Towards Privacy-Preserving Travel-Time-First Task Assignment in Spatial Crowdsourcing

Jian Li[1], An Liu[1](✉), Weiqi Wang[1], Zhixu Li[1], Guanfeng Liu[1], Lei Zhao[1], and Kai Zheng[2]

[1] School of Computer Science and Technology, Soochow University, Suzhou, China
anliu@suda.edu.cn

[2] University of Electronic Science and Technology of China, Chengdu, China

Abstract. With the ubiquity of mobile devices and wireless networks, spatial crowdsourcing (SC) has gained considerable popularity and importance as a new tool of problem-solving. It enables complex tasks at specific locations to be performed by a crowd of nearby workers. In this paper, we study the privacy-preserving travel-time-first task assignment problem where tasks are assigned to workers who can arrive at the required locations first and no private information are revealed to unauthorized parties. Compared with existing work on privacy-preserving task assignment, this problem is novel as tasks are allocated according to travel time rather than travel distance. Moreover, it is challenging as secure computation of travel time requires secure division which is still an open problem nowadays. Observing that current solutions for secure division do not scale well, we propose an efficient algorithm to securely calculate the least common multiple (LCM) of every workers speed, based on which expensive division operation on ciphertexts can be avoided. We formally prove that our protocol is secure against semi-honest adversaries. Through extensive experiments over real datasets, we demonstrate the efficiency and effectiveness of our proposed protocol.

Keywords: Spatial crowdsourcing · Privacy-preserving Task assignment

1 Introduction

Thanks to the ubiquitous wireless networks and powerful mobile devices, spatial crowdsourcing has gained considerable popularity and importance as a new tool of problem-solving. It can be applied to simple tasks such as photo-taking where people act as sensors, or to complex tasks such as handyman service where people work as intelligent processing units. As an emerging crowdsourcing mode, spatial crowdsourcing differs from other crowdsourcing modes in that people in spatial crowdsourcing, also known as workers, must physically move to certain places to perform those spatial tasks. Recently years have witnessed an upsurge of interest

Y. Cai et al. (Eds.): APWeb-WAIM 2018, LNCS 10988, pp. 19–34, 2018.
https://doi.org/10.1007/978-3-319-96893-3_2

in spatial crowdsourcing applications in daily life, ranging from local search-and-discovery (e.g., Foursquare) to home repair and refresh (e.g., TaskRabbit).

A typical workflow of spatial crowdsourcing consists of four steps: task/worker registration, task assignment, answer aggregation, and quality control [1]. Among them, task assignment focuses on allocating a set of tasks to a set of workers according to a set of constraints such as location, time, and budget. Typically finding an optimal assignment subject to multiple constraints is NP-hard, which calls for efficient yet effective algorithms. Based on specific optimization goals, a variety of approaches have been proposed, for example, to maximize the total number of completed tasks [2], to maximize the number of tasks performed by a single worker [3] and to maximize the reliability-and-diversity score of assignments [4].

The problem of task assignment becomes even tougher when privacy issues are taken into account. It is not hard to see that the data used for decision making in task assignment is usually private and thus need to be kept secret due to the lack of trust among workers, task requesters, and the spatial crowdsourcing server. To achieve privacy, these private data should be protected by for example encryption using mature cryptographical algorithms or perturbation using emerging privacy-preserving techniques. However, the noise introduced by these mechanisms will decrease significantly the utility of the data and sometimes even will make the data useless. It is therefore more challenging to deal with task assignment with the extra privacy constraint.

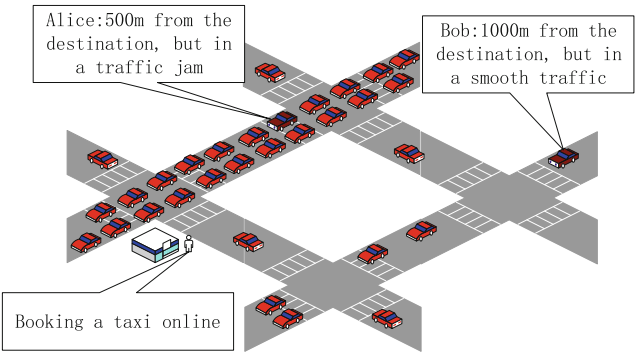

Fig. 1. Spatial crowdsourcing where travel time is more important than travel distance

The above hard problem has been studied by several work recently [5–8]. A common strategy of task assignment adopted by these work is travel-distance-first, that is, a task will be assigned to the worker who has the shortest travel distance to its location. This strategy is simple but sometimes is unreasonable in practice as it is common for some workers to move faster than others. Consider a simple example where a user wants to request a car through a spatial crowdsourcing platform (e.g., Uber). As shown in Fig. 1, two workers (i.e., drivers)

named Alice and Bob are available when the task is issued, and their distances to the user are 500 m and 1,000 m, respectively. Using the aforementioned strategy, the spatial crowdsourcing server will assign this task to Alice as she is nearer to the user. As shown in the figure, however, Alice is in a traffic jam. On the other hand, Bob is in a smooth traffic and he can arrive at the users location before Alice. This simple example motivates us to consider, travel-time-first, a more effective strategy when allocating tasks to workers in practice.

