Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

A Survey Of differential privacy-based techniques and their applicability to location-Based services

Jong Wook Kim[a], Kennedy Edemacu[a], Jong Seon Kim[b], Yon Dohn Chung[b], Beakcheol Jang[c,*]

[a] Department of Computer Science, Sangmyung University, Seoul, Korea
[b] Department of Computer Science and Engineering, Korea University, Seoul, Korea
[c] Graduate School of Information, Yonsei University, Seoul, Korea

ARTICLE INFO

Article history:
Received 10 December 2020
Revised 20 April 2021
Accepted 31 August 2021
Available online 14 September 2021

Keywords:
Location-based services
Differential privacy
Location privacy
Privacy
Security

ABSTRACT

The widespread use of mobile devices such as smartphones, tablets, and smartwatches has led users to constantly generate various location data during their daily activities. Consequently, a growing interest has been seen in location-based services (LBSs), which aim to provide services adjusted to the current locations of users. However, location information may contain sensitive data; therefore, most users are reluctant to provide their exact location data to service providers. This has been identified as the most significant challenge in LBSs. Recently, differential privacy (DP) has emerged as a de facto standard for privacy-preserving data processing. With its strong privacy guarantees, DP has been used in diverse areas such as the collection, analysis, and release of sensitive private data, and several variants of DP have been proposed in the literature. The main objective of this paper is to investigate the applicability of DP-based approaches in an LBS setting. In this paper, we first describe the basic concept of DP and then survey its three variants: (a) geo-indistinguishability, (b) private spatial decomposition, and (c) local differential privacy, which are designed or can be used to protect location privacy in LBSs. Furthermore, we explore the applicability of DP-based schemes in protecting location privacy in different location data processing, collection, and publishing scenarios in LBSs. Finally, certain promising future research directions are discussed to spur further research in this area.

1. Introduction

Nowadays, with the widespread use of mobile devices such as smartphones, tablets, and smartwatches, users are constantly generating various location data during their daily activities. Accordingly, there has been a growing interest in location-based services (LBSs), which provide services adjusted to the locations of users. Examples of LBS include outdoor/indoor navigation (Shi et al., 2017; Teng et al., 2015), location-based advertisement (Rashid et al., 2008), point-of-interest (POI) recommendation (Chang et al., 2018; Cheng et al., 2013), weather applications (Yah, 0000), crowd-sensing (To et al., 2014), etc. According to a recent report by the company, Grand Review Research, the LBS market is expected to reach USD 18.74 billion by 2025, rising at a compounded interest rate of 43.3% from 2018 to 2025 (ind, 2019).

* Corresponding author.
E-mail address: bjang@yonsei.ac.kr (B. Jang).

https://doi.org/10.1016/j.cose.2021.102464
0167-4048/© 2021 Elsevier Ltd. All rights reserved.

LBSs are possible, thanks to the development of localization techniques that are capable of accurately estimating the current position of a user in outdoor/indoor environments. Over the last few decades, extensive studies have been conducted to effectively estimate the positions of users by leveraging various information and communication techniques, such as global positioning system signal, wireless fidelity, bluetooth, vision, inertial sensors, ultra-wideband, and radio-frequency identification (Choi et al., 2004; Harle, 2013; Hightower et al., 2001; Hofmann-Wellenhof et al., 2001; Jang and Kim, 2019; Jang and Sichitiu, 2012; Liu et al., 2007; Pahlavan et al., 2002; Cypriani, Lassabe, Canalda, Spies, system; Feldmann, Kyamakya, Zapater, Lue, An indoor Bluetooth-based positioning system: Concept, evaluation; Kitasuka, Hisazumi, Nakanishi, Fukuda, devices, 802.11; Prasithsangaree, Krishnamurthy, Chrysanthis; Wang et al., 2003).

Advances in localization techniques help users to easily measure their current locations and send various service requests along with their location information to LBSs. However, this may raise serious privacy issues (Terrovitis, 2011) because location information typically contains certain sensitive information. Malicious LBS providers may infer users' sensitive information by tracking and analyzing their position information. For example, by tracking the indoor location information of patients in huge hospital complexes, it is possible to identify the departments they visit, and subsequently infer the diseases they are suffering from. In other examples, location information can reveal users' films of interest watched by them in complex theaters, products of interest in the shopping malls they visited, etc.

With the advances in localization techniques, LBS providers are able to easily collect a vast amount of location data from diverse users. They are typically interested in collecting users' location data for better understanding of user behavior, and consequently, enhancing their service quality. Moreover, LBSs can publish the collected location data to third parties (e.g., researchers and commercial organizations) for research or profit purposes. However, the indiscriminate collection and publication of individuals' location data raises serious privacy concerns owing to the personal and sensitive nature of such information that pertains to an individual's daily routines, lifestyle, social relationships, etc. For example, Nordstrom department stores in the United States track the movement of customers in shopping areas without their consent by using smartphone Wi-Fi signals for efficient and effective store operation (Attention, 2019). This has developed into a serious social issue because most consumers are uncomfortable with the fact that their sensitive location information is collected without their consent. Typically, most users are reluctant to provide their exact location information to service providers owing to privacy concerns, which has been identified as the most significant challenge faced by the LBS industry.

In research literature on data management, extensive studies have been conducted in the area of data privacy protection. Traditional approaches are based on anonymization techniques such as k-anonymity (Lee et al., 2017; Sweeney, 2002; Fung, Wang, Yu, preservation, privacy; LeFevre, DeWitt, Ramakrishnan, k anonymity; LeFevre, DeWitt, Ramakrishnan, k anonymity; Wang et al., 2014), l-diversity (Machanavajjhala et al., 2007), and t-closeness (Li et al., 2007). These approaches transform original data into a more generalized form in which individual data cannot be uniquely identified. Furthermore, there have been several attempts to apply the concept of anonymization techniques to location data (G-Divanis et al., 2010; Gedik and Liu, 2008; Niu et al., 2014; Yang et al., 2013). However, these anonymization-based approaches are based on the assumption that the background knowledge of an attacker is either limited or known in advance. Consequently, they are unable to provide privacy guarantee against attackers with arbitrary background knowledge. Furthermore, more recent techniques such as cryptography and perturbation mechanisms are being used to protect the location privacy of users in LBSs (Primault et al., 2018).

1.1. *Motivation and scope*

The growing demand for customized services and the significant advances in techniques that measure the location of a user have sparked the development of various LBS applications. Although such convenient services based on users' current locations make our lives easier, the users are subject to the risk of privacy leakage. The concerns regarding exposure of sensitive information being exposed to malicious LBS providers or third parties by sending private location information to the LBS servers prevent the wider adoption of various LBSs and consequently deprive users of their benefits. To promote user trust in the use of LBSs, and thereby accelerate the widespread use of such services, it is essential to provide privacy protection guarantees for location data.

Fig. 1 illustrates common use cases of LBS. The privacy issues caused by the leakage of location information can be considered in the following three use cases: location-based query processing, location data collection, and location data publishing.

- Location-based query processing: In a real-time system, users who want to receive services adjusted to their current locations submit the location-based queries to the LBS server either directly or via the trusted server. Usually, location-based queries from users comprise location information of the user's current location and a service request. For example, users who want to locate nearby coffee shops send a query with their location information to an LBS provider using a mobile device. In this case, a malicious LBS provider may collect and track the exact location information of users in outdoor/indoor spaces. Therefore, users want to safeguard the privacy for their locations. *Privacy-preserving location-based query processing* should guarantee that users' exact locations are not disclosed to an LBS provider, while providing the users with a similar level of quality of services adjusted to their current location.
- Location data collection: LBS providers aim to collect a vast amount of location data from diverse users over a long period of time, and leverage the collected data set to enhance the quality of services to be provided to their customers. Location-based recommendation systems consider the location information of users to provide appropriate recommendations to them (Rodriguez-Hernandez et al., 2015).

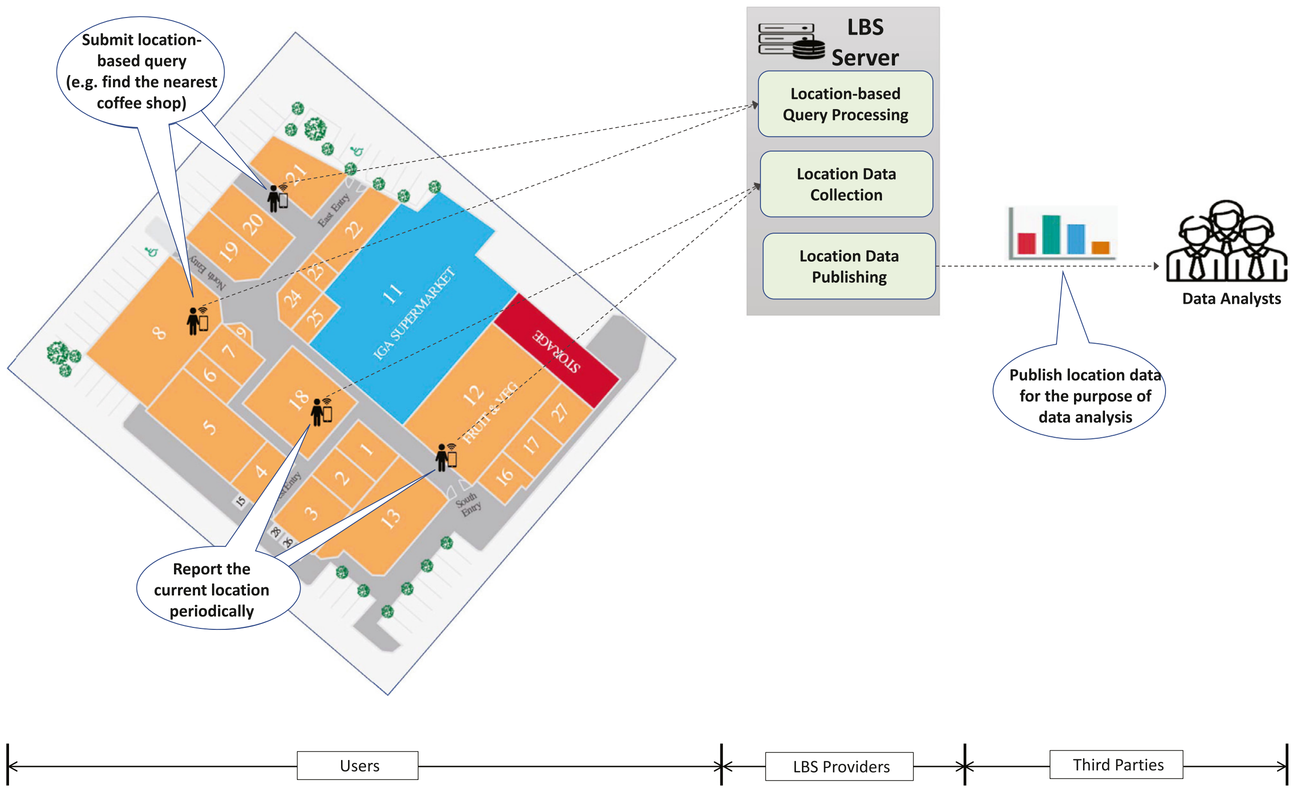

Submit location-based query (e.g. find the nearest coffee shop)
Report the current location periodically
IGA SUPERMARKET
STORAGE
FRUIT & VEG
East Entry
North Entry
South Entry
LBS Server
Location-based Query Processing
Location Data Collection
Location Data Publishing
Publish location data for the purpose of data analysis
Data Analysts
Users
LBS Providers
Third Parties

Fig. 1 – Location-based query processing, location data collection and location data publishing in the LBS.

Location-based recommendation systems have been successfully applied to several areas, including traveling and tour planning (Li et al., 2010), shopping recommendation (Takeuchi and Sugimoto, 2005), and news or web content recommendation (Sandholm and Ung, 2011). Most location-based recommendation systems are based on a location-aware collaborative filtering algorithm that requires a large amount of location/item information from diverse users to compute appropriate recommendations effectively.

As an example, consider an LBS provider that recommends retail shops to users based on the locations of the previous shops visited by them. This LBS provider has to collect the location information of diverse users for a location-aware collaborative filtering algorithm. In this case, the current GPS coordinates of each user, which can be determined by using his/her smartphone, can be periodically collected by the LBS server either at fixed time intervals or when predefined events occur (e.g., when a user moves from one location to another). However, owing to privacy concerns, most users are reluctant to provide their exact location information to LBS providers. *Privacy-preserving location data collection* must ensure that an LBS provider can collect the location information of users in such a manner that the exact user location is not disclosed to other users or the LBS provider. Furthermore, it must enable the LBS provider to obtain useful information from the collected location data to improve the quality of services.

- Location data publishing: In recent years, there has been an increasing demand for publishing datasets collected in various application areas to third parties (e.g., researchers and commercial organizations) who intend to conduct various data analysis tasks. Particularly, publishing datasets collected in LBSs is beneficial for both users and service providers. For example, transport LBS, which provides drivers with various convenience and safety services, such as car navigation, vehicle management, and real-time traffic maps, has been one of the most popular application fields of LBS (Huang and Gartner, 2018). The introduction of the Internet of vehicles with its own communication capabilities has enabled the current intelligent transportation system (ITS), which is an important application of transport LBS, to collect a large amount of traffic data, such as driving trajectories and patterns, easily from individual vehicles on the road. The large amount of geospatial information (usually represented by latitude and longitude coordinates) of vehicles collected by the ITS can be published to third parties for performing various data analysis tasks, such as traffic estimation and road maintenance planning, which are beneficial to both vehicle drivers and the ITS.
  However, location data typically comprise sensitive information; therefore, releasing them directly for public use may violate existing privacy requirements. *Privacy-preserving location data publishing* must ensure that an adversary cannot infer the exact location of a specific user from the published data.

In addition to the cases described in Fig. 1, numerous privacy problems can arise during the use of LBSs, such as eavesdropping on the communication channel between users and LBS servers, location disclosure by a malicious positioning server, etc. However, it should be noted that this survey focuses on protecting the location privacy of users in location-based query processing, location data collection, and location data publishing, which are common use cases to handle location data in LBSs, as depicted in Fig. 1.

1.2. *Contributions*

Recently, differential privacy (DP) has emerged as a de facto standard for privacy-preserving data processing. DP is based on a formal mathematical definition that provides a probabilistic privacy guarantee against attackers with arbitrary background knowledge (Dwork, 2006; Dwork et al., 2006). It ensures that an attacker cannot infer with high confidence whether a particular individual is participating in the query result (or the disseminated data). Owing to its strong privacy guarantees, DP is being used in various areas, such as in the collection, analysis, and release of sensitive private data. Furthermore, several variants of DP have been proposed in the literature. Thus, in this study, we survey different DP-based solutions for privacy protection of location data in LBS, and in the process, make the following contributions (Fig. 2):

- We present an overview of LBS and highlight the privacy issues that may arise from the use of location data in LBSs.
- We survey and discuss DP-based solutions (geo-indistinguishability, private spatial decomposition, and local differential privacy) that are designed or can be used for the privacy protection of location data in LBSs.
- We explore the applicability of DP-based schemes for protecting location privacy in different location data processing, collection, and publishing scenarios in LBSs.
- Finally, we highlight future research directions that require further investigation.

Owing to the growing popularity of DP in private data processing environments, a number of comprehensive surveys tracking its progress have been conducted over the years (Dwork, 2008; Dwork and Smith, 2010; Leoni, 2012; Wang et al., 2015a; Xiong et al., 2020; Yang et al., 2020; 2017; Ye and Hu, 2019; Zhu et al., 2017). Additionally, there are numerous previous survey articles that focus on usage of DP in specific application domains such as health care (Akgun et al., 2015; Dankar and Emam, 2012; 2013; G-Divanis et al., 2014), cyber physical systems (Hassan et al., 2019), internet of vehicles (Zhao et al., 2019b), big data (Jain et al., 2018; Yao et al., 2012), machine/deep learning (Ji et al., 2014; Zhao et al., 2019a), and distributed computing systems (Goryczka and Xiong, 2017; Rao and Bertino, 2019). However, to the best of our knowledge, this is the first study to thoroughly survey the existing DP algorithms from the perspective of LBS. Particularly, this survey focuses on protecting the privacy of users in scenarios involving location-based data, such as location-based query processing, location data collection, and location data publishing and investigates the applicability of DP algorithms to these scenarios to provide protection to safeguard the users from location disclosure.

1.3. *Paper organization*

We organize the rest of the paper as follows: Section 2 presents an overview of LBS and highlights the privacy issues from the

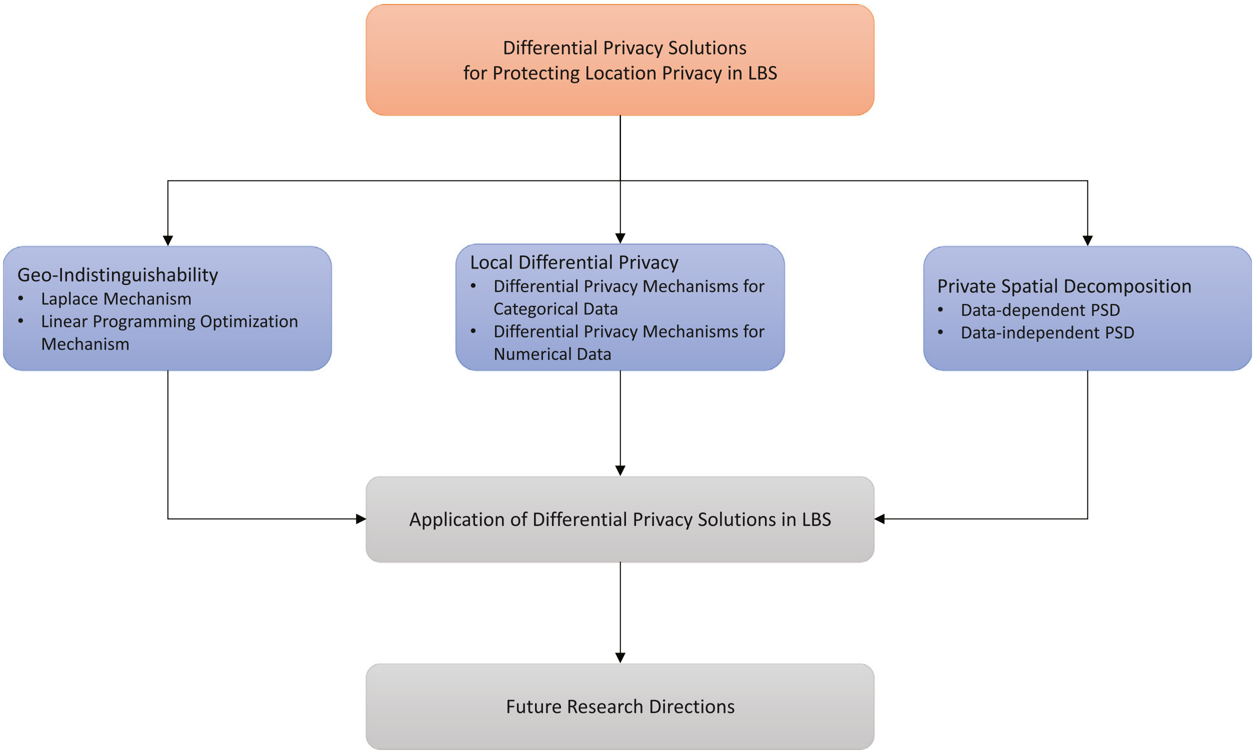

Differential Privacy Solutions
for Protecting Location Privacy in LBS
Geo-Indistinguishability
Laplace Mechanism
Linear Programming Optimization Mechanism
Local Differential Privacy
Differential Privacy Mechanisms for Categorical Data
Differential Privacy Mechanisms for Numerical Data
Private Spatial Decomposition
Data-dependent PSD
Data-independent PSD
Application of Differential Privacy Solutions in LBS
Future Research Directions

Fig. 2 – Taxonomy of the contributions in this survey.

use of LBS. In Section 3, we present DP and its properties. In Sections 4, 6, and 5, we survey the variants of DP that are designed or can be used for privacy-preserving location data processing in LBSs. Section 7 contains the discussion on the applicability of DP based techniques in different data processing environments. Then, Section 8 highlights future research directions and in Section 9, we conclude the paper.

