Hindawi
Security and Communication Networks
Volume 2021, Article ID 5855745, 18 pages
https://doi.org/10.1155/2021/5855745

Research Article

From Centralized Protection to Distributed Edge Collaboration: A Location Difference-Based Privacy-Preserving Framework for Mobile Crowdsensing

Zihao Shao, Huiqiang Wang, Yifan Zou, Zihan Gao, and Hongwu Lv

College of Computer Science and Technology, Harbin Engineering University, Harbin 150001, China

Correspondence should be addressed to Huiqiang Wang; wanghuiqiang@hrbeu.edu.cn

Received 26 May 2021; Revised 1 September 2021; Accepted 7 September 2021; Published 21 September 2021

Academic Editor: Stelvio Cimato

Copyright © 2021 Zihao Shao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile Crowdsensing (MCS) has evolved into an effective and valuable paradigm to engage mobile users to sense and collect urban-scale information. However, users risk their location privacy while reporting data with actual sensing locations. Existing works of location privacy-preserving are primarily based on single-region location information, which rely on a trusted and centralized sensing platform and ignore the impact of regional differences on user privacy-preserving demands. To tackle this issue, we propose a Location Difference-Based Privacy-Preserving Framework (LDPF), leveraging the powerful edge servers deployed between users and the sensing platform to hide and manage users according to regional user characteristics. More specifically, for popular regions, based on the edge servers and the k-anonymity algorithm, we propose a Coordinate Transformation and Bit Commitment (CTBC) privacy-preserving method that effectively guarantees the privacy of location data without relying on a trusted sensing platform. For remote regions, based on a more realistic distance calculation mode, we design a Paillier Encryption Data Coding (PDC) privacy-preserving method that realizes the secure computation for users' location and prevents malicious users from deceiving. The theoretical analysis and simulation results demonstrate the security and efficiency of the proposed framework in location difference-based privacy-preserving.

1. Introduction

Nowadays, the ubiquity of mobile devices equipped with various functional built-in sensors (e.g., camera, microphone, and GPS) and increasingly powerful wireless and 5G network has enabled the prosperity of MCS [1] such as traffic monitoring [2] and point-of-interest characterization [3]. Besides, many commercial MCS platforms have been developed like Gigwalk [4] and Streetspottr [5].

A typical MCS includes a centralized platform at the cloud layer responsible for publishing sensing tasks, collecting user data, and providing match and select services. However, as the performance of intelligent terminals and the complexity of sensing tasks are continuously growing, the platform will have an increasing number on available sensing users, which will inevitably overload the MCS sensing platform. Although the pervasive deployment of 5G substantially improves the responsiveness of sensing services, the centralized MCS sensing platform is not able to meet the requirements of security and efficient processing of raw data. On the other hand, considering centralized sensing platforms are honest-but-curious entities, a trusted platform is challenging to achieve in the real world. It may lead to serious privacy threats and further discourage people from sharing their data. Besides, due to the complexity of the MCS sensing environment, the difference of location privacy-preserving requirements between geo-distributed and user scale becomes a severe research challenge [6].

As an alternative approach, edge computing possesses the advantages of near-zero latency, low network load, and superior flexibility and enables a distributed way of preserving user privacy. The principle of edge computing is to process the uploaded data by users in their close proximity, where only processing results are sent to the cloud [7].

Therefore, edge computing can be employed to realize the various data collected by participants of MCS, dramatically enhancing its data processing efficiency. An edge computing layer consists of edge nodes that have access to storage and computing resources. These nodes are responsible for processing data uploaded by users through mobile devices. Besides, another advantage of deploying an edge computing layer is the reduced privacy risk because these nodes can collaborate to anonymize the local data submissions without relying on a trusted and centralized sensing platform [8].

From the perspective of geo-distributed, various tasks published by the MCS platform are different, in which user location privacy-preserving should be adequately matched with regional characteristics. Particularly, popular regions are usually characterized by either many users or high data redundancy. Therefore, it is necessary to develop a location privacy-preserving method with high efficiency and low complexity. In contrast, due to the small number of participants and strong privacy awareness, a high-security location privacy-preserving method is often needed for remote regions. Unfortunately, most of the existing location privacy-preserving solutions were designed for single-region data, and the impact of regional differences was ignored on user privacy-preserving demands. Additionally, when calculating the user's movement distance, the calculation method based on Euclidean distance may produce errors if any obstacles (e.g., buildings, trees, and other shelters) block users.

In light of the above research challenges, we propose LDPF, a Location Difference-Based Privacy-Preserving Framework for MCS. Firstly, sensing regions are divided into popular regions and remote regions. Next, for popular regions, the edge layer collaborates to change location information and continuously protect participant location through CTBC without relying on a trusted sensing platform. Finally, for remote regions, the edge layer collaborates with the sensing platform. PDC is adopted to realize the secure computation for Manhattan distance and prevent malicious users from deceiving. The main contributions of this paper are as follows:

(i) We present a Location Difference-Based Privacy-Preserving Framework (LDPF) based on the powerful edge servers to solve centralization and situation of no regional differences in user location privacy-preserving.

(ii) We propose a Coordinate Transformation and Bit Commitment (CTBC) privacy-preserving method based on the k-anonymity algorithm that can effectively guarantee location data privacy without relying on a trusted sensing platform.

(iii) We design a Paillier Encryption Data Coding (PDC) privacy-preserving method to realize moving distance calculation without exposing users' actual location and preventing malicious users from deceiving. In addition, we adopt a more realistic distance calculation mode (i.e., Manhattan distance) to overcome the error caused by obstacles (e.g., buildings, trees, and other shelters).

The rest of this paper is organized as follows. Related work is summarized in Section 2. In Section 3, the problem formulation for location difference-based privacy-preserving is presented. Solutions for popular and remote regions are presented in Section 4. In Section 5, security analysis and experimental results are discussed in detail. Conclusions are drawn in Section 6.

2. Related Work

In a broad sense, our work is under the umbrella of research on location privacy-preserving. Roughly speaking, this line of work shares the common goal of selecting an appropriate result from the data uploaded from a set of users without revealing the individual user's location. As shown in Table 1, the solutions can be mainly divided into two privacy-preserving approaches: data-oriented and edge-assisted.

Data-oriented protection includes anonymization, obfuscation, and encryption. Anonymization has been extensively studied since the introduction of MCS. The main idea of anonymization is to hide users' exact location in a hidden region, confusing adversaries [9]. The k-anonymity mechanism is the most common method for centralized anonymization-based location privacy protection. Gruteser and Grunwald [10] first introduced k-anonymity into privacy-preserving, which aims to put a user and at least $k-1$ other users together constitute an anonymous region, so that the probability of the user's real identity being recognized is no more than $1/k$. Chi et al. [11] proposed a location privacy-preserving mechanism for a mobile crowdsensing system, which combined k-anonymity and differential privacy protection technology. For the distributed anonymization, spatial domain decomposition technology has gained extensive attention. Habeeb et al. [12] applied the Voronoi diagram to privacy-preserving kNN spatial query, balancing data confidentiality and integrity. Jadallah and Al Aghbari [13] designed an Aman algorithm to protect user privacy with the least number of communication rounds between the user and the server. However, the quadtree-based anonymous technology also disadvantages a single partition mode and unbalanced privacy protection. Although it is useful, most current works focus on anonymization mechanisms that treat sensing platforms without considering the reliability.

Obfuscation tends to modify the original location of users independently, without mixing with other users' locations. The core of obfuscation is to generate false positions. False data replacement and data denoising are the most common methods. Zhang et al. [14] published privacy-preserving data aggregation for mobile crowdsensing in an auction framework and designed a data aggregation that allows each worker to report noisy data, which can guarantee using of each worker's data in a differentially private manner. Wei et al. [15] proposed a differential privacy-based location protection scheme, which protects both the users' and tasks' location privacy, and it has high data utility. Wang et al. [16] proposed a location obfuscation mechanism to reduce the data quality loss incurred by location obfuscation.

Table 1: Classification of privacy-preserving approaches.

Classification	Approach	Reference
Data-oriented protection	Anonymization	[9–13]
	Obfuscation	[14–19]
	Encryption	[20–25]
Edge-assisted protection	Cloud-edge-user	[8, 26–29]

However, the original Laplacian noise used in the proposed solutions is unbounded, which affects the data utility. For truth discovery in crowdsourced binary-choice question answering systems, Sun et al. [17] defined a ε-local differential privacy-preserving algorithm, which could provide personalized payments for workers with different privacy preferences, achieving accurate truth discovery. Jin et al. [18] proposed an MCS system framework that integrates an incentive, a data aggregation, and a data perturbation mechanism. Its data perturbation reduced workers' privacy leakage to a reasonable degree by adding controlled random noises to the original aggregated results that compensates their costs. In addition, it has been found in this study that Geohash coding technology [19] can encode geographic coordinates, which means it has the advantages of fast retrieving neighbors and low computational overhead. Therefore, employing coding technology to solve user location privacy-preserving is worthy of attention.

Anonymization and obfuscation are achieved by sacrificing the accuracy of the location. In contrast, the location information was protected by using encryption cryptographic methods. For example, Shu et al. [20] proposed an encryption scheme to protect the location privacy of both tasks and users. However, these methods only allow users with the key to obtain task data, which hinders data availability by credible but keyless users. Huang et al. [21] designed a comparable homomorphic encryption scheme based on Lagrange's interpolation theorem, enabling ciphertext comparison between multiple users. Zheng et al. [22] introduced a confidence-aware truth discovery method, where users send encrypted sensory data to the cloud and requesters are responsible for decrypting the data. Xiong et al. [23] provided an additively homomorphic encryption scheme to effectively protect the confidentiality, substitution, and real-time nature of uploaded data. Paillier encryption is the most common encryption method for remote regions. Li et al. [24] proposed a privacy-preserving multisubset data aggregation scheme in a smart grid based on the Paillier cryptosystem. To protect users' sensory data and avoid user participation in the iterative truth discovery procedure, Zhang et al. [25] proposed a privacy-preserving truth discovery scheme based on the Paillier encryption.

