electronics

MDPI

Article

PKGS: A Privacy-Preserving Hitchhiking Task Assignment Scheme for Spatial Crowdsourcing

Peicong He, Yang Xin *, Bochuan Hou and Yixian Yang

School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing 100876, China; hepeicong@bupt.edu.cn (P.H.); houbochuan@bupt.edu.cn (B.H.); yxyang@bupt.edu.cn (Y.Y.)
* Correspondence: yangxin@bupt.edu.cn

Abstract: Privacy-preserving task assignment is vital to assign a task to appropriate workers and protect workers' privacy or task privacy for spatial crowdsourcing (SC). Existing solutions usually require each worker to travel to the task location on purpose to perform this task, which fails to consider that workers have specific trajectories and carry out the task on their way in a hitchhiking manner. To this end, this paper proposes a privacy-preserving hitchhiking task assignment scheme for SC, named PKGS. Specifically, we formulate the privacy-preserving hitchhiking task assignment as a decision problem of the relationship between dot and line under privacy protection. In particular, we present a privacy-preserving travel distance calculation protocol and a privacy-preserving comparison protocol through the Paillier cryptosystem and the SC framework. Results of theoretical analysis and experimental evaluation show that PKGS can not only protect the location privacy of both each worker and the task simultaneously but also assign the task to the worker holding a minimum travel distance. In contrast to prior solutions, PKGS outperforms in the computation of travel distance and task assignment.

Keywords: spatial crowdsourcing; task assignment; privacy-preserving; Paillier homomorphic encryption

check for updates

Citation: He, P.; Xin, Y.; Hou, B.; Yang, Y. PKGS: A Privacy-Preserving Hitchhiking Task Assignment Scheme for Spatial Crowdsourcing. *Electronics* **2023**, *12*, 3318. https://doi.org/10.3390/electronics12153318

Academic Editors: Chuan Zhang, Tong Wu and Weiting Zhang

Received: 30 June 2023
Revised: 29 July 2023
Accepted: 29 July 2023
Published: 2 August 2023

1. Introduction

Spatial crowdsourcing (SC) [1,2] has been widely applied in our daily life, such as crowdsourcing taxis (Uber, DiDi, Didachuxing, etc.) crowdsourcing logistics, and crowdsourced take-out delivery. In these applications, task assignment is an essential requirement for SC. As shown in Figure 1, the task assignment is to tackle how to assign a crowdsourcing task to appropriate workers. For example, an appropriate worker has a minimum travel distance. The task publishers are called task requesters (TRs), and the task completers are called task workers (TWs).

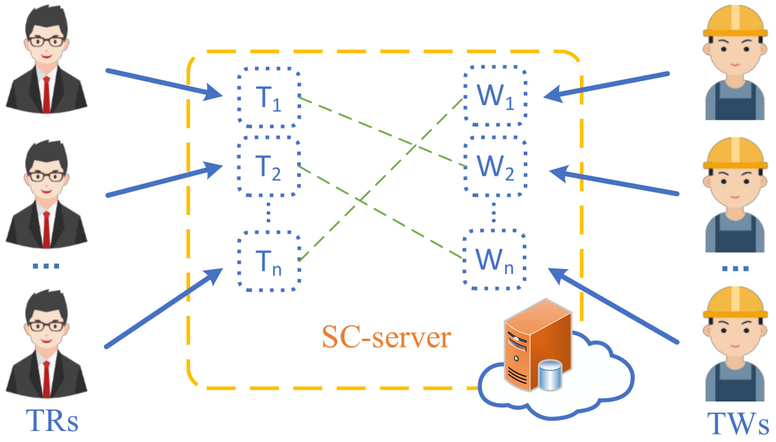

Figure 1. Working Models in SC.

Privacy concerns are widespread in various scenarios in cloud environments, such as neural network training and prediction [3] and support vector machine training in the cloud [4]. It is also a key barrier faced by task assignment of SC. Task assignment of SC always needs to utilize the location both of TWs and task to perform task assignment. However, in terms of TWs and TRs, location data are sensitive information, which involves location privacy [5]. Thus, TWs may refuse to take part in the crowdsourcing task due to privacy concerns. If all TWs do not participate in any crowdsourcing task, the SC paradigm fails to tackle any task by the capability of crowds. On the other hand, in terms of an SC platform, it suffers from high penalties due to disclosing collected TWs' locations and task locations. For example, DiDi is fined $1.28 billion, and Uber is fined $148 million.

Privacy-preserving task assignment simultaneously resolves the task assignment and privacy concerns of SC. Privacy-preserving task assignment is to assign the crowdsourcing task to TWs without compromising any TW or task privacy. Privacy-preserving task assignment has become a research hotpot, and a number of solutions have been proposed. In existing solutions, encryption-based and differential privacy (DP)-based are common methods to protect the privacy of TWs' locations or task locations. DP-based solutions have an advantage over efficiency but are weak in accuracy. In contrast, encryption-based solutions can perform accurate computations but face high computation burdens. Although there are abundant solutions to enable privacy-preserving task assignment, hardly any existing scheme supports the privacy-preserving hitchhiking paradigm, in which each worker has a specific trajectory and is allowed to perform a task on her way without scarifying location privacy of any worker and task. Privacy-preserving hitchhiking task assignment faces with following technical challenges.

The first challenge is how to compute the travel distance without knowing any TW location and task location. The travel distance is the distance between TW's location and a task location. The travel distance is usually measured by Euclidean distance or Manhattan distance. However, whether it is Euclidean distance or Manhattan distance, computing them is not straightforward when the worker's location and the task location are unknown. Technically, it requires computing the travel distance over encrypted or obscured locations. The second challenge is how to estimate reachability without compromising privacy. An SC task is spatiotemporal dependent, thus, and it requires not only computing the travel distance but also estimating whether a TW can reach the task location or not under the time constraint. Intuitively, if all TWs' locations and task locations as well as their temporary information are known, it is easy to estimate whether a TW can reach the location on schedule. However, to protect privacy, a privacy-preserving task assignment solution fails to learn any TW and task spatiotemporal sensitive information. The third challenge is how to search for an optimal worker under privacy protection. The task assignment is usually regarded as an optimization problem, such as searching for the minimum travel distance under multiple constraints. The challenge for privacy-preserving task assignment is to resolve the optimization problem under all data being encrypted or obscured.

