

# Tradeoff Between Location Quality and Privacy in Crowdsensing: An Optimization Perspective

Yuhui Zhang, *Student Member, IEEE*, Ming Li, *Student Member, IEEE*, Dejun Yang<sup>ID</sup>, *Senior Member, IEEE*, Jian Tang<sup>ID</sup>, *Fellow, IEEE*, Guoliang Xue<sup>ID</sup>, *Fellow, IEEE*, and Jia Xu<sup>ID</sup>, *Member, IEEE*

**Abstract**—Crowdsensing enables a wide range of data collection, where the data are usually tagged with private locations. Protecting users' location privacy has been a central issue. The study of various location perturbation techniques, e.g.,  $k$ -anonymity, for location privacy has received widespread attention. Despite the huge promise and considerable attention, provable good algorithms considering the tradeoff between location privacy and location information quality from the optimization perspective in crowdsensing are lacking in the literature. In this article, we study two related optimization problems from two different perspectives. The first problem is to minimize the location quality degradation caused by the protection of users' location privacy. We present an efficient optimal algorithm OLoQ for this problem. The second problem is to maximize the number of protected users, subject to a location quality degradation constraint. To satisfy the different requirements of the platform, we consider two cases for this problem: 1) overlapping and 2) nonoverlapping perturbations. For the former case, we give an efficient optimal algorithm OPUM<sub>O</sub>. For the latter case, we first prove its NP-hardness. We then design a  $(1 - \epsilon)$ -approximation algorithm NPUM<sub>N</sub> and a fast and effective heuristic algorithm HPUM<sub>N</sub>. Extensive simulations demonstrate that OLoQ, OPUM<sub>O</sub>, and HPUM<sub>N</sub> significantly outperform an existing algorithm.

**Index Terms**—Crowdsensing, location data quality, location privacy,  $k$ -anonymity.

## I. INTRODUCTION

OVER the last decade, there has been an explosion of smart devices, e.g., smartphones and tablets. In 2015, there were available 3.2 billion smartphone subscriptions, with 6.2 billion predicted to be available in 2021 [13]. Current smart devices are embedded with increasingly powerful processors and a multitude of sensors (e.g., GPS, thermometer,

microphone, and camera). The ubiquity of mobile devices into everyday life can provide sufficient geographic coverage, especially, in densely populated areas. The mobile crowdsensing paradigm serves as a critical building block for the emerging Internet-of-Things (IoT) applications [19], [23], [24], [29], [30], [39], which takes advantage of the widely distributed mobile devices for sensing and collecting ubiquitous data, such as P-Sense to monitor air pollution [11], Nericell to sense road and traffic conditions [20], and Ear-Phone to construct urban noise maps [25]. The sensing data are usually tagged with locations to form a database or a map for information release or decision making.

It is essential to achieve location privacy protection, since mobile users' locations are tightly correlated with their identities and vulnerable to malicious attacks. Upon preserving location privacy in crowdsensing, various methods, including information caching [27], spatial cloaking [33], data perturbation with noise [40], and microaggregation [36], are proposed. The goal is to prevent the servers or platforms from inferring users' actual locations. However, these privacy-preserving methods need to hide the users' actual locations, which usually degrade the location (information) quality [6].

Location privacy and location quality are two conflicting concerns in crowdsensing. On the one hand, disclosing users' actual locations to the platform may severely discourage their participation, because users are increasingly wary of location privacy. On the other hand, the platform desires the actual locations of users to ensure the location quality. It is necessary to strike a good balance between location privacy and location quality in crowdsensing. To quantify the impact of location privacy protection on location quality, we define the *location quality degradation* as the maximum distance between users' actual locations and their corresponding perturbed locations. Note that the summation of squared location errors (SSE) [28] has also been used to measure location quality in the literature. Although minimizing the SSE is not our objective, our simulation results demonstrate that a low location quality degradation also implies a low SSE.

In this article, we study the tradeoff between location quality and location privacy. Location quality and location privacy are two conflicting concerns in crowdsensing, which naturally leads to a duality relation. On the one hand, for the isolated users, the distances between their actual locations and perturbed locations might be very large, because they have to share the same perturbed locations with the other users to protect their location privacy. Thus, if all users' location

Manuscript received September 27, 2019; revised December 18, 2019 and January 23, 2020; accepted January 26, 2020. Date of publication February 10, 2020; date of current version April 14, 2020. This work was supported in part by NSF under Grant 1525920, Grant 1704092, Grant 1717197, and Grant 1717315. This article was presented in part at 2019 IEEE GLOBECOM. (Corresponding author: Dejun Yang.)

Yuhui Zhang, Ming Li, and Dejun Yang are with the Department of Computer Science, Colorado School of Mines, Golden, CO 80401 USA (e-mail: yuhzhang@mines.edu; mili@mines.edu; djyang@mines.edu).

Jian Tang is with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY 13244 USA (e-mail: jtang02@syr.edu).

Guoliang Xue is with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85287 USA (e-mail: xue@asu.edu).

Jia Xu is with the Jiangsu Key Laboratory of Big Data Security and Intelligent Processing, Nanjing University of Posts and Telecommunications, Nanjing 210023, China (e-mail: xujia@njupt.edu.cn).

Digital Object Identifier 10.1109/IIOT.2020.2972555

privacy must be protected, this may cause a large location quality degradation. On the other hand, if the location quality degradation is a constraint, it might not be possible to find perturbed locations for the isolated users to protect their location privacy, because they are too far from the others. Depending on the preference of the crowdsensing platform, we consider the optimization from two perspectives. If it desires to protect all users' location privacy, the problem can be formulated as the location quality degradation minimization (LQDM) problem: minimizing the location quality degradation, while guaranteeing the location privacy for all users; if it desires to bound the location quality degradation, the problem can be formulated as the protected user maximization with location quality degradation constraint (PUM) problem: maximizing the number of users whose location privacy is protected, subject to a location quality degradation constraint. For the second problem, we consider overlapping and nonoverlapping cases to satisfy various requirements of the platform. The difference is whether one user is allowed to be tagged with more than one perturbed location. The rationale behind the overlapping case is that the sensed data at one location can well represent the results at nearby locations in many crowdsensing applications, e.g., noise, temperature, and signal coverage. Note that we focus on only the overlapping case for the LQDM problem because the nonoverlapping case often results in large location quality degradation due to the constraint of including all users.

#### A. Contributions

We summarize the main contributions as follows.

- 1) To the best of our knowledge, we are the first to consider the tradeoff between location privacy and location quality in crowdsensing from an optimization perspective.
- 2) We first study the problem of optimizing the location quality in terms of the location quality degradation, while guaranteeing the location privacy for all users. We design an efficient optimal algorithm to minimize the location quality degradation among all users.
- 3) We then investigate the problem of maximizing the number of users whose location privacy is protected, while guaranteeing the location quality with a location quality degradation bound. Specifically, there are two cases depending on the platform's requirement: 1) overlapping and 2) nonoverlapping. For the former case, we design an efficient optimal algorithm. For the latter case, we prove its NP-hardness and design a near-optimal approximation algorithm and a fast and effective heuristic algorithm that achieves near-optimal performance in simulations.

The remainder of this article is organized as follows. In Section II, we give a brief review of existing location privacy-preserving mechanisms in the literature. In Section III, we formally introduce the system model and give a precise problem description. In Section IV, we present a polynomial-time optimal algorithm for LQDM and analyze its properties. In Section V, we study the PUM problem under the overlapping perturbation and nonoverlapping perturbation cases and design corresponding algorithms. Section VI demonstrates the experimental evaluations. Section VII concludes this article.

## II. RELATED WORK

### A. Location Privacy Approaches

There is a rich collection of literature on location privacy in general frameworks. Surveys for location privacy-preserving methods can be found in [4] and [9]. Following the discussions in [9], we classify location privacy-preserving techniques into three types: 1) location generation [15], [38]; 2) cryptographic techniques [19]; and 3) differential privacy [14]. Along the line of location generation, various methods, including position dummies [15], mix zone [2], pseudonym [10], and  $k$ -anonymity [38], are proposed.

Much effort has also been made to protect location privacy in crowdsensing systems [1], [8], [14], [18], [34], [35]. This line of work aims at preventing location privacy leakage from sensing reports submitted by crowdsensing users. Gao *et al.* [8] designed a partner selection algorithm and construct several trajectories that are closer to the users. Agir *et al.* [1] proposed a scheme which estimated the expected location-privacy level at the user side locally in real time, which satisfies each user's privacy requirement adaptively. Vu *et al.* [34] utilized a Voronoi diagram to partition a space into cells that contain at least  $k$  users in each, without considering to minimize the cloaking area. Differential location privacy in the crowdsourced spectrum sensing was preserved in [14], [18], and [35]. However, a significant problem neglected in these works is to optimize the crowdsensing platform's location quality, while protecting the users' location privacy.