Besides, the emerging edge computing paradigm is adopted by researchers to enhance the performance of MCS. Zhou et al. [26] proposed a novel context-aware MCS task allocation framework suitable for edge computing scenarios. In the cloud layer, a contextual has been used online for the learning algorithm to manage the participants' reputations. In the edge layer, the task allocation strategy was optimized directly based on users' real-time information. To ensure the user reputation for edge computing-assisted MCS, Ma et al. [27] proposed a novel reputation value updating method based on the deviations of the encrypted sensing data from the final aggregating result. Considering the characteristics of user-generated content and heterogeneity of resources, an intelligent framework has been designed by Yang et al. [28], which is based on "cloud-user-edge" cooperation, further reducing the end-to-end service delay and network traffic load. However, the privacy concern in edge computing-assisted MCS is still in its infancy. Huo et al. [29] designed a fog computing architecture and proposed a real-time streaming data aggregation framework with adaptive ω-event differential privacy. Experimental results showed that this method can relieve the overhead of servers, improve communication efficiency, and protect data privacy. Wu et al. [8] proposed a privacy-preserving task assignment framework for MCS, leveraging the powerful edge servers deployed between users and the platform to cluster and manage users according to user attributes.

One line of the past literature [14, 17, 18], highly related to this study, investigates mobile crowdsensing that preserves workers' privacy and data aggregation. These prior works invariably protect workers' privacy in a centralized framework. In contrast, we construct a three-tier distributed framework, exploiting the advantageous processing capability of edge servers, which reduces the workload of the sensing platform. Furthermore, unlike this paper, the characteristics of regional differences have not been considered as much as this study in most of these works. That is, the state-of-the-art location privacy-preserving methods assume that the privacy-preserving requirements of users are constant, which cannot ensure satisfactory consequences for the protection of users' privacy.

3. Problem Formulation

In this section, assumptions, the system model, and the threat model are given.

3.1. Assumptions. Considering actual application scenarios in the MCS, we make the following hypotheses for facilitating the proposed framework analysis.

Hypothesis 1. Users and attackers are absolutely rational, where the former will not recklessly expose location data and the latter will not launch attacks with no profits.

Hypothesis 2. Communication in the edge layer is secure and is not vulnerable to being attacked.

Hypothesis 3. The data quality of users is negatively correlated to the location (i.e., the closer to the task center, the better data quality), which meets the consensus of existing location-based privacy-preserving methods.

Hypothesis 4. The platform/users are honest-but-curious [30]. The platform/users would honestly execute every operation in mobile crowdsensing but try to grasp private information (e.g., location information).

3.2. System Model. Based on the typical architecture of MCS, an edge layer is introduced into the MCS architecture as a bridge connecting the platform in the cloud layer and users in the terminal layer. Thus, the edge-assisted privacy-preserving framework in this paper consists of three layers: cloud layer (a distributed sensing platform), edge layer (parameter generator and certificate authority), and user layer (a set of I users, denoted as $I = \{i_1, i_2, \ldots, i_n\}$), as illustrated in Figure 1. Their main function can be described as follows.

3.2.1. Cloud Layer. The distributed sensing platform in the cloud layer summarizes the needs of service providers, including the transformed region-of-interest POI$\prime$, the center of the transformed region-of-interest $L_{p-\text{center}}{}'$. In addition, the sensing platform predefines the region classification of the user layer (i.e., the red regions are the popular regions and the black regions are the remote regions) and leverages the difference between $L_{p-\text{center}}{}'$ and their candidate users to select the optimal users.

3.2.2. Edge Layer. Since the scale of users in various regions will affect the performance of location privacy-preserving, the edge layer will assist the cloud layer and user layer to implement data encryption, verification, and management. Specifically, users and service providers implement the data commitment at the edge layer, ensuring the authenticity of data and results. The edge layer verifies the identity of users and service providers before notifying them.

3.2.3. User Layer. Users (denoted as $I = \{i_1, i_2, \ldots, i_n\}$) in the user layer are ordinary participants who use mobile sensing devices (such as intelligent terminal devices, wearable devices, and vehicle-mounted devices). They use wired/wireless networks to perform tasks and gain revenue.

Specifically, the workflow of the proposed LDPF is as follows.

Firstly, service providers and users send request parameters to the edge layer to hide location information. The edge layer will provide different privacy-preserving methods according to the predefined region classification (i.e., the red regions are the popular regions and the black regions are the remote regions).

Coordinate Transformation and Bit Commitment (CTBC) Privacy-Preserving Method. When a task is in a popular region, the parameter generator at the edge layer sends $\text{CCP}_t = (\theta_t, b_t)$ to both participants to realize location hiding. Then, the edge layer implements the bit commitment through the certification authority, which aims to ensure the authenticity of the data. Next, users upload the transformed data, and the sensing platform performs matching operations (see detailed discussion in Section 4.2).

Paillier Encryption Data Coding (PDC) Privacy-Preserving Method. When a task is in a remote region, the parameter generator at the edge layer sends hidden location coding to both participants to realize location hiding. Then, the edge layer employs a cheating-prevention protocol through the certification authority, which aims to calculate the Manhattan distance. Next, users upload the transformed data, and the sensing platform calculates the Manhattan distance between users and tasks (see detailed discussion in Section 4.3).

Finally, the sensing platform releases the matching result to the edge layer. The edge layer notifies service providers and the selected users to perform identity authentication.

3.3. Adversarial Model. There are two types of attackers in MCS [31]: (1) internal attackers (i.e., people who participate in MCS, such as users and service providers) and (2) external attackers (i.e., people who do not participate in MCS). Our adversarial model assumes that both users and the sensing platform are honest-but-curious entities who comply with the transaction rules yet may be curious about private information (e.g., location information). We use data anonymity and encryption to resist external attacks, while internal attacks and collusion (multiple participants) attacks are the core research of LDPF:

(1) *External Attacks.* External attackers eavesdrop on the communications between users and the platform to steal real-location data to impact the system availability. A malicious attacker chooses a historical record to attack and runs a data analysis program by which queries MCS to infer the participant sensitive information.

(2) *Internal Attacks.* Internal attackers may forge their identities or submit low-quality data to reduce the efficiency of MCS. Specifically, during the data collection process, malicious users can submit authenticated but faulty reports to the sensing platform, which can degrade the usefulness of MCS.

(3) *Collusion Attacks.* Collusion attacks are another form of internal attack, which refers to multiple users cooperate and jointly provide forged data. Therefore, the malicious attackers in MCS may generate faked data and submit them to the edge layer or the sensing platform for their own benefit (for example, gaining higher compensation for contributing to a crowdsensing task).

4. Location Difference-Based Privacy-Preserving

Due to the differentiated demands of sensing regions on location privacy-preserving, a typical MCS system issues various tasks, which should adopt different location privacy-preserving methods to meet the privacy needs of users and achieve accurate and efficient location privacy-preserving. Therefore, our proposed LDPF divides sensing regions into popular and remote regions, analyzes user characteristics and location privacy-preserving needs in various regions, and designs different location privacy-preserving methods.

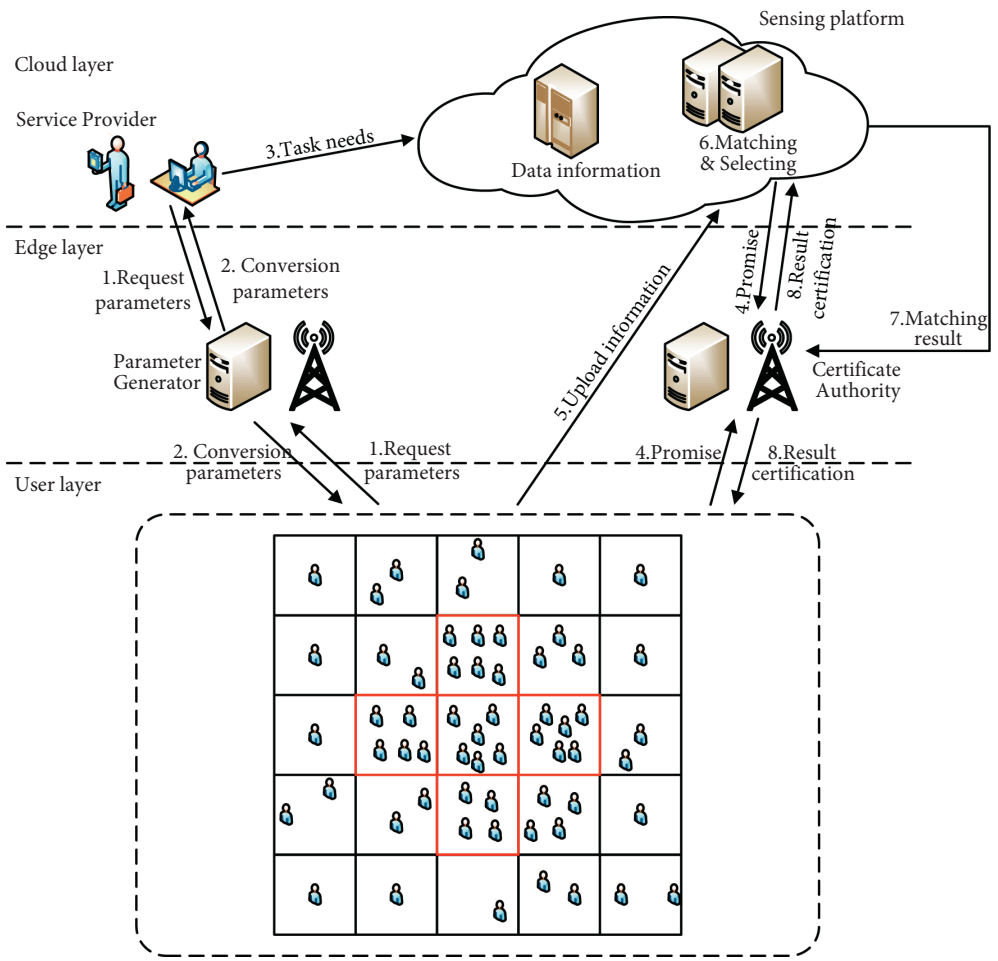

Figure 1: System model of edge-assisted privacy-preserving framework.

4.1. Region Classification. To the best of our knowledge, the state-of-the-art location privacy-preserving methods assume that the privacy-preserving requirements of users are constant. However, in practical MCS activities, the sensing platform has to consider the diversity of privacy-preserving needs for several reasons: (1) users are privacy-sensitive, and the strength of privacy-preserving largely depends on the scale of users; (2) prior works also found evidence that the utility of platform may also be affected by the complexity of privacy-preserving methods [32].

