# A Lightweight Privacy-Preserving Truth Discovery Framework for Mobile Crowd Sensing Systems

Chenglin Miao*, Lu Su*, Wenjun Jiang*, Yaliang Li* and Miaomiao Tian†
*Department of Computer Science and Engineering, SUNY at Buffalo, Buffalo, NY USA
†School of Computer Science and Technology, Anhui University, Hefei, P. R. China
Email: {cmiao, lusu, wenjunji, yaliangl}@buffalo.edu, miaotian@mail.ustc.edu.cn

***Abstract*—The recent proliferation of human-carried mobile devices has given rise to the mobile crowd sensing (MCS) systems. However, the sensory data provided by the participating workers are usually not reliable. As an efficient technique to extract truthful information from unreliable data, truth discovery has drawn significant attention. Currently, the privacy concern of the participating workers poses a major challenge on the design of truth discovery mechanisms. Although the existing mechanism can conduct truth discovery with high accuracy and strong privacy guarantee, tremendous overhead is incurred on the worker side. In this paper, we propose a novel lightweight privacy preserving truth discovery framework, $L$-PPTD, which is implemented by involving two non-colluding cloud platforms and adopting additively homomorphic cryptosystem. This framework not only achieves the protection of each worker's sensory data and reliability information but also introduces little overhead to the workers. In order to further reduce each worker's overhead in the scenarios where only the sensory data need to be protected, we propose another more lightweight framework named $L^2$-PPTD. The desirable performance of the proposed frameworks is verified through extensive experiments conducted on real world MCS systems.**

## I. Introduction

Driven by the explosion of mobile devices (e.g., smartphones, smartwatches, and smartglasses) equipped with various sensors (e.g., accelerometer, compass, GPS, camera), mobile crowd sensing (MCS) [1–3] has recently emerged as a widely employed sensing paradigm. In a typical MCS system, the collection of sensory data is outsourced to a large crowd of users (usually referred to as *workers*) carrying mobile devices. Such MCS systems potentially can serve a wide spectrum of applications that have significant societal and economic impacts, including urban and environment monitoring, smart transportation, healthcare, etc.

However, in MCS systems, the sensory data collected by individual workers are usually not reliable. The reasons include environment noise, the hardware quality, as well as the ways in which workers use the hardware. A possible solution is to aggregate the sensory data of multiple workers who observe the same objects (or events). When aggregating crowd sensing data, however, the traditional methods (e.g., average and voting) would not be able to derive accurate aggregated results, since they regard all the workers equally. An ideal approach should have the capability to capture the difference in the quality of information among different participating workers. However, the challenge here is that the reliability level (referred to as *weight*) of each worker is usually unknown *a priori*. To address this challenge, the problem of truth discovery [4–9], which aims at discovering truthful facts from unreliable data, has recently been widely studied. The common principle of truth discovery approaches is that a worker will be assigned a higher weight if her data is closer to the aggregated results, and the data of a worker will be counted more in the aggregation procedure if she has a higher weight.

Although truth discovery approaches have brought significant improvement to the aggregation accuracy in MCS systems, they fail to address the privacy concerns of the participating workers. In many MCS applications, the workers are usually not willing to provide their sensory data since the private information of them may be inferred from these data. For example, through crowd wisdom system, some difficult questions can be solved by aggregating the answers collected from a large crowd of workers. However, the personal information of each worker can be inferred from her answers.

To address this challenge, a recent paper [10] presents a mechanism called PPTD, which adopts *threshold Paillier cryptosystem* [11] to protect each worker's private information. This mechanism can achieve strong privacy guarantee, however, at a cost of significant computation and communication overhead. The reason is that each worker in this mechanism has to conduct considerable amount of ciphertext-based calculations and communication with the cloud server during the truth discovery procedure. In MCS systems, the mobile device carried by each participating worker usually has limited energy resources. Therefore, there is a great need to design a privacy-preserving truth discovery scheme which can not only guarantee high accuracy and strong privacy protection but also introduce little overhead to the participating workers.

In the light of this need, in this paper we propose a novel lightweight privacy-preserving truth discovery framework ($L$-PPTD) for MCS systems, in which the sensory data and reliability information of each worker are both protected from being disclosed to others. The proposed framework is implemented by involving two non-colluding cloud platforms and adopting additively homomorphic cryptosystem. In this framework, the aggregated results (referred to as *truth*) are cooperatively estimated by the two cloud platforms without disclosing any worker's private information. Additionally, instead of directly encrypting the data to be uploaded, each worker in this framework preserves her privacy through perturbing the data with some random numbers, and all the ciphertext-based

978-1-5090-5336-0/17/$31.00 ©2017 IEEE

calculations are moved onto the cloud side, which substantially reduce the overhead incurred on each worker.

Although $L$-PPTD can achieve tremendous reduction of the overhead on the worker side, each worker is still responsible for calculating her weight so as to protect reliability information. To further reduce the workload of workers, we propose a more lightweight framework ($L^2$-PPTD) suiting for the scenarios where only the sensory data of each worker need to be protected. In this framework, all each worker needs to do is just uploading the perturbed sensory data and random numbers before the truth discovery procedure starts.

In summary, the main contributions of this paper are:

- In order to protect each worker's sensory data and reliability information, we propose a novel lightweight privacy-preserving truth discovery framework called $L$-PPTD, which significantly reduces the overhead on the worker side.
- For the scenarios where only the sensory data need to be protected, a more lightweight framework, $L^2$-PPTD, is proposed to further reduce the overhead on each worker.
- Extensive experiments based on real world MCS systems are conducted to evaluate the performance of the proposed framework.

## II. Problem Formulation

In this section, we formulate the problem addressed by the proposed lightweight privacy preserving truth discovery framework. This framework consists of two different types of parties: *data requester* and *participating workers*. The data requester is an individual or organization who posts sensing tasks which usually require the observations on a collection of *objects* (e.g., the potholes or litters in geotagging campaigns), while the participating workers are a group of mobile device users who carry out the sensing tasks and generate sensory data with their mobile devices.

In MCS systems, the sensory data and the reliability information of each participating worker may be disclosed to the data requester or other workers during the data aggregation process, resulting in the leakage of workers' privacy. Here we mainly consider the attacks coming from the inside malicious parties (data requester or participating workers). For the sake of curiosity and financial purpose, the data requester may try to infer the sensitive personal information of each participating worker. On the other hand, each participating worker may also want to know the private information of other workers. In this paper, we adopt the *semi-honest threat model*, in which all the parties will strictly follow the designed protocols, but each of them may backup all the data she has sent and received, and then try to learn the private information of other parties. Additionally, we assume there is no collusion in the designed framework, which means the parties will not collude with each other outside the designed protocols.

The problem addressed in this paper is formulated as follows: Suppose there are $M$ objects in the posted sensing task, denoted as $O = \{o_1, o_2, ..., o_M\}$, and these objects will be observed by $K$ participating workers represented as $U = \{u_1, u_2, ..., u_K\}$. We use $W = \{w_1, w_2, ..., w_K\}$ to denote the weights (i.e., reliability) of these workers. Let $x_m^k$ denote the sensory data of worker $u_k$ for object $o_m$. For every object $o_m \in O$, there is a ground truth which is not known by all the parties in this framework. Our goal is *to calculate the estimated values $\{x_m\}_{m=1}^M$ of the ground truths for all the objects while protecting the sensory data and reliability information of each worker from being disclosed to others.*

## III. Preliminary

In this section, we will review the concepts and general procedure of truth discovery and additively homomorphic encryption, which are the two major techniques adopted in our proposed framework.