### B. Location Information Quality

As pointed out by Krause *et al.* [17], it is challenging to balance between location privacy and location quality. Rodhe *et al.* [26] considered two strategies based on different types of system servers to reconstruct the data distribution and investigated the impact of location privacy-preserving mechanisms on the quality of information. Xiao *et al.* [37] developed a directed-graph-based cloaking algorithm for protecting location privacy in location-based service while meeting user-specified quality-of-service requirements. Murshed *et al.* [21] proposed a subset-coding scheme to achieve almost lossless data integrity.

Another related topic is the microaggregation problem: divide a set of data into several disjoint subsets, such that the size of each subset is more than  $k$  and the sum of squared error is minimized. This problem has been studied to strike a balance between privacy protection and information loss reduction [5], [16], [28]. However, the LQDM is not considered in this problem.

## III. SYSTEM MODEL AND PROBLEM FORMULATION

In this section, we introduce the system model and give a precise problem description.

### A. System Model

We consider a location-based crowdsensing system consisting of a set  $\mathcal{U} = \{1, 2, \dots, n\}$  of  $n$  users, a trusted third party [32], [36] (e.g., a cellular service provider), and a

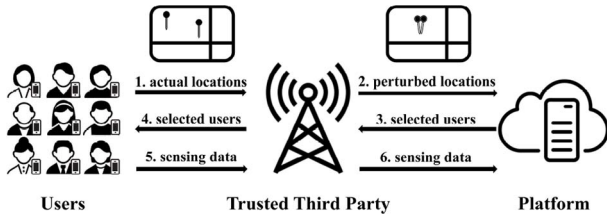


Fig. 1. Location privacy-preserving crowdsensing system.

crowdsensing platform. Each user carries an advanced mobile device with sensing capabilities and wishes to earn rewards by completing crowdsensing tasks. The user registers with the platform and communicates with the platform via an app installed on his mobile device. Developed by the platform, the app is assumed to pass the strict vetting process of the trusted app store and has no unauthorized access to the user's locations.

We assume that the platform is honest but curious, which is commonly used to characterize a reasonable crowdsensing platform. Particularly, the platform is trusted to faithfully follow the protocol but also interested in learning users' locations. We assume that the platform can have arbitrary prior knowledge for attempting to breach the users' location privacy.

The precision-aware location privacy-preserving crowdsensing system is shown in Fig. 1. The platform publishes crowdsensing tasks and collects location-aware sensing data from the users. The trusted third party, which is a cellular service supposed to protect location privacy. The workflow of the system is as follows.

- 1) All the users report their actual locations  $\mathcal{L} = \{l_1, l_2, \dots, l_n\}$  to the trusted third party for location privacy protection.
- 2) The trusted third party processes the actual locations and reports a set of perturbed locations  $\{h_1, h_2, \dots, h_m\}$  to the platform, where a perturbed location  $h_j$  is tagged to at least  $k$  users.
- 3) The users tagged with perturbed locations are reported to the platform, and the rest users are discarded.

### B. Problem Formulation

To formally formulate our studied problems, we introduce the following necessary concepts. In order to preserve location privacy, one solution is to make a user's location indistinguishable from at least  $k - 1$  others' locations. This property is proposed in [31] and is called *k-anonymity*.

*k-Anonymity*: To protect user's privacy, it requires that at least  $k$  reports are combined together before releasing.

*Location Perturbation*: Location perturbation is defined as deliberately degrading the quality of location information about a user's location in order to protect that user's location privacy. A similar definition has been proposed as obfuscation [6]. However, the mechanism in [6] cannot be directly applied to crowdsensing. We have discussed the difference between LBS and crowdsensing in Section II-A.

*Location Quality Degradation*: The location quality degradation is the maximum of a set of distances between users' actual locations and their corresponding perturbed locations.

*Perturbed Group*: A perturbed group is a set of users  $\mathcal{S} \subseteq \mathcal{U}$  tagged with the same perturbed location, denoted by  $(h, \mathcal{S})$ , satisfying *k-anonymity*.

Apparently, the perturbation operation for protecting users' location privacy causes inevitable location errors, which can diminish the quality of the crowdsensing results. Thus, it is necessary to strike a good balance between location privacy and location quality in crowdsensing. Therefore, it is essential to control location quality degradation while preserving users' location privacy. Toward this goal, we consider the following two related optimization problems from two different perspectives in this article.

- 1) *LQDM*: Given a set of  $n$  users' actual locations and an integer  $k \leq n$ , form a set of perturbed groups, denoted by  $\mathcal{H}$ , including all users to minimize the location quality degradation.
- 2) *PUM*: Given a set of  $n$  users' actual locations, an integer  $k \leq n$ , and a location quality degradation bound  $\delta$ , form a set of perturbed groups, denoted by  $\mathcal{H}$ , to include a maximum number of users, such that the location quality degradation is no more than  $\delta$ .

Depending on the platform's requirement, we consider two cases: 1) *overlapping perturbation*, where one user is allowed to be tagged with more than one perturbed location and 2) *nonoverlapping perturbation*, where one user is tagged with at most one perturbed location. The rationale behind the overlapping case is that the sensed data at one location can well represent the results at nearby locations in many crowdsensing applications, e.g., temperature and signal coverage. We denote the PUM problem under these two cases by  $\text{PUM}_O$  and  $\text{PUM}_N$ , respectively. Note that we focus on only the overlapping case for the LQDM problem because the nonoverlapping case often results in large location quality degradation due to the constraint of including all users.

Note that in the literature, the SSE [28] has been used to measure data quality. In this article, we use the location quality degradation. Some large errors are still detrimental to the crowdsensing application with a small SSE, whereas a small location quality degradation guarantees that none of the errors exceeds this value. Although we do not focus on minimizing the SSE, extensive simulations show that our algorithm achieves a lower SSE, compared to an existing *k-anonymity* location privacy-preserving algorithm.

### C. Geometric Problem Transformation

Both LQDM and PUM problems can be transformed into equivalent geometric problems. Before the transformation, we introduce the following definition.

Let  $\mathcal{P}$  denote a plane. For any two points  $p \in \mathcal{P}$  and  $q \in \mathcal{P}$ , we use  $\|p, q\|$  to denote the Euclidean distance between  $p$  and  $q$ . A disk centered at  $c$  of radius  $r$  is denoted by  $D(c, r)$ . We say  $D(c, r)$  covers  $p$ , if  $p \in D(c, r)$ , i.e.,  $\|p, c\| \leq r$ . Let  $B(c, r)$  denote the closed boundary of  $D(c, r)$ . Given a set  $\mathcal{L}$  of  $n$  points, let  $\mathcal{D}(\mathcal{L}, r)$  denote a set of disks of radius  $r$  centered at points in  $\mathcal{L}$ .

*Definition 1 (k-Enclosing Disk)*: Let  $\mathcal{L}$  be a set of  $n$  points on the plane  $\mathcal{P}$ . Given an integer  $k \leq n$ , a *k-enclosing disk* is a disk that covers at least  $k$  points in  $\mathcal{L}$ .

The transformed LQDM and PUM problems are as follows.

- 1) *LQDM*: Given a set  $\mathcal{L}$  of  $n$  points on the plane  $\mathcal{P}$  and an integer  $k \leq n$ , find a minimum  $r$  and a set of  $k$ -enclosing disks  $\mathcal{D} = \{D(h_1, r), D(h_2, r), \dots\}$ , such that any  $l_i \in \mathcal{L}$  is covered by at least one disk in  $\mathcal{D}$ .
- 2) *PUM*: Given a set  $\mathcal{L}$  of  $n$  points on the plane  $\mathcal{P}$ , an integer  $k \leq n$  and a constant  $\delta$ , find a set of  $k$ -enclosing disks  $\mathcal{D} = \{D(h_1, \delta), D(h_2, \delta), \dots\}$ , such that a maximum number of points in  $\mathcal{L}$  are covered by disks in  $\mathcal{D}$ .

To solve these problems, we need the following definitions and claims from [7].

**Definition 2 (Depth of a Point)**: Given a point  $p \in \mathcal{P}$  and a disk set  $\mathcal{D}(\mathcal{L}, r)$ , the depth of  $p$  with respect to  $\mathcal{D}(\mathcal{L}, r)$ , denoted by  $d_{\mathcal{D}(\mathcal{L}, r)}(p)$ , is the number of disks in  $\mathcal{D}(\mathcal{L}, r)$  covering  $p$ .

**Definition 3 (Depth of a Disk)**: Given a point  $l_i \in \mathcal{L}$  and a disk set  $\mathcal{D}(\mathcal{L}, r)$ , the depth of  $D(l_i, r)$ , denoted by  $d_{D(l_i, r)}$ , is the maximum depth of all points  $p \in D(l_i, r)$

$$d_{D(l_i, r)} = \max_{p \in D(l_i, r)} \{d_{\mathcal{D}(\mathcal{L}, r)}(p)\}.$$

**Claim 1**: Given two points  $p, q \in \mathcal{P}$ ,  $p \in D(q, r)$  if and only if  $q \in D(p, r)$ .

**Claim 2**: The depth of a point  $p \in \mathcal{P}$  in  $\mathcal{D}(\mathcal{L}, r)$  is the number of points in  $\mathcal{L}$  covered by  $D(p, r)$ .

