5-2019

PPTDS: A privacy-preserving truth discovery scheme in crowd sensing systems

Chuan ZHANG

Liehuang ZHU

Chang XU

Kashif SHARIF

Ximeng LIU
Singapore Management University, xmliu@smu.edu.sg

Citation

ZHANG, Chuan; ZHU, Liehuang; XU, Chang; SHARIF, Kashif; and LIU, Ximeng. PPTDS: A privacy-preserving truth discovery scheme in crowd sensing systems. (2019). *Information Sciences*. 484, 183-196.
Available at: https://ink.library.smu.edu.sg/sis_research/5152

Published in Information Sciences, 2019 May, 484, 183-196.
https://doi.org/10.1016/j.ins.2019.01.068

PPTDS: A privacy-preserving truth discovery scheme in crowd sensing systems

Chuan Zhang[a], Liehuang Zhu[a], Chang Xu[a,*], Kashif Sharif[a], Ximeng Liu[b]

[a] Beijing Engineering Research Center of Massive Language Information Processing and Cloud Computing Application,School of Computer Science and Technology, Beijing Institute of Technology, Beijing, China
[b] School of Information Systems, Singapore Management University and College of Mathematics and Computer Science, Fuzhou University, China

Keywords:
Crowd sensing
Truth discovery
Privacy-preserving
Efficiency

ABSTRACT

Benefiting from the fast development of human-carried mobile devices, crowd sensing has become an emerging paradigm to sense and collect data. However, reliability of sensory data provided by participating users is still a major concern. To address this reliability challenge, truth discovery is an effective technology to improve data accuracy, and has garnered significant attention. Nevertheless, many of state of art works in truth discovery, either failed to address the protection of participants' privacy or incurred tremendous overhead on the user side. In this paper, we first propose a privacy-preserving truth discovery scheme, named PPTDS-I, which is implemented on two non-colluding cloud platforms. By capitalizing on properties of modular arithmetic, this scheme is able to protect both users' sensory data and reliability information, and simultaneously achieve high efficiency and fault-tolerance. Additionally, for the scenarios with resource constrained devices, an efficient truth discovery scheme, named PPTDS-II, is presented. It can not only protect users' sensory data, but also avoids user participation in the iterative truth discovery procedure. Detailed security analysis shows that the proposed schemes are secure under a comprehensive threat model. Furthermore, extensive experimental analysis has been conducted, which proves the efficiency of the proposed schemes.

1. Introduction

The proliferation of portable, mobile, and wearable devices (e.g., smart phones, tablet computer, smart glasses, etc.) installed with various sensors (e.g., GPS, camera, compass, thermometer), has enabled crowd sensing to become an effective sensing paradigm [12,18,22]. In a typical crowd sensing system, sensing tasks, such as monitoring air quality, collecting health data, etc., are assigned to a large crowd of users carrying mobile devices. By collecting and analyzing the sensory data, such crowd sensing systems can serve a wide spectrum of applications for urban sensing [3], smart transportation [2], public opinion analysis [5], and many others, which directly make a positive impact on people's daily life.

Although the benefits of crowd sensing services are undisputed, the sensory data furnished by different users may not be completely reliable or directly usable. Major reasons for this include lack of sensor calibration, background noise, poor sensor quality, and in some cases malicious deception. Thus, an important task in crowd sensing systems is to find reliable

[*] Corresponding author.
E-mail address: xuchang@bit.edu.cn (C. Xu).

and usable information before utilizing it in real time systems for public benefits. A potential solution is to aggregate the sensory data collected from users participating in the same task, and use the methods such as average or voting to estimate final results. However, the challenge here is that the traditional methods usually treat all users equally, which makes it difficult to derive accurate results. Thus, an ideal approach is to capture the difference between the sensory data and the object truths, and further assign different reliability weights to different users. This approach, also called truth discovery, has received considerable attention [26,28,40]. Most of existing truth discovery approaches use the following basic principle: The user who usually provides truthful information is assigned a higher weight, while the data is more likely to be selected as the truth value if it is provided by a user with a higher weight.

This fundamental principle can significantly improve data accuracy in crowd sensing systems, but existing truth discovery schemes do not consider protecting users' privacy. In fact, if users' private information cannot be protected, users are usually reluctant to provide or share their sensory data. For example, by aggregating opinions from multiple users, some challenging problems can be easily tackled [4]. However, personal information such as professions and education levels may also be inferred from their answers. Similarly, aggregated medical treatment outcomes can provide valuable information for testing new drugs, but may result in a risk of disclosing users' health status.

To address these problems, several schemes have been proposed [29,30,36]. Miao et al. [29] proposed to protect users' sensory data and reliability information by adopting threshold Paillier cryptosystem [6]. Although their mechanism is able to guarantee users' privacy, it is not efficient, as each user needs to perform considerable amount of ciphertext-based calculations. To improve efficiency, Xu et al. [36] and Miao et al. [30] further proposed EPTD and L-PPTD respectively. Nevertheless, both schemes cannot achieve fault tolerance, i.e., if some users stop reporting data, these schemes will not work. In a nutshell, there is still a need to design a reliable, efficient, and privacy-preserving scheme for truth discovery, which can guarantee privacy of each user, and can adapt to the scenarios where users or devices do not report data for certain time periods.

In the light of above mentioned needs, in this paper we present a privacy-preserving truth discovery scheme (PPTDS) for crowd sensing systems. The proposed scheme is implemented by using two non-colluding clouds and employing some properties under the modulo. The three-fold contributions of PPTDS scheme are given below.

- Firstly, we propose PPTDS-I to protect users' sensory data and weight information. In this scheme, two clouds cooperatively estimate the object truths without disclosing users' private information. In addition, to reduce computational overhead incurred on the user side, the sensitive information is perturbed by adding random numbers, and all complex ciphertext-based operations are outsourced to the cloud platforms.
- Secondly, to further reduce computational and communication overhead of each user, we propose PPTDS-II, suited for the applications where only sensory data needs protection. In this scheme, each user is only required to provide perturbed sensory data along with encrypted random values for initialization of the truth discovery procedure.
- Thirdly, we present detailed analysis to prove that both PPTDS schemes can protect users' privacy, and personal or identifiable information is not disclosed to others. In addition, extensive experiments have been conducted to show the efficiency of PPTDS-I and PPTDS-II schemes.

The remainder of this paper is organized as follows. Related work is discussed in Section 2. In Section 3, we introduce the problem definition including system model, security model, and design goals. In Section 4, we describe necessary preliminary information. The PPTDS-I scheme and PPTDS-II scheme are presented in Sections 5 and 6 respectively. Then, we show the performance evaluation in Section 7. Lastly, we draw the conclusion in Section 8.

2. Related work

As an effective technology to extract reliable and truthful information, truth discovery has received considerable attention and many schemes targeting different applications have been proposed [16,26,28,40]. However, these schemes do not consider the protection of participating users' privacy. In fact, due to the sensitivity of users' sensory data, numerous users have shown great concern about their private information and usually refuse to offer their data if the privacy concerns cannot be solved [11,17,21]. Thus, some recent privacy-preserving truth discovery works have been proposed [29,30,36,44,45]. In [29], Miao et al. proposed a scheme called PPTD which can protect users' privacy when conducting truth discovery. Although high accuracy and strong privacy can be guaranteed, it introduces tremendous computational and communication overhead on the user side. To address this challenge, Xu et al. [36] proposed a lightweight and privacy-preserving truth discovery scheme based on [35,46]. In this scheme, each user is allocated a random value to perturb the raw data before delivering the data to the cloud. With the aggregated random values, the cloud is able to recover the summation of the raw data without disclosing each user's private information. However, the challenge is that if some users do not send perturbed data timely, this scheme will not work, as some of the random values are missing. Miao et al. [30] presented another lightweight and privacy-preserving truth discovery scheme via two non-colluding cloud platforms. Similar to [36], each user is also assigned random values to perturb the sensory data, weight, and weighted data. The difference is that the perturbed data is uploaded to a cloud S_A, while the random values are sent to another cloud S_B. By cooperating with each other, the two clouds can estimate the truthful values without disclosing any user's information. However, we find that if cloud S_B observes some users' perturbed data, it is able to recover the raw data, since it holds the corresponding random values. In [45], Zheng et al. proposed two schemes for single-server and two non-colluding server models respectively. However,