### A. Truth Discovery

The truth discovery approaches usually take a two-step iterative procedure:

*1) Weight Estimation:* In this step, each worker's weight will be estimated based on the difference between its sensory data and the estimated truths. Typically, the weight of a worker $u_k$ is calculated as $w_k = f(\sum_{m=1}^M d(x_m^k, x_m))$, where $f$ is a monotonically decreasing function, and $d(x_m^k, x_m)$ is the distance function which is used to measure the difference between workers' sensory data and the estimated truths.

In the proposed framework, we consider both cases when the sensory data are continuous or categorical. For continuous data, the squared distance function $d(x_m^k, x_m) = (x_m^k - x_m)^2$ is adopted. For categorical data, we assume each task has multiple candidate results or answers. Then the sensory data $x_m^k = (0, ..., \underset{q}{1}, ..., 0)^T$ represents worker $k$ selects the $q$-th result or answer for object $o_m$. In this case, the distance function is defined as $d(x_m^k, x_m) = (x_m^k - x_m)^T(x_m^k - x_m)$.

In this paper, we aim to develop a general framework that is compatible with different types of function $f$. Without loss of generality, we will first use the following logarithmic weight function adopted in the widely used truth discovery method CRH [4, 5] as an example when presenting our framework

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

and then discuss the generalization of the proposed framework to other weight functions.

*2) Truth Estimation:* After worker weights are calculated, the ground truth for each object $o_m$ can be estimated as

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

When the sensory data is continuous, this value is actually the weighted average of the workers' observations on object $o_m$. But when the data is categorical, $x_m$ is a vector in which each element represents the probability of a particular candidate result or answer being the truth. The estimated truth of object $o_m$ will be the result or answer with the largest value in vector $x_m$.

In truth discovery algorithms, the above two steps will be iteratively conducted until some convergence criterion is satisfied. The convergence criterion can be a predefined iteration

number or a threshold of the change in the estimated truths in two consecutive iterations.

### B. Additively Homomorphic Encryption

Let $\mathcal{M}$ denote the message space, an encryption scheme is said to be additively homomorphic if the encryption function $E$ satisfies

$$\forall\, m_1, m_2 \in \mathcal{M}, \quad E[m_1 + m_2] = E[m_1] \oplus E[m_2] \tag{3}$$

$$\forall\, m_3 \in \mathcal{M}, \quad E[a \cdot m_3] = a \otimes E[m_3] \tag{4}$$

for some operators $\oplus$ and $\otimes$, where $a$ is a constant. In other words, by using the additively homomorphic cyptosystem, the encrypted sum of messages can be directly calculated from the ciphertexts of these messages.

In this paper, we adopt a widely used additively homomorphic cryptosystem, namely, Paillier's cryptosystem [12], in which the message $m \in \mathbb{Z}_n$ ($\mathbb{Z}_n$ is the set of integers modulo the large positive integer $n$) can be encrypted as $E[m] = g^m r^n mod\ n^2$ with the public key $pk = (n, g)$, where $r \in \mathbb{Z}_n^*$ ($\mathbb{Z}_n^*$ denotes the multiplicative group of invertible elements of $\mathbb{Z}_n$) is privately and randomly selected by the user who calculates the ciphertexts. Then, $\forall\, m_1, m_2, m_3 \in \mathbb{Z}_n$, the following equations show the additively homomorphic properties of Paillier's cryptosystem:

$$D(E[m_1 + m_2]) = D(E[m_1] \cdot E[m_2]) = m_1 + m_2 \tag{5}$$

$$D(E[a \cdot m_3]) = D(E[m_3]^a) = a \cdot m_3 \tag{6}$$

where $D(\cdot)$ is the decryption function with the private key $sk$.

## IV. $L$-PPTD Framework

In this section, we will first introduce the $L$-PPTD framework, and then analyze its computational complexity and communication overhead.

In this framework, we aim to protect the sensory data and reliability information of each worker from being disclosed to others while accurately estimating the object truths. To achieve this goal, two non-colluding cloud platforms are involved as the third parties in $L$-PPTD. As shown in Fig. 1, we use $S_A$ and $S_B$ to denote the two cloud platforms respectively. The key idea of $L$-PPTD is to calculate the summed distances (i.e., $\sum_{k=1}^{K}\sum_{m=1}^{M} d(x_m^k, x_m)$) used to estimate worker weights through the cooperation of $S_A$ and $S_B$ without letting $S_A$ or $S_B$ know the raw data of each worker. With this distance information, each worker will be able to calculate her weight by herself. After the weight of each user is derived, the object truth can be estimated via a similar cooperation between $S_A$ and $S_B$. The final estimated object truths will be forwarded to the data requester.

### A. The Detailed Procedure of L-PPTD

As shown in Fig. 1, $L$-PPTD is mainly composed of two phases: *Initialization Phase* and *Iteration Phase*. Next, we will elaborate on each step in the two phases.

*1) Initialization Phase:* This phase will be conducted only once during the whole truth discovery procedure. Firstly, cloud $S_B$ generates the public key $pk$ and private key $sk$. The public key can be known by all the parties while the private key is only known by $S_B$. Then the following steps are taken.

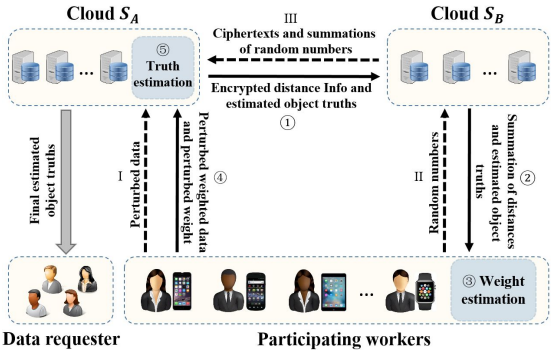



**Fig. 1:** The workflow of $L$-PPTD

**Step I:** Each worker $u_k$ generates random numbers $\{\alpha_m^k\}_{m=1}^M$, $\{\beta_m^k\}_{m=1}^M$ and $\gamma_k$, which are used to perturb her sensory data, weighted data and weight respectively. For categorical data, $\alpha_m^k$ and $\beta_m^k$ are two vectors in which each element is a random. Then $u_k$ perturbs the sensory data $x_m^k$ as $\tilde{x}_m^k = x_m^k - \alpha_m^k$, and uploads all the perturbed data $\{\tilde{x}_m^k\}_{m=1}^M$ to cloud $S_A$.

**Step II:** Each worker uploads all the random numbers generated in step I to cloud $S_B$. Thus, the random numbers of each worker are only known by herself and cloud $S_B$.

**Step III:** After receiving the random numbers from each worker, cloud $S_B$ calculates the ciphertexts $\{E[\alpha_m^k]\}_{k,m=1}^{K,M}$ and summations $\{\sum_{k=1}^K \beta_m^k\}_{m=1}^M$, $\sum_{k=1}^K \gamma_k$. For categorical data, $E[\alpha_m^k]$ represents a ciphertext vector in which each element is the ciphertext of the corresponding element in vector $\alpha_m^k$. Then the ciphertexts and the summations are sent to cloud $S_A$.

Please note that when the data needed to be encrypted in this paper is not integer, we will round it by multiplying a parameter $R$ (a magnitude of 10) and at last the original data will be recovered by dividing the same parameter.