Nowadays, under the background of 'Smart city,' the traffic congestion bothers the managers and causes severe societal problems. Inspired by the previous work [33], we define popular regions and remote regions as follows:

Popular Regions. They possess abundant users and high-traffic sensing regions (e.g., shopping malls and popular tourist destinations)

Remote regions. They possess scarce users and low-traffic sensing regions (e.g., suburban factories)

A significant feature of popular regions is an abundant number of users, which leads to the diverse choice of users for sensing platforms. In addition, more public service personnel are usually active in popular regions (e.g., police and taxi drivers) and provide more accurate information through government equipment without requiring strong location privacy-preserving. Therefore, a low complexity privacy-preserving method is needed to improve the efficiency of the sensing platform. In contrast, remote regions only have a relatively fixed choice of users for the sensing platform due to the scarcity of users and low traffic. That is, a high-security location privacy-preserving method is required to guarantee user security. Table 2 shows the difference between popular regions and remote regions.

In brief, various environments can result in vastly different privacy-preserving needs. Therefore, our proposed LDPF mainly considers two distinct scenarios (i.e., popular regions and remote regions) to realize location difference-based privacy-preserving.

4.2. Location Privacy-Preserving for Popular Regions. In conventional cloud-based MCS architecture, user information is generally reported to the platform and periodically updated for task requirements, relying on a trusted sensing platform, and incurs long communications latency and privacy risks threatening sensitive user location. In addition, k-anonymity mechanisms are widely employed to protect

Table 2: Difference between popular regions and remote regions.

Classification	Number of users	Privacy sensitivity	Public service personnel	Traffic flow
Popular regions	Abundant	Low	High	High
Remote regions	Scarce	High	Low	Low

centralized location privacy-preserving. If the size meets the demand of conventional k-anonymity, the attackers cannot discriminate the participant from the other $k-1$ users in the same group. However, anonymous servers often have accurate user location information that still risks privacy disclosure when anonymous services are subject to external attacks. To protect the identity privacy of users, especially from the vulnerable, honest-but-curious MCS platform, we introduce the edge layer and propose a Coordinate Transformation and Bit Commitment (CTBC) method, which satisfies the low computational complexity of location privacy-preserving and hides user's actual coordinates. As a result, edge computing servers can collaborate to protect the security of these sensitive data substantially.

The sensing platform receives a large number of signed data, which should be timely verified without revealing the identity of data information. The edge layer generates the hash function for each user. Subsequently, users employ the bit commitment protocol to sign their information before transmitting it to the platform via the edge layer. The platform selects users according to their data and verifies commitments. Upon receiving the match result, the edge layer notifies the selected users. The process of an entire location privacy-preserving is as follows, as shown in Figure 2.

Step 1 (location hiding). In k-anonymity privacy-preserving, an anonymous server only cares about the relative distance between users. Therefore, we introduce a coordinate transformation method, which ensures the stability of the relative distance between users and hides users' actual coordinates. Specifically, users and the sensing platform send a location hiding request (i.e.,1. request parameters) to the edge layer. When receiving a parameter request, the parameter generator at the edge layer sends $\mathrm{CCP}_t = (\theta_t, b_t)$ (i.e., 2. CCP_t) to both participants to realize location hiding. The coordinate transformation method is as follows.

We have the accurate coordinates of each user i, denoted by $u_i(x_i, y_i)$. Then, we perform the coordinate transformation, which can be expressed as follows:

$$\begin{cases} x_i' = x_i \sin\theta_t - y_i \cos\theta_t + b_t, \\ y_i' = x_i \cos\theta_t - y_i \sin\theta_t + b_t, \end{cases} \tag{1}$$

where θ_t and b_t are coordinate transformation parameters in time t and $\mathrm{CCP}_t = (\theta_t, b_t)$, in which $\theta_t \in [0, 2\pi]$ and $b_t \in [0, 10]$.

Proposition 1. *Equation (1) does not change the relative distance between users.*

Proof. $u_A(x_A, y_A)$ and $u_B(x_B, y_B)$ are known, and the transformed locations are (x_A', y_A') and (x_B', y_B'). Then, we have

$$\begin{aligned} &\sqrt{(x_B' - x_A')^2 + (y_B' - y_A')^2} \\ &= \sqrt{[(x_B \sin\theta_t - y_B \cos\theta_t + b_t) - (x_A \sin\theta_t - y_A \cos\theta_t + b_t)]^2 + [(x_B \cos\theta_t + y_B \sin\theta_t + b_t) - (x_A \cos\theta_t + y_A \sin\theta_t + b_t)]^2} \\ &= \sqrt{[(x_B - x_A)\sin\theta_t - (y_B - y_A)\cos\theta_t]^2 + [(x_B - x_A)\cos\theta_t + (y_B - y_A)\sin\theta_t]^2} \\ &= \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}. \end{aligned} \tag{2}$$

The relative distance between u_A and u_B remains unchanged after a coordinate transformation. We also derive the coordinate inverse transformation according to equation (1), which can be expressed as follows:

$$\begin{cases} x_i = x_i' \sin\theta_t + y_i' \cos\theta_t - b_t(\sin\theta_t + \cos\theta_t), \\ y_i = y_i' \sin\theta_t - x_i' \cos\theta_t - b_t(\sin\theta_t - \cos\theta_t). \end{cases} \tag{3}$$

□

Step 2. (bit commitment). To ensure the authenticity of data, we employ the well-known bit commitment [34]. In this paper, the process of bit commitment (i.e., 3. bit commitment and 6. commitment verification) is implemented at the edge layer. Users and service providers can bind their identities to a number to prevent deception from each other. Meanwhile, to ensure user privacy and reduce communication overhead, all users need to participate in the commitment phase but only verify the selected user. The protocol is as follows.

Protocol 1. Bit commitment.

Commitment Phase. User i generates two random numbers (i.e., r_{i1} and r_{i2}) and binds the random number to ID_i $(r_{i1} \| r_{i2} \| \mathrm{ID}_i)$, specific steps are as follows:

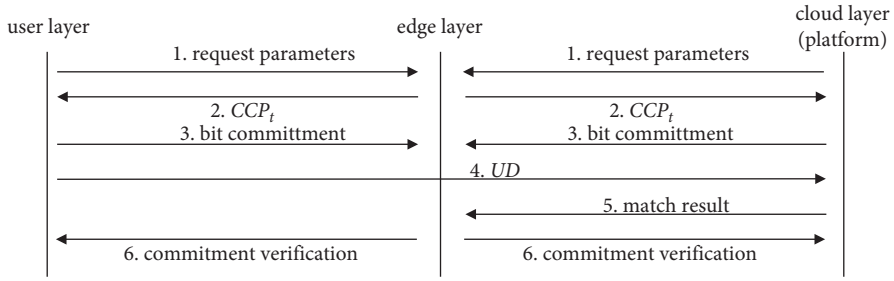

Figure 2: Privacy-preserving for popular regions.

i calculates the hash value of $(r_{i1} \| r_{i2} \| \mathrm{ID}_i)$, which can be expressed as follows:

$$c = \mathrm{Hash}\left(r_{i1} \| r_{i2} \| \mathrm{ID}_i\right), \tag{4}$$

where c and r_{i1} are sent to the sensing platform as a commitment to ID_i.

Reveal the Commitment Phase. i sends $(r_{i1} \| r_{i2} \| \mathrm{ID}_i)$ and the sensing platform verifies that.

In Protocol 1, for the commitment phase, a successful commitment scheme needs to ensure that users will not disclose the commitment value to service providers, cannot change the commitment value, and complete it in the probability of polynomial time. For the reveal phase, users need to provide promised values and random values for service providers to verify. When the publisher successfully validates the message, the promised value is accepted.

Step 3 (upload data). Users upload the transformed data (i.e., 4. UD), which can be expressed as follows:

$$\begin{cases} \mathrm{UD} = \left\{L'_{k_j-\mathrm{center}}, k_j\right\}, \\ I = \sum_{j=1}^{m} k_j = \{i_1, i_2, \ldots, i_n\}, \end{cases} \tag{5}$$

where I is the number of users participating in MCS and k_j is the number of k-anonymity group users.

Increasing the value of k_j will lead to more users in a hidden region. In other words, attackers are hard to achieve the specific information of each user. However, it will lead to more resource consumption. $L'_{k_j-\mathrm{center}} = (x'_{k_j-\mathrm{center}}, y'_{k_j-\mathrm{center}})$ represents the transformed center point coordinates of k-anonymity group users. Considering the slight difference of user data in the same region, we define the center of the k-anonymity group as the average value of all users, which can be expressed as follows:

$$\begin{cases} x'_{k_j-\mathrm{center}} = \sum_{j=1}^{k_j} \dfrac{x'_i}{k_j}, \\ y'_{k_j-\mathrm{center}} = \sum_{j=1}^{k_j} \dfrac{y'_i}{k_j}. \end{cases} \tag{6}$$

Step 4 (select appropriate user information). The sensing platform performs matching operations and returns results to the edge layer (i.e., 5. match result) when receiving UD.

Firstly, the sensing platform confirms the transformed center coordinates of the interest region $L'_{p-\mathrm{center}} = (x'_{p-\mathrm{center}}, y'_{p-\mathrm{center}})$, the number of required users k_p, and the transformed coordinates of the interest region POI′, which can be expressed as follows:

$$\mathrm{POI}' = \left(x'_{p-\min}, y'_{p-\min}\right) \times \left(x'_{p-\max}, y'_{p-\max}\right), \tag{7}$$

where $x'_{p-\min}$, $x'_{p-\max}$, $y'_{p-\min}$, and $y'_{p-\max}$ represent the limit for each user's location, respectively.

Then, the sensing platform performs matching operations and selects k-anonymity users with the highest matchmaking degree. In this paper, we focus on the privacy-preserving of user location. The data quality of users is negatively correlated to the location (i.e., the closer to the task center, the better data quality), which meets the consensus of existing location-based privacy-preserving methods (see Hypothesis 3 in Section 3.1). Therefore, our matching calculation method improves root mean squared error (RMSE) and reflects the difference between user data and task requirement data, which can be expressed as follows:

$$\mathrm{match}_{k_j-\mathrm{center},p-\mathrm{center}} = \frac{1}{1 + \sqrt{\left(x'_{k_j-\mathrm{center}} - x'_{p-\mathrm{center}}\right)^2 + \left(y'_{k_j-\mathrm{center}} - y'_{p-\mathrm{center}}\right)^2}}. \tag{8}$$

Equation (8) aims to calculate the similarity between user data and task center data. In other words, high-quality user data have higher matching values and can be easier to select. *Note*. The number of users selected should meet the

needs of service providers (i.e., $\|\text{match}_{k_j-\text{center},p-\text{center}}\| = k_p$). To solve the problem of user selection, we consider the following two cases.