To tackle the above challenges, in this paper, we propose a privacy-preserving hitchhiking task assignment solution for SC, called PKGS (PKGS comes from a Privacy-preserving hitchhiKing task assiGnment solution for spatial crowdSourcing). Specifically, we first formulate the privacy-preserving hitchhiking task assignment as a decision problem of the relationship between dot and line within privacy protection. We then adopt the Paillier cryptosystem that enables computation over encrypted data to encrypt the worker and task spatiotemporal information and protect their privacy. To enable the Paillier cryptosystem supporting operations of privacy-preserving task assignment, including computation of traveling distance, accessibility estimation, and optimization under constraints, we carefully design a privacy-preserving travel distance calculation protocol and a privacy-preserving comparison protocol. Our contributions are three-fold as follows.

(1) We formulate a privacy-preserving hitchhiking task assignment. To the best of our knowledge, this paper formulates a privacy-preserving hitchhiking task assignment as

a decision problem of the relationship between dot and line under privacy protection for the first time.

(2) We present two privacy-preserving computation protocols. To enable privacy protection and computation over encrypted data, we present a privacy-preserving travel distance calculation protocol to measure the Manhattan distance between a TW and a task. We also present a privacy-preserving comparison protocol to estimate the accessibility and search for the TW holding a minimum travel distance.

(3) We propose a privacy-preserving hitchhiking task assignment solution and evaluate its effectiveness and efficiency. Based on the proposed two privacy-preserving computation protocols, we construct a privacy-preserving hitchhiking task assignment solution, which can search for the optimal TW with a minimum travel distance within several seconds.

The rest of this paper is organized as follows. We introduce the related work in Section 2. In Section 3, we describe the system model, security model, Paillier cryptosystem, and problem formulation. In Section 4, we introduced Scheme PKGS in detail, including how to select potential TWs and how to select optimal TW. In Section 5, we analyze the security of PKGS. Section 6 reports and evaluates the experimental results. Finally, we draw a conclusion in Section 7.

2. Related Work

In recent research, task assignment protecting user privacy can be roughly divided into two types. One is to protect user privacy by processing user-uploaded information or query feedback information. The other is to achieve the purpose by encrypting this information. The first type mainly builds a cloaking region by processing information data. Intuitively, anonymous and perturbation techniques can do this. Based on k-anonymity [6] and l-diversity [7], a lot of anonymous task assignments have been proposed [8–11]. In [8], Kazemi et al. proposed the PiRi scheme, in which multiple users co-create anonymous zones and send requests to the SC server through anonymous zones, thus achieving privacy protection. Users in the anonymous zone share the results of requests to the SC server. Vu et al. [9] partition at least k adjacent users into a group with locality-sensitive hashing (LSH) that maintains both localization and k-anonymity. They design an algorithm for kNN querying based on this partition. Pournajaf et al. [10] study a spatial task assignment method when workers utilize spatial cloaking to obfuscate their locations, which assigns tasks by managing location regions with resource constraints. Based on Pournajaf's work, Hu et al. [11] consider the extent to which a worker is willing to move so that the worker's position in a task assignment is an area of activity. Like the anonymous technique, perturbation techniques can protect users' privacy by modifying values, such as differential privacy or perturbation of geographical position [12–15]. To et al. [12] utilize a trusted third party to preprocess data and then utilize differential privacy protection to protect the worker's location. After that, they tackle the moving TWs challenge by continuously sending privacy-preserving location information and reducing the noise generated in this way by Kalman filter-based post-processing technique [15]. Based on [12], Gong et al. [13] introduced reputation parameters for quality assessment. In a similar framework, Zhang et al. [14] use contour plots to characterize the distribution of workers and thus have less noise generation than others.

Cryptographic systems to encrypt data are the other way to protect the user's privacy [16–21]. Without further consideration of efficiency, the method is strongly theoretically supported for security. These schemes for encrypting data require the construction of protocols that conform to that encryption method to accomplish the assignment of tasks. Zhao et al. [16] propose the iTAM scheme that achieves multiple constrained exact matching of TR and TW by Paillier. However, the full ciphertext comparison leads to the need to compare all the candidates. Although Shu et al. [17] achieve bidirectional privacy-preserving task assignment through proxy re-encryption, their proposed solution requires additional servers such as proxy servers or fog nodes. Then they proposed pMatch [18], that is, a mechanism based on Shamir secret sharing that solves the problem of needing a proxy in privacy-preserving task assignment. Wang et al. [19] provided a personalized privacy-preserving task assignment mechanism PWSM that uses fuzzy location information to assign tasks to the maximum probability worker closest to the task. In [20], Ni et al. proposed the SPOON scheme, which recruits workers based on user location and protects privacy through proxy re-encryption and BBS+ signature. Liu et al. [21] proposed a range search retrieval method that can perform ciphertext retrieval in a small area, thus improving efficiency, but did not propose a corresponding range determination and scaling method. The encryption scheme has the advantage of accurate retrieval, but its retrieval matching process is time-consuming. Workers need to wait for a long matching time after completing a task before they can be assigned the next task. Therefore, we propose a hitchhiking scheme that enables workers to complete tasks within an existing trip, thus increasing their motivation.

3. Problem Formulation and Preliminaries

3.1. System Model

As depicted in Figure 2, our system model consists of Task Requesters (TRs), Task Workers (TWs), Key Generation Center (KGC), and CS server. The CS server is an online platform provided by a service provider (SP) where it matches the TRs and TWs according to their requirement. In other words, it receives task requests from TRs as well as TWs and matches them online. KGC is a trusted organization for secret key generation and distribution. The TRs are the users of SP who want to find a suitable worker to help them complete a task within space-time by the platform. Similarly, the TWs are the other kind of users of SP who want to find a job satisfying their requirements. In our system, the whole task assignment process is divided into four main steps as follows.