In this paper, we propose a privacy-preserving task assignment protocol for spatial crowdsourcing platforms taking travel-time-first strategy, that is, workers who can first arrive at the location of a given task have priority over others. While it is more effective than travel-distance-first in practice, travel-time-first makes privacy-preserving task assignment more challenging due to the required division operation involved in the computation of travel time. For every user, his/her location and speed are both private and should be protected. How to perform division efficiently and accurately on encrypted or perturbed data is still an open problem. In [9], the authors propose a protocol for secure division based on ElGamal cryptosystem. However, this protocol does not scale well and cannot be applied to large spatial crowdsourcing system for the key length should be set large enough to avoid computation overflow and this will introduce prohibitive computation cost. To overcome this weakness, we transform the secure division problem into a secure least common multiple (LCM) problem. We propose an efficient way to calculate the LCM securely. Through extensive experiments, we demonstrate the feasibility and efficiency of our solution.

The remainder of this paper is organized as follows: Sect. 2 discusses related work. Problem definition and background knowledge are presented in Sect. 3. Section 4 introduces our approach in details. Section 5 analyzes the security and complexity of our approach theoretically. Section 6 evaluates our approach on real datasets. Section 7 concludes the paper.

2 Related Work

To be consistent with our contributions, we only review the works that are relevant to task assignment and privacy-preserving. Kazemi and Shahabi [2] propose several solutions to maximize the overall number of assigned tasks under the constraints of workers. Similarly, The assignment protocol proposed by [10] is to assign the time-constrained and multi-skill-required spatial tasks with dynamically moving workers. In [11], Zheng et al. take workers' rejection into consideration and try to maximize workers' acceptance in order to improve the system throughput. Tong et al. [12] devise efficient algorithms with provable competitive radio with online dynamic scenarios. And in [13], Tong et al. propose an online task assignment framework based on offline guidance to maximize the task allocation while maintaining the efficient task assignment. In [14], Gao et al. design a two-level-based framework to recommend suitable teams to accomplish a task. However, these works are all based on a pre-condition that workers do not refuse to disclose their private information to the SC platform that is hard to achieve

in reality. Our work focuses on privacy-preserving during an execution of task assignment.

In recent years, the public concern over privacy has stimulated lots of research efforts in privacy-preserving. A location based query solution is proposed by Paulet et al. [15] that employs two protocols that enables a user to privately determine and acquire location data. In [16], Liu et al. propose an efficient approach to protecting mutual privacy in location-based queries by performing two rounds of oblivious transfer (OT) extension on two small key sets. A solution built on the Paillier public-key cryptosystem is presented by Yi et al. [17] for mutual privacy-preserving kNN query with fixed k and is extended in [18] where k is dynamic. Unfortunately, these solutions where workers location are private data of the SC platform are not suitable for our framework for workers location should be known to the SC platform in a secret way. Also, in [19], Sun et al. focus on the privacy-preserving task assignment in SC by presenting an approach where location privacy of workers can be protected in a k-annoymity manner. In [5], To et al. propose a framework for protecting location privacy of workers participating in SC tasks without protecting task location. Liu et al. [20] propose an efficient solution to securely compute the similarity between two encrypted trajectories without revealing nothing about the trajectories. However, their protocols also cannot be applied to our framework for they have too heavy computation cost to solve large task assignment problems.

3 Problem and Preliminary

In this section, we first present some definitions used in our work and then briefly introduce some cryptosystems based on which our protocol is built.

3.1 Problem Definitions

Definition 1 *(Spatial Task). A spatial task, denoted as T, is a task to be performed l_T.*

Definition 2 *(Workers). Let $W = \{w_1, \cdots, w_n\}$ be a set of n workers. Each worker w has an ID id_w, a location l_w, a constant speed s_w, and an acceptance rate AR_w which is the probability that he/she accepts a task assigned to him/her.*

As mentioned in the introduction, we mainly consider travel-time-first, a new task assignment strategy in privacy-preserving spatial crowdsourcing. Ideally, we only need to find a worker $w \in W$ who can first arrive at l_T and then assign T to w. This works if the worker is certain to accept the assigned task, but sometimes it is not. Therefore we consider a more general case where every worker w has an acceptance ratio denoted as AR_w for an assignment, and we need to ensure the probability that a task T is accepted by at least one worker is larger than a given threshold α_T. In this case, we need to find a set of workers $U \subset W$ rather than a single worker. It is easy to see that the probability that T is accepted by at least one worker in U is $\alpha_U = 1 - \prod_{w \in U}(1 - AR_w)$. Hence the travel-time-first task assignment problem can be formalized as follows:

Definition 3 *(Travel-time-first Task Assignment Problem). Given a set of workers W, a task T and its acceptance threshold α_T, the travel-time-first task assignment assigns task T to a set of workers $U \subset W$ such that:*

$$\frac{d(l_i, l_T)}{s_i} \leq \frac{d(l_j, l_T)}{s_j} \quad and \quad \alpha_T \leq \alpha_U \tag{1}$$

for $\forall i \in U$ and $\forall j \in W \setminus U$.

