Hindawi
Wireless Communications and Mobile Computing
Volume 2018, Article ID 9173519, 15 pages
https://doi.org/10.1155/2018/9173519

Research Article

A Context-Aware Location Differential Perturbation Scheme for Privacy-Aware Users in Mobile Environment

Xuejun Zhang,[1,2] Haiyan Huang,[1] Shan Huang,[3] Qian Chen,[1] Tao Ju,[1] and Xiaogang Du[1]

[1] *School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China*
[2] *The Key Laboratory of Opto-Technology and Intelligent Control Ministry of Education, Lanzhou 730070, China*
[3] *School of Civil Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China*

Correspondence should be addressed to Xuejun Zhang; zxjlyl_new@163.com

Received 14 March 2018; Revised 24 June 2018; Accepted 16 July 2018; Published 6 August 2018

Academic Editor: Wolfgang H. Gerstacker

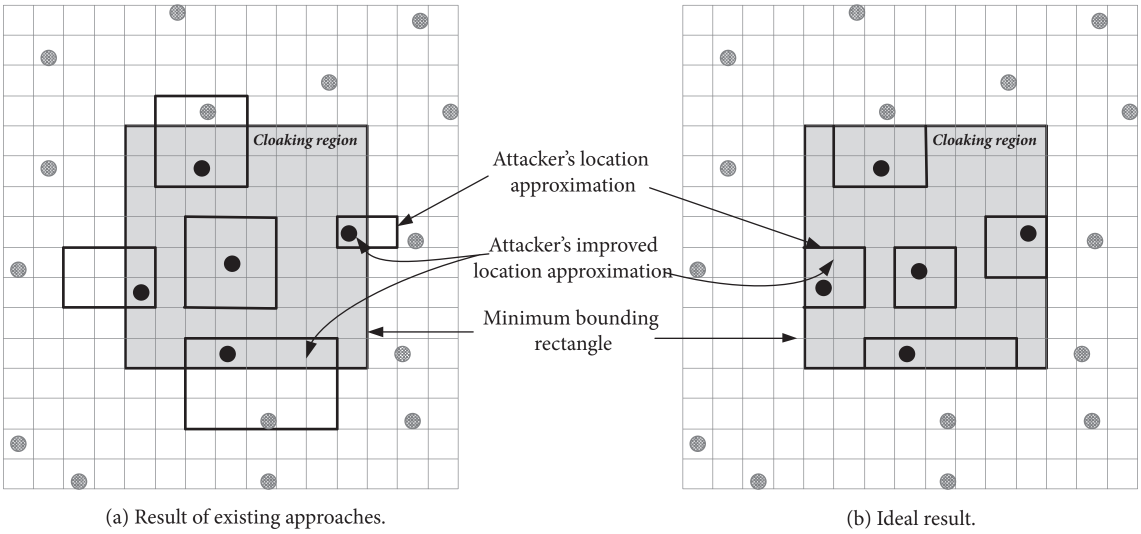
The proliferation of location-based services, representative services for the mobile networks, has posed a serious threat to users' privacy. In the literature, several privacy mechanisms have been proposed to preserve location privacy. Location obfuscation enforced using cloaking region is a widely used technique to achieve location privacy. However, it requires a trusted third-party (TTP) and cannot sufficiently resist various inference attacks based on background information and thus is vulnerable to location privacy breach. In this paper, we propose a context-aware location privacy-preserving solution with differential perturbations, which can enhance the user's location privacy without requiring a TTP. Our scheme utilizes the modified Hilbert curve to project every 2-d location of the user in the considered map to 1-d space and randomly generates the reasonable perturbation by adding Laplace noise via differential privacy. In order to solve the resource limitation of mobile devices, we use a quad-tree based scheme to transform and store the user context information as bit stream which achieves the high compression ratio and supports efficient retrieval. Security analysis shows that our proposed scheme can effectively preserve the location privacy. Experimental evaluation shows that our scheme retrieval accuracy is increased by an average of 15.4% compared with the scheme using standard Hilbert curve. Our scheme can provide strong privacy guarantees with a bounded accuracy loss while improving retrieval accuracy.

1. Introduction

As the indispensable parts of the communications and networks field, the green mobile networks are seen as a potential enabler to realize green communications and networks by minimizing energy consumption while guaranteeing the quality of service [1]. Recently, the rapid development of green wireless communication technologies and personal mobile devices equipped with GPS chips enable location-based services (LBSs) become very popular in almost all social and business domains. Some potential applications of LBS include location-aware information retrieval (e.g., Around Me), GPS navigator (e.g., TomTom), mapping application (e.g., Google Maps), and location-aware social networks (e.g., Foursquare) [2]. With the help of these applications, users can easily issue LBS queries from their smartphones to the LBS providers (LSP) and obtain services related to their current locations. For example, users can search for their friends, share information with each other, and provide check-in data by using the Foursquare.

Despite the enormous benefits of LBSs provided to individual and society, they also raise major privacy concerns when location information has to leave users' devices to untrusted LSP. Location data contained into the LBS queries can be easily linked to a variety of other information about an individual and reveal his sensitive private information such as his home and work address, sexual preferences, political views, religious inclinations, and health conditions.

To address the privacy issues for mobile users in LBSs, a variety of privacy-preserving mechanisms and metrics have been proposed to allow users to make use of the LBSs while mitigating privacy concerns over the past few years [3–15]. These LBS privacy protection mechanisms (LPPMs) provide different privacy-utility trade-off, which offer alternatives


Cloaking region
Attacker's location approximation
Attacker's improved location approximation
Minimum bounding rectangle
Cloaking region

(a) Result of existing approaches.

(b) Ideal result.

Figure 1: Location privacy as a result of using CR.

to better meet personal requirements of different mobile users. Roughly speaking, these LPPMs can be divided into two categories according to their architecture [16]: trusted anonymization server-based schemes [3, 5, 7, 8, 11, 13] and mobile devices-based schemes [4, 6, 9, 10, 14, 15]. In trusted anonymization serve-based schemes, a trusted third-party server (e.g., anonymizer [3]) is employed to perturb, obfuscate, and cloak user's query location by using the notion of k-anonymity [3]. To achieve k-anonymity, a user issues his location to LSP via a trusted third-party server (TTP), which subsequently generates a cloaking regions (CR) that covers not only this user, but also k-1 other users geographically. Therefore, it is difficult for the untrusted LSP to distinguish a user among at least k-1 others. Although such schemes can indeed strengthen the location privacy of users, they heavily rely on the TTP, which would easily be a bottleneck due to handling query requests, frequent updates of user locations, and result postprocessing. Moreover, since the TTP knows the complete knowledge of the locations and queries of all users, it would suffer from a single point of failure. If the adversary seizes control of it, the privacy of all users will be compromised. Recent research [7] attempts to solve this problem by using dynamic grid system, while it requires changing the system mode of the client-side, TTP, and sever-side. Furthermore, it incurs the high computation overhead at client-side. Mobile device-based approaches remove the requirement of a TTP by using k-anonymity [10, 15], location obfuscation and perturbation [4, 6, 9], and private information retrieval (PIR) [14]. However, PIR may incur high computation and communication costs unaffordable to mobile devices and LBS server. The k-anonymity [7, 10, 15] assumes that the adversary has no *side information* about the user [11, 12], such as approximate location, mobility profile, query frequency, and user profiles. In reality, since some adversary (e.g., the LSP) may possess such *side information*, these methods are inadequate to protect the user's location privacy [8]. In Figure 1(a), for example, when approximate location knowledge (e.g., an area) is available to an adversary, he can exploit k-anonymous CR to enhance the precision of location knowledge of multiple users. The CR can therefore provide additional location knowledge to the adversary, thereby leading to a location privacy breach. As shown in Figure 1(b), the problem can be eliminated only if the cloaking regions are guaranteed to encompass the approximated regions corresponding to each of the k users. Unfortunately, it is difficult to judge the extent of knowledge that an adversary possesses. Furthermore, sometime it is difficult to find enough users in a reasonable CR. Thus, in order to achieve the desired level of privacy, CR may be unnecessary expansion. In the worst case, the services for users would be denied. Local obfuscation and differential perturbation approaches [4, 6, 9] may be used to protect user's privacy against an adversary with such *side information*, as they consider the adversary's knowledge and capability to better make a trade-off between location privacy and LBS utility. Further, the differential perturbation [9] abstracts from the *side information* of the adversary, which promises strong theoretical privacy guarantees with a bounded accuracy loss [17]. Nevertheless, these methods are unlikely suffice for LBS because they do not take the contextual information, such as map information, points of interest (POIs) density, the scale of location, and the user's privacy requirement into account. In real scenario, the LBS privacy protection level and accuracy, achieved by location obfuscation and differential perturbation approach, depend highly on the contextual information surround a user. For instance, intuition suggests that a LBS user should deviate from his query location in a rural area than in a downtown area in order to achieve the same privacy level and LBS utility. To the best of our knowledge, how to design a TTP-free and context-aware privacy-preserving LBS system suitable for mobile devices is still challenging.

