Efficient task assignment in spatial crowdsourcing with worker and task privacy protection

Geoinformatica
DOI 10.1007/s10707-017-0305-2

Efficient task assignment in spatial crowdsourcing with worker and task privacy protection

An Liu[1] · Weiqi Wang[2] · Shuo Shang[1] · Qing Li[3] · Xiangliang Zhang[1]

Received: 1 February 2017 / Revised: 2 May 2017 / Accepted: 18 July 2017

Abstract Spatial crowdsourcing (SC) outsources tasks to a set of workers who are required to physically move to specified locations and accomplish tasks. Recently, it is emerging as a promising tool for emergency management, as it enables efficient and cost-effective collection of critical information in emergency such as earthquakes, when search and rescue survivors in potential ares are required. However in current SC systems, task locations and worker locations are all exposed in public without any privacy protection. SC systems if attacked thus have penitential risk of privacy leakage. In this paper, we propose a protocol for protecting the privacy for both workers and task requesters while maintaining the functionality of SC systems. The proposed protocol is built on partially homomorphic encryption schemes, and can efficiently realize complex operations required during task assignment over encrypted data through a well-designed computation strategy. We prove that the proposed protocol is privacy-preserving against semi-honest adversaries. Simulation on two real-world datasets shows that the proposed protocol is more effective than

✉ Shuo Shang
jedi.shang@gmail.com

An Liu
an.liu@kaust.edu.sa

Weiqi Wang
20154227018@stu.suda.edu.cn

Qing Li
itqli@cityu.edu.hk

Xiangliang Zhang
xiangliang.zhang@kaust.edu.sa

[1] King Abdullah University of Science and Technology (KAUST), Thuwal 23955, Saudi Arabia

[2] School of Computer Science and Technology, Soochow University, Suzhou, China

[3] Department of Computer Science, City University of Hong Kong, Hong Kong, China

Published online: 01 August 2017

existing solutions and can achieve mutual privacy-preserving with acceptable computation and communication cost.

Keywords Spatial crowdsourcing · Spatial task assignment · Location privacy · Mutual privacy protection

1 Introduction

Crowdsourcing has revolutionized the landscape of problem solving by outsourcing a task, usually performed by a designated agent, to a large group of people in the form of an open call [19]. Providing on-demand talent capacity and expert services with significantly less cost than hiring dedicated professionals, it has been successfully applied, for example, to transcribe books [41], fold proteins [5], classify galaxies [13], and monitor traffic [42]. Recently, crowdsourcing has also been widely used in emergency management, as it enables efficient and cost-effective collection of critical information in emergency and disaster such as the area of impact, population at risk, and potential areas where search and rescue operations are likely to be required. For example, on April 25, 2015, Nepal was struck by a 7.8 magnitude earthquake. To provide detailed damage assessment, DigitalGlobe collected pre- and post-earthquake high-resolution satellite images of the affected areas, which are divided into small sections and given to an online crowd to identify damaged buildings and blocked roads. Thanks to crowdsourcing, more than 21,000 damaged buildings and roads were identified and tagged within a month [8], providing valuable data for both relief and rebuilding.

The role crowdsourcing played in emergency management can be more proactive due to the fast development of ubiquitous wireless networks and smart mobile devices [12, 21, 33]. A novel type of crowdsourcing, *Spatial Crowdsourcing* (SC) [20], outsources a spatial task (i.e., a task related to a location) to a number of workers with mobile devices who are required to physically move to some specified locations and accomplish the task. Let us continue the above example of emergency management in an earthquake. An SC-server sends a spatial task of assessing whether or not there are survivors in a specific collapsed building to all available workers consisting of both volunteers and professionals with life detector instrument. Workers who are willing to perform the task go to the building, check it and send the results back to the SC-server, based on which subsequent rescue plan can be made, for example, professional heavy rescue equipments will be deployed on site if someone are identified to be trapped in rubble.

The success of crowdsourcing depends on the active participation of the crowd despite its application domains [22, 25, 31, 35, 47]. For spatial crowdsourcing, location privacy concerns are a major factor in hindering workers from engaging in spatial tasks [26, 37]. To enable effective task assignment (effectiveness here means spatial-tasks can be completed quickly by being assigned to nearby workers), the SC-server needs to continuously collect workers' locations via their mobile devices [30, 32, 34]. However, it is very difficult for workers to control the usage of their location data stored by the SC-server, an untrusted third-party. In fact, the collected location data is likely to be shared, rent or sold with other parties, which has serious privacy implications [23, 24, 29, 45]. Based on these location data, an adversary can stage a broad spectrum of attacks against individuals such as physical surveillance and stalking, identity theft, and breach of sensitive information, for instance, home address and lifestyle habits. Hence, location privacy protection, or more

generally, worker privacy protection, is an important aspect of spatial crowdsourcing, as it can stimulate workers to take active part in spatial tasks. This is of particular importance for emergency management, as more active workers typically mean tasks can be completed more quickly.

Tasks on existing crowdsourcing platforms such as Amazon Mechanical Turk are publicly available to all workers. This mode may not be suitable for spatial crowdsourcing in the scenario of emergency management. Once the location of a task is known publicly, overeager workers motivated by altruism can go there to perform the task even if they are not required to, which may set off more disruption, for example, a traffic jam [36]. Therefore, the task's location should not be learned by workers except the one to whom the task is assigned. Sometimes task location protection is also welcome from the task requester viewpoint. For example, someone suffering a health problem at home can ask for help by crowdsourcing, but publication of her health problem together with her home address clearly breaches her privacy. Hence, task location privacy should also be respected in the course of spatial crowdsourcing.

While a lot of efforts [16, 29, 45] have been made towards location privacy protection in the scenario of location-based services, there are only a couple of works [37, 38] that study location privacy preserving in spatial crowdsourcing applications. In [37], worker locations are collected and perturbed by a trusted party which injects calibrated noises into the raw data according to differential privacy [9]. Upon receiving a spatial task, an SC-server queries the disguised location data to determine a region that is very likely to contain sufficient workers nearby the task's location. Workers in this region are notified about this task and have the right to decide whether or not to perform it. The proposed solution in this pioneering work has several weaknesses. First, it only takes into account worker location privacy without considering task location privacy. Second, it performs task assignment primarily based on worker travel distance, and fails to consider other important factors, such as worker velocity, which makes assignment results sometimes unsatisfactory. Further, it works on a very strong assumption that there is a trusted party who has the access to all workers' locations.

In this paper, we propose a protocol for privacy-preserving task assignment in spatial crowdsourcing with the following objectives:

- *Mutual privacy protection*. Not only worker privacy but also task privacy should be protected during task assignment. We adopt well-known cryptosystems to encrypt private data of both parties, thus achieving strong mutual privacy guarantee.
- *Effective task assignment*. During task assignment, travel time is more meaningful than travel distance especially for tasks with deadlines, so worker velocity is considered to be an important metric in recent spatial crowdsourcing applications [4, 7, 39]. We integrate worker velocity with worker location to realize more effective task assignment.
- *Acceptable overheads*. The strength of privacy protection comes at the expense of additional computation or communication cost. We combine partially homomorphic encryption schemes to efficiently realize complex operations required during task assignment over encrypted data, thus avoiding significant performance penalties.

To summarize, our contributions in this paper are listed as follows:

- We propose a protocol for efficient task assignment in spatial crowdsourcing with worker and task privacy protection. To the best of our knowledge, this is the first work that achieves mutual privacy protection in spatial crowdsourcing.

– We devise a computation strategy to eliminate some complex operations that cannot be supported by existing practical cryptosystems. By this strategy, our protocol achieves mutual privacy protection with acceptable overheads.
– We theoretically analyze the security and complexity of our protocol. We also conduct extensive experiments to evaluate the performance of our protocol on two real-world datasets.

The rest of the paper is organized as follows. Section 2 presents problem statement and introduces background knowledge. Section 3 presents the mutual privacy-preserving task assignment protocol and analyzes its complexity and security. Section 4 gives the experimental results. Finally, Section 5 discusses some related work and Section 6 concludes the paper.

2 Preliminaries

In this section, we will formally define the problems of privacy-preserving task assignment in spatial crowdsourcing and briefly review some related cryptographic building blocks. Table 1 summarizes the notation used throughout this paper.

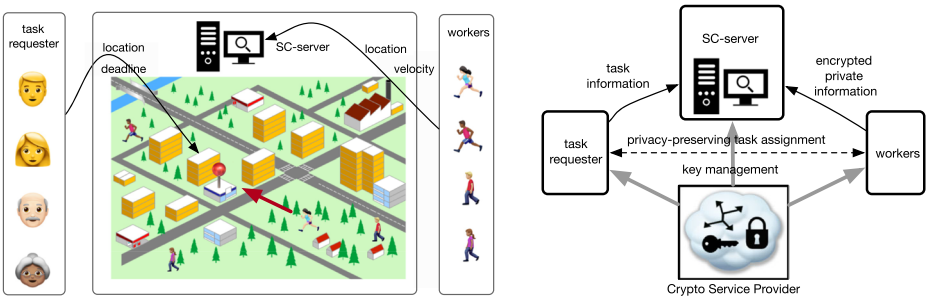
2.1 System model and problem definitions

Figure 1 depicts the system model of spatial crowdsourcing. For the non-private spatial crowdsourcing (see Fig. 1a), there are three parties: the SC-server, workers with mobile devices, and spatial task requesters. The SC-server is in charge of assigning appropriate workers to spatial tasks created by task requesters. Workers are required to report their private information (e.g., location and velocity) to the SC-server via their mobile devices. Based on this framework, we give the following definitions.

Definition 1 (Spatial Task) A spatial task is a task s to be performed at location l_s and is associated with a deadline e_s.

Table 1 Summary of notation

Notation	Meaning
W	a set of n moving workers w_i
l_i	the location of worker w_i
v_i	the velocity of worker w_i
s	a spatial task with location l_s and deadline e_s
$d(l_i, l_s)$	the Euclidean distance between worker w_i and task s
E	Paillier encryption algorithm
D	Paillier decryption algorithm
E'	ElGamal encryption algorithm
D'	ElGamal decryption algorithm
$\overline{\mathrm{E}'}$	ElGamal re-encryption algorithm
$\overline{\mathrm{D}'}$	ElGamal re-decryption algorithm
f_k	a pseudo-random function (PRF)



(a) Non-private SC (b) Privacy-preserving SC

Fig. 1 System model of spatial crowdsourcing (SC)

Definition 2 (Worker) A worker is a person w who is willing to perform spatial tasks. Each worker is associated with an ID id_w designated by the SC-server, a velocity v_w and a location l_w where she is currently located.

