Privacy-Preserving Truth Discovery for Collaborative-Cloud Encryption in Mobile Crowdsensing

Xingting Liu, Siwang Zhou, Wei Zhang, Ting Dong, and Keqin Li, *Fellow, IEEE*

***Abstract*—In mobile crowdsensing (MCS) system, a variety of sensors are required to operate together to glean and upload sensory data to clouds for processing. In real practice, truth discovery has been widely explored to find reliable information from various mobile devices. Under the requirement of MCS security, privacy-preserving truth discovery (PPTD) causes wide concern, which refers to discovering truthful information from these unreliable uploaded data while protecting users' private information. Although many PPTD mechanisms have been proposed, they can either not guarantee low communication from users to the cloud, or fail to realize fully strong privacy protect including sensing data privacy, weight privacy, intermediate privacy, and estimated truth privacy. This study designs a collaborative cloud encryption architecture. In this framework, we propose a new system architecture that adapts a two-cloud peer model while leveraging garbled circuit (GC). Users only need to transfer data to two clouds once, which realizes low users workloads while supporting dynamic users. The two cloud terminals cooperate to complete the weight update through the GC but execute the truth discovery algorithm, respectively. In this way, the noninteractive comes true and the users' workloads are transferred to the cloud server side. The weight update finish and weight privacy are fulfilled through GC, meanwhile the other intermediate values maintain strong privacy. At the same time, because the collaborative cloud architecture of the two clouds ensures the confidentiality of the truth value in the cloud and the homomorphism at the inquiry side, it ensures the strong privacy of the whole process from users to inquiries. The performance of this proposed method is verified through extensive evaluations.**

***Index Terms*—Crowdsensing, cryptosystem, privacy preserving, truth discovery (TD).**

I. Introduction

With interaction between mobile communication and intelligent terminal technology, the mobile crowdsensing (MCS) systems can monitor the physical world effectively by analyzing the sensory data collected and uploaded from sensors to clouds [1], [2], [3], [4]. Some applications have been developed for certain domains, like environmental monitoring [5], traffic navigation [6], and so on. However, in MCS applications, due to poor quality of sensors, incomplete observation results, background noises, or even intentional deception, among other factors, the sensory data provided by a single participant are often unreliable [7].

For this problem, an intuitive method is to collect sensory data from the same object, and then, use the average or voting method to evaluate an approximate result. However, these methods always treat all users equally, thus failing to provide accurate results. Truth discovery (TD) has been used to extract the truth from data collected with different precisions, including some noisy or even conflicting data [8], [9]. In contrast, the truth discovery method assigns different weights to users iteratively according to the data qualities of different users and then computes the weighted average of all claims as the estimated truth. The truth is determined by iteratively updating the user weight and the ground truth, with a user who is nearer to the truth being given a higher weight. Data from a user with a higher weight has a better probability of being chosen as the truth. This process is repeated until the truth of the estimate converges [10].

Privacy-preserving truth discovery (PPTD) has recently received noteworthy attention to protect users' privacy while distilling truthful information efficiently [10], [11], [12], [13]. Not only sensing data, but also weight data are private information. The weights reflect the reliability of the user providing the data. It is another sensitive piece of information that needs to be securely guarded. The attacker may infer specifics about the education, abilities, and personality attributes of participating users using user reliability information. For instance, combining viewpoints on difficult societal issues could result in a better solution. However, the leakage of reliability might reveal the level of intelligence and education of consumers. Miao et al. [13] introduced the first secure truth discovery scheme to protect both the sampling data and the weight information of participants. The scheme adopts the threshold paillier cryptosystem, but it is not efficient, because too many calculations are needed on the side of users' terminals. Later, they further presented a lightweight PPTD framework to reduce user costs by introducing two noncollusion servers. Zhang et al. [12] proposed

Manuscript received 19 September 2022; revised 26 February 2023; accepted 6 May 2023. Date of publication 18 May 2023; date of current version 30 August 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62172153, and in part by the Hunan Provincial Natural Science Foundation of China under Grant 2021JJ30455. *(Corresponding author: Siwang Zhou.)*

Xingting Liu and Siwang Zhou are with the College of Computer Science and Electrical Engineering, Hunan University, Changsha 410012, China (e-mail: xingtingliu@hnu.edu.cn; swzhou@hnu.edu.cn).

Wei Zhang is with the College of Computer Science and Engineering, Changsha University, Changsha 410012, China (e-mail: zweihnu@hnu.edu.cn).

Ting Dong is with the Hunan Vocational College of Science and Technology, Changsha 410012, China (e-mail: 6938741@qq.com).

Keqin Li is with the Department of Computer Science, State University of New York, New Paltz, NY 12561 USA (e-mail: lik@newpaltz.edu).

Digital Object Identifier 10.1109/JSYST.2023.3274812

TABLE I
COMPARISON WITH OTHER RELATED WORKS

Scheme	Sensory data privacy	Weight privacy	User dynamics	Noninteractive	Truth privacy	Users and Inquirers overload
PPTD [11]	✓	✓	×	×	×	×
LPTD [12]	✓	✓	✓	×	×	×
RPTD-I [14]	✓	✓	✓	×	×	×
RPTD-II [14]	✓	✓	✓	✓	×	✓
NPPTD [17]	✓	✓	✓	✓	×	✓
RPPTD [21]	✓	✓	✓	✓	✓	×
ECTD [20]	✓	✓	✓	✓	✓	×
CEPTD(our work)	✓	✓	✓	✓	✓	✓

a lightweight and practical PPTD framework, which can not only protects devices' privacy but also achieves high efficiency. Unfortunately, all of these scenarios entail frequent communications between users and cloud servers to ensure smooth implementation. In a real-world application, users may not be able to send data to clouds in time due to unreliable network connections, human interventions, sensor device battery problems, and other reasons [14].

Recently, many lightweight approaches are explored to deduce the overloads, especially the users' workload because of the limited resources [14], [15], [16], [19]. These methods always adopt a two-server model instead of a single server model, to shift most of the users' workload to the cloud. Tang et al. [17] constructed a noninteractive PPTD system that follows the two noncolluding server architecture and leverages Yao's garbled circuit (GC). According to this honest-but-curious model, they completely remove the online requirement and fulfill the privacy of providers and intermediate values. Liu et al. [18] also designed a two noncolluding servers scheme against the sparse sensory data model. In sparse data scenarios, their algorithm can solve the privacy issues, which can also guarantee sensory data privacy and weight privacy [18]. However, in these categories of existing methods, the estimated truths are directly exposed to the cloud server, which may lead to leakage or abuse of the truths. A curious cloud may try to spy on private information, which produces cloud attacks.

In the literature, only a few works have started to study the privacy of estimated truths in cloud servers. Zheng et al. [20] developed a noninteractive truth discovery algorithm. In their system architecture, users send sensing data to the cloud in the form of encryption, which is followed by the production of confidence-aware truth discovery (CATD) in the encryption domain. While encrypted sensing data transfer to the cloud, the scheme performs CATD in the encrypted domain. The requesters finally decrypt the encrypted inferred truths [20]. The disadvantage of this approach is that the decoding process is implemented on the query side, but in many scenarios, the query sides are resource-limited devices such as mobile phones or smart watches, which cannot load complex decoding tasks. Liu et al. [21] fulfill estimated truths privacy in the cloud but also need the requesters to decrypt. Then the requesters' workload depends on the complexity of the decryption algorithms. Other solutions that address the privacy issue of estimated truth values require honest-but-curious third parties [22]. They also realize the estimated truths privacy in the cloud with the assistance of fog nodes. But trusted third parties such as fog nodes or other devices are not pervasive and can also introduce many other costs.

Most of the existing PPTD works do not guarantee strong privacy protection for the whole truth discovery process, especially the privacy of the reconstructed truths to the cloud. Some recent work achieves privacy preservation for true values, for example, [22] and [23] achieve privacy protection of the estimated truth to clouds, but they either require additional security devices or add a lot of computational overhead on both the user side and the query side. Table I shows the comparison of prior private truth discovery schemes, where we compare the whole strong privacy, practicability, and reliability. From Table I, we can see comparison of prior private truth discovery schemes, where we compare the whole strong privacy, practicability, and reliability. Table I shows that a practical truth discovery scheme that can achieve high efficiency while also providing strong privacy protection for the whole process to be investigated.