In this paper, we propose a context-aware differentially private location perturbation solution for location privacy-preserving which operates solely on the devices and does not require any TTP. Different from existing approaches, our scheme considers the contextual information around the user's location and can prevent privacy breach against an adversary with some *side information*. We first use the modified Hilbert curve (MHC) to transform and store every 2-d geographical location in the considered map to 1-d space in terms of the contextual information of a user's location and then randomly perturb the user's location, by adding a controlled amount of noise from a carefully selected Laplace distribution, according to the desired level of privacy. The perturbed value is then submitted as the user's location to the LSP. To address the resource limitation of mobile devices, we use quad-tree based scheme to transform and to store users' context as bit stream. The generated bit stream can achieve a high compress ratio and support efficient retrieval. Our major contributions are as follows:

(1) We propose a context-aware differentially private location perturbation scheme that does not require a TTP and can protect a user's location privacy against an adversary with *side information*.

(2) We construct a MHC according to the density distribution of POIs in the considered local map and design a differential location perturbation algorithm based on it to protect user's location privacy in LBSs. This scheme provides strong privacy guarantees through the differential privacy. Due to the dimension reduction property of the modified Hilbert curve, the system overhead can also be reduced.

(3) We provide thorough security analysis and a comprehensive set of experiments to demonstrate the effectiveness of our approach to location privacy-preserving.

The remainder of this paper is organized as follows. In Section 2, we review the related works. Section 3 introduces some preliminaries of this paper. Section 4 presents the details of our proposed schemes. In what follows, we give the security and performance evaluation in Section 5. Finally, Section 6 concludes the paper.

2. Related Works

In the last few years, various privacy threats in terms of sharing location data have been identified in the literature. For instance, sharing location of a user not only diminishes his own privacy but also the privacy of others [18]. Even sharing the locations sporadically can still make adversary identify the user [19].

To cope with these threats, a variety of location privacy-preserving mechanisms and metrics have been proposed. In this section, we will review these related works.

2.1. Location Privacy Metrics. Since a location can be specified as single coordinate, to quantify the location privacy, we should find out how accurately an adversary might infer about this coordinate. Based on this principle, numerous privacy metrics have been proposed for quantifying the capability of the adversary. Location k-anonymity [3] and its variation like l-diversity [20] and t-closeness [21] are proposed to measure the ability of the adversary to differentiate the real user from others within the anonymity set. To overcome the drawbacks of k-anonymity in quantifying location privacy, entropy-based metrics have been adopted in [5, 13, 22, 23] for quantifying the information an adversary can obtain from one (or a series) of location update(s). Nonetheless, Shokri et al. [24] show a lack of satisfactory correlation between these two metrics and the success of the adversary in inferring the users' actual position. Therefore, they proposed the expected distance error metric to quantify the degree of accuracy by which an adversary can estimate a user's real position. However, this metric is explicitly defined in terms of the adversary's *side information* [25]. Once the adversary has no such *side information*, the expected distance error is not sufficient for quantifying location privacy. As a result, differential privacy [6, 26] that abstract from the adversary's *side information* has been growing popularity in LBS privacy protection, which measures the ability of the adversary with arbitrary background knowledge to obtain the user's real location. However, as noted in [27], this metric can be problematic if prior is taken into account.

2.2. Location Privacy Protection. In the past few years, many approaches for protecting location privacy are proposed to allow users to enjoy the LBSs while limiting the amount of disclosed sensitive information [3–15, 22, 26–33]. Although, among them, policy-based approaches and cryptography-based approaches [14] have also been investigated, most existing works are based on location obfuscation. For location obfuscation mechanisms, most of them employ well-known location k-anonymity to protect user's privacy by blurring user's exact location into a sufficiently larger CR. Because of its simplicity, k-anonymity metric has been widely adopted in many different methods, including IntervalCloak [3], clique-based cloak [5], location differential perturbations [8], game-theoretic approach [12], dummy location selection [13], and hilbASR [28]. However, these methods suffer from the single point of failure due to the reliance on a TTP named anonymizer. If an adversary seizes control of the TTP, the privacy of all users will be breached. This TTP is also a performance bottleneck since all the submitted LBS queries have to go through it. Moreover, these methods are vulnerable to background knowledge attacks and homogeneity attacks [20].

To avoid the use of TTP, many mobile device-based schemes [4, 6, 9, 10, 14, 15, 29–33] are introduced into LBS privacy protection LBS system. However, k-anonymity based schemes [10, 13, 29–33] still need to generate CR via exchanged information from other encountered mobile users. Thus, they also cannot resist homogeneity attacks and background knowledge attacks. Expected distance error based schemes [4, 9] obfuscate user's location by taking the adversary's *side information* into account, which also suffer from background knowledge attacks. Differential privacy based schemes [6, 26] have gained popularity as they abstract from the adversary's side information and are capable of providing strong worst-case privacy guarantees. However,

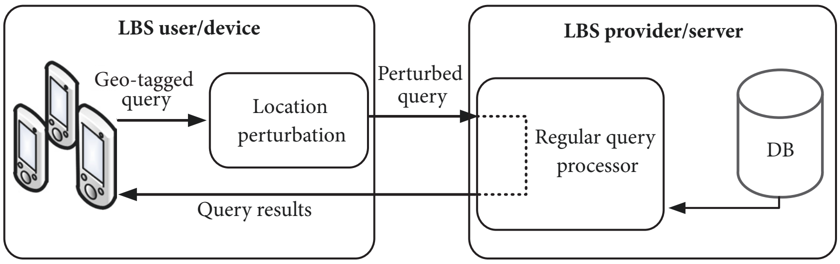


Figure 2: Architecture of our proposed framework.

these approaches do not take the contextual information of the user's location and are not sufficient to protect users from reidentification [34].

Different from existing works, our proposed method use MHC to store the context of the user's location, achieving robust privacy guarantee against the adversary with *some information*. It provides desired privacy level for mobile users without relying on any TTP.

Standard Hilbert curve (SHC) has been applied to some privacy protection schemes (e.g., [30, 32]), which is different from our MHC. Our MHC mapping is similar to the VHC-mapping [15], but there are several key differences. First, VHC-mapping is constructed from road density, but our MHC mapping is based on the density of POIs. Second, VHC-mapping is used to perturb a single location, but our MHC is used to select k POIs to preserve reciprocity [28].

3. Preliminaries

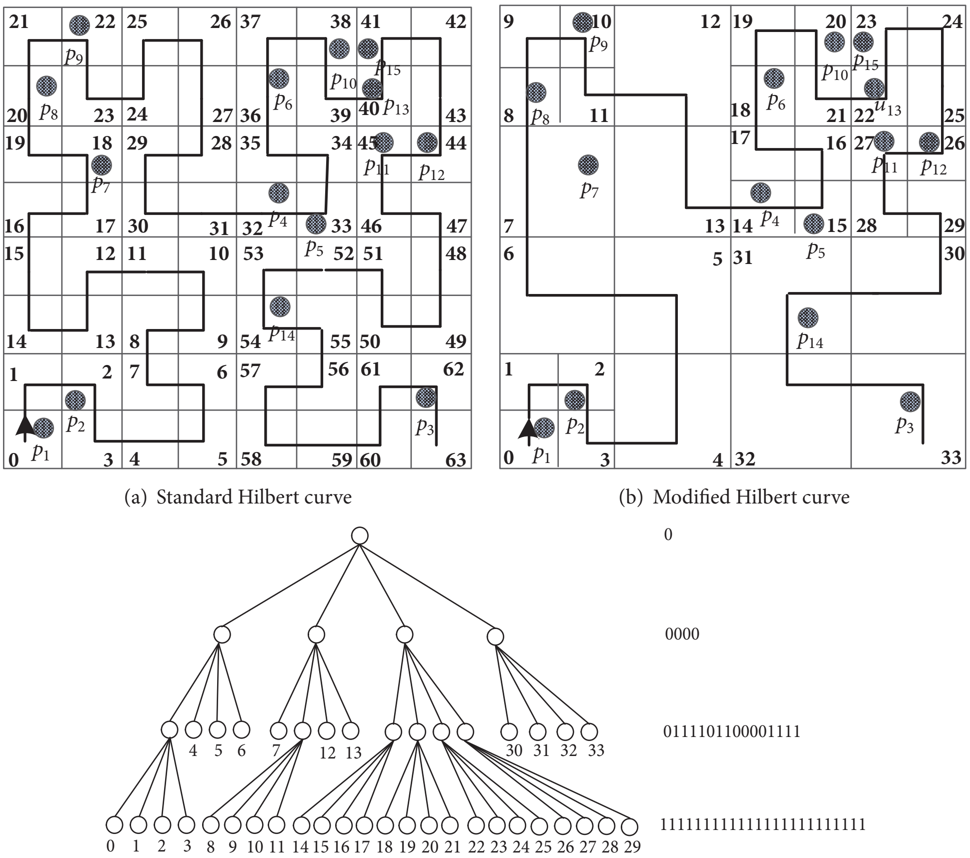
In this section, we first introduce the system model and some basic concepts used in this paper and then present the motivation and basic ideas of our scheme.

3.1. System Model and Basic Concepts. Our system model is composed of two parties: LBS user/device and the LBS provider/server, as shown in Figure 2. LBS user possesses a location-aware wireless device, capable of connecting to the network through a wireless protocol such as WiFi, GPRS, or 3G. LBS user uses location perturbation algorithm to perturb his location included in the LBS query, and he submits the perturbed query to LBS provider. The LBS provider is untrusted and considered as the adversary. He responds to the LBS user's requests and returns query results. He can also obtain all the side information by monitoring the queries issued from the LBS user. Additionally, he knows the location perturbation algorithm and noise distribution used in the system. Based on this information, he tries to perform inference attacks to deduce the user's location information.

