future internet

MDPI

Article

A Privacy Preserving Framework for Worker's Location in Spatial Crowdsourcing Based on Local Differential Privacy

Jiazhu Dai * and Keke Qiao

School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China; qiaokk@shu.edu.cn

* Correspondence: daijz@i.shu.edu.cn

Received: 15 April 2018; Accepted: 13 June 2018; Published: 14 June 2018

Abstract: With the development of the mobile Internet, location-based services are playing an important role in everyday life. As a new location-based service, Spatial Crowdsourcing (SC) involves collecting and analyzing environmental, social, and other spatiotemporal information of individuals, increasing convenience for users. In SC, users (called requesters) publish tasks and other users (called workers) are required to physically travel to specified locations to perform the tasks. However, with SC services, the workers have to disclose their locations to untrusted third parties, such as the Spatial Crowdsourcing Server (SC-server), which could pose a considerable threat to the privacy of workers. In this paper, we propose a new location privacy protection framework based on local difference privacy for spatial crowdsourcing, which does not require the participation of trusted third parties by adding noises locally to workers' locations. The noisy locations of workers are submitted to the SC-server rather than the real locations. Therefore, the protection of workers' locations is achieved. Experiments showed that this framework not only preserves the privacy of workers in SC, but also has modest overhead performance.

Keywords: spatial crowdsourcing; location privacy; local differential privacy

1. Introduction

With the popularity of location-aware mobile devices, such as global position system (GPS) navigation or smart phones, Spatial Crowdsourcing (SC) [1] services have made daily life more convenient. SC is a new type of service that combines location and crowdsourcing, which is used in applications such as environmental sensing, urban planning, convenient travel and so on. In SC, users who are called requesters release their tasks to the spatial crowdsourcing server (SC-server), and other users who are called workers upload their location to the SC-server. The SC-server then assigns the tasks to the workers based on the locations of both the tasks and the workers.

To assign tasks of the requesters to the right workers, SC-server must know the locations of the workers because tasks are time-sensitive and workers should not have to travel long distances to perform and complete a task. However, locations are privacy information, which may reveal private information, such as home address or work places, and may be used to infer personal health, political view, or religious belief information by attackers. Because the SC-server may be untrusted, it may leak the locations of workers to other parties and may also use the locations for other commercial uses. Therefore, protecting the locations of workers is an urgent issue in SC.

In this paper, we propose a framework for protecting the privacy of workers' locations in spatial crowdsourcing based on local differential privacy. Workers' locations are obfuscated locally with local differential privacy and then sent to SC-server for task assignment. The SC-server can only access workers' noisy locations rather than the real ones. The proposed framework does not require the participation of trusted third parties and workers can customize their privacy requirements.

Our contributions are as follows:

1. We propose a new framework that protects the location privacy of workers in SC. In comparison with other solutions, workers' locations are obfuscated locally with noises in the proposed framework so that our framework does not require the participation of any trusted third parties for data collection and privacy processing, and the workers can customize their privacy requirements.
2. We design a task assignment algorithm that assign workers to tasks with obfuscated locations.
3. We conduct experiments on real-world datasets which show that the proposed framework has privacy guarantees and modest overhead performance.

The rest of the paper is organized as follows. Section 2 presents related work. Section 3 introduces the threat model and privacy protection framework. Section 4 discusses the local differential privacy generation algorithm. Experimental set-up and results are presented in Section 5. Finally, Section 6 concludes the paper.

2. Related Work

Spatial crowdsourcing is becoming increasing popular, whereas its location privacy is causing concern. K-anonymity [2–4] has been widely used for privacy protection in location-based systems. The main idea of k-anonymity is that the location of a user is hidden among k other users [5–7]. The other component of k-anonymity involves generating k–1 properly selected dummy points [8,9], and performing k queries to the service provider, using the real and dummy locations. However, the security of k-anonymity is poor when the attacker has background knowledge about the users.

Differential privacy [10] provides a strong privacy guarantee when attackers have background knowledge. To et al. [11] proposed a framework for protecting the privacy of worker locations in SC based on differential privacy. A trusted third party cell service provider (CSP) is needed to clean and noise the worker locations for the SC-server's queries. However, the trusted third party, such as a CSP, is not available sometimes, and users' privacy is not guaranteed even if CSP claim to keep sensitive information safe. On the other hand, the privacy requirements of some workers cannot be satisfied because the privacy budgets of all workers are uniformly distributed by CSP.