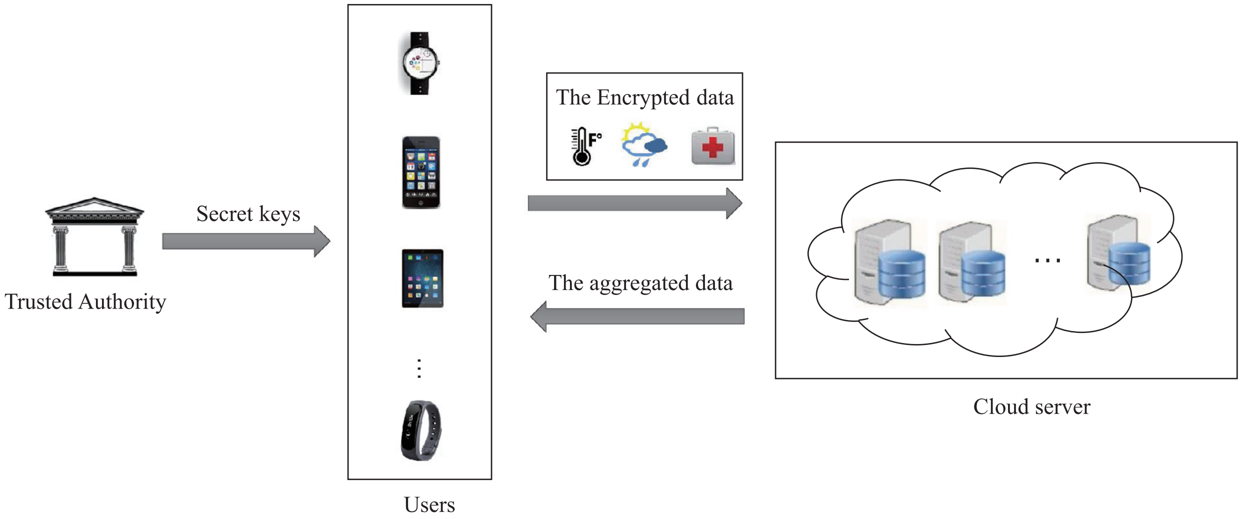

Fig. 1. System model.

in their second design, users are still required to submit data in the iteration phase. In [44], Zheng et al. adopted Garbled Circuit (GC) to design privacy-preserving truth discovery schemes. Nevertheless, it is not efficient in generating the garbled circuit.

With respect to the crowd sensing systems, we also discuss some privacy protection mechanisms in this work, which have been recognized in many areas [9,13,14,34,42,43]. Here we introduce some representative methods including (1) anonymization [1,33], which can achieve privacy protection by removing the information exchanged between participating users and other parties, (2) perturbation [7,19,32], which adds noises to the raw data to complete privacy protection before sending them to others, and (3) cryptology [8,10,11,20,41], in which the data is encrypted by using appropriate encryption tools such as Paillier cryptosystem to realize privacy protection. Specifically, in [24], Li et al. proposed a privacy-preserving multisubset data aggregation scheme in smart grid based on Paillier cryptosystem. In [25], Li et al. proposed a privacy-preserving prepayment scheme based on a power request and trading scheme in smart grid. Xu et al. [37] further proposed a fully Homomorphic encryption-based Merkle Tree construction to authenticate unbounded data. Finally, multi-party protocols have been widely studied [15,38,39], which inspired us to use two non-colluding cloud servers to design the privacy-preserving truth discovery schemes.

3. Problem definition

3.1. System model

The system model shown in Fig. 1 is mainly composed of three kinds of entities: a trusted authority (TA), users, and the cloud server. TA is fully trusted and it initializes secret keys for all users. Each user sends the encrypted data to the cloud, and the cloud is responsible to conduct truth discovery procedures and return the aggregated results to all users. Generally, the problem of truth discovery is formalized as follows. Assume there are M objects $\{o_m\}_{m=1}^{M}$ in the sensing task and K participating users $\{u_k\}_{k=1}^{K}$. Each user u_k observes the sensory data x_m^k for object o_m, and we use w_k to denote the weight of u_k. The goal of our scheme is to estimate the object truths $\{x_m^*\}_{m=1}^{M}$, while protecting each user's privacy from being disclosed to others.

3.2. Security model

Under the security model, TA is considered as fully trusted, while the cloud and users are honest-but-curious, which means they will follow the protocols formally, but are curious about users' privacy. Specifically, the cloud and some malicious users may try to infer the privacy of participating users because of curious or greedy forces. On the other side, users are usually paid for participating in sensing tasks, thus they are also motivated to deduce others' sensory data for economic benefits. Although the entities are semi-honest, we assume there is no collusion in this work, which is based on similar assumption in [29,30,36].

3.3. Design goals

Keeping in view the system and security model, the goal of this paper is to design a reliable, efficient and privacy-preserving truth discovery scheme for crowd sensing systems. In particular, the following design goals should be guaranteed.

- Privacy: The scheme should preserve privacy, such that each user's sensitive information is not disclosed to any other entity.
- Efficiency: The scheme should be efficient, such that the computational costs of tasks performed at the user end should be as minimum as possible. In addition, to save bandwidth costs, the communication overhead of each user should be minimal.
- Fault-tolerance: The scheme should achieve fault-tolerance, such that even if some users stop sending their data to the cloud, the proposed scheme is still able to estimate the object truths accurately.

4. Preliminary

In order to better present the schemes, we first introduce the basics of truth discovery and properties of modulo arithmetic.

4.1. Truth discovery

The general procedure of truth discovery mainly consists of two phases: weight update and truth update. In the following, we will give the detailed description of these.

4.1.1. Weight update

In this step, each user's weight information will be calculated on the basis of distance between their sensory data and the object truths. Generally, the weight of each user w_k can be obtained as $w_k = f(\sum_{m=1}^{M} d(x_m^k, x_m^*))$, where f is a function holding decreasing property, and $d(x_m^k, d_m^*)$ denotes a distance function. Without loss of generality, a weight calculating function widely used in truth discovery approaches [23,26,30] is used in this paper,

$$w_k = \log\left(\frac{\sum_{k=1}^{K}\sum_{m=1}^{M} d(x_m^k, x_m^*)}{\sum_{m=1}^{M} d(x_m^k, x_m^*)}\right). \tag{1}$$

According to Eq. (1), a user is assigned a higher weight if the sensory data is closed to the ground truth.

4.1.2. Truth update

After each user calculates the weight, the object truth can be estimated as

$$x_m^* = \frac{\sum_{k=1}^{K} x_m^k \cdot w_k}{\sum_{k=1}^{K} w_k}. \tag{2}$$

Note that, the selection of distance function depends on data types used in different application scenarios. In this paper, two common data types (i.e., continuous data and categorical data) are discussed.

In case of continuous data, a widely approved distance function $d(x_m^k, x_m^*) = (x_m^k - x_m^*)^2$ is adopted to calculate the difference between the sensory data and the object truth. In case of categorical data, as described in [29], the sensory data is defined as a vector $x_m^k = (0, \dots, 1(q\text{th}), \dots, 1)^T$, which represents u_k selecting qth answer for object o_m. Hence, the distance between x_m^k and x_m^* is defined as $d(x_m^k, x_m^*) = (x_m^k - x_m^*)^T(x_m^k - x_m^*)$. Combing the above phases, the general procedure of truth discovery is summarized in Algorithm 1.

Algorithm 1: Truth Discovery Algorithm.

```
Input: Sensory data: $\{x_m^k\}_{k,m=1}^{K,M}$
Output: Object truths: $\{x_m^*\}_{m=1}^{M}$
Initialize the object truths $[x_1^*, x_2^*, \cdots, x_M^*]$ and send them to each user.
for $iteration = 1, 2, \cdots, iteration_{\max}$ do
    for $k = 1, 2, \dots, K$ do
        Update each user's weight based on Eq. (1);
    for $m = 1, 2, \dots, M$ do
        Update the object truth based on Eq. (2);
return $\{x_m^*\}_{m=1}^{M}$;
```