**Definition 4 (Critical Radius)**: Given any  $l_i \in \mathcal{L}$ , a radius  $r$  is a critical radius, if  $d_{D(l_i, r)}$  decreases, when  $r$  is decreased by an arbitrarily small amount.

Finally, we have three geometrical facts as follows.

- 1) The point on  $D(l_i, r)$  with maximum depth must be an intersection point on  $B(l_i, r)$ , if  $B(l_i, r)$  intersects with the boundary of any other disk in  $\mathcal{D}(\mathcal{L}, r)$ . Then, we only focus on the intersection points on  $B(l_i, r)$  for computing  $d_{D(l_i, r)}$ .
- 2) Given any  $l_i \in \mathcal{L}$ , let  $r_i^*$  denote the minimum radius  $r$ , such that  $d_{D(l_i, r)} \geq k$ . We can locate  $r_i^*$  within a feasible range of  $r$  using the following criteria:
  - a)  $d_{D(l_i, r)} < k \rightarrow r < r_i^*$ ;
  - b)  $d_{D(l_i, r)} > k \rightarrow r > r_i^*$ ;
  - c)  $d_{D(l_i, r)} = k \rightarrow r \geq r_i^*$ .
- 3) A radius  $r$  can be a critical radius only if  $B(l_i, r)$  is tangent to  $B(l_j, r)$ , or  $B(l_i, r)$  is concurrent with  $B(l_j, r)$  and  $B(l_k, r)$ , where  $l_i, l_j, l_k \in \mathcal{L}$ . In other words, a critical radius is either  $1/2||l_i, l_j||$ , denoted by  $r_{ij}$ , or a circum-radius of a triangle with  $l_i, l_j$ , and  $l_k$  as the vertices, denoted by  $r_{ijk}$ .

The main notations are summarized in Table I.

#### IV. OPTIMAL ALGORITHM FOR LQDM

In this section, we present an efficient optimal algorithm OLoQ for the LQDM problem.

##### A. Overview

Since  $r_i^*$  is the minimum radius, such that  $l_i$  is covered by a  $k$ -enclosing disk, the minimum radius in the optimal solution to the LQDM problem equals  $\max_{l_i \in \mathcal{L}} r_i^*$ , denoted by  $r^*$ . Thus, the LQDM problem boils down to finding  $r_i^*$  for each  $l_i \in \mathcal{L}$ . Based on Fact 2) in Section III-C, it is necessary to determine

TABLE I  
MAIN NOTATIONS

Notation	Meaning
$\mathcal{U}$	a set of users $\{1, 2, \dots, n\}$
$(h, S)$	a perturbed group, where all users in $S$ are tagged with $h$
$\mathcal{H}$	a set of perturbed groups
$\mathcal{P}$	a plane
$l_i$	the actual location (a point on $\mathcal{P}$ ) of user $i$
$\mathcal{L}$	the set of actual locations (points on $\mathcal{P}$ ) of users in $\mathcal{U}$
$D(p, r)$	the disk of radius $r$ , centered at $p \in \mathcal{P}$
$\mathcal{D}(\mathcal{L}, r)$	the set of disks with radius $r$ , centered at points in $\mathcal{L}$
$d_{\mathcal{D}(\mathcal{L}, r)}(p)$	the depth of a point $p$ with respect to $\mathcal{D}(\mathcal{L}, r)$
$d_{D(l_i, r)}$	the depth of a disk $D(l_i, r)$
$B(p, r)$	the closed boundary of $D(p, r)$
$r_i^*$	the minimum radius of a $k$ -enclosing disk to cover $l_i$
$p_i^*$	the center of the smallest $k$ -enclosing disk to cover $l_i$

a range in order to locate  $r_i^*$ . To locate the exact value of  $r_i^*$ , we need to discretize its range. By the definition of critical radius,  $r_i^*$  must be a critical radius of  $l_i$ . Thus, we focus on critical radii and conduct a binary search among them for locating  $r_i^*$ . According to Fact 3), a critical radius of  $l_i$  can only be  $r_{ij}$  or  $r_{ijk}$ , where  $l_j, l_k \in \mathcal{L}, i \neq j \neq k$ . Once  $r_i^*$  is located, we find the point of maximum depth on  $D(l_i, r_i^*)$ , denoted by  $p_i^*$ . Then, a set of  $k$ -enclosing disks  $\mathcal{D} = \{D(p_i^*, r_i^*) \mid l_i \in \mathcal{L}\}$  can cover all  $l_i \in \mathcal{L}$ . However, not all disks in  $\mathcal{D}$  are necessary. Thus, we select a minimal  $\mathcal{D}^* \subseteq \mathcal{D}$  covering all points in  $\mathcal{L}$ . The centers of the selected disks are the perturbed locations.

##### B. Algorithm Design

OLoQ includes one key algorithm to find the smallest  $k$ -enclosing disk covering  $l_i \in \mathcal{L}$ , illustrated in Algorithm 1.

In Algorithm 1, we narrow the range where  $r_i^*$  can lie and locate  $r_i^*$ . To narrow the range where  $r_i^*$  can lie, we collect the  $n - 1$  values of  $r_{ij}$  and sort them in nondecreasing order. Note that each  $r_{ij}$  is corresponding to a tangent point  $p_{ij}$  of  $B(l_i, r_{ij})$  and  $B(l_j, r_{ij})$ , which is the midpoint of line  $l_i l_j$ . Then, the range can be narrowed to  $(\underline{r}_{ij}, \bar{r}_{ij}]$ .

Then we collect  $[(n - 1)(n - 2)]/2$  values of  $r_{ijk}$  and only keep the values of  $r_{ijk}$  within the range  $(\underline{r}_{ij}, \bar{r}_{ij}]$ . If there is no  $r_{ijk}$  within this range, then  $r_i^*$  is  $\bar{r}_{ij}$  and its corresponding  $p_i^*$  is  $\bar{p}_{ij}$ . Otherwise, we sort the values of  $r_{ijk}$  within the range  $(\underline{r}_{ij}, \bar{r}_{ij}]$  in a nondecreasing order. Note that each  $r_{ijk}$  is corresponding to a point  $p_{ijk}$ , which is the circumcenter of the triangle with  $l_i, l_j$ , and  $l_k$  as the vertices. Using binary search, we further restrict the range to  $(\underline{r}_{ijk}, \bar{r}_{ijk}]$ , which is the smallest range such that  $d_{D(l_i, \underline{r}_{ijk})} < k$  and  $d_{D(l_i, \bar{r}_{ijk})} \geq k$ . Therefore,  $r_i^*$  is  $\bar{r}_{ijk}$ , and  $p_i^*$  is the  $\bar{p}_{ijk}$  corresponding to  $\bar{r}_{ijk}$ .

At the end, Algorithm 1 outputs  $(r_i^*, p_i^*)$ , which forms the smallest  $k$ -enclosing disk  $D(r_i^*, p_i^*)$  that covers  $l_i$ . We shall run Algorithm 1 for each  $l_i \in \mathcal{L}$ . Then, the minimum radius in the optimal solution to the LQDM problem is  $\max_{l_i \in \mathcal{L}} r_i^*$ .

Next, we generate a set of  $k$ -enclosing disks  $\mathcal{D}^* = \{D(h_1, r^*), D(h_2, r^*), \dots\}$ , such that any  $l_i \in \mathcal{L}$  is covered by at least one disk in  $\mathcal{D}^*$ . By the previous steps, we can obtain a set of  $k$ -enclosing disks  $\mathcal{D} = \{D(p_1^*, r^*), D(p_1^*, r^*), \dots, D(p_n^*, r^*)\}$  covering all points in  $\mathcal{L}$ . However, not all of them are necessary. So we design Algorithm 2 to select a minimal  $\mathcal{D}^* \subseteq \mathcal{D}$  covering all points in  $\mathcal{L}$ . The idea is to select disks iteratively. In each iteration, we select a disk covering as many points as possible. Thus,