In this paper, we propose a framework based on local differential privacy [12,13] to protect workers' locations. The framework can defend against attackers with background knowledges because it achieves differentially-private protection guarantees. Furthermore, the framework does not require a trusted third party such as CSP which used in [11] and workers can customize their own privacy requirements because workers can add noises to their locations locally.

3. Threat Model and Privacy Protection Framework

3.1. Threat Model

Figure 1 shows the basic model of the SC system. There are three participants: requesters, workers, and the SC-server. Firstly, the worker submits their location to the SC-server, which collects and updates a dataset of worker locations. Then, requesters publish their tasks on the SC-server. Next, the SC-server queries the worker locations dataset to assign tasks as they are received from requesters. Finally, workers travel to the designated locations to execute the tasks.

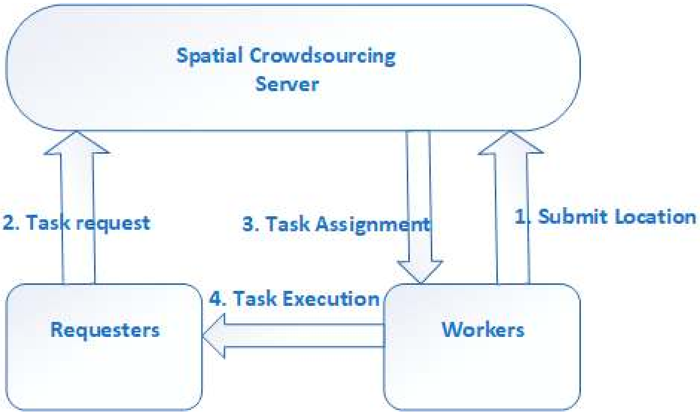

Figure 1. The model of the spatial crowdsourcing (SC) system.

In SC, whether a worker is willing to accept a task or not may depend on the distance between the worker and task. Therefore, the SC-server must obtain the location of workers when assigning a task. However, locations are very important private information for the workers. From the workers' locations, attackers may infer the workers' home or work places, political views, religious inclinations, etc. A worker may not be willing to send their locations to the SC-server with a guarantee of location privacy.

Intuitively, the SC-server cannot be fully trusted. An unpredictable loss may occur to workers when the SC-server leaks workers' locations due to system security vulnerabilities. Additionally, the SC-server may use these locations illegitimately because the location contains commercial value.

3.2. Privacy Protection Framework

Our goal in this paper was to protect the location privacy of workers in SC. We designed a framework (Figure 2) where workers report noisy locations to the SC-server based on local differential privacy. Different from the SC model shown in Figure 1, we added an essential step (step 0) where workers add noises to their real locations by themselves based on a local differential privacy algorithm. Therefore, the SC-server can obtain the noisy locations of workers rather than the real locations during task assignment.

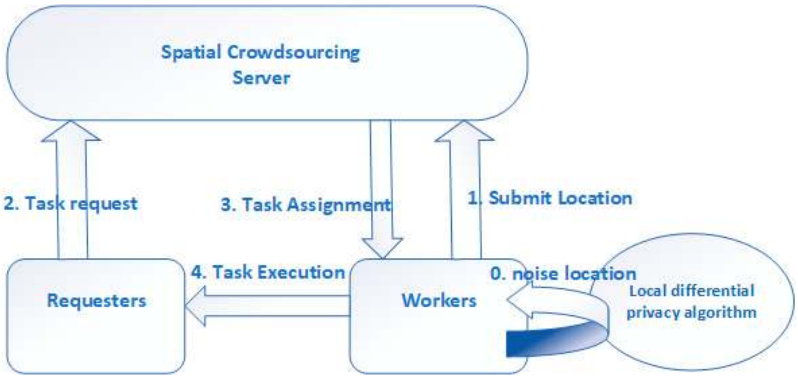

Figure 2. Privacy protection framework.