To deal with the aforementioned challenges, i.e., high user overloads, two unequal clouds, failure to prevent cloud attacks, or rely on third parties. In this article, we propose a collaborative-cloud encryption privacy-preserving truth discovery (CEPTD) approach in MCS. The characteristics of this approach are strong privacy and accuracy guarantees under low users' workloads and requesters' workloads. In addition, we propose a two-cloud peer model, which applies to all truth discovery algorithms, and is available to prevent transmission attacks where eavesdroppers manipulate the sensory data transmitted from the users to the cloud. The contributions of this article can be summarized as follows.

First, we design a new client encryption mechanism, which splits each sampled data from each user into two unrelated data using masking keys for transmission to the cloud. And the masking keys would use only once, which could guarantee the strong privacy of transmission from users to clouds.

Secondly, we shift most of the users' workloads to clouds, which can contribute to achieving a noninteractive truth discovery algorithm. We also realize the synchronous update of weights in the collaborative cloud structure, and finally, ensure the homomorphism from the user side to the query side, so that the inquires can perform a simple homomorphic operation to obtain the final truth values. This mechanism ensures that the information is not visible to two clouds, because the final processing for reconstructing truth values is not performed in the cloud.

Third, we conduct a detailed analysis to demonstrate that our proposed CEPTD scheme is practical and guarantees strong

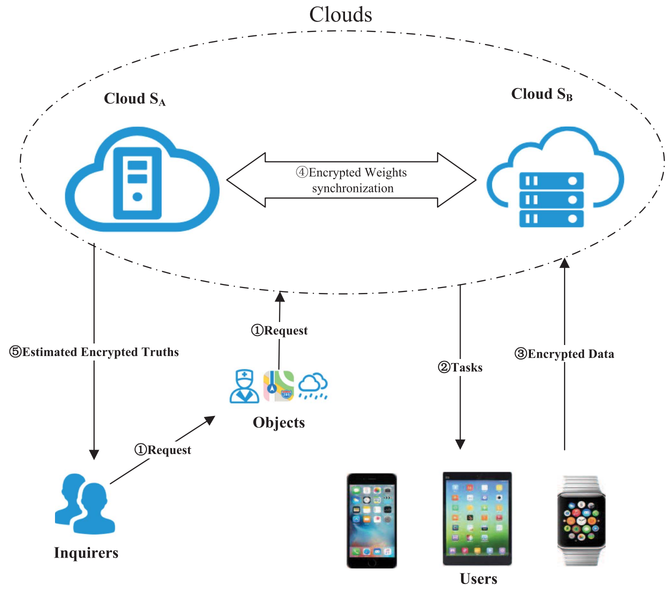

Fig. 1. System architecture of CEPTD.

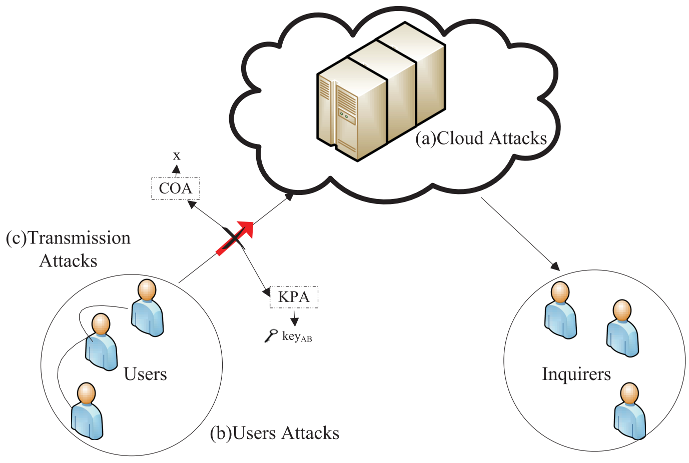

Fig. 2. Threat model. (a) Cloud attacks: Curious cloud tries to spy on private information. (b) Users attacks: Users deduce other observations financially. (c) Transmission attacks: Eavesdroppers manipulate the sensory data transmitted from the users to the cloud. The picture shows two transmission attacks: COA: An eavesdropper tries to recover the sensed claims only by encrypted data, and KPA: A second eavesdropper tries to recover the security key by the sensed claims and encrypted data.

privacy. Also, extensive experiments over real-world mobile crowdsensing datasets demonstrate that our design achieves efficient performance on mobile devices.

The rest of this article is organized as follows. The system architecture and the analysis of the threat model are presented in Section II. In Section III, the background on truth discovery and the GC is provided. Section IV introduces the detail of our system. The theoretical analysis is shown in Section V, and the experimental results are discussed in Section VI. Finally, Section VII concludes this article.

II. System Architecture and Threat Model

In this part, the system architecture is presented, which is followed by the analysis of the privacy attacks therein.

A. System Architecture

The widely used two noncolluding servers are adopted by the system architecture of the CEPTD system, which is honest-but-curious. To prevent transmission attacks and realize the noninteractive nature, users transport the decomposed data to two cloud servers, and the security weights of synchronization are implemented by the GC. Our system workflow is shown in Fig. 1, which consists of four types of entities: inquirers, cloud servers, users, and objects. An inquirer is a unit of the initial crowd and participatory sensing application. It sends requests through the cloud server to the users who collect data required by the inquirer. The cloud servers first issue multiple objects and assign tasks to various users. After that, each user collects the sensor data of the targeted objects and generates random data. In the meantime, it should be noted that the random data help the sample data decomposition. Then, users send the decomposed claims to the two cloud servers, respectively. The two cloud servers cooperate to ensure the encryption and synchronization of weights. Afterward, two servers execute the truth discovery algorithm to receive the results of the truth discovery value, respectively. The inquirer finally reconstructs the estimated truths through the two values obtained from two cloud servers. We can give an example, I am a user of amap, and I want to query the road condition information of the road section A. At this time I am the inquirer and send the task to amap's cloud by request, the cloud sends the task to many car owners who pass by the road section A. These car owners are users of amap and they upload the encrypted road condition information to the cloud, the cloud process the true discovery to return the encrypted true value information to the inquirer.

It is worth noting that the two cloud servers have no knowledge of claims during the cloud's work. Since in our design, the inputs from the users and the outputs to the inquirers are all concealed claims, and the weights for the truth discovery algorithm working in clouds are also ciphertext. This mechanism is available for the protection of the sensory data and weights privacy based on the fact that the clouds cannot deduce any privacy information with their known data. Therefore, our design can prevent cloud attacks. At the same time, the data transferred from the user to the cloud are in the form of decomposition, while the decomposing key is updated randomly each time. In this sense, our system can also prevent transmission attacks.

B. Threat Model

The objective of security is to protect the whole process's privacy under the accuracy requirements. In MCS systems, we mainly consider the following three attack models, as shown in Fig. 2, that can break users' privacy and reduce accuracy. We need to keep strong privacy of the users' sensory data, reliability degree, intermediate data, and estimated truths when facing these attacks. The threat models are divided into three kinds: cloud attacks, user attacks, and transmission attacks. The cloud attacks are considered under the honest-but-curious model and there is no collusion between clouds. All clouds strictly execute the protocol, yet they remains curious when trying to independently infer users' private data and will do so independently [10], [11]. Cloud servers can make use of their huge computing power and storage capacity to infer the observation value and weight of each user. Such a security assumption adapts in cloud-based MCS applications since in practice cloud service providers are business drivers with good reputations. They do not want to damage their reputation, to avoid malicious acts and collusion [20]. User attacks mainly come from other users, and each user, for economic benefits, tries to infer observations except for those of their own [23]. In the meantime, one needs to consider the transmission attacks, for which eavesdroppers want to manipulate the sensory data from users to the clouds. Tang et al. [17] protected the claims from users before transferring them to the clouds. And the way they adapt is the users generate fresh random masks to conceal their claims. Zheng et al. [20] use asymmetric encryption to encrypt the transmitted data from users to clouds. Clouds generate an asymmetric key pair of the Paillier cryptosystem, and the public key is published. Each user encrypts the sensing values through the corresponding public key. In addition to these security targets, the truth value received by the inquirer should be within a small margin of error for the accuracy requirement.