In this paper, the side information is limited to the approximate location knowledge of users (an area instead of exact coordinates), which can be obtained by a variety of means, i.e., device communication logs such as cell towers used, public records such as parking violations, or social engineering methods such as during a casual conversation [8]. Unless regulated by legislations, the approximate location knowledge can more simply be inferred directly from the information broadcast from cell towers and wireless access points.

3.2. Motivation and Basic Ideas. In Figure 2, location perturbation component perturbs the user's location contained in the geo-tagged query to generate the perturbed query. It also rearranges the query results returned from the regular query processor of the LSP, in order to provide better LBS utility. Location perturbation is a straightforward approach to achieve efficient location privacy-preserving. However, this method may lead to other challenges, e.g., how to achieve context-aware privacy protection without incurring the cost of storing and retrieving a full-scale map in a mobile device, and how to generate a reasonable perturbation to make a trade-off between privacy and LBS utility. Most of existing works generate the perturbation by adding a random noise (to the true location) drawn from a standard probability distribution. However, it is not a good way to protect user's privacy against the adversaries with *side information* (e.g., a set of likely positions including the true location). With the *side information* and noise distribution, the adversary can calculate the probability of generating the observed perturbation from each of the likely positions. If the probability is significantly high for the real location, the adversary will confidently infer the user real position. To enhance user privacy, these probabilities should be within a small constant factor of each other. Our main idea is to employ a MHC mapping and a carefully selected Laplace distribution to achieve effective privacy-preserving. Our approach can be presented in two parts: (1) we use a MHC based on POIs density in considered local map to achieve the contextual information of the user's location and store it as bit stream; (2) then, we employ a carefully selected Laplace noise distribution to generate a reasonable perturbation and transmit the perturbed value as the user's location to the LSP.

Specifically, we observe that in location-based applications such as nearby searches and check-in posts in geo-social networks (e.g., Foursquare and WeChat), users tend to query an LBS from places that are meaningful to them (e.g., offices and restaurants). In such places, users are most likely to perform an activity without too much movement. We call these places the points of interest of users and refer them to the real POIs in local map. In addition, we assume that users request LBSs from their POIs. Let $\mathscr{R}$ be the (rectangle) boundary of the local map, Ψ be the set of all possible real POIs in $\mathscr{R}$, and $\Psi_u \in \Psi$ be the set of all POIs of user u. For



(a) Standard Hilbert curve

(b) Modified Hilbert curve

(c) Quad-tree for storage of MHC

Figure 3: Our modified Hilbert curve and its quad-tree storage.

simplicity, each $\psi_i \in \Psi_u$ can be approximately represented as a (x_i, y_i, ζ_i), where (x_i, y_i) is the location coordinate; ζ_i represents the semantic attribute of location coordinate (x_i, y_i), i.e., its semantic location. In this way, user's exposed locations can be transformed into his exposed POIs where he queries LBSs [35]. Indeed, it has been demonstrated that inference of POIs leads to a sever privacy breach [36].

To protect privacy, user usually selects a perturbed POI around him as his real location to request LBSs. Intuitively, to achieve the same level of privacy and LBS accuracy, the perturbed POI should be far more from his real location in a rural area than in downtown. To capture the contextual information in this case, we modify the standard Hilbert curve according to the density of POIs in considering local map and use it to fill the local map (as shown in Figure 3). Figure 3(a) shows the standard Hilbert curve (SHC) that covers the local map. Correspondingly, Figure 3(b) shows the modified Hilbert curve (MHC), which projects every 2-dimension POI of users in local map to a 1-d space. Normally, higher density of POIs leads to finer gains such that every point in the 1-dimension space has homogenous context (e.g., equal density). As we will show in the section of location differential perturbation, this prevents a location at a high POIs density area from receiving too large a perturbation to remain utility for LBSs.

Because of the superior distance preserving properties of Hilbert curve that two adjacent points in 1-dimension space are likely to be close in the original space, and vice versa [15], given a particular point, we can easily discover the adjacent points around. With this property, we first project all POIs in considering local map to 1-dimension space by using MHC. Then we randomly perturb the user's POI where he queries LBS, based on a carefully selected Laplace distribution to guarantee that the probabilities to report the same perturbed POI from a set of likely noise POIs including the true POI are similar. However, the distribution of these noise POIs can affect the proximity of the perturbed POI to the true POI. For instance, if POIs are perturbed based on the locations of every known POI within a city, the scale parameter in the noise distribution will become considerably high, thus leading to heavy noise addition. To solve the problem, we compute the perturbation from a restricted set of k POIs by using the reciprocal framework algorithm [28]. In this way, the probabilities of any POI in these k POIs generating the same perturbed POI are within a small constant factor (up to e^{ε}) of each other.

Formally, let $l_1, \ldots, l_k$ represent a set of k noise POIs, one of which is the real POI l_r=(x_r, y_r) of query user, and $p(\cdot)$ indicates the probability density function. For any two POIs l_i and l_j in these k noise, the perturbed POI l_p=(x_p, y_p)

```
Input: $\mathscr{R}$ as the (rectangle) boundary of the map, $\Psi$ as the set of all POIs in the map
Output: a quad-tree root node T
(1) if $|\Psi|$ in $\mathscr{R}$ is great than pre-determined threshold $\sigma$, then
(2)   partition $\mathscr{R}$ equally into four sub-cells $\mathscr{R}_{nw}$, $\mathscr{R}_{ne}$, $\mathscr{R}_{se}$, $\mathscr{R}_{sw}$;
(3)   for $i$=$nw$, $ne$, $se$, $sw$ do
(4)       return 0 //recursively partition $\mathscr{R}_i$ according to the condition $|\Psi_i|>\sigma$;
(5)   end for
(6) else
(7)   return 1 //outputs quad-tree root node T;
(8) end if
```

Algorithm 1: Modified Hilbert curve construction algorithm.

corresponding to the l_r is generated in a manner such that

$$p\left(l_p l_i\right) \le e^{\varepsilon} p\left(l_p l_j\right), \tag{1}$$

where $\varepsilon > 0$ and $i, j \in [1, k]$. The privacy parameter ε corresponds to the strength of the privacy guarantee: smaller ε yield more privacy. It has been shown that adding noise to each coordinate independently (by applying Laplace noise to each coordinate) provides the stronger protection than adding noise to each point independently (by generating 2-dimensional noise vector) [37]. Therefore, we use a Laplace distribution with scale b>0 to perturb each coordinate of the l_i=(x_i, y_i) independently such that

$$\begin{aligned} p\left(x_p \mid x_i\right) &= \frac{1}{2b} e^{-|x_i - x_p|/b}, \\ p\left(y_p \mid y_i\right) &= \frac{1}{2b} e^{-|y_i - y_p|/b}. \end{aligned} \tag{2}$$

The amount of noise to be added to each coordinate is given as $-b*\text{sign}(rnd)*\ln(1\text{-}2|rnd|)$, where rnd is a uniform random value in (-1/2, 1/2). Based on the following observation, x_p is generated by setting b as $(\max_n x_n\text{-}\min_n x_n)/\varepsilon$, and y_p is generated by setting b as $(\max_n y_n\text{-}\min_n y_n)/\varepsilon$. l_p is obtained as (x_p, y_p).

Observation 1. Using the triangle inequality, we have $|l_j - l_p| \le |l_j - l_i| + |l_i - l_p|$. After rearrangement, dividing by b, raising as a power of e and multiplying by $1/2b$, we get

$$\begin{aligned} \frac{1}{2b} e^{-|l_i - l_p|/b} &\le \frac{1}{2b} e^{-|l_j - l_p|/b} e^{|l_i - l_j|/b} \\ or\ p\left(l_p \mid l_i\right) &\le p\left(l_p \mid l_j\right) e^{|l_j - l_i|/b}. \end{aligned} \tag{3}$$

Therefore, for each coordinate, we have

$$\begin{aligned} p\left(x_p \mid x_i\right) &\le e^{|x_j - x_i|/b} p\left(x_p \mid x_j\right), \\ p\left(y_p \mid y_i\right) &\le e^{|y_j - y_i|/b} p\left(y_p \mid y_j\right), \end{aligned} \tag{4}$$

and the power of the exponent is bounded as

$$p\left(x_p x_i\right) \le e^{|\max_n x_n - \min_n x_n|/b} p\left(x_p x_j\right), \tag{5}$$

$$p\left(y_p y_i\right) \le e^{|\max_n y_n - \min_n y_n|/b} p\left(y_p y_j\right). \tag{6}$$

Consequently, the probability of a POI generating a certain perturbed POI is always with a factor e^{ε} of the probability of some other POIs in the set of k noise generating the same perturbed POI.

4. Location Differential Perturbation

In this section, we introduce the modified Hilbert curve construction algorithm and location differential perturbation algorithm in detail.

4.1. Modified Hilbert Curve Filling. Without loss of generality, we consider a set of users $\mathscr{U} = \{u_1, u_2, \ldots, u_n\}$ who subscribe certain LBSs and move in a local map. The (rectangle) boundary $\mathscr{R}$ of the local map is taken as a large cell. We recursively partition a cell into four equal-size cells if and only if the number of POIs within the original cell is greater than a predetermined threshold σ. One can see that each cell contains roughly σ or fewer POIs. Figure 3(b) depicts an example of such a partition. From the figure, we can see that each cell is either partitioned into four equal-size square cells, or not partitioned (i.e., becoming a base cell). The partitioning scheme can be readily represented as a quad-tree. Figure 3(c) depicts an example of such a quad-tree for the MHC mapping in Figure 3(b). In particular, each node in the tree either is a leaf node (if corresponding to a base cell) or contains four children (if further partitioned). Thus, to efficiently store the tree, we construct a breath-first traversal of the tree, storing 1 bit for each node indicating whether it is a leaf node or not. Since a quad-tree with n leaf nodes has at most $4n/3$ total nodes, the space required by the serialized map file is at most $4n/3$ bits. Thus, the total storage overhead is O (n). One can see that MHC covers the regions of high density of POIs with finer gains.

