SRA: Secure Reverse Auction for Task Assignment in Spatial Crowdsourcing

Mingjun Xiao, *Member, IEEE*, Kai Ma, An Liu, *Member, IEEE*, Hui Zhao, Zhixu Li, Kai Zheng, *Member, IEEE*, and Xiaofang Zhou, *Fellow, IEEE*

Abstract—In this paper, we study a new type of spatial crowdsourcing, namely competitive detour tasking, where workers can make detours from their original travel paths to perform multiple tasks, and each worker is allowed to compete for preferred tasks by strategically claiming his/her detour costs. The objective is to make suitable task assignment by maximizing the social welfare of crowdsourcing systems and protecting workers' private sensitive information. We first model the task assignment problem as a reverse auction process. We formalize the winning bid selection of reverse auction as an n-to-one weighted bipartite graph matching problem with multiple 0-1 knapsack constraints. Since this problem is NP-hard, we design an approximation algorithm to select winning bids and determine corresponding payments. Based on this, a Secure Reverse Auction (SRA) protocol is proposed for this novel spatial crowdsourcing. We analyze the approximation performance of the proposed protocol and prove that it has some desired properties, including truthfulness, individual rationality, computational efficiency, and security. To the best of our knowledge, this is the first theoretically provable secure auction protocol for spatial crowdsourcing systems. In addition, we also conduct extensive simulations on a real trace to verify the performance of the proposed protocol.

Index Terms—Privacy, reverse auction, spatial crowdsourcing, task assignment

1 Introduction

The prevalence of smart mobile devices and ubiquitous wireless networks have resulted in the emergence of a new crowdsourcing paradigm, called spatial crowdsourcing [1], [2], [3]. A typical spatial crowdsourcing system consists of a crowd of workers and a platform on the cloud. The platform will publicize a variety of location-relative spatial tasks. Workers can physically move to these locations to perform the corresponding spatial tasks. Since spatial crowdsourcing can accomplish plenty of spatial tasks that individual users cannot cope with, it has stimulated many commercial applications in practice, such as Gigwalk, Waze and Uber.

- *M. Xiao, K. Ma, and H. Zhao are with the School of Computer Science and Technology/Suzhou Institute for Advanced Study, University of Science and Technology of China, Hefei, Anhui 230022, China. E-mail: xiaomj@ustc.edu.cn, {makaizh, huiz16}@mail.ustc.edu.cn.*
- *A. Liu is with the Department of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215000, China, and the Jiangsu Engineering Laboratory of Big Data Intelligence, Suzhou, China. E-mail: anliu@suda.edu.cn.*
- *Z. Li is with the Department of Computer Science and Technology, Soochow University, Suzhou, Jiangsu 215000, China, and IFLYTEK Research (Suzhou), Suzhou, China. E-mail: zhixuli@suda.edu.cn.*
- *K. Zheng is with the Big Data Research Center, University of Electronic Science and Technology of China, Chengdu 611731, China. E-mail: zhengkai@uestc.edu.cn.*
- *X. Zhou is with the School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, QLD 4072, Australia. E-mail: zxf@itee.uq.edu.au.*

Manuscript received 25 Mar. 2018; revised 20 Nov. 2018; accepted 31 Dec. 2018. Date of publication 16 Jan. 2019; date of current version 5 Mar. 2020. (Corresponding author: An Liu.)
Recommended for acceptance by J. Xu.
Digital Object Identifier no. 10.1109/TKDE.2019.2893240

An important research problem in spatial crowdsourcing is to assign spatial tasks to suitable workers [4]. Existing solutions to this problem can be generally divided into two categories according to the underlying task publishing modes: *worker selected tasks* (WST) [5] and *server assigned tasks* (SAT) [6], [7], [8], [9]. In WST mode, workers can choose any task according to their preferences (e.g., choosing the closest task), which does not necessarily coincide with the objective of the crowdsourcing platform (e.g., maximizing the total number of completed tasks). This weakness can be overcome by the SAT mode in which the platform knows all workers' data, and therefore, can assign to workers suitable tasks while achieving its own objective.

In this paper, we investigate task assignment in the SAT model, where a novel task setting, namely *competitive detour tasking*, is considered. Specifically, each worker has a travel path (e.g., the path from home to office) and is willing to detour from the path to perform some tasks nearby. He/she competes for his/her preferred tasks by strategically claiming his/her detour cost. The platform assigns workers to tasks according to their quotations and pays them some rewards to compensate for their detour costs. This task setting has two advantages when compared with traditional ones. First, workers are not constrained to perform tasks that are close to their current locations. Instead, they can choose distant tasks as long as the detour cost could be tolerated. Second, tasks can be solved in a cost-effective way due to the free quotation of users and the competition among their quotations. Therefore, it is useful in practice and can be applied to ridesharing [10], destination-aware spatial

1041-4347 © 2019 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

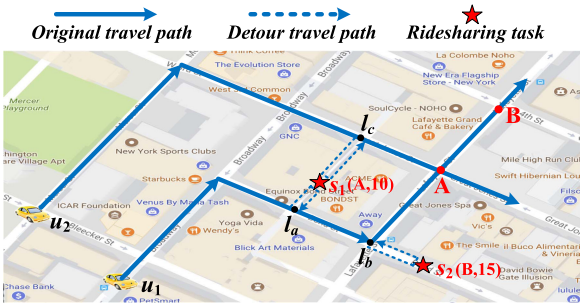


Fig. 1. Example of competitive detour tasking.

crowdsourcing [11], noise pollution monitoring and roadside advertisement collection.

Fig. 1 shows an example of competitive detour tasking. There are two riders and each requests a ridesharing task with a specific budget (e.g., \$10 for task s_1 and \$15 for task s_2) in the crowdsourcing platform. Two drivers (i.e., workers) move along their travel paths (denoted by solid lines) but they can make detours (denoted by dotted lines) to pick riders up at specified locations. Driver u_1 finds the destinations of s_1 and s_2 (i.e., A and B, respectively) are both in his/her travel path but s_1 is in a crowded street, so he/she asks for a high detour cost say \$10 for s_1 and a normal detour cost say \$8 for s_2 based on his/her true travel cost and expected profit. On the other hand, only s_1's destination is in the travel path of u_2, so u_2 is eager to perform s_1 with a competitive detour cost say \$5. Since u_2's quotation is lower than u_1's, the platform selects u_2 to perform s_1 and assigns s_2 to u_1. Clearly, the total income of the platform is $\$10 + \$15 = \$25$ and the total expenditure is $\$5 + \$8 = \$13$, making a profit of \$12. If the platform adopts the conventional distance-first task setting, then both s_1 and s_2 will be assigned to u_1. In this case, the expenditure increases to $\$10 + \$8 = \$18$, so the profit becomes \$7.

Compared to previous task assignment studies, the hardness of competitive detour tasking problem lies in the following two aspects. First, every worker in this problem setting may make multiple detours to perform different tasks but the accumulative detour distance budget is bounded. As a result, the whole task assignment involves a combinatorial optimization problem that combines n-to-one weighted bipartite graph matching and multiple 0-1 knapsack constraints, which is not a trivial bipartite graph matching problem. Second, since workers are allowed to compete for their preferred tasks, designing truthful auction mechanism is one of the most efficient ways to solve this problem. However, the detour costs claimed by workers are sensitive in our problem setting, and should be kept secret in the auction design. This is because a malicious worker can always win an order by deliberately asking for a price lower than other quotations, so honest workers have no chance to perform tasks. Although many auction-based task assignment mechanisms have been proposed for crowdsourcing systems [12], [13], [14], [15], [16], none of them can protect the quotations of workers from being disclosed. On the other hand, although there have been some studies on privacy-preserving task assignment for spatial crowdsourcing [17], [18], [19], they mainly focus on the location privacy, which cannot be applied to our problem setting.

To this end, we propose a Secure Reverse Auction (SRA) protocol to address the above challenges. More specifically, we let the platform conduct the task assignment periodically. Each round of task assignment is formalized as a reverse auction process, which includes a secure winning bid selection problem and a payment computation problem. Since winning bid selection is NP-hard, we propose an approximation algorithm to select winning bids and determine the payments, in which homomorphic encryption is adopted to protect workers' bids (i.e., the detour cost claimed by workers) from being revealed to others. The major contributions in this work are summarized as follows:

1) We transform the problem of competitive detour tasking in spatial crowdsourcing into the problem of designing a secure reverse auction protocol, including secure winning bid selection and secure payment computation. Moreover, the winning bid selection is formalized as an n-to-one weighted bipartite graph matching problem with multiple 0-1 knapsack constraints.
2) We prove that the secure winning bid selection problem is NP-hard, and design a greedy algorithm to select winning bids, by which the SRA protocol can achieve an approximately optimal task assignment solution. We analyze the approximation ratio, and demonstrate that it is an urgent bound.
3) We design the secure payment computation algorithm for the SRA protocol and prove that it makes SRA have the properties of truthfulness and individual rationality, that is, all workers will rationally compete for tasks with their true costs.
4) We prove that the SRA protocol is efficient and secure. To the best of our knowledge, this is the first theoretically provable secure auction protocol for spatial crowdsourcing.

The remainder of the paper is organized as follows. We introduce models, problem, and preliminary in Section 2. The SRA protocol is proposed in Section 3. The theoretical analysis is presented in Section 4. In Section 5, we evaluate the performances of SRA. After reviewing the related work in Section 6, we conclude the paper in Section 7.

2 Model and Problem

2.1 System Model

We consider a typical spatial crowdsourcing system. First, there is a platform receiving spatial tasks from crowdsourcing service requesters. A spatial task is defined as follows:

Definition 2.1 (Spatial Task). *A spatial task, or a* task *for short, is denoted by a triple* $s_j \stackrel{\text{def}}{=} \langle l_j a_j, e_j \rangle$, *where* l_j *is the location in a 2D space where* s_j *needs to be performed,* a_j *is the type of* s_j, *e.g., taking photos, and* e_j *is the reward that the requester of* s_j *is willing to pay to the platform.*

Second, the spatial crowdsourcing system also includes a crowd of mobile workers who can detour from their original paths to perform some spatial tasks using their smartphones. If a worker is ready to perform some spatial tasks, it will send its state information to the platform. Each worker can be identified by its state information:

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

Definition 2.2 (Crowd Worker). *A crowd worker, or a* worker *for short, is denoted by a triple* $u_i \stackrel{\text{def}}{=} \langle L_i, A_i, \delta_i \rangle$, *where* L_i *is the shortest path from his/her current location to his/her destination,* A_i *is the set of task types that he/she can deal with, and* δ_i *is the largest accumulative distance that he/she is willing to detour for performing tasks, called* detour distance budget.

Moreover, we use $\mathcal{S}$ and $\mathcal{U}$ to denote the sets of tasks and workers, respectively. Besides, we define another two notations: detour distance and performable task, as follows.

Definition 2.3 (Detour Distance). *The* detour distance d_{ij} *is the extra travel distance that worker* u_i *detours from his/her path* L_i *to perform task* s_j.

Definition 2.4 (Performable Task). *A task* s_j *is a* performable task *of worker* u_i *if* u_i *can deal with this task and the detour distance is not larger than the detour distance budget. Denote the set of performable tasks of* u_i *as* $\mathcal{S}_i$. *Then, it satisfies* $\mathcal{S}_i = \{s_j | a_j \in A_i, d_{ij} \leq \delta_i, s_j \in \mathcal{S}\}$.

Finally, the spatial crowdsourcing system adopts a periodical task assignment model, defined as follows:

Definition 2.5 (Periodical Task Assignment Model). *The platform continuously receives tasks and periodically assigns them to workers. If a task is not assigned in the current task assignment round, it will be handled in later rounds. Each worker can perform one or more tasks as long as the total detour distance is not larger than his/her budget.*

Remark: Clearly, the detour distance for a task s_j can be easily decided at runtime. However, it is hard to estimate d_{ij} at the start of each round of task assignment, because worker u_i cannot know his/her location in the future (noting that he/she will travel to different places to perform other tasks assigned to him/her). To facilitate later discussion, we first consider the detour distance *in the worst case*, where u_i goes back to his/her original path after completing a task. Here, the original path refers to the path from his/her current location (at the start of one round of task assignment) to his/her destination. It is clear that the detour distance estimated in this way is always larger than the one in practice, as u_i can always follow a shorter path to his/her destination if there exists one. Based on the detour distance in the worst case, our algorithms can be presented much more clearly as this distance is a constant in this circumstance. However, it should be noted that our algorithms can also work well in practice where the detour distance is not a constant (as it will change according to u_i's location). As will be discussed later, we only need to ensure the accumulative detour distance is not larger than the detour distance budget, that is, it does not matter whether the detour distance is a constant or not.