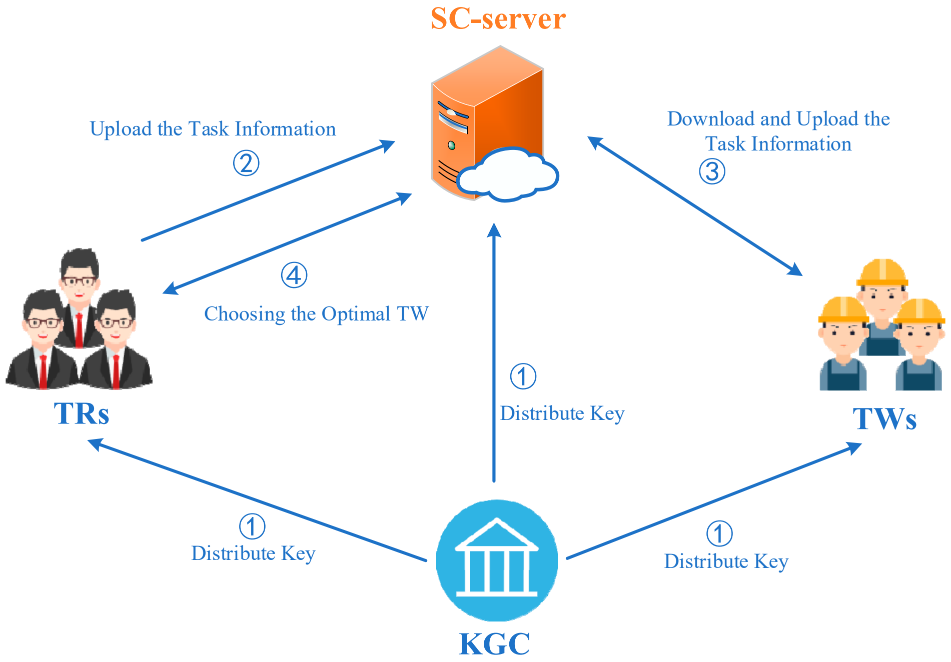

Figure 2. System Model of PKGS.

In ①, KGC generates public and private keys and manages them for TRs, TWs, and the SC server.

In ②, TRs use public keys to encrypt key task information and upload task descriptions and encrypted task information to the SC server.

In ③, TWs download the information about the task they want to perform and upload their work information to the SC server with the same public key encryption.

In ④, the SC server cooperates with the TR issued the task to complete the selection of the optimal worker without revealing private information.

3.2. Security Model

In SC, due to the strongly spatiotemporal properties of tasks, the task request of both TRs and TWs includes much personal sensitive information. This information reveals the current approximate location and needs of a TR or TW to a certain extent, and it has also become the target that the adversary wants to obtain. We assume a strong adversary $\mathcal{A}$ in our model, who has the following capabilities.

(1) $\mathcal{A}$ can disguise as a legal TR or TW and communicate with the SC server to obtain information.
(2) $\mathcal{A}$ can eavesdrop on the communication channel between TRs or TWs and CS server to capture communication information.
(3) $\mathcal{A}$ can compromise the SC server to obtain some information of an unselected TW stored on it.

As Figure 2 shows, the potential threats mainly come from TRs, TWs, and the SC server. Here, we assume that the SC server is semi-honest. In other words, the SC server is honest-but-curious, which follows the protocol but wants to snoop on the user's privacy. This assumption is reasonable because a formal service provider needs to register when releasing its products, and it will provide services under the pre-agreed agreement. However, it also wants to optimize and promote its products by collecting user information, but the collected user information is harmful to users. So, we need to face security threats from these aspects of them, not just adversaries but curious users or the CS server. We should avoid the three of them casually viewing the plaintext of task request information of TRs and TWs. This can prevent an adversary from pretending to be a legal user to obtain the task information of TRs or TWs by his task requirement.

3.3. Paillier Cryptosystem

As a homomorphic public key cryptosystem, the Paillier cryptosystem [22] is widely used for secure multiparty computer protocols.

Key generation: p and q are large prime numbers independently, set $N = pq$ and $\lambda = lcm(p-1, q-1)$, and select random integer $g \in \mathbb{Z}^*_{N^2}$, and then compute $\mu = (L(g^\lambda \bmod N^2))^{-1} \bmod N$, where the function L is defined as $L(x) = \frac{x-1}{N}$. The public key is $pk = (g, N)$ and the private key is $sk = (\lambda, \mu)$.

Encryption: Given a message $m \in \mathbb{Z}_N$ to be encrypted with the public key pk, $\mathcal{C} = [\![m]\!]_{pk} = g^m \cdot r^N \bmod N^2$, where r is a selected random integer $r \in \mathbb{Z}^*_N$.

Decryption: Given the ciphertext $\mathcal{C}$ to be decrypted with the private key sk, $m = D_{sk}(\mathcal{C}) = L(\mathcal{C}^\lambda \bmod N^2) \cdot \mu \bmod N$.

Let $x, y \in \mathbb{Z}_N$, its homomorphic properties are described below.

(1) Additive homomorphism:

$$[\![x]\!] \cdot [\![y]\!] = [\![x+y]\!]. \tag{1}$$

(2) Scalar-multiplicative homomorphism:

$$[\![x]\!]^y = [\![x \cdot y]\!]. \tag{2}$$

3.4. Problem Formulation

In PKGS, we try to help TWs with an existing itinerary to match a suitable task. A TW has her/his departure time, departure location, and destination location. A TR publishes a task that requires someone to complete the specified task at the agreed place at the specified time. To transform the real problem into a mathematical problem, we formulate Task, TW, Travel Distance, and Task Assignment. For the convenience of description, we use loc instead of specific coordinates (x, y).

Definition 1 (Task). *A Task denoted by $T = \{loc_R, T_{Rstart}, T_{task}, TD\}$ requires a TW to travel to the physical location loc_R to perform the task. The task start time is T_{Rstart}, and the time required to complete the task is T_{task},TD is a task description without involving privacy of the task.*

Definition 2 (TW). *A TW denoted by $W = \{loc_S, loc_D, T_{Wstart}, T_{Wend}, v\}$ starts from point loc_S to point loc_D , her/his departure time is T_{Wstart} , the cutoff time to reach loc_D is T_{Wend} , and the travel speed is v.*

Definition 3 (Travel Distance). *Travel distance is Manhattan Distance (MD) denoted by $m(A, B)$ that is a new metric in which the distance between two points is the sum of the absolute differences of their cartesian coordinates and is also known as city block distance or taxicab geometry. Assuming there are two points $A(x_1, y_1)$ and $B(x_2, y_2)$, their Manhattan distance is calculated as follows.*

$$m(A, B) = |x_1 - x_2| + |y_1 - y_2| \tag{3}$$

Compared with Euclidean distance, Manhattan distance is more suitable for calculating the distance between two points in the city. As we know, it is not possible to get from point A to point B in a city directly through a building, which is why it is called city block distance or taxicab geometry.

