Efficient and Privacy-Preserving Truth Discovery in Mobile Crowd Sensing Systems

Guowen Xu, *Student Member, IEEE*, Hongwei Li, *Senior Member, IEEE*, Sen Liu, *Student Member, IEEE*, Mi Wen, *Member, IEEE*, and Rongxing Lu, *Senior Member, IEEE*

***Abstract*—With the advancement of mobile crowd sensing systems and vehicular ad hoc networks, the human-carried mobile devices (e.g., smartphones, smart navigators, and smart tablets) equipped with a variety of sensors (such as GPS, accelerometer, and compass) can work together to collect sensory data consequently delivered to the cloud for processing purposes, which supports a wide range of promising applications such as traffic monitoring, path planning, and real-time navigation. To ensure the authenticity and privacy of data, privacy-preserving truth discovery has attracted much attention since it can find reliable information among uneven quality of data collected from mobile users, while protecting both the confidentiality of users' raw sensory data and reliability. However, these methods always incur tremendous overhead and require all participants to keep online for interacting frequently with the cloud server. In this paper, we design an efficient and privacy-preserving truth discovery (EPTD) approach in mobile crowd sensing systems, which can tolerate users offline at any stage, while guaranteeing practical efficiency and accuracy under working process. More notably, our EPTD is the first solution to resolve the problem that users must be online all times during the truth discovery under a single cloud server setting. Moreover, we design a double-masking protocol to ensure the strong security of users' privacy even if the cloud server colludes with multiple users. Extensive experiments conducted on real-world mobile crowd sensing systems also demonstrate the high performance of our proposed scheme compared with existing models.**

***Index Terms*—Truth discovery, privacy protection, cloud computing, crowd sensing, vehicular ad hoc networks.**

I. Introduction

Ubiquitous sensors in mobile devices (example including smart phone, bracelet and tablets) has made more mature for mobile crowd sensing (MCS) systems [1]–[3], where the cloud server pays to a crowd of users carrying mobile devices for outsourcing sensing tasks, and collects their sensory data for specific requirements. Particularly, with the deep integration of mobile communication and intelligent terminal technology [4]–[6], the mobile crowd sensing (MCS) systems [7], [8] provide a new way to alleviate the traffic congestion of the transportation system [9]–[11], which works seamlessly through numerous mobile devices to upload collected sensory data to the cloud for further traffic analysis. For example, in this sensing paradigm, drivers can forward the traffic data obtained from mobile devices to the cloud. Then traffic data is analyzed and informed to the drivers or the relevant agencies to reflect the current road conditions. Such MCS systems have been widely exploited to large-scale vehicular sensing including traffic monitoring (e.g., collecting average speed or traffic density), real-time traffic prediction and many other application scenarios, which bring tremendous social and economic benefits in our daily life [12]–[14].

However, data collected by the mobile users is not always reliable [15]–[17], since sensor damage, hardware quality problems and the like that often occur during the collection process, where even the observation values of different users on the same objects may be quite different. Therefore, the power of crowd sensing system can be released completely only by filtering out the unreliable data. A possible solution is to aggregate the sensory data of all users who observe the same objects. However, it may lead to uncertainty of the final results as the reliability of each user is considered equal. To address this challenge, truth discovery [18]–[20], which aims to estimate the value (called *ground truth*) closest to the true value based on users' reliability (called *weight*) and inputs, have received much attention from both industry and academia. The main criteria for most of truth discovery methods are that the provider will be given a higher weight (i.e., reliability) if the data provided by him/her is closer to the *ground truth*, and the data provided by a user will be counted more in the aggregation procedure if this user has a higher weight. A variety of truth discovery approaches have been proposed to calculate user weight and aggregated results in a joint manner based on this principle.

Truth discovery mechanisms have been widely used to improve the accuracy of the aggregation in (MCS) systems.

Manuscript received October 9, 2018; revised November 29, 2018; accepted January 23, 2019. Date of publication January 29, 2019; date of current version April 16, 2019. This work was supported in part by the National Key R&D Program of China under Grants 2017YFB0802300 and 2017YFB0802000, in part by the National Natural Science Foundation of China under Grants 61802051, 61772121, 61872230, 61572311, 61728102, and 61472065, and in part by the Fundamental Research Funds for Chinese Central Universities under Grant ZYGX2015J056. The review of this paper was coordinated by Prof. G. Gui. *(Corresponding author: Hongwei Li.)*

G. Xu is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the CETC Big Data Research Institute Co., Ltd., Guiyang 550022, China (e-mail: guowen.xu@foxmail.com).

H. Li is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China, and also with the Science and Technology on Communication Security Laboratory, Chengdu 610041, China (e-mail: hongweili@uestc.edu.cn).

S. Liu is with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: 893551724@qq.com).

M. Wen is with the School of Computer Science and Engineering, Shanghai University of Electric Power, Shanghai 200090, China (e-mail: miwen@shiep.edu.cn).

R. Lu is with the Faculty of Computer Science, University of New Brunswick, Saint John, NB E3B5A3, Canada (e-mail: rlu1@unb.ca).

Digital Object Identifier 10.1109/TVT.2019.2895834

However, users' privacy, e.g., identity information, phone number, and personal health status may be implicitly included in the collected data, which may be abused or leaked by the cloud server (generally considered not completely trusted) if users submit their sensory data to the cloud without any pre-processing [21], [22]. Besides, some users may also try to trick the cloud server by providing fake data, since the private information of them may be inferred from raw data [23]–[25]. It further hinders the smooth implementation of the truth discovery.

To combat that, Miao *et al.* [26] proposed the first secure truth discovery scheme utilizing threshold paillier cryptosystem [27] to protect both the privacy of participants' data and reliability information. Later, Miao *et al.* [28] further presented a lightweight privacy-preserving truth discovery framework by introducing two non-colluding servers to reduce the cost on users. Zheng *et al.* [29] also devised a privacy-aware truth discovery based on a secure sum protocol in mobile crowdsensing. Unfortunately, all of these solutions require that users stay online at all times for interacting frequently with the cloud server consequently to ensure the smooth execution. Otherwise, the entire system will have to fail or start over. In real life, it is universal that some users may fail to send data to the cloud along with unreliable networks, human interventions, sensing device battery issues, etc. Besides, mobile devices are often widely distributed in various geographic locations, thus a practical truth discovery mechanism must be robust to users dropping out in any subprocess of the workflow.

Recently, two non-colluding servers [30], [31] are brought to alleviate the problem of all users keeping online under truth discovery. However, intuitively, no one can guarantee that two servers will never collude in real-world applications [32], [33], regardless of whether the two servers are from the same carrier. Motivating to resolve above challenge, in this paper, we propose an Efficient and Privacy-preserving Truth Discovery (EPTD) approach in mobile crowd sensing systems, which can endure the withdrawal of users in any subprocess of truth discovery, and requires only a single cloud server setting with limited trust. Compared with the preliminary version [34], our contributions are enhanced and can be summarized in three aspects as follows:

- Secret sharing technology and key agreement protocol among multi-users are exploited as the underlying structure in our EPTD, which can efficiently support users dropping out with proper modification.
- We design a double-masking with one-time pads protocol to achieve the high aggregated accuracy along with privacy protection on both users' data and reliability information under working process, where the privacy of those users who have logged out will be still protected.
- We present a comprehensive security analysis of our EPTD. We stress that even if the cloud server colludes with any set of less than t (explained in the following section) users, they will not get any useful information about other users' privacy (i.e., users' sensory data and reliability information), except what can be inferred from the aggregated results. Moreover, extensive experiments conducted on real-world mobile crowd sensing systems also demonstrate the practical performance of our proposed scheme.

The remainder of this paper is organized as follows. In Section II, we outline the problem statement. In Section III and Section IV, we describe the preliminaries and give an intuitive technical presentation about our core ideas. Then, we discuss the EPTD in detail in Section V and carry out the security analysis in Section VI, respectively. Next, performance evaluation and related work are discussed in VII and VIII. Finally, Section IX concludes the paper.

II. Problem Statement

A. Background on Truth Discovery

Specifically, truth discovery always starts from estimating each user's reliability subsequently to integrate the users' weights and sensory data for further inferring the ground truths. In this paper, truth discovery mechanism [7] is adopted in our EPTD due to its superior accuracy and efficiency in MCS systems, which can be divided into two closely related parts: *Weight Update phase* and *Truth Update phase*.

Suppose that a total of $\mathcal{H}$ objects' data need to be collected, and we use $\mathcal{D}$ to denote the number of users in our system. Hence the objected values of the d-th user for the h-th object can be represented as x_h^d, and the aggregated result (i.e., *ground truth*) of object h can be indicated as x_h^*.

Weight Update: Fixing the ground truth of each object x_h^*, every user's weight w_d is iteratively updated as follows.

$$w_d = f\left(\sum_{h=1}^{\mathcal{H}} d_{ist}\left(x_h^d, x_h^*\right)\right) \tag{1}$$

where f is a monotonically decreasing function, and $d_{ist}(\cdot)$ is to calculate the distance between each user's observed values and the corresponding ground truth. Besides, two types of data (i.e., continuous and categorical sensory data) are considered in our EPTD. For continuous data, the distance function is constructed as $d_{ist}(x_h^d, x_h^*) = (x_h^d - x_h^*)^2$. As for categorical data, we take a vector $x_h^d = (0, \ldots, 1(q\text{-th}), \ldots, 0)^T$ to denote the q-th choice chosen by user d for object h, and adopt $d_{ist}(x_h^d, x_h^*) = (x_h^d - x_h^*)^T(x_h^d - x_h^*)$ to indicate the distance between two vectors. In addition, the monotonically decreasing function f [7] is adopted in our EPTD, and the specific weight update is described as below.

$$w_d = log\left(\frac{\sum_{d'=1}^{\mathcal{D}} \sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^{d'}, x_h^*)}{\sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^d, x_h^*)}\right) \tag{2}$$

where if the data provided by a user is closer to the *ground truth*, the provider will be given a higher weight (i.e.,reliability).

