IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 8, 2020, accepted June 13, 2020, date of publication June 23, 2020, date of current version July 1, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3004470

Protecting Location Privacy for Crowd Workers in Spatial Crowdsourcing Using a Novel Dummy-Based Mechanism

RAED S. ALHARTHI[1], (Member, IEEE), ESAM ALOUFI[1], (Member, IEEE), IBRAHIM ALRASHDI[2], ALI ALQAZZAZ[3], MOHAMED A. ZOHDY[1], (Senior Member, IEEE), AND JULIAN L. RRUSHI[1], (Member, IEEE)

[1]School of Engineering and Computer Science, Oakland University, Rochester, MI 48309, USA
[2]College of Computer and Information Sciences, Jouf University, Sakaka 42421, Saudi Arabia
[3]College of Computer and Information Technology, University of Bisha, Bisha 61922, Saudi Arabia

Corresponding author: Raed S. Alharthi (rsalharthi@oakland.edu)

ABSTRACT Spatial Crowdsourcing (SC) is a new valuable paradigm, relies on crowd workers to perform a set of spatial-temporal tasks at specific locations. It has garnered attention in collecting and processing social, environmental, and other spatio-temporal data by the contribution of individuals, communities and groups of workers in the physical world. The objective of SC is to outsource a set of spatio-temporal tasks to a set of workers, which requires the workers to be physically traveling to the tasks' locations in order to perform them, i.e., taking photos or collecting real time weather information at pre-specified location. Existing solutions require crowd workers to disclose their precise locations to untrustworthy service providers. Location updates and tracking in spatial crowdsourcing raise several privacy concerns in that malicious parties could snoop on crowd workers' whereabouts. Thus, the crowd workers' privacy could be compromised by disclosing their locations to untrusted and possibly malicious parties. This paper provides a novel framework called Dummies' Centroid *(DCentroid)*, which aims at preserving location privacy for crowd workers in SC. The framework adapts an anonymous communication technique using a dummy based approach to generate dummy locations, i.e. decoy locations, and send their centroid points (pseudolocations) to service providers for processing. This paper theoretically analyzes the DCentroid framework and guarantees the crowd workers' privacy, while preserving the functionality of SC, such as the success rate of task assignments, worker travel distances, and system overhead. Practical experimentation on real-world datasets shows that the DCentroid framework protects the crowd workers' location privacy without affecting the various performance parameters of task assignment.

INDEX TERMS DCentroid, location privacy, pseudolocation, spatial crowdsourcing.

I. INTRODUCTION

The term crowdsourcing was first coined by Jeff Howe in 2006 in his article titled ''The Rise of Crowdsourcing'' in [1]. Since then, it has been a widely used umbrella term and a hot topic in the field of computer science. As stated by Jeff Howe, crowdsourcing is simply clarified as, an open call to an undefined large group of people to take a job that was traditionally performed by a designated agent (usually an employee) [2]. Typically, plenty of people can easily participate in crowdsourcing since it happens online such as Amazon Mechanical Turk (AMT) [3]. Those people can complete any desired task posted by corporations or individuals based on their own knowledge, usually for a small amount of money.

With the significant growth of crowdsourcing, the area of Spatial Crowdsourcing (SC) has recently been a popular research topic. SC is a platform for performing spatial-temporal tasks that requires crowd workers to physically travel to the locations of the tasks to perform them. Typically, the tasks submitted by a requester to a centralized spatial crowdsourcing server (SC-server) that act as a

The associate editor coordinating the review of this manuscript and approving it for publication was Mansoor Ahmed.

speculator between the requester and the crowd workers. The SC-server matches and distributes the tasks to interested crowd workers based on their locations and eligibility of the execution. These tasks could be private transportation, traffic information, weather condition, storm updates, or any other tasks at a specific location during a specific period of time.

In typical spatial crowdsourcing tasks, participants contribute to collect data and perform tasks that were requested by the requesters then submit the results to the crowdsourcing platform server. Crowd workers must present at the location of the given tasks in order to perform them and collect the data conventionally using their mobile devices. To this end, when considering a scenario for which requesters are interested in collecting pictures and videos after a natural disaster from its location. With spatial crowdsourcing, the requesters post a task query to a spatial crowdsourcing platform (i.e., service provider) to obtain efficient and faster results instead of traveling to the location. Accordingly, the service provider appeals the mission to the nearby crowd workers in the district area of the disaster to be accomplished. Thereafter, the results are sent back to the requester once it is performed by the participants. Consequently, crowdsourcing has become a valuable tool for collecting data and propagating instant information faster to the requesters.

With the ever-growing tasks and the anonymity of workers in the field of spatial crowdsourcing, the security and privacy concerns have increased, especially when the tasks contain sensitive information such as locations. For instance, the service providers require crowd workers' precise locations in order to efficaciously match them to tasks. Hence, revealing individuals' locations may lead to a wide spectrum of attacks such as stalking, physical surveillance, identity theft, and inferring sensitive information (e,g., individual's health status, alternative lifestyles, activities, political affiliation, relationships and religion) [4], [5]. In addition, many of spatial crowdsourcing servers may not be trusted and leak their data to malicious attackers, which cause crowd workers to not accept participating in spatial crowdsourcing tasks. Moreover, certain attributes of spatial crowdsourcing tasks make it more vulnerable to some types of spatial attacks [6]. Consequently, to ensure that crowd workers engage and contribute in spatial crowdsourcing, ensuring location privacy is substantial.

This paper identifies the location privacy of crowd workers as a uniquely challenge in spatial crowdsourcing. The existing SC-servers require crowd workers to disclose their precise locations in order to efficaciously assign them the tasks and avoid long travel distance. A knowledge of location information of the crowd workers may be utilized to perform several location privacy attacks, which are demonstrated in [7], where adversaries can disclose the crowd workers' private information. In fact, many of the SC-servers may not be trusted and reveal crowd workers location information [8]. Hence, crowd workers hesitate to participate in spatial crowdsourcing tasks since their location information may disclosed by untrusted SC-servers. Consequently, location privacy is a critical privacy issue in SC that need to be preserved to attract crowd workers to participate in spatial-temporal tasks.

Therefore, this paper develops a novel framework called Dummies' Centroid point (DCentroid), which aims at preserving location privacy for crowd workers in SC. More specifically, our framework generates n dummy locations for a crowd worker and calculates the dummies' centroid point to be used as a (pseudolocation) for the crowd worker, and send it to the SC-server for task assignment. The SC-server calculates the distance from the pseudolocation to the task location for assignment qualification. The rationale for generating several dummy locations is to counterbalance any possible faulty choices of such locations. The negative effect of faulty locations would be corrected by the remaining properly-chosen dummy locations. In this paper, we use an n equal to 3, which may tolerate 1 erroneous dummy location. Nevertheless, higher values of n may be chosen for increased tolerance and correctness.

The remainder of the paper is organized as follows. Section II discusses various privacy concerns and location privacy attacks in spatial crowdsourcing. The Objectives of the new mechanism DCentroid are presented in Section III. Then, Section IV summarizes the related works and the location privacy protection techniques. The introduced DCentroid framework is presented in Section V. Section VI, presents the location privacy mechanism. The goal and performance metrics are explained in Section VII. Next, Section VIII demonstrates the performance evaluation and results of DCentroid. Finally, Section X concludes this work and outlines the potential directions that can be followed in order to improve this work.

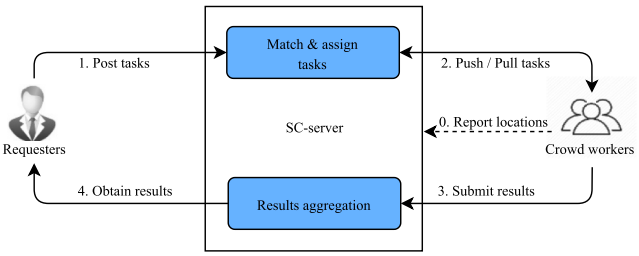

FIGURE 1. Task assignment flow in spatial crowdsourcing.

II. BACKGROUND

A. SPATIAL CROWDSOURCING (SC)

There are many crowdsourcing platforms that provide services for requesters to design and post their tasks. These tasks can be pulled by the crowd workers or pushed to them by the service providers to be executed [9]. Figure 1 illustrates the tasks assignment flow between the requesters and the crowd workers. The three main parties of the spatial crowdsourcing system are:

1) REQUESTERS

The end users who post their tasks through the crowdsourcing services to be executed by crowd workers. They also

determine the eligible criteria to evaluate the crowd workers before they accept performing the tasks. Thereafter, the requesters access and consult the collected data to obtain and verify the results from the service providers after being submitted by crowd workers. Ordinarily, requesters' interest is to maximize the quality of task performance.

2) CROWD WORKERS

The crowd workers who accept to accomplish the tasks that are pushed to them by the service providers or the desired tasks they pull to perform. Then, they return the output data back to the servers and gain their monetary incentive if applied in such tasks. Crowd workers are commonly interested to maximize the exert and value they obtain from performing tasks.