2. Location-based services

In this section, we provide an overview of the different architectures of LBS, and then present the existing methods to mitigate location privacy threats in LBSs.

2.1. Architecture

The architecture of LBS can be broadly classified into two categories: trusted and untrusted LBS providers.

In the scenario with trusted LBS providers, which is depicted in Fig. 3(a), the users who want to receive services adjusted to their current locations send their exact current locations with the service request to the LBS server without worrying about disclosure of personal information. However, in a real application environment, assuming that such trusted LBS providers exist, is unrealistic.

The untrusted LBS provider scenario, which is a realistic scenario in practice, is further categorized into two depending on the presence of the trusted server (i.e., data curator) that is located between the users and the LBS provider (Terrovitis, 2011). In the untrusted LBS provider scenario with trusted server, users communicate with the LBS provider through the trusted server (Fig. 3(b)). More specifically, the trusted data curator receives the location-aware service request from the users, perturbs (or anonymizes) it, and forwards the perturbed (or anonymized) request to the LBS server. In contrast, in the untrusted LBS provider scenario without trusted server, which is depicted in Fig. 3(c), users directly communicate with the LBS provider; therefore, they are responsible for perturbing (or anonymizing) their location data before forwarding them to the LBS server.

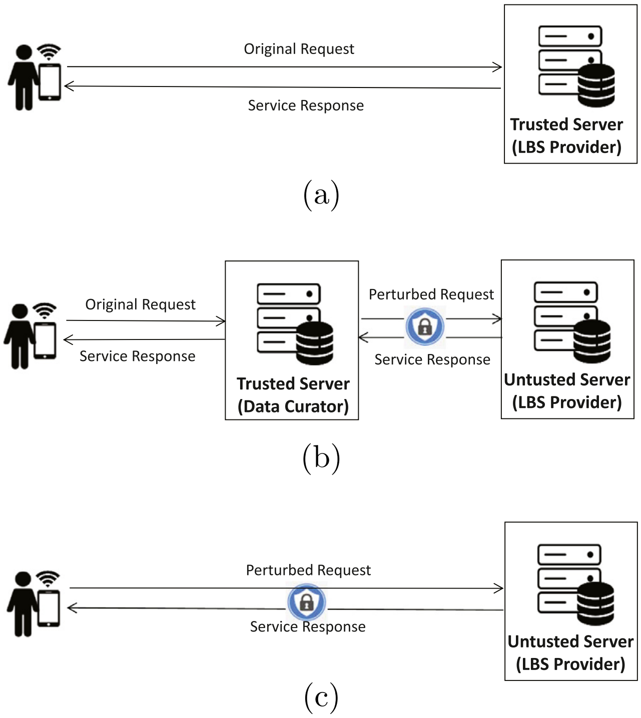


Fig. 3 – The different architectures of LBS: (a) trusted LBS server (LBS provider), (b) untrusted LBS server (LBS provider) with trusted server (data curator), and (c) untrusted LBS server (LBS provider).

2.2. Privacy-Preserving location-based services

Several techniques are being used to mitigate location privacy threats in LBSs such as conventional anonymization techniques, dummy location approaches, and more recent techniques such as cryptography and perturbation mechanisms. In this subsection, we briefly summarize existing studies on protecting location privacy in LBSs.

Anonymization technique: Solutions in this category can be further classified into generalization-based and mix-zone approaches. In spatial cloaking (Gruteser and Grunwald, 2003), which is one of the most popular generalization-based techniques, the LBS user first sends the exact location to the trusted server located between the users and the LBS server. Then, the trusted server computes a broad region (i.e., the number of users who exist in this broad region is greater than or equal to k, thereby achieving k-anonymity), termed as the cloaking area, and reports it to the LBS server instead of reporting the exact location. The mix-zone approaches are recognized as an effective technique for preventing continuous exposure of location information. These approaches break the continuity of location exposure by establishing a specific area, called a mix-zone, where an adversary cannot trace users' movements (Beresford and Stajano, 2003). The k-anonymity is achieved by enforcing that inside mix-zones at least k users change pseudonyms together such that the mapping between old and new pseudonyms of users are not revealed. The concept of this anonymization technique has been successfully applied to location data. However, it is well known that anonymization-based approaches usually fail to guarantee adequate levels of privacy (Ohm, 2010).

Dummy location technique: In this scheme, an LBS user dispatches a service request along with multiple location points instead of a single one to an LBS (Kido et al., 2005; You et al., 2007). For this purpose, the user first needs to generate a set of fake locations that are different from their true location, but are located within a close distance. The basic idea is that, among the set of answers provided by the LBS, one correct answer exists for the user's actual location. Furthermore, the LBS, which receives multiple location points, is unable to exactly determine the user's position. The fake location points are randomly generated by the user's mobile device; therefore, this technique does not require a trusted server. Furthermore, this approach is known to achieve adequate levels of privacy (Liu et al., 2019).

Cryptography technique: Solutions in this category rely on encryption schemes to protect user locations (Mascetti et al., 2011; Narayanan et al., 2011; Popa et al., 2011); therefore, they can achieve the highest level of privacy guarantee. For example, in (Mascetti et al., 2011), which proposed a method for proximity notifications of nearby friends without exposing the current location to the LBS server, each user reported their encrypted current location to the LBS server. Consequently, the user's true location is not revealed to an unauthorized LBS server. The main drawback of cryptography-based mechanisms is that they require expensive computational overhead on low-end mobile devices of the users.

Perturbation technique: Most of the solutions in this category work either by deliberately reducing the precision of the location information or adding carefully designed random noise to the true location (Liu et al., 2019; Primault et al., 2018). For example, Micinski et al. (2013) studied the effect of coarsening the location information sent to the app. Gutscher (2006) proposed an approach to protect location privacy by obfuscating the location coordinates by using coordinate transformation. The challenge in perturbation-based techniques is to achieve a tradeoff between the level of privacy and data utility. More specifically, a large perturbation to the original data enforces a stronger privacy guarantee; however, it introduces a larger loss in accuracy. It should be noted that the DP-based methods surveyed in this paper belong to this category

3. Overview of differential privacy

DP assumes the existence of a centralized trusted aggregator (or curator) located between data contributors (i.e., data owners) and data users. DP is used in two different settings: non-interactive and interactive settings. In the non-interactive setting, a trusted aggregator collects original data from individual data owners, computes aggregate statistics by using the collected data, and publishes the perturbed aggregate statistics, which are obtained by adding random noise to the true aggregate statistics, to data users for data analysis (Li et al., 2014b; Machanavajjhala et al., 2008; Xiao et al., 2011b). In the interactive setting, the trusted curator receives a query from a data user, computes the true result of the user query computed over original database, perturbs the true result by adding random noise to it, and returns the perturbed result to the data user (McSherry, 2010; Peng et al., 2013; Xiao et al., 2011a). DP is defined as follows:

Definition 1. (ϵ-DP) *A randomized algorithm $\mathcal{A}$ satisfies ϵ-DP, if and only if for (1) any two neighboring datasets, D_1 and D_2, and (2) any output O of $\mathcal{A}$, the following is satisfied:*

$$\Pr[\mathcal{A}(D_1) = O] \le e^{\epsilon} \times \Pr[\mathcal{A}(D_2) = O].$$

Here, two datasets, D_1 and D_2, are considered as neighboring, if and only if they exactly differ in one record. The above definition implies that, given any output of $\mathcal{A}$, an adversary who has arbitrary background knowledge cannot distinguish with high confidence (which is controlled by ϵ) if the input of $\mathcal{A}$ is D_1 or D_2. Here, the parameter ϵ is a privacy budget, which controls the level of privacy. That is, smaller (larger) values of ϵ ensure a stronger (weaker) privacy guarantee but introduce larger (smaller) noise to the true result.

Table 1 – Example of credit card payment data for subway fares.

Transaction ID	Origin Station	Destination Station	Time
1001	s1	s2	13:02
1002	s1	s2	13:15
1003	s1	s4	14:28
1004	s2	s1	14:55
1005	s2	s1	15:21
1006	s1	s2	15:30

Definition 2. (Global sensitivity) *Given any two neighboring datasets D_1 or D_2, the global sensitivity of a query $\mathcal{F}$ is defined as:*

$$\Delta\mathcal{F}_{gs} = \max_{D_1, D_2} |\mathcal{F}(D_1) - \mathcal{F}(D_2)|.$$

Global sensitivity denotes the maximum difference in the query result caused by one record difference between any two neighboring datasets. For instance, the global sensitivity of count query is one because a particular record may or may not satisfy the query predicate.

DP is typically implemented by using the Laplace mechanism (Dwork, 2006; Dwork et al., 2006) or the exponential mechanism (McSherry, 2009).

- *Laplace Mechanism:* Let $\mathcal{F}(D)$ denote a query result from dataset D. The Laplace mechanism satisfies ϵ-DP if a random noise sampled from a Laplace distribution with mean $\mu = 0$ and scale $b = \frac{\Delta\mathcal{F}_{gs}}{\epsilon}$ is added to $\mathcal{F}(D)$ as follows:

  $$\mathcal{F}(D) + Lap\left(\frac{\Delta\mathcal{F}_{gs}}{\epsilon}\right)$$

  In other words, ϵ-DP is achieved by adding noise randomly drawn from the Laplace distribution, which is calibrated using the privacy budget and the global sensitivity, to the true result.

  For example, Table 1 presents an example of credit card payment data for subway fares. Let us further assume that an analyst who wants to count the number of transactions associated with an itinerary from “s1” to “s2” submits a count query to a trusted curator. In this example, there are three transactions (i.e., 1001, 1002, and 1006) associated with the itinerary from “s1” to “s2”. The global sensitivity of the count query corresponds to one; thus, through the Laplace mechanism, the trusted curator returns $3 + \eta$ (where η is drawn from a Laplace distribution with the mean $\mu = 0$ and scale $b = \frac{1}{\epsilon}$) to the analyst.
- *Exponential Mechanism:* The exponential mechanism is used in the case of non-numerical query outputs. Let D be a dataset and O be a set of possible outputs of a query $\mathcal{F}$ over the dataset D. A score function $s : D \times O \to \mathbf{R}$ maps a dataset/output pair to a real-valued score. For a score function s, a mechanism that outputs $o \in O$ with a probability that is proportional to $exp(\frac{\epsilon \times s(D,o)}{2 \times \Delta\mathcal{F}_{gs}})$ satisfies ϵ-DP. In other words, the exponential mechanism first assigns a score to possible query results using a score function in a way that an output with a higher score indicates that it is closer to the true output. This mechanism, then, randomly selects

Table 2 – Example of exponential mechanism with the dataset in Table 1.

Route	Frequency	Probability of being selected as a result		
	in Table 1	$\epsilon = 0.01$	$\epsilon = 0.1$	$\epsilon = 1.0$
s1 → s2	3	0.335	0.350	0.507
s2 → s1	2	0.333	0.333	0.307
s1 → s4	1	0.332	0.317	0.186

an output from the possible query result set such that the higher the score, the more appeals the result.

For instance, let us consider a scenario in which an analyst wants to know “What is the most frequently traveled route?” and submits a query to the trusted curator. Table 2 presents an example of an exponential mechanism, in which three different epsilon values, $\epsilon = 0.01$, $\epsilon = 0.1$, and $\epsilon = 1.0$, are used to compute the probability that each route is selected as a query result. For example, the route from “s1” to “s2” (i.e., s1 → s2) appears three times in Table 1; thus, when ϵ is set to 0.1, the probability that route s1 → s2 is selected as a query result is proportional to $exp(\frac{0.1\times3}{2\times1})$. As presented in the table, route s1 → s2 has the maximum probability of being selected as a result, whereas route s1 → s4 has the minimum probability. As ϵ increases, the gap between the maximum and minimum probabilities increases, resulting in a higher utility but a lower privacy level. In contrast, as ϵ decreases, the gap between the maximum and minimum probabilities decreases, which enforces a stronger privacy guarantee, but leads to a lower utility.

DP has two important properties, sequential and parallel compositions, which have to be considered when deploying it in real-world scenarios.

Theorem 1. *(Sequential Composition) Let assume that a randomized algorithm $\mathcal{A}_i$ provides ϵ_i-DP. The sequence of $\mathcal{A}_i(D)$ over same dataset, D, provides $(\sum_i \epsilon_i)$-DP.*

Theorem 2. *(Parallel Composition) Let assume that a randomized algorithm $\mathcal{A}_i$ provides ϵ_i-DP. Let D_i be a disjoint subset of dataset, D. The sequence of $\mathcal{A}_i(D_i)$ provides $\max_i(\epsilon_i)$-DP.*

In particular, the sequential composition property of DP is useful when iteratively running an algorithm over same dataset multiple time. For example, let us consider a scenario where we need to run the algorithm n times over same dataset. Then, in this case, given an available privacy budget ϵ, we can partition ϵ into n smaller privacy budgets, $\epsilon_1, \epsilon_2, \cdots, \epsilon_n$, such that $\epsilon = \sum_{i=1}^{n} \epsilon_i$ and make the i-th running of the algorithm satisfy ϵ_i-DP by consuming the privacy budget ϵ_i. In the context of LBS, the sequential composition property of DP can be usefully used for periodically collecting location data to track movements of the users over time.

Although DP has been used in diverse areas with its strong privacy guarantees, it can not be directly used to protect location privacy of users in LBSs. Thus, in the next subsequent sections, we survey three DP-based schemes, geo-indistinguishability (GeoInd), private spatial decomposition (PSD), and local differential privacy (LDP), that can be applied to location data of users.

4. Geo-Indistinguishability

GeoInd, which was first introduced in Andres et al. (2013), extends DP with a distance metric to provide a privacy-preserving mechanism for location data. The basic idea behind GeoInd is to add random noise to the user’s actual location in such a manner that an adversary, who has arbitrary background knowledge, cannot infer the user’s location with high confidence. GeoInd is formally defined as follows:

Definition 3. (Geo-Indistinguishability) *Let $\mathcal{X}$ be a set of possible user locations and $\mathcal{Z}$ be a set of reported locations, respectively (it is often assumed that $\mathcal{X}$ is equal to $\mathcal{Z}$). Let K be a randomized mechanism that probabilistically generates an obfuscated location, given an actual user location. Then, a randomized mechanism, K, satisfies ϵ-GeoInd, if and only if for (1) all $x, x' \in \mathcal{X}$ and (2) any output location, $z \in \mathcal{Z}$, the following equation is satisfied:*

$$K(x)(z) \le e^{\epsilon\cdot d(x,x')} \times K(x')(z),$$

where $d(x, x')$ corresponds to the distance metric (e.g., Euclidean distance between two locations x and x').

This definition denotes that, given a reported location, z, the ability of an adversary to identify whether the actual location of a user is x or x' is limited by the privacy budget (i.e., ϵ) and the distance between x and x'. This implies that the closer two locations are, the more indistinguishable they are. A practical interpretation of the above definition provided by Chatzikokolakis et al. (2015) is as follows. For all locations x' within radius r from x, a user enjoys ϵr-indistinguishability within r. This implies that the user’s level of privacy is proportional to the radius, r; the smaller (larger) the value of r is, the higher (lower) is the level of privacy. It should be noted that, unlike the DP described in Section 3, GeoInd does not require a trusted server because the data perturbation is performed on the user side.

4.1. *Differential privacy mechanisms of geoind*

In this subsection, we briefly summarize existing privacy mechanisms to realize GeoInd.

4.1.1. *Laplace mechanism*

A simple mechanism used to realize GeoInd is the planar Laplace (PL) mechanism (Andres et al., 2013). The main idea of the PL mechanism is to perturb the real user location, x, by adding random noises drawn from a 2-dimensional Laplace distribution centered at x with density defined as:

$$D_\epsilon(x, z) = \frac{\epsilon^2}{2\pi} e^{-\epsilon d(x,z)}$$

Drawing a random noise, which is expressed as a vector in polar coordinates, from this distribution is performed as follows:

1. Draw an angle, θ, uniformly in $[0, 2\pi)$,
2. Draw p uniformly in [0,1), and
3. Select a radius, r, such that $r = C^{-1}(p)$, where $C^{-1}(p) = -\frac{1}{\epsilon}(W_{-1}(\frac{p-1}{\epsilon}) + 1)$ and W_{-1} is the Lambert W function, (−1 branch).

Once the random vector is generated in polar coordinates, the perturbed location is computed as $z = x + \langle r\cos(\theta), r\sin(\theta)\rangle$. The PL mechanism provides a simple and efficient method to achieve ϵ-GeoInd. However, it is known that this approach may introduce large noise, resulting in a lower utility.

4.1.2. *Optimization mechanism*

Although the aforementioned PL mechanism is simple and efficient, its utility is not always guaranteed to be optimal. To address this issue, Bordenabe et al. (2014) developed a mechanism that can provide the maximum utility while preserving GeoInd. Let us assume that a set of possible user locations, $\mathcal{X}$, and a set of reported locations, $\mathcal{Z}$, are finite and small in size. Given a privacy metric, $d_\mathcal{X}$, a quality metric, $d_\mathcal{Q}$ (which represents the quality loss by reporting z when the real location is x), and the prior probability distribution on the user's possible locations, $\pi_\mathcal{X}$, an optimal mechanism, K, can be obtained by solving the following linear programming problem:

$$\begin{array}{lll}
\textit{min}: & \sum_{x\in\mathcal{X}, z\in\mathcal{Z}} \pi_\mathcal{X} \cdot K(x)(z) \cdot d_\mathcal{Q}(x, z) & \\
\text{s.t.}: & K(x)(z) \le e^{\epsilon \cdot d_\mathcal{X}(x,x')} \times K(x')(z) & x, x' \in \mathcal{X}, z \in \mathcal{Z} \\
 & \sum_{z\in\mathcal{Z}} K(x)(z) = 1 & x \in \mathcal{X} \\
 & K(x)(z) > 0 & x \in \mathcal{X}, z \in \mathcal{Z}
\end{array}$$

Here, $d_\mathcal{X}$ and $d_\mathcal{Q}$ are typically computed as the Euclidean distance between two locations. Furthermore, the prior probability distribution is usually defined as $\pi_\mathcal{X} = \frac{1}{|\mathcal{X}|}$. The mechanism, K, generated by solving the above optimization problem is guaranteed to satisfy ϵ-GeoInd, while providing optimality guarantees for utility.

The abovementioned linear optimization problem can be solved using several techniques. However, considering that the number of constraints of this linear optimization problem is proportional to $O(|\mathcal{X}|^3)$ (by assuming that the two sets, $\mathcal{X}$ and $\mathcal{Z}$, coincide), solving this optimization problem is feasible only when $\mathcal{X}$ is small. In Bordenabe et al. (2014), an approximation technique that can significantly reduce the number of constraints in a linear program is proposed. The main idea of the approximation technique is to first build a spanner graph and then consider the GeoInd constraints for every edge in the spanning graph, instead of considering them for any par of two locations.

In Chatzikokolakis et al. (2017), the authors proposed an optimal mechanism that is constructed on a coarser grid, $\mathcal{X}_c$, to reduce the cardinality of $\mathcal{X}$. The coarser grid, $\mathcal{X}_c$, comprises larger cells that are created by merging several locations of $\mathcal{X}$, and a user location, $x \in \mathcal{X}$, is mapped to the cell of the coarse grid, where x is located. Although an optimal mechanism constructed on a coarse grid reduces computational overhead, it incurs relatively higher quality loss compared with those in Andres et al. (2013) and Bordenabe et al. (2014).

To alleviate the computational overhead of the optimal mechanism, Ahuja et al. (2019) developed a multi-step algorithm that recursively applies the optimal mechanism along a multi-level index structure. The multi-level index structure comprises several grids such that grids closer to the top level are coarser than those lower in the index structure. The multi-level index structure enables effective pruning of the search space while seeking an optimal solution, and consequently, achieves high computational efficiency.