*2) Iteration Phase:* This phase starts with the random initialization of the object truths on cloud platform $S_A$.

**Step ①:** Based on the homomorphic property of the Paillier cryptosystem, cloud $S_A$ firstly calculates the ciphertext $C_{conti}$ for continuous data as

$$\begin{aligned} C_{conti} =& E[\sum_{k=1}^{K}\sum_{m=1}^{M}(x_m^k - x_m)^2 - \sum_{k=1}^{K}\sum_{m=1}^{M}(\alpha_m^k)^2] \\ =& \prod_{k=1}^{K}\prod_{m=1}^{M} E[(x_m^k - x_m)^2 - (\alpha_m^k)^2] \\ =& \prod_{k=1}^{K}\prod_{m=1}^{M} E[(x_m^k - \alpha_m^k - x_m)^2 + 2\alpha_m^k(x_m^k - \alpha_m^k - x_m)] \\ =& \prod_{k=1}^{K}\prod_{m=1}^{M} \{E[(\tilde{x}_m^k - x_m)^2] \cdot E[\alpha_m^k]^{2(\tilde{x}_m^k - x_m)}\} \end{aligned} \tag{7}$$

where $E[\alpha_m^k]$ is received from cloud $S_B$. Based on the perturbed sensory data $\tilde{x}_m^k$ received from the $k$-th worker, and the estimated truth $x_m$ for each object $o_m$ calculated in the previous iteration ($x_m$ is randomly initialized in the first iteration), $E[(\tilde{x}_m^k - x_m)^2]$ is calculated with public key $pk$.

For categorical data, we assume $x_{mi}^k$, $\alpha_{mi}^k$, $\tilde{x}_{mi}^k$ and $x_{mi}$ represents the $i$-th element of vector $x_m^k$, $\alpha_m^k$, $\tilde{x}_m^k$ and $x_m$ respectively, and the number of elements in each vector is $l$. Similarly, The ciphertext $C_{cate}$ can be calculated by $S_A$ as

$$\begin{aligned} C_{cate} =& E[\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{i=1}^{l}(x_{mi}^k - x_{mi})^2 - \sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{i=1}^{l}(\alpha_{mi}^k)^2] \\ =& \prod_{k=1}^{K}\prod_{m=1}^{M}\prod_{i=1}^{l}E[(x_{mi}^k - x_{mi})^2 - (\alpha_{mi}^k)^2] \\ =& \prod_{k=1}^{K}\prod_{m=1}^{M}\prod_{i=1}^{l}\{E[(\tilde{x}_{mi}^k - x_{mi})^2] \cdot E[\alpha_{mi}^k]^{2(\tilde{x}_{mi}^k - x_{mi})}\}. \end{aligned} \tag{8}$$

At last, the ciphertext $C_{conti}$ or $C_{cate}$ together with the estimated object truths $\{x_m\}_{m=1}^M$ are sent to cloud $S_B$.

**Step ②:** After receiving the ciphertext, cloud $S_B$ decrypts it with his private key $sk$ and calculates the summation of distances (i.e., $\sum_{k'=1}^{K}\sum_{m=1}^{M} d(x_m^{k'}, x_m)$ in Equation (1)) by adding the value $\sum_{k=1}^{K}\sum_{m=1}^{M}(\alpha_m^k)^2$ (for categorical data, the value is $\sum_{k=1}^{K}\sum_{m=1}^{M}\sum_{i=1}^{l}(\alpha_{mi}^k)^2$) to the decrypted data. Then the summation $\sum_{k=1}^{K}\sum_{m=1}^{M} d(x_m^k, x_m)$ together with the estimated object truths $\{x_m\}_{m=1}^M$ are sent to each worker.

**Step ③:** In this step, each worker $u_k$ first calculates $\sum_{m=1}^{M} d(x_m^k, x_m)$ based on the estimated object truths $\{x_m\}_{m=1}^M$ received from cloud $S_B$, then estimates her weight $w_k$ according to Equation (1).

**Step ④:** After the weight is estimated, each worker $u_k$ calculates the weighted data $w_k x_m^k$ for object $o_m$ and perturbs it as $w_k x_m^k - \beta_m^k$, where $\beta_m^k$ is the random number (random number vector for categorical data) generated in the initialization phase. Additionally, the weight $w_k$ is perturbed as $w_k - \gamma_k$. Then the perturbed weighted data (i.e., $w_k x_m^k - \beta_m^k$) and perturbed weight (i.e., $w_k - \gamma_k$) are sent to cloud $S_A$.

**Step ⑤:** Based on the information received from all the participating workers, cloud $S_A$ first calculates the value $\sum_{k=1}^{K}(w_k x_m^k - \beta_m^k)$ for each object $o_m$, then derives the summation of the weighted data as

$$\sum_{k=1}^{K} w_k x_m^k = \sum_{k=1}^{K}(w_k x_m^k - \beta_m^k) + \sum_{k=1}^{K}\beta_m^k \tag{9}$$

where the value $\sum_{k=1}^{K}\beta_m^k$ is received from cloud $S_B$ in the initialization phase. Similarly, the summation of all workers' weights is calculated as $\sum_{k=1}^{K} w_k = \sum_{k=1}^{K}(w_k - \gamma_k) + \sum_{k=1}^{K}\gamma_k$. With all the above information, $S_A$ is able to estimate the object truth $x_m$ according to Equation (2).

In this phase, step ① $\sim$ ⑤ are repeated until the convergence criterion is satisfied. The final estimated truth for each object will be sent to the data requester by cloud $S_A$. Please note that, for categorical data, the final result for each object $o_m$ should be the candidate answer with the largest value in the vector $x_m$ calculated in the final iteration.

In the procedure of $L$-PPTD, each worker only communicates with the two cloud platforms once respectively in each iteration and all the calculations on the worker side are based on plaintexts. Thus, very little overhead will be introduced to each worker, which is confirmed by the experimental results presented in section VI.

### B. Security Analysis

The security goal of $L$-PPTD can be summarized as Theorem 1, followed by the proof.

**Theorem 1.** *Suppose the number of participating workers satisfies $K \geq 3$ and for each object, there are at least two workers providing different sensory data. If the parties (including the two cloud platforms) are semi-honest and there is no collusion among them, the sensory data and weight information of each worker will not be disclosed to any other party under the $L$-PPTD framework.*

*Proof.* Firstly, we prove the security of workers' private information on the cloud side. For cloud $S_A$, besides the ciphertexts $\{E[\alpha_m^k]\}_{k,m=1}^{K,M}$, he knows the plaintexts $\{x_m^k - \alpha_m^k\}_{k,m=1}^{K,M}$, $\{w_k x_m^k - \beta_m^k\}_{k,m=1}^{K,M}$, $\{w_k - \gamma_k\}_{k=1}^{K}$, $\{\sum_{k=1}^{K}\beta_m^k\}_{m=1}^{M}$, $\sum_{k=1}^{K}\gamma_k$ and $\{x_m\}_{m=1}^{M}$. Since the private key $sk$ is only known by cloud $S_B$, cloud $S_A$ cannot decrypt the ciphertexts. Although the values $\{\sum_{k=1}^{K}\beta_m^k\}_{m=1}^{M}$ and $\sum_{k=1}^{K}\gamma_k$ are known by cloud $S_A$, he cannot learn anything about the individual random numbers just based on these summations. In this way, as long as the two cloud platforms do not collude with each other, cloud $S_A$ cannot infer the plaintexts of $\{x_m^k\}_{k,m=1}^{K,M}$, $\{w_k x_m^k\}_{k,m=1}^{K,M}$ and $\{w_k\}_{k=1}^{K}$. For cloud $S_B$, he knows the plaintexts of $\{\alpha_m^k\}_{k,m=1}^{K,M}$, $\{\beta_m^k\}_{k,m=1}^{K,M}$, $\{\gamma_k\}_{k=1}^{K}$, $\{x_m\}_{m=1}^{M}$ and $\sum_{k=1}^{K}\sum_{m=1}^{M} d(x_m^k, x_m)$. However, based on these values, he cannot learn anything about the private information of each worker.

