World Wide Web (2019) 22:2017–2040
https://doi.org/10.1007/s11280-018-0638-2

Towards secure and truthful task assignment in spatial crowdsourcing

Dongjun Zhai[1] · Yue Sun[1] · An Liu[1] · Zhixu Li[1] · Guanfeng Liu[2] · Lei Zhao[1] · Kai Zheng[3]

Received: 15 April 2018 / Revised: 12 July 2018 / Accepted: 7 September 2018 /
Published online: 18 September 2018

Abstract
The ubiquity of mobile device and wireless networks flourishes the market of spatial crowdsourcing, in which location constrained tasks are sent to workers and expected to be performed in some designated locations. To obtain a global optimal task assignment scheme, the platform usually needs to collect location information of all workers. During this process, there is a significant security concern, that is, the platform may not be trustworthy, so it brings about a threat to workers location privacy. In this paper, to tackle the privacy-preserving task assignment problem, we propose a privacy-preserving reverse auction based assignment model which consists of two key parts. In the first part, we generalize private location to travel cost and protect it by an anonymity based data aggregation protocol. In the second part, we propose a reverse auction task assignment algorithm, which is a truthful incentive mechanism, to encourage workers to offer authentic data. We theoretically show that the proposed model is secure against semi-honest adversaries. Experimental results show that our model is efficient and can scale to real SC applications.

Keywords Privacy-preserving · Spatial crowdsourcing · Task assignment · Reverse auction

1 Introduction

With the rapid development of GPS-equipped mobile devices and wireless networks, Spatial Crowdsourcing (SC) has become an emerging platform to tackle human intrinsic tasks

This article belongs to the Topical Collection: *Special Issue on Web Information Systems Engineering 2017*
Guest Editors: Lu Chen and Yunjun Gao

✉ An Liu
anliu@suda.edu.cn

[1] School of Computer Science and Technology & Institute of Artificial Intelligence, Soochow University, Suzhou, China

[2] Department of Computing, Macquarie University, Sydney, NSW 2122, Australia

[3] Big Data Research Center, University of Electronic Science and Technology of China, Chengdu, China

[3]. In a traditional SC model, location-constrained tasks are sent to workers (e.g. users with smart phones) and the workers are expected to physically travel to some locations to perform corresponding tasks. As the embodiment of *Wisdom of Crowd*, SC has applications in numerous domains such as transportation (e.g. Uber), handyman service (e.g. TaskRabbit), collaborative mapping (e.g. OpenStreetMap) and food delivery (e.g. Eleme).

Based on the publishing modes of spatial tasks, SC can be divided into two different categories [3]: *worker selected tasks* (WST) and *server assigned tasks* (SAT). Due to the global view of generating task assignment, SAT is more popular and has more application scenarios than WST. The task assignment problem in SAT mode can be viewed as task-worker bipartite graph matching problem, in which each task and worker node may have multiple matching edges. Specifically, each task can be assigned to more than one workers and each worker can hold more than one tasks at a time. So far, existing approaches mainly focus on optimizing task assignments problem. Given some temporal, budget or capacity constraints, these works try to maximize, for example, the overall number of assigned tasks [16], the number of worker's self-selected tasks [9], the budget-minus-cost score of task assignment [6], the acceptance of workers [46] and so on.

Compared with conventional crowdsourcing, SC has some special privacy concerns, as the SC platform needs to collect all workers' private locations to perform task assignment. Protecting workers' private information can encourage them to participate SC, so this is an important problem that needs to be solved in spatial crowdsourcing. Existing privacy-protection methods can be divided into three categories: spatial cloaking [15, 22, 36], differential privacy [33, 38], and homomorphic encryption [23, 25]. Methods based on spatial cloaking and differential privacy typically cannot obtain the accurate task assignment result due to the decreased data utility, which results from the noise introduced by these mechanisms. Homomorphic encryption based methods can generate accurate task assignment, but they suffer from computation time issue due to the expensive operations over ciphertext and thus cannot scale to large SC applications.

To overcome the above weaknesses of existing methods, we consider a novel SAT mode in which workers can set their preferences for tasks and the SC platform makes task assignment based on not only its own optimization goal but also workers' preferences. If a worker prefers some tasks, s/he sends his/her *travel cost* instead of locations to the platform. This setting gives more rights to workers as their preferences are taken into account during task assignment. Further, it protects location privacy to a certain extent by generalizing a worker's exact location to the circumference of a circle whose center is the task location and its radius is proportional to the travel cost. However, the untrustworthy platform can still infer a worker's location approximately by considering the intersections of these circles. To deal with this, we propose an Anonymity-based Data Aggregation (ADA) protocol, which utilizes a bit-wise XOR homomorphic cipher [44] and oblivious transfer[11]. On the other hand, the accuracy of task assignment relies on the authenticity of the travel cost claimed by workers. Hence, we also propose a Reverse Auction based Task Assignment (RATA) algorithm, which is a truthful incentive mechanism, to stimulate workers to provide authentic travel cost.

The main contributions of our work can be summarized as follows:

(1) We consider a novel task assignment mode where workers report their travel cost instead of their exact locations to the SC platform. This not only gives more rights to workers but also can protect location privacy to a certain extent.
(2) We design an anonymity based data aggregation protocol for task assignment. This protocol can protect worker location privacy in an k-anonymity manner, and its security is formally proved.

(3) Compared with our conference version [31], we propose a new reverse auction based task assignment algorithm to ensure the authenticity of reported travel cost and formally show its correctness.
(4) We analyze the complexity of our proposed method and present the performance evaluation. Both theoretical and empirical results show that our method is not only secure but also efficient.

The rest of the paper is organized as follows. Section 2 summarizes the preliminary technology and knowledge knowledge we used in this paper. In Section 3, we presents the model and problem statement. In Section 4, we clarify our privacy-preserving reverse auction based assignment model from two aspects: anonymity-based data aggregation protocol and reverse auction based task assignment algorithm. Then, the theoretical analysis is presented in Section 5. Further, we evaluate the performances of our model in Section 6. In the end, we review some related works in Section 7 and conclude our work in the Section 8.

2 Preliminary

2.1 Security model

In this paper, we assume all parties involved in spatial crowdsourcing are *semi-honest* (also known as *honest-but-curious*). In other words, each entity will follow a protocol exactly, showing the honest aspect. On the other hand, they will also try to learn as much as possible about other's private data, showing the curious aspect. The semi-honest model is reasonable since each entity is generally willing to follow and accomplish the protocol so as to benefit from the crowdsourcing system.

The security of each worker's private data is defined in an k-anonymity manner. If an adversary can only learn that a private data comes from one of k workers, we say this data has a privacy level of k. Obviously, the larger k is, the stronger privacy is. Formally, we use *computational indistinguishability* to model the anonymity of a worker's data and define the security of a protocol under *semi-honest* model as follows:

Definition 1 (**Security under the Semi-honest Model**) Let the view of the SC platform in the execution of a protocol, denoted as $VIEW$, is a triple (m, r, x), where m represents the received data, r represents coin flips, and x represents the output of received data. For any worker w_i and w_j in the worker set W, their data can be denoted as m_i and m_j. Then this protocol is secure under semi-honest model if

$$VIEW((..., m_i, ..., m_j, ...), r, x) \equiv VIEW((..., m_j, ..., m_i, ...), r, x), \tag{1}$$

where $\equiv$ denotes computationally indistinguishability of two random variable ensembles and $(..., m_i, ..., m_j, ...)$ is a random permutation of $(..., m_j, ..., m_i, ...)$.

The above definition states that if we switch any two workers' data, the platform cannot efficiently notice any difference.

2.2 Bitwise XOR homomorphic cipher

Bitwise XOR homomorphic cipher system [44] makes it possible that plaintexts can be directly derived by performing bitwise XOR operations on the ciphertexts. The basic idea

of this cipher system is that the result of any data m XOR itself is zero, that is, $m \oplus m = 0$.

This cipher system can be set up as follows. Assume that there exist n independent string $\{S|s_i \in \{0,1\}^l\}(i = 0, ..., n-1)$ and n pieces of data $\{D|d_i \in \{0,1\}^l\}(i = 1, ..., n)$ that need to be encrypted. Each piece of data d_i will be distributed by two strings s_{i-1} and $s_{i \mod n}$ and be encrypted by performing bitwise XOR operation on itself with these two strings. Then, it is obvious that the bitwise XOR of all ciphertexts equals to the bitwise XOR of plaintexts.

2.3 Oblivious Transfer

Oblivious Transfer (OT) is a cryptographic primitive in which a sender transfers one of the potentially many pieces of information to a receiver, but it remains oblivious as to what piece (if any) has been transferred. The basic form of OT which is called 1-out-of-2 OT and denoted as OT_2^1, was developed by Shimon Even et al. [11]. A 1-out-of-n OT (denoted as OT_n^1) can be defined as a natural generalization of an OT_2^1. Specifically, a sender has n messages, and the receiver has an index i. The receiver wishes to receive the i-th message, without the sender learning i. Meanwhile, the sender wants to ensure that the receiver receives only one of the n messages. In this paper, we use OT_n^1 protocol as the building block of our model. To learn more about it, please refer to [27].