**Algorithm 1:** Find- $k$ -Enclosing-Disk( $l_i, k, \mathcal{L}$ )

---

```

1 Sort all values in  $\{r_{ij} \mid l_j \in \mathcal{L} \setminus \{l_i\}\}$  in a nondecreasing
  order and obtain a sorted list  $R_{ij}$ ;
2 Run a binary search in  $R_{ij}$  to find two consecutive values
  of  $r_{ij}$ , denoted by  $\underline{r}_{ij}$  and  $\bar{r}_{ij}$ , such that  $d_{D(l_i, \underline{r}_{ij})} < k$  and
   $d_{D(l_i, \bar{r}_{ij})} \geq k$ ;
3 Sort all values in  $\{r_{ijk} \mid r_{ijk} \in (\underline{r}_{ij}, \bar{r}_{ij}], l_j, l_k \in \mathcal{L} \setminus \{l_i\}\}$  in a
  nondecreasing order and obtain a sorted list  $R_{ijk}$ ;
4 if  $R_{ijk} = \emptyset$  then
5    $r_i^* \leftarrow \bar{r}_{ij}$ ;  $p_i^* \leftarrow \bar{p}_{ij}$ ;
6 else
7   Run a binary search in  $R_{ijk}$  to find two consecutive
    values of  $r_{ijk}$ , denoted by  $\underline{r}_{ijk}$  and  $\bar{r}_{ijk}$ , such that
     $d_{D(l_i, \underline{r}_{ijk})} < k$  and  $d_{D(l_i, \bar{r}_{ijk})} \geq k$ ;
8    $r_i^* \leftarrow \bar{r}_{ijk}$ ;  $p_i^* \leftarrow \bar{p}_{ijk}$ ;
9 return  $(r_i^*, p_i^*)$ 

```

---

**Algorithm 2:** OLoQ ( $\mathcal{L}, k$ )

---

```

1  $\mathcal{H} \leftarrow \emptyset$ ;  $\mathcal{D}^* \leftarrow \emptyset$ ;
2 for  $l_i \in \mathcal{L}$  do
3    $r_i^* \leftarrow \text{Find-}k\text{-enclosing-Disk}(l_i, k, \mathcal{L})$ ;  $r^* \leftarrow \max_{l_i \in \mathcal{L}} r_i^*$ ;
4 Sort points in  $\mathcal{L}$  based on  $r_i^*$  in a nonincreasing order
  and obtain a sorted list  $L$ ;
5 for  $l_i \in L$  do
6   if  $l_i$  is uncovered by  $\mathcal{D}^*$  then
7      $\mathcal{D}^* \leftarrow \mathcal{D}^* \cup \{D(p_i^*, r^*)\}$ ;
8      $\mathcal{H} \leftarrow \mathcal{H} \cup \{p_i^*, \{i \mid l_i \in D(p_i^*, r^*)\}\}$ ;
9 return  $(\mathcal{H}, r^*)$ 

```

---

we sort  $n$  values of  $r_i^*$  for all  $l_i \in \mathcal{L}$  in a nonincreasing order and select disks sequentially according to the sorted list. If  $l_i$  has not been covered, we add  $D(p_i^*, r^*)$  to  $\mathcal{D}^*$ . For all users whose actual locations are covered by  $D(p_i^*, r^*)$ , we form a perturbed group with  $p_i^*$  as their perturbed location.

The time complexity of Algorithm 1 is  $O(n^2 \log n)$ . The time complexity of Algorithm 2 is  $O(n)$ . Therefore, the time complexity of OLoQ is  $O(n^3 \log n)$ . It can be proved that OLoQ returns an optimal solution to LQDM [41].

## V. ALGORITHMS FOR PUM

In this section, we study the PUM problem under the overlapping perturbation and nonoverlapping perturbation cases, denoted by PUM<sub>O</sub> and PUM<sub>N</sub>, respectively. Then, we design corresponding algorithms for them.

A. Optimal Algorithm for PUM<sub>O</sub>

In this section, we develop an efficient optimal algorithm OPUM<sub>O</sub> for the PUM<sub>O</sub> problem.

We first introduce the intuition behind OPUM<sub>O</sub>. We know that  $r_i^*$  is the minimum radius  $r$ , such that  $l_i$  is covered by a  $k$ -enclosing disk. It implies that for any point  $l_i \in \mathcal{L}$ , if  $r_i^* > \delta$ , then  $l_i$  can never be covered in a  $k$ -enclosing disk of radius  $\delta$ . Thus, we can discard such points from  $\mathcal{L}$ . For

the remaining points in  $\mathcal{L}$ , we select a subset  $\mathcal{D}^\delta \subseteq \mathcal{D} = \{D(p_1^*, \delta), D(p_1^*, \delta), \dots, D(p_n^*, \delta)\}$  to cover all of them. The centers of the selected disks are the perturbed locations.

The details of OPUM<sub>O</sub> are described as follows. Using Algorithm 1 in Section IV-B, we can obtain a set of  $k$ -enclosing disks  $\mathcal{D} = \{D(p_1^*, \delta), D(p_1^*, \delta), \dots, D(p_n^*, \delta)\}$ . Similar to Algorithm 2, we sort the values of  $r_i^*$  for all  $l_i \in \mathcal{L}$  in a nonincreasing order and select a subset of disks  $\mathcal{D}^\delta \subseteq \mathcal{D}$  sequentially according to the sorted list. The fundamental difference from OLoQ is that for any point  $l_i \in \mathcal{L}$ , if  $r_i^* > \delta$ , we discard  $l_i$  from  $\mathcal{L}$ . Then, if  $l_i$  has not been covered and its corresponding  $r_i^* \leq \delta$ , we add  $D(p_i^*, \delta)$  to  $\mathcal{D}^\delta$ . For all users whose actual locations are covered by  $D(p_i^*, \delta)$ , we form a perturbed group  $(p_i^*, \{i \mid l_i \in D(p_i^*, \delta)\})$ . Then,  $p_i^*$  is set to be their perturbed location.

Since OPUM<sub>O</sub> is similar to OLoQ, the overall time complexity of OPUM<sub>O</sub> is  $O(n^3 \log n)$  as well. The optimality of OPUM<sub>O</sub> is guaranteed by the following theorem.

**Theorem 1:** OPUM<sub>O</sub> returns an optimal solution to PUM<sub>O</sub>.

*Proof:* We first prove that each user  $i$  included in the perturbed groups is tagged with the same perturbed location as at least  $k-1$  other users and then prove that the maximum number of users is included in the formed perturbed groups.

For each  $l_i \in \mathcal{L}$ , it guarantees that  $d_{D(l_i, r_i^*)} \geq k$ , based on lines 2 and 7 in Algorithm 1. Since  $p_i^*$  is the point with maximum depth on  $D(l_i, r_i^*)$ , we have  $d_{D(\mathcal{L}, r_i^*)}(p_i^*) \geq k$ . By Claim 2,  $D(p_i^*, r_i^*)$  covers at least  $k$  points in  $\mathcal{L}$ . By Claim 1, we have  $l_i \in D(p_i^*, r_i^*)$ . With  $r_i^* \leq \delta$ , we know that  $D(p_i^*, \delta)$  covers at least  $k$  points in  $\mathcal{L}$  and  $l_i \in D(p_i^*, \delta)$  as well. Thus, there are at least  $k$  users in each perturbed group  $(p_i^*, \{i \mid l_i \in D(p_i^*, \delta)\})$ . Therefore, each user  $i$  included in the perturbed groups is tagged with the same perturbed location with at least  $k-1$  other users.

We learned from the above proof that  $r_i^*$  is the minimum radius, such that user  $i$  is tagged with the same perturbed location as at least  $k-1$  other users. So we know that if  $r_i^* > \delta$ , then user  $i$  can never be included in a perturbed group. Because the users with  $r_i^* > \delta$  are discarded, and the users with  $r_i^* \leq \delta$  are included in the perturbed groups, we know that the maximum number of users are included in the formed perturbed groups. ■

B. Algorithms for PUM<sub>N</sub>

In this section, we design a near-optimal approximation algorithm NPUM<sub>N</sub> and an effective heuristic algorithm HPUM<sub>N</sub>. The PUM<sub>N</sub> problem can be proved to be NP-hard [41] by a reduction from the disjoint unit-disk cover problem, which has been proved to be NP-hard in [12]. Therefore, the PUM<sub>N</sub> problem is unlikely to have an efficient optimal algorithm unless  $P = NP$ . Thus, we design an approximation algorithm NPUM<sub>N</sub> by applying the shifted grid technique. The shifting technique has two stages. In the first stage, the plane is partitioned into squares with each having a size of  $s \times s$ . By shifting the partition grid lines over unit distance, a new way of partitioning can be derived. We call each way of partitioning a “shift.” Thus, there are  $s \times s$  shifts in total.