Truth Update: Similarly, fixing the weight w_d of each user, the ground truth x_h^* of each object can be iteratively updated as follows.

$$x_h^* = \frac{\sum_{d=1}^{\mathcal{D}} w_d \cdot x_h^d}{\sum_{d=1}^{\mathcal{D}} w_d} \tag{3}$$

The final ground truth x_h^* of each object will be output by iteratively running the above two phases until satisfying the agreed-upon convergence conditions, where a data will be counted more in the aggregation procedure if the data provider has a higher weight. **Algorithm 1** shows the whole process of truth discovery.

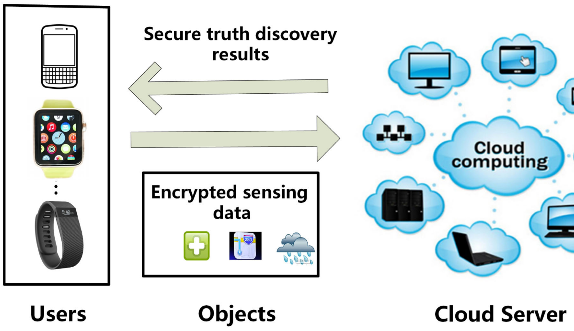


Fig. 1. System architecture.

Algorithm 1: Truth Discovery Process.

```
Input: $\mathcal{D}$ users, $\mathcal{H}$ objects, all users' observed data:
    $\{x_h^d\}, d = (1, 2, \cdots \mathcal{D}), h = (1, 2, \ldots \mathcal{H})$.
Output: Ground truth $\{x_h^*\}_{h=1}^{h=\mathcal{H}}$.
Initialize all ground truth data $x_h^*(h = 1, 2, \ldots \mathcal{H})$
    randomly and send them to each user.
for $d = 1$ to $d = \mathcal{D}$ do
    for $h = 1$ to $h = \mathcal{H}$ do
        Weight Update based on Eqn.(2)
        Truth Update based on Eqn.(3)
    end for
end for
output $\{x_h^*\}_{h=1}^{h=\mathcal{H}}$ when the recursion termination
    condition is satisfied.
```

Please note that we use a parameter L (a magnitude of 10) to round the fractional part of users' inputs in our protocol, since floating-point numbers are very common in real collection, and we can recover them by dividing L.

B. *System Architecture*

As shown in Fig. 1, we consider two types of entities in our system: the cloud server and users. Generally speaking, the cloud server first assigns tasks to various users and releases a set of objects to be collected by them. Then, each user will collect the sensor data of the targeted objects and get paid from the cloud server according to the prior agreement. For the purpose of privacy, the sensory data will be encrypted independently before being submitted to the cloud. Afterwards, the cloud server and users collaboratively execute the secure truth discovery mechanism (i.e.,EPTD) with constant interactions to find the ground truth.

The above system architecture is ubiquitous in MCS applications, such as indoor floorplan reconstruction [1], [8], where the cloud server can aggregate the sensory data collected by mobile devices to reconstruct the indoor floorplan. This will effectively improve the accuracy of the driver's 3D perspective during path planning. In addition, it is worth noting that our system does not require users to be online at all times to ensure the correct execution of EPTD.

C. *Threat Model*

The goal of our EPTD is to protect the privacy of users' sensory data and reliability information (i.e., weight) in whole working process, where all participants are deemed to be honest-but-curious [35]–[37]. Particularly, honest-but-curious indicates that all participants will strictly follow the agreement to execute instructions, yet it will also "curious" and try to spy on the users' private data independently.

Therefore, the cloud server is considered as the most threatening adversary in our EPTD since it possesses the full access right to all the encrypted sensory data. In our EPTD, we allow the cloud server to collude with any set of less than t users to attack EPTD for stealing users' privacy. However, they will not get any useful information except what can be inferred from the aggregated results. Please note that we do not consider the malicious operations of the cloud server and users, such as ingenious tampering with data and malicious execution protocols, which can be addressed by cryptographic techniques like zero-knowledge proofs and digital signature.

III. Preliminaries

In this section, we review the cryptographic primitives needed in our proposed model.

A. *Secret Sharing Protocol*

Shamir's t-out-of-$\mathcal{D}$ secret sharing protocol [38] is utilized in our work for splitting each user's secret s to $\mathcal{D}$ shares independently, where the secret s can be reconstructed with any t shares, but it is impossible to get any useful information about secrets even attackers possess $t - 1$ current secret shares. In our EPTD, each user can be distinctly represented in a finite field $\mathcal{F}$, where $\mathcal{F}$ is parameterized with size of $\mathcal{L} > 2^k$, and k is the security parameter. Here we use symbol $\mathcal{U}$ to denote the set of users' IDs, and the Shamir's t-out-of-$\mathcal{D}$ protocol can be described as below.

1) $\mathbf{Shamir.share}(s, t, \mathcal{U}) \rightarrow \{(d, s_d)\}_{d \in \mathcal{U}}$: Input a secret s, a threshold $t \leq |\mathcal{U}|$ and a set $\mathcal{U}$ denoted the users' ID in finite field $\mathcal{F}$, output the share s_d of secret for each user d, where $|\mathcal{U}| = \mathcal{D}$.
2) $\mathbf{Shamir.recon}(\{(d, s_d)\}_{d \in \mathcal{M}}, t) \rightarrow s$: Input a threshold t and a subset $\mathcal{M} \subseteq \mathcal{U}$, where $t \leq |\mathcal{M}|$. It outputs the secret s.

where the correctness requires that $\forall s \in \mathcal{F}$ and $\forall t, \mathcal{D}$ with $1 < t < \mathcal{D}$. We have $\mathbf{Shamir.recon}(\{(d, s_d)\}_{d \in \mathcal{M}}, t) \rightarrow s$ if $\mathbf{Shamir.share}(s, t, \mathcal{U}) \rightarrow \{(d, s_d)\}_{d \in \mathcal{U}}$, where $\mathcal{M} \subseteq \mathcal{U}$ and $t \leq |\mathcal{M}|$. Security requires $\forall s, s' \in \mathcal{F}$ and any $\mathcal{M} \subseteq \mathcal{U}$ with $t > |\mathcal{M}|$. We have

$$\{(d, s_d)\}_{d \in \mathcal{U}} \leftarrow \mathbf{Shamir.share}(s, t, \mathcal{U}) : \{(d, s_d)\}_{d \in \mathcal{M}}$$
$$\equiv$$
$$\{(d, s_d)\}_{d \in \mathcal{U}} \leftarrow \mathbf{Shamir.share}(s', t, \mathcal{U}) : \{(d, s_d)\}_{d \in \mathcal{M}}$$

where "≡" indicates the indistinguishability of two distribution.

B. Key Agreement

We exploit Diffie-Hellman key agreement called SIGMA [39] in our EPTD to generate private shared key between two users. Informally, SIGMA consists of three parts as below.

1) **KA.param** $(k) \rightarrow (\mathcal{G}, g, q, H)$: Given a security parameter k, it samples a group $\mathcal{G}$ with prime order q, and a generator g along with a hash function H, where H is always set as SHA-256 for practicability.
2) **KA.gen** $(\mathcal{G}, g, q, H) \rightarrow (x, g^x)$: Input a group $\mathcal{G}$ with prime order q, and a generator g along with a hash function H, it samples a random secret key $x \leftarrow \mathcal{Z}_q$ and a public key g^x, where x and g^x will be referred as D_d^{SK} and D_d^{PK} in following sections, respectively.
3) **KA.agree** $(sign_m(g^{x^d}, g^{x^m}), MAC_{k_v}(m), x_d, g^{x_m}, d, m) \rightarrow s_{d,m}$: Input the user d's secret key x_d, the public key g^{x_m} of user m, signed signature $sign_m(g^{x^d}, g^{x^m})$ and $MAC_{k_v}(m)$ from user m, it outputs the current shared session key between user d and m, where k_v is used as the MAC key. For simplicity, we use **KA.agree** $(x_d, g^{x_m}) \rightarrow s_{d,m}$ to represent above process in following sections.

where the correctness requires **KA.agree** $(D_d^{SK}, D_m^{PK})=$ **KA.agree**(D_m^{SK}, D_d^{PK}) for any public and private key generated by users d and m (utilizing the same parameters). Security requires that the shared key $s_{d,m}$ is indistinguishable from a uniformly random string to arbitrary adversaries who have public keys D_d^{PK} and D_m^{PK} (without the corresponding secret key D_d^{SK} and D_m^{SK}).

IV. Technical Intuition

We note that the truth discovery mainly involves the aggregation operations (i.e., computing $\sum_{d'=1}^{\mathcal{D}} \sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^{d'}, x_h^*)$, $\sum_{d=1}^{\mathcal{D}} w_d \cdot x_h^d$ and $\sum_{d=1}^{\mathcal{D}} w_d$) of multiple users' data in a secure manner. Other mathematical operations such as $\sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^d, x_h^*)$ and logarithm can be calculated in user-sides in the form of plaintext. In our EPTD, assuming each user holds the private input x_d, hence the goal of EPTD is to compute the $\sum_{d \in \mathcal{D}} x_d$ privately. As shown in Fig. 2, at a high level, we guarantee that the cloud server only learns the sum of users' inputs and that users learn nothing.

A. One Masking to Protect Security

Assume all of the users are ordered in the system according to certain criteria, and suppose that any pair of users $(d, m), d < m$ agree on a random value $r_{d,m}$, where user d adds the random value $r_{d,m}$ to x_d and user m subtracts it from x_m. Hence, the actual inputs of all users are perturbed but the aggregated results will float out if we add them together. In other words, each user d obscures his/her inputs as follows.

$$\mathbf{y}_d = x_d + \sum_{m \in \mathcal{D}: d<m} r_{d,m} - \sum_{m \in \mathcal{D}: d>m} r_{m,d} \pmod R \tag{4}$$

where we assume both x_d and $\sum_{m \in \mathcal{D}} r_{d,m}$ falling in $\mathcal{Z}_R$ with order R for simplicity.