2.2 Security Model

In the course of task assignment, we need to protect each worker's private sensitive information from being revealed to the platform or to other workers. For this privacy-preserving issue, we adopt the well-known and widely-used semi-honest security model [20]. In this model, each participator will follow the whole task assignment protocol, showing the honest aspect. On the other hand, the participator will also try to derive the extra information from the received data, showing the dishonest aspect. The semi-honest model is reasonable since each participator is generally willing to follow the protocol so as to benefit from the task assignment. The privacy under the semi-honest model can formally be defined as follows:

Definition 2.6 (Privacy under Semi-honest Model). *[20]] Suppose that* $\mathcal{F}(x_0, x_1, \ldots, x_n) = (\mathcal{F}_0, \mathcal{F}_1, \ldots, \mathcal{F}_n)$ *is a functionality computed by* $n+1$ *parties jointly, where* x_i *and* $\mathcal{F}_i$ *are the input and output of the* i*th party* ($0 \leq i \leq n$, *where* $i = 0$ *represents the platform and* $1 \leq i \leq n$ *represents* n *workers), both belonging to a prime field* $\mathcal{Z}_q$. *For* $\mathcal{I} = \{i_1, \ldots, i_\kappa\} \subset \{0, \ldots, n\}$, *we let* $\mathcal{F}_\mathcal{I}$ *denote the subsequence* $\mathcal{F}_{i_1}, \ldots, \mathcal{F}_{i_\kappa}$. *Consider a protocol for computing* $\mathcal{F}$. *The view of the* i*th party during an execution of this protocol, denoted as* $VIEW_i$, *is* (x_i, y, m_i) *where* y *represents the outcome of the* i*th party's internal coin tosses (i.e., a random integer) and* m_i *represents the messages that the party has received. In other words,* $VIEW_i$ *is all the data that the* i*th party can observe during the execution of the protocol. Let* $VIEW_\mathcal{I} \stackrel{\text{def}}{=} (\mathcal{I}, VIEW_{i_1}, \ldots, VIEW_{i_\kappa})$. *Then, we say that the protocol privately computes* $\mathcal{F}$ *if there exists a polynomial-time algorithm, denoted as* $\mathcal{A}$, *such that for every* $\mathcal{I}$ *above*

$$\mathcal{A}(\mathcal{I}, (x_{i_1}, \ldots, x_{i_\kappa}, \mathcal{F}_\mathcal{I})) \stackrel{C}{=} VIEW_\mathcal{I}, \tag{1}$$

where $\stackrel{C}{=}$ *denotes computational indistinguishability.*

Eq. (1) asserts that the view of each party in $\mathcal{I}$ can be efficiently simulated based solely on its inputs and outputs. In other words, it cannot derive extra information during the execution of the protocol. In addition, a semi-honest third party is introduced into the crowdsourcing model to assist the platform and workers to complete the task assignment, whose defined as follows:

Definition 2.7 (Semi-honest Agent). *A semi-honest agent is a third party that works in the semi-honest model and provides the service of encryption key generation and some auxiliary computations for the crowdsourcing system.*

Remark: In practice, many public key infrastructures can serve as the semi-honest agent. As a well-known service provider, the agent typically does not collude with either the platform or the workers. This fact has been widely-used in many other secure computation systems, e.g., [20], [21], [22].

2.3 Problem Formalization

In the spatial crowdsourcing system, the platform conducts the task assignment through the manner of reverse auction. Specifically, the platform acts as the auctioneer, and the workers are seen as the sellers of service of performing tasks, as shown in Fig. 2. First, the platform publicizes all tasks in $\mathcal{S}$ to the workers in $\mathcal{U}$. Then, each worker submits the tasks that it can deal with and the corresponding bids (i.e., the claimed detour costs) to the platform. According to these bids, the platform determines the winning bids and computes the payments for winners, based on which the platform conducts the task assignment and pays the rewards. Meanwhile, the agent helps to protect all workers' true costs from being revealed. The whole auction process mainly involves two key problems: the Secure Winning Bid

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

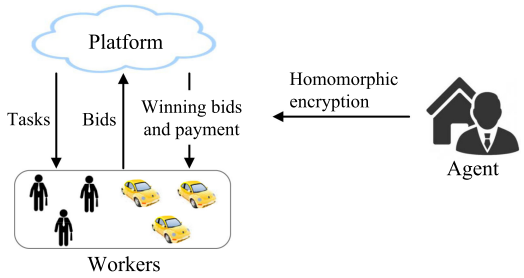


Fig. 2. Privacy-preserving reverse auction model.

Selection (SWBS) problem and the Secure Payment Computation (SPC) problem, which are formalized as follows.

First, we define four basic notations for the auction:

Definition 2.8 (True Cost, Bid, Winning Bid, and Payment). *If a worker $u_i \in \mathcal{U}$ performs a task $s_j \in \mathcal{S}$, he/she will result in a* true (detour) cost, *denoted by c_{ij}. When u_i wants to compete for performing s_j in the auction, he/she will submit a* bid, *denoted by b_{ij}, which is a claimed cost. Since u_i might manipulate the claimed cost, b_{ij} is not necessarily equal to c_{ij}. When u_i wins task s_j in the auction by using the bid b_{ij}, we say that b_{ij} is a* winning bid. *Accordingly, for the winning bid b_{ij}, the platform will determine a* payment, *i.e., the amount of money given to u_i for performing the task, denoted by p_{ij}. We denote all bids, winning bids, and payments as three sets $\mathcal{B}$, $\tilde{\mathcal{B}}$, and $\mathcal{P}$, respectively. Moreover, $c_{ij}, b_{ij}, p_{ij} \in \mathcal{Z}_q$, where q is a prime and $\mathcal{Z}_q$ is a field. Additionally, we let $\rho_i = \sum_{s_j \in \mathcal{S}} p_{ij}$ and $\rho = \sum_{u_i \in \mathcal{U}} \rho_i$.*

Remark: In this paper, the values of c_{ij}, b_{ij}, and p_{ij} are the private sensitive information of worker u_i, which will be protected from being revealed in the whole auction process. $\tilde{\mathcal{B}}$ is the set of winning bids. If $b_{ij} \in \tilde{\mathcal{B}}$, it means that b_{ij} wins the auction and s_j will be assigned to u_i. Thus, $\tilde{\mathcal{B}}$ can also be seen as the solution of the task assignment problem.

Second, the optimization objective of SWBS is to maximize the social welfare, defined as follows.

Definition 2.9 (Social Welfare). *The* social welfare *is the total profit of the whole spatial crowdsourcing system. Let $\tilde{\mathcal{B}}$ be the set of winning bids. Then, social welfare is denoted as:*

$$\Phi(\tilde{\mathcal{B}}) \stackrel{\text{def}}{=} \sum_{b_{ij} \in \tilde{\mathcal{B}}} (e_j - c_{ij}). \tag{2}$$

Remark: Here, if worker u_i wins task s_j in the auction, the social welfare includes not only the profit of the platform, but also the profit of the worker, i.e., $e_j - c_{ij} = (e_j - p_{ij}) + (p_{ij} - c_{ij})$. In practice, the platform and workers might share their profits with the agent as the reward of security service. Here, we just ignore the profit sharing for simplicity, since it will not affect the correctness of our task assignment protocol.

Third, based on the above concepts, we can formalize the SWBS problem as follows.

Definition 2.10 (The SWBS Problem).

$$\begin{aligned} Maximize: \sum_{b_{ij} \in \tilde{\mathcal{B}}} (e_j - c_{ij}) &= \sum_{b_{ij} \in \tilde{\mathcal{B}}} (e_j - b_{ij}) \\ &= \sum_{s_j \in \mathcal{S}} \sum_{u_i \in \mathcal{U}} (e_j - b_{ij}) x_{ij} \end{aligned} \tag{3}$$

$$Subject\ to: \sum_{u_i \in \mathcal{U}} x_{ij} \le 1, \quad s_j \in \mathcal{S}, x_{ij} \in \{0,1\} \tag{4}$$

$$\sum_{s_j \in \mathcal{S}_i} d_{ij} x_{ij} \le \delta_i, \ u_i \in \mathcal{U}, x_{ij} \in \{0,1\} \tag{5}$$

$$Security \quad : \text{Eq. 1 holds.} \tag{6}$$

Here, $x_{ij} = 1$ indicates that bid b_{ij} wins the auction and task s_j will be assigned to worker u_i, i.e., $b_{ij} \in \tilde{\mathcal{B}}$. Otherwise, if $x_{ij} = 0$, s_j will not be assigned to u_i and $b_{ij} \notin \tilde{\mathcal{B}}$. Eq. (4) indicates that each task can be assigned to at most one worker and Eq. (5) means that the total detour distance of each worker is not larger than its budget.

Remarks: Our task assignment protocol, i.e., SRA, is a truthful reverse auction protocol, which means that all workers will always submit the true costs as their bids during the whole auction process. Hence, when discussing the winning bid selection problem, we can directly assume $b_{ij} = c_{ij}$. Then, the optimization objective of the SWBS problem is maximizing the social welfare of the whole system, as shown in Eq. (3). In Section 4, we will prove the truthfulness of SRA, which implies that this assumption holds.

Next, the SPC problem is defined as follows:

Definition 2.11 (The SPC Problem). *The SPC problem is how to determine the payment for each winner so that the whole auction protocol satisfies truthfulness and the individual rationality and the worker's true cost values will not be disclosed to others.*

The concepts of truthfulness and individual rationality are defined as follows:

Definition 2.12 (Truthfulness). *Let b be an arbitrary bid for worker u_i that wins the task s_j, and $p_{ij}(b)$ is the corresponding payment determined by the payment computation algorithm of an auction protocol. Then, if*

$$p_{ij}(b) - c_{ij} \le p_{ij} - c_{ij}, \tag{7}$$

then the auction protocol is truthful. Here, the payment is seen as a function about the bid, and p_{ij} is the payment when u_i claims its true cost as its bid, i.e., $p_{ij} = p_{ij}(c_{ij})$.

Definition 2.13 (Individual Rationality). *For each winning bid b_{ij}, if the corresponding payoff is nonnegative, i.e.,*

$$p_{ij} - c_{ij} \ge 0, \tag{8}$$

then the auction protocol satisfies individual rationality.

Remarks: In Definition 2.12, Eq. (7) can guarantee that each worker claims its cost truthfully, since an untruthful bid will lead to a worse payoff. In Definition 2.13, Eq. (8) shows that each worker can receive a nonnegative payoff if it participates in the auction.

In addition, the whole reverse auction protocol needs to meet the property of computation and communication efficiency, defined as follows:

Definition 2.14 (Computation and Communication Efficiency). *Each round of reverse auction process can*

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

TABLE 1
Description of Major Notations

Variable	Description
s_j, $\mathcal{S}$, $\mathcal{S}_i$	the jth task (Def. 2.1), the set of all tasks, the set of tasks that u_i can deal with (Def. 2.4).
u_i, $\mathcal{U}$	the ith worker (Def. 2.2), and the set of all workers in the auction.
d_{ij}, δ_i	detour distance for u_i performing s_j (Def. 2.3), the detour distance budget of u_i (Def. 2.2).
q, $\mathcal{Z}_q$	a prime and a prime field (Defs. 2.6, 2.8).
e_j, c_{ij}	the reward for completing task s_j (Def. 2.1), and the true cost for u_i performing s_j (Def. 2.8).
b_{ij}, p_{ij}, ρ_i, ρ	the bid of worker u_i competing for task s_j, the corresponding payment, the total payment of u_i, and the total payment of all workers (Def. 2.8).
$\mathcal{B}$, $\tilde{\mathcal{B}}$, $\mathcal{P}$	the sets of all bids, all winning bids, and the corresponding payments. (Def. 2.8).
$E[\cdot]$, $D[\cdot]$	homomorphic encryption function (Def. 2.15) and decryption function.
G, $E[\mathcal{W}]$	a weighted bipartite graph and an ordered set of encrypted edge weights (Def. 3.2).
w_{ij}, $E[w_{ij}]$	the edge weight and the encrypted edge weight in graph G (Def. 3.2).
α, β	random numbers selected from the prime field $\mathcal{Z}_q$ for hiding bids (Def. 3.2).

terminate in a polynomial time with a polynomial communication overhead.

2.4 Preliminary

From the definitions presented in the last section, task assignment through reverse auction mainly involves two basic arithmetic operations: addition and multiplication. To protect sensitive data and enable direct computation, we employ homomorphic encryption to encrypt sensitive data. Note that, in this scheme all kinds of data including real numbers (e.g., bid and payment) are transformed into integers in some prime field.

Definition 2.15 (Homomorphic Encryption). *A homomorphic encryption scheme is a public-key cryptosystem with such a homomorphic property that the "addition" operation can be applied to the encrypted data without decrypting them. Let $\mathcal{Z}_q$ be a prime field, $\otimes$ and $\oplus$ be the multiplication and addition operations in this field, i.e., $x \otimes y \stackrel{\text{def}}{=} xy \bmod q$ and $x \oplus y \stackrel{\text{def}}{=} x + y \bmod q$ for $\forall x, y \in \mathcal{Z}_q$. Then, the homomorphic encryption scheme satisfies:*

$$E[m_1] \otimes E[m_2] = E[m_1 \oplus m_2], \tag{9}$$

where $m_1, m_2 \in \mathcal{Z}_q$ are two plaintexts, and $E[\cdot]$ is the homomorphic encryption operation.

Remarks: Eq. (9) shows that when multiplying the homomorphic encrypted ciphertexts of two messages, we can directly get the ciphertext of the addition of them. In this paper, we adopt the well-known Paillier cryptosystem [23] for this encryption scheme. To facilitate later discussion, for a set of values M, we let $E[M] = \{E[m] | m \in M\}$, that is, the encryption operation is performed on every element $m \in M$. Therefore, $E[M]$ is a set of encrypted values.

For ease of reference, we list main notations in Table 1.

