O(2j^2 + 3(m - j) + 1)$.

Finally, for n workers, TAMTA algorithm calculates the distance between each worker and a task. When the task requires w workers to perform it, thus TAMTA will need to select the workers with top-w acceptance rates. The total time complexity of TAMTA algorithm is $O(n + w)$.



(a) different ϵ_1

(b) different σ_1

FIGURE 5. The effects of different ϵ_1 and σ_1 on the LoPub algorithm and our LDPCG algorithm.

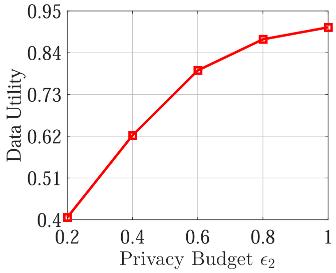


FIGURE 6. The performance evaluation of DSLDP algorithm for different ϵ_2.

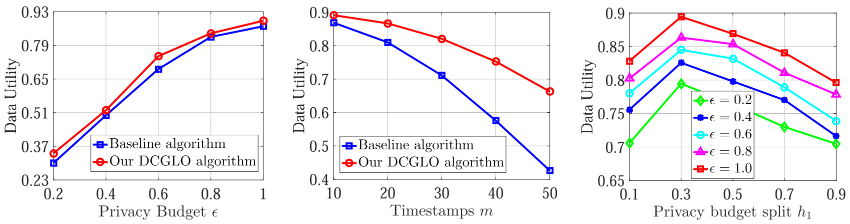
VI. EXPERIMENTS

In the following, our LDPDW scheme is evaluated by performing experiments. The python2.7 platform is used to run our experimental programs, and the experimental environment is based on the windows system having 8GB memory and a @2.5GHz Intel Core CPU.

A. EXPERIMENTAL DATASET AND SETUP

Here, our experiments uses a real crowdsourcing dataset named Gowalla [48]. Similar to the works in [32], [34], we take the users of Gowalla dataset in San Franciso area as the workers. The most recent check-ins locations are the workers' locations, and the locations of tasks are the check-ins points except the most recent points. Since our scheme contains four modules, the proposed scheme is evaluated by the following phases: correlation and location protection evaluation, dynamic location protection evaluation and task allocation evaluation. According to the work in [34], the 90% workers' maximum move distance is no bigger than 3.6 km, thus we set $R_W = 3.6$km.

Following the parameter settings of works in [32], [34], [37], the privacy budgets $\epsilon_1, \epsilon_2 \in \{0.1, 0.2, \cdots, 1\}$. The correlation threshold $\sigma_1 \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$. The budget split $h_1 \in \{0.1, 0.2, \cdots, 0.9\}$ and the timestamps $m \in \{10, 20, 30, 40, 50\}$. We set worker's acceptance rate $PA = \{0.1, 0.2, \cdots, 0.9\}$ and task acceptance threshold $\sigma_2 \in \{0.6, 0.7, 0.8, 0.9\}$. The number of the workers needed by the task is set to $k \in \{2, 4, 6, 10\}$. Since we are the



(a) different ϵ. (b) different m. (c) different h_1.

FIGURE 7. Effects of ϵ, m and h_1 on the performance of different algorithms in dynamic worker location protection.

first to consider the privacy-preserving dynamic workers in SC, based on different parameters, we only evaluate the performance of our scheme without considering comparing the state-of-the-art static LDP methods.

B. CORRELATION AND LOCATION PROTECTION EVALUATION

To evaluate the utility of noisy data, the data utility is defined as: $\frac{|R\cap\widetilde{R}|}{|R|}$, where $|R \cap \widetilde{R}|$ is the number of common elements between real results and noisy results. The LDPCG algorithm in Section IV-A protects the location correlation privacy. Thus we first evaluate the data utility of our LDPCG algorithm by varying privacy budget ϵ_1 and the correlation threshold σ_1, compared with the performance of the LoPub algorithm in [37]. In Fig. 5, when ϵ_1 increases, the data utilities of our LDPCG algorithm and the LoPub algorithm in [37] increase. This is because we add less noise to the correlated locations for a larger ϵ_1 so the noisy correlated locations are closer to the real results. These results are consistent with the nature of LDP. For the same ϵ_1, our LDPCG algorithm's data utility is higher than the LoPub algorithm in [37] since our LDPCG algorithm only adds noise to the high correlated locations instead of all locations. In Fig. 5(b), the data utility is larger when the threshold σ_1 is larger. The reason is that only the locations with higher correlations are added noise when the σ_1 increases, so the less locations in original dataset are added noise, which results in higher data utility.

Second, for different privacy budget ϵ_2, our DSLDP algorithm proposed in Section IV-B is evaluated as shown in Fig. 6. As the ϵ_2 increases, our DSLDP algorithm's data utility is gradually decreasing, which is in line with the LDP's nature. Especially, when the $\epsilon_2 = 1$, the data utility of our DSLDP algorithm up to 90.687%. These results show that the proposed DSLDP algorithm achieves high data utility while ensuring the worker location privacy.