On the worker side, besides the sensory data $\{x_m^{k'}\}_{m=1}^{M}$ and weight $w_{k'}$, each worker $u_{k'}$ also knows the plaintexts of $\{x_m\}_{m=1}^{M}$ and $\sum_{k=1}^{K}\sum_{m=1}^{M} d(x_m^k, x_m)$, based on which the value $\sum_{k=1}^{K}\sum_{m=1}^{M} d(x_m^k, x_m) - \sum_{m=1}^{M} d(x_m^{k'}, x_m)$ can be calculated. However, since the number of all workers satisfies $K \geq 3$, worker $u_{k'}$ cannot infer the private information of any other individual workers. For the data requester, he knows nothing about workers' private information except the final aggregated results $\{x_m\}_{m=1}^{M}$.

In summary, the sensory data and weight information of each worker will not be disclosed to other parties under the $L$-PPTD framework. □

### C. Efficiency Analysis

Next, we will discuss the computational complexity and communication overhead of $L$-PPTD.

*1) Computational Complexity:* On the worker side, all the computations are conducted on the plaintexts, and thus will introduce less overhead compared with the ciphertext-based calculations. In the initialization phase, each worker only needs to generate some random numbers and then perturb her sensory data. The computational complexity is $O(M)$ for each worker in this phase. In the iteration phase, each worker $u_k$ calculates the weight $w_k$ based on $\sum_{m=1}^{M} d(x_m^k, x_m)$, the perturbed weighted data $\{w_k x_m^k - \beta_m^k\}_{m=1}^{M}$ and the perturbed weight $w_k - \gamma_k$. The computational costs of these calculations are $O(M)$, $O(M)$ and $O(1)$ respectively in each iteration.

On the cloud side, we mainly consider the overhead introduced by the ciphertext-based calculations, which dominate

the overall computational cost when the key size is fixed. In each iteration, cloud $S_A$ has to conduct $O(KM)$ encryptions in order to encrypt the values $\{(\tilde{x}_m^k - x_m)^2\}_{k,m=1}^{K,M}$, and $O(KM)$ ciphertext multiplications and exponentiations to calculate $C_{conti}$ or $C_{cate}$. For cloud $S_B$, he needs to take $O(KM)$ encryptions to encrypt the random numbers $\{\alpha_m^k\}_{k,m=1}^{K,M}$ in the initialization phase and conduct decryption once in each iteration.

*2) Communication Overhead:* Since the proposed framework is used in MCS systems, in which the mobile devices usually have limited energy resources, we hope that each worker communicates with the clouds as little as possible. Thus, here we mainly analyze the amount of communication between the parties in $L$-PPTD. In addition to the analysis provided here, we also conduct real-world experiments to measure the communication overhead in section VI.

In $L$-PPTD, each worker needs to upload the perturbed data to cloud $S_A$ and the random numbers to cloud $S_B$, both of which are conducted only once during the whole truth discovery procedure. In every iteration, each worker first receives the summation $\sum_{k=1}^{K}\sum_{m=1}^{M} d(x_m^k, x_m)$ and the estimated object truths from cloud $S_B$, then uploads the perturbed weight and perturbed weighted data to cloud $S_A$. So the total number of communication times between each worker and the two cloud platforms is $2(t+1)$, where $t$ is the number of iterations. For the cloud platforms, cloud $S_B$ needs to send the ciphertexts and summations of the random numbers to cloud $S_A$ in the initialization phase. In each iteration, cloud $S_A$ sends the ciphertext $C_{conti}$ or $C_{cate}$ together with the estimated truths to cloud $S_B$. So the total number of communication times between cloud $S_A$ and cloud $S_B$ is $t+1$.

### D. Generalization

Although the logarithmic weight function is adopted in this paper, $L$-PPTD can also incorporate other types of weight function, such as the reciprocal function $w_k = {d_k}^{-p}$ and the affine function $w_k = 1 - pd_k$, where $d_k = \sum_{m=1}^{M} d(x_m^k, x_m)$ and $p$ is a parameter chosen based on the specific application scenarios. In $L$-PPTD, besides the summation $\sum_{k=1}^{K} d_k$ received from $S_B$, each worker $u_k$ can calculate the value $d_k$ by herself. So as long as the weight function $f$ (presented in section III-A) does not involve other information about all workers except the summation $\sum_{k=1}^{K} d_k$, it can be calculated on the worker side.

## V. $L^2$-PPTD Framework

Although little overhead is introduced on the worker side, each worker in $L$-PPTD framework is still involved in the calculation of her own weight and weighted data. To make the proposed framework more efficient in the scenarios where only the sensory data of each worker need to be protected, we propose an even more lightweight framework, called $L^2$-PPTD, in which the workers need not to be involved in the iterative procedure. Similar to $L$-PPTD, $L^2$-PPTD also contains two phases (i.e., *initialization phase* and *iteration phase*) as shown in Fig. 2.

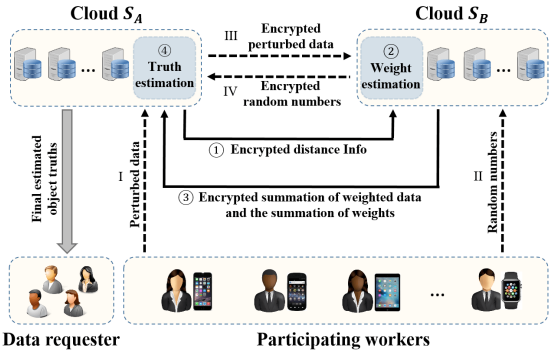



**Fig. 2:** The workflow of $L^2$-PPTD

### A. The Detailed Procedure of $L^2$-PPTD

In the $L^2$-PPTD framework, each worker only needs to take part in the initialization phase. Both the weight estimation and truth estimation are completed on the cloud side.

*1) Initialization Phase:* This phase is also conducted only once during the whole truth discovery procedure. Different from $L$-PPTD, both of the two cloud platforms $S_A$ and $S_B$ need to generate their own public keys and private keys. We use $(pk_A, sk_A)$ and $(pk_B, sk_B)$ to denote the key pairs of $S_A$ and $S_B$ respectively.

**Step I:** Each worker $u_k$ generates a random number $\alpha_m^k$ for each object $o_m$ (for categorical data, $\alpha_m^k$ is a vector in which each element is a random). Then she perturbs the sensory data $x_m^k$ as $\tilde{x}_m^k = x_m^k - \alpha_m^k$, and uploads all the perturbed data $\{\tilde{x}_m^k\}_{m=1}^{M}$ to cloud $S_A$.