4.1.3. *Comparative summary*

A comparative summary of the aforementioned GeoInd mechanisms is presented in Table 3 on the parameters: focus of the work, main technique used, computational demand, and quality loss. As explained previously, the mechanisms can be broadly categorized into Laplace and optimization techniques. In the Laplace technique, the noise is directly injected to obfuscate the actual location of the user. Therefore, it has low computation demand but high quality loss. On the contrary, optimization techniques involve solving linear programming problems to generate a noise generation mechanism that optimizes utility at a given privacy level. Hence, compared with the Laplace technique, the computation demand of the optimization techniques is expensive; however, their quality loss is low.

4.2. *Application of geo-Indistinguishability*

By abstracting from adversaries' side information and requiring no trusted servers for perturbation, GeoInd achieves better privacy and optimal utility. Therefore, it has provided a promising future for privacy provision in several areas. Here, we present some application areas of GeoInd.

4.2.1. *Spatial crowdsourcing*

Spatial crowdsourcing (SC) is a platform in which individuals, groups, and communities are engaged to collect, analyze, and disseminate their surrounding information. The ubiquity and power of smartphones richly equipped with sensors have been instrumental in the rise of this transformative platform. In SC, individual users are considered as workers. When a new task (e.g., traffic monitoring, air quality monitoring, etc.) arrives, the platform allocates the task amongst the workers within a specified area (or who are willing to travel to the specified area) for execution. Hence, there is a demand for the workers to share their locations, which violates the participants' location privacy and can be a deterrent factor in workers' willingness to take part in executing tasks. Several studies have proposed the use of GeoInd to obfuscate the workers' locations, i.e., instead of the worker revealing her/his actual location, she/he shares a perturbed location. In Wang et al. (2017a), proposed a differential geo-obfuscation mechanism to protect the actual location of the

Table 3 – Comparative summary of GeoInd mechanisms.

Ref No.	Target	Technique	Quality Loss	Comp. Demand
(Andres et al., 2013) (Laplace mechanism)	Location obfuscation	Planar laplace	High	Low
(Bordenabe et al., 2014) (Optimization mechanism)	Optimize tradeoff between privacy and quality loss	Linear optimization	Low	High
(Chatzikokolakis et al., 2017) (Optimization mechanism)	Enhance utility in a larger number of locations	Remap and coarse grid mechanisms	Medium (for remap) and High (for course)	Low - Medium
(Ahuja et al., 2019) (Optimization mechanism)	Improve utility and performance	Multi-step algorithm used alongside spatial index	Medium	Medium

workers during task assignment by an SC platform. To minimize the expected distance traveled by the selected task executors, the scheme leverages a mixed-integer nonlinear program(MINLP). In Jin et al. (2019); Wang et al. (2018d), GeoInd is utilized to create personalized privacy for participants during task allocation in crowdsensing, i.e., the task allocation server takes into consideration the preferred privacy levels of the bidding participants before assigning them tasks. Yan et al. (2019) designed a crowdsensing framework that takes into consideration the privacy of the workers' mobility. In the framework, a worker's shortest path from the source to destination is protected using the GeoInd mechanism and she/he is bounded from deviating from this path in a privacy-preserving manner. In Wang et al. (2015b), Wang *et al.* explored privacy-preserving crowdsensing aggregated data reporting in the form of aggregated histograms. In their scheme, each worker u_i observes data d_i at location l_i, and reports a sanitized (l_i, d_i) to an aggregator. The sanitizations are performed locally and independently by the workers using the GeoInd mechanism. The aggregator then generates statistical information from the sanitized tuples received from all the workers. Tehrani et al. (2018) implemented a spatial crowdsourcing system for emergency situations and adopted GeoInd to protect the location information of the workers.

With the emergency of intelligent vehicles, several studies have suggested the use of vehicles as workers in spatial crowdsourcing (leading to the birth of the term, *vehicle-based spatial crowdsourcing*) (Ou et al. 2015; Wang et al. 2016; Wu et al. 2013). In Qiu and Squicciarini (2019), location privacy protection in vehicle-based spatial crowdsourcing is investigated. The authors realized that directly applying GeoInd to protect the locations of vehicles participating in crowdsourcing leads to a loss in quality of service. This is because, privacy and quality of service in GeoInd depend on the Euclidean distance between the actual and obfuscated locations, which is only measurable in the 2D plane. Thus, the Euclidean distance that is only measurable in straight lines cannot be suitable in vehicle-based spatial crowdsourcing. In the same work, the authors then addressed this challenge by modeling the road map as a weighted directed graph, with the tasks' and participants' locations as points on the graph. The location obfuscation is achieved through probabilistic distribution over the graph. Reported results achieve an optimal quality of service.

4.2.2. *Vehicle networks*

Vehicle networks are gaining popularity as a promising method to improve driving experiences and road safety. Because of the development of IoT, vehicles are equipped with sensors and internet access which are then used to gather and receive information such as vehicle operational and configuration data, and vehicle location and speed (Pudar et al., 2014; Zhang et al., 2020). The recipients of these information can include service centers and other vehicles (Pudar et al., 2014). For example, the service center can use the information to track the vehicle in cases of it being stolen. Some information can be used to provide hands-free calling, traffic collision update, road condition update and audio services for the driver, and keep him/her better informed for safe navigation. Thus, in this perspective, location information is a very vital parameter for service provision and should be guarded at all cost from compromises. In Zhou et al. (2019) introduced edge-assisted vehicle networks to improve the quality of service in the vehicle networks, and to provide location privacy in the proposed architecture. The authors proposed the deployment of GeoInd at the edge nodes. In this case, a vehicle first submits its location to an edge node at which the privacy preservation mechanism to protect the actual location of the vehicle is executed. Next, the query with the protected location information is then submitted to the service provider which returns the required service to the requesting vehicle through the edge node.

4.2.3. *Ride-sharing*

The improved vehicle occupancy through the sharing of unoccupied seats has led togain in popularity of ride-sharing service, i.e., there is an increasedpularity in the trend of people traveling through the same route at the same time sharing the same car and travel expenses. This has additional benefits of reduced traffic congestion, energy conservation and environmental conservation (Tong et al., 2017). With the emergency of many ride-sharing applications such as Uber, Waze, and Lyft, there has been a steady increase in the number of ride-sharing users, as demonstrated by the 2 billion bookings for Uber in 2016 (Bugador, 2019). To a larger extent, this rapid development in ride-sharing services is attributed to the developments in mobile techniques, GPS techniques, and cloud computing (Dong et al., 2018). Here, users of ride-sharing applications need to share their location information (e.g., boarding/current location and destination location) to properly benefit from the service. However, location information is sensitive and if not well managed, it can be used for unethical reasons. To address the location privacy challenge, in Tong et al. (2017) proposed a scheduling scheme that uses DP to protect the location information of ride-sharing users while scheduling them to minimize the travel distance of the vehicles. To further improve the privacy, the authors enhanced their scheme by constructing spatial indexes for road networks and then adopted the GeoInd mechanism when querying the grid distance matrix. This way, the actual location information of the users gets protected from adversaries. Shi et al. (2019) proposed a route scheduling scheme for ride-sharing services based on deep reinforcement learning algorithm and in particular, deep Q-network. The authors considered a scenario of ride-sharing vehicles roaming around searching for passengers. In the study, a service center guides the vehicle to an area with a high likelihood of getting a passenger (after computing strategies with the deep Q-network algorithm) and the vehicle keeps updating the service center about its occupancy. When the vehicle is occupied, the information sent to the service center contains the location of the passenger. The service provider can store this information for future use e.g., for future training of the deep Q-network model. To protect the location information from disclosure, GeoInd is employed, i.e., instead of reporting a passenger's actual location x to the service center, an obfuscated location $z = x + \langle r\cos(\theta), r\sin(\theta)\rangle$ is reported. The noise $\langle r\cos(\theta), r\sin(\theta)\rangle$ added to x is generated using the Laplace mechanism of GeoInd described earlier in this section.

4.2.4. *Critical infrastructure networks*

Critical infrastructure networks (CIN) such as electrical power grids and public transportation networks, which are heavily reliant on cyber-physical systems, play a vital role in socio-economic well-being (Fioretto et al., 2019). A lot of the operations of these networks rely on machine learning mechanisms. Thus, like any other machine-learning dependent system, any research on these networks requires massive amounts of data, some of which might be sensitive and can reveal extra information about an entity (e.g., electrical loads, locations, etc.). To address the privacy challenges, Fioretto et al. (2019) proposed a privacy-preserving obfuscation scheme for CINs. The proposed scheme leverages location-indistinguishability (similar to GeoInd) to protect the locations of the cyber-physical components of CINs, and the conventional DP to hide the sensitive values generated by CINs.

4.2.5. *Location-aware social networks*

As the ubiquity of smartphones grows, there is an increasing popularity of social networks which are capable of providing location based services. Amongst the popular services provided, is the location based social discovery (LBSD) service. LBSD service provides users with a list of nearby people (friends and strangers) and their respective distances. The user can use the provided list in an attempt to befriend the strangers or meet with the nearby friends (Ma and Chen, 2014; Pan et al., 2016; Xue et al., 2015). Although location-aware social networks help people get connected, we should not ignore the location privacy of the users. That is because tracking a user's locations can easily leak her/his daily movements to both intended and unintended people (Aronov et al., 2018). To protect location privacy in location-aware social networks, (Ma and Chen, 2014) combined GeoInd with homomorphic encryption for privacy-preserving nearby friend discovery. The integration of homomorphic encryption algorithm enables users to compute distances between each other without leaking their locations to the server. In Huang et al. (2016), Huang *et al.* proposed a privacy-preserving proximity test for a group of nearby persons to discover each other using a fast scale product technique (Lu et al., 2014) and GeoInd. In their work, each user periodically sends her/his encrypted-perturbed locations to the service provider for storage. Each user can then request the server to execute the proximity test mechanism and the server returns the list of people within the required distance to the requesting user.

4.2.6. *Proximity services*

With the growing popularity of handheld devices, location tracking and location-specific services are becoming more common. For accuracy, different environments (e.g., outdoor, indoor, etc.) demand different location detection techniques (e.g., GPS, GSM, Wi-fi, etc) (Roxin et al., 2007). Mobile phones are equipped with location detection capabilities that enable them to provide location-based services such as proximity to specific vendors, notification of safety issues, and closeness to POI (e.g., restaurants, banks, schools). However, these services require users to reveal their locations during service requisition. Location exposure through techniques such as tracking identification, and profiling threats (Ateniese et al., 2015; Chen et al., 2015; Fawaz et al., 2015), have posed serious privacy challengesward these services. Eltarjaman et al. (2017) proposed a location privacy-preserving algorithm that generates the most prominent POI sets through operations limited to the user device, i.e., there is no need for the client device to send location information to the service provider, and the service provider is simply a data source. The main idea in this work is, first, the client defines an area of interest (e.g.,$1000km^2$) within her/his vicinity and sends the coordinates of the defined area together with a keyword to the server. Next, the server identifies the POIs within the defined area and returns their location and prominence value (e.g., reviews) to the client. Based on the returned prominence values, the client then ranks the returned POIs locally. To prevent inversion attacks, the authors proposed the return of POIs beyond the defined area and perturb the returned location information using the GeoInd mechanism.

In inversion attacks, the attacker uses his/her knowledge of the cloaking algorithm and simulates the application to identify candidate locations in the cloaked region (Dewri et al., 2010; Jagwani and Kaushik, 2017). As a general example, suppose that an attacker is able to observe a request for a cloaked region. Also, suppose that the attacker knows the identity of six potential issuers of the request. But, he is unable to identify who among the six is the actual issuer since cloaking is applied to ensure 6-anonymity. If the attacker has knowledge of the cloaking algorithm, he/she can simulate his/her application to the location of each candidate and eliminate those whose cloaked region differs from the observed request. However, with obfuscated location information, the attacker's job of obtaining the actual location information of a target is made more difficult.

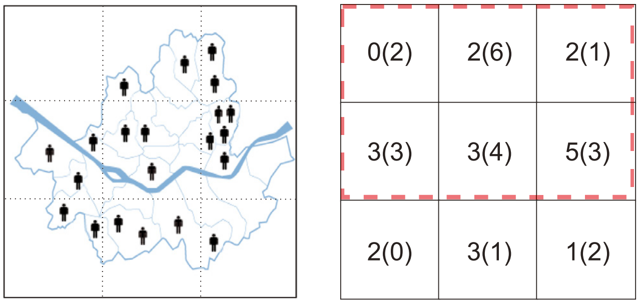
5. Private spatial decomposition

The semantic definition of DP guarantees that an adversary is unable to distinguish with high confidence whether a particular individual is included in the dataset published by a data publisher. This property of DP is especially useful in publishing aggregated population statistics (e.g., histograms), and accordingly, DP has recently become a de facto standard for privacy-preserving histogram publication.

Additionally, DP can be used for publishing location data by using a spatial histogram, where a spatial domain is partitioned into several cells, and each cell contains the information of the number of objects located within the corresponding cell. Recent studies on publishing DP-complaint spatial histograms have been conducted under the name of PSD (Cormode et al., 2012; Hay et al., 2010; Kim et al., 2018a; Qardaji et al., 2013; To et al., 2015; Xiao et al., 2010). PSD methods first partition a spatial domain into several cells and then add carefully calibrated noise to the true count of objects located within the boundaries of each cell. Then, the perturbed spatial histogram that can be used to answer the number of objects within a certain region (e.g., range queries), is released for public use. For example, Fig. 4 represents a spatial domain and the corresponding PSD. This PSD can then be used to estimate the number of objects within a certain region. For example, let us consider the region highlighted by a dotted rectangle in Fig. 4. Then, the number of objects in this region is estimated as (2 + 6 + 1 + 3 + 4 + 3) = 19 using this PSD.

While estimating the number of objects within a certain region (i.e., answering range queries) using PSD, two types of errors can occur.

- *Perturbation error:* This error is caused by the difference between the actual and the perturbed counts. As explained earlier, a standard approach to achieve DP is the Laplace mechanism, which adds random noise to the true value. Given a range query, the perturbation error depends on the number of cells that are included within the query region. For example, Fig. 5 shows two different space decompositions for the same spatial domain that comprises 16 location data. To satisfy ϵ-DP, an independently generated noise sampled from the Laplace distribution with a variance $\sigma = 2(\Delta\mathcal{F}_{gs}/\epsilon)^2 = 2(1/\epsilon)^2$, is added to each true cell count. It must be noted that the global sensitivity of a range query is 1. These noises are independently generated; therefore, the variance of the perturbation error is proportional to the number of cells included within the query region of a range query. Assume a range query that includes the entire spatial domain. The PSD in Fig. 5(a) will provide the result with an error variance of 4σ, whereas that in Fig. 5(b) is 16σ. Therefore, the more the number of cells contained in a query range and the finer the granularity of a PSD, the larger the perturbation error.
- *Non-uniform error:* This error occurs owing to the cells that are partially included in the query region of a range query. A straightforward solution for the partially included cells is the uniform distribution assumption method that assumes that location data in each cell are uniformly distributed. Let us consider the example in Fig. 6, where most of the location data exist in the right part of this cell. Given a range query, which is represented by a dotted red rectangle, one-half of the cell overlaps the query region. Based on the uniform distribution assumption, the answer to the range query is estimated as $\frac{(16+X\sim Lap(1/\epsilon))}{2}$. However, this estimated count significantly differs from the real count, 2. In other words, the non-uniform error can increase when the location data in a cell is not uniformly distributed.



0(2)	2(6)	2(1)
3(3)	3(4)	5(3)
2(0)	3(1)	1(2)

Fig. 4 – An example of a spatial domain and the corresponding PSD: The number in each cell represents the actual (perturbed) number of people located in that cell.

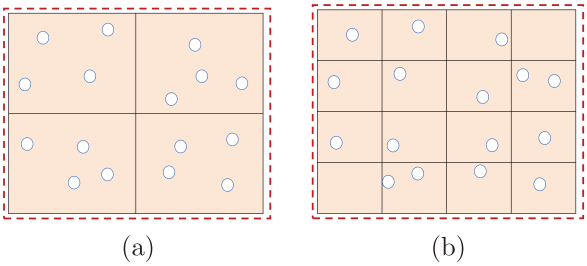


Fig. 5 – Two different PSDs for the same spatial domain: (a) 2 x 2 grid PSD and (b) 4 x 4 grid PSD.

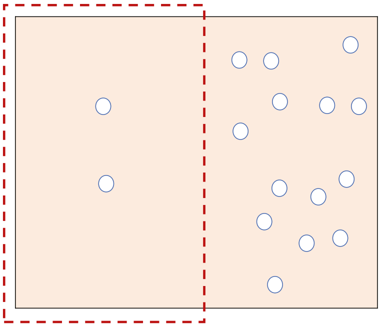


Fig. 6 – A PSD cell that contains location data with a skewed distribution.

In this section, we first review various PSD techniques and then summarize several key application areas of PSD.

5.1. *Differential privacy mechanisms of PSD*

Existing PSD approaches can be classified into two categories: (i) data-independent methods that partition the spatial domain without considering the distribution of data and (ii) data-dependent methods that partition the spatial domain based on the distribution of data.

5.1.1. *Data-dependent PSD*

Data-dependent PSD performs a series of split operations determined by the underlying data. This situation incurs a challenge because simply adding noise to the cell counts is not sufficient to guarantee DP. In other words, the split operations also have to be differentially private to preserve the location privacy of users included in the data-dependent PSD. The location of users in this line of work is used for both determining splitting points and computing noisy counts; therefore, the privacy budget should be distributed among both procedures. Consequently, data-dependent PSD techniques can better capture the distribution of location points, and they are expected to be more balanced than data-independent PSD methods. On the contrary, the accuracy of data-dependent PSD methods largely depends on the data distribution and the parameters related to spatial indexing structures.

Xiao et al. (2010) proposed a method to publish a multidimensional histogram for answering random count queries in an interactive interface. Specifically, the authors first proposed a cell-based partitioning method as a baseline strategy that splits on every domain and adds Laplace noise to the count of each partition. Each cell is completely enclosed by range queries— this allows the cell-based partitioning method to introduce perturbation error only. However, if a range query covers multiple cells, then the perturbation error is aggregated, which makes the query result unacceptably inaccurate. Accordingly, the authors further proposed a partitioning method based on a kd-tree that aims to generate uniform cells so that the non-uniform error is minimized. The kd-tree begins from a node that covers the entire space. At each step, this node is recursively binary-partitioned along a chosen dimension until the height of the tree is reached on a pre-defined height or the uniformity heuristic is not satisfied. Here, the objective of the uniformity heuristic is to minimize the variance among the counts of the generated cells.

Cormode et al. (2012) first coined the PSD concept and proposed several optimization techniques that can improve the utility of hierarchical tree structure PSDs. According to the sequential composition theorem, replacing one specific record affects the counts of every node on the path from the root to the leaf containing that record; therefore, the PSD should divide the budget among all levels of the tree. The authors first proved that allocating more privacy budget to leaf nodes can improve the accuracy of queries. They further enhanced the query accuracy by exploiting consistency constraints (Hay et al., 2010) on noisy counts. They proposed generalized consistency constraints for any non-uniform budget allocation scheme, which can greatly improve the accuracy of any range queries. Finally, they introduced several differentially private mechanisms to determine split points to prevent the tree structures from revealing user location. They combined the above techniques and demonstrated that a hybrid method that uses data-dependent splits (i.e., kd-tree) for the first few levels and splits the remaining levels with the data-independent scheme (i.e., quadtree), can achieve the most accurate results.

