DistPreserv: Maintaining User Distribution for Privacy-Preserving Location-Based Services

Yanbing Ren, Xinghua Li, *Member, IEEE*, Yinbin Miao, Robert H. Deng, *Fellow, IEEE*, Jian Weng, *Member, IEEE*, Siqi Ma, and Jianfeng Ma, *Member, IEEE*

Abstract—Location-Based Services (LBSs) are one of the most frequently used mobile applications in the modern society. Geo-Indistinguishability (Geo-Ind) is a promising privacy protection model for LBSs since it can provide formal security guarantees for location privacy. However, Geo-Ind undermines the statistical location distribution of users on the LBS server because of perturbed locations, thereby disabling the server to provide distribution-based services (e.g., traffic congestion maps). To overcome this issue, we give a privacy definition, called DistPreserv, to enable the LBS server to acquire valid location distributions while providing users with strict location protection. Then we propose a privacy-preserving LBS scheme to benefit both users and the server, in which a location perturbation mechanism is designed to achieve the given definition under the guide of the incentive compatibility, and a retrieval area determination method is presented to ensure query accuracy of users by using the dynamic programming on the two-dimensional map plane. Finally, we theoretically prove that the designed mechanism can achieve the definition of DistPreserv and the property of incentive compatibility. Experimental explorations using a real-world dataset indicate that our proposal prominently improves the availability of users' location distributions by over 90%, while providing high precision and recall of queries.

Index Terms—Location privacy, query accuracy, location distributions, incentive compatibility, location-based services

1 Introduction

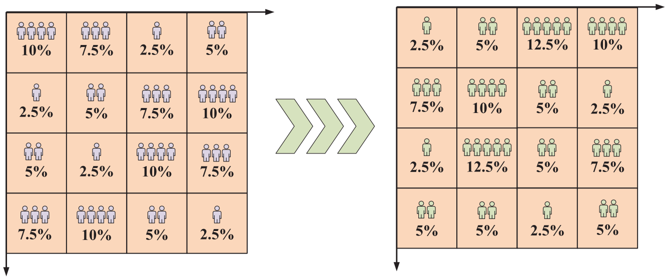
With the growing popularity of mobile devices equipped with GPS chips and the increasing availability of wireless data connections, Location-Based Services (LBSs), which enable a user to obtain real-time services related to his/her current location, have gained much attention from both academic and industrial fields. A recent business research predicts that the market of global LBSs can rise at a robust 19.9% CAGR (Compound Annual Growth Rate) between 2017 and 2025, where the market will be worth US$99.77 billion [1].

These LBSs can facilitate the users' daily lives, but still cause serious privacy concerns. When users report their current locations to query for nearby Points-of-Interests (POIs), the LBS server may collect their locations and learn about sensitive information related to them such as home addresses, income levels, *etc.*, thereby posing threats to users' privacy or even personal safety. Thus, the issue of location privacy is being a key factor in determining the popularity of LBSs in the coming years [2], [3]. To protect location privacy for LBSs, a series of approaches have been proposed on the basis of traditional privacy models, such as k-anonymity [4], l-diversity [5] and t-closeness [6], *etc.* Since these models are designed heuristically, they cannot provide strict and formal privacy guarantees [7], [8], [9]. To solve the above issue, Geo-Indistinguishability (Geo-Ind) [10] proposed based on differential privacy provides a strict location perturbation paradigm for protecting users' location privacy. According to Geo-Ind, a user submits a perturbed location and a retrieval size to get nearby POIs in a privacy-preserving way by adding noise to his/her current location. Thus, Geo-Ind has become a hot research topic for location privacy and has been put into practical use (e.g., SpatialVision, Location-Guard) due to its strict privacy definition and convenient implementation [11], [12], [13], [14], [15].

Although Geo-Ind can protect a user's location privacy effectively, it undermines the statistical location distribution of users on the LBS server since it adds noise in reported locations without considering the distribution. In fact, the statistical location distribution is important for LBSs since it enables location service providers (i.e., LBS servers) to learn the overall user distribution on the spatial domain and further acquire a real insight of spatial patterns of users. Specifically, location distributions can be used for many purposes, such as detecting the popularity of scenic spots, perceiving traffic jams, and warning crowded areas during the COVID-19 pandemic, *etc.* [16], [17], [18]. Nevertheless, existing Geo-Ind works mainly focus on the location privacy protection on the user

- *Yanbing Ren, Yinbin Miao, and Jianfeng Ma are with the State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an 710071, China. E-mail: yanbing_ren@foxmail.com, ybmiao@xidian.edu.cn, jfma@mail.xidian.edu.cn.*
- *Xinghua Li is with the State Key Laboratory of Integrated Services Networks, School of Cyber Engineering, Xidian University, Xi'an 710071, China, and also with the Engineering Research Center of Big Data Security, Ministry of Education, Xi'an 710071, China. E-mail: xhli1@mail.xidian.edu.cn.*
- *Robert H. Deng is with the School of Information Systems, Singapore Management University, Singapore 178902. E-mail: robertdeng@smu.edu.sg.*
- *Jian Weng is with the College of Cyber Security, Jinan University, Guangzhou 510632, China. E-mail: cryptjweng@gmail.com.*
- *Siqi Ma is with the School of Information Technology and Electrical Engineering, University of Queensland, St Lucia, QLD 4072, Australia. E-mail: xdmasiqi@hotmail.com.*

Manuscript received 21 Aug. 2020; revised 29 Dec. 2021; accepted 4 Jan. 2022. Date of publication 7 Jan. 2022; date of current version 5 May 2023.
This work was supported in part by the National Natural Science Foundation of China under Grants 62125205, U1708262, U1736203, and 62072361 in part by the Key Research and Development Program of Shaanxi under Grant 2021ZDLGY05-04, and in part by the Fundamental Research Funds for the Central Universities under Grant JB211505.
(Corresponding author: Xinghua Li.)
Digital Object Identifier no. 10.1109/TMC.2022.3141398



Real distribution of LBS users **Distribution after perturbations**

Fig. 1. A false statistical location distribution can be derived by the service provider resulting from perturbed locations in Geo-Ind.

side, while neglecting the availability requirement of users' location distributions on the server side, which results in false statistical location distributions, as illustrated in Fig. 1. This disables the LBS server to provide correct distribution-based services, which will do harm to the promotion and adoption of Geo-Ind in LBSs. Through extensive experiments we have confirmed that the average statistical distribution changes dramatically after users' Geo-Ind location perturbations, as shown in Figs. 5, 6, and 7.

To address this issue, it is critical to take the user distribution into account while generating reported noisy locations. Therefore, from the perspective of the user distribution availability on the LBS server side, we give a new privacy definition that can provide users with provable location privacy guarantees while allowing the LBS server to obtain the valid distribution information from queries. Then a privacy-preserving LBS scheme is presented to achieve the defined definition. As far as we know, we are the first effort to take into account of the privacy-preserving statistical location distribution of users in LBSs. Specifically, the major contributions of this paper are summarized below:

- We give a new privacy definition, called DistPreserv, as the enhancement of Geo-Ind with considerations of users' location distributions. Specifically, DistPreserv largely maintains the users' collective location distributions on the basis of location privacy protection. This feature is achieved by requiring the reported locations and true locations to be indistinguishable in both euclidean distance and distribution differences.
- We design a privacy-preserving scheme for LBSs. First, to achieve the definition of DistPreserv, a location perturbation mechanism is designed according to differential privacy exponential mechanism under the guidance of incentive compatibility. Then, a dynamic programming method on the two-dimensional map plane is utilized to determine the retrieval area, thereby achieving high accuracy of queries with privacy guarantees.
- We provide theoretical analysis to show that our scheme satisfies both the definition of DistPreserv and the property of incentive compatibility, and analyze that DistPreserv can achieve lower distribution divergence. Furthermore, we conduct extensive experiments using a real-world dataset to demonstrate that our proposal prominently improves the availability of user distribution by over 90% when compared with the classic Geo-Ind approach.

In Section 2, we introduce some related works briefly. In Section 3, we describe some preliminaries and problem formulations. Section 4 presents the privacy definition in detail. Section 5 gives our designed privacy-preserving scheme. The theoretical analysis is provided in Section 6. Experimental results are demonstrated in Section 7. Finally, we give a conclusion in Section 8.

2 Related Work

The location privacy protection for LBSs has attracted much attention due to its necessity [2], and efforts to provide users with location privacy have experienced a long-term development.

2.1 Traditional Location Privacy Protection Methods

To achieve the goal of location privacy, some works have given suggestions relying on cryptographic measures such as homomorphic encryption and Private Information Retrieval (PIR) [19], [20], [21], [22], [23], but these solutions cannot be widely deployed in practice due to heavy computational burdens and little interactivity with existing LBSs [2]. To protect location privacy more efficiently, a sequence of efforts have explored this challenge by introducing computational privacy models (e.g., k-anonymity, l-diversity) into location privacy. Since the k-anonymity was proposed, there has been a series of works using this notion to prevent users' location privacy from being leaked [4], [24], [25], where the intuitive idea is to obscure the user's true location with other $k-1$ fake locations.

The above schemes equipped with k-anonymity do not consider the semantics of $k-1$ fake locations, thus several works based on l-diversity [5] were promoted to provide location privacy protection for LBSs [26], [27]. In l-diversity, it is required to have l different semantic features among k locations based on k-anonymity. However, l-diversity is still claimed to be insufficient for location privacy in some works, thus the notion of t-closeness [6] is introduced. Instead of just guaranteeing l-diversity of locations, t-closeness based works further require that the distribution of the semantic features inside k anonymous locations is as similar as possible to the distribution of these features in the total number of users [28], [29]. Nevertheless, these privacy models cannot render formal privacy guarantees since they are designed according to the methodology of heuristics thereby lacking a strictly theoretical basis [2], [9].

2.2 Differential Privacy and Geo-Ind Based Approaches

To deal with the above issue, Andrés *et al.* [10] proposed a location privacy model namely Geo-Ind to protect LBS users' location privacy effectively. Specifically, they first gave the definition of Geo-Ind by adapting generalized differential privacy[30] to the LBS scenario. Then they designed a location perturbation mechanism to achieve this definition through the distribution of planar Laplace. After that, they discussed the accuracy of the LBS query and gave a method to calculate the retrieval radius in the query. Similar to the differential privacy, the outstanding characteristic of Geo-Ind is that no matter what priori knowledge the adversary has, it cannot get more about the user's true location based on observing the reported location generated by the Geo-Ind mechanism. This makes Geo-Ind an attractive privacy model to protect location privacy in LBSs.

In recent works, taking into account the constraints of road networks, Qiu *et al.* [31] designed a location perturbation mechanism to implement Geo-Ind on road networks with the help of linear programming. Besides, aiming at the vulnerability of existing Geo-Ind mechanisms under long-term observation attacks, Niu *et al.* [32] combined Geo-Ind with k-anonymity and proposed a new location protection mechanism, which uses the differential privacy exponential mechanism to generate the perturbed location from predetermined possible outputs. Since the above works did not consider the issue of privacy budget consumption of the user, Hua *et al.* [33] proposed a location perturbation mechanism that divides a targeted place into cell layouts to slow down the privacy budget consumption of Geo-Ind. By shifting perturbed locations to the center points of their corresponding cells, this mechanism reduces the privacy cost for a single LBS query.

However, the remarkable defect shared by existing schemes is that they do not take into account of the availability of user distribution, which severely disrupts the spatial patterns of users, regardless of its critical values on LBSs.

It is worth noting that some works [17], [34], [35] adopt local differential privacy [36], the variant of differential privacy for user-specified privacy intensity, to keep users' whereabouts private while allowing the server to obtain their distribution information. However, the major difference between these works and our work is that they can not provide users with LBSs since they mainly focus on acquiring users' collective information in a privacy-preserving way. Without the restriction of LBS accuracy, users in these works can produce pseudo-locations without considering the distance between the fake location and true location, which is not allowed in our scenario.

3 Preliminaries and Problem Formulations

In this section, we first concisely introduce some preliminaries, and then describe our system and threat model. Finally, we present the design goal of our work. The symbols and notations used frequently are listed in Table 1.

TABLE 1
Symbols and Notations Used in Our Paper

Symbol	Description
ε	The privacy parameter set by a user
$\mathcal{M}$	Randomized mechanism to achieve differential privacy
K	Randomized mechanism to achieve Geo-Ind
$\mathcal{K}$	Randomized mechanism to achieve DistPreserv
$d(\cdot,\cdot)$	euclidean distance between two locations
$\Pr(\cdot)$	Probability of an event
x_0	True location of a user
z	Reported location of a user for the LBS query
G	The considered macro area where the user locates
D_G	Current users' location distribution in G
f_{x_i}	The users' request rate at location x_i
u	Utility function to score possible perturbed locations
c	The accuracy requirement of the LBS query
$\mathcal{C}(x,r)$	A circle with a center x and a radius r

3.1 Preliminaries

3.1.1 Differential Privacy and Geo-Indistinguishability

Differential privacy is an attractive privacy model first proposed in the field of statistical databases [37], which can prevent the user's privacy from being revealed from aggregated queries. Since it can provide a formal privacy guarantee abstracting from the adversary's priori knowledge, differential privacy has become the mainstream paradigm in the field of privacy protection.

