Hindawi
Security and Communication Networks
Volume 2021, Article ID 2296386, 15 pages
https://doi.org/10.1155/2021/2296386

Research Article

A Secure Truth Discovery for Data Aggregation in Mobile Crowd Sensing

Taochun Wang,[1] **Chengmei Lv**,[1] **Chengtian Wang**,[1] **Fulong Chen**,[2] **and Yonglong Luo**[1]

[1]*School of Computer and Information, Anhui Normal University, Wuhu, Anhui 241003, China*
[2]*Anhui Provincial Key Laboratory of Network and Information Security, Wuhu, Anhui 241003, China*

Correspondence should be addressed to Taochun Wang; wangtc@nuaa.edu.cn

Received 12 April 2021; Revised 15 May 2021; Accepted 10 June 2021; Published 25 June 2021

Academic Editor: Jinbo Xiong

Copyright © 2021 Taochun Wang 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.

With the rapid development of portable mobile devices, mobile crowd sensing systems (MCS) have been widely studied. However, the sensing data provided by participants in MCS applications is always unreliable, which affects the service quality of the system, and the truth discovery technology can effectively obtain true values from the data provided by multiple users. At the same time, privacy leaks also restrict users' enthusiasm for participating in the MCS. Based on this, our paper proposes a secure truth discovery for data aggregation in crowd sensing systems, STDDA, which iteratively calculates user weights and true values to obtain real object data. In order to protect the privacy of data, STDDA divides users into several clusters, and users in the clusters ensure the privacy of data by adding secret random numbers to the perceived data. At the same time, the cluster head node uses the secure sum protocol to obtain the aggregation result of the sense data and uploads it to the server so that the server cannot obtain the sense data and weight of individual users, further ensuring the privacy of the user's sense data and weight. In addition, using the truth discovery method, STDDA provides corresponding processing mechanisms for users' dynamic joining and exiting, which enhances the robustness of the system. Experimental results show that STDDA has the characteristics of high accuracy, low communication, and high security.

1. Introduction

With the rapid popularization of portable mobile sensing devices (such as smart phones and smart watches), which carry many sensors (gravity sensors, GPS, acceleration sensors, fingerprint, etc.), MCS has been extensively studied [1–4]. Participants with mobile sensing devices are encouraged to upload, analyze, and process their sensing data. After receiving the sensing data, the system is applied to all walks of life in society, such as transportation planning [5], environmental monitoring [6], and medical health [7]. For example, in MCS, participants upload the specific geographic location data of an object (such as supermarkets and schools) to the server, which analyzes and processes the data. And the obtained results are fed back to the corresponding application platforms. Then the platform utilizes these data to satisfy the needs of other participants, while enabling participants to quickly and accurately locate the specific location of the required objects, and to facilitate the activities of participants.

Due to the unprofessionalism and mobility of participants, the sensing data uploaded by participants is often unreliable or even conflicting data. Moreover, malicious participants may upload outdated or wrong data, which possibly have serious consequences for decision-making. For example, getting misleading geographic location information on the application platform, ordinary participants miss the best viewing time for tourist attractions. In addition, in many applications, data needs to be obtained from multiple data sources, and multiple data sources may also provide conflicting information. For example, a natural event that may be observed and recorded by multiple laboratories, or a patient record composed of multiple different hospitals,

makes these pieces of data or information conflict with each other. Therefore, the service quality of MCS can be guaranteed by filtering out the incorrect sensing data and identifying the real information. Elimination of abovementioned classification data conflict can be resolved by majority voting; that is, the most frequent information is considered to be the correct answer. For continuous data (e.g., height and weight), the mean/median value can be taken as the answer. The problem with voting or averaging method is that it assumes that the reliability of data from all sources is the same. Because normal participants continuously provide real and meaningful data, while malicious participants may generate biased or even false data, such traditional aggregation methods (such as voting and average) will not be able to get accurate aggregation results. In this case, in order to solve this problem, the truth discovery [7] approach, which is discovering truthful facts from unreliable or conflict information, has received extensive attention. The common principle of truth discovery is that the weight of the participant will be higher if the data provided by a participant is close to the aggregated result, and the reliability of the participant is higher and the data of participant will be counted more during the aggregation process if the participant's weight is higher. Based on this principle, the researchers have proposed multiple truth discovery methods to update the participant's weight and estimate the ground truth of each object.

However, the existing MCS faces serious privacy leakage issues which reduce the enthusiasm of participants. If the scheme based on truth discovery in MCS does not consider privacy, the server will obtain various types of information of participants, which may contain personal identity information and sensitive information such as phone number, home address, and health status. Attackers may take advantage of this sensitive information to conduct malicious deals. Based on this, our paper proposes a secure truth discovery for data aggregation in mobile crowd sensing (STDDA) in MCS. STDDA obtains final result by iteratively updating participant's weights and evaluating ground truth of each object. In order to protect data privacy, STDDA divides participant nodes into several clusters according to the location and number of participants. There are several participant nodes in each cluster which compute the corresponding secret random number according to the common parameters shared by the predecessor and successor nodes, while adding the secret random number to the sensing data to ensure data privacy. At the same time, the cluster head node uses secure sum protocol to fuse the sensing data in the cluster and sends it to the server which does corresponding storage and processing, so that the sensing data and weight of individual will not be known by the server, further ensuring the privacy of the participant's sensing data and weight. Using the truth discovery technology, STDDA gives the corresponding processing mechanism to the participant's failure exit and dynamic join, while enhancing the robustness of the system.

In summary, the contribution of our paper is summarized as follows:

(1) STDDA not only accurately compute the final aggregation result and estimated ground truth but also protects the data and weight information of the participants. In addition, it greatly improves the calculation speed and reduces the communication overhead of the participants.

(2) STDDA meets requests that participants fail to exit and join dynamically through cluster management and at the same time protects their data.

(3) Finally, extensive experiments were conducted in the MCS, and the results verified that STDDA can generate accurate aggregate results while protecting the privacy of participant data and weights.

The rest of this article is arranged as follows. In Section 2, we discuss the related work of this article. Then, we describe the preliminaries and give the details of our proposed algorithm in Sections 3 and 4. In Section 5, we conduct a series of experiments and performance evaluation to demonstrate the claims given in this article. Finally, we make a conclusion in this article in Section 6.

2. Related Work

Recently, truth discovery is an effective method to obtain truth values of each object from many sensing data, which has received more and more attention [8–17]. TruthFind [8] first proposed the problem of truth discovery, which provides a probabilistic approach based on the following assumptions: different data sources are independent, so the unreliable pieces of information that appear on different data sources should be different from each other. Then, AcuSim [9] is suitable for Bayesian analysis, and CRH [12] is suitable for processing heterogeneous data. However, all the abovementioned truth discovery methods ignore important privacy issues and may lead to the disclosure of personal sensitive information. For example, in order to deal with heterogeneous data, a CRH [12] way with high precision and accuracy is proposed, but this method only takes into account the problem of work efficiency, and the protection of data privacy of participants is not within the scope of its research.

Once the user's privacy is leaked, such as home address and office address, malicious attackers may use this information to attack users, which will directly threaten users' property and life safety. Xiong et al. [18] proposed an edge-assisted privacy-preserving raw data sharing framework. The framework uses additional secret sharing technology to encrypt the original data into two ciphertexts and constructs two types of security functions. Tian et al. [19] proposed a secure key management based on blockchain solution (BC-EKM). They use secure cluster formation algorithm and secure node movement algorithm to realize key management.

At the same time, this damages the interests of users and restricts users' enthusiasm for participating in MCS. Privacy protection is a key factor in expanding and motivating MCS applications. Representative ways for solving various privacy issues include (1) anonymization [20, 21], i.e., removing

participant's identifying information during communication, (2) data disturbing [22], i.e., adding noise during communication to interfere with the identification of participant data, (3) cryptography or secure multiparty computation [23–25], which uses various encryption algorithms to protect participants' sensitive data or denoting multiple participants collaborating and cooperating under the condition of mutual distrust and outputting the calculation results.