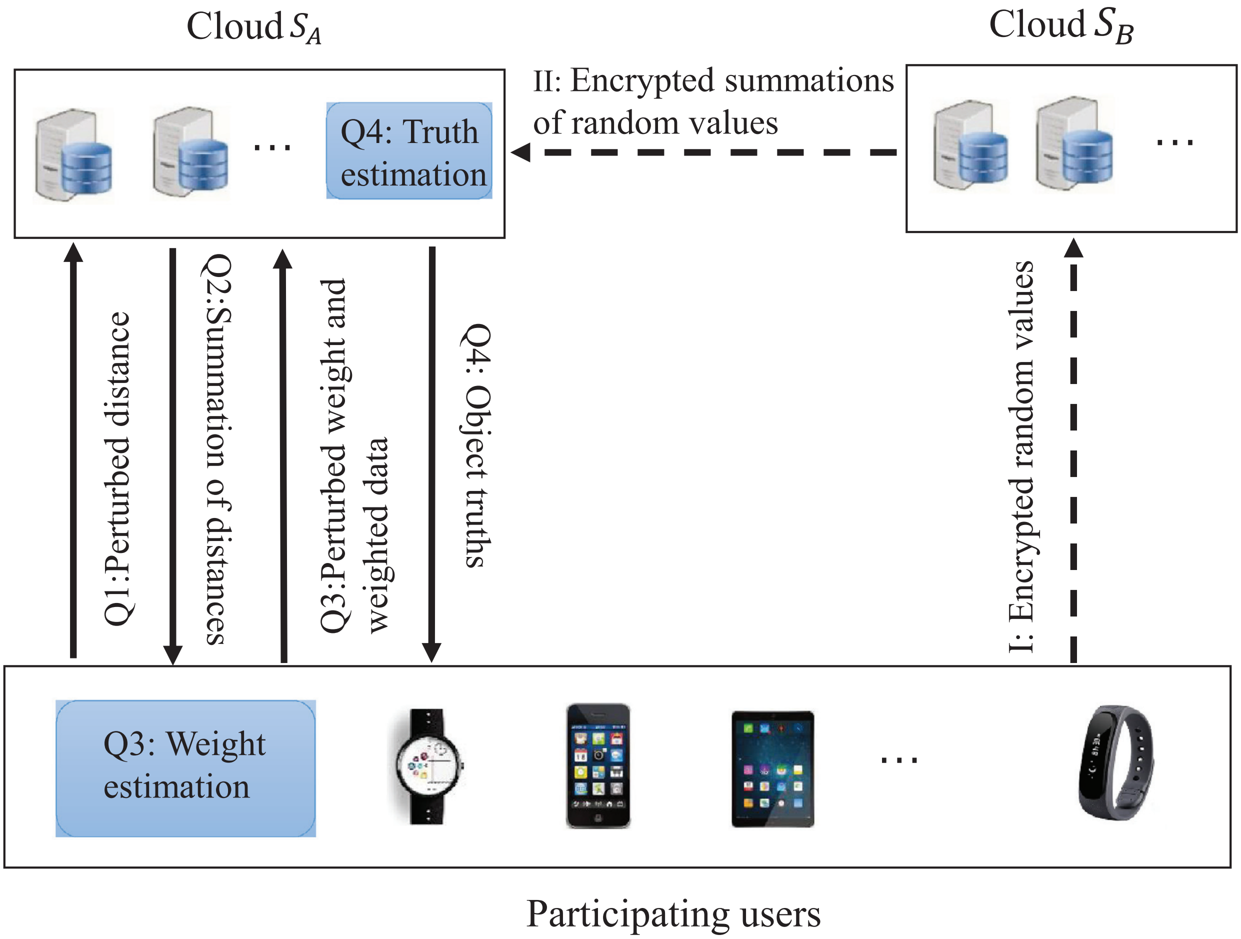
Cloud S_A
Cloud S_B
II: Encrypted summations of random values
Q4: Truth estimation
Q1:Perturbed distance
Q2:Summation of distances
Q3:Perturbed weight and weighted data
Q4: Object truths
I: Encrypted random values
Q3: Weight estimation
Participating users

Fig. 2. Work flow of PPTDS-I scheme. (The labels for data flow correspond to steps discussed in initialization and iteration phases of Algorithm 2).

4.2. Properties under modulo n^2

Select two large safe primes p, q, where $p = 2p' + 1, q = 2q' + 1$ and p', q' are also two primes. Based on p, q, compute $n = pq$ and $\lambda = 2p'q'$. From this, the following properties of modulo n^2 can be utilized.

First, for any message $x \in \mathbb{Z}^*_{n^2}$, we have $x^{n\lambda} \equiv 1 \bmod n^2$. This property has been widely used in designing Paillier Homomorphic encryption mechanisms, and has also been proven in [31]. Second, for any $\{x_i \in \mathbb{Z}_n\}_{i=1}^m$, we have the following property,

$$\prod_{i=1}^{m}(1 + n \cdot x_i) \equiv \left(1 + n \cdot \sum_{i=1}^{m} x_i\right) \text{ mod } n^2. \tag{3}$$

This property can be proven by mathematical induction, and the detailed proof is given in [27].

5. PPTDS-I scheme

The proposed work in this paper has two schemes. Here we first present the working of PPTDS-I, followed by detailed analysis of security and efficiency. The goal of PPTDS-I scheme is to accurately estimate truthful values for objects, while preventing the disclosure of sensory data and weight information to other system entities. Similar to [30,44,45], we also involve two non-colluding cloud platforms to realize this goal. As shown in Fig. 2, S_A and S_B are denoted as two cloud platforms. After receiving the distance (i.e., $\sum_{m=1}^{M} d(x_m^k, x_m^*)$) calculated by each user, S_A and S_B cooperatively calculate the summed distances without letting each cloud obtain the raw data. With this summed distance, each user is able to compute their weight and weighed data. After receiving the weight information, S_A and S_B will estimate the object truths via a similar cooperation mechanism.

5.1. PPTDS-I mechanism

The work flow of PPTDS-I can be observed in Fig. 2. We introduce the detailed procedure of PPTDS-I from two aspects: Initialization phase and Iteration phase.

5.1.1. Initialization phase

TA is a trusted third entity, and it is assumed to bootstrap the whole system. Given two security numbers κ and *l*, TA first selects two safe prime numbers *p, q* as $p = 2p' + 1, q = 2q' + 1$, where $|p| = |q| = \kappa$, and p', q' are also two prime numbers. Then TA calculates $n = pq$ and $\lambda = lcm(p-1, q-1) = 2p'q'$. Assume there are *K* users, TA generates *K* random values [s_1, s_2, $\cdots$, s_K], such that,

$$\sum_{k=1}^{K} s_k \equiv 0 \bmod \lambda. \tag{4}$$

TA selects a random number *h*, and calculates $h^{n \cdot s_k}$ for each user u_k. In addition, TA chooses a secure cryptographic hash function *H*: $\{0,1\}^* \to \{0,1\}^l$, selects a random value r_k for u_k, and computes its corresponding key as $t_k = H(r_k)$. After that, $h^{n \cdot s_k}$ and t_k are sent to u_k, λ is sent to S_A, and $\{t_k\}_{k=1}^K$ are sent to S_B. Then, the following steps are executed.

Step I: Each user u_k selects random values α_k, $\{\beta_m^k\}_{m=1}^M$, and γ_k to perturb the distance, weighted data and weight respectively in each iteration. Then, it encrypts the random values as,

$$c_{k1} = (1 + n \cdot \alpha_k) \cdot h^{n \cdot s_k} \bmod \ n^2, \tag{5}$$

$$c_{k2,m} = (1 + n \cdot \beta_m^k) \cdot h^{n \cdot s_k} \bmod \ n^2, \tag{6}$$

$$c_{k3} = (1 + n \cdot \gamma_k) \cdot h^{n \cdot s_k} \bmod \ n^2, \tag{7}$$

where $1 \le m \le M$. The key t_k is used to compute $C_{k1} = \text{AES}_{t_k}(c_{k1})$, $C_{k2} = \text{AES}_{t_k}(c_{k2})$, and $C_{k3} = \text{AES}_{t_k}(c_{k3})$. Following this, the ciphertexts are uploaded to S_B. Since the random values are doubly encrypted, they are only known to the users.

Step II: After receiving the ciphertexts from each user, S_B begins to decrypt the ciphertexts by using t_k, and calculates summations $\sum_{k=1}^K \alpha_k$, $\{\sum_{k=1}^K \beta_m^k\}_{m=1}^M$, and $\sum_{k=1}^K \gamma_k$. Particularly, we select c_{k1} to show how to obtain the aggregated values,

$$\begin{aligned} S_1 &= \prod_{k=1}^{K} c_{k1} \bmod n^2 \\ &= \prod_{k=1}^{K} (1 + n \cdot \alpha_k) \cdot h^{n \cdot s_k} \bmod n^2 \\ &= \prod_{k=1}^{K} (1 + n \cdot \alpha_k) \cdot h^{n \cdot \sum_{k=1}^K s_k} \bmod n^2. \end{aligned} \tag{8}$$

Note that $\sum_{k=1}^K s_k = L \cdot \lambda$, where *L* is an integer. Thus, Eq. (8) equals to $1 + n \cdot \sum_{k=1}^K \alpha_k \bmod n^2$, and we have

$$\sum_{k=1}^{K} \alpha_k = \frac{S_1 - 1}{n}. \tag{9}$$

Similarly, $\{\sum_{k=1}^K \beta_m^k\}_{m=1}^M$ and $\sum_{k=1}^K \gamma_k$ can be also obtained. Then, the summations are sent to S_A. Note that, the above procedures are only executed once during the whole PPTDS-I procedure.

5.1.2. Iteration phase

In this phase, S_A first initializes the ground truths for all objects, and delivers them to each user.

Step Q1: When a user u_k receives the object truths, it first calculates the distance between x_m^k and x_m^* as $d(x_m^k, x_m^*) = (x_m^k - x_m^*)^2$, and then computes the summation of distances for all objects (i.e., $\sum_{m=1}^M d(x_m^k, x_m^*)$). To protect users' privacy, the summation is further perturbed by adding the random value (i.e., α_k) before sending to S_A.