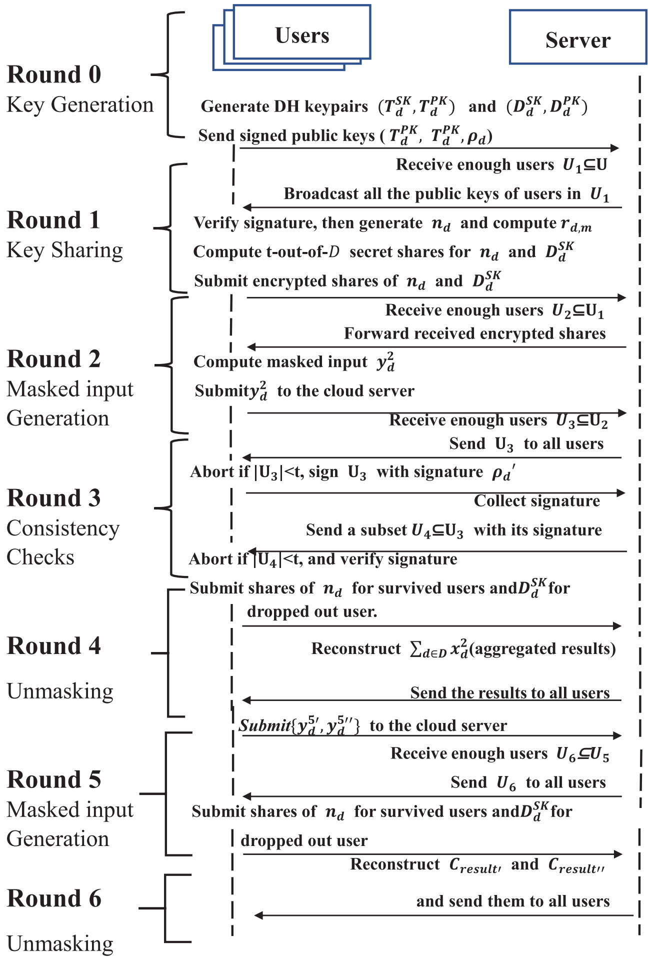

Users
Server
Round 0
Key Generation
Generate DH keypairs (T_d^{SK}, T_d^{PK}) and (D_d^{SK}, D_d^{PK})
Send signed public keys (T_d^{PK}, T_d^{PK}, ρ_d)
Receive enough users $U_1 \subseteq$U
Broadcast all the public keys of users in U_1
Round 1
Key Sharing
Verify signature, then generate n_d and compute $r_{d,m}$
Compute t-out-of-D secret shares for n_d and D_d^{SK}
Submit encrypted shares of n_d and D_d^{SK}
Receive enough users $U_2 \subseteq$U$_1$
Forward received encrypted shares
Round 2
Masked input
Generation
Compute masked input y_d^2
Submity_d^2 to the cloud server
Receive enough users $U_3 \subseteq$U$_2$
Send U$_3$ to all users
Abort if |U$_3$|<t, sign U$_3$ with signature $\rho_d{}'$
Round 3
Consistency
Checks
Collect signature
Send a subset $U_4 \subseteq$U$_3$ with its signature
Abort if |U$_4$|<t, and verify signature
Submit shares of n_d for survived users and$D_d^{SK}$for
dropped out user.
Round 4
Unmasking
Reconstruct $\sum_{d \in D} x_d^2$(aggregated results)
Send the results to all users
Submit$\{y_d^{5'}, y_d^{5''}\}$ to the cloud server
Receive enough users $U_6 \subseteq U_5$
Round 5
Masked input
Generation
Send U_6 to all users
Submit shares of n_d for survived users and$D_d^{SK}$for
dropped out user
Reconstruct $C_{result'}$ and $C_{result''}$
Round 6
Unmasking
and send them to all users

Fig. 2. High-level view of our EPTD.

Then, each $\mathbf{y}_d$ will be submitted to the cloud server, and the cloud server calculates

$$\begin{aligned}
\mathbf{z} &= \sum_{d \in \mathcal{D}} \mathbf{y}_d \\
&= \sum_{d \in \mathcal{D}} \left(x_d + \sum_{m \in \mathcal{D}: d<m} r_{d,m} - \sum_{m \in \mathcal{D}: d>m} r_{m,d} \right) \\
&= \sum_{d \in \mathcal{D}} x_d \pmod R
\end{aligned} \tag{5}$$

However, this method has two drawbacks. One is that each user d needs to agree on the random value $r_{d,m}$ with all other users. A naive way is one-to-one communication, which may cause quadratic communication overhead $(\mathcal{D}^2)$. Another is that this protocol is failure to users dropping out. If a user cannot summit $\mathbf{y}_d$ in time after agreeing on the random value $r_{d,m}$ with all other users, the random value $r_{d,m}$ associated with user d will not be eliminated in the final aggregated results.

B. Double-Masking with One-Time Pads

We note that we can use Pseudorandom Generator [40] to reduce the high communication overhead of each user, and utilize the shared keys obtained by engaging Diffie-Hellman key agreement between two users as the common seeds after the cloud server broadcasting all of the Diffie-Hellman public keys. Therefore, each user can calculate the masks $r_{d,m}$ by exploiting Diffie-Hellman key agreement and pseudorandom generator if he/she receives the public keys of corresponding users.

Threshold secret sharing scheme is adopted in our work to deal with the problem of users dropping out. Specifically, once some users cannot upload data in time, other users will submit shares of these users' secrets to the cloud for recovering masks associated with these users. In this way, our protocol can tolerate additional parties dropping out during recovery process as long as t (i.e., threshold of Shamir's secret sharing) users remain alive and reply shares of dropped users' key to the cloud server.

However, there is still a problem. Users' data may be inadvertently leaked to the cloud server. In real-life scenario, a user d may be too slow in sending $\mathbf{y}_d$ to the cloud server, which results that the cloud server may mistakenly think that user d has logged out and ask all other users to send their shares of d's key to it for removing masks associated with d. Just then, the cloud server receives the delayed $\mathbf{y}_d$ from d. As a result, the cloud server can now remove all the masks $r_{d,m}$ and get the plaintext of x_d.

To combat that, we improve the double-masking protocol proposed in [32] to ensure the privacy of x_d even the cloud server can remove the masks above of each user d. More concretely, each user randomly selects a seed $\boldsymbol{n_d}$ on the same round of generating $r_{d,m}$, and creates the shares of $\boldsymbol{n_d}$ and distributes them to all other users during the secret sharing round. Particularly, the generation process of $\mathbf{y}_d$ is shown as below.

$$\begin{aligned}\mathbf{y}_d = x_d + \mathbf{PRG}(\boldsymbol{n_d}) + \sum_{m\in\mathcal{D}:d<m} \mathbf{PRG}(r_{d,m})\\ - \sum_{m\in\mathcal{D}:d>m} \mathbf{PRG}(r_{m,d}) \pmod R\end{aligned} \tag{6}$$

where $\mathbf{PRG}(\boldsymbol{n_d})$ represents the pseudorandom generator with seed $\boldsymbol{n_d}$. When the cloud server needs to remove the double masks for obtaining the aggregated results $\sum_{d\in\mathcal{D}} x_d$, it will send a request to all alive users to get either the $r_{d,m}$ or the shares of $\boldsymbol{n_d}$ for each user d. After receiving $r_{d,m}$ from at least t users for all dropped users and t shares of $\boldsymbol{n_d}$ for all surviving users, it can get rid of the remaining masks and obtain the sum.

V. Our Proposed Scheme

In this section, we discuss our EPTD protocol in detail. Fig. 2 shows the high-level view of our proposed model, where has one cloud server and a set containing $\mathcal{D}$ users. Each user's inputs x_d will be submitted to the cloud privately, and the goal of the cloud server is to aggregate all the survivors' data. Every user can drop out of the protocol at any time, which means that these users stop uploading data and interacting with the cloud server completely. The cloud server can recover the aggregated results $\sum_{d\in\mathcal{D}} x_d$ as long as at least t users are still surviving in each round. For our purposes, we split our EPTD into *Secure Weight Update* and *Secure Truth Update* based on the properties of truth discovery.

A. Secure Weight Update

Round 0 *(Key Generation):* Assume there are $\mathcal{D}$ users in our EPTD. Given the threshold value t and security parameter k, three key pairs are created by trusted third party (TA) as follows.

$$\{(T_d^{PK}, T_d^{SK}), (D_d^{PK}, D_d^{SK}), (C_d^{PK}, C_d^{SK})\} \leftarrow \mathbf{KA}.\mathbf{gen}(k) \tag{7}$$

Then, each user signs public keys as $\rho_d \leftarrow \boldsymbol{sign}.(C_d^{SK}, T_d^{PK}||D_d^{PK})$, and sends $(\rho_d||T_d^{PK}||D_d^{PK})$ to the cloud server, where C_d^{SK} is the user d's secret key. Once the cloud server receives messages from at least t users (denoted as set $\mathcal{U}_1 \subseteq \mathcal{U}$ for these surviving users), it broadcasts $\{(m, T_m^{PK}, D_m^{PK}, \rho_m)\}_{m\in\mathcal{U}_1}$ to all users. For other cases, discard.