In order to ensure the security of the truth discovery technology, researchers have recently proposed various privacy-oriented truth discovery schemes. For example, Miao et al. [26] first proposed a secure truth discovery scheme PPTD using the threshold Paillier cryptosystem [24] to protect the privacy of the sensing data and weights of participants. However, due to the complexity of the threshold Paillier cryptosystem, the participants undertake huge communication and computational overheads. To reduce the communication overhead of participants and improve system efficiency, Miao et al. [27] used homomorphic encryption to further propose a lightweight truth discovery privacy protection scheme, while designing dual noncollusive servers to achieve a lightweight privacy protection truth discovery system L2-PPTD. However, the premise assumption of the system is that the server does not have any collusion with other participants. Once collusion occurs, the privacy of the participants will be revealed. Zheng et al. [28] proposed a new system architecture that enables an encrypted truth discovery method to be implemented in MCS. In this system, participants send encrypted sensing data to the cloud, while performing CATD (Confidence-Aware Truth Discovery) in the encrypted domain, and the final encrypted inference truth value is sent to the requester for decryption. Xu et al. [29] proposed an EPTD framework to solve the problem that all participants must be online. However, this framework does not solve the problem of dynamic participation of participants, and the practicality is lacking. Therefore, it is a challenge to propose a practical privacy protection solution based on truth discovery. This scheme can solve the failure and join of participants and reduce the communication overhead and cost of participants.

3. Preliminaries

3.1. Network Model. MCS mainly includes three parts: server S, participants, and cluster head nodes CH. Among them, S is responsible for managing all participants and storing and processing the sensing data uploaded by participants. Participants accept the sensing tasks issued by the platform, collect the sensing data, and process it accordingly. CH manages the participant nodes in the cluster and processes related data. At the same time it has the role of ordinary participants. In STDDA, according to the location and number of participants, the network is divided into multiple clusters by the server S. Each cluster is composed of a CH and multiple participants. The CH forms a ring of all nodes in the cluster; that is, each node has a unique predecessor and successor node. The network topology is shown in Figure 1. In each cluster, participants collect, process, and upload sensing data to CH. Then, CH aggregates all sensing data in the cluster and uploads them to S. Finally, S takes advantage of these data for various applications.

3.2. Truth Discovery. Truth discovery can effectively solve the problem of heterogeneous data information conflicts while extracting reliable information in MCS, where the object represents the description of the sensing task in the MCS, and the sensing data denotes the answers to the observations or questions collected by the participants. There are n participants, and a total of m objects require participants to collect data. x_j^i denotes the sensing data provided by the ith participant for the jth object. x_j^* represents the ground truth of jth object. w_i denotes the weight of ith participant, that is, the reliability of the ith participant. In addition, the goal of our article is to enable the server S to aggregate the sensing data of each participant $\{x_j^i\}_{i,j=1}^{m,n}$ and then accurately estimate ground truth of each object $\{x_j^*\}_{j=1}^{j}$, at the same time guaranteeing sensing data (i.e., $\{x_j^i\}_{i,j=1}^{m,n}$) and weights (i.e., $\{w_i\}_{i=1}^{n}$) are not known by other parties.

At present, existing truth discovery algorithms can generally be summarized in two procedures: weight update and truth evaluation. Before the weight is updated, the estimated ground truth of each object is first randomly initialized by the server S, and the weight and the estimated ground truth are updated iteratively until the convergence conditions are satisfied.

Weight update: it is assumed that the estimated ground truth of each object is fixed. Usually, the weight of each participant can be obtained as follows:

$$w_i = f\left(\sum_{j=1}^{m} d_{\text{ist}}\left(x_j^i, x_j^*\right)\right), \tag{1}$$

where f represents a monotonically decreasing function, and $d_{\text{ist}}(\cdot)$ represents the distance function between the sensing data and the estimated ground truth of participant. Since the CRH algorithm proposed has good practical performance, our paper uses the CRH algorithm to update the weight:

$$w_i = \log\left(\frac{\sum_{i=1}^{n}\sum_{j=1}^{m} d_{ist}\left(x_j^i, x_j^*\right)}{\sum_{j=1}^{m} d_{ist}\left(x_j^i, x_j^*\right)}\right), \tag{2}$$

where the distance function $d_{ist}(\cdot)$ is selected according to the application environment. This article considers the two most common data types (continuous data and categorical data) in the actual application of MCS.

In the continuous data (such as height and weight), the distance function $d_{ist}(\cdot)$ can be described as

$$d_{ist}\left(x_j^i, x_j^*\right) = \frac{\left(x_j^i - x_j^*\right)^2}{\text{std}_j}, \tag{3}$$

where std_j represents the standard deviation of the sensing data based on object j.

In the categorical data (such as gender and weather), this paper uses the vector $x_j^i = (0, \ldots, 1(q\text{th}), \ldots, 0)^T$ to represent

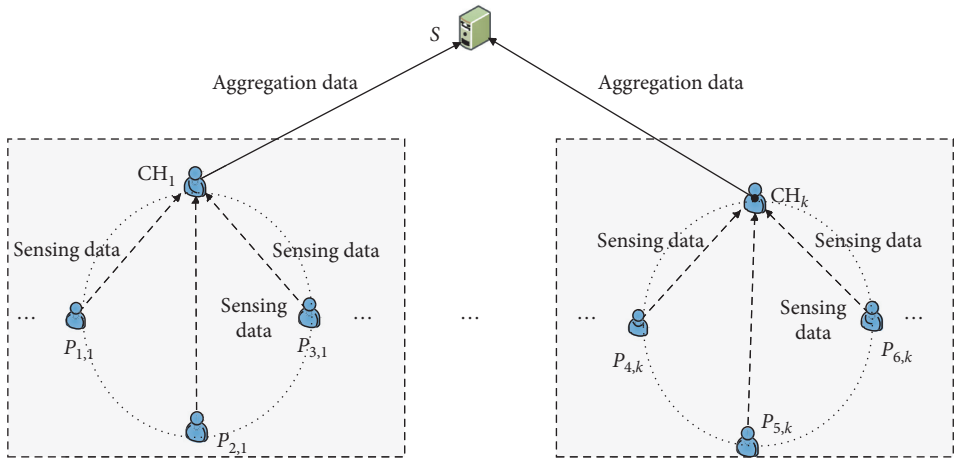

Figure 1: Topology of networks.

the qth choice of the ith participant based on the object j, and the calculation of $d_{ist}(\cdot)$ is

$$d_{ist}\left(x_j^i, x_j^*\right) = \left(x_j^i - x_j^*\right)^T. \tag{4}$$

Truth estimate: it is assumed that the weight of each participant is fixed. The ground truth of the jth object is estimated as

$$x_j^* \leftarrow \frac{\sum_{i=1}^n w_i x_j^i}{\sum_{i=1}^n w_i}. \tag{5}$$

Finally, the estimated ground truth of each object is obtained by iterating the above two procedures until the convergence condition is satisfied. The general truth discovery procedure can be described by Algorithm 1.

3.3. Attack Type. Attacks in MCS mainly include external attacks and internal attacks. (1) External attacks: since the information in MCS is transmitted wirelessly, the most common attack method is network eavesdropping to destroy data confidentiality. Our article assumes that the attacker can eavesdrop the entire network. (2) Internal attack: internal nodes or server S tries to obtain information to deduce the privacy information of other participants in MCS under the premise of completing the agreement. For example, the participant/server S tries to deduce the privacy information (such as location) of other participants on account of curiosity or interest. Our article adopts a semihonest model; that is, all parties of the MCS strictly implement the agreement, but the members retain the data obtained during the execution of the agreement and try to derive the privacy information of other members. Finally, our article, which can prevent collusion attacks (e.g., participants collude with S), uses data encryption to resist external attacks, so this article focuses on preventing internal attacks.

4. Security Truth Discovery

STDDA can accurately estimate the ground truth of each object based on the sensing data transmitted by participants. At the same time, in order to ensure the security of sensitive information, the sensing data and weight of participants are not obtained by other participants and server S. We first introduce the idea of STDDA algorithm, second describe the process of STDDA algorithm, and finally discuss and analyze the dynamics and security of the network.

4.1. STDDA Framework. In STDDA, participants are divided into several clusters by server S according to the location and number of participants. All processing is in units of clusters, and the process of each cluster is divided into three steps. (1) Initialization: S provides initial estimated ground truth of each object for each participant node. Then participant nodes compute the corresponding secret random numbers based on the common parameters shared by the predecessor and successor nodes. (2) Secure weight update: based on the sensing data and the initial ground truth provided by S, each participant calculates D_i, which is the sum of object distance function, while encrypting and transmitting it to CH. After obtaining all the ciphertext data in the cluster, CH uses the secure sum protocol to fuse ciphertext data to get D_C, which is the sum of object distance function of the cluster, and uploads it to S. Finally S aggregates all cluster data to obtain D, which is the sum of object distance function of all participants in the entire system, and then broadcasts D to all participants to update the weight. (3) Secure truth evaluation: participant P_i encrypts the weight W_i and WO_i, the product of weight and sensing data, and transmits them to CH. Then CH takes advantage of the secure sum protocol to get W_C, which is the sum of weight of cluster, and WO_C, which is the product of weight and sensing data of cluster. Next, CH encrypts and uploads them to S. At the same time, S aggregates W_C and WO_C to obtain W, the sum of weight of all participants, and WO, the sum of product of the weight and the sensing data of all participants in the entire system. Finally, the ground truth evaluation is performed until the convergence condition is satisfied; otherwise steps (2) and (3) are repeated. The procedure can be shown in Figure 2.