Algorithm 1 depicts the offline construction of MHC. In the algorithm, we partition original map based on predetermined threshold parameter σ (line (1)), recursively partition their children according to the given conditions and store the quad-tree into a bit stream (line (4)). The computational complexity of Algorithm 1 is O (n).

After the partitioning process, we construct the mapped 1-d space as variation of the Hilbert space-filling curve [38] to connect all cells in the original 2-d space. To assign a corresponding range in the 1-d space for each base cell, we

```
Input: the T obtained from Algorithm 1, starting point $S_0$ and curve orientation $\theta$
Output: a updated quad-tree root node T
(1)  initializes $S(T) = S_0$, $\theta(T) = \theta$, $m = 0$;
(2)  push T into the stack;
(3)  while (stack is not empty) do
(4)      N = pop the top element from the stack
(5)      if (N has child node) then
(6)          for ($i$= $sw$, $se$, $ne$, $nw$) do
(7)              set $S(N_i)$, and $\theta(N_i)$
(8)              push $N_i$ into the stack
(9)          end for
(10)     else
(11)         $h_N = m$
(12)         set the values of all corresponding POIs in the node N as $m$
(13)         $m = m + 1$
(14)     end if
(15) end while
(16) outputs the updated quad-tree root node T.
```

Algorithm 2: Hilbert value generation algorithm for each base cell.

need to traverse every leaf node. To this end, we conduct a depth-first traversal of tree T, assigning the Hilbert value in the 1-d projected space for each leaf node according to its visiting orders. Let $S(\mathrm{N})$, $\theta(\mathrm{N})$ be the orientation and starting point of the Hilbert curve of the node N. The Hilbert value corresponding to the node N is denoted by h_{N}. The formal description of our Hilbert value generation algorithm can be found in Algorithm 2.

In Algorithm 2, we construct a depth-first traversal over the quad-tree. In particular, we start from the root node T (lines (1)-(2)) and determine its every child node's curve orientation θ and starting point S in the manner of drill-down according to the fractal rules used in our recent work [39] (lines (7)-(8)). We repeat this process until reaching a leaf node and set the Hilbert value of this leaf node as m (line (11)). In such way, every leaf node is assigned to a unique Hilbert value. Correspondingly, the Hilbert value of all POIs in every leaf node is also obtained (line (12)). The computational complexity of Algorithm 2 is $O\ (n)$.

4.2. Location Perturbation Algorithm. In Section 3.2, we provide a method to generate a perturbed POI for query point from the carefully selected k POIs by using the Laplace distribution. As mentioned before, the k POIs should be chosen to preserve reciprocity. That is, the same anonymous set should be obtained irrespective of which of the k POIs is the query point. This is achieved by using the reciprocal framework algorithm [28], which partitions the POIs of user into k size buckets based on the Hilbert value of the POIs. The anonymous set is selected as the bucket to which the query point belongs. Each of the k POIs is used for perturbation and the one having the minimum average distance to all POIs in the anonymous set is chosen as the user's location to issue the query. The formal description of our location perturbation algorithm can be found in Algorithm 3.

In Algorithm 3, we firstly index the all-possible POIs by a quad-tree spatial index and assign the Hilbert value for each POI (Line (1)). This step has time complexity of $O(n)$. Then we find the mapped value based on the 1-d value range of the base cell which contains ψ_u (line (2)). One can see that the retrieval process requires access at most $\log n$ (the depth of tree) nodes, leading to computational complexity of $O(\log n)$. Based on the Hilbert indices of the POIs, we determine the k size bucket to which the ψ_u belongs by using reciprocal framework (lines (3)-(9)), which has time complexity of $O(\log n)$. The locality preserving properties of Hilbert curves guarantee the formation of buckets with POIs that are at close proximity to each other. Lines (10)-(14) compute a perturbed value corresponding to the k POIs in the bucket by using Laplace distribution. Thus, each coordinate c of a POI is perturbed to $c - b * \mathrm{sign}(rnd) * \ln(1\text{-}2|rnd|)$, where rnd is a uniform random value in (-1/2, 1/2), and b is set as $(\max_n c_n\text{-}\min_n c_n)/\varepsilon$. This makes perturbation Laplace distributed around c. In the following experiment, the retrieval of the MHC mapping requires less than 0.1s in our system, and the perturbation requires less than 0.5s.

4.3. Security Analysis. In this section, we provide security analysis. In the context of location privacy, we consider two types of adversaries: active adversary and passive adversary. The purpose of the passive adversary is to obtain sensitive information about a particular user by eavesdropping on the wireless channel or compromising the LBS provider. Actually we can use some cryptography tools such as public key infrastructure (PKI) to cope with the eavesdropping attacks on the wireless channel between users and other entities. Thus, we mainly focus on how to avoid collusion attacks and inference attacks from active adversary, both of which can cause serious privacy problems.

Adversary may be collusion with some users or the LBS server to capture the other user's private information.

```
Input: Query user $u$ with associated k, his POI $\psi_u$ where he queries LBSs, and the pre-
        computed MHC filling file mhcFile
Output: A perturbed location $l_p$ for $u$
load a quad-tree T of the partition from mhcFile and use Algorithm 2 to assign the Hilbert
    value for each POI;
find the leaf node N containing $\psi_u$;
while (there is non-empty node at the same level as N with < k POIs) do
    N = parent of N //bottom-up traversal
end while
while (N is not a leaf and (each child of N is either empty or contains ≥ k POIs)) do
    N=child of N that contains $\psi_u$ //top-down traversal
end while
obtain the L={$l_1, \ldots, l_r, \ldots l_k$} by splitting the POIs inside sub-tree of N into buckets
    containing between k and 2k-1 POIs using reciprocal algorithm
$L_p = \phi$
for ($l \in$ L) do
    $l_p = l + z_i$, where $z_i$ is additive noise generated by Laplace distribution
    $L_p = L_p \cup \{l_p\}$
end for
output $l_p \in L_p$ such that $l_p$ has minimum average distance from L.
```

Algorithm 3: Location perturbation algorithm.

Theorem 1. *Our scheme is collusion attack resistant.*

Proof. We contemplate that the collusion attack occurs between a set of users. On the one hand, each user is independent with others in our scheme. He only needs to use his position and the stored Hilbert index file to generate the perturbation instead of interacting with the other users. On the other hand, Algorithm 3 in our scheme guarantees that all the processes are executing locally, not dependent on other users at all. That is, it is useless for the adversary to capture and collude with nearby users. The best case to this kind of adversary is that he can obtain the global information by capturing the LBS server and all the users, but in this case he becomes an active adversary to perform inference attack.

In our scheme, we directly contemplate the untrusted LBS server as the active adversary to perform the inference attack. He can get *side information* by monitoring all the users in the system, including their interests, approximately location (e.g., a set of likely positions including the true location), LBS queries, and observed perturbation. His aim is to use this *side information* to confidently infer real position of the query user. □

Theorem 2. *Our scheme is inference attack resistant under ε-differential privacy.*

Proof. In our scheme, users need to issue the queries to the adversary in order to enjoy the LBSs. Ideally, due to the perturbation, the adversary cannot construct any direct linkage from the perturbed locations to a user. However, the adversary knows the POIs density of the whole map, approximately locations for a user and noise distribution. Based on this information, the adversary can perform inference attacks to gain the real location of the query user. More formally, the adversary knows the set of all POIs, Ψ, a set of positions, $l_1, \ldots, l_r, \ldots, l_k$ (including query user's real location), location perturbation mechanism, and the noise distribution $p(l_i)$. As certain position in the adversary's approximate knowledge is highly unlikely to generate the observed perturbation under the used noise distribution, the adversary can use newly learned distribution to improve its probability of successfully guessing the real location from these equally likely positions. In our algorithm, the inference attack is avoided by using reciprocity framework and ε-differential privacy. First, since the k POIs set $L=\{l_1, \ldots, l_r, \ldots, l_k\}$ generated by using MHC method satisfies reciprocity, the probability of identifying the query user's real POI does not exceed $1/k$ [28]. Second, as discussed above, due to the usage of differential privacy, the probability to report the same observed perturbed location l_p from the positions $l_1, \ldots, l_r, \ldots l_k$ is within a small constant factor of each other. The Laplace noise added to a POI depends on the component-wise maximum distance between two positions. As long as the scale parameters use these maximum, the perturbed POI (x_p, y_p) will satisfy the probability ratio. In our scheme, we use differential privacy notion to abstract the *side information* and guarantee the security efficiently. For any two POIs l_i and l_j in the set L, the adversary's *side information* can be modelled by two prior distributions $p(l_i)$ and $p(l_j)$. After observing the perturbed POI l_p, the adversary could use the l_i and l_j as input to differential location perturbation algorithm and compute the conditional probabilities $p(l_p \mid l_i)$ and $p(l_p \mid l_j)$. For the purpose of modelling the adversary's observation, we use Bayes' rule to obtain the posterior distribution:

$$p\left(l_i = l_r \mid l_p\right) = \frac{p\left(l_p \mid l_i\right) p\left(l_i\right)}{\sum_{c=1}^{k} p\left(l_p \mid l_c\right) p\left(l_c\right)},$$
$$p\left(l_j = l_r \mid l_p\right) = \frac{p\left(l_p \mid l_j\right) p\left(l_j\right)}{\sum_{c=1}^{k} p\left(l_p \mid l_c\right) p\left(l_c\right)}. \qquad (7)$$

We use multiplicative distance to metric the distance between two distributions as

$$d_p = \sup_{l \subseteq L} \left| \ln \frac{p\left(l_i \mid l_p\right)}{p\left(l_j \mid l_p\right)} \right| \leq \varepsilon \tag{8}$$

According to the definition of ε-differential privacy, the d_p should be at most ε. Substituting formula (7) into formula (8), we can get

$$p\left(l_p \mid l_i\right) \leq e^{R/b} p\left(l_p \mid l_j\right), \tag{9}$$

where R is the radius of maximum perturbation range which satisfies Laplace distribution, also the maximum distance between any two noise POIs, and b is the scale parameter of Laplace distribution. Thus, formula (9) can be extended to formulas (5) and (6).