3 Problem definition

A typical spatial crowdsourcing system includes a platform, which receives the spatial tasks from crowdsourcing service requesters and assigns spatial tasks to suitable crowd workers to perform. The task and worker are defined as follows:

Definition 2 (**Spatial task**) A spatial task (task for short) is denoted by $t_j = \langle l_{t_j}, a_{t_j}, d_{t_j}, v_{t_j} \rangle$, where l_{t_j} is the location in a 2D space where task t_j needs to be performed, a_{t_j} and d_{t_j} make up a range time during which the task is valid and v_{t_j} is the value to the platform, i.e. the budget.

Definition 3 (**Crowd worker**) A crowd worker (worker for short) is denoted by $w_i = \langle l_{w_i}, a_{w_i}, d_{w_i}, c_{w_i} \rangle$, arrives at the platform with initial location l_{w_i} at time a_{w_i} and performs several tasks before the deadline d_{w_i}. In addition, capacity c_{w_i} is the maximum number of tasks that worker w_i intends to finish.

If a task t_j is allocated to a worker w_i, the worker needs some *travel cost* to move to location l_{t_j} and expects to attain some *payment* from the platform after finishing the task. The travel cost and payment are defined as follows:

Definition 4 (**Travel cost**) The travel cost of w_i with respect to t_j is defined to be proportional to the distance from l_{w_i} to l_{t_j}, which can be calculated as follows:

$$c_i^j = C * dist(l_{w_i}, l_{t_j}), \tag{2}$$

where C is the unit cost per mile and $dist$ is a distance calculation function (e.g., Euclidean distance).

Definition 5 (**Payment**) The payment of w_i with respect to t_j is the monetary cost that the platform has to pay to w_i for performing t_j. It should be larger than c_i^j to motivate workers to participate crowdsourcing, that is

$$p_i^j \geq c_i^j. \tag{3}$$

For the platform, its profit can be regarded as the difference between the sum of all tasks' values and the sum of payments to workers, that is: $\sum_{t_j \in S} v_{t_j} - \sum_{(w_i,t_j)\in S} p_i^j$, where S is the final task assignment schema (i.e., worker-and-task assignment pairs). For w_i, his/her profit u_i^j is simply the difference between the monetary s/he obtains from the platform and his/her travel cost: $p_i^j - c_i^j$.

In this paper, we want to maximize the total profit of the platform and all workers, denoted as social welfare, whose definition is as follows:

Definition 6 (**Social welfare**) Let S be a task assignment schema, that is, a set of worker-and-task assignment pairs, the social welfare $\phi(S)$ is calculated as follows:

$$\phi(S) = \sum_{t_j \in S} v_{t_j} - \sum_{(w_i,t_j)\in S} p_i^j + \sum_{(w_i,t_j)\in S} (p_i^j - c_i^j) = \sum_{t_j \in S} v_{t_j} - \sum_{(w_i,t_j)\in S} c_i^j. \tag{4}$$

Now, we are ready to give the definition of privacy-preserving task assignment problem in spatial crowdsourcing.

Definition 7 (**Privacy-preserving task assignment problem**) Given a set of workers and a set of tasks, the privacy-preserving task assignment problem is to generate a task assignment schema S such that:

(1) each worker w_i is assigned to no more than c_{w_i} tasks;
(2) the total payment of the task t_j does not exceed its budget v_{t_j};
(3) each worker-and-task assignment pair satisfies the time range requirement (i.e., $(a_{t_j}, d_{t_j}) \cap (a_{w_i}, d_{w_j}) \neq \emptyset$);
(4) each worker's private information is indistinguishable from $k-1$ other workers;
(5) the social welfare $\phi(S)$ is maximized.

The above problem can be formalized as follows:

$$Maximize: \quad \phi(S) = \sum_{t_j \in S} v_{t_j} - \sum_{(w_i,t_j)\in S} c_i^j \tag{5}$$

$$Subject\ to: \quad \sum_{(w_i,t_j)\in S} X_i^j \leq c_{w_i},\ X_i^j \in \{0,1\} \tag{6}$$

$$\sum_{(w_i,t_j)\in S} b_i^j \leq v_{t_j} \tag{7}$$

$$\forall_{(w_i,t_j)\in S},\ (a_{t_j}, d_{t_j}) \cap (a_{w_i}, d_{w_j}) \neq \emptyset \tag{8}$$

$$Security: \quad (1)\ holds. \tag{9}$$

4 Privacy-preserving reverse auction based task assignment

4.1 System overview

As mentioned earlier, workers in our model are more active: they can choose their interested tasks, compute travel costs and send these data to the platform. This brings the following advantages in spatial crowdsourcing:

(1) workers' preferences are taken into consideration, avoiding the situation in which the platform allocates tasks to workers by force;
(2) the workers do not need to report their exact locations to the platform, which protects their location privacy to a certain extent;
(3) the travel costs submitted by workers determine the potential matching edge in the worker-and-task bipartite graph matching problem, thus reducing the computation overhead.

However, when a worker submits travel costs for different tasks, the platform can infer his/her location approximately by intersecting these circumferences. Therefore, in Section 4.2, we design an Anonymity based Data Aggregation (ADA) protocol to delink the data from its sources and generalize the worker location to multiple circumferences, guaranteeing the security of workers' location privacy further. On the other hand, workers are selfish and may submit a high travel cost to attain more payment. To solve this problem, in Section 4.3, we propose a Reverse Auction based Task Assignment (RATA) algorithm to motivate workers to send truthful travel cost.

As shown in Figure 1, our privacy-preserving task assignment model includes three parties: the SC platform, workers and an auxiliary agent, who provides some cryptographic services such as the generation and the distribution of keys. In the beginning, the platform poses a set of tasks to all online workers. Then the agent establishes a bitwise XOR cipher system by generating and distributing secret keys to all workers. Afterwards, each worker

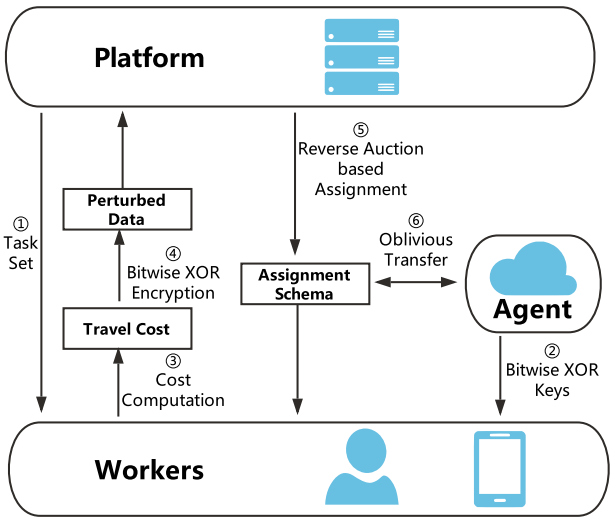


Figure 1 System model

can figure out the travel costs from his/her location to task locations and encrypt this origin data (travel costs) into perturbed data via his/her bitwise XOR secret key. After receiving all workers' perturbed data, the platform produces a task assignment schema via a reverse auction based task assignment algorithm. However, the platform only learns that some workers are selected but does not know their IDs. Thus, the platform then performs oblivious transfer with the agent who holds workers' IDs to allocate tasks to corresponding workers. Table 1 summarizes the main notations in this paper.

4.2 Anonymity-based data aggregation (ADA)

In this subsection, we present an anonymity based data aggregation protocol. Without loss of generality, we only consider the situation where one task is announced on the platform and a set of workers intend to cooperate for the task. However, the protocol can be easily extended to multiple tasks vs. multiple workers circumstance. To be specific, this protocol can be decomposed into four subprocesses.

Key generation and distribution phase Firstly, the agent generates n strings $\{S|S_i \in \{0,1\}^l\}$ uniformly and independently. Secondly, the agent performs a random permutation of workers' ID sequence I to get a new position sequence Pos. Then, the agent sends key pairs $\langle s_i^a = S_{i-1}, s_i^b = S_{i \mod n}, Pos_i \rangle$ to worker w_i.