We first analyze the cloud attacks. Two honest-but-curious cloud servers are assumed, i.e., P_1 and P_2. Being without collusion, P_1 and P_2 will honestly execute our algorithm, but curiously want to infer the users privacy information to gain benefits. Therefore, our security MCS scheme is of significance under this assumption.

Then from the perspective of user attacks, we have to protect users' sample data and their weights, which should not be known by any party, for example, the case that the users' weights are disclosed to other entities, such as the cloud servers, which can use this information to deduce some private information of users. Moreover, if some eavesdroppers gain the sample data by attacking the users' transfer, the cloud server can cooperate with the eavesdropper to modify the value.

Finally, we consider transmission attacks. One remarkable fact is that user sample data and meanwhile upload the data to the cloud, which is vulnerable to transmission attacks [24]. Traditional PPTD algorithms require constant user interaction with the cloud and need fresh encrypted keys, which is not practical in real mobile applications. The latest no-interactive algorithms demand low communication costs from users to clouds. The main transmission methods are to encrypt the sensory data by random masks or leverage asymmetric encryption to conceal their claims [25]. But if the conceal keys are used multiple times, it can be reconstructed by blind source separation and the cryptosystem can be crack [26]. Only the secret key is available for update in real time, the cryptosystem can prevent transmission attacks efficiently. Also the indistinguishable is an important security attribute, which can protect significant information about the sensing values [27].

We do not believe that users maliciously manipulate their sensory data, thereby damaging the system. The defense against this threat is an orthogonal problem and also requires self-validation by the user, which can be accomplished by techniques of bilinear pairing and zero-knowledge proofs [14], [28]. We also did not particularly consider the impact of different user object choices on privacy, which can be solved by deploying appropriate anonymization technologies [29]. The security focus of our work is to ensure the confidentiality of data to the cloud server.

III. Background

In this section, we summarize the background of truth discovery and GC, respectively.

A. Truth Discovery

In MCS, as an effective information processing technology, truth discovery can be used to extract reliable and truthful information from most sensors. The basic principle of the truth discovery scheme is that users have higher weights if they frequently provide effective information. At the same time, these users' sample data are closer to the truth. While the truth discovery algorithm usually assigns an initial truth value randomly, and then, iteratively updates the weight and truth until the occurrence of convergence. We briefly introduce the functions used in truth discovery from the representative conflict resolution on heterogeneous data (CRH) framework [8].

1) Weight Update: In CRH, the assumption of objective truth is fixed. According to the distance between the sensory data and the object truth, users can obtain their weight information. The basic principle of weight assignment is that in the case that the data provided by the user are close to the estimated values, a higher value should be assigned to the user's weight. In general, the user's weight is calculated as

$$w_k = f\left(\Sigma_{m=1}^{M} d(x_m^k, x_m^*)\right) \tag{1}$$

where f refers to the decreasing function, and $d(x_m^k, x_m^*)$ represents the distance function, k is the kth user, and the total number of objects is M. We adopt the weight calculation function of CRH as f for the excellent performance, as shown in

$$w_k = \log\left(\frac{\Sigma_{k'=1}^{K}\Sigma_{m=1}^{M} d(x_m^{k'}, x_m^*)}{\Sigma_{m=1}^{M} d(x_m^k, x_m^*)}\right). \tag{2}$$

2) Truth Update: The truth update function is another important function for the ability to estimate objective truths. When the weights are given to each user, the objective truth can be calculated according to

$$x_m^* = \frac{\Sigma_{k=1}^{K} x_m^k \cdot w_k}{\Sigma_{k=1}^{K} w_k}. \tag{3}$$

Algorithm 1 describes the general truth discovery procedure, which mainly contains two phases: weight update and truth update. These two phases execute continuously and iteratively until they meet the requirements of certain convergence conditions. As for the generality, we follow Algorithm 1 with the production of weight update and truth update as the truth discovery procedure to instantiate the design of our CEPTD system in this article.

Algorithm 1: Truth Discovery Algorithm.

```
Input: $K$ users, $M$ objects, $x_M^K$,Maximum Iteration $T$
Output: Object truths:$x_M^*$
Initialize the object truths $[x_1^*, x_2^*, ..., x_M^*]$ and send them to
each user
for iteration = 1 to iteration=T do
    for k=1 to k=K do
        Update each user's weight based on Eq. (2) // (I);
    end
    for m=1 to m=M do
        Update the object truth based on Eq. (3) // (II);
    end
end
return $x_M^*$
```

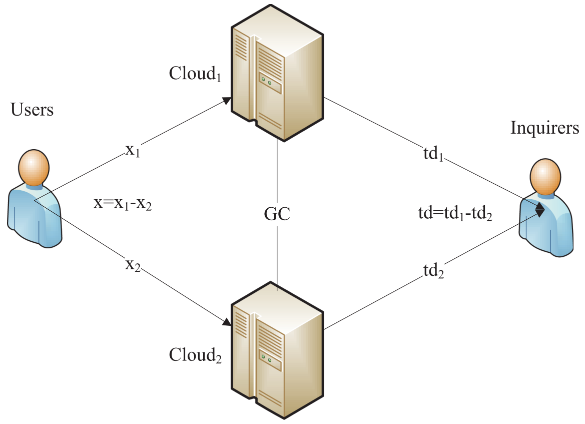

Fig. 3. Core part of our CEPTD system.

B. Garbled Circuit (GC)

Yao first presented the general GCs for the secure two-party computation against the semi-honest model [30]. In a secure two-party computation condition, the two parties collaboratively execute a function, and they both know only the output of the function [31]. Due to the advantage of the two-party computation setting, GC can securely compute arbitrarily complicated functions. Under GC, the involved users do not need to process these intermediate results, however, the iterations are available for sequent and secure process.

Suppose that P_1 and P_2 are the two noncolluding parties, and x_1 and x_2 represent the input from P_1 and P_2, respectively. A polynomial-time function $g(x_1; x_2)$ is used to compute the input gate by gate through the input wires to the output wires. At the same time, the two parties cannot reveal their inputs x_1 and x_2 to each other. To facilitate secure computation, on the one hand, P_1 sends the random value corresponding to its input boolean value to P_2. On the other hand, P_2 engages in a 1-out-of-2 oblivious transfer protocol with P_1 to obtain the random value corresponding to its input boolean value. Therefore, P_1 does not know the inputs of P_2, nor does P_2 know the input of P_1. Once P_2 has both the GC and garbled inputs, it can directly compute the entire circuit. And the outputs will be ultimately shared by P_1 and P_2 [32].

IV. Proposed Scheme

In this section, we propose a CEPTD framework, which is composed of three phases: initialization phase, iteration phase, and truths reconstruction phase. In this part, we first give a detailed introduction to the algorithm core which is high-level system architecture. Then, we provide the cooperation protocol from initialization to iteration and show the truths reconstruction phase in inquirers. When executing the whole program, the real truths can be estimated and securely transmitted to the inquirer.

Before we formally introduce CEPTD, we define some notation. Let M be the whole set of objects from requesters, and K be the set of system users that can sample values. We denote the subset of objects chosen by user $k \in K$ as M_k, where M_k is a subject of M. We denote the subset of users who choose the sensing object m by K_m. We will refer to the sensory data provided to the user as sensory values. User k samples the object m to obtain the sensing value and is defined as x_m^k. The estimated values for object m is represented as x_m^*.

A. Detailed Algorithm Core

The core is a high-level system architecture that contains the constituent entities of the algorithm and algorithm workflow. Fig. 3 depicts our algorithm program in three parts: data encryption, cloud truth discovery, and user data reconstruction. A brief introduction of the proposed CEPTD scheme is shown as follows.