With spatial crowdsourcing, a task requester creates a spatial task s and specifies its location l_s and deadline e_s. To perform this task, workers have to physically move to l_s and arrive at l_s before the deadline e_s. Upon receiving a spatial task, the SC-server assigns it to appropriate workers based on some predefined strategy. In this paper, we assume that the SC-server prefers the worker who might be the first to arrive at l_s. Following [37], we also assume that every worker accepts an assigned task with a certain probability, denoted as acceptance rate (AR). We first define simple task assignment problem as follows, assuming the AR of every worker is 100%:

Definition 3 (Task Assignment Problem) Let $W = \{w_1, w_2, \cdots, w_n\}$ be a set of n workers. Given a spatial task s, the task assignment problem, $\mathbb{P}_{\text{TA}}(W, s)$, is to assign the task s to a worker w_{i^*} such that:

1. w_{i^*} can arrive at location l_s before the deadline e_s;
2. no other workers can arrive at l_s before w_{i^*}.

In Definition 3, the first requirement means $t_c + d(l_{i^*}, l_s)/v_{i^*} \leq e_s$ where t_c is the current time, l_{i^*} is w_{i^*}'s current location, v_{i^*} is w_{i^*}'s velocity, and $d(l_{i^*}, l_s)$ is the Euclidean distance between locations l_{i^*} and l_s. The second requirement means there does not exist w_j such that $d(l_j, l_s)/v_j < d(l_{i^*}, l_s)/v_{i^*}$. To facilitate later discussion, we call w_{i^*} the *winner* of this problem and take i^* as her ID. Note that such a winner does not exist when all workers cannot arrive at l_s before the deadline. In this case, the SC-server informs the task requester that there are no winners.

In practice, however, workers do not necessarily accept tasks assigned to them. To guarantee a task is accepted with a high probability, more than one worker can be asked to perform the task. Suppose the AR of worker w_i is a_i. Denote by $\eta(W, s)$ the probability that at least one worker in W accepts the task s. Clearly, $\eta(W, s) = 1 - \prod_{i=1}^{n}(1 - a_i)$. We thus define below another task assignment problem:

Definition 4 (Task Assignment with Acceptance Guarantee Problem) Let $W = \{w_1, w_2, \cdots, w_n\}$ be a set of n workers. Given a spatial task s, the task assignment with acceptance guarantee problem, $\mathbb{P}_{\text{TAG}}(W, s)$, is to assign the task s to a set of workers W^* (called winner set) such that:

1. every worker $w_{i^*} \in W^*$ can arrive at location l_s before the deadline e_s;
2. no other worker $w_j \in W \setminus W^*$ can arrive at l_s before any worker $w_{i^*} \in W^*$;
3. $\eta(W^*, s) \geq \alpha$ where α is the expected probability that s is accepted by at least one worker in W^*.

Adversary model Figure 1b shows the system model of privacy-preserving spatial crowdsourcing. A new party crypto service provider (CSP) is introduced to provide cryptographic services, such as key generation, to the SC-server and workers. Regarding adversary model, we assume all parties are semi-honest [17], that is, they follow a protocol exactly as specified, but may try to learn as much as possible about other parties' private input from what they see during the protocol's execution. In particular, the SC-server is interested in every worker's location and velocity, and every winner's ID. CSP is also interested in that, as well as the task's location. Every worker is interested in other workers' location and velocity, every winner's ID, and the task's location. As a special worker, every winner is entitled to know her ID and the task's location, but she is also interested in other workers' location and velocity, and other winners' IDs. Base on the adversary model, we have the following definition:

Definition 5 (Privacy-preserving Task Assignment Problem) Let $W = \{w_1, w_2, \cdots, w_n\}$ be a set of n workers. Given a spatial task s, the privacy-preserving task assignment problem, $\mathbb{P}_{\text{PTA}}(W, s)$, is to find w_{i^*} the winner of $\mathbb{P}_{\text{TA}}(W, s)$ in a way that:

1. for each worker $w_i \in W$, her location l_i and velocity v_i cannot be learned by the SC-server, CSP, and any other worker $w_j \in W$, $w_j \neq w_i$;
2. the task's location l_s cannot be learned by CSP and all workers except w_{i^*};
3. the ID of w_{i^*} cannot be learned by the SC-server, CSP and all workers except w_{i^*}.

While its non-private version (i.e., $\mathbb{P}_{\text{TA}}$) is simple, $\mathbb{P}_{\text{PTA}}$ is very challenging as it tries to protect worker privacy and task privacy at the same time. In particular, the winner is determined by not only worker position but also worker velocity, both of which should be kept secret during computation. At the first glance, this requirement means that we need to perform division over ciphertext. Unfortunately, efficient homomorphic division is still an open question nowadays. Moreover, the task's location l_s is required to be hidden from all workers except the winner, which makes the computation of $d(l_i, l_s)$ much harder than that over plaintext. Note that the winner has to know l_s as she needs to physically move there to perform the task, so this does not count as a breach of privacy. The last requirement of $\mathbb{P}_{\text{PTA}}$ indicates that the SC-server is not allowed to know the identity of the winner. If the SC-server knows who is the winner, it is likely to infer the winner's approximate location based on some background knowledge, for example, the task's location and deadline. Clearly, it is the SC-server that decides the winner in $\mathbb{P}_{\text{TA}}$. In $\mathbb{P}_{\text{PTA}}$, however, the SC-server is not allowed to know who is the winner. This contradiction is another difficult challenge of $\mathbb{P}_{\text{PTA}}$.

Similarly, we have the following definition of privacy-preserving task assignment with acceptance guarantee problem:

Definition 6 (Privacy-preserving Task Assignment with Acceptance Guarantee Problem) Let $W = \{w_1, w_2, \cdots, w_n\}$ be a set of n workers. Given a spatial task s, the privacy-preserving task assignment with acceptance guarantee problem, $\mathbb{P}_{\text{PTAG}}(W, s)$, is to find W^*, the winner set of $\mathbb{P}_{\text{TAG}}(W, s)$ in a way that:

1. for each worker $w_i \in W$, her location l_i and velocity v_i cannot be learned by the SC-server, CSP, and any other worker $w_j \in W$, $w_j \neq w_i$;
2. the task's location l_s cannot be learned by CSP and all workers except the winners in W^*;
3. the ID of w_{i^*} cannot be learned by the SC-server, CSP and all workers except w_{i^*}.

2.2 Formal privacy definition

We use real-ideal paradigm [17] to define the security of a protocol. Intuitively, a protocol is secure or privacy-preserving if every party involved in the protocol learns no more knowledge from the execution of the protocol than the knowledge that this party is entitled to know. This can be formally defined by the real-ideal paradigm as follows: for all adversaries, there exists a probabilistic polynomial-time simulator, so that the view of the adversary in the real world and the view of the simulator in the ideal world are computationally indistinguishable.

Let P_{-1} be CSP, P_0 be the SC-server, and $P_1, \cdots, P_n$ be n workers. Let $view_i$, x_i, and K_i $(-1 \leq i \leq n)$ be party P_i's view, its private input, and the extra knowledge it can learn, respectively, during an execution of protocol $\mathscr{P}$. The privacy requirement of $\mathscr{P}$ is formally defined as follows:

Definition 7 A protocol $\mathscr{P}$ is *perfectly privacy-preserving* against party P_i in the sense that it reveals no more knowledge than the final output to P_i, if there exists a probabilistic polynomial-time simulator S_i such that:

$$S_i(x_i, \mathscr{P}(x_{-1}, x_0, \cdots, x_n), K_i)_{x_{-1}, x_0, \cdots, x_n} \equiv view_i(x_{-1}, x_0, \cdots, x_n)_{x_{-1}, x_0, \cdots, x_n}$$

and $K_i = \emptyset$ given all possible inputs $(x_{-1}, x_0, \cdots, x_n)$, where $\equiv$ denotes computational indistinguishability. If $K_i \neq \emptyset$, $\mathscr{P}$ is said to be *privacy-preserving with K_i disclosure* against P_i in the sense that it reveals no more knowledge than K_i and the final output to P_i.

It is clear that perfectly privacy-preserving is a very strong privacy guarantee. However, such a strong guarantee sometimes cannot be achieved by efficient protocols. In practice, an extra knowledge K disclosure during an execution of protocol $\mathscr{P}$ can be allowed (for efficiency) as long as it does not breach privacy in the sense that, even based on K, the probability that an adversary learns the private input of any party during an execution of protocol $\mathscr{P}$ is negligible.

2.3 Cryptographic building blocks

To solve $\mathbb{P}_{\text{PTA}}$ and $\mathbb{P}_{\text{PTAG}}$ defined above, we employ several cryptographic tools: pseudo-random function, Paillier cryptosystem [27] and ElGamal cryptosystem [10], which are briefly introduced as follows.

A pseudo-random function (PRF) is a function that cannot be distinguished from a truly random one by observing the result in a black-box manner. Usually, the PRF is denoted by f_k, a function of the PRF family $\mathbb{F}_\lambda = \{f_k : \{0,1\}^\lambda \to \{0,1\}^\lambda\}_{k\in\{0,1\}^\lambda}$ indexed by k. Our working assumption is that keyed one-way hash functions (such as HMAC) can be modeled as a pseudo-random function [1]. Therefore, the function $f_k(x)$ can be implemented by keying a hash function with k and applying it to x.

Paillier is a public-key cryptosystem whose security is based on an assumption related (but not known to be equivalent) to the hardness of factoring. It consists of the following three algorithms:

- **Key generation**: Choose two distinct large random primes p, q and compute $N = pg$. Choose an element $g \in \mathbb{Z}^*_{N^2}$. The public key pk is (N, g) and the secret key sk is (p, q).
- **Encryption** E: Let m be a message in $\mathbb{Z}_N$. It is encrypted by selecting a random number r in $\mathbb{Z}^*_N$ and computing

$$c = \mathrm{E}(m) = g^m r^N \mod N^2, \tag{1}$$

  where N and g are from the public key pk and c is the ciphertext of m.
- **Decryption** D: The ciphertext c is decrypted by computing

$$m = \mathrm{D}(c) = \frac{(c^\lambda \mod N^2) - 1}{(g^\lambda \mod N^2) - 1} \mod N, \tag{2}$$

  where $\lambda = lcm(p-1, q-1)$ can be computed from the private key sk.

One of the most important properties of Paillier cryptosystem is *homomorphic addition*. Specifically, multiplying an encryption of m_1 and an encryption of m_2 results in an encryption of $m_1 + m_2$, and raising an encryption of m to a constant k results in an encryption of km, that is,

$$\mathrm{E}(m_1)\mathrm{E}(m_2) = \mathrm{E}(m_1 + m_2), \tag{3}$$

$$\mathrm{E}(m)^k = \mathrm{E}(km). \tag{4}$$

Besides, Paillier is *semantic secure*, that is, an adversary cannot learn any partial information about the plaintext from the ciphertext. As a result, it is also a probabilistic encryption scheme, which means when encrypting the same message several times, it will produce different ciphertexts. This is clear from Eq. (1) where a random number r is used in encryption.