Round 1 *(Key Sharing):* For each user $m \in \mathcal{U}_1$, every user d needs to check whether $|\mathcal{U}_1| \geq t$ and utilizes public key C_m^{PK} to verify that the signature ρ_m is valid, while he/she receives the responses from the cloud server. If not, discard. Subsequently, each user randomly selects a parameter $\boldsymbol{n_d} \leftarrow \mathcal{F}$ and creates the shares of both D_d^{SK} and $\boldsymbol{n_d}$ as follows.

$$\begin{aligned}\{(m, D_{m,d}^{SK})\}_{m\in\mathcal{U}_1} &\leftarrow \mathbf{Shamir}.\mathbf{share}(D_d^{SK}, t, \mathcal{U}_1)\\ \{(m, \boldsymbol{n_{m,d}})\}_{m\in\mathcal{U}_1} &\leftarrow \mathbf{Shamir}.\mathbf{share}(\boldsymbol{n_d}, t, \mathcal{U}_1)\end{aligned} \tag{8}$$

Next, each user d exploits the symmetric Authenticated Encryption [41] to encrypt all of the shares above for all the users $m \in \mathcal{U}_1 \backslash \{d\}$ as below.

$$\begin{aligned}\mathcal{T}_{m,d} \leftarrow \mathbf{AE}.\mathbf{enc}\big(\mathbf{KA}.\mathbf{agree}(T_d^{SK}, T_m^{PK}),\\ d||m||D_{m,d}^{SK}||\boldsymbol{n_{m,d}}\big)\end{aligned} \tag{9}$$

where the symmetric authenticated encryption [41] can guarantee the confidentiality and integrity of messages exchanged between two parties. It possesses the indistinguishability under chosen plaintext attack (IND-CPA) and ciphertext integrity attack (IND-CTXT) [41]. Here we do not repeat them for simplicity.

We stress that if any of the above operations such as verification, key sharing and encryption fail, abort. Otherwise, each user d submits its $\mathcal{T}_{m,d}$ to the cloud server.

Once the cloud server receives messages from at least t users (denoted as set $\mathcal{U}_2 \subseteq \mathcal{U}_1$ for these surviving users), it initializes the ground truth $x_h^*(h = 1, 2, \cdots \mathcal{H})$ randomly, and broadcasts all of the $\{\mathcal{T}_{m,d}\}_{m\in\mathcal{U}_2}$ and x_h^* to all users. For other cases, discard.

Round 2 *(Masked Input Generation):* Similarly, each user first checks whether $|\mathcal{U}_2| \geq t$ upon receiving $\{\mathcal{T}_{m,d}\}_{m\in\mathcal{U}_2}$ and x_h^* $(h = 1, 2, \cdots \mathcal{H})$. Next, each user d computes $r_{d,m} \leftarrow \mathbf{KA}.\mathbf{agree}(D_d^{SK}, D_m^{PK})$ for every user $m \in \mathcal{U}_2 \backslash \{d\}$, and generates the masked input $\mathbf{y}_d^2$ as below.

$$\begin{aligned}\mathbf{y}_d^2 &= x_n^2 + \mathbf{PRG}(\boldsymbol{n_d}) + \sum_{m\in\mathcal{U}_2:d<m}\mathbf{PRG}(r_{d,m})\\ &\quad - \sum_{m\in\mathcal{U}_2:d>m}\mathbf{PRG}(r_{m,d}) \pmod R\\ &= \sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^d, x_h^*) + \mathbf{PRG}(\boldsymbol{n_d}) + \sum_{m\in\mathcal{U}_2:d<m}\mathbf{PRG}(r_{d,m})\\ &\quad - \sum_{m\in\mathcal{U}_2:d>m}\mathbf{PRG}(r_{m,d}) \pmod R\end{aligned} \tag{10}$$

where $\mathbf{y}_d^2$ denotes the masked input in the second round, and we use x_d^2 to indicate $\sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^d, x_h^*)$ for convenience of following description. Similarly, if any of the above operations such as **PRG** and key agreement fail, abort. Otherwise, all of the $\{\mathbf{y}_n^2\}_{n\in\mathcal{U}_2}$ will be sent to the cloud server.

Once the cloud server receives $\mathbf{y}_d^2$ from at least t users (denoted as set $\mathcal{U}_3 \subseteq \mathcal{U}_2$ for these surviving users), the list of $\mathcal{U}_3$ will be sent to all users. For other cases, discard.

Round 3 *(Consistency Checks):* Each user first checks whether $|\mathcal{U}_3| \geq t$ upon receiving the list of $\mathcal{U}_3$. Next, user d computes the signature $\rho_{d'} \leftarrow \boldsymbol{sign}.(C_d^{SK}, \mathcal{U}_3)$ and sends it to the cloud server.

Once the cloud server receives $\rho_{d'}$ from at least t users (denoted as set $\mathcal{U}_4 \subseteq \mathcal{U}_3$ for these surviving users), the list of $\{m, \rho_{m'}\}_{m\in\mathcal{U}_4}$ will be sent to all users. For other cases, discard.

Round 4 *(Unmasking):* For each user $m \in \mathcal{U}_4$, each user d first verifies whether $\mathcal{U}_4 \subseteq \mathcal{U}_3$ and $|\mathcal{U}_4| \geq t$, and that the signature $\rho_{m'}$ is valid by utilizing public key C_m^{PK} upon receiving the list $\mathcal{U}_4$. Next, user d decrypts $\mathcal{T}_{m,d}$ for those users $m \in \mathcal{U}_2\backslash\{d\}$ as below.

$$\begin{aligned}&d'||m'||D_{m,d}^{SK}||\boldsymbol{n_{m,d}}\\ &\quad\leftarrow \mathbf{AE}.\mathbf{dec}\left(\mathbf{KA}.\mathbf{agree}(T_d^{SK}, T_m^{PK}), \mathcal{T}_{m,d}\right)\end{aligned} \tag{11}$$

Then, both $D_{m,d}^{SK}$, $m \in \mathcal{U}_2\backslash\mathcal{U}_3$ and $\boldsymbol{n_{m,d}}$, $m \in \mathcal{U}_3$ will be sent to the cloud server if the condition $d = d' \wedge m = m'$ holds. Similarly, if any of above operations such as decryption and verification fail, abort.

Once the cloud server receives messages from at least t users (denoted as set $\mathcal{U}_5 \subseteq \mathcal{U}_4$ for these surviving users), both the secret key D_d^{SK} and masks $\mathbf{PRG}(r_{d,m})$, $d \in \mathcal{U}_2\backslash\mathcal{U}_3$ can be reconstructed as follows.

$$\begin{aligned}D_d^{SK} &\leftarrow \mathbf{Shamir}.\mathbf{recon}\left(\{(D_{m,d}^{SK})\}_{m\in\mathcal{U}_5}, t\right)\\ \mathbf{PRG}(r_{d,m}) &\leftarrow \mathbf{PRG}\left(\mathbf{KA}.\mathbf{agree}\left(\{D_d^{SK}, D_m^{PK}\}_{m\in\mathcal{U}_3}\right)\right)\end{aligned} \tag{12}$$

Similarly, the $\mathbf{PRG}(\boldsymbol{n_d})$, $d \in \mathcal{U}_3$ can be further reconstructed as follows.

$$\mathbf{PRG}(\boldsymbol{n_d}) \leftarrow \mathbf{PRG}\left(\mathbf{Shamir}.\mathbf{recon}\{(\boldsymbol{n_{m,d}}, t)\}_{m\in\mathcal{U}_5}\right) \tag{13}$$

Therefore, the final aggregated results of x_d^2 can be restored as follows.

$$\begin{aligned}\sum_{d\in\mathcal{U}_3} x_d^2 &= \sum_{d\in\mathcal{U}_3}\mathbf{y}_d^2 - \sum_{d\in\mathcal{U}_3}\mathbf{PRG}(\boldsymbol{n_d})\\ &\quad - \sum_{d\in\mathcal{U}_3, m\in\mathcal{U}_2\backslash\mathcal{U}_3:d<m}\mathbf{PRG}(r_{d,m})\\ &\quad + \sum_{d\in\mathcal{U}_3, m\in\mathcal{U}_2\backslash\mathcal{U}_3:d>m}\mathbf{PRG}(r_{m,d}) \pmod R\\ &= \sum_{d\in\mathcal{U}_3}\sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^d, x_h^*)\end{aligned} \tag{14}$$

In order to ensure that neither each user nor the cloud server knows the users' weight information, the cloud server randomly selects a positive noise r to obscure the raw aggregated result as below.

$$C_{result} = Log\left(\sum_{d\in\mathcal{U}_3}\sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^d, x_h^*)\right) + r \tag{15}$$

where we assume $t \cdot r > \max|\sum_{d=1}^{\mathcal{D}} x_h^d \cdot w_d|$. It is reasonable since the observed values x_h^d are often within a limited numerical range, and the values of w_d can be set within a certain positive range in advance. Then, the cloud server sends the C_{result} to all users.

Round 5 *(Masked Input Generation):* After receiving the C_{result}, each user d computes $r_{d,m} \leftarrow \mathbf{KA.agree}(D_d^{SK}, D_m^{PK})$ for every surviving user $m \in \mathcal{U}_5\backslash\{d\}$. Then, the masked weight of each user can be calculated as below.

$$\begin{aligned}\mathbf{y}_d^{5'} &= C_{result} - Log\left(\sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^d, x_h^*)\right)\\ &\quad + \mathbf{PRG}(\boldsymbol{n_d}) + \sum_{m\in\mathcal{U}_5:d<m}\mathbf{PRG}(r_{d,m})\\ &\quad - \sum_{m\in\mathcal{U}_5:d>m}\mathbf{PRG}(r_{m,d}) \pmod R\\ &= w_d + r + \mathbf{PRG}(\boldsymbol{n_d}) + \sum_{m\in\mathcal{U}_5:d<m}\mathbf{PRG}(r_{d,m})\\ &\quad - \sum_{m\in\mathcal{U}_5:d>m}\mathbf{PRG}(r_{m,d}) \pmod R\end{aligned} \tag{16}$$

For each object h, user d calculates

$$\begin{aligned}\mathbf{y}_d^{5''} &= \left(C_{result} - Log\left(\sum_{h=1}^{\mathcal{H}} d_{ist}(x_h^d, x_h^*)\right)\right) \cdot x_h^d\\ &\quad + \mathbf{PRG}(\boldsymbol{n_d}) + \sum_{m\in\mathcal{U}_5:d<m}\mathbf{PRG}(r_{d,m})\\ &\quad - \sum_{m\in\mathcal{U}_5:d>m}\mathbf{PRG}(r_{m,d}) \pmod R\end{aligned}$$

$$\begin{aligned}&= (w_d + r) \cdot x_h^d + \mathbf{PRG}(\boldsymbol{n_d}) + \sum_{m \in \mathcal{U}_5 : d < m} \mathbf{PRG}(r_{d,m}) \\ &\quad - \sum_{m \in \mathcal{U}_5 : d > m} \mathbf{PRG}(r_{m,d}) \pmod R\end{aligned} \tag{17}$$

We use $(\mathbf{y}_d^{5'}, x_d^{5'}), (\mathbf{y}_d^{5''}, x_d^{5''})$ to denote the masked input and raw data input pairs in the fifth round, where $x_d^{5'} = w_d + r$, $x_d^{5''} = (w_d + r) \cdot x_h^d$, respectively. Similarly, if any of the above operations such as **PRG** and key agreement fail, abort. Otherwise, all of the $\{\mathbf{y}_d^{5'}, \mathbf{y}_d^{5''}\}_{d \in \mathcal{U}_5}$ will be sent to the cloud server.