To et al. (2015) built a PSD with a two-level data-dependent tree, called an h-tree. H-tree is an equi-depth histogram that divides the space such that all cells have similar counts. The h-tree has merely two levels; therefore, it can afford a higher privacy budget to compute split points, and consequently, provide more accurate range query results under the skewed data distribution. Before building the histogram, the optimal granularity, m, which is the number of cells per node, is first calculated. Then, the h-tree divides the space into $m - 1$ cells by containing a comparable number of location points. Here, the h-tree splits the chosen dimension (i.e., latitude) along with the median points, thereby reducing the number of split points from $m - 1$ to $\lceil log_2 m \rceil$. In the same manner, each cell is further split along with the other dimension (i.e., longitude), thereby h-tree of size $m \times m$ is built. Experimental results in To et al. (2015) reveal that the h-tree outperforms kd-tree on sparse and skewed data. However, if the dataset has a dense distribution and the fan-out of the tree is large, the accuracy of h-tree may decrease because splitting each leaf node constantly consumes the privacy budget.

Kim et al. (2018a) proposed a skew-aware grid partitioning method called SAGA that aims to robustly provide enhanced range query accuracy even for a dataset with a highly skewed distribution. For this purpose, the authors first define the notion of a hotspot, where the location points are densely clustered. SAGA privately detects hotspots based on an exponential mechanism. Subsequently, it constructs the histogram by dividing the entire space based on the detected hotspots, unlike other tree-based PSDs that divide the space along with split points. The hotspots can have very different densities; therefore, SAGA further lays a uniform grid with a cell size that minimizes the sum of perturbation and non-uniform errors within the hotspots.

5.1.2. *Data-independent PSD*

Unlike data-dependent methods, in data-independent PSD, the split points are independent of user locations. The partition does not disclose the location of users during spatial decomposition; therefore, the privacy budget is only consumed while calculating the noisy counts of cells. However, when the data distribution is highly skewed, these methods can build unbalanced histograms by dividing the regions where no users exist, thereby creating zero-count cells. Thus, this line of work attempts to approximate the data distribution based on noisy counts.

Qardaji et al. (2013) proposed uniform grid (UG) and adaptive grid (AG) algorithms. They analyzed how the choice of grid size affects the perturbation and non-uniform errors. Specifically, the error of an arbitrary range query, q, can be represented as $Err(q) = \frac{\sqrt{2}rm}{\varepsilon} + \frac{\sqrt{r}N}{mc_0}$, where r is the ratio of the area covered by the query, N is the number of location points, m is the grid size, and c_0 is a constant. By using the Cauchy Schwarz inequality, if m is set to $\sqrt{\frac{N\varepsilon}{2c_0}}$, the sum of the two errors is minimized. UG partitions the underlying space into $m \times m$ grid cells of equal size and adds Laplace noise to each cell count.

UG treats all cells equally; therefore, its performance greatly depends on the input data distribution. When the cells are very sparse, the grid size must be reduced because the perturbation error is considerably large compared to the non-uniform error. On the contrary, when the cells are highly dense, a more fine-grained grid size is required so that the cells overlapping on the border of the query do not incur a non-uniform error. To mitigate this problem, AG employs a two-phase partitioning strategy. Specifically, it lays a more coarse-grained $m_1 \times m_1$ uniform grid in the space domain, where m_1 is determined as $\max(10, 0.25\lceil\sqrt{\frac{\varepsilon_1}{2c_0}}\rceil)$. Subsequently, AG adds the Laplace noise sampled from $Lap(1/\varepsilon_1)$ to the coarse grid cells. Based on the noisy counts, AG further divides the cells whose noisy count is greater than the product of a threshold and $m_2 \times m_2$ grid cells and releases their noisy count using another privacy budget, ε_2. The second-level grid size, m_2, is computed as $\lceil\sqrt{\frac{2N'\varepsilon_2}{\sqrt{2}c_0}}\rceil$, where N' is the noisy count of the cells at the first level. According to the sequential composition, the sum of ε_1 and ε_2 should not exceed a predetermined total privacy budget, ε.

Although AG partially captures the data distribution through a two-layer partitioning strategy, the number of levels is still not sufficient to partition the domain according to the actual distribution of data. Consequently, AG produces numerous empty cells when the input data have a severely skewed distribution, which results in a large perturbation error. Another problem with grid-based methods is that the private information of users may be leaked while tuning the grid size, as raised in Fanaeepour and Rubinstein (2018).

Zhang et al. (2016) proposed a hierarchical quadtree partitioning method, PrivTree. One important aspect of PrivTree is that it does not require a predefined tree height, h, which must be determined in advance in other hierarchical approaches. Therefore, the choice of h does not reveal private information, which also makes PrivTree more adaptive to the underlying data distribution (i.e., generates more child nodes in dense regions). The previous methods add Laplace noise to the cell count and judge whether to continue partitioning a node based on the noise count. Instead of adding noise in the intermediate node counts to determine whether to split, PrivTree computes a biased count with a decaying factor and adds Laplace noise to the biased count. Then, the obfuscated biased counts are used to determine whether to split the nodes further. In summary, PrivTree adaptively computes the tree height through a series of binary questions (i.e., "Does the obfuscated biased count exceed a predefined threshold?") and divides the underlying space into quadrants until the determined tree height is reached. A constant amount of noise is used to determine the height of the tree; therefore, PrivTree should use another privacy budget to publish the noisy count of leaf nodes.

5.1.3. *Comparative summary*

We summarize the comparative analysis of PSD methods in Table 4. As kd-tree based approaches (Cormode et al., 2012; Xiao et al., 2010) determine the split positions based on the distribution of data points (e.g., mean, median), they can avoid the unbalance in partitioning. However, kd-tree based approaches consume large amount of privacy budget when building a kd-tree with a high height, which leads to a higher perturbation error. H-tree (To et al., 2015) can achieve a good performance with sparse and skewed data. However, its performance may decrease when the dataset has a dense distribution. SAGA (Kim et al., 2018a) attempts to solve the skewness problem by detecting hotspots where the data points are densely populated. SAGA splits the entire space based on hotspots and then lays UG in each hotspot, thereby preventing the UG from creating many empty cells. However, SAGA requires expensive computational overhead, because it needs to check all possible sub-regions to detect hotspots.

UG and AG (Qardaji et al., 2013) split the entire space with equal-sized cells, and thus there is no privacy leak in space partitioning. The key factor in the grid-based PSDs is the granularity of cells, which balances the perturbation and non-uniform errors. While the granularity largely depends on the number of data points, such partitions can be unbalanced when the data has a skewed distribution. PrivTree (Zhang et al., 2016) has an advantage that it does not need to predefine the maximum tree height. However, it is still difficult to find the optimal biased count according to the distribution of data.

5.2. *Application of PSD*

In this subsection, we summarize several key application areas of PSD.

Table 4 – Comparative summary of PSD Schemes.

Category	PSD Scheme	Pros	Cons
Data-dependent	Kd-tree (Cormode et al., 2012; Xiao et al., 2010)	Possible to avoid the unbalance in partitioning	High perturbation error when the height of a kd-tree is high
	H-tree (To et al., 2015)	Possible to achieve a good performance on sparse and skewed data.	Decreased performance over dataset with dense distribution
	SAGA (Kim et al., 2018a)	Possible to avoid many empty cells by using hotspots	Expensive computational overhead to detect hotspots for a large dataset
Data-independent	UG (Qardaji et al., 2013)	Possible to balance the perturbation and non-uniform errors	High non-uniform error when the data has a sparse distribution
	AG (Qardaji et al., 2013)	Possible to mitigate the non-uniform error of UG	Hard to adapt to the data distribution with two levels
	PrivTree (Zhang et al., 2016)	No need to predefine the maximum tree height	Difficulty in choosing optimal biased count

5.2.1. *Mobile crowdsourcing*

Mobile crowdsourcing (MC) is a framework in which a group of users voluntarily participates in the collection and sharing of data using their mobile devices. The MC system comprises three subsystems: an MC platform (MCP), workers, and requesters. The system operates as follows. First, requesters send their tasks to the MCP. Then, MCP distributes the received tasks to the workers and matches the tasks to workers considering several costs (e.g., distance, time, reputation). Ideally, MCP intends to assign tasks to workers in such a manner that the overall costs associated with the task completion are minimized.

Current solutions require workers to expose their locations to the MCP. However, MCP is typically untrustworthy. Therefore, protecting the location privacy of workers is necessary. Otherwise, the workers are reluctant to accept the tasks owing to likely violation of their location privacy. To address this problem, To et al. (2014) proposed a framework to protect the location privacy of workers participating in MC. Specifically, they considered the problem in a spatial crowdsourcing (SC) environment, which requires workers to be present or travel to physical locations to accomplish tasks. Here it is crucial to know the exact locations of the workers in the SC, because the workers are more likely to perform tasks closer to their current locations. The authors assumed the presence of a trusted cellular service provider (CSP) between the workers and the SC platform. The CSP directly receives the exact locations of workers instead of the SC platforms, which do not have trust relationships with the workers. Then, the CSP constructs a worker PSD using the collected locations of the workers. The SC platform can access only the released worker PSD, and it determines the regions that include the workers located close to the requested tasks based on the noisy counts in the worker PSD.

Similarly, Gong et al. (2015) utilized PSD to protect location privacy in the mobile cloud computing (MCC) environment. Specifically, they focused on the ad-hoc MCC environment, where any mobile device can launch tasks or perform the tasks of other devices. The proposed framework aims to safeguard the location privacy of mobile devices that accept the tasks of other devices. Additionally, while building the PSD, they considered the reputation of workers, which is referred to as R-PSD (reputation-based PSD). Consequently, the platforms distribute the tasks to regions that contain the workers located in nearby and high reputation scores.

Maruseac et al. (2015) used PSD to detect anomalous phenomena (e.g., environmental accidents and dangerous weather events) in the MC setting. The MCP assigns the tasks of measuring environmental parameters (e.g., temperature and air pollution) to mobile devices for detecting such anomalous cases. Then, the collected data is used to generate a heatmap representing the spatial distribution of environmental parameters. Here, the MCP requires the sum of environmental parameters to decide whether a region is anomalous, which cannot be achieved through simple range queries. Accordingly, they extended the grid-based PSD to maintain the sum of sensor values and analyzed the optimal grid size considering both the number of workers and the sum of sensor values.

5.2.2. *Location recommendation*

Recommendation systems require a large amount of data to provide high-quality recommendations. Thanks to the recent widespread use of location-aware mobile devices, users can easily record their daily movement. Accordingly, several LBS providers attempt to collect massive amounts of location data to recommend new places that the users are likely to visit. However, if the recommenders are untrustworthy, the location privacy of users can be breached. For example, De et al. (2013) showed that an adversary can identify most of the users by tracking four location points where they visited before.

Zhang et al. (2014) applied several PSD methods to provide statistics to the recommenders, which is associated with the results of range queries. The authors assumed that a trusted repository exists between the users and a recommender. Thus, the repository gathers the exact locations from users and releases the PSD to the recommender. They used quadtree and kd-tree approaches to build PSDs and demonstrated that PSD can provide reasonably accurate results for the problem of private location recommendations.

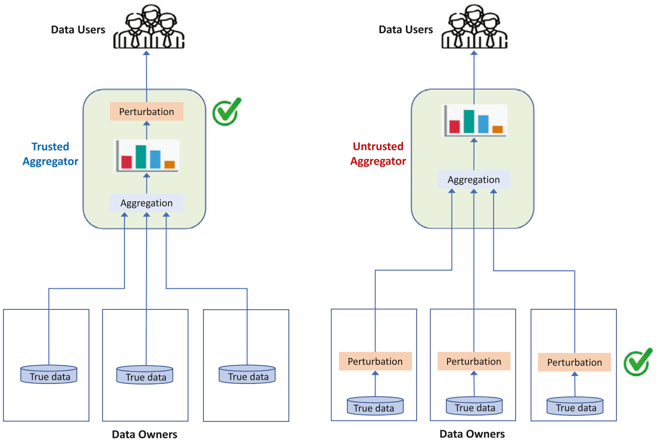
5.2.3. *Traffic monitoring*

Traffic monitoring systems require real-time spatial statistics to guarantee traffic efficiency. Traffic statistics can also be shared with other third-party researchers and used as a tool for environmental planning and urban design. However, the continual release of traffic statistics enables an untrusted third party to infer the trajectory of drivers, which can reveal their behaviors and home and work locations. Most of the traffic monitoring systems require summarized information (i.e., traffic volume in an area), and PSD can deal with such typical range queries while preserving individual privacy.

Fan et al. (2013) focused on the problem of sharing PSD during a given time for real-time traffic monitoring tasks. Although previous research aimed to provide PSD at one time, continuously publishing PSDs imposes another challenge. For example, if a traffic monitoring system requires PSDs during T timestamps to perform real-time tasks, then by the composition theorem, each release should satisfy $\frac{\varepsilon}{T}$-DP in the case of a uniform budget distribution. Therefore, the utility of the published PSDs is severely exacerbated when T is large. To address this problem, the authors applied a framework Fan and Xiong (2012) called FAST that comprises sampling and filtering components to assist the estimation of each cell count. Specifically, they used internal domain knowledge such as the human movement model to predict the variation in cell counts between timestamps. Then, for each timestamp, the framework determines whether to predict the cell counts from the internal model or perturb the cell counts by the Laplace mechanism. In this manner, the Laplace noise is added only when the difference between the prediction and real cell count is large.

6. Local differential privacy

In the DP data collection scenario described in Section 3, a centralized trusted aggregator aggregates original data from individual data owners, perturbs aggregated original datasets



(a) (b)

Fig. 7 – (a) Data aggregation in DP and (b) LDP: Each green check mark indicates the point of time when ϵ-DP is satisfied. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

by adding random noise to centrally satisfy DP, and disseminates the perturbed datasets to data users (Fig. 7(a)). However, in a real application environment, we cannot always assume the presence of such a trusted aggregator between the data owners and users. LDP, which does not require a trusted aggregator, has been developed to solve such constraints of DP in a data collection environment. In LDP, each data owner, who does not fully trust the aggregator, perturbs their original data by adding carefully designed random noise to locally satisfy the DP, and reports the perturbed data to the (untrusted) aggregator (Fig. 7(b)). Formally, LDP is defined as follows (Erlingsson et al., 2014).

Definition 4. (Local Differential Privacy) *A randomized algorithm, $\mathcal{A}$, satisfies ϵ-LDP if and only if for (1) all pairs of the data owner's local data v_a and v_b, and (2) any output O of A, the following equation is satisfied:*

$$Pr[\mathcal{A}(v_a) = O] \leq e^{\epsilon} \times Pr[\mathcal{A}(v_b) = O].$$

The meaning of the aforementioned definition is that, regardless of the data that the aggregator receives from a data owner, the aggregator cannot infer with high confidence (which is controlled by the privacy budget ϵ) whether the data owner has sent v_a or v_b. This provides a plausible deniability to the data owner.

The important property regarding the privacy budget is the sequential composition property. LDP follows the sequential composition property of DP. In other words, the available privacy budget, ϵ, can be partitioned into n smaller privacy budgets, $\epsilon_1, \epsilon_2, \cdots, \epsilon_n$, such that $\epsilon = \sum_{k=1}^{n} \epsilon_k$, and the individual data owner uses each smaller privacy budget to report their local data to the aggregator.

6.1. *Differential privacy mechanisms of LDP*

In this subsection, we describe existing privacy mechanisms that locally achieve DP. Depending on the type of sensitive data of a data owner, the existing privacy mechanisms of LDP can be classified into two categories.

6.1.1. *Categorical data*

The randomized response technique is a survey method that aims to eliminate or reduce the concerns of survey respondents by providing them an opportunity to select a question at a certain probability (Warner, 1965). Given a sensitive survey question of which answer is either "Yes"or "No" (e.g., "Did you cheat during midterm at school?"), a survey respondent is asked to flip a fair coin in secret. If the coin comes up heads, the survey respondent answers the sensitive question truthfully. Otherwise (if the coin comes up tails), the survey respondent flips another coin in secret, and then answers either "Yes" (if the coin comes up a head) or "No" (if the coin comes up a tail). Through this method, the survey respondent has a strong denial for the "Yes" or "No" answer of the sensitive question. Since the randomized response technique provides a simple way for achieving ϵ-DP locally with data in vector form, it has been commonly used in many recent LDP algorithms.

The randomized response is a very basic LDP mechanism for the frequency estimation of a categorical attribute whose domain corresponds to an integer value in $\{1, 2, \cdots, t\}$. For easy explanation purpose, in this subsection, we focus on the a scenario where each data owner reports a (sensitive) single integer value $k \in \{1, 2, \cdots, t\}$ to an aggregator. The data owner's sensitive value k is also represented as a length t binary vector $\vec{v}$ by using one-hot encoding. That is, the k-th bit of $\vec{v}$ is 1 (i.e., $V[k] = 1$) and other bits are 0. Let n be the number of data owners who agree to report their sensitive values to the aggregator. Then, the aggregator wishes to estimate the number of data owners whose sensitive value is to k (i.e., the frequency of k). We classify the randomized response based algorithms into two methods: multi-bit protocol and one-bit protocol.

Multi-bit Protocol. RAPPOR (Erlingsson et al., 2014) is developed for collecting users' data, such as their browser's default homepage, in Google Chrome browser.

- *Data owner:* Given a true binary vector $\vec{v}$ of the data owner, the basic RAPPOR (i.e., ignoring the second randomized response step) perturbs it using the randomized response as follows:

$$\vec{z}[d] = \begin{cases} 1, & with\ probability\ of\ \frac{1}{2}f \\ 0, & with\ probability\ of\ \frac{1}{2}f \\ \vec{v}[d], & with\ probability\ of\ (1-f) \end{cases}$$

  In the next step, the data owner reports the perturbed vector $\vec{z}$ to the aggregator, instead of forwarding $\vec{v}$.
- *Aggregator:* Let $\vec{z_1}, \vec{z_2}, \cdots, \vec{z_n}$ be all perturbed vectors received from n data owners. Here, $\vec{z_i}$ represents the pertrubed vector received from the i-th data owner. All such n vectors are added together to give $\bar{\vec{z}} = \frac{1}{n}\sum_{i=1}^{n} \vec{z_i}$. Then, the frequency of k is estimated as $\frac{\bar{\vec{z}}[k] - 0.5fn}{1-f}$.

Here, $f \in [0, 1]$ is a parameter that controls the level of privacy. That is, a larger value of f enforces stronger privacy protection, adding a larger random noise to the true data, while a smaller value of f provides weaker privacy protection, adding a smaller random noise to the true data. This perturbation mechanism achieves ϵ-differential privacy where $\epsilon = 2 \times \ln(\frac{1-0.5f}{0.5f})$.

Optimized unary encoding (OUE) (Wang et al., 2017c) is considered as one of the protocols providing optimal accuracy in aggregation results.

- *Data owner:* By using OUE, the perturbed vector $\vec{z}$ is obtained from $\vec{v}$ as follows:

$$Pr[\vec{z}[d] = 1] = \begin{cases} p = \frac{1}{2}, & if\ \vec{v}[d] = 1 \\ q = \frac{1}{e^{\epsilon}+1}, & if\ \vec{v}[d] = 0 \end{cases}$$

  That is, if the i-th element of $\vec{v}$ is 1, the probabilities of randomly setting the i-th element of $\vec{z}$ to 1 is $p = .5$. Otherwise, the probability of randomly setting the i-th element of $\vec{z}$ to 1 is $q = \frac{1}{e^{\epsilon}+1}$. This perturbation mechanism satisfies ϵ-differential privacy. Then, the noised vector $\vec{z}$ is reported to the aggregator.
- *Aggregator:* Such as RAPPOR, let $\vec{\bar{z}} = \frac{1}{n}\sum_{i=1}^{n} \vec{z_i}$. Then, the estimated frequency of k is computed by $\frac{\vec{\bar{z}}[k]-nq}{p-q}$.

The communication cost of previous approaches, RAPPOR (Erlingsson et al., 2014) and OUE (Wang et al., 2017c), corresponds to $\mathcal{O}(t)$, because each data owner needs to report all bits in a vector of a length t to the aggregation. Considering that t might be very large for some applications, both RAPPOR and OUE incur high communication cost.

One-bit Protocol. The frequency estimation scheme by Bassily and Smith (2015) reduces communication cost by reporting randomly chosen one bit to the aggregator. Specifically, let $\Phi \in \{\frac{1}{\sqrt{m}}, \frac{1}{\sqrt{m}}\}^{m \times t}$ be an $m \times t$ random projection matrix where m is a pre-defined parameter that affects the confidence level of error guarantee.