Definition 1 (differential privacy). *A randomized algorithm $\mathcal{M}$ is $\varepsilon-$differential privacy if for all subset $\mathcal{S} \subseteq Range(\mathcal{M})$ and all datasets x, y with $\|x-y\|_1 \leq 1$, it has $\Pr[\mathcal{M}(x) \in \mathcal{S}] \leq \exp(\varepsilon)\Pr[\mathcal{M}(y) \in \mathcal{S}]$.*

In the above definition, ε is a positive real number and reflects the level of required privacy, and a smaller ε implies the higher expected privacy level. Besides, Hamming distance is used to measure the difference between the mechanism's possible inputs and it is required that the difference of the compared inputs is at most one Hamming distance in the definition. Thus, to let the metrics of distance no longer be confined to Hamming distance and the difference of inputs no longer limited by one, the generalized differential privacy [30] is developed as a more abstract privacy definition, and then it is adapted to the scenario of LBSs by introducing geo-indistinguishability [10].

Definition 2 (Geo-indistinguishability). *A randomized algorithm $K : \mathcal{X} \rightarrow \mathcal{D}(\mathcal{Z})$ is ε-geo-indistinguishability iff $\forall x, x' \in \mathcal{X}$ it has $d_{\mathcal{D}}(K(x), K(x')) \leq \varepsilon d(x, x')$.*

In this definition, $\mathcal{X}$ and $\mathcal{Z}$ denote the set of a user's all possible true and perturbed locations, respectively. Besides, $\mathcal{D}(\mathcal{Z})$ denotes a probability distribution on $\mathcal{Z}$, so that $K(x)$ indicates the distribution of possible perturbed locations while the true location is x. Besides, it is defined that $d_{\mathcal{D}}(\varpi_1, \varpi_2) = \sup_{Z \subseteq \mathcal{Z}} |\ln \frac{\varpi_1(Z)}{\varpi_2(Z)}|$ to measure the difference between two distributions ϖ_1 and ϖ_2 with the rule that $|\ln \frac{\varpi_1(Z)}{\varpi_2(Z)}| = 0$ if both $\varpi_1(Z)$ and $\varpi_2(Z)$ are zero, and $|\ln \frac{\varpi_1(Z)}{\varpi_2(Z)}| = \infty$ if only one of them is zero. ε is the privacy parameter set by the user and $d(x, x')$ denotes the euclidean distance between x and x'. Note that this definition can also be presented as $K(x)(Z) \leq e^{\varepsilon d(x,x')} K(x')(Z)$ for all $x, x' \in \mathcal{X}$, $Z \subseteq \mathcal{Z}$, in which $K(x)(Z)$ denotes the probability of the user's perturbed location z being in the set Z while his/her true location is x.

Intuitively, the privacy of Geo-Ind comes from requiring that any two close locations should be perturbed to the same location with indistinguishable probabilities. If the user intends to achieve stronger privacy protection, he/she needs to make the value of the privacy parameter ε smaller.

3.1.2 Incentive Compatibility

Incentive compatibility [38] is the concept to characterize those mechanisms in which participants would not find it advantageous to violate the rules of the process, and these rules are formulated to provide public benefits. This means that individual interests and collective interests are compatible, and no one can expand their own interests by

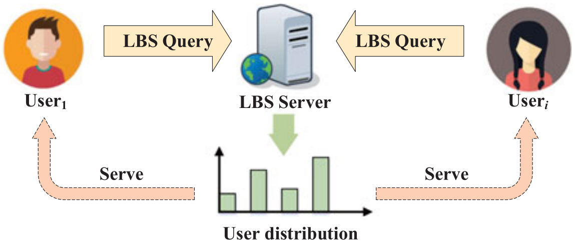


Fig. 2. System model of our proposal.

damaging collective interests. Formally, the definition of incentive compatibility can be defined below.

Definition 3 (incentive compatibility). *If a mechanism satisfies the incentive compatibility, the following condition* $User_i \in \{User_i | i \in \mathbb{N}^*\}$: $v_i(\chi_i) \geq v_i(\tilde{\chi}_i)$ *should be hold for any situation that satisfies* $b(\chi_1, \chi_2, \ldots, \chi_n) \geq b(\tilde{\chi}_1, \tilde{\chi}_2, \ldots, \tilde{\chi}_n)$, *where* $v_i(\cdot)$ *represents the individual utility obtained by* $User_i$, *and* $b(\cdot)$ *represents the public utility obtained by users for a specific strategy combination of them.* χ_i *and* $\tilde{\chi}_i$ *represent the strategies that* $User_i$ *follows and not follows, respectively. Specifically, in the strategy of* χ_i, $User_i$ *reports his/her true location to query if he/she is in an insensitive location without privacy protection. Otherwise, he/she adopts our proposal to perform the LBS query instead of reporting true location or adopting Geo-Ind approaches.*

Note that in some scenarios (e.g., auctions), $b(\chi_1, \chi_2, \ldots, \chi_n)$ is simply defined as $v_1(\chi_1) + v_2(\chi_2) + \ldots + v_n(\chi_n)$, yet in our scenario $v_i(\chi_i)$ represents the comprehensive services received by $User_i$ according to his/her privacy preference, and $b(\chi_1, \chi_2, \ldots, \chi_n)$ denotes the availability of overall users' distributions counted by their reported locations.

3.2 System Model & Threat Model

System Model. In our work, the system model is consistent with the general LBS framework that consists of users and an LBS server [2]. We consider a set of users that send queries based on their locations to the LBS server to obtain nearby POIs (e.g., discovering nearby hotels or restaurants). The location contained in each query can be the current location of the user or another one generated according to the user's different privacy requirements. Users can obtain two aspects of utility from LBSs, one is the individual utility achieved by obtaining the information of nearby POIs from the LBS server, the other is the public utility which corresponds to the user distribution aggregated from their reported locations. At the same time, users expect their whereabouts to be hidden from the LBS server that can observe the locations in the queries. The framework of the system is depicted in Fig. 2.

Instead of making users simply submit their locations as LBS queries, we propose a new paradigm of user-server interaction to ensure that the user distribution does not collapse while protecting location privacy of users. When a user from $U = \{User_i | i \in \mathbb{N}^*\}$ initiates an LBS query, he/she first reports his/her macro location G (e.g., the city) to the LBS server to request the current user distribution D_G in this area. After receiving the user's macro location G, the LBS server returns D_G to the user as a response. Then, the user adds noise to his/her location to produce a perturbed location according to the true location and D_G (which is detailed in Algorithm 1), and reports the pseudo-location to the LBS server along with the retrieval radius for querying POIs. After receiving the query, the LBS server retrieves POIs in its database according to the received location and the retrieval radius, and then returns the query results to the user. Finally, the user filters the received POIs based on his/her true location and actual area of interest.

Threat Model. Similar to the general assumption for LBSs, the LBS server itself is regarded as the honest-but-curious adversary [39], [40], [41], [42], which means that it will honestly provide services to users according to established rules but may be curious to infer users' true locations from their LBS queries. More formally, we introduce an adversary $\mathcal{A}^*$ whose target is the LBS server. The capability of $\mathcal{A}^*$ is described as follows: $\mathcal{A}^*$ can compromise the LBS server to infer users' true locations based on the received LBS queries.

Moreover, it is assumed that the users in the system are rational. Specifically, when the user is currently privacy-insensitive, he/she prefers to report his/her true location in the LBS query for achieving better services. Otherwise, he/she prefers to perform a privacy-preserving LBS query to make his/her whereabouts private. In both cases above, the user does not need to care about the privacy of other users. Besides, since the LBS server is considered honest-but-curious, it cannot be seen as a rational participant, but only as an adversary.

3.3 Design Goals

In our work, we aim at protecting the location privacy of users in the process of LBS query, while making their overall location distribution obtained by the LBS server as available as possible. In this way, both users and servers can gain benefits, which shows that our work is promising in the actual economic environment. Besides, if users adopt our proposal to perform privacy-preserving LBS queries, they need to obtain useful results, which means that the accuracy of the query should be guaranteed. It is worth noting that in our proposal, users are allowed to submit true locations for LBS queries if they are in an insensitive location or have no privacy requirements. In this case, the interests of users who adopt privacy-preserving queries should not be reduced. In general, the design goals of our work can be summarized as follows:

1) *Location privacy*: Users' location privacy should be protected while querying LBS;
2) *Distribution preserving*: Users' privacy-preserving LBS queries should provide the LBS server with valid location distribution information as much as possible;
3) *Incentive compatibility*: Both users and the LBS server should benefit from this model, and the interests of users who adopt privacy-preserving queries should not be harmed by other users performing true location queries;
4) *LBS accuracy*: For a single user, the accuracy of his/her LBS query should be guaranteed.

To achieve these goals, we will first give a new privacy definition considering the availability of the distribution in Section 4, and then propose a location privacy-preserving scheme to keep the user's whereabouts private while querying LBSs in Section 5.

4 DistPreserv: A New Privacy Definition

Note that all paradigms of location privacy are essentially the trade-off between privacy and utility [2]. Compared with the existing location perturbation models in which the trade-off is made between each user's privacy and his/her individual utility, we further treat the user distribution as the public utility by adding a new dimension to this trade-off, thereby effectively protecting the users' location privacy while maintaining the availability of the user distribution as much as possible.

Therefore, inspired by generalized differential privacy and Geo-Ind, we propose a new notion of location privacy, named DistPreserv. It is worth noting that in DistPreserv, euclidean distance is not the only metric for measuring the difference between alternative perturbed locations on the plane. The metrics further include a special attribute we defined between these locations, which is the difference in "request rate". Specifically, the users' request rate f_{x_i} at location x_i is defined as $f_{x_i} = \frac{n_{x_i}}{n_{total}}$, where n_{x_i} is the number of queried users at x_i, and n_{total} is the total number of users. Intuitively, the request rate f_{x_i} reflects the normalized proportion of users who have submitted queries at x_i to the total number of users. Besides, we define a mechanism $\mathcal{K}$ as a probabilistic function that assigns a probability distribution to each location $x_i \in G$, from which the mechanism can determine the sampling probability of each location when the user is at x_0. Then we let $\mathcal{K}(x)(z)$ denote the probability of perturbing x to z, and $d(\cdot,\cdot)$ denote the euclidean metric. Considering the dynamic characteristics of user distributions over time, the time is treated as continuous time slots, and the request rate will be computed independently in each time slot. Formally, the definition of DistPreserv can be given as follows.

Definition 4 (DistPreserv). *In the privacy-preserving LBSs, the location perturbation mechanism $\mathcal{K} : \mathcal{X} \rightarrow \mathcal{D}(\mathcal{Z})$ achieves $\varepsilon-$DistPreserv iff it satisfies that $\mathcal{K}(x)(z) \leq e^{\varepsilon \cdot d(x,x') \cdot |f_x - f_{x'}|}\mathcal{K}(x')(z)$, where $x, x' \in \mathcal{X}$, $d(\cdot,\cdot)$ represents euclidean distance and $f_x, f_{x'} \in [0,1]$ are the request rates at location x, x', respectively.*

This definition requires x and x' to be more indistinguishable on producing the pseudo-location z as the similarity increases between x and x', where the criterion of "similarity" considers both euclidean distance and the request rate in that either $d(x,x')$ or $|f_x - f_{x'}|$ can make sense for the measurement. Note that our DistPreserv is actually a tripartite trade-off among user privacy, individual utility and public utility, which can make the user distribution, a typical public utility, retained as much as possible.

Since DistPreserv adds public utility as a new dimension, it is actually an enhancement of Geo-Ind. Specifically, we review the definition of DistPreserv, and there is $\mathcal{K}(x)(z) \leq e^{\varepsilon \cdot d(x,x') \cdot |f_x - f_{x'}|}\mathcal{K}(x')(z)$. Then we rewrite it as $\mathcal{K}(x)(z) \leq e^{(\varepsilon \cdot |f_x - f_{x'}|) \cdot d(x,x')}\mathcal{K}(x')(z)$ and denote $\varepsilon' = \varepsilon \cdot |f_x - f_{x'}|$. From that we get $\mathcal{K}(x)(z) \leq e^{\varepsilon' \cdot d(x,x')}\mathcal{K}(x')(z)$, which is a formulation in the form of Geo-Ind. Besides, due to $f_x, f_{x'} \in [0,1]$, it is clear that $\varepsilon' \leq \varepsilon$, which means that a mechanism satisfying $\varepsilon-$DistPreserv can also meet the privacy of $\varepsilon'-$Geo-Ind. As the request rate $f_{x'}$ of each location x' in G is different, the privacy level obtained from a DistPreserv mechanism is equivalent to applying Geo-Ind protection of different parameters ε' to different potential perturbed locations x' adaptively. It is worth noting that for each location x', the level of privacy provided by $\varepsilon-$DistPreserv cannot be lower than that of privacy provided by $\varepsilon-$Geo-Ind due to $\varepsilon' \leq \varepsilon$. This also means that under the same privacy budget, DistPreserv is a stronger privacy definition than Geo-Ind. In fact, since the sum of request rates of all locations in G is 1, if and only if the request rate at the user's true location is 1, and the request rates at all other locations are 0, $\varepsilon-$DistPreserv is completely degraded to $\varepsilon-$Geo-Ind.