3) SERVICE PROVIDERS

Centralized Spatial Crowdsourcing servers (SC-servers) who serve as the speculators between the requesters and the crowd workers. The SC-servers distribute the interaction mechanism between requesters and crowd workers. Therefore, the SC-servers manage the tasks and distribute them to the interested crowd workers based on their locations and eligibility to execute the tasks. Then, they receive the results of the executed tasks from the crowd workers. Finally, the SC-servers send the aggregation results back to the requesters. Generally, the third-party service providers intention is to maximize the value obtained from using the platform and its functionality.

B. TASKS ASSIGNMENT

In spatial crowdsourcing, there are two types of task assignments in order to match and assign crowd workers to tasks, according to the taxonomy in [10].

1) WORKER SELECTED TASKS (WST)

This type of task assignment known as pull mode, where the service provider publishes all spatial tasks publicly to crowd workers to choose the task that can be accomplished in their neighborhood without coordinating to the service provider [10]. Figure 2 shows an example of *WST* that posted

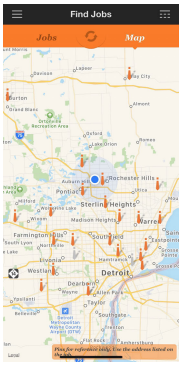

FIGURE 2. Example of distributed pull mode tasks in Field Agent platform. Crowd workers are able to browse all requested tasks that are distributed around their locations.

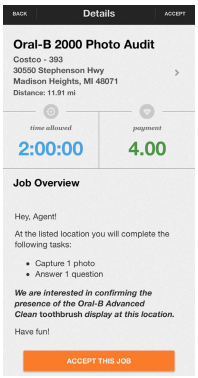

FIGURE 3. An Example of a pull mode task in Field Agent platform. All interested crowd workers are able to see the posted task and have the choice to accept it. Otherwise, the task appears until the time expires.

around a crowd worker in a crowdsourcing platform known as Field Agent [11]. Figure 3 shows the description of a particular task demand to be performed by crowd workers and the payment and time allowed to complete the task.

The advantage of *WST* mode is that crowd workers report their locations only when they choose to perform their desired tasks. However, the disadvantage of this mode is that some spatial tasks may never be selected since the service provider does not have any control over task allocation. Moreover, since crowd workers choose their desired tasks based on their own objectives, which may not result in an inclusive optimal tasks assignment.

2) SERVER ASSIGNED TASKS (SAT)

This type of tasking is known as push mode, where the service provider assigns the tasks to the nearby crowd workers based on their locations and the tasks' requirements.

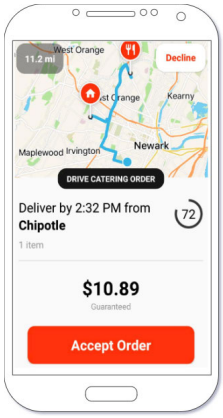

FIGURE 4. An example of a push mode tasking in DoorDash platform. Based on the crowd worker's exact location, the notification is sent to the crowd worker to accept the requested task within a limited time frame.

Figure 4 shows an example of a SAT tasking, where the SC-server (DoorDash) pushes the task request to a nearby crowd worker. The crowd worker then can accept the task

based on the stated motivation. Otherwise, the SC-server pushes the task to the next available crowd worker.

The advantage of this mode is that the SC-server can assign tasks to the closest crowd workers and maximize the overall tasks assignment. However, the disadvantage of this mode is that crowd workers are required to update their locations to the service provider continuously for effective tasks assignment. Figure 5 observes an example of tracking a crowd worker's exact location to be able to participate in performing altruism and volunteer's task such as in PulsePoint's tasks [12].

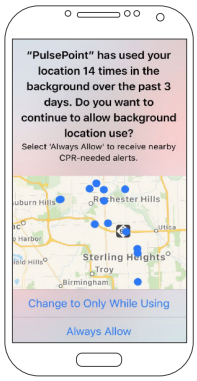

FIGURE 5. An example of a SC-server (PulsePoint) that keeps tracking of a crowd worker's exact location in order to be assigned tasks.

C. LOCATION PRIVACY

Location tracking in spatial crowdsourcing increases numerous privacy concerns about who can snoop into crowd workers whereabouts. The existing SC-servers require crowd workers to disclose their precise locations in order to efficaciously assign them the tasks and avoid long travel distance. In fact, many of the SC-servers may not be trusted and disclose crowd workers location information [8].

Disclosing individual's locations may lead to critical privacy implications such as stalking, physical surveillance, identity theft, and inferring sensitive information (e,g., individual's health status, alternative lifestyles, activities, political affiliation, relationships, and religion) [4]. Krumm [13] shows in his experiments that it is possible to estimate home location of a user within a range of 60 meters by only utilizing the last location information used that day. Sharing more location causes the situation more serious and make it easier to detect private locations. Consequently, location privacy is a critical privacy issue in SC that needs to be preserved to attract crowd workers to participate in spatial-temporal tasks.

III. DCentroid OBJECTIVES

DCentroid adopts an anonymous communication technique using the dummy based technique that generates pseudolocations, which are sent to the SC-server while keeping the actual location hidden from the SC-server. To protect the location privacy of crowd workers using dummy based technique, it is substantial to (i) avoid the situation that adversaries can guess the real location from the dummies and (ii) avoid map matching attack, where the adversaries can remove the locations that crowd workers are not expected to be in (i.e., lakes, mountains, and forests) in order to disclose the real location. However, generating dummy locations that can satisfy the above requirements is challenging in spatial crowd sourcing, since the real location is required to assign tasks. To overcome these challenges, this research designed and implemented a novel scheme that generates dummy locations while the real locations are kept hidden from the SC-server, and guaranteed the success of tasks assignment performance metrics. This is the first dummy generation method that dose not sent the real locations of the crowd workers along with the generated dummy locations to the SC-server.

IV. RELATED WORK

Numerous recent research addressed the topic of location privacy in spatial crowdsourcing such as [14]–[16], [17], [18]–[20]. Location Privacy-preserving in pull mode has been studied in term of participatory sensing in [21]–[24]. A recent survey provides an overview of location privacy attacks in the pull mode and push mode of tasking in [7]. The latest survey that focuses on this topic can be found in [6]. Specifically, Zhang *et al.* [25] proposes a crowd worker coordination framework to protect crowd workers privacy and ensure the quality of the collected information. In their framework, crowd workers coordinate with each other to exchange their locations before submitting the results to the service provider. Consequently, the actual location of each crowd worker cannot be disclosed since all sensitive locations could be evenly visited by any other crowd workers.

Several approaches so far have been proposed to preserve crowd workers location privacy in SC push mode [15], [17], [26]. To *et al.* [19] proposes the first framework based on differential privacy for protecting crowd workers location privacy in SAT mode. In the framework, crowd workers subscribe to a Cellular Service Provider (CSP) that has access to the workers' locations. However, due to the restrictions of the customer privacy protection law, the CSP does not have the right to reveal individuals' locations to third parties [26]. Moreover, their framework requires an online trusted-third-party (TTP) to assign tasks, which incurs in unnecessary task assignment delays [20]. In addition, TTP has to issue new statistical location data when crowd workers update their locations, which causes high communication overheads [27]. A novel solution based on anonymous credentials have been proposed to preserve crowd workers location privacy [28]. Nevertheless, the anonymous technique does not ensure crowd workers locations privacy since it could be inferred by analyzing the reported data and the location's tracing attack [18].

Pournajaf *et al.* [6] proposed another privacy preserving approach that uses a cloacked locations technique. In their approach, crowd workers can cloak their location using either distributed or centralized mechanisms based on other crowd workers locations. Unfortunately, the adversaries can infer

the crowd worker's locations if the crowd worker lactated in a sparsely populated area [7]. Another differential *k*-anonymity-based has been proposed for location privacy [29]. In their technique, they combine *k*-anonymity and differential privacy-preserving to achieve the anonymity of crowd workers. However, in *k*-anonymity, an adversary can guess the location of the crowd worker with probability no higher than $1/k$ [30].

V. DCentroid: LOCATION PRIVACY PROTECTION TECHNIQUE

DCentroid is a new introduced approach that is motivated by the dummy based technique to achieve location privacy for crowd workers in SC [31]. The presented privacy model takes into account the required travel distance from crowd workers to tasks, as well as keeping the location hidden from the SC-server. The duality of this approach makes it more appropriate in spatial crowdsourcing than other purely techniques. This section presents the introduced system model and its efficient algorithm. The major notations used throughout this section present in Table 1.

TABLE 1. Summary of notations.