For simplicity, we did not re-execute the *Key Generation* round and reselect random values $\boldsymbol{n_d}$ in *Key Sharing* round. Therefore, all the signature operations and consistency verification operations are hidden in the following rounds.

B. Secure Truth Update

For the cloud server, after receiving $(\mathbf{y}_d^{5'}, \mathbf{y}_d^{5''})$ from at least t users (Denoted as set $\mathcal{U}_6 \subseteq \mathcal{U}_5$ for these alive users), it sends the list of $\mathcal{U}_6$ to all users. For other cases, abort.

Round 6 *(Unmasking):* Once each user receives the list $\mathcal{U}_6$, he first needs to verify $\mathcal{U}_6 \subseteq \mathcal{U}_5$ and $|\mathcal{U}_6| \geq t$. Otherwise, abort. Then, for each user $m \in \mathcal{U}_5 \backslash \{d\}$, user d decrypts the ciphertext $\mathcal{T}_{m,d}$ as follows.

$$\begin{aligned}&d' || m' || D_{m,d}^{SK} || \boldsymbol{n_{m,d}} \\ &\quad \leftarrow \mathbf{AE.dec}\left(\mathbf{KA.agree}(T_d^{SK}, T_m^{PK}), \mathcal{T}_{m,d}\right)\end{aligned} \tag{18}$$

Next, user d checks whether $d = d' \wedge m = m'$ and sends both $D_{m,d}^{SK}$ for every user $m \in \mathcal{U}_5 \backslash \mathcal{U}_6$ as well as $\boldsymbol{n_{m,d}}$ for every user $m \in \mathcal{U}_6$ to the cloud server. Similarly, if any of above operations such as decryption and verification fail, abort.

After receiving the responses from at least t users (Denoted as set $\mathcal{U}_7 \subseteq \mathcal{U}_6$ for these alive users), the cloud server reconstructs the secret key D_d^{SK} and masks $\mathbf{PRG}(r_{d,m})$ for each user $d \in \mathcal{U}_5 \backslash \mathcal{U}_6$ as below.

$$\begin{aligned}&D_d^{SK} \leftarrow \mathbf{Shamir.recon}\left(\{(D_{m,d}^{SK})\}_{m \in \mathcal{U}_7}, t\right) \\ &\mathbf{PRG}(r_{d,m}) \leftarrow \mathbf{PRG}\left(\mathbf{KA.agree}\left(\{D_d^{SK}, D_m^{PK}\}_{m \in \mathcal{U}_6}\right)\right)\end{aligned} \tag{19}$$

Similarly, the cloud server further reconstructs all the $\{\mathbf{PRG}(\boldsymbol{n_d})\}_{d \in \mathcal{U}_6}$ as follows.

$$\mathbf{PRG}(\boldsymbol{n_d}) \leftarrow \mathbf{PRG}\left(\mathbf{Shamir.recon}\{(\boldsymbol{n_{m,d}}, t)\}_{m \in \mathcal{U}_7}\right) \tag{20}$$

Hence, the actual aggregated results can be restored as below.

$$\begin{aligned}\sum_{d \in \mathcal{U}_6} x_d^{5'} &= \sum_{d \in \mathcal{U}_6} \mathbf{y}_d^{5'} - \sum_{d \in \mathcal{U}_6} \mathbf{PRG}(\boldsymbol{n_d}) \\ &\quad - \sum_{d \in \mathcal{U}_6, m \in \mathcal{U}_5 \backslash \mathcal{U}_6 : d < m} \mathbf{PRG}(r_{d,m}) \\ &\quad + \sum_{d \in \mathcal{U}_6, m \in \mathcal{U}_5 \backslash \mathcal{U}_6 : d > m} \mathbf{PRG}(r_{m,d}) \pmod R \\ &= \sum_{d \in \mathcal{U}_6} (w_d + r)\end{aligned} \tag{21}$$

Similarly,

$$\begin{aligned}\sum_{d \in \mathcal{U}_6} x_d^{5''} &= \sum_{d \in \mathcal{U}_6} \mathbf{y}_d^{5''} - \sum_{d \in \mathcal{U}_6} \mathbf{PRG}(\boldsymbol{n_d}) \\ &\quad - \sum_{d \in \mathcal{U}_6, m \in \mathcal{U}_5 \backslash \mathcal{U}_6 : d < m} \mathbf{PRG}(r_{d,m}) \\ &\quad + \sum_{d \in \mathcal{U}_6, m \in \mathcal{U}_5 \backslash \mathcal{U}_6 : d > m} \mathbf{PRG}(r_{m,d}) \pmod R \\ &= \sum_{d \in \mathcal{U}_6} (w_d + r) \cdot x_h^d\end{aligned} \tag{22}$$

Based on the assumption $t \cdot r > \max |\sum_{d=1}^{\mathcal{D}} x_h^d \cdot w_d|$, and $|\mathcal{U}_6| \geq t$, the cloud server can eliminate the random values r as follows.

$$\begin{aligned}C_{result'} &= \sum_{d \in \mathcal{U}_6} x_d^{5'} \quad mod \left(\sum_{d \in \mathcal{U}_6} r\right) \\ &= \sum_{d \in \mathcal{U}_6} w_d + \sum_{d \in \mathcal{U}_6} r \quad mod \left(\sum_{d \in \mathcal{U}_6} r\right) \\ &= \sum_{d \in \mathcal{U}_6} w_d \\ C_{result''} &= \sum_{d \in \mathcal{U}_6} x_d^{5''} \quad mod \left(\sum_{d \in \mathcal{U}_6} r\right) \\ &= \sum_{d \in \mathcal{U}_6} (w_d + r) \cdot x_h^d \quad mod \left(\sum_{d \in \mathcal{U}_6} r\right) \\ &= \sum_{d \in \mathcal{U}_6} w_d \cdot x_h^d\end{aligned} \tag{23}$$

Hence, the temporary ground truth x_h^* of each object h can be deduced according to the Eqn.(2), and we will get the final ground truth x_h^* by repeating the rounds 0-6 until the convergence conditions are met.

Discussion: According to the protocol of our EPTD, our model can tolerate users offline at any stage, while guaranteeing confidentiality of users' data and weight under a single cloud server setting. However, it is obvious that our EPTD requires frequent interactions between participants to complete the aggregation. This will result in a certain communication overhead with the increase of total number of users in the system. Although some recent works [30], [31] brought two non-colluding servers to alleviate this problem, as mentioned before, no one can guarantee that two servers will never collude in real-world applications [32], [33], regardless of whether the two servers are from the same carrier. We have not yet found a better solution up to now, but the problem of frequent interactions between participants will be our main focus of our attention in the future studies.

VI. Security Analysis

We stress that our EPTD is secure against *honest but curious* setting, where the cloud server and users (less than t) will not infer any privacy data even if they collude with each other, except what can be inferred from the aggregated results.

As explained before, the security parameter of the cryptographic primitives in EPTD is k, and the cloud server S interacts with a set $\mathcal{U}$ of $\mathcal{D}$ users. Because users can drop out in any time under the workflow, we utilize the subset $\mathcal{U}_7 \subseteq \mathcal{U}_6 \subseteq \mathcal{U}_5 \subseteq \mathcal{U}_4 \subseteq \mathcal{U}_3 \subseteq \mathcal{U}_2 \subseteq \mathcal{U}_1 \subseteq \mathcal{U}$ to represent the set of surviving users in corresponding running rounds. For example, users in $\mathcal{U}_5 \backslash \mathcal{U}_6$ denote those users that correctly send message to the cloud server in round 4, but dropout before submitting data in round 5. For simplicity, we adopt x_d to indicate the input of user d. In addition, the *view* of a party consists of its input and all of the messages received from other parties in our model, and the *view* of a party will be not extended past the last messages received if this party drops out. Given a threshold t, and any subset $\mathcal{M} \subseteq \mathcal{U} \cup \{S\}$ of parties, the real *joint view* of these parties can be described as $\mathbf{Real}_{\mathcal{M}}^{\mathcal{U},t,k}(x_d, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \mathcal{U}_4, \mathcal{U}_5, \mathcal{U}_6, \mathcal{U}_7)$. Hence, we have

Theorem 1 (Against Users Attack Only, Under Honest But Curious Setting): Given a threshold t, for all security parameter k, $\mathcal{U}$ with $|\mathcal{U}| > t$, $C \subseteq \mathcal{U}$, and $\mathcal{U}_7 \subseteq \mathcal{U}_6 \subseteq \mathcal{U}_5 \subseteq \mathcal{U}_4 \subseteq \mathcal{U}_3 \subseteq \mathcal{U}_2 \subseteq \mathcal{U}_1 \subseteq \mathcal{U}$, there exists a PPT simulator **SIM** whose output is indistinguishable from the output of $\mathbf{Real}_{\mathcal{M}}^{\mathcal{U},t,k}$ in polynomial time.