As can be seen from formula (9), our scheme is independent of the prior distribution. This is to say, the probabilities that the adversary uses *side information* to report the same observed perturbed location l_p from the positions $l_1, \ldots, l_r, \ldots l_k$ are within a small constant factor of each other. Thus, the adversary cannot use such *side information* to improve its probability of successful guessing the real location. □

4.4. Query Accuracy Analysis. In this section, we provide LBS query accuracy analysis. Using perturbed locations do affect the accuracy of query results. However, difference in the results may or may not exist depending on the distance between the perturbed location and the true location. Therefore, one has to trade-off between location privacy and LBS utility. In order to formally analyze the query accuracy of our location perturbation scheme, we consider three metrics with respect to KNN query [8]: *Nearness*, *Resemblance,* and *Displacement*.

(i) *Nearness*: it indicates the ratio of perturbation at close proximity to the true location.

(ii) *Resemblance:* it depicts the accuracy rate of query results retrieved by a KNN query related to a perturbed location. Let $O = \{o_1, o_2, \cdots, o_K\}$ be the objects retrieved by a KNN-query relative to the true location of user u, and $O' = \{o'_1, o'_2, \cdots, o'_K\}$ be the objects retrieved relative to the perturbed location. The resemblance is the rate of common objects between O and O', given as

$$Q_{AR} = \frac{\left|O \cap O'\right|}{|O|}, \tag{10}$$

where $|O|$ is the number of query objects in the real results set O, $|O \cap O'|$ is the number of common objects between O and O'.

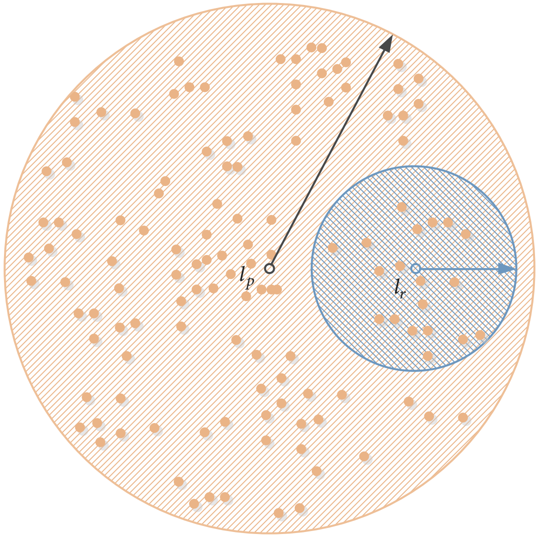


Figure 4: AOI and AOR with centre l_r and l_p.

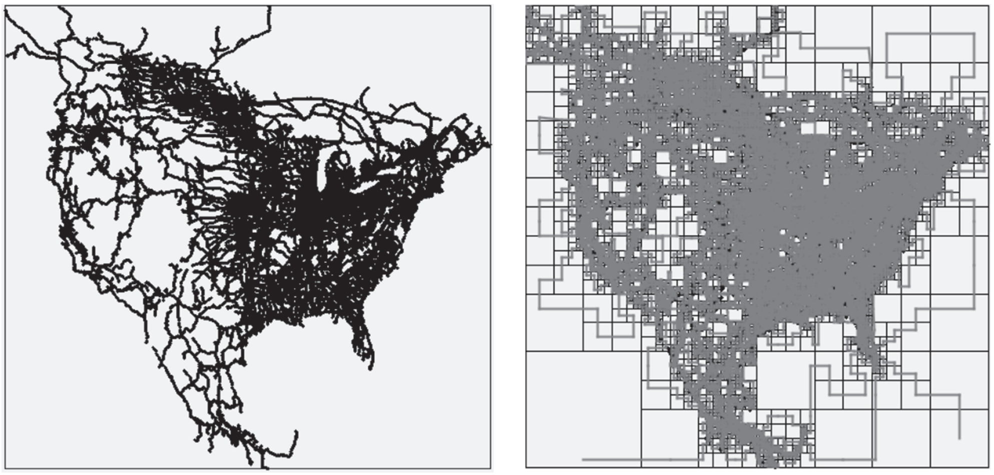
(iii) *Displacement:* it captures the average difference in distance between the actual results and real retrieval results of a KNN-query, given as

$$Q_P = \begin{cases} \dfrac{1}{K}\left[\displaystyle\sum_{i=1}^{K} dist\left(O'_i, q\right) - \sum_{i=1}^{K} dist\left(O, q\right)\right], & O \neq O' \\ 0, & O = O', \end{cases} \tag{11}$$

where q is the real query POIs of a user and the $dist(\cdot)$ is the Euclidean distance between an object's location and the true location of a user. It should be noted that the lowercase k is used to calculate anonymous set and the uppercase K is used to calculate KNN query.

These three metrics are used to measure the effectiveness of our scheme. The resemblance measures the query accuracy with respect to the perturbed location, while the displacement measures the expected distance error between the real query results and actual retrieval results. In this part of theoretical analysis, we adopt the *Resemblance* metric as the query accuracy measure. Nevertheless, in the following experimental evaluation, we also evaluate the *Nearness* and *Displacement* metrics.

As shown in Figure 4, we define the blue circle as the query area of interest (AOI) with regard to the real location l_r and the orange circle as the area of retrieval (AOR) with respect to the perturbed location l_p. In order to guarantee high *Resemblance* (see formula (10)), ideally, the AOR should always completely contain the AOI. Unfortunately, this condition cannot be guaranteed because of the nature of our location perturbation (note that the AOR is centred on a randomly generated location that can be arbitrarily distant from the real location). In order to measure the probability of such event, we introduce the notion of accuracy. Specifically, we use r_I and r_R to represent the radius of the AOI and



(a) Road network of NA

(b) MHC filling of NA with σ=10

Figure 5: NA dataset.

the AOR, respectively, M to denote the location perturbation mechanism, and $C(x, r)$ to denote the circle with centre x and radius r.

Definition 3. An LBS perturbation is $(c,\ r_I)$-accurate *iff* for all locations x we have that $C(x, r_I)$ is fully contained in the $C(M(\mathrm{x}), r_R)$ with probability at least c.

Give a privacy parameter ε and accurate parameters $(c,\ r_I)$, our goal is to obtain an LBS perturbation $(M,\ r_R)$ satisfying both ε-differential privacy and $(c,\ r_I)$-accurate. As for a perturbation mechanism M, we use the Laplace perturbation M_ε discussed in Section 3.2, which satisfies ε-differential privacy. As for r_R, we attempt to find a minimum value validating the accurate condition. To achieve this goal, we use the notion of (α, δ)-usefulness, which was introduced in [40]. A location perturbation mechanism M is (α, δ)-usefulness if for every location x the perturbed location $z = M(x)$ satisfies $dist(x, z) \leq \alpha$ with probability at least δ. In our perturbation mechanism M_ε, we computer the perturbation from a restricted set of k POIs that preserve reciprocal. This guarantee that our M_ε can generate reasonable perturbation range. Therefore, the α and δ values which express M_ε usefulness are related by $-b{*}\mathrm{sign}(rnd){*}\ln(1\text{-}2|rnd|)$, the noise amount of our perturbation.

Observation 2. For any $\alpha > 0$, M_ε is (α, δ)-usefulness if $\alpha < \max_{i,j\in[1,k]} dist(l_i, l_j)$, where l_i and l_j are determined by $-b{*}\mathrm{sign}(rnd){*}\ln(1\text{-}2|rnd|)$.

In the following experimental evaluation (as shown in Table 2), we set various ε to compute the percentage of the perturbations which are within 1km, 0.5km, and 0.1km of the user's true position. As our running example, our perturbation mechanism M_ε (ε-differential privacy, with $\varepsilon = 0.5$) generates a perturbed location l_p falling within 1km of the real position l_r with probability 0.9426.

According to the definition of usefulness, if M_ε is (α, δ)-usefulness, then the LBS perturbation $(M_\varepsilon,\ r_R)$ is $(\delta,\ r_I)$-accurate if $\alpha < \max_{i,j\in[1,k]} dist(l_i, l_j)$. The converse also holds if δ is maximal. By Observation 2, we have the following.