Step Q2: After receiving the perturbed distance from each user, S_A aggregates all ciphertexts to obtain $\sum_{k=1}^K \sum_{m=1}^M d(x_m^k, x_m^*) + \sum_{k=1}^K \alpha_k$, and computes the summation of distances (i.e., $\sum_{k=1}^K \sum_{m=1}^M d(x_m^k, x_m^*)$) by subtracting $\sum_{k=1}^K \alpha_k$, which has been obtained in the initialization phase (see Eq. (9)). Then, S_A sends the summed distance to all users.

Step Q3: On receiving the summation, u_k will calculate its weight w_k according to Eq. (1), and compute the weighted data as $\{w_k \cdot x_m^k\}_{m=1}^M$, which will be further perturbed by adding γ_k and $\{\beta_m^k\}_{m=1}^M$ respectively. The perturbed weight information is then sent to S_A.

Step Q4: Based on the ciphertexts received from all users, S_A first calculates the summation of perturbed weighted data as $\{\sum_{k=1}^K (w_k \cdot x_m^k + \beta_m^k)\}_{m=1}^M$. Similarly, the summation of weighted data (i.e., $\{\sum_{k=1}^K w_k \cdot x_m^k\}_{m=1}^M$) can be obtained by subtracting $\{\sum_{k=1}^{K}\beta_m^k\}_{m=1}^{M}$. Following the same analysis, the summation of weight can be obtained by calculating $\sum_{k=1}^{K}(w_k+\gamma_k)-\sum_{k=1}^{K}\gamma_k$. With the above information, S_A is able to estimate the object truths $\{x_m^*\}_{m=1}^{M}$ according to Eq. (2), and then sends them to all users.

In this phase, step Q1 ~ Q4 are iterated until the convergence criterion is satisfied. Note that, we only consider continuous data in PPTDS-I. If the sensory data provided by users are categorical, the distance is different, which can be easily computed by using the equation $d(x_m^k, x_m^*) = (x_m^k - x_m^*)^T(x_m^k - x_m^*)$. Thus, to a certain extent, the categorical data is another special case in our proposed scheme. Combing the above phases, the general procedure of PPTDS-I can be described by Algorithm 2.

Algorithm 2: PPTDS-I Algorithm.

```
Input: Sensory data: ${x_m^k}_{m,k=1}^{M,K}$
Output: Object truths: ${x_m^*}_{m=1}^{M}$
Step I: Each user generates random values and sends them to $S_B$ after encryption;
Step II: $S_B$ aggregates the encrypted random values and delivers them to $S_A$ for decryption;
Step Q1: Each user calculates the difference between $x_m^k$ and $x_m^*$, and perturbs it before sending to $S_A$;
Step Q2: $S_A$ aggregates all perturbed distances and calculates the summation of distance by subtracting the summed
    random values. Then, $S_A$ sends the summed distance to all users;
Step Q3: Each user calculates the weight and weighted data, and perturbs them by adding random values. Then, the
    perturbed weight information is sent to $S_A$;
Step Q4: $S_A$ calculates the object truths based on the perturbed weight information and summed random values, and
    then sends the object truths ${x_m^*}_{m=1}^{M}$ to all users;
Repeat Step Q1 ~ Q4 until convergence criterion is satisfied.
return the object truths ${x_m^*}_{m=1}^{M}$;
```

Fault tolerance: Due to device malfunctioning, network delay, or intent to break the system, some users (u_b and u_c for example) may not submit their data in a given time frame. In this case, the summations calculated by S_B cannot be used, as the perturbed data of u_b and u_c are not available. To deal with this problem, S_A explicitly reports the information that "u_b and u_c have not provided their data" to S_B and S_B will recompute the summations as

$$\begin{aligned} S_1' &= \prod_{k=1,k\neq b,c}^{K} c_{k1} \bmod n^2 \\ &= \left(1+n\cdot\sum_{k=1,k\neq b,c}^{K}\alpha_k\right)\cdot h^{n\cdot\sum_{k=1,k\neq b,c}^{K}s_k} \bmod n^2, \end{aligned} \tag{10}$$

$$S_{2,m}' = \left(1+n\cdot\sum_{k=1,k\neq b,c}^{K}\beta_m^k\right)\cdot h^{n\cdot\sum_{k=1,k\neq b,c}^{K}s_k} \bmod n^2, \tag{11}$$

$$S_3' = \left(1+n\cdot\sum_{k=1,k\neq b,c}^{K}\gamma_k\right)\cdot h^{n\cdot\sum_{k=1,k\neq b,c}^{K}s_k} \bmod n^2. \tag{12}$$

Then, S_1' together with $\{S_{2,m}'\}_{m=1}^{M}$ and S_3' are sent to S_A. Note that, S_B will not response if S_A sends a report that only one user does not upload its data, since S_A can obtain the user's random value by calculating the difference between two returned aggregations.

On receiving the above information, S_A uses the secret key λ to decrypt the summation as

$$\begin{aligned} S'' &= \left(1+n\cdot\sum_{k=1,k\neq b,c}^{K}M_k\right)^{\lambda}\cdot h^{\lambda\cdot n\cdot\sum_{k=1,k\neq b,c}^{K}s_k} \bmod n^2 \\ &= \left(1+n\cdot\lambda\cdot\sum_{k=1,k\neq b,c}^{K}M_k\right) \bmod n^2, \end{aligned} \tag{13}$$

where M_k denotes α_k, $\{\beta_m^k\}_{m=1}^{M}$, or γ_k, and M_k can be obtained by computing

$$\sum_{k=1,k\neq b,c}^{K} M_k = \frac{S''-1}{n\cdot\lambda}. \tag{14}$$

Thus, even though some users may not submit their data timely, our proposed PPTDS-I is still workable. In summary, the proposed PPTDS-I achieves fault-tolerance.

5.2. Security analysis

In order to prove that the proposed PPTDS-I scheme can protect the sensory data & weight information of each user, we perform the following analysis.

Theorem 1. *Suppose there are $K > 3$ users, and at least two users provide sensory data for each object. PPTDS-I guarantees privacy of each user's sensory data and weight information if there is no collusion among the entities.*

Proof. First, we establish that the private information can be protected on user side. For a user u_k, besides the sensory data $\{x_m^k\}_{m=1}^{M}$ and the weight w_k, it is aware of the object truths $\{x_m^*\}_{m=1}^{M}$ and the summation of distances (i.e., $\sum_{k=1}^{K}\sum_{m=1}^{M} d(x_m^k, x_m^*)$). However, based on the assumption that the number of participating users is more than 3, u_k cannot know other users' private information.

Then, for the cloud S_B, the data provided by each user and the aggregated ciphertext can be expressed as a valid Paillier ciphertext i.e., $c_k = (1 + n \cdot M_k) \cdot h^{n \cdot s_k}$, if we consider the item $\alpha_k, \beta_m^k, \gamma_k$, and the aggregated random values as a message M_k. Since the secret key λ is only known by S_A, S_B cannot recover M_k, i.e., it cannot obtain each user's individual random values. Although S_B can obtain the plaintexts of $\sum_{k=1}^{K} \alpha_k, \sum_{k=1}^{K} \beta_m^k$, and $\sum_{k=1}^{K} \gamma_k$ by using the properties under the modulo n^2, it still has no way to recover each individual random value from these summations. For the cloud S_A, besides the perturbed data $\{\sum_{m=1}^{M} d(x_m^k, x_m^*) + \alpha_k\}_{k=1}^{K}, \{w_k \cdot x_m^k + \beta_m^k\}_{k,m=1}^{K,M}$, and $\{w_k + \gamma_k\}_{k=1}^{K}$, S_A also receives the summations of random values from S_B. Similarly, based on these summations, it cannot deduce any information which is private to each user. Moreover, S_A may observe the perturbed random values sent from each user. However, since the random value is encrypted as $C_k = \mathrm{AES}_{t_k}(c_k)$, as long as there is no collusion between S_A and S_B, it cannot obtain the random value even though it holds the secret key λ.

From the above analysis, the presented PPTDS-I scheme is able to protect the privacy of the sensory data and weight information of each user. □

5.3. Efficiency analysis

In order to evaluate the efficiency of the proposed PPTDS-I scheme in terms of computational cost and communication overhead, we perform the following analysis.

5.3.1. Computational costs

The computations performed on the user end are all in plaintext. Hence, compared against traditional ciphertext-based calculations, fewer computational cost will be introduced. More specifically, during the initialization phase, every user generates some random values, and encrypts them by performing addition and multiplication operations, which costs $O(1)$ for distances, $O(M)$ for weighted data, and $O(1)$ for weight. Although big integer multiplications are used in our scheme, we emphasize the fact that, they are performed only once during the whole process. In the iteration phase, each user calculates the perturbed distance, weighted data, and weight information to calculate the weight w_k, which cost $O(M)$, $O(M)$, and $O(1)$ respectively.