- *Data owner:* A sensitive data, $k \in \{1, 2, \cdots, t\}$, of an data owner is first encoded as a pair of $\langle r, x\rangle$ where r is a random value uniformly drawn from $\{1, 2, \cdots, m\}$ and x corresponds to the element in the r-th row and k-th column of Φ (i.e., $x = \Phi[r, k]$). In the next step, the pair of $\langle r, x\rangle$ is perturbed as $\langle r, y\rangle$ as follows:

$$y = \begin{cases} m \cdot x \cdot \frac{(e^{\epsilon}+1)}{(e^{\epsilon}-1)}, & with\ probability\ of\ \frac{e^{\epsilon}}{e^{\epsilon}+1} \\ -m \cdot x \cdot \frac{(e^{\epsilon}+1)}{(e^{\epsilon}-1)}, & with\ probability\ of\ \frac{1}{e^{\epsilon}+1} \end{cases}$$

  The data owner reports the pair of $\langle r, y\rangle$ to the aggregator. Note that in fact, $\langle r, y\rangle$ represents the t-dimensional vector $\vec{z}$ where the r-th entry is y and the other entries are 0.
- *Aggregator:* The estimated frequency of k is computed as follows: Let $\bar{z} = \frac{1}{n}\sum_{i=1}^{n} \vec{z_i}$. Then, the frequency of k is estimated by the inner product of the k-th column of Φ and $\bar{z}$.

Note that unlike the previous multi-bit protocol scheme, each data owner needs to report only information of one bit (i.e., $\langle r, y\rangle$) to the aggregator, which leads to the significant reduction in communication cost.

Besides, there have been several works to use Fourier (Hadamard) transform (Acharya et al., 2019; Cormode et al., 2018). The Hadamard transform matrix is an orthogonal, symmetric matrix Φ of dimension $t \times t$ where t is a power of 2. Each entry in the matrix Φ is defined as

$$\Phi[i, j] = \frac{1}{\sqrt{t}}(-1)^{\langle i, j\rangle},$$

where $\langle i, j\rangle$ is a dot product of the binary representations of i and j. The perturbation steps of data owner and the aggregation steps of aggregator are similar with (Bassily and Smith, 2015).

Optimal local hashing (OLH) (Wang et al., 2017c) is based on the scheme by Bassily and Smith (Bassily and Smith, 2015).

- *Data owner:* Given a value of $k \in \{1, 2, \cdots, t\}$, each data owner first randomly selects a hash function, H (which outputs a value in $\{1, \cdots, g\}$), from a universal hash function family $\mathcal{H}$ and computes $x = H(k)$. Then, the data owner computes the perturbed value, y, as follows:

$$\forall_{i \in [g]} Pr[y = i] = \begin{cases} p = \frac{e^{\epsilon}}{e^{\epsilon}+g-1}, & if\ x = i \\ q = \frac{1}{e^{\epsilon}+g-1}, & if\ x \neq i \end{cases}$$

  The data user reports y, along with H, (i.e., $\langle H, y\rangle$), to the aggregator. Note that like the scheme by Bassily and Smith (2015), $\langle H, y\rangle$ represents the t-dimensional vector $\vec{z}$ where the r-th entry set to 1, if and only if $H(r) = y'$ and the other entries set to 0.
- *Aggregator:* All n noised vectors, received from all data owners, are added together to give $\vec{\bar{z}} = \frac{1}{n}\sum_{i=1}^{n} \vec{z_i}$. Then, the frequency of k is estimated as $\frac{\vec{\bar{z}}[k]-\frac{N}{g}}{p-\frac{1}{g}}$.

According to Wang et al. (2017c), the variance of the frequency estimation is minimized by setting $g = \epsilon + 1$.

6.1.2. Numerical data

In this subsection, we summary existing privacy mechanisms of LDP which can be applicable for numerical data.

As explained in Section 3, a classical mechanism to achieve differential privacy is the Laplace mechanism, which can be used for the LDP setting. Let k_i be the sensitive data of the i-th data owner. For easy explanation purpose, we assume that each k_i lines in range $[-1, 1]$. Given the true value k_i, the Laplace Mechanism is used to compute the noised version k_i' as follows:

$$k_i' = k_i + Lap(\frac{2}{\epsilon}).$$

Here, $Lap(\frac{2}{\epsilon})$ denotes a random noise sampled from a Laplace distribution with mean $\mu = 0$ and scale $b = \frac{2}{\epsilon}$. Note that the sensitive of k_i is 2, because k_i lines in range $[-1, 1]$. The noised value, k_i', is then reported to the aggregator. Once receiving all noised data, $k_1', k_2', \cdots, k_n'$, the mean value is simply estimated as $\frac{1}{n}\sum_{i=1}^{n} k_i'$. More sophisticated variants of the Laplace mechanism, such as (Geng et al., 2015; Soria-Comas and Domingo-Fer, 2013), that yield better estimation results can be used in the LDP setting.

The methods based on the randomized response are used for the collection of data represented as a vector, where each bit corresponds to either 0 or 1, and thus cannot be directly applied to the scenario in which the sensitive data of a data owner is represented as numerical values. One solution is to discretize a numerical value into a binary one and to perturb the discretized value by the randomized response. The perturbation method of Duchi et al. (2018), which can be applied to numeric data under the LDP setting, is based on this mechanism. Given a true sensitive value $k_i \in [-1, 1]$ of the i-th data owner, it computes a noised value, k_i', as follows:

$$k_i' = \begin{cases} \frac{e^{\epsilon}+1}{e^{\epsilon}-1}, & \textit{with probability of } \frac{1}{2} + \frac{e^{\epsilon}-1}{2e^{\epsilon}+2} \cdot k_i \\ -\frac{e^{\epsilon}+1}{e^{\epsilon}-1}, & \textit{with probability of } \frac{1}{2} - \frac{e^{\epsilon}-1}{2e^{\epsilon}+2} \cdot k_i \end{cases}$$

Each data owner reports the noised value, k_i', to the aggregator. Upon receiving all the noised values, the aggregator simply computes their average, $\frac{1}{n}\sum_{i=1}^{n} k_i'$, to estimate the mean value. Works in Ding et al. (2017); Ding et al. (2018); Nguyen et al. (2016); Ye et al. (2019) also leverage this similar mechanism to develop the perturbation methods that can be applied to numerical data.

Wang et al. (2019a) proposed the randomized response method, called piecewise mechanism (PM), that focuses on the problem of estimating the mean of numeric values under LDP. Given a true sensitive value $k_i \in [-1, 1]$ of the i-th data owner, PM outputs a noised value $k_i' \in [-C, C]$. Here, C is defined as

$$C = \frac{e^{\epsilon/2}+1}{e^{\epsilon/2}-1}.$$

Specifically, the PM builds a probability distribution with three pieces: the left piece in the range $[-C, l(k_i)]$, the center piece in the range $[l(k_i), r(k_i)]$, and the right piece in the range $[r(k_i), C]$. Here, $l(k_i)$ and $r(k_i)$ are defined respectively as

$$l(k_i) = \frac{C+1}{2} \cdot k_i - \frac{C-1}{2} \quad \textit{and}$$

$$r(k_i) = l(k_i) + C - 1.$$

Given k_i, the PM outputs a noised numeric value, k_i', that lies in the center piece with a relatively high probability. Specifically, let $RV([a, b])$ be a function that outputs a random value uniformly drawn from the range $[a, b]$, Then, the noised value k_i' is obtained as follows:

$$k_i' = \begin{cases} RV([l(k_i), r(k_i)]) & \textit{with prob. } \frac{e^{\epsilon/2}}{e^{\epsilon/2}+1} \\ RV([-C, l(k_i)) \cup (r(k_i), C]), & \textit{with prob. } \frac{1}{e^{\epsilon/2}+1} \end{cases}$$

Then, the data owner reports the noised value, k_i', to the aggregator. After receiving all noised data, $k_1', k_2', \cdots, k_n'$, the aggregator estimates the mean value by computing their average, $\frac{1}{n}\sum_{i=1}^{n} k_i'$.

6.1.3. *Comparative summary*

Table 5 summarizes the communication cost and the variance of LDP schemes for categorical data. The communication costs of both basic RAPPOR (Erlingsson et al., 2014) and OUE (Wang et al., 2017c) are t, and can be costly when t is large. Thus, both schemes are not suitable for high categorical data. On the other hands, the communication costs of Bassily and Smith (Bassily and Smith, 2015), Hadamard Randomized Response (Cormode et al., 2018), and OLH (Wang et al., 2017c) are $\log m$, $\log t$, and $\log n$ respectively, which are much less than t. The variance of basic RAPPOR is $\frac{e^{\epsilon/2}}{(e^{\epsilon/2}-1)^2}$, while that of other approaches corresponds to $\frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}$.

Table 6 shows the comparison of LDP schemes for numerical data. Both Laplace mechanism and PM (Wang et al., 2019a) output a value in a continuous distribution, while the output of perturbation step in Duchi *et al.*'s solution (Duchi et al., 2018) is either $\frac{e^{\epsilon}+1}{e^{\epsilon}-1}$ or $-\frac{e^{\epsilon}+1}{e^{\epsilon}-1}$ (binary perturbation). Regarding the variance, PM provides fairly good performance in an entire privacy budget range. Laplace mechanism incurs a high variance of $\frac{8}{\epsilon^2}$, but as ϵ increases, it's variance decreases quadratically. Duchi *et al.*'s solution achieves the best performance when the privacy budget is small (according to Wang et al. (2019a), when $\epsilon \leq 2$). However, it's variance is very slightly reduced with the increase of ϵ, and thus shows the worst performance when ϵ is large.

6.2. *Applications of LDP*

LDP has attracted considerable attention in academic literature. Furthermore, there have been several successful LDP-based industrial deployments in major technology companies, including Google (Erlingsson et al., 2014), Samsung (Nguyen et al., 2016), Apple (Tang et al., 10; Differential privacy team, 2018), and Microsoft (Ding et al., 2017). In this subsection, we briefly summarize several key application areas of LDP.

6.2.1. *Marginal distribution*

The marginal (joint) distribution of multiple variables is utilized for advanced data analysis. Fanti *et al.* proposed a method to estimate the joint distribution of multiple variables that are built on the top of RAPPOR (Fanti et al., 2016).

Table 5 – Comparison of communication cost and variance of LDP schemes for categorical data.

LDP Scheme	Communication Cost	Variance
Basic RAPPOR Erlingsson et al. (2014)	t	$\frac{e^{\epsilon/2}}{(e^{\epsilon/2}-1)^2}$
OUE Wang et al. (2017c)	t	$\frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}$
Bassily and Smith Bassily and Smith (2015)	$\log m$	$\frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}$
Hadamard Randomized Response Cormode et al. (2018)	$\log t$	$\frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}$
OLH Wang et al. (2017c)	$\log n$	$\frac{4e^{\epsilon}}{(e^{\epsilon}-1)^2}$

Table 6 – Comparison of LDP schemes for numerical data.

LDP Scheme	Variance	Output of Perturbation Step
Laplace mechanism	$\frac{8}{\epsilon^2}$	a value in a continuous distribution
Duchi's solution (Duchi et al., 2018)	$(\frac{e^\epsilon+1}{e^\epsilon-1})^2$	either $\frac{e^\epsilon+1}{e^\epsilon-1}$ or $-\frac{e^\epsilon+1}{e^\epsilon-1}$ (binary perturbation)
PM (Wang et al., 2019a)	$\frac{4e^{\epsilon/2}}{3(e^{\epsilon/2}-1)^2}$	a value in a continuous distribution

In the proposed strategy, the joint distribution of multiple variables is modeled by Bayes' theorem, and the expectation maximization (EM) algorithm is employed to iteratively compute the joint and conditional probabilities of Bayes rule. Cormode et al. (2018) developed a method to estimate the k-way marginal table under LDP, where the maximum k should be determined in advance. The perturbation phase of the proposed method in Cormode et al. (2018) is based on the Hadamard randomized response in which Fourier (Hadamard) transformations are first applied to the sensitive data of a data owner, and subsequently, the randomized response technique is applied to the transformed data. Ren et al. (2018) proposed LoPub to aggregate high-dimensional crowdsourced data from distributed users and infer marginal distribution under LDP. Their proposed approach is based on EM methods, which were originally developed by Fanti et al. (2016), and Lasso regression. Zhang et al. (2018b) proposed CALM to estimate k-way marginal tables for high-dimensional attributes. Unlike the method by Cormode et al. (2018), CALM does not require the maximum k to be determined in advance; therefore, it can support queries of arbitrary k values. Instead of directly constructing all marginal tables, CALM first chooses sets of attributes and reconstructs all k-way marginal tables from the chosen sets of attributes.

6.2.2. *Heavy hitter estimation*

Heavy hitter estimation aims to find the top-k items with the highest frequency and the estimated frequency for each such item. This estimation is a well-studied topic with diverse important applications. A straightforward solution to this problem under LDP is to compute the frequency of every item by using the frequency estimation described in Subsection 6.1.1 and identify the most frequent k-items. However, this simple solution incurs high communication cost when the number of items, d, is considerably large because each data user needs to report all d bits to the data collector. Furthermore, the accuracy of estimated heavy hitters can be inaccurate because the privacy budget is wasted in perturbing and reporting items that are not included in the most frequent k-items. Bassily and Smith (Bassily and Smith, 2015) proposed an efficient protocol for heavy hitter estimation that is based on a succinct histogram, S-Hist. A succinct histogram, which is a data structure comprising a list of heavy hitters (items) and their estimated frequencies is efficiently constructed using the one-bit protocol described in Subsection 6.1.1. In a subsequent research, Bassily *et al.* proposed a new heavy hitter algorithm, TreeHist (Bassily et al., 2017), which outperforms the previous approach (Bassily and Smith, 2015) in terms of time complexity and estimated accuracy. The main idea of TreeHist is to transform users' items into binary strings and build a binary prefix tree to compute the top-k items with the highest frequency. Bun et al. (2018) developed an algorithm, PrivateExpanderSketch, that outperforms the heavy hitter algorithm in Bassily et al. (2017), in terms of worst-case error. Qin et al. (2016) proposed LDPMiner for heavy hitter estimation. The proposed LDPMiner comprises two phases: (i) a candidate set selection phase that uses a portion of the privacy budget to identify the candidate set that can be included in the most frequent k-items with high probability, and (ii) a refining phase that selects heavy hitters from the candidate set by leveraging the remaining privacy budget. Wang et al. (2019d) developed the prefix extending method (PEM) for heavy-hitter problems. In PEM, users are partitioned into several equal-sized groups, and users in each group report a prefix of their values. The authors in Wang et al. (2018a) utilized a trie structure, PrivTrie, to directly collect frequent items (or terms) from users by iteratively constructing a trie under LDP.

6.2.3. *Complex statistical analyses*

In most of the early research works, LDP was used to collect basic statistics from simple data types, such as frequency (or count) estimation or mean value estimation. However, in recent years, there have been growing efforts to apply LDP to more advanced statistical estimation tasks.

The key-value model is one of the most popular data models used in most NoSQL databases. Ye et al. (2019) proposed PrivKV to effectively estimate the frequency and mean on key-value datasets. To maintain the correlation between keys and values, the main idea of PrivKV is to first perturb the keys, and then perform the perturbation step of values based on the perturbed keys. Furthermore, to reduce the network transmission overhead and improve the estimation accuracy, the authors devised an optimization strategy called virtual iterations, in which the aggregator executes virtual PrivKV iteratively without data user involvement.

The estimation of probability distributions is a fundamental statistical problem that is essential for various applications in several fields such as machine learning, probability theory, and data analysis. To this end, it has been studied extensively. Several LDP-based frameworks have been proposed to privately estimate various probability distributions, such as discrete distribution (Duchi et al., 2013; Kairouz et al., 2016; Murakami et al., 2018; Pastore and Gastpar, 2016; Wang et al., 2019b; Ye and Barg, 2018), ordinal data distribution (Wang et al., 2017b), and Gaussian distribution (Joseph et al., 2019).

The authors in Joseph et al. (2018) studied the problem of continuously computing static over evolving data under LDP. Owing to the sequential composition property of DP, naively repeating a differentially private computation over evolving data results in quick exhaustion of the privacy budget. Therefore, the authors introduced a mechanism in which the population of users is first partitioned into subgroups, and given a subgroup, the privacy budget of DP is spent only if the distribution of data within the group has changed significantly.

Recently, there have been several attempts to support various types of queries beyond simple count queries in the LDP setting. Cormode et al. (2019) proposed two approaches, based on hierarchical histogram and the Haar wavelet transform, to accurately answer range queries on one-dimensional discrete data under LDP. Wang et al. (2019c) invented a mechanism to answer multi-dimensional analytical queries, which are common SQL queries against a fact table, under LDP.

6.2.4. *Deep/machine learning and data mining*

In recent years, deep learning has achieved remarkable success in various applications. This can be attributed to the massive amount of data available for network model training. However, massive data collected from multiple sources and stored in the central server may raise privacy issues. To address these privacy issues, multi-parity deep learning (collaborative deep learning), wherein multiple participants jointly train a deep neural network model through a central server without sharing their private data has received significant attention from the research community. LDP acts as a de facto technique in enabling multi-parity deep learning in a privacy-preserving manner (Gong et al., 2020; Shokri and Shmatikov, 2015). In other words, in a privacy-preserving multi-parity deep learning, each participant first computes the gradients of a neural network model using their local data, perturbs them under LDP, and reports the perturbed gradients to the untrusted central server. After receiving all perturbed gradients from the participants, the server computes the combined gradients that are downloaded by each participant. This process is repeated iteratively until the pre-defined object is achieved. Arachchige et al. (2019) proposed LATENT, an LDP mechanism based on a randomized response, to train a convolutional neural network in the setting with an untrusted curator.

Various data mining problems have been investigated under LDP. Wang et al. (2018b) studied a protocol for frequent itemset mining in the set-valued LDP setting. Their approach is based on the concept of “padding and sampling,” which was originally developed in Qin et al. (2016). In the padding and sampling strategy, each data owner first creates the fixed-size padded set by padding their set of values with dummy items, samples one item from the padded set, and perturbs the sampled item before reporting it to the aggregator. While estimating the frequency of an item, the aggregator multiplies the estimated frequency by the size of the padded set. In Wang et al. (2018b), this scheme is further improved by introducing a privacy amplification method and adaptively selecting the size of the padded set—the improved scheme is used for frequent itemset mining. Guo et al. (2019) studied item-based collaborative filtering under LDP, wherein private historical data of users are protected without significant deterioration of the recommendation accuracy. In Nissim and Stemmer (2018), the popular k-means clustering problem that arises in several different data mining applications was studied in the local model of DP. In Fan et al. (2020), classification, which is the most widely used data mining technique, was studied under LDP.

The matrix factorization algorithm is widely used to identify the relationship between users and items in recommender systems. Several studies have adopted LDP in matrix factorization to protect user information (i.e., user’s items and ratings). Shin et al. (2018) proposed a recommendation system that can protect both the user items and ratings. For this purpose, they developed novel matrix factorization algorithms under LDP. In their method, the LDP mechanism was employed in a stochastic gradient descent step of iterative learning, wherein the perturbed gradients of each user are iteratively reported to the recommendation server. Sun et al. (2018) developed an LDP-based matrix factorization method to protect sparse crowdsourcing data (i.e., sparse worker answers) on a crowdsourcing environment. Asada et al. (2019) developed an LDP-based matrix factorization method for location privacy preference recommendation.

6.2.5. *IoT Data analysis*

The widespread use of Internet of Things (IoT) devices, such as smartphones and wearable health devices, enables easy collection of massive amounts of personal data produced by individuals during their daily activities. The collected data, which are typically transmitted to and stored in the cloud, are further processed and used by data-driven services such as analytics, advertisement, and recommendation systems. However, as data collection through IoT devices is becoming a popular method to collect various data from various users, it raises privacy concerns because data collected via personal IoT devices can be used to reveal sensitive personal information. Therefore, in recent years, there have been growing efforts to leverage LDP to collect individuals’ sensitive data through IoT devices.