The inherent reason that DistPreserv is a stronger privacy definition than Geo-Ind is that, although the introduction of request rates in the privacy definition is to retain users' location distribution as much as possible, the difference of request rates can also reflect specific privacy demands. Specifically, the $|f_x - f_{x'}|$ term in the definition implies that the privacy level should be increased as the request rate is closer to that at the user's true location, i.e., the closer request rate with the true location requires the greater degree of indistinguishability. This characteristic reflects that the proximity of request rates implies a certain degree of homogeneity of locations [43]. That is, different locations with similar request rates are more likely to be semantically identical locations. Thus, requiring the perturbed location to be indistinguishable on the request rate can not only achieve the preservation of location distributions, but also protect the user's semantic location privacy in the sense of request rates. Under these circumstances, if the request rates of $x \in G$ are equal (i.e., the request rates is uniformly distributed in G), x and x' should be completely indistinguishable according to the definition, which implies that the user should adopt the uniform distribution to generate his/her perturbed location (as detailed in Algorithm 1).

It is worth noting that the mathematical form of definition 4 also satisfies $d_{\mathcal{X}}-$privacy of the generalized differential privacy [30] iff $d_{\mathcal{X}}(x,x') = \varepsilon \cdot d'(x,x')$ and $d'(x,x') = d(x,x') \cdot |f_x - f_{x'}|$. Thus, DistPreserv can also be regarded as another concretization of the abstract $d_{\mathcal{X}}-$privacy by considering the new aspect of location distribution of users on the basis of Geo-Ind. Besides, the reason $d'(x,x')$ is defined with multiplication (i.e., $d(x,x') \cdot |f_x - f_{x'}|$) instead of addition (i.e., $d(x,x') + |f_x - f_{x'}|$) is that $d(x,x')$ and $|f_x - f_{x'}|$ are observations from different dimensions to measure the location differences. Since $d(x,x') \in \mathbb{R}^+$ and $|f_x - f_{x'}| \in [0,1]$, making $d'(x,x') = d(x,x') \cdot |f_x - f_{x'}|$ can allow $d(x,x')$ and $|f_x - f_{x'}|$ to make sense together for the measurement of location differences.

5 Privacy-Preserving LBS Scheme

In our proposal, to establish a research foundation of the new privacy paradigm, we only discuss the situation where each user performs a single LBS query in a time slot. To this end, we use the euclidean metric to measure the geographical distance between locations and discretize the area G into a grid to facilitate computer processing. In addition, we regard a single cell in the grid as the basic observation unit of locations, which means that one cell in the grid corresponds to one location in G. Thus, we use the terms "location(s)" and "location cell(s)" interchangeably in the following discussions. In this

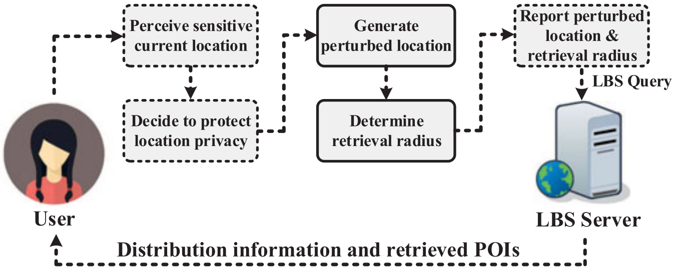

Perceive sensitive current location
Decide to protect location privacy
Generate perturbed location
Determine retrieval radius
Report perturbed location & retrieval radius
LBS Query
User
LBS Server
Distribution information and retrieved POIs

Fig. 3. Overview of the privacy-preserving LBS scheme.

setting, the distance between locations is measured by their cell centers. Specifically, in this section we first give an overview of our proposed scheme, and then give a detailed explanation of the scheme process.

5.1 Overview

To get nearby POIs while protecting location privacy, a user should perturb his/her current location to produce a pseudo-location. In this process, the user first submits a macro area to get the information of user location distributions in this area. Note that if he/she merely submits this pseudo-location to the LBS server for querying, the server will not know how large area to retrieve the POI for him/her. Thus, to perform the LBS query, the user also needs to determine the retrieval area to acquire POI information, where the generated pseudo-location is the center of the retrieval area. Once the pseudo-location and the retrieval area are both clarified, the user can perform the LBS query to acquire information about nearby hotels, restaurants and so on.

On the other side, the LBS server listens and receives users' queries, and then retrieves and returns POI information. Besides, it can count the valid user location distribution of the current time slot from the locations reported by users, regardless of whether the reported location is true or perturbed. Generally, the overview of our scheme can be illustrated in Fig. 3.

It is worth noting that in our work, users do not care about the interests of other users when perturbing their locations, nor do they consider whether the LBS server can obtain valid user distribution information. However, by adopting our scheme, users can get valid user distribution and nearby POIs from the LBS server, even though in this process they only pursue their own needs of privacy-preserving LBSs. In the following subsection, we give detailed designs of the privacy-preserving LBS scheme.

5.2 Designs in Detail

Corresponding to the previous discussion, we first introduce the method of generating a pseudo-location for a user, then explain how the user can determine the retrieval area based on the pseudo-location and the expected accuracy of returned results. Finally, we give a method for the LBS server to obtain the approximate distribution of users while providing LBS services.

1) Location Perturbation Mechanism for User. To produce a pseudo-location for privacy, the user should first get his/her current location x_0 through GPS on the mobile device. Then, he/she submits the macro area (e.g., city and district) to the LBS server to request the latest user distribution grid D_G within this area instead of directly reporting the true location. In this way, the user can obtain the distribution information to acquire distribution-related services. At the same time, this approach is necessary for the LBS server to infer that the user is in Beijing instead of Shanghai to provide such as weather services, *etc.*

Following the above steps, the user maps his/her true location to the obtained distribution D_G and expects to produce a perturbed location according to euclidean distance and the request rate difference between the user's true location x_0 and other locations $x_i \in G$. To make this process satisfy DistPreserv, we will adapt the differential privacy exponential mechanism [44]. Our goal is to protect the user's location privacy by making his/her true location indistinguishable from its similar locations measured by euclidean distance and the request rate, which means that the user is required to perturb his/her true location to a more similar location in terms of the space and the request rate with a higher probability.

To this end, a suitable utility function $u : G^2 \rightarrow \mathbb{R}$ needs to be designed to evaluate the utility of each discrete location cell $x_i \in G$. Specifically, we take $u(x_0, x_i) = -d(x_0, x_i) \cdot |f_{x_0} - f_{x_i}|$, where x_0 and x_i are location cells in G, and f_{x_0} and f_{x_i} represent the request rates at x_0 and x_i, respectively. To determine the selection probability of each location in G, the sensitivity of the utility function $u : G^2 \rightarrow \mathbb{R}$ can be introduced in our scenario. Intuitively, the sensitivity of the utility function reflects the maximum change in the utility value when the difference between alternative inputs is limited within 1 under at most one random unit metric. Since the designed utility function contains two different metrics (i.e., $d(x_0, x_i)$ and $|f_{x_0} - f_{x_i}|$), its sensitivity can be described as the maximum change in the utility value $u(x_0, x_i)$ when x_0 changes at most 1 under one random metric while being fixed under another metric. Specifically, we let $\Gamma(x_0, x_0^{'})$ denote the constraint that x_0 changes at most 1 under one random metric and is fixed under another metric, where $x_0^{'} \in G$ is the comparison location of x_0. Then the constraint $\Gamma(x_0, x_0^{'})$ can be specified as $\Gamma(x_0, x_0^{'}) = (d(x_0, x_0^{'}) \leq 1 \wedge |f_{x_0} - f_{x_0^{'}}| = 0) \vee (d(x_0, x_0^{'}) = 0 \wedge |f_{x_0} - f_{x_0^{'}}| \leq 1)$. According to it, the sensitivity of the utility function can be formally defined as

$$\Delta u = \max_{x_i \in G} \max_{\Gamma(x_0, x_0^{'})} |u(x_0, x_i) - u(x_0^{'}, x_i)|,$$

where $x_0, x_0^{'} \in G$ represent the user's true locations before and after the change, $x_i \in G$ represents any observed location. The constraints of euclidean distance and request rates in the sensitivity can be illustrated in Fig. 4, respectively. Combining the formalization of the utility function, it can be readily obtained that $\Delta u = 1$.

Based on the designed utility function, the differential privacy exponential mechanism can be invoked to select a pseudo location in G as the reported location for the query, by which the whole process satisfies the definition of DistPreserv (which is proved in section "6.1 Privacy Analysis"). The detailed algorithm for generating the perturbed location is given in Algorithm 1.

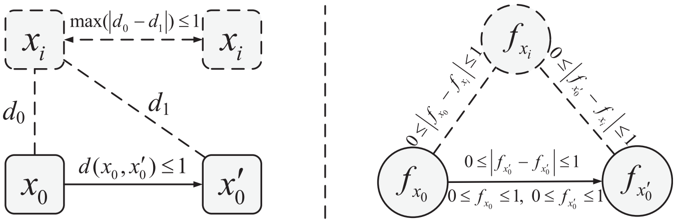

x_i
$\max(|d_0 - d_1|) \leq 1$
x_i
d_0
d_1
x_0
$d(x_0, x_0') \leq 1$
x_0'
f_{x_i}
$0 \leq |f_{x_0} - f_{x_i}| \leq 1$
$0 \leq |f_{x_0'} - f_{x_i}| \leq 1$
f_{x_0}
$0 \leq |f_{x_0'} - f_{x_0'}| \leq 1$
$0 \leq f_{x_0} \leq 1,\ 0 \leq f_{x_0'} \leq 1$
$f_{x_0'}$

Fig. 4. Illustration about the constraints of distance and request rate.

Algorithm 1. Location Perturbation Mechanism $\mathcal{K}$

```
Input: $x_0$, $\varepsilon$, $G$
Output: $z$
1: $sum \leftarrow 0$
2: for $x_i$ in $G$ do
3:     // Compute utility values for $x_i \in G$
4:     $u(x_0, x_i) \leftarrow -d(x_0, x_i) \cdot \left| f_{x_0} - f_{x_i} \right|$
5:     $expweight_i \quad e^{\frac{\varepsilon \cdot u(x_0, x_i)}{2}}$
6:     $sum \quad sum + expweight_i$
7: end for
8: // Generate the distribution for drawing $x_i$
9: for $x_i$ in $G$ do
10:    $\Pr(x_i) \quad \frac{expweight_i}{sum}$
11: end for
12: Randomly sample $x_k$ according to the computed probabilities.
13: $z \quad x_k$
14: return $z$
```

The time complexity of Algorithm 1 is $O(|G|)$, and the space complexity is $O(|G|)$, where $|G|$ is the number of locations in G. By performing this algorithm, the user can get a perturbed location z to report for privacy-preserving LBSs.

2) Determination of Retrieval Area. A user expects to get information about the surrounding POIs after generating the perturbed location. This is because if the perturbation of the true location prevents users from getting their desired POI information from the LBS server, the location perturbation will be meaningless and unattractive. Similar to most real-world LBSs (e.g., Google Map, AMAP) using the circular area to retrieve, we also make our retrieval area as a circle to keep it compatible with existing commercial applications. In this way, the determination of the retrieval area is actually the determination of the center point and retrieval radius. Obviously, the reported pseudo-location generated by Algorithm 1 is the center point for retrieval. Thus, we mainly discuss the determination method of the retrieval radius below.

Ideally, the Area of Retrieval (AOR) should always fully contain the area that the user is really interested in, because in this case the user can acquire all the POIs he/she expects. However, due to the random nature of the location perturbation as well as the uncertainty of the size of the user's Area of Interest (AOI), making the AOR always cover the AOI will reveal that the user is always inside the AOR, thereby compromising the user's location privacy. Therefore, the radius of AOR should be determined independently with the perturbed location z, which also means that we cannot make AOI always be fully contained in AOR. In this way, the process of retrieval radius determination can provide "plausible deniability" where privacy comes from it [45].

Based on the above discussion, we introduce the notion of LBS accuracy [10], which is used to indicate the probability that the user obtains full POI information he/she expects, that is, the probability that AOR fully contains AOI. Specifically, we make r_{AOR} and r_{AOI} to denote the radius of AOR and AOI. $\mathcal{C}(x, r)$ represents a circle with a center x and a radius r, and c denotes the accuracy requirement. Then we say that $(\mathcal{K}, r_{AOR})$ is (c, r_{AOI})-accurate, iff the probability of $\mathcal{C}(x, r_{AOI})$ (i.e., AOI) being fully contained in $\mathcal{C}(\mathcal{K}(x), r_{AOR})$ (i.e., AOR) is no less than c for all $x \in G$.