Contrary to this, in the cloud since ciphertexts are valid Paillier ciphertexts, overhead is observed, as it dominates the overall computational costs. For cloud S_B, in the initialization phase, it performs $O(KM)$ decryption operations to obtain the perturbed random values, and $O(KM)$ ciphertext multiplications to calculate the summations. For the cloud S_A, $O(K)$ additions are required to aggregate $\{\sum_{m}^{M} d(x_m^k, x_m^*) + \alpha_k\}_{k=1}^{K}, \{w_k + \gamma_k\}_{k=1}^{K}$, and $O(KM)$ additions are needed to aggregate $\{w_k \cdot x_m^k + \beta_m^k\}_{k,m=1}^{K,M}$ in each iteration.

5.3.2. Communication overhead

To evaluate the communication overhead, we mainly consider the amount of communication instances between different parties in PPTDS-I.

On the user side, in the initialization phase, each user uploads encrypted random values to S_B, which is a one-time cost. In each iteration, perturbed data (i.e., $\sum_{m=1}^{M} d(x_m^k, x_m^*) + \alpha_k, \{w_k \cdot x_m^k + \beta_m^k\}_{m=1}^{M}$, and $w_k + \gamma_k$) is required to be uploaded to S_A. Thus, the total number of communication instances introduced on each user is $4t + 2$, where t is the number of iterations. In the initialization phase for S_B, it delivers the summations of random values to S_A. In each iteration, S_A needs to deliver the summed distance, together with the estimated object truths to each user. Hence, the total number of communication instances for cloud S_A and S_B is $4t + 2$ and 1 respectively.

6. PPTDS-II framework

Although PPTDS-I achieves high efficiency on the user side, each user still needs to perform calculations on distance, weight, and weighted data, let alone the upload of data to the cloud for truth and weight update. Thus, we design PPTDS-II

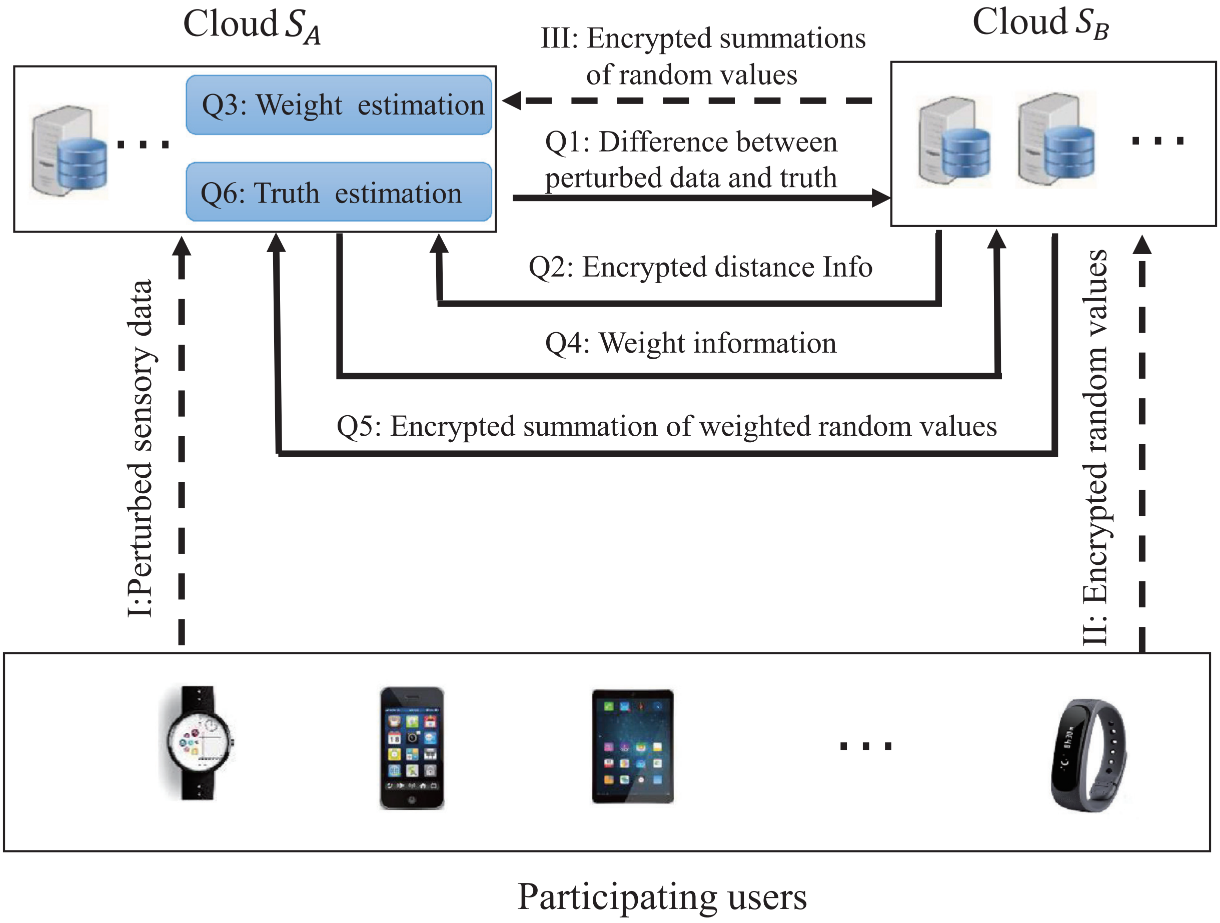
Cloud S_A
Cloud S_B
Q3: Weight estimation
Q6: Truth estimation
III: Encrypted summations of random values
Q1: Difference between perturbed data and truth
Q2: Encrypted distance Info
Q4: Weight information
Q5: Encrypted summation of weighted random values
I:Perturbed sensory data
II: Encrypted random values
Participating users

Fig. 3. System model of PPTDS-II. (The labels of data flow correspond to steps shown in Algorithm 3).

to create a more efficient truth discovery scheme suited for scenarios where only the sensory data needs protection. Similar to PPTDS-I, PPTDS-II also contains the initialization and iteration phase, as illustrated in Fig. 3.

6.1. *Initialization phase*

Step I: u_k generates random values $\{\alpha_m^k\}_{m=1}^M$ for the objects $\{o_m\}_{m=1}^M$. Then, the sensory data is perturbed by adding random values (i.e., $\widetilde{x}_m^k = x_m^k + \alpha_k$), and all perturbed data is sent to S_A.

Step II: To protect the privacy of random values, u_k selects random values s_m^k and r_m^k to encrypt α_m^k as $C_{m1}^k = \text{AES}_{t_k}((1 + n \cdot \alpha_m^k) \cdot h^{n \cdot s_m^k})$, and $(\alpha_m^k)^2$ as $C_{m2}^k = \text{AES}_{t_k}((1 + n \cdot (\alpha_m^k)^2) \cdot h^{n \cdot r_m^k})$. Then, $\{C_{m1}^k, C_{m2}^k\}_{m=1}^M$ are delivered to S_B.

Step III: S_B first decrypts the ciphertexts and obtains the perturbed random values. Then, the process of Eq. (8), S_B calculates the ciphertext of summed $(\alpha_m^k)^2$ for each user u_k as $\prod_{m=1}^M (1 + n \cdot (\alpha_m^k)^2) \cdot h^{n \cdot r_m^k} \bmod n^2$, which equals to $(1 + n \cdot \sum_{m=1}^M (\alpha_m^k)^2) \cdot h^{n \cdot \sum_{m=1}^M r_m^k} \bmod n^2$. Then S_B delivers the summed ciphertext to S_A.

Step IV: On receiving the ciphertext, S_A recovers each user's summation of $(\alpha_m^k)^2$, i.e., $\sum_{m=1}^M (\alpha_m^k)^2$, as $\frac{(1+n \cdot \sum_{m=1}^M (\alpha_m^k)^2)^\lambda \cdot h^{\lambda \cdot n \cdot \sum_{m=1}^M r_m^k} \bmod n^2 - 1}{n \cdot \lambda}$.

6.2. *Iteration phase*

S_A starts this phase by generating random object truths (i.e., $\{x_m^*\}_{m=1}^M$).

Step Q1: Based on the object truths, S_A first calculates $\{\widetilde{x}_m^k - x_m^*\}_{k,m=1}^{K,M}$, and then sends them to S_B.

Step Q2: S_B calculates the ciphertext C_k for each user u_k as

$$
\begin{aligned}
C_k &= \prod_{m=1}^{M} \left(\left(1 + n \cdot \alpha_m^k\right) \cdot h^{n \cdot s_m^k} \right)^{2(\widetilde{x}_m^k - x_m^*)} \mod n^2 \\
&= \prod_{m=1}^{M} \left(1 + n \cdot 2\left(\widetilde{x}_m^k - x_m^*\right) \cdot \alpha_m^k\right) \cdot h^{n \cdot s_m^k \cdot 2(\widetilde{x}_m^k - x_m^*)} \mod n^2 \\
&= \left(1 + n \cdot \sum_{m=1}^{M} 2\left(\widetilde{x}_m^k - x_m^*\right) \cdot \alpha_m^k\right) \cdot h^{n \cdot \sum_{m=1}^{M} s_m^k \cdot 2(\widetilde{x}_m^k - x_m^*)} \mod n^2.
\end{aligned} \tag{15}
$$

Then, all the ciphertexts $\{C_k\}_{k=1}^K$ are sent to S_A.

