Research Article

Privacy-Preserving Incentive Mechanism for Mobile Crowdsensing

Tao Wan ¹, Shixin Yue ¹ and Weichuan Liao ²

¹School of Information Engineer, East China Jiaotong University, Nanchang 330013, China

²School of Science, East China Jiaotong University, Nanchang 330013, China

Correspondence should be addressed to Tao Wan; wantao217@163.com

Received 1 May 2021; Revised 24 June 2021; Accepted 25 July 2021; Published 15 August 2021

Academic Editor: Jinbo Xiong

Copyright © 2021 Tao Wan 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.

Incentive mechanisms are crucial for motivating adequate users to provide reliable data in mobile crowdsensing (MCS) systems. However, the privacy leakage of most existing incentive mechanisms leads to users unwilling to participate in sensing tasks. In this paper, we propose a privacy-preserving incentive mechanism based on truth discovery. Specifically, we use the secure truth discovery scheme to calculate ground truth and the weight of users' data while protecting their privacy. Besides, to ensure the accuracy of the MCS results, a data eligibility assessment protocol is proposed to remove the sensing data of unreliable users before performing the truth discovery scheme. Finally, we distribute rewards to users based on their data quality. The analysis shows that our model can protect users' privacy and prevent the malicious behavior of users and task publishers. In addition, the experimental results demonstrate that our model has high performance, reasonable reward distribution, and robustness to users dropping out.

1. Introduction

As more and more sensors are integrated into human-carried mobile devices, such as GPS locators, gyroscopes, environmental sensors, and accelerometers, they can collect various types of data [1]. Therefore, the MCS system [2–4] can utilize the sensors equipped in mobile devices to collect sensing data and complete various sensing tasks [5], such as navigation service [6], traffic monitoring [7], indoor positioning [8], and environmental monitoring [9]. In general, the MCS system consists of three entities: a task requester, a sensing server, and participating users, as shown in Figure 1. The task requester publishes sensing tasks and pays awards for sensing results. The server recruits users according to the sensing task, processes the data from users, and sends the results to the task publisher. Users collect sensing data based on the requirements of the sensing task and get rewards.

In the practical MCS system, the sensing data collected by users are not always reliable [10, 11] due to various factors (such as poor sensor quality, lack of effort, and background noise). Therefore, the final result may be inaccurate if we treat the data provided by each user equally (e.g., averaging).

To solve this problem, truth discovery [12–14] has been widely concerned by industry and academia. The main idea of most truth discovery schemes is that the user will be given a higher weight (i.e., reliability) if the user's data are closer to the ground truth. Also, the data provided by a user will be counted more in the aggregation procedure if this user has a higher weight. Recently, a number of truth discovery methods [15] have been proposed to calculate user's weight and aggregated results based on this basic idea. But one problem with these methods is that users have to be online to interact with the server. Otherwise, the MCS system may fail and have to restart. Therefore, if we design a truth discovery scheme that allows users to exit, the MCS system can get stronger robustness.

The proper functioning of the truth discovery requires enough users and high-quality sensing data. Generally, the MCS system utilizes an incentive mechanism [16–18] to motivate sufficient users to participate in sensing tasks. However, because of monetary incentives, malicious users attempt to earn rewards with little or no effort. Although the truth discovery can assign low weight to malicious users, their continuous input of erroneous data can result in the

FIGURE 1: A general MCS system.

unavailability of the MCS system [19]. Consequently, the evaluation of data quality is critical to the MCS system. To improve data quality, users who provide incorrect data can be removed before sensing data get aggregated [20]. On the one hand, we can get more accurate aggregation results. On the other hand, users who provide eligible data can get more monetary rewards.

Although the incentive mechanism has been improved a lot, users' privacy protection remains inadequate. When users submit sensing data, their sensitive or private information [21–23] may be leaked, including identity privacy [24], location privacy, and data privacy. Also, privacy disclosure [25] will reduce users' willingness to participate in sensing tasks. Some incentive mechanism methods only consider the cost of users to collect sensing data but do not consider the potential cost of privacy disclosure. Recently, some researchers have designed privacy-preserving incentive mechanisms [26–28]. In [20], an incentive method is proposed to protect the user's identity and data privacy. Still, the user's sensing data will be submitted to the task publisher regardless of the privacy of the sensing data. In [29], the incentive mechanism is designed under the assumption of a trusted platform, which may not hold in practice since the platform itself might be attacked by hackers.

To address these issues, we propose a privacy-preserving incentive mechanism based on truth discovery, called PAID. In our PAID, the task publisher sets data constraints, such as time, location [30], budget [31], and sensing data. If the user does not collect the sensing data at the required time and location or sensing data are not in the qualified range, we believe that the user's sensing data are not credible (i.e., unqualified). After removing the unqualified user's data, the qualified user's sensing data will be submitted to the server to calculate the ground truth and weight. We also design a secure truth discovery scheme, which uses secret sharing technology and key agreement protocol and can still work when some users drop out. Moreover, our truth discovery can ensure that other parties cannot obtain users' sensing data except users themselves. Finally, we calculate every user's data quality according to the weight and distribute the reward.

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

- (i) We introduce a privacy-preserving interval judgment scheme to remove users who provide unreliable data before performing the truth discovery scheme. Removing unqualified users in advance can greatly improve the quality of the sensing data used in the truth discovery scheme, improve the accuracy of results, and save the reward budget.

- (ii) We introduce a secure truth discovery scheme so that our incentive mechanism model can obtain the ground truth and the weight of each user's data while protecting the user's privacy. Then, we design a reasonable reward distribution scheme based on the data weight of users. Moreover, our incentive mechanism model can allow users to drop out at any time.

- (iii) Analysis shows that our model is secure. Also, experimental results demonstrate that our model has high performance and can achieve reasonable reward distribution.

The remainder of this paper is organized as follows. In Section 2, we describe the problem statement. In Sections 3 and 4, we introduce cryptography primitives and intuitive technology in our model. Then, we discuss PAID in detail in Section 5. Next, Sections 6 and 7 carry out the analysis and performance evaluation. Finally, we discuss the related work and conclude the paper in Sections 8 and 9.

2. Problem Statement

In this section, we introduce the background of truth discovery and our system model. Then, we describe the threat model and our design goals. Table 1 summarizes the main notations in this paper.

2.1. Truth Discovery. Truth discovery [32] is widely used in the MCS system to solve the conflicts between sensing data collected from multiple sources. Although the methods of estimating weights and calculating ground truth are different, their general processes are similar. Specifically, truth discovery initializes a random ground truth and then iteratively updates the weight and ground truth until convergence.

2.1.1. Weight Update. Suppose that the ground truth of the object is fixed. If the user's sensing data are close to the ground truth, a higher weight should be assigned to the user. The weight w_i of each user u_i can be iteratively updated as follows:

$$w_i = \log \left(\frac{\sum_{i'=1}^{|U|} d_{ist}(x_{i'}, x^*)}{d_{ist}(x_i, x^*)} \right), \quad (1)$$

where $d_{ist}(x_i, x^*)$ is a distance function and $d_{ist}(x_i, x^*) = (x_i - x^*)^2$. We use U to represent the set of users, and $|U|$ is the number of users in the set U . The sensing data collected by the user u_i are denoted as x_i , in which i is the number of u_i , and x^* is the estimated ground truth.

2.1.2. Truth Update. Similarly, we assume that the weight w_i of each user u_i is fixed. Then, we can calculate the ground truth x^* as follows:

TABLE 1: Summary of notations.

Notations	Description
$\mathcal{T}$	A sensing task
TP	A task publisher
$\mathcal{S}$	A server
u_i	A user who performs a sensing task
U	A set of users
$ U $	Number of users in the set U
w_i	The weight of the user u_i
x^*	Ground truth of the sensing object
x_i	Sensing data collected by the user u_i
τ_i	Time for u_i to collect sensing data
$\hat{l}_i$	The longitude for u_i to collect sensing data
$\tilde{l}_i$	The latitude for u_i to collect sensing data
$\mathcal{D}_i$	The data submitted by the user u_i which is denoted by $(x_i, \tau_i, \hat{l}_i, \tilde{l}_i)$
$\mathcal{E}$	Eligibility rank for $\mathcal{D}_i$ is denoted by $(\mathcal{E}_x, \mathcal{E}_\tau, \mathcal{E}_{\hat{l}}, \mathcal{E}_{\tilde{l}})$
B	Budget constraint of a sensing task
$(pk_{\mathcal{T}}, sk_{\mathcal{T}})$	Key pair of public-key encryption
$\text{Enc}(P, pk_{\mathcal{T}})$	(IND-CPA) Public-key encryption function $C = \text{Enc}(P, pk_{\mathcal{T}})$, where P is a plaintext
$\text{Dec}(C, sk_{\mathcal{T}})$	Public-key decryption function $P = \text{Dec}(C, sk_{\mathcal{T}})$, where C is a ciphertext
$\text{SEnc}(P, k_i)$	Symmetric encryption function $C = \text{SEnc}(P, k_i)$, where k_i is the key
$\text{SDec}(C, k_i)$	Symmetric encryption function $P = \text{SDec}(C, k_i)$
π	A reward control parameter
q_i	The data quality of the user u_i , and $q_i = (w_i / \sum_{u_i \in U_6} w_i)$
$\bar{q}$	Mean of data quality, and $\bar{q} = (\sum_{u_i \in U_6} q_i / U_6) = (1/ U_6)$
p_i	Monetary reward of the user u_i , and $p_i = (B/ U_6) + \pi \cdot (q_i - \bar{q}) \geq 0$
c_i	The cost of the user u_i performing a sensing task
ut_i	Utility of the user u_i

$$x^* = \frac{\sum_{i=1}^{|U|} w_i \cdot x_i}{\sum_{i=1}^{|U|} w_i}. \quad (2)$$

The final ground truth x^* is obtained by iteratively running the weight update and the truth update until the convergence condition is satisfied.