Input: Sensing Data for n participants: $\{x_j^i\}_{j,i=1}^{m,n}$
output: Estimation Truth for m objects: $\{x_j^*\}_{j=1}^{m}$
(1) Randomly initialize the ground truth x_j^* for each object;
(2) **repeat**
(3) **for** $i = 1, 2, \ldots, n$ **do**
(4) **for** $j = 1, 2, \ldots, m$ **do**
(5) Weight Update base on equation (2);
(6) Truth Estimate base on equation (5);
(7) **end for**
(8) **end for**
(9) **until** Convergence criterion is satisfied;
(10) **return** $\{x_j^*\}_{j=1}^{m}$;

Algorithm 1: Truth discovery process.

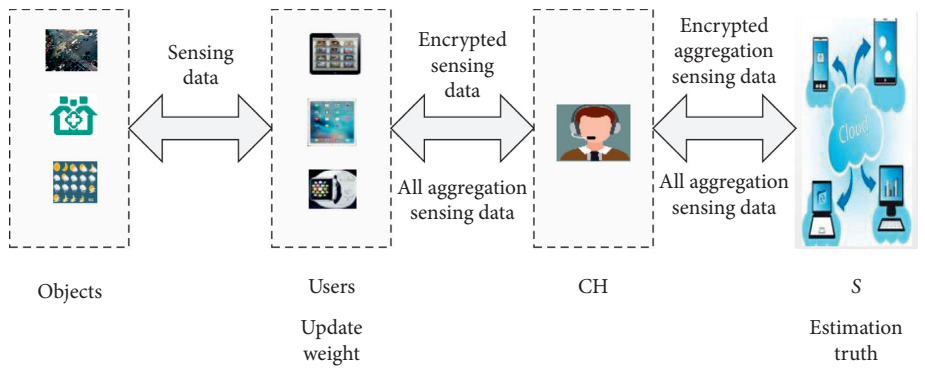

Figure 2: Flowchart of secure weight update and secure truth estimation.

4.2. STDDA Mechanism. In STDDA, it is assumed that n ($\{P_1, P_2, \ldots, P_n\}$) participants participate in MCS and collect sensing data of m objects. Participants are divided into t clusters by server S. There are $k(k = n/t$ and $k \geq 3)$ participants in each cluster, and some participant is randomly selected as the cluster head node (CH), and each cluster head node CH_i is assigned a secret key k_i. All participant nodes are formed into a ring; that is, each node has a unique precursor and successor node. For example, CH is P_1; that is, its precursor and successor nodes are P_k and P_2. P_i node precursor and successor nodes are P_{i-1} and P_{i+1}, respectively. On this basis, the following specifically explains the initialization of the algorithm, the secure weight update, and truth evaluation.

4.2.1. Initialization. The server S generates initialization ground truth of all objects $\{x_j^*\}_{j=1}^{m}$ and broadcasts them to each participant P_i, at the same time, generating two q-order multiplication groups G_1, G. p, q are large prime numbers with the same number of digits, and q is divided by $p - 1$. At the same time $g_1 = h^{(p-1/q)} \bmod p$ is the generator of g_1, where h is a random number. Moreover, $g_2 = g_1^p \bmod p^2$ is the generator of g_2.

Within each cluster, the node P_i randomly generates an integer $u_i \in Z$ and computes the common parameter $\beta_i = g_2{}^{u_i} \bmod p^2$. Then, β_i is shared with its predecessor and successor nodes P_{i-1} and P_{i+1}. After a round of exchanges, P_i calculates the secret random number $R_i = (g_2{}^{u_{i+1}}/g_2{}^{u_{i-1}})^{u_i} \bmod p^2$, as shown in Figure 3.

4.2.2. Secure Weight Update. The main process of secure weight update is divided into four parts. (1) Participants compute $D_i = \sum_{j=1}^{m} d_{ist}(x_j^{P_i}, x_j^*)$, which is the sum of object distance function. It is encrypted and transmitted to the cluster head node CH. (2) CH fuses the ciphertext data to get the sum of object distance function of the cluster D_C. It is encrypted and transmitted to the server S. (3) S gets D and broadcasts it to the participants. (4) All participants complete the weight update. When the participant P_i calculates the sum of object distance function $D_i = \sum_{j=1}^{m} d_{ist}(x_j^{P_i}, x_j^*)$ between the sensing data and the evaluation ground truth, the distance function $d_{ist}(\cdot)$ calculation methods of continuous data and categorical data are different. So, they need to be considered separately in the calculation. For categorical data, $d_{ist}(\cdot)$ is simply computed according to equation (4). For continuous data, the $d_{ist}(\cdot)$ is calculated according to equation (3), which needs to first compute the std of the sensing data, which is standard deviation. Since the std calculation is performed only once in the entire algorithm, it is not included in the iterative process. Therefore, this section first introduces the general steps (Step 1–Step 4) of all data types in the weight update and then introduces the

calculation process of the std_j in continuous data, which is the standard deviation of object j. See Step 5 for details.

Step 1 (each participant P_i encryption): P_i receives the evaluation ground truth sent by the server S (the first round is a random value generated by the S or a specific value). Then, P_i computes and encrypts D_i to form a ciphertext $E(D_i)$ as follows. At the same time, $E(D_i)$ is transmitted to the corresponding CH:

$$E(D_i) = (1 + p \times D_i) \times R_i \bmod p^2. \tag{6}$$

Step 2 (CH fusion): we can derive equation (7) from literature [30], where p represents a large prime number:

$$\begin{aligned} \prod_{i=1}^{n} (1+p)^{D_i} &= \prod_{i=1}^{n} (1 + p \times D_i) \\ &= \left(1 + p \sum_{i=1}^{n} D_i\right) \bmod p^2. \end{aligned} \tag{7}$$

After receiving $E(D_i)$ in the cluster (including its own ciphertext), CH performs the calculation as shown in equation (8), according to equation (7):

$$\begin{aligned} E_C^D &= \prod_{i=1}^{k} E(D_i) \bmod p^2 \\ &= \prod_{i=1}^{k} (1 + p \times D_i) \times R_i \bmod p^2 \\ &= \prod_{i=1}^{k} (1 + p \times D_i) \times \left(\frac{g_2^{u_{i+1}}}{g_2^{u_{i-1}}}\right)^{u_i} \bmod p^2 \\ &= \prod_{i=1}^{k} (1 + p \times D_i) \times g_2^{u_{i+1} \times u_i - u_{i-1} \times u_i} \bmod p^2 \\ &= \left(1 + p \sum_{i=1}^{k} D_i\right) \times g_2^{\sum_{i=1}^{k} u_{i+1} \times u_i - u_{i-1} \times u_i} \bmod p^2 \\ &= \left(1 + p \sum_{i=1}^{k} D_i\right) \bmod p^2, \end{aligned} \tag{8}$$

where $u_{k+1} = u_1$ and $u_0 = u_k$. In order to ensuring accurate results, p needs to be large enough. CH gets $D^{C_y} = \sum_{P_i \in C_y} D_i$, which is the sum of object distance function of k participants in the cluster, based on $(E_C^D - 1)/p = \sum_{i=1}^{k} D_i \bmod p$, while using the secret key k_i to form ciphertext $E_{k_i}(D^{C_y})$. Finally, the ciphertext is uploaded to the server S.

Step 3 (the server S aggregation): after receiving all the data uploaded by CH, S decrypts and aggregates the cluster data to obtain $D = \sum_{i=1}^{n} D_i = \sum_{y=1}^{t} D^{C_y}$, which is the sum of object distance function of n participants in the entire system, while broadcasting D to all participants for weight update.

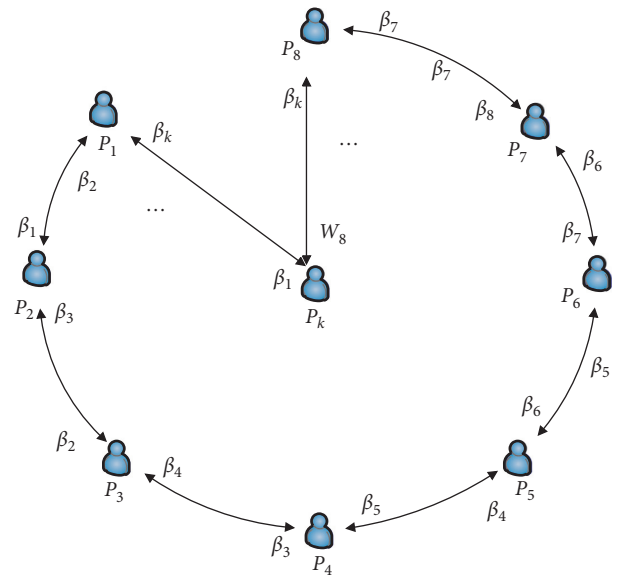

Figure 3: Secret random number in the setup.