Notation	Definition
w_i	Crowd worker $_i$
t_j	Task location
r_i	Possible dummies area
$w.l_i$	Real location of the crowd worker $_i$
$w.d_{a-c}$	Dummy location of crowd worker $_i$
p_i	pseudolocations (the centroid point of the dummies $_{a-c}$)
$P_t(p_i, t_j)$	Distance between location p_i and task location $_j$
$\widehat{w.l}_{i,j}$	Estimated distance between crowd workers and tasks locations

A. DCentroid SYSTEM MODEL

The presented model intends to hide the real location of a crowd worker by sending a pseudolocation, the centroid points of the dummy locations, to the SC-server instead of sending the real location; whereas the tasks will be assigned successfully based on the pseudolocations. Figure 6 demonstrates the overall procedure of the introduced system model, which operates as follows:

1) The crowd workers generate three dummy locations (A, B, and C) around their real locations.
2) The crowd workers calculate the centroid points of the generated three dummies and considers it as their private pseudolocations.
3) The crowd workers send only the generated pseudolocations to the SC-server.
4) The SC-server utilizes the pseudolocations to calculate the estimated distance from the crowd workers to tasks

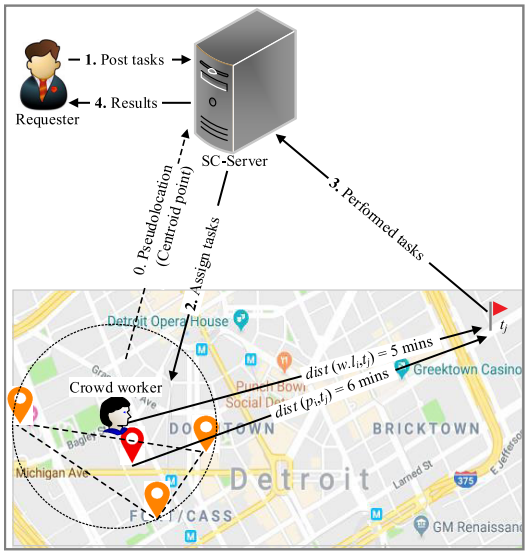

FIGURE 6. System model.

using the standard Euclidean distance $\widehat{w.l}_{i,j}$ as below.

$$\widehat{w.l}_{i,j} = P_t(p_i, t_j) \tag{1}$$

where P_t is the Euclidean distance between the locations of the centroid point p_i, and t_j is the targeted tasks' locations.

5) The SC-server assigns the tasks to crowd workers based on the calculated estimated distance $\widehat{w.l}_{i,j}$.

In this manner, the privacy of the crowd workers' locations are preserved since the SC-server is dealing with pseudolocations and the real locations are kept hidden from adversaries.

B. DUMMY GENERATION MODEL

Generating dummy locations randomly de-emphasize the privacy requirements, where the adversary can quietly distinguish the real locations from the dummy locations [32]. To adopt the dummy generation based in spatial crowdsourcing and to satisfy the requirements of the anonymous area of the real locations, this research introduces a novel dummy generation algorithm, called Direct-Dummy algorithm. This algorithm prevents the downplay by employing sixteen direction to constrains and vary all the possible dummies to specific direction, each time crowd workers want to update their locations, as shown in Figure 7. To the best of our knowledge, this is the first dummy based technique that does not send the real locations along with the dummy locations.

C. DIRECT-DUMMY ALGORITHM

The Direct-Dummy algorithm runs locally on the crowd workers' devices to have access to their exact locations without a third party involvement. Algorithm 1 illustrates the pseudocode for generating the Direct-Dummy algorithm. DCentroid constrains and varies the random locations pick,

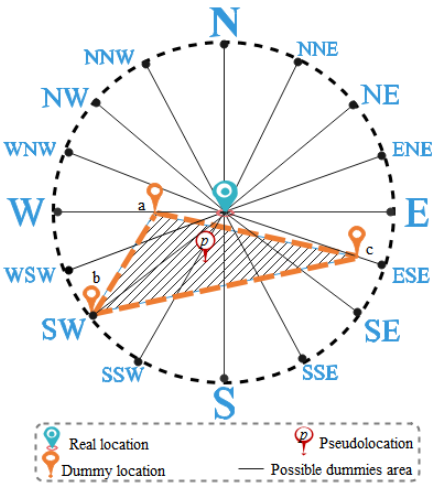

FIGURE 7. Direct-Dummy based.

the algorithm has a custom Direct-Dummy set of array that contains multiple point (x, y) coordinates that are distributed in the sixteen directions. The algorithm starts by initializing all elements of its dummies to the position of (x_i, y_i), the exact real location of the crowd worker. Each direction has one to three positions, where each dummy randomly selects one position from the set to allocate its location. The algorithm then removes the selected point from the set to avoid duplicate selection of the same location.

D. DIRECT-DUMMY DISTANCE

To guarantee the crowd worker's location privacy, the distances between the real location and its dummy locations are specified as shown in Equation 2:

$$dist(w.l_i, w.d_a) \geqslant \frac{1}{2} dist(w.d_a, w.d_b) \leqslant \frac{2}{3} dist(w.d_b, w.d_c) \tag{2}$$

where $w.l_i$ is the real location of a crowd worker and $w.d_a$, $w.d_b$, and $w.d_c$ are the generated dummy locations as shown in Figure 8.

In addition, the minimum distance from the real location to the dummy locations D^{min}_{dist} is set to be one Direct-Dummy Unit Size (DUS) as follows:

$$D^{min}_{dist} = min_{dist}(w.l_i, w.d_{a-c}) = DUS \tag{3}$$

To achieve the travel distance metrics, the maximum dummy location distance is defined to not exceed more than three DUSs as follows:

$$D^{max}_{dist} = max_{dist}(w.l_i, w.d_{a-c}) = 3 \times DUS \tag{4}$$

Hence, the pseudolocation of a crowd worker must be within the maximum distance between the real location and generated random dummy locations, as pseudolocation $< D^{max}_{dist}$. Note that the DUS corresponds to the level of the crowd workers' privacy. Thus, the DUS range is set based on the desired Location Privacy radius (LP_r) area, where $LP_r = (w.l_i, p_i)$. Figure 9 shows the obtained crowd worker's LP_r when the $DUS = 335$, which means that the maximum possible distance of the generated dummy locations are within radius 1.05 km to the real location of the crowd worker, and all dummy locations are located in the maximum possible distance region.

Algorithm 1 Direct-Dummy Algorithm for Generating Dummy Locations

Input: Crowd worker exact real location $w.l_i = (x_i, y_i)$
Output: Dummy locations $w.d_i = \{w.d_a, w.d_b, w.d_c\}$

All elemnts of $w.d_i$, are initialized to (x_i, y_i) position.
Direct-Dummy=
$[(x+0, y+1), (x+0, y+2), (x+0, y+3)$ **N**
$(x+1, y+2)$ **NNE**
$(x+1, y+1), (x+2, y+2)$ **NE**
$(x+1, y+2)$ **ENE**
$(x+1, y+0), (x+2, y+0), (x+3, y+0)$ **E**
$(x+2, y-1)$ **ESE**
$(x+0, y-1), (x+2, y-2)$ **SE**
$(x+1, y-2)$ **SSE**
$(x+0, y-1), (x+0, y-2), (x+0, y-3)$ **S**
$(x-1, y-2)$ **SSW**
$(x-1, y-1), (x-2, y-2)$ **SW**
$(x-2, y-1)$ **WSW**
$(x-1, y-0), (x-2, y-0), (x-3, y-0)$ **W**
$(x-2, y+1)$ **WNW**
$(x-1, y+1), (x-2, y+2)$ **NW**
$(x-1, y+2)$ **NNW**]
for *each* $w.d$ in $w.d_i$ **do**
    r = *RandomSelect* from (**Direct-Dummy**)
    $w.d \leftarrow r$
    Direct-Dummy.*remove*(r)
end for
return $w.d_a \leftarrow (x_a, y_a)$, $w.d_b \leftarrow (x_b, y_b)$, $w.d_c \leftarrow (x_c, y_c)$

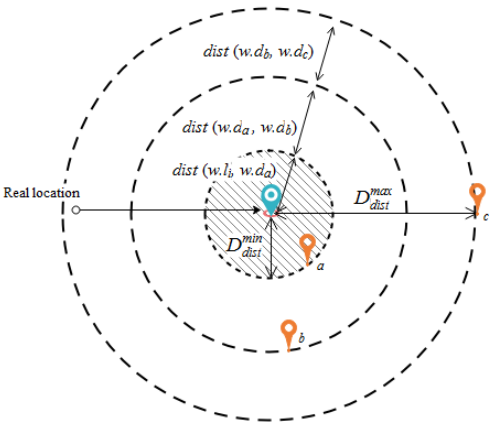

FIGURE 8. Maximum and minimum distance between a real location and its dummy locations.