3 The SRA Protocol

In this section, we propose the SRA protocol to solve the task assignment problem in our spatial crowdsourcing system. The SRA protocol mainly includes two algorithms: the Secure Winning Bid Selection (SWBS) algorithm and the Secure Payment Computation (SPC) algorithm. SWBS securely determines the winning bids, each of which corresponds to a task assignment. SPC determines the payment for each winning bid. First, we analyze the complexity of the task assignment problem. Then, we propose SWBS and SPC as the building blocks. Next, based on the algorithms, we design the SRA protocol by using the homogeneous encryption techniques. Finally, we demonstrate the execution of this protocol through an example.

3.1 Problem Hardness Analysis

First, we analyze the complexity of the SWBS problem:

Theorem 3.1. *The SWBS problem is NP-hard.*

Proof. We consider a special case of the SWBS problem, where there is only one worker totally, i.e., $|\mathcal{U}| = 1$. Without loss of generality, we let the worker be u_i. Then, this special problem is determining a subset of tasks $\mathcal{S}' \subseteq \mathcal{S}_i$ so as to maximize $\sum_{s_j \in \mathcal{S}'} (e_j - c_{ij})$, while ensuring $\sum_{s_j \in \mathcal{S}'} d_{ij} \leq \delta_i$. This is equivalent to the 0-1 knapsack problem: given a set of items $\mathcal{S}_i$, each item has a value $e_j - c_{ij}$ and a weight d_{ij}, determining a subset of items to maximize the total value, while ensuring the total weight is not larger than a budget δ_i. This is a well-known NP-hard problem, so the special SWBS problem is also NP-hard. Consequently, the general SWBS problem is at least NP-hard. □

Remark: In the SWBS problem, each constraint of detour distance budget (i.e., Eq. (5)) can be seen as a 0-1 knapsack constraint. The whole problem can be modeled as an n-to-one weighted bipartite graph matching problem with multiple 0-1 knapsack constraints.

3.2 Winning Bid Selection

Due to the NP-hardness of the task assignment problem, we design the Secure Winning Bid Selection (SWBS) algorithm to determine the winning bids. In order to maximize the social welfare, SWBS greedily selects the bid from $\mathcal{B}$ that can produce the largest profit under the constraints of detour distance budgets, until no bids can be selected.

Before the winning bid selection, we construct a weighted bipartite graph, defined as follows:

Definition 3.2 (Bipartite Graph with Ordered and Encrypted Edge Weights). *$G = \{\mathcal{U}, \mathcal{S}, \mathcal{D}, E[\mathcal{W}]\}$ is a weighted bipartite graph, including two separate vertex sets: worker set $\mathcal{U}$ and task set $\mathcal{S}$. In graph G, $\mathcal{D} = \{\delta_i | u_i \in \mathcal{U}\}$ is the set of worker vertex weights, where δ_i is u_i's detour distance budget. $E[\mathcal{W}] = \{E[w_{ij}] | u_i \in \mathcal{U}, s_j \in \mathcal{S}\}$ is an* ordered set of encrypted edge weights, *in which $E[w_{ij}]$ is the* encrypted weight *of edge $\langle u_i, s_j \rangle$, satisfying*

$$w_{ij} \stackrel{\text{def}}{=} \alpha(e_j - b_{ij}) + \beta, \tag{10}$$

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

$$E[w_{ij}] = E[e_j]^{\alpha} \otimes E[b_{ij}]^{-\alpha} \otimes E[\beta], \\ = E[\alpha(e_j - b_{ij}) + \beta], \tag{11}$$

where α and β are two numbers randomly selected from prime field $\mathcal{Z}_q$, and $e_j - b_{ij}$ is the profit of worker u_i performing task s_j. Moreover, $E[w_{ij}]$ is the edge weight encrypted by homogeneous encryption. All $E[w_{ij}]$'s in $E[\mathcal{W}]$ are ranked in the descending orders of w_{ij}.

The SWBS algorithm is conducted on the weighted bipartite graph G. The SWBS algorithm selects the bids that have the largest edge weights within the constraints of detour distance budgets as the winning bids, in turn. Since $E[\mathcal{W}]$ is ranked in the descending orders of edge weights, the SWBS algorithm directly lets the bid that correspond to the first element of the sets $E[\mathcal{W}]$ as the winning bid, if the detour distance constraint is not broken. After each selection, the first element of the set $E[\mathcal{W}]$ will be removed and the corresponding detour distance budget will also be updated. For example, if b_{ij} is selected as the winning bid by SWBS, the first element $E[w_{ij}]$ will be removed from $E[\mathcal{W}]$ and the detour distance budget of worker u_i will become $\delta_i - d_{ij}$ from δ_i. Such selection processes will be repeatedly conducted until $E[\mathcal{W}]$ becomes an empty set. Finally, we will get a set of winning bids as the result, denoted by $\tilde{\mathcal{B}}$.

The detailed SWBS algorithm is shown in Algorithm 1. In Step 1, the solution is initialized. The largest edge weight is determined in Step 3. If the corresponding task assignment does not break the constraint of detour distance budget, the related bid b_{ij} is selected as a winning bid in Step 5. Accordingly, the sets of vertices, edges, and weights are updated in Steps 6-8. Otherwise, if the detour distance constraint is broken, this weight will be removed in Step 10. When $E[\mathcal{W}]$ becomes an empty set, the algorithm will terminate to produce the solution.

Algorithm 1. Secure Winning Bid Selection

Input: $G = \{\mathcal{U}, \mathcal{S}, \mathcal{D}, E[\mathcal{W}]\}$
Output: $E[\tilde{\mathcal{B}}]$
1: Initialize: $E[\tilde{\mathcal{B}}] \leftarrow \emptyset$;
2: **while** $E[\mathcal{W}] \neq \emptyset$ **do**
3: $E[w_{ij}] \leftarrow$ the first element in $E[\mathcal{W}]$;
4: **if** $d_{ij} \leq \delta_i$ **then**
5: $E[\tilde{\mathcal{B}}] \leftarrow E[\tilde{\mathcal{B}}] + \{E[b_{ij}]\}$;
6: $\mathcal{S} \leftarrow \mathcal{S} - \{s_j\}$;
7: $E[\mathcal{W}] \leftarrow E[\mathcal{W}] - \{E[w_{ij}] | u_i \in \mathcal{U}\}$;
8: $\delta_i \leftarrow \delta_i - d_{ij}$;
9: **else**
10: $E[\mathcal{W}] \leftarrow E[\mathcal{W}] - \{E[w_{ij}]\}$;

Remarks: Note that the SWBS algorithm is conducted on the graph G, where the edge weight and corresponding bid have been encrypted by homomorphic encryption operations in advance. Thus, the solution produced by SWBS is actually a set of encrypted wining bids, i.e., $E[\tilde{\mathcal{B}}]$, as shown in Algorithm 1. Despite this, it will not affect the correctness of the SWBS algorithm. This is because $E[b_{ij}] \in E[\tilde{\mathcal{B}}]$ can still indicate that b_{ij} is a winning bid, although we cannot derive the true value of b_{ij} from $E[b_{ij}]$.

Algorithm 2. Secure Payment Computation

Input: $G = \{\mathcal{U}, \mathcal{S}, \mathcal{D}, E[\mathcal{W}]\}$, $E[\tilde{\mathcal{B}}]$
Output: $E[\mathcal{P}] = \{E[p_{i^*j^*}] | E[b_{i^*j^*}] \in E[\tilde{\mathcal{B}}]\}$
1: **for** each $E[b_{i^*j^*}] \in E[\tilde{\mathcal{B}}]$ **do**
2: $E[\mathcal{W}'] \leftarrow E[\mathcal{W}] - \{E[w_{i^*j^*}]\}$;
3: **while** $E[\mathcal{W}'] \neq \emptyset$ **do**
4: $E[w_{ij}] \leftarrow$ the first element in $E[\mathcal{W}']$;
5: **if** $d_{ij} \leq \delta_i$ **then**
6: $E[b_{ij}] \leftarrow E[e_j] \otimes E[w_{ij}]^{-\frac{1}{\alpha}} \otimes E[\frac{\beta}{\alpha}]$;
7: **if** $j^* = j$ **then**
8: $E[p_{i^*j^*}] \leftarrow E[b_{ij}]$;
9: **break**;
10: **if** $i^* = i$ and $d_{i^*j^*} > \delta_i - d_{ij}$ **then**
11: $E[p_{i^*j^*}] \leftarrow E[e_{j^*}] \otimes E[e_j]^{-1} \otimes E[b_{ij}]$;
12: **break**;
13: $\mathcal{S} \leftarrow \mathcal{S} - \{s_j\}$;
14: $E[\mathcal{W}'] \leftarrow E[\mathcal{W}'] - \{E[w_{ij}] | u_i \in \mathcal{U}\}$;
15: $\delta_i \leftarrow \delta_i - d_{ij}$;
16: **else**
17: $E[\mathcal{W}'] \leftarrow E[\mathcal{W}'] - \{E[w_{ij}]\}$;

3.3 Payment Computation

The payment computation algorithm is to determine the payment for each winning bid, ensuring that each worker honestly claims its true costs. According to the well-known statement by Myerson [24], in order to guarantee the truthfulness, each winning bid should be paid with a critical payment:

Definition 3.3 (Critical Payment). *A payment p is said to be critical value of a bid b_{ij} if bid b_{ij} can win the auction when $b_{ij} \leq p$ and b_{ij} will lose the auction when $b_{ij} > p$.*

According to Definition 3.3, the critical payment of a bid is equal to the largest bid value that still makes the worker win the corresponding task in the auction, that is, it is equal to the smallest bid value by which the worker will lose the corresponding task in the auction. This value can be determined by using an alternative bid defined as follows:

Definition 3.4 (Alternative Bid). *The alternative bid of a winning bid $b_{i^*j^*}$ is such a bid that will replace $b_{i^*j^*}$ to become a winning bid when we remove $b_{i^*j^*}$ from $\mathcal{B}$.*

We design the Secure Payment Computation (SPC) algorithm to determine the alternative bid of a winning bid and compute the corresponding critical payment. Consider a given weighted bipartite graph $G = \{\mathcal{U}, \mathcal{S}, \mathcal{D}, E[\mathcal{W}]\}$ and a winning bid $b_{i^*j^*} \in \tilde{\mathcal{B}}$, where $E[\mathcal{W}]$ is an ordered set of encrypted edge weights. Then, SPC determines the corresponding encrypted critical payment $E[p_{i^*j^*}]$ as follows.

First, we consider the winning bid selection without the bid $b_{i^*j^*}$. By removing edge $\langle u_{i^*}, s_{j^*} \rangle$ from G, we get a weighted bipartite graph without $b_{i^*j^*}$, denoted by G':

$$E[\mathcal{W}'] = E[\mathcal{W}] - \{E[w_{i^*j^*}]\}, \tag{12}$$

$$G' = \{\mathcal{U}, \mathcal{S}, \mathcal{D}, E[\mathcal{W}']\}. \tag{13}$$

Then, we conduct the greedy winning bid selection algorithm over G' to get a solution $E[\tilde{\mathcal{B}}']$. The alternative bid of $b_{i^*j^*}$ must belong to $\tilde{\mathcal{B}}'$. We consider two cases: the task s_{j^*} is assigned to another worker u_i, or the worker u_{i^*} has win

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

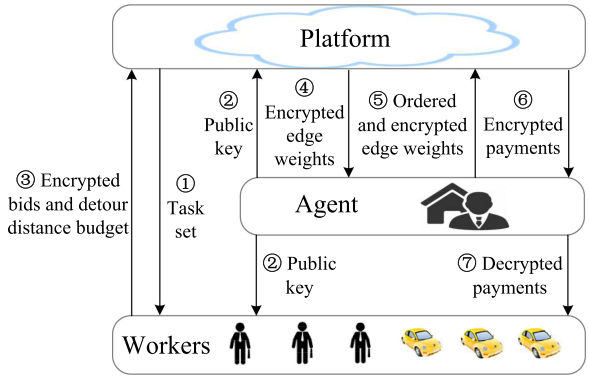


Fig. 3. SRA: Interactions among platform, agent and workers.

some tasks so that it has no budget to compete for task s_{j^*}. For the first case, b_{ij^*} becomes a winning bid in $\tilde{\mathcal{B}}'$ and it is exactly a candidate alternative bid of $b_{i^*j^*}$. For the second case, we assume that $b_{i^*j_1}$, $b_{i^*j_2}$, ..., $b_{i^*j_k} \in \tilde{\mathcal{B}}'$ are winning bids that correspond to worker u_{i^*}. Then, there must be a bid $b_{i^*j'}$ satisfying:

$$w_{i^*j'} = \min\{w_{i^*j_1}, \ldots, w_{i^*j_k} : \delta_{i^*} - \sum_{w_{i^*j_h} > w_{i^*j'}} d_{i^*j_h} \geq d_{i^*j^*}\}, \quad (14)$$

$$b_{i^*j'} = e_{j'} - \frac{w_{i^*j'} - \beta}{\alpha}. \quad (15)$$

Here, $b_{i^*j'}$ is exactly another candidate alternative bid of $b_{i^*j^*}$. Moreover, if $w_{ij^*} \geq w_{i^*j'}$, b_{ij^*} will become the alternative bid of $b_{i^*j^*}$. Otherwise, if $w_{ij^*} < w_{i^*j'}$, the alternative bid of $b_{i^*j^*}$ will be $b_{i^*j'}$.