4. Noisy Location with Local Differential Privacy

The first step in the proposed framework is to add noise to the real worker location for task assignment at the SC-server. In this section, we address the specific requirements of the SC framework based on the geo-indistinguishability method previously proposed [14]. In our approach, we consider the level of privacy within a radius *r* where the worker enjoys *l-privacy* within *r*, where *l* represents the worker's level of privacy for radius and $l = \epsilon r$. The definition of local differential privacy based on location (LDPL) is as follows.

4.1. Definition Local Differential Privacy on Location (LDPL)

A mechanism K satisfies *LDPL* if for all l, l', and $d(l, l') \leq r$:

$$Dp(K(l), K(l')) \leq \epsilon d(l, l') \tag{1}$$

where $d(l, l')$ is the Euclidean distance between l and l', and r is the radius of zone of privacy, and ϵ denote differential privacy parameter.

4.2. Achieve Local Differential Privacy on Location

In this section, we describe a method to satisfy LDPL that can protect workers' locations privacy. First, we used the multivariate Laplacians method to add noise to workers' locations, so we can generate a noisy point. Then, we remap each point to the worker's location.

4.2.1. Generating Noise Point

In this paper's framework, instead of uploading the worker's real location l_0 to the SC-server, we used the noise function to generate a noisy location l to send to the SC-server. Because the location is two-dimensional (2D), we could not use the standard Laplacian method. Instead, we used the method previously described [15,16], which was obtained from the standard Laplacian by replacing $|x - u|$ with $d(x, u)$. The probability density function is given as:

Given the parameter $\epsilon \in R^+$, the actual location $l_0 \in R^2$, and any other location $l \in R^2$, the probability density function (pdf) of the noise mechanism is:

$$D_\epsilon(l_0)(l) = \frac{\epsilon^2}{2\pi} e^{-\epsilon d(l_0, l)} \tag{2}$$

where $\frac{\epsilon^2}{2\pi}$ is a normalization factor.

To draw the point l, we switched the pdf defined in Equation (2) to a system of polar coordinates. Therefore, following the standard transformation formula, the pdf of the polar Laplacian is:

$$D_\epsilon(r, \theta) = \frac{\epsilon^2}{2\pi} r e^{-\epsilon r} \tag{3}$$

where r is the distance of l from l_0, and θ is the angle that the line ll_0 forms with respect to the horizontal axis of the Cartesian system.

Basing the pdf of the polar Laplacian, we can express it as two marginals. We denote these two random variables by r (radius) and θ (angle). The two marginals are:

$$D_{\epsilon,R}(r) == \int_0^{2\pi} D_\epsilon(r, \theta) d\theta = \epsilon^2 r e^{-\epsilon r} \tag{4}$$

and

$$D_{\epsilon,\Theta}(\theta) = \int_0^{\infty} D_\epsilon(r, \theta) dr = \frac{1}{2\pi} \tag{5}$$

Hence, we generate θ as a random number in the interval $[0, 2\pi)$ with uniform distribution base on the $D_{\epsilon,\Theta}(\theta)$ defined in (5).

In addition, we can obtain the cumulative distribution function (cdf) of variable r with $D_{\epsilon,R}(r)$ defined in Equation (4). The cdf is:

$$C_\epsilon(r) = \int_0^r D_{\epsilon,R}(\rho)d(\rho) = 1 - (1+\epsilon r)e^{-\epsilon r} \tag{6}$$

Finally, we can generate r by $C_\epsilon(r)$ defined in Equation (6), and

$$r = C_\epsilon^{-1}(p) = -\frac{1}{\epsilon}\left(W_{-1}\left(\frac{p-1}{e}\right)+1\right) \tag{7}$$

where W_{-1} is the Lambert W function (the -1 branch).

Therefore, we can generate the noisy point *(r, θ)* by following Algorithm 1.

Algorithm 1 Noisy Point Generate Algorithm

```
Input: privacy budget $\epsilon$
Output: $(r, \theta)$
01. $\theta$ = random [0, 2π)
02. $p$ = random [0,1)
03. $r = C_\epsilon^{-1}(p)$
04. return $(r, \theta)$
```

Consequently, in Algorithm 1, we first generate θ as a random number in the interval [0, 2π) with uniform distribution, then we generate a random number p with uniform probability in the interval [0,1). Finally, we generate r using Equation (7).