Privacy-preserving means all the private data should be hidden from unauthorized parties in the procedure of task assignment. To accurately define the ability of unauthorized parties, we adopt a typical adversary model, i.e., the semi-honest model [21]. Specifically, all parties in this model are assumed to be semi-honest, that is, they follow a given protocol exactly as specified, but may try to learn as much as possible about other parties private input from what they see during the protocols execution. 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. Specifically, the security of a protocol Π is defined as follows:

Definition 4. *Let $p_i(1 \leq i \leq n)$ be n parties involved in a protocol Π. For $p_i(1 \leq i \leq n)$, its view, private input and extra knowledge it can infer during an execution of P_i are defined as V_i,X_i and K_i respectively. A protocol Pi has a strong privacy guarantee, that is, p_i cannot learn any knowledge except the final output of p_i, if these exists a probabilistic polynomial-time simulator P_i such that:*

$$P_i(X_i, \Pi(X_1, \cdots, X_n), K_i)_{X_1, \cdots, X_n} \equiv V_i(X_1 \cdots, X_n)_{X_1 \cdots, X_n} \tag{2}$$

and $K_i = \emptyset$, where $\equiv$ means computational indistinguishability. However, this strong guarantee cannot be achieved sometimes for $K_i \neq \emptyset$. If $K_i \neq \emptyset$, Π 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.

Now we are ready to define the problem of privacy-preserving travel-time-first task assignment as follows:

Definition 5 *(Privacy-preserving Travel-time-first Task Assignment Problem). Given a set of workers W, a task T and its acceptance threshold α_T, the travel-time-first task assignment assigns task T to a set of workers $U \subset W$ such that Eqs. (1) and (2) hold.*

3.2 Cryptosystems

The privacy-preserving property of our protocol is built on several well-known cryptosystems: PRG [22], Paillier [23] and ElGamal [24]. The details of PRG, Paillier and ElGamal can be found in the given references and all of them are proved to be secure. Here we only emphasize some important properties of these cryptosystems.

PRG can be implemented by using a one-way hash function denoted as G_k. For Paillier, its encryption and decryption are denoted as E_p and D_p, respectively. For ElGamal, its encryption and decryption are denoted as E_e and D_e, respectively. The important properties of Paillier and ElGamal are listed as follows:

Homomorphic Properties of Paillier: Given m_1 and m_2 are two messages, we have:

$$E_p(m_1)E_p(m_2) = E_p(m_1 + m_2). \tag{3}$$

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

Commutative-Like Property of ElGamal: Given a message m, we have:

$$E_e^{h_a}(E_e^{h_b}(m)) = E_e^{h_b}(E_e^{h_a}(m)). \tag{5}$$

4 Proposed Privacy-Preserving Framework

In this section, we will introduce our privacy-preserving framework in details and explain how to get LCM in a safe and secret way by AP encryption strategy.

4.1 Framework Overview

As Fig. 2 shows, our proposed framework consists of six stages, namely Initialization, Distance, LCM, Time, Comparison and Verification respectively. Different colors mean different stages.

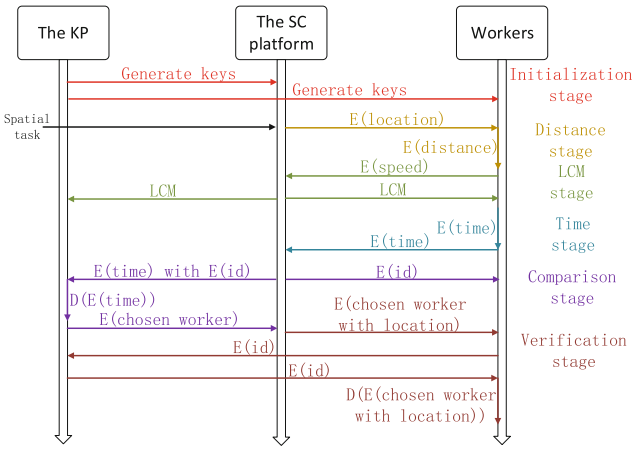

Fig. 2. Framework overview

4.2 Detailed Framework

Next, every stage in details is described in this subsection.

Initialization Stage. Firstly, the KP generates a pair of keys for Paillier. Then the KP keeps the encryption key public and the decryption key private respectively because the computations of the SC platform and workers are based on encrypted data while the KP has to decrypt data to find the chosen worker. Besides, the KP generates a cyclic group G for ElGamal based on which the KP and all workers generate their own pair of keys and keep them secret.

Distance Stage. Given a spatial task T, the SC platform encrypts task location $l_T(x_T, y_T)$ with the encryption key of Paillier by calculating $E_p(x_T^2+y_T^2)$, $E_p(x_T)$ and $E_p(y_T)$. Then these three ciphertexts are sent to all workers. Without holding the decryption key of Paillier, every worker w_i can calculate the encrypted square of the distance based on the Euclidean distance and the homomorphic properties shown in Eqs. 3 and 4 as follows:

$$E_p(d^2(l_T, l_i)) = E_p(x_T^2 + y_T^2)E_p(x_T)^{-2x_i}E_p(y_T)^{-2y_i}E_p(x_i^2 + y_i^2) \tag{6}$$

It should be noted that it also works when every worker encrypts location and the SC platform calculates $E_p(d^2(l_T, l_i))$. However, it will cost much more computing resources for every worker can calculate in parallel. That is to say, our proposed method is good for reducing the computation cost of the SC platform.