C. DYNAMIC LOCATION PROTECTION EVALUATION

In Section IV-C, a *baseline algorithm* that directly uses the LDPCG and DSLDP algorithms with privacy budget $\frac{\epsilon}{m}$ at each timestamp and the DCGLO algorithm are proposed to protect the dynamic correlation and the location privacy. Here, we evaluate these two algorithms' performance by using different privacy budget ϵ ($\epsilon = \epsilon_1 + \epsilon_2$), budget split h_1 and the timestamps m.

First, in Fig. 7(a), for the different privacy budget ϵ, as ϵ increases, the data utilities of the baseline algorithm and the DCGLO algorithm increase. For a larger ϵ, we add less noise to the real results so that the data utility is larger. Besides, when the ϵ is the same, our DCGLO algorithm's data utility is larger than the baseline algorithm. Since our DCGLO algorithm adopts the optimal privacy budget allocation model, which can ensure the data utility.

In Fig. 7(b), two algorithms' data utilities are evaluated for different m. Our DCGLO's performance is better than the baseline algorithm. This is because the privacy budget $\frac{\epsilon_1}{m}$ of each location for the baseline algorithm is smaller when m is larger, which reduces the data utility. The privacy budget of the noisy correlated locations generated by the DCGLO is always ϵ_1 for different m, thus our DCGLO algorithm cannot significantly reduce the data utility.

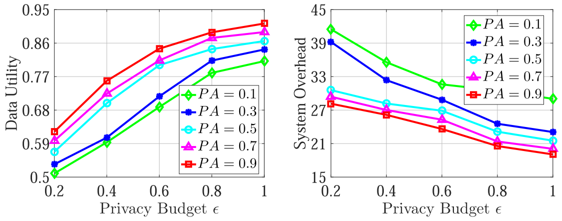
Our DCGLO algorithm's data utility is evaluated by varying h_1 as shown in Fig. 7(c). As the ϵ increases, the data utility is first gradually increasing and then decreasing. The reasons are: When the h_1 is larger, although the accuracy of the clustered noisy graph classes is higher, the data utility of the generated correlated locations is smaller since the smaller privacy budget $h_1\epsilon_1$ is allocated to the location correlation protection. Especially, when $h_1 = 0.3$, our DCGLO algorithm has higher data utility.

D. TASK ALLOCATION EVALUATION

Most importantly, we evaluate the performance of our LAMTA algorithm over the Gowalla dataset by varying ϵ, acceptance rate *PA*, acceptance threshold σ_2 and the number k required by the task. The data utility and system overhead indicators designed by the Section II-C are used to evaluate the task allocation performance of LAMTA algorithm.

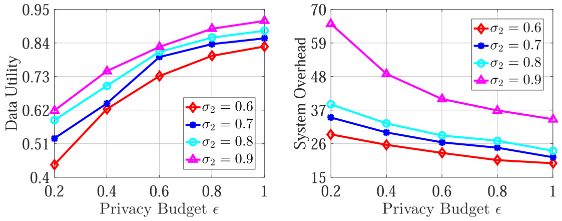
1) THE EFFECTS OF ϵ AND *PA*

First, different ϵ and *PA* affects the data utility and system overhead of our LAMTA algorithm as shown in Fig. 8. For the Fig. 8(a), when ϵ increases, the LAMTA algorithm's data utility increases. The reason is that a larger ϵ results in the noisy locations with higher data utility so that the allocated results have high accuracy, which is consists with the previous



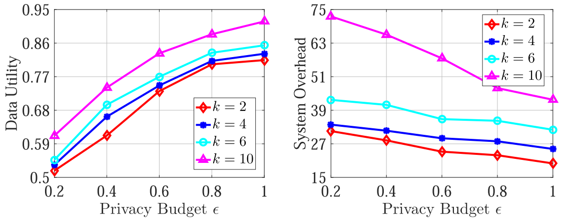
(a) data utility (b) system overhead

FIGURE 8. Effects of different ϵ, and *PA* in task allocation.



(a) data utility (b) system overhead

FIGURE 9. Effects of different ϵ and σ_2 in task allocation.



(a) data utility (b) system overhead

FIGURE 10. Effects of different ϵ and k in task allocation.

results. When ϵ is the same, the data utility is larger for a larger *PA* since the workers are more like to perform the task. In Fig. 8(b), the system overhead of our LAMTA algorithm decreases when ϵ increases. This is because LAMTA algorithm allocates a smaller of number of workers for obtaining the same acceptance rate when the ϵ is larger. When ϵ is the same, the system overhead is gradually decreasing for a larger *PA* since the less workers are allocated.

2) THE EFFECTS OF ϵ AND σ_2

Here, our LAMTA algorithm's data utility and system overhead are evaluated in Fig. 9. The data utility is larger when the ϵ is larger. For the same ϵ, as the σ_2 increases, the data utility is larger. The reason is that more number of workers are required to perform tasks for a larger σ_2. In Fig. 9, both the data utility and the system overhead are larger when ϵ and σ_2 increase. This is because the larger ϵ and σ_2 require allocating more workers.