ElGamal is a public-key cryptosystem whose security is based on the difficulty of the discrete logarithm problem. It consists of some public domain parameters that can be shared by a number of users and three algorithms:

- **Domain parameters**. Let p be a large prime and q be a medium prime such that $q|p-1$, g be an element of $\mathbb{F}^*_p$ of prime order q, that is, $g = r^{(p-1/q)} \mod p \neq 1$ for some $r \in \mathbb{F}^*_p$. These public parameters create a public finite abelian group G of prime order q with generator g.
- **Key generation**. Choose an integer x such that $0 \leq x \leq q-1$ and compute $h = g^x \mod p$. The public key pk is h and the secret key sk is x.
- **Encryption** E'. Let m be a message in G. It is encrypted by selecting a random number r such that $0 \leq r \leq q-1$ and computing

$$c_1 = g^r, c_2 = mh^r. \tag{5}$$

  The ciphertext c of m is $\mathrm{E}'(m) = (c_1, c_2)$.

– **Decryption** D'. The ciphertext c is decrypted by computing

$$m = \mathrm{D}'(c) = c_2(c_1^x)^{-1} \tag{6}$$

ElGamal is also a probabilistic encryption scheme as each message is encrypted by a different random number r, as shown in Eq. (5). One interesting property of ElGamal cryptosystem is *homomorphic multiplication*. Specifically, multiplying an encryption of m_1 and an encryption of m_2 results in an encryption of $m_1 m_2$, that is:

$$\mathrm{E}'(m_1)\mathrm{E}'(m_2) = \mathrm{E}'(m_1 m_2). \tag{7}$$

A *commutative encryption* satisfies the property that the order of two encryptions is irrelevant. ElGamal can be extended to support commutative-like encryption [6]. In particular, two new algorithms are defined as follows [6]:

– **Re-encryption** $\overline{\mathrm{E}'}$. Given a ciphertext $\mathrm{E}'_{h_a}(m) = (g^{r_a}, mh_a^{r_a})$ encrypted by public key h_a, it can be re-encrypted by selecting a random number r_b such that $0 \leq r_b \leq q-1$ and computing $c_1 = g^{r_a}$, $c_2 = g^{r_b}$, and $c_3 = mh_a^{r_a}h_b^{r_b}$ where h_b is the public key. The ciphertext of $\mathrm{E}'_{h_a}(m)$ is $\overline{\mathrm{E}'}_{h_b}(\mathrm{E}'_{h_a}(m)) = (c_1, c_2, c_3)$.
– **Re-decryption** $\overline{\mathrm{D}'}$. The ciphertext (c_1, c_2, c_3) can be decrypted by secret keys x_a and x_b in different orders, producing the same result. Using secret key x_a first, we have $\overline{\mathrm{D}'}_{x_a}(\overline{\mathrm{E}'}_{h_b}(\mathrm{E}'_{h_a}(m))) = (c_2, c_3(c_1^{x_a})^{-1}) = (g^{r_b}, mh_b^{r_b}) = \mathrm{E}'_{h_b}(m)$, which can be decrypted again by x_b to obtain m. It is easy to verify that the result is also m when first using x_b and then x_a.

3 Privacy-preserving task assignment protocol

According to Definition 5, our objective is to find the winner of $\mathbb{P}_{\mathrm{TA}}$ without disclosing worker location information. Though some existing privacy protection tools such as k-anonymity and differential privacy could be adopted to protect individual privacy, they typically assume a trusted third party that has the access to the whole raw data (e.g., all workers' locations), which is too strong in practice. Besides, they preserve individual privacy at the cost of decreasing data utility, which means approaches based on them may not be able to find the correct winner of $\mathbb{P}_{\mathrm{TA}}$. Therefore, we decide to make use of cryptographic tools to solve $\mathbb{P}_{\mathrm{PTA}}$ accurately. To prevent privacy leakage, the private data of each worker are encrypted by herself before being sent to the SC-server. From Definition 3, the key problem in $\mathbb{P}_{\mathrm{PTA}}$ is to determine which worker will be the first to arrive l_s. To solve this problem, we need to compare the travel times of two workers w_i and w_j, that is, to evaluate the following inequality:

$$\frac{d(l_i, l_s)}{v_i} < \frac{d(l_j, l_s)}{v_j}. \tag{8}$$

Clearly, the evaluation consists of several fundamental operations: addition and multiplication (for distance computation), division, and comparison. It is important to note that these operations should be performed over ciphertext since, for example, l_i and v_i have been encrypted for privacy protection. In theory, we can design an approach based on a fully homomorphic encryption (FHE) scheme [14, 15] to enabling the above evaluation, but this will incur prohibitive computation cost which makes the approach being of limited practical significance. We therefore consider to use partially homomorphic encryption schemes.

Though they are much more efficient than FHE, none of them can support all the operations required in the evaluation of inequality (8). We will show how to solve this difficult challenge in the next subsections.

3.1 Protocol overview

Figure 2 gives a high level picture of our privacy-preserving task assignment protocol. Based on the above discussion, we adopt two partially homomorphic encryption schemes, Paillier and ElGamal, to construct our solution, which consists of five stages depicted in different colors in Fig. 2. In stage 0, CSP generates domain parameters for ElGamal and the pair of keys for Paillier and ElGamal according to the requirement of security. It keeps the secret keys private and issues the public keys to the SC-server and all workers. The creation of a spatial task by a task requester triggers the start of stage 1 during which the SC-server and all workers run a privacy-preserving distance computation protocol which outputs encrypted distance values based on encrypted locations. In stage 2, each worker's velocity is encrypted and sent to the SC-server which cooperates with CSP to compute the travel time for every worker. Based on the encrypted travel times obtained in stage 2, the SC-server computes the winner with the help of CSP in stage 3, but the result is still in the encrypted form. In the last stage, the encrypted task's location is broadcast to all workers but only the winner is able to recover the task's location. After that, the winner moves to the specified location to perform the corresponding task.

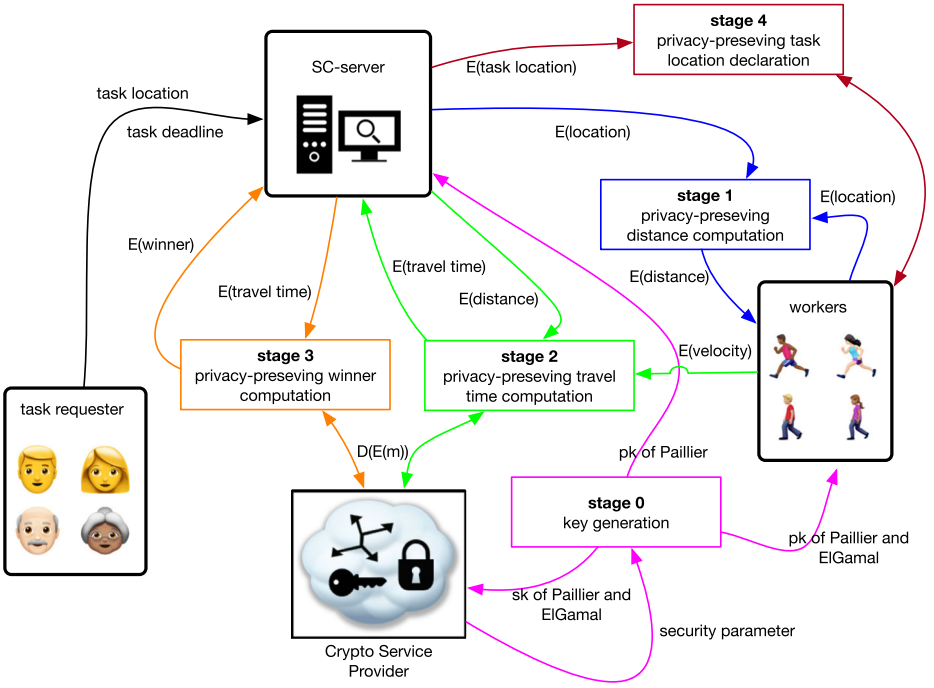


Fig. 2 Overview of privacy-preserving task assignment protocol

3.2 Detailed construction

Algorithm 1 shows our protocol for privacy-preserving task assignment. We explain it in details as follows.

Stage 1. We start to present our detailed construction from stage 1 since key generation for Paillier and ElGamal cryptosystems required in stage 0 has been introduced in Section 2.3. Holding the public key of Paillier, the SC-server encrypts the task location $l_s = (x_s, y_s)$ and sends three ciphertexts $\mathrm{E}(x_s^2 + y_s^2)$, $\mathrm{E}(x_s)$ and $\mathrm{E}(y_s)$ to all workers. On receiving encrypted values from the SC-server, every worker w_i computes the encrypted square of the distance between l_s and her current location $l_i = (x_i, y_i)$ as follows:

$$\mathrm{E}(d^2(l_i, l_s)) = \mathrm{E}(x_s^2 + y_s^2)\mathrm{E}(x_s)^{-2x_i}\mathrm{E}(y_s)^{-2y_i}\mathrm{E}(x_i^2 + y_i^2), \tag{9}$$

whose correctness can be easily verified based on Eqs. (3) and (4). Note that we can also ask all workers to send the SC-server their encrypted locations (in the form of $\mathrm{E}(x_i^2 + y_i^2)$, $\mathrm{E}(x_i)$ and $\mathrm{E}(y_i)$) and ask the SC-server to compute $\mathrm{E}(d^2(l_i, l_s))$ for every worker. Though this procedure is similar to what we do in the non-private case, it incurs much more computation cost for the SC-server. In other words, our current design has an advantage of amortizing the computation cost to all workers.

Stage 2. As discussed earlier, privacy-preserving travel time computation needs division operation over ciphertext. Efficient implementation of homomorphic division, however, is still an open question. Hence, our objective here is not to design an efficient protocol of homomorphic division, but to technically eliminate division during the computation of travel time. To do this, we utilize one interesting property of the comparison of travel times, that is, the exact travel time computation is not necessary. This property is guaranteed by the following lemma:

Lemma 1 *Let $W = \{w_1, w_2, \cdots, w_n\}$ be a set of n workers, $\mathcal{V}$ be the product of all workers' velocities, that is, $\mathcal{V} = \prod_{k=1}^{n} v_k$, and $v'_k = \mathcal{V}/v_k$ for $1 \le k \le n$. For any two workers $w_i, w_j \in W$, $d(l_i, l_s)/v_i < d(l_j, l_s)/v_j$ holds if and only if $d(l_i, l_s)v'_i < d(l_j, l_s)v'_j$.*

Proof

$$\frac{d(l_i, l_s)}{v_i} < \frac{d(l_j, l_s)}{v_j} \Leftrightarrow d(l_i, l_s)\frac{v'_i}{\mathcal{V}} < d(l_j, l_s)\frac{v'_j}{\mathcal{V}} \Leftrightarrow d(l_i, l_s)v'_i < d(l_j, l_s)v'_j$$

□

Based on the lemma, we compute for each worker w_i a virtual travel time $t'_i = d(l_i, l_s)v'_i$ which is equivalent of the exact travel time $t_i = d(l_i, l_s)/v_k$ in the sense that a worker having the shortest virtual travel time necessarily has the shortest exact travel time. Specifically, every worker w_i encrypts her velocity v_i by the ElGamal cryptosystem and sends $\mathrm{E}'(v_i)$ to the SC-server which can obtain $\mathrm{E}'(\mathcal{V})$ by multiplying all the encrypted velocities it received. The SC-server then asks CSP to decrypt $\mathrm{E}'(\mathcal{V})$ and gives $\mathcal{V}$ to all workers. By dividing $\mathcal{V}$ by her velocity v_i, every worker w_i has the value of v'_i and computes $\mathrm{E}(d^2(l_i, l_s))^{v'^2_i} = \mathrm{E}(d^2(l_i, l_s)v'^2_i) = \mathrm{E}(t'^2_i)$. The encrypted virtual travel times are sent to the SC-server for further processing. Note that the exact value of $\mathcal{V}$ is known to CSP and all workers in the

above procedure. However, this does not breach the individual privacy of any worker, which will be proved in the next subsection.