2.2. System Model. Similar to the general MCS system, our PAID comprises three entities: a task publisher (TP), a server ($\mathcal{S}$), and users. In our PAID, the TP publishes tasks and requirements to $\mathcal{S}$ and gets the ground truth of the object from $\mathcal{S}$. The server $\mathcal{S}$ recruits adequate users and removes the users who provide unqualified data. After receiving the sensing data of all users, $\mathcal{S}$ performs the truth discovery scheme and gets the ground truth and the weight of each user. To prevent the TP from refusing to pay the reward, we require the TP to prepay the reward to $\mathcal{S}$ as a guarantee. After getting the weight of each user, the server $\mathcal{S}$ calculates the data quality and distributes the rewards. Users collect sensing data and earn monetary rewards by providing qualified data. Moreover, our PAID can protect users' privacy of time, location, identity, and sensing data. Unlike general MCS models, in our PAID, the TP and $\mathcal{S}$ can only get the aggregated result instead of users' sensing data. Figure 2 shows the flow of our PAID. The specific process of our model is as follows.

- (1) *Task Publish.* The TP publishes a sensing task to $\mathcal{S}$, including sensing objects, data eligibility requirements, and budget.

FIGURE 2: System model of PAID.

- (2) *User Recruitment.* The server $\mathcal{S}$ broadcasts the sensing task and recruits participating users.
- (3) *Eligibility Assessment.* The server $\mathcal{S}$ judges whether every user's sensing data meet qualification requirements.
- (4) *Prepayment.* The TP prepays $\mathcal{S}$ monetary reward to avoid the denial of payment attack.
- (5) *Submission Notification.* The server $\mathcal{S}$ notifies qualified users to submit sensing data.

- (6) *Data Submission and Eligibility Confirmation.* Users submit the masked sensing data to $\mathcal{S}$. And the server $\mathcal{S}$ needs to confirm whether the submitted sensing data are qualified to prevent malicious users from tampering with the data.
- (7) *Deviation Elimination.* The server $\mathcal{S}$ removes users who tamper with their sensing data and eliminates the deviation of data aggregation caused by these dropped users.
- (8) *Secure Truth Discovery.* The server $\mathcal{S}$ calculates the ground truth and weight of each user by performing the security truth discovery scheme.
- (9) *Reward Distribution.* The server $\mathcal{S}$ calculates the data quality of each user and distributes the rewards.
- (10) *Task Completion.* The server $\mathcal{S}$ sends the ground truth of the sensing object to TP.

2.3. Threat Model. In this section, we mainly consider the potential threats from TP, the server $\mathcal{S}$, and users.

We suppose that TP is dishonest. After getting data from $\mathcal{S}$, TP may launch a *denial of payment attack* (DoP) and refuse to pay rewards.

The server $\mathcal{S}$ is considered as honest-but-curious [33, 34]. Specifically, the server $\mathcal{S}$ follows the agreement execution instructions, but it also attempts to spy on users' private data. In other words, the server $\mathcal{S}$ may launch *inference attacks* (IAs) on the users' private data.

We assume that users are untrusted. Some malicious users may provide erroneous data and launch a *data pollution attack* (DPA). Besides, untrusted users may forge multiple identities and initiate a *Sybil attack* (SA), to earn more monetary rewards.

2.4. Design Goals. In this section, we introduce the design goals of our PAID, which are divided into privacy and security goals and property goals.

The privacy goals can protect the user's private data, and the security goals can avoid malicious attacks. The details are as follows.

- (i) *Privacy Goals.* PAID can protect user's location privacy, data privacy, and identity privacy. Specifically, the location and sensing data of a user cannot be obtained by any other parties except the user himself. And users' real identities would not be disclosed when performing a sensing task.
- (ii) *Security Goals.* In our PAID, users can avoid the *denial of payment attack* (DoP) of TP. The server $\mathcal{S}$ cannot initiate an *inference attack* (IA) on users. The server $\mathcal{S}$ can resist the *data pollution attack* (DPA) launched by malicious users. And our PAID guarantees fairness by resisting the *Sybil attack* (SA).

Our PAID also requires the following property goals.

- (i) *Eligibility.* If users' data do not meet the eligibility requirements, they cannot pass the eligibility

assessment. In other words, the sensing data adopted by our PAID must be eligible.

- (ii) *Zero Knowledge.* When the server $\mathcal{S}$ assesses whether users' data meet the eligibility requirements, it cannot obtain the content of users' private data.
- (iii) *Payment Rationality.* Each user can get non-negative utility as long as the user provides qualified data.
- (iv) *Budget Rationality.* The total monetary reward paid by the TP does not exceed the budget constraint.

3. Preliminaries

In this section, we review the cryptographic primitives used in our PAID.

3.1. Secret Sharing. We use Shamir's t -out-of- N secret sharing protocol [35], which can split each user's secret s into N shares, where any t shares can be used to reconstruct s . Still, it is impossible to get any information about s if the shares obtained by attackers are less than t .

We assume that some integers can be identified with distinct elements in a finite field $\mathcal{F}$, where $\mathcal{F}$ is parameterized with a size of $l > 2^k$ (in which k is the security parameter). These integers can represent all users' IDs, and we use a symbol U to denote the set of users' IDs. Then, Shamir's secret sharing protocol consists of two steps as below.

- (i) $\text{Shamir.share}(s, t, U) \rightarrow \{(u_i, s_i)\}_{u_i \in U}$: the inputs of the sharing algorithm are a secret s , a threshold $t \leq |U|$, and a set U of N field elements denoting the users' ID, where $|U| = N$. It outputs a set of shares s_i , each of which is associated with its corresponding the user u_i .
- (ii) $\text{Shamir.recon}(\{(u_i, s_i)\}_{u_i \in \mathcal{M}}, t) \rightarrow s$: the inputs of the reconstruction algorithm are the shares corresponding to a subset $\mathcal{M} \subseteq U$ and a threshold t , where $t \leq |\mathcal{M}|$, and it outputs the secret s .

Correctness requires that $\forall s \in \mathcal{F}, \forall t, N$ with $1 \leq t \leq N$. If $(u_i, s_i)_{u_i \in U} \leftarrow \text{Shamir.share}(s, t, U)$, where $\mathcal{M} \subseteq U$ and $t \leq |\mathcal{M}|$, then $\text{Shamir.recon}(\{(u_i, s_i)\}_{u_i \in \mathcal{M}}, t) \rightarrow s$.

Security requires $\forall s, s' \in \mathcal{F}$ and any $\mathcal{M} \subseteq U$ with $t > |\mathcal{M}|$. We have

$$\begin{aligned} & \{(u_i, s_i)\}_{u_i \in U} \leftarrow \text{Shamir.share}(s, t, U): \{(u_i, s_i)\}_{u_i \in \mathcal{M}} \\ & \equiv \{(u_i, s_i)\}_{u_i \in U} \leftarrow \text{Shamir.share}(s', t, U): \{(u_i, s_i)\}_{u_i \in \mathcal{M}}, \end{aligned} \quad (3)$$

where " $\equiv$ " indicates that the two distributions are indistinguishable.

3.2. Key Agreement. We utilize the Diffie–Hellman key agreement called SIGMA [36] in our PAID to generate a session key between two users. Typically, SIGMA is described in three parts as follows.

- (i) $\text{KA.param}(k) \rightarrow (\mathbb{G}, g, q, H)$: the algorithm's input is a security parameter k . It samples a group $\mathbb{G}$ of prime order q , along with a generator g and a hash function H , where H is set as SHA-256 for practicability in our model.
- (ii) $\text{KA.gen}(\mathbb{G}, g, q, H) \rightarrow (x, g^x)$: the algorithm's inputs are a group $\mathbb{G}$ of prime order q , along with a generator g and a hash function H . It samples a random $x \leftarrow Z_q$ and g^x , where x and g^x will be marked as the secret key SK_i and the public key PK_i in the following sections.
- (iii) $\text{KA.agree}(\text{sign}_j(g^{x_i}, g^{x_j}), \text{MAC}_k(u_j), x_i, g^{x_i}, i, j) \rightarrow s_{i,j}$: the algorithm's inputs are the user u_i 's secret key x_i , the user u_j 's public key g^{x_j} , signed signature $\text{sign}_j(g^{x_i}, g^{x_j})$, and $\text{MAC}_k(u_j)$ from the user u_j , where k is used as the MAC key. It outputs a session key $s_{i,j}$ between user u_i and user u_j . For simplicity, we use $\text{KA.agree}(x_i, g^{x_j}) \rightarrow s_{i,j}$ to represent the above process in the following sections.

Correctness requires that $\text{KA.agree}(\text{SK}_i, \text{PK}_j) = \text{KA.agree}(\text{SK}_j, \text{PK}_i)$ for any private and public key generated by the users u_i and u_j if two users use the same parameters. Security requires that the shared key $s_{i,j}$ is indistinguishable from a uniformly random string for any adversary who is given public keys PK_i and PK_j (but do not have the corresponding secret keys SK_i and SK_j).

3.3. Paillier Cryptosystem. The Paillier cryptosystem [37] is a probabilistic public key cryptosystem. It consists of three parts as follows.

- (i) $\text{Paillier.gen}(N, g) \rightarrow (sk_p, pk_p)$: the key distribution algorithm inputs are a number N and $g \leftarrow Z_{N^2}^*$, where N is the product of two large primes p, q . It outputs a secret key sk_p and a public key pk_p , where pk_p is computed by (N, g) , and $sk_p = \text{lcm}(p-1, q-1)$.
- (ii) $\text{Paillier.enc}(m, pk_p) \rightarrow c$: the encryption algorithm inputs are a plaintext m (which $m < N$) and a public key pk_p . It outputs a ciphertext c .
- (iii) $\text{Paillier.dec}(c, sk_p) \rightarrow m$: the decryption algorithm inputs are a ciphertext c (which $c < N^2$) and a secret key sk_p . It outputs a plaintext m .

The Paillier cryptosystem has the property of homomorphic addition.

$$E_{pk}(a + b) = E_{pk}(a) \cdot E_{pk}(b) \pmod{N^2}. \quad (4)$$

We assume that E is an encryption function.

4. Technical Intuition

In this section, we first introduce how the interval judgment scheme can judge users' data eligibility while protecting users' privacy. Then, we notice that truth discovery mainly

involves the aggregation of multiple users' data in a secure manner. Therefore, we require that the server $\mathcal{S}$ only get the sum of users' input, not content. And we propose a double-masking scheme to achieve this goal.