3) THE EFFECTS OF ϵ AND k

In Fig. 10, as the k increases, the data utility and the system overhead of our LAMTA algorithm all increases. This is because more workers are allocated to perform a tasks for a larger k. Besides, when the k is the same, for a larger ϵ, the data utility is larger and the system overhead is smaller since the allocated results contain less noise and less workers are allocated. Particularly, when $\epsilon = 1$, $PA = 0.9$ and $k = 2$, the data utility achieves the maximum value and the system overhead achieves the smallest value, they are 91.816% and 19.997, respectively. These results show that our LAMTA algorithm can allocate high-quality tasks and has high efficiency.

VII. RELATED WORKS

At present, there are many privacy-preserving methods to research the worker location privacy in spatial crowdsourcing. In the following, these methods will be reviewed and they mainly contain three categories: anonymous methods, encryption methods and differential privacy methods.

A. ANONYMOUS AND ENCRYPTION METHODS

There are some state-of-the-art anonymous methods, which can ensure that the SC-server allocates tasks without knowing the identities of the workers. To protect the location privacy, Alharthi *et al.* [23] proposed a DCentroid framework that adopts a dummy-based anonymous method to calculate dummy locations in SC. In [24], Shu *et al.* proposed a single-keyword task allocation method to protect the location privacy and identity privacy based on anonymous method. The work in [25] designed a ZebraLancer system to implement the data and anonymous identity protection in decentralized SC services. However, although these anonymous can provide the worker privacy protection, the anonymous method may be vulnerable caused by the strong background knowledge attacks and the reference attacks [27], [28].

Another traditional privacy-preserving method known as encryption methods are used to ensure the worker location privacy in SC services. Zhang *et al.* [14] proposed an additive homomorphic encryption method that can protect the both tasks and workers' location privacy in vehicle-based SC applications. Yuan *et al.* [20] proposed a grid-based attribute encryption method to achieve the privacy-preserving location privacy in SC without depending on a TTP. In [21], Wu *et al.* introduced a fog-assisted SC framework to allocate the tasks based on homomorphic encryption method. In [22], Xiao *et al.* protected the location privacy and allocated the workers with winning bids based on a secure reverse auction protocol. However, these homomorphic encryption methods do not have high efficiency [26]. Besides, the users who are trusted but have no keys cannot access to the data encrypted by the encryption methods, thus the data availability is limited by the encryption methods [32].

B. DIFFERENTIAL PRIVACY METHODS

Different from the traditional anonymous methods and homomorphic encryption methods, differential privacy (DP) methods [27], [29]–[32] become popular and have been used to ensure the location privacy of workers. For example, To *et al* [33], [34] proposed a PSD method to protect the location privacy of workers in SC. In [35], Gong *et al.* adopted a cellular service provider (CSP) to protect the location and reputation privacy of workers by using a reputation-based DP method. Wang *et al.* [36] published the noisy crowdsourced data based on an agent-based DP framework. However, these DP methods protect the worker location privacy by relying on a TTP. The worker location privacy may be leaked once the TTP is attacked.

To remove the TTP, the works in [37]–[39], [41] adopted LDP methods to allow the users to protect their own data privacy without relying on a TTP. In [40], Wang *et al.* introduced a distortion geo-obfuscation method to ensure the task allocation privacy. The work in [42] protected the location privacy in vehicle-based SC system based on a geo-indistinguishability method. To *et al.* [43] proposed a LDP framework to protect the worker location privacy based on geo-indistinguishability. However, these LDP methods only protect the worker location privacy in static case without considering dynamic location privacy. In practice, the dynamic worker locations are outsourced to the SC server [43], [44], and public dynamic worker locations can increase the privacy disclosure risk [34], [43]. Therefore, in our paper, we propose a LDPDW scheme to locally protect the dynamic worker location privacy and achieve high-quality task allocation.

VIII. CONCLUSION

In this paper, we explore using LDP method to solve the problems of locally privacy-preserving dynamic worker locations in SC. A LDPDW scheme is proposed to achieve the dynamic worker location locally and the high-quality task allocation. In our LDPDW scheme, the high correlated locations are added noise based on a LDPCG algorithm, which protects the location correlation privacy. Then a DSLDP algorithm is proposed to obfuscate the worker locations. To achieve the tasks with high acceptance rates, we use a linear acceptance model (LAM) to allocate the tasks to the nearby workers, which reduces the system overhead. Furthermore, we propose a DCGLO algorithm to perform the dynamic worker location obfuscation, which ensures the correlation privacy and the worker location privacy. Privacy analysis and extensive experiments confirm that our LDPDW scheme achieves ϵ-LDP and has high data utility.