Data computation and encryption phase Each worker computes a travel cost for moving to performing task based on the distance. The travel cost data is private and can be converted into l-bit binary string m_i. Then each worker w_i chooses two pseudo-random function $h_{s_i^a}$

Table 1 Summary of notations

Notation	Definition
$H_{l,m,o}$	$\{h_s : \{0,1\}^m \to \{0,1\}^l\}_{s\in\{0,1\}^o}$
$h_s(t)$	A function indexed by s in $H_{l,m,o}$
W, w_i	Worker set and the ith worker
I, I_i	ID sequence and the ith ID
Pos, Pos_i	Position sequence and the ith position
T, t_j	Task set and the jth task
l	Bit length of worker's data
m_i	Bit string of w_i's origin data
c_i	Bit string of workers' encrypted data
$\oplus$	Bitwise XOR operation
$\|$	Concatenation operation
c_i^j	True travel cost for w_i performing t_j
b_i^j	Claimed travel cost for w_i performing t_j
p_i^j	Payment for w_i performing t_j
X_i^j	Winning bid indicator for w_i performing t_j
S	Winning bid set
(i, j)	Index of worker and task in S

and $h_{s_i^b}$ from family $H_{l,m,o} = \{h_s : \{0,1\}^m \to \{0,1\}^l\}_{s\in\{0,1\}^o}$ and generates n random l-bit strings $k_i^j (j = 1, ..., n)$ via computing

$$k_i^j = h_{s_i^a}(t|j) \oplus h_{s_i^b}(t|j), \tag{10}$$

where $t|j$ is the concatenation of time stamp t and position index Pos_i. Each worker gets the same t when executing a data aggregation. Afterwards, worker w_i uses $k_i^{Pos_i}$ to encrypt its real data m_i and uses $k_i^j (j \neq Pos_i)$ to encrypt dummy data $\{0\}^l$ respectively. To be specific, worker w_i gets n encrypted l-bit strings by computing:

$$\{0\}^l \oplus k_i^1, ..., \{0\}^l \oplus k_i^{Pos_i-1}, m_i \oplus k_i^{Pos_i}, \{0\}^l \oplus k_i^{Pos_i+1}, ..., \{0\}^l \oplus k_i^n. \tag{11}$$

In the end, worker w_i can get the encrypted nl-bit string c_i by concatenating these n encrypted l-bit strings successively and send c_i to the platform.

Data decryption and analysis phase After the platform receives submitted ciphertexts $\{c|c_i \in \{0,1\}^{nl}\}(i = 1, ..., n)$, it can directly compute the concatenation of all workers' origin data by performing bitwise XOR operation:

$$m_1 \mid ... \mid m_{Pos_i} \mid ... \mid m_n = c_1 \oplus ... \oplus c_{Pos_i} \oplus ... \oplus c_n. \tag{12}$$

Then, the platform divides the concatenation into n pieces of data and generates task assignment and payment schema based on RATA algorithm, which will be discussed in Section 4.3.

Remark 1 Because only the Pos_i-th data in ciphertext c_i is the worker's origin data, the rest are dummy data $\{0\}^l$ and Pos is the random permutation of I. It is easy to prove that the bitwise XOR of all ciphertexts $\{c|c_i \in \{0,1\}^{nl}\}(i = 1, ..., n)$ equals the concatenation of origin data $\{m|m_i \in \{0,1\}^l\}(i = 1, ..., n)$.

Oblivious transfer phase After generating task assignment schema, the platform needs to allocate the task to selected workers. However, the platform only konws the position index Pos_i instead of ID I_i of selected worker w_i due to random permutation of received data. On the other hand, the agent holds the corresponding relation between ID sequence I and position sequence Pos. Therefore, the platform needs to perform oblivious transfer with the agent. In an OT_n^1 model, the agent (sender) holds the corresponding relation data and the platform (receiver) wants to get the Pos_i-th data. When they finish one round of OT_n^1 protocol, the platform only knows the ID I_i of the Pos_i-th worker other than the rest $n-1$ workers and the agent learns nothing about which piece of data the platform gets. In the end, the platform can allocate the task to the selected workers.

Example To make the proposed protocol more clear, we illustrate the main procedure via a simple example. As shown in Figure 2, there are 3 workers who have travel cost data $7_{10} = 0111_2$, $11_{10} = 1011_2$ and $3_{10} = 0011_2$ respectively. Further, the Pos_i number for 3 workers are 2, 3 and 1 respectively. For each worker, s/he chooses two pseudo-random functions with private key pair (s_i^a, s_i^b) and generates random bit string k_i^j with time stamp t and position number Pos_i. To encrypt the data, each worker needs to perform bitwise XOR operations between k_i^j and the real (or dummy) data. Then all 3 workers send their ciphertexts to the platform. After receiving all 3 bit strings, the platform performs the decryption by bitwise XOR operation on ciphertexts and breaks the bit string into 3 parts. Afterwards, the platform

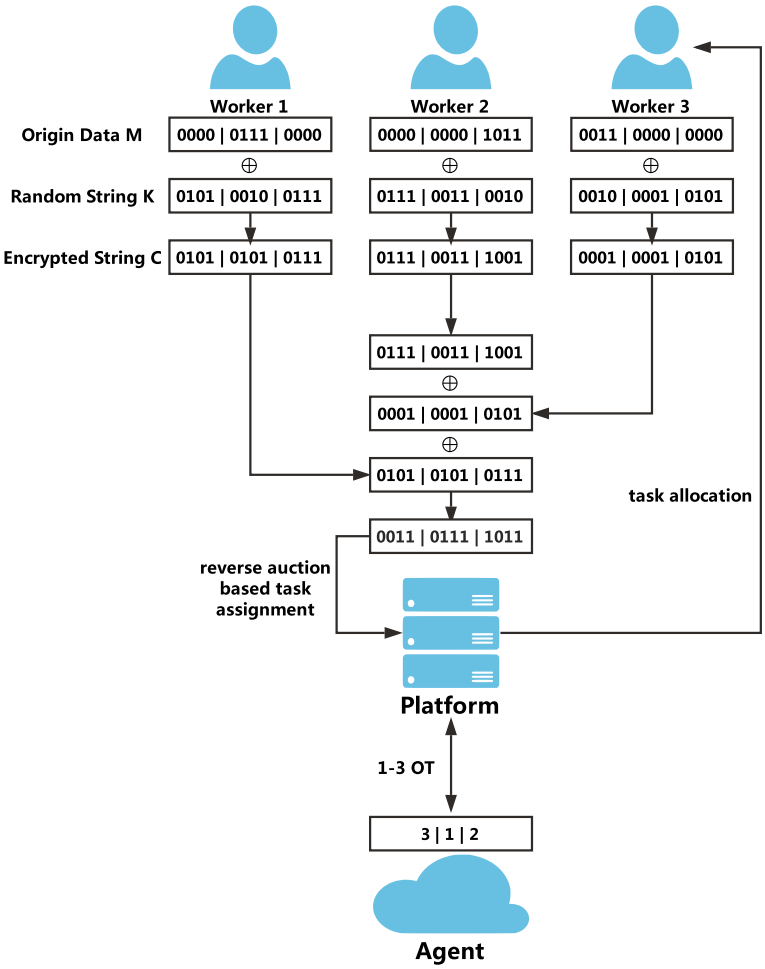


Figure 2 Example of ADA

can generate the task assignment schema based on received plaintexts. Here, the task will be assigned to the worker who is in the 1st position for simplification. However, the platform only knows the position of the selected worker other than its ID, while the agent holds the corresponding relations between IDs and positions. Hence, the platform executes an OT_3^1 protocol with the agent and learns that the selected worker is worker 3. Finally, the task will be assigned to worker 3.

4.3 Reverse auction based task assignment (RATA)

In our proposed model, workers send travel cost instead of private location data to the platform. Obviously, the performance of the generated task assignment schema severely depends on the authenticity of the submitted data. Since workers are different individually, it is reasonable to assume that workers are *rational* and *selfish*. In other words, each worker will not participate in tasks unless there is sufficient incentive and always tries to maximize

his/her own profit. Hence we need to design a truthful incentive mechanism to stimulate workers to compete for tasks at a truthful cost.

Reverse auction theory is the perfect theoretical tool to design a truthful incentive mechanism for task assignment in our model. To be specific, we model the platform as an *auctioneer*, workers as *sellers* of service by performing the tasks and submitted travel costs as *bids*. The platform takes these bids as input, selects a subset of bids as *winning bid* set and determines the *payment* for each winning bid. Based on this reverse auction model, we design a task assignment algorithm, which mainly involves two key problems: the *Winning Bid Selection Problem* and the *Payment Determination Problem*.

First, we define some basic notations for the auction:

Definition 8 (**Bid and winning bid**) When w_i wants to perform t_j, it will submit a *bid*, denoted by b_i^j, which is a cost claimed by w_i. Since w_i always tries to maximize his/her own profit, s/he may submit a bid b_i^j which is greater than his/her true travel cost c_i^j. When w_i wins task t_j by bid b_i^j, we call b_i^j *winning bid*.

Obviously, the Winner Selection Problem is an optimization problem: Given a set of bids B, selects a subset (task assignment schema) S such that the social welfare $\phi(S)$ is maximized over all possible subsets. The problem is defined as follows:

Definition 9 (**Winner selection problem**)

$$Maximize: \quad \phi(S) = \sum_{t_j \in S} v_{t_j} - \sum_{(w_i,t_j)\in S} b_i^j \tag{13}$$

$$Subject\ to: \quad \sum_{(w_i,t_j)\in S} X_i^j \leq c_{w_i}, X_i^j \in \{0,1\} \tag{14}$$

$$\sum_{(w_i,t_j)\in S} b_i^j \leq v_{t_j} \tag{15}$$

$$\forall_{(w_i,t_j)\in S}, (a_{t_j}, d_{t_j}) \cap (a_{w_i}, d_{w_j}) \neq \phi \tag{16}$$

Remark 2 The reverse auction based task assignment algorithm is a truthful incentive mechanism, which means that each worker must submit a bid that equals to his/her true travel cost. Therefore, we can say $b_i^j = c_i^j$. Then, (13) is equivalent to (5). In addition, $X_i^j = 1$ indicates that bid b_i^j wins the auction and task t_j will be assigned to worker w_i. The three constraints are the same as Definition 7 and the ADA protocol guarantees the security.