At the core shown in Fig. 3, the design contains three types of entities: inquirers, clouds, and users. The inquirer refers to a unit of the initial crowd and participatory sensing application. It sends the query task to the cloud which returns the data result of the query to the inquirer. An example of a crowdsensing application is indoor floorplan dataset, of which the purpose is to reconstruct the floorplan based on the sample data collected by the mobile terminals containing various sensors. In this set, the object refers to the measurement of the distances between two special locations. Each user only senses a subset of the object rather than all objects. The cloud server assigns tasks to various users who are responsible for sampling data. While our cloud servers consist of P_1 and P_2, which are two noncolluding servers and equally important. They execute the truth discovery algorithm, respectively. At last, P_1 and P_2 send estimated truths to the inquirer for decryption and reconstruction. It is practically important to know that we only need one-time interaction between users and clouds during the whole process. Therefore, the noninteractive of our CEPTD is realized.

We note the core program from Fig. 3. First, user k samples the object m to gain sensory data x_m^k, and then generates a fresh random number $x_{m,1}^k$. After that, the user k calculates $x_{m,2}^k$ with the equation of $x_m^k = x_{m,1}^k - x_{m,2}^k$. At last, the user sends $x_{m,1}^k$ to the cloud server P_1 and $x_{m,2}^k$ to the cloud server P_2. From this encrypted program, it can be seen that one single cloud server has no idea about the original sample data of x_m^k. Besides, the whole encrypted program introduces only a

Algorithm 2: CEPTD Algorithm.

```
Input: Users set $K$, Objects set $M$, Cloud Servers $P_1$ and
       $P_2$, Inquirers $Q$, iterations $T$, Sensory data:$x_m^k, k =
       1, 2, ..., K, m = 1, 2, ..., M$
Output: Estimated truths $\text{Truth}_m$
Initialization Phase: users initialize variables $\text{ID}_i$, $S_1$, $S_2$;
for m=1 to m=M do
    for k=1 to k=K do
        Generate a fresh random number $x_{m,1}^k$
        Decompose $x_m^k$ into $x_{m,1}^k$ and $x_{m,2}^k$ through $x_m^k =
        x_{m,1}^k - x_{m,2}^k$
        Send ($x_{m,1}^k$, $\text{ID}_{k,m}$), $S_1(k)$ to $P_1$ and ($x_{m,2}^k$, $\text{ID}_{k,m}$),
        $S_2(k)$ to $P_2$
    end
end
Iteration Phase: initialize truth values $(td_1)_m^*$ and $(td_2)_m^*$,
for iteration=1 to iteration=T do
    for m=1 to m=M do
        for k=1 to k=K do
            Perform Eq. (4)−(7)
        end
    end
end
Reconstruction Phase:
for q=1 to q=Q do
    Query the cloud, gain $(td_1)_m^*$ from $P_1$ and $(td_2)_m^*$ from
    $P_2$,
    Perform equation $td_m = (td_1)_m^* - (td_2)_m^*$.
end
return Truth_m=$td_m$
```

little overhead to the users, which conforms to the practical characteristics of the mobile crowdsensing networks. Moreover, this encrypted process adopts patterns of data decomposition, which can deal with massive transmission attacks. The objective of this article is to ensure the confidentiality of user-perceived data and the reliability of the information in the MCS scenario and to estimate the authenticity of the target accurately. Therefore, we implement the specialized Algorithm 2 of truth discovery for crowd sensed data streams with the security cooperate protocol, which prevents the users' privacy data from leakage.

B. Initialization Phase

To enable practical CEPTD, at a high level, users first send the decomposed signal to two noncolluding cloud servers. Then, the two servers perform the truth discovery algorithm, respectively. The algorithm exceeds the estimated truth from each cloud. Finally, an inquirer searches two clouds, and collaboratively works to reconstruct the final real truths. In the meantime, it should be noted that each claim from the same user has a different decomposition key to protect privacy.

In the initialization phase, each user k uses its sensors to collect sensory data for object m which is set as $x_{m,1}^k$. The user k also need to generate a random vector $\mathbf{r_k} = r(k,m)_{m=1}^M$, where the random range is $\left[\frac{x_{m,1}^k}{2} - \theta_m, \frac{x_{m,1}^k}{2} + \theta_m\right]$. We set θ_m based on the type of object m and each value $r(k,m)$ conceals the mth object in the kth user's data. This initialization ensures that the decomposition key is used in a one-time scheme where each secret key is used only once, and different keys are statistically independent. After that, the user performs the equation $x_m^k = x_{m,1}^k - x_{m,2}^k$ to get $x_{m,2}^k$. The user transmits $x_{m,1}^k$ and $x_{m,2}^k$ to the cloud servers of P_1 and P_2, respectively [17], [33]. Another note is that the two transferred data have range of similarity based on θ_m, then secure indistinguishable obfuscation has been realized. When the initialization is finished, the whole sensory data encryption in users side is completed as well. It should be noticed that we only transport two unrelated data instead of the original sensory data. As long as there is no collusion between the two cloud servers, this irrelevance makes the eavesdropper unable to crack this encryption method.

When the users submit the decomposed data to the two cloud servers, we should verify whether the user successfully submits it or not. The user k generates a random unique identifier ID_k. Later, it sends $(x_{m,1}^k, \text{ID}_k)$ to P_1, and $(x_{m,2}^k, \text{ID}_k)$ to P_2, respectively. The work on the user side is completed until the transmission finishes. Compared with the original truth discovery algorithm, we only add a small number of operations that are in line with the actual characteristics of a mobile crowd perception network with limited mobile terminal resources. At the same time, we believe that the capabilities and resources of the cloud are infinite, regardless of the added overhead of the cloud. The two cloud servers designate T iterations until the occurrence of results convergence. After that, the estimated truths are sent to the inquirers.

To maintain the homomorphism between the user and the inquirer, we need two clouds to keep the weights in synchronization when performing the truth discovery algorithm. However, the weight information also presents a kind of private information of the user, which cannot be disclosed to the cloud. Therefore, we want to encrypt the weight information and ensure that the update weight for calculation from two clouds keep synchronization. The secret key encrypted to the cloud can be generated on the user side. Therefore, in the initialization stage, we generate other two random values of s_1^k and s_2^k for each user. Then, s_1^k and s_2^k are available for perturbation encryption for the output weight of the GC. Besides, the security of the weight is guaranteed since the perturbation information is random, and cannot be known by cloud servers.

It should be noticed that, in this approach, if the data to be concealed are not in integer form, we multiply a parameter L (a magnitude of 10) to round it. Therefore, we can recover the approximate value by dividing L.

C. Iteration Phase

In this part, the cloud servers execute the truth discovery procedure for convergence and security weight synchronization to realize PPTD. Here, the total number of successful users and the summation of the data collected from users are under privacy protection for these information are also the users' privacy, which cannot be leaked. The cloud P_1 and P_2 receive the $x_{m,1}$ and $x_{m,2}$, respectively. When the cloud servers receive the data,

P_1 and P_2 cooperation leverage GC calculates the estimated truths in each iteration.

The detailed GC design is introduced as follows: P_1 assembles a garbled circuit GC with the functionality of truth discovery algorithm. It then garbles the conceal vectors $(x_{m,1}^k, \mathrm{ID}_k)_M^K$, random values s_1^k, and sends garbled circuit GC to cloud P_2. P_2 first obtains garbled claims $(x_{m,2}^k, \mathrm{ID}_k)_M^K$ and a garbled random values $(s_1^k,\, td_{m,1}^*)$ through oblivious transfer with P_1. Next, P_1 and P_2 exchange with each others the ID_k they received. When ID_k appears both in P_1 and P_2, P_2 is designed to perform the following functions inside the circuit.

Given the decomposed truths of $x_{m,1}^k$ and $x_{m,2}^k$, $k \in [1, K]$, random values s_1^k, s_2^k, the initial truth vectors $(td_1)_0^*$, $(td_2)_0^*$, and the GC from P_1 to P_2 are designed to perform the following equations. With GC, two servers, P_1 and P_2 can get the result of a function without knowing each other's input. Because of this, all iterations can be carried out sequentially and safely without the need for providers to handle any intermediate results. The GC that P_2 receives is designed to perform the following functions inside the circuit.