$$\begin{aligned} &\mathbf{Real}_{\mathcal{C}}^{\mathcal{U},t,k}(x_d, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \mathcal{U}_4, \mathcal{U}_5, \mathcal{U}_6, \mathcal{U}_7) \\ &\equiv \\ &\mathbf{SIM}_{\mathcal{C}}^{\mathcal{U},t,k}(x_c, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \mathcal{U}_4, \mathcal{U}_5, \mathcal{U}_6, \mathcal{U}_7) \end{aligned} \tag{24}$$

Proof: In this theorem, we assume that the cloud server is honest, but some users will collude with each other in order to illegally access other users' private data. This kind of assumption is prevalent in real life, such as the famous event of Apple icloud leakage [42], where a number of hackers (also legitimate users of Apple) teamed up to crack the Apple system and steal a lot of celebrity photos. We take **Round 0** to **Round 4** (i.e., a complete aggregation process) for example to prove our theorem. Intuitively, the **SIM** can easily deceive those honest but curious users in $\mathcal{C}$ since the *view* of the cloud server is omitted. More concretely, in **Round 0**, for all users in $\mathcal{C}$, **SIM** creates the true key pairs $(T_d^{PK}, T_d^{SK}) \leftarrow \mathbf{KA}.\mathbf{gen}(k)$, $(D_d^{PK}, D_d^{SK}) \leftarrow \mathbf{KA}.\mathbf{gen}(k)$, and sends the public keys (T_d^{PK}, D_d^{PK}) to the cloud server. For other honest users, **SIM** can product a perfect simulation by uniformly generating the dummy keys (such as a vector of 0s) based on the Diffie-Hellman assumption, and output the simulated *view* of $\mathcal{C}$. Similarly, in **Round 1** and **Round 2**, only the values sent by users in $\mathcal{C}$ are depended on the true inputs and executions. All other users' inputs can be replaced by uniform dummy values based on the properties of Shamir's t-out-of-$\mathcal{D}$ secret sharing protocol [38] and Two Oracle Diffie-Hellman assumption (2ODH) [32]. Besides, the cloud server only returns a list of users' ID to all users in each round except in **Round 4**, but not the specific value of $\mathbf{y}_d$. Therefore, the **SIM** only needs to guarantee the dummy value r selected in **Round 4** that meets the condition $t \cdot r > \max |\sum_{d=1}^{\mathcal{D}} x_h^d \cdot w_d|$, which means that **SIM** can successfully generate a perfect simulation by using random values to replace real inputs for all honest users, and the *view* of $\mathbf{SIM}_{\mathcal{C}}^{\mathcal{U},t,k}(x_c, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \mathcal{U}_4, \mathcal{U}_5, \mathcal{U}_6, \mathcal{U}_7)$ is indistinguishable from the output of $\mathbf{Real}_{\mathcal{W}}^{\mathcal{U},t,k}$ in polynomial time.

Theorem 2 (Against Joint Attack By Server And Users, Under Honest But Curious Setting): Given a threshold t, for all security parameter k, $\mathcal{U}$ with $|\mathcal{U}| > t$, $C \subseteq \mathcal{U} \cup \{S\}$, $|C \backslash \{S\}| < t$ and $\mathcal{U}_7 \subseteq \mathcal{U}_6 \subseteq \mathcal{U}_5 \subseteq \mathcal{U}_4 \subseteq \mathcal{U}_3 \subseteq \mathcal{U}_2 \subseteq \mathcal{U}_1 \subseteq \mathcal{U}$, there exists a PPT simulator **SIM** whose output is indistinguishable from the output of $\mathbf{Real}_{\mathcal{W}}^{\mathcal{U},t,k}$ in polynomial time.

$$\begin{aligned} &\mathbf{Real}_{\mathcal{C}}^{\mathcal{U},t,k}(x_d, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \mathcal{U}_4, \mathcal{U}_5, \mathcal{U}_6, \mathcal{U}_7) \\ &\equiv \\ &\mathbf{SIM}_{\mathcal{C}}^{\mathcal{U},t,k}(x_c, \mathcal{U}_1, \mathcal{U}_2, \mathcal{U}_3, \mathcal{U}_4, \mathcal{U}_5, \mathcal{U}_6, \mathcal{U}_7) \end{aligned} \tag{25}$$

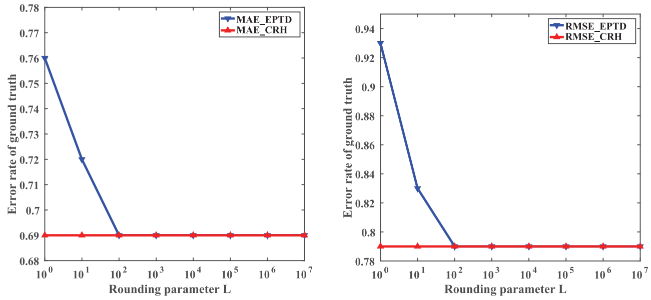
Proof: In this theorem, we allow the cloud server to collude with any set of less than t users to attack EPTD for stealing users' privacy. This assumption is also very common in real life. Examples include the Baidu Tieba event in China [43], where the Baidu authorities sell the data of the post users to third parties without the users' permission. The idea of proof is similar to Theorem 1. The **SIM** will try to deceive those honest but curious parties in $\mathcal{C}$ by utilizing dummy values to replace the true inputs of honest users not in $\mathcal{C}$. Our goal is to protect both the sensory data and weight privacy of the users who belong to $\mathcal{U} \backslash \{C \cup \{S\}\}$ even if the cloud server colludes with any set of less than t users. Here we omit the specific proof and interested readers can refer to the literature [32] for more relevant details.

VII. Performance Evaluation

To evaluate the performance of EPTD, we recruit 100 users with mobile devices (including mobile phones and smart watches) to detect about 40 objects and collect their sensory data in the application of floorplan construction [18], where floorplan construction has recently drawn much attention since many location-based services can be facilitated by it [44]–[46]. In our experiments, the observations were mainly the height, length, layout, and other sensory data of a building, and the floorplan can be automatically reconstructed by integration of these sensor data. Most smart devices come with 1GB of RAM and are equipped with Android 6.0 system. Besides, the " Cloud " is simulated with a Lenovo server which has Intel(R) Xeon(R)E5-2620 2.10GHZ CPU, 16GB RAM, 256SSD, 1TB mechanical hard disk and runs on the Ubuntu 18.04 operating system.

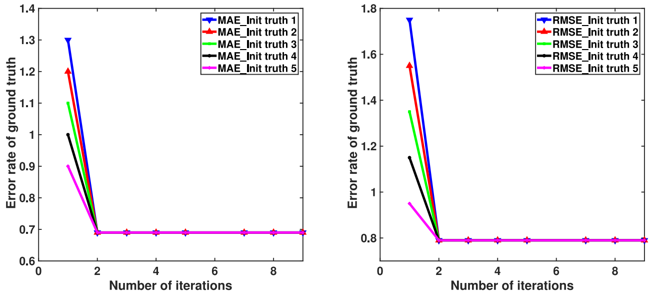
A. Accuracy

As mentioned in Section. II, we use a parameter L to round the fractional part of users' inputs in EPTD, and we know that the accuracy of the final ground truths may be affected by the value of L, because the fractional part of some inputs are discarded if we take a small L. In our experiments, we exploit mean of absolute error (MAE) and the root of mean squared error (RMSE) to measure the error rates between EPTD and the original truth discovery approach named CRH [7], respectively, and consequently to evaluate the impact of L on the ground truths. Fig. 3 shows the error rate of ground truth under different number of rounding parameter L, where we make the number



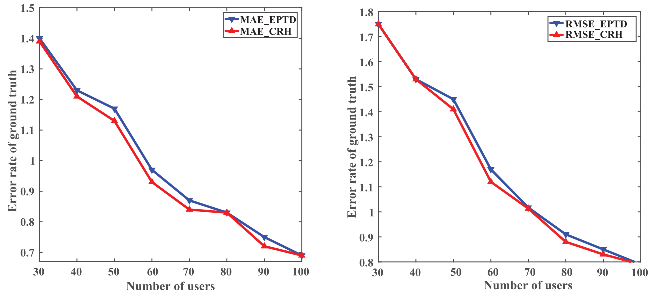
(a) (b)

Fig. 3. Error rate of ground truth for different rounding parameter L. (a) MAE. (b) RMSE.



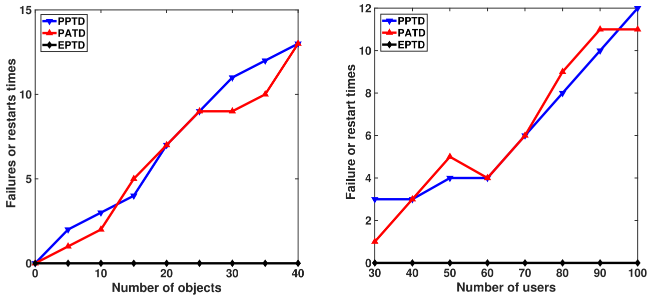
(a) (b)

Fig. 5. Error rate of ground truth for different number of iterations. (a) MAE. (b) RMSE.



(a) (b)

Fig. 4. Error rate of ground truth for different number of users. (a) MAE. (b) RMSE.



(a) (b)

Fig. 6. Failure or restart time. (a) For the different number of objects. (b) For the different number of users.

of objects and users 40 and 100, respectively, and output the average of 10 experimental results. We can see that EPTD and CRH have basically no difference in error rate while choosing a large L like 10^7. Therefore, it can be concluded that with the increase of L, EPTD has a gradual increase in the accuracy since less data are missed in the workfolw.