Then, we can formalize the Payment Determination Problem as follows.

Definition 10 (**The Payment Determination Problem**) The Payment Determination Problem is to compute the payment for each bid $b_i^j \in S$, satisfying *truthfulness* and *individual rationality*.

– **Truthfulness:** Let b_i^j be the truthful bid submitted by worker w_i for task t_j and $\hat{b_i^j}$ be the untruthful bid which is manipulated by w_i. The payments for the truthful bid and untruthful bid are p_i^j and $\hat{p_i^j}$. Then, we say that the auction protocol is truthful if

$$p_i^j - c_i^j \geq \hat{p_i^j} - c_i^j \tag{17}$$

– **Individual Rationality:** For each winner worker, we say that the auction protocol satisfies individual rationality if the corresponding profit is non-negative, i.e.,

$$p_i^j - c_i^j \geq 0 \tag{18}$$

The Winner Selection Problem is NP-hard, which can be proved as follows:

Theorem 1 *The Winner Selection Problem is NP-hard.*

Proof First, we consider a special case of the Winner Selection Problem, where these is only one worker w_i and his/her bid is B_i. Then, this special problem is to select a subset S_i from B_i to maximize the social welfare $\phi(S_i)$ with the constraint of $|S_i| \leq c_{w_i}$. This is equivalent to the well-known 0-1 knapsack problem which is NP-hard: given a set B_i, each item in which has a value $v_{t_j} - c_i^j$ and a weight 1, determining a subset to maximize the whole value, while ensuring the total weight is not large than c_{w_i}. Therefore, the special problem is NP-hard and the general Winner Selection Problem is also at least NP-hard. □

Due to the Winner Selection Problem is an NP-hard problem, we turn to design an approximation algorithm via *submodularity* of the social welfare function.

Definition 11 (**Submodular function**) Let σ be a finite set, a function $f : 2^\sigma \to \mathbb{R}$ is submodular if

$$f(X \cup \{x\}) - f(X) \geq f(Y \cup \{x\}) - f(Y) \tag{19}$$

for any $X \subseteq Y \subseteq \sigma$ and $x \in \sigma \setminus Y$.

We now first prove the submodularity of the social welfare function $\phi(S)$.

Lemma 1 *The social welfare function $\phi(S)$ is submodular.*

Proof By Definition 11, we need to show that

$$\phi(X \cup \{b_i^j\}) - \phi(X) \geq \phi(Y \cup \{b_i^j\}) - \phi(Y) \tag{20}$$

for any $X \subseteq Y \subseteq S$ and $b_i^j \in S \setminus Y$. As one task can be assigned to multiple workers, after adding a new bid b_i^j, X may expand while Y does not. We use $flag_x$ and $flag_y$ with value 0 or 1 to denote whether the set X and set Y expand or not, respectively. If $flag_y = 1$ then $flag_x = 1$, however, if $flag_y = 1$, $flag_x$ can be 0 or 1. Thus we have

$$\begin{aligned}
\phi(X \cup \{b_i^j\}) - \phi(X) &= v_{t_j} * flag_x - b_i^j \\
&\geq v_{t_j} * flag_y - b_i^j \\
&= \phi(Y \cup \{b_i^j\}) - \phi(Y)
\end{aligned}$$

Therefore social welfare function $\phi(S)$ is submodular. □

Based on the submodularity of the optimization objective, we design an algorithm as illustrated in Algorithm 1. The winner bid selection phase follows a greedy approach: Bids are essentially sorted according to the difference of their marginal values. Given the selected bid set S, the marginal value of bid b_i^j is $V_i^j(S) = V(S \cup \{b_i^j\}) - V(S) = \sum_{t_k \in \tau(S \cup \{b_i^j\})} v_{t_k} - \sum_{t_k \in \tau(S)} v_{t_k}$. In the sorting, the (k + 1) bid is the bid b_i^j such that $V_i^j(S_k) - b_i^j$ is maximized

over $B \setminus S_k$, where S_k is the winning set in the k-th iteration and $S_0 = \emptyset$. We use V_k instead of $V_i^j(S_{k-1})$ and b_k instead of the kth bid in the sorting for simplification. The sorting implies the submodularity of social welfare function by:

$$V_1 - b_1 \geq V_2 - b_2 \geq ... \geq V_{|B|} - b_{|B|} \tag{21}$$

The *If* statement(line 3) ensures the three constraints: for each worker-and-task matching pair, the time requirements need to be met; for each task t_j, the total payment should not be greater the value v_{t_j} of t_j; for each worker w_i, it can afford no more than c_{w_i} tasks.

Algorithm 1 Winner selection

Input : task set T, worker set W, bid set B
Output: winning bid set S, winning indicator matrix X

1 **Initialize**: $S \leftarrow \emptyset$, $X \leftarrow 0$, $(w_i, t_j) \leftarrow \arg\max_{b_i^j \in B}(V_i^j(S) - b_i^j)$;
2 **while** $B \neq \emptyset$ **do**
3 **if** $\forall_{(w_i,t_j)\in S}, (a_{t_j}, d_{t_j}) \cap (a_{w_i}, d_{w_j}) \neq \emptyset$ *and* $\sum_{b_k^j \in S} b_k^j \leq v_{t_j}$ *and* $\sum_{b_i^l \in S} X_i^l \leq c_{w_i}$ **then**
4 $S \leftarrow S \cup \{b_i^j\}$;
5 $X_i^j \leftarrow 1$;
6 $B \leftarrow B \setminus b_i^j$;
7 $(w_i, t_j) \leftarrow \arg\max_{b_i^j \in B}(V_i^j(S) - b_i^j)$;

4.3.1 Payment determination

The payment determination algorithm is to compute the payment for each winning bid, ensuring each worker honestly claims its true cost. The truthfulness of algorithm relies on the well-known Myerson's theory [26].

Theorem 2 *An auction mechanism is truthful if and only if:*

- *The selection rule is monotone: If worker w_i wins the auction by bid b_i^j, s/he also wins by bid $\tilde{b_i^j} \leq b_i^j$;*
- *Each winner is paid the critical value: worker w_i would not win the auction if s/he bids higher than this value.*

Based on Theorem 2, we design our payment determination algorithm, as shown in Algorithm 2. This algorithm also computes the payment p_i^j for each winning bid b_i^j in a greedy manner. As similar with (21), bids are sorted over $\tilde{B} = B \setminus b_i^j$ by

$$V_1 - b_1 \geq V_2 - b_2 \geq ... \geq V_{|\tilde{B}|} - b_{|\tilde{B}|} \tag{22}$$

where $V_m = V_k^l(T_{m-1} \cup b_k^l) - V_k^l(T_{m-1})$ and T_{m-1} is the winning set in the m-th iteration for b_i^j. For each position m in the sorting, we compute the maximal price that b_i^j can reach such that b_i^j can be selected instead of the bid in m-th position. The iteration will be repeated

until the last position M satisfying $V_M \geq B_M$. In the end, we set the value of p_i^j to the maximum of these M bids.

Algorithm 2 Payment determination

Input : bid set B, winning bid set S
Output: payment set P

Initialize: $P \leftarrow 0$;
for $b_i^j \in S$ **do**
  $\tilde{B} \leftarrow B \setminus b_i^j$;
  $T \leftarrow \emptyset$;
  repeat
    $b_k^l \leftarrow \arg\max_{b_k^l \in \tilde{B} \setminus T}(V_k^l(T) - b_k^l)$;
    $p_i^j \leftarrow \max\{p_i^j, \min\{V_i^j(T) - (V_k^l(T) - b_k^l), V_i^j(T)\}\}$;
    $T \leftarrow T \cup b_k^l$;
  until $b_k^l > V_k^l(T)$ *or* $T = \tilde{B}$;
  if $T = \tilde{B}$ **then**
    $p_i^j \leftarrow \max\{p_i^j, V_i^j(T)\}$;

5 Theoretical analysis

5.1 Validity analysis

Theorem 3 *The task assignment algorithm is truthful.*

Proof According to Theorem 2, it suffices to prove that the selection rule of Algorithm 1 is monotone and the payment p_i^j computed in Algorithm 2 is critical value. The monotonicity is obvious as bidding a smaller value cannot push b_i^j backwards in the sorting order. We next show that p_i^j is the critical value making the bid b_i^j win the auction exactly. Note that

$$p_i^j = \max\left\{\max_{1 \leq m \leq M}(V_i^j - (V_m - b_m)), V_{M+1}\right\}. \tag{23}$$

If worker bids $b_i^j > p_i^j$, the bid will be placed after the position M since $b_i^j > V_i^j - (V_m - b_m)$. At the $(M+1)$th iteration, b_i^j will not be selected as winning bid as $b_i^j > V_{M+1}$. Hence, p_i^j is the critical value and the theorem holds. □

Theorem 4 *The task assignment algorithm satisfies individual rationality.*

Proof We need to prove $p_i^j - c_i^j \geq 0$. If $b_i^j \notin S$, then $p_i^j - c_i^j = 0$. Now assume that $b_i^j \in S$, the iteration guarantees that $b_i^j \leq V_i^j$ and $p_i^j \leftarrow \max\{p_i^j, \min\{V_i^j(T) - (V_k^l(T) - b_k^l), V_i^j(T)\}\}$. Hence, $b_i^j \leq V_i^j \leq p_i^j$. Further, due to the truthfulness of task assignment algorithm, we have $b_i^j = c_i^j$. Therefore, we have $p_i^j - c_i^j \geq 0$. The theorem holds. □

5.2 Security analysis

Lemma 2 *Our privacy-preserving task assignment protocol is secure against semi-honest adversaries.*

Proof Given a set of semi-honest workers $W = (w_1, ..., w_n)$, a sequence of their data $M = (m_1, ..., m_n)$, a sequence of their encrypted data $C = (c_1, ..., c_n)$ which will be sent to the platform as its input.