Proposition 4. *The LBS perturbation* (M_ε, r_R) *is* $(\delta,\ r_I)$-*accurate if* $r_R \geq r_I + \delta \cdot \max_{i,j\in[1,k]} dist(l_i,\ l_j)$.

Therefore, it is sufficient to set $r_R = r_I + \delta \cdot \max_{i,j\in[1,k]} dist(l_i, l_j)$. Thus, our perturbation (M_ε, r_R) satisfies both ε-differential privacy and $(\delta,\ r_I)$-accurate.

5. Experimental Evaluation

This section evaluates the proposed differential location perturbation algorithms. We implemented the algorithms using Java program. All experiments were executed on an Intel Core i7-4790 3.6GHz machine with 4G RAM and Windows OS. The perturbation scheme indexes the all-possible POIs of the considering local map, which are taken from the NA dataset (available at http://www.cs.utah.edu/~lifeifei/SpatialDataset.htm) containing 175813 real POIs of the North America road network (see Figure 5(a)). The parameter k is set from 10 to 1000. The results are obtained by taking the average of 100 times simulation of the corresponding algorithms.

Several parameters are employed in our evaluation. S_0 is the starting point of Hilbert curve, and its default value is (0, 0). The θ represents the Hilbert curve direction, and its default value is D_1 (see Figure 6) [39]. Γ is the scale factor of Hilbert curve, and its default value is 1. k is related to k-anonymity. Figure 5(b) shows a real MHC filling for the North America road network with σ=10. From the figure, an intuitive observation is that the denser regions represent large cities, while the sparse regions represent the rural areas. Thus, MHC mapping captures the contextual information well. In [9], Shokri proposes an optimal location privacy preservation strategy by solving a linear program, which avoids TTP.

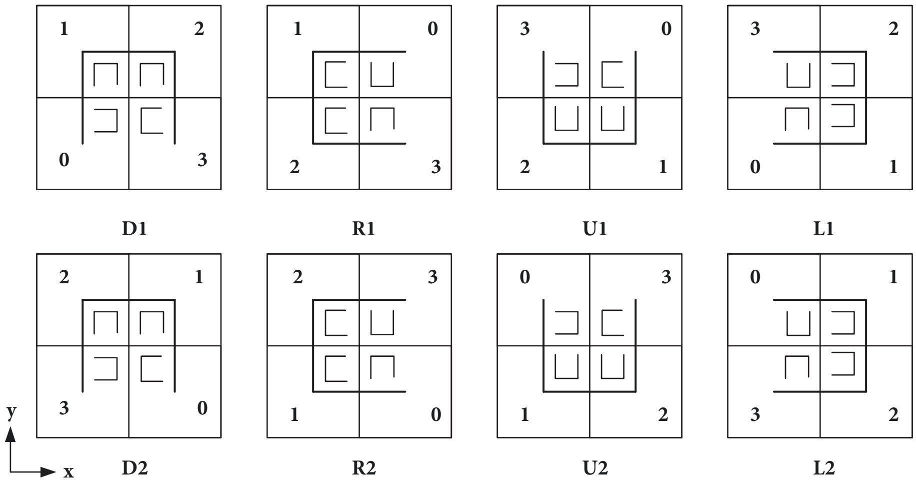


Figure 6: Fractal rules of Hilbert curve.

This optimal strategy computes location obfuscation probability distribution function to maximize the location privacy, subject to service quality constraints. However, this method depends on the modelling of adversary's side information and thus suffers from background knowledge attacks. As can be seen from previous analysis, our method abstracts adversary's side information. Furthermore, this optical strategy has nothing to do with contextual information and Hilbert curve. Therefore, our method cannot be comparable to this optical strategy. Location perturbation method in [8] is similar to our approach and uses SHC, whereas it still employs a TTP. For the purpose of comparing with [8], we implement the method in [8] under the same setting as our method. To generate the k-anonymous sets, the location perturbation scheme in [8] employs SHC to calculate the Hilbert index value of users' location online. Different from the method in [8], our scheme employs MHC to calculate the Hilbert indices of the POIs and stores them as a binary map file. We use the quad-tree recovered from the binary map file to find the node where the user's POIs are located and then generate the k-anonymous set. Therefore, in the following experiment, we evaluate the performance of the location perturbation by comparing MHC mapping with SHC mapping.

5.1. Parameters Selection for MHC and SHC. During the partitioning process, MHC and SHC employ different curve parameters. To carry out the following experiments under the same standard, we first examine the parameters selection for MHC and SHC. When the geographic space is filled by using MHC mapping, a unique index value is assigned to each atomic region according to the traversal order of the Hilbert curve. The index values of the POIs contained in the atomic region are also index value of the atomic region. Thus, we can obtain the Hilbert index of all POIs. If some POIs are in the same base cell, they are overlapped. We define the overlap factor λ to describe the overlap of the POIs for each base cell such that

$$\lambda = \frac{1}{M}\sum_{i=0}^{H} n_i \tag{12}$$

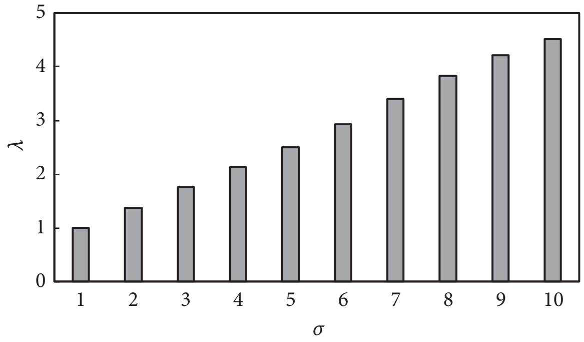


Figure 7: The relationship between σ and λ.

where M is the number of base cells that contain POIs, H is the upper bound of the Hilbert index value, and n_i represents the number of the POIs whose index value is equal to i.

Figure 7 illustrates the case that the MHC overlaps factor λ changes with the division threshold σ. We find that λ grows slowly as σ increases, and when σ is 1, the overlap factor is also 1. This is determined by the definition of the overlap factor and the division in Algorithm 2. The MHC can achieve finer grains via setting the threshold σ.

Figure 8 illustrates the relationship between the standard Hilbert curve degree D and the overlap factor λ when the map of the NA dataset is divided by the SHC. As can be seen from the figure, the larger D is, the smaller λ is and the finer grain that the partition leads to. Since the SHC employs the uniform standard to divide the space, the λ changes greatly with the changing of the D when D is small. In [8], the SHC mapping technique was employed to divide the entire map into a grid of $2^{14} * 2^{14}$ while calculating the Hilbert indices, which guarantees that there is not more than one user in each division. Objects in the same division have the same Hilbert index. This is because a larger curve degree D can lead to a finer granularity division of spatial maps. Nevertheless, the greater curve degree may lead to high computational overhead unaffordable to the server.

Table 1: Hilbert index generation time (ms).

Algorithm	$\lambda = 1$	$\lambda = 1.5$	$\lambda = 2.7$	$\lambda = 4.9$
	σ=1, D=13	σ=2, D=11	σ=5, D=10	σ=10, D=9
SHC	1237	1067	988	903
MHC	892	521	300	214

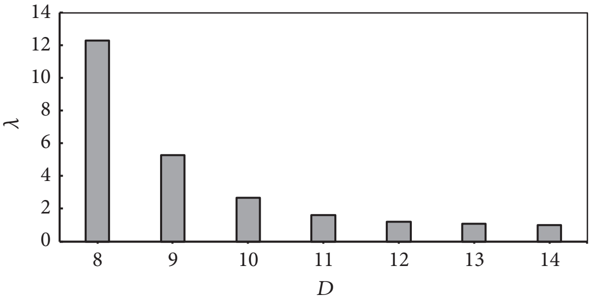


Figure 8: The relationship between λ and D.

The generation time of the index of POIs is an important measure when using the spatial filling curve to divide the considered map. We compare the index generation time of our scheme using MHC padding algorithm HVGA with the scheme using SHC padding algorithm EDHO in [8]. The HVGA represents the Hilbert value generation algorithm (see Algorithm 2). The results are shown in Table 1.

As seen from the table, in the case of the same λ, the efficiency of Hilbert index generation via using the MHC mapping technology in our scheme is significantly higher than that via using SHC mapping technology in [8], and with the increase of λ, the result of using MHC is more obvious. This is because the MHC partition considers the density distribution of POIs and uses different curve degree D for different density regions, which enable the partitioning of the lower density region not use high D, thus improving the efficiency of index generation.

When $\sigma \geq 10$, the MHC index generation time changes very slowly. Therefore, in all the following experiments we considered the $\lambda = 4.9$, $\sigma = 10$, $D = 9$.

5.2. Anonymous Evaluation. We compare the average anonymous set generation time of our scheme and the scheme in [9] for varying k (see Figure 9).

As can be seen from Figure 10, as the k increase, the anonymous set generation time for MHC perturbation (see Algorithm 3) and SHC perturbation (see [8]) does not vary significantly. This is because, in both perturbation algorithms, to select the k-anonymous set that satisfies reciprocity we only need to traverse the small subtree determined by node N in the quad-tree T, and the data structure of the intermediate nodes of the quad-tree T contains all the POIs in its subtree, so there is no need to traverse their subtrees to obtain this information. In the case of the same k, the anonymous sets generation time of MHC perturbation is much lower than that of SHC perturbation, with an average reduction about 66%. This is because when the location index is generated, the SHC perturbation is partitioned by the uniform granularity for all the regions; nevertheless, the MHC is divided according to the density distribution of the POIs. The MHC partition of the sparse regions uses lower curve orders and thus reduces the division time. That is to say, in the same case, the MHC partition traverses fewer subtrees than the SHC partition.