4.1. Interval Judgment Scheme for Privacy Protection. In our PAID, we use the interval judgment scheme [38] based on the Paillier cryptosystem to determine the sensing data eligibility. Every user u_i provides sensing data x_i , and the server $\mathcal{S}$ provides a continuous integer interval $[y_1, y_2]$ ($y_1, y_2 \leftarrow Z^*$). The server $\mathcal{S}$ can judge whether the user u_i 's sensing data x_i meet the interval range $[y_1, y_2]$ without knowing the data x_i . The user u_i also cannot obtain any information about the integer interval. The scheme is divided into four steps as follows.

- (i) The user u_i gets $(pk_p, sk_p) \leftarrow \text{Paillier.gen}(N, g)$ and then u_i computes $E(x_i)$ using pk_p and sends it to $\mathcal{S}$.
- (ii) The server $\mathcal{S}$ picks two random numbers k, b ($k, b \leftarrow Z^*$) to construct a monotone increasing (or decreasing) function $f(x_i) = kx_i + b$. Then, the server $\mathcal{S}$ computes $f(y_1), f(y_2), c = E(x_i)^k E(b) = E(kx + b)$ and sends them to u_i .
- (iii) After receiving the information from the server $\mathcal{S}$, the user u_i gets $f(x_i) \leftarrow \text{Paillier.dec}(c, sk)$ and then compares the size of $f(y_1), f(y_2)$, and $f(x_i)$. Next, the message is sent to the server $\mathcal{S}$.
- (iv) After receiving the message from u_i , the server $\mathcal{S}$ judges whether $f(y_1) < f(x_i) < f(y_2)$. If so, we can know $x_i \in [y_1, y_2]$ because of the monotonicity of the function $f(x_i) = kx_i + b$, i.e., the user u_i passes the data eligibility assessment. Otherwise, the user u_i fails to pass the eligibility assessment of the server $\mathcal{S}$.

It should be noted that since the user u_i does not know the monotonicity of the function $f(x) = kx + b$, it is impossible to infer whether the data x_i are in the range of the interval $[y_1, y_2]$ from the size relationship. For simplicity, we formulate the above process as an interval judgment function denoted by $\text{ins}(x_i, y_1, y_2)$. If the user u_i passes the eligibility assessment of the server $\mathcal{S}$, $\text{ins}(x_i, y_1, y_2) = 1$; otherwise, $\text{ins}(x_i, y_1, y_2) = 0$.

4.2. One-Masking Scheme. Assume that all users are represented in sequence as integers $1, n$. And any pair of users (u_i, u_j) , $i < j$, agrees on a random value $r_{i,j}$. Let us add $r_{i,j}$ to the user u_i 's data x_i and subtract $r_{i,j}$ from the user u_j 's data x_j to mask all users' raw data. In other words, each user u_i computes as follows.

$$y_i = x_i + \sum_{u_j \in \mathcal{U}: i < j} r_{i,j} - \sum_{u_j \in \mathcal{U}: i > j} r_{j,i} \pmod{R}, \quad (5)$$

where we assume x_i and $\sum_{u_j \in \mathcal{U}} r_{i,j}$ are in Z_R with order R for simplicity.

Then, each user u_i submits y_i to the server $\mathcal{S}$, and $\mathcal{S}$ computes

$$\begin{aligned}
z &= \sum_{u_i \in U} y_i \\
&= \sum_{u_i \in U} \left(x_i + \sum_{u_j \in U: i < j} r_{i,j} - \sum_{u_j \in U: i > j} r_{j,i} \right) \quad (6) \\
&= \sum_{u_i \in U} x_i \pmod{R}.
\end{aligned}$$

However, this approach has two shortcomings. The first one is that every user u_i needs to exchange the value $r_{i,j}$ with all other users, which will result in quadratic communication overhead ($|U|^2$) if done naively. The second one is that the protocol will fail if any user u_i drops out since the server cannot eliminate the value $r_{i,j}$ associated with u_i in the final aggregated results z .

4.3. Double-Masking Scheme. To solve these security problems, we introduce a double-masking scheme [39, 40]. In the work [40], the double-masking scheme is used for privacy-preserving data aggregation. And the scheme in [40] can also protect location privacy and verify the aggregation results. In our model, location privacy protection is implemented by the interval judgment scheme, and our secure truth discovery will confirm the data consistency. The details of the double-masking scheme are as follows.

$$y_i = x_i + \text{PRG}(n_i) + \sum_{u_j \in U: i < j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U: i > j} \text{PRG}(r_{j,i}) \pmod{R}. \quad (7)$$

Note that an honest user will never reveal both kinds of shares of the same user to the server $\mathcal{S}$. During the recovery round, the server $\mathcal{S}$ can request either a share of $r_{i,j}$ or a share of n_i from each surviving user u_j . After gathering at least t shares of $r_{i,j}$ for all dropped users and t shares of n_i for all surviving users, the server $\mathcal{S}$ can eliminate the remaining masks to reveal the sum.

5. Our Proposed Scheme

In this section, we first provide an overview of our PAID. Then, we show the details of the three critical designs in our PAID, including eligibility assessment, truth discovery, and reward distribution. In the eligibility assessment stage, the server $\mathcal{S}$ judges whether users' sensing data meet the requirements of a sensing task. In the truth discovery stage, the server $\mathcal{S}$ can calculate each user's weight and the ground truth required by the sensing task without knowing their sensing data. In the reward distribution stage, the server $\mathcal{S}$ computes the quality of sensing data by each user's weight and then pays a reward to users.

5.1. Overview. For convenience, we introduce a simple case. We set up a sensing task $\mathcal{T}$ to collect the temperature of urban roads in the evening. There are range requirements for time, location, and sensing data (i.e., temperature). To be more precise, the time range is required to be 5–8 pm on

Every user u_i can get a session key $r_{i,j}$ with other user u_j by engaging the Diffie–Hellman key agreement after the server $\mathcal{S}$ broadcasts all of the Diffie–Hellman public keys. Then, we can utilize a pseudorandom generator (PRG) to reduce the high communication overhead by having the parties agree on a common seed instead of the whole mask $r_{i,j}$.

We use the threshold secret sharing scheme to solve the issue that users are not allowed to drop out. Every user u_i can send his secret shares to other users. Once some users cannot submit data in time, other users can recover masks associated with these users by submitting shares of these users' secrets to $\mathcal{S}$, as long as the number of dropped users is less than t (i.e., threshold of Shamir's secret sharing).

However, there is a problem that may lead to users' data leaked to $\mathcal{S}$. There is a scenario where a user u_i is very slow to send data to $\mathcal{S}$. The server $\mathcal{S}$ considers that the user u_i has dropped and asks for their shares of the user u_i 's secret from all other users. Then, the server receives the delayed data y_i after recovering u_i 's mask. At this time, the server $\mathcal{S}$ can remove all the masks $r_{i,j}$ and get the plaintext x_i .

To improve the scheme, we introduce an additional random seed n_i to mask the data. Specifically, each user u_i selects a random seed n_i on the round of generating $r_{i,j}$ and then creates and distributes shares of n_i to all other users during the secret sharing round. Now, users calculate y_i as follows:

February 3rd, the location range is required to be 12.45–12.55 E and 41.79–41.99 N, and the temperature requirement is 10–15°C. In our PAID, we consider the range requirement as the data eligibility requirement $\mathcal{E}$. The data $\mathcal{D}_i$ ($\mathcal{D}_i = (x_i, \tau_i, \hat{l}_i, \tilde{l}_i)$) collected by a user u_i meet the eligibility requirements $\mathcal{E}$, meaning that $10 \leq x_i \leq 15, 5 \leq \tau_i \leq 8, 12.45 \leq \hat{l}_i \leq 12.55, 41.79 \leq \tilde{l}_i \leq 41.99$. Since the data collected by mobile devices are usually rational numbers, in our PAID, we transform the eligible interval into an integer interval by moving the decimal point right. The sensing task $\mathcal{T}$ consists of three entities: a task publisher (TP), a server ($\mathcal{S}$), and users. And the specific steps are as follows.

Step 1 (Task Publish). The task publisher TP initializes a public key $pk_{\mathcal{T}}$ and a private key $sk_{\mathcal{T}}$, a reward control parameter π (π is a decimal number), a task budget B , the number of users N , and eligibility requirements $\mathcal{E}$ for a sensing task $\mathcal{T}$. The public key $pk_{\mathcal{T}}$ is used to encrypt the information that the server $\mathcal{S}$ needs to send to the TP, and the TP decrypts the ciphertext using the private key $sk_{\mathcal{T}}$. Then, the TP sends the information $\{\mathcal{T}, pk_{\mathcal{T}}, \pi, N, B, \mathcal{E}\}$ to $\mathcal{S}$ as a task request.

Step 2 (User Recruitment). The server $\mathcal{S}$ broadcasts the sensing task information $\{\mathcal{T}, \pi, N, B\}$ and recruits N users who request to participate in the sensing task. Then, $\mathcal{S}$ generates a key pair $\{PK_{\mathcal{S}}^i, SK_{\mathcal{S}}^i\}$ using the key agreement scheme for every user u_i and sends $PK_{\mathcal{S}}^i$ to u_i .

Step 3 (Eligibility Assessment). Each user u_i confirms whether $c_i \leq (B - \pi/N)$, where c_i denotes the sensing cost of u_i , and the posted lowest reward is denoted as $(B - \pi/N)$. If $c_i \leq (B - \pi/N)$, and u_i starts the sensing task and collects the data $\mathcal{D}_i$. The user u_i then generates a key pair $\{PK_i, SK_i\}$ using the key agreement scheme and computes a session key $k_i \leftarrow \text{KA.agree}(SK_i, PK_{\mathcal{S}}^i)$ as u_i 's anonymous identity information. Then, the user u_i performs the interval judgment scheme $\text{ins}(\mathcal{D}_i, \mathcal{E})$ and sends the public key PK_i to $\mathcal{S}$. Specifically, $\text{ins}(\mathcal{D}_i, \mathcal{E})$ is divided into $\text{ins}(x_i, \mathcal{E})$, $\text{ins}(\tau_i, \mathcal{E})$, $\text{ins}(\hat{t}_i, \mathcal{E})$, $\text{ins}(\tilde{t}_i, \mathcal{E})$.