Based on that, our goal is to determine an appropriate r_{AOR} with any given (c, r_{AOI}) such that $(\mathcal{K}, r_{AOR})$ meets (c, r_{AOI})-accurate. Although a simple way to achieve this goal is to set r_{AOR} to a very large constant, this method will cause users to receive a large amount of returned POIs, resulting in excessive bandwidth consumption. Thus, to reduce bandwidth cost as much as possible, the r_{AOR} we want should be the minimum value meeting the accuracy requirement.

To achieve this goal, we should note that for any x_0, there is $d(x_0, z) \leq \alpha$ with the probability of no less than $\sigma(\alpha) = \sum_{x_i \in \mathcal{C}(x_0, \alpha)} \Pr(x_i)$, where $\alpha \in \mathbb{R}^*$, $z = \mathcal{K}(x_0)$, and $\Pr(x_i)$ is the probability that $x_i \in G$ is selected as the perturbed location, which is computed by the (Line 1–Line 11) of Algorithm 1. Besides, $x_i \in \mathcal{C}(x_0, \alpha)$ indicates the situation that $d(x_0, x_i) \leq \alpha$. From this view, we can get that when $c \leq \sigma(\alpha)$, the mechanism $\mathcal{K}$ satisfies the notion of (α, c)-usefulness [46], which means that for any location x_0, the reported location $z = \mathcal{K}(x_0)$ satisfies $d(x_0, z) \leq \alpha$ with the probability at least c. Combined with the notion of (c, r_{AOI})-accuracy, we can get that if a mechanism $\mathcal{K}$ is (α, c)-useful, then $(\mathcal{K}, r_{AOR})$ satisfies (c, r_{AOI})-accuracy if and only if $r_{AOR} \geq r_{AOI} + \alpha$. Due to $c \leq \sigma(\alpha)$, we can have $\alpha \geq \sigma^{-1}(c)$, which requires $r_{AOR} \geq r_{AOI} + \sigma^{-1}(c)$. Therefore, in order to satisfy (c, r_{AOI})-accuracy, the minimum r_{AOR} that meets the accuracy requirement is $r_{AOR} = r_{AOI} + \sigma^{-1}(c)$.

Note that $\sigma^{-1}(c)$ is the smallest α that makes $\sum_{x_i \in \mathcal{C}(x_0, \alpha)} \Pr(x_i)$ no less than c, we can rewrite it as $\sigma^{-1}(c) = \arg\min_{\alpha} |\sum_{x_i \in \mathcal{C}(x_0, \alpha)} \Pr(x_i) - c|$ while $\sum_{x_i \in \mathcal{C}(x_0, \alpha)} \Pr(x_i) \geq c$. Since the map plane is divided into grids and the request rate at $x_i \in G$ is unpredictable, there is no simple analytical function relationship between the selection probability of x_i and the user's true location x_0. Considering that these probabilities are known to us in Algorithm 1, we adopt the approach of dynamic programming to calculate $\sigma^{-1}(c)$ to avoid repeated considerations of the probability at each location.

Our basic approach is to examine the probability sum for the candidate locations in $\mathcal{C}(x_0, \alpha)$ when α gradually increases with a unit distance step, that is, the sum of all $\Pr(x_i)$ satisfying $d(x_0, x_i) \leq \alpha$ is denoted as $\mathcal{P}$. The purpose of utilizing dynamic programming is that the selection probability of each location cumulated in $\mathcal{P}$ will not be traversed repeatedly when α is explored from a small value to a large value. To this end, we dynamically update $\mathcal{P}$ when α increases by examining location grid layers in turn according to the distance between x_0 and each layer. Within each quadrilateral layer of the grid, the location x_i is also traversed from near to far according to $d(x_0, x_i)$. In this process, we record the starting sequence of the grid layer where there are location cells that still have not been traversed, and the current traversal position of each layer. As probabilities within all location cells of the starting traversal layer are counted into $\mathcal{P}$, the ordinal number of the starting traversal layer is increased by one (i.e., a unit length of distance). In the process of cumulation, while $\mathcal{P}$ is no less than c for the first time, α at that time is $\sigma^{-1}(c)$.

In terms of procedural details of this algorithm, we use the array Tr to store the latest position for traversal in each quadrilateral layer, and denote $startLayer$ as the ordinal number of the first remaining layer that has not been fully traversed. Note that while we say that the algorithm traverses to the $Tr[k]$'th position of the k'th quadrilateral layer, we mean to count four probabilities of positions that are equidistant from x_0 with distance $\sqrt{k^2 + Tr[k]^2}$ at the same time. The detailed process for the retrieval radius determination is shown in Algorithm 2.

Algorithm 2. Determination of the AOR Radius

```
Input: $x_0$, $r_{AOI}$, $D_G$, $c$
Output: $r_{AOR}$
$Tr$   Declare an empty array
$\mathcal{P} \leftarrow 0$
// Store the first remaining layer that is not fully traversed
$startLayer$   1
for $r$ from 1 to $\infty$ do
    $Tr[r] \leftarrow 0$ // Traverse each layer from near to far
    for $k$ from $startLayer$ to $r$ do
        Traverse $x_i$ in $k$'th layer from position $Tr[k]$, and:
        if $x_i$ meets $r-1 \leq d(x_0, x_i) \leq r$ then
            $\mathcal{P} \leftarrow \mathcal{P} + \Pr(x_i)$
        end if
        if $\mathcal{P} \geq c$ then
            // The accuracy requirement is achieved
            Break all loops
        end if
        Update $Tr[k]$ to the beginning position not traversed
    end for
    Update $startLayer$ to the beginning layer not fully traversed
end for
$r_{AOR} \leftarrow r_{AOI} + r$
return $r_{AOR}$
```

The essence of Algorithm 2 is to traverse the growing circular area on the grid plane. In Algorithm 2, the time and space complexities are both $O(|G|)$, and the effect achieved by this algorithm is to determine the value of r_{AOR} that satisfies the requirement of (c, r_{AOI})-accuracy by using the dynamic programming.

3) Query Process and User Distribution Generation. In order to query POIs near the current locations and obtain the distribution information in their macro areas in a privacy-preserving manner, the process required by users for obtaining services is: (1) Perform a pre-query. Specifically, the user sends his/her macro position G to the LBS server, and then obtains the user distribution information D_G in G as its public utility; (2) Generate perturbed location z and retrieval radius r_{AOR} according to Algorithms 1 and 2, respectively. Then the user reports them to the LBS server to obtain POIs in AOR; (3) The user filters the obtained POIs according to his/her AOI, thereby obtaining individual utility.

Since the location distribution of users is not static, the LBS server has to dynamically maintain and update the user distribution. Our approach is to discretize continuous time into consecutive equal time slots and make the LBS server always maintain the user distribution in the latest time slot. Therefore, the LBS server performs dynamic updates of the distribution in the process of counting users' reported locations. Specifically, the steps performed by the LBS server are: (1) The server waits for and receives the users' pre-queries. If it receives a pre-query in the time slot i, it returns the global user distribution D_G aggregated in the time slot $i-1$ in macro position G; (2) The LBS server receives the reported location and retrieval radius submitted by the user, and then returns the POIs in the AOR; (3) When the time slot $i+1$ arrives, the LBS server updates the distribution in the time slot i according to reported locations collected in time slot i. The processing steps of the LBS server in the time slot i are shown in Algorithm 3.

Algorithm 3. User Distribution Generation Process on Server

```
Input: $z$, $r_{AOR}$
Output: $D_G^{i-1}$, POIs in AOR
Initialize $U^{(i)} = \varnothing$
According to reported locations $\{x_u^{(i-1)} | u \in U^{(i-1)}, x_u^{(i-1)} \in G\}$
  collected in time slot $i-1$, count and obtain
  $\{(x,k) | x \in G, k \in \mathbb{N}^+\}$
Normalize $\{(x,k) | x \in G, k \in \mathbb{N}^+\}$ to get $D_G^{i-1}$
Receive a pre-query from $User_k$
Respond $D_G^{i-1}$ within $G$ to the user
Receive $z$ and $r_{AOR}$ reported by $User_k$
Retrieve POIs in AOR, then respond the retrieved POIs
if $User_k \notin U^{(i)}$ then
    add $User_k$ to $U^{(i)}$
    add $x_k^{(i)}$ to $\{x_u^{(i)} | u \in U^{(i)}, x_u^{(i)} \in G\}$
end if
```

In Algorithm 3, the LBS server initializes and maintains a set of user identifiers at each time slot to record users having initiated queries in this time slot. When a pre-query of the user is received, the LBS server returns the distribution information to the user based on the reported locations collected in the previous time slot, and records the location information of users reported in the current time slot. It is worth noting that the cold start process (i.e., the initialization of the distribution D_G) at the very beginning of the system operation needs to be discussed. Specifically, to reflect the location distribution of users, the initial D_G should not be randomly generated or artificially set. Instead, since the existing LBS service providers (e.g., Google Map, Amap, *etc.*) have been stably operated for a long time, they can count users' location distributions through almost real-time received locations. Thus, the LBS server can set the initial D_G to the distribution obtained from its previous non-privacy-preserving services. This also means that, in real-world business operations, the service provider does not need to start from scratch for additional privacy-preserving functions. Instead, the service provider can directly apply

DistPreserv based on its obtained statistical location distribution, which also makes DistPreserv more potentially practical. Besides, the operation of line 2 in the algorithm is flexible, because the diversified implementations of LBS servers allow them to obtain valid location distributions in various ways as far as possible. For example, the operator of LBSs may set privacy protection options in the user-side APPs, so that the client application can identify whether the user has performed location privacy protection. In this case, the LBS server can count location distributions of users who report true locations following the default configuration. Locations reported by users who adopt privacy protection are excluded in the distribution statistics.

Through Algorithm 3, users who query in the time slot i can obtain user distributions in their areas and POI information near their whereabouts. Meanwhile, the server can provide users with LBS services while gaining knowledge about overall users' distributions. We prove in Section 6.4 that our proposal is incentive compatible, which also means that no user can increase his/her own interests by harming the collective interests, thereby ensuring the feasibility and stability of the entire system. The time complexity of Algorithm 3 is $O^{(1)}_{time} + O^{(2)}_{time}$, where $O^{(1)}_{time}$ represents the time complexity of Step 2 in the algorithm, and $O^{(2)}_{time}$ is the time complexity of Step 7 in the algorithm. They are all related to the specific implementation of the LBS server. The space complexity is $O^{(1)}_{space} + O^{(2)}_{space}$, where $O^{(1)}_{space}$ is the space complexity implemented by the LBS server to store $\{x^{(i-1)}_u | u \in U^{(i-1)}, x^{(i-1)}_u \in G\}$, and $O^{(2)}_{space}$ is the space complexity implemented by the LBS server to store $\{(x,k) | x \in G, k \in \mathbb{N}^+\}$. They are also related to the detailed design of the data structure on the LBS server.

6 Theoretical Analysis

In this section, we perform the theoretical analysis of our proposal. Specifically, we first prove that the proposed location perturbation mechanism meets the definition of DistPreserv. Then we discuss the utility performance of the mechanism when selecting a reported location. Finally, we prove that our proposal satisfies the property of incentive compatibility.

6.1 Privacy Analysis

Compared to previous works, our proposal should allow users to keep their whereabouts private while querying for LBSs, and in that process, the distribution of users learned by the LBS server should be similar to the real distribution of users. To this end, we give a theorem which shows that our proposal can satisfy our newly introduced privacy definition.