Step Q3: After receiving the ciphertexts, similar to step IV, S_A can also use λ to obtain $\sum_{m=1}^{M} 2(\widetilde{x}_m^k - x_m^*) \cdot \alpha_m^k$ by computing $\frac{C_k^\lambda - 1}{n \cdot \lambda}$. Then, each user's summation of distance can be calculated as

$$
\begin{aligned}
\mathrm{sumd}_k &= \sum_{m=1}^{M} \left(\widetilde{x}_m^k - x_m^*\right)^2 - \sum_{m=1}^{M} 2\left(\widetilde{x}_m^k - x_m^*\right)\alpha_m^k + \sum_{m=1}^{M} (\alpha_m^k)^2 \\
&= \sum_{m=1}^{M} \left(x_m^k - x_m^*\right)^2.
\end{aligned} \tag{16}
$$

Then, S_A can estimate the weight of u_k following Eq. (2).

Step Q4: Based on the estimated weight, S_A calculates the weighted data for each object (i.e., $\{p_m^k = w_k \cdot (x_m^k + \alpha_m^k)\}_{k,m=1}^{K,M}$), and then sends $\{w_k\}_{k=1}^K$ to S_B.

Step Q5: Similar to Eq. (15), S_B calculates the ciphertext C_m for each object o_m as $\prod_{k=1}^K ((1 + n \cdot \alpha_m^k) \cdot h^{n \cdot s_m^k})^{w_k} \mod n^2$. Then, the ciphertexts of all objects (i.e., $\{C_m\}_{m=1}^M$) are sent to S_A.

Step Q6: After receiving the data from S_B, S_A begins to update the object truth as

$$
\begin{aligned}
x_m^* &= \frac{\sum_{k=1}^K p_m^k - \frac{C_m^\lambda - 1}{n \cdot \lambda}}{\sum_{k=1}^K w_k} \\
&= \frac{\sum_{k=1}^K w_k \cdot (x_m^k + \alpha_m^k) - \sum_{k=1}^K w_k \cdot \alpha_m^k}{\sum_{k=1}^K w_k} \\
&= \frac{\sum_{k=1}^K w_k \cdot x_m^k}{\sum_{k=1}^K w_k}.
\end{aligned} \tag{17}
$$

In this phase, step Q1~Q6 will be iteratively conducted until the number of desired iteration is reached or convergence criterion is satisfied. The detailed procedure of PPTDS-II can be seen in Algorithm 3.

Algorithm 3: PPTDS-II Algorithm.

```
Input: Sensory data: $\{x_m^k\}_{m,k=1}^{M,K}$
Output: Object truths: $\{x_m^*\}_{m=1}^M$
Step I: Each user perturbs the sensory data using random values, and then sends the perturbed data to $S_A$;
Step II: Each user encrypts the random values and sends the ciphertexts to $S_B$;
Step III: $S_B$ aggregates the encrypted random values and delivers the result to $S_A$;
Step IV: $S_B$ decrypts the ciphertext to obtain the summed random value;
Step Q1: $S_A$ calculates $\{\widetilde{x}_m^k - x_m^*\}_{m,k=1}^{M,K}$ and sends them to $S_B$.
Step Q2: $S_B$ calculates the encrypted distance Info and sends them to $S_A$;
Step Q3: $S_A$ estimates the weight information;
Step Q4: $S_A$ sends the weight to $S_B$;
Step Q5: $S_B$ calculates the weighted random value in ciphertexts and sends them to $S_A$ after aggregation;
Step Q6: $S_A$ performs the truth estimation;
Repeat Step Q1 $\sim$ Q6 until convergence criterion is satisfied.
return the object truths $\{x_m^*\}_{m=1}^M$.
```

Fault tolerance: PPTDS-II can achieve fault tolerance by performing similar operations as in PPTDS-I.

6.3. Security analysis

Theorem 2. *Suppose there are $K > 3$ users and $M > 3$ objects, and at least two users provide sensory data for at least two objects. PPTDS-II guarantees the privacy of each user's sensory data if there is no collusion among the entities.*

Proof. In PPTDS-II, each user only takes part in the initialization phase. Thus, we only need to prove that the sensory data is not disclosed to the cloud.

For the cloud S_A, it knows the ciphertexts $\{\widetilde{x}_m^k\}_{k,m=1}^{K,M}$ and $\{C_{m1}^k, C_{m2}^k\}_{k,m=1}^{K,M}$. Without random values $\{\alpha_m^k\}_{k,m=1}^{K,M}$ and the AES secret keys $\{t_k\}_{k=1}^K$, S_A cannot decrypt the above ciphertexts. Besides the ciphertexts, S_A also knows the plaintexts $\{\sum_{m=1}^M (\alpha_m^k)^2\}_{k=1}^K$, $\{\sum_{k=1}^K 2(\widetilde{x}_m^k - x_m^*) \cdot \alpha_m^k\}_{m=1}^M$, $\sum_{k=1}^K w_k \cdot x_m^k$, $\{w_k\}_{k=1}^K$. However, with the assumption that the number of users and objects is larger than 3, S_A cannot learn anything about each user's sensory data based on these summations.

For the cloud S_B, it knows the ciphertexts $\{C_{m1}^k, C_{m2}^k\}_{k,m=1}^{K,M}$, $\{\widetilde{x}_m^k - x_m^*\}_{k,m=1}^{K,M}$, $\{C_k\}_{k=1}^K$, $\{C_m\}_{m=1}^M$, and the plaintexts $\{w_k\}_{k=1}^K$. Nevertheless, without the secret key λ and the random value $\{\alpha_m^k\}_{k,m=1}^{K,M}$, the above ciphertexts cannot be decrypted. Hence, each user's sensory data will not be disclosed to cloud S_A and S_B. In summary, PPTDS-II can guarantee the privacy of each user's sensory data. □

6.4. Efficiency analysis

The efficiency of this scheme is also evaluated in terms of computational and communication costs.

6.4.1. Computational costs

In PPTDS-II, each participating user is only involved in the initialization phase. Following similar analysis in PPTDS-I, each user costs $O(M)$ to encrypt $\{\alpha_m^k\}_{m=1}^M$ and $\{(\alpha_m^k)^2\}_{m=1}^M$, respectively. S_B is involved in both initialization and iteration phases. In the initialization phase, it needs to conduct $O(KM)$ decryption operations to recover the perturbed random values, and $O(KM)$ ciphertext multiplication operations to obtain $\{\sum_{m=1}^M (\alpha_m^k)^2\}_{k=1}^K$. In the iteration phase, to calculate $\{C_k\}_{k=1}^K$ and $\{C_m\}_{m=1}^M$, S_B needs to perform $O(KM)$ exponentiations and $O(KM)$ ciphertext multiplications respectively. For S_A, in the initialization phase, it needs to perform $O(K)$ ciphertext exponentiations to recover $\{\sum_{m=1}^M (\alpha_m^k)^2\}_{k=1}^K$. In the iteration phase, S_A first takes $O(KM)$ to calculate $\{\widetilde{x}_m^k - x_m^*\}_{k,m=1}^{K,M}$, and conducts $O(K)$ and $O(M)$ ciphertext exponentiations to decrypt $\{C_k\}_{k=1}^K$ and $\{C_m\}_{m=1}^M$ respectively. Additionally, it costs $O(M)$ to compute the object truths $\{x_m^*\}_{m=1}^M$.

6.4.2. Communication overhead

In PPTDS-II, as each user is only involved in the initialization phase, it only sends $\{\widetilde{x}_m^k\}_{m=1}^M$, and $\{C_{m1}^k, C_{m2}^k\}_{m=1}^M$ to S_A and S_B respectively. Hence, the total communication instances between each user and the cloud platforms is only 2. Note that, this is a one-time cost during the whole truth discovery process. For S_B, in the initialization phase, it sends $\{\prod_{m=1}^M C_{m2}^k\}_{k=1}^K$ to S_A. In the iteration phase, to calculate the weight information, S_A needs to send the perturbed data to S_B, while S_B returns $\{C_k\}_{k=1}^K$ in each iteration. To update the object truths, S_A sends K user weights to S_B, while S_B returns M ciphertexts $\{C_m\}_{m=1}^M$. Thus, the total number of communication instances between cloud S_A and S_B is $4t + 1$.

7. Performance evaluation

In this section, we evaluate our proposed PPTDS schemes in comparison to the PPTD scheme [29]. All schemes are implemented in Java, and the experiments are conducted on an Intel Core i7 2.5GHz system. The bit-length of n and l is set as 512 and 128 respectively. We set the number of iteration as 10, and obtain the average results for comparative analysis.