Case 1 ($k_{j-\text{best}} \geq k_p$).

$k_{j-\text{best}}$ is the k-anonymity user with the best matching degree. In order to ensure the privacy of users, the sensing platform selects users randomly, and the selected users perform verification.

Case 2 ($k_{j-\text{best}} < k_p$).

The sensing platform first selects the optimal anonymity users and then selects users from the remaining sorting to meet the number of users required by service providers. Finally, the selected users perform verification.

Step 5 (verification). According to the received match result, the edge layer performs the commitment verification (i.e., 6. commitment verification) for the selected users.

We use a one-way function $(r_{i1}\|r_{i2}\|\text{ID}_i)$ to construct a bit commitment, where ID and UD correspond one by one. Then, we compare the value with the initially received value and a random number. If it matches, the commitment is valid.

In this paper, to ensure data integrity and prevent the dependence on the trusted sensing platform, we convert the problem into maximizing the user matching degree, which can be expressed as follows:

$$\begin{aligned} &\text{max}\text{match}_{k_j-\text{center},p-\text{center}} \\ &\quad k_j, \quad \forall k_j \in I \\ &\quad x'_{p-\min} \leq x'_{k_j-\text{center}} \leq x'_{p-\max} \\ &\text{s.t.} \quad y'_{p-\min} \leq y'_{k_j-\text{center}} \leq y'_{p-\max} \\ &\quad \left\|\text{match}_{k_j-\text{center},p-\text{center}}\right\| = k_p. \end{aligned} \tag{9}$$

Here, the matching degree of all users can be calculated through equation (8). Then, the sensing platform selects the optimal users by the value of matching degree, which is a simple baseline comparison. The objective function in the first line is to maximize the matching degree of all users. The second to fourth line defines the limit for each user's number and location, respectively, and the fifth line indicates the number of users required by service providers.

Algorithm 1 provides the detailed process of privacy-preserving for popular regions. From (2) to (4), the algorithm is used to implement coordinate transformation and k-anonymity construction. From (5) to (11), it is used to judge whether the center of k-anonymity satisfies the publisher's location requirements. From (12) to (19), it is used to determine whether the optimal number of k-anonymity group users meets the requirements of service providers.

4.3. Location Privacy-Preserving for Remote Regions. For remote regions, considering the small number of candidate users and intense awareness of privacy-preserving, a high-security location privacy-preserving method is needed to ensure the security of users and complete data collection inefficiently. Moreover, the existence of obstacles prevents users from adopting the optimal movement method (i.e., the Euclidean distance). That is, the moving distance algorithm underlying the Euclidean distance is not suitable for real-world MCS applications. To tackle this problem, we design a Paillier Encryption Data Coding (PDC) privacy-preserving method that realizes calculating moving distance without exposing users' actual location and preventing malicious users. Furthermore, PDC adopts a more realistic distance calculation mode (i.e., Manhattan distance) to overcome the error caused by obstacles (e.g., buildings, trees, and other shelters).

The sensing platform receives a large number of location information, which should be timely verified without revealing the identity of data information. The edge layer generates hash functions and encoding functions for each user. Users then implement encoding functions to hide their information and employ the Paillier encryption algorithm to sign their information before transmitting them to the platform. Subsequently, the platform uses the Paillier decryption algorithm to match users. Upon receiving the match result, the edge layer notifies the selected users, and users need to verify the authenticity of the information. The process of an entire location privacy-preserving is as follows, as shown in Figure 3.

Step 6 (location hiding). Users and the sensing platform send a location hiding request (i.e., 1. request parameters) to the edge layer. When receiving a parameter request, the parameter generator at the edge layer sends hidden location coding (i.e., 2. Encode 1 and 2) to both participants to realize location hiding.

Assume that D (x_D, y_D) denotes the user location and C (x_C, y_C) denotes the task center location. We determine the integer set (i.e., $S = \{s_1, s_2, \ldots, s_n\}$) through POI $(x_{\text{POI}}, y_{\text{POI}})$, where $s_1 < s_2 < \cdots < s_n$, $s_n = \max\langle \max x_{\text{POI}}, \max y_{\text{POI}}\rangle$ and $s_1 = \min\langle \min x_{\text{POI}}, \min y_{\text{POI}}\rangle$. x_{POI} and y_{POI} represent the abscissa and ordinate of POI, respectively. The data coding method is expressed as follows:

Encode 1. According to $D(x_D, y_D)$, $x_D, y_D \in S$, and $S = \{s_1, s_2, \ldots, s_n\}$, we construct a $2n$-dimensional array (i.e., $V(D)$). Assuming $x_D = s_k$ and $y_D = s_l$, we code the first k elements of the abscissa as 0 and the rest as 1; the first l elements of the ordinate as 0 and the rest as 1, which can be expressed as follows:

$$\begin{cases} v_{D_x 1} = \cdots = v_{D_x k} = 0, \\ v_{D_x (k+1)} = \cdots = v_{D_x n} = 1, \\ v_{D_y 1} = \cdots = v_{D_y l} = 0, \\ v_{D_y (l+1)} = \cdots = v_{D_y n} = 1. \end{cases} \tag{10}$$

The encoding array of D under S is as follows:

Input: $I = \{i_1, i_2, \ldots, i_n\}$, $\mathrm{CCP}_t = (\theta_t, b_t)$, k_j, POI$\prime$, $L'_{p-\mathrm{center}}$, k_p
Output: selected users
(1) Select $\mathrm{CCP}_t = (\theta_t, b_t)$ to realize a coordinate transformation
(2) Calculate L' by equation (1)
(3) Calculate $L'_{k_j-\mathrm{center}}$ by k_j and equations (1) and (6)
(4) Determine POI$\prime$, $L'_{p-\mathrm{center}}$, k_p
(5) for $j \leftarrow 1$ to m DO
(6) if $x'_{k_j-\mathrm{center}} \in [x'_{p-\min}, x'_{p-\max}]$ and $y'_{k_j-\mathrm{center}} \in [y'_{p-\min}, y'_{p-\max}]$
(7) Calculate the matchmaking degree by equation (8), and sort user data in descending order (i.e., $\mathrm{STC} = \left\{\mathrm{match}_{k_j-\mathrm{center},p-\mathrm{center}} | j = 1, 2, \ldots, m\right\}$)
(8) else
(9) Delete j
(10) end for
(11) Determine $k_{j-\mathrm{best}}$ from STC
(12) if $k_{j-\mathrm{best}} \geq k_p$ then
(13) Random select users from $k_{j-\mathrm{best}}$
(14) else
(15) $k_{j-\mathrm{best}}$ is selected
(16) Determine $k_{j-\mathrm{other}}$ from STC
(17) until $\|\mathrm{match}_{k_j-\mathrm{center},p-\mathrm{center}}\| = k_p$
(18) end if
(19) **End**

Algorithm 1: CTBC.

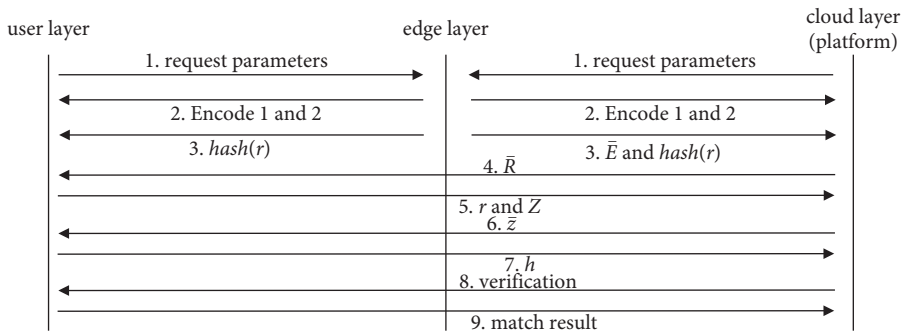

Figure 3: Privacy-preserving for remote regions.

$$\begin{cases} V(D_x) = \left(v_{D_x1}, \ldots, v_{D_xn}\right), \\ V\left(D_y\right) = \left(v_{D_y1}, \ldots, v_{D_yn}\right), \\ V(D) = \left(V(D_x), V\left(D_y\right)\right). \end{cases} \tag{11}$$

Encode 1 enables privacy protection of users' location. However, private key owners possess complete data information and send it to opponents in traditional encryption and decryption algorithms. As a result, private key owners may tamper with data for more excellent benefits. To overcome the inadequacy of location privacy-preserving with a single encoding method, we also added Encode 2.

Encode 2. According to $D(x_D, y_D)$, $x_D, y_D \in S$, and $S = \{s_1, s_2, \ldots, s_n\}$, we construct a $2n$-dimensional array (i.e., $W(D)$). Assuming $x_D = s_k$ and $y_D = s_l$, we code the first k elements of the abscissa as 1 and the rest as 0; the first l elements of the ordinate as 1 and the rest as 0, which can be expressed as follows:

$$\begin{cases} w_{D_x1} = \cdots = w_{D_xk} = 1, \\ w_{D_x(k+1)} = \cdots = w_{D_xn} = 0, \\ w_{D_y1} = \cdots = w_{D_yl} = 1, \\ w_{D_y(l+1)} = \cdots = w_{D_yn} = 0. \end{cases} \tag{12}$$

The encoding array of D under S is as follows:

$$\begin{cases} W(D_x) = \left(w_{D_x1}, \ldots, w_{D_xn}\right), \\ W\left(D_y\right) = \left(w_{D_y1}, \ldots, w_{D_yn}\right), \\ W(D) = \left(W(D_x), W\left(D_y\right)\right). \end{cases} \tag{13}$$

Step 7 (the Paillier encryption). To ensure the authenticity of data, we employ a cheating-prevention protocol to calculate

the Manhattan distance, which is implemented at the edge layer. The protocol is as follows.

Protocol 2. A cheating-prevention protocol to calculate the Manhattan distance.

Preparation. The edge layer determines a hash function and a Paillier encryption algorithm (i.e., 3. $\overline{E}$ and hash (r)).

(1) The sensing platform (i.e., C) encrypts the task location information (i.e., $V(C)$), as shown in the following:

$$\overline{R} = \left(\overline{E}\left(v_{C_x 1}\right), \ldots, \overline{E}\left(v_{C_y n}\right)\right), \tag{14}$$

where $\overline{E}$ is the Paillier encryption algorithm. Then, users achieve the encrypted task location information (i.e., 4. $\overline{R}$).