Actually, we can only determine an encrypted alternative bid for $b_{i^*j^*}$ and the encrypted critical payment. First, we determine $E[w_{ij^*}]$ and $E[w_{i^*j'}]$ after a round of scan in $E[\mathcal{W}'_P]$. Then, based on $E[w_{ij^*}]$ and $E[w_{i^*j'}]$, we compute the encrypted candidate alternative bids and the encrypted critical payment $E[p_{i^*j^*}]$ as follows:

$$E[b_{ij^*}] = E[e_{j^*}] \otimes E[w_{ij^*}]^{-\frac{1}{\alpha}} \otimes E[\frac{\beta}{\alpha}]; \quad (16)$$

$$E[b_{i^*j'}] = E[e_{j'}] \otimes E[w_{i^*j'}]^{-\frac{1}{\alpha}} \otimes E[\frac{\beta}{\alpha}]; \quad (17)$$

$$E[p_{i^*j^*}] = \begin{cases} E[b_{ij^*}], & \text{if } w_{ij^*} \geq w_{i^*j'}; \\ E[e_{j^*}] \otimes E[e_{j'}]^{-1} \otimes E[b_{i^*j'}], & \text{else.} \end{cases} \quad (18)$$

The detailed SPC algorithm is shown in Algorithm 2, and the solution is denoted by $E[\mathcal{P}]$. For each winning bid $b_{i^*j^*}$, the weighted bipartite graph G' is constructed in Step 2. From Step 3 to 17, the greedy winning bid selection algorithm is conducted over G'. The encrypted candidate alternative bid and the corresponding critical payment for the first case is determined in Steps 7-9. The critical payment that corresponds to the second candidate alternative bid is determined in Steps 10-12.

3.4 The Detailed SRA Protocol

We design the SRA protocol based on the SWBS algorithm (i.e., Algorithm 1) and the SPC algorithm (i.e., Algorithm 2). In SRA, we introduce a semi-honest agent to provide the homomorphic encryption service, by which each worker encrypts his/her bids and uses the encrypted bids to participate in the auction. The platform applies SWBS to determine the encrypted winning bids and uses SPC to compute the encrypted payments. The interactions among the platform, agent, and workers in the SRA protocol are presented in Protocol 3 and are also illustrated in Fig. 3.

Protocol 3 The SRA Protocol

Input: Platform: $\mathcal{S}$; Workers: $\mathcal{U}$; Agent: $E[\cdot]$, $D[\cdot]$
Output: Platform: ρ; Workers: $\{\rho_i | u_i \in \mathcal{U}\}$

1: The platform publicizes task set $\mathcal{S}$ to the workers in $\mathcal{U}$.
2: The agent creates a pair of public and private keys of homogeneous encryption, i.e., $E[\cdot]$, $D[\cdot]$, and sends $E[\cdot]$ to the platform and the workers.
3: After receiving $\mathcal{S}$ from the platform and $E[\cdot]$ from the agent, each worker $u_i \in \mathcal{U}$ computes the detour distance for each task in $\mathcal{S}$, and determines the set of performable tasks $\mathcal{S}_i$. For each task $s_j \in \mathcal{S}_i$, u_i produces a bid b_{ij} and a random number σ_{ij}. Next, u_i encrypts them to get $E[b_{ij}]$ and $E[\sigma_{ij}]$. Then, u_i sends $\{\langle s_j, d_{ij}, E[b_{ij}], E[\sigma_{ij}]\rangle | s_j \in \mathcal{S}_i\}$ and the detour distance budget δ_i to the platform.
4: After receiving encrypted bids from workers, the platform randomly selects two numbers α, $\beta \in \mathcal{Z}_q$, and constructs the bipartite graphs with encrypted edge weights $G = \{\mathcal{U}, \mathcal{S}, \mathcal{D}, E[\mathcal{W}]\}$, where each encrypted edge weight $E[w_{ij}]$ in $E[\mathcal{W}]$ satisfies Eq. (11). Then, it sends the encrypted edge weights $E[\mathcal{W}]$ to the agent.
5: When receiving $E[\mathcal{W}]$ from the platform, the agent decrypt these encrypted edge weights: $w_{ij} = D[E[w_{ij}]]$ for each $E[w_{ij}] \in E[\mathcal{W}]$. Then, the agent ranks $E[\mathcal{W}]$ in the descending order of edge weight w_{ij}. Finally, the agent sends the ordered sets $E[\mathcal{W}]$ back to the platform.
6: The platform conducts Algorithms 1 and 2 to produce $\tilde{\mathcal{B}}$ and $E[\mathcal{P}]$. If $E[b_{ij}] \in E[\tilde{\mathcal{B}}]$, the platform sends x_{ij}=1 to worker u_i. Otherwise, x_{ij}=0 is sent to worker u_i. Moreover, the platform computes $E[p'_{ij}] = E[p_{ij}] \otimes E[\sigma_{ij}]^{-1}$ and sends this hidden payment to the agent. Finally, the platform sends $E[\sigma] = \otimes_{E[p_{ij}] \in E[\mathcal{P}]} E[\sigma_{ij}]$ to the agent.
7: When receiving each encrypted and hidden payment $E[p'_{ij}]$ and $E[\sigma]$, the agent decrypts it to get all p'_{ij} and σ. Then, it computes the hidden payment $\sum_{s_j \in \mathcal{S}} x_{ij} p'_{ij}$ and sends it to each u_i. Each u_i gets its payment by computing $\rho_i = \sum_{s_j \in \mathcal{S}} x_{ij}(p'_{ij} \oplus \sigma_{ij})$. Moreover, the agent sends the total payment $\rho = \sum p'_{ij} \oplus \sigma$ to the platform.

3.5 Example

The key parts of SRA are using Algorithms 1 and 2 to select winning bids and compute the corresponding payments. To better understand the two algorithms, we use an example, as shown in Fig. 4, to illustrate the winning bid selection and payment computation procedures. The example includes two tasks and three workers: $\mathcal{U} = \{u_1, u_2\}$ and $\mathcal{S} = \{s_1, s_2, s_3\}$. The reward of each task, the encrypted bids, detour distances, and detour distance budgets of each worker are listed in Fig. 4a. The corresponding weighted bipartite graph $G = \{\mathcal{U}, \mathcal{S}, \mathcal{D}, E[\mathcal{W}]\}$ with encrypted edge weights is shown in Fig. 4b, where $\mathcal{D} = \{\delta_1 = 5, \delta_2 = 6\}$ and the ordered set of edge weights is $E[\mathcal{W}] = \{E[12], E[10], E[9], E[8], E[6], E[5]\}$. Here, we let $\alpha = 1$ and $\beta = 0$, i.e., $E[w_{ij}] = E[e_j - b_{ij}]$, to simplify the

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

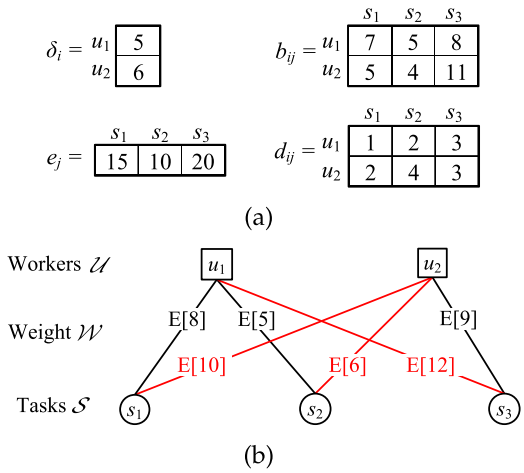


Fig. 4. Illustration of SRA.

presentation of the example. The winning bid selection and payment computation are conducted on the graph G:

Winning Bid Selection. The platform determines the winning bids in graph G through three rounds of greedy selections. In the first round, the platform selects $E[b_{13}]$ as a winning bid since $E[w_{13}]=E[12]$ is the first element of $E[\mathcal{W}]$, which implies that w_{13} is the largest edge weight, and the detour distance satisfies $d_{13}<\delta_1$. After this selection, edge $\langle u_1, s_3\rangle$ is removed from G and the graph becomes $G=\{\mathcal{U},\mathcal{S}=\{s_1,s_2\},\mathcal{D}=\{\delta_1=2,\delta_2=\ \ 6\},E[\mathcal{W}]=\{E[10],E[8],E[6],E[5]\}\}$. In the second round, $E[b_{21}]$ is selected as a winning bid since $E[w_{21}]=E[10]$ is the first element of $E[\mathcal{W}]$ and the detour distance satisfies $d_{21}<\delta_2$. After that, the graph becomes $G=\{\mathcal{U},\mathcal{S}=\{s_2\},\mathcal{D}=\ \{\delta_1=2,\delta_2=4\},E[\mathcal{W}]=\{E[6],E[5]\}\}$. Likewise, in the third round, $E[b_{22}]$ is selected as a winning bid. As a result, the winning bids are $E[b_{13}]$, $E[b_{21}]$, $E[b_{22}]$, and the social welfare is 28.

Payment Computation. To compute the critical payment of the winning bid b_{13}, we first remove the edge $\langle u_1, s_3\rangle$ from bipartite graph G to get $G'=\{\mathcal{U},\mathcal{S}=\{s_1,s_2\},\mathcal{D}=\{\delta_1=5,\delta_2=6\},E[\mathcal{W}]=\{E[10],E[9],E[8],E[6],E[5]\}\}$. Then, the platform conducts Algorithm 1 on the graph G' to get a new solution $E[b_{21}]$, $E[b_{23}]$, $E[b_{12}]$. Since $w_{23}=9>5=w_{12}$, $E[b_{23}]$ becomes the alterative bid of $E[b_{13}]$. Thus, the payment of winning bid b_{13} is $E[p_{13}]=E[b_{23}]=E[11]$. Similarly, the payments of bids b_{21} and b_{22} can be computed as $E[p_{21}]=E[b_{11}]=E[7]$ and $E[p_{22}]=E[b_{12}]=E[5]$.

4 Theoretical Analysis

In this section, we analyze the approximation performance of SRA. Moreover, we prove that SRA can achieve the desired properties of truthfulness, individual rationality, computation and communication efficiency, and security.

4.1 Truthfulness and Individual Rationality

First, we prove that the SRA protocol is truthful, so that each worker will honestly claim its true costs as the bids. According to Myerson's theorem [24], an auction protocol is truthful if and only if the two conditions hold: (1) the winning bid selection is monotonic; (2) each winning bid is paid with a critical value. Based on this theorem, we analyze the truthfulness of SRA as follows.

Lemma 4.1. *The winning bid selection in Algorithm 1 is monotonic. Specifically, for each worker $u_i\in\mathcal{U}$ and task $s_j\in\mathcal{S}_i$, if u_i wins the task s_j by using a bid b_{ij}, then a smaller bid $b'_{ij}<b_{ij}$ can still win the auction.*

Proof. Without loss of generality, we assume that b_{ij} is selected as a winning bid in the kth loop of Algorithm 1 and let the corresponding encrypted edge weight in graph G be $E[w_{ij}]=E[\alpha(e_j-b_{ij})+\beta]$. Now, if worker u_i claims a smaller bid b'_{ij} for task s_j, the corresponding edge weight will become $E[w'_{ij}]=E[\alpha(e_j-b'_{ij})+\beta]$. Since $b'_{ij}<b_{ij}$, we can get $w'_{ij}>w_{ij}$. According to the greedy winner selection strategy in Algorithm 1, the edge with the weight $E[w'_{ij}]$ will be selected in the kth or an even earlier iteration. Thus, bid b'_{ij} will still win the auction. The lemma holds. □

Lemma 4.2. *Algorithm 2 will produce a critical payment $p_{i^*j^*}$ for each winning bid $b_{i^*j^*}$. If worker u_{i^*} claims a bid no larger than $p_{i^*j^*}$ for task s_{j^*}, u_{i^*} will win the task; otherwise, $b_{i^*j^*}$ will lose the auction.*

Proof. According to the payment computation scheme in Algorithm 2, the corresponding payment satisfies Eq. (18). Then, we have

$$p_{i^*j^*}=\begin{cases} b_{ij^*}, & \text{if } w_{ij^*}\geq w_{i^*j'};\\ e_{j^*}-e_{j'}+b_{i^*j'}, & \text{if } w_{ij^*}<w_{i^*j'}.\end{cases} \tag{19}$$

Here, we ignore the encryption operations since we only discuss the truthfulness. Moreover, b_{ij^*} and $b_{i^*j'}$ are the candidate alternative bids of $b_{i^*j^*}$ when we remove the bid $b_{i^*j^*}$, which satisfy Eqs. (16) and (17).

First, we assume that worker u_{i^*} claims a larger bid $b>p_{i^*j^*}$ for task s_{j^*}. Then, if $w_{ij^*}\geq w_{i^*j'}$, we have $b>p_{i^*j^*}=b_{ij^*}$. As a result, $w_{i^*j^*}=\alpha(e_{j^*}-b)+\beta<\alpha(e_{j^*}-b_{ij^*})+\beta=w_{ij^*}$. According to the greedy winning bid selection strategy in Algorithm 1, b_{ij^*} will be selected prior to b. This means that bid b loses the auction. Otherwise, if $w_{ij^*}<w_{i^*j'}$, we have $b>p_{i^*j^*}=e_{j^*}-e_{j'}+b_{i^*j'}$. Then, $w_{i^*j^*}=\alpha(e_{j^*}-b)+\beta<\alpha(e_{j'}-b_{i^*j'})+\beta=w_{i^*j'}$. This means that $b_{i^*j'}$ will be selected prior to b. According to Eq. (14), b cannot be selected after $b_{i^*j'}$ any more due to the constraint of detour distance budget. Therefore, bid b still loses the auction.