Algorithm 1 Privacy-preserving task assignment protocol

Input: A set of n workers, each worker w_i has an ID i, a location l_i, and a velocity v_i; a spatial task s (created by a task requester) with a location l_s and deadline e_s; an SC-server and a CSP.
Output: the winner w^* obtains task location l_s.

Stage 0 - Key generation
CSP generates a pair of keys (pk, sk) of Paillier cryptosystem and a pair of keys (pk', sk') of ElGamal cryptosystem. The SC-server and all workers know the public keys pk and pk'. Only the CSP knows the private keys sk and sk'.
CSP generates another set of domain parameters for ElGamal cryptosystem and makes them publicly available. Based on these parameters, the CSP generates another public key pk'' but keeps it secret, and each worker w_i also generates a pair of keys (pk_i'', sk_i'') and keeps it secret.
Stage 1 - Privacy-preserving distance computation
the SC-server encrypts $x_s^2 + y_s^2$, x_s, and y_s by pk and sends the results to all workers.
for each worker w_i $(1 \le i \le n)$ **do**
    w_i encrypts $x_i^2 + y_i^2$ by pk to obtain $\mathrm{E}(x_s^2 + y_s^2)$.
    w_i computes $\mathrm{E}(d^2(l_i, l_s)) = \mathrm{E}(x_s^2 + y_s^2)\mathrm{E}(x_s)^{-2x_i}\mathrm{E}(y_s)^{-2y_i}\mathrm{E}(x_i^2 + y_i^2)$.
end for
Stage 2 - Privacy-preserving travel time computation
for each worker w_i $(1 \le i \le n)$ **do**
    w_i encrypts v_i by pk' and sends $\mathrm{E}'(v_i)$ to the SC-server.
end for
the SC-server computes $\mathrm{E}'(\mathcal{V}) = \prod_{i=1}^{n} \mathrm{E}'(v_i)$ and sends it to CSP.
CSP decrypts $\mathrm{E}'(\mathcal{V})$ and sends it to the SC-server.
the SC-server broadcasts $\mathcal{V}$ to all workers.
for each worker w_i $(1 \le i \le n)$ **do**
    w_i computes $\mathrm{E}(t_i'^2) = \mathrm{E}(d^2(l_i, l_s))^{(\mathcal{V}/v_i)^2}$ and sends $\mathrm{E}(t_i'^2)$ to the SC-server
end for
Stage 3 - Privacy-preserving winner computation
the SC-server sends $f_k(i)$ to worker w_i where f_k is a PRF.
the SC-server sends CSP $\langle f_k(i), \mathrm{E}(t_{f_k(i)}'^2)\rangle$ for $1 \le i \le n$.
CSP decrypts $\mathrm{E}(t_{f_k(i)}'^2)$ and computes $t_{f_k(i)} = \sqrt{t_{f_k(i)}'^2/\mathcal{V}^2}$ for $1 \le i \le n$.
CSP finds the winner w_{i^*} with the smallest travel time $t_{f_k(i^*)}$.
CSP encrypts $f_k(i^*)$ by k' and sends $\mathrm{E}'_C(f_k(i^*))$ to the SC-server.
Stage 4 - Privacy-preserving task location declaration
the SC-server encrypts l_s by computing $h(\mathrm{E}'_C(f_k(i^*))) \oplus l_s$ where h is a length-match hash function and broadcasts $\dot{\mathrm{E}}(l_s)$ to all workers.
for each worker w_i $(1 \le i \le n)$ **do**
    w_i encrypts $f_k(i)$ by pk_i'' and sends $\mathrm{E}'_{w_i}(f_k(i))$ to CSP.
    CSP encrypts $\mathrm{E}'_{w_i}(f_k(i))$ by pk'' and sends $\overline{\mathrm{E}'}_C(\mathrm{E}'_{w_i}(f_k(i)))$ to w_i.
    w_i decrypt $\overline{\mathrm{E}'}_C(\mathrm{E}'_{w_i}(f_k(i)))$ by sk_i'' to obtain $\mathrm{E}'_C(f_k(i))$.
    w_i tries to decrypt $\dot{\mathrm{E}}(l_s)$ by computing $h(\mathrm{E}'_C(f_k(i))) \oplus \dot{\mathrm{E}}(l_s)$.
end for

Stage 3. Now, the SC-server has a list of 2-tuples $\langle i, \mathrm{E}(t_i'^2)\rangle$ where i is the ID of worker w_i and $1 \leq i \leq n$. To protect the identities of workers especially the winner, it encrypts every worker's ID by a PRF f_k and sends CSP the list $\langle f_k(i), \mathrm{E}(t'^2_{f_k(i)})\rangle$ to find which worker has the shortest travel time and whether she can arrive the task location before the deadline e_s. As CSP has the private key of Paillier, it can obtain $t_i'^2$ by decrypting $\mathrm{E}(t_i'^2)$ and compute the real travel time $t_i = \sqrt{t_i'^2/\mathcal{V}^2}$. Then, it is easy for CSP to find the worker having the shortest travel time and to check whether she can meet the deadline constraint. If not, CSP informs the SC-server that there are no winners. Otherwise, it encrypts the winner's ID $f_k(i^*)$ using ElGamal and sends $\mathrm{E'}_C(f_k(i^*))$ to the SC-server. This encryption is necessary, as the SC-server can infer who is the winner once it knows $f_k(i^*)$. On the other hand, the winner's privacy is still protected due to the pseudo-randomness of the PRF.

Stage 4. On receiving $\mathrm{E'}_C(f_k(i^*))$, the SC-server encrypts the task location l_s and broadcasts $\dot{\mathrm{E}}(l_s)$ to all workers. Specifically, l_s is encrypted as follows:

$$\dot{\mathrm{E}}(l_s) = h(\mathrm{E'}_C(f_k(i^*))) \oplus l_s, \tag{10}$$

where h is a length-match hash function which is used to map a long bit string to a shorter bit string. A specific construction of h proved to be semantically secure [1] is to truncate a long bit string into multiple shorter bit strings of fixed length and output the exclusive-OR on these strings. Clearly, only the worker knowing $\mathrm{E'}_C(f_k(i^*))$ can recover the task location by computing $l_s = \dot{\mathrm{E}}(l_s) \oplus h(\mathrm{E'}_C(f_k(i^*)))$. The following procedure ensures that only the winner knows $\mathrm{E'}_C(f_k(i^*))$.

First, every workers w_i obtains her encrypted ID $f_k(i)$ from the SC-server and encrypts it by ElGamal using her own public key and sends the encrypted value $\mathrm{E'}_{w_i}(f_k(i))$ to CSP. When receiving it, CSP encrypts it again by ElGamal using its public key and the same random number r used for encryption of $\mathrm{E'}_C(f_k(i^*))$. The result $\overline{\mathrm{E}}'_C(\mathrm{E'}_{w_i}(f_k(i)))$ is send to every worker i who can decrypt it by her private key to obtain $\mathrm{E'}_C(f_k(i))$. Clearly, only the winner $w_{f_k(i^*)}$ knows $\mathrm{E'}_C(f_k(i^*))$. It is important to note that the *public* keys used here should be kept secret for privacy protection.

Remark 1 During the computation of $\mathrm{E'}(\mathcal{V})$, an appropriate key length should be set to avoid overflow of the multiplication of all workers' velocities. For example, we use 2048-bits keys in our experiments to deal with 1,000 workers. For very large number of workers, a possible way is to use least common multiple (LCM) instead of multiplication. However, privacy-preserving LCM computation (i.e., compute the LCM of multiple encrypted numbers) is a very challenging problem and we leave it as one of our future directions.

3.3 Performance analysis

Computation cost Table 2 summarizes the computation cost of our protocol. We assume all workers can perform computation (e.g., encryption and decryption) in parallel and can interact with the SC-server/CSP in parallel, so we only need to consider the cost of one user. Besides, we ignore cheaper operations such as big integer multiplication and exclusive-or of bit strings. The detailed analysis is as follows. In Algorithm 1, three Paillier encryptions

Table 2 Computation cost of the proposed protocol

	SC-server	CSP	Worker
stage 1	3E	0	$1\mathrm{E} + 2e$
stage 2	0	$1\mathrm{D}'$	$1\mathrm{E} + 3e$
stage 3	n PRF	$1\mathrm{E}' + n\mathrm{D}$	0
stage 4	0	$n\overline{\mathrm{E}'}$	$1\mathrm{E}' + 1\overline{\mathrm{D}'}$
total	$3\mathrm{E} + n$ PRF	$1\mathrm{E}' + n\overline{\mathrm{E}'} + n\mathrm{D} + 1\mathrm{D}'$	$2\mathrm{E} + 1\mathrm{E}' + 1\overline{\mathrm{D}'} + 5e$

E, D, E', D', $\overline{\mathrm{E}'}$, $\overline{\mathrm{D}'}$, e, PRF represent Paillier encryption, Paillier decryption, ElGamal encryption, ElGamal decryption, ElGamal re-encryption, ElGamal re-decryption, modular exponentiation, and pseudorandom function, respectively

are performed by the SC-server (Line 5) and one Paillier encryption and two modular exponentiations are performed by worker w_i (Line 7 and 8) for privacy-preserving travel distance computation. In stage 2, worker w_i protects her velocity by one ElGamal encryption (Line 12). The product of encrypted velocities is decrypted by CSP (Line 15) to enable subsequent travel time computation which requires one modular exponentiation for worker w_i (Line 18). In stage 3, the SC-server uses n PRFs to protect workers' ID (Line 21) and CSP performs n ElGamal decryptions (Line 23) and one ElGamal encryption (Line 25) to find the winner and protect her ID. In the last stage, to exchange decryption key, worker w_i performs one ElGamal encryption (Line 29) and one ElGamal re-decryption (Line 31), and CSP performs n ElGamal re-encryptions (Line 30).