**Step II:** Each worker $u_k$ uploads the random numbers $\{\alpha_m^k\}_{m=1}^{M}$ to cloud $S_B$.

**Step III:** Cloud $S_A$ first encrypts each perturbed sensory data $\tilde{x}_m^k$ with his own public key $pk_A$, then sends all the ciphertexts $\{E_A[\tilde{x}_m^k]\}_{k,m=1}^{K,M}$ ($E_A$ is the encryption function based on $pk_A$) to cloud $S_B$. For categorical data, $E_A[\tilde{x}_m^k]$ is a ciphertext vector in which each element is the ciphertext of the corresponding element in vector $\tilde{x}_m^k$.

**Step IV:** Cloud $S_B$ encrypts each random number $\alpha_m^k$ with his own public key $pk_B$, and sends all the ciphertexts $\{E_B[\alpha_m^k]\}_{k,m=1}^{K,M}$ ($E_B$ is the encryption function based on $pk_B$) to cloud $S_A$.

*2) Iteration Phase:* This phase also starts with the random initialization of the object truths on cloud $S_A$.

**Step ①:** For continuous data, cloud $S_A$ calculates the ciphertext $C_{conti}^k$ for each worker $u_k$ as

$$\begin{aligned} C_{conti}^k =& E_B[\sum_{m=1}^{M}(x_m^k - x_m)^2 - \sum_{m=1}^{M}(\alpha_m^k)^2] \\ =& \prod_{m=1}^{M}\{E_B[(\tilde{x}_m^k - x_m)^2] \cdot E_B[\alpha_m^k]^{2(\tilde{x}_m^k - x_m)}\}. \end{aligned} \tag{10}$$

For categorical data, $S_A$ calculates the ciphertext $C_{cate}^k$ for each worker $u_k$ as

$$\begin{aligned} C_{cate}^k =& E_B[\sum_{m=1}^{M}\sum_{i=1}^{l}(x_{mi}^k - x_{mi})^2 - \sum_{m=1}^{M}\sum_{i=1}^{l}(\alpha_{mi}^k)^2] \\ =& \prod_{m=1}^{M}\prod_{i=1}^{l}\{E_B[(\tilde{x}_{mi}^k - x_{mi})^2] \cdot E_B[\alpha_{mi}^k]^{2(\tilde{x}_{mi}^k - x_{mi})}\}. \end{aligned} \tag{11}$$

Then, all the ciphertexts $\{C^k_{conti}\}^K_{k=1}$ or $\{C^k_{cate}\}^K_{k=1}$ are sent to cloud $S_B$ (here the estimated object truths are not sent).

**Step ②:** After receiving the ciphertexts from cloud $S_A$, cloud $S_B$ decrypts them with his private key $sk_B$ and calculates the summation of distances (i.e., $\sum_{m=1}^{M} d(x^k_m, x_m)$) for each worker $u_k$ by adding the value $\sum_{m=1}^{M}(\alpha^k_m)^2$ or $\sum_{m=1}^{M}\sum_{i=1}^{l}(\alpha^k_{mi})^2$ to the decrypted data. Then, cloud $S_B$ estimates the weight of $u_k$ according to Equation (1).

**Step ③:** When the sensory data is continuous, based on the estimated weights, cloud $S_B$ first calculates the value $\sum_{k=1}^{K} w_k\alpha^k_m$ for each object $o_m$ and encrypts it with $S_A$'s public key $pk_A$ as $E_A[\sum_{k=1}^{K} w_k\alpha^k_m]$. Then, cloud $S_B$ calculates the encrypted summation of weighted data for each object $o_m$ as

$$\begin{aligned} E_A[\sum_{k=1}^{K} w_k x^k_m] =& E_A[\sum_{k=1}^{K} w_k(x^k_m - \alpha^k_m) + \sum_{k=1}^{K} w_k\alpha^k_m] \\ =& E_A[\sum_{k=1}^{K} w_k\tilde{x}^k_m] \cdot E_A[\sum_{k=1}^{K} w_k\alpha^k_m] \\ =& \prod_{k=1}^{K}\{E_A[\tilde{x}^k_m]^{w_k}\} \cdot E_A[\sum_{k=1}^{K} w_k\alpha^k_m]. \end{aligned} \tag{12}$$

When the sensory data is categorical, $E_A[\sum_{k=1}^{K} w_k x^k_m]$ is a ciphertext vector in which each element is calculated similarly with Equation (12). Then, the ciphertexts (ciphertext vectors for categorical data) of all objects together with the summation of all workers' weights (i.e., $\sum_{k=1}^{K} w_k$) are sent to cloud $S_A$.

**Step ④:** After receiving the data from cloud $S_B$, cloud $S_A$ decrypts the ciphertexts and gets the summation $\sum_{k=1}^{K} w_k x^k_m$ for each object $o_m$. Then, the truth of each object is estimated according to Equation (2).

Step ① $\sim$ ④ in this phase will also be iteratively conducted until the convergence criterion is satisfied. The final estimated truths are then sent to the data requester. In this framework, although the reliability information (i.e., weight) of each worker is known by cloud $S_B$, much less overhead will be introduced on the worker side since the workers only take part in the initialization phase.

### B. Security Analysis

**Theorem 2.** *Suppose there are at least two workers providing different sensory data for each object. If the parties (including the two cloud platforms) are semi-honest and there is no collusion among them, the sensory data of each worker will not be disclosed to any other party under $L^2$-PPTD framework.*

*Proof.* In $L^2$-PPTD, since each worker does not receive any information about other parties, we just need to prove the sensory data of each worker would not be disclosed to the cloud platforms and the data requester.

For cloud $S_A$, the values he knows include the ciphertexts $\{E_B[\alpha^k_m]\}^{K,M}_{k,m=1}$, $\{C^k_{conti}\}^K_{k=1}$ (or $\{C^k_{cate}\}^K_{k=1}$) and the plaintexts $\{x^k_m - \alpha^k_m\}^{K,M}_{k,m=1}$, $\{\sum_{k=1}^{K} w_k x^k_m\}^M_{m=1}$, $\sum_{k=1}^{K} w_k$, $\{x_m\}^M_{m=1}$. Without the private key $sk_B$, above ciphertexts cannot be decrypted by $S_A$. Since the weight of each worker estimated on cloud $S_B$ is not sent to cloud $S_A$, $S_A$ cannot infer the individual values $w_k$, $w_k x^k_m$ just based on the summations $\{\sum_{k=1}^{K} w_k x^k_m\}^M_{m=1}$ and $\sum_{k=1}^{K} w_k$. Additionally, as the two cloud platforms do not collude with each other, the sensory data of each worker will not be inferred by cloud $S_A$ from the values $\{x^k_m - \alpha^k_m\}^{K,M}_{k,m=1}$.

For cloud $S_B$, he knows the ciphertexts $\{E_A[x^k_m - \alpha^k_m]\}^{K,M}_{k,m=1}$, $\{E_A[\sum_{k=1}^{K} w_k x^k_m]\}^M_{m=1}$ and the plaintexts $\{\alpha^k_m\}^{K,M}_{k,m=1}$, $\{\sum_{m=1}^{M} d(x^k_m, x_m)\}^K_{k=1}$, $\{w_k\}^K_{k=1}$. Since the object truths $\{x_m\}^M_{m=1}$ estimated on cloud $S_A$ are not sent to cloud $S_B$, $S_B$ can not learn the individual sensory data $x^k_m$ from the values $\{\sum_{m=1}^{M} d(x^k_m, x_m)\}^K_{k=1}$. Additionally, $S_B$ cannot decrypt the ciphertexts without $S_A$'s private key. So each worker's sensory data will not be known to cloud $S_B$. Additionally, similar to $L$-PPTD, the data requester can only know the final aggregated results $\{x_m\}^M_{m=1}$ in $L^2$-PPTD.