1) Perform

$$w_k = \log\left(\frac{\Sigma_{k'=1}^K \Sigma_{m=1}^M d(x_{m,1}^{k'} - x_{m,2}^{k'}, x_{1,m}^* - x_{2,m}^*)}{\Sigma_{m=1}^M d(x_{m,1}^k - x_{m,2}^k, x_{m,1}^* - x_{m,2}^*)}\right). \tag{4}$$

2) Perturb the weights data using

$$w_k' = (s_1^k + s_2^k) \times w_k. \tag{5}$$

When the weight is estimated, P_1 and P_2 calculate the estimated truths from

$$td_{m,i}^* = \frac{\Sigma_{k=1}^K x_{m,i}^k \cdot w_k'}{\Sigma_{k=1}^K w_k'}, i \in [1, 2]. \tag{6}$$

The two cloud servers repeat the truth estimation until $t_{m,1}^*$ and $t_{m,2}^*$ convergence, and this process iterates for T times.

In the procedure of CEPTD, users only need to interact with the two cloud servers once in the initialization phase. At the same time, all calculations on the user's terminal are based on plaintexts. Therefore, each user would introduce very little overhead. In addition, the costs can be confirmed by the performance evaluation specified in Section VI-C.

D. Truths Reconstruction Phase

Since inquirers are usually mobile devices that are limited in terms of computation and storage, reconstruction tasks should not perform too many operations. In our scheme, when performing the collaborative protocol, we can recover the truths with a simple computation process, which satisfies the requirements of the system character. Upon the receipt of the two components of $td_{m,1}^*$ and $td_{m,2}^*$ by the inquirer, these components are used to reconstruct the final truth. When performing the collaborative truth discovery protocol, P_1 outputs the truth discovery result of $td_{m,1}^*$, and P_2 outputs the truth discovery result of $td_{m,2}^*$. After that, both results of the algorithm are sent to the inquirer. In the whole scheme, the two noncolluding servers adopt different input observation values, however, their weight information keeps being synchronized. The inquirer can recover the final truth by simply compute the equation of $td_m^* = td_{m,1}^* - td_{m,2}^*$. Detailed proof will be given in Section V-A.

V. Theoretical Analysis of Advantages

In this section, we analyze the correctness, security, and complexity of the CEPTD algorithm. Correctness ensures the reconstruction accuracy of sensory data. Strong security proves the privacy protection performance of the CEPTD algorithm. And lower complexity reflects the efficiency of our design.

A. Correctness Analysis

In this part, a theoretical analysis of the correctness of our scheme is presented. In our scheme, to encrypt the acquired original sample signal, x is divided into two parts by the equation of $x = x_1 - x_2$. Intuitively, for a truth discovery algorithm, we do not have $td = td_1 - td_2$. However, we can ensure that the final result satisfies the requirements of the aforementioned equation through collaborative design. The detailed procedure of the iteration process is referred to [32]

$$\begin{aligned}(td')_m^* &= \frac{\Sigma_{k=1}^K w_k' \cdot x_m^k}{\Sigma_{k=1}^K w_k'} = \frac{\Sigma_{k=1}^K (s_1 + s_2) \times w_k \cdot x_m^k}{\Sigma_{k=1}^K (s_1 + s_2) \times w_k} \\ &= \frac{\Sigma_{k=1}^K w_k \cdot x_m^k}{\Sigma_{k=1}^K w_k} = td_m^*.\end{aligned} \tag{7}$$

It can be seen from (7) that the weight information is encrypted to prevent cloud leakage. The result of the truth value is not changed when performing the truth value update

$$x_m = x_{m,1} - x_{m,2} \tag{8}$$

$$td_{m,1}^* = (td')_{m,1}^* = \frac{\Sigma_{k=1}^K w_k' \cdot (x')_{m,1}^k}{\Sigma_{k=1}^K w_k'} \tag{9}$$

$$td_{m,2}^* = (td')_{m,2}^* = \frac{\Sigma_{k=1}^K w_k' \cdot x_{m,2}^k}{\Sigma_{k=1}^K w_k'} \tag{10}$$

$$\begin{aligned}td_{m,1}^* - td_{m,2}^* &= \frac{\Sigma_{k=1}^K w_k \cdot x_{m,1}^k}{\Sigma_{k=1}^K w_k} - \frac{\Sigma_{k=1}^K w_k \cdot x_{m,2}^k}{\Sigma_{k=1}^K w_k} \\ &= \frac{\Sigma_{k=1}^K w_k \cdot x_m^k}{\Sigma_{k=1}^K w_k} = td_m^*.\end{aligned} \tag{11}$$

It is proved by (8)–(11) that homomorphism is maintained by the truth discovery algorithm. It means that $td = td_1 - td_2$ still be hold from (11).

B. Security Analysis

In this subsection, it is introduced that our CEPTD algorithm is secure under honest-but-curious model. The two noncolluding cloud servers and users cannot infer any privacy information, including privacy information of users, weights data, intermediate results, and the final estimated truths. Intermediate results include intermediate weights and intermediate truths, which may be exposed during each iteration. If these intermediate results are in plaintext form, they could leak some privacy information about users to a certain degree. As previously discussed, the security threats mainly consist of three aspects: cloud attacks, user-to-user attacks, and transmission attacks. The objective to deal with these attacks is to protect users' information and inquires' requirements. Therefore, the whole dataset containing users' information and the final estimated truths need to be protected. The security of the scheme is analyzed from the perspective of the three types of attacks.

Theorem 5.1: (Against Cloud Attacks) Our basic architecture and transmission mode ensures that the two noncolluding cloud servers are not sensitive to the sensing values, weights of users, intermediate results, and truths of the inquirers.

Proof: In this theorem, the two clouds servers are assumed to have no collusion with each other, which, however, are curious and try to infer the users' private information independently. The assumption is practical in a real MCS system.

P_1 and P_2 receive decomposed sensing values, which cannot be decrypted only when two servers collude. During the encrypted CEPTD procedure, P_1 has some interactions with P_2 when executing weight updates. As shown at the iteration phase, the weight information is updated in the form of encryption. And the secret key cannot be known to the two servers at the same time. Since the encrypted weight key S are randomly generated and statistically independent, they do not contain any users' private information. Combining with their one-time update feature, they hold strong privacy. Therefore, neither P_1 nor P_2 is able to infer anything about the users' private information unless they collude with each other. All the transmission values, including the intermediate data, are well kept confidential to both cloud servers. Even in the last iteration, the truth is also stored in a decomposed form in each cloud, and only the inquirer can recover the truth. □

Theorem 5.2: (Against User Attacks: Under noncolluding Setting) Suppose K users sample M objects to two clouds P_1 and P_2. After executing the collaborate protocol, each user's observation value cannot be disclosed to others.

Proof: In this theorem, we only consider the attacks from users but the cloud servers are honest. Some users may collude with each other to illegally derive other users' private information for economic benefit, which is a common assumption in real MCS systems. For simplicity, we indicate x_k as the input of the user k. As every user only holds its own data and secret keys, the cooperative users can infer the information of the black users. And these cooperative users can work together with the cloud to get the summation aggregate of the noncolluding users in each cloud. However, they also cannot get any individual user's input from this summation since this summation is also irrelevant to the original summation. □

Theorem 5.3: (Against Transmission Attacks) Users transport values to cloud in an encrypted form, but the encrypted keys are not transported. The transmission attacks cannot break the cryptosystem.