Second, we assume that u_{i^*} claims a bid $b\leq p_{i^*j^*}$ for task s_{j^*}. Like the first case, we have $w_{i^*j^*}\geq w_{ij^*}$ if $w_{ij^*}\geq w_{i^*j'}$ or $w_{i^*j^*}\geq w_{i^*j'}$ if $w_{ij^*}<w_{i^*j'}$. That is, $w_{i^*j^*}\geq\max\{w_{ij^*},w_{i^*j'}\}$. According to the greedy winning bid selection strategy, b will be selected prior to b_{ij^*} and $b_{i^*j'}$. Since b_{ij^*} and $b_{i^*j'}$ are also prior to the remaining bids corresponding to u_{i^*} and s_{j^*}, b will be selected first and becomes the winning bid.

Based on the two cases, we can conclude that $p_{i^*j^*}$ is exactly the critical value. The lemma holds. □

Theorem 4.3. *The SRA protocol is truthful.*

Proof. Lemmas 4.1 and 4.2 show that the winning bid selection of SRA is monotonic and each winning bid is paid with a critical value. Thus, SRA is truthful according to [24]. □

Theorem 4.4. *The SRA protocol meets the condition of individual rationality.*

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

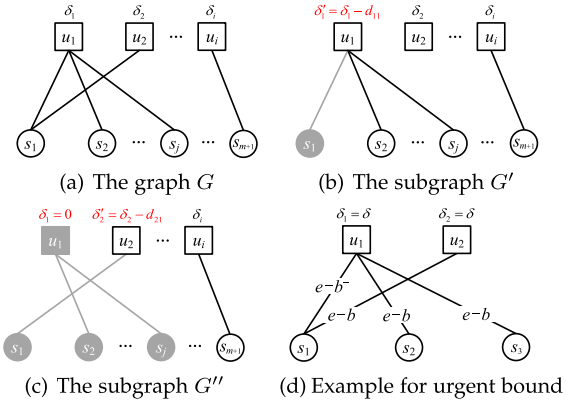


Fig. 5. Illustration for the approximation ratio analysis.

Proof. We consider an arbitrary winning bid $b_{i^*j^*}$. Like the proof of Lemma 4.2, we can ignore the encryption operations to derive the payment of bid $b_{i^*j^*}$ from Eq. (18). As a result, the payment $p_{i^*j^*}$ satisfies Eq. (19), in which b_{ij^*} and $b_{i^*j'}$ are the alternative bids of $b_{i^*j^*}$. First, we consider the case where $w_{ij^*} \geq w_{i^*j'}$. Since $b_{i^*j^*}$ is a winning bid, we have $w_{i^*j^*} = \alpha(e_{j^*} - b_{i^*j^*}) + \beta \geq \alpha(e_{j^*} - b_{ij^*}) + \beta = w_{ij^*}$. As a result, $b_{i^*j^*} \leq b_{ij^*} = p_{i^*j^*}$. Second, we consider the case where $w_{ij^*} < w_{i^*j'}$. We also have $w_{i^*j^*} = \alpha(e_{j^*} - b_{i^*j^*}) + \beta \geq \alpha(e_{j'} - b_{i^*j'}) + \beta = w_{i^*j'}$. Then, we can get $b_{i^*j^*} \leq e_{j^*} - e_{j'} + b_{i^*j'} = p_{i^*j^*}$. Thus, we have $b_{i^*j^*} \leq p_{i^*j^*}$ for both cases. Further, due to the truthfulness of the SRA protocol, we have $b_{i^*j^*} = c_{i^*j^*}$. Therefore, $p_{i^*j^*} \geq c_{i^*j^*}$. The theorem holds. □

4.2 The Approximation Performance Analysis

In SRA protocol, Algorithm 1 is used to select winning bids, which can achieve an approximately optimal solution for the task assignment problem. We analyze the approximation ratio as follows.

Theorem 4.5. *Denote the optimal solution of task assignment as $\tilde{\mathcal{B}}_F$, the approximation ratio γ of the solution $\tilde{\mathcal{B}}$ produced by the SRA protocol satisfies:*

$$\gamma = \frac{\Phi(\tilde{\mathcal{B}}_F)}{\Phi(\tilde{\mathcal{B}})} \leq \lambda + 1, \tag{20}$$

where $\lambda = \max\{\frac{\delta_i}{d_{ij}} | u_i \in \mathcal{U}, s_j \in \mathcal{S}_i\}$. Moreover, $\lambda+1$ is a tight bound.

Proof. We prove the approximation ratio $\gamma \leq \lambda + 1$ by using the mathematical induction method. First, when $|\mathcal{S}| = 1$, it is straightforward for $\tilde{\mathcal{B}}_F = \tilde{\mathcal{B}}$ and $\gamma = 1 < \lambda + 1$. Second, we assume that $\gamma \leq \lambda + 1$ holds when $|\mathcal{S}| \leq m$ and consider the case where $|\mathcal{S}| = m+1$. Without loss of generality, we assume that $e_1 - b_{11} = \max\{e_j - b_{ij} | b_{ij} \in \mathcal{B}\}$. Then, we have $w_{11} = \alpha(e_1 - b_{11}) + \beta = \max\{w_{ij} | u_i \in \mathcal{U}, s_j \in \mathcal{S}\}$. According to the greedy winning bid selection strategy in Algorithm 1, b_{11} must belong to the winning bid set $\tilde{\mathcal{B}}$. Now, we consider the following two sub-cases:

In the first sub-case, b_{11} also belongs to the optimal solution $\tilde{\mathcal{B}}_F$. Removing vertex s_1 from the graph G, we can get a subgraph $G' = \{\mathcal{U}, \mathcal{S}', \mathcal{D}', E[\mathcal{W}']\}$, where $\mathcal{S}' = \mathcal{S} - \{s_1\}$, $\mathcal{D}' = \mathcal{D} - \{\delta_1\} + \{\delta'_1 = \delta_1 - d_{11}\}$, and $E[\mathcal{W}'] = E[\mathcal{W}] - \{E[w_{11}]\}$, as shown in Fig. 5b. After running the greedy and optimal winning bid selection strategies on graph G', we can get the corresponding solutions. We denote them as $\tilde{\mathcal{B}}|_{G'}$ and $\tilde{\mathcal{B}}_F|_{G'}$, respectively. Then, we have $\Phi(\tilde{\mathcal{B}}) = \Phi(\tilde{\mathcal{B}}|_{G'}) + (e_1 - b_{11})$ and $\Phi(\tilde{\mathcal{B}}_F) = \Phi(\tilde{\mathcal{B}}_F|_{G'}) + (e_1 - b_{11})$. Note that $|\mathcal{S}'| \leq m$. According to the assumption of the induction, we can get $\Phi(\tilde{\mathcal{B}}_F|_{G'}) \leq (\lambda+1)\Phi(\tilde{\mathcal{B}}|_{G'})$. Thus, we have:

$$\gamma = \frac{\Phi(\tilde{\mathcal{B}}_F)}{\Phi(\tilde{\mathcal{B}})} = \frac{\Phi(\tilde{\mathcal{B}}_F|_{G'}) + (e_1 - b_{11})}{\Phi(\tilde{\mathcal{B}}|_{G'}) + (e_1 - b_{11})} \leq \lambda + 1. \tag{21}$$

In the second sub-case, b_{11} does not belong to the optimal solution $\tilde{\mathcal{B}}_F$. Without loss of generality, we assume that some other bids from worker u_1 are selected in $\tilde{\mathcal{B}}_F$, denoted as a set $\tilde{\mathcal{B}}_F^{(1)} = \{b_{1j} | b_{1j} \in \tilde{\mathcal{B}}_F\}$, and task s_1 is assigned to another worker u_2, i.e., $b_{21} \in \tilde{\mathcal{B}}_F$. Since $e_1 - b_{11} = \max\{e_j - b_{ij} | b_{ij} \in \mathcal{B}\}$, we have:

$$\Phi(\tilde{\mathcal{B}}_F^{(1)}) = \sum_{b_{1j} \in \tilde{\mathcal{B}}_F^{(1)}} (e_j - b_{1j}) \leq \lambda(e_1 - b_{11}). \tag{22}$$

Next, we construct two subgraphs of G: $G' = \{\mathcal{U}, \mathcal{S}', \mathcal{D}', E[\mathcal{W}']\}$ and $G'' = \{\mathcal{U}, \mathcal{S}'', \mathcal{D}'', E[\mathcal{W}'']\}$, where $\mathcal{S}' = \mathcal{S} - \{s_1\}$, $\mathcal{D}' = \mathcal{D} - \{\delta_1\} + \{\delta'_1 = \delta_1 - d_{11}\}$, $E[\mathcal{W}'] = E[\mathcal{W}] - \{E[w_{11}]\}$, $\mathcal{S}'' = \mathcal{S} - \{s_1\} - \{s_j | b_{1j} \in \tilde{\mathcal{B}}_F\}$, $\mathcal{D}'' = \mathcal{D} - \{\delta_1, \delta_2\} + \{\delta'_2 = \delta_2 - d_{21}\}$, and $E[\mathcal{W}''] = E[\mathcal{W}] - \{E[w_{11}], E[w_{21}]\} - \{E[w_{1j}] | b_{1j} \in \tilde{\mathcal{B}}_F^{(1)}\}$, as shown in Figs. 5b and 5c. Then, we have:

$$\Phi(\tilde{\mathcal{B}}) = \Phi(\tilde{\mathcal{B}}|_{G'}) + (e_1 - b_{11}), \tag{23}$$

$$\Phi(\tilde{\mathcal{B}}_F) = \Phi(\tilde{\mathcal{B}}_F|_{G''}) + \Phi(\tilde{\mathcal{B}}_F^{(1)}) + (e_1 - b_{21}). \tag{24}$$

Note that G'' is also a subgraph of G'. Thus, we can get $\Phi(\tilde{\mathcal{B}}_F|_{G''}) \leq \Phi(\tilde{\mathcal{B}}_F|_{G'})$. Further, because $|\mathcal{S}'| \leq m$. According to the assumption of the induction, we can get $\Phi(\tilde{\mathcal{B}}_F|_{G'}) \leq (\lambda+1)\Phi(\tilde{\mathcal{B}}|_{G'})$. Therefore, we have:

$$\Phi(\tilde{\mathcal{B}}_F|_{G''}) \leq (\lambda+1)\Phi(\tilde{\mathcal{B}}|_{G'}). \tag{25}$$

Now, according to Eqs. (22), (23), (24), and 25 and $e_1 - b_{11} = \max\{e_j - b_{ij} | b_{ij} \in \mathcal{B}\}$, we have:

$$\begin{aligned} \gamma &= \frac{\Phi(\tilde{\mathcal{B}}_F)}{\Phi(\tilde{\mathcal{B}})} = \frac{\Phi(\tilde{\mathcal{B}}_F|_{G''}) + \Phi(\tilde{\mathcal{B}}_F^{(1)}) + (e_1 - b_{21})}{\Phi(\tilde{\mathcal{B}}|_{G'}) + (e_1 - b_{11})} \\ &\leq \frac{(\lambda+1)\Phi(\tilde{\mathcal{B}}|_{G'}) + \lambda(e_1 - b_{11}) + (e_1 - b_{21})}{\Phi(\tilde{\mathcal{B}}|_{G'}) + (e_1 - b_{11})} \\ &\leq \lambda + 1. \end{aligned} \tag{26}$$

In addition, we show that $\lambda+1$ is a tight bound through an example. Consider a special case, where $\mathcal{U} = \{u_1, u_2\}$, $\mathcal{S} = \{s_1, s_2, s_3\}$, $\mathcal{D} = \{\delta_1 = \delta, \delta_2 = \delta\}$, $d_{11} = \delta$, $d_{12} = \frac{\delta}{2}, d_{13} = \frac{\delta}{2}, d_{21} = \delta$, $e_1 = e_2 = e_3 = e$, and $\mathcal{B} = \{b_{11} = b^-, b_{12} = b, b_{13} = b, b_{21} = b\}$, where b^- is a number smaller than b but infinitely close to b, as shown in Fig. 5d. For this example, we have $\lambda = \frac{\delta_1}{d_{12}} = 2$, $\tilde{\mathcal{B}}_F = \{b_{12}, b_{13}, b_{21}\}$, and $\tilde{\mathcal{B}} = \{b_{11}\}$. Then, $\gamma = \frac{3(e-b)}{e-b^-}$ is infinitely close to $\lambda+1 = 3$. Thus, $\lambda+1$ is a tight bound. The theorem holds. □

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

4.3 Efficiency

We analyze the computational efficiency of SRA as follows.