Step 4 (Prepayment). After recruiting N eligible users, the server $\mathcal{S}$ requests TP to prepay a budget reward B for the sensing task $\mathcal{T}$ to prevent the denial of payment attack. And the server $\mathcal{S}$ calculates the session key $k_i \leftarrow \text{KA.agree}(SK_{\mathcal{S}}^i, PK_i)$ with the eligible user u_i .

Step 5 (Submission Notification). After getting the budget reward B , the server $\mathcal{S}$ informs the eligible user u_i ($1 \leq i \leq N$) to submit data.

Step 6 (Data Submission and Eligibility Confirmation). After receiving the submission notification, each user u_i performs double-masking scheme to mask the sensing data x_i and get y_i and, at the same time, executes eligibility confirmation $\text{ins}(\mathcal{D}_i, \mathcal{E})$ to prevent malicious users from modifying data. Then, u_i encrypts the data y_i using the symmetric encryption algorithm and sends the ciphertext $\text{SEnc}(y_i, k_i)$ to $\mathcal{S}$. The session key k_i is the key of symmetric encryption.

Step 7 (Deviation Elimination). For users who tamper with data during data submission, the server $\mathcal{S}$ regards them as dropped users and discards their data. Then, $\mathcal{S}$ gets plaintext $\text{SDec}(\text{SEnc}(y_i, k_i), k_i)$ and requests seed n_i and the noise $r_{i,j}$ between the dropped user u_i and the surviving user u_j to eliminate the impact on the aggregate result.

Step 8 (Secure Truth Discovery). The server $\mathcal{S}$ computes the surviving user u_i 's weight w_i and the ground truth x^* of the sensing object utilizing the truth discovery algorithm. The detailed algorithm process will be introduced later.

Step 9 (Reward Distribution). The server $\mathcal{S}$ calculates the sensing data quality $q_i = (w_i / \sum_{i=1}^m w_i)$ of u_i , where $\sum_{i=1}^m q_i = 1$, m is the number of online users. Then, $\mathcal{S}$ pays a monetary reward $p_i = (B/m) + \pi \cdot (q_i - \bar{q})$ for u_i , where $\pi \cdot (q_i - \bar{q})$ denotes the payment parameter, $m \leq N$, and $1 \leq i \leq m$.

Step 10 (Task Completion). The server $\mathcal{S}$ encrypts the ground truth x^* using $pk_{\mathcal{T}}$ and sends $\text{Enc}(x^*, pk_{\mathcal{T}})$ to TP. And the TP can decrypt the data using $sk_{\mathcal{T}}$, i.e., $x^* = \text{Dec}(\text{Enc}(x^*, pk_{\mathcal{T}}), sk_{\mathcal{T}})$.

In our PAID, only users who passed the eligibility assessment and eligibility confirmation can obtain the monetary reward. Thus, users cannot cheat $\mathcal{S}$ to get a reward with unreliable data. We can also ensure the quality of the sensing data used by the truth discovery algorithm and obtain more accurate ground truth x^* . Moreover, since the TP pays the

task reward to $\mathcal{S}$ in advance and $\mathcal{S}$ will pay a reward to u_i according to the quality of u_i 's sensing data after the task is accomplished, the TP cannot refuse to pay the reward. Besides, $\mathcal{S}$ cannot get users' raw sensing data, time, and location information, which can protect the users' privacy. The anonymous identity of each user is determined by both the user and $\mathcal{S}$. $\mathcal{S}$ only assigns one random identity token to each user, so malicious users cannot forge multiple identities.

5.2. Eligibility Assessment. In our PAID, there are three benefits to the design of the eligibility assessment. First, it can prevent users who provide unreliable or erroneous sensing data from receiving monetary rewards, which avoids wasting budgets. Secondly, filtering out unqualified sensing data can improve the accuracy of the sensing task result. Thirdly, the data quality q_i of each user u_i is related to the sensing object's ground truth x^* , and inaccurate ground truth will lead to unfair incentives.

The process of eligibility assessment and eligibility confirmation is similar. The purpose of the eligibility assessment is to filter out unqualified users preliminarily. Thus, the unqualified users do not need to communicate with other users to perform the double-masking scheme, by which the communication overhead can be reduced. The eligibility confirmation is designed to prevent malicious users from altering the original qualified data. The detailed process of eligibility assessment and eligibility confirmation is as follows.

Step 1. Each user u_i initializes a key pair $(pk_{p_i}, sk_{p_i}) \leftarrow \text{Paillier.gen}(N, g)$. Then, u_i encrypts the sensing data $\mathcal{D}_i$ using pk_{p_i} and sends the ciphertext $E(\mathcal{D}_i)$ to $\mathcal{S}$. Generally, $E(\mathcal{D}_i)$ consists of four parts: $E(x_i)$, $E(\tau_i)$, $E(\hat{t}_i)$, and $E(\tilde{t}_i)$.

Step 2. After receiving $E(\mathcal{D}_i)$, the server $\mathcal{S}$ picks different random k, b ($k, b \leftarrow Z^*$) and constructs a monotone increasing (or decreasing) function $f(\mathcal{D}_i) = k\mathcal{D}_i + b$ for each value in the quadruples $(x_i, \tau_i, \hat{t}_i, \tilde{t}_i)$. The monotonicity of the four functions is inconsistent. For eligibility requirement interval $\mathcal{E}$ ($\mathcal{E} = \{\mathcal{E}_x, \mathcal{E}_\tau, \mathcal{E}_{\hat{t}}, \mathcal{E}_{\tilde{t}}\}$, $\mathcal{E}_x = [x_l, x_r]$, $\mathcal{E}_\tau = [\tau_l, \tau_r]$, $\mathcal{E}_{\hat{t}} = [\hat{t}_l, \hat{t}_r]$, $\mathcal{E}_{\tilde{t}} = [\tilde{t}_l, \tilde{t}_r]$), the server $\mathcal{S}$ calculates

$$\begin{cases} f(x_l), f(x_r), c_1 = E(x_l)^{k_1}, E(b_1) = E(k_1 x_l + b_1), \\ f(\tau_l), f(\tau_r), c_2 = E(\tau_l)^{k_2}, E(b_2) = E(k_2 \tau_l + b_2), \\ f(\hat{t}_l), f(\hat{t}_r), c_3 = E(\hat{t}_l)^{k_3}, E(b_3) = E(k_3 \hat{t}_l + b_3), \\ f(\tilde{t}_l), f(\tilde{t}_r), c_4 = E(\tilde{t}_l)^{k_4}, E(b_4) = E(k_4 \tilde{t}_l + b_4). \end{cases} \quad (8)$$

Then, $\mathcal{S}$ sends $\{f(\mathcal{E}_l), f(\mathcal{E}_r), c\}$ ($c = \{c_1, c_2, c_3, c_4\}$) to u_i . For convenience, we will not describe $\mathcal{D}_i$ and $\mathcal{E}$ separately in the following text.

Step 3. After receiving $\{f(\mathcal{E}_l), f(\mathcal{E}_r), c\}$ from $\mathcal{S}$, each user u_i gets $f(\mathcal{D}_i) \leftarrow \text{Paillier.dec}(c, sk_{p_i})$ and then

compares the size of $f(\mathcal{E}_l), f(\mathcal{E}_r), f(\mathcal{D}_i)$. Next, the size relationship is sent to $\mathcal{S}$.

Step 4. After the server $\mathcal{S}$ receives the information from u_i , if $f(\mathcal{E}_l) \leq f(\mathcal{D}_i) \leq f(\mathcal{E}_r)$, then $\mathcal{D}_i \in [\mathcal{E}_l, \mathcal{E}_r]$ because of the monotonicity of the functions. And $\mathcal{S}$ determines whether u_i passes the eligibility assessment. Otherwise, it fails.

Because users do not know the function's monotonicity, they cannot infer the size relationship between the qualified data and eligibility requirement. Therefore, we can think that malicious users have a very low probability of passing the eligibility assessment. Moreover, during the eligibility assessment, u_i cannot know the specific qualified interval. $\mathcal{S}$ also cannot get u_i 's sensing data, which can protect u_i 's privacy. The above process is represented by $\text{ins}(\mathcal{D}_i, \mathcal{E})$. If u_i passes the eligibility assessment, then $\text{ins}(\mathcal{D}_i, \mathcal{E}) = 1$. If not, $\text{ins}(\mathcal{D}_i, \mathcal{E}) = 0$.

5.3. Secure Truth Discovery. In the secure truth discovery scheme [15], data exchange is between users and the server $\mathcal{S}$. The user u_i needs to collect sensing data x_i , perform the double-masking scheme to mask the raw input data y_i ($y_i = d_{\text{ist}}(x_i - x^*)$), and then send the masked input data z_i to $\mathcal{S}$. The server $\mathcal{S}$ receives masked input data z_i from each user u_i and aggregates the input data of online users. Each user u_i can drop out at any time. As long as the number of surviving users is not less than the threshold t , $\mathcal{S}$ can eliminate the deviation caused by dropped users and restore the aggregation results. The detailed process is as follows.

Step 0 (Key Generation). Assume N users submit sensing data in the data submission phase. Given the security parameter k and threshold value t , a trusted third party creates three key pairs for each user u_i as follows.

$$\{(PK_i^s, SK_i^s), (PK_i^a, SK_i^a), (PK_i^r, SK_i^r)\} \leftarrow \text{KA.gen}(k), \quad (9)$$

where (PK_i^s, SK_i^s) are used for signature, (PK_i^a, SK_i^a) are used to generate a session key with other users for symmetric encryption, and (PK_i^r, SK_i^r) are used to generate a session key with other users u_j as the noise $r_{i,j}$. Then, each user u_i signs two public keys using SK_i^s

as $\rho_i \leftarrow \text{sign} \cdot (SK_i^s, PK_i^s \| PK_i^r)$ and sends $\{(\rho_i, \|PK_i^a \| PK_i^r)\}$ to $\mathcal{S}$.

When receiving messages from at least t users (which denotes the surviving users as a set $U_1 \subseteq U$), $\mathcal{S}$ broadcasts $\{(u_j, \rho_j, PK_j^a, PK_j^r)\}_{u_j \in U_1}$ to all users. Otherwise, abort.