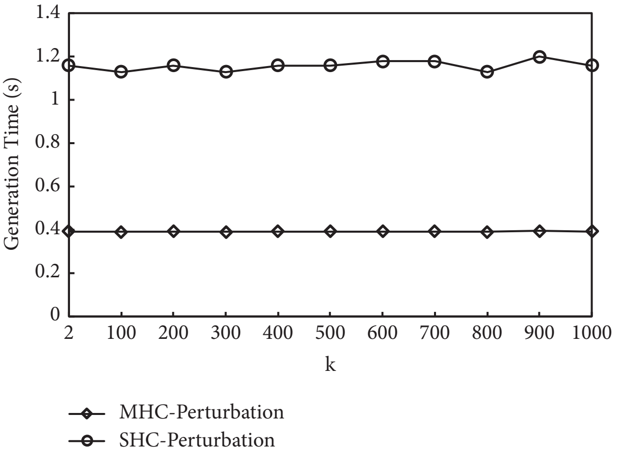


Figure 9: Anonymous set generation time for varying k.

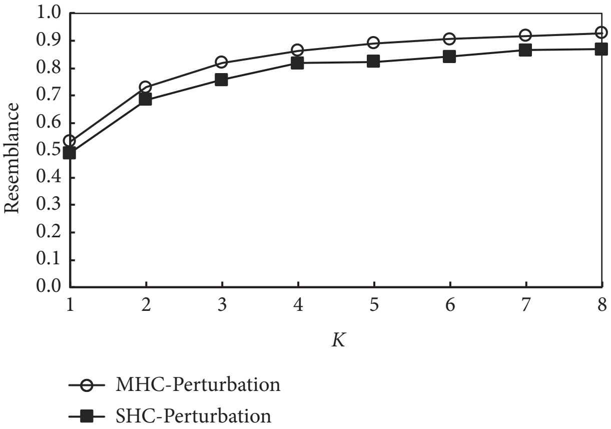


Figure 10: Query accuracy rate of KNN retrieval.

5.3. Differential Location Perturbation Evaluation. In this section, we evaluate the performance of our differential location perturbation algorithm by comparing MHC mapping with SHC mapping. From Section 3.2, we know that the probability ratio of generating the perturbed position (x_p,

Table 2: Percentage of the generated perturbed positions that is at close proximity to true location.

ε	Nearness/%		
	d ≤1000m	d ≤500m	d ≤100m
0.01	2.39	0.95	0.01
0.1	28.2	16.27	8.13
0.3	89.3	61.24	7.18
0.5	94.26	62.20	11.48
1.0	99.04	73.68	27.27
2.0	99.99	91.87	33.97

y_p) at any two POIs in the anonymous is bounded as e^{ε}. The amount of Laplace noise to be added to the position depends on the maximum distance of the corresponding coordinates of the two positions. As long as the Laplace scale parameter b uses these maximum distances, the probability ratio of the any two POIs in anonymous set generating a perturbed position (x_p, y_p) always satisfies the bounded e^{ε}. ε is privacy budget and smaller ε yields more privacy, but leading to less accuracy. In the following experiment, we evaluate the accuracy of our scheme for a scenario, where we issue a KNN query for nearest POIs. In particular, we use the *Nearness*, *Resemblance,* and *Displacement* metrics to measure LBS accuracy.

(1) ***Nearness***: for the Nearness metric, we set the different privacy parameter ε to calculate the percentage of the perturbations that resulted in the perturbed point being generated within 1000 m, 500 m, and 100 m of the user's true position. The results are shown in Table 2.

As can be seen from the table, a value of $\varepsilon = 0.01$ indicates that two users should have the same probability ($e^{\varepsilon} = 1.01$) to generate perturbations. This is difficult to achieve most values of k. When the ε value reaches 0.5 ($e^{0.5} = 1.65$), more than 90 percent of the perturbed points are within 1000 meters of the real position. More than 60 percent of the perturbed points are within 500m of the true position. The number of perturbed points increases with increasing of the ε value. However, higher ε value reduces the practical significance of the approach. For example, the value of $\varepsilon = 2.0$ means that a factor of 7 differences in the probability estimates ($e^{2.0} = 7.39$) must be accepted. Nonetheless, high nearness values with smaller value of ε are also possible as well.

(2) ***Resemblance and Displacement***: as previously observed in Table 2, about 95% perturbed points fall within 1km of the real position when ε=0.5. Therefore, for the resemblance metric Q_{AR} and displacement metric Q_P, we set $\varepsilon = 0.5$ to generate the perturbations in the experiments. Figures 10 and 11 show the evaluation results of the *Resemble* and *Displacement* corresponding to different values of K (the number of the nearest neighbour objects retrieved by KNN).

From Figure 10, we can see that increasing the number of nearest neighbouring objects to search K enhances the similarities of the result set. As K increases, the query accuracy rate of KNN retrieval varies from around 60 percent to almost 90 percent. This is because that a greater number of retrieved results can be seen as enlarging the search radius, in which case, an object becomes more likely in the KNN set of more number of location-based queries. For example, searching for the nearest theatres from two different locations in a city, we can expect that these two different locations have higher overlap in their list of 10 nearest theatres. The extent of the overlap depends on how proximal the two locations are to each other. Therefore, the noise added to a location is important in this regard. Meanwhile, the retrieval accuracy of a KNN using MHC is increased by an average 15.4% compared with the approach using SHC. This is because the MHC partition considered the density distribution of the POIs, which needs only a small perturbation to achieve a high level of privacy-preserving in a densely populated area.

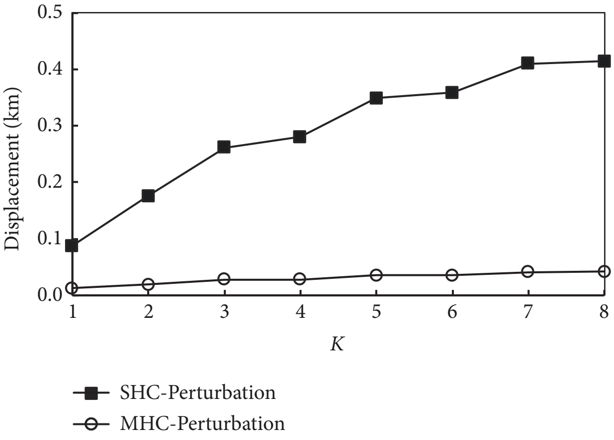


Figure 11: Query precision of the approximate KNN search.

The query precision indicates the average difference in the distance between the actual results and real retrieval results of a KNN-query based on the real location and the perturbed position, which is more effective in measuring the quality of retrieved results. Figure 11 shows the results of the query precision.

From Figure 11, we can clearly see that the query precision of a KNN retrieval related to the perturbed location generated by using SHC perturbation varies from about 120m to 350m and that the query precision of a KNN retrieval related to the perturbed location generated by using MHC perturbation is within 50m. This shows that the query precision of a KNN retrieval related to MHC perturbation is smaller than that of a KNN retrieval related to SHC perturbation. The reason is that MHC considers the contextual information of the POIs, thereby resulting in smaller perturbation than

SHC. The results show that the quality of a KNN retrieval results related to MHC perturbation is higher than that of the results related to SHC, which also corresponds with the character that MHC conducts granularity partition of the defined spaces according to density distribution of the POIs.

6. Conclusion

Driven by the prosperity of smart mobile devices equipped with GPS, location-based services, as an import part of Green Mobile Communications and Networks (GMCNs), have become very popular recently in almost all business and society domain. Since these services access private position information, location privacy protection mechanisms are mandatory to ensure the user acceptance of such services. The location-based confounding mechanism based on the cloaking area is a wide range of research techniques to achieve location privacy protection, but most of these technologies rely on TTP and assume that the attacker does not have side information, thus easy to cause location privacy disclosure.

In this paper, we proposed a context-aware differential location perturbation technique to protect user privacy. Our scheme, the context information of the user's location is considered in the event of a perturbation, and the attack of the background information can be effectively prevented without depending on any TTP. We use MHC mapping technology to project each 2-d geographic location of the user on the map into 1-d space and combine the k anonymous with the differential privacy techniques to randomly disturb the user's location, and then to submit the perturbation as the user's real location to the location service provider. In order to solve the limited resources of mobile devices, we use a quad-tree based approach to transform and to store the users' context to support efficient retrieval and storage. Through the security analysis and experimental evaluation, we can find that our scheme can resist the inference attacks of approximate position knowledge. Using the perturbation position will not significantly improve the attacker's prior knowledge about the user's position, so it has strong privacy protection. However, the identification of some unreasonable perturbation is still a problem. In the future work, we will consider abandoning the anonymity sets to address this problem.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 61762058, in part by the Scientifically and Technological Project of Gansu Province under Grant 1610RJZA056, in part by the Key Laboratory Opening Project of Opto-Technology and Intelligent Control Ministry of Education under Grant KFKT2016-7, and in part by the Youth Science Foundation of Lanzhou Jiaotong University under Grant 2014026.