LCM Stage. At first, we explain why we need to get the LCM of all worker's speed. As defined in Definition 3, our framework prefers the worker who has the shortest travel time. To this end, we have to face division operation on ciphertexts which is still an open problem nowadays during the computation of travel time. Though we cannot solve the problem of division operation, a transformation can be employed to avoid the division operation based on the following lemma:

Lemma 1. *Let $W = \{w_1, \cdots, w_n\}$ be a set of n workers, $D = \{d_1, \cdots, d_n\}$ be the distance between task location and the worker w_i, S_{lcm} be the LCM of every worker's speed s_i and $s'_i = S_{lcm}/s_i$ where $1 \leq i \leq n$. So for any two different workers $w_i, w_j \in W$, if $d_i s'_i < d_j s'_j$ holds then we must infer $d_i/s_i < d_j/s_j$.*

Proof. $d_i s'_i < d_j s'_j \iff d_i s'_i/S_{lcm} < d_j s'_j/S_{lcm} \iff d_i/s_i < d_j/s_j$.

Deforming the formula of travel time can help us avoid the division operation over ciphertexts, which is the reason why we need to get the LCM. Note that the product of all speeds is not suitable here for it may cause the overflow of the multiplication of all speeds [9]. The process of calculating the LCM by AP encryption strategy in a safe and secret way will be introduced in the next subsection. In the end, the SC platform will inform the KP and all workers of the LCM.

Time Stage. Upon receiving the LCM S_{lcm}, every worker w_i can calculate an equivalent encrypted travel time t'_i to replace real encrypted travel time t_i based

on the Lemma 1 where $t_i' = d(l_i, l_T)s_i'$ and $t_i = d(l_i, l_T)/s_i$. For no worker holds the decryption key of Paillier, homomorphic properties of Paillier are used again as follows:

$$E_p(t_i'^2) = E_p((d(l_t, l_i)s_i')^2) = E_p((d(l_t, l_i)S_{lcm}/s_i)^2) = E_p(d^2(l_T, l_i))^{(S_{lcm}/s_i)^2} \tag{7}$$

where $E_p\left(d^2\left(l_t, l_i\right)\right)$ is calculated by Eq. 6 and s_i is the speed of worker w_i. Then the worker sends the encrypted equivalent travel time with his own ID to the SC platform for comparison in the form of $(i, E_p(t_i'^2))$.

Comparison Stage. When receiving the list of $(i, E_p(t_i'^2))$, the SC platform adopts a PRG G_k to encrypt the ID of workers as $(G_k(i), E_p(t'^2_{G_k(i)}))$ for the protection of workers especially the chosen worker. Then the SC platform sends the list $(G_k(i), E_p(t'^2_{G_k(i)}))$ to the KP and sends every $G_k(i)$ to the corresponding worker w_i. With the decryption key of Paillier, the KP can decrypt $E_p(t'^2_{G_k(i)})$ to obtain the $t'^2_{G_k(i)}$ and the real travel time $t_{G_k(i)}$ can be computed by $\sqrt{\frac{tt'^2_{G_k(i)}}{S^2_{lcm}}}$ where S_{lcm} is achieved in the LCM. And then the KP can easily find the chosen worker who has the shortest $t_{G_k(i)}$. Then, the ID of the chosen worker $G_k(i^*)$ is encrypted by ElGamal, whose output is $E_e^{KP}(G_k(i^*))$. At last, the KP sends $E_e^{KP}(G_k(i^*))$ to the SC platform. This encrypting operation is essential because the SC platform can infer that who is the chosen worker from $G_k(i^*)$. However, when AR is not always 100%, we will return a set of chosen workers instead of a chosen worker.

Verification Stage. To ensure only the chosen worker can learn the true task location, the SC platform hides the true task location by encrypting $E_e^{KP}(G_k(i^*))$ and l_T as follows:

$$E\left(l_T\right) = h\left(E_e^{KP}\left(G_k\left(i^*\right)\right)\right) \oplus l_T \tag{8}$$

where function h is a length-match hash function which is used shorten a long bit-string and it is proved to be semantically secure. We perform exclusive-OR on the l_T and the output of function h because an important property of exclusive-OR is $a \oplus b \oplus a = b$. Based on this property, only the chosen worker w_i^* can infer the true task location by $l_s = E(l_s) \oplus h(E_e^{KP}(G_k(i^*)))$. The detailed procedure is as follows:

With their own ElGamal, every worker encrypts their own encrypted ID $G_k(i)$ received in the comparison stage as $E_e^{w_i}(G_k(i))$ and sends it to the KP. For all ElGamals are based on the same cyclic group G, commutative-like encryption can be implemented by $E_e^{KP}(E_e^{w_i}(G_k(i))) = E_e^{w_i}(E_e^{KP}(G_k(i)))$ with the same random number for the consistence of E_e^{KP} and the result is sent back to workers. Every worker w_i can decrypt it by the decryption key of his own ElGamal and get $E_e^{KP}(G_k(i))$. It is obvious that only the chosen worker can infer $E_e^{KP}(G_k(i^*))$ and thus infer the true task location.