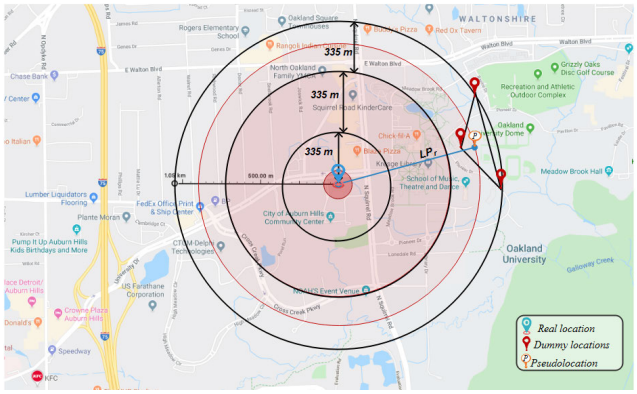

FIGURE 9. An example of the obtained crowd worker's LP_r when the $DUS = 335$.

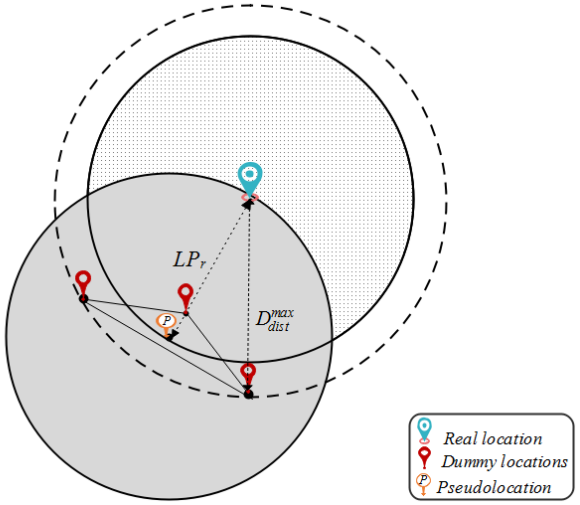

FIGURE 10. Location privacy region.

VI. LOCATION PRIVACY MECHANISM

The essential purpose of this work is to protect the crowd workers' locations from the untrusted SC-Server and guarantees that their locations are hidden to preserve their privacy.

Moreover, to ensure the privacy, DCentroid's mechanism dose not require any third parties involvement. However, the SC-Server may enable the communications between crowd workers and the task requesters once the crowd workers are assigned the tasks, where the crowd workers could directly disclose their information to the task requesters. This case is outside this work's scope since the crowd workers have the right to disclose their information to the task requesters. To protect their information after the task is assigned, existing techniques and identity protection can be applied such as pseudonyms and anonymous methods.

Since crowd workers have to report their locations to the SC-servers to engage in the task's assignment procedure, the privacy of their location is exposed. On the other hand, existing location privacy-preserving mechanisms are limited to be applied in crowdsourcing during task assignment. Therefore, guaranteeing the privacy to the crowd workers is sufficient motivation to engage them in crowdsourcing task assignment.

A. LOCATION PRIVACY ANALYSIS

DCentroid ensures the privacy protections for crowd workers by preventing any possibility of guessing the exact real locations from the adversary when having the pseudolocation's information, and even if the construction algorithm of the generating dummy locations is known. Assuming the adversaries know the generation algorithm of the generated pseudolocations of the crowd workers, the way the real locations appear on their sides are as shown in Figure 10 within the gray area. Due to the random generation of the pseudolocations, the adversaries received the pseudolocations of the crowd workers and their assumptions of the real locations would be within the maximum Location Privacy radius (LP_r) of the generation dummies. Thus, the probability of estimating the real location is extremely low.

Therefore, the expected location privacy performance of DCentroid can be measured in term of the Expected Estimation Error (EE) of "rational" Bayesian adversary presented in [33], called Location Privacy (LP). Table 2 summarizes the notations introduced throughout this section.

Formally, LP is computed as:

$$LP = \sum_{w.l, \widehat{w.l}, w.l'} \psi(w.l) f(w.l'|w.l) h(\widehat{w.l}|w.l') d(\widehat{w.l}, w.l) \quad (5)$$

Assuming that the adversary has access to the crowd worker's profile, and use this side knowledge to guess the real locations, as expressed in terms of prior location information. The adversary's goal is to use such prior location information, and combine it with the provided information by the privacy mechanism to infer the real locations of the crowd workers. Accordingly, the probability of error between the estimated location $\widehat{w.l}$ and the real location $w.l$, if the crowd worker's goal is to hides the real location would be:

$$d(\widehat{w.l}, w.l) = \begin{cases} 0, & \text{if } \widehat{w.l} = w.l \\ 1, & \text{otherwise} \end{cases} \quad (6)$$

In this term, any location that is different from the real location of the crowd worker produces in a high level of LP. On the other hand, if the crowd worker' location privacy is to minimize the distance between the real location $w.l$ and the estimated location $\widehat{w.l}$, the distortion function would be the squared-error distortion as follow:

$$d(\widehat{w.l}, w.l) = (\widehat{w.l} - w.l)^2 \quad (7)$$

In particular, DCentroid can protect crowd workers from the following location privacy attacks mentioned in [7].

- Location distribution attacks: This framework does not rely on other crowd workers when generating the pseudolocations. Hence, adversaries cannot infer the

TABLE 2. Summary of notations.

Notation	Definition
$w.l$	Real location of the crowd worker $_i$
$w.l'$	Reported pseudolocation of the crowd worker (the centroid point p_i)
$\psi\ (w.l)$	Prior location information (probability of being at location $w.l$)
$f\ (w.l' \mid w.l)$	Probability of replacing $w.l$ with the mechanism $w.l'$
$\widehat{w.l}$	Adversary's estimate of the crowd worker's real location
$h\ (\widehat{w.l} \mid (w.l')$	Adversary's attack function (probability of the reported location $w.l'$ is remapped into $\widehat{w.l}$)
$d\ (\widehat{w.l}\ , w.l')$	The distance between location $\widehat{w.l}$ and the real location $w.l$
LP	Expected location privacy for a crowd worker with profile $\psi\ (.)$ using protection f against attack h

real locations of the crowd workers if they are not distributed homogeneously.

- Map matching attacks: The generated pseudolocations of the crowd workers can be located at any location points around their real locations. Hence, the adversary does not have a specific region of the exact location of a crowd worker. Therefore, eliminating the areas that the crowd worker is unexpected to be in, does not help the adversary to assume the exact location of the crowd worker.
- Task tracking attacks: This attack occurs when a crowd worker is requested to perform more than one continuous task, and cloaked with different crowd workers in each task while the continuous tasks are still running. The adversary can identify the crowd worker's location by linking the respective tasks locations to the respective crowd worker. However, DCentroid prevents such an attack since crowd workers are not cloaked with each other in such a framework.
- Location trajectory attacks: In order to prevent this attack in DCentroid approach, crowd workers update the SC-server with their generated pseudolocations instead of their exact locations. For each update, crowd workers generate the pseudolocations with varying directions.
- Maximum movement boundary attacks: The adversaries are unable to link two consecutive pseudolocations to the crowd workers exact locations, consecutively. Hence, knowing only the maximum possible movement speed of the crowd workers is not efficacious way for the adversaries to launch such attacks.
- Location inference attacks: Sharing crowd workers' earlier exact locations could help the adversaries to deduce their private and sensitive locations by knowing their paths. However, DCentroid approach prevent crowd workers from such attacks by updating their pseudolocations to the SC-server and varying directions on each generations of the dummies locations. Thus, the crowd workers' paths are hidden from the SC-server.

B. LOCATIONS PRIVACY FOR STATIC CROWD WORKERS

Since the Spatial Crowdsourcing server (SC-server) has to keep track of crowd workers' locations to assign them tasks, they have to send and update their locations continuously to SC-server to request task assignment. Hence, with the introduced DCentroid technique, the history of crowd workers' dummy locations and pseudolocation can be exploited to anticipate their real locations if the crowd workers send numerous pseudolocations related to the identical static locations (e.g., home and work).

This is considerably a challenging problem for the static crowd workers in the DCentroid system model. However, to overcome this issue, crowd workers' pseudolocations remain the same as in the previous instance when generating pseudolocations in regards to the same location. DCentroid system obtains the exact dummy locations to the previous dummies to get the exact pseudolocations when updating the same locations to the SC-server. In this manner, crowd workers will not send to the SC-server more than one pseudolocations related to their static locations and their background knowledge will be limited to the SC-server.

C. LOCATIONS PRIVACY IN SPARSE AREA

Most of the exciting location privacy techniques relay on location-cloaking techniques, where they customize a cloaking region for users and distinguish their identity with each others. However, such a technique dose not protect crowd workers' locations privacy if they are located in a sparsely populated area. Location distribution attacks occurs when crowd workers are not distributed homogeneously in a sparsely populated area [7]. Thus, DCentroid approach takes this limitation into consideration. Applying the designed approach in a sparsely populated area fulfills the locations privacy for crowd workers, since the generation of their pseudolocations do not relay on any other crowd workers' locations or third-parties involvements.