Step 1 (Key Sharing). After receiving the information from $\mathcal{S}$, each user u_i confirms whether $|U_1| \geq t$; then, u_i verifies whether the signature ρ_j is valid using the public key PK_j^s for other user u_j . If not, abort. Next, u_i selects a random parameter $n_i \leftarrow \mathcal{F}$ and generates shares of n_i and SK_i^r as follows.

$$\begin{aligned} \{(u_j, n_{j,i})\}_{u_j \in U_1} &\leftarrow \text{Shamir.share}(n_i, t, U_1), \\ \{(u_j, SK_{j,i}^r)\}_{u_j \in U_1} &\leftarrow \text{Shamir.share}(SK_i^r, t, U_1). \end{aligned} \quad (10)$$

Then, each user u_i generates a session key with other users $u_j \in U_1 \setminus \{u_i\}$ and uses the symmetric authenticated encryption to encrypt two types of shares as follows.

$$\mathcal{T}_{j,i} \leftarrow \text{AE.enc}(\text{KA.agree}(SK_i^a, PK_j^a), u_i \| u_j \| n_{j,i} \| SK_{j,i}^r), \quad (11)$$

where the symmetric authenticated encryption is indistinguishable under ciphertext integrity attack and chosen plaintext attack. It can ensure the confidentiality and integrity of messages, which are exchanged between two parties. We do not repeat the details here. If any of the above processes fails, abort. Otherwise, each user u_i sends $\mathcal{T}_{j,i}$ to $\mathcal{S}$.

When receiving messages from at least t users (which denotes the surviving users as a set U_2), $\mathcal{S}$ randomly initializes the ground truth x^* and then broadcasts $\{\mathcal{T}_{j,i}\}_{u_j \in U_2}$ and x^* to all users. Otherwise, abort.

Step 2 (Masking Input Data). After receiving x^* and $\{\mathcal{T}_{j,i}\}_{u_j \in U_2}$ from $\mathcal{S}$, each user u_i confirms whether $|U_2| \geq t$, then computes $r_{i,j} \leftarrow \text{KA.agree}(SK_i^r, PK_j^r)$ for every user $u_j \in U_2 \setminus \{u_i\}$, and gets masked input data z_i^2 as follows.

$$\begin{aligned} z_i^2 &= y_i^2 + \sum_{j \in U_2: i < j} \text{PRG}(r_{i,j}) - \sum_{j \in U_2: i > j} \text{PRG}(r_{i,j}) + \text{PRG}(n_i) \pmod{R} \\ &= d_{\text{ist}}(x_i, x^*) + \sum_{j \in U_2: i < j} \text{PRG}(r_{i,j}) - \sum_{j \in U_2: i > j} \text{PRG}(r_{i,j}) + \text{PRG}(n_i) \pmod{R}, \end{aligned} \quad (12)$$

where $d_{ist}(x_i, x^*)$ is the input data in the second round, represented by y_i^2 for convenience, and z_i^2 indicates the masked input data. If any of the above processes fails, abort. Otherwise, each user u_i sends $\{z_i^2\}_{u_i \in U_2}$ to $\mathcal{S}$.

When receiving z_i^2 from at least t users (which denotes the surviving users as a set U_3), $\mathcal{S}$ sends the list of U_3 to all users. Otherwise, abort.

Step 3 (Consistency Check). After receiving the list of U_3 from $\mathcal{S}$, each user u_i confirms whether $|U_3| \geq t$. Then, u_i calculates the signature $\rho_i' \leftarrow \text{sign} \cdot (SK_i^s, U_3)$ and sends it to $\mathcal{S}$.

When receiving ρ_i' from at least t users (which denotes the surviving users as a set U_4), $\mathcal{S}$ sends $\{u_j, \rho_j'\}_{u_j \in U_4}$ to all users. Otherwise, abort.

Step 4 (Unmasking). After receiving the list of U_4 from $\mathcal{S}$, each user u_i confirms whether $U_4 \subseteq U_3$, $|U_4| \geq t$, and the signature ρ_j' is valid using the public key PK_j^s . Then, u_i decrypts $\mathcal{T}_{j,i}$ for users $u_j \in U_2 \setminus \{u_i\}$ as follows.

$$u_i' \| u_j' \| n_{j,i} \| SK_{j,i}^r \leftarrow \text{AE.dec}(\text{KA.agree}(SK_i^a, PK_j^a), \mathcal{T}_{j,i}). \quad (13)$$

Then, $n_{j,i}$ ($u_j \in U_3$) and $SK_{j,i}^r$ ($u_j \in U_2 \setminus U_3$) will be sent to $\mathcal{S}$ if $u_i = u_i'$ and $u_j = u_j'$. If any of the above processes fails, abort.

After receiving messages from users, $\mathcal{S}$ performs the deviation elimination and regards users who modify the data as dropped users, and $\mathcal{S}$ discards dropped users' data. The surviving users are then denoted as a set $U_5 \subseteq U_4$. If $|U_5| \geq t$, the secret key SK_i^r and masks $\text{PRG}(r_{i,j})$, $u_i \in U_2 \setminus U_3$ can be reconstructed as follows.

$$\begin{aligned} SK_i^r &\leftarrow \text{Shamir.recon}\left(\{(SK_{j,i}^r)\}_{u_j \in U_5}, t\right), \\ \text{PRG}(r_{i,j}) &\leftarrow \text{PRG}\left(\text{KA.agree}\left(\{SK_i^r, PK_j^r\}_{u_j \in U_3}\right)\right). \end{aligned} \quad (14)$$

Furthermore, the $\text{PRG}(n_i)$, $u_i \in U_3$ can be reconstructed as follows.

$$\text{PRG}(n_i) \leftarrow \text{PRG}\left(\text{Shamir.recon}\left(\{(n_{j,i}, t)\}_{u_j \in U_5}\right)\right). \quad (15)$$

Next, the aggregated results of y_i^2 can be calculated as follows.

$$\begin{aligned} \sum_{u_i \in U_3} y_i^2 &= \sum_{u_i \in U_3} y_i^2 - \sum_{u_j \in U_3} \text{PRG}(n_i) - \sum_{u_i \in U_3, u_j \in U_2 \setminus U_3: i < j} \text{PRG}(r_{i,j}) \\ &+ \sum_{u_i \in U_3, u_j \in U_2 \setminus U_3: i < j} \text{PRG}(r_{i,j}) \pmod{R} = \sum_{u_i \in U_3} d_{ist}(x_i, x^*). \end{aligned} \quad (16)$$

Then, $\mathcal{S}$ selects a random positive noise value m to mask the raw aggregation results to prevent users from obtaining weight information.

$$\mathcal{W}_{\text{result}} = \text{Log}\left(\sum_{u_i \in U_3} d_{ist}(x_i, x^*)\right) + m. \quad (17)$$

Next, $\mathcal{S}$ sends $\mathcal{W}_{\text{result}}$ to all users.

Step 5 (Masked Input Generation). After receiving $\mathcal{W}_{\text{result}}$ from $\mathcal{S}$, each user u_i computes $r_{i,j} \leftarrow \text{KA.agree}(SK_i^r, PK_j^r)$ for every surviving user $u_j \in U_5 \setminus \{u_i\}$. Then, each user u_i calculates the masked weight information as follows.

$$\begin{aligned} z_i^{5'} &= \mathcal{W}_{\text{result}} - \text{Log}(d_{ist}(x_i, x^*)) + \text{PRG}(n_i) + \sum_{u_j \in U_5: i > j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U_5: i < j} \text{PRG}(r_{i,j}) \pmod{R} \\ &= w_i + m + \text{PRG}(n_i) + \sum_{u_j \in U_5: i > j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U_5: i < j} \text{PRG}(r_{i,j}) \pmod{R}, \\ z_i^{5''} &= (\mathcal{W}_{\text{result}} - \text{Log}(d_{ist}(x_i, x^*))) \cdot x_i + \text{PRG}(n_i) + \sum_{u_j \in U_5: i > j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U_5: i < j} \text{PRG}(r_{i,j}) \pmod{R} \\ &= (w_i + m) \cdot x_i + \text{PRG}(n_i) + \sum_{u_j \in U_5: i > j} \text{PRG}(r_{i,j}) - \sum_{u_j \in U_5: i < j} \text{PRG}(r_{i,j}) \pmod{R}. \end{aligned} \quad (18)$$

So, the masked input data are denoted as (z_i^b, z_i^b) , where the raw input data are $y_i^b = w_i + m$, $y_i^b = (w_i + m) \cdot x_i$. If any of the above processes fails, abort. Otherwise, each user u_i sends (z_i^b, z_i^b) to $\mathcal{S}$.

Step 6 (Unmasking). After receiving (z_i^b, z_i^b) from at least t users (which denotes the surviving users as a set U_6), $\mathcal{S}$ sends the list of U_6 to all users. Otherwise, abort. Then, each user $u_j \in U_6 \setminus \{u_i\}$ decrypts $\mathcal{T}_{j,i}$ as follows.

$$u'_i \| u'_j \| n_{j,i} \| SK_{j,i}^r \leftarrow \text{AE.dec}(\text{KA.agree}(SK_i^a, PK_j^a), \mathcal{T}_{j,i}). \quad (19)$$

Then, $n_{j,i}$ ($u_j \in U_6$) and $SK_{j,i}^r$ ($u_j \in U_5 \setminus U_6$) will be sent to $\mathcal{S}$ if $u_i = u'_i$ and $u_j = u'_j$. If any of the above processes fails, abort.