Algorithm 1. Calculating LCM

Input: the maximal speed S_{max}, the speed s_i of every worker $w_i (1 \leq i \leq n)$
Output: the LCM of all speeds S_{lcm}

1: The SC platform and all workers perform the same exclusion algorithm on S_{max} to get a same list L of 2-tuples $<p, c_p>$ where p is a prime meeting $p \leq S_{max}$ and c_p is the maximal times of p meeting $p^{c_p} \leq S_{max}$.
2: Every worker w_i computes his own factorization F_i of s_i by Pollard's rho algorithm.
3: AP performs $\sum_{p \in P} p*(c_p+1)$ key generations and assigns these secrets respectively
4: **for** each prime p in L **do**
5: **for** number $k(0 \leq k \leq c_p)$ **do**
6: Every worker w_i generates his own flag data $f[k]$, encrypts it by
7: the assigned AP secrets and sends it to the SC platfrom.
8: $S_{lcm} = 1$
9: **for** each prime p in L **do**
10: **for** number $k(c_p \geq k \geq 0)$ **do**
11: The SC platform decrypts the sum of all $f[k]$, denoted as H.
12: **if** $H > 0$ **then**
13: $S_{lcm} = S_{lcm} * p^H$
14: break
15: **return** S_{lcm}

4.3 Calculating LCM

To compute the LCM securely, we adopt an aggregation protocol denoted as AP [25] which can calculate the sum of multiple messages in a privacy-preserving manner. It works as follows:

Key Generation: Let S be a set of nc random numbers where n is the number of workers and c is a random number. Then, divide S into n random disjoint subsets S_i with c numbers and define $M = 2^{\lceil \log_2 n\Delta \rceil}$ where Δ is maximum value of workers's data. At last, send k_i to w_i and the sum k_0 to the SC platform where $k_i = (\sum_{s' \in S_i} s') \mod M$ and $k_0 = (\sum_{s' \in S} s') \mod M$.

Encryption E_a: For each worker w_i, he encrypt data m_i by computing:

$$c_i = (k_i + m_i) \mod M \tag{9}$$

Encryption D_a: The SC platform can decrypt the sum by computing:

$$S(\sum_{i=1}^{n} m_i) = (\sum_{i=1}^{n} c_i - k_0) \mod M \tag{10}$$

Based on a credible assumption that the maximal worker's speed is limited and known to all, we explain the Algorithm 1 as follow: In line 1 and 2, exclusion algorithm is performed to get the list L of 2-tuples $<p, c_p>$ whose complexity is $O(n \log(\log n))$. For example, our maximal speed is 10. Then 3 is one prime where $3 < 10$, and its maximal times is 2 for $3^2 \leq 10$. So the tuple $<3, 2>$ will

be inserted into the list. Besides, every worker calculates the factorization F_i of his own speed s_i by Pollard's rho algorithm whose complexity is $O(n^{\frac{1}{4}})$. For example, the factorization F of a worker($s_i = 6$) is $F = 2 * 3$ for $6 = 2 * 3$. Based on the list L, the AP generates $\sum_{p \in L} p * (c_p + 1)$ different keys for same key may disclose workers' speed in line 3. In line 4 to 7, each worker w_i generates his flag data $f[k](k \in [0, c_p])$ as follows:

$$f[k] = \begin{cases} 1, & AT[p] = k \\ 0, & otherwise \end{cases} \tag{11}$$

where $AT[p]$ is the appearance times of p in the corresponding F_i. Then, encrypts and sends flag data. In the above examples, when $p = 3$, this worker ($s_i = 6$) generates these flag data $f[0] = 0, f[1] = 1, f[2] = 0$. In line 9 to 14, the LCM is computed by $S_{lcm} = \prod_{p \in L} p^H$. For example, the factorization of another worker($s_i = 9$) is $3 * 3$. If $p = 3$, this worker generates flag data $f[0] = 0, f[1] = 0, f[2] = 1$. So the maximal times of 3 is 2 for the decrypted sum of $f[2]$ meets the condition in line 12. Meanwhile, the maximal times of $2, 5, 7$ are $1, 0, 0$ respectively. So $S_{lcm} = 2^1 * 3^2 * 5^0 * 7^0 = 18$ will be returned.

5 Security and Complexity Analysis

Denoting the LCM stage as $E_a(s_i)$ and $D_a(S_{lcm})$, we will prove the security and complexity of our framework next.

5.1 Security Analysis

Theorem 1. *Our framework is allowed to be privacy-preserving with* $K_0 = S_{lcm}, K_{-1} = \{S_{lcm}, t_{G_k(i)}\}$ *and* $K_i = S_{lcm}(1 \leq i \leq n)$ *extra knowledge.*

Proof. We firstly consider the SC platform w_0 with $K_0 = S_{lcm}$. Then the view is $V_0 = \{E_e^{KP}(G_k(i^*)), S_{lcm}, E_a(s_j), E_p(t_j'^2)\}(1 \leq j \leq n)$. There is a probabilistic polynomial-time simulator P_0 that generates $V_0' = \{E_e^{KP}(x_1), S_{lcm}, E_a(y_i), E_p(z_i)\}$ where x_1 is random number from a cyclic group G, $y_i(1 \leq i \leq n)$ are random numbers distributed in $\mathbb{Z}$ and $z_i(1 \leq i \leq n)$ are random numbers uniformly distributed in $\mathbb{Z}_N$. As Paillier, ElGamal and AP are all secure, it is clear that $V_0 \equiv V_0'$.