Theorem 1. *For any privacy parameter ε, the proposed location perturbation mechanism satisfies the definition of DistPreserv.*

Proof. Since DistPreserv is formally defined with a limitation of the mechanism's input-output relationship, it is necessary for the mechanism to satisfy $\Pr[\mathcal{K}(x) = z] \le e^{\varepsilon \cdot d(x,x') \cdot |f_x - f_{x'}|} \Pr[\mathcal{K}(x') = z]$, note that the location perturbation mechanism produces the reported pseudo-location z with probability $\Pr[\mathcal{K}(x) = z]$ when the user's true location is x. Therefore we have

$$\frac{\Pr[\mathcal{K}(x)=z]}{\Pr[\mathcal{K}(x')=z]} = \frac{\exp\left(\frac{\varepsilon \cdot u(x,z)}{2}\right) / \sum_{z' \in G} \exp\left(\frac{\varepsilon \cdot u(x,z')}{2}\right)}{\exp\left(\frac{\varepsilon \cdot u(x',z)}{2}\right) / \sum_{z' \in G} \exp\left(\frac{\varepsilon \cdot u(x',z')}{2}\right)}$$

$$= \frac{\exp\left(\frac{\varepsilon \cdot d(x,z) \cdot |f_x - f_z|}{2}\right)}{\exp\left(\frac{\varepsilon \cdot d(x',z) \cdot \left|f_{x'} - f_z\right|}{2}\right)} \cdot \frac{\sum_{z' \in G} \exp\left(\frac{\varepsilon \cdot d(x',z') \cdot \left|f_{x'} - f_{z'}\right|}{2}\right)}{\sum_{z' \in G} \exp\left(\frac{\varepsilon \cdot d(x,z') \cdot \left|f_x - f_{z'}\right|}{2}\right)}$$

$$= \exp\left(\frac{\varepsilon \cdot (d(x',z) \cdot |f_{x'} - f_z| - d(x,z) \cdot |f_x - f_z|)}{2}\right) \cdot \frac{\sum_{z' \in G} \exp\left(\frac{-\varepsilon \cdot d(x',z') \cdot |f_{x'} - f_{z'}|}{2}\right)}{\sum_{z' \in G} \exp\left(\frac{-\varepsilon \cdot d(x,z') \cdot \left|f_x - f_{z'}\right|}{2}\right)}$$

$$\le \exp\left(\frac{\varepsilon \cdot d(x,x') \cdot |f_x - f_{x'}|}{2}\right) \cdot \exp\left(\frac{\varepsilon \cdot d(x,x') \cdot \left|f_x - f_{x'}\right|}{2}\right) \cdot \frac{\sum_{z' \in G} \exp\left(\frac{\varepsilon \cdot d(x',z') \cdot \left|f_{x'} - f_{z'}\right|}{2}\right)}{\sum_{z' \in G} \exp\left(\frac{\varepsilon \cdot d(x',z') \cdot \left|f_{x'} - f_{z'}\right|}{2}\right)}$$

$$= \exp(\varepsilon \cdot d(x,x') \cdot |f_x - f_{x'}|)$$

The theorem is proved. □

6.2 Utility Analysis for Perturbed locations

Although it is probabilistic to select a reported location in G by the differential private exponential mechanism, users do not have to worry too much about elements with very low utility values being selected as reported locations for querying. An element with a very low utility value means that it is too far away from the true location of the user, which can reduce the individual utility. Meanwhile, its difference in the request rate from the true location is too large, which can reduce the public utility. Specifically, as the utility of the location $x' \in G$ is $u(x,x') = -d(x,x') \cdot |f_x - f_{x'}|$, we denote the maximum utility value of locations in G as $\mathrm{OPT}_u(G) = \max_{x' \in G} u(x,x')$, and denote the set $\mathcal{R}_{\mathrm{OPT}} = \{x' \in G : u(x,x') = \mathrm{OPT}_u(G)\}$. Then we can have $\Pr[u(x,\mathcal{K}(x)) \le \mathrm{OPT}_u(G) - \frac{2}{\varepsilon}(\ln(|G|) + t)] \le e^{-t}$ [45]. This means that for any specified value, a strict upper bound exists for the probability that the actual perturbed location's utility value is less than this given value. For example, if we already know that x_k is the element with the smallest utility value in G, that is $x_k = \arg\min_{x' \in G}(u(x,x'))$, note that $u(x,x_k) \le 0$. Then, using the above inequalities, we can directly obtain that the probability of our proposed mechanism outputting x_k as the reported location does not exceed $|G| \cdot e^{\frac{\varepsilon \cdot u(x,x_k)}{2}}$.

6.3 Distribution Divergence Analysis

In this section, we theoretically evaluate the public utility, i.e., the quality of the users' spatial distribution obtained by the LBSs. Specifically, we examine the user distribution divergence between the true distribution and the distribution after perturbations by comparing our method with Geo-Ind under the metric of JS-Divergence. To this end, we give a theorem to show the distribution divergence features.

Theorem 2. *For any privacy parameter ε, the JS-Divergence between the true distribution and the distribution after perturbations by $\varepsilon-$DistPreserv is no more than that perturbed by $\varepsilon-$Geo-Ind.*

Proof. When the user's current location is x_0, according to Algorithm 1, the probability that the user perturbs x_0 to x_k is

$$\Pr(z = x_k) = \frac{e^{\frac{\varepsilon \cdot u(x_0, x_k)}{2}}}{\sum_{x_i \in G} e^{\frac{\varepsilon \cdot u(x_0, x_i)}{2}}}, \tag{1}$$

where $u(x_0, x_k) = -d(x_0, x_k) \cdot |f_{x_0} - f_{x_k}|$. Since $\sum_{x_i \in G} e^{\frac{\varepsilon \cdot u(x_0, x_i)}{2}}$ is invariant when the user at x_0 needs to perturb his/her location, we only focus on the numerator part of Eq. (1). Thus, we have $\Pr(z = x_k) \propto e^{\frac{\varepsilon \cdot u(x_0, x_k)}{2}}$, i.e.,

$$\Pr(z = x_k) \propto e^{\frac{-\varepsilon \cdot d(x_0, x_k) \cdot \left|f_{x_0} - f_{x_k}\right|}{2}}, \tag{2}$$

which can be rewritten as $\Pr(z = x_k) \propto \left(e^{\frac{-\varepsilon \cdot d(x_0, x_k)}{2}}\right)^{\left|f_{x_0} - f_{x_k}\right|}$. Note that with $\Pr(z = x_k) \propto e^{\frac{-\varepsilon \cdot d(x_0, x_k)}{2}}$, the location perturbation process is essentially based on the planar Laplace mechanism of Geo-Ind, thus distPreserv can be regarded as making the probability improvement of location perturbations by Geo-Ind. Since there are $e^{\frac{-\varepsilon \cdot d(x_0, x_k)}{2}} \in [0, 1]$ and $|f_{x_0} - f_{x_k}| \in [0, 1]$, we always have

$$\left(e^{\frac{-\varepsilon \cdot d(x_0, x_k)}{2}}\right)^{\left|f_{x_0} - f_{x_k}\right|} \geq e^{\frac{-\varepsilon \cdot d(x_0, x_k)}{2}}. \tag{3}$$

This shows that with the decrease of $|f_{x_0} - f_{x_k}|$, (i.e., f_{x_0} is closer to f_{x_k}), DistPreserv perturbs x_0 to x_k with a greater probability than Geo-Ind. Recall that the request rate f_{x_k} reflects the proportion of the number of users submitting queries at x_k to the total number of users. According to it, since DistPreserv can perturb the true location to another location with a more similar request rate by a greater probability, for a perturbed location x_k, the request rates f_{x_k} before and after the perturbation processed by DistPreserv can be closed with a greater probability. Let $f_{x_k}^{(true)}$ denote the true request rate at x_k, $f_{x_k}^{(DistPreserv)}$ denote the request rate at x_k after perturbations by DistPreserv, $f_{x_k}^{(Geo-Ind)}$ denote the request rate at x_k after perturbations by Geo-Ind. According to the above discussion, we have $|f_{x_k}^{(true)} - f_{x_k}^{(DistPreserv)}| < |f_{x_k}^{(true)} - f_{x_k}^{(Geo-Ind)}|$. This inequality means that for x_k, $f_{x_k}^{(DistPreserv)}$ is closer to $f_{x_k}^{(true)}$ than $f_{x_k}^{(Geo-Ind)}$.

This inequality means that for x_k, $f_{x_k}^{(distPreserv)}$ is closer to $f_{x_k}^{(true)}$ than $f_{x_k}^{(Geo-Ind)}$.

Then we denote $D_G^{(T)}$ as the true location distribution of users before perturbations, $D_G^{(D)}$ as the location distribution of users after the distPreserv perturbations, and $D_G^{(L)}$ as the location distribution of users after the Geo-Ind perturbations. Thereby, there are $f_{x_k}^{(true)} \in D_G^{(T)}$, $f_{x_k}^{(distPreserv)} \in D_G^{(D)}$ and $f_{x_k}^{(Geo-Ind)} \in D_G^{(L)}$. According to the formula of JS-Divergence, we have

$$JS\left(D_G^{(T)} || D_G^{(D)}\right) = \frac{1}{2} D_{KL}\left(D_G^{(T)} || D_G^{(M)}\right) + \frac{1}{2} D_{KL}\left(D_G^{(D)} || D_G^{(M)}\right), \tag{4}$$

where $D_G^{(M)} = \frac{1}{2}(D_G^{(T)} + D_G^{(D)})$ and $D_{KL}(D_G^{(T)} || D_G^{(M)}) = \sum_{x_k \ G} f_x^{(true)} \log \frac{f_{x_k}^{(true)}}{{}^{average}_{x}}$, $f_{x_k}^{(average)} \in D_G^{(M)}$. Take the form of ed KL-Divergence into JS-Divergence, we can obtain the formula expansions for distPreserv and Geo-Ind, respectively.

$$JS(D_G^{(T)} || D_G^{(D)}) = \frac{1}{2} \sum_{x_k \in G} f_{x_k}^{(true)} \log \frac{2 \cdot f_{x_k}^{(true)}}{f_{x_k}^{(true)} + f_{x_k}^{(average)}}\Bigg) + \frac{1}{2} \sum_{x_k \in G} f_{x_k}^{(distPreserv)} \log \frac{2 \cdot f_{x_k}^{(distPreserv)}}{f_{x_k}^{(distPreserv)} + f_{x_k}^{(average)}}\Bigg) \tag{5}$$

$$JS(D_G^{(T)} || D_G^{(L)}) = \frac{1}{2} \sum_{x_k \in G} f_{x_k}^{(true)} \log \frac{2 \cdot f_{x_k}^{(true)}}{f_{x_k}^{(true)} + f_{x_k}^{(average)}}\Bigg) + \frac{1}{2} \sum_{x_k \in G} f_{x_k}^{(Geo-Ind)} \log \frac{2 \cdot f_{x_k}^{(Geo-Ind)}}{f_{x_k}^{(Geo-Ind)} + f_{x_k}^{(average)}}\Bigg) \tag{6}$$

The Eqs. (5) and (6) can be continuously written as follows.

$$JS(D_G^{(T)} || D_G^{(D)}) = \frac{1}{2} \sum_{\mathrm{X}_k \in G} f_{x_k}^{(true)} \log \frac{4 \cdot f_{x_k}^{(true)}}{3 \cdot f_{x_k}^{(true)} + f_{x_k}^{(distPreserv)}}\Bigg) + \frac{1}{2} \sum_{\mathrm{X}_k \in G} f_{x_k}^{(distPreserv)} \log \left(\frac{4 \cdot f_{x_k}^{(distPreserv)}}{3 \cdot f_{x_k}^{(distPreserv)} + f_{x_k}^{(true)}}\right) \tag{7}$$

$$JS(D_G^{(T)} || D_G^{(L)}) = \frac{1}{2} \sum_{\mathrm{X}_k \in G} f_{x_k}^{(true)} \log \frac{4 \cdot f_{x_k}^{(true)}}{3 \cdot f_{x_k}^{(true)} + f_{x_k}^{(Geo-Ind)}}\Bigg) + \frac{1}{2} \sum_{\mathrm{X}_k \in G} f_{x_k}^{(Geo-Ind)} \log \frac{4 \cdot f_{x_k}^{(Geo-Ind)}}{3 \cdot f_{x_k}^{(Geo-Ind)} + f_{x_k}^{(true)}}\Bigg) \tag{8}$$

Since there is $|f_{x_k}^{(true)} - f_{x_k}^{(distPreserv)}| < |f_{x_k}^{(true)} - f_{x_k}^{(Geo-Ind)}|$, we have:

$$\left|4 \cdot f_{x_k}^{(true)} - (3 \cdot f_{x_k}^{(true)} + f_{x_k}^{(distPreserv)})\right| < \left|4 \cdot f_{x_k}^{(true)} - (3 \cdot f_{x_k}^{(true)} + f_{x_k}^{(Geo-Ind)})\right| \tag{9}$$

and

$$\left|4 \cdot f_{x_k}^{(distPreserv)} - (3 \cdot f_{x_k}^{(distPreserv)} + f_{x_k}^{(true)})\right| < \left|4 \cdot f_{x_k}^{(Geo-Ind)} - (3 \cdot f_{x_k}^{(Geo-Ind)} + f_{x_k}^{(true)})\right|. \tag{10}$$

Based on it, there are

$$\left|\frac{4 \cdot f_{x_k}^{(true)}}{3 \cdot f_{x_k}^{(true)} + f_{x_k}^{(distPreserv)}} - 1\right| < \left|\frac{4 \cdot f_{x_k}^{(true)}}{3 \cdot f_{x_k}^{(true)} + f_{x_k}^{(Geo-Ind)}} - 1\right| \tag{11}$$

and

$$\left|\frac{4 \cdot f_{x_k}^{(distPreserv)}}{3 \cdot f_{x_k}^{(distPreserv)} + f_{x_k}^{(true)}} - 1\right| < \left|\frac{4 \cdot f_{x_k}^{(Geo-Ind)}}{3 \cdot f_{x_k}^{(Geo-Ind)} + f_{x_k}^{(true)}} - 1\right|, \tag{12}$$

which indicates that the logarithmic expression in Eq. (7) is closer to zero than that in Eq. (8). Based on the above formulas, we can get that $JSD(D_G^{(T)} || D_G^{(D)}) < JSD(D_G^{(T)} || D_G^{(L)})$.