Theorem 4.6. *The SRA protocol has a polynomial-time computation complexity and a polynomial-time communication overhead.*

Proof. The computation overhead of SRA is dominated by Steps 5 and 6. In Step 5, all encrypted edge weights in $E[\mathcal{W}]$ are decrypted and ranked. The corresponding overhead is $O(|\mathcal{W}|)$ decryption operations and $O(|\mathcal{W}|\log|\mathcal{W}|)$ comparison operations, where $|\mathcal{W}|$ is the cardinal number of set $\mathcal{W}$. In Step 6, Algorithms 1 and 2 are conducted by the platform to determine the winning bids and compute critical payments. The computation overhead is dominated by Step 6 of Algorithm 2, i.e., $O(|\mathcal{W}||\tilde{\mathcal{B}}|)$ multiplication operations on ciphertexts. Since $O(|\mathcal{B}|)=O(|\mathcal{W}|)$ and $O(|\mathcal{B}|)=O(|\tilde{\mathcal{B}}|)$, the total computation overhead of SRA is $O(|\mathcal{B}|^2)$ multiplication operations on ciphertexts. In addition, the communication overhead of SRA mainly includes $E[b_{ij}], E[\sigma_{ij}]$ in Step 3, $E[w_{ij}]$ in Steps 4-5, and $E[p'_{ij}]$ in Step 6, which is $O(|\mathcal{B}|)$ ciphertexts of homomorphic encryption. Therefore, the theorem holds. □

4.4 Security

Finally, we prove that the SRA protocol is secure against any semi-honest adversaries.

Theorem 4.7. *The SRA protocol can protect the bid values of each worker from being revealed to any other semi-honest workers, the platform, and the agent.*

Proof. According to Definition 2.6, we construct three simulators SP, SA, and SW for the platform, the agent, and an arbitrary worker u_i such that their views can be efficiently simulated by the outputs of the simulators SP, SA, and SW. That is to say, the outputs of the simulators and the views are computational indistinguishable.

Denote the views of the platform, the agent, and worker u_i as $VIEW_P$, $VIEW_A$, and $VIEW_{u_i}$. Then, according to the SRA protocol, these views can be represented as follows:

$$VIEW_P = \big(\alpha, \beta, E[b_{ij}], E[\sigma_{ij}], E[w_{ij}]\big), \tag{27}$$

$$VIEW_A = \big(D[\cdot], E[p'_{ij}], E[\sigma], E[w_{ij}]\big), \tag{28}$$

$$VIEW_{u_i} = \big(\sigma_{ij}, \rho_i\big), \tag{29}$$

where $D[\cdot]$ is the input of the agent, α, β are the internal coin tosses of the platform, σ_{ij} is the coin toss of worker u_i, and the others are the messages received by the three parts during the execution of the SRA protocol. Here, we ignore the public messages such as $\mathcal{U}$, $\mathcal{S}$, $E[\cdot]$, d_{ij}, and δ_i for simplicity.

Simulator SP randomly selects two numbers α', β', and three numbers b'_{ij}, σ'_{ij}, w'_{ij} from the prime field $\mathcal{Z}_q$ for each b_{ij}. By using the public homomorphic encryption key, SP creates $E[b'_{ij}]$, $E[\sigma'_{ij}]$, $E[w'_{ij}]$, and then, SP outputs $(\alpha', \beta', E[b'_{ij}], E[\sigma'_{ij}], E[w'_{ij}])$. Because both α, β and α', β' are the numbers randomly selected from $\mathcal{Z}_q$ and all of $E[b_{ij}]$, $E[\sigma_{ij}]$, $E[w_{ij}]$, $E[b'_{ij}]$, $E[\sigma'_{ij}]$, and $E[w'_{ij}]$ are the ciphertexts of the homomorphic encryption $E[\cdot]$, the outputs of simulator SP and $VIEW_P$ are computational indistinguishable. Likewise, simulator SA randomly selects the numbers p''_{ij}, σ', w'_{ij} from the prime field $\mathcal{Z}_q$ and outputs $(D[\cdot], E[p''_{ij}], E[\sigma'], E[w'_{ij}])$ by using the input $D[\cdot]$ and homomorphic encryption operations. As a result, the outputs of simulator SA and $VIEW_A$ are also computational indistinguishable. In addition, simulator SW also randomly selects the numbers σ'_{ij} from the prime field $\mathcal{Z}_q$ and directly outputs (σ'_{ij}, ρ_i), where ρ_i is the output of the SRA protocol. Since both σ_{ij} and σ'_{ij} are randomly selected from $\mathcal{Z}_q$, the outputs of simulator SW and $VIEW_{u_i}$ are also computational indistinguishable. Thus, the theorem holds. □

5 Evaluation

5.1 Algorithms in Comparison

Since the task assignment of spatial crowdsourcing is generally an NP-Hard problem, most of existing works adopt greedy selection strategies to assign tasks, such as the algorithms in [17], [25]. However, the spatial crowdsourcing models and problems in these works are different from ours. The existing algorithms cannot be used for comparison directly. In order to evaluate the task assignment performance of SRA, we tailor the basic idea in these algorithms for our model and carefully design two task assignment algorithms for comparison: the task assignment algorithm based on Social welfare per Detour distance (SD), and the Spatial-First task assignment (SF) algorithm.

Like the SRA protocol, the two algorithms are also conducted on the weighted bipartite graph G, where the detour distances of each worker for performing tasks are considered. For each round of task assignment, SD greedily selects the bid that has the largest profit per detour distance, i.e., $\max\{\frac{e_i-b_{ij}}{d_{ij}}|u_i\in\mathcal{U}, s_j\in\mathcal{S}\}$, within the constraints of detour distance budgets. In contrast, the SF algorithm treats the detour distance as a kind of cost and greedily selects the bid with the smallest detour distance in each round of task assignment.

5.2 Simulation Parameters and Settings

We adopt a widely-used real world dataset in [26] to conduct the simulations. The dataset contains approximately 320 mobility traces of taxi cabs collected over 30 days in Rome, Italy. Each trace is represented by a sequence of GPS coordinates with time labels, which are collected about every 7 seconds. From these records, we directly extract the original travel path of each taxi cab in each day to form a travel map. In our simulations, we first randomly select 10 days of the mobility traces. Then, we see each taxi cab in the traces on different days as different candidate mobile workers. Particularly, a taxi cab with 10 days of mobility traces is seen as 10 mobile workers with the corresponding mobility traces. Finally, we randomly select a group of taxi cabs from the original dataset to form the worker set $\mathcal{U}$. The number of workers $|\mathcal{U}|$ is set as 1000, 1500, 2000, 2500, and 3000, respectively.

Since there are no tasks in this dataset, we randomly deploy a number of tasks, each of which can be performed by as least two workers. First, we divide the time into

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

TABLE 2
Evaluation Settings

Parameter name	Values
number of workers $\|\mathcal{U}\|$	1000, 1500, **2000**, 2500, 3000
number of tasks $\|\mathcal{S}\|$	1000, 1500, **2000**, 2500, 3000
ratio of average detour distance budget $\frac{\delta}{\|L\|}$	0.1, 0.2, **0.3**, 0.4, 0.5
ratio of average detour distance $\frac{d}{\delta}$	0.1, 0.2, **0.3**, 0.4, 0.5
range of bids	[1,50],**[1,100]**,[1,150],[1,200],[1,250]
range of reward	[10,30],**[30,50]**,[50,70],[70,90], [90,110]

equal-length time intervals. Each time interval is equal to a reverse auction cycle and is set as one hour. Then, we derive the length of the travel path of each worker u_i in each auction cycle, denoted by $\|L_i\|$. The average path length of all workers is denoted as $\|L\|$. Next, we denote the average detour distance budget of all workers in $\mathcal{U}$ as δ. Moreover, we let $\frac{\delta}{\|L\|} = 0.1$, 0.2, 0.3, 0.4, and 0.5, and call it *the ratio of average detour distance budget*. For each worker u_i , we let δ_i be randomly selected from $[\frac{\delta}{2}, \frac{3\delta}{2}]$. After that, we define another simulation parameter, i.e., *the ratio of average detour distance*, and denote it by $\frac{d}{\delta}$, where $\frac{d}{\delta}$ is set as 0.1, 0.2, 0.3, 0.4, and 0.5, respectively. Finally, we randomly produce all tasks in $\mathcal{S}$, where *the number of tasks* $|\mathcal{S}|$ is set as 1000, 1500, 2000, 2500, and 3000, respectively. Here, if the detour distance of a worker u_i performing a task s_j satisfies $d_{ij} < \frac{\delta_i d}{\delta}$, we say that the task s_j is covered by the worker u_i. When a randomly produced task is covered by less than two workers, we will reproduce the task.

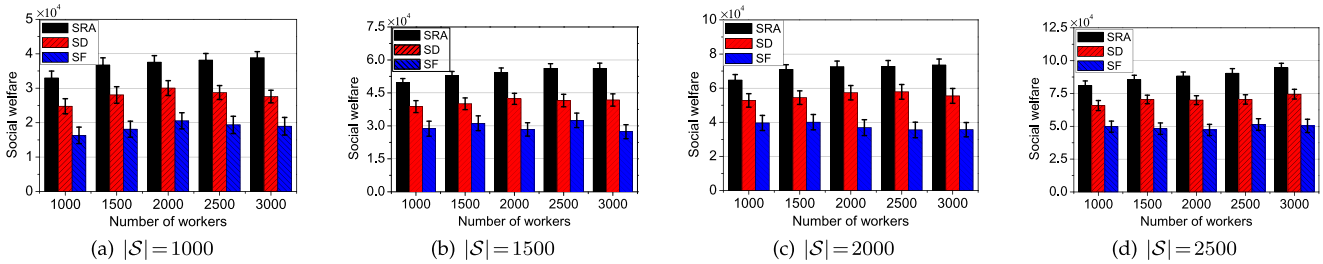
In addition, we set *the range of task reward* $[e_{min}, e_{max}]$ as [10, 30], [30, 50], [50, 70], [70, 90], and [90, 110], respectively. The reward of each task is randomly selected from these ranges. Each bid b_{ij} is randomly selected from *the range of workers' bids* $[b_{min}, b_{max}]$, which is set as [1, 50], [1, 100], [1, 150], [1, 200], and [1, 250], respectively. The units of rewards, bids, costs, and payments are all assumed to be dollar. Moreover, all simulation parameters are listed in Table 2, where default values are in bold fonts. Each simulation is conducted 1,000 times. The average social welfare value and the average running time are recorded for comparison.

5.3 Evaluation on Social Welfare

We evaluate the effects of the number of workers $|\mathcal{U}|$ and the number of tasks $|\mathcal{S}|$ on the social welfare. The results are shown in Figs. 6 and 7.

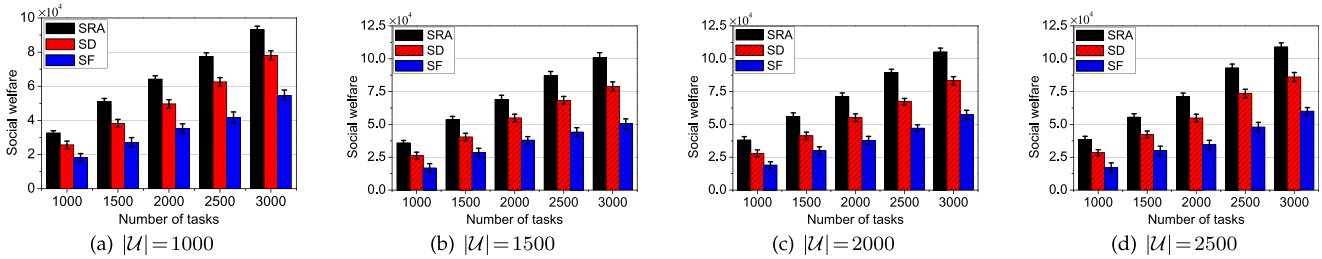
First, we can find that our SRA protocol achieves the highest social welfare, while the SF algorithm obtains the smallest social welfare. This is because SF only takes the distance between the workers and tasks into consideration, which leads to a higher probability of selecting the workers with large bids. SRA outperforms SD due to the following reasons. The SRA protocol directly chooses the bids who can bring the highest social welfare, while the SD algorithm prefers the bids with large social welfare per detour distance. The latter outperforms the former only in some special cases where there are many such bids that the incurred profits are very large but the profits per detour distance are small. However, the bids, rewards, and detour distances are randomly produced, and each simulation is conducted 1,000 times. The probability that the above-mentioned special cases appear is very low. Thus, as an average result of 1,000 times of simulations, SRA outperforms SD significantly.

Second, when the number of workers is increased from 1000 to 3000, the social welfare of SRA increases slightly but steadily, as shown in Figs. 6a, 6b, 6c, and 6d. This is because when we keep the tasks unchanged and let more candidate workers emerge, there are also more bids with higher social welfare, leading to a better selection than before. In contrast, the social welfare values of SD and SF change irregularly. For the SD algorithm, the increasing workers will not result in the increase of the probability of the above-mentioned special cases. In some cases, the probability might decrease. Thus, the social welfare value of SD changes irregularly. As



(a) $|\mathcal{S}| = 1000$ (b) $|\mathcal{S}| = 1500$ (c) $|\mathcal{S}| = 2000$ (d) $|\mathcal{S}| = 2500$

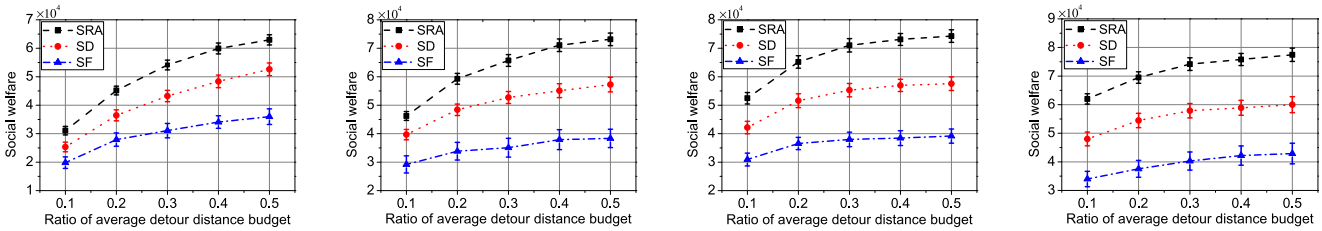
Fig. 6. Social welfare versus the number of workers $|\mathcal{U}|$.