APPENDIX A

Theorem 1. *For each original class $C_l = \{L_{W_it_{l_1}}, L_{W_it_{l_2}}, \cdots, L_{W_it_{l_v}}\}$, when the correlation score between the locations $L_{W_it_{l_j}}$ and $L_{W_it_{l_k}}$ is S_{jk}, thus the query sensitivity $\Delta S(C_l) = \max_{L_{W_it_{l_j}}, L_{W_it_{l_k}} \in C_l} (S_{jk}) \leq 1$.*

Proof: Given the original class $C_l = \{L_{W_it_{l_1}}, L_{W_it_{l_2}}, \cdots, L_{W_it_{l_v}}\}$, we can obtain its neighboring class $C_l^* = \{L^*_{W_it_{l_1}}, L^*_{W_it_{l_2}}, \cdots, L^*_{W_it_{l_v}}\}$ when any correlation edge S_{jk} ($j, k \in [1, v]$) in C_l is deleted. Therefore, when $r \neq j$ and $p \neq k$, the correlation score $S_{rp} = S^*_{rp}$. The correlation $S^*_{rp} = 0$ when $r = j$ and $p = k$. According to the Eq. (1), any correlation $S_{rp}, S^*_{rp} \in [0, 1]$. Based on the query sensitivity Definition 3 in Section III, thus the correlation sensitivity $\Delta S(C_l)$ is calculated as:

$$\begin{aligned}
\Delta S(C_l) &= \max_{C_l, C_l^*}(\|S(C_l) - S(C_l^*)\|_1) \\
&= \max_{C_l, C_l^*}(\sum_{r,p}^{v} \|S_{rp} - S^*_{rp})\|_1) \\
&= \max_{L_{W_it_{l_j}}, L_{W_it_{l_k}} \in C_l, L^*_{W_it_{l_j}}, L^*_{W_it_{l_k}} \in C_l^*} (|S_{jk} - 0|) \\
&= \max_{L_{W_it_{l_j}}, L_{W_it_{l_k}} \in C_l} (S_{jk}) \\
&\leq 1
\end{aligned}$$

■

Based on the nature of LDP, since our query sensitivity is smaller, which can add less noise to the correlations of dynamic locations. It means that our LDPCG algorithm ensures the data utility while protecting the correlation privacy.

APPENDIX B

Theorem 2. *Given an original location dataset $L^c_{W_it} = (L^c_{W_it_1}, \cdots, L^c_{W_it_m})$, for two neighbor locations $L^c_{W_it_j}$ and $L^c_{W_it_{j+1}}$, their distance score is DS_{t_j}. Thus the query sensitivity $\Delta DS(L^c_{W_it}) = \max_{L^c_{W_it_j}, L^c_{W_it_{j+1}} \in L^c_{W_it}} (DS_{t_j} + DS_{t_{j+1}}) \leq 2$.*

Proof: For the original dataset $L^c_{W_it} = (L^c_{W_it_1}, \cdots, L^c_{W_it_m})$, based on the LDP, the neighboring dataset $L^{c*}_{W_it} = (L^{c*}_{W_it_1}, \cdots, L^{c*}_{W_it_m})$ is obtained by arbitrarily deleting a location $L^c_{W_it_j}$ in $L^c_{W_it}$. According to the distance score definition, the $DS_{t_k} \leq 1$ ($k \in [1, m]$). When $k = j = 1$, it means that the 1-th location $L^c_{W_it_1}$ is deleted, thus $DS^*_{t_1} = 0$ and $DS_{t_k} = DS^*_{t_k}$ ($k \in [2, m-1]$). According to the query sensitivity definition 3 in Section III, the $\Delta DS(L^c_{W_it}) = \max_{L^c_{W_it}, L^{c*}_{W_it}} |DS(L^c_{W_it}) - DS(L^{c*}_{W_it})|_1 = DS_{t_1}$. Similarly, when the last location $L^c_{W_it_m}$ is deleted, thus $DS^*_{t_{m-1}} = 0$ and $DS_{t_k} = DS^*_{t_k}$ ($k \in [1, m-2]$). The query sensitivity $\Delta DS(L^c_{W_it}) = \max_{L^c_{W_it}, L^{c*}_{W_it}} |DS(L^c_{W_it}) - DS(L^{c*}_{W_it})|_1 = DS_{t_{m-1}}$. When we delete the location $L^c_{W_it_j}$ from the second location to the $(m-1)$-th location, and if $j = k$, $DS^*_{t_j} = 0$ and $DS^*_{t_{j+1}} = 0$. Otherwise, $DS_{t_k} = DS^*_{t_k}$ ($k \in [2, m-2]$). Based on the query sensitivity Definition 3 in Section III, thus the distance score sensitivity $\Delta DS(L^c_{W_it})$ is calculated as:

$$\begin{aligned}
\Delta DS(L^c_{W_it}) &= \max_{L^c_{W_it}, L^{c*}_{W_it}} (\|DS(L^c_{W_it}) - DS(L^{c*}_{W_it})\|_1) \\
&= \max_{L^c_{W_it}, L^{c*}_{W_it}} (\sum_{k=2}^{m-2} \|DS_{t_k} - DS^*_{t_k}\|_1) \\
&= \max_{L^c_{W_it_j}, L^c_{W_it_{j+1}} \in L^c_{W_it}} |DS_{t_j} + DS_{t_{j+1}} - 0| \\
&= \max_{L^c_{W_it_j}, L^c_{W_it_{j+1}} \in L^c_{W_it}} (DS_{t_j} + DS_{t_{j+1}}) \\
&\leq 2
\end{aligned}$$