Thus, the JS-Divergence perturbed by DistPreserv is less than that perturbed by Geo-Ind. □

6.4 Incentive Compatibility Analysis

An incentive-compatible mechanism requires that the individual interests of the participants are consistent with the collective interests. Thus, a mechanism that satisfies incentive compatibility can attract participants to spontaneously follow the defined rules and attract more participants to join the system. In our scenario, users are considered rational, while the LBS server is considered honest-but-curious, implying that it is not a rational player. Thus, consistent with the discussion in Section 3.2, we analyze the property of incentive compatibility of users by taking the availability of the overall user distributions as the public utility. To this end, we denote UT, UG, and UD to indicate a user's obtained individual utility by directly reporting the true location, applying Geo-Ind, or utilizing DistPreserv when performing LBS queries, respectively. We chose Geo-Ind as the baseline for comparison because it has a strict theoretical foundation and has been extensively studied, which makes it a de facto standard for location privacy. Besides, we use PT, PG, and PD as public utilities to indicate the similarity between the users' true location distribution and the location distribution obtained from their reported locations for the above three cases, respectively. It is obvious that $PT > PD > PG$. Then we give a theorem and prove it.

Theorem 3. *The DistPreserv mechanism we proposed satisfies the property of incentive compatibility.*

Proof. We consider the following two cases to prove the property.

Case A: For users who are privacy-insensitive currently, they tend not to adopt any privacy protection strategy in this case and want to submit their current true locations to obtain LBSs. For such users, we have $UT > UG > UD$. Since the users do not have motivations to pursue privacy in this case, the optimal strategy for them is to submit their true locations directly for LBS queries. Besides, since UT and PT are respectively the best individual and public utilities, the strategy of submitting true locations directly is optimal for both individual and collective interests of users. Therefore, the user has no incentive to harm the public utility to make it lower than PT, because this action will also harm his/her own individual utility UT.

Case B: For users who are privacy-sensitive currently, it is obvious that the level of privacy is the most concerning factor for users. For these users, combined with the analysis in Section 4, we have $UD > UG > UT$, which means that the optimal strategy for users at this time is to use DistPreserv for LBS queries. On the one hand, since $PD > PG$, if the user wants to reduce the public utility by using the strategy of Geo-Ind query, it will also harm the individual utility of himself/herself due to $UD > UG$. Therefore, a rational user who cares about his/her privacy will use DistPreserv instead of Geo-Ind while issuing the LBS query. On the other hand, although the strategy of querying with true location will improve the public utility due to $PT > PD$, a rational user will not adopt this strategy to leak privacy due to $UT < UD$.

Thus, no matter whether users are privacy-sensitive or not, they cannot harm the public utility in the process of pursuing their own individual utilities. This property shows that our proposed mechanism satisfies the incentive compatibility. □

7 Experimental Evaluation

In this section, we focus on the performance of our proposal through extensive experiments. We divide G into a grid of 50×50 in simulation experiments and divide the area within Fifth Ring of Beijing City into a grid of 100×100 on a real-world dataset. We demonstrate the performance of our proposed scheme by comparing with that of the planar Laplace mechanism in Geo-Ind, since it is the original and most typical way to achieve Geo-Ind in existing practical applications. Besides, we perform experiments on a PC with Intel Core i7-6700 3.4GHz CPU, 8GB RAM, and Windows 7-64bit OS. All the experiments are programmed using Python, and the relevant code can be found on GitHub.[1]

7.1 Availability Comparison of User Distribution: Simulations Experiment

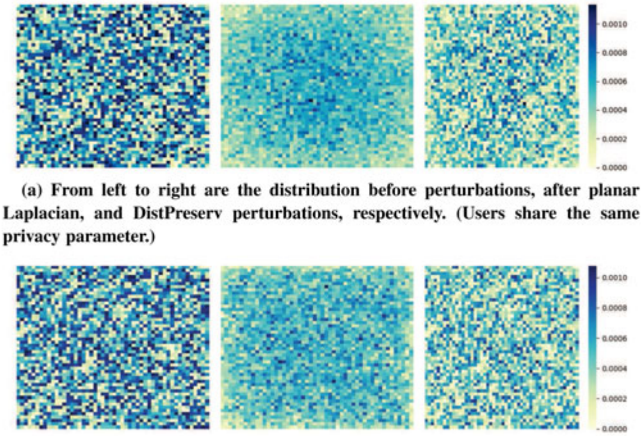
In this section, we discuss the availability of statistical distribution of users after they perturb locations. Specifically, we first divide G into a grid of 50×50, and then evaluate the distance between users' perturbed distributions and their true distributions. To this end, we still use JS-divergence to evaluate the difference of user distributions before and after their perturbations, and employ the base e logarithm to calculate KL-divergence in the computation.

Intuitively, we first show the comparison between users' true distributions, perturbed distributions based on planar Laplacian and our proposed mechanism, respectively. In this experiment, we control the number of users on each grid to follow a uniform distribution on [0, 50]. The experimental results are demonstrated in Fig. 5.

As shown in Fig. 5, after the perturbation using planar Laplacian, the distribution of reported locations is intuitively different from the users' true distribution significantly. However, adopting our proposed mechanism, the distributions are visually closer generally. In fact, if we denote the true distribution of users in G as $D_G^{(T)}$, the distribution of users after planar Laplace perturbation as $D_G^{(L)}$, and the distribution of users after DistPreserv perturbation as $D_G^{(D)}$, then we can get $JS(D_G^{(T)}||D_G^{(L)}) = 0.064$ and $JS(D_G^{(T)}||D_G^{(D)}) = 0.005$ when all users share $\varepsilon = 0.5$, which indicate that our proposed mechanism improves user distribution availability by 92.2%. When each user randomly chooses his/her privacy parameters ranging from 0.1 to 1, we have $JS(D_G^{(T)}||D_G^{(L)}) = 0.061$ and $JS(D_G^{(T)}||D_G^{(D)}) = 0.005$, which implies that the user distribution availability of our proposal is 91.8% higher than the baseline.

Then we evaluate JS-divergence between the distribution of user-reported locations and the users' true location distribution as ε and the number of users at each location cell

[1] github.com/MeetSiddhartha/spatialPatternLocationPert

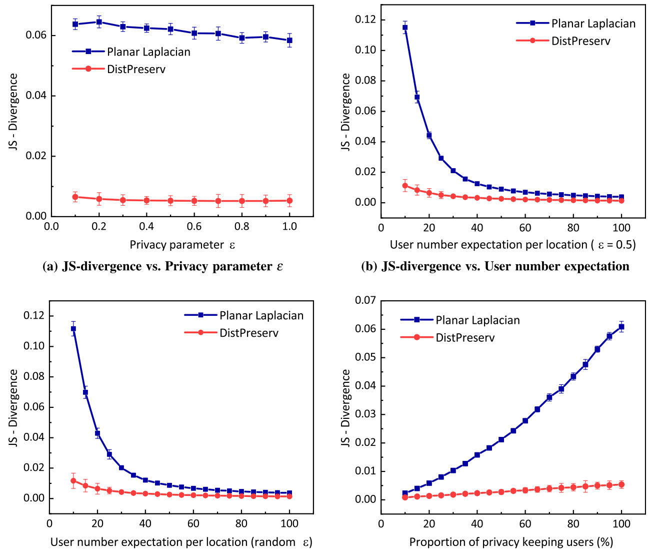


(a) From left to right are the distribution before perturbations, after planar Laplacian, and DistPreserv perturbations, respectively. (Users share the same privacy parameter.)

(b) From left to right are the distribution before perturbations, after planar Laplacian, and DistPreserv perturbations, respectively. (Users choose privacy parameters respectively.)

Fig. 5. Comparison of user distribution before and after perturbations. In this figure, heat maps are presented to give an intuitive effect of our scheme, where the shade of colors expresses the density of users. In Fig. (a), we make all users share the same privacy parameter $\varepsilon = 0.5$. In Fig. (b), we make each user choose a privacy parameter respectively and randomly on [0.1, 1]. Intuitively, Figs. (a)(b) show that the location distribution after perturbations through our proposal is closer to the true distribution than that after the planar Laplacian perturbations.

gradually changes, respectively. Specifically, in Fig. 6a, we set the users to share the same ε and make the number of participant users at each location follow a uniform distribution on [0, 50]; In Fig. 6b, we make all users share the same $\varepsilon = 0.5$, and make the number of users at each location follow the normal distribution of (μ, 10); In Fig. 6c, we allow each user to randomly choose ε within [0.1, 1], and still make the number of querying users at each location follow the normal distribution of (μ, 10); In Figs. 6a, 6b and 6c, we assume that all querying users in the current time slot adopt location perturbation to keep their whereabouts private. In Fig. 6d, we make the number of querying users in each location follow the uniform distribution on [0, 50], and make all users share the same $\varepsilon = 0.5$. Then we make the proportion of the users who adopt the strategy of location privacy protection gradually increase, which also means that we examine the situation where some users choose to submit their true locations without privacy protection. Besides, we perform each of these evaluations 100 times and calculate the means and error bars of the observed results, in which the error bars are measured by $\pm 2 \times SE$ (i.e., 95% CI), where SE and CI represent the standard error and the confidence interval of observations, respectively [47]. The results are shown in Fig. 6.



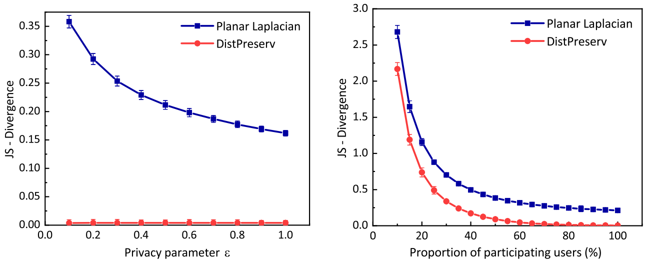
(a) JS-divergence vs. Privacy parameter ε

(b) JS-divergence vs. User number expectation

(c) JS-divergence vs. User number expectation

(d) JS-divergence vs. Proportion of sensitive users

Fig. 6. JS-divergence evaluation. We evaluate JS-divergence between the distribution before and after perturbations. From the results we learn that our proposal improves the availability of user distribution after perturbations effectively compared to the baseline.



(a) JS-divergence vs. Privacy parameter ε

(b) JS-divergence vs. Participating users

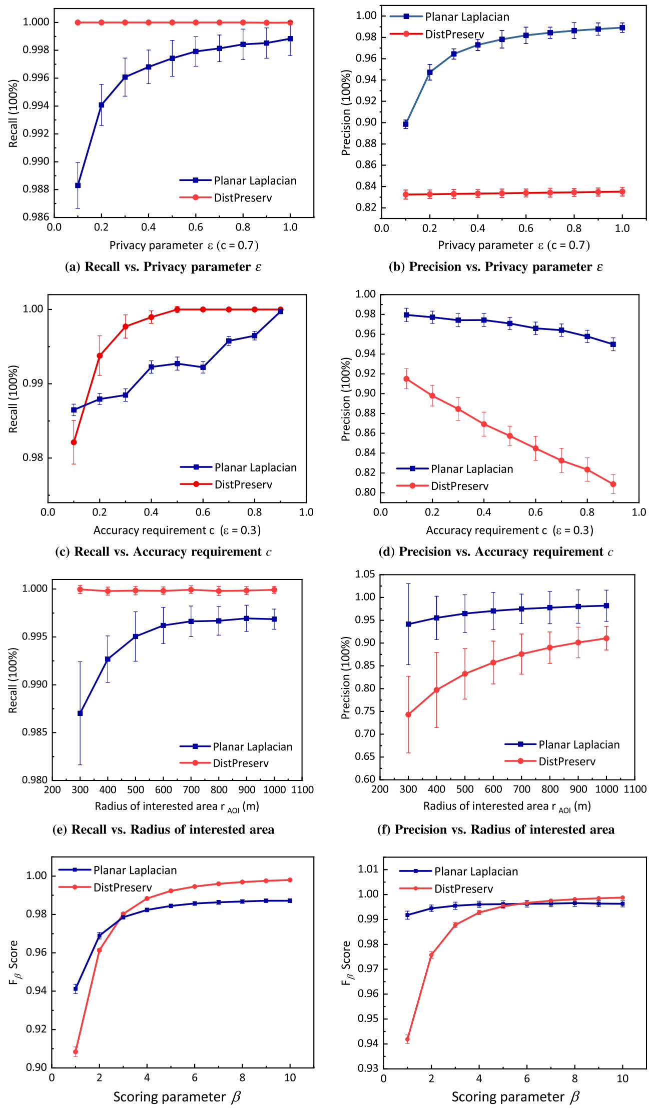
Fig. 7. JS-divergence evaluation on a real-world dataset. Fig. (a) (b) both illustrate that DistPreserv provides the better availability of the distribution after perturbations than planar laplacian. Besides, we can learn from Fig. (b) that if we want to obtain the better distribution availability, we need more users to participate, whether for DistPreserv or Geo-Ind.