Proof: In this theorem, the main attacks that are taken into consideration are the transmission attacks. These attacks break the system by collecting some plaintext and ciphertext. To cope with transmission attacks, the original cryptosystem uses multiple secret keys to encrypt sensory data. However, it is not suitable for the limit of MCS systems. Our cryptosystem employs two split parties which are then transmitted to the two cloud servers, respectively. After that, the split part also refers to the encrypted phase, and the split key is the secret key. As mentioned in the initialization phase, the split key is a one-time value and statistically independent. Therefore, the eavesdroppers cannot recover the secret key for the frequent update. Although eavesdroppers can collect a part of ciphertext, the ciphertext is not related to the original data. Therefore, the eavesdroppers cannot reconstruct any users' private information. When eavesdroppers collect part plaintext, the idea of the proof is similar to Theorem 5.1. □

C. Complexity Analysis

The performance of the CEPTD system is analyzed when it runs for a while. In the following calculations, it is assumed that there are two cloud servers and K users. We consider the capabilities and resources of the clouds are infinite, regardless of the added overhead of the clouds. So, our main concern is complexity on the users' side. Communication costs are considered resource-expensive processes. We first analyze the communication costs of users. In the initialization phase, each user needs to produce random values and decompose the sample data. Then, these values are transported to the two cloud servers, respectively. The communication cost can produce only $O(1)$ for each user k. For the total K users, the communication complexity of the user side is $O(K)$. Considering the cloud communication costs in each iteration, the two clouds interact with each other and transport $O(K)$ encrypted weights. When Algorithm 1 produces T times in our cryptosystem, M truths will be transported to the inquirers. Therefore, the whole communication cost in the whole cryptosystem is $O(TK + M)$. It can be seen that the communication complexity depends highly on the total sensory data collected from all the users. It can put up a good performance in real MCS applications.

VI. Performance Evaluation

This part shows extensive performance evaluations of our CEPTD scheme. First, we analyze the accuracy of our system. Then, different truth discovery algorithms are embedded in our scheme to indicate that our algorithm can be applied in different scenarios containing different types of datasets. After that, time cost, energy cost, and communication cost are evaluated. Finally, we simulate the convergence rate of our algorithm. During the implementation of CEPTD, we simulate our experiment with two servers in Microsoft Azure, each with 2 Intel Xeon E5-2667 3.2-GHz cores and 256-GB RAM. Besides, the operating system is Windows Server 2012 R2 standard.

In this experiment, we use both the real dataset and the simulation dataset as experimental sets to record the performance of the algorithm and test the robustness of the system. In particular, we use continuous datasets in our experiments, since categorical datasets also have similar results. In crowdsensing applications with categorical data, there are usually multiple candidate choices, and only one is correct. In this case, the sensory data x_c^i collected on an object by the user i can be represented as a vector, i.e., $x_c^i = [0, \ldots, 1, \ldots 0]^T$, which means that the lth choice is

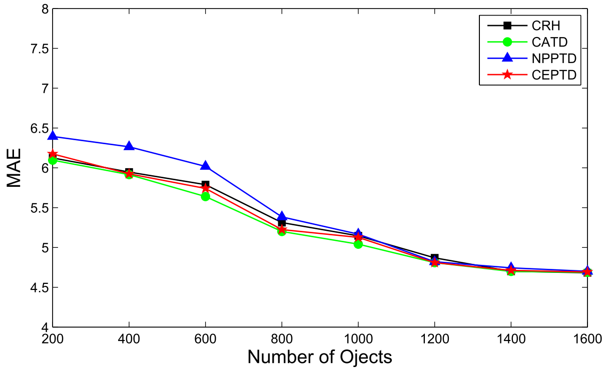

Fig. 4. MAE with weather dataset for different PPTD algorithms.

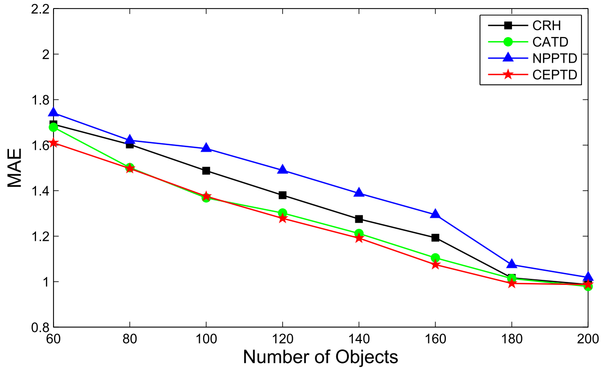

Fig. 5. MAE with indoor floorplan dataset for different PPTD algorithms.

selected by the user i. The distance between two categorical data is defined as: $d(x_c^i, x_c^j) = (x_c^i - x_c^j)^T(x_c^i - x_c^j)$. The estimated ground truth x_c is a vector of probability values, and truth discovery can be performed on each component of the sensory data vector for the item.

A. Accuracy

Now, the accuracy of the final truth of CEPTD is evaluated. The main target for the accuracy experiment is to show that our CEPTD algorithm does not compromise the accuracy compared with traditional truth discovery algorithms. In our experiment, the accuracy of the CEPTD algorithm includes the mean absolute error (MAE) and root mean square error (RMSE). Two real-world datasets are used to demonstrate the effectiveness of the CEPTD algorithm, one is the weather forecast dataset, in which the weather forecasting data are collected from three platforms includes nine sources [8]. About 16 000 temperature data are selected, and there are 1700 ground truths. Ground truth values are used only for accuracy evaluation, and will not participate in truth discovery. Other datasets also follow this setting [8]. The other is the floorplan dataset. The sensory data in the dataset refer to the sensing values of the distance information for any two specified indoor points. It contains 2400 sensing values from 247 users on 260 objects. The datasets for the rest of the experiment also come from these shown in [34].

Next, it is proved that our CEPTD framework will not lower the accuracy. Fig. 4 shows the MAE results of the weather dataset from different objects for different PPTD algorithms. And Fig. 5 shows the MAE results of the indoor floorplan dataset from different objects for different plaintext truth discovery algorithms. As shown in these figures, we compare our CEPTD approach with CRH, CEPTD, and CATD. It is found that our proposed CEPTD approach achieves almost the same estimation accuracy as CRH and other classic plaintext truth discovery algorithms like CATD.

B. Use Different Truth Discovery Algorithms

We now evaluate the accuracy of the CEPTD design using different truth discovery algorithms. Figs. 6–9 show the MAE and RMSE results after T iterations, respectively. The results prove the substitutability of the CEPTD framework. Our scheme under

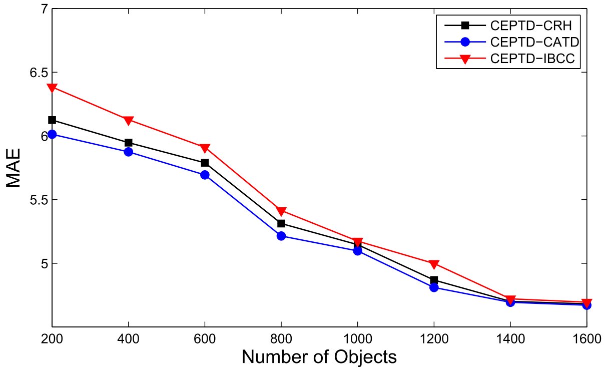

Fig. 6. MAE with weather dataset for three truth discovery algorithms.

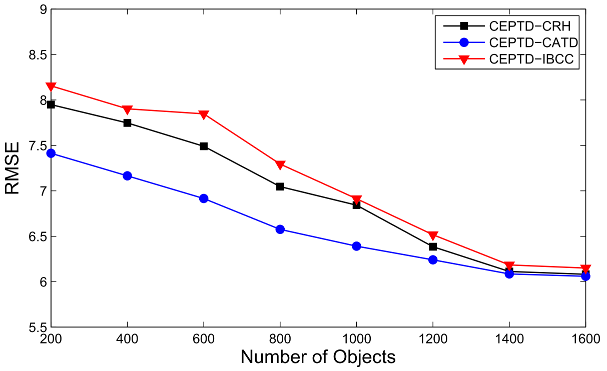

Fig. 7. RMSE with weather dataset for three truth discovery algorithms.

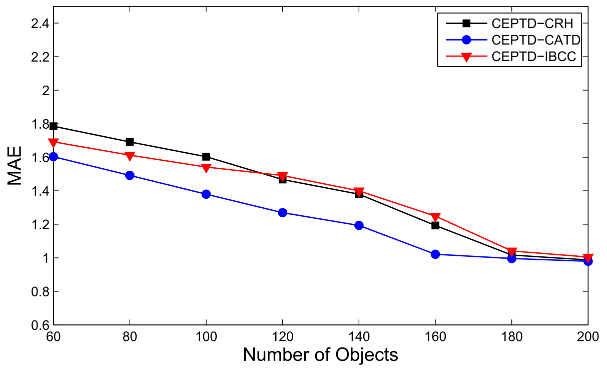

Fig. 8. MAE with indoor floorplan dataset for three truth discovery algorithms.