D. LOCATION PRIVACY TRADE-OFF

The DCentroid approach is dealing with pseudolocations of crowd workers, which may cause Travel Distance Error (TDE). The *TDE* may affect both the SC-server and the crowd workers. The effect can be an advantage for the crowd workers and disadvantage for the SC-server, and vice versa. Moreover, the crowd workers may be rewarded more than they deserve if they are assigned to tasks of a lesser

distance. Alternatively, crowd workers may be assigned to tasks with distances more than its promised benefit. For instance, a crowd worker is assigned to a task where the actual distance from the real location to the task is 600 meters, and the distance from the pseudolocation is 900 meters as shown in Figure 11. Thus, the crowd worker benefits from the SC-server cost of 300 meters more than the actual cost.

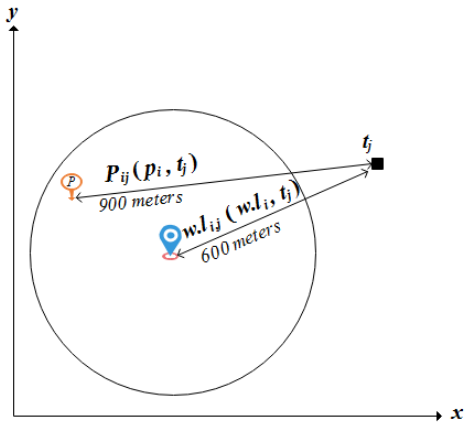

FIGURE 11. An example of server impact due to TDE.

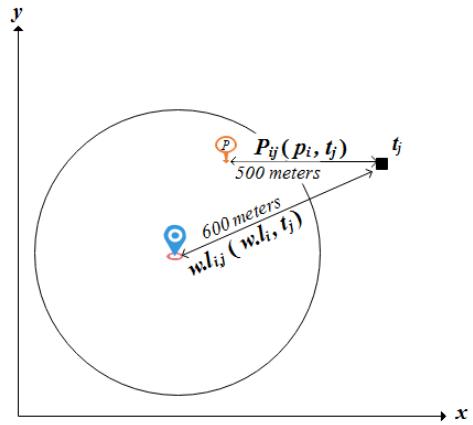

FIGURE 12. An example of crowd worker impact due to TDE.

On the other hand, a crowd worker may be assigned to a task where the actual distance from the real location to the task is 600 meters, and the distance from the pseudolocation is 500 meters as shown in Figure 12. In this case, the crowd worker losses a travel cost of 100 meters from the actual cost. However, the variation of *DTE* does not affect the privacy level of the crowd workers. In other words, less *TDE* does not mean less privacy as shown in Figure 13.

The second effect is from the task assignment perspective. The crowd workers may be notified of task assignments where the actual distance from the task location is undetermined, whereas nearer crowd workers are not notified. In this case, the tasks may not be accepted. The challenge is to

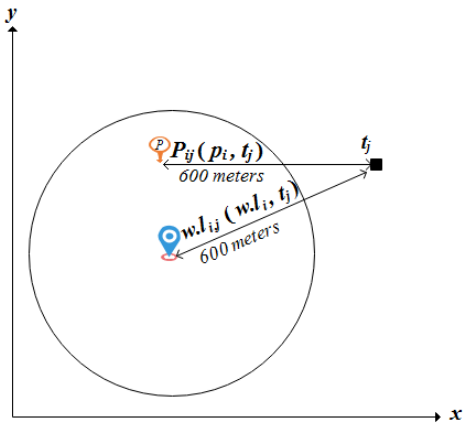

FIGURE 13. An example of a non-impacted task from TDE.

maintain the *TDE* as low as possible, which can be defined as:

$$TDE = |(w.l_i, t_j) - (p_i, t_j)| \tag{8}$$

where $w.l_{i,j}$ is the distance between the real locations of the crowd workers and the tasks, and $p_{i,j}$ is the distance between the pseudolocation of the crowd workers and the tasks.

VII. GOALS AND PERFORMANCE METRICS

Due to crowd workers having to physically visit the task locations, travel distance is critical in SC. The SC-server requires the crowd workers exact locations to calculate the travel distance to the targeted task locations. Accordingly, the goal of the DCentroid system is to substitute the exact locations of crowd workers with alternative locations to be used by the SC-server to calculate the estimation travel distance to tasks while preserving crowd workers location privacy.

To evaluate DCentroid framework and demonstrate its effectiveness, this research focuses on the following performance metrics adopted from a similar approach [18]:

1) ASSIGNMENT SUCCESS RATE (ASR)

Since the DCentroid system is dealing with imprecise locations of crowd workers and sending their estimation locations to a SC-server, the SC-server may incorrectly assign a crowd worker to a task that is too far from the task's location and the crowd worker can reject it. On the other hand, a closer crowd worker does not receive the request. The measurement of *ASR* is the ratio of the accepted tasks by crowd workers to the total number of task requests. The challenge here is to keep *ASR* close to 100% by performing all or most of the requested tasks.

2) WORKER TRAVEL DISTANCE (WTD)

Without the precise locations of crowd workers, the SC-server is not capable to accurately evaluate the distance from crowd workers to tasks. Thus, crowd workers may have to travel

longer distance to their assigned tasks. The challenge is to minimize the *WTD*, even when the real locations of crowd workers are hidden from the SC-server.

3) SYSTEM OVERHEAD

Assigning the tasks based on imprecise locations raises task assignment complexity, which demonstrates scalability problems. The considerable metric to measure the system overhead is the average number of notified crowd workers (ANW). *ANW* impacts the computation overhead of task assignment, which relies on how many crowd workers need to be notified of a task assignment request.

VIII. PERFORMANCE EVALUATION

The DCentroid scheme determines an effective location privacy protection mechanism for the crowd workers in SC. The optimal DCentroid is designed under the constraint of ensuring a task's assignment performance metrics. This section evaluates the relation between the introduced location privacy approach (DCentroid), and a non-privacy approach that has access to crowd workers real locations (GroundTruth), to present the promised work's optimization. First, it presents the experimental methodology followed by the performance evaluation in spatial tasks assignment and then discusses the results under various experimental datasets.

A. EXPERIMENTAL SETUP

This research performs a set of experiments on real-world data to evaluate the introduced framework's performance. In fact, this evaluation adapts a similar approach to demonstrate the effectiveness of the introduced location privacy method under the varying metrics [18]. The experiments were performed by varying the *MTD* with the other parameters and run on an Intel Core i7-7700 CPU @ 3.60GHz with 32 GB RAM. The algorithms were implemented using the open source libraries in Python.

B. REAL DATASET

A real-world dataset that was collected from a sparsely populated area, and another dataset that was collected from a densely populated area have been used to evaluate the performance of DCentroid scheme. The first real-world dataset was collected from Yelp,[1] a popular location-based social network. The dataset corresponds to a collection of user reviews about restaurants in the greater Phoenix, Arizona area (*Ye.-PHO*). It includes 70,817 users, locations of 15,583 restaurants, and 11,434 check-ins at different locations [34]. In this experiment, the Yelp users were assumed to be the crowd workers with their check-ins as their real locations, and the restaurants are considered as the targeted tasks locations.

The second real-world dataset is based on data collected from another popular location-based social network called Foursquare.[2] The dataset contains 1,083 users, 38,333 locations, and 227,428 check-ins at various locations in New York City from April 2012 to February 2013 (*FO.-NYC*) [35]. The users were assumed to be the crowd workers and their last check-ins were considered the real locations. The locations associated with the check-in were considered as the tasks locations.

C. DATASET SETTINGS

To simulate crowdsourcing tasks assignment in the experiment, the Yelp dataset uniformly sampled 66,000 real locations for crowd workers and 15,000 restaurants to emulate the tasks' locations. And total numbers of 7,935 check-ins were randomly sampled to simulate tasks and 34,925 real locations for crowd workers from Foursquare dataset. The experiment parameters are listed in Table 3. For each dataset independently, the total tasks were divided into 20% tasks for each assignment rounds, and the crowd workers' Maximum Travel Distance (*MTD*) $\in \{1, 2, 4, 6, 8, 10\}$ in km. The travel distance between crowd workers and tasks' locations is proportional to their Euclidean distances.

D. DIRECT-DUMMY UNIT SIZE SETTINGS

To determine the Direct-Dummy Unit Size (*DUS*), all the crowd workers' locations were generated by their dummy locations to obtain the pseudolocations with four various *DUS* generations. For each generation, the *DUS* $\in \{110, 220, 330, 440\}$ in meters, independently. The distributions of crowd workers' pseudolocations in the first two generations 110 and 220, did not obtain the best locations privacy destinations for the crowd workers, since the majority of the generated pseudolocations were distributed near to the real locations, as shown in Figure 14 and Figure 15, respectively. On the other hand, the *DUS* (330 meters) generation performed satisfied distribution, as shown in Figure 16, and was nominated for the pseudolocations generation in this experiment.