Definition 4 (Task Assignment). *Task assignment is a set of conditions matching TR and TW. As Figure 3 shows, the TW plans to go from loc_S to loc_D . The task location is loc_R . Task assignment requires the following conditions to be met. For the TW, she/he needs to reach loc_D before T_{Wend} . For the TR, the task needs to start at moment T_{Rstart} . The time taken by the TW to travel from loc_S to loc_R and from loc_R to loc_D is T_1 and T_2, respectively. So, the task assignment needs to fulfill the following conditions.*

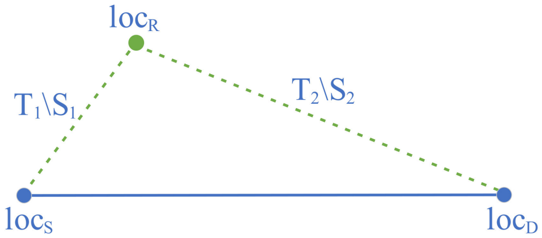

Figure 3. Task Assignment Constraint.

$$\begin{cases} T_{Wstart} + T_1 + T_{task} + T_2 \le T_{Wend} \\ T_{Wstart} + T_1 \le T_{Rstart} \end{cases} \tag{4}$$

To make the matching condition satisfy the homomorphic cryptosystem, we do a little trick with it. We transform a comparison of time into a comparison of distance. Assume there is a TR's task denoted by $T = \{loc_R, T_{Rstart}, T_{task}, TD\}$ and a TW denoted by $W = \{loc_S, loc_D, T_{Wstart}, T_{Wend}, v\}$. If the following constraints are met

$$\begin{cases} S_1 + S_2 \le S \\ S_1 \le S' \end{cases} \tag{5}$$

the TW can perform the TR's task. In this set of equations, $S_1 = m(loc_R, loc_S)$, $S_2 = m(loc_R, loc_D)$, $S = (T_{Wend} - T_{Wstart} - T_{task}) * v$, $S' = (T_{Wstart} - T_{Rstart}) * v$.

Remark 1. *The distance of the TW from the starting point to the task point is denoted as S_1. The distance from the task point to the end point of the TW is denoted as S_2. The farthest moving distance of the TW under the premise of completing the task is recorded as S. This is the product of the time* *the TW can spend moving and his or her speed. Before the task starts, the maximum distance that the TW can move is recorded as S'. Its rationale is like that of S. $S_1+S_2 \leq S$ means that the TW can go to the mission point to complete the mission without delaying his original journey. $S_1 \leq S'$ means that the TW can reach the task point before the task needs to be started.*

Task allocation needs to consider not only whether the task will be completed on time, but also who is the optimal TW. The set of candidate TWs is denoted by $\mathbb{W}= \{W_1, W_2, \ldots, W_{\mathrm{n}}\}$, and the set of S_1 of each TW is denoted by $\mathbb{W}_{\mathrm{S}_1}= \{W_1\mathrm{S}_1, W_2\mathrm{S}_1, \ldots, W_{\mathrm{n}}\mathrm{S}_1\}$. A TW is an optimal worker if he satisfies the following conditions.

$$WS_1 = min(\mathbb{W}_{\mathrm{S}_1}) \tag{6}$$

As we know, S_1 is the distance of the TW from the start point to the task point. Equation (6) means that WS_1 is the nearest worker to the task point. If the TW is closer, the probability is higher that the TW will arrive at the task point as early as possible to start working.

4. Design of PKGS

4.1. Overview of PKGS

We have a preliminary understanding of PKGS through the previous system model, and now we will introduce its scheme process in detail. In Figure 4, we illustrate the process of PKGS with an example where a TR matches TWs for clarity. This situation can be seen as a case where, among the many TRs tasks, this group of TWs chose this task at the same time.

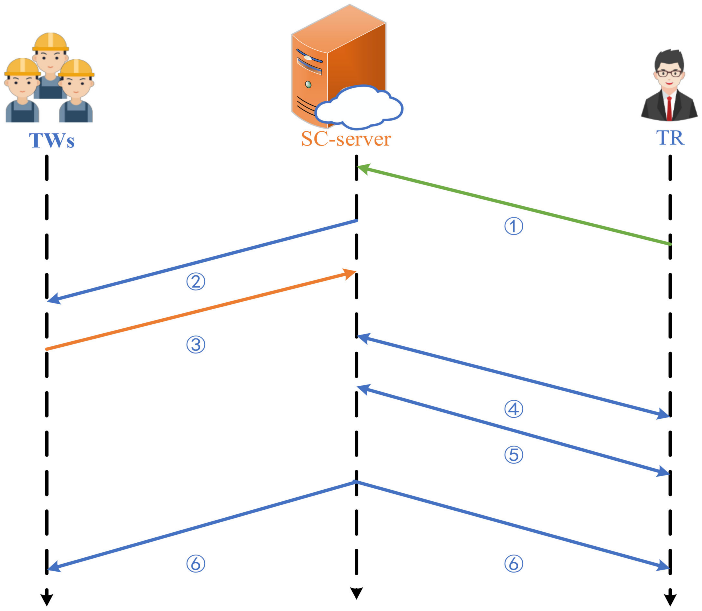

Figure 4. Scheme Process of PKGS.

In ①, a TR sends a request to the SC server to find a suitable TW to complete her/his CS task. As mentioned in Section 3.4, the TR's task is denoted by $T = \{loc_R, T_{Rstart}, T_{task}, TD\}$. To make it easier for TWs to choose, he uploads the encrypted location information loc_R and the time required to complete the task T_{task}, as well as the unencrypted task start time T_{Rstart} and task description TD to the SC server.

In ②, TWs select a task on the SC server and download the task information from it.

In ③, each TW adds their encrypted information to the downloaded encrypted task information and sends the encrypted information back to the SC server.

In ④, the SC server figures out candidate TWs who met the constraints with the help of TR.

In ⑤, the SC server finds the optimal TW from the candidate TWs with TR's assistance.