Step 4 (weight update): after P_i receives the D sent by S, the weight W_i is updated according to (2) as

$$w_i = \log\left(\frac{D}{D_i}\right). \tag{9}$$

Step 5: the standard deviation std_j computing

① The ciphertext of P_i's sensing data based on the jth object is $E(x_j^i) = (1 + p \times x_j^i) \times R_i \bmod p^2$ and is transmitted to the CH of the cluster where P_i is located.

② After receiving $E(x_j^i)$ of all nodes in the cluster (including its own ciphertext), according to (7), the CH computes $\sum_{i=1}^{k} x_j^i$, which is the sum of the sensing data of k participants in the cluster based on the object j, and adopts the secret key k_i to form $E_{k_i}(\sum_{i=1}^{k} x_j^i)$, while uploading it to server S.

③ After receiving the data uploaded by CH, the server S decrypts and aggregates all cluster data to obtain $\sum_{i=1}^{n} x_j^i = \sum_{t=1}^{t} \sum_{i=1}^{k} x_j^i$, which is the sum of sensing data of n participants in the system based on object j. Then S calculates the average value $\overline{x_j} = \sum_{i=1}^{n} x_j^i / n$ based on the sensing data of object j and sends it to all participants.

④ After receiving $\overline{x_j}$, the participant P_i calculates $(x_j^i - \overline{x_j})^2$. It is encrypted to $E((x_j^i - \overline{x_j})^2) = (1 + p \times (x_j^i - \overline{x_j})^2) \times R_i \bmod p^2$ and transmitted to CH.

⑤ The CH calculates $\sum_{i=1}^{k} (x_j^i - \overline{x_j})^2$ of the k participants in the cluster and encrypts and uploads it to S. After receiving all the data $\text{SUM} = \sum_{i=1}^{n} (x_j^i - \overline{x_j})^2 = \sum_{t=1}^{t} \sum_{i=1}^{k} (x_j^i - \overline{x_j})^2$ uploaded by CH, S can obtain

and calculate the standard deviation $\text{std}_j = \sqrt{\text{SUM}/n}$ of participant's sensing data based on object j according to SUM.

4.2.3. Secure Truth Evaluation. The secure truth evaluation phase can be divided into three parts: (1) Participants compute WO_i, which is the product of weight and sensing data, and the weight W_i. They are transmitted to CH. (2) The ciphertexts of product and weight are fused by CH separately, while being encrypted and uploaded to the server S. (3) S obtains the sum of weight and product of all participates, respectively, and finally completes the truth evaluation. The specific process is show as follows.

Step 1 (each participant P_i encryption): P_i computes the WO_i, which is the product of weight and sensing data according to the obtained weight W_i, encrypts W_i and WO_i to form ciphertext $E(W_i) = (1 + p \times W_i) \times R_i \bmod p^2$ and $E(WO_i) = (1 + p \times WD_i) \cdot R_i \bmod p^2$, and then transmits them to the CH.

Step 2 (CH fusion): after receiving the ciphertext of all nodes in the cluster (including its own ciphertext), the CH performs calculations such as (10) and (11) in combination with (7):

$$\begin{aligned} E_C^W &= \prod_{i=1}^{k} E(W_i) \bmod p^2 \\ &= \prod_{i=1}^{k} (1 + p \times W_i) \times \left(\frac{g_2^{u_{i+1}}}{g_2^{u_{i-1}}} \right)^{u_i} \bmod p^2 \\ &= \left(1 + p \sum_{i=1}^{k} W_i \right) \bmod p^2, \end{aligned} \tag{10}$$

$$\begin{aligned} E_C^O &= \prod_{i=1}^{k} E(WO_i) \bmod p^2 \\ &= \left(1 + p \sum_{i=1}^{k} WO_i \right) \cdot \left(\frac{g_2^{u_{i+1}}}{g_2^{u_{i-1}}} \right)^{u_i} \bmod p^2 \\ &= \left(1 + p \sum_{i=1}^{k} WO_i \right) \bmod p^2. \end{aligned} \tag{11}$$

CH computes $E_C^W - 1/p$ and $E_C^W - 1/p$ to obtain $W^{C_y} = \sum_{P_i \in C_y} W_i$ and $WO^{C_y} = \sum_{P_i \in C_y} WO_i$, which are the sum of weight and product of the k participants in the cluster, and then uses the secret key k_i to form ciphertexts $E_{k_i}(W^{C_y})$ and $E_{k_i}(WO^{C_y})$, uploading them to the server S.

Step 3. Truth Evaluation. After receiving all the data uploaded by the CH, S decrypts and aggregates all the cluster data to obtain $W = \sum_{i=1}^{n} W_i = \sum_{y=1}^{t} W^{C_y}$, which is the sum of weight of n participants in the entire system, and $WO = \sum_{i=1}^{n} WO_i = \sum_{y=1}^{t} WO^{C_y}$, which is the sum of the product of the weight and the sensing data in the entire system. Finally the ground truth of each object is estimated based on (3) as

$$x_j^* \leftarrow \frac{WO}{W}. \tag{12}$$

The algorithm iteratively and securely updates participants' weight and estimates ground truth of object until the convergence condition is satisfied. The server S finally obtains the estimated ground truth of each object j as Algorithm 2, where steps 1–3 are the initialization procedure. Step7–10 are secure weight update process, and steps 11–13 are secure truth evaluation procedure.

4.3. Participant Dynamics. Because of the unprofessional nature of MCS participants and the characteristics of wireless transmission, it is often the case that participants are often (temporarily) invalid or newly join. In order to increase the robustness of the system, STDDA gives the corresponding processing mechanism which solves the failure exit or dynamic join of participant nodes.

4.3.1. Node Join. In order to encourage users to participate in MCS, STDDA allows new nodes to participate in the system and enhances the usability of the system. When the node P_j wants to join the MCS system, the node P_j first sends a join request message to the server S and S verifies its identity and determines whether the number of cluster nodes is less than the upper limit k. If it exists, select the cluster C_y according to the number of nodes in the cluster and the position of P_j and then forward the request message to the cluster head node CH_y. After CH_y receives the message, CH_y randomly informs two consecutive nodes in the cluster C_y (without loss of generality, such as nodes P_i, P_{i+1}) as the predecessor and successor nodes of P_j. At the same time, the nodes P_i, P_{i+1} and P_j update the public parameters ($\beta_i^{C_y}$, $\beta_{i+1}^{C_y}$, $\beta_j^{C_y}$) and secret random numbers ($R_i^{C_y}$, $R_{i+1}^{C_y}$, $R_j^{C_y}$). After the above work is completed, P_j will participate in the next truth discovery process. If the number of nodes in the existing cluster reaches the upper limit (=k), the server randomly selects the cluster C_y and randomly selects a ($2 \le a < k$) nodes in the cluster to establish a new cluster N_y with the newly added node. Updating the public parameters and secret random numbers are added to the next truth discovery process. The procedure can be described by Algorithm 3.