(2) D chooses a random number (i.e., r), calculates $w = \text{hash}(r)$, and uses the Paillier encryption algorithm to achieve $\overline{E}(r)$.

$$Z = \prod_{i=1}^{2n} \left(\overline{E}\left(v_{Di}\right)^{w_{Di}}\right)\overline{E}(r), \tag{15}$$

D sends r and Z to C (i.e., 5. r and Z).

(3) C decrypts Z to achieve $\overline{z} = \overline{H}(Z)$ and sends to D, where $\overline{H}$ is the decryption algorithm (i.e., 6. $\overline{z}$).

(4) D calculates $h = \overline{z} - r$ and sends to C (i.e., 7. h).

(5) C verifies $\text{hash}(\overline{z} - h) = w$. If the verification is successful, then it outputs h; otherwise, does not accept h (i.e., 8. verification).

Step 8 (select appropriate users). The sensing platform calculates the Manhattan distance between D and C and returns results to users (i.e., 8. match result) when receiving $\overline{R}$.

To calculate the Manhattan distance between D and C, we encode D (C) with Encode 1 and then encode C (D) with Encode 2, which can be expressed as follows:

$$\begin{cases} VW_{DC} = \left(V(D_x) \odot W(C_x), V\left(D_y\right) \odot W\left(C_y\right)\right), \\ WV_{DC} = \left(W(D_x) \odot V(C_x), W\left(D_y\right) \odot V\left(C_y\right)\right), \end{cases} \tag{16}$$

where $\odot$ represents an XNOR (Exclusive NOR) operation.

Proposition 2. *The Manhattan distance between D and C is equal to the dot product of VW_{DC} and WV_{DC}:*

$$\begin{aligned} &\left|x_D - x_C\right| + \left|y_D - y_C\right| = VW_{DC} \cdot WV_{DC} \\ &= \left[V(D_x) \odot W(C_x)\right] \cdot \left[W(D_x) \odot V(C_x)\right] \\ &+ \left[V\left(D_y\right) \odot W\left(C_y\right)\right] \cdot \left[W\left(D_y\right) \odot V\left(C_y\right)\right]. \end{aligned} \tag{17}$$

Proof. (i) $|x_D - x_C| = [V(D_x) \odot W(C_x)] \cdot [W(D_x) \odot V(C_x)]$.

Assume that the position of x_D in S is p'_d, the position of x_C in S is q'_c, and $x_D \le x_C$ (i.e., $p'_d \le q'_c$). x_D performs Encode 1 and Encode 2, respectively:

$$\begin{cases} V_{D_x} = \left(\underbrace{0, 0, \ldots, 0}_{p'_d}, \underbrace{1, 1, \ldots, 1}_{n-p'_d}\right), \\ W_{D_x} = \left(\underbrace{1, 1, \ldots, 1}_{p'_d}, \underbrace{0, 0, \ldots, 0}_{n-p'_d}\right), \end{cases} \tag{18}$$

x_C performs Encode 2 and Encode 1, respectively:

$$\begin{cases} W_{C_x} = \left(\underbrace{1, 1, \ldots, 1}_{q'_c}, \underbrace{0, 0, \ldots, 0}_{n-q'_c}\right), \\ V_{C_x} = \left(\underbrace{0, 0, \ldots, 0}_{q'_c}, \underbrace{1, 1, \ldots, 1}_{n-q'_c}\right). \end{cases} \tag{19}$$

XNOR operation is performed between V_{D_x} and W_{C_x}; W_{D_x} and V_{C_x}, respectively:

$$\begin{cases} V(D_x) \odot W(C_x) = \left(\underbrace{0, 0, \ldots, 0}_{p'_d}, \underbrace{1, 1, \ldots, 1}_{q'_c - p'_d}, \underbrace{0, 0, \ldots, 0}_{n-q'_c}\right), \\ W(D_x) \odot V(C_x) = \left(\underbrace{0, 0, \ldots, 0}_{p'_d}, \underbrace{1, 1, \ldots, 1}_{q'_c - p'_d}, \underbrace{0, 0, \ldots, 0}_{n-q'_c}\right). \end{cases} \tag{20}$$

The dot product of equation (20) is performed as follows:

$$[V(D_x) \odot W(C_x)] \cdot [W(D_x) \odot V(C_x)] = q'_c - p'_d = \left|x_D - x_C\right|. \tag{21}$$

When $x_D > x_C$, $[V(D_x) \odot W(C_x)] \cdot [W(D_x) \odot V(C_x)] = p'_d - q'_c = |x_D - x_C|$ can be proved in the same way. □

Proof. (ii) $|y_D - y_C| = [V(D_y) \odot W(C_y)] \cdot [W(D_y) \odot V(C_y)]$.

Assume that the position of y_D in S is p''_d, the position of y_C in S is q''_c, and $y_D \le y_C$ (i.e., $p''_d \le q''_c$). y_D performs Encode 2 and Encode 1, respectively:

$$\begin{cases} W_{D_y} = \left(\underbrace{1, 1, \ldots, 1}_{p''_d}, \underbrace{0, 0, \ldots, 0}_{n-p''_d}\right), \\ V_{D_y} = \left(\underbrace{0, 0, \ldots, 0}_{p''_d}, \underbrace{1, 1, \ldots, 1}_{n-p''_d}\right), \end{cases} \tag{22}$$

y_C performs Encode 1 and Encode 2, respectively:

$$\begin{cases} V_{C_y} = \left(\underbrace{0, 0, \ldots, 0}_{q''_c}, \underbrace{1, 1, \ldots, 1}_{n-q''_c}\right), \\ W_{C_y} = \left(\underbrace{1, 1, \ldots, 1}_{q''_c}, \underbrace{0, 0, \ldots, 0}_{n-q''_c}\right). \end{cases} \tag{23}$$

XNOR operation is performed between V_{D_y} and V_{D_y}; W_{D_y} and V_{C_y}, respectively:

$$\begin{cases} V(D_y) \odot W(C_y) = \left(\underbrace{0, 0, \ldots, 0}_{p_d''}, \underbrace{1, 1, \ldots, 1}_{q_c'' - p_d''}, \underbrace{0, 0, \ldots, 0}_{n - q_c''} \right), \\ W(D_y) \odot V(C_y) = \left(\underbrace{0, 0, \ldots, 0}_{p_d''}, \underbrace{1, 1, \ldots, 1}_{q_c'' - p_d''}, \underbrace{0, 0, \ldots, 0}_{n - q_c''} \right). \end{cases} \tag{24}$$

The dot product of equation (24) is performed.

$$\left[V(D_y) \odot W(C_y)\right] \cdot \left[W(D_y) \odot V(C_y)\right] = q_c'' - p_d'' = |y_D - y_C|. \tag{25}$$

When $y_D > y_C$, $[V(D_y) \odot W(C_y)] \cdot [W(D_y) \odot V(C_y)] = p_d'' - q_c'' = |y_D - y_C|$ can be proved in the same way.

In our methods, based on equations (21) and (25), the Manhattan distance between D and C is equal to the dot product of VW_{DC} and WV_{DC}. □

Step 9 (verification). According to the match result, the sensing performs verification (i.e., 7. h and verification) for the selected users.

When users receive h from the sensing platform, users calculate the value of hash$(\overline{z} - h)$. If hash$(\overline{z} - h) = w$, the verification is valid. Otherwise, users will refuse it.

In general, to ensure data security and reduce the calculation error of the moving distance, we convert the problem into minimizing the Manhattan distance, which can be expressed as follows:

$$\begin{aligned} &\min f(D, C) \\ &\quad f(D, C) = |x_D - x_C| + |y_D - y_C| \\ &\text{s.t.} \quad \min x_{\text{POI}} \le x_D \le \max x_{\text{POI}}, \quad \forall x_D \in S \\ &\quad \min y_{\text{POI}} \le y_D \le \max y_{\text{POI}}, \quad \forall y_D \in S. \end{aligned} \tag{26}$$

Here, the objective function in the first line is to minimize the Manhattan distance based on secure computation. The second line defines the calculation method of Manhattan distance. Finally, the third and fourth line represents the limit for each user's location, respectively. At the same time, we combine the Paillier encryption method to prevent malicious users from cheating.

Algorithm 2 provides the detailed process of privacy-preserving for remote regions. From (2) to (3), the algorithm is used to filter users, which aims to select the proper users. From (4) to (6), it is used for data encoding, which seeks to calculate the Manhattan distance confidentially. From (7) to (11), it performs the Paillier encryption for users and the sensing platform, where Protocol 2 introduces the detailed encryption steps. From (12) to (14), it is used to delete users who do not meet the requirements. Thus, Algorithm 2 can ensure the secret calculation of Manhattan distance and realize the identification of malicious users.

5. Theoretical Analysis and Simulation

In this section, we elaborately evaluate the effectiveness of the method from the aspects of security analysis and performance evaluation.

5.1. Experimental Setup. The aid of Python 3.6 software implements all simulations designed to validate our proposed LDPF framework on a computer with Windows 10 operating system, Intel Core I7 CPU @ 2.2 GHz, and 8 GB RAM and use the real-world datasets and position data reported by Dias et al. [35] from the city of Rio de Janeiro to evaluate our scheme. Selected performance indicators include running time and drift degree [36]. During the location privacy-preserving process, the insignificant communications time and parameter distribution time can be neglected, whereas the running time in our simulation involves the verification time and user matching time.

5.2. Security Analysis. We evaluate the security performance of our proposed LDPF in three attacks (i.e., external attacks, internal attacks, and collusion attacks).

5.2.1. External Attacks. A common attack method is that external attackers eavesdrop on the communications between users and the platform to steal real-location data. In this paper, we assume that attackers can eavesdrop on the whole network.

Our proposed LDPF divides sensing regions into popular and remote regions and designs different location privacy-preserving methods (i.e., CTBC and PDC). For popular regions, CTBC converts and anonymizes the user data, and the data eavesdropped by external attackers are UD. Firstly, to obtain real user data, external attackers must obtain $\text{CCP}_t = (\theta_t, b_t)$ from the edge server. However, it is difficult for external attackers to eavesdrop on the truth $\text{CCP}_t = (\theta_t, b_t)$ in the edge layer when considering the secure edge server (i.e., Hypothesis 2). Secondly, when malicious external attackers capture the edge layer, CTBC still guarantees the anonymity of users by Protocol 1, whereas external attackers can only achieve c and r_{i1} but cannot identify their specific sources. According to the characteristics of the hash function, it is impossible to find the same value from different messages. That is, attackers have no clue about the actual location. The probability of successful guessing is $1/k$ since each location in the intercepted set has the same query probability.