After receiving the information from at least t users (which denotes the surviving users as a set U_7), $\mathcal{S}$ restores the secret key SK_i^r , $u_j = u'_j$ for each user ($u_j \in U_5 \setminus U_6$) and $\{\text{PRG}(n_i)\}_{u_i \in U_6}$ as follows.

$$\begin{aligned} SK_i^r &\leftarrow \text{Shamir.recon}\left(\{(SK_{j,i}^r)\}_{u_j \in U_7}, t\right), \\ \text{PRG}(r_{i,j}) &\leftarrow \text{PRG}\left(\text{KA.agree}\left(\{SK_i^r, PK_j^r\}_{u_j \in U_6}\right)\right), \\ \text{PRG}(n_i) &\leftarrow \text{PRG}\left(\text{Shamir.recon}\left(\{n_{j,i}\}_{u_j \in U_7}, t\right)\right). \end{aligned} \quad (20)$$

Then, $\mathcal{S}$ can calculate the aggregation results as follows.

$$\begin{aligned} \sum_{u_i \in U_6} y_i^{5'} &= \sum_{u_i \in U_6} z_i^{5'} - \sum_{u_i \in U_6} \text{PRG}(n_i) \\ &\quad - \sum_{u_i \in U_6, u_j \in U_5 \setminus U_6: i < j} \text{PRG}(r_{i,j}) \\ &\quad + \sum_{u_i \in U_6, u_j \in U_5 \setminus U_6: i > j} \text{PRG}(r_{j,i}) \pmod{R} \\ &= \sum_{u_i \in U_6} (w_i + m), \\ \sum_{u_i \in U_6} y_i^{5''} &= \sum_{u_i \in U_6} z_i^{5''} - \sum_{u_i \in U_6} \text{PRG}(n_i) \\ &\quad - \sum_{u_i \in U_6, u_j \in U_5 \setminus U_6: i < j} \text{PRG}(r_{i,j}) \\ &\quad + \sum_{u_i \in U_6, u_j \in U_5 \setminus U_6: i > j} \text{PRG}(r_{j,i}) \pmod{R} \\ &= \sum_{u_i \in U_6} (w_i + m) \cdot x_i. \end{aligned} \quad (21)$$

Next, $\mathcal{S}$ eliminates the random noise value m as follows.

$$\begin{aligned} \mathcal{W}_{\text{result}'} &= \sum_{u_i \in U_6} y_i^{5'} \bmod \left(\sum_{u_i \in U_6} m \right) \\ &= \sum_{u_i \in U_6} w_i + \sum_{u_i \in U_6} m \bmod \left(\sum_{u_i \in U_6} m \right) \\ &= \sum_{u_i \in U_6} w_i, \\ \mathcal{W}_{\text{result}''} &= \sum_{u_i \in U_6} y_i^{5''} \bmod \left(\sum_{u_i \in U_6} m \right) \\ &= \sum_{u_i \in U_6} (w_i + m) \cdot x_i \bmod \left(\sum_{u_i \in U_6} m \right) \\ &= \sum_{u_i \in U_6} w_i \cdot x_i. \end{aligned} \quad (22)$$

Therefore, the current ground truth x^* and the weight w_i of every user $u_i \in U_6$ can be calculated using formulas (1) and (2) as follows.

$$x^* = \frac{\sum_{u_i \in U_6} w_i \cdot x_i}{\sum_{u_i \in U_6} w_i} = \frac{\mathcal{W}_{\text{result}''}}{\mathcal{W}_{\text{result}'}} \quad (23)$$

$$w_i = \text{Log} \left(\frac{\sum_{u_i \in U_6} d_{ist}(x_i, x^*)}{d_{ist}(x_i, x^*)} \right).$$

Thus, $\mathcal{S}$ can get the final ground truth x^* and the weight w_i of every user u_i by repeating steps 0 to 6 until the convergence conditions are met. And the weight w_i will be used to calculate the data quality q_i of each user u_i .

5.4. Reward Distribution. The weight u_i calculated by truth discovery can represent the effective contribution of users. Still, to facilitate reward distribution, we need to quantify the data quality q_i of every user u_i further. Then, $\mathcal{S}$ can compute the monetary reward p_i according to the data quality q_i of u_i .

To achieve the rationality of reward distribution, we set $\sum_{u_i \in U_6} q_i = 1$, so the data quality q_i of each user u_i can be calculated as follows.

$$q_i = \frac{w_i}{\sum_{u_i \in U_6} w_i}. \quad (24)$$

Next, we calculate the monetary reward p_i of each user u_i as follows. And the higher the quality q_i of u_i 's data, the more reward u_i can get.

$$p_i = \frac{B}{|U_6|} + \pi \cdot (q_i - \bar{q}), \quad (25)$$

where $\bar{q} (\bar{q} = (\sum_{u_i \in U_6} q_i / |U_6|) = (1/|U_6|))$ is the average quality of all surviving users. $\pi (0 < \pi < (B/N))$ represents the reward control parameter, which is a small rational number. The function of π is to ensure that the reward p_i is non-negative. And $|U_6|$ is the number of surviving users.

Since $|U_6| \leq N$, we can know that the lowest reward p_i which a user can get is $(B - \pi/N)$. When the number of final online users $|U_6| = N$, each user u_i 's reward is $p_i = (B/N) + \pi \cdot (q_i - \bar{q})$. If some users dropped out and $|U_6| < N$, $\mathcal{S}$ will distribute the task budget to each surviving user $u_i \in U_6$, and each user's reward is $p_i = (B/|U_6|) + \pi \cdot (q_i - \bar{q})$. Therefore, our reward distribution formula is applicable regardless of whether there are users offline.

6. Analysis

In this section, we introduce property analysis, privacy analysis, and security analysis to illustrate the feasibility of our PAID.

6.1. Property Analysis. In this section, we introduce eligibility, zero knowledge, payment rationality, and budget rationality of our PAID.

Theorem 1. (eligibility) If the data $\mathcal{D}_i$ ($\mathcal{D}_i = (x_i, \tau_i, \hat{t}_i, \tilde{t}_i)$) collected by users do not meet the eligibility requirement $\mathcal{E}$, these users cannot pass the eligibility assessment.

Proof. We assume that the user's data are denoted as s , and the eligibility requirement interval is $[a, b]$. The user gets ciphertext $E(s)$ using homomorphic encryption. Then, $\mathcal{S}$ picks different random k, b and constructs a monotone increasing (or decreasing) function $f(x) = kx + b$. Then, $\mathcal{S}$ computes $f(a)$, $f(b)$, and $c = E(s)^k E(b) = E(ks + b)$. When receiving $f(a)$, $f(b)$, c from $\mathcal{S}$, the user decrypts c to get $f(s)$ and compares the sizes of $f(a)$, $f(b)$, $f(s)$. Because the user does not know the monotonicity of the function, it is impossible to determine the size relationship among the three numbers. Therefore, if the user's data are not qualified, then it cannot pass the qualification judgment. $\square$

Theorem 2. (zero knowledge) The server $\mathcal{S}$ can determine whether the user's data meet the eligibility requirements, but it cannot know the user's specific data content.

Proof. Similar to the description in Theorem 1, we assume that the user's data are s , and the server $\mathcal{S}$ can receive the user's homomorphic encrypted ciphertext $E(s)$. Since the Paillier cryptosystem is indistinguishable under the chosen plaintext attack, a malicious user has no way to recover the plaintext s . The server $\mathcal{S}$ may be curious about each user's data, but it cannot obtain each user's data s without knowing the secret key. $\square$

Theorem 3. (payment rationality) If an honest user u_i provides qualified data, u_i can obtain a non-negative utility.

Proof. The utility ut_i of each user u_i is determined by the cost of u_i and the real reward from task publisher TP, i.e., $ut_i = p_i - c_i$.

If the data provided by an untrusted user u_i are not qualified, u_i cannot pass the eligibility assessment, so the untrusted user's utility $ut_i = 0$. However, when $c_i > (B - \pi/N)$ ($(B - \pi/N)$ is the posted lowest pricing), an honest user u_i will refuse to participate in the sensing task, so the trusted user's utility $ut_i = 0$. When $c_i \leq (B - \pi/N)$, an honest user u_i will participate in the sensing task and earn a reward $p_i = (B/m) + \pi \cdot (q_i - \bar{q})$. Since $\bar{q} = \sum_{i=1}^m q_i/m = 1/m$, $0 < q_i < 1$, and $m \leq N$, we have

$$p_i = \frac{B}{m} + \pi \cdot (q_i - \bar{q}) = \frac{B - \pi}{m} + \pi \cdot q_i > \frac{B - \pi}{N}. \quad (26)$$

Therefore, we can know that $p_i - c_i > 0$. To summarize, a user's real reward is always non-negative. $\square$

Theorem 4. (budget rationality) The total payment of the task publisher TP is no larger than budget B in our PAID.

Proof. The total rewards for all users are calculated as follows.

$$\begin{aligned} \sum_{i=0}^m p_i &= \sum_{i=0}^m \left(\frac{B}{m} + \pi \cdot (q_i - \bar{q}) \right) \\ &= B + \pi \cdot \sum_{i=0}^m \left(q_i - \frac{\sum_{i=1}^m q_i}{m} \right) \\ &= B + \pi \cdot \left(\sum_{i=0}^m q_i - 1 \right) \\ &= B. \end{aligned} \quad (27)$$

Hence, $\sum_{i=0}^m p_i \leq B$, i.e., our PAID is budget rational. $\square$

6.2. Privacy Analysis. In this section, we demonstrate the protection of user's sensing data, location, and identity privacy in our PAID.

Theorem 5. (data and location privacy protection) Except for the user himself, other parties cannot obtain the user's sensing data and location data.

Proof. In PAID, the objects that steal users' data and location privacy are mainly the server $\mathcal{S}$ and external attackers. Specifically, the server $\mathcal{S}$ may obtain users' sensing data and location privacy in eligibility assessment and truth discovery. External attackers steal data and location privacy by eavesdropping on the communication between the server $\mathcal{S}$ and users.

According to Theorem 2, we can know that our PAID has the property of zero knowledge, so the server $\mathcal{S}$ cannot learn users' sensing data and location data in the eligibility assessment. In truth discovery, users' sensing data are sent to $\mathcal{S}$ after performing the double-masking scheme. However, the server $\mathcal{S}$ cannot recover users' raw sensing data by double-masking sensing data. Furthermore, before the communication between the user u_i and $\mathcal{S}$, the data are encrypted by AES symmetric encryption function $\text{SEnc}(y_i, k_i)$. Therefore, as long as $\text{SEnc}(y_i, k_i)$ is secure, external attackers cannot steal the data y_i by eavesdropping communication. $\square$

Theorem 6. (identity privacy protection) When users participate in a sensing task, they use an anonymous identity rather than their real identity. Therefore, any PPT adversary cannot distinguish the users' identities.