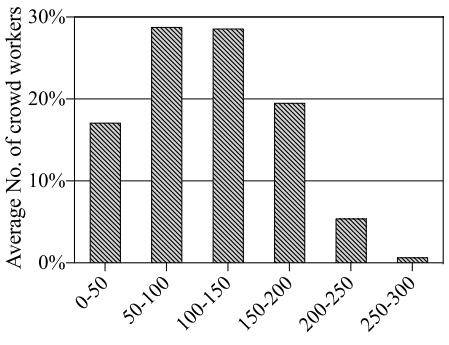

FIGURE 14. Distributions of pseudolocations when *DUS* = 110 meters.

E. DISTANCE ESTIMATION

As mentioned earlier, the actual locations of crowd workers are hidden from the SC-server in this work.

[1] https://www.yelp.com/dataset/challenge

[2] https://foursquare.com/

TABLE 3. Experiment parameters.

Parameters	Yelp: Phoenix	Foursquare: New York City
Name	(*Ye.-PHO*)	(*Fo.-NYC*)
No. of workers' real locations	66,000	34,925
No. of tasks	15,000	7,935
Avg. No. of tasks/ round	3,000	1,587
MTD	1, 2, 4, 6, 8, 10 (km)	1, 2, 4, 6, 8, 10 (km)
DUS	110, 220, 330*, 440 (m)	110, 220, 330*, 440 (m)

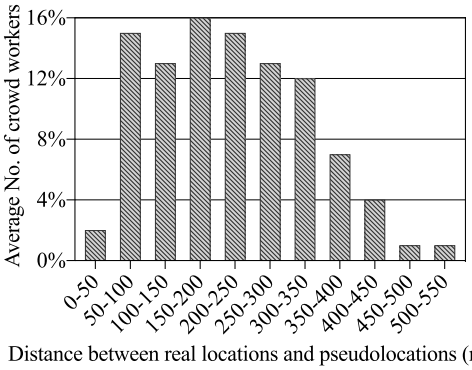

FIGURE 15. Distributions of pseudolocations when *DUS* = 220 meters.

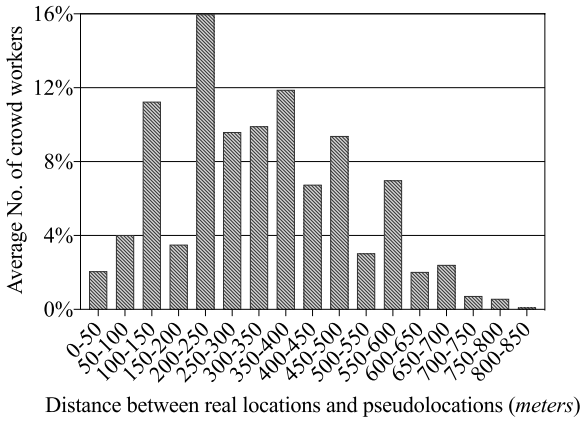

FIGURE 16. Distributions of pseudolocations when *DUS* = 330 meters.

Therefore, the SC-server is required to deal with the locations of the crowd workers uncertainty to estimate the travel distances. Thus, this work introduces a simple method to assume the expected travel distances between crowd workers' and tasks as follows. Considering that the positions of the three dummies are in a triangle shape (ABC) as illustrated in Figure 7, the average of the three vertices can be utilized to obtain pseudolocations, the centriod point of the dummy locations, as follows:

$$p_i = \left(\frac{\sum_{i=1}^{n} x_{w.d_i}}{n}, \frac{\sum_{i=1}^{n} y_{w.d_i}}{n} \right) \tag{9}$$

where x and y represent the coordinate difference between the dummy locations $w.d_{a-c} \in r_i$.

Taking into consideration the pseudolocations p_i are sufficient to estimate the distance from the crowd workers w_i to the task locations t_j, which is the Euclidean distance from the pseudolocations p_i to the task location t_j.

IX. EXPERIMENTAL RESULTS

A. OVERVIEW OF RESULTS

DCentroid scheme substitutes the crowd workers real locations to pseudolocations during the assignment simulation. Hence, the most significant factor that might impact the task assignment performance metrics is the travel distance to the tasks, which may cause the crowd workers to reject the tasks. Therefore, to assign tasks to the crowd workers in the simulation, the available crowd workers are notified for task assignment based on first come first serve, where they accept or reject the tasks based on their Maximum Travel Distance (MTD) to the tasks.

To evaluate the experimental results, the two simulated task assignments (*Ye.-PHO* and *Fo.-NYC*) are compared with a non-privacy approach that has access to the crowd workers' exact locations (GroundTruth). All reported metrics are based on the average of five task assignments rounds for various parameters in each experiment of a dataset. The overall results confirm that the DCentroid scheme does not affect all the task assignment metrics by estimating the distance between crowd workers and tasks, especially, when the travel distance between the crowd workers and tasks increased. The most important factors in SC are the *ASR* and the *WTD*, which are not significantly affected by the introduced DCentroid location privacy approach comparing to GroundTruth.

B. DCentroid PRIVACY TRADE-OFF

Figure 17 shows the *TDE* compared with the GroudTruth approach, which obtains zero *TDE*. This is expected since it has access to the real location data of crowd workers. However, the results show that the *TDE* drops in DCentroid scheme as the *MTD* increases between crowd workers and tasks. Moreover, the results show that there is a slight difference between the *TDE* in *Ye.-PHO* and *Fo.-NYC*. This is because the *Fo.-NYC* dataset were collected from a densely populated area, and there are more crowd workers near to each other. Thus, the DCentroid approach performs better for longer distance task assignments.

C. DETAILS OF RESULTS

Below are the details of each of the performance metrics illustrated in Section VII. Each metric, compared with a

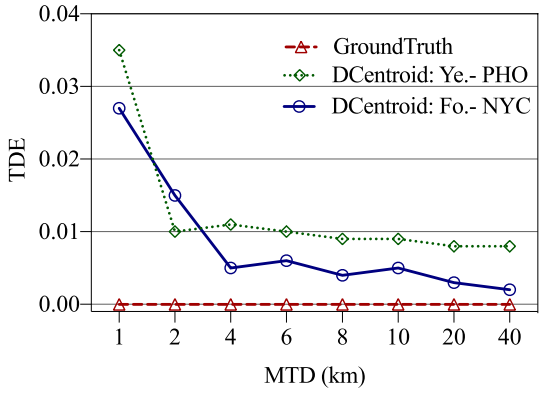

FIGURE 17. Location privacy trade-off.

GroudTruth approach that has access to the real locations of the crowd workers by varying the *MTD* of the crowd workers in *Ye.-PHO* and *Fo.-NYC* task assignments. In particular, ASR reports the average number of assigned tasks and *WTD* states the average travel distance across the assigned tasks. The system overhead was calculated as the average number of candidate crowd workers per task. All reported metrics are based on the average of five task assignments rounds for various parameters in each experiment as follows:

1) ASSIGNMENT SUCCESS RATE (ASR)

Figure 18 shows the *ASR* results when varying the *MTD* of crowd workers, which obtains a slight difference between DCentroid scheme and the GroudTruth, when the crowd workers' *MTD* decreased. It shows that DCentroid performs higher *ASR* in *Ye.-PHO* and *Fo.-NYC*.

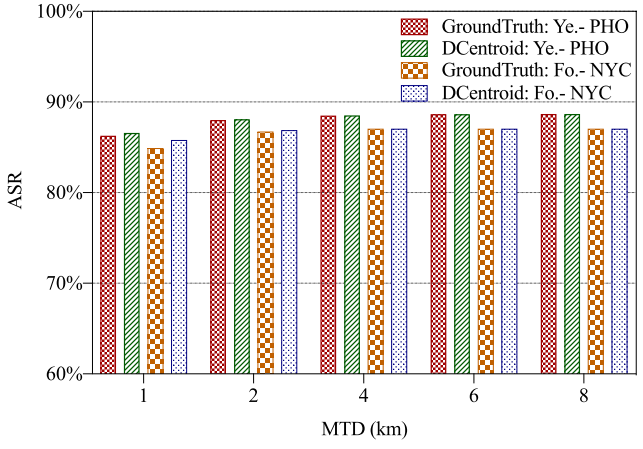

FIGURE 18. Average *ASR* over *DUS* = 330 meters.

This is because the crowd workers might be assigned to tasks in which the actual distances are less than the estimated distances, which yields a higher number of accepted tasks. On the other hand, when increasing the *MTD* of the crowd workers, the *ASR* is not affected by the DCentroid privacy scheme in both cities. This is because the *TDE* decreased when increasing the *MTD* of crowd workers. To this end, the evaluation of the two datasets indicates that the *ASR* is not impacted greatly by the introduced scheme, which is perhaps one of the most significant factors of task assignment in spatial crowdsourcing.

2) WORKER TRAVEL DISTANCE (WTD)

Figure 19 shows the *WTD* results in various *MTD* between crowd workers and tasks. The GroudTruth: *Ye.-PHO* and GroudTruth: *Fo.-NYC* achieve lower travel costs, which is predictable as they have access to the real locations of crowd workers. It is observed that DCentroid and GroudTruth for both cities obtain similar travel cost when increasing the *MTD*. Note that the travel cost in *Fo.-NYC* is decreases in the long travel distance comparing to the travel cost in *Ye.-PHO*. This is due to the significant difference in density area where the tasks distributions do not require high travel cost in the densely populated area *Fo.-NYC*.