**Algorithm 3:** NPUM<sub>N</sub>( $\mathcal{L}, k, \delta, \epsilon$ )

---

```

1 Normalize the plane with respect to  $\delta$ ;
2  $s \leftarrow 2/\epsilon$ ;
3 in parallel for  $i \leftarrow 1$  to  $s$  and  $j \leftarrow 1$  to  $s$  do
4   Partition the plane into squares with each having a
   size of  $s \times s$ , by drawing grid lines  $x = i + as$  and
    $y = j + bs$ , where  $a, b \in \mathbb{Z}$ ;
5   in parallel for each square do
6     Construct  $s(s-1)$  unit disks s.t. at least 2 points
     in  $\mathcal{L}$  are on the boundary;
7     Use a brute-force algorithm to select an optimal
     disk set for this square, which contains at most
      $2s^2$  disks;
8   Combine the selected disk sets as the global solution;
9 Pick the solution  $\mathcal{D}^\delta$  that includes the maximum number
  of points among;
10 Construct  $\mathcal{H}$  based on disks in  $\mathcal{D}^\delta$ ;
11 return  $\mathcal{H}$ 

```

---

The second stage is to use a local algorithm to find an optimal solution within each square. Then, the union of all squares' solutions is the global solution for a shift. The final solution is the one with the best performance among all shifts. A brute-force algorithm can find the optimal solution within  $s \times s$  square in exponential time. Since  $s \times s$  square can be covered with  $2s^2$  unit disks compactly, there are  $O(8/\epsilon^2)$  disks in the optimal solution to cover  $n_p$  points inside the square. Since we can always move a disk in the optimal solution, such that at least two points are on its boundary, there are  $O(n_p^2)$  disk positions. To select  $O(8/\epsilon^2)$  from  $O(n_p^2)$  disks, we check  $O(n_p^{16/\epsilon^2})$  disk arrangements. In each arrangement, we check  $n_p$  points' positions in  $O((8/\epsilon^2)^{n_p})$  time. Therefore, the overall time complexity of the local algorithm is  $O(8^{n_p}/\epsilon^{2n_p}n_p^{16/\epsilon^2})$ . For the shifted grid technique, we only discuss the local algorithm's complexity, because the local algorithm (lines 6 and 7 in Algorithm 3) can be run in parallel in multiple squares.

**Theorem 2:** The approximation ratio of NPUM<sub>N</sub> is  $1 - \epsilon$ , where  $\epsilon > 0$  is an arbitrarily small constant.

*Proof:* Let  $OPT$  be the set of points in an optimal solution,  $OPT_{(i,j)}$  be the set of points in the global solution of shift  $(i, j)$ , and  $OPT'_{(i,j)}$  be the set of points in  $OPT$  intersecting the lines  $x = as + i$  and  $y = bs + j$ , where  $a, b \in \{0, 1, 2, \dots\}$ . Then, we have  $|OPT_{(i,j)}| + |OPT'_{(i,j)}| \geq |OPT|$ , and thus

$$\sum_{i=1}^s \sum_{j=1}^s (|OPT_{(i,j)}| + |OPT'_{(i,j)}|) \geq s^2 |OPT|$$

because  $\sum_{i=1}^s \sum_{j=1}^s |OPT'_{(i,j)}| \leq 2s|OPT|$ , we have  $\sum_{i=1}^s \sum_{j=1}^s |OPT_{(i,j)}| \geq (s^2 - 2s)|OPT|$ , and thus

$$\max_{i,j \in \{1, \dots, s-1\}} |OPT_{(i,j)}| \geq \left(1 - \frac{2}{s}\right) |OPT| = (1 - \epsilon) |OPT|.$$

Even though the time complexity is exponential to the maximum number of points in any square, the number of users in

**Algorithm 4:** HPUM<sub>N</sub>( $\mathcal{L}, k, \delta$ )

---

```

1  $\mathcal{H} \leftarrow \emptyset$ ;  $\mathcal{D}^\delta \leftarrow \emptyset$ ;
2 while  $|\mathcal{L}| \geq k$  do
3   for  $l_i \in \mathcal{L}$  do
4     Update  $d_{D(l_i, \delta)}$ ;
5     if  $d_{D(l_i, \delta)} < k$  then  $\mathcal{L} \leftarrow \mathcal{L} \setminus \{l_i\}$ ;
6   if  $\max_{l_i \in \mathcal{L}} d_{D(l_i, \delta)} < k$  then break;
7    $l_{\min} \leftarrow \arg \min_{l_i \in \mathcal{L}} d_{D(l_i, \delta)}$ ;
8   Find the point  $p_{\min}^* \in D(l_{\min}, \delta)$  of maximum depth;
9    $\mathcal{D}^\delta \leftarrow \mathcal{D}^\delta \cup \{D(p_{\min}^*, \delta)\}$ ;
10   $\mathcal{H} \leftarrow \mathcal{H} \cup \{(p_{\min}^*, \{i \mid l_i \in D(p_{\min}^*, \delta) \cap \mathcal{L}\})\}$ ;
11   $\mathcal{L} \leftarrow \mathcal{L} \setminus \{l_i \mid l_i \in D(p_{\min}^*, \delta)\}$ ;
12 return  $\mathcal{H}$ 

```

---

TABLE II  
MAXIMUM NUMBER OF USERS IN A SQUARE

Dataset	$1.5 \times 1.5 \text{ km}^2$	$2 \times 2 \text{ km}^2$	$2.5 \times 2.5 \text{ km}^2$
roma taxi [3]	17	22	30
SF taxi [22]	23	29	34

each square is small in practice. We conduct studies on the roma taxi data set [3] and the San Francisco taxi dataset [22]. Table II shows the maximum number of users in a square.

To further reduce the time complexity, we design a fast and effective heuristic algorithm HPUM<sub>N</sub> that can achieve near-optimal performance in practice, although not theoretically.

The details of HPUM<sub>N</sub> are as follows. When  $\mathcal{L}$  contains more than  $k$  points, we compute the disk depth  $d_{D(l_i, \delta)}$  for each  $l_i \in \mathcal{L}$ . If  $d_{D(l_i, \delta)} < k$ , then we discard  $l_i$  from  $\mathcal{L}$ . For the remaining points in  $\mathcal{L}$ , we extract the point  $l_{\min}$ , whose corresponding disk depth  $d_{D(l_i, \delta)}$  is the minimum among all disks. Then, we find the point with maximum depth on  $d_{D(l_{\min}, \delta)}$ , denoted by  $p_{\min}^*$ . It is obvious that  $d_{D(\mathcal{L}, \delta)}(p_{\min}^*) \geq k$ , and we obtain a  $k$ -enclosing disk  $D(p_{\min}^*, \delta)$ . Then, we add  $D(p_{\min}^*, \delta)$  to  $\mathcal{D}^\delta$ . For the remaining points covered by  $D(p_{\min}^*, \delta)$ , we form a perturbed group  $(p_{\min}^*, \{i \mid l_i \in D(p_{\min}^*, \delta) \cap \mathcal{L}\})$  and tag  $p_{\min}^*$  to the corresponding users as their perturbed location. Then, we discard these locations from  $\mathcal{L}$ . We keep forming perturbed groups and discarding points until there are less than  $k$  points in  $\mathcal{L}$ .

**Theorem 3:** HPUM<sub>N</sub> preserves  $k$ -anonymity location privacy.

*Proof:* We prove that each user  $i$  included in the perturbed groups is tagged with the same perturbed location as at least  $k-1$  other users.

For each  $l_{\min} \in \mathcal{L}$ , it guarantees that  $d_{D(l_{\min}, \delta)} \geq k$ , based on line 5 in Algorithm 4. Since  $p_{\min}^*$  is the point with maximum depth on  $D(l_{\min}, \delta)$ , we have  $d_{D(\mathcal{L}, \delta)}(p_{\min}^*) \geq k$ . By Claim 2,  $D(p_{\min}^*, \delta)$  covers at least  $k$  points in  $\mathcal{L}$ . Thus, there are at least  $k$  users in each perturbed group  $(p_{\min}^*, \{i \mid l_{\min} \in D(p_{\min}^*, \delta)\})$ . Therefore, each user  $i$  included in the perturbed groups is tagged with the same perturbed location with at least  $k-1$  other users. ■

The time complexity of HPUM<sub>N</sub> is dominated by the nested while-loop and for-loop. The while-loop takes  $O(n/k)$  time.



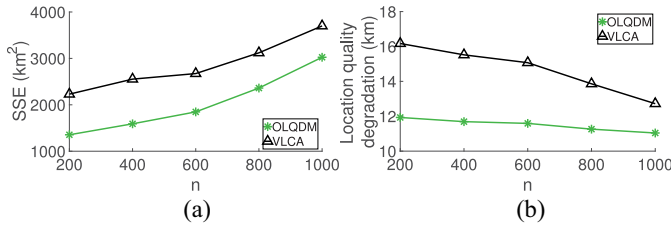


Fig. 2. Impact of  $n$  on OLoQ and VCLA. (a) SSE. (b) Location quality degradation (km).

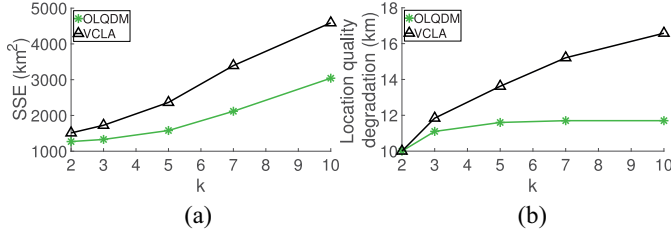


Fig. 3. Impact of  $k$  on OLoQ and VCLA. (a) SSE. (b) Location quality degradation (km).