For $\forall(w_i, w_j) \in W$ where $1 < i < j < n$, the view of the platform in execution of our protocol are as follows:

$$VIEW = ((c_1, ..., c_{i-1}, c_i, c_{i+1}, ..., c_{j-1}, c_j, c_{j+1}, ..., c_n), r, x) \tag{24}$$

where x represents the output of received data which is immutable and r represents coin flips.

Suppose that worker w_i and w_j switch their data, the sequence of data changes to

$$M' = (m_1, ..., m_{i-1}, m_j, m_{i+1}, ..., m_{j-1}, m_i, m_{j+1}, ...) \tag{25}$$

and then view of platform in the protocol also becomes

$$VIEW' = ((c_1, ..., c_{i-1}, c_j, c_{i+1}, ..., c_{j-1}, c_i, c_{j+1}, ..., c_n), r, x) \tag{26}$$

According to Definition 1, to prove the security of out protocol is equivalent to prove

$$VIEW \equiv VIEW' \tag{27}$$

holds for $\forall(w_i, w_j) \in W$ where $\equiv$ denotes computationally indistinguishability of two random variable ensembles.

To clarify this, we construct a simulator Sim that takes $C = (c_1, ..., c_n)$ as input and outputs a view $VIEW''$ which is computationally indistinguishable to both $VIEW$ and $VIEW'$. Specifically, Sim runs the same protocol with all workers and gets received data C as input. Then Sim performs a random permutation function π ($[1, n] \rightarrow [1, n]$) on input and converts it $C'' = \pi(c_1, ..., c_n)$. Further, the view of Sim is updated at the same time:

$$VIEW'' = (\pi(c_1, ..., c_n), r, x). \tag{28}$$

Obviously, we know $VIEW \equiv VIEW''$, because it is impossible to distinguish C and permutation C'' in polynomial time. In a similar way, we also know $VIEW' \equiv VIEW''$. Therefore, we have

$$VIEW \equiv VIEW' \tag{29}$$

□

Theorem 5 *Our privacy-preserving task assignment model can prevent workers' location from being revealed.*

Proof Firstly, based on Lemma 2, the platform cannot efficiently notice any difference if we switch two workers' data. After analyzing the received data, the platform only knows the source of selected worker is one of the n workers because of the random permutation. Secondly, the agent only knows the corresponding relations between ID sequence I and position sequence Pos which cannot help it infer other private information. Last but not least, the oblivious transfer protocol ensures that the platform only learns the IDs of selected workers and the agent learns nothing about task assignment schema. The theorem holds. □

5.3 Complexity analysis

5.3.1 Efficiency of ADA

The ADA protocol has a high efficiency both on the computation time and communication. We will analyze the complexity of ADA from the aspects of the platform, the agent and each worker. The result is shown in Table 2. Note that, *XOR* represents bitwise XOR operation, *HASH* represents one-way hashing for pseudo-random function and OT_2^1 represents computation (or communication) overhead of a round of basic 1-2 OT protocol. In addition, we use n instead of workers number $|W|$ for simplification.

Platform On each round of ADA protocol, the platform needs to perform n^2l bitwise XOR operations on received nl-bit data submitted by n workers. Therefore, the communication overhead is n^2l bits. Based on efficient implementation in [27], OT_n^1 invokes $logn$ basic OT_2^1. Hence, the computation and communication cost of the platform are n^2l XOR $+ logn\ OT_2^1$ and $n^2l + logn\ OT_2^1$, respectively.

Agent For the agent, the time of generating and distributing keys is negligible, while the communication overhead is $2nl$ bit. In the round of OT_n^1, the agent needs to invoke $logn$ basic OT_2^1 and perform the other $nlogn$ hash operations. Therefore, the computation and communication overhead of the agent are $nlogn$ HASH $+ logn\ OT_2^1$ and $2nl + logn\ OT_2^1$, respectively.

Worker For each worker, the time and space overhead rely on the ADA protocol. In each round of the ADA protocol, each worker w_i firstly performs $2n$ hash operations to compute $h_{s_i^a}(t|j)$ and $h_{s_i^b}(t|j)$, then performs nl bitwise XOR operations to compute $h_{s_i^a}(t|j) \oplus h_{s_i^b}(t|j)$ and performs nl bitwise XOR operations to encryption real or dummy data in the end. Hence, the computation cost is $2n$ HASH + $2nl$ XOR. Moreover, the communication overhead is nl-bits encrypted data.

5.3.2 Efficiency of RATA

The computation overhead of RATA is dominated by finding the bid with maximum marginal value $V_i^j(S) - b_i^j$ which takes $O(|B|)$. Hence, the while-loop (line 2–7) in Algorithm 1 takes $O(|B|^2)$ time to select winning bid set S from origin bid set B and the for-loop (line 2–11) in Algorithm 2 takes $O(|S||B|^2)$ time to compute payment for $|S|$. It is noted that $|S|$ is proportional to $|B|$, the latter complexity can be deduced to $O(|B|^3)$. In conclusion, the RATA has a polynomial-time computation complexity.

Table 2 Complexity of ADA

	Computation time	Communication cost
Platform	n^2l XOR $+ logn\ OT_2^1$	$n^2l + logn\ OT_2^1$
Agent	$nlogn$ HASH $+ logn\ OT_2^1$	$2nl + logn\ OT_2^1$
Worker w_i	$2n$ HASH $+2nl$ XOR	nl

6 Experimental evaluation

In this section, we present experimental evaluations of our model from the following aspects: the computation and communication efficiency of ADA, the social welfare of RATA, truthfulness and individual rationality. Since the computation overhead of RATA only depends on the number of bids, which has been deduced theoretically in Section 5.3, we will not evaluate the aspect.

We adopt the widely-used read world dataset Gowalla [7] which has a total of 6,442,890 check-ins of 196,591 users over the period of Feb. 2009–Oct. 2010. In our simulation, we assume that Gowalla users are the workers of the spatial crowdsourcing platform and their locations are those of most recent check-in points. We also model each check-in point as a task which needs to be performed by workers. The travel cost of each $\langle worker, task \rangle$ pair can be measured by *unit cost* (u_c for short). We select a subset of all users as workers, and the number of workers $|W|$ is set to 100,200,500,1000,2000. The capacity c_{w_i} of each worker is set to 1, 3 and 5. Then we randomly deploy a number of tasks where the number is set to 100,200,500,1000,2000. The value v_{t_j} of each task is randomly sampled in the range [5, 30] times of u_c. The bit length l are set to 5,10,15,20,30. When l reaches 30, it means a quite large data space which is $[0,2^{30}-1]$. All of these simulation parameters are listed in Table 3, where the default value for each parameter is shown in boldface.

Furthermore, all experiments are performed on a PC running 64-bit windows 10 with Intel Core i5-3470 3.2 GHz CPU and 8GB memory. The code is implemented in Java and executed in JDK 1.8. We use SHA512 as pseudo-random function family for bitwise XOR homomorphic cipher and OT. Results of experiments on our proposed model are averaged by 100 runs.

6.1 Evaluation of ADA

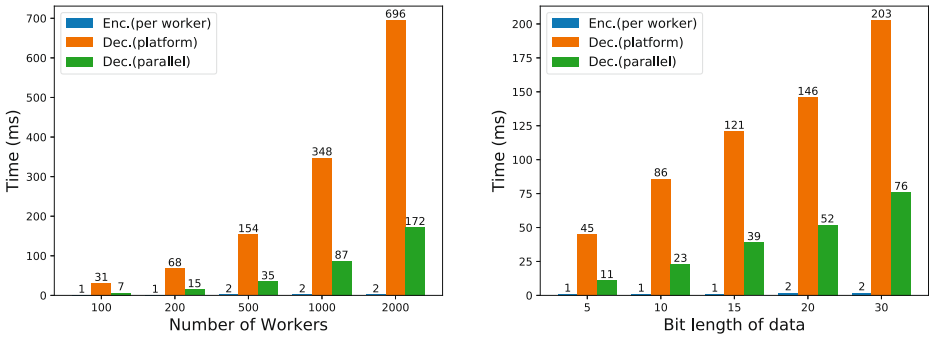
6.1.1 Computation time

Based on complexity analysis in Section 5.3, there are two critical factors effecting the time efficiency of ADA protocol: the number of workers n and the bit length l of data. We evaluate the computation overhead on two aspects: Encryption/Decryption time and OT execution time.