In ⑥, finally, the SC server establishes a connection between them.

Note that according to the descriptions in Sections 3.4 and 4.1, there are mainly two difficult problems to be solved. The travel distance is calculated from the encrypted task information and the encrypted travel distance is compared. Given this, we will focus on these two types of challenges and address two solutions detailed in the following two subsections.

4.2. Privacy-Preserving Comparison Protocol

Both absolute value symbol removal and constraint judgment require a comparison operation on the ciphertext. In this subsection, we propose a PC (privacy-preserving comparison) protocol to solve the comparison of encrypted information.

The SC server owns the computed ciphertext but cannot get the size relation because it does not have the private key. So, it needs the help of TR. However it cannot send the ciphertext directly to the TR because the TR has the private key and the location of the task. If the ciphertext is sent directly to the TR, then the TR can project the location of the TWs. So, as shown in Figure 5, the SC server performs a perturbation operation before transmission, and the TR decrypts the received message and returns the size relation to the SC server. The details are described below.

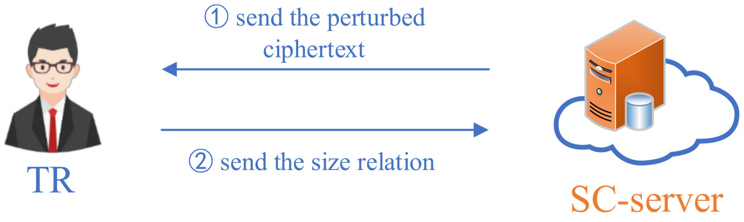

Figure 5. Process of PC.

Assume the length of a public key (N) is L and the encrypted numbers to be compared are recorded as $m = [\![A-B]\!]$. For security purposes, a typical public key length needs to be greater than or equal to 256 bits. Therefore, the number of bits in the values of A, B is much smaller than L. Let the perturbation value be $m_i{}' = [\![2^{L/2}]\!] \cdot [\![A-B]\!]^{x_i}$. x_i is a random integer for perturbing, ranging from -64 to 64 without 0. Since it is all positive integers in the Paillier cryptosystem, here, we add $2^{L/2}$ to the difference to avoid an overflow situation. The TR compares the sizes of $D_{sk}(m_i{}')$ and $2^{L/2}$. If $D_{sk}(m_i{}') > 2^{L/2}$, $R = 1$, if $D_{sk}(m_i{}') < 2^{L/2}$, $R = -1$, if $D_{sk}(m_i{}') = 2^{L/2}$, $R = 0$. Then, the SC server can determine the size relationship of A, B based on the values of R and x_i as shown in the following equation.

$$\begin{cases} x_i * R > 0 \ \Rightarrow \ A > B \\ x_i * R < 0 \ \Rightarrow \ A < B \\ x_i * R = 0 \ \Rightarrow \ A = B \end{cases} \tag{7}$$

Remark 2. *$[\![A-B]\!]^{x_i}$ is a number. When we add $2^{L/2}$ to it, if the result is greater than $2^{L/2}$, it means that $[\![A-B]\!]^{x_i}$ is a positive number, if it is less than $2^{L/2}$ then $[\![A-B]\!]^{x_i}$ is negative, and if it is equal to $2^{L/2}$, then $[\![A-B]\!]^{x_i}$ is equal to 0. There are two levels of perturbation of xi. On the one hand, there is its sign, which denotes $A-B$ if it is positive and $B-A$ if it is negative. on the other hand, we amplify the result of $A-B$ by a factor of x_i. The result of $|x_i|$ is the number of times the result of $A-B$. After the perturbation process, the TR can still determine $[\![2^{L/2}]\!]$ and $[\![A-B]\!]^{x_i}$ in relation to each other, but cannot infer one value of A, B from the other. With $R = 1$ and $x_i > 0$, we get $A-B > 0$; with $R = 1$ and $x_i < 0$, we get $A-B < 0$; with $R = -1$ and $x_i > 0$, we get $A-B < 0$; with $R = -1$ and $x_i < 0$, we get $A-B > 0$; with $R = 0$, we get $A = B$. So, we can get the above equation. With this, we can easily complete the comparison operation of encrypted data without revealing private information.*

4.3. Privacy-Preserving Travel Distance Calculation Protocol

In this subsection, we propose a privacy-preserving travel distance calculation protocol based on PC. Denote the PC operation as $P(A,B)$. A and B are the ciphertexts to be compared and the return value is the size relationship between A and B. Now, we design the protocol that implements steps ④ and ⑤. In ④, the goal is to find candidate TWs who meet the requirements. In ⑤, the goal is to find the optimal TW. To satisfy the judgment condition of the task assignment, as shown in Equation (5), we need to calculate S_1, S_2, S , S' . Assume that the coordinates of loc_R, loc_S, loc_D are (lat_R, lng_R), (lat_S, lng_S), (lat_D, lng_D), respectively. We can expand these equations as follows.

$$S_1 = m(loc_R, loc_S) = |lat_R - lat_S| + |lng_R - lng_S| \tag{8}$$

$$S_2 = m(loc_R, loc_D) = |lat_R - lat_D| + |lng_R - lng_D| \tag{9}$$

We convert the time point to the number of minutes elapsed so that the time can be converted into an integer for operation.

$$S = (T_{Wend} - T_{Wstart}) * v - T_{task} * v \tag{10}$$

$$S' = (T_{Wstart} - T_{Rstart}) * v \tag{11}$$

To protect privacy, we encrypt this information with TR's Paillier public key. We use $\llbracket x_1 \rrbracket$ to denote the Paillier encryption ciphertext of x_1. According to the Paillier cryptosystem, we can use the following equation with the encryption task information to calculate the desired distance cipher.

$$\begin{aligned} \llbracket S_1 \rrbracket &= \llbracket m(loc_R, loc_S) \rrbracket \\ &= |\llbracket lat_R - lat_S \rrbracket| + |\llbracket lng_R - lng_S \rrbracket| \\ &= |\llbracket lat_R \rrbracket \cdot \llbracket lat_S \rrbracket^{-1}| + |\llbracket lng_R \rrbracket \cdot \llbracket lng_S \rrbracket^{-1}| \end{aligned} \tag{12}$$

$$\begin{aligned} \llbracket S_2 \rrbracket &= \llbracket m(loc_R, loc_D) \rrbracket \\ &= |\llbracket lat_R - lat_D \rrbracket| + |\llbracket lng_R - lng_D \rrbracket| \\ &= |\llbracket lat_R \rrbracket \cdot \llbracket lat_D \rrbracket^{-1}| + |\llbracket lng_R \rrbracket \cdot \llbracket lng_D \rrbracket^{-1}| \end{aligned} \tag{13}$$

In ①, a TR sends the $\llbracket lat_R \rrbracket$, $\llbracket lng_R \rrbracket$, $\llbracket T_{\text{task}} \rrbracket$, T_{Rstart}, and TD to the SC server.