(a) $|\mathcal{U}| = 1000$ (b) $|\mathcal{U}| = 1500$ (c) $|\mathcal{U}| = 2000$ (d) $|\mathcal{U}| = 2500$

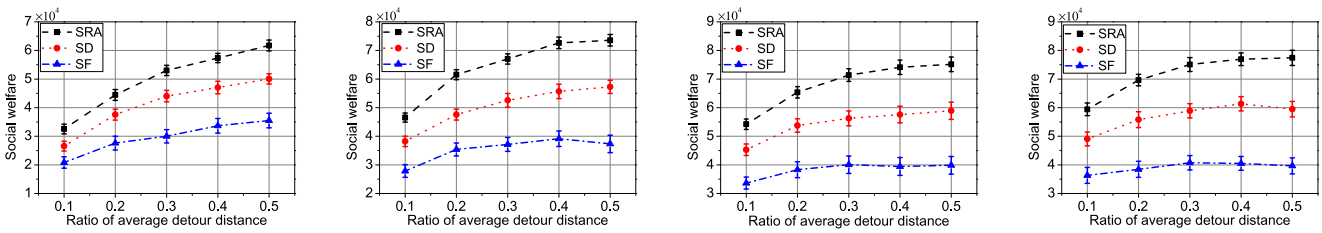
Fig. 7. Social welfare versus the number of tasks $|\mathcal{S}|$.

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.



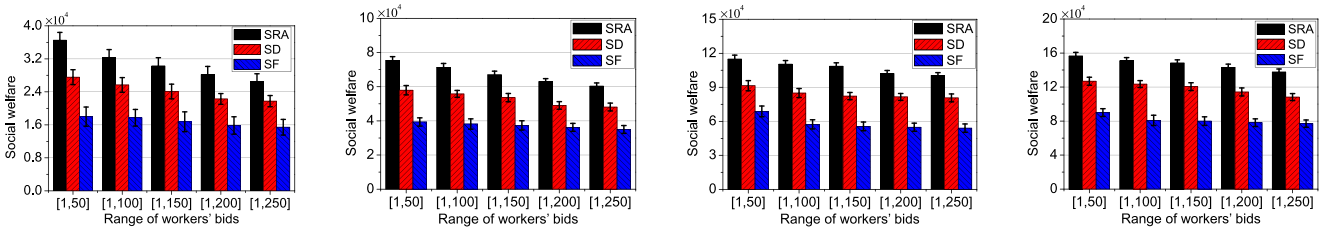
(a) $\frac{d}{\delta}=0.1$ (b) $\frac{d}{\delta}=0.2$ (c) $\frac{d}{\delta}=0.3$ (d) $\frac{d}{\delta}=0.4$

Fig. 8. Social welfare versus the ratio of average detour distance budget $\frac{\delta}{\|L\|}$.



(a) $\frac{\delta}{\|L\|}=0.1$ (b) $\frac{\delta}{\|L\|}=0.2$ (c) $\frac{\delta}{\|L\|}=0.3$ (d) $\frac{\delta}{\|L\|}=0.4$

Fig. 9. Social welfare versus the ratio of average detour distance $\frac{d}{\delta}$.



(a) $[e_{min}, e_{max}]=[10,30]$ (b) $[e_{min}, e_{max}]=[30,50]$ (c) $[e_{min}, e_{max}]=[50,70]$ (d) $[e_{min}, e_{max}]=[70,90]$

Fig. 10. Social welfare versus the range of workers' bids $[b_{min}, b_{max}]$.

for SF , increasing the number of workers does not affect the social welfare monotonically, since SF only focuses on the bids with small detour distances.

Third, when we increase the number of tasks $|\mathcal{S}|$ from 1000 to 3000, SRA, SD, and SF all produce increasing social welfare values, as shown in Figs. 7a, 7b, 7c, and 7d. This is because there are more selections for each worker along with the increase of the number of tasks. Consequently, SRA and SD can select the tasks with larger social welfare values for each worker. Likewise, SF can select the tasks with shorter detour distance for each worker, so that more tasks can be accomplished and higher social welfare can be achieved.

5.4 Evaluation on Detour Distance

In order to evaluate the effect of the detour distance on social welfare, we report the simulation results of social welfare with different ratios of average detour distance budget $\frac{\delta}{\|L\|}$ and different ratios of average detour distance $\frac{d}{\delta}$, as shown in Figs. 8 and 9, respectively.

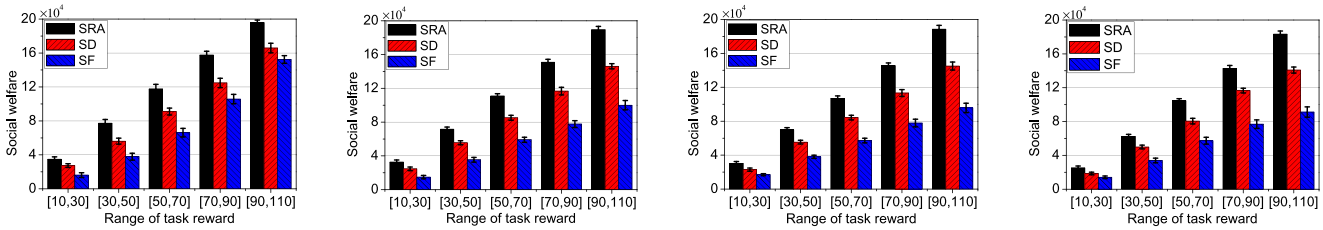
From Fig. 8, we can find that when ratio $\frac{\delta}{\|L\|}$ increases, larger social welfare values are achieved by SRA, SD and SF. The reason is that the detour distance budget of each worker will increase along with $\frac{\delta}{\|L\|}$, which will not only bring more tasks to the unsaturated workers, but also increase the number of available workers for some tasks. Therefore, the fact that more tasks may be assigned and better workers may be selected increases the social welfare values. Additionally, owing to the fixed number of tasks, the growth rate of the social welfare values is getting slower.

In Fig. 9, when ratio $\frac{d}{\delta}$ increases from 0.1 to 0.5, the trend of social welfare of SRA is similar to the results caused by increasing $\frac{\delta}{\|L\|}$. The reason is that, with the increasing $\frac{d}{\delta}$, the number of tasks covered by each worker is also no less than before. This means that more tasks may be performed and more workers with lower bids may be selected, so higher social welfare will be obtained.

5.5 Evaluation on Workers' Bids and Task Rewards

We evaluate the effect of the range of workers' bids $[b_{min}, b_{max}]$ on social welfare by changing the range of task rewards from [10,30] to [70,90] and setting other parameters to their default values, as shown in Figs. 10a, 10b, 10c, and 10d. When we change the range $[b_{min}, b_{max}]$ from [1,50] to [1,200], there are more and more workers with higher bids according to the random distribution of workers' bids. Therefore, SRA, SD and SF have to select more workers with higher bids as the winners, which leads to smaller social welfare values. Moreover, compared to SD and SF, SRA has a relatively larger decrease on the social welfare values. This is because the social welfare performance of SRA is more sensitive than those of SD and SF on the average bid value.

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.



(a) $[b_{min}, b_{max}] = [1, 50]$ (b) $[b_{min}, b_{max}] = [1, 100]$ (c) $[b_{min}, b_{max}] = [1, 150]$ (d) $[b_{min}, b_{max}] = [1, 200]$

Fig. 11. Social welfare versus the range of tasks' rewards $[e_{min}, e_{max}]$.

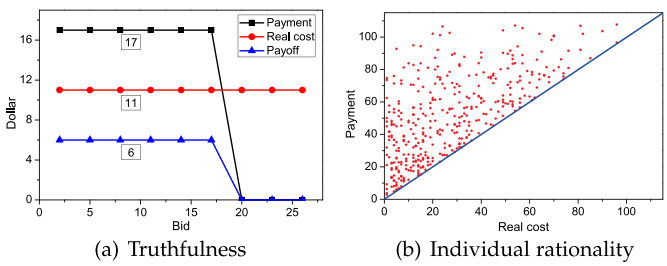
Fig. 11 reports the social welfare values obtained by SRA, SD and SF with different task rewards $[e_{min}, e_{max}]$. Similar to the change of bids, the rewards of all tasks increase when we change the task reward ranges from [10,30] to [70,90]. Consequently, large social welfare values are achieved, as shown in Figs. 11a, 11b, 11c, and 11d. Additionally, when we fix the task reward range and increase the range of workers' bids, the social welfare values of the three approaches all decrease, which coincides with the previous results about the effect of workers' bids.

5.6 Evaluation on Truthfulness & Individual Rationality

To verify the truthfulness of SRA, we randomly choose a worker and allow it to claim some bid values different from its real cost. The results are shown in Fig. 12a, where the critical payment is \$17. When the claimed bid value is not larger than the critical payment \$17, the payment and payoff remain unchanged, which are \$17 and \$17-11=6, respectively. Otherwise, if the claimed bid value exceeds the critical payment, the corresponding payoff and payment become zero. This means that the worker cannot improve its payoff by claiming a false cost. Thus, the SRA protocol is truthful. To verify the individual rationality, we randomly select a number of winning bids and depict them in Fig. 12b according to the corresponding real cost and payment values. The results show that each payment is higher than the corresponding cost, which means that SRA has the property of individual rationality.

5.7 Evaluation on Efficiency

We run SRA in a desktop with 3.2 GHZ CPU and 4 GB RAM. The number of workers changes from 500 to 1000 and the number of tasks changes from 50 to 500, while all other parameters are set as default values. As shown in Fig. 13, the running time of SRA increases slowly when the number of workers and tasks increase. Furthermore, when the number of workers and tasks are 1000 and 500, respectively, the running time is less than 150s. Hence, SRA can work efficiently in real applications.



(a) Truthfulness (b) Individual rationality

Fig. 12. Truthfulness and individual rationality of SRA.

6 Related Work

Crowdsourcing utilizes the capabilities of crowd to deal with computer-hard tasks. It is challenging as different workers may need different times and costs to do the same task, and their answers may have different qualities. Therefore, a large number of work on crowdsourcing has been reported recently, trying to achieve high quality answers in a cost-effective and efficient way [27], [28], [29], or infer the truth based on workers' answers [30], [31], [32]. In this section, we review related studies on a special kind of crowdsourcing, namely spatial crowdsourcing, from three aspects: task assignment, incentive mechanism, and privacy protection.

In [6], Kazemi and Shahabi propose several heuristics to maximize the number of assigned tasks in a given time interval while meeting the constraints specified by workers. Deng et al. [5] devise both exact and approximation algorithms to find a schedule for a worker such that the number of performed tasks by the worker is maximized. Spatial-temporal diversity and reliability are taken into account in the course of task assignment. [7] shows task assignment with these constraints is NP-hard and proposes several approximation algorithms. In [8], efficient methods are designed to assign workers to complex tasks that require more than one skill. In practice, tasks often arrives dynamically. This kind of online scenarios is more challenging and has been addressed in [33], [34], [35] where efficient algorithms with provable competitive ratio are proposed. Song et al. in [25] extend conventional task assignment from two objects matching problem to trichromatic matching problem. In [36], spatial distribution of workers and tasks are taken into account when maximizing a global assignment quality score. [37] tackles the problem of assigning tasks to workers such that mutual benefit are maximized. Our work differs from the above studies in that we take into account both competition and security requirements.

In order to stimulate workers to compete for tasks with low costs so that the overall profit of crowdsourcing

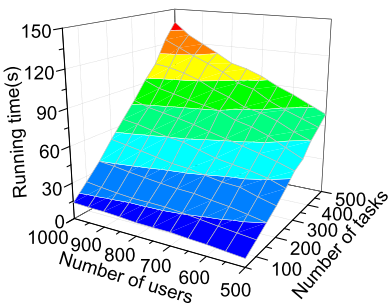


Fig. 13. Running time versus the numbers of users and tasks.

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

systems could be improved, many incentive mechanisms, such as auction-based task assignment strategies, have been reported in recent research efforts. In [12], data quality is introduced into the design of incentive mechanism, in particular, the payments given to workers depend on how well they perform tasks. In [13], the authors propose a double auction interaction process among service users and service providers in dynamic mobile crowdsourcing systems. A special task assignment problem, binary labeling, is studied in [14] and a reverse auction mechanism is proposed to maximize the platform's utility. [38] proposes a match-based approach to solve the dynamic pricing problem in spatial crowdsourcing. Though competitiveness of workers are considered in these achievements, none of them can protect the private information of workers during auction, failing to meet the security requirement in crowdsourcing.