Communication cost Table 3 summarizes the communication cost of our protocol. As the size of ciphertext is typically larger than that of plaintext, we only consider the ciphertexts sent and received by every party. It is important to note that the ciphertext of ElGamal encryption and re-encryption are twice and three times longer than the key, respectively. We omit the detailed analysis as the result is clear from the protocol.

3.4 Security analysis

The following analysis show the security of the proposed protocol.

Theorem 1 *Our task assignment protocol (Algorithm 1) is privacy-preserving with extra knowledge* $K_0 = \mathcal{V}$, $K_{-1} = \{\mathcal{V}, t_{f_k(1)}, \cdots, t_{f_k(n)}\}$ *and* $K_i = \mathcal{V}$ $(1 \leq i \leq n)$ *disclosure against the SC-server, CSP and all workers, respectively.*

Table 3 Communication cost of the proposed protocol. L and L' are the key size of Paillier and ElGamal cryptosystems, respectively

	SC-server	CSP	Worker
stage 1	$3L$	0	$3L$
stage 2	$nL + (2n+2)L'$	$2L'$	$L + 2L'$
stage 3	$nL + 2L'$	$nL + 2L'$	0
stage 4	0	$3nL'$	$3L'$
total	$(2n+3)L + (2n+4)L'$	$nL + (3n+4)L'$	$4L + 5L'$

Proof We first show that there is a probabilistic polynomial-time simulator S_0 which can simulate the SC-server's view based on $K_0 = \mathcal{V}$. As the SC-server's view is $view_0 = \{\mathrm{E}'(v_1), \cdots, \mathrm{E}'(v_n), \mathrm{E}(t_1'^2), \cdots, \mathrm{E}(t_n'^2), \mathrm{E}'_C(f_k(i^*)), \mathcal{V}\}$, S_0 generates $view'_0 = \{\mathrm{E}'(x_1), \cdots, \mathrm{E}'(x_n), \mathrm{E}(y_1), \cdots, \mathrm{E}(y_n), \mathrm{E}'(x_{n+1}), \mathcal{V}\}$ where x_i $(1 \le i \le n+1)$ are random elements uniformly distributed in G and y_i $(1 \le i \le n)$ are random elements uniformly distributed in $\mathbb{Z}_N$. It is easy to verify that $view_0 \equiv view'_0$ since Paillier and ElGamal are both semantic secure.

Next, we show that there is a probabilistic polynomial-time simulator S_i which can also simulate worker w_i's view based on $K_i = \mathcal{V}$. For w_i who is not the winner, $view_i = \{\mathrm{E}(x^2 + y^2), \mathrm{E}(x), \mathrm{E}(y), f_k(i), \overline{\mathrm{E}'}_C(\mathrm{E}'_{w_i}(f_k(i^*))), \mathcal{V}\}$ is her view. To simulate it, S_i generates $view'_i = \{\mathrm{E}(x_1), \mathrm{E}(x_2), \mathrm{E}(x_3), k, \overline{\mathrm{E}'}(\mathrm{E}'(y)), \mathcal{V}\}$ where x_i $(i = 1, 2, 3)$ are random elements uniformly distributed in $\mathbb{Z}_N$, y is randomly sampled from G, and k is a random element uniformly distributed over $\{0, 1\}^\lambda$. For w_{i^*}, her view is $view_{i^*} = \{\mathrm{E}(x^2 + y^2), \mathrm{E}(x), \mathrm{E}(y), f_k(i), i^*, \mathcal{V}\}$, so s_{i^*} generates $\{\mathrm{E}(x_1), \mathrm{E}(x_2), \mathrm{E}(x_3), k, i^*, \mathcal{V}\}$ as its $view'_{i^*}$. In both cases, we can verify that $view_i \equiv view'_i$ due to the semantic security of Paillier and ElGamal, as well as the pseudo-randomness of PRF.

Finally, we show that there is a probabilistic polynomial-time simulator S_{-1} which can simulate CSP's view based on $K_{-1} = \{\mathcal{V}, t_{f_k(1)}, \cdots, t_{f_k(n)}\}$. In our protocol, CSP's view is $view_{-1} = \{\mathrm{E}'_{w_1}(f_k(i)), \cdots, \mathrm{E}'_{w_n}(f_k(n))\} \bigcup K_{-1}$. To simulate it, S_{-1} generates $view'_{-1} = \{\mathrm{E}'(x_1), \cdots, \mathrm{E}'(x_n)\} \bigcup K_{-1}$ where x_i $(1 \le i \le n)$ are random elements uniformly distributed in G. $view_{-1} \equiv view'_{-1}$ clearly holds due to the semantic security of ElGamal. □

The above theorem proves our protocol is secure with K disclosure. Before showing K has limited effects on individual privacy, we give the following lemmas.

Lemma 2 *The product $\pi = \prod_{i=1}^n \tilde{x}_i$ is generated by randomly chosen integers $\tilde{x}_i \in \mathcal{Z}^+$ between 1 to d $(d > n)$. As $d \to \infty$, the equation $\prod_{i=1}^n x_i = \pi$ s.t. $\forall x_i \in \mathcal{Z}^+$ has at least $n!$ solutions with probability 1.*

Proof The probability for $\{\tilde{x}_1, \cdots, \tilde{x}_n\}$ are all different is given by

$$\eta(d, n) = \frac{P_d^n}{d^n} = \frac{d}{d}\frac{d-1}{d} \cdots \frac{d-n+1}{d}.$$

Any permutation of the sequence $\{\tilde{x}_1, \cdots, \tilde{x}_n\}$ is a valid solution. Therefore there are at least $n!$ different solutions to the equation $\prod_{i=1}^n x_i = \pi$ with probability $\eta(d, n)$, and we have $\lim_{d\to\infty} \eta(d, n) = 1$. □

Lemma 3 *The product π and a set of positive rational numbers $\{b_1, \cdots, b_n\}$ are generated by randomly chosen positive integers $\tilde{x}_1, \cdots, \tilde{x}_n$, $\tilde{y}_1, \cdots, \tilde{y}_n$ between 1 to d $(d > n)$ based on the equation*

$$\begin{cases} \prod_{i=1}^n x_i = \pi \\ \frac{y_i}{x_i} = b_{\sigma(i)} \\ \vdots \\ \frac{y_n}{x_n} = b_{\sigma(n)}, \end{cases}$$

where $(\sigma(1), \cdots, \sigma(n))$ is an unknown permutation of $(1, \cdots, n)$, then the equation has at least $n!$ solutions with probability 1 as $d \to \infty$.

Proof The proof is very similar to that of Lemma 2, by noting that as $d \to \infty$, with probability 1, $\tilde{x_1}, \cdots, \tilde{x_n}$ will be all different, and any permutation of a solution yields a different solution. □

Lemma 4 *Pick a random number in* $1, \cdots, d$*, the probability that it is prime is* $1/\log d$*, as* $d \to \infty$.

The above lemma is straightforward from the prime number theorem [18], which states that the number of primes before d converges to $d/\log d$ as $d \to \infty$.

Remark 2 Using Lemma 4, the probability for x_i being a prime or 1 can be approximated as $(1/\log d + 1/d)$. The probability for every x_i having at least two prime factors is therefore

$$\left(1 - \frac{1}{\log d} - \frac{1}{d}\right)^n \tag{11}$$

which converges to 1 at the limit $d \to \infty$. This means that with probability 1 the product π will have at least $2n$ prime factors, as long as d is chosen large enough. In practice, the number of solutions to the equation $\pi = \prod_{i=1}^{n} x_i$ is even larger than the stated $n!$.

Theorem 2 *Based on knowledge* K_i $(-1 \leq i \leq n)$*, the probability that an adversary* P_i *learns the private input of any party during an execution of the task assignment protocol (Algorithm 1) is negligible in general.*

Proof First consider P_0, the SC-server, which has the knowledge $K_0 = \mathcal{V}$. It can construct an equation $\prod_{i=1}^{n} v_i = \mathcal{V}$. Assume $1 \leq v_i \leq d$ and denote by $\eta(v_i)$ and $\eta(v_i|K_0)$ the probability that P_0 learns v_i and the probability that P_0 learns v_i given K_0, respectively. From Lemma 2, we have

$$\lim_{d\to\infty} (\eta(v_i|K_0) - \eta(v_i)) = \lim_{d\to\infty} (1/n! - 1/d) = 1/n!,$$

which is clearly negligible in general.

The proof for P_i $(1 \leq i \leq n)$ is similar, so we consider P_{-1} (i.e., CSP) now. As $K_{-1} = \{\mathcal{V}, t_{f_k(1)}, \cdots, t_{f_k(n)}\}$, it can construct a non-linear system which consists of the following $n + 1$ equations:

$$\begin{cases} \prod_{i=1}^{n} v_i = \mathcal{V} \\ \frac{d(l_s,l_1)}{v_1} = t_{f_k(1)} \\ \vdots \\ \frac{d(l_s,l_n)}{v_n} = t_{f_k(n)}. \end{cases}$$

From Lemma 3, we also have

$$\lim_{d\to\infty} (\eta(v_i|K_{-1}) - \eta(v_i)) = \lim_{d\to\infty} (1/n! - 1/d) = 1/n!,$$

which is negligible in general. Moreover, it is clear that CSP cannot know l_s and l_i with probability significantly better than random guess even if it knows the exact value of $d(l_s, l_i)$, which completes the proof. □

Remark 3 It is important to note that Theorem 2 says the privacy-preserving task assignment protocol is secure in general. In some extreme cases, for example, $\mathcal{V} = 1$, an adversary

can immediately know that the velocity of every worker is 1, but the probability that this happens drops dramatically when the number of workers grows.

3.5 Extension to task assignment with acceptance guarantee

The protocol shown in Algorithm 1 can be easily extended to solve $\mathbb{P}_{\text{PTAG}}$, the privacy-preserving task assignment with acceptance guarantee problem. In particular, only Line 24 of Algorithm 1 needs to be updated, as we need to find a winner set rather than a single winner. On obtaining the travel times of all workers (Line 23 of Algorithm 1), CSP sorts them in an ascending order and then keeps adding workers to the winner set one by one until the expected acceptance rate is achieved. Following [37], we also model worker's AR (acceptance rate) as a decreasing function ϕ of travel time and consider two cases: 1) *linear*, where AR decreases linearly with travel time starting from an initial MAR (maximum AR) value (when a worker is just in the task's location); and 2) *Zipf*, where AR follows Zipf distribution. Then, the termination condition of adding new workers to the winner set W^* is $1 - \prod_{w_i \in W^*}(1 - a_i) \geq \alpha$ where $a_i = \eta(t_{f_k(i)}, MAR)$.