Fig. 4(a) illustrates the running time against varying number of users with the number of objects fixed at 100. Fig. 4(b) on the other hand, plots the same against increasing number of objects with fixed 100 users. As can be seen, the running time increases with the number of users and objects, since more data needs to be encrypted. In Fig. 4(a), when the number of users reaches 700, PPTDS-I takes 2.468 s to initialize the system, while PPTDS-II needs 3.860 s. The reason is that PPTDS-I executes multiplication operations to compute the aggregated random values. But for PPTDS-II, ciphertext exponentiations are performed. Similarly, when the number of objects ranges from 100 to 700, PPTDS-I also performs better than PPTDS-II, as illustrated in Fig. 4(b).

In Fig. 5(a) and (b), we plot the running time by varying the number of users and objects in the iteration phase, where the number of objects and users is set as 100 in (a) and (b) respectively. From Fig. 5(a), it is observed that the running time increases with varying degrees. PPTD consumes the most time among the three schemes. For example, when the number of users reaches 700, PPTD needs 1772.952 s, while PPTDS-I and PPTDS-II need 0.286 s and 44.517 s respectively. This is because in PPTD each user needs to perform time-consuming (p, t)-threshold Paillier cryptosystem to decrypt the ciphertexts, while only ciphertext multiplications are required in the other two schemes. As PPTDS-II needs to perform ciphertext exponentiations to compute the distance and weighted data, it costs more time than PPTDS-I. Similarly, from Fig. 5(b), it can be seen that the running time of PPTD is more than that of PPTDS when the number of objects is in the range of 100–700. When it reaches 700, PPTDS-I and PPTDS-II need 0.258 s and 39.361 s respectively, while PPTD takes 419.912 s. This also confirms the efficiency of the proposed schemes. Although PPTDS-II takes more time than PPTDS-I, we

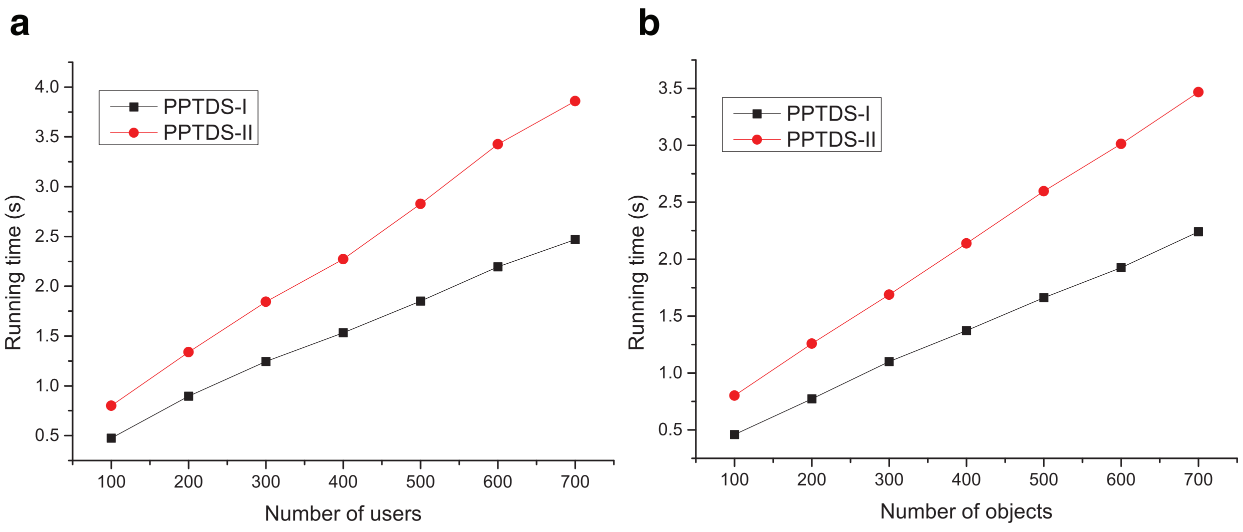

Fig. 4. (a) Running time with varying number of users in initialization phase. (b) Running time with varying number of objects in initialization phase.

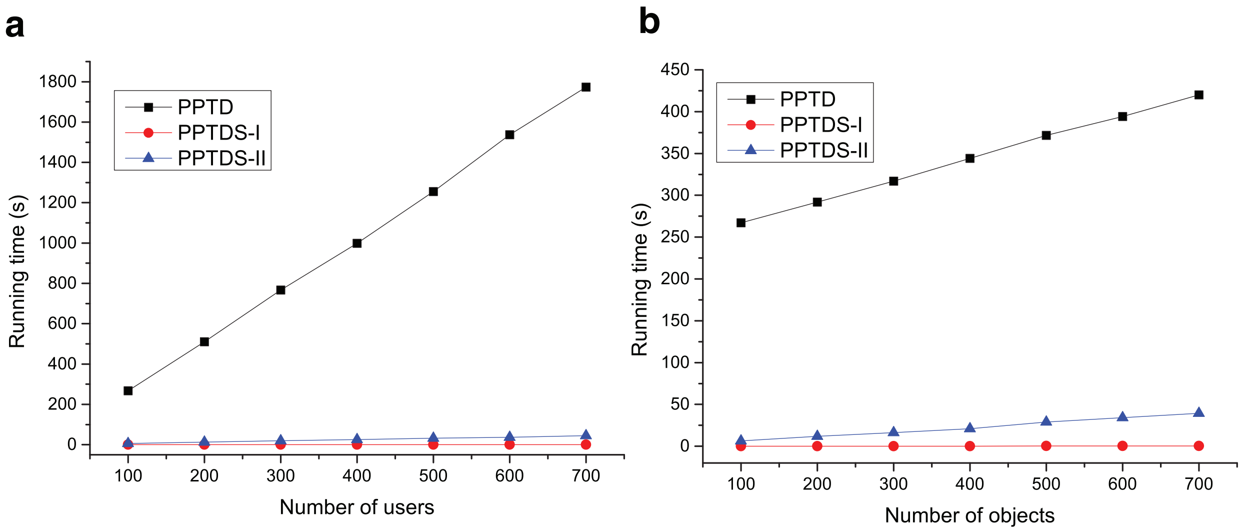

Fig. 5. (a) Running time with varying number of users in iteration phase. (b) Running time with varying number of objects in iteration phase.

emphasize that all operations in PPTDS-II are executed in the cloud and the users do not need to participate in the iterative process, which makes it suitable for resource constrained devices on user side.

8. Conclusion

In this paper, we investigate how to realize efficient and privacy-preserving truth discovery in crowd sensing systems. Our first design is fit for the scenarios where users can join the iterative truth discovery procedure. Data perturbation and Paillier cryptosystem are used to protect users' privacy. To further reduce users' computational and communication costs, we also propose another non-iterative truth discovery scheme, which can be applied in the scenarios with requirement of sensory data protection but the users cannot perform large computations. In the future, we aim to address privacy concerns in different truth discovery mechanisms that use types of weight functions.

Acknowledgements

The authors want to thank the associate editor and reviewers for their constructive and generous feedback. This research is supported by the National Natural Science Foundation of China (grant nos. 61402037, 61272512, 61702105), and Graduate Technological Innovation Project of Beijing Institute of Technology (no. 1320012211801).

References

[1] A. Abdrashitov, A. Spivak, Sensor data anonymization based on genetic algorithm clustering with l-diversity, in: 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology, FRUCT 2016, St-Petersburg, Russia, April 18–22, 2016, 2016, pp. 3–8.

[2] S. An, H. Yang, J. Wang, N. Cui, J. Cui, Mining urban recurrent congestion evolution patterns from gps-equipped vehicle mobility data, Inf. Sci. 373 (2016) 515–526.

[3] P. Baier, F. Dürr, K. Rothermel, Efficient distribution of sensing queries in public sensing systems, in: IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems, MASS 2013, Hangzhou, China, October 14–16, 2013, 2013, pp. 272–280.

[4] S. Chatterjee, M. Bhattacharyya, Judgment analysis of crowdsourced opinions using biclustering, Inf. Sci. 375 (2017) 138–154.

[5] S. Chatterjee, A. Mukhopadhyay, M. Bhattacharyya, Dependent judgment analysis: a Markov chain based approach for aggregating crowdsourced opinions, Inf. Sci. 396 (2017) 83–96.

[6] R. Cramer, I. Damgård, J.B. Nielsen, Multiparty computation from threshold homomorphic encryption, in: Advances in Cryptology - EUROCRYPT 2001, International Conference on the Theory and Application of Cryptographic Techniques, Innsbruck, Austria, May 6–10, 2001, Proceeding, 2001, pp. 280–299.

[7] J. Domingo-Ferrer, S. Ricci, C. Domingo-Enrich, Outsourcing scalar products and matrix products on privacy-protected unencrypted data stored in untrusted clouds, Inf. Sci. 436–437 (2018) 320–342.