The for-loop takes  $O(n)$  time to update disk depth. Hence, the overall time complexity of HPUM<sub>N</sub> is  $O(n^3/k)$ .

## VI. PERFORMANCE EVALUATION

In this section, we evaluate the performance of OLoQ, OPUM<sub>O</sub>, and HPUM<sub>N</sub> by comparing them with the existing  $k$ -anonymity location privacy-preserving algorithms [36].

### A. Evaluation Setup

As we surveyed in Section II, there is no existing algorithm that aims to minimize the location quality degradation or to maximize the number of protected users. The most related work for  $k$ -anonymity location privacy is VCLA [36], which is a heuristic algorithm that uses the microaggregation approach to obtain anonymized locations and aims to minimize the SSE.

We use the CRAWDAD data set roma/taxi [3] for our simulations. The data set contains the mobility traces of approximately 320 taxis collected over 30 days in Rome, Italy. Each mobility trace consists of a sequence of GPS coordinates collected roughly every 7 s along with corresponding timestamps. Because our model does not require the time information, we removed the timestamps from the whole 30-day data set and treated all the data points as independent. We then randomly select data points as input to our algorithms. Investigating the tradeoff between location quality and privacy with both spatial and temporal information will be one of our future research directions as we will discuss in Section VII.

### B. Performance Metrics

We are interested in the following performance metrics.

- 1) *SSE*: Suppose a point set  $\mathcal{L}$  is divided into  $m$  groups. The sum of squared errors of perturbed group  $j$  is defined as

$$\text{sse}_j = \sum_{p=1}^{n_j} [(x_{jp} - \bar{x}_j)^2 + (y_{jp} - \bar{y}_j)^2]$$

where  $n_j$  is the number of users in the  $j$ th group satisfying  $n_j \geq k$ , and  $(x_{jp}, y_{jp})$  is the location of the  $p$ th user

with  $(\bar{x}_j, \bar{y}_j)$  the perturbed location of the  $j$ th group. The SSE is the sum of  $\text{sse}_j$

$$\text{SSE} = \sum_{j=1}^m \text{sse}_j = \sum_{j=1}^m \sum_{p=1}^{n_j} [(x_{jp} - \bar{x}_j)^2 + (y_{jp} - \bar{y}_j)^2]$$

where SSE describes the overall group homogeneity after group formation. When nearby points are grouped together, SSE will be small and the groups are more homogeneous.

### 2) Location quality degradation.

### 3) Number of protected users.

In our evaluation, we show the impact of the number of users ( $n$ ) and  $k$  on OLoQ and VCLA in terms of SSE and location quality degradation. For the impact of  $n$ , we vary it from 200 to 1000 with an increment of 200, while fixing  $k = 5$ . For the impact of  $k$ , we set it to be 2, 3, 5, 7, and 10, while fixing  $n = 400$ .

Then, we show the impact of the number of users ( $n$ ), the value of  $k$ , and the location quality degradation bound ( $\delta$ ) on OPUM<sub>O</sub>, HPUM<sub>N</sub>, and VCLA in terms of the number of protected users. For the impact of  $n$ , we vary it from 200 to 1000 with an increment of 200, fixing  $k = 5$  and  $\delta = 500$  m. For the impact of  $k$ , we set it to be 2, 3, 5, 7, and 10, fixing  $n = 400$  and  $\delta = 500$  m. For the impact of  $\delta$ , we vary it from 500 to 2500 m with an increment of 500 m, fixing  $n = 400$  and  $k = 5$ . We choose this range of  $\delta$ , because this error is tolerable for location quality in most crowdsensing applications. All the results are averaged over 100 independent runs.

In addition, we show the impact of  $\epsilon$  and  $k$  on NPUM<sub>N</sub> in terms of the number of protected users. For the impact of  $\epsilon$ , we set it to be 1/750, 1/1000, and 1/1250, while fixing  $n = 50$  and  $\delta = 500$ . For the impact of  $k$ , we vary it from 3 to 5 with an increment of 1, while fixing  $n = 50$  and  $k = 4$ . Due to the high time complexity of NPUM<sub>N</sub>, all the results are averaged for 30 independent runs.

### C. Evaluation Results and Analysis

Fig. 2 shows the impact of  $n$  on OLoQ and VCLA. Fig. 2(a) shows the impact of  $n$  on SSE. We observe that OLoQ can always introduce lower SSE, which is very essential to obtain accurate sensing data. Besides, the SSE increases with  $n$ , because sparser location distribution will lead to larger errors. In Fig. 2(b), the location quality degradations of OLoQ and VCLA decrease with  $n$ . We also observe that OLoQ performs better than VCLA, especially, with fewer users, because OLoQ minimizes the location quality degradation optimally, while VCLA heuristically aggregate locations by first choosing the farthest point and then aggregating the nearest points to it.

Fig. 3 shows the impact of  $k$  on OLoQ and VCLA. Fig. 3(a) illustrates that the SSE gradually increases with more stringent privacy protection in both OLoQ and VCLA. To protect more users' locations in one perturbed group, it is inevitable to diminish the location quality to some degree. OLoQ has a lower SSE, because it minimizes the location quality degradation. From Fig. 3(b), we observe that OLoQ outputs perturbed groups with minimum location quality degradation and performs significantly better than VCLA. The common

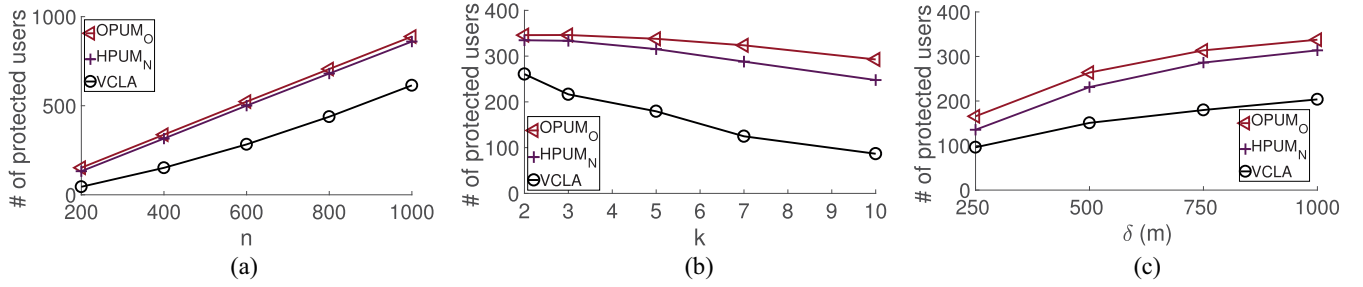


Fig. 4. Impact of  $n$ ,  $k$ , and  $\delta$  on OPUM<sub>O</sub>, HPUM<sub>N</sub>, and VCLA. (a) Impact of  $n$  ( $k = 5$ ,  $\delta = 1000$ ). (b) Impact of  $k$  ( $n = 400$ ,  $\delta = 1000$ ). (c) Impact of  $\delta$  ( $k = 5$ ,  $n = 400$ ).

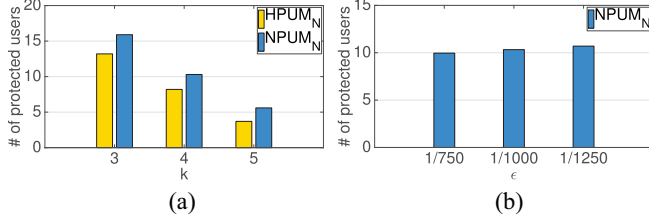


Fig. 5. Impact of  $k$  and  $l$  on HPUM<sub>N</sub> and NPUM<sub>N</sub>. (a) Impact of  $k$  ( $\epsilon = 10^{-3}$ ). (b) Impact of  $\epsilon$  ( $n = 50$ ).

trend is that with more stringent privacy protection the location quality degradation increases in both OLoQ and VCLA.

Fig. 4 shows the impact of  $n$ ,  $k$ , and  $\delta$  on OPUM<sub>O</sub>, HPUM<sub>N</sub>, and VCLA. Fig. 4(a) illustrates that the number of protected users increases with more users involved. OPUM<sub>O</sub> includes most protected users, because it is an optimal algorithm that allows overlapping. We can also see that the performance of HPUM<sub>N</sub> is very close to OPUM<sub>O</sub>. Since OPUM<sub>O</sub> allows overlapping and already includes all the possible users with the location quality degradation constraint, the optimal solution in the nonoverlapping case can never include more users than OPUM<sub>O</sub>. Thus, HPUM<sub>N</sub> achieves near-optimal performance. In addition, VCLA includes fewer protected users, because VCLA always chooses the farthest point to form a new group. Fig. 4(b) illustrates that the number of protected users decreases, when  $k$  increases. The reason is that a larger  $k$  requires more stringent privacy protection, while the location quality degradation bound remains the same, which makes some users unprotected. We also notice that HPUM<sub>N</sub>'s performance is very close to OPUM<sub>O</sub>. Since the optimal solution in the nonoverlapping case is no greater than that of the overlapping case, we can say that HPUM<sub>N</sub> achieves near-optimal performance, whereas VCLA includes fewer protected users, because it forms perturbed groups by choosing the farthest point and then aggregating the nearest points to it. Fig. 4(c) demonstrates that the number of protected users increases when  $\delta$  is larger, because more users' locations can be included in a perturbed group with a larger  $\delta$ .