Proof. In PAID, the anonymous identity of a user u_i is represented by $k_i \leftarrow \text{KA.agree}(SK_i, PK_{\mathcal{S}}^i)$, and the real identity of u_i is SK_i where $SK_i = x_i \leftarrow Z_q$, and $PK_{\mathcal{S}}^i = g^{x_i s}$ ($PK_{\mathcal{S}}^i$ is a token assigned by $\mathcal{S}$). The user u_i uses an anonymous identity k_i rather than a real identity SK_i to participate in a sensing task. Because of the DDH problem, the PPT adversary cannot get the real identity SK_i of the user u_i by the anonymous identity k_i . We omit the detailed proof, and interested readers can learn more details in the literature [36]. $\square$

6.3. Security Analysis. In this section, we describe the attacks our PAID can resist, including denial of payment attack (DoP), inference attack (IA), data pollution attack (DPA), and Sybil attack (SA).

- (1) *Resistance to Denial of Payment Attack (DoP).* We use the prepayment mechanism in our PAID. At the beginning of a sensing task, the task publisher TP pays the monetary rewards of users to $\mathcal{S}$ in advance. If a malicious TP refuses to pay the monetary reward after receiving the data, $\mathcal{S}$ can pay the reward to users according to the reward distribution formula. Therefore, the TP cannot refuse to pay users the reward.
- (2) *Resistance to Inference Attack (IA).* The server $\mathcal{S}$ cannot initiate an inference attack on users' data due to the zero-knowledge property of our PAID.
- (3) *Resistance to Data Pollution Attack (DPA).* Our PAID introduces eligibility assessment, and the unqualified data submitted by users are not used in the truth discovery algorithm. Therefore, our PAID can resist the data pollution attack (DPA).
- (4) *Resistance to Sybil Attack (SA).* The anonymous identity k_i of a user u_i needs the information PK_i provided by the user and the token $PK_{\mathcal{S}}^i$ assigned by $\mathcal{S}$. Each user can only obtain one token from $\mathcal{S}$ and then get the anonymous identity k_i using the key agreement algorithm. Hence, untrusted users cannot forge vast fake identities to launch the Sybil attack (SA).

7. Performance Evaluation

In this section, we use a temperature dataset from Roma for performance evaluation. First, we describe the computational and communication overhead of the eligibility assessment. Then, we show the performance of the truth discovery algorithm. Finally, the comparison with the related work shows that the quality quantification and incentive mechanism are effective.

In our experiment, the server has Intel(R) Xeon(R) E3-1231v3 3.4 GHz CPU, 16 GB RAM, 256 GB SSD, and 1 TB mechanical hard disk and runs on Ubuntu 18.04 operating system. These mobile devices are equipped with Android system with 2.2 GHz CPU and 4 GB RAM. The Roma temperature dataset includes users' ID, date, time, longitude, latitude, and temperature. In particular, the range accuracy of location, time, and sensing data (temperature) is 1 meter, 1 second, and 0.01°C, respectively. Before performing the eligibility assessment, we convert the decimal interval to the corresponding integer interval by moving the decimal point to the right. Figure 3 shows the statistical results of 232 qualified users. And we select 100 data from all qualified data for performance evaluation.

7.1. Evaluation of Eligibility Assessment. In this section, we analyze the computational and communication overhead in the eligibility assessment. Table 2 shows the performance comparison between our PAID and related work.

7.1.1. Computational Overhead. The Paillier homomorphic encryption requires two exponents (exp), one multiplication (mul), and one modular operation (mod). One decryption operation needs to perform two exponents (exp), three divisions (div), and two modular operations (mod). And in our interval judgment scheme, the user u_i needs to perform one encryption and one decryption, so the computational cost of the user is $4n \cdot \text{exp} + n \cdot \text{mul} + 3n \cdot \text{div} + 3n \cdot \text{mod}$, where n is the number of users. The server needs to perform one encryption $E(b)$ and calculates $c = E(x_i)^k E(b)$, so the computational overhead of the server is $3n \cdot \text{exp} + 2n \cdot \text{mul} + 2n \cdot \text{mod}$. Consequently, the total computational overhead is $n \cdot (7 \cdot \text{exp} + 3 \cdot \text{mul} + 3 \cdot \text{div} + 5 \cdot \text{mod})$ and the computation complexity of the interval judgment scheme is $O(n)$.

7.1.2. Communication Overhead. According to our interval judgment scheme, users need to send encrypted data $E(x_i)$ to the server $\mathcal{S}$, and the communication overhead is $\|N^2\|$ bits, where N is the product of two large primes p, q . After receiving the encrypted data $E(x_i)$, the server $\mathcal{S}$ calculates $c = E(x_i)^k E(b)$ and sends it to the user. The communication overhead is $\|c\|$ bits, where $\|c\|$ denotes the bit length of ciphertext. So, we can conclude that the total communication overhead is $\|N^2\| + \|c\|$ bits.

7.2. Evaluation of Truth Discovery. In this section, we select 100 users to participate in the performance comparison of truth discovery. We compare the truth discovery of our PAID with the related work from five aspects, including accuracy, convergence, robustness to users dropping out, computational overhead, and communication overhead. The evaluation results show that our truth discovery algorithm has good accuracy, quick convergence, and high robustness to users dropping out. Besides, the computational overhead and communication overhead of our algorithm are better than those of the related work. Therefore, our truth discovery algorithm is reasonable.

7.2.1. Accuracy. We utilize the root of mean squared error (RMSE) to measure the resulting accuracy between PAID and CRH [32]. Figure 4 shows that the accuracy rates of PAID and CRH are similar when different numbers of users participate in a sensing task.

7.2.2. Convergence. To prove the convergence ability of our truth discovery algorithm in PAID, we choose four different initial values to calculate the error rate of ground truth. As shown in Figure 5, our PAID can converge quickly in a few iterations when choosing different initial values.

7.2.3. Robustness to Users Dropping Out. To analyze the robustness of our PAID to dropped users, we count the number of PAID failures and compare with related work PPTD [41]. Failure means that the model cannot continue to

FIGURE 3: Statistics of dataset. (a) Space-time statistics of users. (b) Temperature statistics of users.

TABLE 2: Performance comparison between PAID and related work.

Protocol	Computational overhead	Communication overhead (bits)
PAID	$n \cdot (7 \cdot \text{exp} + 3 \cdot \text{mul} + 3 \cdot \text{div} + 5 \cdot \text{mod})$	$\ N^2\ + \ c\ $
[20]	$n \cdot (10 \cdot \text{exp} + 5 \cdot \text{mul} + 5 \cdot \text{comp} + 5 \cdot \text{mod})$	$3\ p\ + 2\ c\ $

Note. p is a large prime. And exp, mul, div, comp, and mod represent one exponent arithmetic, one multiplication, one division, one comparison operation, and one modular arithmetic.

FIGURE 4: Accuracy evaluation.

FIGURE 5: Convergence evaluation.

run and have to restart because of users' exit. In the PPTD, it is considered as a failure once a user quits in the whole truth discovery process. In our PAID, it is deemed to be a failure only when the number of online users is less than the threshold t ($t = 25$ in our experiment). And we repeat the experiment 50 times to count the failure times of the two

models. Figure 6 shows the failure times of the two models when different users participate in a sensing task. We can know that the number of PPTD failures increases as the number of users increases. However, as long as online users' number is greater than the threshold, our PAID is robust to dropped users.

FIGURE 6: Robustness evaluation.

7.2.4. Computational Overhead. We compare the computational overhead of PAID and PPTD [41]. Figure 7 shows the running time of the two schemes for different users. It is evident that the running time of our PAID is far less than that of PPTD.

7.2.5. Communication Overhead. We count the communication overhead of users in a complete iterative process and compare our scheme with PPTD [41]. And we do not count the server's communication overhead because we can regard the total communication overhead of all users as the communication cost of the server. Table 3 shows that the communication overhead of our PAID is far less than that of PPTD, although the number of users is different.

7.3. Evaluation of Incentive Mechanism. In this section, we compare the monetary rewards of our PAID and related work. In the experiment, we select 100 users, including 80 qualified users and 20 unqualified users. And the budget $B = 100$, $\pi = 0.3$. DQTE [42] is a scheme that includes unqualified users in reward distribution, while DQTE+ removes unqualified users before reward distribution. As Figure 8 shows, users in DQTE get almost the same rewards. Although DQTE+ removes unqualified users, there is no obvious difference for users' rewards except for the increase in each user's monetary rewards. However, our scheme can provide higher monetary rewards for users who submit higher quality data. Therefore, our scheme can effectively motivate users to provide high-quality sensing data.

8. Related Work

Truth discovery is an effective technology that can calculate the ground truth and users' data quality from conflicting sensing data. Li et al. [32] proposed a general truth discovery scheme, but privacy protection is not in their work

FIGURE 7: Running time for different number of users.

TABLE 3: Communication overhead of users (kB).

Number of users	PAID	PPTD
50	0.16	1.38
60	0.24	1.78
70	0.32	1.86
80	0.45	2.31
90	0.58	2.63
100	0.76	2.97

FIGURE 8: The reward evaluation of users.

scope. To protect users' privacy data, Miao et al. [41] proposed the first privacy-preserving truth discovery scheme using the Paillier cryptosystem, but the computational and communication costs are huge. Zheng et al.

[43] designed a privacy-aware truth discovery, which greatly reduced the computational and communication overhead through a secure sum protocol. Zhang et al. [44] designed a truth discovery scheme using a one-way hash chain to ensure privacy security, and all truth discovery operations are completed by fog and cloud platforms. Tang et al. [45] used two servers to complete the calculation process of truth discovery, which can effectively protect users' sensing data privacy. However, these works do not take into account the failure of the MCS system caused by users' exit. Bonawitz et al. [39] proposed a double-masking scheme for secure data aggregation, and this scheme allows users to exit. After that, Xu et al. [15] designed a privacy-preserving truth discovery scheme based on the double-masking scheme. However, these truth discovery schemes do not incorporate incentive mechanisms. If malicious users constantly input erroneous data, it will affect the reliability of the results in the MCS system.