Since $DS_{t_1} < \max(DS_{t_j} + DS_{t_{j+1}})$ and $DS_{t_{m-1}} < \max(DS_{t_j} + DS_{t_{j+1}})$, the query sensitivity $\Delta DS(L^c_{W_it}) = \max_{L^c_{W_it_j}, L^c_{W_it_{j+1}} \in L^c_{W_it}} (DS_{t_j} + DS_{t_{j+1}}) \leq 2$. ■

APPENDIX C PRIVACY PROOF

Theorem 3. *Our LDPCG algorithm follows ϵ_1-LDP.*

Proof: As described in Section IV-A, LDPCG generates high correlated graph classes $\{C_1, C_2, \cdots, C_q\}$ and low correlated locations $L_{low}(W_i)$. To protect the correlation privacy, each class C_l ($l \in [1, q]$) is added noise and the low correlated locations $L_{low}(W_i)$ do not add noise. Based on the LDP, for each C_l, we delete any correlation edge in C_l to obtain a neighboring graph $C^*_l = \{L^*_{W_{il_1}}, L^*_{W_{il_2}}, \cdots, L^*_{W_{il_k}}\}$. To protect the privacy, the class C_l is added $U_l \times Lap(\frac{\Delta S(C_l)}{\epsilon_1})$ Laplace noise based on our LDPCG algorithm, where U_j satisfies the Bernoulli distribution so $Pr(U_{l_j}) = Pr(U^*_{l_j}) = \frac{1}{2}$. According to the Theorem 1 in Section IV-A, the query sensitivity $\Delta S(C_l) = \max_{L_{W_it_{l_j}}, L_{W_it_{l_k}} \in C_l} (S_{jk}) \leq 1$. Since $LDPCG(C_l) = L^c_{W_{il}} = \{L^c_{W_{il_1}}, L^c_{W_{il_2}}, \cdots, L^c_{W_{il_k}}\}$, we have:

$$
\begin{aligned}
\frac{Pr[LDPCG(C_l) \in L^c_{W_{il}}]}{Pr[LDPCG(C^*_l)) \in L^{c*}_{W_{il}}]} &= \frac{Pr[LDPCG(L_{W_{il_j}})] \times Pr[U_j]}{Pr[LDPCG(L^*_{W_{il_j}})] \times Pr[U^*_j]} \\
&= \frac{\prod_{p,j=1}^{k} \frac{\epsilon_1}{2\Delta S} e^{(-\frac{\epsilon_1}{\Delta S}|L^c_{W_{il_j}} - S_{jp}|)}}{\prod_{p,j=1}^{k} \frac{\epsilon_1}{2\Delta S} e^{(-\frac{\epsilon_1}{\Delta S}|L^c_{W_{il_j}} - S^*_{jp}|)}} \\
&= \prod_{p,j=1}^{k} \exp(\frac{\epsilon_1}{\Delta S}(|L^c_{W_{il_j}} - S^*_{jp}| \\
&\quad -|L^c_{W_{il_j}} - S_{jp}|)) \\
&\leq e^{(\frac{\epsilon_1 \sum_{p,j=1}^{k}(|S_{jp} - S^*_{jp})|)}{\Delta S(C_l)})} \\
&\leq \exp(\frac{\epsilon_1 \Delta S(C_l)}{\Delta S(C_l)}) = e^{\epsilon_1}
\end{aligned}
$$

Based on the LDP Definition 1 in Section III, the algorithm $LDPCG(C_l)$ follows ϵ_1-LDP. Similarly, the algorithms $\{LDPCG(C_1), \cdots, LDPCG(C_k)\}$ all achieves ϵ_1-LDP. Besides, since low correlated locations $L_{low}(W_i)$ are not added noise, $LDPCG(L_{low}(W_i))$ achieves 0-LDP. Based on the parallel combination principle of LDP in [29], our LDPCG algorithm follows ϵ_1-LDP since $\max(\epsilon_1, 0) = \epsilon_1$. ■

Theorem 4. *Our DSLDP algorithm achieves ϵ_2-LDP.*

Proof: For the dataset $L^c_{W_{it}} = \{L^c_{W_{it_1}}, L^c_{W_{it_2}}, \cdots, L^c_{W_{it_m}}\}$, the noisy locations $DSLDP(L^c_{W_{it}}) = \widetilde{L}_{W_{it}} = \{\widetilde{L}_{W_{it_1}}, \cdots, \widetilde{L}_{W_{it_m}}\}$ are obtained in Section IV-B, in which each noisy location $\widetilde{L}_{W_{it_j}} = L^c_{W_{it_j}} + Lap(\frac{\Delta DS(L^c_{W_it})}{\epsilon_2}) \times \Upsilon_j$, where query sensitivity $\Delta DS(L^c_{W_it}) = \max_{L^c_{W_it_j}, L^c_{W_it_{j+1}} \in L^c_{W_it}} (DS_{t_j} + DS_{t_{j+1}})$ and Υ_j follows the follows Bernoulli distribution. Similar to the proof of Theorem 3, the neighboring dataset $L^{c*}_{W_{it}} = \{L^{c*}_{W_{it_1}}, L^{c*}_{W_{it_2}}, \cdots, L^{c*}_{W_{it_m}}\}$ and $Pr[\Upsilon_t] = Pr[\Upsilon^*_t]$. We prove:

$$
\begin{aligned}
&\frac{Pr[DSLDP(L^c_{W_{it}}) \in \widetilde{L}_{W_{it}}]}{Pr[DSLDP(L^{c*}_{W_{it}})) \in \widetilde{L}_{W_{it}}]} \\
&\quad = \frac{Pr[DSLDP(L^c_{W_{it}})] \times Pr[\Upsilon_t]}{Pr[DSLDP(L^{c*}_{W_{it}})] \times Pr[\Upsilon^*_t]} \\
&\quad = \frac{\prod_{j=1}^{m-1} e^{(-\frac{\epsilon_2}{\Delta DS}|\widetilde{L}_{W_{it_j}} - (DS_{t_j} + DS_{t_{j+1}})|)}}{\prod_{j=1}^{m-1} e^{(-\frac{\epsilon_2}{\Delta DS}|\widetilde{L}_{W_{it_j}} - (DS^*_{t_j} + DS^*_{t_{j+1}})|)}} \\
&\quad = \prod_{j=1}^{m-1} e^{(\frac{\epsilon_2}{\Delta DS}(|\widetilde{L}_{W_{it_j}} - (DS_{t_j} + DS_{t_{j+1}})| - |\widetilde{L}_{W_{it_j}} - (DS^*_{t_j} + DS^*_{t_{j+1}})|))} \\
&\quad \leq e^{(\frac{\epsilon_2 \sum limits_{j=1}^{m-1}(|DS_{t_j} + DS_{t_{j+1}} - DS^*_{t_j} - DS^*_{t_{j+1}}|)}{\Delta DS(L^c_{W_it})})} \\
&\quad \leq \exp(\frac{\epsilon_2 \Delta DS(L^c_{W_it})}{\Delta DS(L^c_{W_it})}) = e^{\epsilon_2}
\end{aligned}
$$

Therefore, our DWLDP algorithm achieves ϵ_2-LDP based on the LDP in [29]. ■

REFERENCES

[1] W. Ni, P. Cheng, L. Chen, and X. Lin, "Task allocation in dependency-aware spatial crowdsourcing," in *Proc. IEEE 36th Int. Conf. Data Eng. (ICDE)*, Dallas, TX, USA, Apr. 2020, pp. 985–996, doi: 10.1109/ICDE48307.2020.00090.

[2] Y. Wang, C. Zhao, and S. Xu, "Method for spatial crowdsourcing task assignment based on integrating of genetic algorithm and ant colony optimization," *IEEE Access*, vol. 8, pp. 68311–68319, Apr. 2020.

[3] B. Guo, Y. Liu, L. Wang, V. O. K. Li, J. C. K. Lam, and Z. Yu, "Task allocation in spatial crowdsourcing: Current state and future directions," *IEEE Internet Things J.*, vol. 5, no. 3, pp. 1749–1764, Jun. 2018, doi: 10.1109/JIOT.2018.2815982.

[4] M. Kadadha, R. Mizouni, S. Singh, H. Otrok, and A. Ouali, "ABCrowd: An auction mechanism on blockchain for spatial crowdsourcing," *IEEE Access*, vol. 8, pp. 12745–12755, Jan. 2020.

[5] Y. Zhao and Q. Han, "Spatial crowdsourcing: Current state and future directions," *IEEE Commun. Mag.*, vol. 54, no. 7, pp. 102–107, Jul. 2016, doi: 10.1109/MCOM.2016.7509386.

[6] N. Wang and J. Wu, "Cost-efficient heterogeneous worker recruitment under coverage requirement spatial crowdsourcing," *IEEE Trans. Big Data*, early acess, Aug. 17, 2018, doi: 10.1109/TBDATA.2018.2865755.

[7] (Nov. 2020). *DiDi*. [Online]. Available: http://www.didiglobal.com/#/

[8] (Nov. 2020). *Waze*. [Online]. Available:https://www.waze.com

[9] (Nov. 2020). *OpenStreetMap*. [Online]. Available: https://www.openstreetmap.org

[10] (Nov. 2020). *Uber*. [Online]. Available: https://en.wikipedia.org/wiki/Uber

[11] E. Mazareanu. (Oct. 2020). *Number Rides Uber Gave Worldwide From Q2 2017 to Q2 2020*. [Online]. Available: https://www.statista.com/statistics/946298/uber-ridership-worldwide/

[12] L. Yu, L. Liu, and C. Pu, "Dynamic differential location privacy with personalized error bounds," in *Proc. Netw. Distrib. Syst. Secur. Symp.*, San Diego, CA, USA, 2017, pp. 1–15.

[13] G. Cormode, C. Procopiuc, D. Srivastava, E. Shen, and T. Yu, "Differentially private spatial decompositions," in *Proc. IEEE 28th Int. Conf. Data Eng.*, Washington, DC, USA, Apr. 2012, pp. 20–31, doi: 10.1109/ICDE.2012.16.

[14] J. Zhang, F. Yang, Z. Ma, Z. Wang, X. Liu, and J. Ma, "A decentralized location privacy-preserving spatial crowdsourcing for Internet of vehicles," *IEEE Trans. Intell. Transp. Syst.*, early access, Aug. 4, 2020, doi: 10.1109/TITS.2020.3010288.