[8] X. Du, H. Chen, Security in wireless sensor networks, IEEE Wireless Commun. 15 (2008) 60–66.

[9] X. Du, M. Guizani, Y. Xiao, H. Chen, Secure and efficient time synchronization in heterogeneous sensor networks, IEEE Trans. Veh. Technol. 57 (2008) 2387–2394.

[10] X. Du, M. Guizani, Y. Xiao, H. Chen, A routing-driven elliptic curve cryptography based key management scheme for heterogeneous sensor networks, IEEE Trans. Wireless Commun. 8 (2009) 1223–1229.

[11] C. Gao, Q. Cheng, P. He, W. Susilo, J. Li, Privacy-preserving naive bayes classifiers secure against the substitution-then-comparison attack, Inf. Sci. 444 (2018) 72–88.

[12] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, X. Li, Jigsaw: indoor floor plan reconstruction via mobile crowdsensing, in: The 20th Annual International Conference on Mobile Computing and Networking, MobiCom'14, Maui, HI, USA, September 7–11, 2014, 2014, pp. 249–260.

[13] X. Hei, X. Du, Biometric-based two-level secure access control for implantable medical devices during emergencies, in: INFOCOM 2011. 30th IEEE International Conference on Computer Communications, Joint Conference of the IEEE Computer and Communications Societies, 10–15 April 2011, Shanghai, China, 2011, pp. 346–350.

[14] X. Hei, X. Du, J. Wu, F. Hu, Defending resource depletion attacks on implantable medical devices, in: Proceedings of the Global Communications Conference, 2010. GLOBECOM 2010, 6–10 December 2010, Miami, Florida, USA, 2010, pp. 1–5.

[15] J. Hong, K. Xue, L. Wei, Comments on "dac-macs: effective data access control for multiauthority cloud storage systems" /security analysis of attribute revocation in multiauthority data access control for cloud storage systems, IEEE Trans. Inf. Foren.Secur. 10 (2017) 1315–1317.

[16] S. Hu, L. Su, H. Liu, H. Wang, T.F. Abdelzaher, Smartroad: smartphone-based crowd sensing for traffic regulator detection and identification, ACM Trans. Sens. Netw. (TOSN) 11 (2015) 55:1–55:27.

[17] Z. Huang, S. Liu, X. Mao, K. Chen, J. Li, Insight of the protection for data security under selective opening attacks, Inf. Sci. 412 (2017) 223–241.

[18] Y. Kim, W. Jung, K. Shim, Integration of graphs from different data sources using crowdsourcing, Inf. Sci. 385 (2017) 438–456.

[19] C. Li, D. Lin, J. Lu, Cryptanalyzing an image-scrambling encryption algorithm of pixel bits, IEEE MultiMedia 24 (2017) 64–71.

[20] J. Li, X. Huang, J. Li, X. Chen, Y. Xiang, Securely outsourcing attribute-based encryption with checkability, IEEE Trans. Parallel Distrib. Syst. 25 (2014) 2201–2210.

[21] J. Li, J. Li, X. Chen, C. Jia, W. Lou, Identity-based encryption with outsourced revocation in cloud computing, IEEE Trans. Comput. 64 (2015) 425–437.

[22] M. Li, L. Zhu, Z. Zhang, R. Xu, Achieving differential privacy of trajectory data publishing in participatory sensing, Inf. Sci. 400 (2017) 1–13.

[23] Q. Li, Y. Li, J. Gao, B. Zhao, W. Fan, J. Han, Resolving conflicts in heterogeneous data by truth discovery and source reliability estimation, in: International Conference on Management of Data, SIGMOD 2014, Snowbird, UT, USA, June 22–27, 2014, 2014, pp. 1187–1198.

[24] S. Li, K. Xue, Q. Yang, P. Hong, Ppma: privacy-preserving multi-subset aggregation in smart grid, IEEE Trans. Ind. Inf. 14 (2018) 462–471.

[25] S. Li, X. Zhang, K. Xue, L. Zhou, H. Yue, Privacy-preserving prepayment based power request and trading in smart grid, China Commun. 15 (2018) 14–27.

[26] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, J. Han, Conflicts to harmony: a framework for resolving conflicts in heterogeneous data by truth discovery, IEEE Trans. Knowl. Data Eng. 28 (2016) 1986–1999.

[27] R. Lu, K. Heung, A.H. Lashkari, A.A. Ghorbani, A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced iot, IEEE Access 5 (2017) 3302–3312.

[28] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, Y. Cheng, Truth discovery on crowd sensing of correlated entities, in: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, SenSys 2015, Seoul, South Korea, November 1–4, 2015, 2015, pp. 169–182.

[29] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, K. Ren, Cloud-enabled privacy-preserving truth discovery in crowd sensing systems, in: Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems, SenSys 2015, Seoul, South Korea, November 1–4, 2015, 2015, pp. 183–196.

[30] C. Miao, L. Su, W. Jiang, Y. Li, M. Tian, A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems, in: 2017 IEEE Conference on Computer Communications, INFOCOM 2017, Atlanta, GA, USA, May 1–4, 2017, 2017, pp. 1–9.

[31] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2–6, 1999, Proceeding, 1999, pp. 223–238.

[32] M. Savi, C. Rottondi, G. Verticale, Evaluation of the precision-privacy tradeoff of data perturbation for smart metering, IEEE Trans. Smart Grid 6 (2015) 2409–2416.

[33] J. Shen, T. Zhou, X. Chen, J. Li, W. Susilo, Anonymous and traceable group data sharing in cloud computing, IEEE Trans. Inf. Foren. Secur. 13 (2018) 912–925.

[34] Y. Xiao, V.K. Rayi, B. Sun, X. Du, F. Hu, M. Galloway, A survey of key management schemes in wireless sensor networks, Comput. Commun. 30 (2007) 2314–2341.

[35] G. Xu, H. Li, D. Liu, H. Ren, Y. Dai, X. Liang, Towards efficient privacy-preserving truth discovery in crowd sensing systems, in: 2016 IEEE Global Communications Conference, GLOBECOM 2016, Washington, DC, USA, December 4–8, 2016, 2016, pp. 1–6.

[36] G. Xu, H. Li, C. Tan, D. Liu, Y. Dai, K. Yang, Achieving efficient and privacy-preserving truth discovery in crowd sensing systems, Comput. Secur. 69 (2017) 114–126.

[37] J. Xu, L. Wei, Y. Zhang, A. Wang, F. Zhou, C. Gao, Dynamic fully homomorphic encryption-based merkle tree for lightweight streaming authenticated data structures, J. Netw. Comput. Appl. 107 (2018) 113–124.

[38] K. Xue, P. Hong, C. Ma, A lightweight dynamic pseudonym identity based authentication and key agreement protocol without verification tables for multi-server architecture, J. Comput. Syst. Sci. 80 (2014) 195–206.

[39] K. Xue, S. Li, J. Hong, Y. Xue, N. Yu, P. Hong, Two-cloud secure database for numeric-related SQL range queries with privacy preserving, IEEE Trans. Inf. Foren. Secur. 12 (2017) 1596–1608.

[40] S. Yao, M.T.A. Amin, L. Su, S. Hu, S. Li, S. Wang, Y. Zhao, T.F. Abdelzaher, L.M. Kaplan, C.C. Aggarwal, A. Yener, Recursive ground truth estimator for social data streams, in: 15th ACM/IEEE International Conference on Information Processing in Sensor Networks, IPSN 2016, Vienna, Austria, April 11-14, 2016, 2016, pp. 14:1–14:12.

[41] X. Zhang, L. Zhou, S. Li, K. Xue, H. Yue, Privacy-preserving power request and trading by prepayment in smart grid, in: 2017 IEEE/CIC International Conference on Communications in China, ICCC 2017, Qingdao, China, October 22–24, 2017, 2017, pp. 1–6.

[42] Y. Zhang, R. Yu, M. Nekovee, Y. Liu, S. Xie, S. Gjessing, Cognitive machine-to-machine communications: visions and potentials for the smart grid, IEEE Netw. 26 (2012) 6–13.

[43] Y. Zhang, R. Yu, S. Xie, W. Yao, Y. Xiao, M. Guizani, Home M2M networks: architectures, standards, and qos improvement, IEEE Commun. Mag. 49 (2011) 44–52.

[44] Y. Zheng, H. Duan, C. Wang, Learning the truth privately and confidently: encrypted confidence-aware truth discovery in mobile crowdsensing, IEEE Trans. Inf. Foren. Security 13 (2018) 2475–2489.

[45] Y. Zheng, H. Duan, X. Yuan, C. Wang, Privacy-aware and efficient mobile crowdsensing with truth discovery, IEEE Trans. Depend. Secure Comput. PP (2017) 1.

[46] J. Zhou, Z. Cao, X. Dong, X. Lin, Security and privacy in cloud-assisted wireless wearable communications: challenges, solutions, and future directions, IEEE Wireless Commun. 22 (2015) 136–144.