For remote regions, PDC anonymizes and encrypts the user data, and the data eavesdropped by external attackers are $\overline{R}$. Firstly, to obtain real user data, external attackers must master the data coding methods (i.e., Encode 1 and Encode 2). It is difficult for external attackers to capture the encoding methods in the edge layer when considering the secure edge server (i.e., Hypothesis 2). At the same time, since PDC uses two encoding methods to prevent the private key owner

Input: $I = \{i_1, i_2, \ldots, i_n\}$, C (x_C, y_C), POI, $S = \{s_1, s_2, \ldots, s_n\}$
Output: $f(I, C) = |x_i - x_C| + |y_i - y_C|$
(1) Confirm $S = \{s_1, s_2, \ldots, s_n\}$ by POI
(2) for $i \leftarrow 1$ to n DO
(3) if $(x_i, y_i) \in$ POI
(4) Calculate $V(I)$ and $V(C)$ by equation (11)
(5) Calculate $W(I)$ and $W(C)$ by equation (13)
(6) Calculate VW and WV by equation (16)
(7) Encryption $V(I)$ by equation (14)
(8) Calculate Z by equation (15) and select a random number r
(9) Decrypt Z
(10) Calculate h
(11) Verification hash $(\overline{z} - h) = w$
(12) else
(13) Delete (x_i, y_i)
(14) end if
(15) **End for**

Algorithm 2: PDC.

from tampering with the data to gain greater benefits, attackers capture a single encoding method which is invalid. Secondly, when malicious external attackers capture the edge layer, PDC still promises the security of users by the Paillier encryption, whereas external attackers can only use the group public key (i.e., $\overline{R}$ and $\overline{z}$) to verify the data but cannot identify their specific sources. Moreover, due to the random parameters in the Paillier encryption, the difficulty of tracing data has greatly increased. All in all, LDPF avoids external attacks.

5.2.2. Internal Attacks. Internal attackers forge their identities or submit faked data to gain higher benefits. For popular regions, CTBC leverages the well-known bit commitment [34] to ensure the authenticity of data. A complete bit commitment (i.e., Protocol 1) includes a commitment phase and a reveal phase. For the commitment phase, attackers may not disclose the commitment value (i.e., UD) to the sensing platform and complete the task in the probability of polynomial time. As a result, they may submit faked data. However, attackers cannot repudiate their promises in the data, and ID_i and UD_i are corresponding one by one. For the reveal phase, selected attackers need to provide promised values (i.e., $(r_{i1} \| r_{i2} \| \mathrm{ID}_i)$) to verify the authenticity of the data. When attackers fail to provide the promised data, the sensing platform will refuse compensation and reselect the appropriate user.

For remote regions, both data encoding method and the Paillier encryption scheme can prevent internal attacks. Firstly, PDC uses two encoding methods to prevent attackers from possessing complete data information and provide forged data to the sensing platform. Next, PDC ensures that participants in the Paillier encryption scheme cannot deny its presence. Protocol 2 points out that C (i.e., the sensing platform) is not the first to achieve the calculation results, which avoids the deception of internal attackers in the cloud layer. At the same time, to avoid cheating by malicious users, users should send r to C and make promise. The sensing platform verifies the selected user using hash $(\overline{z} - h) = w$ to detect the attacker's cheating behaviors. Therefore, participants cannot repudiate their promises in the data. In summary, LDPF ensures the authenticity of data and implements against internal attacks.

5.2.3. Collusion Attacks. Collusion attacks refer to multiple users cooperate and jointly provide forged data. One notable feature of popular regions is the large number of public service personnel (e.g., police and taxi drivers) and the provision of more accurate information through government equipment without strong location privacy protections. Therefore, similar to internal attacks, low-quality attacker data are difficult to achieve high benefits, which reduce the probability of collusion attacks.. Similar to internal attacks, low-quality attacker data are difficult to achieve high benefits, which reduce the probability of collusion attacks. In contrast, attackers with false high-quality data still need to perform bit commitment to ensure the authenticity and immutability of data. In other words, CTBC controls the frequency of collusion attacks from the perspective of data authenticity.

Collusive attacks in remote regions are easy to identify due to the scarcity of users and low traffic. Similar to internal attacks, it is meaningless for low-quality attackers to launch collusion attacks because attackers cannot obtain the ultimate benefit. Moreover, PDC ensures that participants in the Paillier encryption scheme cannot deny its presence, and high-quality attackers still need to provide real data. On the whole, LDPF can resist collusion attacks.

5.3. Experimental for Popular Regions. For popular regions, CTBC possesses the security performance almost equivalent to the famous k-anonymity. Therefore, this method is adopted for performance comparison purposes. Table 3 compares CTBC and the k-anonymity method.

From Table 3, we observe that CTBC achieves higher security by adding some system overhead. In terms of system security, since the anonymous server can directly obtain the

Table 3: Comparison of privacy-preserving methods.

Scheme		CTBC	k-anonymity
System overhead	Information loss	M	L
	Computational complexity	$(k+2)\ n$	kn
Security	Trusted third party	No	Yes
	User anonymity	Yes	Yes
	Location authenticity protection	Yes	No

L: low; M: medium; H: high.

user's actual location information, the k-anonymity method needs to rely on a trusted third-party platform. In contrast, CTBC leverages the powerful edge servers to avoid dependence on trusted platforms and adopts coordinate transformation parameters to hide accurate user information. In terms of system overhead, due to the simple anonymity method, the k-anonymity method has low information loss. The computational complexity of the k-anonymity is kn. By comparison, CTBC increases the information loss since the addition of a coordinate transformation parameter. In addition, to prevent participants from cheating, we also add bit commitment, and therefore the computational complexity of CTBC is $(k+2)\ n$.

Figure 4 shows the running time of CTBC and k-anonymity under a varying number of users, where the number of users varies from 100 to 1000, and elements in θ_t, b_t, k_j, and POI$\prime$ are $(\pi/3)$, 2, 5, and 2, respectively. As clearly illustrated in Figure 4, k-anonymity always has the lowest running time given the same number of users. The reason is that the k-anonymity method simply hides the user's location, uses a new place to achieve user matching, and effectively improves the platform's runtime. However, anonymous servers in the k-anonymity method can directly obtain the user's actual location information and cause the disclosure of location privacy information. In addition, the running time of both schemes is positively correlated to the user scale. Large-scale users will increase the workload of user matching degree calculation and ultimately reduce the running time of schemes. Worth noting is that CTBC leverages CCP_t and bit commitment to hide the real location of the users and ensure the authenticity of data and therefore has a long-running time.

The drift degree $\text{Drift}_{\text{deg}}$ is the difference between a transformed location $L'_{k_j-\text{center}}(x'_{k_j-\text{center}}, y'_{k_j-\text{center}})$ and its corresponding location $u_i(x_i, y_i)$. Both the mean and standard deviation (STD) are computed to measure the usefulness and stability of the location anonymization. The mathematical formulation of its mean is defined as follows:

$$\text{drift}_{\text{deg}} = \frac{1}{n}\sum_{i=1}^{n}\sqrt{\left(x'_{k_j-\text{center}} - x_i\right)^2 + \left(y'_{k_j-\text{center}} - y_i\right)^2}. \tag{27}$$

As shown in Figure 5, as the number of users grows, the drift degree of both methods remains basically unchanged, where k-anonymity-avg/min/max represents average/minimum/maximum drift in the k-anonymity method and CTBC-avg/min/max represents average/minimum/maximum drift

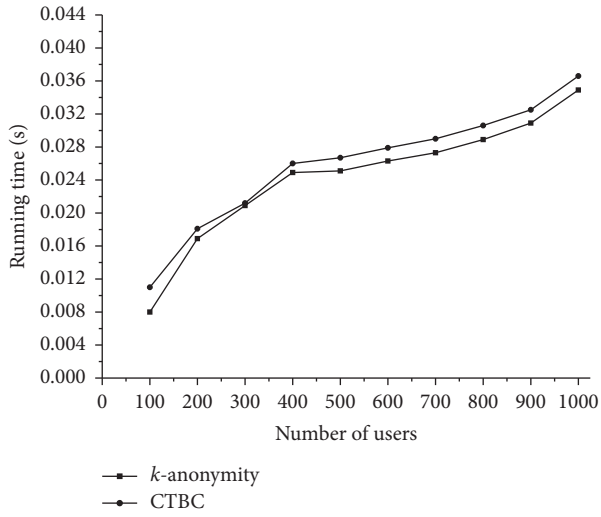

Figure 4: Running time for different users.

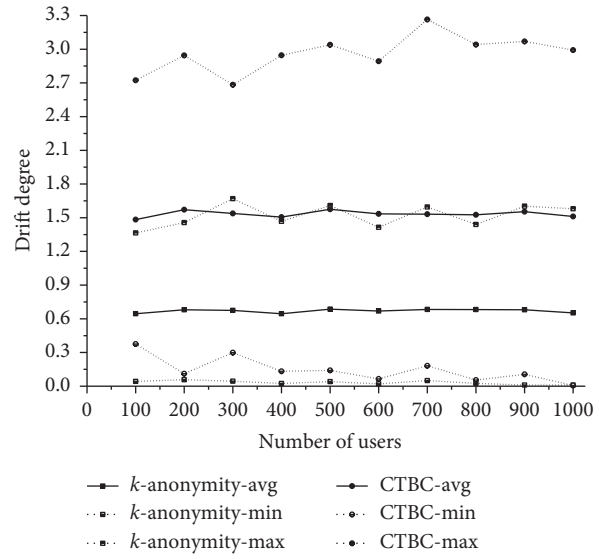

Figure 5: Drift degree for different users.

in the k-anonymity method. The reason is that the k-anonymity-based strategy aims to put a user and at least $k-1$ other users together constitute an anonymous region so that the probability of the user's real identity being recognized is not more than $1/k$. That is, the drift degree of users may be related to k_j. At the same time, CTBC selects the optimal users to form a group of anonymous users through the maximum matching degree, which leads to the best performance in

minimum drift degree (i.e., CTBC-min is better than k-anonymity-min). However, it should be noted that the average drift of CTBC is inferior to the k-anonymity method even though CTBC prevents the disclosure of location privacy information.