Encryption/Decryption time First, as shown in Figure 3, the computation cost increases a little faster on the influence of n than that on the influence of l. This is reasonable because l is in a small value interval, while n can be very large. Second, the encryption time for per worker is negligible so that the protocol is suitable for mobile device with limited resource.

Table 3 Evaluation settings

Parameter	Value
Number of workers $\|W\|$	100, 200, **500**, 1000, 2000
Number of tasks $\|T\|$	100, 200, **500**, 1000, 2000
Bit length of bid l	5, 10, **15**, 20, 30
Value v_{t_j} of task t_j	[5, 30]
Capacity c_{w_i} of worker w_i	1, **3**, 5



(a) Effect of the number of workers $|W|$ (b) Effect of the bit length of data l

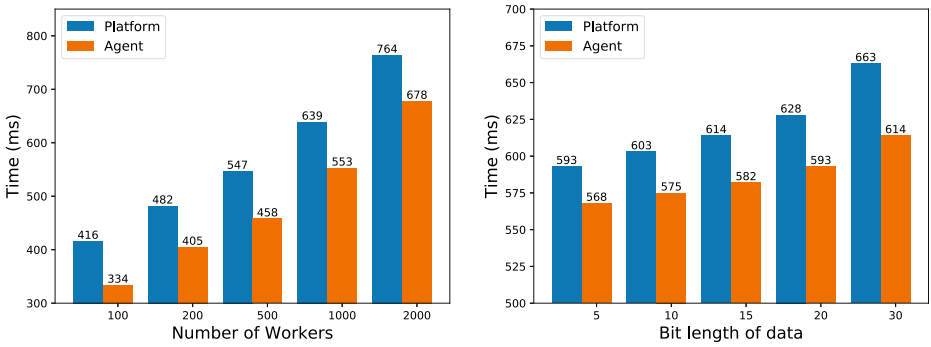
Figure 3 Encryption/Decryption time of ADA

Third, with the help of parallel technology, the decryption time of the platform can be deduced to a satisfactory level. For example, the decryption time of parallel situation comes down to 172 ms which is a quarter of the origin time when $n = 2000$ in Figure 3a.

OT execution time Similarly, as shown in Figure 4, the OT execution time is prone to be effected by n other than l. Further, the execution time of the platform is slightly more than that of the agent. This is because the platform needs to perform more asymmetric cryptography operations in the round of OT.

6.1.2 Communication cost

The communication cost of per worker is proportional to l and n, and is negligible for each worker. Therefore, we focus on the communication cost of the execution phase of OT_n^1 using the same experimental setting as in the computation evaluations. As shown in Table 4, communication is prone to be affected by n. However, even when n reaches 2000, the communication cost between the platform and the agent is only 119.02 kb.



(a) Effect of the number of workers $|W|$ (b) Effect of the bit length of data l

Figure 4 OT execution time of ADA

Table 4 Communication cost of OT

	l = 5	l = 10	l = 15	l = 20	l = 30
Comm. (kb)	76.85	77.63	78.06	78.59	79.81
	n = 100	n = 200	n = 500	n = 1000	n = 2000
Comm. (kb)	41.25	48.31	58.29	78.65	119.02

6.2 Evaluation of RATA

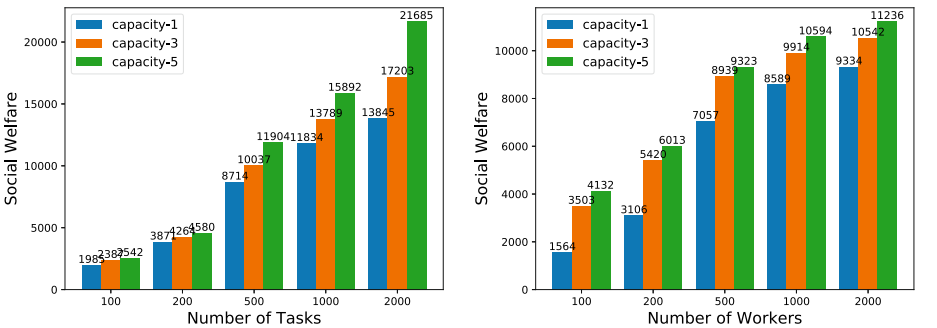
6.2.1 Social welfare

As shown in Figure 5, we evaluate the effect of the number of tasks $|T|$ and the number of workers $|W|$ on the social welfare. The effect of capacity of each worker is also considered.

We can find that capacity-5 achieves the highest social welfare, while capacity-1 obtains the smallest social welfare. This is because capacity represents the number of tasks one worker can handle at some period. Especially, capacity-1 means each worker can perform 1 task or not. The higher capacity is, the more tasks one worker can obtain. Hence, in the multi capacity situation, tasks can be assigned to more potential workers to achieve the higher overall social welfare.

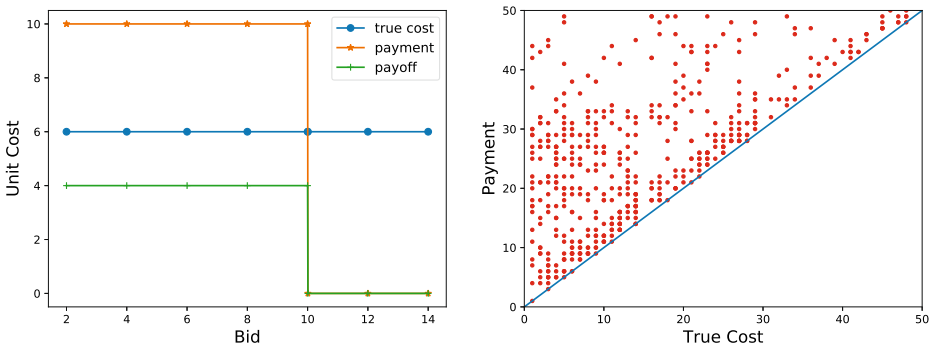
Effect of the number of tasks We set the number of workers $|W|$ to 500 and increase the number of tasks $|T|$ from 100 to 2000. As shown in Figure 5a, the social welfare grows sharply when $|W| \geq |T|$ and increases slightly when $|W| < |T|$. This is because almost each task will be assigned when $|W| \geq |T|$. The more tasks are, the higher social welfare is. While $|W| < |T|$, only the tasks with high values can be performed by workers. On the other hand, the social welfare with capacity-5 is higher than capacity-3 and capacity-1 when $|W| \geq |T|$. It is reasonable that workers with higher capacity will obtain more tasks, hence more tasks will be performed to achieve higher total social welfare.

Effect of the number of workers We set the number of tasks $|T|$ to 500 and increase the number of workers $|W|$ from 100 to 2000. The result is shown in Figure 5b. When $|W| \leq |T|$, the social welfare increases with the growth of workers and the capacity of each worker. While $|W| > |T|$, the social welfare grows slightly due to the shortage of tasks.



(a) Effect of the number of tasks $|T|$ (b) Effect of the number of workers $|W|$

Figure 5 Social welfare



(a) Truthfulness (b) Individual rationality

Figure 6 Evaluation of truthfulness and individual rationality

6.2.2 Truthfulness and individual rationality

Truthfulness To verify of truthfulness of RATA, we randomly choose a worker and allow s/he to claim his/her bids different from his/her real travel cost. The result is shown in Figure 6a, where the critical payment is 10 u_c and the true travel cost is 6 u_c. When the claimed bid value is not large than the critical payment, the payment and payoff remain unchanged, which are 10 u_c and $10 - 6 = 4u_c$, respectively. Otherwise, if the claimed bid exceeds the critical payment, the corresponding payment and payoff become 0. Hence, the RATA is truthful.

Individual rationality To verify the individual rationality of workers, we randomly choose some winning bids and depict them in Figure 6b, according to the corresponding payment and real travel cost. The results show that each payment is higher than the real travel cost, which means that RATA satisfies the property of individual rationality.