4.3.2. Node Invalid. When the node P_j fails to transmit data normally due to its own aspiration or software and hardware problems, STDDA needs to perform invalidation processing on the node P_j. This section considers two situations of node failure:

① Active failure: the node sends a leave request message to the server S before the node fails and applies to leave the cluster C_y. If the number of nodes of the

Input: n participants, m objects, sensing data for n participants base on m objects: $\{x_j^i\}_{j,i=1}^{m,n}$
output: Estimation ground truths for m objects: $\{x_j^*\}_{j=1}^{m}$

(1) Server S randomly initializes the estimated ground truth x_j^* for each object and sends to n participants;
(2) P_i randomly produces a integer $u_i \in Z_p$ and calculates the public parameters W_i, while sharing W_i with the precursor and successor nodes;
(3) After a round of swapping, P_i computes secret random number R_i
(4) **repeat**
(5) **for** $i = 1, 2, \ldots, n$ **do**
(6) **for** $j = 1, 2, \ldots, m$ **do**
(7) P_i calculates $D_i = \sum_{j=1}^{m} d_{ist}(x_j^{P_i}, x_j^*)$, then encrypting them forms ciphertext $E(D_i) = (1 + p \times D_i) \times R_i \bmod p^2$ and sending $E(D_i)$ to CH;
(8) CH fuses E_C^D, which is transmitted by the P_i in the cluster based on the secure sum protocol, to obtain $D^{C_y} = \sum_{P_i \in C_y} D_i$, and uploads it as ciphertext to S by using the secret key k_i;
(9) S decrypts and aggregates all the cluster data to obtain $D = \sum_{i=1}^{n} D_i = \sum_{y=1}^{t} D^{C_y}$ and sends them to P_i;
(10) After receiving D sent by S, P_i update the W_i according to equation (9);
(11) P_i calculates ciphertext $E(W_i) = (1 + p \times W_i) \times R_i \bmod p^2$ with $E(WO_i) = (1 + p \times WD_i) \cdot R_i \bmod p^2$ respectively and sends them to CH;
(12) CH fuses E_C^W and E_C^O based on the secure sum protocol to obtain $W^{C_y} = \sum_{P_i \in C_y} W_i$ with $WO^{C_y} = \sum_{P_i \in C_y} WO_i$, while uploading them as ciphertext to S by using the secret key k_i;
(13) S decrypts and aggregates all the cluster data to obtain $W = \sum_{i=1}^{n} W_i = \sum_{y=1}^{t} W^{C_y}$ with $WO = \sum_{i=1}^{n} WO_i = \sum_{y=1}^{t} WO^{C_y}$, and estimates the ground truths for m objects according to equation (12);
(14) **end for**
(15) **end for**
(16) **until** Convergence criterion is satisfied;
(17) **return** $\{x_j^*\}_{j=1}^{m}$;

Algorithm 2: Truth discovery process.

(1) Denoting k_{C_y} is the number of nodes in the cluster C_y;
(2) $P_j \longrightarrow S$; //P_j sends a request to join message to server S
(3) if ($\exists k_{C_y} < k$)
(4) S selects C_y;
(5) $S \longrightarrow \mathrm{CH}_y$; //$S$ forwards the join request to the cluster head node CH_y
(6) $\mathrm{CH}_y \longrightarrow P_i$;
(7) $\mathrm{CH}_y \longrightarrow P_{i+1}$;
(8) Denoting k_{C_y} is the number of nodes in the cluster C_y;
(9) $P_j \longrightarrow S$; //P_j sends a request to join message to server S
(10) if ($\exists k_{C_y} < k$)
(11) S selects C_y;
(12) $S \longrightarrow \mathrm{CH}_y$; //$S$ forwards the join request to the cluster head node CH_y
(13) $\mathrm{CH}_y \longrightarrow P_i$;
(14) $\mathrm{CH}_y \longrightarrow P_{i+1}$;
(15) $u_j = \mathrm{random}()$, $u_j \in Z_p$; //P_j randomly generates an integer
(16) $\beta_i^{C_y}, \beta_j^{C_y}, \beta_{i+1}^{C_y}$; //updating the public parameters
(17) $R_i^{C_y}, R_j^{C_y}, R_{i+1}^{C_y}$; //updating secret random numbers
(18) else
(19) establish a new cluster N_y;
(20) $N_y : \beta_{N_y}$;
(21) $N_y : R_j^{N_y}$;
(22) end if

Algorithm 3: Node join.

cluster C_y after P_j leaves is less than 3, the cluster is disbanded. And the remaining nodes are added to other clusters according to Algorithm 3. If the number of nodes in the cluster C_y after P_j leaves is greater than 3, the cluster head node CH_y notifies P_j's predecessor node P_{j-1} and successor node P_{j+1} to update the public parameters and secret random numbers, while processing to the next iteration.

② Passive failure: node P_j has sent relevant data, but the phenomenon of data loss occurs during the transmission. That is, the receiver has not received the message sent by P_j within the specified time. STDDA adopts a fast retransmission mechanism to solve this type of passive failure problem. Its main idea is that when the receiver receives every piece of data, it needs to reply with an acknowledgement ACK (value 1). When the receiver does not receive the data within the specified time, it sends a redundant ACK (value 0) to the node. STDDA selects 3 redundant ACKs as the threshold. Specifically, after the node P_j continuously receives 3 redundant ACKs, it immediately retransmits the data that has not been received by the other party. When the receiver has not received the sender's data within the specified time after sending 3 redundant ACKs, it is determined that the sender is passively invalid. The server can determine the number of remaining nodes in the cluster according to the node failure situation ①, while updating the public parameters and secret random numbers of the relevant nodes, so that the next iteration can be performed normally.

4.4. Security Analysis. We will conduct a theoretical analysis of the security of the STDDA algorithm in this section. Since attacks can be divided into external attacks and internal attacks according to the source in MCS, this chapter will conduct a theoretical analysis of security from both external and internal attacks.

4.4.1. External Attack. External attacks are attacks initiated by malicious nodes outside the network. The most common attack method is network eavesdropping. This article assumes that the attacker can conduct network-wide eavesdropping.

Theorem 1 *(under honest but curious setting). During the execution of the STDDA algorithm, the sensing data and weight of participant can resist theft attacks.*

Proof. In this article, we prove the participants' sensing data and weight against eavesdropping attacks from both the participants and the server. (1) Participants: In the secure weight update procedure, since the transmitted sensing data is encrypted by participants, the external attacker eavesdrops to obtain the encrypted ciphertext $E(D_i) = (1 + p \times D_i) \times R_i \bmod p_2$, so the attacker must infer the large prime number p and the secret random number R_i to get the plaintext D_i. However, the secret random number R_i is only known by the participant, so the attacker cannot eavesdrop on the ciphertext (D_i) to infer the plaintext D_i. Similarly, in the secure truth evaluation procedure, the transmitted weight is encrypted by participants, and the attacker cannot get the plaintext of weight. In addition, in order to further increase data privacy, participants update the secret random number R_i after N rounds of transmission. (2) Server: In the secure weight update procedure, the attacker eavesdrops on the sum of the object distance D ($D = \sum_{i=1}^{n} D_i = \sum_{t=1}^{t} \sum_{i=1}^{k} D_i$) of n participants transmitted by the server. Because D is aggregated data, the attacker cannot determine D is obtained by fusion of which nodes; that is, the sensing data of any node cannot be derived. In summary, the participant's sensing data and weight can prevent external eavesdropping attacks. □

4.4.2. Internal Attack. Internal attack refers to internal participants/server S or participants and S colluding to derive the sensing data and weight of other nodes.

Theorem 2 *(under honest but curious setting). During the execution of the STDDA algorithm, the sensing data and weight of participant can resist internal attacks.*

Proof. Internal attacks that derive the sensing data and weight of participants can be attributed to three types: participants, servers, and participants and servers colluding. (1) When an internal attacker is a participant: Because the transmitted sensing data and weight are encrypted by the target node in the cluster which uses the secret random number $R_i = (g_2^{u_{i+1}}/g_2^{u_{i-1}})^{u_i} \bmod p^2$, the attacker must obtain the secret random number Ri to obtain the plaintext of the target node. But the integer ui is only known by the target node. Therefore, the attacker cannot obtain the plaintext of sensing data and weight. (2) When the internal attacker is a server: the attacker can only get the aggregated plaintext data but cannot derive the plaintext data of a single node. (3) A collusion attack between participants and the server: When the server colludes with $(k-1)$ nodes in the cluster, the data of the target node will be leaked. Assuming that the probability of malicious nodes in the cluster is p, the probability of the target node leaking is related to the number of member nodes in the cluster, and its specific probability is $p^{k-1} \times (1-p) \times k$. So, when k is large, its probability is negligible. In summary, the participant's sensing data and weight can prevent internal attacks. □

5. Experiment and Performance Evaluation

5.1. Performance Evaluation. The performance evaluation of the truth discovery algorithm with privacy protection capability mainly includes the following: (1) whether the correct truth discovery results can be obtained; (2) whether the privacy of users can be guaranteed; (3) whether to rely on a trusted third party; (4) whether the user and the server (user) are required to not collude with each other; (5) whether to consider the dynamics of users in mobile crowd sensing. From Table 1, we can see that STDDA has advantages in the above five aspects.

5.2. Experiment Verification. In order to more realistically estimate the performance of STDDA, we design and develop a privacy protection truth discovery APP and background processing system. The front-end experimental environment is a smartphone (Huawei, iPhone, etc.), the operating system is Android 9.0 and above, the running memory is 4 GB and

Table 1: Performance comparison with existing approaches.