Next we analyze every worker w_i with $K_i = S_{lcm}$. There is a probabilistic polynomial-time simulator P_i to simulate worker w_i's view. However, There are two kinds of workers to be analyzed. The difference between them is that only the chosen worker can infer the chosen ID is his ID. For the chosen worker w_i^*, his view is $V_{i^*} = \{G_k(i), i*, S_{lcm}, E_p(x_T^2 + y_T^2), E_p(x_T), E_p(y_T)\}$. So simulator P_{i^*} generates $V_{i^*}' = \{g, i*, S_{lcm}, E_p(x_1), E_p(x_2), E_p(x_3)\}$ where $x_i(i = 1, 2, 3)$ are random numbers uniformly distributed in $\mathbb{Z}_N$ and g is a random element uniformly distributed over $\{0, 1\}^\lambda$. For others, the view for them

is $V_i = \{G_k(i), E_{re}(E_e^{w_i}(G_k(i^*))), S_{lcm}, E_p(x^2+y^2), E_p(x), E_p(y)\}$ and simulator P_i generates $V_i' = \{g, E_{re}(E_e^{w_i}(y)), S_{lcm}, E_p(x_1), E_p(x_2), E_p(x_3)\}$ where x_i and g are the same as V_{i^*}' and y is a random number from G. Based on the semantic security of Paillier, ElGamal and PRG, we can easily verify that $V_i \equiv V_i'(1 \leq i \leq n)$.

Finally, we analyze the KP w_{-1} with $K_{-1} = \{S_{lcm}, t_{G_k(i)}\}(1 \leq i \leq n)$. The view of the KP is $V_{-1} = \{S_{lcm}, t_{G_k(i)}, E_e^{w_i}(G_k(i))\}(1 \leq i \leq n)$. There is a probabilistic polynomial-time simulator P_{-1} that generates $V_{-1}' = \{S_{lcm}, t_{x_i}, E_e^{w_i}(x_i)\}$ where $x_i(1 \leq i \leq n)$ are random numbers uniformly distributed in G. Due to the semantic security of ElGamal, $V_{-1} \equiv V_{-1}'$ is clearly true.

Based on the above proofs, our framework is secure with K disclosure where K has neglected effects on individual privacy.

5.2 Complexity Analysis

In our framework, every worker computes and communicates in parallel. To this end, we only need to consider one user. Ignoring some cheap operations, the computation and communication cost are summarized in Table 1 where $L_i(i = p, e)$ is the key size of encryption strategy,e is modular exponentiation and $+, -$ means sending and receiving. Note that ElGamal encryption and communicative-like encryption is two and three times longer than L_e. Due to the size of ciphertext by Paillier and ElGamal are larger than plaintext and the ciphertext by AP, we exclude the latter two from communication cost. In the situation when the AR is not always 100%, the KP needs $|W^*|E_e$ instead of $1E_e$ in computation cost and the communication cost changes from $|2L_e|$ to $2|W^*|L_e$ during the comparison stage.

Table 1. Computation and communication cost

	Computation cost			Communication cost		
	The SC platform	The KP	Workers	The SC platform	The KP	Workers
Distance	$3E_p$	0	$1E_p + 2e$	$+3L_p$	0	$-3L_p$
LCM	D_a	0	E_a	0	0	0
Time	0	0	3e	$-L_p$	0	$+L_p$
Comparison	$nPRG$	$nD_p + 1E_e$	0	$+nL_p - 2L_e$	$-nL_p + 2L_e$	0
Verification	0	nE_e	$E_e + D_e$	0	$-2nL_e + 3nL_e$	$+2L_e - 3L_e$

6 Experiment Study

In the first subsection, we introduce our experiment settings and evaluation criteria. Then we show and analyze the experiment results in the second subsection.

6.1 Experiment Settings

We conduct our experiments on an area in Pennsylvania of Gowalla dataset with latitude from 39.804250 to 41.787732 and longitude from −80.418515 to −75.189944 with 3036 workers.

Three criteria are introduced to evaluate our proposed framework, namely computing time, travel distance, and worker number respectively. For computing time, we compare our framework with Liu et al.'s framework [9] for all of them are based on the public-key cryptosystems. In these two frameworks, it is meaningless to take the computing time of the SC platform and the KP into consideration because we pay more attention on the workers computing time in the task assignment and these two parties are the same in these two frameworks. For travel distance and worker number, we compare our framework with To et al.'s framework [7] for Liu et al.'s framework has the same values as ours in travel distance and worker number. Tables 2 and 3 summarize the parameters in these two comparisons.

Table 2. Computing time

Parameters	Default	Range	Description
W	200	100, 200, 300, 400, 500	The number of workers
S_{max}	10	5, 10, 15, 20, 15	The maximal speed

Table 3. Travel distance and worker number

Parameters	Default	Range	Description
AR_{max}	0.6	0.2, 0.4, 0.6, 0.8, 1.0	The maximal AR
α	0.9	0.8, 0.85, 0.9, 0.95, 0.99	The expected rate of a task
ϵ	0.6	0.2, 0.4, 0.6, 0.8, 1.0	The privacy budget of To et al.'s framework

6.2 Performance Analysis

Computing Time. In the computing time comparison, two key sizes (1024 and 2048) of Paillier and ElGamal are considered in our framework and Liu et al.'s framework.