It is easy to verify that our protocol is still secure with this modification when all parties are entitled to know $|W^*|$, that is , the number of winners. By assuming all workers have the same AR, we can compute the size of W^* is $\lceil \frac{\lg(1-\alpha)}{\lg(1-a_i)} \rceil$. Consequently, in stage 3, CSP needs to perform ElGamal encryption $|W^*|$ times and the communication cost between CSP and the SC-server changes from $2L'$ to $2|W^*|L'$.

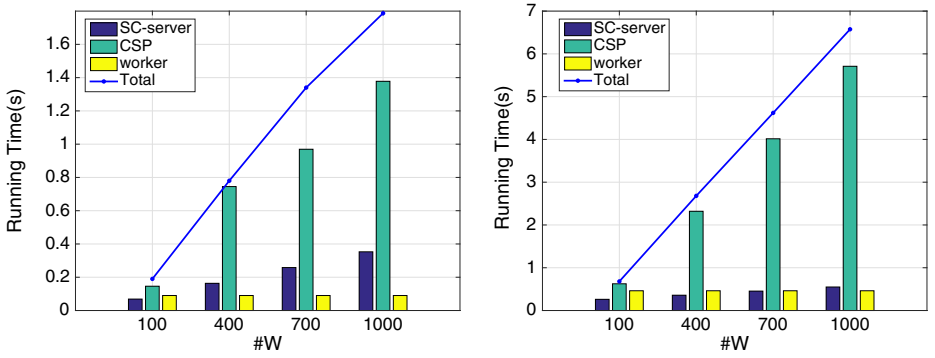
4 Performance evaluation

4.1 Experimental settings

We evaluate the performance of our basic protocol (Algorithm 1) and its extended version (Section 3.5) in terms of two categories of metrics: efficiency-related and effectiveness-related. The former includes running time and communication cost, while the latter includes worker travel distance (WTD), worker travel time (WTT), and the number of notified workers (NNW). Generally, workers perfer shorter WTD, and so do task requesters as their tasks can be performed early by assuming workers have the same velocity. If workers have different velocities, however, shorter WTD is not necessarily better. In this case, short WTT is preferred by both workers and task requesters. NNW should be kept low to decrease both computation cost and communication cost.

For effectiveness evaluation, we take To et al.'s approach [37] as the baseline. As their approach does not take velocity into account, the velocity of every worker is set to be 1 in the experiment. In this case, WTT is equal to WTD. Besides, the deadline of every task is set to be a large value so that all workers can arrive before the deadline. As our basic protocol does not consider worker acceptance rate and always returns one worker (i.e., NNW always equal 1), we only report the comparison results between the extended version and the baseline. We randomly generate 1,000 tasks and report the average results.

For efficiency evaluation, we notice that differential privacy is clearly much computationally cheaper than public-key cryptosystems, but it cannot protect data during computation (e.g., a trusted third party is allowed to see all workers' locations). It is thus meaningless to compare our protocl (based on public-key cryptosystems) with To et al.'s approach (based on differential privacy) in terms of running time. Therefore, we focus only on the efficiency



(a) key length = 1024 (b) key length = 2048

Fig. 3 Effect of number of workers on running time

of our basic protocol and its extension, testing whether their overheads could be accepted in practice. We execute our protocol 10 times and report the average results.

We use two real-world datasets, Gowalla[1] and Yelp,[2] for performance evaluation. Gowalla contains the check-in history of users in a location-based social network. We select an area in California with latitude from 33.720183 to 34.149932 and longitude from -118.399999 to -117.900516. This area has the check-ins of 5,830 users who are assumed to be the workers in a spatial-crowdsourcing system. We take the location where a user has most check-ins as her current location, and assume that a spatial task can be created in any location that has check-in. For Yelp, we select an area in Phoenix with latitude from 33.205308 to 33.924407 and longitude from -112.400283 to -111.218100. This area has about 67,000 users and 11,200 businesses. Business locations are regarded as tasks while the location of a user is randomly selected from businesses that she has a review.

We set the number of workers $\#W \in \{100, 400, 700, \mathbf{1000}\}$, the maximum acceptance rate $MAR \in \{0.4, 0.6, \mathbf{0.8}, 1\}$, and the expected task acceptance probability $\alpha \in \{0.7, 0.8, \mathbf{0.9}, 0.99\}$. As the baseline relies on differential privacy whose performance is based on a privacy budget ϵ, we also set $\epsilon \in \{0.1, 0.4, \mathbf{0.7}, 1.0\}$, as proposed in [37]. For security parameter of Paillier and ElGamal, we refer to NIST recommendations (2016)[3] and set the key length $KL \in \{\mathbf{1024}, 2048\}$, where 1024 is appropriate for current applications and 2048 is recommended for next 15 years (2016-2030). The default value for each parameter is shown in boldface.

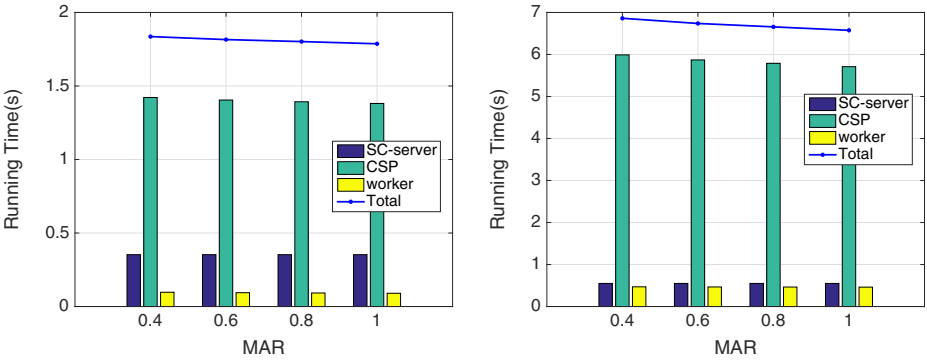
In our experiment, the SC-server and CSP are run on a machine with four Intel Xeon E7-8860 2.2GHz CPUs (each CPU has 16 cores) and 1TB RAM, and a worker is simulated by a Mi 2 cellphone with APQ 8064 1.5GHz CPU and 2GB RAM. We implement our protocol using the Bouncy Castle Crypto package.[4] The code is written in Java and executed in JDK 1.8. From Table 2, the performance bottleneck of our protocol is a number of Paillier decryptions. Fortunately, these expensive operations can be parallelized easily as they are

[1] https://snap.stanford.edu/data/loc-gowalla.html

[2] https://www.yelp.com/dataset_challenge

[3] https://www.keylength.com/

[4] https://www.bouncycastle.org/java.html



(a) key length = 1024 (b) key length = 2048

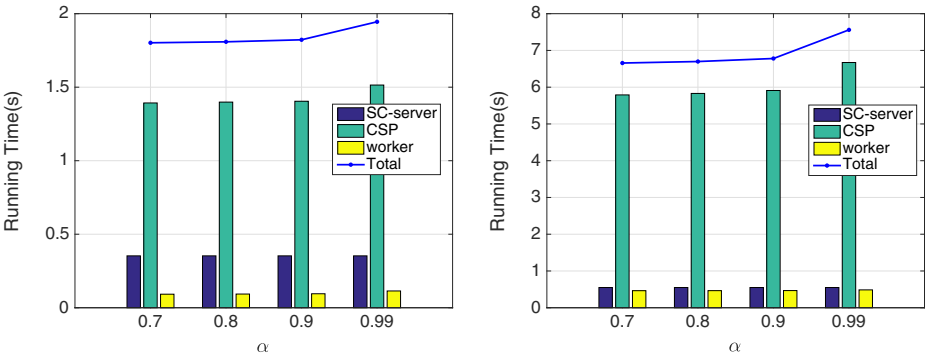
Fig. 4 Effect of MAR on running time

performed on independent values. In our experiment, we use 64 threads to perform these decryptions.

4.2 Experimental results

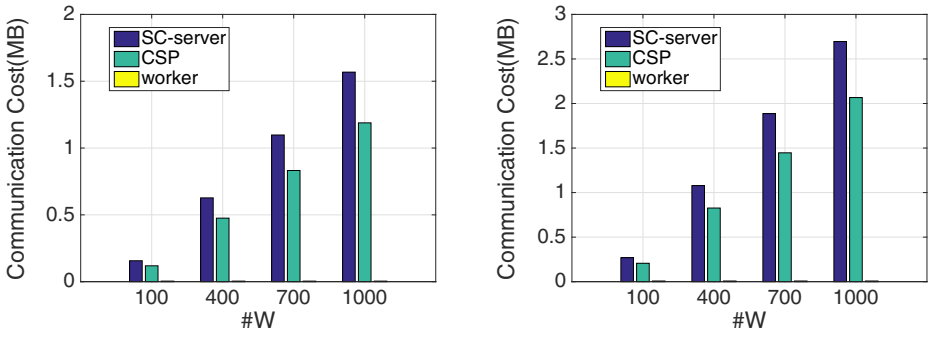
4.2.1 Efficiency

Figure 3a shows the running time of our basic protocol when the number of workers $\#W$ increases from 100 to 1,000 with a step of 300. As expected, when $\#W$ increases, the CPU times of SC-server and CSP also increase, but in a linear manner, as their computation costs are mainly dominated by cryptographical operations whose number is proportional to the number of workers. On the other hand, the computation cost of a worker is almost a constant, for example, about 0.1 seconds in a moderate cellphone, despite of the number of workers. Therefore, our protocol has a good scalability in practice. In terms of total running time, our protocol only needs less than 2 seconds to achieve privacy-preserving task assignment over 1,000 workers. Similar performance trend can be observed in Fig. 3b where 2048-bits



(a) key length = 1024 (b) key length = 2048

Fig. 5 Effect of α on running time



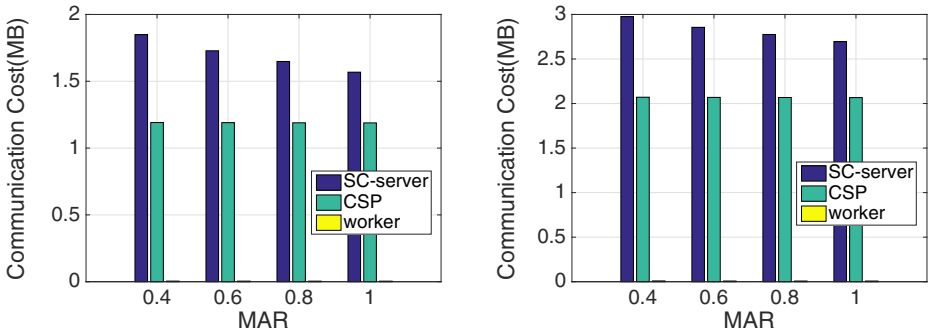
(a) key length = 1024 (b) key length = 2048

Fig. 6 Effect of number of workers on communication cost

keys are used to provide much stronger security guarantee (recall that this key length is recommended for next 15 years). Even in this case, the total running time of our protocol is less than 7 seconds.