[15] X. Chu, J. Liu, D. Gong, and R. Wang, "Preserving location privacy in spatial crowdsourcing under quality control," *IEEE Access*, vol. 7, pp. 155851–155859, Oct. 2019, doi: 10.1109/ACCESS.2019.2949409.

[16] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K.-L. Tan, "Private queries in location based services: Anonymizers are not necessary," in *Proc. ACM SIGMOD Int. Conf. Manage. Data (SIGMOD)*, Vancouver, BC, Canada, 2008, pp. 121–132.

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

[18] G. Wang, B. Wang, T. Wang, A. Nika, H. Zheng, and B. Y. Zhao, "Defending against Sybil devices in crowdsourced mapping services," in *Proc. 14th Annu. Int. Conf. Mobile Syst., Appl., Services*, Singapore, May 2016, pp. 179–191.

[19] S. Han, J. Lin, S. Zhao, G. Xu, S. Ren, D. He, L. Wang, and L. Shi, "Location privacy-preserving distance computation for spatial crowdsourcing," *IEEE Internet Things J.*, vol. 7, no. 8, pp. 7550–7563, Aug. 2020, doi: 10.1109/JIOT.2020.2985454.

[20] D. Yuan, Q. Li, G. Li, Q. Wang, and K. Ren, "PriRadar: A privacy-preserving framework for spatial crowdsourcing," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 299–314, 2020, doi: 10.1109/TIFS.2019.2913232.

[21] H. Wu, L. Wang, and G. Xue, "Privacy-aware task allocation and data aggregation in fog-assisted spatial crowdsourcing," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 589–602, Jan. 2020, doi: 10.1109/TNSE.2019.2892583.

[22] M. Xiao, K. Ma, A. Liu, H. Zhao, Z. Li, K. Zheng, and X. Zhou, "SRA: Secure reverse auction for task assignment in spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 782–796, Apr. 2020, doi: 10.1109/TKDE.2019.2893240.

[23] R. S. Alharthi, E. Aloufi, I. Alrashdi, A. Alqazzaz, M. A. Zohdy, and J. L. Rrushi, "Protecting location privacy for crowd workers in spatial crowdsourcing using a novel dummy-based mechanism," *IEEE Access*, vol. 8, pp. 114608–114622, 2020, doi: 10.1109/ACCESS.2020.3004470.

[24] J. Shu, X. Liu, X. Jia, K. Yang, and R. H. Deng, "Anonymous privacy-preserving task matching in crowdsourcing," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 3068–3078, Aug. 2018, doi: 10.1109/JIOT.2018.2830784.

[25] Y. Lu, Q. Tang, and G. Wang, "ZebraLancer: Private and anonymous crowdsourcing system atop open blockchain," in *Proc. IEEE 38th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Vienna, Austria, Jul. 2018, pp. 853–865, doi: 10.1109/ICDCS.2018.00087.

[26] 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. 26th Int. Conf. World Wide Web*, Perth, WA, Australia, Apr. 2017, pp. 627–636.

[27] K. Han, H. Liu, S. Tang, M. Xiao, and J. Luo, "Differentially private mechanisms for budget limited mobile crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 18, no. 4, pp. 934–946, Apr. 2019, doi: 10.1109/TMC.2018.2848265.

[28] C. Clifton and T. Tassa, "On syntactic anonymity and differential privacy," in *Proc. IEEE 29th Int. Conf. Data Eng. Workshops (ICDEW)*, Brisbane, QLD, Australia, Apr. 2013, pp. 88–93, doi: 10.1109/ICDEW.2013.6547433.

[29] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, Xi'an, China, 2008, pp. 1–19.

[30] Z. Xu, S. Shi, A. X. Liu, J. Zhao, and L. Chen, "An adaptive and fast convergent approach to differentially private deep learning," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Toronto, ON, Canada, Jul. 2020, pp. 1867–1876, doi: 10.1109/INFOCOM41043.2020.9155359.

[31] Z. Chi, Y. Wang, Y. Huang, and X. Tong, "The novel location privacy-preserving CKD for mobile crowdsourcing systems," *IEEE Access*, vol. 6, pp. 5678–5687, 2018, doi: 10.1109/ACCESS.2017.2783322.

[32] J. Wei, Y. Lin, X. Yao, and J. Zhang, "Differential privacy-based location protection in spatial crowdsourcing," *IEEE Trans. Services Comput.*, early access, Jun. 6, 2019, doi: 10.1109/TSC.2019.2920643.

[33] H. To, G. Ghinita, and C. Shahabi, "PrivGeoCrowd: A toolbox for studying private spatial crowdsourcing," in *Proc. IEEE 31st Int. Conf. Data Eng.*, Seoul, South Korea, Apr. 2015, pp. 1404–1407, doi: 10.1109/ICDE.2015.7113387.

[34] 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, doi: 10.1109/TMC.2016.2586058.

[35] 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. 2018.