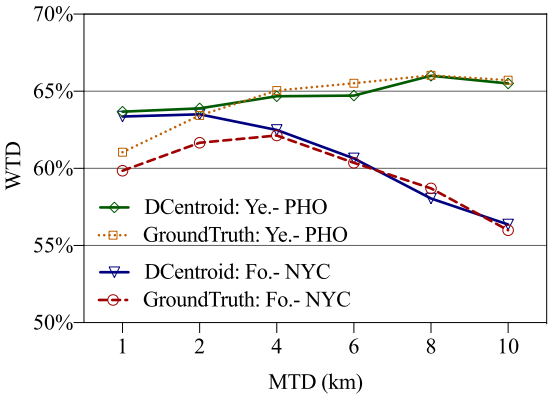

FIGURE 19. Average *WTD* over *DUS* = 330 meters.

3) SYSTEM OVERHEAD (ANW)

Figure 20 shows the results of the system overhead when varying the crowd workers' *MTD*. It can be noticed that the system overhead of the introduced privacy approach is slightly higher than the GroudTruth. This is expected, because DCentroid may notify crowd workers of tasks that are not reachable to them where they reject the requests, since the distance between crowd workers to tasks are estimated in the privacy approach. However, this overhead occurs only with less *MTD* of crowd workers to tasks, for both the introduced privacy approach and the GroudTruth.

The reason for this is that most of the requested task assignments are higher than crowd workers *MTD*, which forces the system to notify more crowd workers to be assigned the tasks. In fact, the system overhead in *Fo.-NYC* is less than the system overhead in *Ye.-PHO*, which is expected due to the availability of tasks assignments in a densely populated area are more likely in a densely populated area.

D. EFFECTS OF VARYING DATASETS

Table 4 summarizes the variation of the considered performance metrics when increasing *MTD* in *Ye.-PHO*.

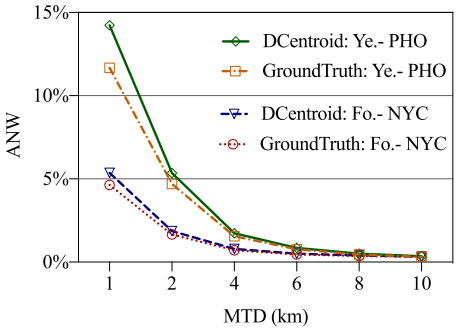

FIGURE 20. Average *ANW* over *DUS* = 330 meters.

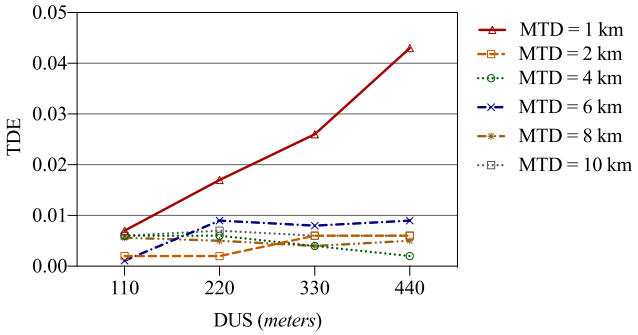

FIGURE 21. Comparison of varying *DUS* on *Ye.-PHO* dataset.

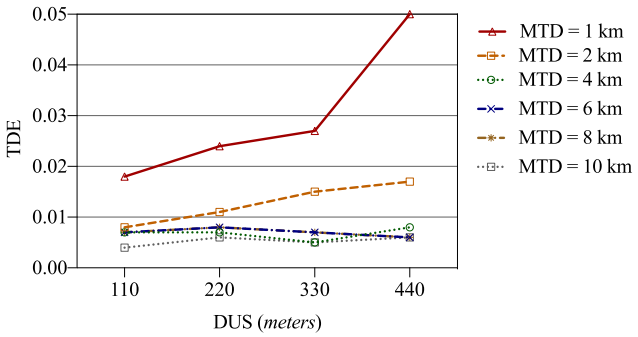

FIGURE 22. Comparison of varying *DUS* on *Fo.-NYC* dataset.

TABLE 4. The average relative performance in percentage when increasing the travel distance in *Ye.-PHO*.

MTD	1 km	2 km	4 km	6 km	8 km	10 km
ASR	86.59%	88.10%	88.51%	88.66%	88.68%	88.69%
ANW	14.23%	5.35%	1.73%	0.85%	0.51%	0.35%
WTD	63.68%	63.90%	64.68%	64.72%	66.00%	65.50%

The average of *ASR* increases when the *MTD* increase, because the more distance crowd workers are able to travel, the more chance far away tasks are assigned to them. The less *MTD* incurs higher system overhead since crowd workers are not accepting far away tasks as distinguished; Therefore, the system overhead reduces when increasing the *MTD*. This reduction is due to the average number of crowd workers who accept tasks increases. The average *WTD* increases slightly, because the average number of farther tasks is more than the nearby tasks to crowd workers.

Table 5 shows the average summaries of varying *MTD* in *Fo.-NYC*. The tasks in this dataset are located closer to crowd workers. This indicates the effectiveness of the *ASR* constants with higher *MTD*. Due to the same reason, the average system overhead in *Fo.-NYC* dataset is lower than the average system overhead in *Ye.-PHO*. Note that, the *WTD* is reducing when the *MTD* increases comparing to the *WTD* in *Ye.-PHO*. This decrease is because the average number of crowd workers who are assigned to nearby tasks is more than those who are assigned farther tasks.

TABLE 5. The average relative performance in percentage when increasing the travel distance in *Fo.-NYC*.

MTD	1 km	2 km	4 km	6 km	8 km	10 km
ASR	85.80%	86.87%	87.03%	87.03%	87.03%	87.03%
ANW	5.36%	1.86%	0.79%	0.51%	0.40%	0.33%
WTD	63.36%	63.50%	62.49%	60.65%	58.05%	56.36%

E. EFFECTS OF VARYING DUS

The performance was evaluated on the *Ye.-PHO* and *Fo.-NYC* datasets by varying the *DUS*. Figures 21 and Figure 22 show the results of varying the *DUS*. The *TDE* exists only when the crowd workers are close to tasks when injecting more privacy. However, DCentroid performs better when increasing the travel distance of the crowd workers even when injecting more privacy.

X. CONCLUSION

Spatial crowdsourcing is growing as a modern framework that facilitates workers to perform tasks in the physical world. With spatial crowdsourcing, requesters submit their spatio-temporal tasks (tasks associated with location and time) to the spatial crowdsourcing server, to be performed by a set of crowd workers who have to physically travel to the tasks' locations for execution. However, current solutions require crowd workers to disclose their exact locations to the spatial crowdsourcing server (untrusted entities).

This paper has reviewed potential location privacy attacks from the adversaries perspective, and presented a counter-measure approach to overcome the incidence of such attacks. It designed and implemented a novel privacy-preserving scheme for spatial crowdsourcing called (DCentroid), which facilitates crowd workers to participate in performing spatial crowdsourcing tasks without disclosing their locations privacy to the server. It developed a dummy generation technique that generates effective dummy locations using a specified algorithm to constrains all the possible dummies as a conceivable solution to hide the exact crowd workers' locations from adversaries. The experimental results on real-world datasets demonstrated that the introduced scheme is effective and practical. Moreover, the results of the tasks

assignment metrics shows that the the privacy trade-off is rational.

The following are future directions that can be pursued to extend and improve the DCentroid framework:

- Matching the *DUS* to the crowd workers' locations, such as minimum *DUS* to dense areas and longer to sparse areas.
- Enabling the crowd workers to choose the *DUS* based on their desired maximum travel distances.
- Enabling the crowd workers to choose the privacy radius based on their locations privacy concerns.
- Including the tasks' locations to the designed framework to insure their location's privacy.

REFERENCES

[1] J. Howe, "The rise of crowdsourcing," *Wired Mag.*, vol. 14, no. 6, pp. 1–4, Jun. 2006.

[2] J. Howe. (2006). *Crowdsourcing: A Definition*. Accessed: Oct. 20, 2018. [Online]. Available: http://www.crowdsourcing.com/cs/2006/06/crowdsourcing-a.html

[3] *Amazon Mechanical Turk*. Accessed: Nov. 16, 2017. [Online]. Available: https://www.mturk.com/

[4] G. Ghinita, "Privacy for location-based services," *Synth. Lectures Inf. Secur., Privacy, Trust*, vol. 4, no. 1, pp. 1–85, 2013.

[5] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, p. 779, 2008.

[6] L. Pournajaf, D. A. Garcia-Ulloa, L. Xiong, and V. Sunderam, "Participant privacy in mobile crowd sensing task management: A survey of methods and challenges," *ACM SIGMOD Rec.*, vol. 44, no. 4, pp. 23–34, May 2016.