Firstly, we study the effect of S_{max}. As described in Fig. 3, no matter what key size is adopted, our framework has much shorter average computing time than Liu et al.'s framework which means tasks can be assigned more quickly and thus improve the service quality of all platforms. Also, there is a fault of Liu et al.'s framework where S_{max} is 10 when key size is 1024 because when S_{max} is larger than 10, theirs framework based on the product of all speeds will face the overflow of product. Meanwhile, our framework can support these calculations

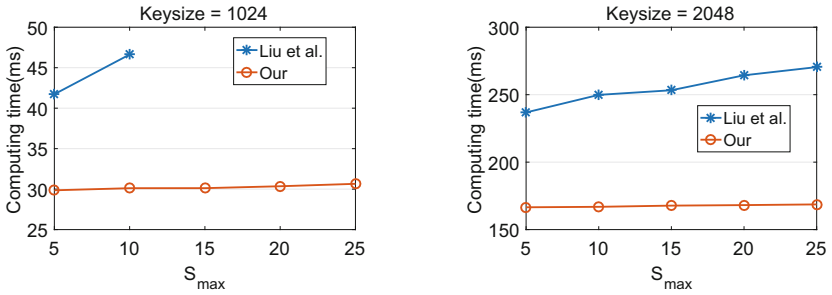

Fig. 3. Effect of S_{max}

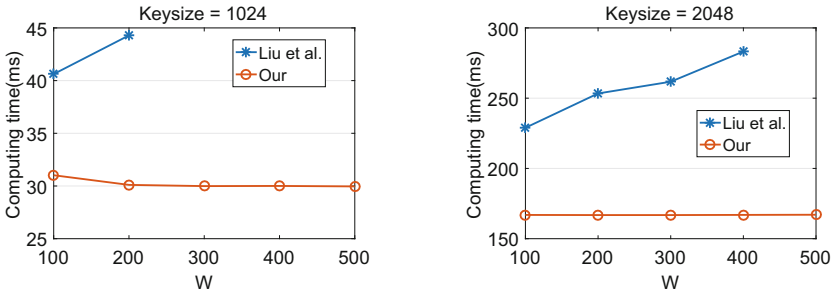

Fig. 4. Effect of W

for our framework is based on the LCM of all speeds. Note that there is still a fault where S_{max} is 100 in Liu et al.'s framework when key size is 2048 which is not shown in the Fig. 3. That is to say, the most important meaning for our framework is to break through the speed limitation of Liu et al.'s framework. Moreover, within our expectations, the computing time of Liu et al.'s framework increases as S_{max} grows while ours is a constant for the same reason as before.

Next, the effect of W is evaluated. Similar performance trend can be observed in Fig. 4 where the larger W is, the computing time grows. In addition, there are two obvious faults in Fig. 4 where W are 200 and 400 when key sizes are 1024 and 2048 respectively for the same reason as first part. Also, our framework has much shorter computing time than Liu et al.'s framework. Based on the LCM, our framework can be applied to more workers and a bigger speed.

Travel Time and Worker Number. In the travel time and worker number comparison, two functions are used to change the AR of every worker (Linear and Zipf). As To et al.'s framework does not consider the speed of workers, we set the speed of all workers is 1.

Firstly, we investigate the effect of AR_{max}. As depicted in Fig. 5, our framework has much shorter travel distance and smaller number of notified workers than To et al.'s framework because theirs is to choose some grid cells which

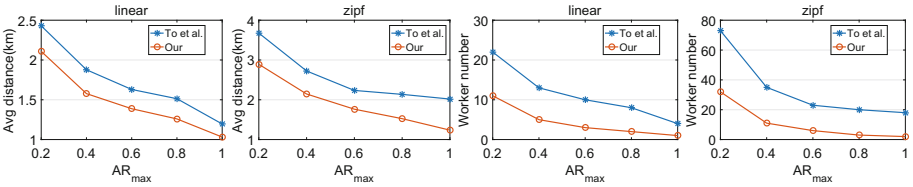

Fig. 5. Effect of AR_{max}

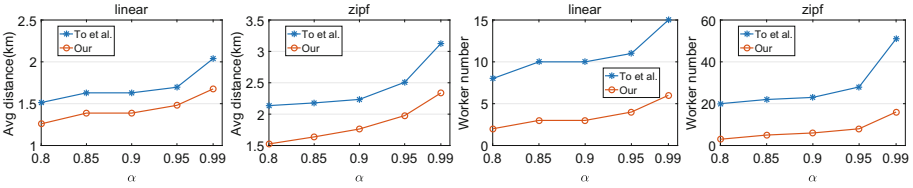

Fig. 6. Effect of α

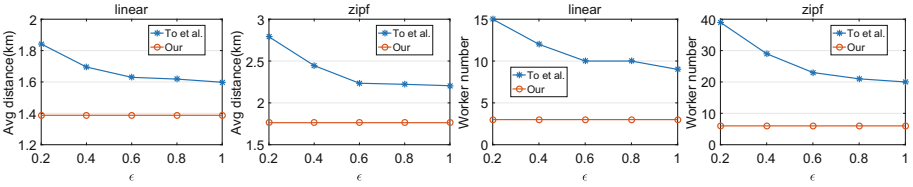

Fig. 7. Effect of ϵ

contains a number of workers. Some of them may be far away from task location. Yet, our framework is to visit the worker sorted by travel distance. In addition, the travel distance and worker number of our framework decrease when AR_{max} increases for a larger AR_{max} means workers are more willing to achieve this task.

Secondly, we study the effect of α. Figure 6 shows that our framework is much better than To et al.'s framework for the same reason as before. Also the travel distance and worker number of our framework grow with α increases for a larger α means a task has a higher expected rate to be accepted and thus more workers are required to accomplish the task.

At last, we assess the effect of ϵ. The higher ϵ is, the weaker privacy guarantee To et al.'s framework has. As expected, the change of ϵ only affects To et al.'s framework for ours is stable which is shown in Fig. 7. Also, with ϵ increases, the travel distance and worker number of their framework decreases by sacrificing of privacy. But ours still works better than theirs even in weakest privacy guarantee.