Another previous work [42, 46] related to this paper is the incentive mechanism in the MCS system. Zhang et al. [47] presented a reverse auction model which can motivate online users to participate in sensing tasks. Jin et al. [16] designed an incentive mechanism model based on reverse combinatorial auctions, which can maximize social welfare and effectively motivate users. Yang et al. [42] introduced a quality-aware incentive mechanism, which can distribute rewards to users after calculating the data quality. However, these works do not consider the privacy of users. In [27], the authors designed a privacy-preserving incentive mechanism model. Nevertheless, these solutions can not eliminate users who provide error data. Zhao et al. [20] presented an incentive mechanism model to evaluate the reliability of users' data while protecting data privacy. Still, the user's sensing data need to be submitted to the task publisher, so the privacy protection of sensing data is still insufficient. Later, Zhao et al. [48] proposed a privacy-preserving incentive mechanism based on truth discovery. This model uses two servers to achieve real-time reward distribution while protecting users' privacy. However, most existing works do not take users' exit into account.

9. Conclusion

In this paper, we propose a privacy-preserving incentive mechanism based on truth discovery in the MCS system. Specifically, we introduce an eligibility assessment scheme to estimate whether the data submitted by users are qualified. Next, the truth discovery scheme calculates the ground truth and the weight of each user. Then, we quantify the data quality of users by the weight and distribute the rewards. Besides, we also demonstrate that PAID meets eligibility, zero knowledge, payment rationality, and budget rationality. And the analysis shows that our PAID can resist the denial of payment attack, inference attack, data pollution attack, and Sybil attack. Finally, experiments illustrate that PAID is effective, efficient, and robust to dropped users. In future work, we will design an incentive mechanism model for the application of multidimensional sensing data collection.

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 study was supported by the National Natural Science Foundation of China (nos. 61962022 and 62062034) and Key Research and Development Plan of Jiangxi Province (no. 20192BBE50077).

References

- [1] R. Gao, M. Zhao, T. Ye et al., "Jigsaw: indoor floor plan reconstruction via mobile crowdsensing," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, pp. 249–260, Association for Computing Machinery, Maui Hawaii, US, September 2014.
- [2] J. Xiong, M. Zhao, M. Z. A. Bhuiyan, L. Chen, and Y. Tian, "An ai-enabled three-party game framework for guaranteed data privacy in mobile edge crowdsensing of iot," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 922–933, 2019.
- [3] X. Wei, B. Sun, and J. Cui, "Task replica assignment in mobile self-organized crowdsensing," *International Journal of Performance Engineering*, vol. 16, no. 1, pp. 152–162, 2020.
- [4] Z. Wang, J. Zhao, J. Hu et al., "Towards personalized task-oriented worker recruitment in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 2080–2093, 2021.
- [5] 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.
- [6] R. K. Ganti, N. Pham, H. Ahmadi, S. Nangia, and T. F. Abdelzaher, "Greengps: a participatory sensing fuel-efficient maps application," in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services*, pp. 151–164, San Francisco, California, USA, June 2010.
- [7] K. Abualsaud, T. M. Elfouly, T. Khatib et al., "A survey on mobile crowd-sensing and its applications in the iot era," *IEEE Access*, vol. 7, pp. 3855–3881, 2018.
- [8] Z. Yin, C. Wu, Z. Yang, and Y. Liu, "Peer-to-peer indoor navigation using smartphones," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 5, pp. 1141–1153, 2017.
- [9] F. Montori, L. Bedogni, and L. Bononi, "A collaborative internet of things architecture for smart cities and environmental monitoring," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 592–605, 2017.
- [10] S. Zhang, H. Li, Y. Dai, J. Li, M. He, and R. Lu, "Verifiable outsourcing computation for matrix multiplication with improved efficiency and applicability," *IEEE Internet of Things Journal*, vol. 5, no. 6, pp. 5076–5088, 2018.
- [11] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in internet of things with privacy preserving: challenges, solutions and opportunities," *IEEE Network*, vol. 32, no. 6, pp. 144–151, 2018.

- [12] R. W. Ouyang, M. Srivastava, A. Toniolo, and T. J. Norman, "Truth discovery in crowdsourced detection of spatial events," *IEEE Transactions on Knowledge and Data Engineering*, vol. 28, no. 4, pp. 1047–1060, 2015.
- [13] Y. Li, J. Gao, C. Meng et al., "A survey on truth discovery," *ACM Sigkdd Explorations Newsletter*, vol. 17, no. 2, pp. 1–16, 2016.
- [14] F. Ma, Y. Li, Q. Li et al., "Faitcrowd: fine grained truth discovery for crowdsourced data aggregation," in *Proceedings of the 21th Acm Sigkdd International Conference on Knowledge Discovery and Data Mining*, pp. 745–754, Sydney, NSW, Australia, August 2015.
- [15] 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.
- [16] H. Jin, L. Su, D. Chen, K. Nahrstedt, and J. Xu, "Quality of information aware incentive mechanisms for mobile crowd sensing systems," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 167–176, Hangzhou, China, June 2015.
- [17] H. Jin, L. Su, and K. Nahrstedt, "Theseus: incentivizing truth discovery in mobile crowd sensing systems," in *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 1–10, Chennai India, July 2017.
- [18] X. Zhang, Z. Yang, W. Sun et al., "Incentives for mobile crowd sensing: a survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2015.
- [19] D. Peng, F. Wu, and G. Chen, "Data quality guided incentive mechanism design for crowdsensing," *IEEE Transactions on mobile Computing*, vol. 17, no. 2, pp. 307–319, 2017.
- [20] B. Zhao, S. Tang, X. Liu, and X. Zhang, "Pace: privacy-preserving and quality-aware incentive mechanism for mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 20, no. 5, pp. 1924–1939, 2020.
- [21] D. Dharminder and D. Mishra, "Lcpga: lattice-based conditional privacy preserving authentication in vehicular communication," *Transactions on Emerging Telecommunications Technologies*, vol. 31, no. 2, p. e3810, 2020.
- [22] 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.
- [23] K. Huang, X. Liu, S. Fu, D. Guo, and M. Xu, "A lightweight privacy-preserving cnn feature extraction framework for mobile sensing," *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1441–1455, 2021.
- [24] T. Wan, X. Liu, W. Liao, and N. Jiang, "Cryptanalysis and improvement of a smart card based authentication scheme for multi-server architecture using ecc," *IJ Network Security*, vol. 21, no. 6, pp. 993–1002, 2019.
- [25] 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, 2019.
- [26] B. Zhao, S. Tang, X. Liu, X. Zhang, and W.-N. Chen, "Ironm: privacy-preserving reliability estimation of heterogeneous data for mobile crowdsensing," *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5159–5170, 2020.
- [27] 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 IEEE INFOCOM 2019-IEEE Conference on Computer Communications*, pp. 2053–2061, IEEE, Paris, France, April 2019.
- [28] 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.
- [29] 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.
- [30] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4w1h in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 42–48, 2014.
- [31] L. Zhang, Y. Li, X. Xiao et al., "Crowdbuy: privacy-friendly image dataset purchasing via crowdsourcing," in *Proceedings of IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pp. 2735–2743, IEEE, Honolulu, HI, USA, April 2018.
- [32] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, and J. Han, "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, Utah USA, June 2014.
- [33] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: when qoe meets qop," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 74–80, 2015.
- [34] H. Li, D. Liu, Y. Dai, T. H. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Transactions on Emerging Topics in Computing*, vol. 6, no. 1, pp. 97–109, 2015.
- [35] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, 1979.
- [36] H. Krawczyk, "SIGMA: the 'SIGn-and-MAC' approach to authenticated diffie-hellman and its use in the IKE protocols," in *Proceedings of Annual International Cryptology Conference*, pp. 400–425, Springer, Santa Barbara, California, USA, August 2003.
- [37] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, Springer, Prague, Czech Republic, May 1999.
- [38] L. Chen, W. Zhang, S. Li, Q. Huang, and Z. Chen, "Fully privacy-preserving determination of point-range relationship," *SCIENTIA SINICA Informationis*, vol. 48, no. 2, pp. 187–204, 2018.
- [39] K. Bonawitz, V. Ivanov, B. Kreuter et al., "Practical secure aggregation for privacy-preserving machine learning," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1175–1191, Dallas, Texas, USA, October 2017.
- [40] Y. Jiang, B. Zhao, S. Tang, and H. T. Wu, "A verifiable and privacy-preserving multidimensional data aggregation scheme in mobile crowdsensing," *Transactions on Emerging Telecommunications Technologies*, vol. 32, no. 5, p. e4008, 2021.
- [41] 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, South Korea, November 2015.
- [42] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang, and G. Chen, "On designing data quality-aware truth estimation and surplus

- sharing method for mobile crowdsensing,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 832–847, 2017.
- [43] Y. Zheng, H. Duan, X. Yuan, and C. Wang, “Privacy-aware and efficient mobile crowdsensing with truth discovery,” *IEEE Transactions on Dependable and Secure Computing*, vol. 17, no. 1, pp. 121–133, 2017.
 - [44] C. Zhang, L. Zhu, C. Xu, X. Liu, and K. Sharif, “Reliable and privacy-preserving truth discovery for mobile crowdsensing systems,” *IEEE Transactions on Dependable and Secure Computing*, vol. 18, no. 3, pp. 1245–1260, 2019.
 - [45] J. Tang, S. Fu, X. Liu, Y. Luo, and M. Xu, “Achieving privacy-preserving and lightweight truth discovery in mobile crowdsensing,” *IEEE Transactions on Knowledge and Data Engineering*, p. 1, 2021.
 - [46] D. Peng, F. Wu, and G. Chen, “Pay as how well you do: a quality based incentive mechanism for crowdsensing,” in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 177–186, Hangzhou China, June 2015.
 - [47] X. Zhang, Z. Yang, Z. Zhou, H. Cai, L. Chen, and X. Li, “Free market of crowdsourcing: incentive mechanism design for mobile sensing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 12, pp. 3190–3200, 2014.
 - [48] B. Zhao, X. Liu, W. N. Chen et al., “Privacy and reliability-aware real-time incentive system for crowdsensing,” *IEEE Internet of Things Journal*, p. 1, 2021.