4.2.2. Remapping Noisy Point to Worker's Location

To generate the worker's noisy location, we then had to remap the noisy point generated by Algorithm 1 to the location of the worker. We can generate the worker's noisy location by following Algorithm 2.

Algorithm 2 Worker's Noisy Location Generate Algorithm

```
Input: worker's real location $l_0$
Output: worker's noisy location $l_{0'}$
01. $\theta$ = random [0, 2π)
02. $p$ = random [0,1)
03. $r = C_\epsilon^{-1}(p)$
04. $l_{0'} = l_0 + (r * cos(\theta), r * sin(\theta))$
05. return $l_{0'}$
```

Intuitively, in Algorithm 2, we first generated the noisy point *(r, θ)* using Algorithm 1. Therefore, we could easily generate the worker's noisy location in step 04 after we generated the noisy point in steps 01 and 03.

5. Experimental

5.1. Design Goals and Performance Metrics

Tasks assignment with noisy data on the SC-server may reduce the effectiveness and efficiency of worker–task matching. Due to the nature of local differential privacy, it is possible that no workers may be notified of the task request. Alternatively, the real distance between the worker and task may be very far. Therefore, we considered the following performance metrics. (1) Assignment Success Rate

(ASR): Due to the noisy locations of the workers, the SC-server may incorrectly assign workers to tasks that are too far, and workers will not accept it. ASR measures the ratio of tasks accepted by a worker to the total number of task requests; (2) Worker Travel Distance (WTD): The SC-server uses noisy data to assign tasks, which may lead to workers have to travel long distances to tasks. WTD measures the distance for the nearest worker to travel to the task.

5.2. Experimental Methodology

5.2.1. Experimental Data Set

In this paper, we validate our method by using the real location Gowalla data. Gowalla is a dataset that contains the check-in history of users. For our experiments, we only used the check-in data for 6100 users in the area of San Francisco, California. We assumed that the users were the workers in the SC system, and assumed the most recent check-in points as their locations. We also modelled each check-in point as a task that was accepted by a worker. The characteristics of the dataset are shown in Table 1.

Table 1. Dataset characteristics.

Tasks	Workers	Workers/km^2
151,000	6100	35

5.2.2. Task Assignment Algorithm

Given a task *t*, we needed to build a matching region (MR) for the task where workers are notified to accept the task. The algorithm must balance two conflicting requirements: determining if a region contains enough workers so that the probability of acceptance of task *t* is the highest, and the size of the MR must be small. To build a matching region, we first set the expected utility (EU), which represents the expected success rate of a task. We then set a maximum travel distance (MTD) for task matching, which is the maximum distance a worker must travel to perform a task. The task assignment algorithm is shown in Algorithm 3.

Algorithm 3 Task assignment algorithm

```
Input: Task $t$, workers' dataset, MTD, $0 < EU < 1$
Output: MR
01. Init MR = {}, U = 0
02. add nearest workers to MR
03. calculate utility U
04. if U > EU, return MR
05. if the radius of MR > MTD, return MR
06. go to step 02
```

Algorithm 3 initially selects the acceptance area of a task centered on task *t* and determines the utility of the task being accepted. For every additional worker, the utility of the task being accepted is recalculated. The algorithm stops when the utility exceeds the threshold EU or the radius of MR exceeds the MTD.

5.2.3. Evaluation Methodology

Given a task *t*, we first used Algorithm 1 that was proposed in Section 4.2.2 to noisy the worker's locations, and then used Algorithm 3 to perform task assignment. We compared our proposed solution with a non-private algorithm that has access to exact workers' locations so that we could evaluate the overhead of privacy. We considered the privacy budge $\epsilon \in \{0.1, \ldots, 0.5, \ldots 1\}$, ranging from strict to lose privacy requirements. We set the expected utility EU $\in \{0.3, 0.5, 0.7, 0.9\}$. We randomly generated 1000 tasks and measured the performance of ASR and WTD.