[36] Z. Wang, X. Pang, Y. Chen, H. Shao, Q. Wang, L. Wu, H. Chen, and H. Qi, "Privacy-preserving crowd-sourced statistical data publishing with an untrusted server," *IEEE Trans. Mobile Comput.*, vol. 18, no. 6, pp. 1356–1367, Jun. 2019, doi: 10.1109/TMC.2018.2861765.

[37] X. Ren, C.-M. Yu, W. Yu, S. Yang, X. Yang, J. A. McCann, and P. S. Yu, "LoPub : High-dimensional crowdsourced data publication with local differential privacy," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 9, pp. 2151–2166, Sep. 2018, doi: 10.1109/TIFS.2018.2812146.

[38] W. Li, C. Zhang, and Y. Tanaka, "Privacy-aware sensing-quality-based budget feasible incentive mechanism for crowdsourcing fingerprint collection," *IEEE Access*, vol. 8, pp. 49775–49784, Mar. 2020, doi: 10.1109/ACCESS.2020.2974909.

[39] M. Andres, N. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM SIGSAC Conf. Comput. Commun.*, Berlin, Germany, Nov. 2013, pp. 901–914.

[40] L. Wang, D. Yang, X. Han, D. Zhang, and X. Ma, "Mobile crowdsourcing task allocation with differential- and-distortion geo-obfuscation," *IEEE Trans. Dependable Secure Comput.*, early access, Apr. 23, 2019, doi: 10.1109/TDSC.2019.2912886.

[41] J. Wei, J. Li, Y. Lin, and J. Zhang, "LDP-based social content protection for trending topic recommendation," *IEEE Internet Things J.*, early access, Sep. 24, 2020, doi: 10.1109/JIOT.2020.3026366.

[42] C. Qiu and A. C. Squicciarini, "Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability," in *Proc. IEEE 39th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Dallas, TX, USA, Jul. 2019, pp. 1061–1071, doi: 10.1109/ICDCS.2019.00109.

[43] H. To, C. Shahabi, and L. Xiong, "Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server," in *Proc. IEEE 34th Int. Conf. Data Eng. (ICDE)*, Paris, France, Apr. 2018, pp. 833–844, doi: 10.1109/ICDE.2018.00080.

[44] W. Liu, Y. Yang, E. Wang, and J. Wu, "Dynamic user recruitment with truthful pricing for mobile CrowdSensing," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Toronto, ON, Canada, Jul. 2020, pp. 1113–1122, doi: 10.1109/INFOCOM41043.2020.9155242.

[45] L. Ou, Z. Qin, S. Liao, Y. Hong, and X. Jia, "Releasing correlated trajectories: Towards high utility and optimal differential privacy," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 5, pp. 1109–1123, Sep. 2020, doi: 10.1109/TDSC.2018.2853105.

[46] A. Alrawais, A. Alhothaily, X. Cheng, C. Hu, and J. Yu, "SecureGuard: A certificate validation system in public key infrastructure," *IEEE Trans. Veh. Technol.*, vol. 67, no. 6, pp. 5399–5408, Jun. 2018.

[47] J. Zhang, J. Sun, R. Zhang, Y. Zhang, and X. Hu, "Privacy-preserving social media data outsourcing," in *Proc. IEEE INFOCOM-IEEE Conf. Comput. Commun.*, Honolulu, HI, USA, Apr. 2018, pp. 1106–1114.

[48] (Jan. 2014). *Gowalla*. [Online]. Available: https://snap.stanford.edu/data/loc-gowalla.html

FENG LIN received the master's degree from Hunan University, China, in 2007. He is currently a Lecturer with Shaoyang University, China. His research interests include cloud computing and information security.

JIANHAO WEI (Student Member, IEEE) received the B.S. degree in information and computing science from Shangqiu Normal University, China, in 2014, and the M.S. degree in computer application from Hunan Normal University, China, in 2017. He is currently pursuing the Ph.D. degree with the College of Computer Science and Electronic Engineering, Hunan University. His research interests include privacy-preserving in social networks, privacy issues in spatial crowdsourcing, cloud computing, and social big data analysis.

JUNYI LI (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in computer application from Hunan University, in 1993, 2001, and 2008, respectively. Since 2005, he has been an Associate Professor with Hunan University. From 2009 to 2010, he was a Visiting Researcher with Lakehead University, Canada. His research interests include big data analysis, software engineering, and privacy protection.

JIANMING ZHANG (Member, IEEE) received the B.S. degree from Zhejiang University, in 1996, the M.S. degree from the National University of Defense Technology, China, in 2001, and the Ph.D. degree from Hunan University, China, in 2010. He is currently a Full Professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology, China. He has published more than 110 research articles. His research interests include computer vision, pattern recognition, the Internet of Things, and mobile computing. He is a Senior Member of CCF.

BO YIN (Member, IEEE) received the B.S. degree in communication engineering, the M.S. degree in communication and information system, and the Ph.D. degree in computer application technology from Hunan University, Changsha, China, in 2005, 2008, and 2013, respectively. She is currently an Associate Professor with the School of Computer and Communication Engineering, Changsha University of Science and Technology, Changsha. Her current research interests include data management and analysis, parallel and distributed computing, and blockchain systems.