In ②, TWs select a task on the SC server based on the task description and task start time as well as download the task information from it.

In ③, each TW adds their encrypted information to the downloaded encrypted task information and sends the encrypted information back to the SC server, as $\llbracket lat_R \rrbracket \cdot \llbracket lat_S \rrbracket^{-1}$, $\llbracket lng_R \rrbracket \cdot \llbracket lng_S \rrbracket^{-1}$,$\llbracket lat_R \rrbracket \cdot \llbracket lat_D \rrbracket^{-1}$, $\llbracket lng_R \rrbracket \cdot \llbracket lng_D \rrbracket^{-1}$,$\llbracket (T_{Wend} - T_{Wstart}) * v \rrbracket$, $\llbracket T_{task} \rrbracket^{-\mathrm{v}}$, $\llbracket (T_{Wstart} - T_{Rstart}) * v \rrbracket$.

In ④, the SC server performs a perturbation operation on the values to be compared and the following operations with the assistance of TR: PC($(\llbracket lat_R \rrbracket \cdot \llbracket lat_S \rrbracket^{-1})'$,$\llbracket 2^{L/2} \rrbracket$), PC($(\llbracket lng_R \rrbracket \cdot \llbracket lng_S \rrbracket^{-1})'$,$\llbracket 2^{L/2} \rrbracket$), PC($(\llbracket lng_R \rrbracket \cdot \llbracket lng_D \rrbracket^{-1})'$,$\llbracket 2^{L/2} \rrbracket$), PC($(\llbracket lng_R \rrbracket \cdot \llbracket lng_D \rrbracket^{-1})'$, $\llbracket 2^{L/2} \rrbracket$). Based on the well-judged size relation, the server can compute $\llbracket S_1 \rrbracket$, $\llbracket S_2 \rrbracket$, and calculate $\llbracket S \rrbracket$, $\llbracket S' \rrbracket$ according to the following equation.

$$\begin{aligned} \llbracket S \rrbracket &= \llbracket (T_{Wend} - T_{Wstart}) * v - T_{task} * v \rrbracket \\ &= \llbracket (T_{Wend} - T_{Wstart}) * v \rrbracket \cdot \llbracket T_{task} \rrbracket^{-v} \end{aligned} \tag{14}$$

$$\llbracket S' \rrbracket = \llbracket (T_{Wstart} - T_{Rstart}) * v \rrbracket \tag{15}$$

The SC server figures out candidate TWs who meet the constraints based on the following PC operations: PC($(\llbracket S \rrbracket \cdot \llbracket S_1 \rrbracket^{-1} \cdot \llbracket S_2 \rrbracket^{-1})'$,$\llbracket 2^{L/2} \rrbracket$), PC($(\llbracket S' \rrbracket \cdot \llbracket S_1 \rrbracket^{-1})'$,$\llbracket 2^{L/2} \rrbracket$).

In ⑤, the SC server puts the TWs who passed the judgment in ④ into set $\mathbb{W}$ and puts their corresponding S_1 into set $\mathbb{W}_{S_1}$. PC operations are performed continuously in the set $\mathbb{W}_{S_1}$ until the optimal TW is selected and the algorithm is described as follows (Algorithm 1).

Algorithm 1: Find Optimal TW

```
Input: $\mathbb{W}_{S_1}$
Output: The number of $WS_{\min}$
Initialize min = 1;
for i = 1 to n − 1 do                                    //n = $\mathbb{W}_{S_1}$.length
PC($(([\![W_i S_1]\!] \cdot [\![W_{i+1} S_1]\!]^{-1})', [\![2^{L/2}]\!])$,
min = get the number of the small $WS_1$
Return min;
```

In ⑥, finally, the SC server establishes a connection between them according to the number.

5. Security Analysis

In this section, we analyze the security of PKGS in the process of task constraint determination and choosing optimal TW. In other words, PC and travel distance calculation protocol is secure as same as Paillier.

Theorem 1. *If the Paillier cryptosystem is secure, then the PC protocol is secure.*

Proof. For the SC server, the data are always in the form of ciphertext to be computed during the comparison process. Since it has no private key, it cannot decrypt the encrypted data. In other words, if the Paillier is secure, then the server cannot get the user's location information during the comparison process. For the TR, even though she/he has the private key to decrypt the ciphertext, the ciphertext has been perturbed. This also prevents the TR from obtaining the TWs' location information. Since the TR has the private key and the ciphertext is meaningless to him, we next analyze it in plaintext. In the process of perturbation, we randomly disturbed the position of the subtractor and the subtracted number and subjected the result of calculations to random magnification. This makes it impossible for the TR to distinguish the actual difference and to know the exact magnitude relationship, thus making it impossible to speculate on the location of the workers. □

Theorem 2. *If the Paillier cryptosystem is secure, then the travel distance calculation protocol does not reveal the location.*

Proof. In the previous theorem, it was shown that the TR cannot infer the location of TWs from the PC protocol. Next, we analyze the security for TWs and the SC server in the travel distance calculation protocol. For TWs, since they do not have the private key, they cannot decrypt the ciphertext, and no result is returned after they add the encrypted message. If the Paillier cryptosystem is secure, the TWs cannot guess the location of the task. For the SC server, it performs operations on the ciphertext, but the result of the computation is still the ciphertext. Moreover, it has no private key, so it cannot decrypt the ciphertext. The TR decrypts the ciphertext and returns the size relation instead of the specific result of the calculation. That is, if the Paillier cryptosystem is secure, it cannot get the location information of any party. □

6. Performance Analysis

6.1. Experiment Setting

In this section, we will test the performance of our proposed scheme through a series of experiments. As in previous work [16], we used the addition of simulated fields to the real dataset to complete the experiment. In this paper, we use the real Gowalla dataset and select 30 data entries in the range of latitude 37.75431 to 37.80062 and longitude −122.42691 to −122.39382. We choose 10 check-in data as the TWs and add departure time, task working time, and speed attributes at the original latitude of the data. The remaining 20 pieces of data are used as the TRs. Similarly, we add task start time and task need time to it. We assume 10 tasks have been assigned to TWs. In this way, each worker has a certain itinerary. TWs have three transportation tools, walking (90 m/min), bicycles (240 m/min), and cars (660 m/min).