Fig. 5 shows the impact of  $k$  and  $\epsilon$  on NPUM<sub>N</sub>. In Fig. 5(a), we observe that the number of protected users decreases. A larger  $k$  requires more stringent privacy protection, while the location quality degradation bound remains the same, which makes some users unprotected. In addition, NPUM<sub>N</sub> achieves better performance than HPUM<sub>N</sub>, because NPUM<sub>N</sub> has an approximation close to 1 when  $\epsilon$  is very small. In Fig. 5(b),

we notice that the number of protected users increases, because of the approximation ratio increases when the value of  $\epsilon$  decreases.

## VII. CONCLUSION

In this article, we considered the tradeoff between location privacy and location quality in location-based crowdsensing from an optimization perspective. Two optimization problems have been studied. The first problem is to minimize the location quality degradation, while guaranteeing the location privacy for all users. We presented an efficient optimal algorithm OLoQ for this problem. The second problem is to maximize the number of protected users with location quality degradation constraint. To satisfy the different requirements of the platforms, we further considered two cases: 1) overlapping and 2) nonoverlapping perturbations. For the former case, we gave an efficient optimal algorithm OPUM<sub>O</sub>. For the latter case, we proved its NP-hardness, and designed a near-optimal  $(1 - \epsilon)$ -approximation algorithm NPUM<sub>N</sub> and a fast and effective heuristic algorithm HPUM<sub>N</sub>. Extensive simulations show that OLoQ and OPUM<sub>O</sub> achieve optimal performance. In addition, NPUM<sub>N</sub> and HPUM<sub>N</sub> achieve near-optimal performance.

There are two directions that we can work on in the future. In the current work, we assume that all users require the same anonymity level. Our algorithms can be extended to a personalized  $k$ -anonymity model, where each user can specify a different anonymity level requirement. Another direction is to consider temporal information privacy protection, because the crowdsensing data are sometimes time sensitive. An attacker can infer a user's personal preference or behaviors based on the user's location information combined with its temporal information. Thus, it will be better to perturb both the spatial and temporal information.

In addition, we plan to conduct the mobile crowdsensing on constructing urban noise maps [42] as a case study. People in major cities suffer from noise pollution, which compromises working efficiency and mental health. Urban noises usually vary by locations, change over time, and consist of multiple sound sources, e.g., loud music, vehicle traffic, and constructions. New York City (NYC) has opened a platform CityNoise [42] to allow people to submit the urban noise sensing data tagged with locations by using a mobile app, which is location aware and open source. Our proposed algorithms will process the sensing data by tagging them with perturbed locations. With the processed location data, we will be able to



generate a noise map and study the tradeoff between location quality and privacy.

## REFERENCES

- [1] B. Agir, T. G. Papaioannou, R. Narendula, K. Aberer, and J.-P. Hubaux, "User-side adaptive protection of location privacy in participatory sensing," *Geoinformatica*, vol. 18, no. 1, pp. 165–191, 2014.
- [2] A. R. Beresford and F. Stajano, "Mix zones: User privacy in location-aware services," in *Proc. IEEE Annu. Conf. Pervasive Comput. Commun. Workshops (PerCom Workshops)*, 2004, pp. 127–131.
- [3] L. Bracciale, M. Bonola, P. Loreti, G. Bianchi, R. Amici, and A. Rabuffi, *Crowdad Data Set Roma/Taxi* (v. 2014–07-17). Accessed: Feb. 15, 2019. [Online]. Available: <http://crowdad.org/roma/taxi/20140717/>
- [4] D. Christin, A. Reinhardt, S. S. Kanhere, and M. Hollick, "A survey on privacy in mobile participatory sensing applications," *J. Syst. Softw.*, vol. 84, no. 11, pp. 1928–1946, 2011.
- [5] J. Domingo-Ferrer and J. M. Mateo-Sanz, "Practical data-oriented microaggregation for statistical disclosure control," *IEEE Trans. Knowl. Data Eng.*, vol. 14, no. 1, pp. 189–201, Jan./Feb. 2002.
- [6] M. Duckham and L. Kulik, "A formal model of obfuscation and negotiation for location privacy," in *Proc. Int. Conf. Pervasive Comput.*, 2005, pp. 152–170.
- [7] A. Efrat, M. Sharir, and A. Ziv, "Computing the smallest  $k$ -enclosing circle and related problems," *Comput. Geom.*, vol. 4, no. 3, pp. 119–136, 1994.
- [8] S. Gao, J. Ma, W. Shi, and G. Zhan, "Towards location and trajectory privacy protection in participatory sensing," in *Proc. Int. Conf. Mobile Comput. Appl. Services (MobiCASE)*, 2011, pp. 381–386.
- [9] G. Ghinita, "Privacy for location-based services," *Synth. Lectures Inf. Security Privacy Trust*, vol. 4, no. 1, pp. 1–85, 2013.
- [10] X. Gong, X. Chen, K. Xing, D.-H. Shin, M. Zhang, and J. Zhang, "From social group utility maximization to personalized location privacy in mobile networks," *IEEE/ACM Trans. Netw.*, vol. 25, no. 3, pp. 1703–1716, Jun. 2017.
- [11] D. Hasenfratz, O. Saukh, S. Sturzenegger, and L. Thiele, "Participatory air pollution monitoring using smartphones," *Mobile Sens.*, vol. 1, no. 1, pp. 1–5, 2012.
- [12] N. Hu, "Approximation algorithms for geometric covering problems for disks and squares," M.S. thesis, School Comput. Sci., Univ. Waterloo, Waterloo, ON, Canada, 2013.
- [13] C. V. N. Index, *Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2014–2019*. Accessed: Feb. 29, 2020. [Online]. Available: <https://community.cisco.com/t5/wireless-mobility-documents/cisco-visual-networking-index-global-mobile-data-traffic/ta-p/3612708?dtid=ossdc000283>
- [14] X. Jin and Y. Zhang, "Privacy-preserving crowdsourced spectrum sensing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2016, pp. 1–9.
- [15] H. Kido, Y. Yanagisawa, and T. Satoh, "An anonymous communication technique using dummies for location-based services," in *Proc. Int. Conf. Pervasive Services (ICPS)*, 2005, pp. 88–97.
- [16] G. Kokolakis and D. Fouskakis, "Importance partitioning in micro-aggregation," *Comput. Stat. Data Anal.*, vol. 53, no. 7, pp. 2439–2445, 2009.
- [17] A. Krause, E. Horvitz, A. Kansal, and F. Zhao, "Toward community sensing," in *Proc. Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, 2008, pp. 481–492.
- [18] S. Li, H. Zhu, Z. Gao, X. Guan, K. Xing, and X. Shen, "Location privacy preservation in collaborative spectrum sensing," in *Proc. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, 2012, pp. 729–737.
- [19] V. A. Memos, K. E. Psannis, Y. Ishibashi, B.-G. Kim, and B. B. Gupta, "An efficient algorithm for media-based surveillance system (EAMSuS) in IoT smart city framework," *Future Gener. Comput. Syst.*, vol. 83, pp. 619–628, Jun. 2018.
- [20] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich monitoring of road and traffic conditions using mobile smartphones," in *Proc. ACM Conf. Embedded Netw. Sensor Syst. (SenSys)*, 2008, pp. 323–336.
- [21] M. Murshed, A. Iqbal, T. Sabrina, and K. M. Alam, "A subset coding based  $k$ -anonymization technique to trade-off location privacy and data integrity in participatory sensing systems," in *Proc. IEEE 10th Int. Symp. Netw. Comput. Appl. (NCA)*, 2011, pp. 107–114.
- [22] M. Piorkowski, N. Sarafjanovic-Djukic, and M. Grossglauser, *CROWDAD Dataset EPFL/Mobility* (v. 2009–02-24). Accessed: Feb. 2009. [Online]. Available: <https://crowdad.org/epfl/mobility/20090224>
- [23] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient IoT-based sensor big data collection–processing and analysis in smart buildings," *Future Gener. Comput. Syst.*, vol. 82, pp. 349–357, May 2018.
- [24] K. E. Psannis, C. Stergiou, and B. B. Gupta, "Advanced media-based smart big data on intelligent cloud systems," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 1, pp. 77–87, Jan.–Mar. 2019.
- [25] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Earphone: An end-to-end participatory urban noise mapping system," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw. (IPSN)*, 2010, pp. 105–116.
- [26] I. Rodhe, C. Rohner, and E. C.-H. Ngai, "On location privacy and quality of information in participatory sensing," in *Proc. 8th ACM Symp. QoS Security Wireless Mobile Netw. (Q2SWinet)*, 2012, pp. 55–62.
- [27] R. Shokri, G. Theodorakopoulos, P. Papadimitratos, E. Kazemi, and J.-P. Hubaux, "Hiding in the mobile crowd: Location privacy through collaboration," *IEEE Trans. Depend. Secure Comput.*, vol. 11, no. 3, pp. 266–279, May/Jun. 2014.
- [28] A. Solanas, A. Martinez-Balleste, and J. Domingo-Ferrer, "V-MDAV: A multivariate microaggregation with variable group size," in *Proc. COMPSTAT Symp. IASC (IASC)*, 2006, pp. 917–925.
- [29] C. Stergiou, K. E. Psannis, B.-G. Kim, and B. Gupta, "Secure integration of IoT and cloud computing," *Future Gener. Comput. Syst.*, vol. 78, pp. 964–975, Jan. 2018.
- [30] C. Stergiou, K. E. Psannis, A. P. Plageras, Y. Ishibashi, and B.-G. Kim, "Algorithms for efficient digital media transmission over iot and cloud networking," *J. Multimedia Inf. Syst.*, vol. 5, no. 1, pp. 27–34, 2018.
- [31] L. Sweeney, "k-anonymity: A model for protecting privacy," *Int. J. Uncertainty Fuzziness Knowl. Based Syst.*, vol. 10, no. 5, pp. 557–570, 2002.
- [32] H. To, G. Ghinita, and C. Shahabi, "A framework for protecting worker location privacy in spatial crowdsourcing," *Proc. VLDB Endow.*, vol. 7, no. 10, pp. 919–930, 2014.
- [33] I. J. Vergara-Laurens and M. A. Labrador, "Preserving privacy while reducing power consumption and information loss in LBS and participatory sensing applications," in *Proc. IEEE GLOBECOM Workshops (GLOBECOM Workshops)*, 2011, pp. 1247–1252.
- [34] K. Vu, R. Zheng, and J. Gao, "Efficient algorithms for  $k$ -anonymous location privacy in participatory sensing," in *Proc. IEEE INFOCOM*, 2012, pp. 2399–2407.
- [35] W. Wang and Q. Zhang, "Privacy-preserving collaborative spectrum sensing with multiple service providers," *IEEE Trans. Wireless Commun.*, vol. 14, no. 2, pp. 1011–1019, Feb. 2015.
- [36] X. Wang, Z. Liu, X. Tian, X. Gan, Y. Guan, and X. Wang, "Incentivizing crowdsensing with location-privacy preserving," *IEEE Trans. Wireless Commun.*, vol. 16, no. 10, pp. 6940–6952, Oct. 2017.
- [37] Z. Xiao, X. Meng, and J. Xu, "Quality aware privacy protection for location-based services," in *Proc. Int. Conf. Database Syst. Adv. Appl. (DASFAA)*, 2007, pp. 434–446.
- [38] T. Xu and Y. Cai, "Location anonymity in continuous location-based services," in *Proc. 15th Annu. ACM Int. Symp. Adv. Geograph. Inf. Syst. (GIS)*, 2007, pp. 1–8.
- [39] L. Yang, W. Li, M. Ghandehari, and G. Fortino, "People-centric cognitive Internet of Things for the quantitative analysis of environmental exposure," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2353–2366, Aug. 2018.
- [40] F. Zhang, L. He, W. He, and X. Liu, "Data perturbation with state-dependent noise for participatory sensing," in *Proc. IEEE INFOCOM*, 2012, pp. 2246–2254.
- [41] Y. Zhang, M. Li, D. Yang, J. Tang, and G. Xue, "Optimizing location quality in privacy preserving crowdsensing," in *Proc. IEEE GLOBECOM*, 2019, pp. 1–6.
- [42] Y. Zheng, T. Liu, Y. Wang, Y. Zhu, Y. Liu, and E. Chang, "Diagnosing New York city's noises with ubiquitous data," in *Proc. 16th ACM Int. Conf. Ubiquitous Comput. (UbiComp)*, 2014, pp. 715–725.