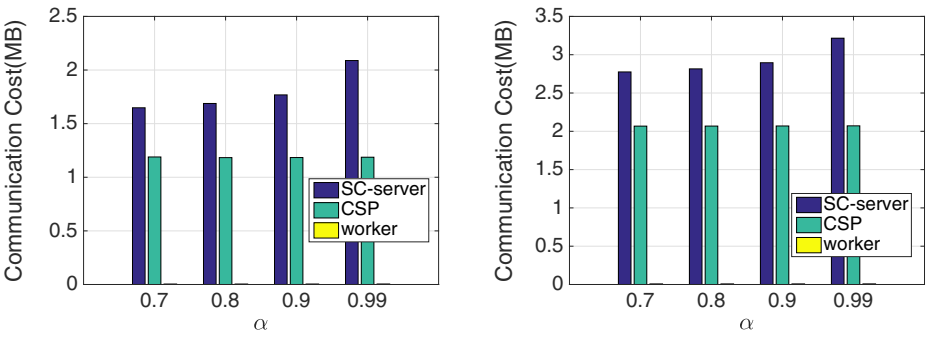
Figures 4 and 5 depict the running time of the extended protocol by varying MAR and α, respectively. Overall, the extended protocol only introduces limited overheads to provide specified acceptance guarantee. For example, to make $\alpha = 0.9$, our protocol needs about 1.79 seconds when $MAR = 1$, but needs only about 1.84 seconds when MAR decreases to 0.4 (see Fig. 4a). For another example, when $MAR = 0.8$, our protocol can find a winner set with $\alpha = 0.7$ in 1.81 seconds. To ensure tasks can be accepted with a very high probability, say $\alpha = 0.99$, our protocol only needs 1.94 seconds. The reason for this performance is that the extra overheads mostly come from ElGamal encryptions and the number of encryptions is bounded by the size of the winner set, which is typically small (more results can be found in Figs. 12, 13 and 14).

In Fig. 6, we measure the communication cost of different parties in our basic protocol. From Fig. 6b, the SC-server, CSP, and worker need to send or receive 2.7, 2.1, and 0.008 MB data, respectively, when performing a task assignment over 1,000 workers with 2048-bits keys. We believe this overheads is not a burden at all in current mobile applications. By



(a) key length = 1024 (b) key length = 2048

Fig. 7 Effect of MAR on communication cost



(a) key length = 1024 (b) key length = 2048

Fig. 8 Effect of α on communication cost

varying the number of workers from 100 to 1,000, we observe in Fig. 6 a linear increase trend for both SC-server and CSP, as the transferred data are largely ciphertexts whose size is proportional to the number of workers. Besides, a worker's communication cost remains to be a small constant, as explained in Section 3.3. We further investigate the communication cost of the extended protocol by varying MAR and α, and report the results in Figs. 7 and 8. For all three parties, their communication costs have a very small increase due to the declaration of multiple winners. In summary, our protocol is also scalable in terms of communication cost.

4.2.2 Effectiveness

Figures 9, 10 and 11 show the performance of our protocol in terms of WTD by varying MAR, α, and ϵ, respectively. In all figures, our protocol performs better than the baseline in all combinations of datasets (Gowalla, Yelp) and acceptance rate functions (Linear, Zipf). Specifically, in Fig. 9, we observe that the difference between our protocol and the baseline increases when MAR decreases. To explain this, we first note that the baseline needs to visit more grid cells to achieve required acceptance rate. Every cell generally contains a number of workers. Some of them may be far from the task location but they can accept the task. Our protocol, however, always selects workers based on their travel times (or travel distances in this case). That is why our protocol is much better than the baseline when MAR is small. The similar result can be observed in Fig. 10. Figure 11 shows that the baseline has a larger

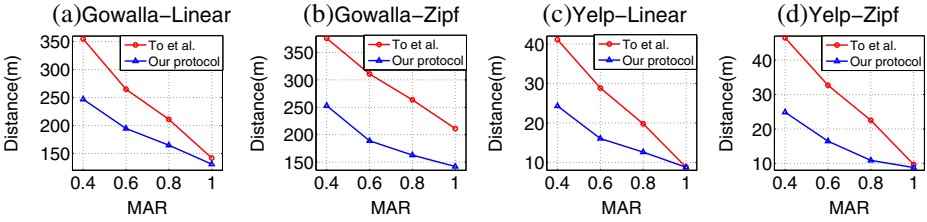


Fig. 9 Effect of MAR on WTD

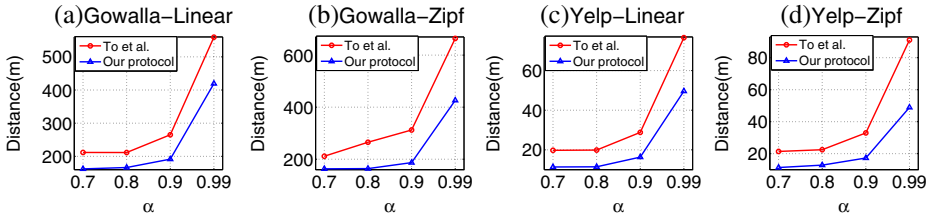


Fig. 10 Effect of α on WTD

WTD when providing stronger privacy guarantee (e.g., $\epsilon = 0.1$). However, our protocol still outperforms the baseline even if it only provides weak privacy guarantee (e.g., $\epsilon = 1$).

We further evaluate the performance of our protocol in terms of NNW by varying MAR, α, and ϵ, and report the results in Figs. 12, 13 and 14, respectively. Again, our protocol performs better than the baseline in all combinations of datasets (Gowalla, Yelp) and acceptance rate functions (Linear, Zipf). In most cases, the number of notified workers is not larger than 5. In some extreme cases, for example, $\alpha = 0.99$, our protocol selects less than 15 workers to perform a task. This can explain why our protocol can be extended to $\mathbb{P}_{\text{PTAG}}$ with very low overheads. On the other hand, the baseline needs to notify a lot of workers since it works on gird cells.

5 Related work

SC is a complex procedure that generally consists of four phases: task and workers registration, task assignment, answer aggregation, and response and quality control [2]. Here we only review the works that are relevant to task assignment, referring the reader to [2] for more results on other phases of SC. Kazemi and Shahabi [20] propose several heuristics to maximize the overall task assignment while conforming to the constraints of workers. Similarly, Deng et al. [7] 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 also taken into account in the course of task assignment. Cheng et al. [3] shows task assignment with these constraints is NP-hard and proposes several approximation algorithms. In [4], efficient methods are designed to assign workers to complex tasks that require more than one skill. Tong et al. [40] consider task assignment in online scenarios and propose efficient algorithms with provable competitive ratio.

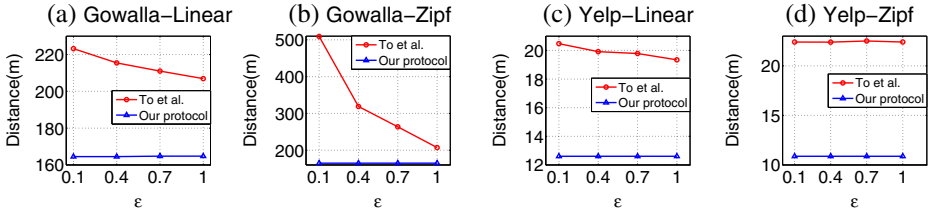


Fig. 11 Effect of ϵ on WTD

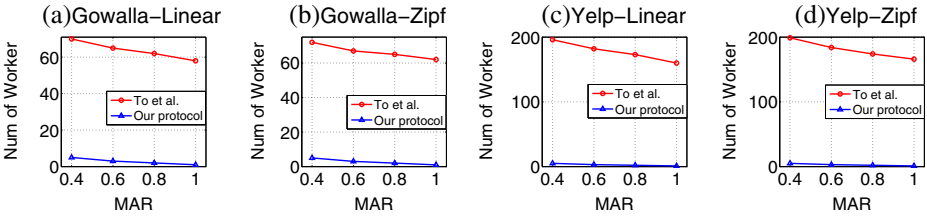


Fig. 12 Effect of MAR on NNW

Zheng and Chen [48] tackle the problem of assigning tasks to workers such that mutual benefit are maximized. All these works assume that workers are willing to give their private location information to the SC-server that is typically untrusted in practice. Our work complements these works by tackling the privacy leakage problem in the phase of task assignment.

Location privacy protection has been studied extensively in recent years. Ghinita et al. [16] adopt private information retrieval (PIR) to enable users to conduct approximate and exact nearest neighbor search without revealing their locations to the server. Paulet et al. [28, 29] combines PIR and oblivious transfer (OT) to achieve mutual privacy-preserving location-based queries. On one hand, the server is unable to know the location of users. On the other hand, users can only get a limited location data for their queries, thus protecting the server's private data. Liu et al. [24] propose a more efficient approach for this problem by utilizing two rounds of OT and show the efficiency improvement can be realized at the expense of acceptable communication cost. Yi et al. [44] present a solution based on Paillier and Rabin cryptosystem for mutual privacy-preserving kNN query where k is fixed. The solution is extended in [45] to support dynamic k up to a constant and sequential queries. However, these solutions cannot be applied to our scenario. This is because, in SC, worker location is not the private data of the SC-server, but rather the sensitive information that workers want to hide from the SC-server. There are also some works focusing on privacy-preserving location-based queries over outsourced location data [43, 46], where the data owner and users sending queries are assumed to trust each other. In SC, however, there is no inherent trust relationship between task requesters and workers. To enable kNN query over encrypted data, Elmehdwi et al. [11] propose a set of protocols based on Paillier. While mutual privacy can be guaranteed due to the security of Paillier, the computation cost of these protocols are very expensive [23]. Thus, we cannot apply these protocols to directly solve large task assignment problems.

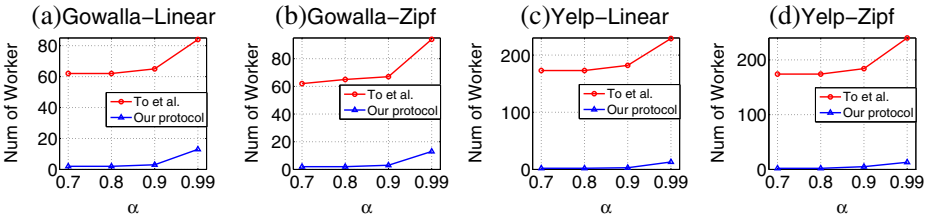


Fig. 13 Effect of α on NNW

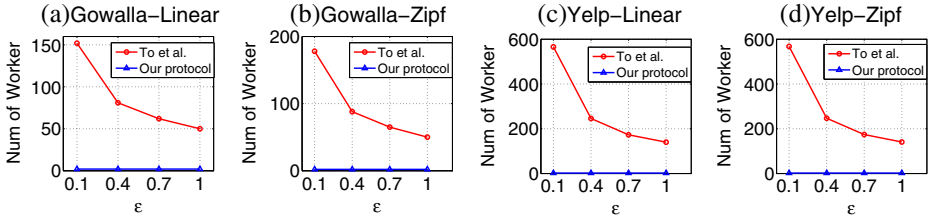


Fig. 14 Effect of ϵ on NNW

6 Conclusion

In this paper, we have identified mutual privacy protection requirement in the course of spatial crowdsourcing. We have presented a privacy-preserving task assignment protocol which makes effective decision based on multiple factors. We have theoretically proved that our approach is secure against semi-honest adversaries. We have conducted extensive experiments on two real-world datasets. Experimental results have shown that our protocol is more effective than state-of-the-art solutions in terms of task assignment quality, and the computation and communication overheads caused by privacy protection are acceptable in practice.