We simulate the SC server with a computer with i7cpu and 16 GB of RAM. We simulate the users with a smartphone with a Kirin 990 core and 8 BG of RAM. We evaluate the performance of the scheme from the two aspects of the total time to complete all tasks and the total distance to complete all tasks.

6.2. Evaluation Results

We choose the related study [16] to complete the comparative experiment. In the experiment, there are 10 TWs with matched 10 tasks. So, each TW has a certain itinerary. We assign the remaining 10 tasks to them. In [16], TWs need to complete tasks and then select new tasks according to their task assignment scheme. In PKGS, we directly match the remaining 10 tasks to the 10 TWs with our task assignment scheme. We reselect 30 recordings and repeat the experiment 10 times. The total task completion time includes the time it takes to get to the task location and the time it takes to complete the task. The total distance traveled to complete all tasks includes the total distance the worker must travel. As shown in Figures 6 and 7, the results of the experiments show that the PKGS has less mission completion time and mission total distance.

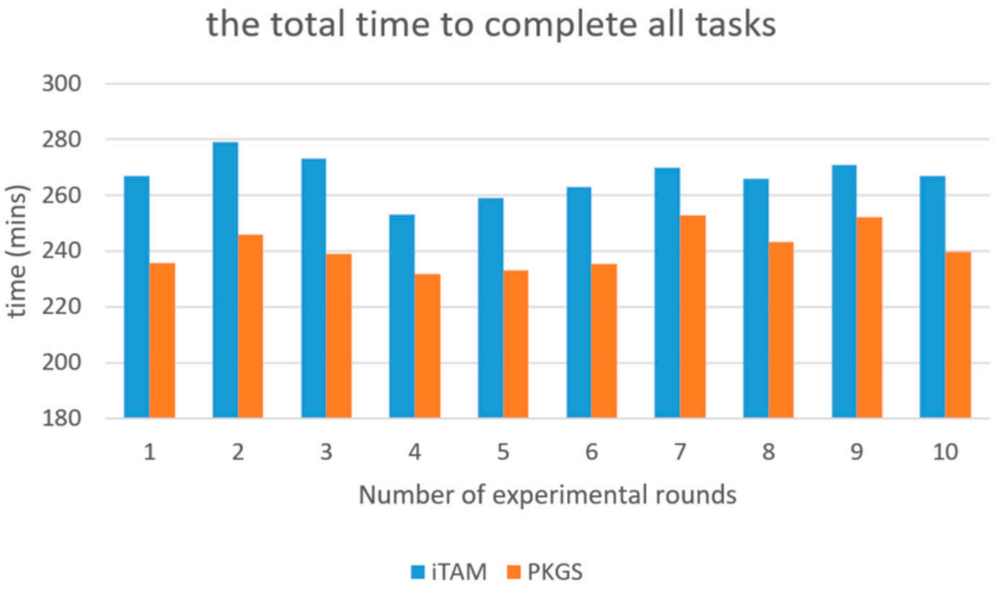

Figure 6. Time of All Tasks Completed.

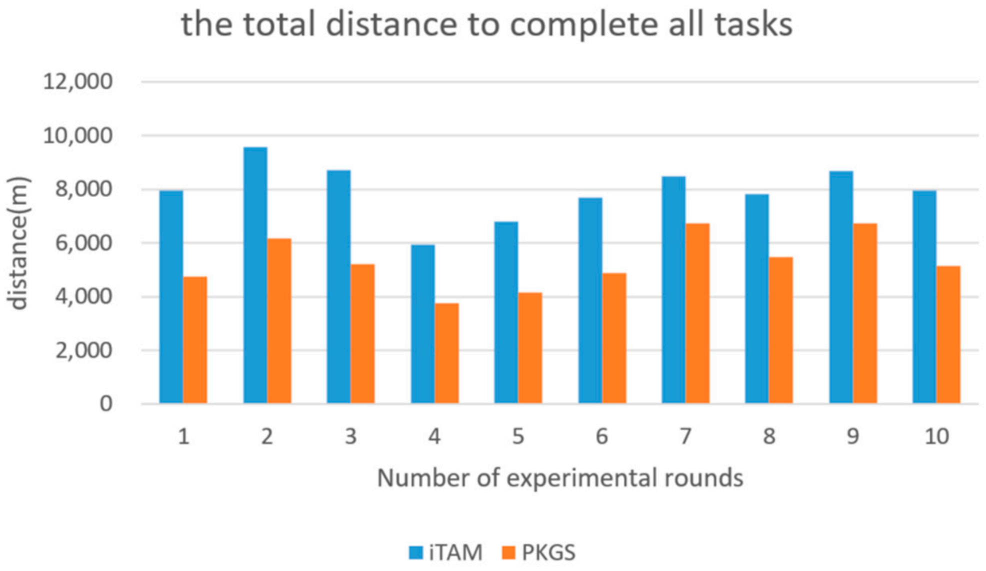

Figure 7. Distance of All Tasks Completed.

7. Conclusions

In this paper, we propose a privacy-preserving hitchhiking task assignment solution for SC, called PKGS, that can help TWs with existing travel trajectories to match with the appropriate TR. Since it can allocate tasks based on the TW's original trajectory under the premise of protecting user privacy. This greatly increases the speed of task completion and reduces the extra distance workers need to travel to complete additional tasks. In future work, we will consider the optimal solution problem of how to package and assign multiple tasks to a single TW.