In summary, the sensory data of each worker will not be disclosed to other parties under the $L^2$-PPTD framework. □

### C. Efficiency Analysis

*1) Computational Complexity:* The participating workers in $L^2$-PPTD are only involved in the initialization phase and the computational cost (based on plaintexts) is $O(M)$ for each worker. For cloud $S_A$ and cloud $S_B$, both of them are involved in the initialization phase and iteration phase. In the initialization phase, they both have to take $O(KM)$ encryptions. In each iteration, cloud $S_A$ has to conduct $O(KM)$ encryptions, $O(M)$ decryptions, $O(KM)$ ciphertext multiplications and exponentiations. On the other hand, Cloud $S_B$ needs to conduct $O(K)$ decryptions, $O(M)$ encryptions, $O(KM)$ ciphertext multiplications and $O(KM)$ ciphertext exponentiations to calculate the weights and ciphertexts $\{E_A[\sum_{k=1}^{K} w_k x^k_m]\}^M_{m=1}$.

*2) Communication Overhead:* In $L^2$-PPTD, since the participating workers are only involved in the initialization phase, the number of communication times between each worker and the cloud platforms is only 2 during the whole truth discovery procedure. As for cloud $S_A$ and $S_B$, both of them need to send data once to each other in the initialization phase and each iteration, so the communication overhead between the two clouds is $2(t+1)$, where $t$ is the number of iterations.

### D. Generalization

Since the weight of each worker in $L^2$-PPTD is estimated based on plaintexts on cloud $S_B$, which has all the values $\{\sum_{m=1}^{M} d(x^k_m, x_m)\}^K_{k=1}$ needed to estimate the workers' weights, any weight function $f$ can be used in this framework.

## VI. Performance Evaluation

In this section, we evaluate the performance of the proposed framework on real world crowdsensing systems. Both continuous and categorical sensory data are considered here. The cloud platforms in this experiment are emulated by two Intel(R) Core(TM) 3.4GHz computers running Ubuntu 14.04, with 16GB RAM. We use Nexus 5 Android phones as the mobile sensing devices. The frameworks are implemented with the Paillier encryption tool[1] and the key size is set as 512 bits. Additionally, we use the rounding parameter $R = 10^7$ to round

[1] http://www.cs.utdallas.edu/dspl/cgi-bin/pailliertoolbox/

the fractional data. The baseline methods are the original truth discovery method CRH [4, 5] and the state-of-the-art privacy preserving truth discovery framework PPTD [10].

### *A. Experiment on Crowdsourced Indoor Floorplan Construction System*

We first evaluate the performance of the proposed framework on one real-world crowdsensing application, i.e., crowdsourced indoor floorplan construction [1, 13]. In such crowdsensing systems, although the indoor floorplan can be automatically constructed from the sensory data (e.g., compass, accelerometer, gyroscope) collected by smartphone users, the sensor readings may disclose a user's private personal information. For the sake of illustration, here we only focus on estimating the distance (continuous data) between two particular location points in a hallway. 10 smartphone users are employed as the participating workers and 20 hallway segments in a building are selected as the objects. We develop an Android App that can estimate the walking distances of a smartphone user through multiplying the user's step size by step count inferred using the in-phone accelerometer. The ground truths are obtained by measuring these segments manually.

*1) Accuracy:* We first compare the accuracy of the final estimated object truths between the proposed method and the baseline approach. In this experiment, the root of mean squared error (RMSE) is used to measure the deviation between the estimated results and the ground truths. Here the number of objects is fixed as 20 and we vary the number of workers from 3 to 10. The results are shown in Fig. 3. As seen, our proposed frameworks have almost the same estimation accuracy as CRH regardless of the number of workers.

*2) Convergence:* In order to evaluate the convergence of the proposed algorithms, we measure the distance between the estimated object truths in two consecutive iterations using a *convergence value* defined as $||\mathbf{x}^t - \mathbf{x}^{t-1}||_2$, where $\mathbf{x}^t$ $(t \geq 1)$ is the vector of estimated truths in iteration $t$ (the values in $\mathbf{x}^0$ are randomly initialized). The convergence values of $L$-PPTD are shown in Fig. 4. Here we calculate the convergence values with 5 different initial truth vectors (i.e., $\mathbf{x}^0$). As we can see, the $L$-PPTD algorithm converges quickly in just a few iterations. $L^2$-PPTD has similar convergence pattern.

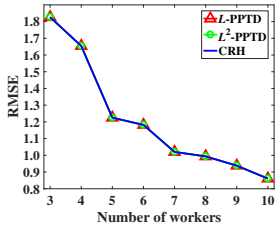



**Fig. 3:** Accuracy for the indoor floorplan construction system

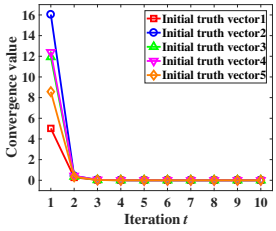



**Fig. 4:** Convergence for the indoor floorplan construction system

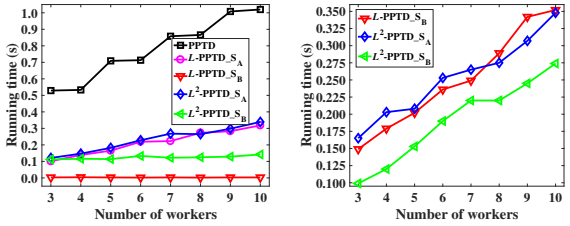
*3) Computational cost:* Here we evaluate the running time of the proposed frameworks on both the worker and the cloud sides. The results are compared with that of the baseline method PPTD.

Table I shows the running time on one worker's smartphone while the number of objects is varying from 4 to 20. Since the workers are not involved in the iteration phase of $L^2$-PPTD, there is no result for that phase in Table I. From this table, we can see the running time of the proposed frameworks is much less than that of PPTD. The reason is that all the computations on the worker side are based on plaintexts in our frameworks while the workers in PPTD need to do some encryptions. Additionally, the results in this table also show that $L^2$-PPTD is more lightweight than $L$-PPTD.

**TABLE I:** Running time on smartphone for the indoor floorplan construction system

| Number of objects | | 4 | 8 | 12 | 16 | 20 |
|---|---|---|---|---|---|---|
| PPTD/iteration ($\times 10^{-2}s$) | | 1.3 | 2.0 | 2.8 | 3.4 | 5.1 |
| $L$-PPTD ($\times 10^{-6}s$) | Initialization phase | 3.0 | 4.6 | 6.0 | 6.7 | 9.3 |
| | Each iteration | 0.9 | 1.1 | 1.5 | 2.1 | 2.6 |
| $L^2$-PPTD ($\times 10^{-6}s$) | | 2.6 | 3.0 | 3.8 | 4.1 | 4.9 |

Figure 5 shows the running time on the cloud platforms with the worker number varying from 3 to 10. The results in one iteration and the initialization phase are provided in Fig. 5a and Fig. 5b, respectively. Since cloud $S_A$ in $L$-PPTD does not have to do any computation in the initialization phase, there is no result for $S_A$ in Fig. 5b. From Fig. 5a, we can see the running time of our frameworks on both cloud $S_A$ and $S_B$ are less than that of PPTD. This is mainly because of the threshold-based decryption scheme used in PPTD. Although the initialization phase is involved in our frameworks, it does not introduce much computational cost, which can be seen from Fig. 5b.