Properties	CRH [12]	PPTD [27]	EPTD [29]	STDDA
Correct truth discovery results	Yes	Yes	Yes	Yes
Ensured privacy	No	Yes	Yes	Yes
Trusted third party	No	No	Yes	No
Anticollusion attack	No	No	Yes	Yes
Dynamic join and quit of participants	No	No	No	Yes

above, and the back-end environment is operating system Win7, CPU Intel Core i5, 16 GB RAM. In our experiment, 100 mobile smart devices are used to target objects (latitude, longitude, etc.) in 10 buildings (such as schools, supermarkets, and hotels) for data collection. The truth discovery processing result of the object in the building and the corresponding map location are displayed as red dots in Figure 4, where the red mark indicates the building collection result and the corresponding display location.

In addition, we also analyze the accuracy, convergence, computational overhead, and communication overhead of the algorithm. In order to more truly reflect the experimental results, each experiment below is repeated 10 times, and the experiment shows that the result is the average value of the experiment.

5.2.1. Accuracy. In this experiment, the accuracy of CRH [12], PPTD [27], and STDDA algorithm is measured by the mean of absolute error (MAE) and the root of mean squared error (RMSE). Since PPTD requires sensing data to be calculated in integers, it is necessary to introduce the parameter L to approximate the data by rounding method [27] when computing the MAE and RMSE of PPTD. Therefore we set $L = 106$. Figures 5(a) and 5(b) show the changes in the MAE and RMSE of the corresponding three algorithm longitudes as the number of participants increases. Figures 5(c) and 5(d), respectively, show the changes of MAE and RMSE of the latitude. From Figure 5, we can see that the accuracy of the STDDA is consistent with CRH, because the parameter L is introduced by PPTD, so the accuracy is lower.

Figure 4: The map display of the building.

5.2.2. Convergence. By setting 5 different initial estimated ground truth values x_j^* to verify the convergence of the STDDA algorithm, it can be seen from Figure 6 that, under different estimated ground truth, basically two iterations can achieve the convergence requirements and higher efficiency.

5.2.3. Computational Overhead. Under the same hardware environment, by experimenting with a different number of objects, we obtain the communication overhead (run time) of the weight update and truth evaluation. We will explain the running time of the weight update, truth evaluation, and the entire process. As the number of objects increases, the running time of STDDA's weight update and truth evaluation is shown in Figure 7. At the same time, Figure 8 shows the running time of STDDA, PPTD, and EPTD for different numbers of users. In the secure weight update procedure, the participant P_i needs to encrypt and decrypt the data twice, respectively, in PPTD. In EPTD, the user needs to perform the Diffie-Hellman key exchange protocol to obtain the public key, and the user needs to perform two encryption operations and one decryption operation, but in STDDA, P_i only needs to encrypt D_i, which is the sum of object distance function, to get $E(D_i)$, while CH only performs simple multiplication. In the secure truth evaluation procedure, the P_i needs to perform two encryption operations and one data decryption in PPTD. In EPTD, the user needs to negotiate a public key, and the user needs to perform two encryption operations and one decryption operation, which is the same as the weight update stage, but in STDDA, the participant P_i needs to perform two encryption operations on W_i and WO_i, and CH only performs multiplication operations. In summary, STDDA has the shortest running time, EPTD is the second, and PPTD is

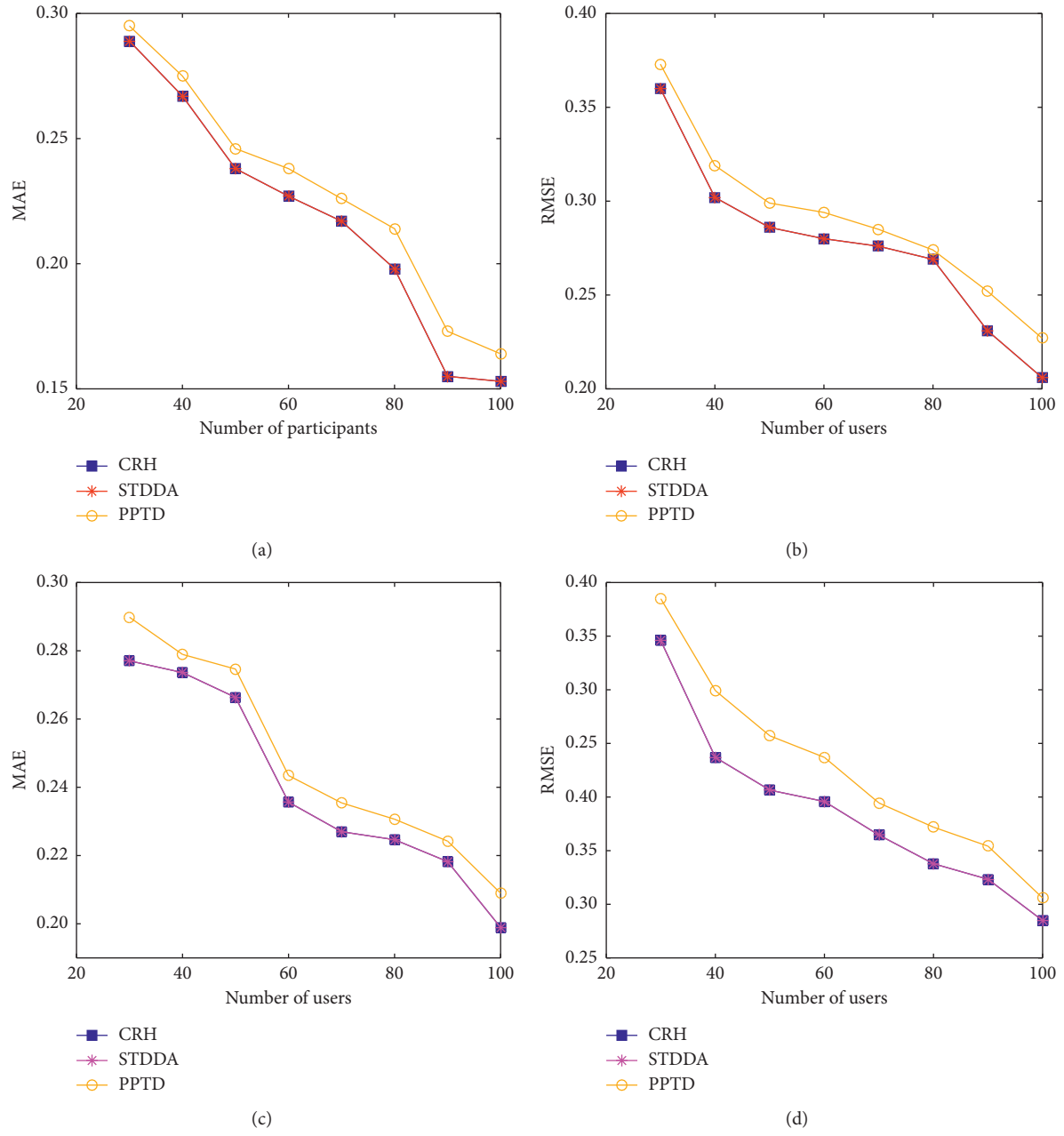

Figure 5: MAE and RMSE of object under different number of participants. (a) MAE. (b) RMSE. (c) MAE. (d) RMSE.

the longest. Figure 9 shows the comparison of total running time of the three algorithms.

5.2.4. Communication Overhead. The truth discovery algorithm mainly includes two procedures: weight update and truth evaluation. In this section, the communication overhead of the algorithm is obtained by analyzing the resource consumption of the participant nodes and the traffic between participant nodes and the CH in the two phases. Our article assumes that the length of all sent ciphertext data is u bits, and the number of iterations is a. (1) Secure weight update procedure: Participant node calculates the sum of object distance function D_i based on the sensing data and the initial ground truth provided by the server S, while encrypting and transmitting it to CH. So the time and space complexity are $O(1)$ and $O(|u|)$ ($|u|$ represents the length of the ciphertext) of a single participant node. And the total time and space complexity of this phase are $O(n)$ and $O(|u|)$. When each node P_i sends $E(D_i)$ to CH, the communication overhead is u. CH receives the ciphertext of all participants in the cluster, while fusing and sending it to the server S. And its traffic is $(k-1) \times u + u$ (each cluster has $(k-1)$ nodes and 1 cluster head node on average). (2) Secure truth evaluation procedure: The participant node encrypts the weight and the product of the weight and the sensing data and transmits it to CH. The time complexity of a single node is $O(1)$ and the space complexity is $O(|u|)$, so the time and space complexity