Xu et al. (2019) developed an EdgeSanitizer that leverages a deep learning model to extract useful features from IoT device data, which is characterized as high-dimensional, obfuscates selected features by adaptively injecting random noise under LDP, and performs data reconstruction to reassemble the perturbed features for data analytics. The authors in Kim et al. (2018c) and Kim and Jang (2019) utilized LDP to collect indoor positioning data from users in a privacy-preserving manner. The collected data is then used to estimate crowd density in indoor locations. Kim et al. (2018b, 2019) developed an LDP-based mechanism that is capable of collecting sensitive health lifelogs, e.g., heart rate, blood pressure, counting steps, from smartwatch users, while protecting the data privacy of individual smartwatch users. Their approach first searches for salient points, where changes in the trends occur, as seen in the health lifelog stream and then perturbs the selected salient points under LDP, before reporting them to the aggregator. Once the perturbed salient points from each smartwatch user are received, the aggregator reconstructs the health lifelog stream based on the received perturbed salient points. Usman et al. (2019) proposed a multi-layer framework PAAL, wherein LDP is used for privacy-preserving multimedia data aggregation in IoT environments.

7. Application of differential privacy-based schemes in LBS

In this section, we first analyze the techniques surveyed in this paper with various parameters from the perspective of the use of LBSs, and then present their applicability to protect users' location privacy in LBSs.

7.1. Comparison between geoind, LDP, and PSD

Table 7 provides a comparison between GeoInd, LDP, and PSD based on five parameters from the perspective of the use of LBSs.

Need for trusted server: The first important consideration when applying DP-based techniques to a real application setting is the presence of a trusted server (e.g., data curator in Fig. 3). If a trusted server is available, users forward their original sensitive data to the trusted server, which performs data perturbation. Otherwise, the users are responsible for perturbing their original sensitive data. Among the three variants of DP surveyed in this study, GeoInd and LDP do not require a trusted server, whereas PSD does.

Direct application to location data: GeoInd and PSD were proposed to provide privacy protection for users' location information; therefore, they are directly applicable to location data. However, LDP is a general-purpose method that enables privacy-preserving collection of users' sensitive data represented in a categorial or numerical form. Therefore, to use LDP for location data, it is necessary to represent the user's location information in a data format that is compatible with LDP. One viable solution is to partition a spatial domain into several cells and represent the user's location using the cell wherein the user is currently located, e.g., Kim and Jang (2019) (Fig. 8). More specifically, assume that the target spatial domain is divided into m cells, $c_1, c_2, \cdots, c_m$. Furthermore, assume that the location of a specific user belongs to the k-th cell, c_k. The location of this user is represented as an m-dimensional vector such that the k-th element, d_k, is set to 1 and the others are set to 0 (i.e., $V = [d_1, \cdots, d_k, \cdots, d_m] = [0, \cdots, 1, \cdots, 0]$). Once the user's location is represented as a vector in this manner, the perturbation mechanisms of LDP discussed in Subsection 6.1 can be applied to it.

Privacy-preserving location-based query processing: GeoInd is originally designed to collect location information from mobile devices to provide LBSs to users. In other words, the utility of GeoInd is based on single location data; therefore, it is appropriate for location-based query processing. As seen in Fig. 9, a user who wants to receive services adjusted to their current location (i.e., P1) perturbs the true location, and reports the perturbed location (i.e., P2) with service requests to the LBS server. Then, the LBS server provides the user with services adjusted to the perturbed location. However, unlike GeoInd, the utility of LDP and PSD is based on aggregating data from a large number of users, making them inappropriate for location-based query processing.

Privacy-preserving location data collection: Depending on the granularity of the data collected by the LBS servers, the location data collection can be categorized into two cases: microdata collection and aggregate information collection. In the microdata collection scenario, a large amount of microlocation data comprising a single location of each individual user is collected by the LBS servers. On the contrary, in the aggregate information collection scenario, LBS servers intend to compute aggregate information (e.g., crowd-density at a specific location) that is obtained by performing aggre-

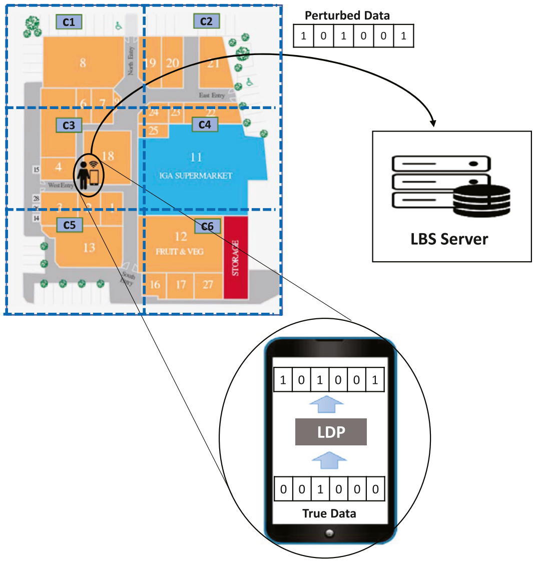


Fig. 8 – An example of applying LDP to location data: The spatial domain is divided into 6 cells, $c_1, c_2, \cdots, c_6$, represented by dotted rectangles and a specific user is located in c_3.

Table 7 – Comparison between GeoInd, LDP, and PSD from the perspective of the use of LBSs.

	Need for trusted server	Direct application to location data	Privacy-preserving location-based query processing	Privacy-preserving location data collection		Privacy-preserving location data publishing	
				microdata	aggregate information	microdata	aggregate information
GeoInd	×	✓	✓	✓	×	✓	×
LDP	×	×	×	×	✓	×	✓
PSD	✓	✓	×	×	✓	×	✓

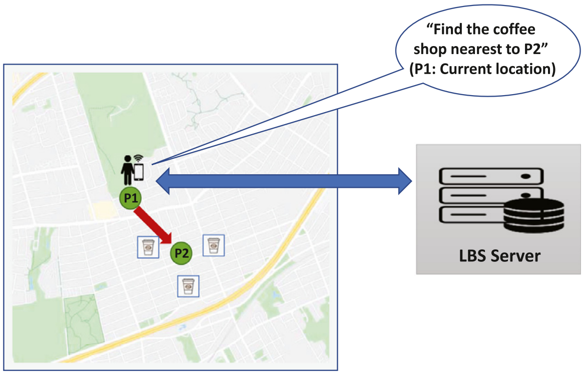


Fig. 9 – An example of location-based query processing with GeoInd.

gate operations over the whole location data set. Among the aforementioned three alternatives, GeoInd is suitable for the scenario of privacy-preserving micro-location data collection because the utility of GeoInd is mostly centric to single location data. However, LDP is appropriate for a scenario of privacy-preserving aggregate information collection because it has been proposed to compute aggregate statistics based on a large amount of collected data; therefore, its utility is centric to aggregating data. Additionally, LBS servers can collect aggregate information via a trusted server that acts as a data curator by using PSD. In this case, a trusted server collects original location information from individual users, computes a perturbed spatial histogram under PSD, and forwards it to the LBS servers (Fig. 10).

Privacy-preserving location data publishing: Similar to the case of location data collection, location data publishing can be considered in two different scenarios: microdata publishing and aggregate information publishing. In the microdata publishing scenario, LBS providers disseminate a large amount of micro-location data containing a single location of each individual user to third parties. On the contrary, in the aggregate information publishing scenario, aggregate information extracted from a large amount of location data is distributed for public use. Similar to the case of location data collection, GeoInd is suitable for privacy-preserving micro-location data publishing, whereas LDP and PSD are appropriate for privacy-preserving aggregate information publishing.

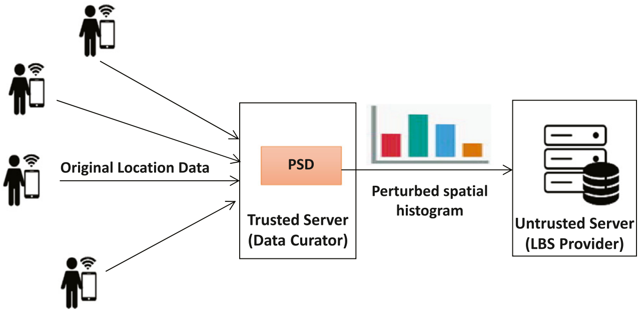

Original Location Data
PSD
Trusted Server (Data Curator)
Perturbed spatial histogram
Untrusted Server (LBS Provider)

Fig. 10 – Privacy-preserving location data collection via a trusted server by using PSD.

7.2. *Applicability of geoind, LDP, and PSD*

Table 8 presents the applicability of GeoInd, LDP, and PSD for different privacy-preserving location data processing, collection, and publishing scenarios in LBS. We discuss the applicability of three alternatives to location data in three different architectures of LBSs.

Trusted LBS Provider: In this case, there is no privacy issue for both location-based query processing and location data collection, because users fully trust the LBS provider. PSD is applicable for privacy-preserving location data publishing in such a manner that a trusted LBS provider, who has access to the original location information of individual users, computes a perturbed spatial histogram and releases it for public use.

Untrusted LBS Provider with trusted server: In this case, users communicate with an untrusted LBS provider via the trusted server (i.e., data curator) that is responsible for performing data perturbation. GeoInd can be employed to process location-based queries in a privacy-preserving manner. Herein, the user sends their original location information to the trusted server, which then perturbs the original location data and forwards the perturbed location data to the LBS server. Furthermore, it can be employed to collect and publish a set of micro-location data via a trusted server. PSD is applicable to privacy-preserving location data collection, as previously explained in Fig. 10. Furthermore, it can be employed for privacy-preserving location data publishing because the perturbed spatial histogram received from the trusted server can be published to be used by third parties for different purposes without further processing.

Untrusted LBS Provider without trusted server: In this case, users are responsible for performing the data perturbation because they directly interact with the untrusted LBS provider. GeoInd is applicable to privacy-preserving location-based query processing, wherein users send perturbed location data to the LBS providers instead of reporting original data. GeoInd can be used to collect micro-location data in a privacy-preserving manner. Furthermore, the collected micro-location data set can be published for data analytics purpose to third parties. LDP is applicable for privacy-preserving location data collection, wherein each user perturbs an original location by adding random noise before reporting it to the LBS server. Furthermore, the aggregate information, which is computed based on the perturbed location data set collected from users, can be published to be used by third parties; therefore, LBS is applicable to privacy-preserving location data publishing.

We further discuss some issues that can arise when applying the DP-based techniques surveyed in this paper to protect location privacy in real-world LBS applications. In inference attacks, an attacker illegally analyzes the data of a target to draw conclusions about the target (Cormode, 2011; Jagwani and Kaushik, 2017). Inference attacks have been reported in several LBS systems (Krumm, 2007; Xiao et al., 2019; Yao et al., 2018). The inference attacks against differentially private location data are broadly categorized into *tracking* and *identification* attacks.

Table 8 – Applicability of GeoInd, LDP and PSD for different location data processing, collection and publishing scenarios in LBS.

	LBS architecture		
	Trusted LBS Provider	Untrusted LBS Provider with a trusted server	Untrusted LBS Provider without a trusted server
Privacy-preserving location-based query processing	No privacy concern	GeoInd	GeoInd
Privacy-preserving location data collection	No privacy concern	GeoInd (microdata), PSD (aggregate information)	GeoInd (microdata), LDP (aggregate information)
Privacy-preserving location data publishing	PSD (aggregate information)	GeoInd (microdata), PSD (aggregate information)	GeoInd (microdata), LDP (aggregate information)
Data Perturbation	Performed by an LBS server	Performed by a trusted server	Performed by each user

- In identification attacks, the attacker aims at identifying a target user in a group of users. Normally, the attacker uses the location information of a set of users to identify its target. For an illustration, consider an adversarial LBS provider. Assume that the LBS provider first collects the location information of a set of users and uses it to compute the actual POIs for each user. Later, if a user submits obfuscated location information to the LBS provider anonymously, the adversarial LBS provider then associates the obfuscated location information with the actual POIs to identify the user. Primault *et al.* demonstrated that, under the GeoInd mechanism, this attack is feasible with a success rate of 63% to 89% depending on the degree of obfuscation (Primault et al., 2014).
- In tracking attacks, the attacker aims at establishing the travel or movement patterns of a target, i.e., the attacker aims to reconstruct the actual trajectory of its target. By design, in most LBSs, users periodically submit their location information. A dishonest LBS provider can use the reported location information and analyze the trace files to establish the trajectory of a target user (Zhang et al., 2018a). With a compromised trajectory, a user loses his/her privacy and becomes vulnerable to further attacks, including physical attacks. In Cao et al. (2017, 2018), Cao *et al.* revealed that differential privacy mechanisms are prone to privacy loss under a temporally correlated data release. Thus, an attacker that repeatedly receives such data can use them to infer the complete trajectory pattern of the target user.

Although in theory, the DP-based techniques surveyed in this paper have been known to guarantee the location privacy of an individual, under weak levels of privacy, they are prone to inference attacks. Therefore, a balance between utility and privacy is required to apply DP-based techniques to real-world LBS applications.

Currently, the location information of a user is generally obtained using mobile devices. Although the computational capacity of mobile devices has increased rapidly in recent years, most of these devices still provide less computational power compared with servers. In particular, the limited computational power of mobile devices can be a problem for GeoInd and LDP, in which data perturbation must be performed on the user device side. As described in Subsection 4.1, for GeoInd, the optimization mechanism can provide a higher utility than the PL mechanism. However, the optimization mechanism involves solving linear programming problems that can be computationally expensive, and thus is less efficient than the PL mechanism. Thus, the optimization mechanism of GeoInd is not suitable when mobile devices with extremely limited computation power are used to measure and perturb the location of a user in LBSs. A similar problem can be encountered when using LDP, particularly when a spatial domain is partitioned into a large number of finer-grained cells. In this case, a high data utility can be achieved, but a large number of bits are required to represent the location of a user, resulting in an increased computational overhead during the data perturbation phase of LDP. Thus, with mobile devices with a lower computational capacity, the efficiency of use of GeoInd and LDP needs to be enhanced.

DP-based techniques protect data privacy by perturbing true data by adding random noise, and such perturbation mechanisms cause a loss in data utility. This problem can be particularly severe for LDP, because the perturbation mechanism of LDP adds a sufficient amount of noise to the true data to achieve ϵ-DP locally. Thus, in terms of the amount of noise, LDP adds a relatively large amount of noise to the true data. LDP requires a large amount of data to be collected to offset the large amount of noise caused by the perturbation mechanism. Therefore, LDP is particularly useful for LBS applications in which it is possible to collect a large amount of location data from many users; however, it is not recommended for LBS applications in which it is not possible to collect a large amount of location data from many users.

8. Future research directions

This section provides examples of some open challenges and research directions for location privacy provision in LBSs.

8.1. *Continuous reporting of location information*

Reporting of location data in LBSs can be categorized as continuous or sporadic (Shokri et al., 2011). It should be noted that GeoInd is based on the assumption that the location reports of a user are independent of each other. Although this assumption may hold for sporadic reporting of location data, it is likely to fail in continuous reporting mode because it neglects the potential threat posed by examining the correlation between different location reports (Mendes et al., 2020).

Continuously reporting location data to an LBS server under an LDP setting raises issues of privacy. Owing to the sequential composition property of LDP, continuously running a differentially private computation over the same dataset causes the privacy budget to become enormous, which leads to the problem of privacy loss (Joseph et al., 2019). For example, Tang et al. (2017) observed that the value of ϵ can become unreasonably large when periodically collecting user data. Furthermore, similar issues can be encountered while continuously publishing a spatial histogram using PSD. Therefore, developing efficient and effective algorithms (that are resilient against tracking attacks) targeting applications that require continuous reporting of location data and quantifying how continuous location data reporting affects the privacy-utility trade-off in GeoInd, LDP, and PSD for different location applications can be further investigated.

8.2. *Quantifying the risk of location privacy disclosure*

As described earlier, the privacy budget, ϵ, controls the level of privacy such that smaller (larger) values of ϵ ensure a stronger (weaker) privacy guarantee but introduce larger (smaller) noise in the true result. However, even though the value of ϵ is notified to LBS users in advance, they are unaware of the exact risk of their location privacy disclosure. This is because privacy budget is a rather theoretical quantity; therefore, it is not easily connected to the practical privacy metric that can quantify the risk of privacy disclosure according to this parameter. Although there have been a few studies, such as Hsu et al. (2014); Lee and Clifton (2011) that have investigated the privacy metric for DP, studies on developing a practical privacy metric, which is able to quantify the risk of location privacy disclosure according to the value of ϵ in the use of LBSs, have received little attention from the scientific community; therefore, it needs to be urgently investigated.

8.3. *One-Size-Fits-All solution for location privacy*

During the use of LBSs, the location information of users can be exposed not only to the LBS server but also the positioning systems (Liu et al., 2019). For instance, in Wi-Fi fingerprint-based indoor localization, which is one of the most representative approaches to measure a user's position in indoor spaces, a to-be-localized user (who eventually wants to receive services adjusted to their current location) measures the radio signal strength of all access points and sends it to the positioning system. Then, the positioning system computes the position and returns it to the user. During this process, the exact location information of the user can be leaked to a malicious positioning system, which raises another privacy concern that should be resolved for the safe use of LBS.

Extensive studies have been conducted on privacy-preserving localization in the literature, such as (Li et al., 2014a; Nieminen and Jrvinen, 2020). Additionally, there has been a recent attempt to leverage the DP mechanism to measure a user's location in a privacy-preserving manner (Wang et al., 2018c). However, a one-size-fits-all privacy solution that is able to protect location privacy both from the LBS server and the positioning system does not exist currently. For example, if the (perturbed) current location that is computed by the positioning system in a privacy-preserving manner, satisfies ϵ-GeoInd (or ϵ-DP), then an LBS user reports it to the LBS server without further perturbation processing. We believe that making such a one-size-fits-all privacy solution available will boost the wide adoption of LBSs.

8.4. *Optimality for mobile crowdsourcing*

Mobile crowd-sourcing applications have greatly benefited organizations and societies (Ind, 2020). To protect location information, several studies have proposed DP-based mechanisms for mobile crowdsourcing applications. However, unlike PSD and LDP, GeoInd mechanisms generally gear towards sporadic and short period services, and moreover, their utility is mostly user-centric. In contrast, mobile crowdsourcing applications involve data collection over longer periods of time, and their utility is based on aggregating data from a large number of users, i.e., the utility is not user-centric (Boukoros et al., 2019). This has an undesirable effect on the required privacy levels (which must be high) even when optimized for utility (Boukoros et al., 2019). Therefore, designing optimal GeoInd mechanisms for location privacy protection in mobile crowdsourcing applications by considering the utility of aggregate data from a large number of users can be performed in the future.

9. Conclusion

With the widespread adoption of mobile devices with their own communication capabilities, various services based on the location of a user are becoming increasingly prevalent. These services provide users with various convenience services, such as location-aware weather reports, navigation and direction, and location-aware assistive services. For a user to receive various benefits from such services, the location information of the user needs to be provided to the service providers, which leads to privacy concerns. The location data of individual users typically contain location-sensitive information; thus, by tracking them, it is viable to infer the sensitive information of the users, such as home and workplace locations, and hospital visits. Therefore, concerns always exist regarding the leakage of sensitive private information, owing to the provision of location information to the service providers. Thus, several privacy-preserving techniques have been proposed to mitigate location privacy threats when using LBSs. One of the most promising solutions to overcome these privacy threats is to enhance location privacy by noising the original data using the perturbation mechanism of DP.