**(a)** The iteration phase **(b)** The initialization phase

**Fig. 5:** Running time on cloud platforms for the indoor floorplan construction system

*4) Communication overhead:* Here we evaluate the data size each worker needs to send or receive in the proposed frameworks. In Table II, we provide a comparison between the proposed frameworks and PPTD. As can be seen, the data needed to be transmitted in each iteration of $L$-PPTD is much less than that of PPTD. The reason is that in the proposed frameworks, workers do not need to send or receive any ciphertext, which is usually much larger than plaintext. Since each worker is not involved in the iteration phase of $L^2$-PPTD and the initialization phases of the two frameworks are only needed to be conducted once, the total communication overhead of our scheme are also much less than that of PPTD. Additionally, the results further verify the conclusion that $L^2$-PPTD is more lightweight than $L$-PPTD since there is no communication overhead on the smartphone in each iteration of $L^2$-PPTD.

**TABLE II:** Communication overhead on smartphone for the indoor floor plan construction system (KB)

| Number of objects | | 4 | 8 | 12 | 16 | 20 |
|---|---|---|---|---|---|---|
| PPTD/iteration | | 1.83 | 3.15 | 4.37 | 5.58 | 6.90 |
| $L$-PPTD | Initialization phase | 0.05 | 0.09 | 0.14 | 0.21 | 0.25 |
| | Each iteration | 0.10 | 0.18 | 0.26 | 0.33 | 0.41 |
| $L^2$-PPTD | | 0.04 | 0.09 | 0.14 | 0.19 | 0.24 |

### *B. Experiment on Crowd Wisdom System*

In this experiment, we evaluate the proposed frameworks on crowd wisdom system in which the sensory data are categorical. Each smartphone user in this system receives questions from the cloud to which she uploads her answers. The cloud infers the true answer for each question through aggregating the answers from smartphone users. However, some users may concern that their personal information may be inferred from their answers. Here we implement the proposed frameworks on this system. 100 smartphone users are employed as the participating workers and 54 questions (each has 4 candidate answers) are used as the objects.

*1) Accuracy and Convergence:* In this part, we measure the accuracy of the proposed frameworks with *Error Rate*, which is defined as the percentage of mismatched values between the estimated results and the ground truths. Since some objects (i.e., questions) are not answered by all the workers in this experiment, we vary the average number of workers that answer each question and then calculate the Error Rate. The results of the two frameworks and CRH are shown in Fig. 6. We can see that the Error Rates of the proposed frameworks are the same with that of CRH in all scenarios, which verifies the accuracy of our proposed frameworks for categorical data. Additionally, the convergence value defined before is also adopted here to measure the convergence. We find both our frameworks and CRH converge within two iterations.

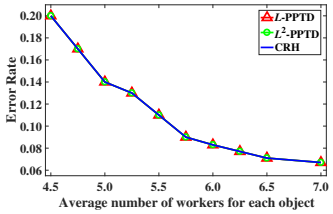


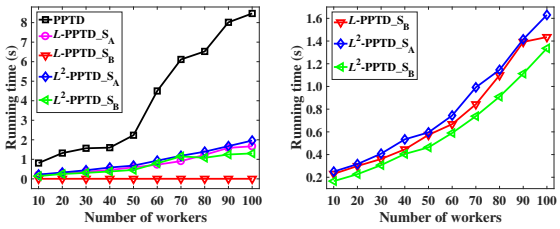

**Fig. 6:** Accuracy for the wisdom system

*2) Computational cost:* Here we also evaluate the running time of the proposed frameworks on both smartphone and cloud platforms. In this experiment, we record the running time on smartphone while the number of objects is varying from 2 to 14, which is the maximum number of questions answered by a single worker. The results are shown in Table III, from which we can see the computational cost of our frameworks on smartphone is much less than that of PPTD.

**TABLE III:** Running time on smartphone for the crowd wisdom system

| Number of objects | | 2 | 5 | 8 | 11 | 14 |
|---|---|---|---|---|---|---|
| PPTD/iteration ($\times 10^{-2}s$) | | 1.8 | 3.9 | 5.8 | 8.0 | 12.7 |
| $L$-PPTD ($\times 10^{-6}s$) | Initialization phase | 3.5 | 10.2 | 12.6 | 17.6 | 23.3 |
| | Each iteration | 1.3 | 2.9 | 7.3 | 6.2 | 8.2 |
| $L^2$-PPTD ($\times 10^{-6}s$) | | 1.8 | 4.6 | 7.1 | 9.4 | 11.6 |

For the cloud platforms, Fig. 7a and Fig. 7b show the running time in each iteration and the initialization phase respectively. As seen, the running time of our frameworks on both cloud $S_A$ and $S_B$ is less than that of PPTD.




**(a)** The iteration phase **(b)** The initialization phase

**Fig. 7:** Running time on cloud platforms for the crowd wisdom system

*3) Communication overhead:* The size of data needed to be transmitted on each smartphone is shown in Table IV, from which we can see the communication overhead on smartphone in each iteration of $L$-PPTD is much less than that of PPTD. Additionally, the data transmitted by each worker in $L^2$-PPTD is less than that of both $L$-PPTD and PPTD.

**TABLE IV:** Communication overhead on smartphone for the crowd wisdom system (KB)

| Number of objects | | 2 | 5 | 8 | 11 | 14 |
|---|---|---|---|---|---|---|
| PPTD/iteration | | 3.1 | 6.9 | 10.6 | 14.4 | 18.1 |
| $L$-PPTD | Initialization phase | 0.09 | 0.24 | 0.38 | 0.52 | 0.66 |
| | Each iteration | 0.11 | 0.21 | 0.32 | 0.43 | 0.51 |
| $L^2$-PPTD | | 0.07 | 0.18 | 0.28 | 0.39 | 0.49 |

### *C. Experiment on Simulated MCS system*

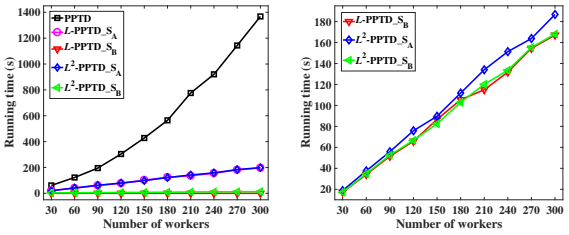
In order to evaluate the scalability and efficiency of the proposed frameworks, we conduct further experiments on a simulated MCS system, in which there are 300 participating workers and 1000 objects. We generate the sensory data of workers through adding Gaussian noise with different intensities to the ground truths. Table V shows the running time of the frameworks on smartphone with varying object number ranging from 200 to 1000. From the table, we can see our proposed frameworks can keep high efficiency even when the number of objects is very large. Especially for $L^2$-PPTD, when the number of objects is 1000, the running time of $L^2$-PPTD on the smartphone is only $2.15 \times 10^{-4}s$ during the whole truth discovery procedure.