Besides, we also evaluate the error rate of ground truth with different number of users under MAE and RMSE. Similarly, the number of objects is set as 40, and we take 10^7 for rounding parameter L. As shown in Fig. 4, it is obvious that the error rate of EPTD is almost same as the CRH despite some users dropping out in execution process.

B. Convergence

We further analyze the convergence of our proposed model, where the number of objects and L are also set as 40 and 10^7, respectively. We take 5 different initial values (i.e., random ground truth x_m^*) and compute the error rate of ground truth for different number of iterations. Based on the Fig. 5, we can see that our EPTD will converge quickly with a few iterations regardless the values of initial inputs.

C. Robust to Users Dropping Out

To analyze the universality of users dropping out from the running process, we count the times of failure or restart of EPTD compared with recent works PPTD [26] and PPAD [29], where both PPTD and PPAD are considered as failure once a user quits in whole truth discovery process, and EPTD is deemed to be a failure when only the current online user is less than the threshold t ($t = 25$ in this experiment) set in advance. It should be noted that literatures [30], [31] are not considered in our experiment since their two non-colluding servers setting is completely different from our purpose. Fig. 6 shows the specific failures or restart times under diverse number of objects and users, where we repeat the experiment 10 times and output the average. We can see that the events of users dropping out occur frequently on both PPTD [26] and PPAD [29]. For example, at least 10 users quit during the entire running process while we initialize the number of users and objects as 100 and 40 respectively. However, our EPTD is robust to users dropping out as long as the number of surviving users are more than the threshold t.

D. Communication Overhead

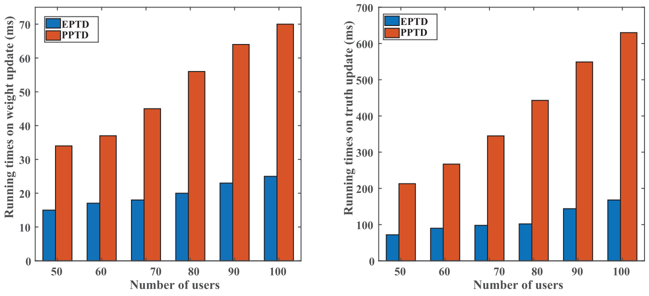
We also analyze the communication overhead of our proposed scheme by comparing with PPTD [26], where we fix the number of users and objects as 100 and 40 in Table I and Table II respectively. We count the communication overhead of a user on one complete iterative process and output the average values of 10 experiments, where the communication overhead of the cloud server is not considered in our experiments since it can be deemed as the sum of the communication overhead of all users. As shown in Table I and Table II, we can conclude that the cost of PPTD is much higher than EPTD regardless of the growth of users or objects. One of the main reasons is that a secure sum protocol based on threshold paillier [27] is used in

TABLE I
UNDER DIFFERENT NUMBER OF OBJECTS

	EPTD	PPTD
Objects=15	340.55KB	1.67MB
Objects=20	440.67KB	2.01MB
Objects=25	539.77KB	2.32MB
Objects=30	638.44KB	2.61MB
Objects=35	741.09KB	2.94MB
Objects=40	850.41KB	3.23MB

TABLE II
UNDER DIFFERENT NUMBER OF USERS

	EPTD	PPTD
Users=50	209.65KB	1.81MB
Users=60	302.16KB	2.14MB
Users=70	411.56KB	2.33MB
Users=80	537.91KB	2.61MB
Users=90	681.11KB	2.92MB
Users=100	847.15KB	3.22MB



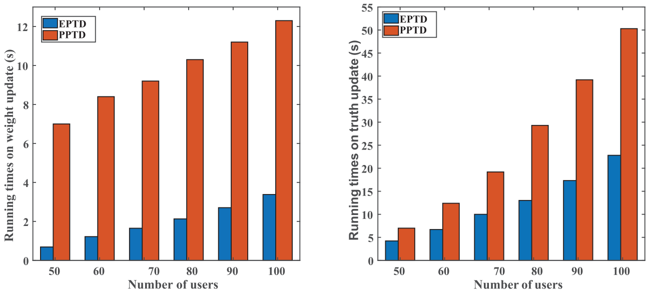
(a) (b)

Fig. 7. Running time of each user. (a) Weight update under different number of users. (b) Truth update under different number of users.

PPTD. Besides, the overhead of EPTD is reasonable since most existing mobile devices are equipped with more than 100MB of RAM, thus the impact on overall performance will be very small if we run EPTD on these mobile devices.

E. Computational Overhead

At last, Fig. 7 and Fig. 8 show the comparison between EPTD and PPTD on the performance of computational overhead, where the overall running time is divided into two parts (i.e., *Weight update* and *Truth update*), and the number of objects is also fixed as 40 in Fig. 7 and Fig. 8, respectively. Fig. 7 shows the running time of each user under *Weight update* and *Truth update*. It is clear that the running time of EPTD is small and far less than PPTD. Besides, Fig. 8(a) and Fig. 7(b) are also demonstrate the lower running time of EPTD compared with PPTD.



(a) (b)

Fig. 8. Running time of the cloud server. (a) Weight update under different number of users. (b) Truth update under different number of users.

VIII. RELATED WORK

Truth discovery has been widely used in many MCS scenarios such as environment monitoring, smart transportation, and health diagnosis [47]–[49]. However, the cloud server can easily access the users' raw data and further stealing the data privacy if there is no security mechanism before submitting sensory data to the cloud server. Li *et al.* [7] proposed a general truth discovery approach to deal with the single data type or heterogeneous data types in execution. However, this solution only considers the efficiency issues, and the privacy protection of users' data is not in its work. For privacy-assured truth discovery, Miao *et al.* [26] proposed the first secure truth discovery scheme utilizing threshold paillier cryptosystem [27] to protect both the privacy of participants' data and reliability information. However, huge communication and computational overhead are generated in their protocol due to the complexity of threshold paillier cryptosystem [27]. Later, Miao *et al.* [28] further presented a lightweight privacy-preserving truth discovery framework by introducing two non-colluding servers to reduce the overhead on users. Zheng *et al.* [29] also devised a privacy-aware truth discovery in mobile crowdsensing through a secure sum protocol. However, all of these solutions require participants to keep online for frequently interacting with the cloud server. Recently, literatures [30], [31] bring two non-colluding servers in their works for reducing the overhead of users-side and alleviating the problem of all users keeping online under truth discovery. However, intuitively, it is not easy to ensure that two servers do not collude in a real-world scenario [32], [33], regardless of whether the two servers belong to the same carrier. Therefore, it is a challenge to propose a practical and privacy-aware truth discovery approach which is fully robust to users dropping out under a single server setting.

IX. CONCLUSION

In this paper, we have presented the EPTD approach in mobile crowd sensing systems, which can achieve high accuracy and protection of both the users' sensory data and weight privacy even if the cloud server colludes with any set of less than t users. Besides, our EPTD is fully robust to users dropping out at any time point, and constructed on more practical single server setting. Extensive experiments conducted on real-world mobile crowd sensing systems also demonstrate the desired performance of EPTD compared with existing models. As for the future research direction, we will consider to further improve the communication overhead of EPTD. Besides, it is also an interesting point to design a more secure protocol against active attacks from the cloud server.

REFERENCES

[1] H. Ma, D. Zhao, and P. Yuan, "Opportunities in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 29–35, Aug. 2014.

[2] B. Guo, Z. Yu, L. Chen, X. Zhou, and X. Ma, "Mobigroup: Enabling lifecycle support to social activity organization and suggestion with mobile crowd sensing," *IEEE Trans. Human-Mach. Syst.*, vol. 46, no. 3, pp. 390–402, Jun. 2016.

[3] H. Li, R. Lu, L. Zhou, B. Yang, and X. Shen, "An efficient merkle-tree-based authentication scheme for smart grid," *IEEE Syst. J.*, vol. 8, no. 2, pp. 655–663, Jun. 2014.

[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] T. G. Rodrigues, K. Suto, H. Nishiyama, and N. Kato, "Hybrid method for minimizing service delay in edge cloud computing through VM migration and transmission power control," *IEEE Trans. Comput.*, vol. 66, no. 5, pp. 810–819, May 2017.

[6] Y. Li, X. Cheng, and G. Gui, "Co-robust-admm-net: Joint admm framework and DNN for robust sparse composite regularization," *IEEE Access*, vol. 6, pp. 47 943–47 952, 2018.

[7] 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.

[8] X. Zhang *et al.*, "Incentives for mobile crowd sensing: A survey," *IEEE Commun. Surv. Tut.*, vol. 18, no. 1, pp. 54–67, Jan.–Mar. 2016.

[9] Z. M. Fadlullah *et al.*, "State-of-the-art deep learning: Evolving machine intelligence toward tomorrow's intelligent network traffic control systems," *IEEE Commun. Surv. Tut.*, vol. 19, no. 4, pp. 2432–2455, Oct.–Dec. 2017.

[10] N. Kato *et al.*, "The deep learning vision for heterogeneous network traffic control: Proposal, challenges, and future perspective," *IEEE Wireless Commun.*, vol. 24, no. 3, pp. 146–153, Jun. 2017.

[11] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and doa estimation based massive mimo system," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8549–8560, Sep. 2018.

[12] H. Zhao, Z. Wang, and F. Nie, "A new formulation of linear discriminant analysis for robust dimensionality reduction," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 4, pp. 629–640, Apr. 2019.

[13] Y. Zhao, D. Wang, J. Hu, and K. Yang, "H-ap deployment for joint wireless information and energy transfer in smart cities," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7485–7496, Aug. 2018.

[14] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2053–2064, Aug. 2014.

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

[16] S. Zhang, H. Li, Y. Dai, J. Li, M. He, and R. Lu, "Verifiable outsourcing computation for matrix multiplication with improved efficiency and applicability," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 5076–5088, Dec. 2018, doi: 10.1109/JIOT.2018.2867113.

[17] H. Ren, H. Li, Y. Dai, K. Yang, and X. Lin, "Querying in internet of things with privacy preserving: Challenges, solutions and opportunities," *IEEE Netw.*, vol. 32, no. 6, pp. 144–151, Nov./Dec. 2018.