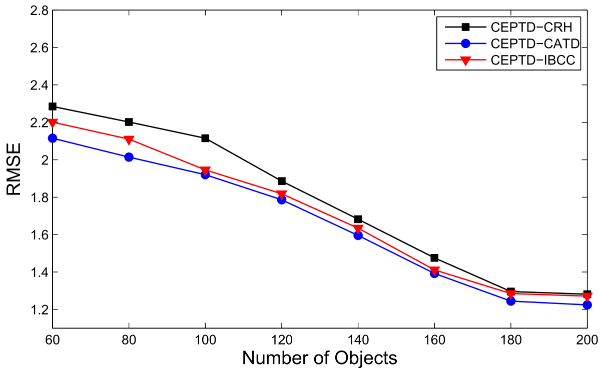

Fig. 9. RMSE with Indoor floorplan dataset for three truth discovery algorithms.

all kinds of truth discovery algorithms shows good performance, and it is important to apply the scheme to different scenarios and different types of data. Some scenarios restrict certain truth discovery algorithms, and sometimes the data types can also restrict truth discovery algorithms. Therefore, universality and practicability are guaranteed by our scheme. Besides, any truth discovery algorithm can also be embedded.

In Figs. 6 and 7, we show the performance of several classical algorithms in terms of MAE and RMSE on continuous data for a real-world dataset called weather datasets. Figs. 8 and 9 show the accuracy of these encrypted truth discovery algorithms in the indoor floorplan datasets. The result of the CEPTD approach achieves the same accuracy compared with normal truth discovery algorithms. These four figures show three truth discovery algorithms, including CRH, CATD, and independent Bayesian classifier combination [35]. The CEPTD-CATD shows better accuracy compared with the other two algorithms. This is because the accuracy of our scheme depends largely on the original truth discovery algorithm, and the original CATD algorithm performs better. However, it should be noted that CRH is very widely used. In order not to lose generality, we use CRH as Algorithm 1 for truth discovery.

C. Communication Overhead

1) Time Cost: In this part, the running time of our CEPTD scheme via M objects and K users are demonstrated. Fig. 10 shows the running time results with a large scale number of users when the observed objects vary from 200 to 1700. Compared with the original algorithm, our method increases the decomposition part of sensory data, and the truth discovery algorithm is executed simultaneously in two clouds. However, our scheme adds less to the encryption overhead. It can be seen from Fig. 10 that our CEPTD approach only increases 4 s, even if the number of objects increases sharply to 1700.

2) Energy Cost: As mentioned previously, the user side and the inquirers are always mobile devices, therefore, battery energy presents a precious resource. It should be noted that in the CEPTD scheme, each user only introduces cost when decomposing the sensing values sampled by it. Besides, each user generates four random values including $(x_{m,1}^{k}, \mathrm{ID}_k, s_1, s_2)$ and decomposes the sensory data into two irrelevant data. Therefore, the amount of energy produced includes the generation energy and the energy of decomposition. We measure the total energy on users' terminals based on our designated computation with Power Tutor 2 Pro. It can be observed from Fig. 11 that the total energy consumed is in direct proportion to the number of objects perceived by users. Particularly, when the number of objects varies from 20 to 80, the total energy cost grows from 37 to 182 mJ, which is acceptable to mobile users with limited resources [36].

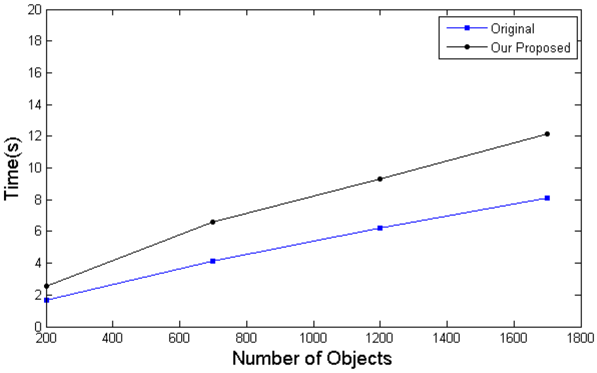

Fig. 10. Time cost.

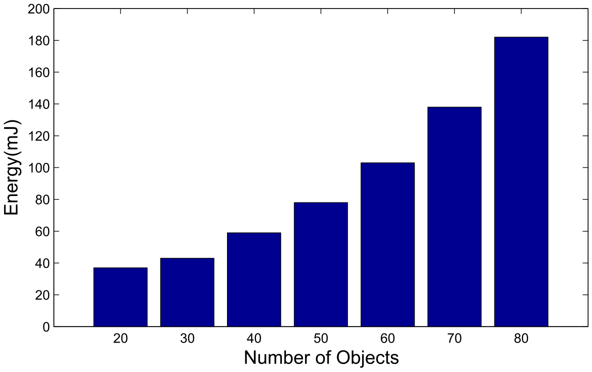

Fig. 11. Communication cost.

VII. Conclusion

In this article, a CEPTD approach is proposed in MCS systems, which realizes noninteractive truth discovery with strong privacy and accuracy guarantees. Our approach adopts a one-time decomposition scheme when running in the cloud. therefore, we do not need to transfer any data related to the original information and security is guaranteed. Moreover, we make user interaction with the cloud into two clouds by GC communication circuit, thus realizing the practical truth discovery algorithm without interaction. Extensive experiments conducted on real-world datasets and simulation datasets also demonstrate that the proposed CEPTD approach shows ideal performance compared with the existing models.

Acknowledgment

The authors would like to thank Prof. J. Yu for his help in this work, especially the literature expansion and valuable guidance.

References

[1] S. Zhou, Y. Lian, D. Liu, H. Jiang, Y. Liu, and K. Li, "Compressive sensing based distributed data storage for mobile crowdsensing," *ACM Trans. Sensor Netw.*, vol. 18, 2022, Art. no. 25.

[2] Q. Liu, Y. Peng, J. Wu, T. Wang, and G. Wang, "Secure multi-keyword fuzzy searches with enhanced service quality in cloud computing," *IEEE Trans. Netw. Service Manage.*, vol. 18, no. 2, pp. 2046–2062, Jun. 2021.

[3] S. Zhou, Y. He, S. Xiang, K. Li, and Y. Liu, "Region-based compressive networked storage with lazy encoding," *IEEE Trans. Parallel Distrib. Syst.*, vol. 30, no. 6, pp. 1390–1402, Jun. 2019.

[4] T. G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration," *IEEE Trans. Comput.*, vol. 67, no. 9, pp. 1287–1300, Sep. 2018.

[5] P. Sun et al., "Towards personalized privacy-preserving incentive for truth discovery in mobile crowdsensing systems," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 352–365, Jan. 2022.

[6] R. Liu and J. Pan, "Lightweight privacy-preserving truth discovery for vehicular air quality monitoring," *Digital Commun. Netw.*, vol. 9, pp. 280–291, 2022.

[7] J. Chen, Y. Liu, Y. Xiang, and K. Sood, "RPPTD: Robust privacy-preserving truth discovery scheme," *IEEE Syst. J.*, vol. 16, no. 3, pp. 4525–4531, Sep. 2022.

[8] 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 *Proc. ACM Sigmod Int. Conf. Manage. Data*, 2014, pp. 1187–1198.

[9] C. Zhang, L. Zhu, C. Xu, K. Sharif, and X. Liu, "Pptds: A privacy-preserving truth discovery scheme in crowd sensing systems," *Inf. Sci.*, vol. 484, pp. 183–196, 2019.

[10] G. Xu, H. Li, D. Liu, R. Hao, Y. Dai, and X. Liang, "Towards efficient privacy-preserving truth discovery in crowd sensing systems," in *Proc. IEEE Glob. Commun. Conf.*, 2017, pp. 1–6.

[11] Y. Zheng, H. Duan, X. Yuan, and W. Cong, "Privacy-aware and efficient mobile crowdsensing with truth discovery," *IEEE Trans. Dependable Secure Comput.*, vol. 17, no. 1, pp. 121–133, Jan./Feb. 2017.