Thus, in this study, we surveyed the DP-based solutions for privacy protection of location data in LBSs. Firstly, we provided background information on LBSs and highlighted the privacy issues that may arise from the use of location data in LBSs. Second, we discussed three DP-based solutions, GeoInd, LDP, and PSD, which are designed or can be used to protect location privacy in LBSs. Third, after a discussion of these approaches, we investigated the applicability of DP-based schemes to protect location privacy in different location data processing, collection, and publishing scenarios in LBSs. Finally, we highlighted potential future research problems related to privacy-preserving LBSs. To the best of our knowledge, this is the first study to thoroughly survey the existing DP algorithms from the perspective of LBSs. We believe that this study will stimulate more interest in the research issues associated with location privacy and promote more research efforts toward the wide adoption of LBSs.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research was supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF-2020R1F1A1072622).

REFERENCES

Acharya J, Sun Z, Zhang H. Hadamard Response: Estimating Distributions Privately, Efficiently, and with Little Communication. Proceedings of the International Conference on Artificial Intelligence and Statistics, 2019.

Ahuja R, Ghinita G, Shahabi C. A Utility-preserving and Scalable Technique for Protecting Location Data with Geo-indistinguishability. In: Proceedings of the International Conference on Extending Database Technology; 2019. p. 210–31. Lisbon, Portuga

Akgun M, Bayrak AO, Ozer B, Sagiroglu MS. Privacy preserving processing of genomic data: asurvey. J Biomed Inform 2015;56:103–11.

Andres ME, Bordenabe NE, Chatzikokolakis K, Palamidessi C. Geo-indistinguishability: Differential Privacy for Location-based Systems. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security; 2013. p. 901–14. Berlin, Germany

Apple differential privacy team. 2018. Learning with Privacy at Scale. https://docs-assets.developer.apple.com/ml-research/papers/learning-with-privacy-at-scale.pdf.

Arachchige PCM, Bertok P, Khalil I, Liu D, Camtepe S, Atiquzzaman M. Local differential privacy for deep learning. IEEE Internet Things J. 2019.

Aronov B, Efrat A, Li M, Gao J, Mitchell JSB, Polishchuk V, Wang B, Quan H, Ding J. Are Friends of My Friends Too Social? Limitations of Location Privacy in a Socially-connected World. In: Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking and Computing; 2018. p. 280–9. New York, NY, USA

Asada M, Yoshikawa M, Cao Y. ”When and Where Do You Want to Hide?”- Recommendation of Location Privacy Preferences with Local Differential Privacy. In: Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy; 2019. p. 164–76.

Ateniese G, Hitaj B, Mancini LV, Verde NV, Villani A. No Place to Hide That Bytes Won’t Reveal: Sniffing Location-based Encrypted Traffic to Track a User’s Position. In: Proceedings of the International Conference on Network and System Security; 2015. p. 46–59.

Attention, Shoppers: Store Is Tracking Your Cell 2019. https://www.nytimes.com/2013/07/15/business/attention-shopper-stores-are-tracking-your-cell.html.

Bassily R, Nissim K, Stemmer U, Thakurta A. Practical Locally Private Heavy Hitters. In: Proceedings of the International Conference on Neural Information Processing Systems; 2017. p. 2285–93.

Bassily R, Smith A. Local, Private, Efficient Protocols for Succinct Histograms. Proceedings of the forty-seventh annual ACM symposium on Theory of computing, 2015. Portland, OR, USA

Beresford AR, Stajano F. Location privacy in pervasive computing. IEEE Pervasive Comput. 2003;2(1):46–55.

Bordenabe NE, Chatzikokolakis K, Palamidess C. Optimal Geo-indistinguishable Mechanisms for Location Privacy. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security; 2014. p. 251–62. New York, NY, USA

Boukoros S, Humbert M, Katzenbeisser S, Troncoso C. On (the Lack of) Location Privacy in Crowdsourcing Applications. In: Proceedings of the USENIX Security Symposium; 2019. p. 1859–76. Santa Clara, CA

Bugador R. The global expansion of UBER in ASIAN markets. International Journal of Supply Chain Management 2019;8(2).

Bun M, Nelson J, Stemmer U. Heavy Hitters and the Structure of Local Privacy. In: Proceedings of the ACM SIGMOD-SIGACT-SIGAI Symposium on Principles of Database Systems; 2018. p. 435–47. Houston, TX, USA

Cao Y, Yoshikawa M, Xiao Y, Xiong L. Quantifying Differential Privacy under Temporal Correlations. In: Proceedings of the IEEE International Conference on Data Engineering; 2017. p. 821–32.

Cao Y, Yoshikawa M, Xiao Y, Xiong L. Quantifying differential privacy in continuous data release under temporal correlations. IEEE Trans Knowl Data Eng 2018;31(7):1281–95.

Chang B, Park Y, Park D, Kim S, Kang J. Content-aware Hierarchical Point-of-interest Embedding Model for Successive POI Recommendation. In: Proceedings of the Twenty-Seventh international joint conference on Artificial Intelligence; 2018. p. 3301–7. Stockholm, Sweden

Chatzikokolakis K, El E, Palamidessi C. Efficient Utility Improvement for Location Privacy. In: Proceedings on Privacy Enhancing Technologies; 2017. p. 210–31. Minneapolis, USA

Chatzikokolakis K, Palamidessi C, Stronati M. Geo-indistinguishability: A Principled Approach to Location Privacy. In: Proceedings of the International Conference on Distributed Computing and Internet Technology; 2015. p. 49–72. Bhubaneswar, India

Chen X, Mizera A, Pang J. Activity Tracking: A New Attack on Location Privacy. In: Proceedings of the IEEE Conference on Communications and Network Security; 2015. p. 22–30.

Cheng C, Yang H, Lyu MR, King I. Where You like to Go Next: Successive Point-of-interest Recommendation. In: Proceedings of the Twenty-Third international joint conference on Artificial Intelligence; 2013. p. 2605–11. Beijing, China

Choi K, Bilich A, Larson KM, Axelrad P. Modified sidereal filtering: Implications for high-rate GPS positioning. Geophys Res Lett 2004;31(22).

Cormode G. Personal Privacy vs Population Privacy: Learning to Attack Anonymization. In: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2011. p. 1253–61.

Cormode G, Kulkarni T, Srivastava D. Marginal Release under Local Differential Privacy. In: Proceedings of the International Conference on Management of Data; 2018. p. 131–46. Houston, TX, USA

Cormode G, Kulkarni T, Srivastava D. Answering Range Queries under Local Differential Privacy. Proceedings of the

International Conference on Very Large Data Bases, 2019. Los Angeles, CA, USA

Cormode G, Procopiuc C, Srivastava D, Shen E, Yu T. Differentially Private Spatial Decompositions. In: Proceedings of the IEEE International Conference on Data Engineering; 2012. p. 20–31. Washington, DC, USA

Cypriani M, Lassabe F, Canalda P, Spies F. Open wireless positioning system: A Wi-Fi-based indoor positioning system. Proceedings of the IEEE Vehicular Technology Conference Fall. USA: Anchorage, AK, 2009.

Dankar FK, Emam KE. The application of differential privacy to health data. Proceedings of the Joint EDBT/ICDT WorkshopsMarch; 2012. p. 158–66.

Dankar FK, Emam KE. Practicing differential privacy in health care: areview. ACM Transactions on Data Privacy 2013;6(1).

De YA, Hidalgo CA, Verleysen M, Blondel VD. Unique in the crowd: the privacy bounds of human mobility. Sci Rep 2013;3:1–5.

Dewri R, Ray I, Ray I, Whitley D. On the Formation of Historically K-Anonymous Anonymity Sets in a Continuous LBS. In: International Conference on Security and Privacy in Communication Systems; 2010. p. 71–88.

Ding B, Kulkarni J, Yekhanin S. Collecting Telemetry Data Privately. In: Proceedings of the International Conference on Neural Information Processing Systems; 2017. p. 3574–83.

Ding B, Nori H, Li P, Allen J. Comparing Population Means under Local Differential Privacy: With Significance and Power. In: Proceedings of the AAAI Conference on Artificial Intelligence; 2018. p. 26–33. New Orleans, LA, USA

Dong Y, Wang S, Li L, Zhang Z. An empirical study on travel patterns of internet based ride-sharing. Transportation research part C: Emerging Technologies 2018;86:1–22.

Duchi JC, Jordan MI, Wainwright MJ. Minimax optimal procedures for locally private estimation. J Am Stat Assoc 2018;113(521):182–215.

Duchi JC, Wainwright MJ, Jordan MI. Local Privacy and Minimax Bounds: Sharp Rates for Probability Estimation. In: Proceedings of the International Conference on Neural Information Processing Systems; 2013. p. 1529–37.

Dwork C. Differential Privacy. In: Proc. Int. Conf. Automata Languages Program.; 2006. p. 1–12. Venice, Italy

Dwork C. Differential Privacy: A Survey of Results. Proceedings of the 5th international conference on Theory and applications of models of computation, 2008.

Dwork C, Mc F, Nissim K, Smith A. Calibrating Noise to Sensitivity in Private Data Analysis. Proceedings of the Third conference on Theory of Cryptography, 2006.

Dwork C, Smith A. Differential privacy for statistics: what we know and what we want to learn. Journal of Privacy and Confidentiality 2010;1(2):135–54.

Eltarjaman W, Dewri R, Thurimella R. Private retrieval of POI details in top-k queries. IEEE Trans. Mob. Comput. 2017;16(9):2611–24.

Erlingsson U, Pihur V, Korolova A. RAPPOR: Randomized Aggregatable Privacy-preserving Ordinal Response. In: Proc. ACM SIGSAC Conf. Comput. Commun. Security; 2014. p. 1054–67. Scottsdale, AZ, USA

Fan L, Xiong L. Real-time Aggregate Monitoring with Differential Privacy. In: Proceedings of the International Conference on Information and Knowledge Management; 2012. p. 2169–73. New York, NY, USA

Fan L, Xiong L, Sunderam V. Differentially Private Multi-dimensional Time Series Release for Traffic Monitoring. In: Proceedings of the IFIP Annual Conference on Data and Applications Security and Privacy; 2013. p. 33–48. Newark, NJ, USA

Fan W, He J, Guo M, Li P, Han Z, Wang R. Privacy preserving classification on local differential privacy in data centers. J Parallel Distrib Comput 2020;135(9):70–82.

Fanaeepour M, Rubinstein BI. Histogramming Privately Ever after: Differentially-private Data-dependent Error Bound Optimisation. In: Proceedings of the IEEE International Conference on Data Engineering; 2018. p. 1204–7. Paris, France

Fanti G, Pihur V, Erlingsson U. Building a RAPPOR with the Unknown: Privacy-preserving Learning of Associations and Data Dictionaries. In: Proceedings of the Privacy Enhancing Technologies Symposium; 2016. p. 41–61.

Fawaz K, Feng H, Shin KG. Anatomization and Protection of Mobile Apps Location Privacy Threats. In: Proceedings of the USENIX Security Symposium; 2015. p. 753–68.

Feldmann S, Kyamakya K, Zapater A, Lue Z. An indoor Bluetooth-based positioning system: Concept i., evaluation. Proc. ICWN 2003;272:109–13.

Fioretto F., Mak T.W.K., Hentenryck P.V.. Privacy-preserving obfuscation of critical infrastructure networks. 2019. arXiv:1905.09778.

Fung B.C.M., Wang K., Yu P.S., preservation T.d.s.f.i., privacy. 2005. Proceedings of the IEEE International Conference on Data Engineering.

G-Divanis A, Kalnis P, Verykios VS. Providing k-anonymity in location based services. ACM SIGKDD Explorations 2010;12(1):1–18.

G-Divanis A, Loukides G, Su J. Publishing data from electronic health records while preserving privacy: a survey of algorithms. J Biomed Inform 2014;50:4–19.

Gedik B, Liu L. Protecting location privacy with personalized k-anonymity: architecture and algorithms. IEEE Trans. Mob. Comput. 2008;7(1):1–18.

Geng Q, Kairouz P, Oh S, Viswanath P. The staircase mechanism in differential privacy. IEEE J Sel Top Signal Process 2015;9(7):1176–84.

Gong M, Feng J, Y X. Privacy-enhanced multi-party deep learning. Neural Networks 2020;121:484–96.

Gong Y, Zhang C, Fang Y, Sun J. Protecting location privacy for task allocation in ad hoc mobile cloud computing. IEEE Trans Emerg Top Comput 2015;6(1):110–21.

Goryczka S, Xiong L. A comprehensive comparison of multiparty secure additions with differential privacy. IEEE Trans Dependable Secure Comput 2017;14(5):463–77.

Gruteser MO, Grunwald D. Anonymous Usage of Location-based Services through Spatial and Temporal Cloaking. In: Proceedings of the International Conference on Mobile Systems, Applications and Services; 2003. p. 31–42. San Francisco, CA, USA

Guo T, Luo J, Dong K, Yang M. Locally differentially private item-based collaborative filtering. Inf Sci (Ny) 2019;502:229–46.

Gutscher A. Coordinate Transformation - a Solution for the Privacy Problem of Location Based Services. Proceedings of the International Parallel and Distributed Processing Symposium, 2006. Rhodes Island, Greece

Harle R. A Survey of Indoor Inertial Positioning Systems for Pedestrians. IEEE Commun. Surv. Tutorials 2013;15(3):2151–66.

Hassan MU, Rehmani MH, Chen J. Differential privacy techniques for cyber physical systems: a survey. IEEE Commun. Surv. Tutorials 2019. Early Access

Hay M, Rastogi V, Miklau G, Suciu D. Boosting the Accuracy of Differentially Private Histograms through Consistency, volume 3; 2010. p. 1021–32.

Hightower J, Borriello G, computing Lsfu. Computer (Long Beach Calif) 2001;34:57–66.

Hofmann-Wellenhof B, Lichtenegger H, Collins J. GPS Theory and practice. New York, NY, USA: Springer; 2001.

Hsu J, Gaboardi M, Haeberlen A, Khanna S, Narayan A, Pierce BC, Roth A. Differential Privacy: An Economic Method for Choosing Epsilon. In: Proceedings of the IEEE Computer

Security Foundations Symposium; 2014. p. 398–410. Vienna, Austria

Huang C, Lu R, Zhu H, Shao J, Alamer A, Lin X. EPPD: Efficient and Privacy-preserving Proximity Testing with Differential Privacy Techniques. In: Proceedings of the IEEE International Conference on Communications; 2016. p. 1–6. Kuala Lumpur, Malaysia

Huang H, Gartner G. Current trends and challenges in location-based services. ISPRS Int J Geoinf 2018;7(6).

SAFECAST Accessed on: Oct. 12, [Online]. Available: https://blog.safecast.org.

2019. Indoor Location-based Services (LBS) Market Analysis Report. https://www.researchandmarkets.com/reports/4661569/indoor-location-based-services-lbs-market.

Jagwani P, Kaushik S. Privacy in location based services: Protection strategies, attack models and open challenges. International Conference on Information Science and Applications 2017:12–21.

Jain P, Gyanchandani M, Khare N. Differential privacy: its technological prescriptive using big data. J Big Data 2018;5(15).

Jang B, Kim H. Indoor positioning technologies without offline fingerprinting map: a survey. IEEE Commun. Surv. Tutorials 2019;21(1):508–25.

Jang B, Sichitiu ML. IEEE 802.11 Saturation throughput analysis in the presence of hidden terminalsm. IEEE/ACM Trans Netw 2012;20(2):557–70.

Ji Z., Lipton Z.C., Elkan C.. Differential privacy and machine learning: A survey and review. 2014. ArXiv preprint arXiv:1412.7584.

Jin W, Xiao M, Li M, Guo L. If You Do Not Care about It, Sell It: Trading Location Privacy in Mobile Crowd Sensing. In: Proceedings of the IEEE Conference on Computer Communications; 2019. p. 1045–53.

Joseph M, Kulkarni J, Mao J, Wu ZS. Locally Private Gaussian Estimation. Proceedings of the Conference on Neural Information Processing Systems, 2019. Vancouver, Canada

Joseph M, Roth A, Ullman J, Waggoner B. Local Differential Privacy for Evolving Data. Proceedings of the Conference on Neural Information Processing Systems, 2018. Montreal, Canada

Kairouz P, Bonawitz K, Ramage D. Discrete Distribution Estimation under Local Privacy. In: Proceedings of the International Conference on International Conference on Machine Learning; 2016. p. 2436–44.

Kido H, Yanagisawa Y, Satoh T. Protection of Location Privacy Using Dummies for Location-based Services. Proceedings of the International Conference on Data Engineering Workshops, 2005. Tokyo, Japan

Kim JS, Chung YD, Kim JW. Differentially private and skew-aware spatial decompositions for mobile crowdsensing. Sensors 2018a;18(11).

Kim JW, Jang B. Workload-aware indoor positioning data collection via local differential privacy. IEEE Commun. Lett. 2019;23(8):1352–6.

Kim JW, Jang B, Yoo H. Privacy-preserving aggregation of personal health data streams. PLoS ONE 2018b;13(11).

Kim JW, Kim DH, Jang B. Application of local differential privacy to collection of indoor positioning data. IEEE Access 2018c;6:4276–86.

Kim JW, Lim JH, Moon SM, Jang B. Collecting health lifelog data from smartwatch users in a privacy-preserving manner. IEEE Trans. Consum. Electron. 2019;65(3):369–78.

Kitasuka T., Hisazumi K., Nakanishi T., Fukuda A., devices W.L., 802.11 m.s.t.o.I.. 2005. Sydney, NSW, Australia. Proc. ICITA, volume 2, 346–349.

Krumm J. Inference attacks on location tracks. International Conference on Pervasive Computing; 2007. p. 127–43.

Lee H, Kim S, Kim JW, Chung YD. Utility-preserving anonymization for health data publishing. BMC Med Inform Decis Mak 2017;17(1).

Lee J, Clifton C. In: Proceedings of the Information Security Conference; 2011. p. 325–40. Xian, China

LeFevre K., DeWitt D.J., Ramakrishnan R., *k* anonymity I.E.F.D. 2005. Proceedings of the ACM SIGMOD International Conference on Management of Data.

LeFevre K., DeWitt D.J., Ramakrishnan R., *k* anonymity M.M. 2006. Proceedings of the IEEE International Conference on Data Engineering.

Leoni D. Non-interactive Differential Privacy: A Survey. In: Proceedings of the First International Workshop on Open Data; 2012. p. 40–52.

Li H, Sun L, Zhu H, Lu X, Cheng X. Achieving Privacy Preservation in Wifi Fingerprint-based Localization. In: Proceedings of the IEEE Conference on Computer Communications; 2014a. p. 2337–45. Toronto, ON, Canada

Li H, Xiong L, Zhang L, Jiang X. DPSynthesizer: Differentially Private Data Synthesizer for Privacy Preserving Data Sharing. Proceedings of the VLDB Endowment, 2014b.

Li N, Li T, Venkatasubramanian S. *T*-Closeness: Privacy beyond *K*-Anonymity and *L*-Diversity. Proceedings of the International Conference on Data Engineering, 2007.

Li X, Mi Z, Zhang Z, Wu J. A Location-aware Recommender System for Tourism Mobile Commerce. In: Proceedings of the International Conference on Information Science and Engineering; 2010. p. 1709–11.

Liu B, Zhou W, Zhu T, Gao L, Xiang Y. Location privacy and its applications: a systematic study. IEEE Access 2019;6:17606–24.

Liu H, Darabi H, Banerjee P, Liu J. Survey of wireless indoor positioning techniques and systems. IEEE Trans Syst Man Cybern 2007;37(6):1067–80.

Lu R, Zhu H, Liu X, Liu JK, Shao J. Toward efficient and privacy-preserving computing in big data era. IEEE Netw 2014;28(4):46–50.

Ma C, Chen CW. Nearby friend discovery with geo-indistinguishability to stalkers. Procedia Comput Sci 2014;34:352–9.

Machanavajjhala A, Kifer D, Abowd J, Gehrke J, Vilhuber L. Privacy: Theory Meets Practice on the Map. Proceedings of the IEEE International Conference on Data Engineering, 2008. Cancun, Mexico

Machanavajjhala A., Kifer D., Gehrke J., Venkitasubramaniam M. *l*-diversity: Privacy beyond *k*-anonymity. ACM transactions on knowledge discovery from data. 2007. volume-1, number-1.