Acknowledgements Research reported in this publication was partially supported by KAUST and Natural Science Foundation of China (Grant Nos. 61572336, 61572335, 61632016, 61402313, 61472337), and has been benefited from discussions with Dr. Ke Sun in MINE lab at KAUST.

References

1. Castelluccia C, Chan AC, Mykletun E, Tsudik G (2009) Efficient and provably secure aggregation of encrypted data in wireless sensor networks. ACM Transactions on Sensor Networks (TOSN) 5(3):20
2. Chen L, Shahabi C (2016) Spatial crowdsourcing: challenges and opportunities. IEEE Data Eng Bull 39(4):14–25
3. Cheng P, Lian X, Chen Z, Fu R, Chen L, Han J, Zhao J (2015) Reliable diversity-based spatial crowdsourcing by moving workers. Proc VLDB Endowment 8(10):1022–1033
4. Cheng P, Lian X, Chen L, Han J, Zhao J (2016) Task assignment on multi-skill oriented spatial crowdsourcing. IEEE Trans Knowl Data Eng 28(8):2201–2215
5. Cooper S, Khatib F, Treuille A, Barbero J, Lee J, Beenen M, Leaver-Fay A, Baker D, Popović Z (2010) Predicting protein structures with a multiplayer online game. Nature 466(7307):756–760
6. Dai W (2010) Commutative-like encryption: a new characterization of ElGamal. arXiv:1011.3718
7. Deng D, Shahabi C, Demiryurek U, Zhu L (2016) Task selection in spatial crowd- sourcing from workers perspective. GeoInformatica 20(3):529–568
8. DigitalGlobe: http://blog.digitalglobe.com/2015/05/29/digitalglobe-winding-down-nepal-earthquake-response-activities/
9. Dwork C (2008) Differential privacy: a survey of results. TAMC, pp 1–19
10. ELGAMAL T (1985) A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans Inf Theory 31(4):469–472
11. Elmehdwi Y, Samanthula BK, Jiang W (2014) Secure k-nearest neighbor query over encrypted data in outsourced environments. ICDE, pp 664–675
12. Feng Y, Wang J, Zhang Z, Zhong H, Ming Z, Yang X, Mao R (2016) The edge weight computation with mapreduce for extracting weighted graphs. IEEE Trans Parallel Distrib Syst 27(12):3659–3672
13. GalaxyZoo: https://www.galaxyzoo.org/
14. Gentry C (2009) Fully homomorphic encryption using ideal lattices. STOC, pp 169–169

15. Gentry C, Halevi S (2011) Implementing Gentry's fully-homomorphic encryption scheme. EUROCRYPT, pp 129–148
16. Ghinita G, Kalnis P, Khoshgozaran A, Shahabi C, Tan KL (2008) Private queries in location based services: anonymizers are not necessary. SIGMOD, pp 121–132
17. Goldreich O (2004) Foundations of cryptography: volume 2, basic applications. Cambridge University Press
18. Hardy GH, Littlewood JE (1916) Contributions to the theory of the Riemann zeta-function and the theory of the distribution of primes. Acta Mathematica 41(1):119–196
19. Howe J (2006) Crowdsourcing: a definition http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html
20. Kazemi L, Shahabi C (2012) Geocrowd: enabling query answering with spatial crowdsourcing. SIGSPATIAL, pp 189–198
21. Liu A, Li Q, Huang L, Xiao M (2010) FACTS: a framework for fault-tolerant composition of transactional web services. IEEE Trans Serv Comput 3(1):46–59
22. Liu A, Li Q, Huang L, Ying S, Xiao M (2013) Coalitional game for community-based autonomous web services cooperation. IEEE Trans Services Computing 6(3):387–399
23. Liu A, Zheng K, Li L, Liu G, Zhao L, Zhou X (2015) Efficient secure similarity computation on encrypted trajectory data. ICDE, pp 66–77
24. Liu S, Liu A, Zhao L, Liu G, Li Z, Zhao P, Zheng K, Qin L (2016) Efficient query processing with mutual privacy protection for location-based services. DASFAA, pp 299–313
25. Mao R, Xu H, Wu W, Li J, Li Y, Lu M (2015) Overcoming the challenge of variety: big data abstraction, the next evolution of data management for AAL communication systems. IEEE Commun Mag 53(1):42–47
26. Mao R, Zhang P, Li X, Liu X, Lu M (2016) Pivot selection for metric-space indexing. Int J Mach Learn Cybern 7(2):311–323
27. Paillier P (1999) Public-Key cryptosystems based on composite degree residuosity classes. EUROCRYPT, pp 223–238
28. Paulet R, Kaosar MG, Yi X, Bertino E (2012) Privacy-Preserving and Content-Protecting Location Based Queries. ICDE, pp 44–53
29. Paulet R, Kaosar MG, Yi X, Bertino E (2014) Privacy-preserving and content-protecting location based queries. IEEE Trans Knowl Data Eng 26(5):1200–1210
30. Shang S, Yuan B, Deng K, Xie K, Zheng K, Zhou X (2012) PNN query processing on compressed trajectories. GeoInformatica 16(3):467–496
31. Shang S, Ding R, Yuan B, Xie K, Zheng K, Kalnis P (2012) User oriented trajectory search for trip recommendation. EDBT, pp 156–167
32. Shang S, Ding R, Zheng K, Jensen CS, Kalnis P, Zhou X (2014) Personalized trajectory matching in spatial networks. VLDB J 23(3):449–468
33. Shang S, Liu J, Zheng K, Lu H, Pedersen TB, Wen JR (2015) Planning unobstructed paths in traffic-aware spatial networks. GeoInformatica 19(4):723–746
34. Shang S, Zheng K, Jensen CS, Yang B, Kalnis P, Li G, Wen JR (2015) Discovery of path nearby clusters in spatial networks. IEEE Trans Knowl Data Eng 27(6):1505–1518
35. Shang S, Chen L, Wei Z, Jensen CS, Wen JR, Kalnis P (2016) Collective travel planning in spatial networks. IEEE Trans Knowl Data Eng 28(5):1132–1146
36. Shove E, Trentmann F, Wilk R (2009) Time, consumption and everyday life: practice, materiality and culture. Berg
37. To H, Ghinita G, Shahabi C (2014) A framework for protecting worker location privacy in spatial crowdsourcing. PVLDB 7(10):919–930
38. To H, Ghinita G, Shahabi C (2015) Privgeocrowd: a toolbox for studying private spatial crowdsourcing. ICDE, pp 1404–1407
39. To H, Shahabi C, Kazemi L (2015) A server-assigned spatial crowdsourcing framework. ACM Transactions on Spatial Algorithms and Systems 1(1):2
40. Tong Y, She J, Ding B, Wang L, Chen L (2016) Online mobile micro-task allocation in spatial crowdsourcing. ICDE, pp 49–60
41. Von Ahn L, Maurer B, McMillen C, Abraham D, Blum M (2008) Recaptcha: Human-based character recognition via web security measures. Science 321(5895):1465–1468
42. Waze: https://www.waze.com/
43. Yao B, Li F, Xiao X (2013) Secure nearest neighbor revisited. ICDE, pp 733–744
44. Yi X, Paulet R, Bertino E, Varadharajan V (2014) Practical k nearest neighbor queries with location privacy. ICDE, pp 640–651

45. Yi X, Paulet R, Bertino E, Varadharajan V (2016) Practical approximate k nearest neighbor queries with location and query privacy. IEEE Trans Knowl Data Eng 28(6):1546–1559
46. Yiu ML, Ghinita G, Jensen CS, Kalnis P (2010) Enabling search services on outsourced private spatial data. VLDB J 19(3):363–384
47. Zhang D, Lu K, Mao R, Feng Y, Liu Y, Ming Z, Ni LM (2014) Fine-grained localization for multiple transceiver-free objects by using RF-based technologies. IEEE Trans Parallel Distrib Syst 25(6):1464–1475
48. Zheng L, Chen L (2016) Mutual benefit aware task assignment in a bipartite labor market. ICDE, pp 73–84

An Liu received his Ph. D. degree in computer science from both City University of Hong Kong and University of Science and Technology of China in 2009. He is currently a postdoctoral fellow at King Abdullah University of Science and Technology (KAUST). His research interests include spatial databases, data mining, data security and privacy. He has published more than 70 papers in referred journals and conferences, including IEEE TKDE, IEEE TSC, ICDE, WWW etc.

Weiqi Wang is a master student in the Department of Computer Science and Technology at Soochow University. His main research interests include data privacy and data management.

Shuo Shang is a research scientist in King Abdullah University of Science and Technology. His research interests include efficient query processing in spatio-temporal databases, spatial trajectory computing, and location based social media. He is/was on Demo Chair of APWeb/WAIM Joint Conference 2017, on Session Chair (session of moving objects) of ICDE 2013, and on PC member of SIGMOD 2018, CIKM 2017, and DASFAA 2015 and 2016. He is on the reviewer board of several top database/data mining journals such as IEEE TKDE, The VLDB Journal, ACM TIST, ACM TSAS, IEEE TITS, Geoinformatica, KAIS, DKE, WWW Journal, JCST, and IEICE Transactions.

Qing Li received the B.Eng. degree from Hunan University, Changsha, China, and the M.Sc. and Ph.D. degrees from the University of Southern California, Los Angeles, all in computer science. He is currently a Professor with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. His current research interests include dynamic object modeling, multimedia and mobile information retrieval and management, distributed databases and data warehousing/mining, and workflow management and web services.

Xiangliang Zhang is an Associate Professor and directs the Machine Intelligence and kNowledge Engineering (MINE) Laboratory in King Abdullah University of Science and Technology (KAUST). She earned her Ph.D. degree in computer science from INRIA-University Paris-Sud 11, France, in July 2010. Her main research interests and experiences are in diverse areas of machine learning and data mining. She has published over 70 papers in referred journals and conferences, including TKDE, SIGKDD, VLDB J, AAAI, IJCAI, ICDM, ECML/PKDD, CIKM, InfoCom etc.