[12] C. Zhang, L. Zhu, C. Xu, K. Sharif, X. Du, and M. Guizani, "LPTD: Achieving lightweight and privacy-preserving truth discovery in CIoT," *Future Gener. Comput. Syst.*, vol. 90, pp. 175–184, 2019.

[13] C. Miao, L. Su, W. Jiang, Y. Li, and M. Tian, "A lightweight privacy-preserving truth discovery framework for mobile crowd sensing systems," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.

[14] C. Zhang, L. Zhu, C. Xu, X. Liu, and K. Sharif, "Reliable and privacy-preserving truth discovery for mobile crowdsensing systems," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 3, pp. 1245–1260, May/Jun. 2021.

[15] C. Lv, T. Wang, C. Wang, F. Chen, and C. Zhao, "ESPPTD: An efficient slicing-based privacy-preserving truth discovery in mobile crowd sensing," *Knowl.-Based Syst.*, vol. 229, 2021, Art. no. 107349.

[16] X. Pang, Z. Wang, D. Liu, J. C. Lui, Q. Wang, and J. Ren, "Towards personalized privacy-preserving truth discovery over crowdsourced data streams," *IEEE/ACM Trans. Netw.*, vol. 30, no. 1, pp. 327–340, Feb. 2022.

[17] X. Tang, C. Wang, X. Yuan, and Q. Wang, "Non-interactive privacy-preserving truth discovery in crowd sensing applications," in *Proc. IEEE Conf. Comput. Commun.*, 2018, pp. 1988–1996.

[18] F. Liu, B. Zhu, S. Yuan, J. Li, and K. Xue, "Privacy-preserving truth discovery for sparse data in mobile crowdsensing systems," in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.

[19] Z. Wang, X. Cheng, S. Su, and L. Wang, "Achieving private and fair truth discovery in crowdsourcing systems," *Secur. Commun. Netw.*, vol. 2022, 2022, Art. no. 9281729.

[20] Y. Zheng, H. Duan, and W. Cong, "Learning the truth privately and confidently: Encrypted confidence-aware truth discovery in mobile crowdsensing," *IEEE Trans. Inf. Forensics Secur.*, vol. 13, no. 10, pp. 2475–2489, Oct. 2018.

[21] Y. Liu et al., "RPTD: Reliability-enhanced privacy-preserving truth discovery for mobile crowdsensing," *J. Netw. Comput. Appl.*, vol. 207, pp. 138–150, 2022.

[22] S. Yuan, B. Zhu, F. Liu, J. Li, and K. Xue, "A fog-aided privacy-preserving truth discovery framework over crowdsensed data streams," in *Proc. IEEE Glob. Commun. Conf.*, 2021, pp. 1–6.

[23] W. Jiang et al., "Towards quality aware information integration in distributed sensing systems," *IEEE Trans. Parallel Distrib. Syst.*, vol. 29, no. 1, pp. 198–211, Jan. 2018.

[24] X. Shen, L. Zhu, C. Xu, K. Sharif, and R. Lu, "A privacy-preserving data aggregation scheme for dynamic groups in fog computing," *Inf. Sci.*, vol. 514, pp. 118–130, 2020.

[25] Z. Li, Z. Zheng, S. Guo, B. Guo, F. Xiao, and K. Ren, "Disguised as privacy: Data poisoning attacks against differentially private crowdsensing systems," *IEEE Trans. Mobile Comput.*, to be published, doi: 10.1109/TMC.2022.3173642 .

[26] T. Bianchi, V. Bioglio, and E. Magli, "Analysis of one-time random projections for privacy preserving compressed sensing," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 2, pp. 313–327, Feb. 2016.

[27] Z. Jing et al., "Security analysis of indistinguishable obfuscation for internet of medical things applications," *Comput. Commun.*, vol. 161, pp. 202–211, 2020.

[28] M. Jawurek, F. Kerschbaum, and C. Orlandi, "Zero-knowledge using garbled circuits: How to prove non-algebraic statements efficiently," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 955–966.

[29] J. An, Z. Wang, X. He, X. Gui, J. Cheng, and R. Gui, "PPQC: A blockchain-based privacy-preserving quality control mechanism in crowdsensing applications," *IEEE/ACM Trans. Netw.*, vol. 30, no. 3, pp. 1352–1367, Jun. 2022.

[30] V. Kolesnikov and T. Schneider, "Improved garbled circuit: Free XOR gates and applications," in *Proc. Int. Colloq. Automata, Lang. Program.*, 2008, pp. 486–498.

[31] Y. Huang, D. Evans, J. Katz, and L. Malka, "Faster secure two-party computation using garbled circuits," in *Proc. Usenix Conf. Secur.*, 2011, pp. 331–335.

[32] K. Huang, M. Xu, S. Fu, and D. Wang, "Practical privacy-preserving compressed sensing image recovery in the cloud," *Sci. China Inf. Sci.*, vol. 60, no. 9, 2017, Art. no. 098103.

[33] V. Nikolaenko, S. Ioannidis, U. Weinsberg, M. Joye, N. Taft, and D. Boneh, "Privacy-preserving matrix factorization," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2013, pp. 801–812.

[34] R. Gao et al., "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Proc. Int. Conf. Mobile Comput. Netw.*, 2014, pp. 249–260.

[35] E. Simpson, S. Roberts, I. Psorakis, and A. Smith, *Dynamic Bayesian Combination of Multiple Imperfect Classifiers, in Intelligent Systems Reference Library Series: Decision Making and Imperfection*. Berlin, Germany: Springer, 2013, pp. 1–35.

[36] M. Dong and L. Zhong, "Self-constructive high-rate system energy modeling for battery-powered mobile systems," in *Proc. 9th Int. Conf. Mobile Syst., Appl., Serv.*, 2011, pp. 335–348.

Xingting Liu received the B.S. degree in information and computer science from the College of Mathematics and Computer Science, Hunan Normal University, Changsha, China, in 2011, and the M.S. degree in computer application technology from Hunan Normal University, Changsha, China, in 2014. He is currently working toward the Ph.D. degree in software engineering with the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China.

His research interests include compressive sensing, mobile crowdsensing, and privacy preserving.

Siwang Zhou received the M.S. degree in computer software and theory from Xiangtan University, Xiangtan, China, in 2004 and the Ph.D. degree in computer applied technology from Hunan University, Changsha, China, in 2007.

He is currently a Professor with the College of Computer Science and Electronic Engineering, Hunan University. His research interests include compressive sensing, information security, and Internet of Things.

Wei Zhang was born in Hunan, China. He received the M.S. and Ph.D. degrees in software engineering from the College of Computer Science and Electronic Engineering, Hunan University, Changsha, China, in 2013 and 2020, respectively.

He is currently working with Changsha University, Changsha. His research interests include wireless networks, signal processing, and machine learning.

Ting Dong received the M.S. degree in computer applied technology from Hunan University, Changsha, China, in 2006.

She is currently an Associate Professor with Hunan Vocational College of Science and Technology, Changsha. Her research interests include image processing and Internet of Things.

Keqin Li (Fellow, IEEE) received the B.S. degree in computer science from Tsinghua University, Beijing, China, in 1985 and the Ph.D. degree in computer science from the University of Houston, Houston, USA, in 1990.

He is a SUNY Distinguished Professor of computer science with the State University of New York, New Paltz, NY, USA. He is also a Distinguished Professor with Hunan University, Changsha, China. He has authored and coauthored more than 760 journal articles, book chapters, and refereed conference papers. His current research interests include cloud computing, fog computing and mobile edge computing, energy-efficient computing and communication, embedded systems and cyber-physical systems, heterogeneous computing systems, Big Data computing, high-performance computing, CPU–GPU hybrid and cooperative computing, computer architectures and systems, computer networking, machine learning, and intelligent and soft computing.

Prof. Li was the recipient of several best paper awards. He has served on the editorial boards of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS, the IEEE TRANSACTIONS ON COMPUTERS, the IEEE TRANSACTIONS ON CLOUD COMPUTING, the IEEE TRANSACTIONS ON SERVICES COMPUTING, and the IEEE TRANSACTIONS ON SUSTAINABLE COMPUTING.