[18] R. W. Ouyang, M. Srivastava, A. Toniolo, and T. J. Norman, "Truth discovery in crowdsourced detection of spatial events," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 4, pp. 1047–1060, Apr. 2016.

[19] M. Wan, X. Chen, L. Kaplan, J. Han, J. Gao, and B. Zhao, "From truth discovery to trustworthy opinion discovery:an uncertainty-aware quantitative modeling approach," in *Proc. ACM SIGKDD*, 2016, pp. 1885–1894.

[20] C. Huang, D. Wang, and N. Chawla, "Scalable uncertainty-aware truth discovery in big data social sensing applications for cyber-physical systems," *IEEE Trans. Big Data*, to be published, doi: 10.1109/TBDATA.2017.2669308.

[21] H. Corrigangibbs and B. Dan, "Prio: Private, robust, and scalable computation of aggregate statistics," in *Proc. 14th USENIX Conf. Netw. Syst. Des. Implementation*, 2017, pp. 259–282.

[22] H. Li, D. Liu, Y. Dai, T. H. Luan, and X. S. Shen, "Enabling efficient multi-keyword ranked search over encrypted mobile cloud data through blind storage," *IEEE Trans. Emerg. Topics Comput.*, vol. 3, no. 1, pp. 127–138, Mar. 2015.

[23] Y. Zhang, C. Xu, X. Liang, H. Li, Y. Mu, and X. Zhang, "Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation," *IEEE Trans. Inf. Forensics Secur.*, vol. 12, no. 3, pp. 676–688, Mar. 2017.

[24] G. Xu, H. Li, Y. Dai, K. Yang, and X. Lin, "Enabling efficient and geometric range query with access control over encrypted spatial data," *IEEE Trans. Inf. Forensics Secur.*, vol. 14, no. 4, pp. 870–885, Apr. 2019.

[25] H. Li, Y. Yi, Y. Dai, and Y. Xiang, "Achieving secure and efficient dynamic searchable symmetric encryption over medical cloud data," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2017.2769645.

[26] C. Miao *et al.*, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *Proc. 13th ACM Conf. Embedded Netw. Sensor Syst.*, 2015, pp. 183–196.

[27] I. Damgrd and M. Jurik, "A generalisation, a simpli.cation and some applications of paillier's probabilistic public-key system," in *Proc. Int. Workshop Pract. Theory Public Key Cryptography*, 2001, pp. 119–136.

[28] 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 INFOCOM*, 2017, pp. 1–9.

[29] Y. Zheng, H. Duan, X. Yuan, and C. Wang, "Privacy-aware and efficient mobile crowdsensing with truth discovery," *IEEE Trans. Dependable Secure Comput.*, to be published, doi: 10.1109/TDSC.2017.2753245.

[30] X. Tang, C. Wang, X. Yuan, and Q. Wang, "Non-interactive privacy-preserving truth discovery in crowd sensing applications," in *Proc. IEEE INFOCOM*, 2018, pp. 1–9.

[31] Y. Zheng, H. Duan, and C. Wang, "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.

[32] K. Bonawitz *et al.*, "Practical secure aggregation for privacy-preserving machine learning," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2017, pp. 1175–1191.

[33] J. Liu, M. Juuti, Y. Lu, and N. Asokan, "Oblivious neural network predictions via minionn transformations," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2017, pp. 619–631.

[34] G. Xu, H. Li, and R. Lu, "POSTER:practical and privacy-aware truth discovery in mobile crowd sensing systems," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2018, pp. 2312–2314.

[35] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: When QoE meets QoP," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 74–80, Aug. 2015.

[36] H. Li, D. Liu, Y. Dai, T. Luan, and S. Yu, "Personalized search over encrypted data with efficient and secure updates in mobile clouds," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 97–109, Jan.–Mar. 2018.

[37] X. Li, Y. Zhu, J. Wang, Z. Liu, Y. Liu, and M. Zhang, "On the soundness and security of privacy-preserving SVM for outsourcing data classification," *IEEE Trans. Dependable Secure Comput.*, vol. 15, no. 5, pp. 906–912, Sep./Oct 2018.

[38] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[39] H. Krawczyk, "Sigma: The SIGn-and-MAc approach to authenticated Diffie–Hellman and its use in the IKE protocols," *Proc. Crypto*, vol. 2729, no. 2, pp. 400–425, 2003.

[40] M. Blum and S. Micali, "How to generate cryptographically strong sequences of pseudo-random bits," in *Proc. Symp. Found. Comput. Sci.*, 2008, pp. 112–117.

[41] M. Bellare and C. Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," *J. Cryptology*, vol. 21, no. 4, pp. 469–491, 2008.

[42] J. Lee, H. Chung, C. Lee, and S. Lee, "Methodology for digital forensic investigation of icloud," in *Information Technology Convergence, Secure and Trust Computing, and Data Management*. New York, NY, USA: Springer, 2012, pp. 197–206.

[43] M. Xu, Y. Ma, X. Liu, F. X. Lin, and Y. Liu, "Appholmes: Detecting and characterizing app collusion among third-party android markets," in *Proc. 26th Int. Conf. Committee*, 2017, pp. 143–152.

[44] J. Liu, H. Nishiyama, N. Kato, and J. Guo, "On the outage probability of device-to-device-communication-enabled multichannel cellular networks: An RSS-threshold-based perspective," *IEEE J. Select. Areas Commun.*, vol. 34, no. 1, pp. 163–175, Jan. 2016.

[45] Y. Wu *et al.*, "Secrecy-based delay-aware computation offloading via mobile edge computing for internet of things," *IEEE Internet Things J.*, to be published, doi: 10.1109/JIOT.2018.2875241.

[46] G. Gui, H. Huang, Y. Song, and H. Sari, "Deep learning for an effective nonorthogonal multiple access scheme," *IEEE Trans. Veh. Technol.*, vol. 67, no. 9, pp. 8440–8450, Sep. 2018.

[47] 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. Manag. Data*, 2014, pp. 1187–1198.

[48] G. Xu, H. Li, D. Liu, H. Ren, Y. Dai, and X. Liang, "Towards efficient privacy-preserving truth discovery in crowd sensing systems," in *Proc. IEEE Global Commun. Conf.*, Washington, DC, USA, Dec. 4–8, 2016, pp. 1–6.

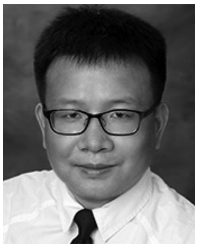
[49] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 3, pp. 312–325, May/Jun. 2016.

Guowen Xu (S'15) received the B.S. degree in information and computing science from the Anhui University of Architecture, Hefei, China, in 2014. He is currently working toward the Ph.D. degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include cryptography, Searchable encryption, and the privacy issues in deep learning.

Hongwei Li (M'12–SM'18) received the Ph.D. degree from the University of Electronic Science and Technology of China, Chengdu, China, in June 2008. He is currently the Head and a Professor with the Department of Information Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China. He worked as a Postdoctoral Fellow with the University of Waterloo from October 2011 to October 2012 under the supervision of Prof. S. Shen. He has authored and coauthored more than 80 technical papers. He is the sole author of the book *Enabling Secure and Privacy Preserving Communications in Smart Grids* (Springer, 2014). His research interests include network security and applied cryptography. His research is supported by the National Science Foundation of China and Ministry of Science and Technology of China, Ministry of Industry and Information Technology, and China Unicom. He is currently the Associate Editor of the IEEE INTERNET OF THINGS JOURNAL and *Peer-to-Peer Networking and Applications*, the Guest Editor of the IEEE NETWORK and IEEE INTERNET OF THINGS JOURNAL. He also serves/served the technical symposium co-chair of ACM TUR-C 2019, IEEE ICCC 2016, IEEE GLOBECOM 2015, and IEEE BigDataService 2015, and many technical program committees for international conferences, such as IEEE INFOCOM, IEEE ICC, IEEE GLOBECOM, IEEE WCNC, IEEE SmartGridComm, BODYNETS, and IEEE DASC. He has received the Best Paper Award from IEEE MASS 2018 and IEEE HELTHCOM 2015. He is a Distinguished Lecturer of IEEE Vehicular Technology Society.

Sen Liu (S'18) received the B.S. degree in information security from Guizhou University, Guiyang, China, in 2017. He is currently working toward the master's degree with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China. His research interests include cryptography and Searchable encryption.

Mi Wen (M'10) received the Ph.D. degree from Shanghai Jiao Tong University, Shanghai, China, in 2008. She is currently the Chairman of CCF Young Computer Scientists & Engineers Forum Shanghai and a Professor of the College of Computer Science and Technology with the Shanghai University of Electric Power, Shanghai, China. Her research interests include privacy preserving in wireless networks, big data, smart grid, etc. Her research is supported by the National Science Foundation of China, Shanghai Science and Technology Commission, and China Scholarship Council. She is an Associate Editor of *Peer-to-Peer Networking and Applications* (Springer). She acts as the TPC Member of many conferences such as the IEEE INFOCOM, IEEE ICC, and IEEE GLOEBECOM.



Rongxing Lu (S'09–M'11–SM'15) received the Ph.D. degree from the University of Waterloo, Waterloo, ON, Canada, in 2012. Since August 2016, he has been an Assistant Professor with the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Canada. He was an Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, from April 2013 to August 2016. He worked as a Postdoctoral Fellow with the University of Waterloo from May 2012 to April 2013. He was awarded the most prestigious "Governor Generals Gold Medal" and received the 8th IEEE Communications Society Asia Pacific Outstanding Young Researcher Award in 2013. He is currently a Senior Member of IEEE Communications Society. He is currently the Vice-Chair (Publication) of IEEE ComSoc CIS-TC. He is the Winner of 2016–2017 Excellence in Teaching Award, FCS, UNB.