Author Contributions: Conceptualization, P.H.; methodology, P.H.; software, P.H. and B.H.; validation, P.H.; investigation, P.H. and B.H.; resources, Y.X. and Y.Y.; writing—original draft preparation, P.H.; writing—review and editing, P.H.; supervision, Y.X. and Y.Y.; All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Key R&D Program of China (No. 2020YFB1805403) and the Foundation of Guizhou Provincial Key Laboratory of Public Big Data (No. 2017BDKFJJ015, No. 2018BDKFJJ008, No. 2018BDKFJJ020, No. 2018BDKFJJ021).

Data Availability Statement: In this paper, we use the Gowalla dataset, the URL of the dataset is https://snap.stanford.edu/data/loc-gowalla.html (accessed on 16 November 2021).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Feng, W.; Yan, Z.; Zhang, H.; Zeng, K.; Xiao, Y.; Hou, T. A survey on security, privacy, and trust in mobile crowdsourcing. *IEEE Internet Things J.* **2017**, *5*, 2971–2992. [CrossRef]
2. Tong, Y.; Zhou, Z.; Zeng, Y.; Chen, L.; Shahabi, C. Spatial crowdsourcing: A survey. *VLDB J.* **2020**, *29*, 217–250. [CrossRef]
3. Zhang, C.; Hu, C.; Wu, T.; Zhu, L.; Liu, X. Achieving efficient and privacy-preserving neural network training and prediction in cloud environments. *IEEE Trans. Dependable Secur. Comput.* **2022**. Available online: https://ieeexplore.ieee.org/abstract/document/9899726 (accessed on 28 July 2023). [CrossRef]
4. Hu, C.; Zhang, C.; Lei, D.; Wu, T.; Liu, X.; Zhu, L. Achieving Privacy-Preserving and Verifiable Support Vector Machine Training in the Cloud. *IEEE Trans. Inf. Forensics Secur.* **2023**, *18*, 3476–3491. [CrossRef]
5. Zhang, C.; Zhu, L.; Xu, C.; Ni, J.; Huang, C.; Shen, X. Location privacy-preserving task recommendation with geometric range query in mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2021**, *21*, 4410–4425. [CrossRef]
6. Sweeney, L. k-anonymity: A model for protecting privacy. *Int. J. Uncertain. Fuzziness Knowl.-Based Syst.* **2002**, *10*, 557–570. [CrossRef]
7. Machanavajjhala, A.; Kifer, D.; Gehrke, J.; Venkitasubramaniam, M. L-diversity: Privacy beyond k-anonymity. *ACM Trans. Knowl. Discov. Data (TKDD)* **2007**, *1*, 3-es. [CrossRef]

8. Kazemi, L.; Shahabi, C. A privacy-aware framework for participatory sensing. *ACM Sigkdd Explor. Newsl.* **2011**, *13*, 43–51. [CrossRef]
9. Vu, K.; Zheng, R.; Gao, J. Efficient algorithms for k-anonymous location privacy in participatory sensing. In Proceedings of the 2012 Proceedings IEEE INFOCOM, Orlando, FL, USA, 25–30 March 2012; IEEE: Piscataway, NJ, USA, 2012; pp. 2399–2407.
10. Pournajaf, L.; Xiong, L.; Sunderam, V.; Goryczka, S. Spatial task assignment for crowd sensing with cloaked locations. In Proceedings of the 2014 IEEE 15th International Conference on Mobile Data Management, Brisbane, Australia, 14–18 July 2014; IEEE: Piscataway, NJ, USA, 2014; Volume 1, pp. 73–82.
11. Hu, J.; Huang, L.; Li, L.; Qi, M.; Yang, W. Protecting location privacy in spatial crowdsourcing. In Proceedings of the Asia-Pacific Web Conference, Guangzhou, China, 18–20 September 2015; Springer: Cham, Swizerland, 2015; pp. 113–124.
12. To, H.; Ghinita, G.; Shahabi, C. A framework for protecting worker location privacy in spatial crowdsourcing. *Proc. VLDB Endow.* **2014**, *7*, 919–930. [CrossRef]
13. Gong, Y.; Zhang, C.; Fang, Y.; Sun, J. Protecting location privacy for task allocation in ad hoc mobile cloud computing. *IEEE Trans. Emerg. Top. Comput.* **2015**, *6*, 110–121. [CrossRef]
14. Zhang, L.; Lu, X.; Xiong, P.; Zhu, T. A differentially private method for reward-based spatial crowdsourcing. In *International Conference on Applications and Techniques in Information Security*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 153–164.
15. To, H.; Ghinita, G.; Fan, L.; Shahabi, C. Differentially private location protection for worker datasets in spatial crowdsourcing. *IEEE Trans. Mob. Comput.* **2016**, *16*, 934–949. [CrossRef]
16. Zhao, B.; Tang, S.; Liu, X.; Zhang, X.; Chen, W.-N. iTAM: Bilateral privacy-preserving task assignment for mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2020**, *20*, 3351–3366. [CrossRef]
17. Shu, J.; Jia, X.; Yang, K.; Wang, H. Privacy-preserving task recommendation services for crowdsourcing. *IEEE Trans. Serv. Comput.* **2018**, *14*, 235–247. [CrossRef]
18. Shu, J.; Yang, K.; Jia, X.; Liu, X.; Wang, C.; Deng, R.H. Proxy-free privacy-preserving task matching with efficient revocation in crowdsourcing. *IEEE Trans. Dependable Secur. Comput.* **2018**, *18*, 117–130. [CrossRef]
19. Wang, Z.; Hu, J.; Lv, R.; Wei, J.; Wang, Q.; Yang, D.; Qi, H. Personalized privacy-preserving task allocation for mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2018**, *18*, 1330–1341. [CrossRef]
20. Ni, J.; Zhang, K.; Xia, Q.; Lin, X.; Shen, X.S. Enabling strong privacy preservation and accurate task allocation for mobile crowdsensing. *IEEE Trans. Mob. Comput.* **2019**, *19*, 1317–1331. [CrossRef]
21. Liu, A.; Li, Z.X.; Liu, G.F.; Zheng, K.; Zhang, M.; Li, Q.; Zhang, X. Privacy-Preserving Task Assignment in Spatial Crowdsourcing. *J. Comput. Sci. Technol.* **2017**, *32*, 905–918. [CrossRef]
22. Paillier, P. Public-key cryptosystems based on composite degree residuosity classes. In Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques, Prague, Czech Republic, 2–6 May 1999; Springer: Berlin/Heidelberg, Germany, 1999; pp. 223–238.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.