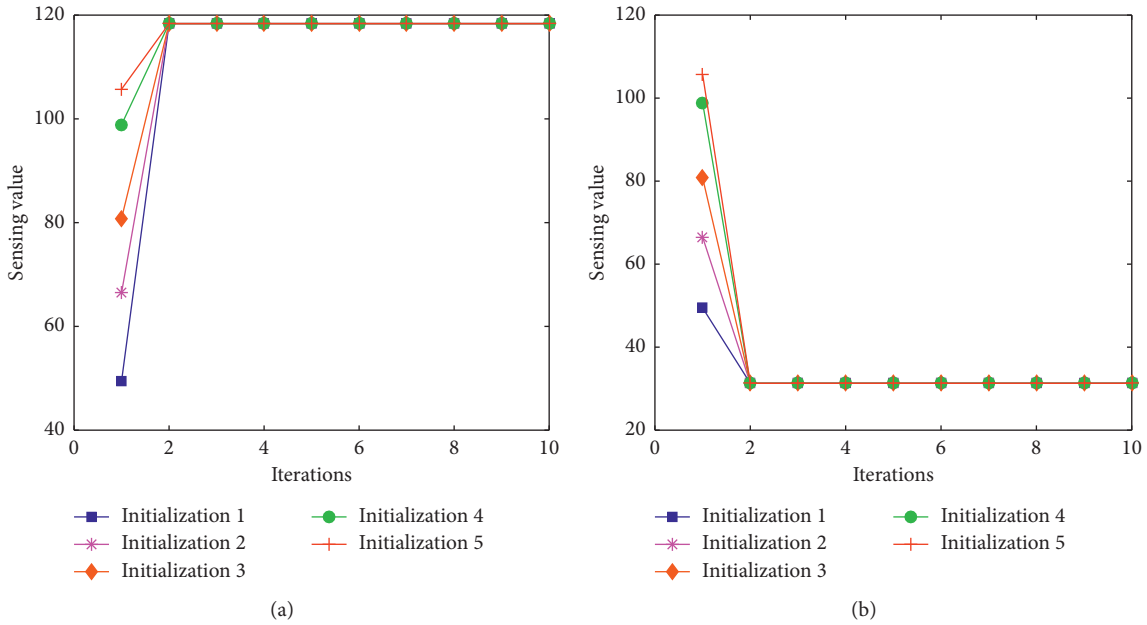

Figure 6: Comparison of convergence. (a) Longitude. (b) Latitude.

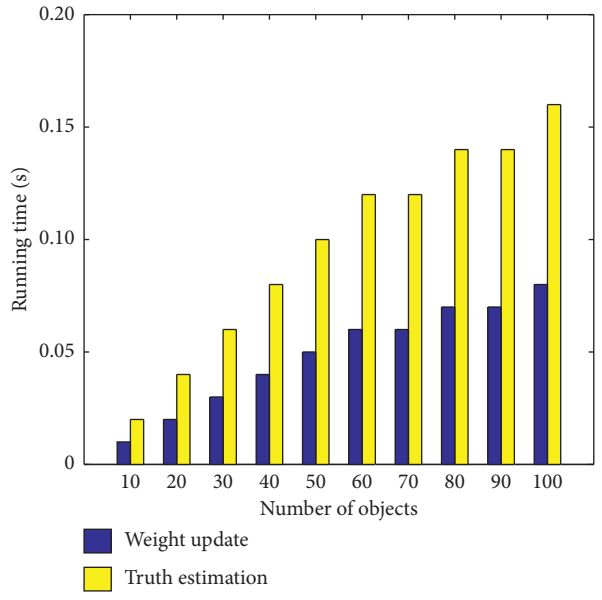

Figure 7: Running time of weight update and truth estimation under different number of objects.

of the STDDA algorithm in the secure truth evaluation phase are $O(n)$ and $O(n|u|)$, respectively. Each node P_i sends $E(W_i)$ and $E(WO_i)$ to CH, whose traffic is $2u$. CH receives $E(W_i)$ and $E(WO_i)$ from all participants in the cluster and fuses and uploads them to S, whose traffic is $(k-1)\times 2u + 2u$. Since the algorithm iterates a times on average, the algorithm traffic is shown in Table 2.

In PPTD, a single user needs to send ciphertext data three times and receive ciphertext data once. $(t'-1)$ users receive three times ciphertext and send three times plaintext data to the server. Therefore, the communication overhead of PPTD is $4\times n\times u\times a + 6\times u\times a\times (t'-1)$ in the whole process, where t' represents the number of users at the time of decryption. In EPTD, a single user needs to use Shamir's

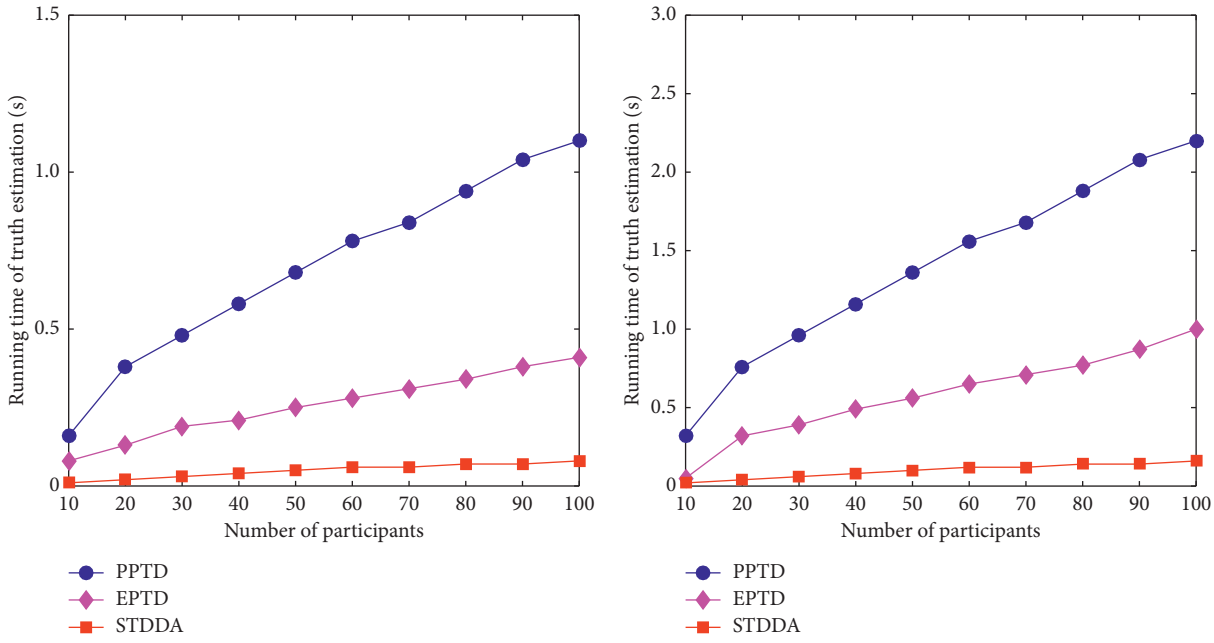

Figure 8: Comparison of running time. (a) The running time of weight update under different number of objects. (b) The running time of truth estimation under different number of objects.

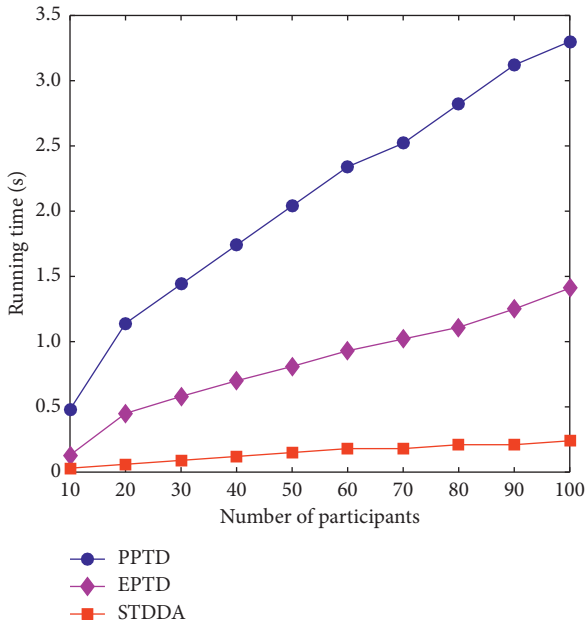

Figure 9: Comparison of total running time.

Table 2: Traffic overhead.

	Each P_i (b)	CH(b)	Total traffic
Secure weight update	$u \times a$	$k \times u \times a$	$(n-t) \times u \times a + t \times u \times a$
Secure truth estimation	$2u \times a$	$k \times 2u \times a$	$(n-t) \times 2u \times a + t \times 2u \times a$
Entire process	$3u \times a$	$k \times 3u \times a$	$3 \times n \times u \times a$

Table 3: Comparison of communication overhead.