To illustrate the impact of k_j on different schemes, Figure 6 presents the running time with a different number of k-anonymity group users (i.e., k_j). Unlike the number of users, the number of k-anonymity group users represents the number of users in a group after k-anonymity hiding, where $I = \sum_{j=1}^{m} k_j = \{i_1, i_2, \ldots, i_n\}$. From the simulation results, the running time of CTBC is basically the same as the k-anonymity method. Among those, the high k_j can make the running time smaller. The reason is that the increase in k_j means that the number of anonymous users in the group are expanded, which decreases the times of matching and reduces the burden of the platform.

Figure 7 shows the curve of the drift degree during the simulation. It can be seen from results that the drift degree with CTBC is more than the k-anonymity method. In addition, as the number of k-anonymity group users grows, both methods' drift degree increases slightly. The reason is that the more significant number of users will enlarge the differences between users in the k-anonymity group users and lead to an increase in drift degree.

Finally, we further illustrate the impact of POI$\prime$ on different schemes. Figure 8 presents the running time of various interest regions. With the expansion of interest regions, both methods need to spend more time to finish the task. This is because that a larger region-of-interest will increase the scope of search optimal users and produce a longer time overhead.

As shown in Figure 9, the expansion of interest regions increases the drift degree of both methods, and the average drift of CTBC is inferior to the k-anonymity method. A larger POI$\prime$ will expand the range of user activity and eventually leads to an increase in the drift when considering that the data quality of users is negatively correlated to the location (i.e., Hypothesis 3). In addition, CTBC reduces the availability of data information to achieve more secure privacy protection, which is acceptable in location-sensitive user privacy-preserving.

5.4. Experimental for Remote Regions. For remote regions, we analyze the computational complexity and communication complexity of the protocol. Specifically, we consider the impact of users and r on the runtime of privacy-preserving, where the number of users varies from 5 to 25 and the value of r varies from 16 bit to 128 bit, respectively.

5.4.1. Computational Complexity. In the Paillier encryption algorithm, users need to perform $4n$ encryption and 1 decryption. The sensing platform needs $4n$ modular multiplication operations at most. Protocol 2 needs n modular multiplication operations. Therefore, the computational complexity of Protocol 2 is $2(6n + 1)$.

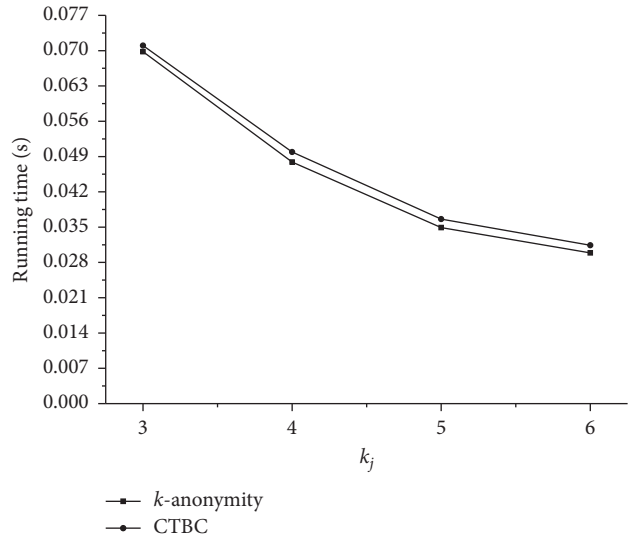

Figure 6: Running time for different k_j.

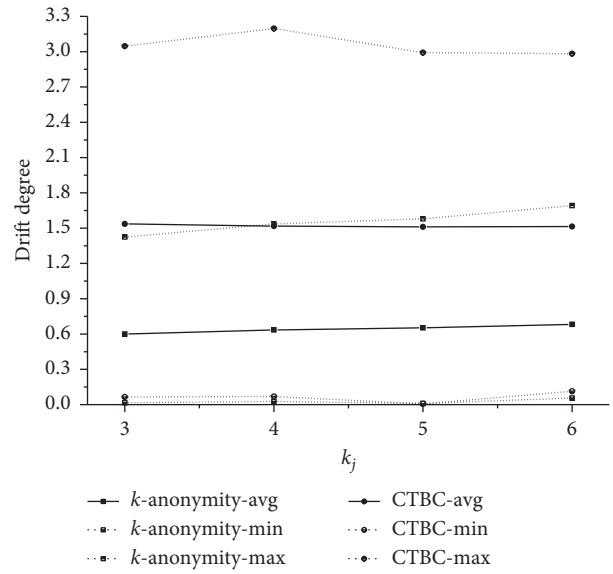

Figure 7: Drift degree for different k_j.

5.4.2. Communication Complexity. Protocol 2 requires 2 rounds of communication because we use the number of communication rounds to calculate.

In this paper, we only analyze the efficiency of PDC since there is currently no method for secretly calculating the Manhattan distance between users and the task center. Encode 1 represents that Manhattan distance is only calculated by Encode 1, which cannot prevent users from cheating. Figures 10–13 show the effect of different values of r on running time.

Figures 10–13 show that the running time of PDC is basically the same as Encode 1. Meanwhile, the increase in r and n will reduce the timeliness of PDC. The main reasons are as follows: firstly, the more significant value of n means that more users will be recruited by PDC, whose winners

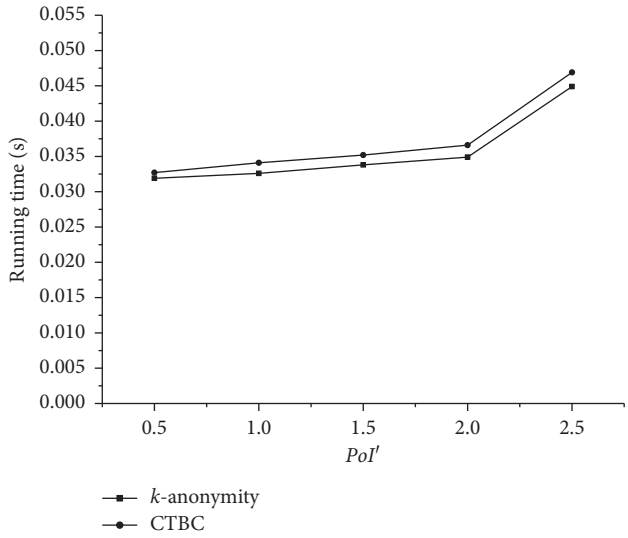

Figure 8: Running time for different PoI′.

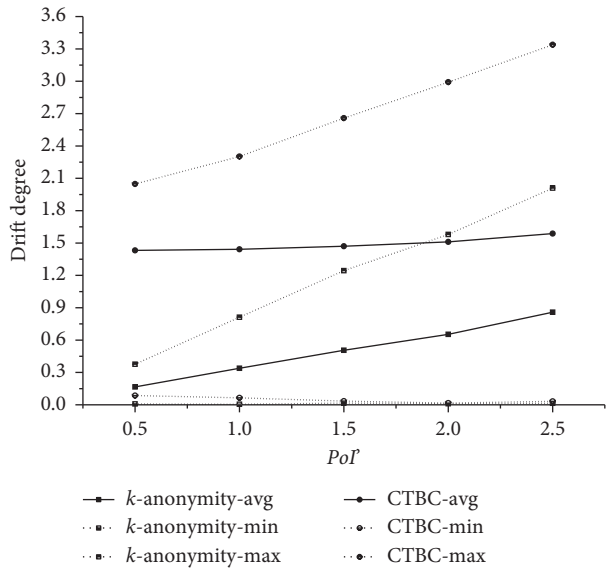

Figure 9: Drift degree for different PoI′.

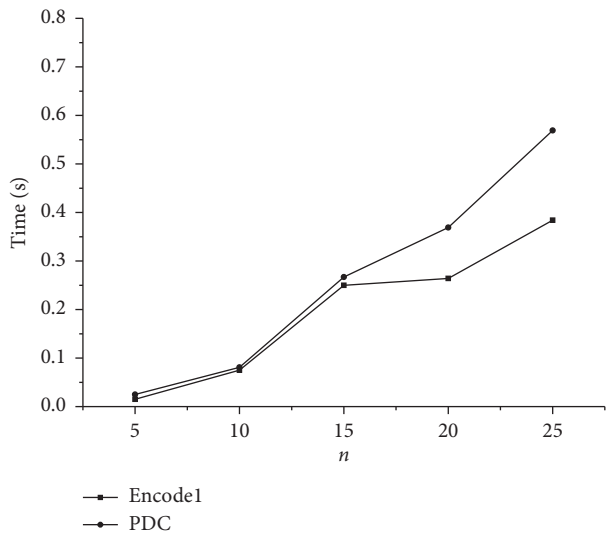

Figure 10: Time vs. 16 bit.

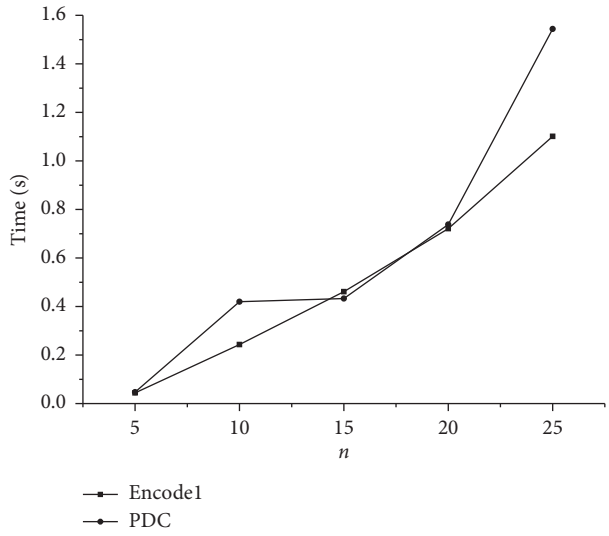

Figure 11: Time vs. 32 bit.

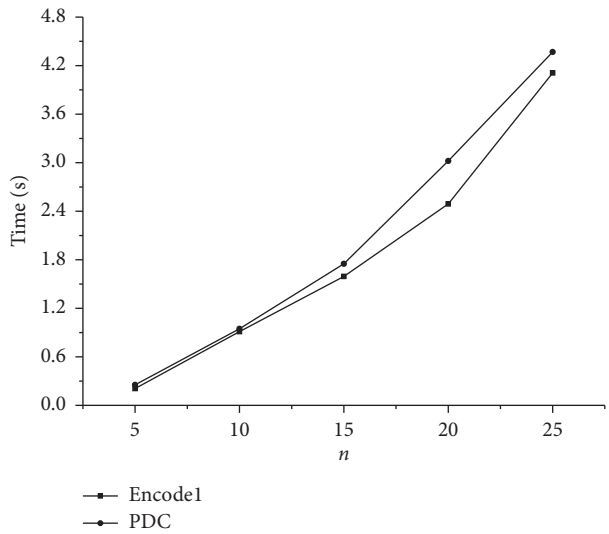

Figure 12: Time vs. 64 bit.