7 Related work

Crowdsourcing is an emerging way to utilize the capabilities of crowd to address computer-hard tasks, such as transcribing books, folding proteins, and classifying galaxies. The involvement of human however makes crowdsourcing much challenging, as different workers may need different times and costs to do the same task, and their answers may have different qualities. Therefore, a great deal of work on crowdsourcing has been reported recently, trying to achieve high quality answers in a cost-effective and efficient way [17, 19]. In [18], the authors design and implement a crowd-powered database system that can generate high-quality answers with small cost and low latency. The superiority of their multi-goal optimization strategy is demonstrated on both simulated and real experiments. Another interesting problem in crowdsourcing is truth inference, that is, how to infer the truth based on workers' answers. In [47], the authors adopt Bayesian Voting (BV) to solve Jury selection problem. As computing Jury quality with BV is NP-hard, an approximation algorithm is proposed to reduce computation cost. In [49], domain knowledge is utilized to model a worker's quality to facilitate the inference of the true answer of a task. In [50], Zheng ets al. provide a detailed experimental study on existing truth inference algorithms

over five real datasets, identifying the limitations of existing work and promising research directions. Task assignment is also a hot research issue in crowdsourcing. In [48], two popular evaluation metrics, i.e., Accuracy and F-score, are considered in task assignment to improve the quality of final answers. In [14], the authors leverage crowdsourcing to improve the quality of POI labeling. Tasks are assigned to workers in a dynamic way so that more accurate inference for next available workers can be made.

As mentioned earlier, Spatial Crowdsourcing (SC) is a special kind of crowdsourcing where workers need to physically move to some locations to perform tasks. This brings some new challenges for spatial crowdsourcing. Among them, how to address location constraint or protect location privacy in task assignment is really appealing, as this is largely overlooked by current studies in non-spatial crowdsourcing. Further, we use incentive mechanism in this paper to encourage workers to report correct information. Therefore, we subsequently discuss related work from the following aspects: task assignment in spatial crowdsourcing, location privacy, privacy-preserving spatial crowdsourcing and incentive mechanism in crowdsourcing.

Task assignment in spatial crowdsourcing In [16], Kazemi and Shahabi formulate task assignment as a matching problem between tasks and workers. They introduce a framework called GeoCrowd to maximize the number of assigned tasks. Similarly, Chen et al. [4] propose a general platform gMission which possesses some features of task recommendation. Deng et al. [9] formulate task assignment as a scheduling problem in which the goal is to maximize the number of performed tasks for each worker. Cheng et al. [6] tackle the situation in which workers have multi-skills. In [46], Zheng et al. take workers' rejection into consideration and try to maximize acceptance. In [35], Tong et al. consider task assignment in online scenarios and formulate it as an online maximum weighted bipartite match problem. All of these works do not take location privacy into consideration as they assume that workers are willing to share their private location information with the platform. Our work complements these works by protecting location privacy in the phase of task assignment.

Location privacy Ghinita et al. [12] adopt private information retrieval (PIR) to enable users to conduct approximate and exact nearest neighbor search without revealing their locations to the server. Paulet et al. [29] combines PIR and oblivious transfer (OT) to achieve mutual privacy-preserving location-based queries. On one hand, the server is unable to know the location of users. On the other hand, users can only get a limited location data for their queries, thus protecting the server's private data. Liu et al. [21] propose a more efficient approach for this problem by utilizing two rounds of OT and show the efficiency improvement can be realized at the expense of acceptable communication cost. Yi et al. [41] present a solution based on Paillier and Rabin cryptosystem for mutual privacy-preserving kNN query where k is fixed. The solution is extended in [42] to support dynamic k up to a constant and sequential queries. However, these solutions cannot be applied to our scenario. This is because, in SC, worker location is not the private data of the SC-server, but rather the sensitive information that workers want to hide from the SC-server. There are also some works focusing on privacy-preserving location-based queries over outsourced location data [40, 43], where the data owner and users sending queries are assumed to trust each other. In SC, however, there is no inherent trust relationship between task requesters and workers. To enable kNN query over encrypted data, Elmehdwi et al. [10] propose a set of protocols based on Paillier. While mutual privacy can be guaranteed due to the security of Paillier, the computation cost of these protocols are very expensive [20]. Thus, we cannot apply these protocols to directly solve large task assignment problems.

Privacy-preserving spatial crowdsourcing Pournajaf et al. [30] summarizes typical privacy-preserving techniques used in spatial crowdsourcing, including k-anonymization, spatial cloaking and differential privacy. In [33], the number of workers in a region is disguised using differential privacy by a trusted party. The SC platform needs to find a region that is very likely to contain enough workers nearby the task location and the workers in this region are notified to perform the task. Similarly, Xiong et al. [38] introduce a trusted third party to process the origin location data according to differential privacy and construct contour line to indicate location distribution. Since dummy data are injected into raw worker location data, the SC platform cannot learn the true location information of each worker. In [34], To et al. perturb the locations of both tasks and workers based on geo-indistinguishability [1] and perform task assignment based on the probabilities of reachability between tasks and workers. In [22], the authors utilize Geohash to realize a cloaked area to hide the exact locations of workers. Methods based on differential privacy or spatial cloaking, however, has an inherent weakness, that is, data utility will be inevitably reduced due to the injected noise [24, 37]. To ensure the accuracy of task assignment, different kinds of cryptosystems have been employed in spatial crowdsourcing. In [23], worker locations are encrypted by Paillier [28] and indexed by an SKD tree, a newly designed data structure which eliminates potential information leakage of normal KD-tree caused by its public splitting dimension. However, workers are not static and they often move from one place to another, so the SKD tree needs to be updated frequently. Unfortunately, the update operation is very time-consuming especially when there are a large number of workers, which makes it unsuitable for large-scale real time SC applications. In [25], Liu et al. combine Paillier and ElGamal to design a protocol which allows the platform to find workers with the shortest travel time without knowing the private data of workers and tasks. Though this protocol provides strong privacy guarantee, it still suffers from computation time issue and cannot scale to large SC applications.

Incentive mechanism From the perspective of participating workers and the platform, Yang et al. [39] propose user-centric and platform-centric incentive mechanisms respectively. The former guarantees the validity of the mechanism based on a reverse auction model and the latter computes the unique Stackelberg Equilibrium to maximize the utility of the platform. In [32], Tham et al. propose two incentive mechanisms, IDF (Incentive with Demand Fairness) and TIf (Iterative Tank Filling), to ensure the fairness of workers and maximize the social welfare. Zhao et al. [45] propose two online incentive mechanism based on the timing of worker participation and the character of non-negative submodular function.

8 Conclusion

In this paper, we have presented a novel privacy-preserving task assignment model in spatial crowdsourcing. First, we have generalized workers' private location data to travel cost and protect it in a k-anonymity manner via combining the bitwise XOR homomorphic cipher and OT. Second, to guarantee the authenticity of the travel costs submitted by workers, we have designed a reverse auction based task assignment algorithm to produce task assignment schema. Further, We have theoretically proved the security and analyzed the complexity of our model. In the end, experimental results have shown that our proposed model has considerable efficiency in both computation time and communication overhead, and the incentive mechanism can achieve good social welfare while satisfying the truthfulness and individual rationality.

The work reported in this paper can still be improved in several ways. For example, a weakness of our method is in the static problem setting. In most real world spatial crowdsourcing scenarios, however, both tasks and workers come dynamically. Intuitively, our method can be easily extended to an online problem setting by using a batched assignment scheme where the assignment is delayed for a period of time (e.g. 10 min) [5, 8, 13]. However, Asghari et al. [2] point out this method has two main disadvantages. On one hand, every task has less available time to be scheduled. On the other hand, this delay cannot be accepted in some real-time applications such as Uber and Didi. Therefore, privacy-preserving online task assignment is one of the next steps we need to consider.

Acknowledgements Research reported in this publication was partially supported by Natural Science Foundation of China (Grant Nos. 61572336, 61632016, 61572335), and the Natural Science Research Project of Jiangsu Higher Education Institution (Grant No. 18KJA520010).

References

1. Andrés, M.E., Bordenabe, N.E., Chatzikokolakis, K., Palamidessi, C.: Geo-indistinguishability: differential privacy for location-based systems. In: 2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13, Berlin, Germany, November 4–8, 2013, pp. 901–914 (2013)
2. Asghari, M., Shahabi, C.: On on-line task assignment in spatial crowdsourcing. In: 2017 IEEE International Conference on Big Data, BigData 2017, Boston, MA, USA, December 11–14, 2017, pp. 395–404 (2017)
3. Chen, L., Shahabi, C.: Spatial crowdsourcing: challenges and opportunities. IEEE Data Eng. Bull. **39**(4), 14–25 (2016)
4. Chen, Z., Fu, R., Zhao, Z., Liu, Z., Xia, L., Chen, L., Cheng, P., Cao, C.C., Tong, Y., Zhang, C.J.: gmission: A general spatial crowdsourcing platform. Proc. VLDB Endow. **7**(13), 1629–1632 (2014)
5. Chen, C., Cheng, S., Lau, H.C., Misra, A.: Towards city-scale mobile crowdsourcing: task recommendations under trajectory uncertainties. In: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25–31, 2015, pp. 1113–1119 (2015)
6. Cheng, P., Lian, X., Chen, L., Han, J., Zhao, J.: Task assignment on multi-skill oriented spatial crowdsourcing. IEEE Trans. Knowl. Data Eng. **28**(8), 2201–2215 (2015)
7. Cho, E., Myers, S.A., Leskovec, J.: Friendship and mobility: user movement in location-based social networks. In: ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, Ca, USA, August, pp. 1082–1090 (2011)
8. Deng, D., Shahabi, C., Zhu, L.: Task matching and scheduling for multiple workers in spatial crowdsourcing. In: Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems, Bellevue, WA, USA, November 3–6, 2015, pp. 21:1–21:10 (2015)
9. Deng, D., Shahabi, C., Demiryurek, U., Zhu, L.: Task selection in spatial crowdsourcing from worker's perspective. Geoinformatica **20**(3), 529–568 (2016)
10. Elmehdwi, Y., Samanthula, B.K., Jiang, W.: Secure k-nearest neighbor query over encrypted data in outsourced environments. In: IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31–April 4, 2014, pp. 664–675 (2014)
11. Even, S., Goldreich, O., Lempel, A.: A Randomized Protocol for Signing Contracts. Springer US (1983)
12. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.: Private queries in location based services: anonymizers are not necessary. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2008, Vancouver, BC, Canada, June 10–12, 2008, pp. 121–132 (2008)
13. Guo, B., Liu, Y., Wu, W., Yu, Z., Han, Q.: Activecrowd: a framework for optimized multitask allocation in mobile crowdsensing systems. IEEE Trans. Human-Mach. Syst. **47**(3), 392–403 (2017)