Methods	Communication overhead
STDDA	$3 \times n \times u \times a$
PPTD [27]	$4 \times n \times u \times a + 6 \times u \times a \times (t' - 1)$
EPTD [29]	$4 \times n \times u \times a \times t'' + 7 \times u \times a \times t''$

(k, n) threshold key sharing protocol to distribute the private key four times to t'' users. A single user sends four ciphertexts to the server. At the same time, t'' users also need to send three times decryption key to the server again. Therefore, the communication overhead of EPTD in the whole process is $4 \times n \times u \times a \times t'' + 7 \times u \times a \times t''$, where t'' represents the number of users when uploading data or decrypting. Table 3 shows three comparisons of the total communication overhead, where $t' > 0$ and $t'' > 0$.

6. Conclusion

The STDDA algorithm proposed in this paper is used to solve the problem of truth discovery for privacy protection data fusion in MCS. Participants are divided into several clusters based on the number and position of participants, and the cluster head node is randomly assigned in each cluster. Then participants inside compute the corresponding secret random number according to the common parameters shared by the predecessor and successor nodes, ensuring the privacy of the data by adding secret random number to the sensing data. At the same time, the cluster head node uses the secure sum protocol to fuse the sensing data in the cluster, while encrypting and uploading it to the server, which decrypts and aggregates all cluster data to

obtain the sum of the sensing data of all participants in the entire system, and finally we iterate weight update and truth evaluation until convergence. So the server cannot obtain the sensing data and weight of a single participant, which further ensures the privacy of participants' sensing data and weight. In addition, using the truth discovery technology, the STDDA algorithm provides corresponding processing mechanisms for the dynamic join and invalid exit of participant nodes, enhancing the system robustness. Theoretical analysis shows that the STDDA algorithm can both defend against external attacks and resist internal attacks. A large number of experimental results prove that the STDDA algorithm has the characteristics of high security, high accuracy, and low communication. Besides, STDDA algorithm has great advantages over existing methods.

Data Availability

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

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the National Science Foundation of China (61972439, 61972438, and 61871412), Key Research and Development Projects in Anhui Province (202004a05020002), 2019 Key Project of Natural Science Research in Colleges and Universities of Anhui Provincial Department of Education (KJ2019A1164), the Anhui Normal University PhD Startup Fund (2018XJJ66), and the Anhui Normal University Innovation Fund (2018XJJ114).

References

[1] A. El, F. El, F. Ennaji, and M. Sadgal, "A mobile crowd sensing framework for suspect investigation: an objectivity analysis and de-identification approach," *Computer Science and Information Systems*, vol. 17, no. 1, pp. 253–269, 2020.

[2] J. Nan, X. Dong, Z. Jie et al., "Toward optimal participant decisions with voting-based incentive model for crowd sensing," *Journal of Information Science*, vol. 512, pp. 1–17, 2020.

[3] D. Wu, J. Liu, and Z. Yang, "Bilateral satisfaction aware participant selection with MEC for mobile crowd sensing," *IEEE Access*, vol. 8, Article ID 48110, 2020.

[4] J. Xiong, X. Chen, Q. Yang, L. Chen, and Z. Yao, "A task-oriented user selection incentive mechanism in edge-aided mobile crowdsensing," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2347–2360, 2020.

[5] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and DOA estimation based massive MIMO system," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8549–8560, 2018.

[6] N. Maisonneuve, M. Stevens, M. E. Niessen, and L. Steels, "Noisetube: measuring and mapping noise pollution with mobile phones," in *Proceedings of the Information Technologies in Environmental Engineering*, pp. 215–228, Springer, Berlin, Germany, April 2009.

[7] S. Vigneshwaran, K. Amit, N. Vikrant et al., "ConferenceSense: a case study of sensing public gatherings using participatory smartphones," in *Proceedings of the International Workshop on Pervasive Urban Crowdsensing Architecture and Applications*, Zürich, Switzerland, September 2013.

[8] X. Xiaoxin Yin, J. Jiawei Han, and P. S. Yu, "Truth discovery with multiple conflicting information providers on the web," *IEEE Transactions on Knowledge and Data Engineering*, vol. 20, no. 6, pp. 796–808, 2008.

[9] X. Li, X. L. Dong, K. Lyons, W. Meng, and D. Srivastava, "Truth finding on the deep web," *Proceedings of the VLDB Endowment*, vol. 6, no. 2, pp. 97–108, 2012.

[10] H. Jin, L. Su, and K. Nahrstedt, "Theseus: incentivizing truth discovery in mobile crowd sensing systems," in *Proceedings of the Mobihoc*, pp. 1–10, Chennai, India, July 2017.

[11] Z. Daniel Yue, B. Jose, Z. Yang et al., "Towards reliable missing truth discovery in online social media sensing applications," in *Proceedings of the IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pp. 143–150, Vancouver Canada, August 2018.

[12] L. Qi, L. Yaliang, G. Jing et al., "Resolving Conflicts in Heterogeneous data by truth discovery and source reliability estimation," in *Proceedings of the 2014 ACM SIGMOD International Conference on Management of Data*, pp. 1187–1198, Snowbird, UT, USA, June 2014.

[13] L. Qi, L. Yaliang, G. Jing et al., "A confidence-aware approach for truth discovery on long-tail data," *Proceedings of the VLDB Endowment*, vol. 8, no. 4, pp. 425–436, 2014.

[14] Y. Yi, B. Quan, and L. Qing, "A probabilistic model for truth discovery with object correlations," *Knowledge-Based Systems*, vol. 165, pp. 360–373, 2019.

[15] J. Yang, J. Wang, and W. P. Tay, "Using social network information in community-based Bayesian truth discovery," *IEEE Transactions on Signal and Information Processing Over Networks*, vol. 5, no. 3, pp. 525–537, 2019.

[16] H. Xiao, J. Gao, Q. Li et al., "Towards confidence interval estimation in truth discovery," *IEEE Transactions on Knowledge and Data Engineering*, vol. 31, no. 3, pp. 575–588, 2019.

[17] Z. Daniel, W. Dong, N. Vance et al., "On scalable and robust truth discovery in big data social media sensing applications," *IEEE Transactions on Big Data*, vol. 5, no. 2, pp. 195–208, 2019.

[18] J. Xiong, R. Bi, M. Zhao, J. Guo, and Q. Yang, "Edge-assisted privacy-preserving raw data sharing framework for connected autonomous vehicles," *IEEE Wireless Communications*, vol. 27, no. 3, pp. 24–30, 2020.

[19] Y. Tian, Z. Wang, J. Xiong, and J. Ma, "A blockchain-based secure key management scheme with trustworthiness in DWSNs," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 9, pp. 6193–6202, 2020.

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

[21] C. Chi-Yin, F. M. Mohamed, H. Tian et al., "A privacy-preserving location monitoring system for wireless sensor networks," *IEEE Transactions on Mobile Computing*, vol. 10, no. 1, pp. 94–107, 2010.

[22] H. Kargupta, S. Datta, W. Qi et al., "On the privacy preserving properties of random data perturbation techniques," in *Proceedings of the 3rd IEEE International Conference on Data Mining(ICDM'03)*, Melbourne, FL, USA, November 2003.

[23] Y. Shen, T. Zhang, Y. Wang, H. Wang, and X. Jiang, "Microthings: a generic iot architecture for flexible data

aggregation and scalable service cooperation," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 86–93, 2017.

[24] I. Damgård and M. Jurik, "A generalisation, a simplification and some applications of Paillier's probabilistic public-key system," in *Public Key Cryptography*Springer, Berlin, Germany, 2001.

[25] J. Liu, Y. Tian, Y. Zhou, Y. Xiao, and N. Ansari, "Privacy preserving distributed data mining based on secure multi-party computation," *Computer Communications*, vol. 153, pp. 208–216, 2020.

[26] C. Miao, W. Jiang, L. Su et al., "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, pp. 183–196, Seoul Republic of Korea, November 2015.

[27] C. Miao, W. Jiang, L. Su et al., "Privacy-preserving truth discovery in crowd sensing systems," *ACM Transactions on Sensor Networks*, vol. 15, no. 1, pp. 9–32, 2019.

[28] 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.

[29] G. Xu, H. Li, S. Liu, M. Wen, and R. Lu, "Efficient and privacy-preserving truth discovery in mobile crowd sensing systems," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 3854–3865, 2019.

[30] T. Jung, X.-Y. Li, and M. Wan, "Collusion-tolerable privacy-preserving sum and product calculation without secure channel," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 1, pp. 45–57, 2015.