5.3. Analysis

5.3.1. Assignment Success Rate

We first conducted experiments on the assignment success rate (ASR). Each worker decides whether or not to accept a received task request based on the distance to the task. Therefore, we denoted by acceptance rate (AR) the probability p^a $(0 \leq p^a \leq 1)$ that a worker accepts a task to complete for which they had received a request. We assumed that all workers were identical and independent of each other in deciding to perform tasks. A task is accepted if at least one worker agrees to perform it. Therefore, the utility of MR in Algorithm 2 is:

$$U = 1 - (1 - p^a)^w \quad (8)$$

where w represents the number of workers in MR. We compared the proposed solution that was described in Section 4.2 with the non-private framework. Figure 3 compares the ASR of the non-privacy framework and our privacy framework. The ASR decreases with the privacy budget ϵ. However, tiny distinctions are evident between the ASR of non-privacy framework and our privacy framework.

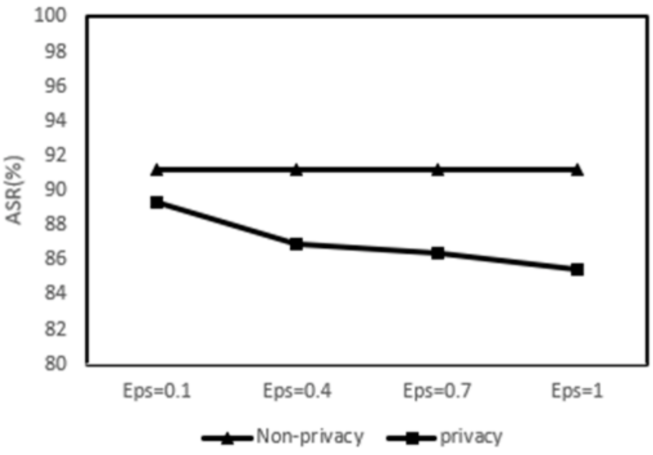

Figure 3. Assignment success rate (ASR) of non-privacy framework and our privacy framework.

5.3.2. Worker Travel Distance

We then examined the worker travel distance under different privacy budgets ϵ. The metric value of WTD was determined as the distance from the task to the nearest worker that accepts the task.

We also compared the WTD between the non-privacy framework and our privacy framework. This was justified by the results shown in Figure 4. The WTD decreases with the privacy budget ϵ. We observed that privacy does not significantly increase WTD compared with the non-privacy case. Therefore, the balance between privacy budget and worker travel distance can be easily found in practical applications.

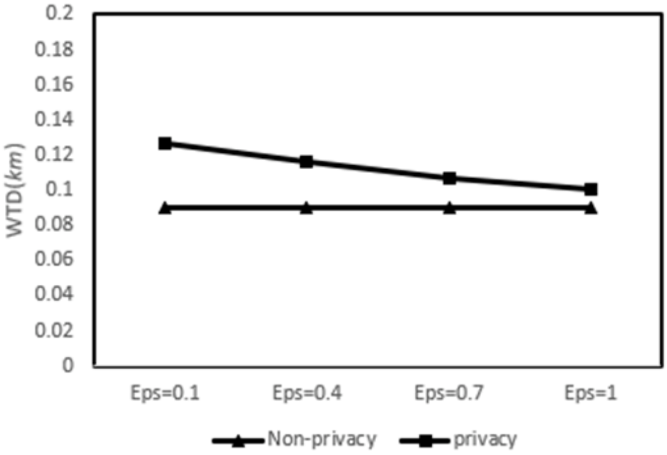

Figure 4. Worker travel distance (WTD) of non-privacy framework and our privacy framework.

6. Conclusions

In this paper, we introduced a novel privacy-aware framework based on local differential privacy for spatial crowdsourcing. We added noises to a worker's location based on multivariate Laplacians. Then, we used the noisy locations to assign tasks. Our experimental results using real data showed that the proposed techniques are effective, and the cost of privacy is practical.

In the future, we will extend our framework to situations where the privacy of both workers and tasks must be protected. We will also focus on protecting the trajectory of workers in SC.

Author Contributions: Conceptualization, J.D. and K.Q.; Methodology, J.D.; Software, K.Q.; Validation, J.D.; Formal Analysis, K.Q.; Investigation, K.Q.; Resources, J.D.; Data Curation, K.Q.; Writing-Original Draft Preparation, K.Q.; Writing-Review & Editing, K.Q.; Visualization, J.D.; Supervision, J.D.; Project Administration, J.D.; Funding Acquisition, K.Q.