[7] R. Alharthi, A. Banihani, A. Alzahrani, A. Alshehri, H. Alshahrani, H. Fu, A. Liu, and Y. Zhu, "Location privacy challenges in spatial crowdsourcing," in *Proc. IEEE Int. Conf. Electro/Inf. Technol. (EIT)*, May 2018, pp. 0564–0569.

[8] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proc. 1st Int. Conf. Mobile Syst., Appl. Services (MobiSys)*. New York, NY, USA: ACM, 2003, pp. 31–42.

[9] U. U. Hassan and E. Curry, "A multi-armed bandit approach to online spatial task assignment," in *Proc. IEEE 11th Int. Conf. Ubiquitous Intell. Comput. IEEE 11th Int. Conf. Autonomic Trusted Comput. IEEE 14th Intl Conf Scalable Comput. Commun. Its Associated Workshops*, Dec. 2014, pp. 212–219.

[10] L. Kazemi and C. Shahabi, "GeoCrowd: Enabling query answering with spatial crowdsourcing," in *Proc. 20th Int. Conf. Adv. Geographic Inf. Syst. (SIGSPATIAL)*. New York, NY, USA: ACM, 2012, pp. 189–198.

[11] Fieldagent. (2010). *Fieldagent*. Accessed: Aug. 10, 2019. [Online]. Available: https://app.fieldagent.net/

[12] PulsePoint. (2010). *Pulsepoint*. Accessed: Aug. 12, 2019. [Online]. Available: http://www.pulsepoint.org/

[13] J. Krumm, "Realistic driving trips for location privacy," in *Proc. Int. Conf. Pervas. Comput.* Berlin, Germany: Springer, 2009, pp. 25–41.

[14] J. Hu, L. Huang, L. Li, M. Qi, and W. Yang, "Protecting location privacy in spatial crowdsourcing," in *Proc. Asia–Pacific Web Conf.* Guangzhou, China: Springer, 2015, pp. 113–124.

[15] B. Liu, L. Chen, X. Zhu, Y. Zhang, and C. Zhang, "Protecting location privacy in spatial crowdsourcing using encrypted data," in *Proc. 20th Int. Conf. Extending Database Technol. (EDBT)*, 2017, pp. 1–4.

[16] L. Pournajaf, L. Xiong, V. Sunderam, and S. Goryczka, "Spatial task assignment for crowd sensing with cloaked locations," in *Proc. IEEE 15th Int. Conf. Mobile Data Manage.*, vol. 1, Jul. 2014, pp. 73–82.

[17] Y. Shen, L. Huang, L. Li, X. Lu, S. Wang, and W. Yang, "Towards preserving worker location privacy in spatial crowdsourcing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[18] H. To, G. Ghinita, L. Fan, and C. Shahabi, "Differentially private location protection for worker datasets in spatial crowdsourcing," *IEEE Trans. Mobile Comput.*, vol. 16, no. 4, pp. 934–949, Apr. 2017.

[19] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proc. VLDB Endowment*, vol. 7, no. 10, pp. 919–930, Jun. 2014.

[20] D. Yuan, Q. Li, G. Li, Q. Wang, and K. Ren, "PriRadar: A privacy-preserving framework for spatial crowdsourcing," *IEEE Trans. Inf. Forensics Security*, vol. 15, no. 1, pp. 299–314, Apr. 2020.

[21] Y. Gong, C. Zhang, Y. Fang, and J. Sun, "Protecting location privacy for task allocation in ad hoc mobile cloud computing," *IEEE Trans. Emerg. Topics Comput.*, vol. 6, no. 1, pp. 110–121, Jan. 2018.

[22] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," in *Proc. IEEE Int. Conf. Pervas. Comput. Commun. Workshops (PERCOM WORKSHOPS)*, Mar. 2014, pp. 593–598.

[23] S. S. Kanhere, "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces," in *Proc. IEEE 12th Int. Conf. Mobile Data Manage.*, vol. 2, Jun. 2011, pp. 3–6.

[24] S. S. Kanhere, "Participatory sensing: Crowdsourcing data from mobile smartphones in urban spaces," in *Proc. Int. Conf. Distrib. Comput. Internet Technol.* Berlin, Germany: Springer, 2013, pp. 19–26.

[25] B. Zhang, C. H. Liu, J. Lu, Z. Song, Z. Ren, J. Ma, and W. Wang, "Privacy-preserving QoI-aware participant coordination for mobile crowdsourcing," *Comput. Netw.*, vol. 101, pp. 29–41, Jun. 2016.

[26] B. Zhu, S. Zhu, X. Liu, Y. Zhong, and H. Wu, "A novel location privacy preserving scheme for spatial crowdsourcing," in *Proc. 6th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jun. 2016, pp. 34–37.

[27] Y. Xiao and L. Xiong, "Protecting locations with differential privacy under temporal correlations," in *Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: ACM, 2015, pp. 1298–1309.

[28] X. Yi, F.-Y. Rao, G. Ghinita, and E. Bertino, "Privacy-preserving spatial crowdsourcing based on anonymous credentials," in *Proc. 19th IEEE Int. Conf. Mobile Data Manage. (MDM)*, Jun. 2018, pp. 187–196.

[29] Y. Wang, Z. Cai, Z. Chi, X. Tong, and L. Li, "A differentially k-anonymity-based location privacy-preserving for mobile crowdsourcing systems," *Procedia Comput. Sci.*, vol. 129, pp. 28–34, 2018.

[30] H. Lu, C. S. Jensen, and M. L. Yiu, "PAD: Privacy-area aware, dummy-based location privacy in mobile services," in *Proc. 7th ACM Int. Workshop Data Eng. for Wireless Mobile Access (MobiDE)*. New York, NY, USA: ACM, 2008, pp. 16–23.

[31] R. Alharthi, E. Aloufi, A. Alqazzaz, I. Alrashdi, and M. Zohdy, "DCentroid: Location privacy-preserving scheme in spatial crowdsourcing," in *Proc. IEEE 9th Annu. Comput. Commun. Workshop Conf. (CCWC)*, Jan. 2019, pp. 0715–0720.

[32] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proc. Int. Conf. Pervas. Services (ICPS)*, 2005, pp. 88–97.

[33] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: Optimal strategy against localization attacks," in *Proc. ACM Conf. Comput. Commun. Secur. (CCS)*. New York, NY, USA: ACM, 2012, pp. 617–627.

[34] Yelp. *Yelp Dataset Challenge*. Accessed: May 12, 2018. [Online]. Available: https://www.yelp.com/dataset/challenge

[35] D. Yang, D. Zhang, V. W. Zheng, and Z. Yu, "Modeling user activity preference by leveraging user spatial temporal characteristics in LBSNs," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 45, no. 1, pp. 129–142, Jan. 2015.

RAED S. ALHARTHI (Member, IEEE) received the B.S. degree in computer science from Taif University, Taif, Saudi Arabia, in 2009, the M.S. degree in computer science and information technology from Sacred Heart University, Fairfield, USA, in 2014, and the Ph.D. degree in computer science and informatics from Oakland University, MI, USA, in 2019. His research interest includes location privacy in crowdsourcing.

ESAM ALOUFI (Member, IEEE) received the bachelor's degree in management information systems from King Abdulaziz University and the master's degree in computer science and information technology from Sacred Heart University. He is currently pursuing the Ph.D. degree in computer science and informatics with Oakland University, Rochester, MI, USA. His research interests include crowdsourcing, software engineering, and the Internet of Things.

IBRAHIM ALRASHDI received the B.S. degree in computer science and information from Jouf University, Saudi Arabia, in 2009, the M.S. degree in computer science from Western Illinois University, IL, USA, in 2013, and the Ph.D. degree in computer science and informatics from Oakland University, MI, USA, in 2019. He is currently an Assistant Professor with the Department of Computer Science, College of Computer and Information Sciences, Jouf University. His research interests include the Internet of Things, cybersecurity, and artificial intelligence.

ALI ALQAZZAZ received the M.S. degree in computer science from Saint Joseph's University, in 2013, and the Ph.D. degree in computer science from Oakland University, USA, in 2019. He is currently an Assistant Professor with the University of Bisha. His research interests include network traffic analysis, information security, the IoT security, cybersecurity, data analytics, and machine learning.

MOHAMED A. ZOHDY (Senior Member, IEEE) received the B.Sc. degree in electrical engineering from Cairo University, Egypt, in 1968, and the M.Sc. and Ph.D. degrees in electrical engineering from the University of Waterloo, Canada, in 1974 and 1977, respectively. He is currently a Professor with the School of Engineering and Computer Science, Oakland University, MI, USA. His research interest includes control and soft computing.

JULIAN L. RRUSHI (Member, IEEE) received the Ph.D. degree from the University of Milan, in 2009. He is on the faculty of the Computer Science and Engineering Department, School of Engineering and Computer Science (SECS), Oakland University, MI, USA. He works on operating systems, computer architectures, AI, and security and privacy.