During crowdsourcing workers are required to report their data to the untrusted crowdsourcing systems. Some kinds of data, such as locations and bids, are sensitive information and should be kept secret. It is therefore important to achieve privacy-preserving during crowdsourcing. In [17], a trusted party collects workers' locations and constructs a private spatial decompositions (PSD) according to differential privacy. The PSD is then given to crowdsourcing systems which can assign tasks effectively based on some well-designed strategies. This idea is extended to online scenario where multiple PSDs are generated for dynamic workers [18]. To improve the effectiveness of task assignment, workers' velocities are also considered in [19] so that tasks can be assigned by travel time other than travel distance. In [39], the authors adopt secret sharing to design a privacy-preserving user recruitment protocol for spatial crowdsourcing. Though security issue has been addressed in these studies, competition among workers is still ignored. By overcoming this weakness, our work can effectively improve workers' enthusiasm to perform crowdsourcing tasks and the overall utility of crowdsourcing systems.

7 Conclusion

In this paper, we have studied secure task assignment problem in the competitive detour tasking scenario where each worker can make multiple detours from its original path to perform some spatial tasks. We have formalized this problem as an n-to-one weighted bipartite graph matching problems with multiple 0-1 knapsack constraints, and we have proposed a protocol named SRA to solve this task assignment problem. We have analyzed its approximation performance and proved that SRA not only has the properties of truthfulness, individual rationality, computation and communication efficiency, but also can protect the private information of workers from being revealed to others. Extensive simulations have been carried out on a real trace to show the performance of our protocol.

Acknowledgments

This research was supported in part by the National Natural Science Foundation of China (NSFC) (Grant No. 61872330, 61572336, 61572457, 61632016, 61379132, 61532018, 61836007, 61832017, U1709217), the Natural Science Foundation of Jiangsu Province in China (Grant No. BK20131174, BK2009150), the Natural Science Research Project of Jiangsu Higher Education Institution (No. 18KJA520010, 17KJA520003), and the Anhui Initiative in Quantum Information Technologies (Grant No. AHY150300).

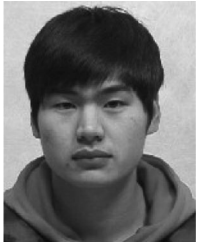
References

[1] G. Li, J. Wang, Y. Zheng, and M. J. Franklin, "Crowdsourced data management: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 9, pp. 2296–2319, Sep. 2016.
[2] Y. Tong, L. Chen, and C. Shahabi, "Spatial crowdsourcing: challenges, techniques, and applications," *Proc. VLDB Endowment*, vol. 10, no. 12, pp. 1988–1991, 2017.
[3] G. Li, Y. Zheng, J. Fan, J. Wang, and R. Cheng, "Crowdsourced data management: Overview and challenges," in *Proc. ACM Int. Conf. Manage. Data*, 2017, pp. 1711–1716.
[4] L. Chen and C. Shahabi, "Spatial crowdsourcing: Challenges and opportunities," *IEEE Data Eng. Bull.*, vol. 39, no. 4, pp. 14–25, Dec. 2016.
[5] D. Deng, C. Shahabi, U. Demiryurek, and L. Zhu, "Task selection in spatial crowdsourcing from worker's perspective," *GeoInformatica*, vol. 20, no. 3, pp. 529–568, 2016.
[6] L. Kazemi and C. Shahabi, "Geocrowd: Enabling query answering with spatial crowdsourcing," in *Proc. 20th Int. Conf. Advances Geographic Inf. Syst.*, 2012, pp. 189–198.
[7] P. Cheng, X. Lian, Z. Chen, R. Fu, L. Chen, J. Han, and J. Zhao, "Reliable diversity-based spatial crowdsourcing by moving workers," *Proc. VLDB Endowment*, vol. 8, no. 10, pp. 1022–1033, 2015.
[8] P. Cheng, X. Lian, L. Chen, J. Han, and J. Zhao, "Task assignment on multi-skill oriented spatial crowdsourcing," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2201–2215, Aug. 2016.
[9] M. Xiao, J. Wu, L. Huang, R. Cheng, and Y. Wang, "Online task assignment for crowdsensing in predictable mobile social networks," *IEEE Trans. Mobile Comput.*, vol. 16, no. 8, pp. 2306–2320, Aug. 2017.
[10] P. Cheng, H. Xin, and L. Chen, "Utility-aware ridesharing on road networks," in *Proc. ACM Int. Conf. Manage. Data*, 2017, pp. 1197–1210.
[11] Y. Zhao, Y. Li, Y. Wang, B. Zheng, H. Su, and K. Zheng, "Destination-aware task assignment in spatial crowdsourcing," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1–10.
[12] D. Peng, F. Wu, and G. Chen, "Pay as how well you do: A quality based incentive mechanism for crowdsensing," in *Proc. 16th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2015, pp. 177–186.
[13] Y. Wei, Y. Zhu, H. Zhu, Q. Zhang, and G. Xue, "Truthful online double auctions for dynamic mobile crowdsourcing," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2074–2082.
[14] Q. Zhang, Y. Wen, X. Tian, X. Gan, and X. Wang, "Incentivize crowd labeling under budget constraint," in *Proc. IEEE Conf. Comput. Commun.*, 2015, pp. 2812–2820.
[15] H. Zhang, B. Liu, H. Susanto, G. Xue, and T. Sun, "Incentive mechanism for proximity-based mobile crowd service systems," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun.*, 2016, pp. 1–9.
[16] G. Gao, M. Xiao, J. Wu, L. Huang, and C. Hu, "Truthful incentive mechanism for nondeterministic crowdsensing with vehicles," *IEEE Trans. Mobile Comput.*, vol. 17, no. 12, pp. 2982–2997, Dec. 2018.
[17] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proc. VLDB Endowment*, vol. 7, no. 10, pp. 919–930, 2014.
[18] H. To, G. Ghinita, L. Fan, and C. Shahabi, "Differentially private location protection for worker datasets in spatial crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 934–949, Apr. 2017.
[19] A. Liu, W. Wang, S. Shang, Q. Li, and X. Zhang, "Efficient task assignment in spatial crowdsourcing with worker and task privacy protection," *GeoInformatica*, vol. 22, no. 2, pp. 1–28, 2018.
[20] O. Goldreich, *Foundations of Cryptography: Volume 2 - Basic Applications*. Cambridge, U. K.: Cambridge Univ. Press, 2004.
[21] C. Dong, L. Chen, and Z. Wen, "When private set intersection meets big data: An efficient and scalable protocol," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2013, pp. 789–800.
[22] T. Araki, J. Furukawa, Y. Lindell, A. Nof, and K. Ohara, "High-throughput semi-honest secure three-party computation with an honest majority," in *Proc. ACM SIGSAC Conf. Comput. Commun. Security*, 2016, pp. 805–817.

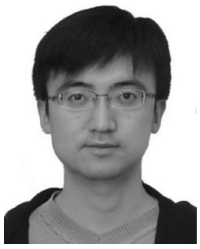
Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.

[23] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. 17th Int. Conf. Theory Appl. Cryptographic Techn.*, 1999, pp. 223–238.
[24] R. B. Myerson, "Optimal auction design," *Mathematics Operations Res.*, vol. 6, no. 1, pp. 58–73, 1981.
[25] T. Song, Y. Tong, L. Wang, J. She, B. Yao, L. Chen, and K. Xu, "Trichromatic online matching in real-time spatial crowdsourcing," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 1009–1020.
[26] "CRAWDAD dataset roma/taxi (v. 2014–07-17)," Jul. 2014. [Online]. Available: https://crawdad.org/roma/taxi/20140717
[27] Y. Zheng, J. W. G. Li, R. Cheng, and J. Feng, "QASCA: A quality-aware task assignment system for crowdsourcing applications," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, 2015, pp. 1031–1046.
[28] H. Hu, Y. Zheng, Z. Bao, G. Li, J. Feng, and R. Cheng, "Crowdsourced POI labelling: Location-aware result inference and task assignment," in *Proc. IEEE 32nd Int. Conf. Data Eng.*, 2016, pp. 61–72.
[29] G. Li, C. Chai, J. Fan, X. Weng, J. Li, Y. Zheng, Y. Li, X. Yu, X. Zhang, and H. Yuan, "CDB: Optimizing queries with crowd-based selections and joins," in *Proc. ACM Int. Conf. Manage. Data*, 2017, pp. 1463–1478.
[30] Y. Zheng, R. Cheng, S. Maniu, and L. Mo, "On optimality of jury selection in crowdsourcing," in *Proc. 18th Int. Conf. Extending Database Technol.*, pp. 193–204, 2015.
[31] Y. Zheng, G. Li, Y. Li, C. Shan, and R. Cheng, "Truth inference in crowdsourcing: Is the problem solved?" *Proc. VLDB Endowment*, vol. 10, no. 5, pp. 541–552, 2017.
[32] Y. Zheng, G. Li, and R. Cheng, "Docs: A domain-aware crowdsourcing system using knowledge bases," *Proc. VLDB Endowment*, vol. 10, no. 4, pp. 361–372, 2016.
[33] Y. Tong, J. She, B. Ding, L. Wang, and L. Chen, "Online mobile micro-task allocation in spatial crowdsourcing," in *Proc. 12th Chinese Conf. Comput. Supported Cooperative Work Social Comput.*, 2016, pp. 49–60.
[34] Y. Tong, J. She, B. Ding, L. Chen, T. Wo, and K. Xu, "Online minimum matching in real-time spatial data: Experiments and analysis," *Proc. VLDB Endowment*, vol. 9, no. 12, pp. 1053–1064, 2016.
[35] Y. Tong, L. Wang, Z. Zhou, B. Ding, L. Chen, J. Ye, and K. Xu, "Flexible online task assignment in real-time spatial data," *Proc. VLDB Endowment*, vol. 10, no. 11, pp. 1334–1345, 2017.
[36] P. Cheng, X. Lian, L. Chen, and C. Shahabi, "Prediction-based task assignment in spatial crowdsourcing," in *Proc. IEEE 33rd Int. Conf. Data Eng.*, 2017, pp. 997–1008.
[37] L. Zheng and L. Chen, "Mutual benefit aware task assignment in a bipartite labor market," in *Proc. IEEE 32nd Int. Conf. Data Eng.*, 2016, pp. 73–84.
[38] Y. Tong, L. Wang, Z. Zhou, L. Chen, B. Du, and J. Ye, "Dynamic pricing in spatial crowdsourcing: A matching-based approach," in *Proc. Int. Conf. Manage. Data*, 2018, pp. 773–788.
[39] M. Xiao, J. Wu, S. Zhang, and J. Yu, "Secret-sharing-based secure user recruitment protocol for mobile crowdsensing," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 793–807.

Mingjun Xiao received the PhD degree from the University of Science and Technology of China, in 2004. He is an associate professor with the School of Computer Science and Technology, University of Science and Technology of China (USTC). His research interests include spatial crowdsourcing, mobile social networks, vehicular ad hoc networks, mobile cloud computing, auction theory, and data security and privacy. He is a member of the IEEE.



Kai Ma is currently a master student in the School of Computer Science and Technology, University of Science and Technology of China (USTC). His research interests include spatial crowdsourcing, vehicular ad hoc networks, auction theory, and privacy-preserving mechanism.



An Liu received the PhD degree in computer science from both the City University of Hong Kong (CityU) and the University of Science and Technology of China (USTC), in 2009. He is an associate professor with the Department of Computer Science and Technology, Soochow University. His research interests include spatial databases, crowdsourcing, data security and privacy, and cloud/service computing. He is a member of the IEEE.

Hui Zhao is currently a master student in the School of Computer Science and Technology, University of Science and Technology of China (USTC). Her research interests include spatial crowdsourcing, vehicular ad hoc networks, auction theory, and privacy-preserving mechanism.

Zhixu Li received the BS and MS degrees in computer science from the Renmin University of China, Beijing in 2006 and 2009, respectively, and the PhD degree in computer science from the University of Queensland, in 2013. He is an associate professor with the Department of Computer Science and Technology, Soochow University, Suzhou. His research interests include data cleaning, machine learning, deep learning, knowledge graph, and crowdsourcing.

Kai Zheng received the PhD degree in computer science from the University of Queensland in Queensland, in 2012. He is a professor with the University of Electronic Science and Technology of China (UESTC). His research interests include finding effective and efficient solutions for managing, integrating, and analyzing big data for business, scientific, and personal applications. He has been working in the area of spatial-temporal databases, uncertain databases, trajectory computing, social-media analysis, and bioinformatics. He is a member of the IEEE.

Xiaofang Zhou is a professor of computer science with the University of Queensland. He is the head of the Data and Knowledge Engineering Research Division. He is a specially appointed adjunct professor under the Chinese National Qianren Scheme hosted by the Renmin University of China (2010-2013), and by Soochow University since July 2013 where he leads the Research Center on Advanced Data Analytics (ADA). He has been working in the area of spatial and multimedia databases, data quality, high performance query processing, Web information systems, and bioinformatics, and co-authored more than 250 research papers with many published in top journals and conferences. He is a fellow of the IEEE.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.

Authorized licensed use limited to: University of Science and Technology of China. Downloaded on March 17,2020 at 01:55:01 UTC from IEEE Xplore. Restrictions apply.