14. Hu, H., Zheng, Y., Bao, Z., Li, G., Feng, J., Cheng, R.: Crowdsourced poi labelling: location-aware result inference and task assignment. In: IEEE International Conference on Data Engineering, pp. 61–72 (2016)
15. Kazemi, L., Shahabi, C.: A privacy-aware framework for participatory sensing. SIGKDD Explor. **13**(1), 43–51 (2011)
16. Kazemi, L., Shahabi, C.: Geocrowd: enabling query answering with spatial crowdsourcing. In: International Conference on Advances in Geographic Information Systems, pp. 189–198 (2012)
17. Li, G., Wang, J., Zheng, Y., Franklin, M.J.: Crowdsourced data management: a survey. IEEE Trans. Knowl. Data Eng. **28**(9), 2296–2319 (2016)
18. Li, G., Chai, C., Fan, J., Weng, X., Li, J., Zheng, Y., Li, Y., Yu, X., Zhang, X., Yuan, H.: Cdb: optimizing queries with crowd-based selections and joins. In: ACM International Conference, pp. 1463–1478 (2017)
19. Li, G., Fan, J., Fan, J., Wang, J., Cheng, R.: Crowdsourced data management: overview and challenges. In: ACM International Conference on Management of Data, pp. 1711–1716 (2017)
20. Liu, A., Zheng, K., Li, L., Liu, G., Zhao, L., Zhou, X.: Efficient secure similarity computation on encrypted trajectory data. In: IEEE International Conference on Data Engineering, pp. 66–77 (2015)
21. Liu, S., Liu, A., Zhao, L., Liu, G., Li, Z., Zhao, P., Zheng, K., Qin, L.: Efficient query processing with mutual privacy protection for location-based services. In: Proceedings, Part II, of the 21st International Conference on Database Systems for Advanced Applications - Volume 9643, pp. 299–313 (2016)
22. Liu, A., Li, Z., Liu, G., Zheng, K., Zhang, M., Li, Q., Zhang, X.: Privacy-preserving task assignment in spatial crowdsourcing. J. Comput. Sci. Technol. **32**(5), 905–918 (2017)
23. Liu, B., Chen, L., Zhu, X., Zhang, Y., Zhang, C., Qiu, W.: Protecting location privacy in spatial crowdsourcing using encrypted data. In: Proceedings of the 20th International Conference on Extending Database Technology, EDBT 2017, Venice, Italy, March 21–24, 2017, pp. 478–481 (2017)
24. Liu, X., Liu, A., Zhang, X., Li, Z., Liu, G., Zhao, L., Zhou, X.: When differential privacy meets randomized perturbation: a hybrid approach for privacy-preserving recommender system. In: International Conference on Database Systems for Advanced Applications, pp. 576–591 (2017)
25. Liu, A., Wang, W., Shang, S., Li, Q., Zhang, X.: Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. GeoInformatica **22**(2), 335–362 (2018)
26. Myerson, R.B.: Optimal auction design. Math. Oper. Res. **6**(1), 58–73 (1981)
27. Naor, M., Pinkas, B.: Computationally secure oblivious transfer. J. Cryptol. **18**(1), 1–35 (2005)
28. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2–6, 1999, Proceeding, pp. 223–238 (1999)
29. Paulet, R., Koasar, M.G., Yi, X., Bertino, E.: Privacy-preserving and content-protecting location based queries. In: 2012 IEEE 28th International Conference on Data Engineering (ICDE), p. 1 (2014)
30. Pournajaf, L., Garcia-Ulloa, D.A., Li, X., Sunderam, V.: Participant privacy in mobile crowd sensing task management: a survey of methods and challenges. ACM Sigmod Record **44**(4), 23–34 (2016)
31. Sun, Y., Liu, A., Li, Z., Liu, G., Zhao, L., Zheng, K.: Anonymity-based privacy-preserving task assignment in spatial crowdsourcing. In: Web Information Systems Engineering - WISE 2017 - 18th International Conference, Puschino, Russia, October 7–11, 2017, Proceedings, Part II, pp. 263–277 (2017)
32. Tham, C.K., Luo, T.: Fairness and social welfare in service allocation schemes for participatory sensing. Comput. Netw. **73**(C), 58–71 (2014)
33. To, H., Ghinita, G., Shahabi, C.: Framework for protecting worker location privacy in spatial crowdsourcing. Proc. VLDB Endow. **7**(10), 919–930 (2014)
34. To, H., Shahabi, C., Li, X.: Privacy-preserving online task assignment in spatial crowdsourcing with untrusted server. In: The IEEE International Conference on Data Engineering (2018)
35. Tong, Y., She, J., Ding, B., Wang, L., Chen, L.: Online mobile micro-task allocation in spatial crowdsourcing. In: IEEE International Conference on Data Engineering, pp. 49–60 (2016)
36. Vu, K., Zheng, R., Gao, J.: Efficient algorithms for k-anonymous location privacy in participatory sensing. In: Proceedings of the IEEE INFOCOM 2012, Orlando, FL, USA, March 25–30, 2012, pp. 2399–2407 (2012)
37. Xiao, Y., Xiong, L.: Protecting locations with differential privacy under temporal correlations. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12–16, 2015, pp. 1298–1309 (2015)
38. Xiong, P., Zhang, L., Zhu, T.: Reward-based spatial crowdsourcing with differential privacy preservation. Enterp. Inf. Syst. **11**(10), 1–18 (2017)
39. Yang, D., Xue, G., Fang, X., Tang, J.: Crowdsourcing to smartphones: incentive mechanism design for mobile phone sensing. MobiCom **2012**, 173–184 (2012)
40. Yao, B., Li, F., Xiao, X.: Secure nearest neighbor revisited. In: 29th IEEE International Conference on Data Engineering, ICDE 2013, Brisbane, Australia, April 8–12, 2013, pp. 733–744 (2013)

41. Yi, X., Paulet, R., Bertino, E., Varadharajan, V.: Practical k nearest neighbor queries with location privacy. In: IEEE 30th International Conference on Data Engineering, Chicago, ICDE 2014, IL, USA, March 31–April 4, 2014, pp. 640–651 (2014)
42. Yi, X., Paulet, R., Bertino, E., Varadharajan, V.: Practical approximate k nearest neighbor queries with location and query privacy. IEEE Trans. Knowl. Data Eng. **28**(6), 1546–1559 (2016)
43. Yiu, M.L., Ghinita, G., Jensen, C.S., Kalnis, P.: Enabling search services on outsourced private spatial data. VLDB J. **19**(3), 363–384 (2010)
44. Zhang, Y., Chen, Q., Zhong, S.: Privacy-preserving data aggregation in mobile phone sensing. IEEE Trans. Inf. Forensics Secur. **11**(5), 980–992 (2016)
45. Zhao, D., Li, X.Y., Ma, H.: How to crowdsource tasks truthfully without sacrificing utility: online incentive mechanisms with budget constraint. In: INFOCOM, 2014 Proceedings IEEE, pp. 1213–1221 (2014)
46. Zheng, L., Chen, L.: Maximizing acceptance in rejection-aware spatial crowdsourcing. In: IEEE International Conference on Data Engineering, pp. 71–72 (2017)
47. Zheng, Y., Cheng, R., Maniu, S., Mo, L.: On optimality of jury selection in crowdsourcing. In: International Conference on Extending Database Technology. Brussels Belgium (2015)
48. Zheng, Y., Wang, J., Li, G., Cheng, R., Feng, J.: Qasca: a quality-aware task assignment system for crowdsourcing applications, pp. 1031–1046 (2015)
49. Zheng, Y., Li, G., Cheng, R.: Docs: a domain-aware crowdsourcing system using knowledge bases. Proc. VLDB Endow. **10**(4), 361–372 (2016)
50. Zheng, Y., Li, G., Li, Y., Shan, C., Cheng, R.: Truth inference in crowdsourcing: is the problem solved? Proc. VLDB Endow. **10**(5), 541–552 (2017)