Funding: This research was funded by Innovation Program of Shanghai Municipal Education Commission grant number 13YZ013 and Natural Science Fundation of Shanghai grant number 16ZR1411200.

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

References

1. Kazemi, L.; Shahabi, C. GeoCrowd:enabling query answering with spatial crowdsourcing. In Proceedings of the International Conference on Advances in Geographic Information Systems, Redondo Beach, CA, USA, 6–9 November 2012; pp. 189–198.
2. Gruteser, M.; Grunwald, D. Anonymous Usage of Location-Based Services Through Spatial and Temporal Cloaking. In Proceedings of the International Conference on Mobile Systems, Applications, and Services, San Francisco, CA, USA, 5–8 May 2003; pp. 31–42.
3. Gedik, B.; Liu, L. Location Privacy in Mobile Systems: A Personalized Anonymization Model. In Proceedings of the IEEE International Conference on Distributed Computing Systems, Columbus, OH, USA, 6–10 June 2005; pp. 620–629.
4. Mokbel, M.F.; Chow, C.Y.; Aref, W.G. The new Casper:query processing for location services without compromising privacy. In Proceedings of the International Conference on Very Large Data Bases, Seoul, Korea, 12–15 September 2006; pp. 763–774.
5. Bamba, B.; Liu, L.; Pesti, P.; Wang, T. Supporting Anonymous Location Queries in Mobile Environments with PrivacyGrid. In Proceedings of the International Conference on World Wide Web, Beijing, China, 21–25 April 2008; pp. 237–246.
6. Duckham, M.; Kulik, L. *A Formal Model of Obfuscation and Negotiation for Location Privacy*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 152–170.

7. Xue, M.; Kalnis, P.; Pung, H.K. Location Diversity: Enhanced Privacy Protection in Location Based Services. In Proceedings of the International Symposium on Location and Context Awareness, Tokyo, Japan, 7–8 May 2009; pp. 70–87.
8. Kido, H.; Yanagisawa, Y.; Satoh, T. Protection of Location Privacy using Dummies for Location-based Services. In Proceedings of the International Conference on Data Engineering Workshops, Tokyo, Japan, 3–4 April 2005; p. 1248.
9. Shankar, P.; Ganapathy, V.; Iftode, L. Privately querying location-based services with SybilQuery. In Proceedings of the 11th International Conference on Ubiquitous Computing, Orlando, FL, USA, 30 September–3 October 2009; pp. 31–40.
10. Dwork, C. Differential privacy. In Proceedings of the International Colloquium on Automata, Languages, and Programming, Prague, Czech Republic, 9–13 July 2018; pp. 1–12.
11. To, H.; Ghinita, G.; Fan, L.; Shahabi, C. Differentially Private Location Protection for Worker Datasets in Spatial Crowdsourcing. *IEEE Trans. Mobile Comput.* **2017**, *16*, 934–949. [CrossRef]
12. Duchi, J.C.; Jordan, M.I.; Wainwright, M.J. Local Privacy and Statistical Minimax Rates. In Proceedings of the IEEE Symposium on Foundations of Computer Science, Berkeley, CA, USA, 26–29 October 2013; pp. 429–438.
13. Kasiviswanathan, S.P.; Lee, H.K.; Nissim, K.; Raskhodnikova, S.; Smith, A. What Can We Learn Privately? In Proceedings of the IEEE Symposium on Foundations of Computer Science, Philadelphia, PA, USA, 25–28 October 2008; pp. 531–540.
14. Andrés, M.E.; Bordenabe, N.E.; Chatzikokolakis, K.; Palamidessi, C. Geo-indistinguishability: Differential privacy for location-based systems. In Proceedings of the 2013 ACM SIGSAC Conference on Computer & Communications Security, Berlin, Germany, 4–8 November 2013; pp. 901–914.
15. Dwork, C. A Firm Foundation for Private Data Analysis. *Commun. ACM* **2011**, *54*, 86–95. [CrossRef]
16. Dwork, C.; Mcsherry, F.; Nissim, K.; Smith, A. Calibrating Noise to Sensitivity in Private Data Analysis. *Lect. Notes Comput. Sci.* **2012**, *3876*, 265–284.