**TABLE V:** Running time on smartphone for the simulated MCS system

| Number of objects | | 200 | 400 | 600 | 800 | 1000 |
|---|---|---|---|---|---|---|
| PPTD/iteration ($s$) | | 0.43 | 0.83 | 1.47 | 2.29 | 2.88 |
| $L$-PPTD ($\times 10^{-4}s$) | Initialization phase | 0.78 | 1.62 | 2.40 | 3.18 | 3.95 |
| | Each iteration | 0.24 | 0.51 | 0.64 | 0.86 | 1.14 |
| $L^2$-PPTD ($\times 10^{-4}s$) | | 0.43 | 0.91 | 1.32 | 1.69 | 2.15 |

Next, we fix the number of objects as 500 and change the number of workers from 30 to 300 in order to evaluate the computational cost of the cloud platforms. The results shown in Fig. 8 further verify that the two frameworks in this paper are more efficient on the cloud platforms than PPTD.

Table VI reports the communication overhead on the smartphone while the number of objects observed by each worker




**(a)** The iteration phase **(b)** The initialization phase

**Fig. 8:** Running time on cloud platforms for the simulated MCS system

is changing from 400 to 1000. When the number of objects is 1000, the communication overhead in each iteration of $L$-PPTD is only 19.6KB while that in PPTD is 314.2KB. Additionally, the communication overhead of $L^2$-PPTD is only 13.5KB during the whole truth discovery procedure when the number of objects is 1000, which means $L^2$-PPTD is a more lightweight scheme for participating workers.

**TABLE VI:** Communication overhead on smartphone for the simulated MCS system (KB)

| Number of objects | | 400 | 600 | 800 | 1000 |
|---|---|---|---|---|---|
| PPTD/iteration | | 126.1 | 188.8 | 251.6 | 314.2 |
| $L$-PPTD | Initialization phase | 6.4 | 9.9 | 13.1 | 16.4 |
| | Each iteration | 7.7 | 11.6 | 15.5 | 19.6 |
| $L^2$-PPTD | | 5.1 | 8.1 | 10.8 | 13.5 |

## VII. Related Work

As an effective technique to extract truthful information from the unreliable data in MCS systems, truth discovery has recently been widely studied [4–9]. However, these schemes do not take actions to protect the participating workers' privacy, which is a key concern in many MCS applications [14]. To address the privacy concern, a recent paper [10] presents a mechanism called PPTD that can protect the workers' privacy during the truth discovery procedure. Although strong privacy and high accuracy can be guaranteed in PPTD, significant computation and communication overhead will be introduced on both the worker and cloud sides.

With respect to the MCS systems, the privacy-preserving problem is also studied in paper [15–17]. However, the problem settings targeted in these papers are different from that of truth discovery aiming at jointly infer both worker weights and object truths. In addition, although a cryptography based scheme is proposed in paper [18] to protect the privacy of each worker in crowdsourcing applications, it mainly focuses on categorical data and is built upon the threshold-based cryptosystem, which would inevitably incur a large amount of overhead. Finally, Catalano et al. propose a two-server based protocol [19] for the delegation of computation on encrypted data. The frameworks presented in this paper, though also involving two cloud servers, are designed for different problem settings and application scenarios.

## VIII. Conclusion

In this paper, we propose a lightweight privacy-preserving truth discovery ($L$-PPTD) framework, which is implemented by involving two non-colluding cloud platforms and adopting additively homomorphic cryptosystem. This framework can not only protect the sensory data and reliability information of each participating worker but also substantially reduce the overhead on the worker side. Additionally, to further reduce each worker's overhead in the scenarios where only the sensory data need to be protected, a more lightweight truth discovery ($L^2$-PPTD) framework is also developed.

## Acknowledgment

This work was sponsored in part by US National Science Foundation under grant CNS-1566374, the National Natural Science Foundation of China under grant No. 61502443 and China Postdoctoral Science Foundation under grant No. 2015M570545.

## References

[1] R. Gao, M. Zhao, T. Ye, F. Ye, Y. Wang, K. Bian, T. Wang, and X. Li, "Jigsaw: Indoor floor plan reconstruction via mobile crowdsensing," in *Mobicom*, 2014.

[2] F. Saremi, O. Fatemieh, H. Ahmadi, H. Wang, T. Abdelzaher, R. Ganti, H. Liu, S. Hu, S. Li, and L. Su, "Experiences with greengpsfuel-efficient navigation using participatory sensing," *IEEE Transactions on Mobile Computing (TMC)*, 2016.

[3] S. Hu, L. Su, H. Liu, H. Wang, and T. F. Abdelzaher, "Smartroad: Smartphone-based crowd sensing for traffic regulator detection and identification," *ACM Transactions on Sensor Networks (TOSN)*, 2015.

[4] 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 *SIGMOD*, 2014.

[5] Y. Li, Q. Li, J. Gao, L. Su, B. Zhao, W. Fan, and J. Han, "Conflicts to harmony: A framework for resolving conflicts in heterogeneous data by truth discovery," *IEEE transactions on Knowledge and Data Engineering (TKDE)*, 2016.

[6] L. Su, Q. Li, S. Hu, S. Wang, J. Gao, H. Liu, T. F. Abdelzaher, J. Han, X. Liu, Y. Gao *et al.*, "Generalized decision aggregation in distributed sensing systems," in *RTSS*, 2014.

[7] C. Meng, W. Jiang, Y. Li, J. Gao, L. Su, H. Ding, and Y. Cheng, "Truth discovery on crowd sensing of correlated entities," in *SenSys*, 2015.

[8] Y. Li, J. Gao, C. Meng, Q. Li, L. Su, B. Zhao, W. Fan, and J. Han, "A survey on truth discovery," *SIGKDD Explorations*, 2015.

[9] S. Yao, M. T. Amin, L. Su, S. Hu, S. Li, S. Wang, Y. Zhao, T. Abdelzaher, L. Kaplan, C. Aggarwal *et al.*, "Recursive ground truth estimator for social data streams," in *IPSN*, 2016.

[10] C. Miao, W. Jiang, L. Su, Y. Li, S. Guo, Z. Qin, H. Xiao, J. Gao, and K. Ren, "Cloud-enabled privacy-preserving truth discovery in crowd sensing systems," in *SenSys*, 2015.

[11] I. Damgård and M. Jurik, "A generalisation, a simpli. cation and some applications of paillier's probabilistic public-key system," in *International Workshop on Public Key Cryptography*, 2001.

[12] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *EUROCRYPT*, 1999.

[13] S. Chen, M. Li, K. Ren, X. Fu, and C. Qiao, "Rise of the indoor crowd: Reconstruction of building interior view via mobile crowdsourcing," in *SenSys*, 2015.

[14] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges." *IEEE Communications Magazine*, 2011.

[15] R. K. Ganti, N. Pham, Y.-E. Tsai, and T. F. Abdelzaher, "Poolview: stream privacy for grassroots participatory sensing," in *SenSys*, 2008.

[16] F. Zhang, L. He, W. He, and X. Liu, "Data perturbation with state-dependent noise for participatory sensing," in *INFOCOM*, 2012.

[17] R. Zhang, J. Shi, Y. Zhang, and C. Zhang, "Verifiable privacy-preserving aggregation in people-centric urban sensing systems," *IEEE Journal on Selected Areas in Communications*, 2013.

[18] H. Kajino, H. Arai, and H. Kashima, "Preserving worker privacy in crowdsourcing," *Data Mining and Knowledge Discovery*, 2014.

[19] D. Catalano and D. Fiore, "Using linearly-homomorphic encryption to evaluate degree-2 functions on encrypted data," in *CCS*, 2015.