Maruseac M, Ghinita G, Avci B, Trajcevski G, Scheuermann P. Privacy-preserving Detection of Anomalous Phenomena in Crowdsourced Environmental Sensing. In: Proceedings of the International Symposium on Spatial and Temporal Databases; 2015. p. 313–32. Hong Kong, China

Mascetti S, Freni D, Bettini C, Wang X, Jajodia S. Privacy in geo-social networks: proximity notification with untrusted service providers and curious buddies. The International Journal on Very Large Data Bases 2011;20(4):541–66.

McSherry F. Privacy Integrated Queries: An Extensible Platform for Privacy-preserving Data Analysis. Proceedings of the ACM International Conference on Management of Data, 2009. Providence, RI, USA

McSherry FD. Privacy integrated queries: an extensible platform for privacy-preserving data analysis. Commun ACM 2010;53(9).

Mendes R, Cunha M, Vilela Jp. Impact of Frequency of Location Reports on the Privacy Level of Geo-indistinguishability, volume 2020; 2020. p. 379–96.

Micinski K, Phelps P, Foster JS. An Empirical Study of Location Truncation on Android. In: Proceedings of the Mobile Security Technologies; 2013. p. 1–10. San Diego, CA, USA

Murakami T, Hino H, Sakuma J. Toward Distribution Estimation under Local Differential Privacy with Small Sample. In:

Proceedings of the Privacy Enhancing Technologies; 2018. p. 84–104.

Narayanan A, Thiagarajan N, Lakhani M, Hamburg M, Boneh D. Location Privacy via Private Proximity Testing. Proceedings of the Network and Distributed System Security Symposium, 2011. San Diego, CA, USA

Nguyen T.T., Xiao X., Yang Y., Hui S.C., Shin H., Shin J.. Collecting and analyzing data from smart device users with local differential privacy. 2016. arXiv:1606.05053.

Nieminen R, Jrvinen K. Practical privacy-preserving indoor localization based on secure two-party computation. IEEE Trans. Mob. Comput. 2020.

Nissim K, Stemmer U. Clustering Algorithms for the Centralized and Local Models. In: Proceedings of the Machine Learning Research; 2018. p. 619–53.

Niu B, Li Q, Zhu X, Cao G, Li H. Achieving k-anonymity in privacy-aware location-based services. Toronto, ON, Canada: IEEE INFOCOM; 2014.

Ohm P. Broken promises of privacy: responding to the surprising failure of anonymization. UCLA Law Review 2010;57(6):1701–77.

Ou Z, Dong J, Dong S, Wu J, Ylä-Jääski A, Hui P, Wang R, Min AW. Utilize signal traces from others? a crowdsourcing perspective of energy saving in cellular data communication. IEEE Trans. Mob. Comput. 2015;14(1):194–207.

Pahlavan K, Li X, Makela JP, technology Igsa. IEEE Commun. Mag. 2002;40(2):112–18.

Pan X, Zhang J, Wang F, Yu PS. DistSD: Distance-based Social Discovery with Personalized Posterior Screening. In: Proceedings of the IEEE International Conference on Big Data; 2016. p. 1110–19. Washington, DC, USA

Pastore A, Gastpar M. Locally Differentially-private Distribution Estimation. In: Proceedings of the IEEE International Symposium on Information Theory; 2016. p. 2694–8. Barcelona, Spain

Peng S, Yang Y, Zhang Z, Winslett M, Yu Y. Query Optimization for Differentially Private Data Management Systems. Proceedings of the IEEE International Conference on Data Engineering, 2013. Brisbane, QLD, Australia

Popa RA, Blumberg AJ, Balakrishnan H, Li FH. Privacy and Accountability for Location-based Aggregate Statistics. In: Proceedings of the ACM conference on Computer and communications security; 2011. p. 653–66. Chicago, IL, USA

Prasithsangaree P., Krishnamurthy P., Chrysanthis P.K., On indoor position location with wireless LANs. Proc. PIMRC, Lisboa, Portugal 2002.

Primault V, Boutet A, Mokhtar SB, Brunie L. The long road to computational location privacy: asurvey. IEEE Commun. Surv. Tutorials 2018;21(3):2772–93.

Primault V., Mokhtar S.B., Lauradoux C., Brunie L.. Differentially private location privacy in practice. 2014. arXiv:1410.7744.

Pudar NJ, Schwinke SP, Tengler SC. Method of usinghicle location informationth a wireless mobile device. U S Patent 2014(8). 744,745

Qardaji W, Yang W, Li N. Differentially Private Grids for Geospatial Data. In: Proceedings of the IEEE International Conference on Data Engineering; 2013. p. 757–68. Brisbane, QLD, Australia

Qin Z, Yang Y, Yu T, Khalil I, Xiao X, Ren K. Heavy Hitter Estimation over Set-valued Data with Local Differential Privacy. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security; 2016. p. 192–203. Vienna, Austria

Qiu C, Squicciarini AC. Location Privacy Protection in Vehicle-based Spatial Crowdsourcing via Geo-indistinguishability. In: Proceedings of the IEEE International Conference on Distributed Computing Systems; 2019. p. 1061–71. Dallas, TX, USA

Rao FY, Bertino E. Privacy techniques for edge computing systems. Proc. IEEE 2019;107(8):1632–54.

Rashid O, Coulton P, Edwards RC. Providing location based information/advertising for existing mobile phone users. Pers Ubiquitous Comput 2008;12(1):3–10.

Ren X, Yu CM, Yu W, Yang S, Yang X, McCann JA, Yu PS. Lopub: high-dimensional crowdsourced data publication with local differential privacy. IEEE Trans. Inf. Forensics Secur. 2018;13(9):2151–66.

Rodriguez-Hernandez MdC, Ilarri S, Trillo-Lado R, Hermoso R. Location-aware Recommendation Systems: Where We Are and Where We Recommend to Go. Proceedings of the Workshop on Location-Aware Recommendations, 2015. Vienna, Austria.

Roxin A, Gaber J, Wack M, Nait-Sidi-Moh A. Survey of Wireless Geolocation Techniques. In: Proceedings of the IEEE Globecom Workshops; 2007. p. 1–9. Washington, DC.

Sandholm T, Ung H. Real-time, Location-aware Collaborative Filtering of Web Content. In: Proceedings of the Workshop on Context-awareness in Retrieval and Recommendation; 2011. p. 14–18.

Shi D, Ding J, Errapotu SM, Yue H, Xu W, Zhou X, Pan M. Deep q-network-based route scheduling for TNC vehicles with passengers' location differential privacy. IEEE Internet Things J. 2019;6(5).

Shi Z, Zhang Z, Shu Y, Cheng P, Chen J. Indoor Navigation Leveraging Gradient Wifi Signals. In: Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems; 2017. p. 1–2. Delft, Netherlands

Shin H, Kim S, Shin J, Xiao X. Privacy enhanced matrix factorization for recommendation with local differential privacy. IEEE Trans Knowl Data Eng 2018;30(9):1770–82.

Shokri R, Shmatikov V. Privacy-preserving Deep Learning. In: Proceedings of the ACM SIGSAC conference on computer and communications security; 2015. p. 1310–21. Denver, CO, USA

Shokri R, Theodorakopoulos G, Boudec JL, Hubaux J. Quantifying Location Privacy. In: Proceedings of the IEEE Symposium on Security and Privacy; 2011. p. 247–62. Berkeley, CA

Soria-Comas J, Domingo-Fer J. Optimal data-independent noise for differential privacy. Inf Sci (Ny) 2013;250:200–14.

Sun H, Dong B, Wang H, Yu T, Qin Z. Truth Inference on Sparse Crowdsourcing Data with Local Differential Privacy. Proceedings of the IEEE International Conference on Big Data, 2018. Seattle, WA, USA

Sweeney L. K-anonymity: A model for protecting privacy. Int. J. Uncertainty Fuzziness Knowledge Based Syst. 2002;10(5):557–70.

Takeuchi Y, Sugimoto M. Cityvoyager: an outdoor recommendation system based on user location history. Proceedings of the International Conference on Ubiquitous Intelligence and Computing 2005:625–35.

Tang J., Korolova A., Bai X., Wang X., Wang X. Privacy loss in Apple's implementation of differential privacy on MacOS 10.12. arXiv:1709.02753, 2017.

Tehrani PF, Restel H, Jendreck M, Pfennigschmidt S, Hardt M, Meissen U. Toward Privacy by Eesign in Spatial Crowdsourcing in Emergency and Disaster Response. In: Proceedings of the International Conference on Information and Communication Technologies for Disaster Management; 2018. p. 1–9. Sendai, Japan

Teng X, Guo D, Zhou X, Liu Z. An Indoor-outdoor Navigation Service for Subway Transportation Systems. In: Proceedings of the 13th ACM Conference on Embedded Network Sensor Systems; 2015. p. 415–16. Seoul, South Korea

Terrovitis M. Privacy preservation in the dissemination of location data. ACM SIGKDD Explorations Newsletter 2011;13(1):6–18.

To H, Fan L, Shahabi C. Differentially Private H-tree. In: Proceedings of the 2nd Workshop on Privacy in Geographic Information Collection and Analysis; 2015. p. 1–8. Bellevue, WA, USA

To H, Ghinita G, Shahabi C. A Framework for Protecting Worker Location Privacy in Spatial Crowdsourcing, volume 7; 2014. p. 919–30.

Tong W, Hua J, Zhong S. A jointly differentially private scheduling protocol for ridesharing sservices. IEEE Trans. Inf. Forensics Secur. 2017;12(10):2444–56.

Usman M, Jan MA, Puthal D. PAAL: A framework based on authentication, aggregation and local differential privacy for internet of multimedia things. IEEE Internet of Things Journal, Early Access 2019.

Yahoo! Weather Website https://mobile.yahoo.com/weather/.

Wang J, Liu S, Li Y. A review of differential privacy in individual data release. Int. J. Distrib. Sens. Netw. 2015a;11(10).

Wang L, Yang D, Han X, Wang T, Zhang D, Ma X. Location Privacy-preserving Task Allocation for Mobile Crowdsensing with Differential Geo-obfuscation. In: Proceedings of the International Conference on World Wide Web; 2017a. p. 627–36.

Wang N, Xiao X, Yang Y, Hoang TD, Shin H, Shin J, Yu G. Privtrie: Effective Frequent Term Discovery under Local Differential Privacy. In: Proceedings of the IEEE International Conference on Data Engineering; 2018a. p. 821–32. Paris, France

Wang N, Xiao X, Yang Y, Zhao J, Hui SC, Shin H, Shin J, Yu G. Collecting and Analyzing Multidimensional Data with Local Differential Privacy. Proceedings of the International Conference on Data Engineering, 2019a. Macao

Wang S, Huang L, Nie Y, Zhang X, Wang P, Xu H, Yang W. Local differential private data aggregation for discrete distribution estimation. IEEE Trans. Parallel Distrib. Syst. 2019b;30(9):2046–59.

Wang S, Huang L, Wang P, Shen Y, Xu H, Yang W. Privacy Preserving Big Histogram Aggregation for Spatial Crowdsensing. In: Proceedings of the IEEE International Performance Computing and Communications Conference; 2015b. p. 1–8. Nanjing, China

Wang S, Nie Y, Wang P, Xu H, Yang W, Huang L. Local Private Ordinal Data Distribution Estimation. Proceedings of the IEEE Conference on Computer Communications, 2017b. Atlanta, GA, USA

Wang T, Blocki J, Li N, Jha S. Locally Differentially Private Protocols for Frequency Estimation. Proceedings of the 26th USENIX Conference on Security Symposium, 2017c. Berkeley, CA, USA

Wang T, Ding B, Zhou J, Hong C, Huang Z, Li N, Jha SK. Answering Multi-dimensional Analytical Queries under Local Differential Privacy. In: Proceedings of the International Conference on Management of Data; 2019c. p. 159–76. Amsterdam, Netherlands

Wang T, Li N, Jha S. Locally Differentially Private Frequent Itemset Mining. In: Proceedings of the IEEE Symposium on Security and Privacy; 2018b. p. 127–43. San Francisco, CA, USA

Wang T, Li N, Jha S. Locally differentially private heavy hitter identification. IEEE Trans Dependable Secure Comput 2019d. Early Access

Wang X, Zheng X, Zhang Q, Wang T, Shen D. Crowdsourcing in ITS: the state of the work and the networking. IEEE Trans. Intell. Transp. Syst. 2016;17(6):1596–605.

Wang Y, Huang M, Jin Q, Ma J. DP3: A differential privacy-based privacy-preserving indoor localization mechanism. IEEE Commun. Lett. 2018c;22(12):2457–550.

Wang K, Yu PS, Chakraborty S. In: Proceedings of the IEEE International Conference on Data Mining. Bottom-up generalization: A data mining solution to privacy protection; 2014.

Wang Z, Hu J, Lv R, Wei J, Wang Q. Personalized privacy-preserving task allocation for mobile crowdsensing. IEEE Trans. Mob. Comput. 2018d;18(6):1330–41.

Warner SL. Randomized response: a survey technique for eliminating evasive answer bias. J Am Stat Assoc 1965;60(309).

Wu D, Zhang Y, Bao L, Regan AC. Location-based crowdsourcing for vehicular communication in hybrid networks. IEEE Trans. Intell. Transp. Syst. 2013;14(2):837–46.

Xiao X, Bender G, Hay M, Gehrke J. Ireduct: Differential Privacy with Reduced Relative Errors. Proceedings of the ACM SIGMOD International Conference on Management of data, 2011a. Athens Greece

Xiao X, Wang G, Gehrke J. Differential privacy via wavelet transforms. IEEE Trans Knowl Data Eng 2011b;23(8):1200–14.

Xiao Y, Jia Y, Cheng X, Yu J, Liang Z, Tian Z. I can see your brain: investigating home-use electroencephalography system security. IEEE Internet Things J. 2019;6(4):6681–91.

Xiao Y, Xiong L, Yuan C. Differentially Private Data Release through Multidimensional Partitioning. In: Proceedings of the Workshop on Secure Data Management; 2010. p. 150–68. Singapore

Xiong X, Liu S, Li D, Cai Z, Niu X. A comprehensive survey on local differential privacy. Security and Communication Networks 2020.

Xu C, Ren J, She L, Zhang Y, Qin Z, Ren K. Edgesanitizer: locally differentially private deep inference at the edge for mobile data analytics. IEEE Internet Things J. 2019;6(3):5140–51.

Xue M, Liu Y, Ross KW, Qian H. I Know Where You Are: Thwarting Privacy Protection in Location-based Social Discovery Services. In: Proceedings of the IEEE Conference on Computer Communications Workshops; 2015. p. 179–84. Hong Kong, China

Yan K, Luo G, Zheng X, Tian L, Sai AMVV. A comprehensive location-privacy-awareness task selection mechanism in mobile crowd-wensing. IEEE Access 2019;7:77541–54.

Wang Y, Jia X, Lee HK, Li GY. An indoors wireless positioning system based on wireless local area network infrastructure. Proceedings of the International Symposium on Satellite Navigation Technology Including Mobile Positioning & Location Services. Australia: Melbourne VIC, 2003.

Yang M., Lyu L., Zhao J., Zhu T., Lam K.Y. Local differential privacy and its applications: A comprehensive survey. 2020. arXiv:2008.03686.

Yang D, Fang X, Xue G. Truthful incentive mechanisms for k-anonymity location privacy. Proceedings of the IEEE INFOCOM, Turin, Italy, 2013.

Yang X, Wang T, Ren X, Yu W. Survey on improving data utility in differentially private sequential data publishing. IEEE Trans. Big Data 2017. Early Access

Yao X, Chen Y, Zhang R, Zhang Y, Lin Y. Beware of what you share: inferring user locations in venmo. IEEE Internet Things J. 2018;5(6):5109–18.

Yao X, Zhou X, Ma J. Differential Privacy of Big Data: An Overview. In: Proceedings of the IEEE International Conference on Big Data Security on Cloud; 2012. p. 158–66.

Ye M, Barg A. Optimal schemes for discrete distribution estimation under locally differential privacy. IEEE Trans. Inf. Theory 2018;64(8):5662–76.

Ye Q, Hu H. Local Differential Privacy: Tools, Challenges, and Opportunities. In: Proceedings of the WISE 2019 Workshop, Demo, and Tutorial; 2019. p. 13–23. Hong Kong and Macau, China

Ye Q, Hu H, Meng X, Zheng H. PrivKV: Key-value Data Collection with Local Differential Privacy. Proceedings of the IEEE Symposium on Security and Privacy, 2019. San Francisco, CA, USA

You T, Peng W, Lee W. Protecting Moving Trajectories with Dummies. Proceedings of the International Conference on Mobile Data Management, 2007. Mannheim, Germany

Zhang J, Guo H, Liu J, Zhang Y. Task offloading in vehicular edge computing networks: a load-balancing solution. IEEE Trans. Veh. Technol. 2020;69(2):2092–104.

Zhang J, Xiao X, Xie X. Privtree: A Differentially Private Algorithm for Hierarchical Decompositions. In: Proceedings of the International Conference on Management of Data; 2016. p. 155–70. San Francisco, CA, USA

Zhang JD, Ghinita G, Chow CY. Differentially Private Location Recommendations in Geosocial Networks. In: Proceedings of the IEEE International Conference on Mobile Data Management; 2014. p. 59–68. Brisbane, QLD, Australia

Zhang X, Wang J, Shu M, Wang Y, Pan M, Han Z. TPP: Trajectory privacy preservation against tensor voting based inference attacks. IEEE Access 2018a;6:77975–85.

Zhang Z, Wang T, Li N, He S, Chen J. Calm: Consistent Adaptive Local Marginal for Marginal Release under Local Differential Privacy. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security; 2018b. p. 212–29. Toronto, Canada

Zhao J, Chen Y, Zhang W. Differential privacy preservation in deep learning: challenges, opportunities and solutions. IEEE Access 2019a;7:48901–11.

Zhao P, Zhang G, Wan S, Liu G, Umer T. A survey of local differential privacy for securing internet of vehicles. J Supercomput 2019b. Early Access

Zhou L, Yu L, Du S, Zhu H, Chen C. Achieving differentially private location privacy in edge-assistant connected vehicles. IEEE Internet Things J. 2019;6(3).

Zhu T, Li G, Zhou W, Yu PS. Differentially private data publishing and analysis: a survey. IEEE Trans Knowl Data Eng 2017;29(8):1619–38.

Jong Wook Kim received the Ph.D. degree from the Computer Science Department, Arizona State University, in 2009. He was a Software Engineer with the Query Optimization Group at Teradata, from 2010 to 2013. He is currently an Associate Professor of computer science with Sangmyung University. His primary research interest is in the area of data privacy, distributed databases, and query optimization.

Kennedy Edemacu received the B.S degree in Computer Science from Gulu University in 2011, the M.S degree in Data Communication and Software Engineering from Makerere University in 2014 and he is currently working towards the Ph.D degree in Computer Science in Sangmyung University. He worked as an Assistant lecturer in Muni University from 2013 to 2016. His current research interests include: Privacy in Cloud Computing, Cryptography and Artificial Intelligence.

Jong Seon Kim is currently working towards the Ph.D degree in Computer Science in Korea University. His primary research interest is in the area of data privacy and recommender system.

Yon Dohn Chung received the B.S. degree in computer science from Korea University, Seoul, in 1994, and the M.S. and Ph.D. degrees in computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 1996 and 2000, respectively. He was an Assistant Professor with the Department of Computer Engineering, Dongguk University, Seoul, South Korea, from 2003 to 2006. He joined the Faculty of the Department of Computer Science and Engineering, Korea University, in 2006, where he is currently a Full Professor. His research interests include spatial databases, data privacy, array databases, and distributed/parallel processing of large-scale data.

Beakcheol Jang received the B.S. degree from Yonsei University in 2001, the M.S. degree from the Korea Advanced Institute of Science and Technology in 2002, and the Ph.D. degree from North Carolina State University in 2009, all in computer science. .He is currently an Associate Professor with the Graduate School of Information, Yonsei University. His primary research interests include wireless networking, big data, Internet of Things, and artificial intelligence.