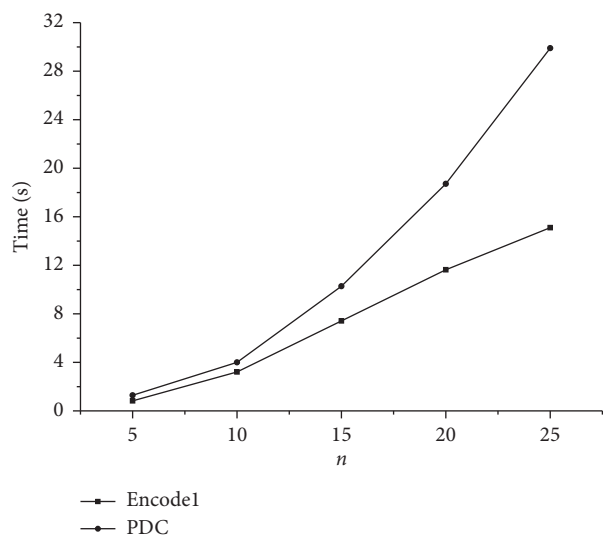

Figure 13: Time vs. 128 bit.

mainly concentrate on the scope of the sensing platform. However, the number of users in remote regions is often insufficient, leading the sensing platform to spend more time finding users. Secondly, the larger value of r will increase the length of the key. That means, the high security of PDC is based on increasing the running time. In this paper, considering the characteristics of MCS in remote regions, when $r = 64$ bit, PDC can both meet the security of user privacy and the requirement of runtime.

6. Conclusions

In this paper, efficient and location privacy-preserving schemes have been introduced for the different regional characteristics in MCS. To be specific, this proposed LDPF can present three advantages: (1) LDPF is suitable for different regional data and is able to prevent malicious participants; (2) it leverages powerful edge computing technology to avoid dependence on trusted platforms and realize distributed location privacy-preserving; and (3) it reduces the calculation error of the moving distance while protecting the privacies for both users and service providers. However, two shortcomings have also been revealed in the simulation experiments: Firstly, the privacy-preserving of participants should not be narrowed to the location privacy-preserving; secondly, the encoding method of users' moving distance should not be limited to the integer value, which will increase the loss of location information. In future work, we will expand the protection of user information (e.g., user data quality and user reputation), and a secure data coding method is going to be designed under noninteger values.

Data Availability

The data used to support this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that there are no conflicts of interest.

Acknowledgments

This research was funded by the Nature Science Foundation of China, grant number 61872104, and Fundamental Research Fund for a Central Universities in China, grant number 3072020CF0603.

References

[1] R. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, pp. 32–39, 2011.

[2] H. Yu, H. Pang, and M. Zhang, "Value-added effects of transit-oriented development: the impact of urban rail on commercial property values with consideration of spatial heterogeneity," *Papers in Regional Science*, vol. 97, no. 4, pp. 1375–1396, 2017.

[3] Y. Chon, N.D. Lane, F. Li, H. Cha, and F. Zhao, "Automatically characterizing places with opportunistic crowdsensing using smartphones," in *Proceedings of the 2012 ACM International Conference on Ubiquitous Computing*, Pittsburgh, PA, USA, 2012.

[4] "Gigwalk," 2010, http://www.gigwalk.com/.

[5] "Streetspotr," 2011, https://streetspotr.com/.

[6] Z. Wang, X. Pang, J. Hu et al., "When mobile crowdsensing meets privacy," *IEEE Communications Magazine*, vol. 57, no. 9, pp. 72–78, 2019.

[7] Z. Peng, S. Gao, B. Xiao, S. Guo, and Y. Yang, "CrowdGIS: updating digital maps via mobile crowdsensing," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 369–380, 2018.

[8] D. Wu, Z. Yang, B. Yang, R. Wang, and P. Zhang, "From centralized management to edge collaboration: a privacy-preserving task assignment framework for mobile crowd sensing," *IEEE Internet Things*, vol. 8, no. 6, pp. 4579–4589, 2020.

[9] I. J. Vergara-Laurens, D. Mendez-Chaves, and M. A. Labrador, "On the interactions between privacy-preserving, incentive, and inference mechanisms in participatory sensing systems," in *Proceedings of the 2013 International Conference on Network & System Security*, pp. 614–620, Madrid, Spain, 2013.

[10] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications, and Services (MobiSys 2003)*, pp. 31–42, San Francisco, CA, USA, 2003.

[11] 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.

[12] E. Habeeb, I. Kamel, and Z. Al Aghbari, "Privacy preserving KNN spatial query with voronoi neighbors," in *Proceedings of the 2019 Advances in Intelligent Systems and Computing*, pp. 900–910, Cham, Switzerland, 2019.

[13] H. Jadallah and Z. Al Aghbari, "Spatial cloaking for location-based queries in the cloud," *Journal of Ambient Intelligence and Humanized Computing*, vol. 10, no. 9, pp. 3339–3347, 2019.

[14] M. Zhang, L. Yang, S. He, M. Li, and J. Zhang, "Privacy-preserving data aggregation for mobile crowdsensing with externality: an auction approach," *IEEE/ACM Transactions on Networking*, vol. 29, no. 3, pp. 1046–1059, 2021.

[15] J. Wei, Y. Lin, X. Yao, and J. Zhang, "Differential privacy-based location protection in spatial crowdsourcing," *IEEE Transactions on Services Computing*, vol. 16, no. 4, pp. 934–949, 2019.

[16] L. Wang, D. Zhang, D. Yang, B. Y. Lim, X. Han, and X. Ma, "Sparse mobile crowdsensing with differential and distortion location privacy," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2735–2749, 2020.

[17] P. Sun, Z. Wang, Y. Feng et al., "Towards personalized privacy-preserving incentive for truth discovery in crowdsourced binary-choice question answering," in *Proceedings of the 2020 IEEE INFOCOM 2020—IEEE Conference on Computer Communications*, pp. 1133–1142, Toronto, Canada, 2020.

[18] H. Jin, L. Su, H. Xiao, and K. Nahrstedt, "Incentive mechanism for privacy-aware data aggregation in mobile crowd sensing systems," *IEEE/ACM Transactions on Networking*, vol. 26, no. 5, pp. 2019–2032, 2018.

[19] W. Xiang, "An efficient location privacy preserving model based on Geohash," in *Proceedings of the 2019 6th International Conference on Behavioral, Economic and Socio-Cultural Computing (Besc 2019)*, pp. 1–5, IEEE, Beijing, China, 2019.

[20] J. Shu, X. Liu, X. Jia, K. Yang, and R. H. Deng, "Anonymous privacy-preserving task matching in crowdsourcing," *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 3068–3078, 2018.

[21] D. Huang, Q. Gan, X. Wang, C. Huang, and Y. Lin, "Toward comparable homomorphic encryption for crowd-sensing network," *IACR Cryptology ePrint Archive*, vol. 2020, pp. 1–19, 2020.

[22] Y. Zheng, H. Duan, and C. Wang, "Learning the truth privately and confidently: encrypted confidence-aware truth discovery in mobile crowdsensing," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 10, pp. 2475–2489, 2018.

[23] J. Xiong, R. Ma, L. Chen et al., "A Personalized privacy protection framework for mobile crowdsensing in IIoT," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4231–4241, 2020.

[24] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: privacy-preserving multisubset data aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 462–471, 2018.

[25] C. Zhang, L. Zhu, C. Xu, K. Sharif, and X. Liu, "PPTDS: a privacy-preserving truth discovery scheme in crowd sensing systems," *Information Sciences*, vol. 484, pp. 183–196, 2019.

[26] P. Zhou, W. Chen, S. Ji, H. Jiang, L. Yu, and D. Wu, "Privacy-preserving online task allocation in edge-computing-enabled massive crowdsensing," *IEEE Internet Things*, vol. 33, no. 3, pp. 18–25, 2019.

[27] L. Ma, X. Liu, Q. Pei, and Y. Xiang, "Privacy-preserving reputation management for edge computing enhanced mobile crowdsensing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 786–799, 2018.

[28] B. Yang, D. Wu, and R. Wang, "CUE: an intelligent edge computing framework," *IEEE Network*, vol. 33, no. 3, pp. 18–25, 2019.

[29] Y. Huo, C. Yong, and Y. Lu, "Re-ADP: real-time data aggregation with adaptive ω-event differential privacy for fog computing," *Wireless Communications and Mobile Computing*, vol. 2018, Article ID 6285719, 13 pages, 2018.

[30] Z. Wang, J. Li, J. Hu, J. Ren, Z. Li, and Y. Li, "Towards privacy-preserving incentive for mobile crowdsensing under an untrusted platform," in *Proceedings of the IEEE INFOCOM 2019—IEEE Conference on Computer Communications*, Paris, France, 2019.

[31] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, "Security, privacy, and incentive provision for mobile crowd sensing systems," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 839–853, 2016.

[32] Z. Zhu, B. Chen, W. Liu, and Z. Yong, "A cost-quality beneficial cell selection approach for sparse mobile crowdsensing with diverse sensing costs," *IEEE Internet Things*, vol. 8, no. 5, pp. 3831–3850, 2020.

[33] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M'hamed, "Sparse mobile crowdsensing: challenges and opportunities," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 161–167, 2016.

[34] M. Naor, "Bit commitment using pseudorandomness," *Journal of Cryptology*, vol. 4, no. 2, pp. 151–158, 1991.

[35] D. Dias, L. Henrique, and M. Kosmalski Costa, "CRAWDAD dataset coppe-ufrj/RioBuses (v. 2018-03-19)," 2018, https://crawdad.org/coppe-ufrj/RioBuses/ 20180319.

[36] P. Huang, X. Zhang, L. Guo, and M. Li, "Incentivizing crowdsensing-based noise monitoring with differentially-private locations," *IEEE Transactions on Mobile Computing*, vol. 20, no. 2, pp. 519–532, 2021.