Figs. 6a, 6b and 6c show that the user-selected privacy parameter ε has little effect on the JS-divergence between user location distributions before and after the perturbation. However, as ε increases, there is a slight downward trend in JS-divergence, which is also in line with the intuition that the smaller ε indicates the more randomness of the location perturbation. Besides, Figs. 6b and 6c show that when the number of users at each location gradually increases, the JS-divergence obtained by the planar Laplace mechanism is always larger than that of our proposal, especially when the number of users is small. Fig. 6d shows that as the proportion of users who adopt the strategy of location privacy protection increases, the planar Laplacian gradually loses the availability of the user distribution after perturbations, yet our mechanism effectively maintains the user distribution after perturbations as much as possible, making it as similar as before.

7.2 Availability Comparison of User Distribution: Real-World Experiment

After examining the availability of user distribution through simulations, we evaluate this issue on a real-world dataset named *Geolife*, which is collected in Beijing City by Microsoft Research. In the experiments, we divide the area within Fifth Ring of Beijing into a grid of 100×100, and randomly sample 30% of the locations in the dataset as the true locations of users. Since in this experiment the number of users at each location is uncontrolled, we evaluate the JS-divergence by varying the privacy parameter ε and the proportion of participating users. In each evaluation, the experiment is performed 100 times to calculate the means and error bars, in which the error bars are also computed by $\pm 2 \times SE$ (i.e., 95% CI). The results are shown in Figs. 7a and 7b.

In Fig. 7a, we make all users share the same privacy parameter ε, and in Fig. 7b, each user randomly chooses a privacy parameter within the range [0.1, 1]. The results demonstrate that DistPreserv prominently improves the



(a) Recall vs. Privacy parameter ε

(b) Precision vs. Privacy parameter ε

(c) Recall vs. Accuracy requirement c

(d) Precision vs. Accuracy requirement c

(e) Recall vs. Radius of interested area

(f) Precision vs. Radius of interested area

(g) F_β score vs. Scoring parameter β

(h) Weighted product vs. Scoring parameter β

Fig. 8. Evaluation on the precision and recall. From figures (a)$\sim$(f) we get that our proposal outperforms planar laplacian in terms of recall and underperforms it in terms of Precision, which indicates that DistPreserv can get more complete queried results, while bearing the cost of greater bandwidth overheads. Besides, figures (g)$\sim$(h) shows that with the increased emphasis on recall, the comprehensive score of both methods increase and our proposal can surpass the baseline at a certain β.

availability of the user distribution after perturbations, which further confirm the advantage of our proposal.

7.3 Precision and Recall Comparison of LBS Query

We examine the accuracy of queried POIs by introducing two metrics that are widely used to evaluate retrieval information, namely precision and recall. In the scenario of LBSs, the precision refers to the proportion of actually interested POIs received by the user to the total POIs within AOR, and the recall refers to the proportion of received interested POIs to the total POIs within AOI. Formally describing, if we make $True$ represent POIs in AOI, make $Positive$ represent POIs in AOR, and denote TP as POIs in AOI $\cap$ AOR, then the precision of the query equals to $\frac{TP}{Positive}$ and the recall of the query equals to $\frac{TP}{True}$ according to these notations.

In this experiment, we still choose the Fifth Ring Road of Beijing City as the macro area G and project the grid with 100×100 cells in this area. Besides, we control the number of users on each grid to follow a uniform distribution on [0, 50], and then select 100 locations uniformly in this area as the user's true locations. According to these settings, we generate the reported location z at each location and determine the radius of retrieval area r_{AOR} based on our proposed method and the baseline method, respectively. Afterwards, AMAP "neighboring search" API is invoked to query POIs around the true and reported locations, respectively. For example, if we want to get POI information about all hotels up to 500 meters from the location (120.101193, 30.238169), we can query by the HTTPS request below:

$restapi.amap.com/v3/place/around?key = ourToken\&$
$location = 120.101193, 30.238169\&radius = 500\&$
$keywords = hotels.$

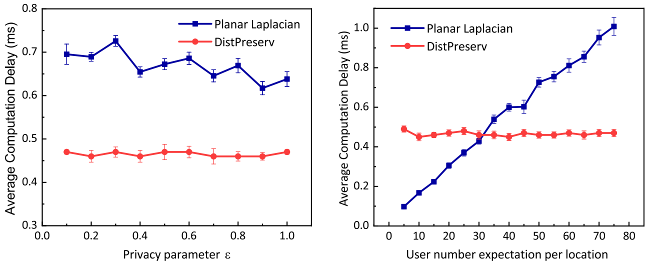
Besides the precision and the recall, we also combine these two factors to compute their F_β score [48]. Specifically, the formula of the F_β score is defined as

$$F_\beta = (1+\beta^2) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall},$$

where β is the scoring parameter to regulate the emphasis for the recall and a bigger β means the higher importance on the recall in evaluations [49]. Other control factors in the evaluations are instantiated as case 1 ($\varepsilon = 0.1$, $R_{AOI} = 500$, $c = 0.7$) and case 2 ($\varepsilon = 0.5$, $R_{AOI} = 800$, $c = 0.5$), respectively. Each evaluation is tested 100 times to get the means and error bars, where the error bars are also measured by $\pm 2 \times SE$ (i.e., 95% CI). Overall, the evaluating results for these indicators are presented in Fig. 8.

From Fig. 8 we learn that precision and recall constitute a trade-off to some extent. In general, our proposal outperforms the baseline in terms of recall and underperforms it in terms of precision. The reason for this phenomenon is that, since our proposal can provide a higher level of privacy as detailed in Section 4, it requires a larger retrieval radius to satisfy the accuracy level specified by users. As a larger retrieval radius means that users can receive more POIs, users can get more complete results, which is reflected by a higher recall. Meanwhile, since more received POIs inevitably include some results that are not in the user's AOI, the evaluated precision would be decreased. Besides, it is worth noting that in LBSs, the recall is the major demonstration of the quality of services since it indicates the completeness of queried results, and the basic goal of LBSs is to provide complete results. In contrast, since the precision reflects the proportion of desired POIs in all received POIs, higher precision reflects the refinement and lower precision reflects the redundancy of the queried results, which means that the precision only embodies the bandwidth overhead in LBSs. Since the bandwidth overhead can be efficiently resolved in current 5G/WiFi and future 6G networks, the recall (which reflects the completeness of returned queries) is a more important indicator compared with the precision.

Besides, through the comprehensive metrics of F_β, we can get that with the increase of the scoring parameter β, the value of F_β can increase for both comparative methods. Moreover, the evaluation shows that as β grows, our proposal can surpass the baseline at a certain scoring parameter



(a) Computation delay vs. Privacy parameter ε **(b) Computation delay vs. User number expectation**

Fig. 9. Evaluation on the computation delay. With the increasing number expectation of users, the computation delay of our mechanism is basically invariant while the baseline approach leads to a growing computation cost. Besides, the maximum delay of our approach is about 0.5ms, which is a quite small cost.

β, which means that as the increased emphasis on the recall, the superiority of our proposal will be highlighted in terms of the POI querying performance. Based on the above discussions, we can get that our proposal is effective and practical since it can provide more complete queried results with affordable bandwidth overheads.

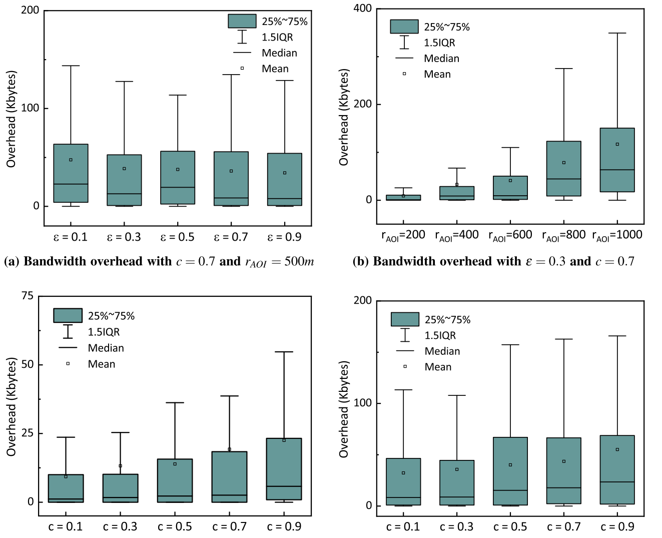
In the next subsection, we will examine the indicator of bandwidth overhead, which also reflects that the maximum bandwidth overhead in the experiment is limited.

7.4 Computation Delay and Bandwidth Overhead

In this subsection, we continue to examine the computation delay and bandwidth overhead of our proposed scheme. Specifically, we first evaluate the computation delay with different privacy parameters and user number expectations, where in Fig. 9a, we make the querying users in each location follow the uniform distribution on [0, 100]; in Fig. 9b, we make $\varepsilon = 0.5$ and the number of users in each location follows the uniform distribution with varying expectations. The baseline approach in this experiment is to employ planar Laplacian repeatedly until the number of users at the perturbed location differs from that at the true location by no more than 10. We perform each of these two tested methods to generate reported locations 100 times and calculate the means and error bars of computation delays. The error bars are also measured by $\pm 2 \times SE$ (i.e., 95% CI). The results are shown in Fig. 9.

Fig. 9 reflects that when the number of querying users follows the uniform distribution on [0, 100] at each location, our mechanism has a lower computation delay than the baseline approach. Besides, with the increase of user number expectation, the computation delay of our mechanism is almost unchanged, while that of the baseline approach gradually increases. Note that the maximum computation delay of our approach is about 0.5ms, which is acceptable to users.

Since our proposal can provide a high level of privacy, preserve the user distribution after perturbations and obtain decent accuracy of queries, it requires a larger retrieval radius in the query, which is embodied in bandwidth overheads. Therefore, it is necessary to examine the bandwidth overhead incurred by our proposal. Specifically, we use the same setting as that in Section 7.3 for G and D_G. Then we make the user set the privacy parameter ε, accuracy requirement c and the radius of interested area r_{AOI}, and call corresponding algorithms to generate the pseudo-location z and the radius of the retrieval area. After that, we submit z and r_{AOR} to AMAP searching API in the same way as that in Section 7.3 to get POI information, by which we can count the bandwidth overhead. We randomly choose 100 users as the querying user and compute the mean value of counted bandwidth overheads. The results are shown in Fig. 10.



(a) Bandwidth overhead with $c = 0.7$ and $r_{AOI} = 500m$ **(b) Bandwidth overhead with $\varepsilon = 0.3$ and $c = 0.7$**

(c) Bandwidth overhead with $\varepsilon = 0.3$ and $r_{AOI} = 300m$ **(d) Bandwidth overhead with $\varepsilon = 0.3$ and $r_{AOI} = 500m$**

Fig. 10. Evaluation on the bandwidth overhead. The bandwidth overhead does not appear to be prominently affected by ε, while it has a slight upward trend with a larger c, and rises with a larger r_{AOI}.

From Fig. 10 we get that the privacy parameter ε has no prominent relationship with the bandwidth overhead. However, with the increase of r_{AOI}, the bandwidth overhead also rises gradually. Besides, the growing accuracy requirement c gives the bandwidth overhead a slight upward trend to some degree. We also learn from the figures that the maximum bandwidth overhead in the experiment is about 350 KB. Since this overhead is approximately equivalent to 0.6 seconds of 720P YouTube video, we believe this overhead is acceptable to mobile users.

8 Conclusion and Future Work

Since Geo-Ind location privacy protection methods undermine the true distribution of querying users after location perturbations, we give a new privacy definition namely DistPreserv, so that the user distribution can be largely retained on the LBS server after location perturbations, thereby providing benefits to both the LBS server and users in LBSs. We first design a detailed mechanism to produce perturbed locations to satisfy the privacy definition, then provide a retrieval radius determination method to enable users to obtain preferred query accuracy while protecting their location privacy, finally discuss the issues about interactions between the LBS server and a user during the implementation process. Theoretical analyses prove that our proposal can achieve the expected level of privacy and the property of incentive compatibility. Experimental results verify that our proposal can better retain the true distribution of querying users and achieve feasibility.

In the future, we aim at extending our discussion to the continuous queries of the user, with taking the mobility patterns of users and the privacy budget consumption of DistPreserv into consideration. Besides, it will be interesting to explore whether it is possible for users to retain the availability of the distribution while reporting perturbed locations through only one round of interaction with the server.