**Yuhui Zhang** (Student Member, IEEE) received the B.S. degree from Sun Yat-sen University, Guangzhou, China, in 2011. She is currently pursuing the Ph.D. degree in computer science with the Colorado School of Mines, Golden, CO, USA.

Her main research interests lie in the areas of blockchain, game theory, location privacy, and crowdsourcing.



**Ming Li** (Student Member, IEEE) received the B.S. degree in geochemistry from Peking University, Beijing, China, in 2013, and the M.S. degree in computer science from the Colorado School of Mines, Golden, CO, USA, in 2015, where she is currently pursuing the Ph.D. degree.

Her main research interests include game theory, contract theory, crowdsourcing, and smartphone-based sensing systems.



**Dejun Yang** (Senior Member, IEEE) received the B.S. degree in computer science from Peking University, Beijing, China, in 2007, and the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2013.

He is currently an Associate Professor of computer science with the Colorado School of Mines, Golden, CO, USA. His research interests include Internet of Things, networking, and mobile sensing and computing with a focus on the application of game theory, optimization, algorithm design, and

machine learning to resource allocation, security and privacy problems.

Prof. Yang has received the IEEE Communications Society William R. Bennett Prize in 2019 (Best Paper Award for the IEEE/ACM TRANSACTIONS ON NETWORKING and the IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT in the previous three years), the Best Paper Awards at the IEEE Global Communications Conference in 2015, the IEEE International Conference on Mobile Ad hoc and Sensor Systems in 2011, and the IEEE International Conference on Communications in 2011 and 2012, as well as the Best Paper Award Runner-up at the IEEE International Conference on Network Protocols in 2010. He currently serves as an Associate Editor for the IEEE INTERNET OF THINGS JOURNAL. He has served as the TPC Vice-Chair for Information Systems for the IEEE International Conference on Computer Communications.



**Jian Tang** (Fellow, IEEE) received the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2006.

He is currently a Professor with the Department of Electrical Engineering and Computer Science, Syracuse University, Syracuse, NY, USA. He has published over 100 papers in premier journals and conferences. His research interests lie in the areas of wireless networking, deep learning, big data, and cloud computing.

Prof. Tang received the NSF CAREER Award in 2009, the 2016 Best Vehicular Electronics Paper Award from the IEEE Vehicular Technology Society, and the Best Paper Awards from ICC in 2014 and GLOBECOM in 2015, respectively. He has been an Editor of the IEEE TRANSACTIONS ON NETWORK SCIENCE AND ENGINEERING since 2017, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS since 2016, and the IEEE INTERNET OF THINGS JOURNAL since 2013, and was an Editor of the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY from 2010 to 2017. He served as the TPC Co-Chair for the IEEE International Conference on Internet of Things in 2015 and for International Conference on Computing, Networking and Communication in 2016.



**Guoliang Xue** (Fellow, IEEE) received the B.S. degree in mathematics and the M.S. degree in operations research from Qufu Normal University, Qufu, China, in 1981 and 1984, respectively, and the Ph.D. degree in computer science from the University of Minnesota, Minneapolis, MN, USA, in 1991.

He is a Professor of computer science and engineering with Arizona State University, Tempe, AZ, USA. His research interests span the areas of quality-of-service provisioning, network security and privacy, crowdsourcing and network economics, RFID

systems and Internet of Things, and smart city and smart grids. He has published over 280 papers in these areas, many of which in top conferences, such as IEEE ICNP, INFOCOM, ACM Conference on Mobile Computing and Networking, ACM International Symposium on Mobile Ad Hoc Networking and Computing, and Network and Distributed System Security Symposium, and top journals, such as the IEEE/ACM TRANSACTIONS ON NETWORKING, the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, and the IEEE TRANSACTIONS ON MOBILE COMPUTING.

Prof. Xue was the elected Vice President of the IEEE Communications Society for Conferences from 2016 to 2017. He has served on the TPC of many conferences, including the ACM Conference on Computer and Communications Security, MobiHoc, ICNP, and INFOCOM. He served on the editorial board for the IEEE/ACM TRANSACTIONS ON NETWORKING and *Computer Networks*. He serves as an Area Editor of the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, overseeing 13 editors in the wireless networking area. He was the TPC Co-Chair of INFOCOM in 2010 and the General Co-Chair of the IEEE Conference on Communications and Network Security in 2014. He was a Keynote Speaker at IEEE Conference on Local Computer Networks in 2011 and ICNC in 2014.



**Jia Xu** (Member, IEEE) received the M.S. degree from the School of Information and Engineering, Yangzhou University, Yangzhou, China, in 2006, and the Ph.D. degree from the School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing, China, in 2010.

He is currently a Professor with the School of Computer Science, Nanjing University of Posts and Telecommunications. He was a Visiting Scholar with the Department of Electrical Engineering and Computer Science, Colorado School of Mines,

Golden, CO, USA, from November 2014 to May 2015. His main research interests include crowdsourcing, edge computing, and wireless sensor networks.