7 Conclusion

In this paper, we have identified a new task assignment strategy, travel-time-first, when allocating workers to tasks in spatial crowdsourcing. We have presented an

efficient privacy-preserving task assignment protocol for this new strategy. The proposed protocol scales well because the expensive secure division operation is replaced by the secure least common multiple (LCM) computation, for which we have designed an efficient algorithm based on data aggregation. We have theoretically proved that our approach is secure against semi-honest adversaries. We have conducted extensive experiments on real-world datasets. Experimental results have shown that our protocol is efficient and effective.

Acknowledgement. Research reported in this publication was partially supported Natural Science Foundation of China (Grant Nos. 61572336, 61632016, 61572335).

References

1. Chen, L., Shahabi, C.: Spatial crowdsourcing: challenges and opportunities. IEEE Data Eng. Bull. **39**(4), 14–25 (2016)
2. Kazemi, L., Shahabi, C.: GeoCrowd: enabling query answering with spatial crowdsourcing. In: SIGSPATIAL, pp. 189–198 (2012)
3. Deng, D., Shahabi, C., Demiryurek, U.: Maximizing the number of worker's self-selected tasks in spatial crowdsourcing. In: SIGSPATIAL, pp. 324–333 (2013)
4. Cheng, P., Lian, X., Chen, Z., Fu, R., Chen, L., Han, J., Zhao, J.: Reliable diversity-based spatial crowdsourcing by moving workers. PVLDB **8**(10), 1022–1033 (2015)
5. To, H., Ghinita, G., Fan, L., Shahabi, C.: Differentially private location protection for worker datasets in spatial crowdsourcing. TMC **16**(4), 934–949 (2017)
6. Liu, B., Chen, L., Zhu, X., Zhang, Y., Zhang, C., Qiu, W.: Protecting location privacy in spatial crowdsourcing using encrypted data. In: EDBT, pp. 478–481 (2017)
7. To, H., Ghinita, G., Shahabi, C.: A framework for protecting worker location privacy in spatial crowdsourcing. PVLDB **7**(10), 919–930 (2014)
8. Liu, A., Li, Z., Liu, G., Zheng, K., Zhang, M., Li, Q., Zhang, X.: Privacy-preserving task assignment in spatial crowdsourcing. J. Comput. Sci. Technol. **32**(5), 905–918 (2017)
9. Liu, A., Wang, W., Shang, S., Li, Q., Zhang, X.: Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. GeoInformatica **22**(2), 335–362 (2018)
10. Cheng, P., Lian, X., Chen, L., Han, J., Zhao, J.: Task assignment on multi-skill oriented spatial crowdsourcing. TKDE **28**(8), 2201–2215 (2016)
11. Zheng, L., Chen, L.: Maximizing acceptance in rejection-aware spatial crowdsourcing. TKDE **29**(9), 1943–1956 (2017)
12. Tong, Y., She, J., Ding, B., Wang, L., Chen, L.: Online mobile micro-task allocation in spatial crowdsourcing. In: ICDE, pp. 49–60 (2016)
13. Tong, Y., Wang, L., Zhou, Z., Ding, B., Chen, L., Ye, J., Xu, K.: Flexible online task assignment in real-time spatial data. PVLDB **10**(11), 1334–1345 (2017)
14. Gao, D., Tong, Y., She, J., Song, T., Chen, L., Xu, K.: Top-k team recommendation and its variants in spatial crowdsourcing. Data Sci. Eng. **2**(2), 136–150 (2017)
15. Paulet, R., Kaosar, M.G., Yi, X., Bertino, E.: Privacy-preserving and content-protecting location based queries. TKDE **26**(5), 1200–1210 (2014)
16. Liu, S., et al.: Efficient query processing with mutual privacy protection for location-based services. In: Navathe, S.B., Wu, W., Shekhar, S., Du, X., Wang, X.S., Xiong, H. (eds.) DASFAA 2016. LNCS, vol. 9643, pp. 299–313. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-32049-6_19

17. Yi, X., Paulet, R., Bertino, E., Varadharajan, V.: Practical k nearest neighbor queries with location privacy. In: ICDE, pp. 640–651 (2014)
18. Yi, X., Paulet, R., Bertino, E., Varadharajan, V.: Practical approximate k nearest neighbor queries with location and query privacy. TKDE **28**(6), 1546–1559 (2016)
19. Sun, Y., Liu, A., Li, Z., Liu, G., Zhao, L., Zheng, K.: Anonymity-based privacy-preserving task assignment in spatial crowdsourcing. In: Bouguettaya, A., et al. (eds.) WISE 2017. LNCS, vol. 10570, pp. 263–277. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-68786-5_21
20. Liu, A., Zheng, K., Li, L., Liu, G., Zhao, L., Zhou, X.: Efficient secure similarity computation on encrypted trajectory data. In: ICDE, pp. 66–77 (2015)
21. Goldreich, O.: The Foundations of Cryptography - Volume 2, Basic Applications. Cambridge University Press, Cambridge (2004)
22. Reddaway, S.: Pseudo-random number generators. US, pp. 57–67 (1974)
23. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48910-X_16
24. ElGamal, T.: A public key cryptosystem and a signature scheme based on discrete logarithms. IEEE Trans. Inf. Theory **31**(4), 469–472 (1985)
25. Li, Q., Cao, G., La Porta, T.F.: Efficient and privacy-aware data aggregation in mobile sensing. TDSC **11**(2), 115–129 (2014)