References

[1] T. M. Research, "Global location based marketing services market," Accessed: Jan. 15, 2022. [Online]. Available: www.transparencymarketresearch.com/pressrelease/location-based-marketing-services-market.htm

[2] V. Primault, A. Boutet, S. B. Mokhtar, and L. Brunie, "The long road to computational location privacy: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2772–2793, Third Quarter 2019.

[3] X. Li *et al.*, "Perturbation-hidden: Enhancement of vehicular privacy for location-based services in internet of vehicles," *IEEE Trans. Netw. Sci. Eng.*, vol. 8, no. 3, pp. 2073–2086, Third Quarter 2021.

[4] F. Fei, S. Li, H. Dai, C. Hu, W. Dou, and Q. Ni, "A k-anonymity based schema for location privacy preservation," *IEEE Trans. Sustain. Comput.*, vol. 4, no. 2, pp. 156–167, Second Quarter 2019.

[5] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, "L-diversity: Privacy beyond k-anonymity," in *Proc. IEEE Int. Conf. Data Eng.*, 2006, pp. 24–24.

[6] N. Li, T. Li, and S. Venkatasubramanian, "T-closeness: Privacy beyond k-anonymity and L-diversity," in *Proc. IEEE Int. Conf. Data Eng.*, 2007, pp. 106–115.

[7] I. Wagner and D. Eckhoff, "Technical privacy metrics: A systematic survey," *ACM Comput. Surv.*, vol. 51, no. 3, 2018, Art. no. 57.

[8] E. Toch *et al.*, "The privacy implications of cyber security systems: A technological survey," *ACM Comput. Surv.*, vol. 51, no. 2, 2018, Art. no. 36.

[9] D. Kahneman, *Thinking, Fast and Slow*. Basingstoke, U.K.: Macmillan, 2011.

[10] M. E. Andrés, N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Geo-indistinguishability: Differential privacy for location-based systems," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2013, pp. 901–914.

[11] S. V. Co., "Spatial vision," Accessed: Jan. 15, 2022. [Online]. Available: github.com/LapisIT/differential_privacy

[12] Chatziko, "location guard," Accessed: Jan. 15, 2022. [Online]. Available: github.com/chatziko/location-guard

[13] K. Fawaz and K. G. Shin, "Location privacy protection for smartphone users," in *Proc. ACM Conf. Comput. Commun. Secur.*, 2014, pp. 239–250.

[14] K. Fawaz, H. Feng, and K. G. Shin, "Anatomization and protection of mobile apps location privacy threats," in *Proc. USENIX Secur. Symp.*, 2015, pp. 753–768.

[15] K. Chatzikokolakis, E. ElSalamouny, and C. Palamidessi, "Practical mechanisms for location privacy," *Proc. Privacy Enhancing Technol.*, vol. 4, pp. 210–231, 2017.

[16] Wikipedia, "Location-based service," Accessed: Jan. 15, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Location-based_service

[17] R. Chen, H. Li, A. K. Qin, S. P. Kasiviswanathan, and H. Jin, "Private spatial data aggregation in the local setting," in *Proc. IEEE Int. Conf. Data Eng.*, 2016, pp. 289–300.

[18] W. Liu, M. F. Rahman, S. Thirumuruganathan, N. Zhang, and G. Das, "Aggregate estimations over location based services," in *Proc. Int. Conf. Very Large Data Bases*, 2015, vol. 8, pp. 1334–1345.

[19] Q. Hu, S. Wang, C. Hu, J. Huang, W. Li, and X. Cheng, "Messages in a concealed bottle: Achieving query content privacy with accurate location-based services," *IEEE Trans. Veh. Technol.*, vol. 67, no. 8, pp. 7698–7711, Aug. 2018.

[20] J. Zhou, Z. Cao, Z. Qin, X. Dong, and K. Ren, "LPPA: Lightweight privacy-preserving authentication from efficient multi-key secure outsourced computation for location-based services in VANETs," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 420–434, 2019.

[21] U. M. Aïvodji, K. Huguenin, M.-J. Huguet, and M.-O. Killijian, "SRide: A privacy-preserving ridesharing system," in *Proc. ACM Conf. Secur. Privacy Wireless Mobile Netw.*, 2018, pp. 40–50.

[22] H. Carter, B. Mood, P. Traynor, and K. Butler, "Secure outsourced garbled circuit evaluation for mobile devices," *J. Comput. Secur.*, vol. 24, no. 2, pp. 137–180, 2016.

[23] F. Wang *et al.*, "Efficient and privacy-preserving dynamic spatial query scheme for ride-hailing services," *IEEE Trans. Veh. Technol.*, vol. 67, no. 11, pp. 11 084–11 097, Nov. 2018.

[24] Y.-B. Zhang, Q.-Y. Zhang, Z.-Y. Li, Y. Yan, and M.-Y. Zhang, "A k-anonymous location privacy protection method of dummy based on geographical semantics," *IJ Netw. Secur.*, vol. 21, no. 6, pp. 937–946, 2019.

[25] L. Zhang, Y. Qian, M. Ding, C. Ma, J. Li, and S. Shaham, "Location privacy preservation based on continuous queries for location-based services," in *Proc. IEEE Conf. Comput. Commun.*, 2019, pp. 1–6.

[26] W. Xu and C.-Y. Chow, "A location-and diversity-aware news feed system for mobile users," *IEEE Trans. Services Comput.*, vol. 9, no. 6, pp. 846–861, Nov./Dec. 2016.

[27] H.-T. Li, L.-X. Gong, F. Guo, Q.-L. Miao, J. Wang, and T. Zhang, "Location privacy protection in mobile social networks based on L-diversity," *J. Inf. Sci. Eng.*, vol. 36, no. 4, pp. 745–763, 2020.

[28] D. Rebollo-Monedero , J. Forne, and J. Domingo-Ferrer , "From t-closeness-like privacy to postrandomization via information theory," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 11, pp. 1623–1636, Nov. 2010.

[29] D. Riboni, L. Pareschi, C. Bettini, and S. Jajodia, "Preserving anonymity of recurrent location-based queries," in *Proc. Int. Symp. Temporal Representation Reasoning*, 2009, pp. 62–69.

[30] K. Chatzikokolakis, M. E. Andrés, N. E. Bordenabe, and C. Palamidessi, "Broadening the scope of differential privacy using metrics," in *Proc. Privacy Enhancing Technol.*, 2013, pp. 82–102.

[31] C. Qiu and A. Squicciarini, "Location privacy protection in vehicle-based spatial crowdsourcing via geo-indistinguishability," in *Proc. IEEE Int. Conf. Distrib. Comput. Syst.*, 2019, pp. 1061–1071.

[32] B. Niu, Y. Chen, Z. Wang, F. Li, B. Wang, and H. Li, "Eclipse: Preserving differential location privacy against long-term observation attacks," *IEEE Trans. Mobile Comput.*, vol. 21, no. 1, pp. 125–138, Jan. 2022.

[33] J. Hua, W. Tong, F. Xu, and S. Zhong, "A geo-indistinguishable location perturbation mechanism for location-based services supporting frequent queries," *IEEE Trans. Inf. Forensics Security*, vol. 13, no. 5, pp. 1155–1168, May 2018.

[34] F. Peng, S. Tang, B. Zhao, and Y. Liu, "A privacy-preserving data aggregation of mobile crowdsensing based on local differential privacy," in *Proc. ACM Turing Celebration Conf.-China*, 2019, pp. 1–5.

[35] G. Cormode, S. Jha, T. Kulkarni, N. Li, D. Srivastava, and T. Wang, "Privacy at scale: Local differential privacy in practice," in *Proc. Int. Conf. Manage. Data*, 2018, pp. 1655–1658.

[36] S. P. Kasiviswanathan, H. K. Lee, K. Nissim, S. Raskhodnikova, and A. Smith, "What can we learn privately?," *SIAM J. Comput.*, vol. 40, no. 3, pp. 793–826, 2011.

[37] C. Dwork, "Differential privacy: A survey of results," in *Proc. Int. Conf. Theory Appl. Models Comput.*, 2008, pp. 1–19.

[38] J. G. Riley, *Incentive Compatibility and Mechanism Design*. Cambridge, U.K.: Cambridge Univ. Press, 2012, pp. 382–435.

[39] X. Zhu, E. Ayday, and R. Vitenberg, "A privacy-preserving framework for outsourcing location-based services to the cloud," *IEEE Trans. Dependable Secure Comput.*, vol. 18, no. 1, pp. 384–399, Jan./Feb. 2021.

[40] H. Jiang, J. Li, P. Zhao, F. Zeng, Z. Xiao, and A. Iyengar, "Location privacy-preserving mechanisms in location-based services: A comprehensive survey," *ACM Comput. Surv.*, vol. 54, Jan. 2022, Art. no. 4.

[41] D. Negi, S. Ray, and R. Lu, "Pystin: Enabling secure LBS in smart cities with privacy-preserving top- spatial–textual query," *IEEE Internet of Things J.*, vol. 6, no. 5, pp. 7788–7799, Oct. 2019.

[42] B. Bostanipour and G. Theodorakopoulos, "Joint obfuscation of location and its semantic information for privacy protection," *Comput. Secur.*, vol. 107, 2021, Art. no. 102310.

[43] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proc. IEEE Conf. Comput. Commun.*, 2014, pp. 754–762.

[44] F. McSherry and K. Talwar, "Mechanism design via differential privacy," in *Proc. 48th Annu. IEEE Symp. Found. Comput. Sci.*, 2007, pp. 94–103.

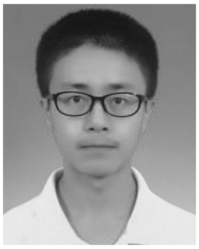
[45] C. Dwork and A. Roth, "The algorithmic foundations of differential privacy," *Found. Trends Theor. Comput. Sci.*, vol. 9, no. 3/4, pp. 211–407, 2014.

[46] A. Blum, K. Ligett, and A. Roth, "A learning theory approach to noninteractive database privacy," *J. ACM*, vol. 60, no. 2, pp. 12:1–12:25, 2013.

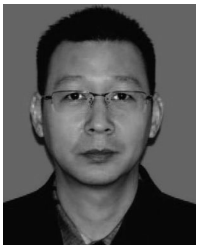
[47] S. Gupta and V. Kapoor, *Fundamentals of Mathematical Statistics*. Delhi, India: Sultan Chand & Sons, 2020.

[48] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-score, with implication for evaluation," in *Proc. Eur. Conf. Inf. Retrieval*, 2005, pp. 345–359.

[49] R. Baeza-Yates *et al.*, *Modern Information Retrieval*, vol. 463. New York, NY, USA: ACM Press, 1999.



Yanbing Ren received the BS degree in information security from Hainan University, China, in 2015. He is currently working toward the PhD degree in security of cyberspace at Xidian University, China. His research interests include blockchain, privacy preserving and intelligent mobile computing.



Xinghua Li (Member, IEEE) received the MS and PhD degrees in computer science from Xidian University, China, in 2004 and 2007, respectively. He is currently a professor in the School of Cyber Engineering, Xidian University, China. His research interests include wireless networks security, privacy protection, cloud computing, and security protocol formal methodology.

Yinbin Miao received the BE degree from the Department of Telecommunication Engineering, Jilin University, Changchun, China, in 2011, and the PhD degree from the Department of Telecommunication Engineering, Xidian University, Xi'an, China, in 2016. He is currently an associate professor with the Department of Cyber Engineering, Xidian University, Xi'an, China. His research interests include information security and applied cryptography.

Robert H. Deng (Fellow, IEEE) is currently AXA chair professor of cybersecurity and professor of information systems in the School of Information Systems, Singapore Management University, Singapore since 2004. His research interests include data security and privacy, multimedia security, network and system security. He served/is serving on the editorial boards of many international journals, including the *IEEE Transactions on Information Forensics and Security*, the *IEEE Transactions on Dependable and Secure Computing*. He has received the Distinguished Paper Award (NDSS 2012), Best Paper Award (CMS 2012), Best Journal Paper Award (IEEE Communications Society 2017).

Jian Weng (Member, IEEE) received the PhD degree from Shanghai Jiao Tong University, Shanghai, China, in 2008. He is currently a professor with the College of Information Science and Technology, Jinan University, Guangzhou, China. His research interests include public key cryptography, cloud security, blockchain, etc. He served as a PC co-chairs or PC member for more than 20 international conferences.

Siqi Ma received the PhD degree in information system from Singapore Management University, Singapore, in 2018. She is currently a lecturer in the School of Information Technology and Electrical Engineering, University of Queensland, Australia. Her research interests are mobile security, web security, IoT security.

Jianfeng Ma (Member, IEEE) received the BS degree in computer science from Shaanxi Normal University, China, in 1982, and the MS and PhD degrees in computer science from Xidian University, China, in 1992 and 1995, respectively, where he is currently a professor with the School of Cyber Engineering. He has published more than 150 journal and conference papers. His research interests include information security, cryptography, and network security.

▷ **For more information on this or any other computing topic, please visit our Digital Library at** www.computer.org/csdl.