References

[1] M. Ismail, W. Zhuang, E. Serpedin, and K. Qaraqe, "A survey on green mobile networking: From the perspectives of network operators and mobile users," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 3, pp. 1535–1556, 2015.

[2] X. J. Zhang, X. L. Gui, and Z. D. Wu, "Privacy preservation for location-based services: a survey," *Journal of Software*, vol. 26, no. 9, pp. 2373–2395, 2015, (in Chinese with English abstract).

[3] M. Gruteser and D. Grunwald, "Anonymous usage of location-based services through spatial and temporal cloaking," in *Proceedings of the 1st International Conference on Mobile Systems, Applications and Services, MobiSys 2003*, pp. 31–42, May 2003.

[4] S. Oya, C. Troncoso, and F. Pérez-González, "Back to the drawing board: Revisiting the design of optimal location privacy-preserving mechanisms," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1959–1972, Dallas, Texas, USA, October 2017.

[5] X. Pan, J. L. Xu, and X. F. Meng, "Protecting location privacy against location-dependent attacks in mobile services," *IEEE Transactions on Knowledge and Data Engineering*, vol. 8, no. 24, pp. 1506–1519, 2012.

[6] N. E. Bordenabe, K. Chatzikokolakis, and C. Palamidessi, "Optimal geo-indistinguishable mechanisms for location privacy," in *Proceedings of the 21st ACM Conference on Computer and Communications Security, CCS 2014*, pp. 251–262, November 2014.

[7] R. Schlegel, C.-Y. Chow, Q. Huang, and D. S. Wong, "User-defined privacy grid system for continuous location-based services," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2158–2172, 2015.

[8] R. Dewri, "Local differential perturbations: location privacy under approximate knowledge attackers," *IEEE Transactions on Mobile Computing*, vol. 12, no. 12, pp. 2360–2372, 2013.

[9] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: Optimal strategy against localization attacks," in *Proceedings of the 2012 ACM Conference on Computer and Communications Security, CCS 2012*, pp. 617–626, October 2012.

[10] H. Lu, C. S. Jensen, and M. L. Yiu, "Pad: privacy-area aware, dummy-based location privacy in mobile services," in *Proceedings of the Seventh ACM International Workshop on Data Engineering for Wireless and Mobile Access*, pp. 16–23, Vancouver, Canada, June 2008.

[11] C. Y. Ma, D. K. Yau, N. K. Yip, and N. S. Rao, "Privacy vulnerability of published anonymous mobility trace," in *Proceedings of the 16th Annual International Conference on Mobile Computing and Networking*, pp. 186–196, Chicago, IL, USA, 2010.

[12] X. Liu, K. Liu, L. Guo, X. Li, and Y. Fang, "A game-theoretic approach for achieving k-anonymity in location based services," in *Proceedings of the 32nd IEEE Conference on Computer Communications (INFOCOM '13)*, pp. 2985–2993, IEEE, Turin, Italy, April 2013.

[13] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Achieving k-anonymity in privacy-aware location-based services," in *Proceedings of*

the 33rd IEEE Conference on Computer Communications, IEEE INFOCOM 2014, pp. 754–762, Toronto, Canada, May 2014.

[14] A. Khoshgozaran, C. Shahabi, and H. Shirani-Mehr, “Location privacy: Going beyond K-anonymity, cloaking and anonymizers,” *Knowledge and Information Systems*, vol. 3, no. 26, pp. 435–465, 2011.

[15] A. Pingley, W. Yu, N. Zhang, X. Fu, and W. Zhao, “CAP: A context-aware privacy protection system for location-based services,” in *Proceedings of the 29th IEEE International Conference on Distributed Computing Systems Workshops (ICDCS ’09)*, pp. 49–57, Montreal, Canada, June 2009.

[16] K. G. Shin, X. Ju, Z. Chen, and X. Hu, “Privacy protection for users of location-based services,” *IEEE Wireless Communications Magazine*, vol. 19, no. 1, pp. 30–39, 2012.

[17] C. Dwork, “Differential privacy,” in *Proceedings of the International Colloquium on Automata, languages and Programming*, vol. 4052, Springer, Berlin, Germany, 2006.

[18] N. Vratonjic, K. Huguenin, V. Bindschaedler, and J.-P. Hubaux, “How others compromise your location privacy: The case of shared public IPs at hotspots,” in *Proceedings of the Privacy Enhancing Technologies Symposium – PETS*, vol. 7981, pp. 123–142, 2013.

[19] J. Freudiger, R. Shokri, and J. Hubaux, “Evaluating the privacy risk of location-based services,” in *Proceedings of the Financial Cryptography and Data Security*, pp. 31–46, 2012.

[20] A. Machanavajjhala, J. Gehrke, D. Kifer, and M. Venkitasubramaniam, “L-diversity: privacy beyond k-anonymity,” in *Proceedings of the 22nd International Conference on Data Engineering (ICDE ’06)*, pp. 1–12, Atlanta, Ga, USA, April 2006.

[21] N. Li, T. Li, and S. Venkatasubramanian, “t-closeness: privacy beyond k-anonymity and *l*-diversity,” in *Proceedings of the 23rd International Conference on Data Engineering*, pp. 1–12, Istanbul, Turkey, April 2007.

[22] J. Meyerowitz and R. Roy Choudhury, “Hiding stars with fireworks: location privacy through camouflage,” in *Proceedings of the 15th Annual ACM International Conference on Mobile Computing and Networking (MobiCom ’09)*, pp. 345–356, Beijing, China, September 2009.

[23] X. Zhang, X. Gui, and F. Tian, “A Framework for measuring query in Location-based Service,” *KSII Transactions on Internet and Information Systems*, vol. 5, no. 9, pp. 1717–1732, 2015.

[24] R. Shokri, G. Theodorakopoulos, J. Boudec, and J. Hubaux, “Quantifying location privacy,” in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 247–262, Oakland, CA, USA, 2011.

[25] X. Zhang, X. Gui, F. Tian, S. Yu, and J. An, “Privacy quantification model based on the Bayes conditional risk in Location-Based Services,” *Tsinghua Science and Technology*, vol. 5, no. 19, pp. 452–462, 2014.

[26] Y. H. Xiao and X. Li, “Protecting location with dynamic differential privacy under temporal correlations,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, pp. 1298–1309, 2015.

[27] K. Dong, T. Guo, H. Ye, X. Li, and Z. Ling, “On the limitations of existing notions of location privacy,” *Future Generation Computer Systems*, vol. 86, pp. 1513–1522, 2018.

[28] G. Ghinita, K. Zhao, D. Papadias, and P. Kalnis, “A reciprocal framework for spatial K-anonymity,” *Information Systems*, vol. 3, no. 35, pp. 299–314, 2010.

[29] C.-Y. Chow, M. F. Mokbel, and X. Liu, “A peer-to-peer spatial cloaking algorithm for anonymous location-based service,” in *Proceedings of the 14th Annual ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS ’06)*, pp. 171–178, New York, NY, USA, November 2006.

[30] G. Ghinita, P. Kalnis, and S. Skiadopoulos, “MOBIHIDE: A mobilea peer-to-peer system for anonymous location-based queries,” in *International Symposium on Spatial and Temporal Databases*, vol. 4605, pp. 221–238, Boston, MA, USA, 2007.

[31] J. Manweiler, R. Scudellari, and L. P. Cox, “SMILE: Encounter-based trust for mobile social services,” in *Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS’09*, pp. 246–255, November 2009.

[32] G. Ghinita, P. Kalnis, and S. Skiadopoulos, “PRIVE: anonymous location-based queries in distributed mobile systems,” in *Proceedings of the 16th International World Wide Web Conference (WWW ’07)*, pp. 371–380, May 2007.

[33] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, “Enhancing privacy through caching in location-based services,” in *Proceedings of the 34th IEEE Annual Conference on Computer Communications (IEEE INFOCOM ’15)*, pp. 754–762, IEEE, May 2015.

[34] V. Primault, S. B. Mokhtar, C. Lauradoux, and L. Brunie, “Differentially Private Location Privacy in Practice,” 2015, https://arxiv.org/abs/1410.7744?context=cs.

[35] X. H. Chen, A. Mizera, and J. Pang, “Quantifying location privacy revisited: preliminary report,” 2014, http://satoss.uni.lu/.

[36] S. Gambs, M.-O. Killijian, and M. N. del Prado Cortez, “Show me how you move and I will tell you who you are,” *Transactions on Data Privacy*, vol. 4, no. 2, pp. 103–126, 2011.

[37] K. Jiang, D. Shao, S. Bressan, T. Kister, and K. Tan, “Publishing trajectories with differential privacy guarantees,” in *Proceedings of the 25th International Conference on Scientific and Statistical Database Management*, pp. 1–12, July 2013.

[38] C. Faloutsos and S. Roseman, “Fractals for secondary key retrieval,” in *Proceedings of the ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, pp. 247–252, New York, NY, USA, 1989.

[39] F. Tian, X. Gui, J. An, P. Yang, and X. Zhang, “A density-based space filling curve for location privacy-preserving,” in *Proceedings of the 2014 IEEE International Conference on Services Computing (SCC)*, pp. 131–138, Atlanta, GA, USA, June 2014.

[40] A. Blum, K. Ligett, and A. Roth, “A learning theory approach to non-interactive database privacy,” in *Proceedings of the Fortieth Annual ACM Symposium on Theory of Computing (STOC ’08)*, pp. 609–618, ACM, New York, NY, USA, 2008.